

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ
ТОЧНОЙ МЕХАНИКИ И ОПТИКИ (ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ)**

**НАУЧНО-ТЕХНИЧЕСКИЙ
ВЕСТНИК**

Выпуск 10

**ИНФОРМАЦИЯ И УПРАВЛЕНИЕ
В ТЕХНИЧЕСКИХ СИСТЕМАХ**



**САНКТ-ПЕТЕРБУРГ
2003**

Выпуск содержит материалы XXXII научной и учебно-методической конференции СПбГИТМО (ТУ), посвященной 300-летию Санкт-Петербурга

Конференция организуется и проводится 4–7 февраля 2003 г. Санкт-Петербургским государственным институтом точной механики и оптики (техническим университетом) в сотрудничестве с

ВНЦ ГОИ им. С.И. Вавилова,
Институтом аналитического приборостроения РАН,
Институтом проблем машиноведения РАН,
Комитетом по науке и высшей школе Администрации Санкт-Петербурга
ВНИИМ им. Д.И. Менделеева,
ОАО "ЛОМО",
ОАО "Техприбор",
ОАО "Электроавтоматика",
ЦНИИ "Электроприбор".

Программный комитет конференции:

Васильев В.Н. (СПбГИТМО (ТУ)) – председатель

Аронов А.М. (ОАО ЛОМО)	Мирошник И.В. (СПбГИТМО (ТУ))
Викторов А.Д. (Администрация Санкт-Петербурга)	Мусалимов В.М. (СПбГИТМО (ТУ))
Гатчин Ю.А. (СПбГИТМО (ТУ))	Парамонов П.П. (ОАО "Электроавтоматика")
Гуров И.П. (СПбГИТМО (ТУ))	Пешехонов В.Г. (ЦНИИ "Электроприбор")
Индейцев Д.А. (ИПМаш РАН)	Путилин Э.С. (СПбГИТМО (ТУ))
Карасев В.Б. (ВНЦ ГОИ им. С.И. Вавилова)	Ханов Н.И. (ВНИИМ им. Д.И. Менделеева)
Козлов С.А. (СПбГИТМО (ТУ))	Храмов В.Ю. (СПбГИТМО (ТУ))
Колесников Ю.Л. (СПбГИТМО (ТУ))	Шехонин А.А. (СПбГИТМО (ТУ))
Курочкин В.Е. (ИанП РАН)	Яковлев Е.Б. (СПбГИТМО (ТУ))
Маслов Ю.В. (ОАО "Техприбор")	

Организационный комитет конференции:

Никифоров В.О. – председатель
Студеникин Л.М. – зам. председателя
Казар Л.Н. – ученый секретарь

Беспалов В.Г.	Ленский А.В.
Волков Д.П.	Метляков А.П.
Горкина Н.М.	Ноздрин М.А.
Гусарова Н.Ф.	Подлесных В.И.
Демин А.В.	Прудентова Т.А.
Коняхин И.А.	Савельева Л.П.

ISBN 5-7577-0139-0

© Санкт-Петербургский государственный институт
точной механики и оптики (технический университет),
2003

ПРЕДИСЛОВИЕ

Наступивший XXI век можно без преувеличения назвать веком информации. К основным его характеристикам можно отнести активное использование информационных технологий, углубление глобализации экономических и социальных процессов и усиление в них роли управления.

В очередном, десятом выпуске "Научно-технического вестника СПбГИТМО(ТУ)" вниманию читателей предлагаются работы по важнейшим направлениям научных исследований в области вычислительной техники, систем управления, автоматизации проектирования и программирования.

В разделе "Управление и автоматика в технических системах" представлены результаты теоретических и экспериментальных исследований систем управления, а также их моделирования. В разделе также затронуты вопросы организации учебного процесса по соответствующей специальности.

Работы, представленные в разделе "Вычислительная техника и программирование", отражают исследования по таким актуальным проблемам, как сетевые технологии, создание интегрированных информационных сред, методы тестирования программного обеспечения.

Развитие научных исследований в области высоких технологий неразрывно связано с автоматизацией проектирования и изучением новых свойств элементов микроэлектроники, отличающихся более высокими характеристиками.

Исследованию полупроводниковых структур на основе кремния посвящено большинство работ в разделе "Автоматизация проектирования, технологии элементов и узлов компьютерных систем". Научная новизна этих работ состоит в развитии методов анализа структурных дефектов в полупроводниках, которые позволили получить целый ряд новых результатов, важных, в частности, для развития технологий микросхемотехники. В нескольких работах рассмотрены проблемы надежности упругих элементов систем управления.

В заключительный раздел "Электротехника, электромеханика и электротехнологии" включены работы сотрудников университета в области электропривода и широтно-импульсных преобразователей. Полученные результаты позволяют расширить знания об электромагнитных процессах в силовых цепях.

Статьи сборника отражают результаты научной и учебно-методической деятельности преподавателей и аспирантов СПб ГИТМО (ТУ), находящие применение в учебном процессе и в практике работы реальных предприятий.

*Ю.А. Гатчин,
профессор, зав. кафедрой
проектирования компьютерных систем*

АЛГОРИТМ КОМПЕНСАЦИИ ВНЕШНИХ
ДЕТЕРМИНИРОВАННЫХ ВОЗМУЩЕНИЙ: ОПЕРАТОРНЫЙ
МЕТОД СИНТЕЗА

Г.В. Лукьянова, В.О. Никифоров

Предложен регулятор выходной переменной линейного объекта, обеспечивающий стабилизацию объекта управления с полной компенсацией внешних детерминированных возмущений выделенного класса.

Введение

Задача компенсации внешних детерминированных возмущений имеет большое значение для теории и практики систем автоматического управления. Один из подходов к ее решению состоит в использовании принципа внутренней модели [1–4]. В соответствии с данным принципом, внешнее детерминированное возмущение рассматривается в качестве выхода автономной системы (так называемого *генератора возмущений*), возбуждаемого ненулевыми начальными условиями. Для полной компенсации внешнего возмущения модель его генератора должна быть соответствующим образом учтена (воспроизведена) в структуре регулятора.

В настоящее время принцип внутренней модели является хорошо разработанным для широких классов линейных [1–6] и нелинейных систем [7–9], систем с неизвестными параметрами [10–12]. Также он нашел свое применение в задачах управления реальными техническими объектами [13–16].

Однако в подавляющем большинстве публикаций принцип внутренней модели формулируется в терминах пространства состояний. С практической точки зрения это означает построение регуляторов состояния или применение в синтезированных системах специальных "расширенных" наблюдателей [5,16].

В настоящей работе представлен новый метод синтеза регулятора выходной переменной с компенсацией внешних детерминированных возмущений. Метод основан на решении специального полиномиального уравнения и предусматривает использование операторной формы записи математической модели объекта и генератора возмущения.

Постановка задачи

Рассматривается линейный динамический объект

$$\alpha(p)\bar{y}(t) = k\beta(p)u(t), \quad (1.1)$$

с регулируемой переменной

$$y = \bar{y}(t) + f(t), \quad (1.2)$$

где $\bar{y}(t)$ – неизмеряемый выход объекта; $p = d/dt$ – оператор дифференцирования; $\alpha(p) = p^n + \alpha_{n-1}p^{n-1} + \alpha_{n-2}p^{n-2} + \dots + \alpha_0$ и $\beta(p) = p^m + b_{m-1}p^{m-1} + b_{m-2}p^{m-2} + \dots + b_0$ – известные нормированные полиномы; k – "высокочастотный" коэффициент усиления; $u(t)$ – сигнал управления; $f(t)$ – внешнее неизмеряемое возмущение. Рассматриваемая задача состоит в синтезе управления, обеспечивающего стабилизацию объекта (1.1) и нулевое значение регулируемой переменной в установившемся режиме, т.е. выполнение условия

$$\lim_{t \rightarrow \infty} y(t) = 0. \quad (1.3)$$

При решении сформулированной задачи будем полагать выполненными следующие гипотезы.

Гипотеза 1: полином $\beta(p)$ является гурвицевым.

Гипотеза 2: ограниченное детерминированное возмущение $f(t)$ имеет изображение Лапласа

$$F(s) = L\{f(t)\} = \frac{\psi(s)}{\varphi(s)}$$

с известным полиномом $\varphi(s)$; при этом взаимонеприводимые полиномы $\varphi(s)$ и $\psi(s)$ комплексной переменной s имеют степени μ и $\gamma \leq \mu$, соответственно.

Вспомогательный результат

Синтез искомого управления основан на решении полиномиального уравнения специального вида. Поэтому приведем сначала вспомогательный математический результат, устанавливающий достаточные условия существования и единственности решения. Предлагаемое утверждение отличается от приведенного в работе [17], а его доказательство содержит конструктивный метод решения полиномиального уравнения рассматриваемого класса.

Лемма. Пусть заданы произвольные нормированные полиномы

$$\delta(p) = p^M + \delta_{M-1}p^{M-1} + \dots + \delta_0.$$

$$A(p) = p^N + a_{N-1}p^{N-1} + \dots + a_0,$$

где $M > N$. Тогда существуют единственные полиномы

$$R(p) = r_{N-1}p^{N-1} + r_{N-2}p^{N-2} + \dots + r_0.$$

$$D(p) = p^{M-N} + d_{M-N-1}p^{M-N-2} + \dots + d_0,$$

удовлетворяющие уравнению

$$\delta(p) = A(p)D(p) + R(p). \quad (2.1)$$

Доказательство леммы является конструктивным и предлагает алгоритм расчета коэффициентов искомого полинома. Рассмотрим правую часть уравнения (2.1)

$$\begin{aligned} A(p)D(p) + R(p) &= (p^N + a_{N-1}p^{N-1} + \dots + a_0)(p^{M-N} + d_{M-N-1}p^{M-N-1} + \dots + d_0) + \\ &+ r_{N-1}p^{N-1} + r_{N-2}p^{N-2} + \dots + r_0 = p^M + (a_{N-1} + d_{M-N-1})p^{M-1} + \\ &+ (a_{N-2} + a_{N-1}d_{M-N-1} + d_{M-N-2})p^{M-2} + (a_{N-3} + a_{N-2}d_{M-N-1} + a_{N-1}d_{M-N-2} + d_{M-N-3})p^{M-3} + \\ &+ \dots + (a_{2N-M} + a_{2N-M+1}d_{M-N-1} + \dots + d_0)p^N + (\mathcal{G}_{N-1}(a_i, d_j) + r_{N-1})p^{N-1} + \dots + \mathcal{G}_0(a_i, d_j) + r_0, \end{aligned}$$

где $\mathcal{G}_i(a_i, d_j)$ – постоянные коэффициенты, зависящие от a_i и d_j .

Очевидно, что для справедливости уравнения (2.1) коэффициенты левой и правой частей, стоящие при одинаковых степенях переменной p , должны быть равны, т.е.

$$\begin{cases} \delta_{M-1} = a_{N-1} + d_{M-N-1}, \\ \delta_{M-2} = a_{N-2} + a_{N-1}d_{M-N-1} + d_{M-N-2}, \\ \vdots \\ \delta_N = a_{2N-M} + a_{2N-M+1}d_{M-N-1} + \dots + d_0, \\ \delta_{N-1} = \mathcal{G}_{N-1}(a_i, d_j) + r_{N-1}, \\ \vdots \\ \delta_0 = \mathcal{G}_0(a_i, d_j) + r_0. \end{cases} \quad (2.2)$$

Разрешая уравнение (2.2) относительно коэффициентов d_i и r_j , находим:

$$\begin{cases} d_{M-N-1} = \delta_{M-1} - a_{N-1}, \\ d_{M-N-2} = \delta_{M-2} - a_{N-2} - a_{N-1}d_{M-N-1}, \\ \vdots \\ d_0 = \delta_N - a_{2N-M} - a_{2N-M+1}d_{M-N-1} - \dots - a_{2N-M+1}d_1, \\ r_{N-1} = \delta_{N-1} - \mathcal{G}_{N-1}(a_i, d_j), \\ \vdots \\ r_0 = \delta_0 - \mathcal{G}_0(a_i, d_j). \end{cases} \quad (2.3)$$

Таким образом, формулы (2.3) устанавливают алгоритм последовательного расчета коэффициентов искомого полинома, чем подтверждается существование единственного решения полиномиального уравнения (2.1). Лемма доказана.

Основной результат

Сформируем искомое управление в виде

$$u(t) = -\frac{1}{k} \frac{R(p)}{\beta(p)D(p)\varphi(p)} y, \quad (3.1)$$

где полиномы

$$R(p) = r_{n+\mu-1}p^{n+\mu-1} + r_{n+\mu-2}p^{n+\mu-2} + \dots + r_0, \quad (3.2)$$

$$D(p) = p^{n-m-1} + d_{n-m-2}p^{n-m-2} + \dots + d_0 \quad (3.3)$$

являются решением уравнения

$$\delta(p) = \alpha(p)\varphi(p)D(p) + R(p) \quad (3.4)$$

с произвольным гурвицевым полиномом

$$\delta(p) = p^{2n+\mu-m-1} + \delta_{2n+\mu-m-2}p^{2n+\mu-m-2} + \dots + \delta_0. \quad (3.5)$$

С учетом подстановок $A(p) = \alpha(p)\varphi(p)$, $N = n + \mu$, $M = 2n + \mu - m - 1$ из леммы следует существование единственных полиномов (3.2) и (3.3), удовлетворяющих равенству (3.4).

Для исследования свойств замкнутой системы (см. рис. 1) запишем ее уравнение

$$\begin{cases} y(t) = \bar{y}(t) + f(t), \\ \alpha(p)\bar{y}(t) = k\beta(p)u(t), \\ k\beta(p)D(p)\varphi(p)u(t) = -R(p)y(t). \end{cases} \quad (3.6)$$

Вычисляя преобразование Лапласа системы (3.6), получим:

$$\begin{cases} Y(s) = \bar{Y}(s) + \frac{\psi(s)}{\varphi(s)}, \\ \alpha(s)\bar{Y}(s) + Q(s) = k\beta(s)U(s), \\ k\beta(s)D(s)\varphi(s)U(s) = -R(s)Y(s) - B(s), \end{cases} \quad (3.7)$$

где $Y(s) = L\{y(t)\}$, $U(s) = L\{u(t)\}$, $\bar{Y}(s) = L\{\bar{y}(t)\}$, а $Q(s)$ и $B(s)$ – полиномы, зависящие от начальных условий.

Из третьего уравнения системы (3.7) найдем управление $U(s)$:

$$U(s) = -\frac{1}{k} \frac{R(s)}{\beta(s)D(s)\varphi(s)} Y(s) - \frac{1}{k} \frac{B(s)}{\beta(s)D(s)\varphi(s)}.$$

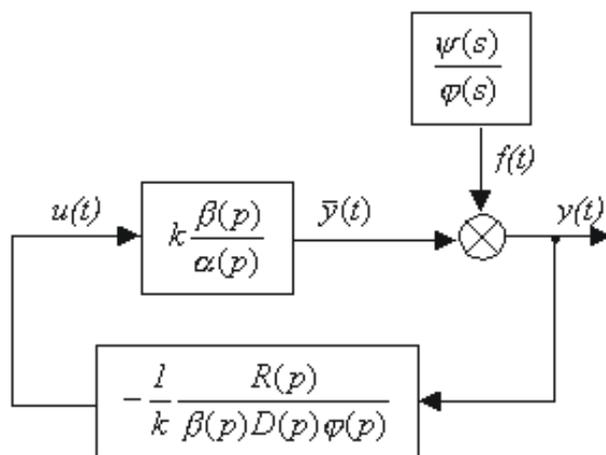


Рис. 1. Структурная схема замкнутой системы (1.1)

Подставим найденное управление во второе уравнение и выразим образ Лапласа выходной переменной динамического объекта:

$$\bar{Y}(s) = -\frac{R(s)}{D(s)\varphi(s)\alpha(s)}Y(s) - \frac{B(s)}{D(s)\varphi(s)\alpha(s)} - \frac{Q(s)}{\alpha(s)}.$$

Тогда образ Лапласа регулируемой переменной замкнутой системы будет иметь вид

$$Y(s) = -\frac{R(s)}{D(s)\varphi(s)\alpha(s)}Y(s) - \frac{B(s)}{D(s)\varphi(s)\alpha(s)} - \frac{Q(s)}{\alpha(s)} + \frac{\psi(s)}{\varphi(s)}.$$

После элементарных преобразований окончательно получаем:

$$Y(s) = \frac{D(s)\alpha(s)\psi(s) - B(s) - D(s)\varphi(s)Q(s)}{\delta(s)}. \quad (3.8)$$

Тогда для выходной переменной замкнутой системы будем иметь

$$y(t) = \sigma(t), \quad (3.9)$$

где функция

$$\sigma(t) = L^{-1} \left\{ \frac{D(s)\alpha(s)\psi(s) - B(s) - D(s)\varphi(s)Q(s)}{\delta(s)} \right\} \quad (3.10)$$

экспоненциально затухает в силу гурвицевости полинома $\delta(s)$.

Таким образом, из уравнения (3.10) следует асимптотическая устойчивость замкнутой системы (в силу гурвицевости характеристического полинома $\delta(s)$), а из уравнения (3.10) – выполнение целевого условия (1.3). Следовательно, предложенный регулятор обеспечивает стабилизацию объекта управления с полной компенсацией внешних детерминированных возмущений выделенного класса.

Работа выполнена при финансовой поддержке в рамках тематического плана НИР СПбГИТМО (ТУ).

Литература

1. Johnson C.D. Accommodation of external disturbances in linear regulator and servomechanism problems // IEEE Transactions on Automatic Control. 1971. V. 16, № 6.
2. Francis B.A., Wonham W.M. The internal model principle for linear multivariable regulators // Applied Mathematics and Optimization. 1975, № 2.
3. Davison E.J. The robust control of a servomechanism problem for linear time-invariant multivariable systems // IEEE Transactions on Automatic Control. 1976. V. 21. № 1.

4. Уонем М. Линейные многомерные системы управления: Геометрический подход. М.: Наука, 1980.
5. Синтез дискретных регуляторов при помощи ЭВМ / В.В. Григорьев, В.Н. Дроздов, В.В. Лаврентьев, А.В. Ушаков Л.: Машиностроение, 1983.
6. Акунов Т.А., Алишеров С., Оморов Р.О., Ушаков А.В. Матричные уравнения в задачах управления и наблюдения непрерывных объектов. Бишкек: Илим, 1991.
7. Di Benedetto M.D. Synthesis of an internal model for nonlinear output regulation // International Journal of Control. 1987. V. 45.
8. Khalil H.K. Robust servomechanism output feedback controller for feedback linearizable systems // Automatica. 1994. V. 30. № 10.
9. Никифоров В.О. Нелинейная система управления с компенсацией внешних детерминированных возмущений // Известия РАН. Теория и системы управления. 1997. № 4.
10. Никифоров В.О. Адаптивная стабилизация линейного объекта, подверженного внешним детерминированным возмущениям // Известия РАН. Теория и системы управления. 1997. № 2.
11. Nikiforov V.O. Adaptive non-linear tracking with complete compensation of unknown disturbances // European Journal of Control. 1998. V. 4. № 2.
12. Nikiforov V.O. Nonlinear servocompensation of unknown external disturbances // Automatica. 2001. V. 37.
13. Buchner H.J., Hemami H. Servocompensation of disturbance in robotic systems // International Journal of Control. 1988. V. 48.
14. Никифоров В.О., Дроздов В.Н. Адаптивное управление мехатронным поворотным столом // Мехатроника, автоматизация и управление. 2002. Часть I – № 4. Часть II – № 5.
15. Дроздов В.Н., Мирошник И.В., Скорубский И.В. Системы автоматического управления с микроЭВМ. Ленинград: Машиностроение, 1989.
16. Андриевский Б.Р., Фрадков А.Л. Избранные главы теории автоматического управления. Санкт-Петербург: Наука, 1999.
17. Острем К., Виттенмарк Б. Системы управления с ЭВМ. М.: Мир, 1987.

РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ ИНЖЕКТОРНОГО ДВИГАТЕЛЯ

С.А. Мигуш, Д.Н. Герасимов, В.О. Никифоров

Введение

На сегодняшний день в технически развитых странах инжекторные двигатели (т.е. двигатели, где подача топлива в цилиндр осуществляется путем принудительного впрыскивания) практически полностью вытеснили классическую карбюраторную схему питания [1–5]. К основным преимуществами инжекторных двигателей (ИД) можно отнести:

- 1) возможность электронного управления двигателем;
- 2) возможность оптимизации процесса смесеобразования и, как следствие:
 - снижение расхода топлива;
 - снижение токсичности выхлопных газов;
 - повышение КПД двигателя;
 - снижение износа цилиндров;
- 3) возможность использования топлива с повышенным октановым числом.

Однако, несмотря на то, что практически все крупные автопроизводители предлагают широкую номенклатуру автомобилей с инжекторными двигателями, работы по математическому моделированию и аналитическому синтезу систем управления ИД находятся на начальной стадии развития. В большинстве случаев проектирование, калибровка и отладка новых образцов систем управления подачей топлива в инжекторные двигатели происходит эмпирическим путем (методом "проб и ошибок") и основывается на богатом производственном и инженерном опыте конструкторов. Микропроцессорные системы управления двигателями, как правило, основаны не на принципах замкнутого управления по отклонению, а на использовании встроенных таблиц данных, сформированных экспериментальным путем на этапе калибровки двигателей [2, 3]. Поэтому проблема получения математических моделей инжекторных двигателей, ориентированных на дальнейшее использование в задачах аналитического синтеза систем управления, является актуальной и представляет несомненный интерес.

Математические модели двигателей внутреннего сгорания, представленные в литературе 70-х, 80-х годов XX столетия, как правило, являются линеаризованными и сохраняют свою справедливость только при малых отклонениях от установившихся режимов работы двигателя [4,5]. Естественно, что такие модели были ориентированы на разработку упрощенных, линейных законов управления. Современный уровень развития методов нелинейного, адаптивного и робастного управления позволяет проводить синтез законов управления с учетом нелинейных характеристик объекта управления, что мотивирует большой объем работ по созданию более точных, нелинейных моделей инжекторных двигателей [6, 9].

В настоящей статье авторы приводят схему инжекторного двигателя; анализируют двигатель как объект управления и формулируют основные задачи управления двигателем; приводят проблемно-ориентированную модель двигателя, описывающую последовательность преобразования входных сигналов (угла поворота дроссельной заслонки, количества впрыснутого топлива и угла опережения зажигания) в регулируемые переменные (момент двигателя и скорость вращения коленчатого вала).

Схема инжекторного двигателя

Схема инжекторного двигателя представлена на рис. 1. Цифрами на схеме обозначены: 1 – акселератор (педаль газа); 2 – дроссельная заслонка; 3 – впускной коллектор; 4 – инжектор; 5 – впускной клапан; 6 – свеча зажигания; 7 – выпускной клапан; 8 – цилиндр; 9 – поршень; 10 – коленчатый вал; 11 – выпускной коллектор; 12 – каталитический конвертер; 13 – датчик количества воздуха, проходящего через заслонку; 14 – датчик положения дроссельной заслонки; 15 – датчик кислорода; 16 – датчик скорости; 17 – датчик давления во впускном коллекторе.

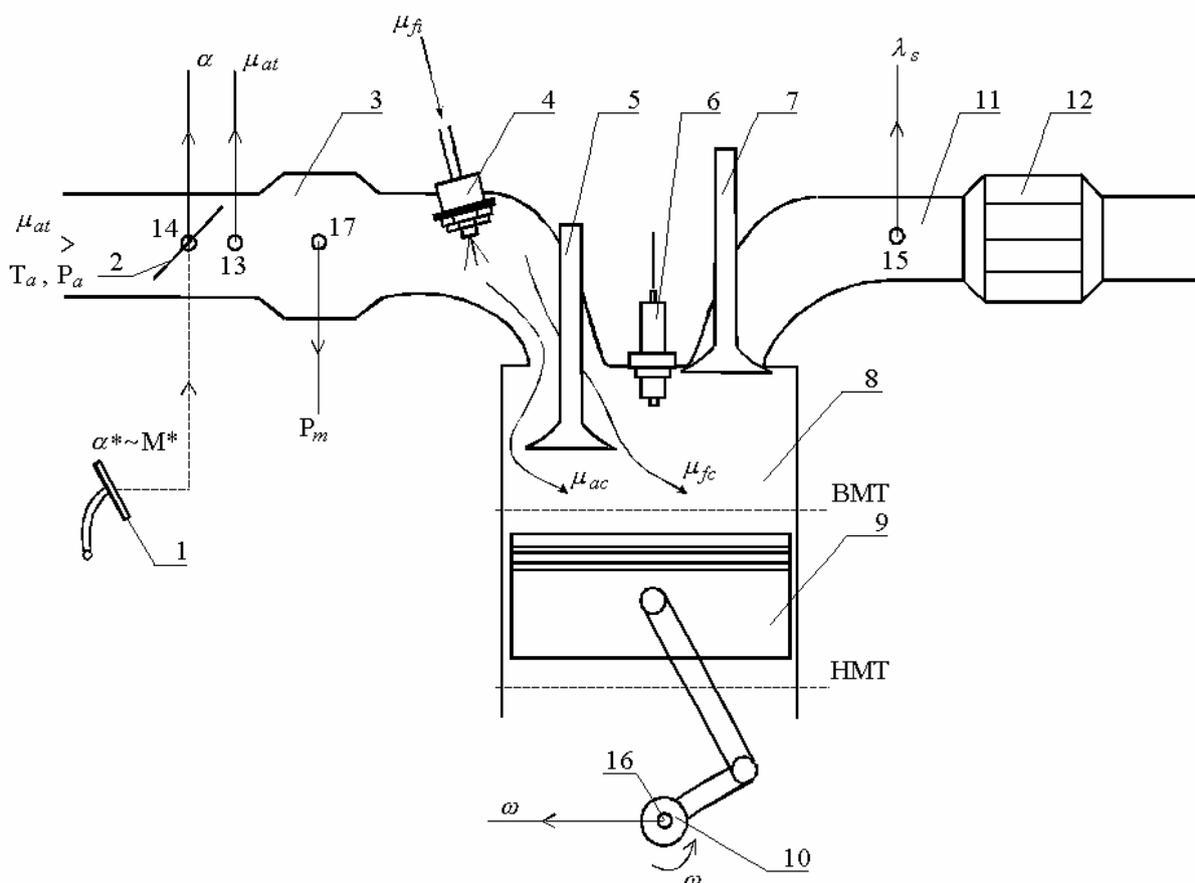


Рис.1. Схема инжекторного двигателя.

Двигатель функционирует следующим образом. Через дроссельную заслонку 2 воздух попадает во впускной коллектор 3. Позади дроссельной заслонки находится датчик 13, измеряющий поток воздуха, проходящий через заслонку в единицу времени. Датчик положения дроссельной заслонки 14 измеряет угол ее поворота α . Во впускном коллекторе находится датчик давления воздуха 17. Рядом со впускным клапаном 5 находится форсунка 4, впрыскивающая топливо в цилиндр 8. Воздух и топливо попадают в цилиндр через впускной клапан во время такта впуска и смешиваются. С помощью свечи зажигания 6 в цилиндре осуществляется воспламенение рабочей смеси. После сгорания топлива продукты сгорания поступают из цилиндра через выпускной клапан в выпускной коллектор 11, где находится датчик кислорода 15, информация с которого позволяет рассчитывать соотношение воздух / топливо в цилиндре. Датчик скорости 16 служит для измерения скорости вращения коленчатого вала. Каталитический конвертер 12 содержит катализатор, ускоряющий химическую реакцию разложения окиси углерода и углеводородов в выхлопных газах.

Для получения максимального крутящего момента необходимо постоянное поддержание оптимального соотношения воздух / топливо в цилиндре и соответствующего данному соотношению угла опережения зажигания. При отклонениях любого из вышеуказанных параметров от точки оптимума происходят потери мощности двигателя, вызванные неполным или несвоевременным сгоранием рабочей смеси. Поэтому важнейшими задачами при управлении инжекторным двигателем является поддержка оптимального соотношения воздух / топливо в цилиндре и оперативная корректировка угла опережения зажигания.

Инжекторный двигатель как объекта управления

Проанализируем теперь инжекторный двигатель как объект управления, выделив входные и выходные (регулируемые) переменные, а также сформулировав основные задачи управления. Для этого рассмотрим процесс управления автомобилем с участием человека – водителя.

При управлении автомобилем водитель изменяет положение акселератора для того, чтобы изменять скорость транспортного средства. Поэтому целью управления замкнутой системы "водитель–автомобиль" является стабилизация желаемой скорости транспортного средства W^* . Однако данный процесс не является непосредственным, так как изменение положения акселератора вызывает сначала изменение момента, развиваемого двигателем. В свою очередь, разница между этим моментом и моментом внешней нагрузки определяет ускорение и, в итоге, скорость автомобиля. Поэтому можно считать, что водитель является антропогенным регулятором, который на основании разницы между желаемой W^* и реальной W скоростью определяет требуемый момент двигателя M^* . Таким образом, положение акселератора может интерпретироваться как желаемая величина момента двигателя M^* . Исходя из этого, в системе управления двигателем (рис. 2) можно выделить следующие контуры обратной связи:

1) контур с обратной связью по моменту (главная обратная связь); для данного контура целью управления является стабилизация желаемой величины момента двигателя M^* (или поддержания желаемого закона изменения момента $M^*(t)$), определяемого положением акселератора; сигналом управления в этом контуре является угол поворота дроссельной заслонки $u_2 = \alpha$;

2) контур с обратной связью по соотношению воздух / топливо (подчиненная обратная связь); для данного контура целью управления является стабилизация желаемой величины соотношения воздух/топливо. Для бензина оптимальным является так называемое *стехиометрическое* соотношение, равное 14.7. Это означает, что для полного сгорания одного килограмма бензина необходимо 14.7 кг воздуха. Для целей оптимального управления двигателем вводится коэффициент λ , показывающий отношение масс впрыснутого в цилиндр топлива к массе сгоревшего в ходе рабочего цикла. Желаемое значение этого коэффициента $\lambda^* = 1$. Сигнал управления для данного контура – количество впрыскиваемого топлива $u_1 = \mu_{fi}$;

3) контур с обратной связью по углу опережения зажигания (оптимизирующая обратная связь); для данного контура целью управления является оптимизация некоторых характеристик работы двигателя; сигналом управления в данном случае является угол опережения зажигания $u_3 = \alpha_{ign}$.

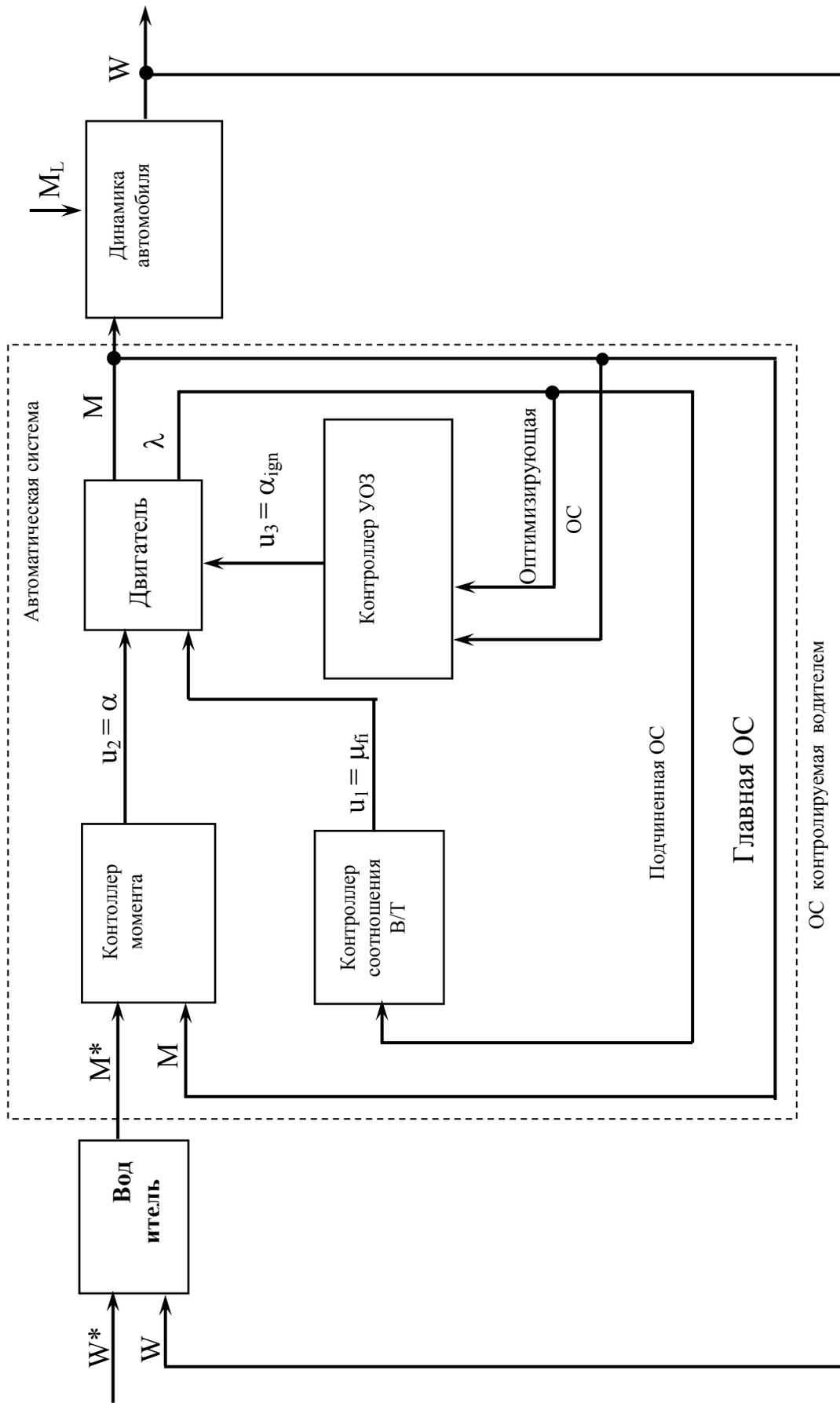


Рис.2. Схема системы автоматического управления инжекторным двигателем

Вывод уравнений модели инжекторного двигателя

Усредненная математическая модель ИД описывает последовательность преобразования входных сигналов (угла поворота дроссельной заслонки α , количества впрыскиваемого топлива μ_{fi} и угла опережения зажигания) в средние значения регулируемых переменных (соотношение воздух/топливо λ и крутящий момент M) без учета пульсаций данных величин внутри каждого цикла работы двигателя. Модель состоит из следующих подсистем.

1. Поток воздуха через дроссельную заслонку. Уравнения данной подсистемы описывают связь потока воздуха через заслонку μ_{at} [кг/с] с углом поворота заслонки α [рад] и давлением во впускном коллекторе P_m [Па]. Это соотношение может быть представлено в форме

$$\mu_{at}(\alpha, P_m) = \eta_t \varphi_1(P_m) \varphi_2(\alpha), \quad 0 \leq \eta_t \leq 1, \quad (1)$$

где η_t – коэффициент эффективности заполнения впускного коллектора, [%]; $\varphi_1(P_m)$ – нелинейная функция, характеризующая поток воздуха, проходящего через единицу площади дроссельной заслонки; $\varphi_2(\alpha)$ – площадь заслонки. Типовой вид функций $\varphi_1(P_m)$ и $\varphi_2(\alpha)$ приведен на рис. 3 и 4.

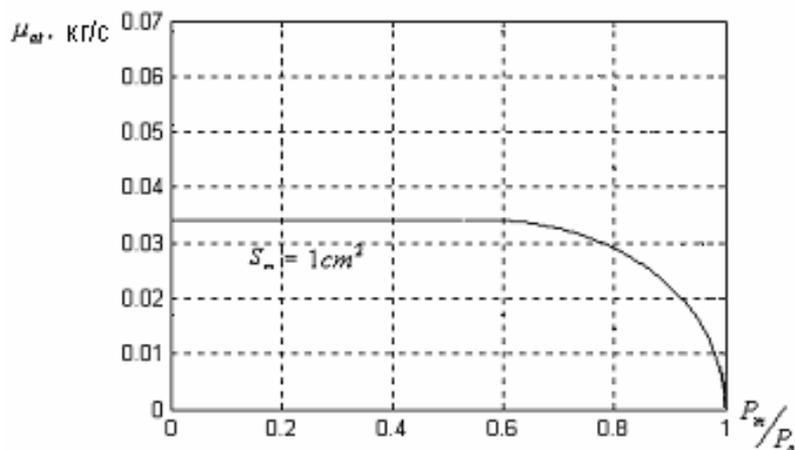


Рис.3. Зависимость потока воздуха через дроссельную заслонку от соотношения атмосферного давления и давления во впускном коллекторе

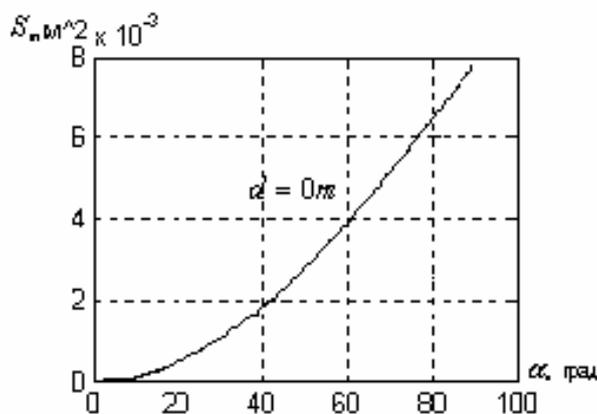


Рис.4. Зависимость площади сечения впускного коллектора от угла поворота дроссельной заслонки

2. Давление во впускном коллекторе. Уравнение давления во впускном коллекторе получено путем дифференцирования уравнения идеального газа в предположении, что температура воздуха во впускном коллекторе является постоянной величиной:

$$\dot{P}_m + \eta_c \frac{V_d \omega}{4\pi V_m} P_m = \frac{RT_m}{V_m} \mu_{at}(\alpha, P_m), \quad 0 \leq \eta_c \leq 1. \quad (2)$$

В уравнении (2) V_m – объем впускного коллектора, м³; T_m – температура в коллекторе, К; μ_{at} – количество воздуха, проходящего через дроссельную заслонку, кг/с; η_c – коэффициент наполнения цилиндра, %.

3. Процесс осаждения топлива на стенках впускного коллектора. Часть распыляемого топлива осаждается на стенках впускного коллектора, образуя пленку [9, 11]. Эта пленка накапливает и поставляет топливо в цилиндр в результате процесса испарения. Таким образом, в цилиндр попадает топливо из двух источников: впрыснутое форсункой и испаренное со стенок коллектора. Точной аналитической модели, описывающей данный процесс, не существует. Однако экспериментальные данные показывают, что его можно описать следующей моделью:

$$\mu_{fc} = \frac{K}{T_p + 1} \mu_{fi} + (1 - K) \mu_{fi}, \quad (3)$$

где K – коэффициент, показывающий, какая часть распыляемого топлива осаждается в пленку ($0 < K < 1$); μ_{fi} – количество топлива, распыляемого форсункой, кг/с; T – постоянная времени процесса осаждения топлива, с.

4. Соотношение воздух/топливо в цилиндре. При рассмотрении данной подсистемы примем следующее идеализированное допущение.

Допущение 1. Все топливо и весь воздух, попадающие в цилиндр, сжигаются и полностью удаляются из него за один цикл. Таким образом, ни топливо, ни воздух не накапливаются в цилиндре от цикла к циклу.

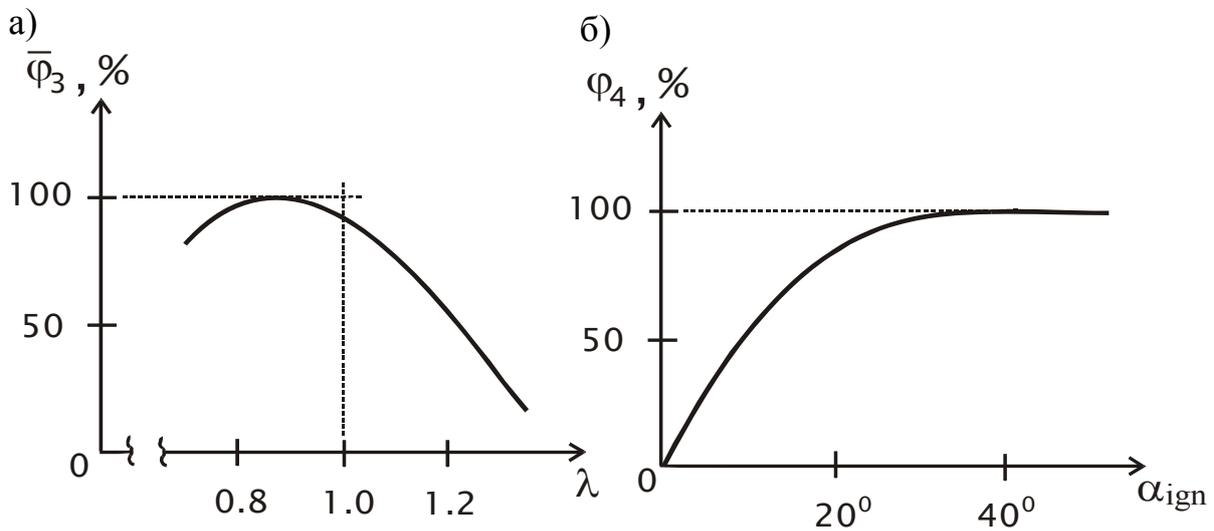


Рис. 5. Типовой вид функций $\bar{\Phi}_3(\lambda)$ и $\Phi_4(\alpha_{igm})$

На основе этого утверждения можно рассчитать соотношение воздух/топливо в виде

$$\lambda = \frac{\mu_{ac}}{\mu_{fc}} = \frac{V_d}{4\pi R T_M F_s} \frac{P_m \omega}{\mu_{ff} + (1 - K) \mu_{fi}} \quad (4)$$

где F_s – стехиометрическое соотношение ($F_s = 14.7$).

5. *Крутящий момент двигателя.* Крутящий момент зависит от количества воздуха в цилиндре, соотношения воздух / топливо и угла опережения зажигания. В первом приближении можно записать следующее выражение для момента:

$$M = c_M \mu_{ac} \bar{\varphi}_3(\lambda) \varphi_4(\alpha_{ign}), \quad (5)$$

где c_M – коэффициент момента, Н·м·с/кг; α_{ign} – угол опережения зажигания, град. Типовой вид функций $\bar{\varphi}_3(\lambda)$, $\varphi_4(\alpha_{ign})$ приведен на рис. 5.

6. *Уравнение движения двигателя.* Механическая цепь двигателя описывается уравнением

$$J\dot{\omega} = M - M_L - M_{\omega}, \quad (6)$$

где J – приведенный момент инерции двигателя, кг/м²; M_L – приведенный момент внешней нагрузки, Н·м; M_{ω} – приведенный момент трения, Н·м.

Модель двигателя в пространстве состояний. На основе представленных выше уравнений запишем модель двигателя в пространстве состояний:

$$\dot{x}_1 = -a_1 x_1 + b_{11} x_1 x_3 \varphi_3(x_1, x_2, x_3, u_1) \varphi_4(u_3) - b_{12} \delta_1(t), \quad (7)$$

$$\dot{x}_2 = -a_2 x_2 + b_2 u_1, \quad (8)$$

$$\dot{x}_3 = -a_3 x_1 x_3 + b_3 \varphi_1(x_3) \varphi_2(u_2), \quad (9)$$

$$y_1 = \frac{1}{x_1 x_3} (c_1 x_2 + d_1 u_1), \quad (10)$$

$$y_2 = c_2 x_1 x_3 \varphi_3(x_1, x_2, x_3, u_1) \varphi_4(u_3), \quad (11)$$

где $x_1 = \omega$ – скорость вращения выходного вала, об/мин; $x_3 = P_m$ – давление во впускном коллекторе, Па; $y_1 = \lambda$ – соотношение воздух/топливо; $y_2 = M$ – крутящий момент, Нм; $\delta_1 = M_L$ – момент внешней нагрузки, Нм; $u_1 = \mu_{fi}$ – количество впрыснутого в цилиндр топлива, кг; $u_2 = \alpha$ – угол поворота дроссельной заслонки, рад; $u_3 = \alpha_{ign}$ – угол опережения зажигания.

Таким образом, математическая модель инжекторного двигателя может быть представлена системой нелинейных дифференциальных уравнений третьего порядка (7)–(8), координатами состояния которой являются: скорость вращения коленчатого вала, масса топливной пленки на стенках впускного коллектора и давление воздуха во впускном коллекторе. Некоторые параметры данной модели не могут быть заранее точно рассчитаны или экспериментально определены (например, масса топлива, образовавшего пленку на стенках впускного коллектора). Поэтому модель инжекторного двигателя является параметрически неопределенной, что мотивирует необходимость использования методов адаптивного или робастного управления.

Дальнейшее уточнение модели должно состоять в определении зависимостей коэффициента наполнения цилиндра и коэффициента эффективности заполнения впускного коллектора от переменных состояния двигателя – давления во впускном коллекторе и скорости вращения коленчатого вала. Особый интерес представляет определение зависимостей основных параметров двигателя от температуры охлаждающей жидкости.

Результаты моделирования

На основе полученных уравнений (7)–(11) в среде пакета Simulink была создана схема моделирования работы двигателя со следующими параметрами: $P_a = 10^5$ Па, $D = 0.035$ м, $d = 0$, $\alpha = 0$, $V_d = 1600$ см³, $V_m = 800$ см³, $T_m = 323$ К, $T_a = 300$ К, $J = 0.14$ кг·м².

Результаты моделирования переходных процессов в инжекторном двигателе при ступенчатом изменении угла поворота дроссельной заслонки и внешнего момента (при $\lambda = 1$ и $\alpha_{ign} = const$) приведены на рис. 6 и демонстрируют хорошее совпадение с экспериментальными данными [9,12,13].

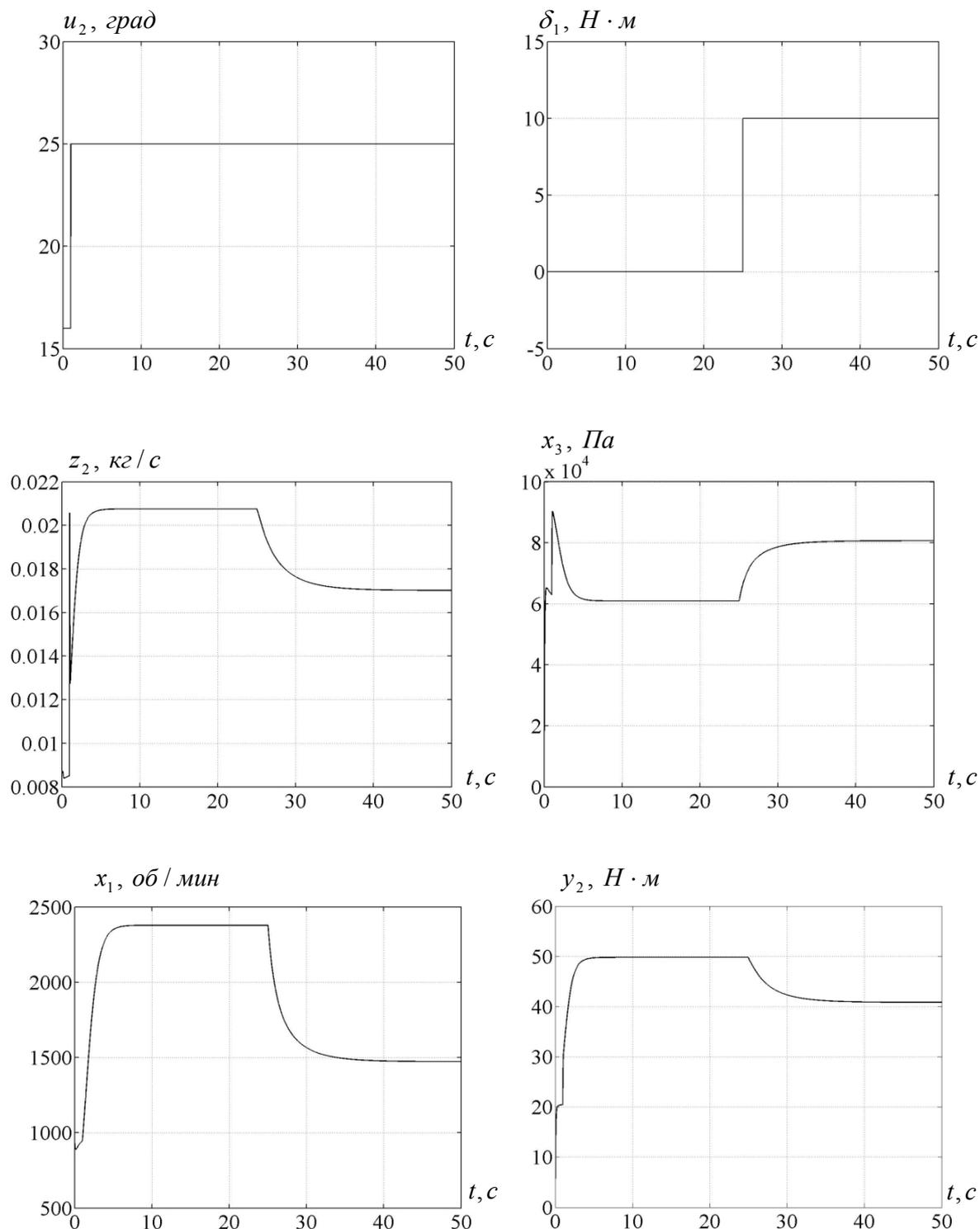


Рис. 6. Изменение давления в коллекторе $x_3 = P_a$, скорости двигателя $x_1 = \omega$, потока воздуха через заслонку z_2 и момента двигателя $y_2 = M$ при ступенчатом изменении угла поворота заслонки u_2 и момента внешней нагрузки $\delta_1 = M_L$

Литература

1. Твег Р. Системы впрыска бензина. Устройство, обслуживание, ремонт: Практ. пособ. М.: Издательство "За рулем", 1999.
2. Пинский Ф.И., Давтян Р.И., Черняк Б.Я. Микропроцессорные системы управления автомобильными двигателями внутреннего сгорания: Учебное пособие. М. "Легион-Автодата", 2001.
3. Грехов Л.В. Топливная аппаратура с электронным управлением дизелей и двигателей с непосредственным впрыском бензина: Учебно-практическое пособие. М.: "Легион-Автодата", 2001.
4. Крутов В.Н. Автоматическое регулирование и управление двигателями внутреннего сгорания: Учебное пособие. М.: Машиностроение, 1989.
5. Богданов С.Н., Буренков М.М., Иванов И.Е. Автомобильные двигатели. М.: Машиностроение, 1987.
6. Ault B.A., Jones V.K., Powell J.D., Franklin G.F. Adaptive air-fuel ration control of a spark ignition engine. // SAE paper. 1993. № 940373.
7. Beaumont A. J., Noble A.D., Scarisbrick A. Adaptive transient air-fuel ratio control to minimize gasoline engine emissions. // FISITA Congress. London, 1992. P. 1–14.
8. Hart M., Ziegler M. Adaptive estimation of cylinder air mass using the combustion pressure. // SAE paper. 1998. № 980791.
9. Kim Y.-W., Rizzoni G., V. Utkin Automotive engine diagnostics and control via nonlinear estimation. // IEEE Control Systems. 1998.
10. Onder C.H., Geering H.P. Model-based multivariable speed and air-to-fuel ration control of an SI engine. // SAE paper. 1993. № 930859. P. 69–80.
11. Simons M.R., Locatelli M., Onder C.H., Geering H.P. A nonlinear wall-wetting model for the complete operating region of a sequential fuel injected SI engine. // SAE paper. 2000. № 1260. P. 1–10.
12. Turin R.C., Geering H.P. Model-based adaptive fuel control in an SI engine. // SAE paper. 1993. № 940374.
13. Turin R.C., Geering H.P. On-line identification of air-to-fuel ratio dynamics in a sequentially injected SI engines. // SAE paper. 1993. No. 930857.

ИЗМЕРИТЕЛЬНАЯ СИСТЕМА ДЛЯ УЧЕБНО - ИССЛЕДОВАТЕЛЬСКОГО КОМПЛЕКСА "МОБИЛЬНЫЕ РОБОТЫ"

К.А. Сергеев, А. В. Котов, Д. Н. Кокшаров

Разработаны датчики внутреннего состояния, предназначенные для контроля скорости и угла поворота колесных модулей. Рассмотрен принцип работы. Приведены результаты обработки текущей информации, приходящих с датчиков.

Введение

Для обучения студентов методам обработки информации и управления транспортным роботом был создан учебно-исследовательский комплекс "Мобильные роботы". Этот комплекс предназначен для моделирования систем управления различными типами транспортных роботов, отладки алгоритмов управления, апробирования алгоритмов управления, управления транспортным роботом в режиме реального времени [1].

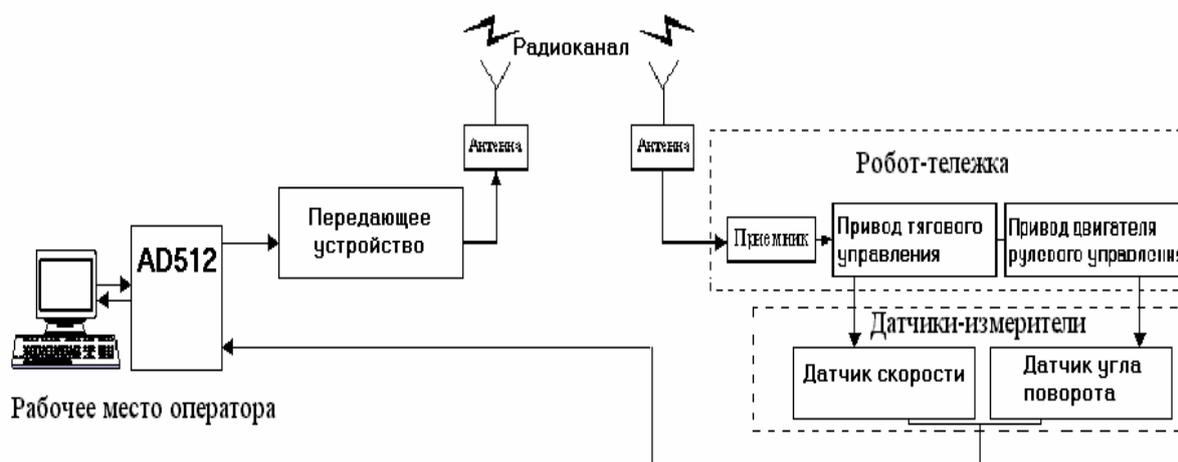


Рис 1. Состав учебно-исследовательского комплекса "Мобильные роботы"

На рис. 1 представлен состав учебно-исследовательского комплекса "Мобильные роботы". Здесь изображена система измерений внутреннего состояния колесного робота, предназначенная для обеспечения возможности его движения в пространстве и использующая датчики угла поворота рулевого привода и скорости вращения заднего привода.

Датчики положения и перемещения находят широкое применение, так как определение положения является одним из основных элементов функционирования робототехнических систем. Существует два базовых способа измерения перемещения [2].

Первый основывается на выработке сигнала, который является функцией положения одной из частей подвижного органа объекта управления, а измерение этого сигнала отражает соответствующее перемещение. В таких датчиках в основном используется зависимость электрического импеданса от положения подвижного элемента.

Второй базируется на выработке единичного импульса на каждом элементарном перемещении. Положение и перемещение определяются подсчетом суммы или разности генерированных импульсов в зависимости от направления перемещения, т.е. такие датчики являются датчиками последовательного приращения.

Выбор реализации и измерительных схем данной системы измерений основывался на простоте реализации и дешевизне технических средств. Поэтому в качестве датчика угла поворота был выбран потенциометрический датчик, который относится к первому типу измерений, а в качестве датчика скорости заднего привода – оптоэлектронный датчик, реализующий второй способ.

Датчик угла поворота рулевого привода

Чтобы понять, какой датчик необходимо выбрать, рассмотрим работу рулевой машинки.

Рулевой привод представляет собой устройство, преобразующее каналный импульс в соответствующее установочное значение, т.е. является серводвигателем [3]. Электрическая часть этого механизма, называемая сервоэлектроникой, состоит из следующих блоков:

- а) генератора опорных импульсов;
- б) каскада сравнения;
- в) усилителя;
- г) серводвигателя.

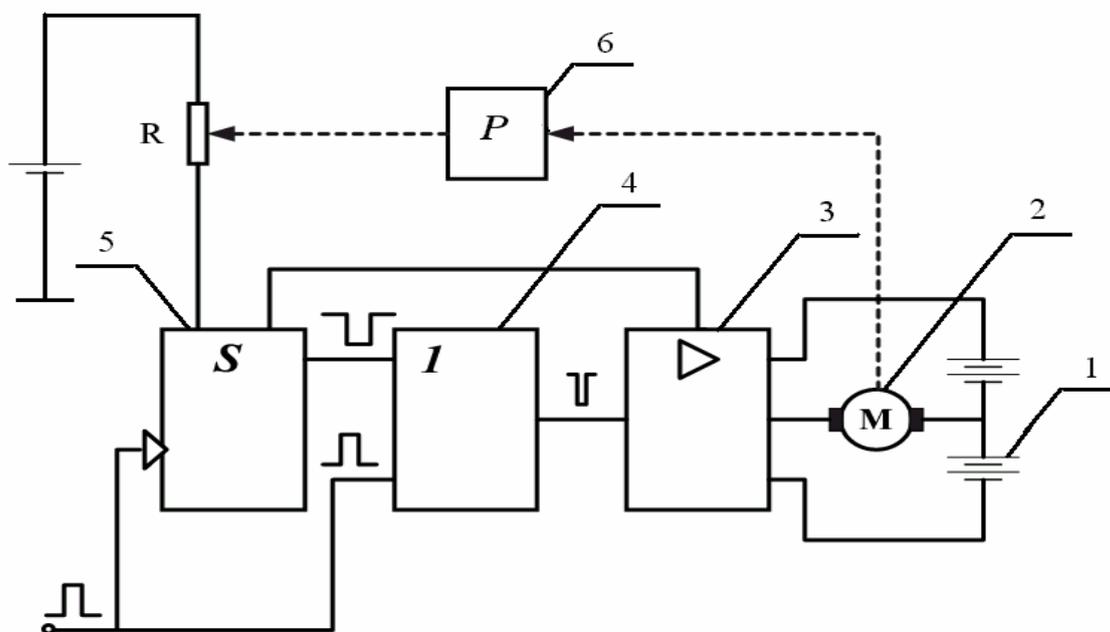


Рис. 2. Схема рулевого привода

Как видно из рис. 2, серводвигатель 2 связан при помощи редуктора 6 с потенциометром, причем генератор опорных импульсов 5 начинает функционировать после прихода каналного импульса и выдает эталонный сигнал. Затем этот эталонный импульс сравнивается с пришедшим каналным, и остаточный сигнал управляет усилителем 3. Серводвигатель 2 подсоединен к выходу усилителя 3 и в зависимости от рассогласованного сигнала осуществляет вращение в ту или иную сторону. При этом, вращаясь, он с помощью редуктора 6 меняет сопротивление потенциометра опорного генератора 5 таким образом, чтобы длительность опорного импульса постепенно менялась в сторону сближения с длительностью каналного импульса. Остаточный импульс с течением времени исчезает, и серводвигатель 2 останавливается в данном

положении. Для того чтобы процесс регулировки протекал быстро, но без качения, в схеме предусмотрено эффективное демпфирование (гашение колебаний).

Как видно из сказанного, самым простым решением для определения угла поворота является потенциометрический датчик. Датчик такого типа имеет простой принцип действия, что приводит к конструированию дешевой измерительной техники. Измеряемый сигнал, который он позволяет получать, может иметь весьма высокий уровень и не требовать специальной электрической схемы обработки.

Сигнал, приходящий с потенциометра R, поступает в плату ввод/вывода AD 512, а затем с помощью специально разработанного программного обеспечения осуществляется определение угла поворота переднего колесного модуля.

Диапазон измерения данного датчика – от минус $35,84^\circ$ до $25,16^\circ$ с точностью до $4,51^\circ$.

Датчик скорости вращения заднего привода

В наиболее простой форме датчик скорости вращения заднего привода состоит из источника света 1 и оптического приемника 4 – фотодиода.

Вращающееся тело 5 соединяется с диском, имеющим попеременно прозрачные и непрозрачные сектора, который располагают между фокусирующей линзой 2, входящей в состав источника 1, и приемником света 4 (см. рис. 3).

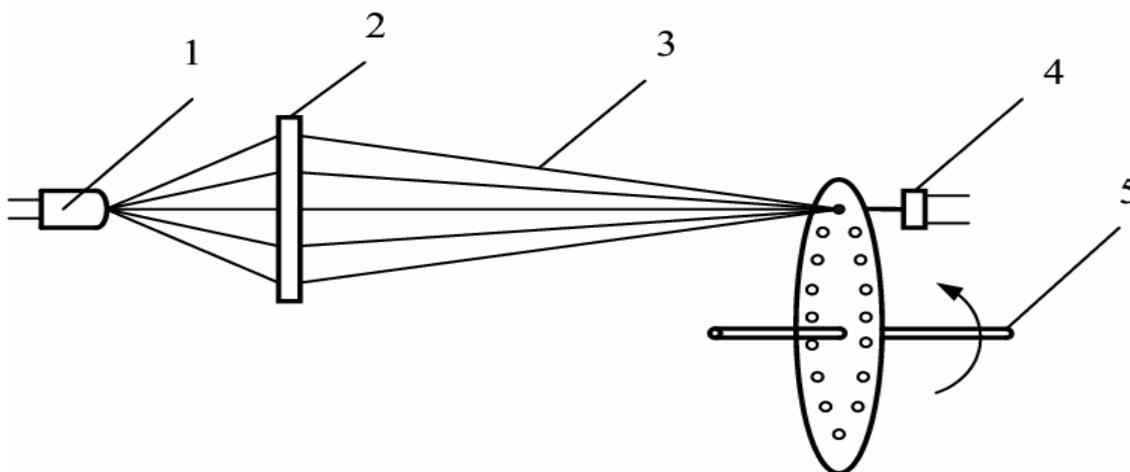


Рис. 3. Схема датчика скорости вращения заднего привода

Принцип действия датчика скорости состоит в следующем. Получив модулированный скачкообразными изменениями пропускания поток, фотоприемник выдает электрический сигнал с частотой, пропорциональной скорости вращения, и амплитудой, не зависящей от этой скорости.

Диапазон измеряемых скоростей зависит, с одной стороны, от числа импульсов, с другой стороны – от полосы пропускания приемника и связанных с ним электрических схем.

Применение двух приемников, сдвинутых на четверть периода по пространству, позволяет определять направление вращения. Если осуществляется вращение в положительном направлении, то сигналы, приходящие с приемника 1 и приемника 2, совпадают. В противном случае имеет место рассогласование между сигналами приемников. Электрическая схема усиливает сигналы приемников. Затем эти сигналы поступают в плату ввод/вывода AD 512, и с помощью специально разработанного

программного обеспечения в среде программирования MATLAB осуществляется определение скорости вращения колеса.

Достоинством датчика является высокое быстродействие (ширина полосы 1 ГГц), низкий уровень шума, простота в обработке.

Измерительная схема

Сигнал с датчиков скорости и угла поворота имеют недостаточную амплитуду для согласования с платой ввода-вывода AD 512. С этой целью используется электрическая схема, представленная на рис. 4. На этой схеме резистор R является потенциометрическим датчиком рулевой машинки, R1 служит для регулировки выходного напряжения, R2 предназначен для определения коэффициента усиления, R3 – для компенсации дрейфа нуля, R4 – для ограничения тока диода, R5 – для установки порога срабатывания компаратора. D1 является операционным усилителем, который усиливает сигнал с датчика угла поворота до уровня, соответствующего уровню опорного напряжения платы ввода-вывода AD 512. D2 – компаратор, предназначенный для выделения полезного сигнала с датчика скорости [4].

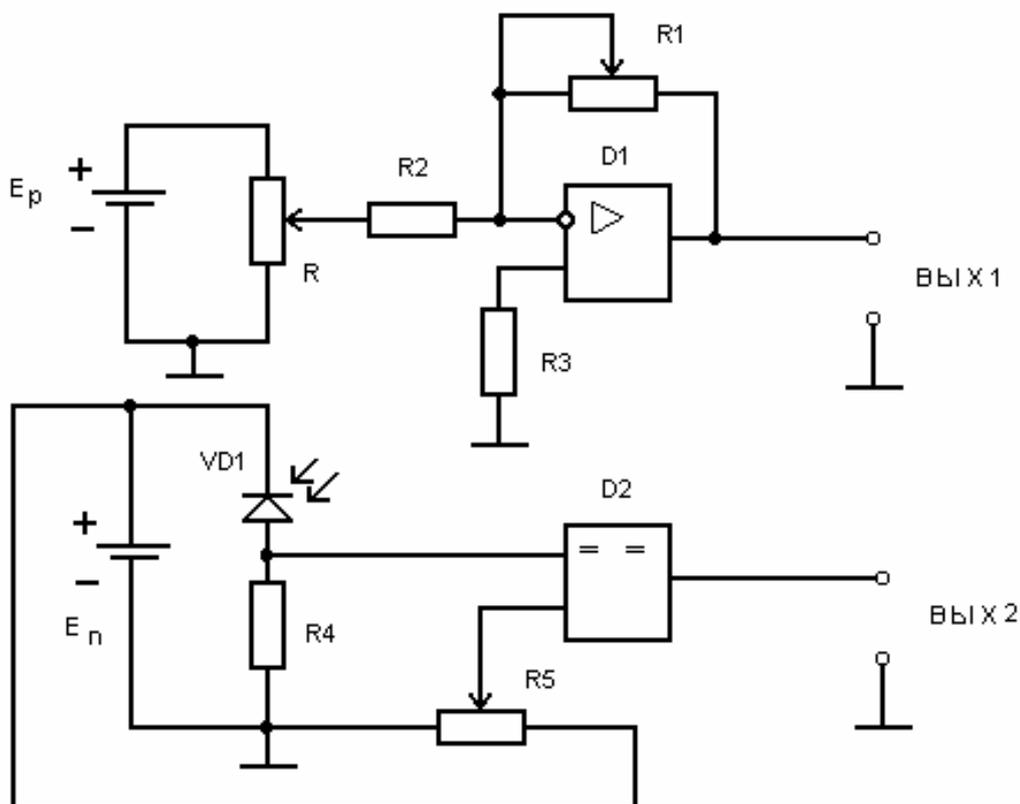


Рис. 4. Согласующая электрическая схема

На рис. 5–6 приведены сигналы, снимаемые с датчиков угла поворота и скорости, а также определенные значения скорости и угла поворота. Проведенные экспериментальные исследования показали работоспособность разработанных измерительных устройств.

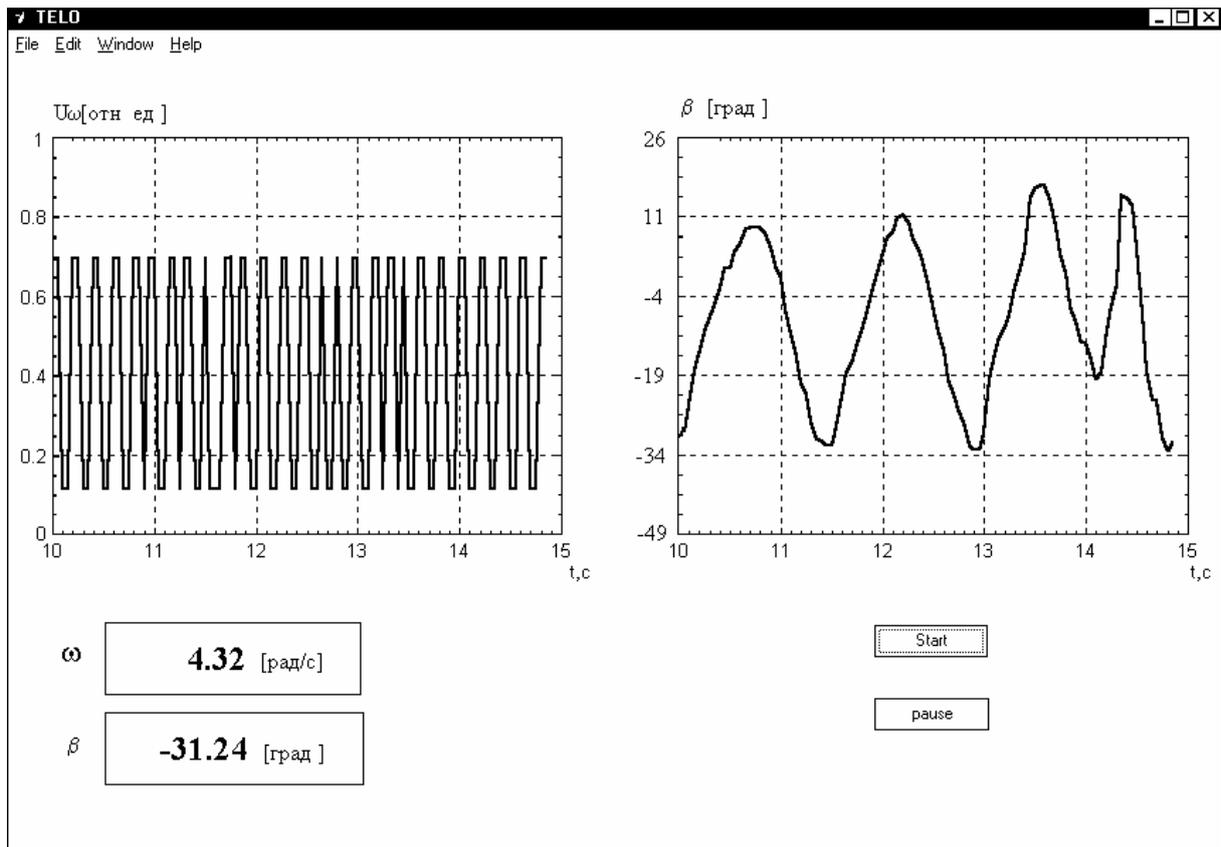


Рис. 5. Результаты эксперимента

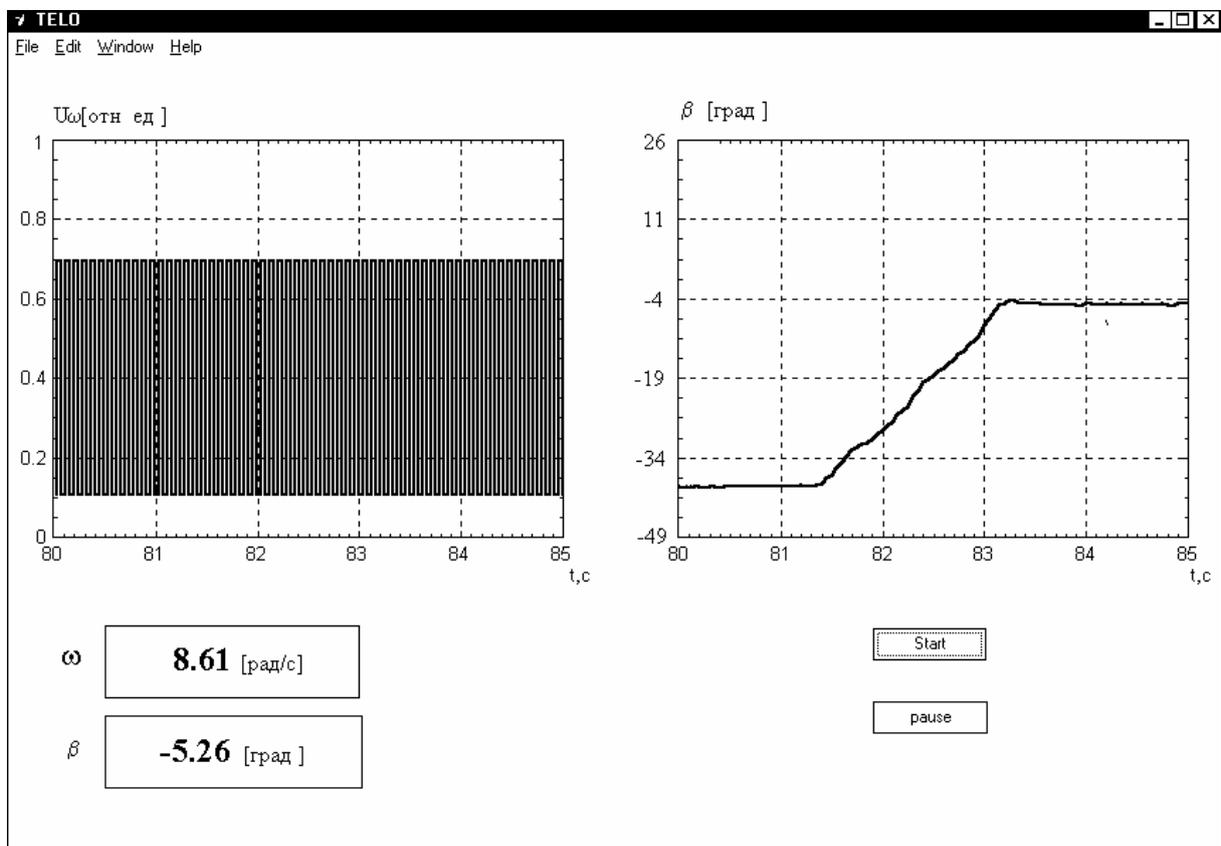


Рис. 6. Результаты эксперимента

Литература

1. Гель П Как превратить персональный компьютер в измерительный комплекс: Пер. с франц. 2-е изд., испр. М.: ДМК, 1999.
2. Аш Ж. и соавторы. Датчики измерительных систем: В 2-х книгах. Пер. с франц. М.: Мир, 1992.
3. Миль Г. Модели с дистанционным управлением: Пер. с нем. Л.: Судостроение, 1984.
4. Хоровиц П., Хилл У. Искусство схемотехники: В 2-х т. Т. 2. Пер. с англ. Изд. 3-е, стереотип. М.: Мир, 1986.

АНАЛИЗ ПРОЦЕССОВ В СИСТЕМАХ С МОДУЛЯЦИЕЙ – ДЕМОДУЛЯЦИЕЙ ПРИ ВНЕШНЕМ КОНЕЧНОМЕРНОМ ВОЗДЕЙСТВИИ С ИСПОЛЬЗОВАНИЕМ КРОНЕКЕРОВСКОГО МОДЕЛЬНОГО ПРЕДСТАВЛЕНИЯ

Л. В. Кожевникова, А. В. Ушаков

Рассматривается задача анализа процессов в системах с модуляцией – демодуляцией при внешнем конечномерном воздействии с использованием возможностей кронекеровских векторно-матричных представлений. Полученные результаты иллюстрируются примером.

Введение

В развитие представлений, изложенных в работах [1, 2] и посвященных анализу возможностей кронекеровских векторно-матричных представлений для целей исследования динамических процессов с модуляцией – демодуляцией и фильтрацией, в настоящей работе ставится задача описания всего цикла прохождения конечномерного сигнала через динамическую систему, использующую сигнал-носитель ненулевой частотной модуляции [6].

Как и в [1, 2], описание всего цикла преобразования сигнала модуляция – демодуляция – фильтрация осуществляется с использованием возможностей кронекеровских векторно-матричных представлений.

Постановка задачи

Весь цикл преобразования задающего воздействия в динамических системах с сигналом-носителем ненулевой частоты, который состоит из процессов модуляции, демодуляции и фильтрации, представлен структурной схемой рис. 1.

Модельные представления всех фаз указанных выше преобразования сигналов в таких системах с использованием кронекеровских векторно-матричных представлений обнаруживают дополнительные возможности. Эти дополнительные возможности порождаются тем обстоятельством, что матрицы состояния системы с модуляцией – демодуляцией (М-ДМ) содержат в своей структуре кронекеровские матричные суммы матрицы состояния источника внешнего задающего воздействия, источника моделирующего сигнала, источника демодулирующего сигнала, а также исследуемой динамической системы.

В силу свойств алгебраического спектра собственных значений таких матриц они составлены из аддитивных композиций собственных значений матричных компонентов кронекеровских сумм на всех возможных сочетаниях их индексов.

По существу это алгебраическое свойство кронекеровских матричных сумм содержит доказательство утверждения: \square

Утверждение 1(У.1). Процессы модуляции и демодуляции являются спектрально-согласованными (в смысле спектров собственных значений), если среди собственных значений $\{\lambda_{Mi}; i = \overline{1, n_M}\}$ и $\{\lambda_{Di}; i = \overline{1, n_D}\}$ матриц состояния Γ_M и Γ_D , соответственно источников модулирующего и демодулирующего сигналов есть такие значения, которые порождают нулевые аддитивные структуры:

$$\lambda_{Mi} + \lambda_{Di} = 0. \quad \blacksquare \quad (1)$$

Нетрудно видеть, что соотношение (1), являясь необходимым условием согласованности процессов модуляции и демодуляции, является содержательным математическим основанием для построения согласованных пар сигналов модуляции и демодуляции. Простейшими из них являются одночастотные модулирующие сигналы.

Как известно, при суммировании модулированных одночастотных сигналов обнаруживается проблема их фазирования, которая не существует в теории и практике в системах, работающих на постоянном токе. Эта проблема в рамках кронекеровских векторно-матричных представлений сводится к проблеме согласования начальных состояний генераторов модулирующего и демодулирующего сигналов. Задача фазирования здесь приобретает постановку оптимизационной задачи, которую можно выразить в форме следующего утверждения.

Утверждение 2 (У.2). Начальные состояния $Z_M(0)$, $Z_D(0)$ источников одночастотных гармонических модулирующего и демодулирующего сигналов являются согласованными, если постоянная составляющая сигнала на выходе фильтра устройства демодуляции при постоянном внешнем задающем воздействии принимает максимальное на множестве пар $Z_M(0)$, $Z_D(0)$ значение, или, что то же самое, выполняется соотношение:

$$\{Z_M(0), Z_D(0)\} = \arg \max_{Z_M(0), Z_D(0)} \left\{ \|Y_\Phi(t)\| \right\} \text{ при } g(t) = g_0 = \text{const}. \quad \blacksquare \quad (2)$$

Основной результат

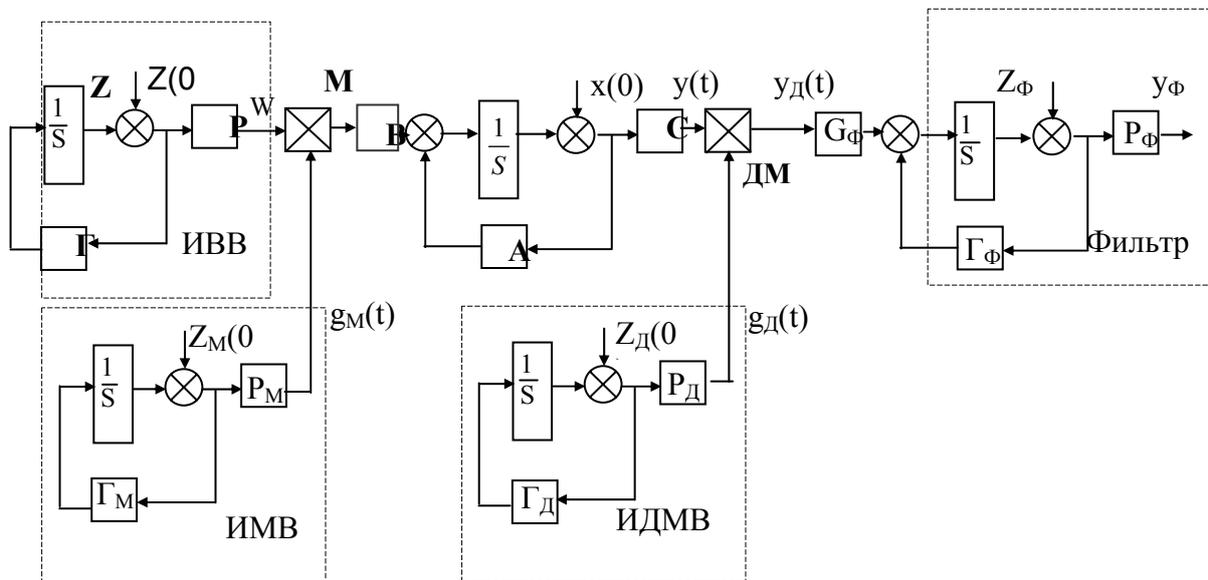


Рис. 1. Содержательная структурная схема:

Полная схема преобразования сигнала источника внешнего воздействия (ИВВ) в модулированный сигнал с последующим его прохождением через динамическую систему устройством демодуляции и фильтрации демодулированного сигнала приведена на рис. 1.

Основные функциональные звенья схемы имеют следующие векторно-матричные представления.

ИВВ – источник внешнего воздействия, описываемый системой

$$\dot{z} = \Gamma z; z(0); g(t) = Pz(t). \quad (3)$$

ИМВ – источник модулирующего воздействия:

$$\dot{z}_M = \Gamma_M z_M; z_M(0); g_M(t) = P_M z_M(t). \quad (4)$$

Объект управления (система):

$$\dot{x} = Ax + Bu; x(0); y(t) = Cx(t). \quad (5)$$

ИДМВ – источник демодулирующего воздействия:

$$\dot{z}_D = \Gamma_D z_D; \quad z_D(0); \quad g_D(t) = P_D z_D(t). \quad (6)$$

Опишем процесс модуляции (выход блока М на схеме), используя оператор кронекеровского сужения с матрицей S , в форме

$$u = S(W \otimes g_M). \quad (7)$$

Процесс демодуляции (на выходе блока Д) задается как

$$y_D = S(y \otimes g_D), \quad (8)$$

а фильтр – как

$$\dot{z}_\Phi = \Gamma_\Phi z_\Phi + G_\Phi y_D; \quad y_\Phi(t) = P_\Phi z_\Phi(t). \quad (9)$$

Далее составим единую векторно-матричную модель. Сначала объединим векторные модели ИВВ и ИМВ $z \otimes z_M$:

$$\frac{d}{dt}(z \otimes z_M) = (\Gamma \otimes I + I \otimes \Gamma_M)(z \otimes z_M); \quad z(0) \otimes z_M(0), \quad (10)$$

т.е. на входе системы будем иметь следующее воздействие:

$$u(t) = S\{g(t) \otimes g_M(t)\} = S\{(P \otimes P_M)(z \otimes z_M)\} = S(P \otimes P_M)(z \otimes z_M). \quad (11)$$

Объединение (11) с моделью (5) дает описание прохождения модулированного сигнала через объект управления:

$$\dot{x} = Ax + Bu(t) = Ax + B \cdot S(P \otimes P_M)(z \otimes z_M). \quad (12)$$

Сформируем вектор с компонентами $(z \otimes z_M)$ и x . Тогда дифференциальная модель его изменения во времени принимает вид

$$\begin{bmatrix} \dot{(z \otimes z_M)} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} \Gamma \otimes I + I \otimes \Gamma_M & | & 0 \\ \hline B \cdot S(P \otimes P_M) & | & A \end{bmatrix} \begin{bmatrix} (z \otimes z_M) \\ x \end{bmatrix}. \quad (13)$$

Соотношение (13) получает эквивалентное представление

$$\tilde{x} = \begin{bmatrix} z \otimes z_M \\ x \end{bmatrix}; \quad \dot{\tilde{x}}(t) = \tilde{A}\tilde{x}; \quad \tilde{x}(0); \quad y(t) = \tilde{C}\tilde{x}(t), \quad (14)$$

где

$$\tilde{A} = \begin{bmatrix} \Gamma \otimes I + I \otimes \Gamma_M & | & 0 \\ \hline B \cdot S(P \otimes P_M) & | & A \end{bmatrix}; \quad \tilde{C} = [0 \ | \ C]. \quad (15)$$

Если теперь систему (14), (15) объединить с процессом демодуляции, то получим следующее дифференциальное представление:

$$\begin{aligned} & \begin{bmatrix} \dot{z \otimes z_M \otimes z_D} \\ \dot{x \otimes z_D} \end{bmatrix} = \\ & = \begin{bmatrix} \Gamma \otimes I_M \otimes I_D + I \otimes \Gamma_M \otimes I_D + I \otimes I_M \otimes \Gamma_D & | & 0 \\ \hline B \cdot S(P \otimes P_M \otimes I_D) & | & A \otimes I_D + I \otimes \Gamma_D \end{bmatrix} \cdot \\ & \cdot \begin{bmatrix} z \otimes z_M \otimes z_D \\ x \otimes z_D \end{bmatrix} = \begin{bmatrix} Si(\Gamma, \Gamma_M, \Gamma_D) & | & 0 \\ \hline B \cdot S(P \otimes P_M \otimes I_D) & | & Si(A, \Gamma_D) \end{bmatrix} \cdot \begin{bmatrix} z \otimes z_M \otimes z_D \\ x \otimes z_D \end{bmatrix} \end{aligned} \quad (16)$$

и сформируем выходной сигнал y_D демодулятора в виде

$$\begin{aligned} y_D &= S(\tilde{C} \otimes P_D)(\tilde{x} \otimes z_D) = S([0 \ | \ C] \otimes P_D)(\tilde{x} \otimes z_D) = \\ &= S[0 \otimes P_D \ | \ C \otimes P_D] \cdot [\tilde{x} \otimes z_D] \end{aligned} \quad (17)$$

при этом

$$\tilde{x} \otimes z_D = [0 \mid I] \begin{bmatrix} z \otimes z_M \otimes z_D \\ \tilde{x} \otimes z_D \end{bmatrix}.$$

Для полного описания процесса в системе с модуляцией – демодуляцией при внешнем конечномерном воздействии с использованием кронекеровского модельного представления дополним систему (17) процессом фильтрации, в результате чего получим:

$$\begin{bmatrix} \dot{z} \otimes z_M \otimes z_D \\ \dot{x} \otimes z_D \\ \dot{z}_\phi \end{bmatrix} = \begin{bmatrix} Si\{\Gamma, \Gamma_M, \Gamma_D\} & 0 & 0 \\ B \cdot S(P \otimes P_M \otimes I_D) & Si\{A, \Gamma_D\} & 0 \\ 0 & G_\phi \cdot S(C \otimes P_D) & \Gamma_\phi \end{bmatrix} \cdot \begin{bmatrix} z \otimes z_M \otimes z_D \\ x \otimes z_D \\ z_\phi \end{bmatrix};$$

$$\begin{bmatrix} z \otimes z_M \otimes z_D \\ x \otimes z_D \\ z_\phi \end{bmatrix} = \begin{bmatrix} z(0) \otimes z_M(0) \otimes z_D(0) \\ x(0) \otimes z_D(0) \\ z_\phi(0) \end{bmatrix},$$

$$y_\phi = \tilde{P} \cdot \begin{bmatrix} z \otimes z_M \otimes z_D \\ x \otimes z_D \\ z_\phi \end{bmatrix} = [0 \mid 0 \mid P_\phi] \begin{bmatrix} z \otimes z_M \otimes z_D \\ x \otimes z_D \\ z_\phi \end{bmatrix}. \quad (18)$$

Итак, объединенную систему можно описать следующим образом:

$$\tilde{\dot{A}} = \begin{bmatrix} Si\{\Gamma, \Gamma_M, \Gamma_D\} & 0 & 0 \\ B \cdot S(P \otimes P_M \otimes I_D) & Si\{A, \Gamma_D\} & 0 \\ 0 & G_\phi \cdot S(C \otimes P_D) & \Gamma_\phi \end{bmatrix},$$

$$\tilde{\tilde{x}} = \begin{bmatrix} z \otimes z_M \otimes z_D \\ x \otimes z_D \\ z_\phi \end{bmatrix};$$

$$\dot{\tilde{\tilde{x}}} = \tilde{\tilde{A}}\tilde{\tilde{x}}; \tilde{\tilde{x}}(0); y_\phi = \tilde{\tilde{C}}\tilde{\tilde{x}}, \quad (19)$$

где $\tilde{\tilde{C}} = [0 \mid 0 \mid P_\phi]$. Нетрудно видеть, что (19) представляет собой описание автономной системы, для решения которой достаточно процедуры вычисления матричной экспоненты. Процессы по внешнему конечномерному воздействию, модулированному сигналу на выходе системы (5), демодулированному сигналу (8) и сигналу на выходе фильтра (18) с помощью строк матрицы выхода могут быть сформированы из вектора состояния системы (19).

Полученная модель (19) предоставляет разработчику богатые возможности по проведению комплексного компьютерного эксперимента с целью исследования динамических систем, дополненных процессом модуляции – демодуляции – фильтрации.

Литература

1. Кожевникова Л.В., Ушаков А.В. Кронекеровская матричная модель динамических процессов с модуляцией. // Научно-технический вестник СПб ГИТМО (ТУ). Выпуск 6. Информационные, вычислительные и управляющие системы / Главный редактор В.Н. Васильев. СПб: СПб ГИТМО (ТУ), 2002.
2. Кожевникова Л.В., научный руководитель А.В. Ушаков. Кронекеровские технологии в исследовании многомерных динамических систем. // Современные технологии. Сборник научных статей / Под редакцией С.А. Козлова и В.О. Никифорова. СПб: СПбГИТМО (ТУ), 2002.
3. Ланкастер П. Теория матриц. / Пер. с англ. М.: Наука, 1978..
4. Гантмахер Ф.Д. Теория матриц. М.: Наука, 1988.
5. Сабинин Ю.А. Позиционные и следящие электромеханические системы: Учебное пособие. СПб: Энергоатомиздат, Санкт-Петербургское отд-ние, 2001.
6. Куракин К.И., Куракин Л.К. Анализ систем автоматического регулирования на несущей переменного тока. М.: Машиностроение, 1978.
7. Заде Л., Дезоер Ч. Теория линейных систем. / Пер. с англ. М.: Наука, 1970.
8. Ту Ю. Современная теория управления. / Пер. с англ. М.: Наука, 1971.

РЕЗИДЕНТНОЕ И ИНСТРУМЕНТАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПРОМЫШЛЕННЫХ КОНТРОЛЛЕРОВ СЕТЕВОГО ВВОДА/ВЫВОДА

И.С. Болгаров, А.А. Патиев, А.О. Ключев

Значительная часть систем промышленной автоматизации построена на основе распределенных сетей управления (PCY). В качестве устройств сопряжения с объектом (УСО) в таких сетях используются различные модули ввода/вывода (контроллеры сетевого ввода/вывода), работающие под управлением основного контроллера сети (к примеру, это может быть ПЛК – программируемый логический контроллер). Модули ввода/вывода осуществляют прием и первичное преобразование информации от датчиков и выдачу управляющих сигналов на исполнительные устройства.

В качестве сетевого интерфейса для связи с главным контроллером могут использоваться как стандартные (PROFIBUS, Interbus-S, DeviceNet, CANopen и др.), так и интерфейсы с протоколами, разработанными фирмой-производителем для своих систем (например, Controller Link и CompoBus/S фирмы OMRON).

Основные понятия

Значительную часть в процессе разработки модулей сетевого ввода/вывода занимает разработка резидентного и инструментального программного обеспечения. Дадим определения этим понятиям.

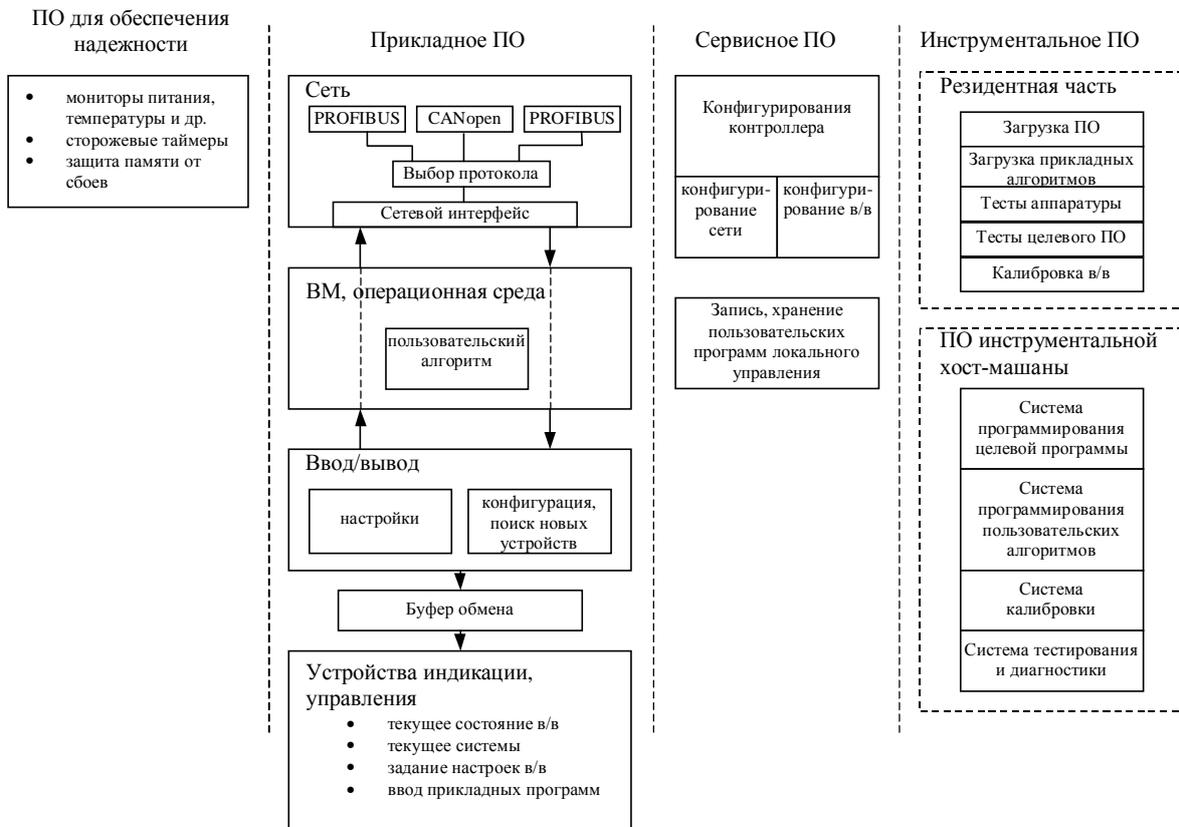


Рис. 1. Структура программного обеспечения модулей сетевого ввода/вывода

Резидентным называется программное обеспечение (ПО), входящее в состав встраиваемой системы и находящееся в памяти контроллера. Инструментальное ПО используется при разработке программно-аппаратных частей системы, при отладке на этапе разработки, при тестировании и диагностике в процессе эксплуатации. Резидентное ПО делится на прикладное, выполняющее целевую задачу в режиме реального времени, сервисное, используемое для настройки системы пользователем, резидентное инструментальное и ПО обеспечения надежности. В отличие от резидентного инструментального ПО часто не входит в состав системы, за исключением некоторых частей, которые могут в дальнейшем понадобиться при эксплуатации системы (например, система загрузки пользовательских алгоритмов, система калибровки входов/выходов контроллера).

Чтобы очертить границы резидентного и инструментального программного обеспечения, необходимо выделить основные компоненты ПО и связи между ними. Однако зачастую состав ПО определяется архитектурой конкретных модулей ввода/вывода. В приведенной на рис. 1 структурной схеме представлены компоненты, используемые при различных решениях по организации модулей. Связи между элементами ПО в различных системах могут меняться в определенных пределах.

Рассмотрим основные части ПО РСУ на примере системы с модулями сетевого ввода-вывода [1, 2].

Сеть

Поскольку для построения РСУ применяются различные сетевые протоколы, очевидно, модули должны поддерживать несколько протоколов. В зависимости от исполнения модуля ввода/вывода изменение сетевой части может потребовать применения новых аппаратных решений и перекомпиляции программы при условии того, что модуль выполнен в виде единого блока. В тех же случаях, когда возможна модульная замена сетевого адаптера, изменение сетевого программного обеспечения может производиться автоматически.

Виртуальная машина (ВМ) или операционная среда

Виртуальная машина применяется в так называемых активных модулях для исполнения пользовательских алгоритмов. В подобном случае часть задачи управления перекладывается на модуль ввода/вывода.

Одной из основных проблем при таком решении является синхронизация работы пользовательских алгоритмов с командами управляющего контроллера.

Для работы ВМ в контроллер должны быть загружены пользовательские алгоритмы, осуществляется это при помощи системы доставки программ в ВМ.

Пользовательские алгоритмы также могут реализовываться при помощи языков, не требующих поддержки виртуальной машины (к примеру, С или С++). Однако в этом случае для замены программы локального управления потребуется заменить все резидентное программное обеспечение контроллера.

Ввод/вывод

Подсистема ввода/вывода содержит набор драйверов для взаимодействия с устройствами ввода/вывода. В состав может входить система поиска новых устройств, если контроллер распределенного ввода/вывода выполнен в модульном варианте. Также для конфигурирования ввода/вывода и для передачи параметров в систему индикации должен присутствовать блок настройки.

Система индикации и управления

Система индикации и управления отображает состояние системы и устройств ввода/вывода, позволяет задавать режимы каналов ввода/вывода и осуществлять ввод прикладных программ. Система может поддерживать различные устройства индикации, начиная от простых светодиодных панелей до жидкокристаллических индикаторов, и различные типы устройств ввода: кнопки, аналоговые и дискретные сенсорные панели, энкодеры и другие. При помощи элементов управления пользователь может:

- выбирать настройки в/в для отображения,
- вводить программы локального управления,
- настраивать устройства в/в.

Система конфигурирования

Система конфигурирования обеспечивает конфигурирование сети и подсистемы ввода/вывода. В задачу системы конфигурирования сети входит выбор соответствующего драйвера сетевого интерфейса (если существует возможность физической замены сетевых адаптеров) и/или конфигурация параметров сетевого соединения, назначение сетевых адресов контроллеров.

Система конфигурирования подсистемы ввода/вывода позволяет задавать определенные режимы работы устройств при инициализации системы. Конфигурация может проводиться через специальный канал или прозрачно через основную информационную сеть.

Система записи и хранения пользовательских программ

Эта система применяется для ввода пользовательской программы с консоли модуля ввода/вывода или по информационной сети. Она позволяет хранить программы и передавать их на исполнение ВМ.

Инструментальное ПО

Инструментальное ПО делится на резидентное и ПО инструментальной хост-машины.

В состав резидентного ПО контроллеров сетевого ввода-вывода может входить система загрузки программ, средства тестирования аппаратуры и ПО, средства калибровки аналого-цифровых трактов и узлов.

Загрузка ПО

Системы начальной загрузки ПО подразделяются на:

- встраиваемые в процессор (для начальной инициализации ПЗУ или ОЗУ программ);
- расположенные в ПЗУ контроллера (осуществляют загрузку программы в ПЗУ или ОЗУ), могут быть ориентированны на различные инструментальные системы программирования.

Загрузчик пользовательских программ

Позволяет загружать/выбирать пользовательские программы через специальный канал программирования или через сеть. Предполагает наличие резидентного системного ПО в контроллере.

Тесты аппаратуры

Эти тесты используются для проверки аппаратной части системы на этапе настройки и/или во время исполнения. Тестирование может производиться с участием пользователя (интерактивный режим тестирования) и без участия пользователя. В последнем случае зачастую необходимо аппаратное обеспечение в виде платы тестирования.

Тесты целевого ПО

Эти тесты применяются как на этапе создания системы, так и для тестирования в процессе производства и эксплуатации. Обычно они включают в себя:

- систему вызова процедур;
- систему проверки корректности работы ПО при длительной работе;
- систему замера времени исполнения участков программы и другое.

Калибровка каналов ввода/вывода

Калибровка используется при настройке каналов ввода/вывода для работы в различных режимах. Наибольшую сложность представляет калибровка аналого-цифровых высокоточных трактов, которые предполагают несколько режимов работы. Задача усложняется в случае широкого температурного рабочего диапазона контроллера и высокой чувствительности его измерительных входов.

ПО инструментальной хост-машины

ПО инструментальной хост-машины_представляет собой сложный комплекс средств, в состав которых в общем случае входят:

- система создания и загрузки целевых программ;
- система программирования и загрузки пользовательских алгоритмов;
- управляющая часть системы калибровки;
- система тестирования и диагностики.

Отдельно следует обозначить программную составляющую РСУ для обеспечения надежности функционирования системы в целом. Такое ПО включает средства, направленные на защиту от сбоев и неисправностей, и может содержать мониторы питания и температуры, сторожевые таймеры, средства защита памяти контроллера и многое другое.

Литература

1. <http://lmt-automation.ifmo.ru>
2. <http://embedded.ifmo.ru>

КОНЦЕПЦИЯ ПОСТРОЕНИЯ ИНТЕГРИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ РЕГИОНАЛЬНОЙ СИСТЕМОЙ ОБРАЗОВАНИЯ

В.Н. Васильев, В.В. Кириллов, Г.Ю. Громов, В.С. Черемухин,
Д.А. Дорохин, А.В. Скатын

Региональная система образования издревле имеет иерархическую структуру (рис. 1): образовательные учреждения (ОУ) – районные отделы народного образования (роно) – городские отделы народного образования (гороно) – областные отделы народного образования (облоно) и т.д.

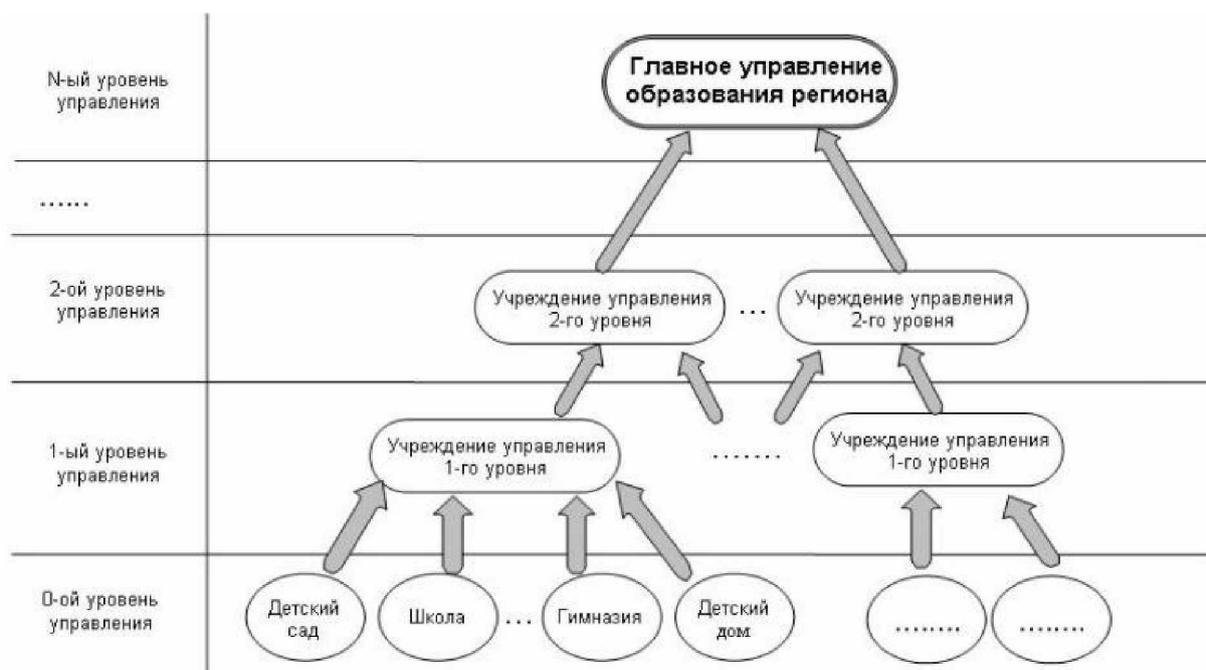


Рис. 1. Иерархическая структура региональной системы образования

Для обеспечения собственных управляющих функций и функционирования вышестоящих организаций "...оно" собирают различную оперативную и тактическую информацию от подчиненных им учреждений, агрегируют ее и пересылают выше (рис. 2).

На нижнем уровне находится повторяющаяся, предсказуемая, "рутинная" и часто обновляемая информация, называемая *оперативной*. Такая информация нужна для повседневной деятельности сотрудников образовательных учреждений (ОУ) и учреждений управления образованием различных уровней, часто создающих новые оперативные и тактические данные.

Следующий уровень составляет *тактическая* информация. Она получается путем дальнейшей обработки оперативных данных. Ранее обработанные данные суммируются или приводятся в другом контексте. Тактические решения касаются размещения ресурсов и планирования. Они сложнее операционных данных и часто в дополнение к фактической информации включают информационные оценки. Тактическая информация предсказуема, она повторяется, но создают ее не так часто, как оперативную. Эта информация более агрегирована, чем оперативная, а отчеты на ее основе составляются по известным формам в плановые сроки.

На вершине иерархии располагается *стратегическая* информация (информация для руководства). На основе этой информации руководство разрабатывает политику управления организацией или организациями региона в целом и осуществляет

долгосрочное планирование. Информация, запрашиваемая руководством для себя или высших инстанций, заранее непредсказуема. Она образуется путем агрегирования тактической и частично оперативной информации. Непредсказуемость и сложность отыскания запрашиваемых данных часто не позволяет получить их в приемлемые сроки и с необходимой степенью достоверности.



Рис. 2. Иерархия информации

Так, в начале учебного года ОУ должны предоставлять в учреждения управления образованием первого уровня множество данных по заданным формам (ОШ-п, п-ДО и т.п.). Среди них есть, например, сведения о составе педагогических работников (возраст, пол, образование, стаж педагогической работы, разряд ЕТС и т.п.) – сведения из личных дел сотрудников ОУ. Учреждения управления образованием первого уровня обобщают полученные данные и пересылают их на следующий уровень и т.д. до Госкомстата России. Рассмотренный процесс сбора, обобщения и пересылки статистической информации достаточно трудоемок, однообразен и, как правило, сопровождается большим количеством ошибок.

Отметим недостатки неавтоматизированной системы управления образованием:

- высокая трудоемкость подготовки и передачи информации с уровня на уровень;
- большое количество ошибок при сборе и агрегировании информации;
- при агрегировании исчезает часть информации о природе данных;
- практически невозможно получить информацию, непосредственно не включенную в статистические отчеты.

Развитие компьютерной обработки данных позволило начать процесс автоматизации деятельности учреждений образования и управления ими. Стали создаваться информационные системы управления (ИС) школами, учреждениями управления образованием различных уровней (Воронеж, Москва, Республика Карелия, Самара, Санкт-Петербург, Ярославль и др.).

Эти ИС автоматизируют деятельность отдельных видов учреждений, находящихся на разных уровнях иерархии региональной системы образования – от разнообразных видов образовательных учреждений (нулевой уровень) до главных управлений образованием региона, находящихся на вершине иерархии. Кроме автоматизации повседневной деятельности сотрудников этих учреждений, подобные

ИС позволяют упростить подготовку множества письменных отчетов, стандартных учетных форм и ответов на запросы, необходимых различным учреждениям вышестоящего уровня иерархии, а также автоматизировать обработку получаемых документов.

При таком подходе в регионе должно быть установлено множество ИС, по крайней мере, двух видов (ИС образовательного учреждения и ИС учреждения управления образованием), и каждая из них должна обслуживаться тем или иным числом квалифицированных специалистов. Однако простая автоматизация труда отдельных сотрудников (учителей, работников и руководителей различных организаций), которые занимаются сбором и обобщением, необходимой для управления, статистической информации, не позволит качественно изменить систему управления образованием региона. Руководители опять смогут использовать для анализа только агрегированные данные из отчетов, в которых уже исчезла информация о природе этих данных. Уточнение же потребует значительных усилий и времени, так как придется спускаться по иерархической лестнице до первичных данных и формировать соответствующий запрос.

Кроме того, при таком подходе почти не уменьшается (а часто даже и увеличивается) число сотрудников, занимающихся обработкой данных. К сотрудникам, занимающимся разработкой разнообразных форм для подчиненных учреждений и технических заданий на создание этих форм, а также форм, затребованных вышестоящим учреждением, добавляются сотрудники, обслуживающие ИС и создающие новые приложения для ее правильного функционирования. Кроме затрат на содержание квалифицированного персонала ИС, производятся и значительные затраты на оборудование. Но самое главное, что при таком подходе сохраняется большинство недостатков неавтоматизированных систем управления образованием.

Рассмотренная организация ИС не позволяет обеспечить оперативную выдачу руководству специальной, не предусмотренной формами, информации, затрудняет обмен данными с другими ИС, затрудняет ведение учета и контроля информации, в значительной мере усложняет процедуры модификации хранимых данных и программ, использующих эти данные. Такую систему трудно администрировать: сложно представить себе администратора, которому нужно обновить программу на сотнях или даже тысячах компьютеров, расположенных в разных помещениях, зданиях, городах и т.п. Ситуация усугубляется неоднородностью компьютеров на рабочих местах. Кроме того, современный персональный компьютер весьма уязвим для непродуманных или злонамеренных действий работающего на нем человека. Он может быть надолго выведен из строя разного рода манипуляциями с файловой системой, правкой конфигурационных файлов, установкой новых пакетов (не говоря уже о тривиальных вирусах).

Исключить указанные недостатки можно с помощью единой интегрированной информационно-аналитической системы управления образованием, которая должна обслуживать все образовательные учреждения и все местные (городские, районные и др.) управления образованием различных уровней (рис. 3).

С этой системой с помощью "тонкого" клиента (стандартного Web-браузера) связаны все пользователи указанных учреждений, для которых она является как бы их собственной системой управления школой, районным или городским отделом образования и т.п.

В центральной интегрированной базе данных ИС должны храниться только данные из первичных документов перечисленных учреждений и организаций. Итоговые документы и те промежуточные документы, от которых нельзя избавиться при существующей системе документооборота, будут генерироваться из первичных с помощью прикладных программ. Агрегированная же информация (многолетние

аналитические, статистические отчеты и т.п.) может содержаться в специально организованном хранилище данных аналитической системы поддержки принятия решений (если возникнет необходимость в создании такого хранилища).



Рис. 3. Предлагаемая структура ИС управления региональной системой образования

Предлагаемая архитектура, которая сконцентрировала и объединила в себе лучшие качества централизованных систем и традиционных систем клиент-сервер, отличается следующими чертами:

- на сервере ИС порождается конечный продукт – информация в форме, предназначенной для представления пользователю (а не полуфабрикат в виде данных);
- для обмена информацией между клиентом и сервером используется протокол открытого стандарта (стандарт передачи данных в сети Интернет);
- информация передается клиентам в виде, пригодном для восприятия человеком (в интерфейсе навигатора Интернет, который уже начинает вытеснять пользовательский интерфейс многих программных продуктов, например, операционных систем);
- прикладная система сконцентрирована на сервере (на клиентах, кроме программ-навигаторов, ничего нет);
- доступ к информации осуществляется через одну и ту же программу, не требующую локальных данных;
- устройство на рабочем месте целиком настраивается из центра, и нет необходимости выполнять какие-то дополнительные действия по его конфигурированию;
- если с устройством что-то происходит, то оно выключается, приносится другое, включается, и работа продолжается.

Одно из полезных качеств подобных систем – облегченное централизованное управление, причем не только серверной частью, но и рабочими местами. Сегодня уже можно говорить о централизованном конфигурировании каждого рабочего места, что на порядок упрощает и удешевляет администрирование информационной системы.

Таким образом, для упрощения администрирования выполнены три действия. Во-первых, созданы универсальные рабочие места, во-вторых, они выбраны очень простыми, в-третьих, предусмотрено централизованное управление серверным оборудованием и рабочими местами.

В подобных системах проще решается и вопрос информационной безопасности. Проблема безопасности сложна в первую очередь не тем, что сложны сами по себе задача и каждая отдельная подзадача обеспечения информационной безопасности, а тем, что задач много и они исключительно разнообразны. В клиент-серверных системах очень трудно обеспечить комплексное решение с одинаковым уровнем надежности, перекрывающее все прикладные системы. Все дело в том, что имеются разнообразные компоненты, разнородные задачи, различные уровни – от системно-технического до прикладного.

Переход к технологии Интернет качественно упрощает рассматриваемую задачу.

Во-первых, гораздо большая часть ресурсов централизована. Централизованными ресурсами не только легче управлять, но их и легче защищать.

Во-вторых, внешние интерфейсы оказываются унифицированными, стандартными. Способов взаимодействия удаленного рабочего места с центральным сервером оказывается очень немного. Не нужно более заботиться о сотнях или даже тысячах приложений на компьютерах-клиентах и для каждого из них решать задачу защиты взаимодействия клиента с сервером. Достаточно обеспечить стандартное решение для одного рабочего места, которое и будет стандартным для всех.

В качестве аппаратной платформы для проектирования, разработки, сопровождения и поддержки системы управления региональной системой образования предлагается платформа SUN SPARC фирмы Sun Microsystems под управлением операционной системы Solaris.

Систему предлагается создавать на базе использования развитых промышленно-действующих лицензионных программных продуктов компании Oracle, предоставляющей полную линейку продуктов для решения обширного круга задач. В нее входят системы управления базами данных, сервера приложений, средства проектирования и разработки, системы поддержки принятия решений, системы обеспечения безопасности хранимых данных и др.

Спецификация моделей бизнес-процессов осуществляется на основе единых технологий и инструментальной среды разработки CASE-технологии ORACLE.

В качестве аппаратной платформы для клиентов информационной системы могут использоваться персональные компьютеры, подключенные к Интернет с программным обеспечением MS Windows 95 SR1 / 98 / ME / NT / 2000, MS Internet Explorer 4.01 или выше или Netscape Communicator 4.76 или выше.

АЛГОРИТМЫ ВЫБОРА СОЕДИНЕНИЙ В ПЛИС

Д.А. Гончаров, В.И. Скорубский

В статье приводятся свойства блока коммутации ПЛИС MAX7000 и ограничения, которые накладываются на этот блок. Ставится задача по разводке сигналов через блок коммутации, приводится алгоритм решения этой задачи. Кратко, для последующей работы, приводится структура блока коммутации для ПЛИС FLEX10K.

Система проектирования MAXPLUS является общепризнанной, надежной и определяет стандарты в технологии построения схем цифровых устройств [1].

Независимо от реализации в конкретной ПЛИС семейства MAX или в других семействах, технология позволяет создать и отладить корректное описание в графической и текстовой форме на основе принятых в этой области стандартов (AHDL, VHDL, VERILOG).

Практическое применение системы показывает, что здесь требуется определенная поддержка (помощь) для получения необходимого качества, а часто и принципиальной реализации конкретного устройства.

В связи с этим представляет интерес создание модели системы управления качеством проектирования. Модель можно представить черным ящиком, в котором выделяются:

- входные данные или описание схемы на стандартном языке или в графической форме;
- управляющие воздействия, позволяющие локально влиять на различные этапы проектирования;
- выходные данные – в виде конкретного размещения схемы в БИС;

Степень влияния управляющих параметров выявляется экспериментальными исследованиями по методике планирования эксперимента с учетом разделения технологического процесса на этапы:

- разбиения схемы на блоки (ячейки);
- разводка соединений (коммутация).

Выбор критерия качества позволяет упорядочить процесс проектирования и представить его как серию экспериментов, в которых оптимизируется некоторая выбранная в качестве критерия функция по одному из наиболее существенных на данном этапе входных параметров.

Если предположить, что зависимость монотонная, то, выбирая параметры в соответствии со стратегией "золотого сечения" или простым делением пополам области определения значений аргументов, можно приблизиться к оптимальным решениям.

Вместе с тем неопределенность этой зависимости не всегда позволяет корректно спланировать процесс оптимизации. В большинстве случаев критерий качества двузначен. Он оценивается возможностью реализации входных данных в конкретной ПЛИС. В дальнейшем может потребоваться только частичная коррекция для достижения заданных временных параметров.

Таким образом, часто используемые критерии – сложность, размеры, количество и др. – не актуальны.

Опираясь на эти положения, ограничимся рассмотрением задачи проектирования на этапе разводки (коммутации) выделенных на предыдущих этапах логических ячеек.

Задача и модели упрощаются по сравнению с традиционной задачей разводки соединений на печатной плате и сводятся к расчетам, учитывающим конкретные жесткие ограничения на структуру соединений в ПЛИС. Эти ограничения, в свою очередь, сводятся к заданию системы (матрицы) программируемых точек соединений,

распределенных по определенным правилам в горизонтальные и вертикальные линии (координаты).

На рис. 1 приведена структура соединений ПЛИС.

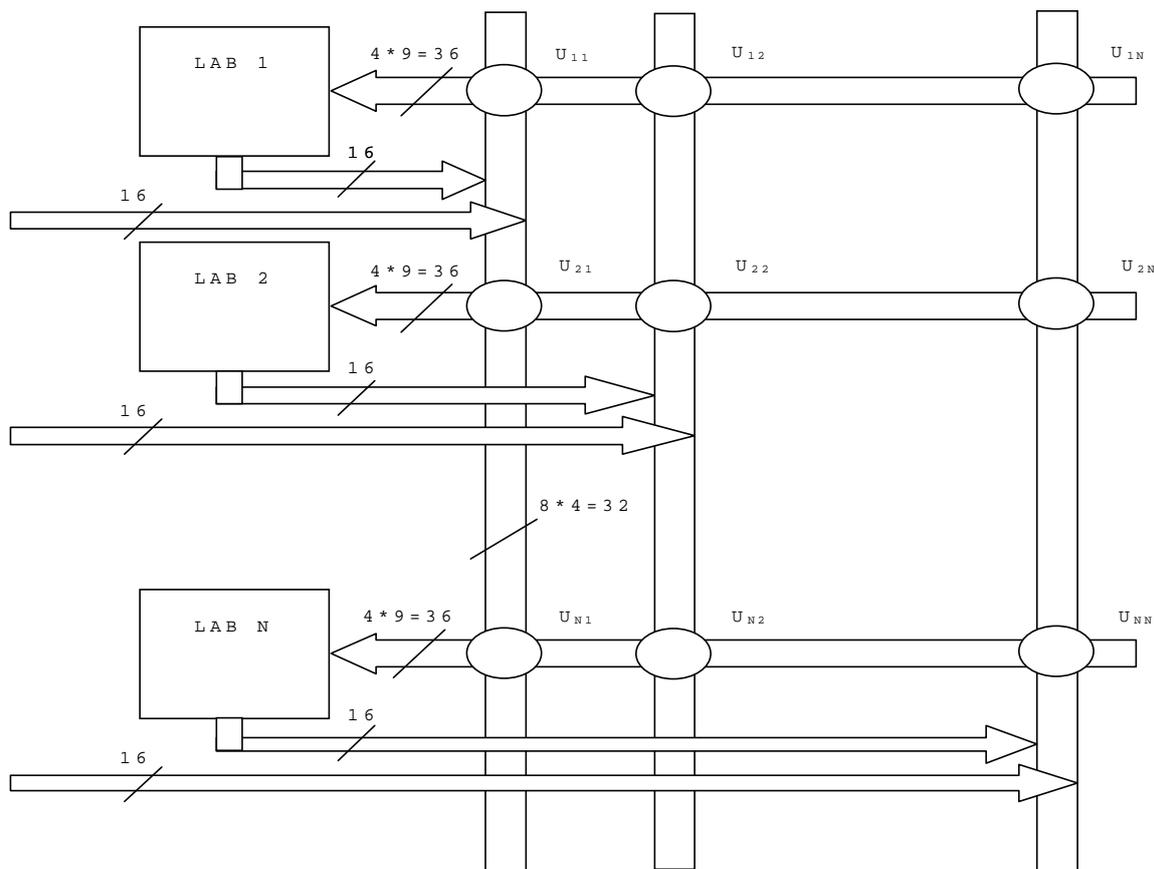


Рис. 1. Структура соединений ПЛИС MAX7000

MAX7000 – структура состоит из узлов коммутации U_{xy} на пересечении групп вертикальных линий (Y) и горизонтальных линий (X).

Далее сформулируем ограничения на коммутацию в узлах на входах и выходах блока коммутации (Programmable Interconnect Array) PIA.

1. Распределение входов (горизонтальные линии X) и выходов (Y), относящихся к ячейкам, зависит от размещения логики и формируется с учетом структуры U_{xy} .

На этапе синтеза и реализации логики можно влиять на их распределение:

- переименованием логических входов и выходов, нумерацией разрядов регистров – индивидуально и во множестве (возрастающая или убывающая нумерация);
- переупорядочением фрагментов схемы, которые на этапе синтеза заполняют ячейки последовательно – в порядке их записи в текстовом файле.

Можно выделить классы симметрии, образующие разбиение контактов ячейки на группы с эквивалентными перестановками.

Таким образом, можно принять свободное распределение контактов при выборе коммутации и в дальнейшем корректировать их распределение в случае неудачи, выбрать распределение с учетом симметрии и назначить их в одну группу линий X, связанных с одним общим узлом коммутации.

Внешние контакты БИС могут быть инварианты к нумерации или ограниченно закреплены, если они влияют на качество соединений на системном – более высоком – уровне иерархии.

Следовательно, существуют различные стратегии нумерации горизонтальных и вертикальных линий в PIA.

2. Ограничения по коэффициенту разветвления – каждая вертикальная линия (входной контакт PIA) может подключаться не более чем к N горизонтальным линиям (одноименные входы N логических ячеек)

3. Ограниченная схема соединений в узле коммутации U_{xy} .

Схема соединений задается мультиплексором частичной коммутации (MPC), который представляет собой матрицу элементов коммутации (по внешнему виду это матрица размещения 4 входов в 9 выходов).

LAB A	PA0-PA8	1-0				2-1						4-0				3-1		5-0			
	PA9-PA17		1-0				2-1					4-0					3-1		5-0		
	PA18-PA26			2-0			1-1		3-0				4-1							5-0	
	PA27-PA35				2-0			1-1		3-0				4-1							5-0
LAB B	PA0-PA8	1-0				2-1						4-0				3-1		5-0			
	PA9-PA17		1-0				2-1					4-0					3-1		5-0		
	PA18-PA26			2-0			1-1		3-0				4-1							5-0	
	PA27-PA35				2-0			1-1		3-0				4-1							5-0
		LC LAB A				LC LAB B				PIN LAB A				PIN LAB B				GLOBAL			
		1	5	9	13	17	21	25	29	4	8	13	18	41	37	32	27	2	1	44	43
		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-				
		4	8	12	16	20	24	28	32	7	12	17	21	38	33	28	24				

Рис. 2. Структура узла коммутации EPM7032

1-0		1-1		3-0		3-1		5-0																		
0	0	0	0	1	1	0	1	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	1	0	0	4	1	0	0	1	0	2	1	0	1	0	0	4	1	0	0	1	0	2	1	0	0
2	1	0	0	0	8	2	0	0	0	1	1	2	1	0	0	1	9	2	1	0	0	1	9	2	0	0
3	0	0	1	0	2	3	0	1	0	0	4	3	0	1	0	0	4	3	0	0	1	0	2	3	0	0
4	1	0	0	0	8	4	0	0	0	1	1	4	0	0	1	0	2	4	0	1	0	0	4	4	0	0
5	0	0	0	1	1	5	1	0	0	0	8	5	1	0	0	0	8	5	0	0	0	1	1	5	0	0
6	0	1	0	0	4	6	0	0	1	0	2	6	0	0	0	1	1	6	1	0	0	0	8	6	0	0
7	0	0	1	0	2	7	0	1	0	0	4	7	0	0	1	0	2	7	0	1	0	0	4	7	0	0
8	0	0	0	0	0	8	0	0	0	0	0	8	0	0	0	0	0	8	0	0	0	0	0	8	0	1
				30						30						30						30				
2-0		2-1		4-0		4-1																				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	1	0	0	4			
1	1	0	0	0	8	1	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0			
2	1	0	0	0	8	2	0	0	0	1	1	2	0	0	0	0	0	2	0	0	0	0	0			
3	0	1	0	0	4	3	0	0	1	0	2	3	0	0	0	1	1	3	1	0	0	0	8			
4	0	1	0	0	4	4	0	0	1	0	2	4	0	1	0	0	4	4	0	0	1	0	2			
5	0	0	1	0	2	5	0	1	0	0	4	5	0	1	0	0	4	5	0	0	1	0	2			
6	0	0	1	0	2	6	0	1	0	0	4	6	1	0	0	0	8	6	0	0	0	1	1			
7	0	0	0	1	1	7	1	0	0	0	8	7	0	0	1	0	2	7	0	1	0	0	4			
8	0	0	0	1	1	8	1	0	0	0	8	8	1	0	0	1	9	8	1	0	0	1	9			
				30						30						30						30				

Рис. 3. Мультиплексоры частичной коммутации EPM7032

На рис. 2 приведена структура узла коммутации ПЛИС EPM7032LC44-7, а на рис. 3 – мультиплексоры частичной коммутации ПЛИС EPM7032LC44-7.

Таким образом, горизонтальные и вертикальные линии должны быть разделены на классы с учетом этих ограничений.

Разбиение может быть инвариантно к перестановкам или связано, тогда достаточно проверить ограничения 9×4 .

Матрица размещения состоит из 4 различных сочетаний $C(2, 9)$, следовательно допускается не более двух вариантов выбора соединений для каждой линии по вертикали.

По горизонтали в одной матрице возможно альтернативное подключение не более одной вертикальной линии, но таких линий не более 4 (для EPM7032LC44-7). Следовательно, при отсутствии выбора достаточно контролировать только это ограничение.

Схему соединений для одного мультиплексора можно представить двудольным графом (X,Y) [2] связей между 4 вертикальными линиями и 9 горизонтальными.

Например, для мультиплексора 1-0 граф имеет вид рис. 4.

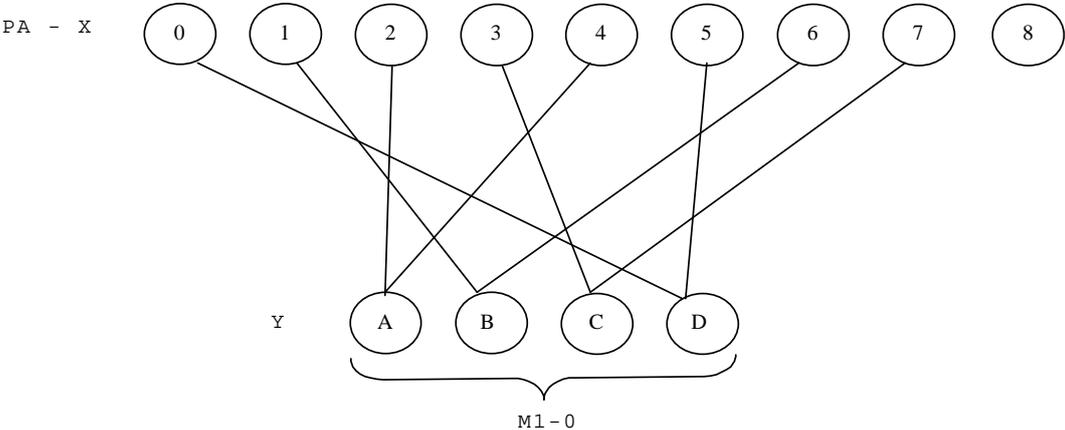


Рис. 4. Двудольный граф связей для MPC 1-0

Требуется выбрать такие пары вершин X и Y, чтобы они не имели общих вершин. Две пары вершин X с одной общей вершиной Y назовем связанными. Построим граф, отражающий связанность вершин, и каждую дугу в нем пометим соответствующим обозначением вершины Y, которая их связывает (рис. 5).

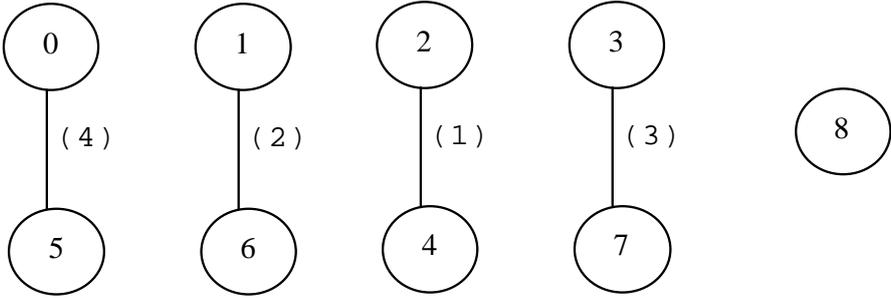


Рис. 5. Граф связности для MPC 1-0

Здесь можно визуально найти допустимые соединения последовательным выбором вершин X и одного ребра, отмечающего связи между вершинами, например, множество $Z=\{0(4), 1(2), 2(1), 3(3)\}$.

Всего существуют 2^4 вариантов выбора, и, следовательно, могут быть учтены и некоторые ограничения по входам и выходам, например, закрепление контактов X и Y.

Построим распределение для MPC 1-0 и 2-1. Двудольный граф связей и граф связности для MPC 1-0 и 2-1 приведены на рис. 6, 7.

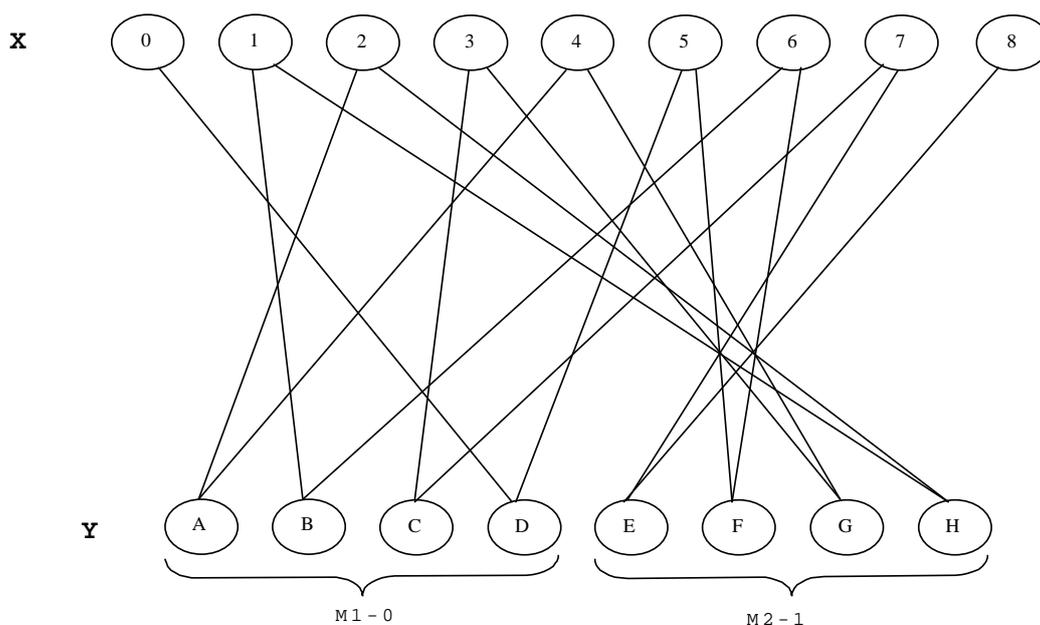


Рис. 6. Двудольный граф связей для MPC 1-0 и 2-1

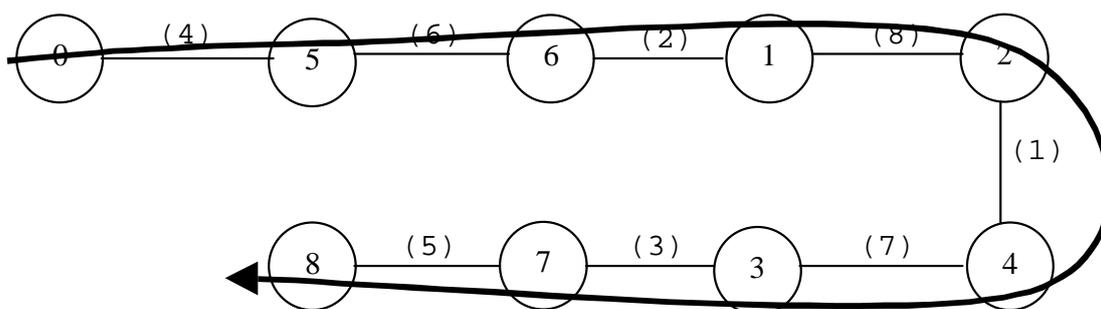


Рис. 7. Граф связности для MPC 1-0 и 2-1. Выбор соединений

Вершины графа обозначают линии X первой строки матрицы PIA, дуги помечены вершинами Y (по порядку 1, 2, 3, ... 8 для двух выбранных групп вертикальных линий).

В данном случае максимальное множество Z формируется последовательным выбором, начиная с вершины 0:

$$Z = \{0(4), 5(6), 6(2), 1(8), 2(1), 4(7), 3(3), 7(5)\}.$$

Если вертикальных групп N (для MAX7032 N=5) и каждая группа объединяет Y линий, то должно быть $\Sigma Y \leq 9$ из $4 \cdot (N-1) + 1$ возможных вариантов.

Приведем пример, когда ΣY должна быть меньше 9.

1. Не размещается больше 8.

Построим распределение для MPC 1-0 и 3-1.

$$Y = \{(1), (2), (3), (4), (13), (14), (15), (16)\}$$

Дуги MPC 1-0 помечены 1-4, а для MPC 3-1 13-16. Двудольный граф можно представить в следующем виде:

$$G = \{0(4), 1(2,15), 2(1,13,16), 3(3,15), 4(1,14), 5(4,16), 6(2,13), 7(3,14), 8(-)\}.$$

Граф связности приведен на рис. 8.

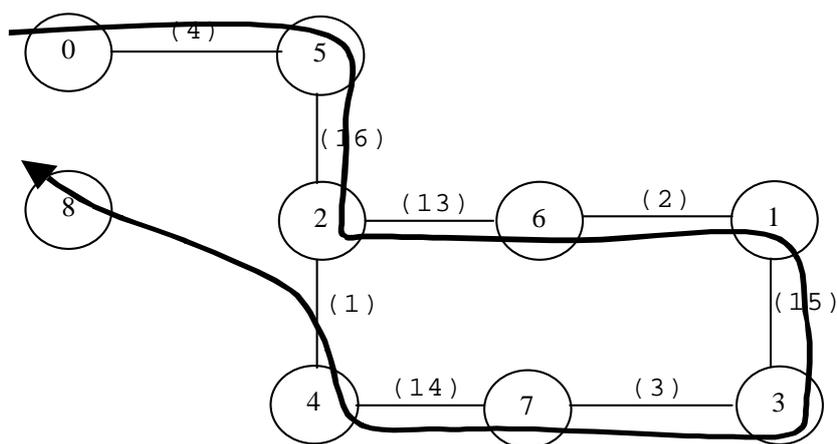


Рис. 8. Граф связности для MPC 1-0 и 3-1. Выбор соединений

Максимальное множество Z формируется последовательным выбором, начиная с вершины 0:

$$Z = \{0(4), 5(16), 2(13), 6(2), 1(15), 3(3), 7(14), 4(1), 8(-)\}.$$

Если в множество Y добавить (5) из MPC 2-1, то двудольный граф будет выглядеть следующим образом:

$$G = \{0(4), 1(2,15), 2(1,13,16), 3(3,15), 4(1,14), 5(4,16), 6(2,13), 7(3,5,14), 8(5)\}.$$

Граф связности приведен на рис. 9.

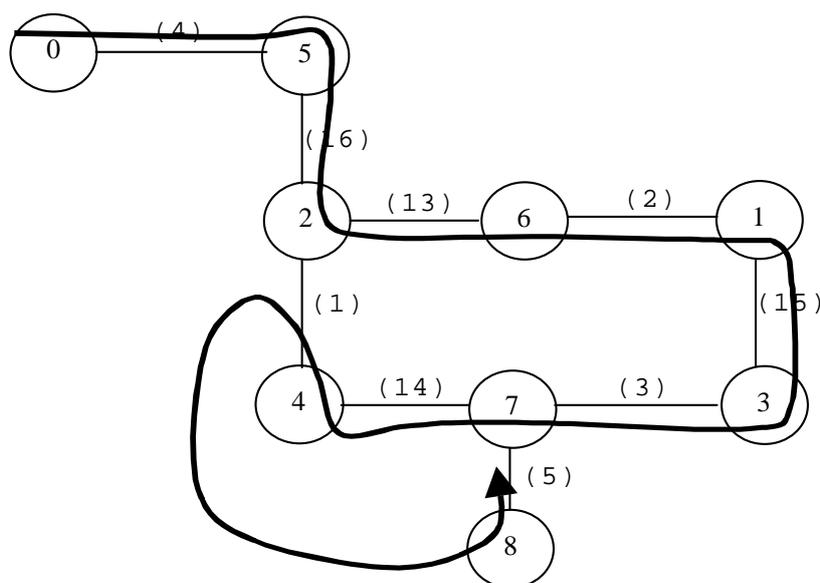


Рис. 9. Граф связности для MPC 1-0, 3-1 и (5) из 2-1. Выбор соединений

Максимальное множество Z формируется последовательным выбором, начиная с вершины 0:

$$Z = \{0(4), 5(16), 2(13), 6(2), 1(15), 3(3), 7(17), 4(1), 8(5)\}.$$

Если в множество Y добавить (6) из MPC 2-1, то двудольный граф будет выглядеть следующим образом:

$$G = \{0(4), 1(2,15), 2(1,13,16), 3(3,15), 4(1,14), 5(4,6,16), 6(2,6,13), 7(3,14), 8(-)\}.$$

Граф связности приведен на рис. 10.

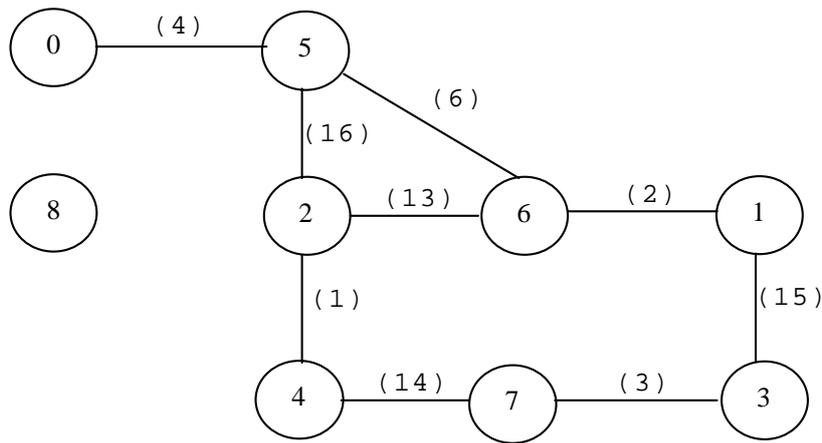


Рис. 10. Граф связности для MPC 1-0, 3-1 и (6) из 2-1

Из графа видно, что не все связи могут быть выбраны, так как задействованных вершин 8, а дуг, которые необходимо развести, 9. Это также видно из того, что одна вершина осталась не задействована, а для разводки 9 связей необходимо 9 вершин.

То же самое произойдет, если добавить к множеству связей Y (7), (8) из MPC 2-1, (10), (11) из MPC 4-0.

2. Не размещается больше 6.

Во множество Y входят следующие связи:

$Y = \{(3), (5), (6), (7), (9), (14)\}$.

Двудольный граф можно представить в следующем виде:

$G = \{0(-), 1(-), 2(-), 3(3,7), 4(7,14), 5(6), 6(6,9), 7(3,5,14), 8(5,9)\}$.

Граф связности приведен на рис. 11.

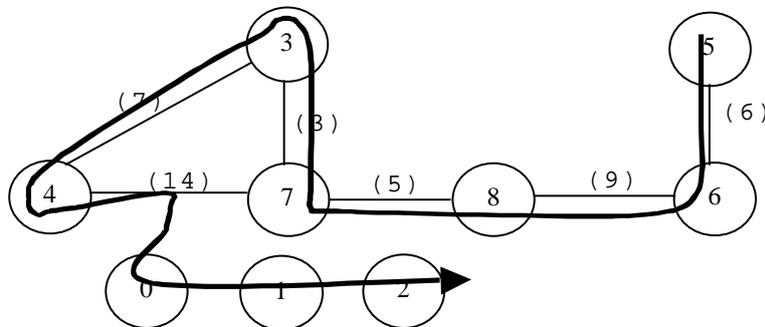


Рис. 11. Граф связности. Выбор соединений

Если во множество Y добавить (12), то двудольный граф будет выглядеть следующим образом:

$G = \{0(-), 1(-), 2(-), 3(3,7,12), 4(7,14), 5(6), 6(6,9), 7(3,5,14), 8(5,9,12)\}$.

Граф связности приведен на рис. 12.

Из графа видно, что не все связи могут быть выбраны, так как задействованных вершин 6, а дуг, которые необходимо развести, 7.

То же самое произойдет, если добавить к множеству связей Y (10).

В дальнейшем с ростом сложности матрицы коммутации необходимо переходить к алгоритмическому методу выбора соединений. Задача имеет следующую общую математическую постановку. Множество дуг, смежных с вершинами в двудольном графе, состоят из ребер, одно из которых должно быть выбрано для соединений.

Следовательно, в множество всех ребер S выделены собственные подмножества ребер $S_i \subset S$ такие что $\sum S_i = S$. Из этих подмножеств должны быть выбраны различные

связи. Говорят, что эти элементы представляют подмножества, и совокупность всех таких элементов, выбранных для всех подмножеств, называют системой различных представителей СРП.

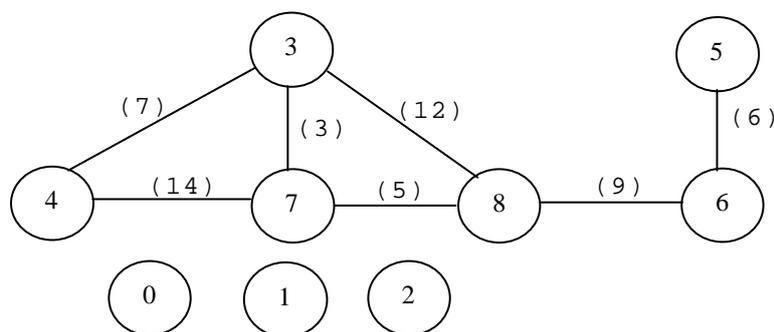


Рис. 12. Граф связности

Теорема Холла [2] содержит необходимые и достаточные условия существования СРП. Условия, по существу, опираются на алгоритм выбора СРП, сформулированный Холлом в виде доказательства теоремы.

Описание алгоритма очень расплывчато, и в нем есть много непонятных моментов. Поэтому пришлось придумывать собственный алгоритм выбора СРП. Упрощенная блок-схема алгоритма представлена на рис. 13.

Исходные данные представлены в виде массива M , где строки – это представители, а столбцы – возможные места размещения. Массив состоит из 0 и 1. Единица говорит о том, что представитель (строка) может размещаться на данном месте (столбце).

Поиск осуществляется последовательно для каждой строки. После выбора строки проверяется, есть ли для этого представителя место. Место ищется из множества столбцов, помеченных единицами и не занятых другими представителями. Если такой столбец найден, то он помечается занятым, т.е. ему присваивается минус единица. После этого переходим к следующему представителю.

Если место не найдено, т.е. все возможные варианты-столбцы заняты, то происходит дальнейший поиск. Для этого используется вспомогательный массив $M1$. В него копируется текущая строка, а в массив координат XU заносятся координаты (строка-столбец) единиц строки. Переменной $Index$ присваивается значение 1. Эта переменная показывает уровень вложенности поиска.

Далее осуществляется поиск по каждому столбцу из XU значения минус единица в массиве M и присвоение в массиве $M1$ этой координате значения следующего уровня вложенности. Эти координаты тоже заносятся в массив координат.

После этого идет поиск по строкам массива M значений 1 и также присвоение найденным координатам значения следующего уровня вложенности.

На текущем уровне вложенности проверяется, заняты ли найденные столбцы. Если столбец занят, то поиск "столбец-колонка" продолжается. Если столбец свободен, то данная координата помечается занятой (минус единицей) и происходит обратный проход по уровням вложенности до нулевого уровня с соответствующими заменами единицы на минус единицу и наоборот.

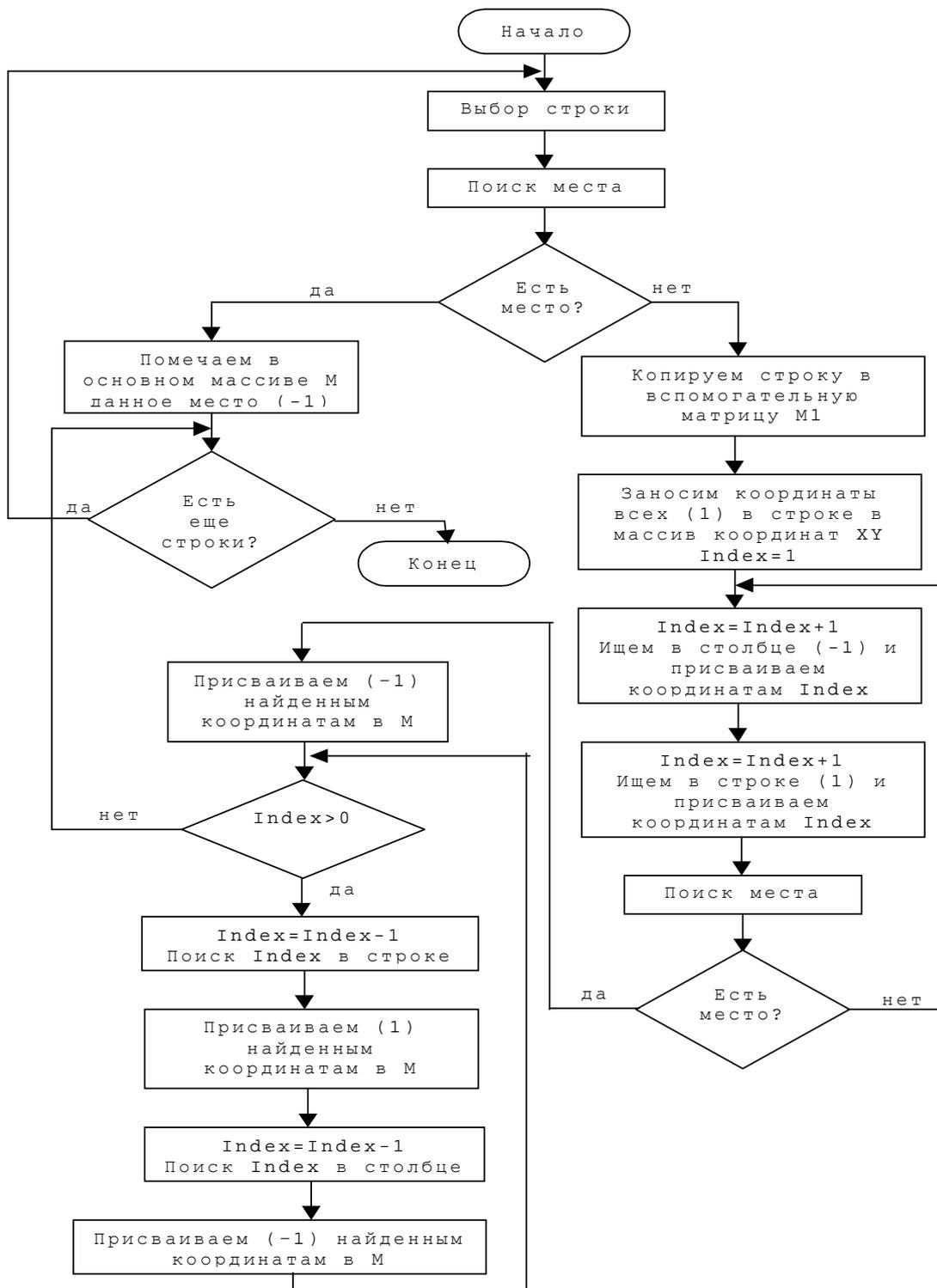


Рис. 13. Блок-схема алгоритма выбора СРП

PIA для ПЛИС EPM7032LC44-7 можно рассматривать как простейший случай одномерной коммутации, где ячейки и MPC лежат в одной плоскости и рассматриваются только параллельные схемы соединений. В случае 2, 3-мерной коммутации рассматриваются несколько последовательно соединенных MPC. Например, PIA ПЛИС FLEX10K обеспечивают двумерную коммутацию, которую можно представить в следующем упрощенном виде (рис. 14).

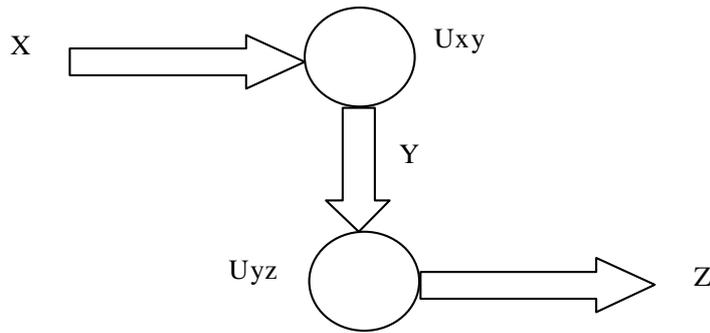


Рис. 14. Двумерная коммутация ПЛИС

Контакты X ячеек одного слоя коммутируются через МРС с множеством промежуточных контактов Y , которые через другую последовательную МРС переключаются на контакты Z и подключаются к ячейкам другого слоя.

Коммутация в этом случае сводится к построению двух связанных по входам-выходам двудольных графов и построением СРП для графов (X, Y) и (Y, Z) .

На рис. 15 показана обобщенная структура соединений ПЛИС FLEX10К.

Таким образом, рассмотрена математическая модель среды коммутации PIA для ПЛИС и приведено решение задачи коммутации контактов ячеек, позволяющие контролировать распределение соединений и корректировать автоматическую разводку.

Литература

1. Мелих В.Ф., Филлипов А.С. Обзор ПЛИС фирмы ALTERA.
2. Кристофидес Н. Теория графов. Алгоритмический подход. М.: Мир, 1978.

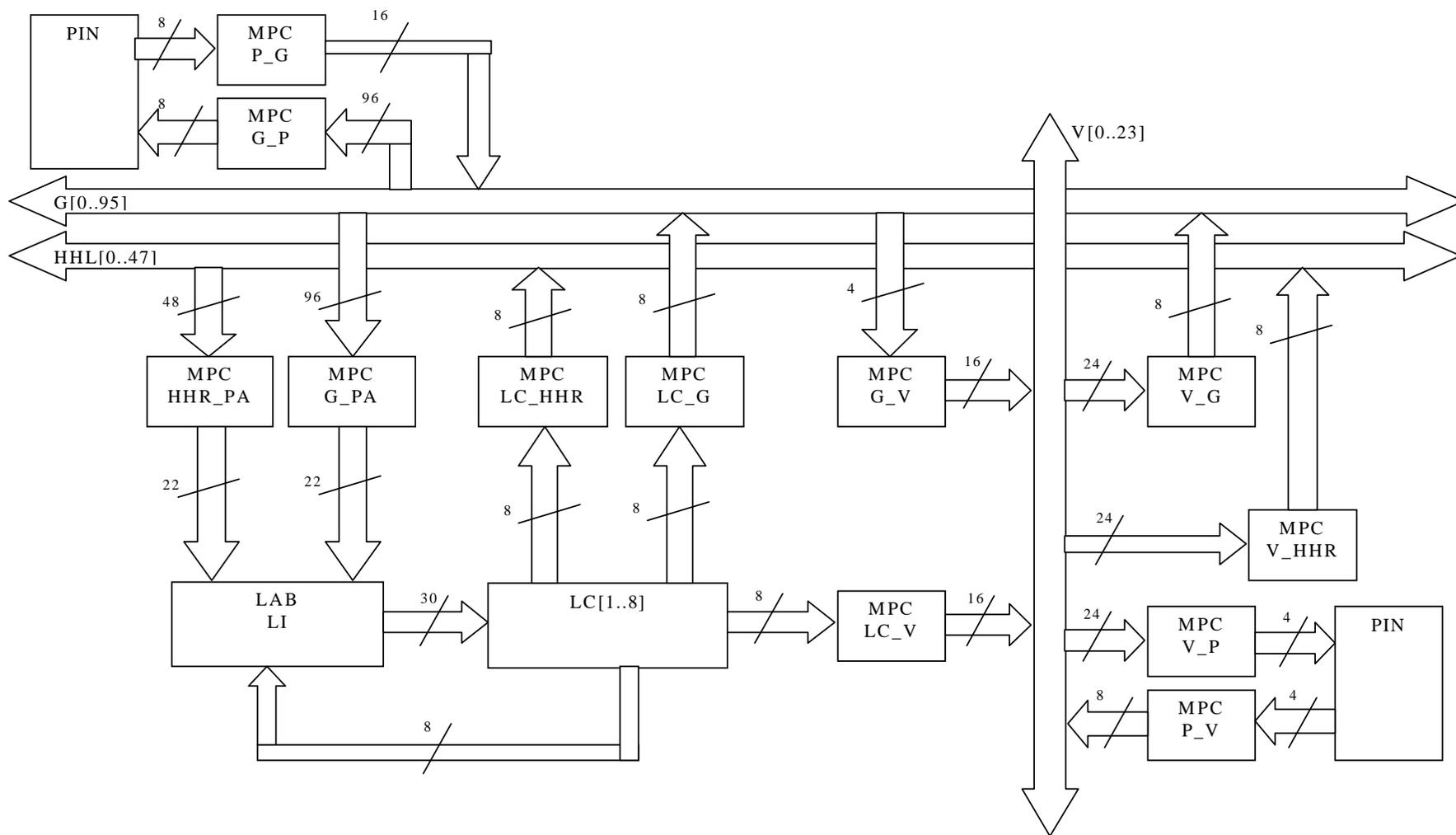


Рис. 15. Структура соединений ПЛИС FLEX10K

СЖАТИЕ ИЗОБРАЖЕНИЙ НА ОСНОВЕ ВЫДЕЛЕНИЯ ЛОКАЛЬНЫХ ОДНОРОДНЫХ ОБЛАСТЕЙ

М.В. Гришин, А.А. Ожиганов, А.Ю. Тропченко

В настоящее время существует достаточно много алгоритмов сжатия изображений. Они осуществляют сжатие либо в пространственной, либо в частотной областях изображения. Наиболее яркими примерами пространственного сжатия изображений являются алгоритмы РСХ, GIF, а частотного сжатия – JPEG.

В пространственной области нельзя добиться достаточно большого коэффициента сжатия, поэтому в настоящее время большинство алгоритмов сжатия работают в частотной области изображения.

В подобных алгоритмах можно выделить три основных шага:

1. применение обратимых дискретных ортогональных преобразований к изображению;
2. выбор наиболее значимых частотных коэффициентов;
3. вторичное сжатие выбранных коэффициентов, например, арифметическим или статистическим алгоритмом сжатия.

В данной работе в качестве обратимого дискретного ортогонального преобразования выбрано вейвлет-преобразование [1]. Основная идея вейвлет-преобразования состоит в иерархическом разложении входного сигнала на последовательности так называемых базовых компонент с последовательно уменьшающимся разрешением и связанных с ними компонент детализации. На каждом уровне разложения базовая компонента и компонента детализации содержат информацию, необходимую для восстановления базового сигнала на следующем уровне с более высоким разрешением.

В рассматриваемых алгоритмах использовались два вида преобразования: преобразование Хаара и вейвлет-преобразование (5,3).

Преобразование Хаара использует следующие фильтры декомпозиции:

$$\tilde{h}_n = \left\{ \frac{1}{2}, \frac{1}{2} \right\}, \quad \tilde{g}_n = \left\{ \frac{1}{2}, -\frac{1}{2} \right\}.$$

Вейвлет-преобразование (5,3) использует следующую пару фильтров:

$$\tilde{h}_n = \left\{ -\frac{1}{8}, \frac{1}{4}, \frac{3}{4}, \frac{1}{4}, -\frac{1}{8} \right\}, \quad \tilde{g}_n = \left\{ \frac{1}{4}, -\frac{1}{2}, \frac{1}{4}, 0, 0 \right\},$$

где \tilde{h}_n – низкочастотный фильтр, а \tilde{g}_n – высокочастотный фильтр.

После применения двухуровневого вейвлет-преобразования к изображению I возможны несколько подходов для сжатия полученных вейвлет-коэффициентов.

Одним из наиболее очевидных способов сжатия вейвлет-коэффициентов является учет взаимосвязей между коэффициентами из различных субполос вейвлет-разложения. В высокочастотных субполосах имеются обычно большие области с нулевой или малой энергией. Области с высокой энергией повторяют от субполосы к субполосе свои очертания и местоположение.

Медленно изменяющиеся, гладкие области исходного изображения хорошо описывают низкочастотные (НЧ) вейвлет-базисы, что приводит к "упаковке" энергии в малом числе коэффициентов НЧ области. Впервые идея нуль-дерева была предложена А. Льюисом и Г. Ноулесом [2]. В их алгоритме применялась древовидная структура данных для описания вейвлет-коэффициентов (рис.1). Корневой узел дерева представляет коэффициент масштабирующей функции в самой НЧ области и имеет три отпрыска. Узлы дерева соответствуют вейвлет-коэффициентам масштаба, равного их высоте в дереве. Каждый из узлов имеет четыре отпрыска, соответствующих вейвлет-

коэффициентам следующего уровня и того же пространственного расположения. Низом дерева являются листовые узлы, не имеющие отпрысков.

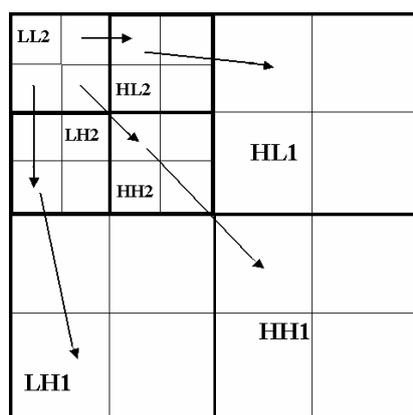


Рис 1. Кодирование вейвлет-коэффициентов нуль-деревом

Кодирование нуль-деревом основано на том, что, если коэффициент мал, его отпрыски на дереве зачастую тоже малы. Это объясняется тем, что значимые коэффициенты возникают вблизи контуров и текстур, которые локальны. Нетрудно увидеть, что это является разновидностью предсказания. Это предсказание можно свести к минимуму, предположив, что, если какой-либо коэффициент незначимый (т.е. он меньше по модулю некоторого порога Thr), то все его потомки также будут незначимыми. Дерево или subtree, которое содержит только незначимые коэффициенты, называется нуль-деревом.

Выбранные таким образом вейвлет-коэффициенты затем сжимаются с помощью вторичного алгоритма сжатия, в нашем случае это был алгоритм ZIP. Сравнение данного метода сжатия вейвлет-коэффициентов с другими методами приведено в таблицах 1 и 2.

В работе также рассматривалась и исследовалась модификация выше описанного метода сжатия вейвлет-коэффициентов. Сжатие ветвей, не являющихся нуль-деревом, происходит с использованием кодовой книги. В начальном состоянии есть одна "чистая" кодовая книга. Первая найденная ветвь, не являющаяся нуль-деревом, помещается в кодовую книгу, а на приемную сторону передается код данной ветви (поскольку ветвь была первой, то передается код 0). В дальнейшем любая найденная ветвь сравнивается со всеми ветвями, уже находящимися в кодовой книге. Пусть S – текущий размер кодовой книги. Если все коэффициенты данной ветви Br_i отличаются от какой-то ветви Br_n с кодом C , находящейся в кодовой книге, не более чем на порог Thr , то эти ветви считаются идентичными, и на приемную сторону передается код C . Если ни одна ветвь в словаре не совпала с кодируемой ветвью, то в словаре заводится новая запись для ветви Br_i и передается новый код $S+1$. Чтобы коды всех ветвей были однобайтовыми, размер кодовой книги был ограничен 250-ю значениями. Таким образом, если размер книги превысит 250, то будет заведена новая кодовая книга, и дальнейшая работа будет производиться только с ней. Эксперименты показали, что такой подход к кодированию вейвлет-ветвей очень эффективен. В большинстве случаев наблюдался большой процент попадания кодируемых ветвей в кодовую книгу. Сравнение такого подхода к кодированию вейвлет-коэффициентов с другими методами кодирования приведено в таблицах 1 и 2.

Еще одним из способов кодирования вейвлет коэффициентов, является метод шаблонно-блокового кодирования, который основан на высокой структурированности и разреженности вейвлет коэффициентов.

В данном методе каждая субполоса вейвлет-преобразования, за исключением низкочастотной составляющей, кодируется отдельно [3]. Субполоса разбивается на блоки размером 4×4 , которые заменяются кодом одного из шаблонов, наилучшим образом соответствующего этому блоку.

Поскольку высокочастотные составляющие вейвлет-преобразования хорошо описывают границы на изображении, то шаблоны также должны отражать различные границы для разных ориентаций (рис. 2).

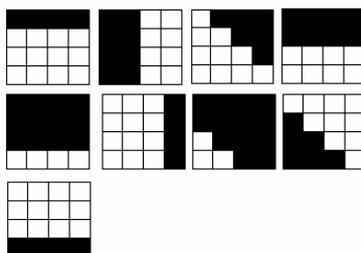


Рис. 2. Примеры шаблонов

Блоки делятся на однородные и граничные. Однородный блок – это блок, в котором все пиксели имеют одинаковую интенсивность. Граничный блок – это блок, который имеет одну границу, проходящую вдоль всего блока.

Для классификации блоков была разработана специальная классификационная схема, которая определяла, является ли блок однородным или граничным. Если блок является граничным, то он заменяется кодом одного из конечных шаблонов, наилучшим образом соответствующим ему. В работе использовались 250 различных шаблонов, которые наиболее часто встречались в ряде исследуемых изображений.

В конечном итоге, на приемную сторону передаются либо 2 числа, если блок однородный, либо 3 числа (индекс блока, наибольшая и наименьшая интенсивность в блоке), если он граничный. Естественно, что передаваемый поток данных также подвергнулся вторичному сжатию. Сравнение данного метода сжатия вейвлет-коэффициентов с другими методами приведено в таблицах 1 и 2.

Поскольку сжатие данных подразумевает учет статистических связей между элементами изображения, то применение обратимых дискретных ортогональных преобразований к отдельным областям изображения, а не к изображению в целом в некоторых случаях может открывать возможности для увеличения коэффициента сжатия.

В работе была предпринята попытка выделения естественных для изображения областей, т.е. изображение не разбивается на блоки размером 8×8 , как в алгоритме сжатия JPEG, а на изображении осуществляется поиск и выделение псевдооднородных сегментов, присущих данному изображению.

В процессе выделения сегментов участвуют 2 копии одного изображения: одна копия хранит оригинальные значения пикселей, а вторая используется для поиска сегментов.

Выделение сегментов можно описать тремя шагами:

1. фильтрация изображения для устранения яркостных всплесков(в качестве фильтра использовался медианный фильтр);
2. квантование изображения (уменьшение уровней градаций серого);
3. выделение полученных сегментов на квантованном изображении.

Остановимся подробнее на последнем шаге – выделении сегментов на изображении.

Классические методы сегментации основаны на использовании порога интенсивности. Если точки изображения имеют значения выше (ниже) некоторого порога, то они принадлежат области, иначе – не принадлежат. Эти методы имеют ряд недостатков. Во-первых, методы распознавания по уровню интенсивности дают плохие результаты, если имеется изменение интенсивности освещенности в пространстве изображения. Во-вторых, если внутри области наблюдается изменение интенсивности, то метод выделит несколько концентрических областей.

В связи с этими недостатками предлагается использовать несколько иной метод сегментации изображения.

В начале ищется точка $B_{\max}(X_{\max}, Y_{\max})$ максимальной интенсивности I_{\max} , эта точка считается стартовой (рис. 3). На следующем шаге k размечаются только те точки, которые имеют соседей из числа точек, размеченных на предыдущем шаге ($k-1$).

Точка считается принадлежащей области, если выполняются следующие условия:

- 1) если хотя бы одна из соседних точек принадлежит выделяемому сегменту;
- 2) если интенсивность точки попадает в определяемый на каждом шаге интервал интенсивностей $[I_{top}, I_{bottom}]$, определяемых как доля интенсивности I_{\max} точки B_{\max} ;

- 3) точка $B(i, j)$ может быть отмечена, если линия, проведенная из стартовой точки B_{\max} , лежащей на границе, построенной на предыдущих шагах области, к точке $B(i, j)$ не пересекает границ этой области. При выполнении алгоритма линия строится по алгоритму Бразенхейма.



Рис. 3. Выделение сегментов А и В

Таким образом, в результате работы алгоритма получается множество точек, составляющих выделяемый сегмент.

Этот сегмент окружается прямоугольной областью, фон области заполняется интенсивностью граничных точек сегмента. К полученному объекту применяется один из вышеописанных методов сжатия, и формируется пакет данных: параметры сегмента (координаты верхнего левого угла, ширина и высота) и сжатые данные. Затем полученный пакет сжимается вторичным алгоритмом сжатия.

Эксперименты показали, что не все вейвлет-преобразования можно применять к сегментам изображения. Некоторые вейвлет-преобразования при восстановлении изображения допускают проявления граничных артефактов, которые не позволяют правильно состыковать сегменты друг с другом. Сравнение методов сжатия изображений с помощью вейвлет-преобразования, применяемого как к целому изображению, так и к его областям приведено в таблицах 1 и 2.

Проведенные эксперименты показали, что применение вейвлет-преобразования к локальным однородным областям, а не к изображению целиком, в общем случае не дает выигрыша в сжатии изображения. Во-первых, это связано с ростом объема дополнительных хранимых данных, таких как ширина и высота области, а также координаты ее левого верхнего угла. Во-вторых, появляющиеся в некоторых случаях граничные артефакты при восстановлении изображения не позволяют точно состыковывать восстановленные области, что приводит к значительному искажению восстановленного изображения. Однако не исключено, что подобные технологии можно будет использовать для создания черновых или предпросмотровых вариантов изображений при их распространении через Интернет.

Метод сжатия	До вторичного сжатия	После вторичного сжатия (zip)
Исходный файл BMP	227k	137k
JPEG	-----	25k
Сжатие однородных локальных областей		
Метод нуль-дерева	101k	50k
Метод шаблонно-блочного кодирования	83k	34k
Сжатие целого изображения		
Метод нуль-дерева	67k	38k
Метод шаблонно-блочного кодирования	54k	24k
Метод нуль-дерева с кодовой книгой	59k	22k

Таблица 1. Объем сжатых данных для различных методов сжатия изображений

Метод сжатия	MSE	SNR	PSNR
JPEG	89.9	22.7	28.6
Сжатие однородных локальных областей			
Метод нуль-дерева	185	18.5	24.4
Метод шаблонно-блочного кодирования	514.8	15.3	21
Сжатие целого изображения			
Метод нуль-дерева	103.0	19.3	25.0
Метод шаблонно-блочного кодирования	357	16.8	22.6
Метод нуль-дерева с кодовой книгой	86.7	22.7	28.7

Примечание: **MSE** – среднеквадратичная ошибка,
SNR – соотношение сигнал/шум,
PSNR – соотношение максимального значения сигнала к MSE

Таблица 2. Сравнение качества сжатого изображения

Литература

1. Дьяконов В.П. От теории к практике. Вейвлеты. М.: Солон-Р, 2002.
2. Воробьев В.И., Грибунин В.Г. Теория и практика вейвлет-преобразований. СПб: Военный университет связи, 2000.
3. Keissarian F., Daem M.F. Block pattern coding of HVS-based on wavelets for image compression”. // SPEE. Volume 4472.

МОДЕЛЬ АБСТРАКТНЫХ ФУНКЦИОНАЛЬНЫХ БЛОКОВ

И.В. Елькин, П.В. Кустарев

Рассмотрены существующие и перспективные методики проектирования распределенных информационно-управляющих систем (РИУС) в части создания архитектурных моделей. Описанные подходы являются в достаточной степени формализованными, закреплены в интернациональных стандартах и рассматриваются в качестве объединяющей платформы для сообщества разработчиков РИУС.

Современные подходы к проектированию РИУС

Проектирование информационно-управляющих систем, в том числе и распределенных (РИУС), построенных на основе цифровых сетей – это сложный процесс, включающий в себя последовательность фаз и этапов, ведущих от прикладной задачи к реализации системы. Перечислим основные шаги при проектировании.

1. Фаза 1 – создание и анализ абстрактного описания прикладной задачи, решение которой возлагается на систему:
 - a. формулирование требований к системе в формальном или неформальном виде;
 - b. разработка высокоуровневой архитектуры системы.
2. Фаза 2 – разработка по данному описанию реализации системы: протоколов, топологий, аппаратуры, программных функций и др.:
 - a. разработка аппаратной части системы;
 - b. разработка программного обеспечения системы.

Типовые подходы к этапам первой фазы – использование модели программируемых логических контроллеров (ПЛК, PLC) с централизованным управлением (стандарт IEC 61131) или использование модели распределенного управляющего приложения на базе функциональных блоков (ФБ) с элементарными функциями (стандарт IEC 61804). Рассмотрим подробнее каждый из этих подходов.

Проектирование на основе PLC

При проектировании РИУС на основе программируемых логических контроллеров идеологически алгоритм управления является централизованным и сосредоточен в одном или нескольких контроллерах большой вычислительной мощности, которые связаны с сетью распределенных датчиков и исполнительных устройств.

Программирование PLC осуществляется на специальных языках в рамках специальной программной модели – виртуальной машины, которая изолирует разработчика от реальной аппаратуры и определяет структуру вычислительного процесса, правила исполнения и взаимодействия задач, типы данных, порядок взаимодействия PLC и УСО.

Стандартом, определяющим архитектуру, параметры аппаратных средств, организацию коммуникационной подсистемы PLC, является IEC-61131. В разделе IEC 61131-3 этого стандарта приводится описание синтаксиса и семантики нескольких языков программирования PLC – как текстовых, так и графических (SFC, LD, FBD, ST, IL). Используя многообразие языковых средств, пользователь может выбирать наиболее удобные для него языковые конструкции для описания логики.

К достоинствам модели на основе PLC можно отнести возможность формального описания прикладной задачи на аппаратно-независимом языке, возможность перепрограммирования и конфигурирования функциональности системы конечным пользователем, набор различных языков для решения разных задач и хорошоую

структурированность получаемого управляющего ПО, что облегчает дальнейшее усовершенствование системы, а также допускает повторное использование кода. Существуют инструментальные средства, охватывающие весь жизненный цикл разработки системы.

Основным недостатком подхода является то, что получаемое решение является централизованным, т.е. алгоритм управления сосредоточен в пределах одного промышленного контроллера, что приводит к ограничениям масштабирования системы производительностью PLC, неэффективности построения распределенных приложений, потенциальному снижению надежности. Кроме того, стоимость PLC достаточно высока.

Системы с распределенным управлением

Системы с распределенным управлением (Distributed Control Systems – DCS) основаны на принципе децентрализованного управления. При этом прикладной алгоритм реализуется в виде распределенного приложения, отдельные модули которого исполняются параллельно на множестве взаимосвязанных интеллектуальных контроллеров. Данная архитектура определена в стандарте IEC 61804, обобщающем аналогичные решения, разработанные в рамках различных технологий РИУС, таких как Foundation Fieldbus, Profibus-PA и ряда других.

Распределенные приложения собираются из функциональных блоков, которые можно гибко использовать и конфигурировать без традиционного программирования. Функциональные блоки (ФБ) являются стандартизированными пакетами функций управления – например, блок аналогового входного сигнала, аналогового выходного сигнала и ПИД-регулирования. Другими стандартными функциональными блоками управления являются дискретный вход, дискретный выход, селектор сигналов, ручной ввод, регулятор смещения/усиления сигнала и регулятор соотношения.

Основное достоинство данного подхода состоит в том, что получаемое решение является распределенным. Это позволяет использовать множество относительно маломощных и недорогих контроллеров для создания распределенного алгоритма управления, соответственно, возрастает масштабируемость системы. Функциональные блоки допускают динамическое конфигурирование в соответствии с конкретными условиями эксплуатации.

Недостаток же состоит в том, что функциональные блоки представляют собой достаточно примитивные пакеты функций управления (например, дискретные, аналоговые вводы/выводы), имеющие минимальные возможности их реконфигурирования, вследствие чего остается необходимость в низкоуровневом программировании при создании программной составляющей системы.

Рассмотренные подходы также имеют ряд общих недостатков, таких как отсутствие формальной методики проектирования и вытекающие отсюда несовместимость инструментальных средств от различных производителей, а также сложность интеграции различных программных решений, полученных с помощью этого инструментария.

Развитие идеи ФБ – абстрактные ФБ IEC 61499

Решение проблем, указанных для обоих подходов, возлагается на обновленную модель описания, названную моделью "абстрактных функциональных блоков" и разрабатываемую в рамках проекта стандарта IEC 61499 (рис. 1). Понятие "абстрактные" отличает их от ФБ стандарта IEC 61804, являющихся примитивами, жестко связанными со своей реализацией и ограниченными ее возможностями.

"Абстрактные ФБ" совмещают лучшее, что есть в ранних моделях: они могут представлять сколь угодно высокоуровневые элементы алгоритма, как в языке FBD стандарта IEC-61131-3, сохраняя ориентацию на распределенную реализацию, как в модели IEC 61804.

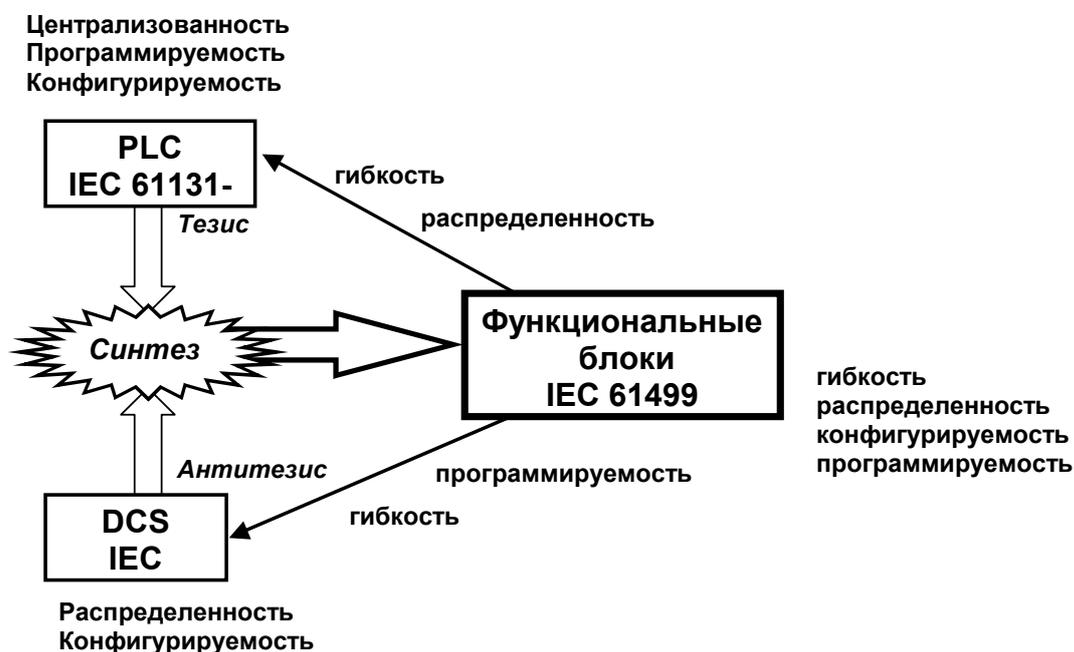


Рис. 1. Развитие стандартов проектирования РИУС

Функциональные блоки служат основой для построения управляющих приложений, распределенных между отдельными узлами управляющей сети. Распределенное управляющее приложение представляет собой совокупность функциональных блоков, связанных управляющими и информационными связями. Каждый блок отвечает за выполнение какой-либо части управляющего алгоритма (например, управление клапаном или некоторой промышленной установкой), инкапсулируя в себе алгоритм управления и предоставляя внешний интерфейс. Таким образом, здесь имеет место объектно-ориентированный подход к созданию приложений.

Существует возможность создания своих блоков и их последующего использования при проектировании, а также расширения функциональности существующих блоков.

Основные принципы построения систем на основе абстрактных функциональных блоков

Рассмотрим основные принципы построения систем на основе абстрактных функциональных блоков.

Управляющие приложения, построенные в соответствии со стандартом, имеют распределенный характер, базирующийся на декомпозиции функции управления на несколько ФБ, закрепленных за разными физическими устройствами (рис. 2). При этом сохраняются абстрактные блоки, не связанные с реализацией. Приложение состоит из одного или нескольких экземпляров таких блоков, соединенных связями по данным и событиям.

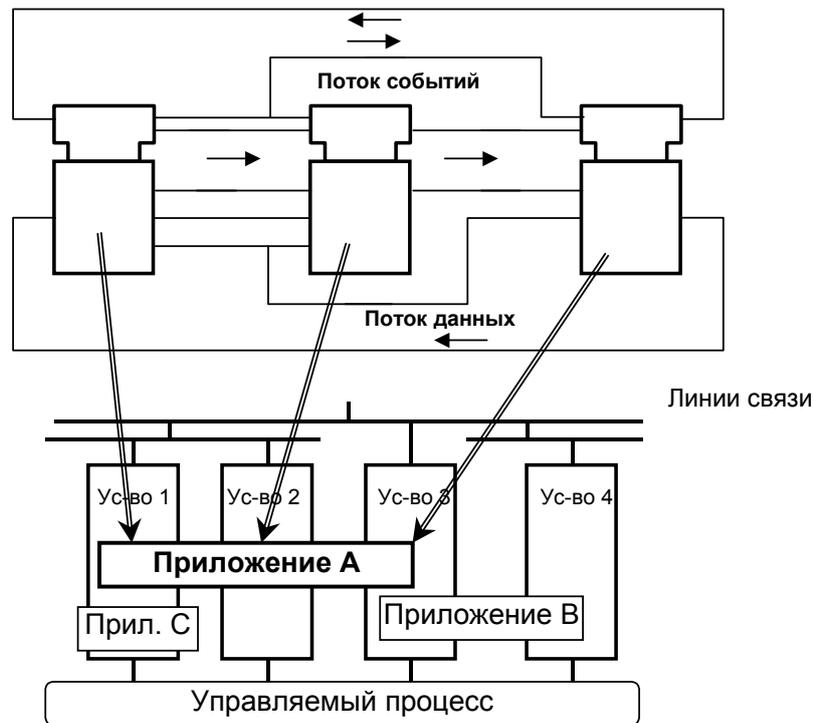


Рис. 2. Структура распределенного приложения на основе ФБ

Все функциональные блоки относятся к тому или иному заранее определенному типу и имеют четкие интерфейсы данных и событий между ФБ (рис. 3). Стандарт предоставляет как текстовые, так и графические средства объявления интерфейсов типов ФБ, а также связи входов и выходов разных блоков между собой. На практике для описания типов данных используются языковые средства IEC 61131-3 или XML.

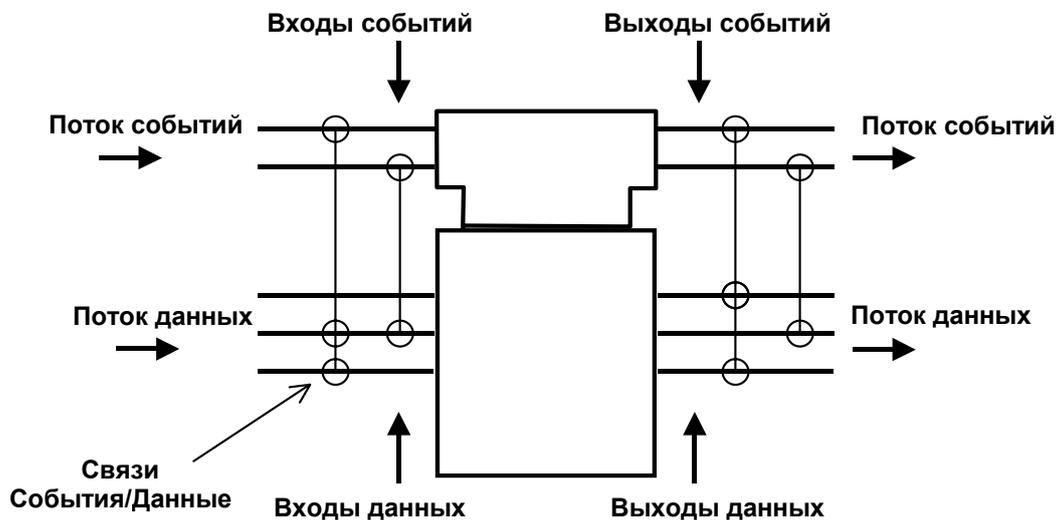


Рис. 3. Интерфейс функционального блока

Функционирование приложения основано на модели управления событиями как на уровне внутреннего алгоритма отдельного ФБ, так и на уровне распределенной системы в целом. По сути дела, приложение представляет собой управляемый событиями конечный автомат. Простые ФБ описываются объявлением внешнего интерфейса, диаграммы управления исполнением (конечный автомат) и алгоритмами, выполняемыми в соответствии с этой диаграммой. Алгоритмы могут быть написаны на

языках IEC 61131-3 или других языках структурного программирования (например, C++, Java). Алгоритмы используют значения входов ФБ и внутренних переменных для вычисления выходных значений.

Стандарт определяет также служебные интерфейсы. Особый тип ФБ – сервисные интерфейсные ФБ (СИФБ). Они предоставляют стандартные интерфейсы для коммуникационных служб, служб ввода/вывода и службы операционной среды, таких как службы времени и синхронизации, функции инициализации приложений, создания экземпляров ФБ, их удаления, связывания и активации. Такие службы приобретают особенное значение в условиях увеличивающихся требований к гибкости системы.

Стандарт предоставляет возможности инкапсуляции и повторного использования алгоритмов-программ функционирования ФБ, создаваемых на языках IEC 61131-3 или ряде других. Кроме того, с помощью так называемых составных ФБ и подсистем приложений поддерживается инкапсуляция сетей ФБ (рис. 4). Таким образом, распределенное приложение может строиться из набора повторно используемых компонентов с четко определенным интерфейсом и функциональностью.

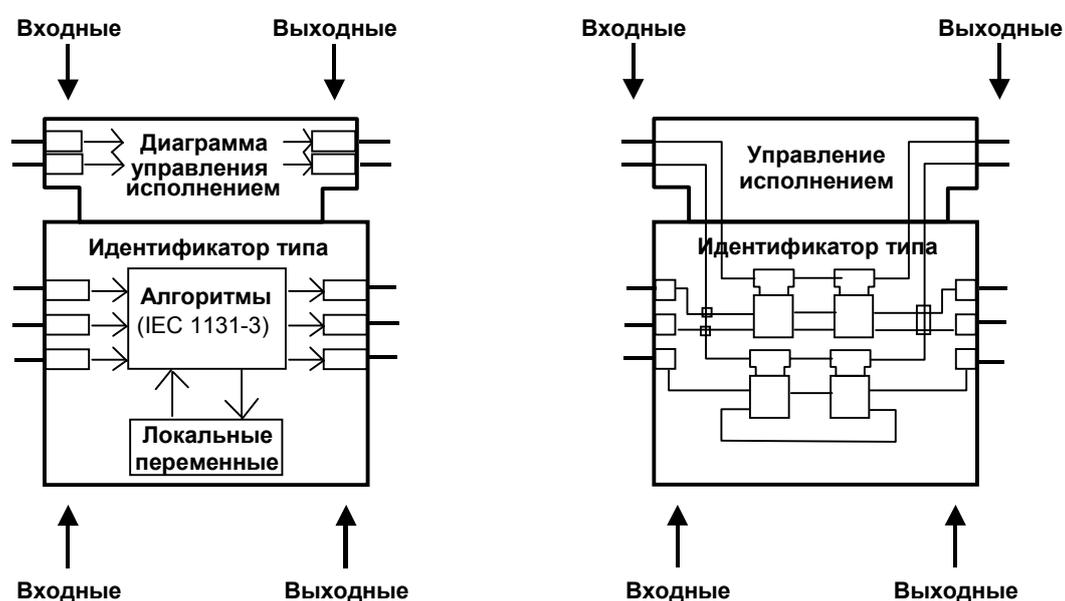


Рис. 4. Механизмы инкапсуляции

Также обеспечивается переносимость алгоритмов-программ. Данная особенность предполагает возможности использования разнородных сред разработки и оборудования от различных поставщиков. С этой целью во второй части стандарта IEC 61499-2 приведены схемы описания языка XML для стандартизованного обмена всеми библиотечными элементами IEC 61499, включая типы данных, типы ФБ, подсистем, устройств, ресурсов и конфигураций, что обеспечивает их переносимость, повторное использование и распространение

Проектирование на базе абстрактных ФБ

Рассмотрим теперь особенности процесса проектирования управляющих систем на основе стандарта IEC 61499.

Проектирование системы состоит в логическом описании прикладного алгоритма, заложенного в требованиях, как совокупности функциональных блоков, связанных между собой, независимо от конкретных аппаратных средств. На конечной стадии

проектирования осуществляется отображение отдельных блоков приложения на вычислительные узлы управляющей сети – построение распределенного решения.

В стандарте зафиксированы ключевые аспекты проектирования: унифицированы требования к функциональности инструментальных средств, правила их взаимодействия. Специальные профили соответствия описывают то, каким образом следует реализовывать модели стандарта, чтобы добиться возможности взаимодействия (interoperability) устройств от различных поставщиков, переносимости ПО (portability of software) между инструментальными средствами от различных поставщиков и конфигурируемости (configurability) устройств от различных поставщиков с помощью различных инструментальных средств.

Также унифицирован цикл проектирования и зафиксирована методология. В общем виде проектирование любой системы можно представить в виде итеративного спирального процесса, направленного на постепенное улучшение проектируемой системы. Каждая итерация состоит из следующих этапов:

1. оценивание (Evaluate),
2. улучшение (Improve),
3. эксплуатация (Operate).

Обобщенная методология, которая применяется в пределах одной итерации, включает следующие шаги.

1. Разработка и тестирование централизованной версии приложения с использованием одиночного устройства. На данном шаге разрабатываются функциональные блоки приложения (простые и составные), их интерфейсы и алгоритмы. На основе этих блоков создается простое (централизованное) приложение, состоящее только из блоков, реализующих требуемую функциональность управляющей системы, диктуемую процессом.
2. Разработка и тестирование распределенной конфигурации системы с использованием симулятора распределенных устройств. На данном шаге создается описание ресурсов и устройств, содержащих вспомогательные функциональные блоки (сервисные – коммуникационные и управляющие), необходимые для работы приложения в распределенной конфигурации. Создается конфигурация распределенного приложения.
3. Разработка и тестирование распределенной конфигурации системы на реальных устройствах. На данном шаге распределенное приложение загружается в реальные устройства и осуществляется удаленное конфигурирование системы через инструментальные средства (для тех устройств, которые поддерживают конфигурирование) либо загрузка сгенерированного промежуточного ПО ("firmware") в не конфигурируемые пользователем устройства.

Данный переход от централизованной конфигурации к реальной системе обеспечивается моделью абстрактных ФБ. На всех этапах разработка сопровождается тестированием функциональности системы с использованием инструментальных средств (симуляция).

Стандарт определяет правила использования предлагаемых методик и дает рекомендации для достижения максимальной эффективности проектирования в различных типовых применениях, а также предлагает "образцы проектирования" – формализованные решения по организации проектирования для типовых задач. Применение "образцов проектирования" позволит упростить знакомство с концепциями стандарта, непривычными для специалистов в области проектирования систем управления, таких как принципы распределенного приложения, управление исполнением алгоритмов на основе событий и использование сервисных интерфейсных ФБ.

Кроме того, предлагается методология, основанная на идее визуального проектирования. Функционирование управляемой машины или процесса, а также соответствующих алгоритмов управления должно визуализироваться на всех этапах разработки – проектирования, симуляции, тестирования, внедрения и эксплуатации системы.

Выводы

Рассмотренный стандарт IEC 61499 показывает общие тенденции развития методов проектирования сложных промышленных управляющих систем на базе цифровой техники. Можно сделать вывод о том, что имеет место постепенный переход к более высокоуровневым подходам при разработке систем, которые должны обеспечить сокращение цикла разработки на основе применения объектно-ориентированного подхода, повторного использования имеющихся наработок, переносимости получаемого решения и упрощения интеграции решений различных поставщиков за счет стандартизации и унификации. Также постоянно растут требования к гибкости получаемых решений. Промышленные системы в будущем должны обладать способностью к изменениям конфигурации системы в связи с изменениями в технологическом процессе и для быстрого восстановления системы после сбоев. В дальнейшем ожидается, что управляющие системы будут представлять собой совокупность автономных интеллектуальных модулей, способных самоконфигурироваться в соответствии с изменяющимися требованиями.

ТЕСТИРОВАНИЕ И АНАЛИЗ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ, РАЗРАБОТАННЫХ ПО ТЕХНОЛОГИИ J2EE

А.А. Ермыкин, Т.И. Алиев

К настоящему времени создано довольно много самых различных методов тестирования и анализа, в том числе методы тестирования распределенных информационных систем (РИС) и их составляющих [1–4]. РИС могут быть построены по различным технологиям [2]. Поэтому методы и средства тестирования таких систем следует создавать с учетом специфики технологий, по которым они разработаны. К тому же существующее разнообразие типов таких систем настолько велико, что разработать для них универсальный метод тестирования и анализа даже в рамках одной технологии – очень непростая задача. Представляется и вовсе невозможным разработать автоматизированное универсальное средство, которое учитывало бы все нюансы конкретной РИС и, в то же время, выдавало бы корректные и достаточно полные результаты [2].

Сейчас существует сравнительно небольшое количество средств тестирования для систем, разработанных в рамках технологии J2EE [5]. Но, во-первых, эти тесты базируются на строго определенном наборе стандартных операций для нескольких наиболее типичных случаев в области использования таких систем. В то же время необходимо учитывать, что, как правило, для каждой задачи существует определенная специфика и определенная, характерная только для этой задачи, функциональность, соответствующая ей нагрузка и распределение нагрузки на все используемые данной системой виды ресурсов, необходимые для решения поставленной задачи. Во-вторых, существует необходимость тестировать и анализировать РИС под влиянием стрессовой нагрузки и анализировать полученные результаты. В-третьих, нужно получать данные о распределении нагрузки на ресурсы для каждой конкретной системы и о характере динамического изменения параметров характеристик этой системы при той или иной входной нагрузке. В-четвертых, эти возможности или, по крайней мере, их часть должны быть доступны уже после окончания первой части этапа разработки, на этапе тестирования, а также на всех последующих этапах эволюции РИС, включая поддержку и модернизацию. В-пятых, нужно в рамках существующей РИС правильно оценивать загрузку всех ресурсов РИС при различных условиях нагрузки. Необходимо анализировать результаты экспериментов и получать дополнительную информацию. Для этого необходимо разработать новый метод и подход к тестированию и анализу таких сложных систем. В нашем случае интерес представляет стресс-тестирование как способ не только протестировать специфические, но важные характеристики РИС, а также проанализировать систему. Стресс-тестирование является очень важным еще и потому, что позволяет выявлять и отслеживать не только функциональные свойства РИС, но и, что является особенно ценным, нефункциональные свойства.

Основные высокоуровневые сущности и понятия типичной системы, разработанной по технологии J2EE – информационная система (ИС), РИС, система управления базами данных (СУБД), сервер приложений (СП; Application Server – AS). Тест и система стресс-тестирования распределенной информационной системы (ССТРИС) могут быть использованы для тестирования и анализа распределенных информационных систем, разработанных по технологии J2EE. Соотношение между этими сущностями можно увидеть на рис. 1. Все диаграммы выполнены в соответствии со спецификацией UML.

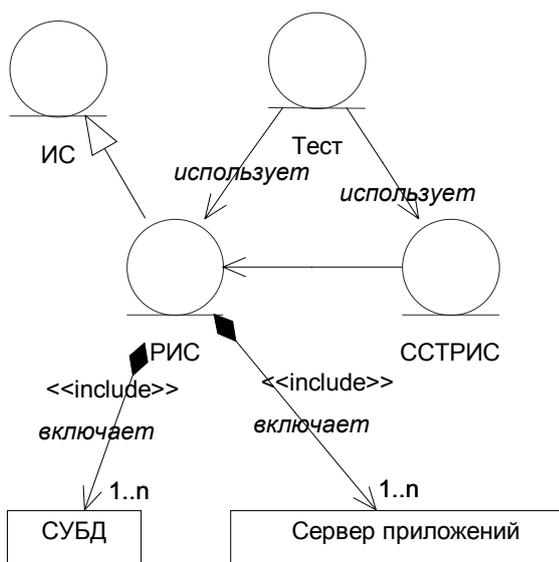


Рис. 1. Основные сущности и понятия системы

Типичная конфигурация РИС, разработанной по технологии J2EE, представлена на диаграмме размещения РИС с кластером серверов приложений (см. рис. 2).

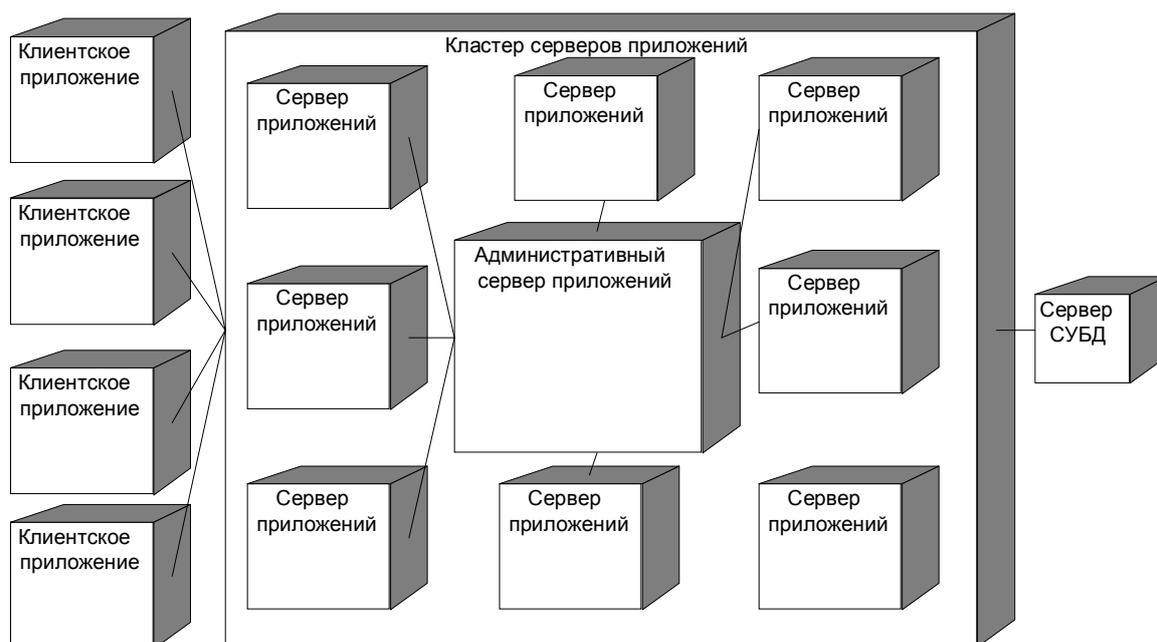


Рис. 2. Диаграмма размещения РИС с кластером серверов приложений

Сейчас уже существуют определенные тестовые программы, комплексы и средства, которые могут быть применены для тестирования и, в частности, для стресс тестирования РИС [5, 10, 12]. Но ни одно из этих средств не учитывает все те пять условий, которые были упомянуты выше. Аспекты тестирования, которые покрываются в случае применения существующих средств можно понять из диаграммы покрытия конкретными тестовыми программами, комплексами и средствами тестируемых сущностей и систем (см. рис. 3).

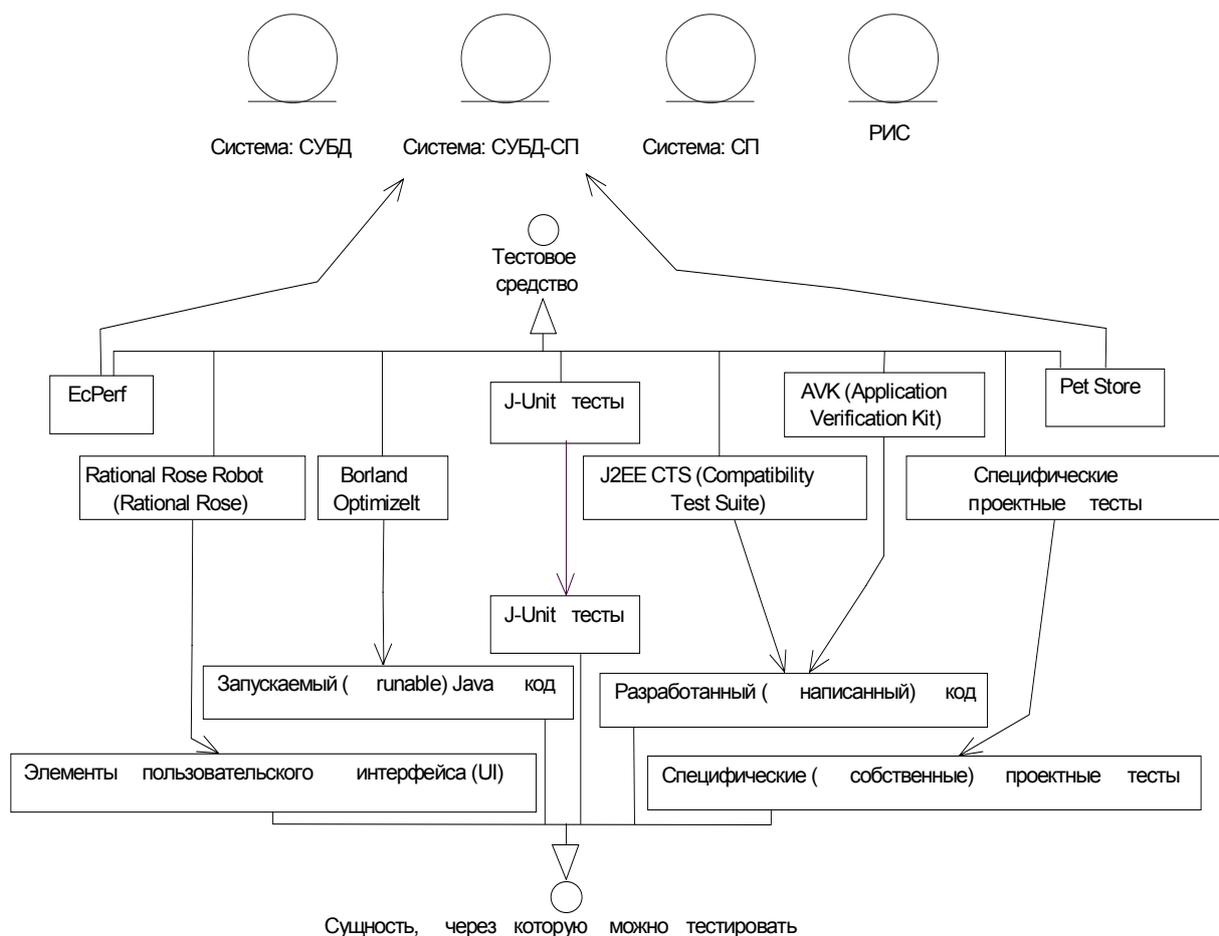


Рис. 3. Диаграмма покрытия конкретными тестовыми программами, комплексами и средствами тестируемых сущностей и систем

Как видно из этой диаграммы, область тестирования РИС не покрывается полностью такими средствами тестирования. Чтобы удовлетворить пяти вышеизложенным условиям, нужен другой подход и соответствующий ему метод. Авторами предложен метод, включающий ССТРИС – см. диаграмму размещения системы ССТРИС-РИС на рис. 4.

ССТРИС является той системой, которая может помочь в достижении поставленных целей. Можно снять данные со всех критических участков РИС с помощью наблюдателей (см. рис. 4). Анализатор – часть ССТРИС, которая должна получить, обработать и проанализировать данные. Клиентская часть ССТРИС имитирует нагрузку от реальных пользователей системы. Здесь важно удовлетворить требованию максимальной приближенности к реальным входным воздействиям. В то же время существует требование экономической целесообразности и эффективности, что обычно противоречит первому. В результате проведения стресс-тестирования путем анализа полученных данных можно наиболее полно и достоверно охарактеризовать РИС и выявить особенно интересующие нефункциональные свойства системы. Последнее качество ССТРИС является очень важным не только для отслеживания состояния системы при эволюции, но и для того, чтобы информировать заказчика о том, насколько большую нагрузку и с какими показателями может выдержать конечная РИС.

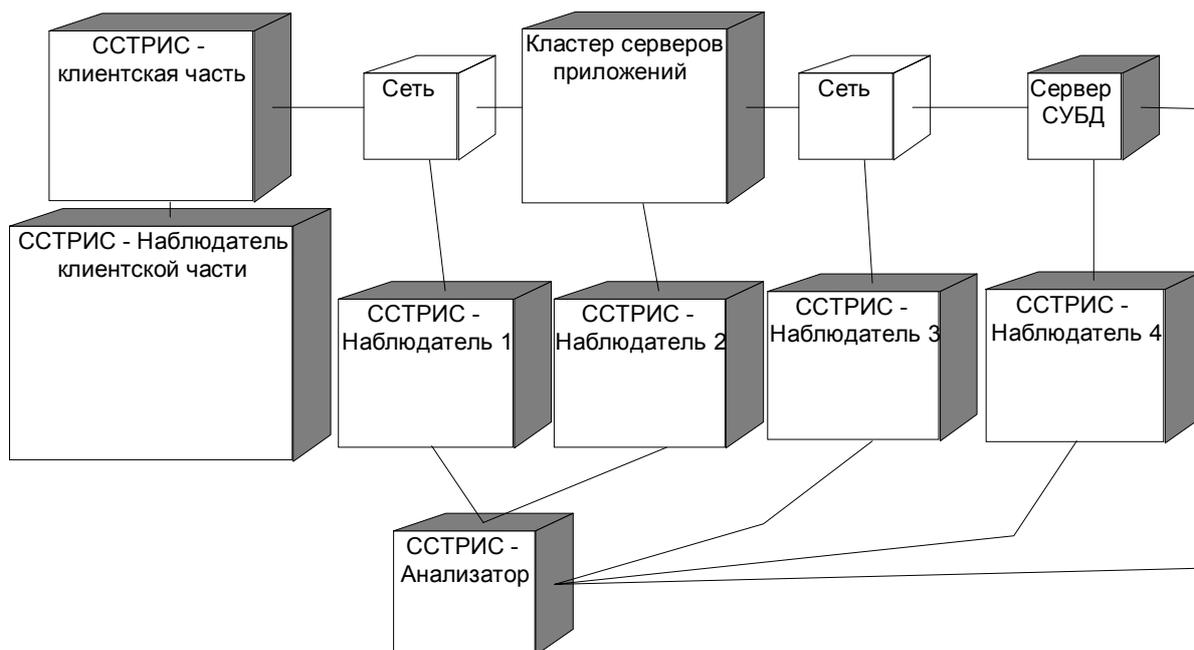


Рис. 4. Диаграмма размещения системы ССТРИС-РИС

Стоит еще раз упомянуть, что одной из важных причин необходимости создания нового метода стресс-тестирования является экономическая целесообразность, о которой нельзя ни в коем случае забывать. Кому будет интересен прекрасный метод стресс-тестирования и анализа, если его применение будет стоить в десятки, а то и сотни раз дороже, чем разработка всей системы?

Сейчас появляются некоторые коммерческие средства стресс-тестирования, но и качество, и полнота, и стоимость их проведения оставляют желать лучшего [11]. В зависимости от конкретной РИС стоимость может колебаться от нескольких сотен тысяч долларов (для средних и малых систем) до нескольких миллионов долларов. Предлагаемый метод призван улучшить полноту и соотношение стоимость-качество для такого рода тестирования.

Литература

1. Dimitra Giannakopoulou, Gary T. Leavens, Murali Sitaraman и др. SAVCBS 2001 Proceedings Specification and Verification of Component-Based Systems // Workshop at OOPSLA 2001. October 14, 200. P. 136.
2. Gopalan Suresh Raj. Web Cornucopia. An Oasis for the parched Enterprise Component Engineer/Developer // URL: <http://www.execpc.com/~gopalan/>
3. Nayeem Islam. Distributed Objects Methodologies for Customizing Systems Software // IEEE Computer Society Press. 1996.
4. Philippe Collet. On Contract Monitoring for the Verification of Component Based Systems // Objects and Software Components Group.
5. Enterprise: Java Pet Store Sample Application // URL: <http://java.sun.com/blueprints/code/>
6. Воас Дж. Сопровождение компонентных систем. // Открытые системы. 1998. № 6 // URL: <http://www.osp.ru/os/1998/06/17.htm>
7. Ахтырченко К. В., Леонтьев В. В. Распределенные объектные технологии в информационных системах. // СУБД. 1997. № 5–6.
8. Эккерсон В. В поисках лучшей архитектуры клиент-сервер // Сети. 1995. № 4.

9. Волков А.А. Про тесты измерения производительности // Computerworld. 1996. № 5.
10. ECPperf Specification // URL: <http://java.sun.com/j2ee/ecperf/>
11. LoadRunner Specification // URL: <http://www-heva.mercuryinteractive.com/products/loadrunner/>
12. Test Smarter, Finish Faster. Rational Robot // URL: http://www-scf.usc.edu/~csci577/teams/team12a/LCA/MDL/MDL_LCA_F02a_T12.pdf

ПРОБЛЕМЫ ТЕСТИРОВАНИЯ СИСТЕМНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ

А.О. Ключев, Н.А. Маковецкая

Рассматривается применимость классических подходов к тестированию системного программного обеспечения (ПО) распределенных информационно-управляющих систем (РИУС).

В настоящее время наблюдается повышение уровня автоматизации в различных сферах человеческой деятельности, переход на новые технологии построения систем автоматизации. В значительной мере этому способствует использование заказных встроенных систем управления объектами. Таким образом, проектирование встроенных вычислительных систем приобретает массовый характер.

Ко многим подобным системам предъявляются повышенные требования по надежности и безопасности функционирования. Так как тестирование – это основной метод измерения качества, определения корректности и реальной надежности функционирования программ [2], приобретает большое значение вопрос о применимости классических методов тестирования программ к ПО встроенных систем и в частности – ПО распределенных информационно-управляющих систем (РИУС).

В русскоязычной литературе преобладают описания методик тестирования вычислительных систем общего назначения, в области же проектирования и тестирования встроенных систем существует определенный информационный вакуум. Возникают опасения, что зарождающиеся в России направления проектирования встроенных систем могут быть серьезно дезориентированы общепринятыми на данный момент подходами к проектированию систем общего назначения.

В данной статье будет сделана попытка проанализировать проблему тестирования встроенных систем с помощью методик, используемых для систем общего назначения, на примере РИУС – одного из самых сложных видов встроенных систем.

Перечислим классические задачи при тестировании ПО [1,2].

1. Тестирование функциональности (тестирование "черного ящика"): проверка функциональности системы на соответствие требованиям.
2. Тестирование производительности: определение характеристик производительности, поиск "узких мест".
3. Стрессовое тестирование системы при предельной нагрузке на различные ресурсы.
4. Проверка устойчивости системы к неполадкам и безопасного оперативного восстановления при сбоях: оценка эффективности защиты от аппаратных сбоев, последствий проявления невыявленных дефектов и ошибок программ и данных.
5. Тестирование совместимости ПО со всеми модификациями аппаратуры, для которых оно предназначено.
6. Анализ исходных текстов (просмотр экспертом и автоматический анализ).
7. Оценка удобства использования системы.
8. Тестирование способности модулей системы к взаимодействию, в том числе перераспределения нагрузки.
9. Тестирование надежности: проверка поведения при длительной непрерывной эксплуатации при большой нагрузке.

Тестирование подразделяют на прогрессивное, при котором проверяются новые компоненты, и регрессивное, когда контролируется качество и корректность изменений в программах и данных после внесения в них изменений. Согласно еще одной классификации, тестирование бывает восходящее, когда сначала тестируются объекты нижнего уровня, и нисходящее, при котором в первую очередь тестируются высшие иерархические уровни системы, а потом низшие.

Несмотря на многообразие методов и инструментов тестирования, как правило, некоторые ошибки все равно остаются невыявленными. Основной причиной этого является недостаток времени и сил для тестирования. Эффективность тестирования можно повысить, встроив средства тестирования в архитектуру системы уже на этапе проектирования; практикуется также создание тестов перед написанием тестируемого кода. Еще один путь – автоматическая генерация тестовых наборов и автоматическое проведение тестов, особенно регрессивных.

На процесс тестирования и отладки системного ПО РИУС влияют следующие факторы:

1. РИУС – сложный программно-аппаратный комплекс, состоящий из множества разнообразных компонентов, часто пространственно распределенный;
2. РИУС интегрирована в объект управления;
3. работа в реальном масштабе времени (РВ);
4. в реальных условиях доступ к модулям РИУС часто затруднен или невозможен, так как они устанавливаются в труднодоступных или вовсе недоступных местах (например, на космических станциях);
5. тяжелые климатические, механические и электромагнитные условия эксплуатации: высокая влажность, неблагоприятная температура, вибрация, сильные электромагнитные поля;
6. жесткие требования к надежности и безопасности функционирования: РИУС управляют такими критическими объектами, как АЭС, железные дороги, военная техника, медицинское оборудование;
7. тесная интеграция программного и аппаратного обеспечения РИУС: качественное тестирование ПО невозможно при отсутствии корректно функционирующей аппаратуры. В то же время на начальных стадиях разработки аппаратура работает нестабильно или не работает вообще. Таким образом, приходится тестировать сразу весь программно-аппаратный комплекс, каждый модуль которого может содержать ошибки. Это значительно усложняет интерпретацию результатов тестирования: во-первых, иногда некоторые ошибки компенсируют друг друга; во-вторых, из-за невозможности тестирования единичных модулей отдельно от всей системы очень трудно локализовать источник проблем.
8. некоторые аппаратные компоненты имеют ограниченный рабочий ресурс: E²PROM и FLASH имеют ограниченное число циклов записи, электромеханические механизмы (клапаны, реле и т.п.) постепенно изнашиваются.

Тестирование функциональности РИУС, а также проверка устойчивости системы к неполадкам и безопасного оперативного восстановления при сбоях затруднены по следующим причинам.

1. Сложность, модульность, распределенность, гетерогенность системы.
2. Для тестирования требуется широкий спектр дополнительной аппаратуры и ПО. При этом многие инструменты либо очень дороги, либо вообще не существуют, и их нужно разрабатывать с нуля, что может быть сопряжено с значительными затратами денег и времени.
3. Проведение испытаний на реальном объекте управления обычно невозможно. Даже если такая возможность существует, аварийная ситуация при проведении подобных испытаний может привести к катастрофическим последствиям. Симуляция объекта управления опасна дополнительными ошибками и требует специального дорогостоящего оборудования для имитации внешних воздействий.
4. Требование работоспособности РИУС в различных внешних условиях означает, что нужно повторить все тестовые испытания при высокой или низкой температуре, высокой влажности, в условиях вибрации, жесткого электромагнитного излучения

и т.д. Это значительно увеличивает трудозатраты на тестирование и требует сложного дорогого оборудования (термокамеры, вибростенды).

Тестирование производительности РИУС малозначимо на фоне остальных задач. Однако существует гораздо более серьезная проблема – проверка способности системы к работе в реальном масштабе времени. Способов формального доказательства соответствия системы требованиям РВ пока не существует, а проверка на практике усложняется тем, что сама система тестирования при совместной работе с целевой системой искажает временные показатели и содержимое памяти. Таким образом, проблема может быть замаскирована инструментальной погрешностью или наоборот, может быть выдано ложное сообщение о несоответствии системы требованиям РВ. Более того, анализ причин проблемы будет затруднен из-за порожденных системой тестирования искажений в дампах памяти. Чтобы избежать этих трудностей, система тестирования интегрируется в целевую систему, но тогда появляется другая проблема: система тестирования съедает ресурсы, которых и так не хватает.

Стрессовое тестирование в некоторых случаях затруднено, так как рабочий ресурс аппаратуры ограничен.

Тестирование совместимости: в отличие от тестирования ПО для универсальных персональных компьютеров, обычно не требуется проверки совместимости с широкой номенклатурой аппаратных и программных продуктов, а нужна только совместимость с продукцией этого же производителя и некоей базовой конфигурацией ПК. Таким образом, тестирование совместимости увеличивает общий объем работ по тестированию всего в несколько раз. Однако в случае, когда требуется протестировать совместимость ПО с продуктами сторонних фирм, возникают проблемы дороговизны этих продуктов и нежелания конкурирующих фирм предоставлять свои системы для подобных целей.

Хотя *анализ исходных текстов* позволил бы обнаружить потенциальные дефекты ПО на ранних стадиях разработки, для проведения подобного анализа вручную необходим эксперт по аппаратуре и ПО. Проблема выбора и подготовки такого эксперта усугубляется сложностью и частой сменой элементной базы. Автоматизированный просмотр исходных текстов не представляется целесообразным: как правило, системное ПО РИУС не содержит сложных алгоритмов, а основное количество ошибок кроется в неправильном использовании регистров специального назначения, несоблюдении временных соотношений и т.п. Автоматическая система анализа исходных текстов вряд ли сможет обнаружить подобные ошибки, а значит, будет практически бесполезна.

Оценка удобства использования системного ПО РИУС не столь актуальна, так как, как правило, с ним редко работают напрямую. Для интерфейсов оператора важна согласованность с пользовательской "моделью мира": интерфейс системы должен соответствовать представлению пользователя о ее устройстве и окружении, а не истинной архитектуре системы. Для РИУС удобство использования отходит на второй план по сравнению с такими показателями, как надежность и безопасность функционирования. Как правило, оценка удобства использования проводится уже на месте заказчиком.

Тестирование способности модулей системы к взаимодействию: несмотря на то, что эта способность является ключевой для РИУС, как правило, подобное тестирование проводится на малом числе модулей (воспроизведение необходимых внешних условий для большого числа модулей трудоемко и дорого).

Тестирование надежности не представляет особых технических проблем, но может отнять очень много времени.

Автоматической генерации тестов мешают такие факторы, как тесная связь ПО с аппаратурой, интеграция РИУС с ОУ, а также их модульность (это увеличивает объем

тестовых испытаний: нужно создать тесты не для одной системы, а для множества разнообразных модулей). Автоматическая генерация тестов для РИУС сложна и в некоторых случаях опасна: некоторые, на первый взгляд совершенно невинные, наборы тестовых значений могут привести к катастрофе. Поэтому перед запуском тесты обязательно должен проверять эксперт.

Регрессивное тестирование вручную очень трудоемко и требует использования сложного тестового оборудования. Практика показывает, что регрессивное тестирование одного контроллера только лишь для одного варианта внешних условий может длиться часами, при этом большое количество ручных операций делает возможным пропуск тестировщиком многих ошибок (человеческий фактор). Еще одним препятствием для регрессивного тестирования является ограниченность рабочего ресурса аппаратуры.

Автоматическое проведение тестов, в том числе регрессивных, ограничивается тем, что для генерации входных тестовых воздействий необходима дополнительная аппаратура и, возможно, ОУ.

В классическом варианте встраивание средств тестирования в архитектуру системы является необязательным; для РИУС это жизненно необходимо. Практика показывает, что тестирование РИУС требует создания соизмеримого с ней или даже превосходящего ее по сложности тестового комплекса.

Как видно из представленных примеров, применение классических способов тестирования либо не дает должного эффекта, либо приводит к очень большим материальным затратам. В связи с тем, что после распада СССР серьезно снизилась активность российских производителей встроенных систем и вся эта область проектирования долгое время находилась в кризисе, возникло значительное отставание от зарубежных методологий. Для ликвидации этого разрыва требуется переосмысление привычных подходов к проектированию встроенных систем.

Литература

1. Web-сайт компании Amphora Quality Technologies (www.aqt.ru).
2. Липаев В.В. Надежность программных средств. Серия «Информатизация России на пороге XXI века». М.: СИНТЕГ, 1998. 232 с.

ПРИМЕНЕНИЕ ТЕХНОЛОГИИ OPC

Р.Р. Ковязин, А.Е. Платунов

Введение

Существует множество встроенных систем, для которых важной составляющей является взаимодействие с персональным компьютером. Наличие компьютера может быть обязательным и необязательным. Среди всех функций персонального компьютера можно выделить следующие функции:

- протоколирование функционирования системы;
- различные виды тестирования;
- получение целевой и отладочной информации (обработка, визуализация);
- настройка системы;
- предварительная обработка данных.

Для связи персонального компьютера и встроенной системы обычно используются либо стандартные интерфейсы (последовательный и параллельный канал, USB), либо специализированные платы (ByteBlaster, BitBlaster), которые могут располагаться в системном блоке или подключаться к нему с помощью стандартного интерфейса. В большинстве случаев персональный компьютер является инструментальной машиной, устройством хранения информации, управляющей подсистемой. Вычислительная система предоставляет интерфейс, а его сложность зависит от множества параметров – физический интерфейс, потоки данных и др. Программное обеспечение на персональном компьютере должно предоставить высокоуровневый интерфейс пользователя, скрывающий особенности связи с вычислительной системой в большей или меньшей степени. Некоторые встроенные системы используют широко распространенные протоколы передачи данных, например терминалы ANSI, VT100, стек TCP/IP. Фирмы, разрабатывающие встроенные системы, микросхемы программируемой логики, микроконтроллеры и другие программируемые электронные компоненты, стараются обеспечить их универсальными инструментальными средствами (САПР), позволяющими работать и конфигурировать не одно устройство, а ряд устройств. Естественно, предлагать пользователям ряд систем и обеспечивать их одним САПР выгоднее, чем создавать для каждой системы свой САПР. Существует ряд стандартов и рекомендаций по организации и программированию вычислительных систем и инструментальных средств к ним, например стандарт IEC1131. Они поддержаны множеством производителей. Сейчас с ростом мощности и объема ресурсов вычислительных систем, с появлением новых технологий появляются новые стандарты. Организация OPC Foundation [1] предлагает свой стандарт, описывающий организацию интерфейса пользователя вычислительной системы, достаточно универсального и гибкого, использующего современные информационные технологии, позволяющего единообразно работать с разными вычислительными системами.

Модель взаимодействия встроенных систем и приложений персонального компьютера

Учитывая вышесказанное, можно выделить три способа взаимодействия встроенных систем с инструментальным и целевым программным обеспечением на персональном компьютере:

1. "Одна встроенная система — одно приложение". Каждой системе соответствует свое уникальное программное обеспечение (см. рис. 1). В большинстве случаев его разрабатывает фирма-разработчик встроенной системы. Используя эту

методику, можно добиться оптимальных результатов в плане скорости обмена информацией, получить возможность использования всех достоинств каждой встроенной системы. С другой стороны, поддержание и модернизация программного обеспечения для большого числа встроенных систем накладно.

2. "Одна встроенная система – много приложений". Каждая система предоставляет открытый всем или ряду фирм, разрабатывающих программное обеспечение, интерфейс и любое приложение, поддержав его, может взаимодействовать со встроенной системой (см. рис. 2). В этом случае разработка и производство систем облегчается, так как разработка программного обеспечения может быть осуществлена как фирмой-разработчиком встроенной системы, так и третьими фирмами.
3. "Много встроенных систем — много приложений". Каждая встроенная система имеет свой интерфейс. Она поставляется вместе с драйвером, предоставляющим стандартный интерфейс, одинаковый для всех встроенных систем (см. рис. 3). Пользовательские приложения тоже предоставляют стандартный интерфейс, так что каждая встроенная система может взаимодействовать с любым из них.

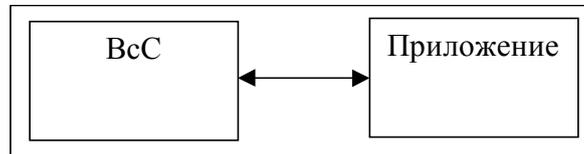


Рис. 1. Встроенная система с уникальным программным обеспечением

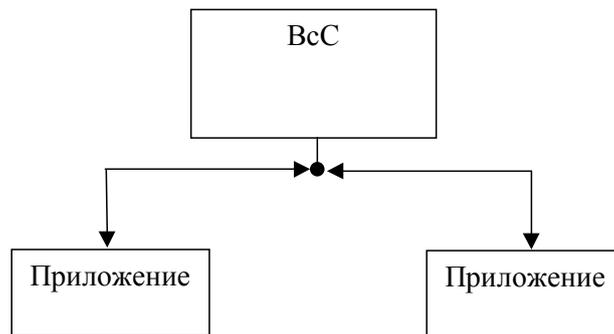


Рис. 2. Встроенная система с открытым интерфейсом для программного обеспечения

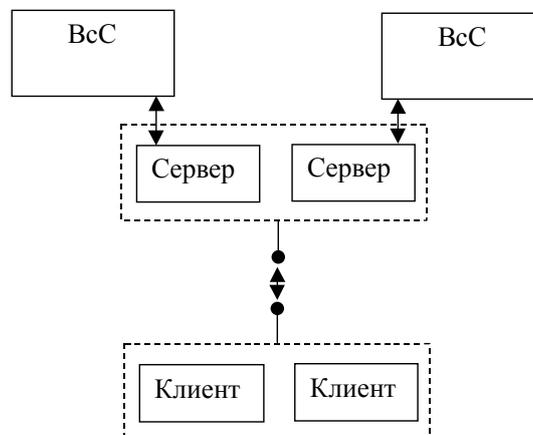


Рис. 3. Встроенная система со стандартным интерфейсом для программного обеспечения

Введение в технологию OPC

В разработке технологии OPC (OLE for Process Control) принимает участие множество корпораций. По приведенной выше классификации концепция OPC предлагает организацию взаимодействия третьим способом ("много встроенных систем – много приложений"). Технология OPC содержит в себе ряд спецификаций. Несмотря на то, что им требуется операционная система WIN32, и большинство приложений, поддерживающих эту технологию, разработано как раз под нее, есть разработки и под другие операционные системы, например, QNX. По сути, стандарты OPC указывают на то, какие интерфейсы должны предоставлять участники обмена, но не указывают ни способа реализации интерфейсов, ни среды взаимодействия [2].

Основой технологии OPC являются следующие понятия:

- сервер и клиент;
- интерфейс;
- интеграция.

Сервер – это программа или библиотека, исполняющаяся на персональном компьютере, к которому непосредственно подключена встроенная система. Взаимодействие сервера и встроенной системы является первым способом взаимодействия. Клиент – это программа, которой нужен доступ ко встроенной системе. Схема взаимодействия клиентов и серверов показана на рис. 4. У каждого сервера может быть несколько клиентов, а клиент может обмениваться данными с несколькими серверами.

Технология OPC базируется на модели COM (Component Object Model). OPC-сервер и OPC-клиент выполнены в виде COM-объектов. В Win32-совместимой операционной системе взаимодействие COM-объектов происходит на системном уровне. Всего существует 3 типа OPC-серверов:

- DataAccess-сервер предназначен для обмена данными;
- Alarm&Event-сервер предназначен для отслеживания и обработки событий;
- HistoricalData-сервер предназначен для дополнительной информации и протоколирования.

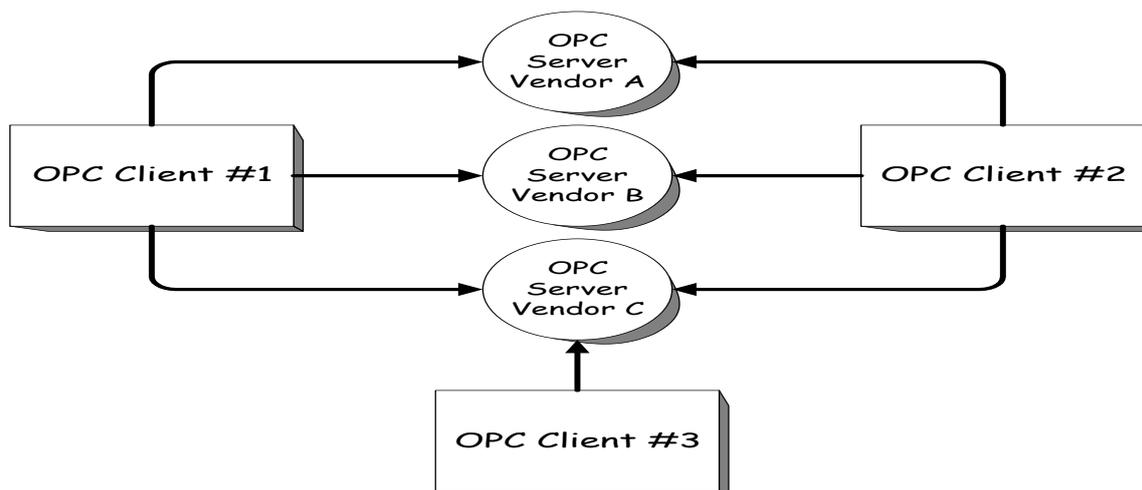


Рис. 4. Взаимодействие OPC-серверов и OPC-клиентов

Все OPC-серверы реализуют, как минимум, функциональность DataAccess-сервера [3], потому что она необходима для работы со встроенной системой, а оставшиеся 2 типа серверов базируются на DataAccess-сервере. Технология OPC содержит как обязательные к реализации интерфейсы, так и необязательные. В каждом интерфейсе также есть обязательные и необязательные функции. В соответствии с

моделью COM набор функций каждого интерфейса фиксирован. Необязательность функций означает, что при вызове они возвращают код ошибки, говорящий о том, что функция не реализована. Например, каждая единица данных Data Access сервера идентифицируются уникальным именем. В минимальном объеме реализации сервер не предоставляет информацию о названиях своих данных. В более полной реализации есть специальный интерфейс для получения списка имен и других свойств данных (тип, поддержка временных меток и т.п.). Имена могут быть произвольными, а могут иметь иерархическую структуру, например MD846.DIN3, MD846.DOUT2, MA444.AIN8. Есть интерфейсы, позволяющие раскрывать эту структуру имен, переходить от одного к другому, но они необязательны. В минимальной реализации клиент может получить доступ к данным OPC-сервера, только если он знает названия данных из других источников.

OPC-клиент предоставляет OPC-серверу интерфейс IAdviceSink. В минимальном объеме он способен принимать и передавать данные. Поддержание более высокоуровневых механизмов требует их реализации как в сервере, так и в клиенте. Могут быть реализованы такие возможности, как совместная работа нескольких клиентов, имеющих общие данные, уведомление о завершении работы сервера, получение дополнительной информации о клиентах и т.п.

Разработать OPC-приложения можно на основе любой среды программирования, позволяющей использовать технологию COM. В том числе можно использовать офисные приложения, например MS Word, в которые встроен язык Basic for Applications или VBScript. В этом случае происходит интеграция OPC-приложений в офисные приложения, что довольно удобно. Например, можно получить последовательность данных от встроенной системы, показать ее в таблице Excel, произвести необходимые преобразования и построить график, пользуясь исключительно средствами Excel. Большинство интерфейсов OPC продублировано для интегрированных приложений. Поддержка интеграции в различные среды является необязательной.

Анализ применения технологии OPC

Использование технологии OPC предоставляет возможности совершенно нового уровня по сравнению с возможностями инструментальных средств, не придерживающихся стандартов такого уровня. Приложения могут быть написаны задолго до момента готовности встроенной системы, так как интерфейсы OPC определены и зафиксированы. Достоинством применения технологии OPC является также легкая и незаметная модернизация приложений, что является следствием применения технологии COM. Серверы и клиенты могут находиться на разных рабочих станциях как в пределах локальной сети, так и по всему миру, а для связи использовать как IPС в рамках одного компьютера, так и обмениваться через Интернет. Применение одного инструмента формирует стандартные или часто применяемые приемы работы с ним. Ряд клиентов могут быть использованы для работы с большим числом встроенных систем. Технология OPC позволяет применять сетевую поддержку, интеграцию и другие сервисы, предоставляемые операционной системой.

Технология OPC является мощным и универсальным средством, но это же является причиной его сложности. Универсальность не эквивалентна удобству, поддержка стандарта OPC увеличивает сложность программного обеспечения, для его разработки требуются более квалифицированные сотрудники. Основным недостатком этой технологии является то, что в силу своей универсальности она не может предоставить достаточно высокоуровневый интерфейс. Возможность работы с большим числом встроенных систем ведет к тому, что множество параметров обмена

данными задается непосредственно в вызовах функций. Например, добавление опрашиваемой клиентом ячейки данных требует 12 параметров, а ее чтение – 7 параметров. Понятно, что для разных систем их значения будут различными, но значительное число параметров придает громоздкость и усложняет применение этой технологии. Ряд систем требует особых свойств от программ, работающих с ними, например работы в реальном масштабе времени, высокого быстродействия. Технология OPC предоставляет наиболее широко используемые функции для работы со встроенными системами и стремится к универсализации, а не оптимальности. Добиться в ее рамках режимов работы, имеющих не заложенную в нее специфику, сложно. Применение технологии OPC в таких системах может привести к потере их выгодных качеств. С другой стороны, встроенная система может быть настолько простой или иметь такой простой интерфейс, что использование сложных средств, предоставляемых технологией OPC, окажется неоправданным.

Как применить технологию OPC

Существует два способа поддержания встроенной системой технологии OPC:

- реализация требуемых СОМ-компонентов;
- использование готовых библиотек компонентов OPC.

Самым сложным способом является полная реализация компонент СОМ, необходимых для поддержания технологии OPC. Стандарт OPC указывает, каким должно быть поведение OPC-серверов и OPC-клиентов, но не указывает, как они должны быть реализованы. К сожалению, множество реализаций имеет те или иные особенности, обусловленные внутренней архитектурой компонент и встроенной системы. В любом случае технология OPC – это только инструмент. Можно доверить разработку подсистемы, реализующей технологию OPC или весь OPC-сервер или клиент, третьим фирмам, но в любом случае это будет самым долгим и самым дорогим способом поддержания технологии OPC.

Другой способ состоит в использовании готовых библиотек. Ряд фирм занимается разработкой библиотек, модулей, предоставляющих, с одной стороны, интерфейсы OPC, а с другой стороны – программный интерфейс для программ, использующих их. Эти библиотеки предоставляют интерфейс, существенно упрощенный по сравнению с интерфейсами OPC, что облегчает работу с ними. К упрощениям относятся, например, поддержание не всех типов данных VARIANT, обработка данных OPC-сервера одним или парой способов, неподдержание ряда необязательных интерфейсов и т.п. С другой стороны, эти упрощения означают, что программист лишается ряда возможностей технологии OPC, которые могут оказаться нужными. Стоимость исполняемых модулей таких библиотек составляет 500–1000 \$, а покупка их вместе с исходными текстами обойдется в 2000–3000 \$. И это всего лишь интерфейсы OPC, а OPC-сервер и OPC-клиент – это не только интерфейсы, а еще и логика обработки данных, которую нужно будет реализовать с использованием приобретенных OPC-библиотек.

С целью ознакомления с технологией OPC для контроллерной сети modbus-устройств были разработаны OPC-сервер и OPC-клиент. Организация всей системы показана на рис. 5. ПЛК MCN 3.0 управляет сетью устройств аналогового и дискретного ввода/вывода по интерфейсу ModBus. Для исследования технологии OPC была использована демонстрационная библиотека WTOPC фирмы WinTech Software Design, предоставляющая основные интерфейсы OPC и позволяющая работать в течение 30 минут. Были выявлены неприятные особенности, которые уже указывались выше – неправильная поддержка структурных типов данных. Кроме того, было

выявлено различное поведение программного обеспечения на разных операционных системах (MS Windows 2000, MS Windows 98).

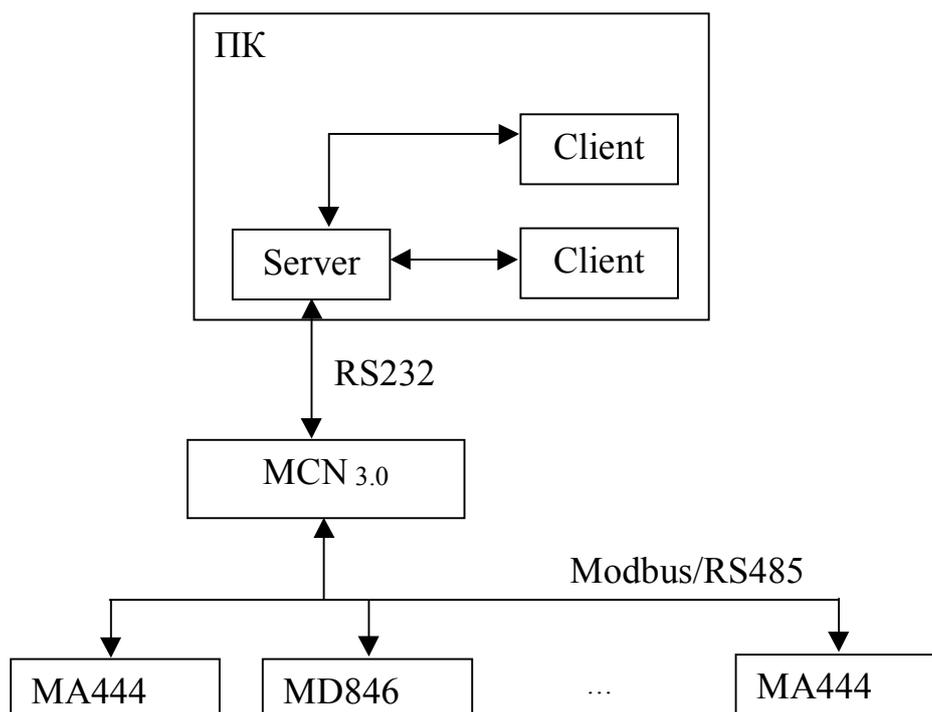


Рис. 5. Организация контроллеров сети modbus-устройств.

Выводы

OPC является мощным и универсальным средством. Технология OPC сделала значительный вклад в стандартизацию встроенных систем, взаимодействующих с компьютерами. Множество фирм-разработчиков применили ее и предоставили пользователям унифицированный способ доступа к своим системам. Появилось множество фирм-разработчиков программного обеспечения, специализирующихся только на разработке систем, поддерживающих технологию OPC. Однако, поддержание этой технологии в системах, при проектировании которых это изначально не было учтено, проблематично. Существуют встроенные системы, в которых применение OPC неоправданно, так что технология OPC на текущем уровне развития пока не может позиционироваться как универсальный инструмент для организации взаимодействия встроенных систем и программного обеспечения на персональном компьютере.

Литература

1. <http://www.opcfoundation.org> – официальный сайт организации OPC Foundation.
2. <http://www.opcfoundation.org/Downloads/Overview/OPC%20Overview%201.00.pdf> – обзор технологии OPC.
3. <http://www.opcfoundation.org/Downloads/Data%20Access/OPC%20DA%202.05a%20Specification.pdf> – стандарт на интерфейс DataAccess-сервера.

СРАВНЕНИЕ АРХИТЕКТУР ПОСТРОЕНИЯ СЕРВЕРОВ МУЛЬТИМЕДИА

М.С. Косяков, Т.И. Алиев

Введение

Современные достижения информационных технологий сделали возможным и экономически обоснованным сетевой доступ к разнообразным источникам информации, таким как периодические и литературные издания, аудиозаписи и видеоролики, научные данные и т.д. Типичная архитектура подобных систем подразумевает использование серверов мультимедиа, обеспечивающих хранение и предоставляющих доступ как к мультимедийным (аудио- и видео-), так и обычным текстовым данным. При этом характеристики мультимедийной информации сильно отличаются от характеристик традиционных текстовых файлов, для работы с которыми создавались современные файловые системы. Для управления новыми типами данных требуются новые файловые системы, реализующие специфические дисциплины дискового планирования и политики замещения страниц буферного кэша [1].

На данный момент существуют два подхода к построению файловых систем подобных серверов [2]:

- файловые системы раздельной архитектуры, использующие независимые буферные кэши и массивы накопителей на магнитных дисках (НМД) для организации хранения и доступа к данным каждого типа;
- интегрированные файловые системы, предоставляющие ресурсы сервера для совместного использования данными обоих типов.

Так как принципы построения файловых систем, оптимизированных под конкретный тип данных, хорошо известны [3], файловые системы раздельной архитектуры просты при разработке и реализации. В этом случае запросы к данным различных типов (запросы различных типов) обслуживаются независимо, и предотвращается их взаимное влияние друг на друга: обслуживание запросов к текстовым данным не влияет на качество воспроизведения мультимедийных объектов. Однако подобный подход приводит к ограничениям.

- Раздельная архитектура не позволяет динамически выделять и перераспределять ресурсы сервера в случае меняющегося во времени соотношения нагрузок к текстовым и мультимедийным данным. При необходимости перераспределение таких ресурсов, как дисковые массивы и буферные кэши, будет являться трудной задачей и может потребовать остановки системы на долгое время. Простое добавление требуемых ресурсов (например, НМД) как альтернатива перераспределению ведет к неэффективному использованию ресурсов, принадлежащих незагруженной компоненте сервера.
- Количество НМД, используемых каждой компонентой сервера, определяется двумя факторами – требованиями объема хранимой информации и требованиями скорости считывания для каждого файла. Несоответствие параметров НМД требованиям для данных каждого типа, хранимых в своем дисковом массиве, может привести к неэффективному использованию либо дискового пространства, либо пропускной способности подсистемы ввода/вывода.

Перечисленные недостатки файловых систем раздельной архитектуры послужили толчком к разработке различных интегрированных файловых систем [4, 5]. Однако, за исключением [2], все работы в этом направлении посвящены изучению специфических для рассматриваемого случая методов управления данными и ресурсами сервера при неявном предположении превосходства предлагаемого подхода над файловыми системами раздельной архитектуры.

В связи с этим для сравнения рассмотренных архитектур построения файловых систем серверов мультимедиа в данной работе проводится анализ производительности работы дисковой подсистемы ввода/вывода для каждого из случаев. В отличие от [2], количественное сравнение характеристик производительности разрабатываемой интегрированной файловой системы и файловой системы отдельной архитектуры проводится для случая рассмотренного в [6] смешанного динамического дискового планирования FCFS с дисциплиной обслуживания очереди запросов к НМД SCAN. По сравнению с применяемым в [2] алгоритмом дискового планирования Cello [7], представленный алгоритм более эффективно использует пропускную способность подсистемы ввода/вывода сервера мультимедиа и тем самым увеличивает его производительность. Кроме того, моделирование проводится с использованием структурных параметров современных НМД IBM UltraStar 36Гб [8] вместо устаревших Seagate Elite3, рассматриваемых в [2], и нагрузочных параметров, полученных из наблюдений за реальными системами [9, 10], а не гипотетическим путем. И наконец, в отличие от [2], в данной работе рассматривалось применение алгоритма контроля допустимости, обеспечивающего стохастический уровень качества предоставляемых услуг (Quality of Service – QoS), что увеличивает эффективность использования ресурсов сервера и позволяет обслуживать большее количество клиентов.

Параметры моделирования

Для сравнения рассмотренных архитектур файловых систем были реализованы имитационные модели работы дисковой подсистемы ввода/вывода серверов мультимедиа различных архитектур. При этом полагалось, что для сервера с файловой системой отдельной архитектуры выделено D1 и D2 дисков для поддержки текстовых и мультимедийных данных соответственно, в то время как для сервера с интегрированной файловой системой выделено D1+D2 дисков для поддержки данных обоих типов. В обоих случаях полагалось, что объем буферной памяти достаточен для обслуживания запросов обоих типов.

Значения структурных параметров моделирования приведены в табл. 1.

Тип параметров	Файловая система отдельной архитектуры	Интегрированная файловая система
Параметры НМД	IBM UltraStar 36LZX	IBM UltraStar 36LZX
Количество НМД	D1 = D2 = 5	D1+D2 = 10
Дисковое планирование	-	FCFS, смешанное динамическое
Длина цикла T	1 сек.	1 сек.
Дисциплина обслуживания очереди запросов к НМД	SCAN	SCAN

Таблица 1. Структурные параметры моделирования

В качестве входного потока запросов к текстовым данным будем рассматривать простейший поток, характерный для серверов глобальных вычислительных сетей, обслуживающих большое количество пользователей (например, WWW-серверов). Размер запрашиваемого блока данных для данного случая будем описывать случайной величиной с распределением, близким к распределению размеров файлов, формирующих web-страницы WWW-серверов [9]. Обслуживание запросов к мультимедийным данным моделируются замкнутой системой с фиксированным количеством воспроизводимых потоков N и циклическим методом доступа с периодом

$T = 1$ с. Согласно [10], распределение длин считываемых фрагментов мультимедийных файлов будет определяться как гамма-распределение со средним значением $\xi = 600000$ Б и стандартным отклонением $\sigma = 200000$ Б.

Значения нагрузочных параметров моделирования приведены в табл. 2.

Тип параметров	Запросы к текстовым данным	Запросы к мультимедийным данным
Тип входящего потока	Простейший поток с интенсивностью Λ	Фиксированное количество обслуживаемых потоков N
Длина запрашиваемого блока данных	Гамма-распределение, $\xi = 24$ Кб, $x_{1/2} = 2$ Кб	Гамма-распределение, $\xi = 600000$ Б, $\sigma = 200000$ Б

Таблица 2. Нагрузочные параметры моделирования

Стохастический критерий качества обслуживания запросов к мультимедийным данным выглядит следующим образом:

$$P(K_C > 12) \leq 0,01, \quad (1)$$

где K_C – количество разрывов в потоке на C циклов, $C = 1200$.

Кроме того, в данной работе стохастический критерий качества обслуживания запросов к мультимедийным данным объединяется с требованием качества обслуживания запросов к текстовым данным. Другими словами, в дополнение к (1) требуется ограничение вероятности того, что время пребывания запроса к текстовым данным превысит заданное значение:

$$P(T_D > 2 \text{ сек.}) \leq 0,01, \quad (2)$$

где T_D – время пребывания запроса в НМД.

Оба указанных критерия качества обслуживания (1) и (2) определяют область допустимой нагрузки сервера мультимедиа, а именно область допустимых значений интенсивности входящего потока Λ и числа одновременно воспроизводимых потоков N .

Для сравнения производительности серверов различных архитектур в пересечении их областей допустимой нагрузки, в данной работе вводятся следующие характеристики производительности:

- для запросов к мультимедийным данным – вероятность возникновения разрыва в потоке в данном цикле P_{glitch} ;
- для запросов к текстовым данным – среднее значение времени пребывания запроса в НМД $M(T_D)$.

Сравнение областей допустимой нагрузки

При определении области допустимой нагрузки сервера мультимедиа, для каждой пары значений Λ и N необходимо проверить критерии качества обслуживания (1) и (2). Результаты, полученные для серверов рассматриваемых архитектур, приведены на рис.1.

При указанных выше параметрах моделирования для сервера отдельной архитектуры максимально допустимые значения интенсивности входящего потока Λ и количества одновременно воспроизводимых потоков N составили соответственно $\Lambda_{Max,Part} = 825 \text{ сек}^{-1}$ и $N_{Max,Part} = 150$. При отсутствии нагрузки к мультимедийным данным максимально допустимая для сервера с интегрированной файловой системой интенсивность $\Lambda_{Max,Int} = 2\Lambda_{Max,Part}$, что вызвано предположением равномерного распределения текстовой нагрузки между всеми используемыми НМД. При отсутствии

текстовой нагрузки для сервера с интегрированной файловой системой также имеем двукратное увеличение максимально допустимого количества одновременно воспроизводимых потоков $N_{Max,Int} = 2N_{Max,Part}$. В свою очередь, это обусловлено использованием указанного выше метода размещения мультимедийных данных в адресном пространстве подсистемы ввода/вывода. В отличие от применяемого в [2] метода разбиения мультимедийных фрагментов на блоки фиксированной длины и их циклического размещения по массиву НМД, рассматриваемый метод обеспечивает практически идеальную балансировку мультимедийной нагрузки.

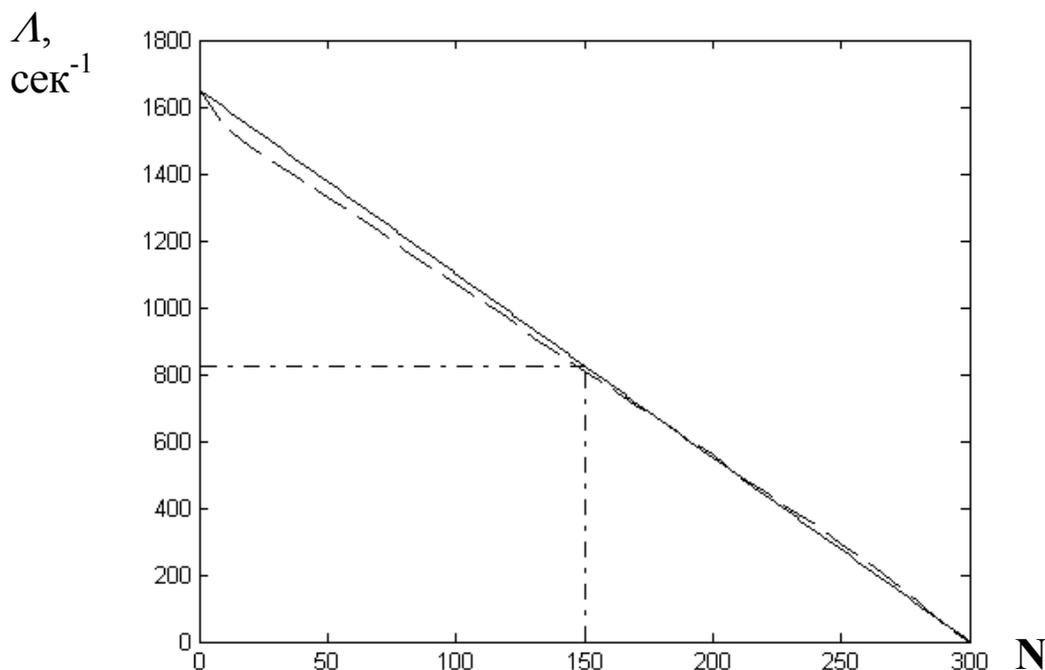


Рис. 1. Области допустимой нагрузки идеального сервера (сплошная линия), сервера раздельной (штрих-пунктирная линия) и интегрированной архитектуры (пунктирная линия).

Благодаря используемому алгоритму дискового планирования, отдающему приоритет обслуживанию запросов к мультимедийным данным, за исключением точки $(\Lambda_{Max,Int}, 0)$ граница области допустимой нагрузки определялась требованием качества обслуживания запросов к текстовым данным (2). При этом зависимость максимально допустимого значения $N_{Max,Int}$ от $\Lambda_{Max,Int}$ близка к линейной (отклонение не превышает 5%). Таким образом, в отличие от результатов, полученных в [2] для алгоритма дискового планирования Cello, рассматриваемая интегрированная файловая система имеет практически идеальную форму области допустимой нагрузки.

Сравнение производительности в области допустимой нагрузки сервера раздельной архитектуры

Для сравнения производительности серверов различных архитектур в работе рассматривалась нагрузка, не превышающая предельную для случая сервера раздельной архитектуры.

На рис. 2 изображена зависимость среднего времени пребывания запросов к текстовым данным $M(T_D)$ от количества воспроизводимых потоков N при фиксированной интенсивности Λ .

Так как сервер раздельной архитектуры использует независимые массивы НМД для хранения и организации доступа к данным каждого типа, его производительность для запросов к текстовым данным не зависит от мультимедийной нагрузки. Для сервера интегрированной архитектуры среднее время пребывания $M(T_D)$ возрастает с увеличением количества воспроизводимых потоков N . Такое поведение объясняется двумя факторами. Во-первых, в данном случае пропускная способность подсистемы ввода/вывода разделяется между обеими нагрузочными компонентами. Во-вторых, рассматриваемый алгоритм дискового планирования отдает приоритет обслуживанию запросов к мультимедийным данным для выполнения требования их непрерывного воспроизведения. Это приводит к увеличению времени ожидания запросов к текстовым данным, и, как следствие, их времени пребывания в НМД.

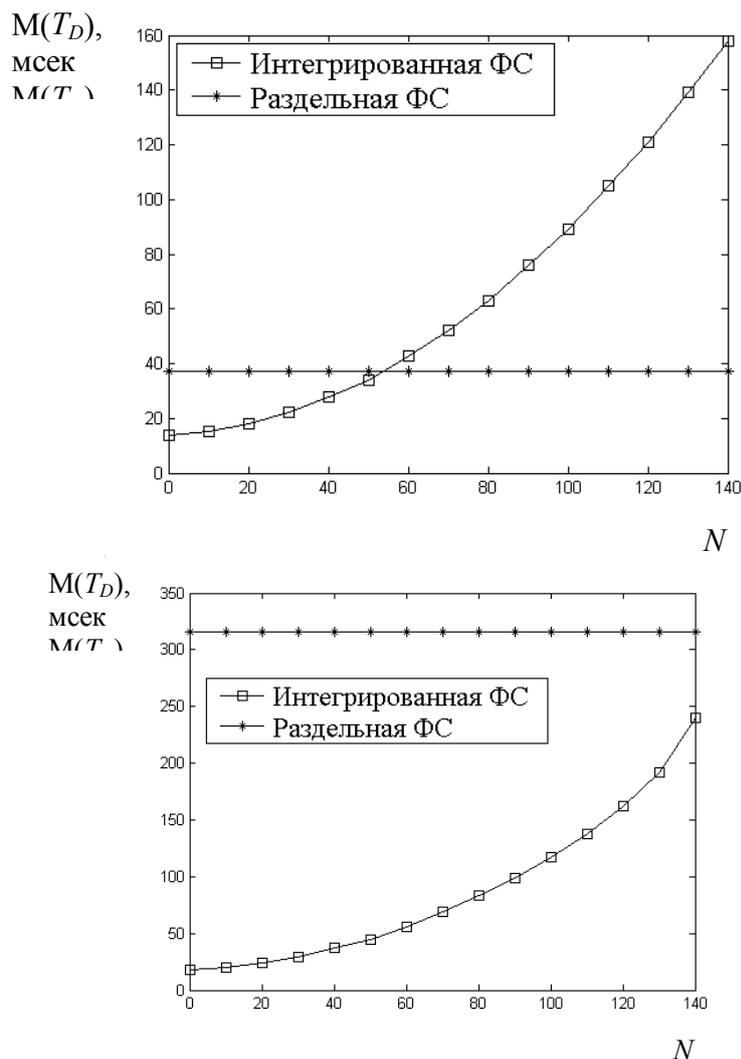


Рис. 2. Зависимости производительности серверов различных архитектур для запросов к текстовым данным от количества одновременно воспроизводимых потоков. Верхний рис.: $\Lambda = 600 \text{ сек}^{-1}$, нижний рис.: $\Lambda = 800 \text{ сек}^{-1}$.

Однако в области умеренных мультимедийных и высоких текстовых нагрузок сервер интегрированной архитектуры обладает лучшей производительностью благодаря тому, что на обслуживание каждого запроса выделяется больше ресурсов.

На рис. 3 изображены зависимости среднего времени пребывания запросов к текстовым данным $M(T_D)$ от интенсивности входящего потока Λ при фиксированном количестве воспроизводимых потоков N .

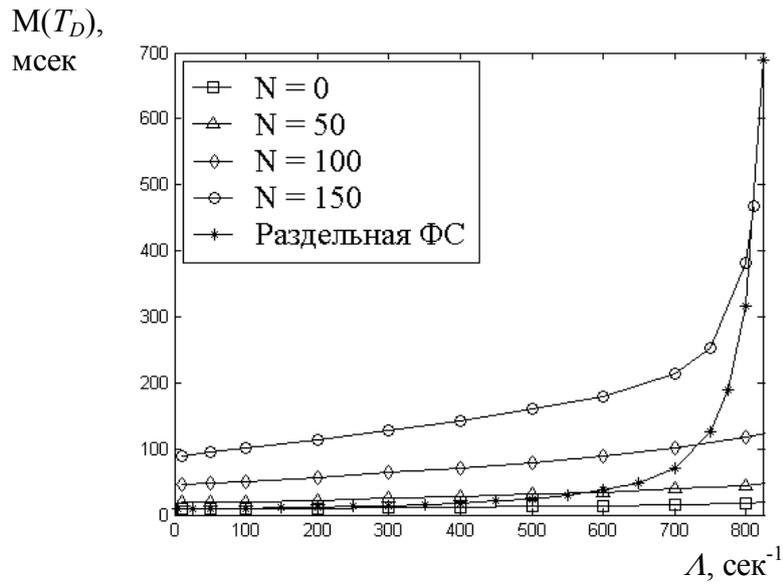


Рис. 3. Зависимости производительности серверов различных архитектур для запросов к текстовым данным от интенсивности входящего потока

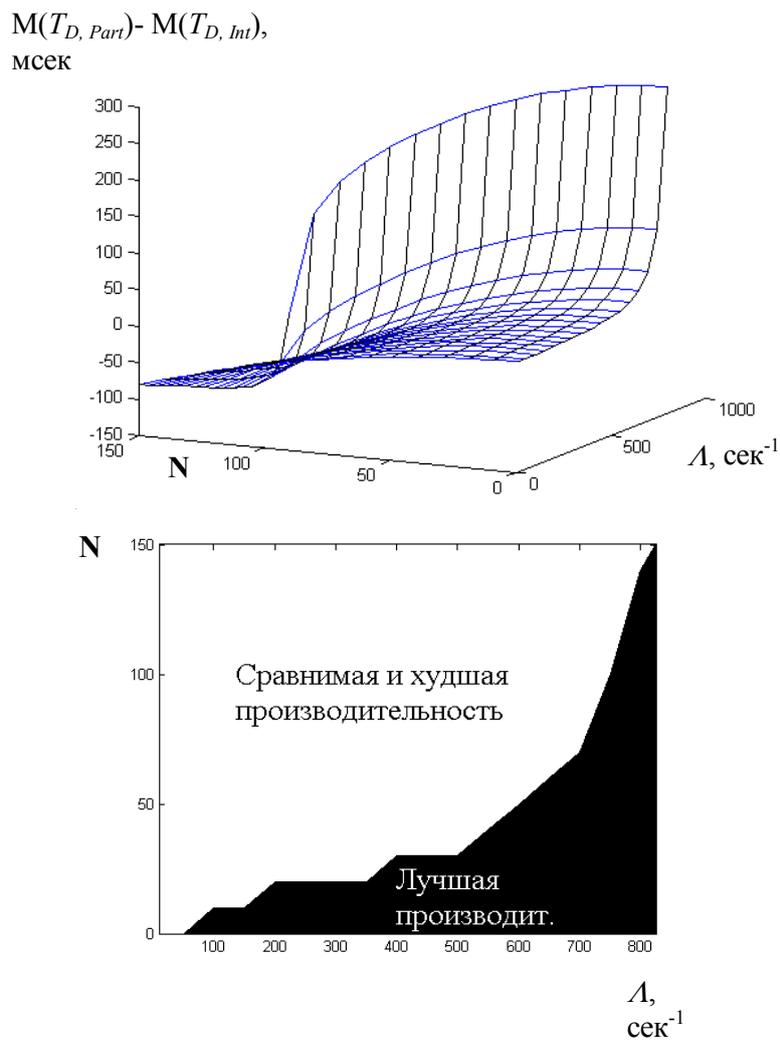


Рис. 4. Разница в производительности серверов различных архитектур для запросов к текстовым данным

Как и ожидалось, для всех рассматриваемых архитектур построения файловых систем среднее время пребывания $M(T_D)$ возрастает с увеличением интенсивности входящего потока Λ . Однако при высоких интенсивностях Λ рост $M(T_D)$ оказывается значительно более быстрым для сервера раздельной архитектуры, чем для сервера интегрированной архитектуры. Представленные зависимости еще раз подтверждают тот факт, что при малых и умеренных мультимедийных нагрузках сервер с файловой системой интегрированной архитектуры имеет сравнимую и лучшую производительность чем сервер раздельной архитектуры.

На рис. 4 показана разница в производительности серверов раздельной и интегрированной архитектур по нагрузке к текстовым данным.

Таким образом, показано, что сервер мультимедиа с файловой системой интегрированной архитектуры имеет лучшую и сравнимую производительность для запросов к текстовым данным в большей части области допустимой нагрузки сервера раздельной архитектуры.

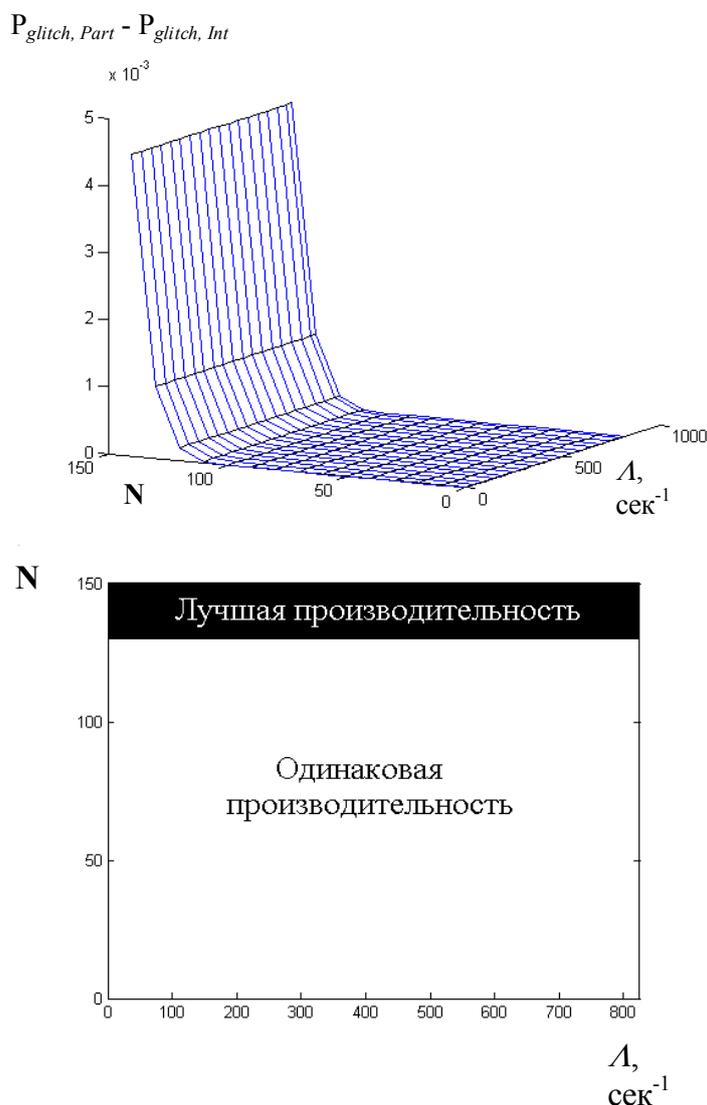


Рис. 5. Разница в производительности серверов различных архитектур для запросов к мультимедийным данным

Кроме того, используемый для сервера интегрированной архитектуры алгоритм дискового планирования обеспечивает непрерывное воспроизведение мультимедийных объектов во всей области допустимой нагрузки сервера раздельной

архитектуры. Таким образом, исходя из критерия качества обслуживания воспроизведения мультимедийных объектов (1), определяющего область допустимой нагрузки, получаем, что сервер мультимедиа с файловой системой интегрированной архитектуры имеет такую же и лучшую производительность для запросов к мультимедийным данным во всей области допустимой нагрузки сервера раздельной архитектуры (см. рис. 5).

Выводы

Для сравнения раздельной и интегрированной архитектур построения файловых систем серверов мультимедиа в данной работе проведен анализ производительности работы дисковой подсистемы ввода/вывода для каждого из случаев. Показано, что сервер мультимедиа с интегрированной файловой системой может поддерживать большее количество клиентов одного типа при малых нагрузках к данным другого типа. При этом рассматриваемая интегрированная файловая система имеет практически идеальную форму области допустимой нагрузки (отклонение не превышает 5%). Так, если для сервера с файловой системой раздельной архитектуры выделено $D1$ и $D2$ дисков для поддержки текстовых и мультимедийных данных соответственно, то при гарантии того же качества обслуживания сервер с интегрированной файловой системой сможет обслуживать в $(D1+D2)/D2$ раз большее количество клиентов, запрашивающих мультимедийные объекты, и выдерживать в $(D1+D2)/D1$ раз большую интенсивность входящего потока к текстовым данным. Более широкая область допустимой нагрузки будет обеспечивать его лучшую производительность при пульсирующих (или "взрывных") нагрузках, часто встречающихся в реальности.

Для сравнения указанных архитектур в области функционирования сервера мультимедиа с файловой системой раздельной архитектуры в работе рассматривалась нагрузка, не превышающая предельную для данного случая, при условии обеспечения требуемого качества обслуживания. Результаты показали, что по производительности для запросов к текстовым данным интегрированные файловые системы сравнимы и превосходят файловые системы раздельной архитектуры в большей части рассматриваемой области и незначительно уступают в оставшейся. Кроме того, используемый для сервера интегрированной архитектуры алгоритм дискового планирования обеспечивает непрерывное воспроизведение мультимедийных объектов во всей области допустимой нагрузки сервера раздельной архитектуры.

Необходимость единой поддержки различных типов данных с различными требованиями к их воспроизведению и наличие механизмов предотвращения их взаимного влияния друг на друга (обслуживание запросов к текстовым данным не должно влиять на качество воспроизведения мультимедийных объектов) приводят к усложнению архитектуры интегрированной файловой системы. В свою очередь, принципы построения файловых систем, оптимизированных под конкретный тип данных, хорошо известны. Поэтому файловые системы раздельной архитектуры просты при разработке и реализации, что, соответственно, снижает стоимость самой файловой системы. Однако более высокая эффективность использования ресурсов и лучшая производительность сервера интегрированной архитектуры позволяет сделать вывод о перспективности данного подхода.

Литература

1. D. James Gemmell, Harrick M. Vin, Dilip D. Kandlur, P. Venkat Rangan. Multimedia Storage Servers: A Tutorial and Survey. // IEEE Computer. 1995.

2. P. Shenoy, P. Goyal, H.M. Vin. Architectural Considerations for Next Generation File Systems. // Proceedings of the Seventh ACM Multimedia Conference, Orlando, November 1999.
3. Таненбаум Э. Современные операционные системы. 2-е изд. СПб.: Питер, 2002.
4. C. Martin, P.S. Narayan, B. Ozden, R. Rastogi, A. Silberschatz. The Fellini Multimedia Storage Server. Multimedia Information Storage and Management, Editor S.M. Chung, Kluwer Academic Publishers, 1996.
5. P. Shenoy, P. Goyal, S.S. Rao, H.M. Vin. Symphony: An Integrated Multimedia File System. // Proceedings of the SPIE/ACM Conference on Multimedia Computing and Networking (MMCN'98), San Jose, CA. January 1998. P. 124–138.
6. Петров А.А. Моделирование дисковой подсистемы мультимедийного сервера при комбинированной нагрузке. Бакалаврская работа. СПбГИТМО, кафедра ВТ. Июнь 2002.
7. P. Shenoy, H.M. Vin. Cello: A Disk Scheduling Framework for Next Generation Operating Systems. // Proceedings of ACM SIGMETRICS Conference, Madison, WI, June 1998. P. 44–55.
8. Hard disk drive specification. Ultrastar 36LZX, Models: DDYS-T36950, DDYS-T18350, DDYS-T09170. Revision 2.1. IBM Corp. 9 June 2000.
9. S. Saroiu, K.P. Gummadi, R.J. Dunn, S.D. Gribble, H.M. Levy. An Analysis of Internet Content Delivery Systems.
10. M. Krunk, H. Hughes. A Traffic Model for MPEG-Coded VBR Streams. // Proceedings ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, Ottawa, Canada, May 1995.

РАСШИРЕНИЕ ВОЗМОЖНОСТЕЙ ЛАБОРАТОРНОГО КОМПЛЕКСА SDK-1.1

А.Н. Лукичев

Рассматриваются области применения, проблемы использования и способы расширения возможностей лабораторного комплекса на основе контроллера-конструктора SDK-1.1.

Основу лабораторного комплекса составляет контроллер-конструктор (микропроцессорный стенд) SDK-1.1 на базе ОКЭВМ фирмы Analog Devices ADuC812. Сам лабораторный комплекс представляет собой совокупность контроллера-конструктора, подключенного к персональному компьютеру, и программного обеспечения для ПК и SDK-1.1. Подключение осуществляется к COM-порту ПК через кабель RS232, комплекс инструментальных программ обеспечивает весь процесс программирования SDK-1.1: компиляцию, доставку и запуск программ в SDK-1.1.

Контроллер-конструктор имеет в своем составе устройства для ввода и отображения информации, снабжен блоком питания и может работать автономно от ПК.

Основными областями использования комплекса являются:

- обучение основам вычислительной и микропроцессорной техники, систем управления;
- автоматизация простых технологических процессов и лабораторных исследований;
- макетирование микропроцессорных систем, отладка программного обеспечения для систем на базе широко распространенного ядра Intel MCS-51;
- радиолюбительство, управление бытовой техникой.

SDK-1.1 с успехом используются во многих университетах и колледжах России и стран СНГ. На кафедре ВТ СПбГИТМО(ТУ) SDK-1.1 составляют основу лабораторного практикума по таким специальным дисциплинам, как информационно-управляющие системы, системы ввода-вывода, операционные системы реального времени.

Контроллер SDK-1.1 оснащен устройствами для обработки и формирования аналоговых и дискретных сигналов, а также приспособлениями для замыкания выходных цепей на входные и симуляции внешних событий. Это дает возможность использовать SDK-1.1 для обучения основам цифровой обработки сигналов, в качестве контрольно-измерительной панели при проведении экспериментов в различных областях, при настройке оборудования и т.п., а также для формирования аналоговых и дискретных сигналов с заданными параметрами в процессе управления различными объектами. Контроллер может обеспечить первичную обработку и хранение результатов измерений с возможностью последующей передачи их на ПК.

Поскольку центральный вычислитель стенда, ОКЭВМ ADuC812, выполнен на базе вычислительного ядра Intel MCS-51 и полностью совместим с ним, SDK-1.1 может быть применен для разработки и отладки встроенного программного обеспечения систем на базе MCS-51. Использованию SDK-1.1 в качестве прототипа еще не созданной системы способствует также наличие в стенде устройств для симуляции входных воздействий, механизмов контроля реального времени, энергонезависимой памяти.

SDK-1.1 имеет дискретный 20-тиразрядный параллельный порт, позволяющий организовать взаимодействие с различными цифровыми устройствами. Он устроен таким образом, что позволяет независимо использовать 2 блока по 8 линий порта, а также каждую из 4 старших линий. Таким образом, стенд способен взаимодействовать как с 8-ми разрядными, так и с 16..20-ти разрядными устройствами. Кроме того, электрические параметры порта позволяют использовать его в "нестандартных"

решениях. В частности, имеется опыт организации параллельного взаимодействия с группой температурных датчиков DS18B20 (фирмы Dallas Semiconductor), подключенных к 8-ми младшим линиям порта, через интерфейс 1-Wire™. Каждый датчик был соединен 2 проводами: с соответствующей информационной линией ("разрядом") порта и с общей точкой. Питание каждого датчика осуществлялось непосредственно через информационную линию, обмен происходил через нее же. Во время получения текущего значения температуры датчик DS18B20 потребляет ток до 1.5 мА, соответственно, суммарный потребляемый в такой системе ток (вытекающий из порта) достигал 12 мА. Подключение такого рода устройств к SDK-1.1 позволяет использовать стенд в качестве управляющего блока в простых, собранных "на коленках" системах.

Однако главной областью применения микропроцессорного стенда SDK-1.1, безусловно, является обучение различным аспектам встраиваемой вычислительной техники. Студенты имеют возможность ознакомиться на практике с проектированием, программированием, отладкой и использованием создаваемой ими из "конструктора" SDK-1.1 системы, убедиться в сложности и научиться находить пути решения задачи эффективного управления ресурсами, освоить механизмы реального времени и многозадачности. Разнообразные устройства, входящие в состав стенда, позволяют изучить круг вопросов, связанных с организацией взаимодействия с ними через типичные интерфейсы, применяемые во встраиваемых вычислительных системах.

Структура стенда

В своем составе контроллер SDK-1.1 имеет:

- вычислительное ядро на основе ОКЭВМ ADuC812, FLASH-памяти (8 Кб) и внешнего ОЗУ объемом до 512Кб;
- гальванически изолированный порт RS232C для связи с ПК;
- 8, 16, 20-тиразрядный порт дискретного ввода-вывода;
- аналоговый порт ввода на базе 8-миканального 12-тиразрядного высокоскоростного АЦП со встроенным термодатчиком и возможностью работы в режиме ПДП;
- аналоговый порт вывода на основе двух 12-тиразрядных ЦАП;
- EEPROM-память емкостью 640 байт;
- второй блок EEPROM-памяти емкостью до 32 Кб, подключенный к вычислителю через интерфейс I²C;
- 3 16-тиразрядных таймера-счетчика с внешними счетными входами (возможностью подачи сигналов через переключатели стенда) и блоком захвата/сравнения для измерения параметров и/или формирования дискретных сигналов;
- сторожевой (Watchdog) таймер;
- жидкокристаллический индикатор для вывода текста с поддержкой русского алфавита (2 строки по 16 символов);
- линейку из 8 сигнальных светодиодов;
- акустический пьезокерамический излучатель;
- матричную клавиатуру на 16 клавиш;
- переключатели-стимуляторы 10-ти линий параллельного порта, сигналов от внешних источников прерываний, коммутаторы сигналов с выходов ЦАП на входы АЦП;
- часы/календарь с возможностью подключения внешней батареи питания.

Несмотря на обилие механизмов и устройств, входящих в SDK-1.1, контроллер имеет габариты 13 см × 12.5 см × 2 см, выполнен в прочном пластмассовом корпусе и

снабжен защитой от возможных повреждений, связанных с постоянным использованием студентами.

Программирование

Подготовка программ для микроконтроллера ADuC812 осуществляется на языке программирования Си на ПК в обычном текстовом редакторе или средах программирования, IDE, предназначенных для разработки программ для MCS-51, например в среде Keil μ Vision. После этого программа компилируется в исполняемый модуль, который доставляется в стенд с помощью разработанного инструментального программного обеспечения.

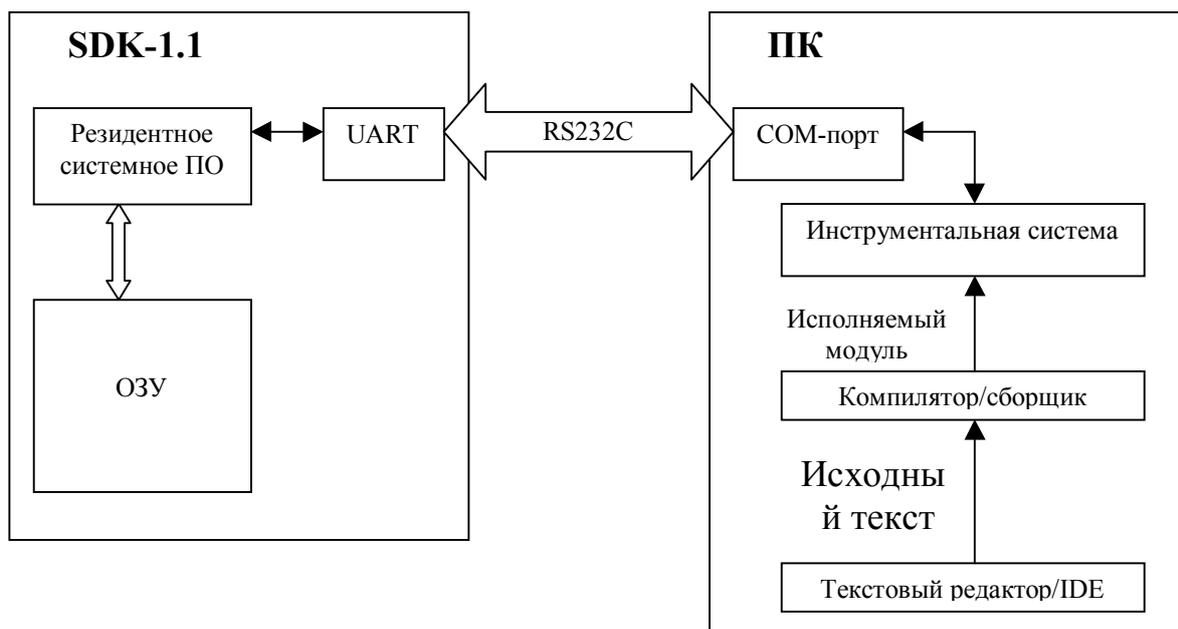


Рис. 1. Процесс подготовки программ для SDK-1.1

Резидентное системное ПО стенда обеспечивает прием, развертывание исполняемого модуля в ОЗУ и запуск на исполнение принятой программы. Для размещения программ пользователя в SDK-1.1 доступно 56 Кб ОЗУ. Остальная память (всего до 512 Кб) может быть использована как память данных.

Расширение возможностей

В ходе эксплуатации лабораторного комплекса SDK-1.1 удалось сформировать перечень предложений, направленных на преодоление ограничений и расширение его учебных и инструментальных возможностей, а также на его применение в смежных областях.

Во-первых, дискретный и аналоговый порты контроллера не снабжены гальванической изоляцией, что ограничивает их использование в области управления различными объектами. К этому же ограничению относятся и электрические параметры портов: аналоговый порт принимает (формирует) в качестве сигнала напряжение в диапазоне от 0 до +5В, дискретный порт способен формировать напряжение от 0 до +5В на старших 4 линиях и от 0 до +3.3 В на 16-ти остальных. В связи с этим было предложено спроектировать ряд типовых схем и аппаратных модулей сопряжения контроллера с датчиками и объектами управления.

Во-вторых, в процессе использования стенда часто возникала необходимость создания собственных цифровых устройств, которые бы взаимодействовали с SDK-1.1. Так как SDK-1.1 выполнен в достаточно компактном корпусе, а создаваемое устройство с большой вероятностью представляет собой несколько микросхем, разъемов и пассивных элементов и выполнено в лучшем случае на макетном поле ("слепыше") без корпуса, эксплуатация такого комплекса затруднена из-за постоянной необходимости подключения и отключения устройства от SDK-1.1 при переносе, хранении, начале и завершении использования. Это влечет за собой постоянные неудобства вроде обрывов проводников, срыва или повреждения элементов устройства. Для решения этой проблемы было предложено дополнить стенд достаточно объемным макетным полем, располагающимся в одном корпусе с печатной платой самого SDK-1.1 и допускающим установку микросхем как в DIP-, так и в SMD-корпусах.

В-третьих, при программировании SDK-1.1 иногда возникает необходимость сохранять программу и после выключения питания. Так как стенд создавался для массового использования студентами, такая возможность в базовой его конфигурации имеется лишь за счет замены содержимого FLASH-памяти вычислителя (программы пользователя обычно располагаются в ОЗУ стенда). Однако ее размер составляет всего 8 Кб, что не всегда достаточно для более-менее серьезных программ, таких как курсовые и дипломные проекты, бакалаврские и другие исследовательские работы. К тому же во FLASH-память ADuC812 может быть записан только один образ, т.е. либо одна программа, либо несколько программ, но скомпонованных в один файл. Это затрудняет обновление отдельной программы в наборе, так как, во-первых, для этого необходимо заново компоновать все программы из набора, во-вторых, невозможно перезаписать часть FLASH-памяти ADuC812 без стирания всего ее содержимого. В связи с этим было предложено:

а) рассмотреть возможность сохранения программ в энергонезависимой памяти (EEPROM) стенда, увеличив ее объем до 32 Кб;

б) модифицировать системное ПО стенда таким образом, чтобы оно могло обеспечить развертывание и запуск программ как из ПК, так и из EEPROM;

в) дополнить резидентное системное ПО библиотекой стандартных системных функций для работы с устройствами и механизмами стенда, "разгрузив" тем самым пользовательские программы и существенно уменьшив объем исполняемых модулей.

Результатом решения первой из вышеперечисленных задач явилось создание аппаратных модулей расширения серии SDX – модуля аналогового ввода-вывода SDX-06 и дискретного ввода-вывода SDX-03. SDX-03 имеет 8 оптически изолированных дискретных входов, 4 оптически изолированных транзисторных (коммутация напряжения до +30 В) выходы и 4 релейных выходы. Кроме того, модуль снабжен собственным блоком питания. SDX-06 позволяет обеспечить формирование напряжения в диапазоне –10..+10В для двух ЦАП, а также измерение напряжений –5..+5В для четырех каналов АЦП.

Кроме того, был разработан модуль сопряжения стенда SDK-1.1 с теплофизическими ячейками для получения в комплексе прибора для определения тепловых свойств твердых веществ. Контроллер обеспечивал управление цепями нагрева, снятие показаний датчиков и контроль времени эксперимента, а также первичную обработку и хранение результатов. Имелась возможность передачи результатов на ПК для последующего исследования. Данный прибор демонстрировался на выставке в рамках VII научно-методической конференции стран СНГ "Современный физический практикум – 2002".

Создана модификация стенда SDK-1.1/М, в состав которой входит дополнительно макетное поле размером 12 см × 10.2 см, размещенное в одном корпусе с печатной

платой SDK-1.1 и допускающее двусторонний монтаж компонент, в частности, монтаж микросхем в QFP- и DIP-корпусах.

Ведутся работы по созданию операционного окружения пользовательских программ на основе резидентной библиотеки системных функций, обеспечивающего управление ресурсами SDK-1.1, загрузку и запуск программ из ПК и serial EEPROM. Создана модификация стенда SDK-1.1/E, имеющая serial EEPROM емкостью 32 Кб. В состав операционного окружения включены функции простейшей файловой системы, базирующейся на serial EEPROM, позволяющие размещать в энергонезависимой памяти стенда не только исполняемые модули, но и файлы данных.

Постоянно создаются и совершенствуются методические материалы по применению SDK-1.1 в учебном процессе. В частности, разработаны курсы лабораторных работ с использованием SDK-1.1, снабженные методическими пособиями. Созданы примеры программ для работы с устройствами стенда в виде исходных текстов на языке Си. Инструментальное ПО стенда постоянно обновляется и доступно для загрузки через Интернет.

Заключение

На данный момент в области проектирования встроенных вычислительных систем число и квалификация российских разработчиков во многом уступают зарубежным. Сократить этот разрыв можно было бы за счет подготовки нового поколения российских специалистов и повышения квалификации имеющихся кадров. Разработанный коллективом лаборатории микропроцессорной техники кафедры ВТ СПбГИТМО(ТУ) учебный лабораторный комплекс SDK-1.1 в сочетании с учебно-методическими материалами уже несколько лет с успехом используется при подготовке специалистов в области проектирования встроенных вычислительных систем как в России, так и в других странах СНГ.

Литература

1. Ключев А.О., Кустарев П.В., Платунов А.Е. Инструментальные и учебные контроллеры семейства SDK. // Компоненты и технологии. 2002. №5. С. 96–99.
2. Ключев А.О., Платунов А.Е. Встроенные инструментальные средства современных микроконтроллеров. // Электронные компоненты. 2002. №7. С. 94–97.
3. Сайт направления "Информационно-управляющие системы" кафедры вычислительной техники СПбГИТМО (ТУ). URL: <http://embedded.ifmo.ru>

ОПТИМИЗАЦИЯ КОНФИГУРИРУЕМЫХ ПАРАМЕТРОВ МАРШРУТИЗАТОРА

С.А. Макеев, Т.И. Алиев

Эффективность функционирования маршрутизируемых вычислительных сетей в значительной степени определяется значениями конфигурируемых параметров маршрутизатора, настройка которых осуществляется в зависимости от выполнения условий доставки сообщений.

К конфигурируемым параметрам, в первую очередь, относится используемый протокол маршрутизации и временные интервалы формирования служебных пакетов.

Разработанные авторами аналитические и имитационные модели на основе математического аппарата теории массового обслуживания позволяют оценить эффективность функционирования процесса маршрутизации, обосновать выбор того или иного протокола маршрутизации, настроить конфигурируемые параметры протокола маршрутизации.

Методика анализа сети базируется на методе декомпозиции сети и сводится к расчету отдельных узлов РСМО как СМО [1,2]. Под сетью в данном случае понимаются узлы (маршрутизаторы), источники нагрузки (ЛВС) и каналы передачи данных.

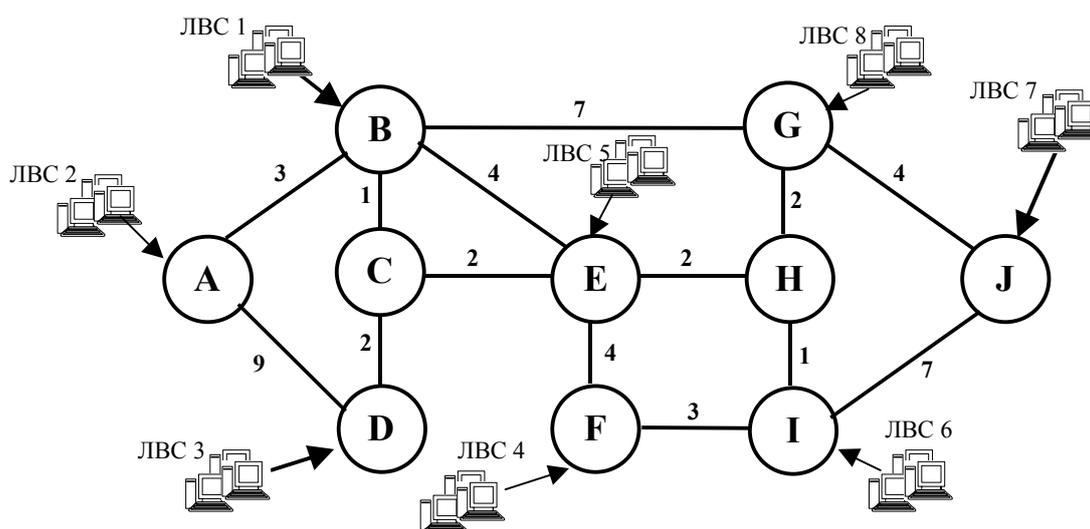


Рис.1. Пример маршрутизируемой сети

Выполнение или невыполнение условия доставки сообщения зависит от адекватности таблицы маршрутизации или, другими словами, от того, насколько правильно выбран маршрут отправки сообщения. Адекватность таблиц маршрутизации прямо зависит от временного интервала рассылки сообщений содержащих маршрутную информацию.

Задачу оптимизации можно сформулировать следующим образом:

$$\lambda \rightarrow \min, \quad \forall t_i \rightarrow \max, \quad \text{при } t_i \leq t_i^*,$$

где t_i^* – максимальный интервал времени между сообщениями i -го типа, при котором выполняются условия доставки сообщений.

Для расчета трафика создаваемого протоколом маршрутизации была получена следующая аналитическая зависимость:

$$\lambda = \sum_{j=1}^M [m_j \sum_{i=1}^H \frac{N_i (l_{IP} + L_i + n_i l_i)}{n_i t_i}],$$

где M – количество узлов в сети, H – количество типов служебных сообщений; L_i – длина заголовка сообщения i -го типа; l_i – количество байт для описания одного маршрута в сообщении i -го типа; $l_{IP} = 20$ байт – длина заголовка IP пакета; n_i – максимальное количество маршрутов, описываемых в одном сообщении i -го типа; N_i – общее количество маршрутов, которые необходимо описать; m_j – количество копий, рассылаемое узлом j ; t_i – интервал времени между сообщениями i -го типа.

Подставив значения переменных для протоколов IGRP и OSPF [3, 4] и предположив, что время рассылки сообщений Hello 3 раза меньше, чем время рассылки сообщений LS для протокола OSPF, получим

$$\frac{\lambda_{IGRP}}{\lambda_{OSPF}} \approx 1,6.$$

Очевидно, что применение протокола OSPF для сети, представленной на рис. 1, снизит трафик служебной информации.

Следующим важным моментом является выбор временных интервалов рассылки служебных сообщений. Для этого:

1. в нескольких узлах изменим входные интенсивности, чтобы несколько сегментов сети изменили свои характеристики, и установим временной порог, когда условия доставки сообщений перестанут выполняться;
2. обновим маршрутные таблицы и дождемся установления стационарного режима в сети;
3. будем выполнять п.1 и п.2 до тех пор, пока не соберем достаточно статистических данных для установления временного интервала рассылки служебных сообщений.

Для оценки t_i^* была разработана имитационная модель, позволяющая имитировать сетевой IP-трафик и работу маршрутизатора, задавать параметры маршрутизатора, количество узлов и топологию сети.

Маршрутизатор представлен следующим образом:

- входящее сообщение попадает в очередь Q_1 ;
- учитывая дисциплину обслуживания ДОО, процессор выбирает заявку, анализирует ее заголовок;
- процессор P_1 определяет направление для отправки сообщения и поддерживает актуальность таблиц маршрутизации путем обмена служебными сообщениями с другими узлами;
- после обработки заголовка сообщения и выбора исходящего направления сообщение попадает в очередь ожидания канала Q_2 .

Примем, что загрузка очереди Q_1 и процессора P_1 не превышает 80%, тогда:

- время обработки заголовка пакета процессором P_1 $\beta \leq 1,25\lambda$, где λ – суммарная интенсивность входного потока;
- объем входного буфера очереди Q_1 $V_{\text{вх. буфера}} \geq 1,25nl_{\text{пакета}}$, где $l_{\text{пакета}}$ – длина пакета, n – число входных каналов связи.

Время пребывания пакета в очереди Q_2 ожидания канала на несколько порядков больше, чем суммарное время пребывания пакета в очереди Q_1 и время обработки заголовка пакета процессором P_1 , поэтому этими значениями можно пренебречь.

Маршрутизатор, согласно определению компании Cisco – "это устройство, использующее одну и более метрик для определения оптимального пути передачи сетевого трафика на основе информации сетевого уровня" [5]. По существу, маршрутизатор представляет собой компьютер с необходимым программным обеспечением и устройствами ввода/вывода. Маршрутизатор выполняет две основные

функции: переключение трафика и обслуживание среды, в которой он работает. Обе функции можно реализовать на одном и том же процессоре, но это вовсе не обязательно.

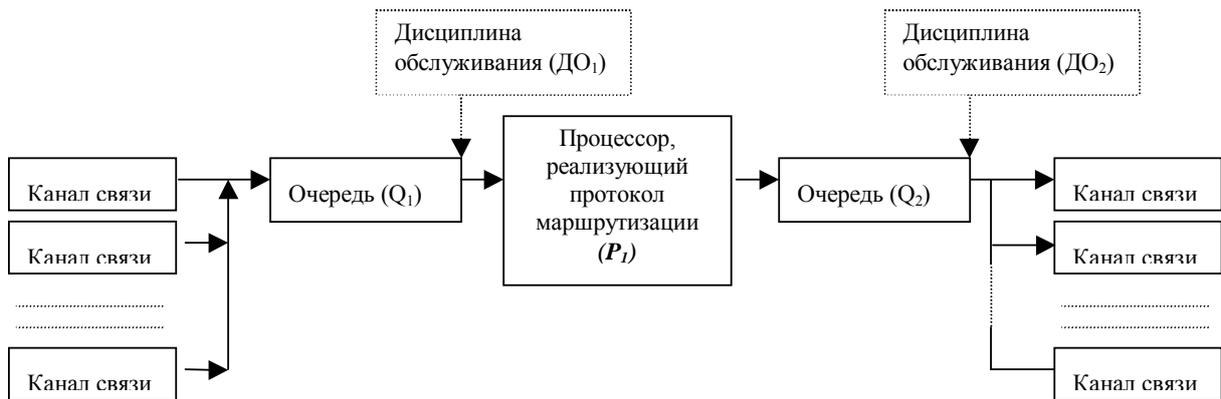


Рис. 2. Модель маршрутизатора

Источники нагрузки, задавались интенсивностями поступления заявок. Интенсивность входного потока сообщений можно рассматривать как сумму трех потоков, а именно:

- поток сообщений от локальной сети, подключенной к данному маршрутизатору;
- поток, создаваемый сообщениями от других узлов сети;
- поток, создаваемый служебными сообщениями протокола маршрутизации, использующимися для обновления маршрутной информации в узлах.

С помощью этой модели результаты для сети, представленной на рис.1, были получены результаты, изображенные на рис. 3.

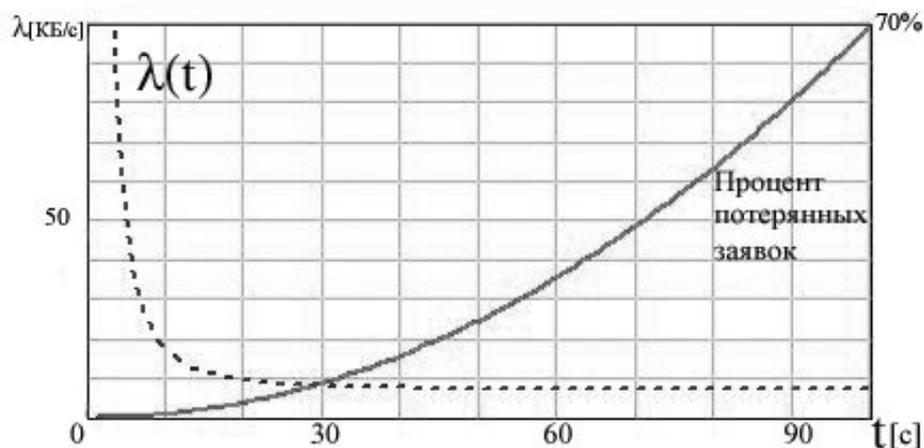


Рис. 3. Зависимость интенсивности служебного трафика и процента потерянных заявок от периода рассылки служебных сообщений

Степень адекватности таблицы маршрутизации (рис. 4) представлена отношением количества маршрутов, на момент обращения имеющих метрику, совпадающую с метрикой, записанной в маршрутной таблице, к количеству устаревших маршрутов.

Таким образом, для сети, представленной на рис.1, целесообразнее всего использовать протокол маршрутизации OSPF с интервалом рассылки LS сообщений, равным 30 с.

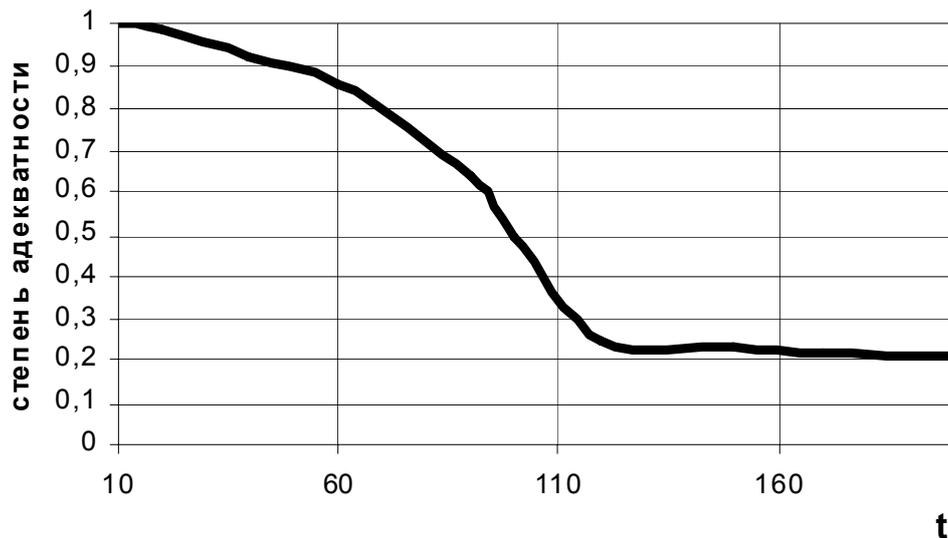


Рис. 4. Степень адекватности маршрутных таблиц в зависимости от периода рассылки служебных сообщений

Для оценки верхней границы допустимого количества узлов при заданной интенсивности поступления служебных сообщений (λ^*) с учетом условий доставки сообщений предположим, что все узлы соединены между собой, тогда получим следующую формулу:

$$M = \sqrt{\frac{\lambda^*}{\sum_{i=1}^H \frac{N_i (l_{IP} + L_i + n_i l_i)}{n_i t_i}}$$

Предложенная методика позволяет оценивать:

- правильность выбора протокола маршрутизации;
- временные интервалы рассылки маршрутных сообщений;
- максимально допустимое количество узлов в сети.

Результаты исследования могут быть использованы как при проектировании маршрутизируемых сетей, так и для устранения узких мест и снижения служебного трафика.

Литература

1. Майоров С.А. Основы теории вычислительных систем. М.: Высш. школа, 1978.
2. Клейнрок Л. Теория массового обслуживания. М.: Машиностроение, 1979.
3. Семенов Ю.А. Протоколы и ресурсы Internet. М.: Радио и связь, 1996.
4. Moy J. OSPF Version 2 (RFC 1583). Proteon Inc., March 1994.
5. Олифер В., Олифер Н. Компьютерные сети. Принципы, технологии, протоколы. СПб: Питер, 2001.

РАЗДЕЛЕНИЕ СИСТЕМНЫХ И ПРИКЛАДНЫХ ЗАДАЧ ПРИ РАЗРАБОТКЕ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ

В.В. Окулевич

Повышение эффективности разработки информационно-управляющих систем является актуальной задачей. Одним из способов повышения эффективности производства является разделение труда.

В ИУС можно условно выделить системную и прикладную части. Системная часть ИУС – это та часть, которая непосредственно не относится к главной функциональности системы. Прикладная часть системы как раз и обеспечивает выполнение пользовательских процессов. Можно сказать, что системная часть является базисом для прикладной части. Такая точка зрения часто выражается в графических представлениях самых разных систем.

Поэтому первым признаком, по которому следует определять принадлежность решаемой задачи к тому или иному уровню системы, является ответ на вопрос, понятна и важна ли она для конечного пользователя системы.

Правильное разделение задач, решаемых в системе, на пользовательские и прикладные весьма важно при распределении работ по проектированию и разработке ИУС. Специфика задач системного уровня состоит в том, что они предоставляют платформу для решения прикладных задач. Таким образом, невозможно обеспечить решение прикладных задач, не решив ряд системных вопросов. С другой стороны, решение даже самых интересных системных задач должно быть востребованным. Без этого они не принесут никакой пользы, и дальнейшее их решение будет вряд ли возможным по материальным соображениям. Таким образом, очевидно, что существует неразрывная взаимосвязь между задачами системного и прикладного уровней.

Разделив системную и прикладную части архитектуры ИУС, далее можно проводить разделение системы на более простые функциональные блоки с явно (и, по возможности, точно) определенными связями между блоками (интерфейсами). За счет этого будет возможно добиться удачного распределения работы по созданию частей системы между группами разработчиков и, далее, между разработчиками в группах для максимально эффективного использования их способностей.

Кроме того, это должно улучшить процесс производства ИУС, поскольку каждый из разработчиков будет заниматься разработкой своей части системы в соответствии со своей специализацией.

Важно заметить, что подобное деление создает условия для повторного использования элементов предыдущих разработок. Процесс подразумевает фиксацию интерфейсов между элементами системы (в первую очередь, за счет явного выделения и фиксации языков в системе). Таким образом, создаются условия для выделения независимых компонент и удачных решений, которые могут накапливаться и применяться в последующих разработках.

Использование разделения труда в разработке ИУС относительно ново из-за специфики разрабатываемых систем. В настоящее время многие специалисты данной области не могут погружаться в современные технологии и инструменты. Один и тот же специалист вынужден заниматься очень большим количеством задач. Одновременно приходится заниматься и прикладной, и системной составляющей. Такая технология разработки ведет к тому, что использование современных средств или совсем отсутствует, или неэффективно, или попросту наивно и даже неуместно.

Решение системных и прикладных задач в общем случае требует различных знаний, умений и навыков. Скажем, для организации обмена данными пользовательских задач между собой в распределенной сетевой системе требуется

использование сервисов системного уровня. При этом приходится решать вопросы организации взаимодействия асинхронных задач, отслеживать целостность и непротиворечивость пользовательских данных, организовывать обработку ошибок передачи, предоставлять удобный (т.е., как правило, достаточно простой) интерфейс обмена данными для пользовательских задач. При этом даже для систем, решающих разные классы задач, оказывается много общих элементов в системных уровнях. Поэтому существует значительный потенциал для организации повторного использования удачных решений в системном уровне.

Пользовательские задачи решают проблемы прикладного уровня. Для систем, применяющихся в различных областях, характерно значительное различие в типах используемых алгоритмов. Важно отметить, что в каждой предметной области существуют свои приемы для обмена информацией (языки) и решения специфических для данной области задач.

Специалисты в той или иной прикладной области общаются между собой и размышляют, как правило, на специальном языке. Для химика – это язык химических реакций и формул, для математика – математический язык, для специалиста в области железнодорожной автоматике – язык релейно-контактных схем и логико-временных зависимостей. Для эффективного решения данными специалистами прикладных задач предпочтительно предоставить им возможность выражать решения именно на нем.

С другой стороны, тот вычислительный базис, который предоставляется современной вычислительной техникой, плохо подходит или вообще не подходит для формализации прикладных задач. В самом деле, самыми подходящими средствами для создания программ с минимальной избыточностью являются компиляторы и интерпретаторы различных языков программирования. Эти современные инструменты являются обычно мощными и универсальными средствами. Однако оборотной стороной универсальности является сложность, влекущая за собой потребность в специальных знаниях и умениях для эффективного использования. Таким образом, возникает разрыв между языком предметной области и языком вычислительной платформы.

Под языком предметной области здесь понимается язык, используемый специалистами в данной области для описания типовых задач и их решения.

При разработке используется некоторое множество языков вычислительной платформы. Это те языки, на которых осуществляется реализация алгоритмов предметной области в виде, пригодном для подготовки к исполнению. Для множества современных универсальных вычислительных платформ языками исполнительской платформы являются языки программирования низкого и высокого уровней. Кроме того, в их качестве могут использоваться различные языки описания аппаратуры (Verilog, HDL, AHDL и др.).

Очевидно, что в общем случае язык исполнительской платформы не совпадает с языком предметной области. Устранение данного разрыва обеспечивает операционная среда. Операционная среда – это набор механизмов, обеспечивающих выполнение прикладных задач на данной вычислительной платформе. При этом считается, что решение прикладной задачи реализовано в терминах языка описания решения прикладной задачи. Операционная среда основывается на языке исполнительской платформы. Можно говорить о том, что верхняя граница операционной среды – это тот базис, который предоставлен разработчикам прикладных алгоритмов. Ясно, что в зависимости от особенностей решаемой прикладной задачи операционная среда будет сильно изменяться.

Эффективность устранения разрыва при помощи операционной среды зависит от:

- масштаба системы,
- типа производства – штучное, мелкосерийное, крупносерийное,

- особенностей разрабатываемой системы,
- характеристик прикладных задач,
- характеристик вычислительной платформы,
- квалификации и количества разработчиков,
- сроков разработки.

Операционная среда должна удовлетворять ряду требований:

1. соответствовать предметной области;
2. удовлетворять требованиям, предъявляемым к целевой системе (требованиям к ИУС);
3. использовать эффективный и удобный язык для описания задачи;
4. быть эффективной на этапе исполнения;
5. позволять пользователю операционной среды абстрагироваться от особенностей конкретной используемой вычислительной платформы;
6. быть переносима на различные вычислительные платформы;
7. позволять строить масштабируемые конфигурации;
8. быть достаточно универсальной и гибкой для внесения новых возможностей.

Рассмотрим перечисленные требования подробнее. Соответствие предметной области подразумевает предоставление полного базиса для решения данного класса прикладных задач. Полнота базиса предполагает возможности для описания всех необходимых элементов предметной области и требуемых операций над ними.

Под удовлетворением требований, предъявляемым к целевой системе, понимается способность данной операционной среды выполнять ограничения, наложенные на систему на допустимом наборе прикладных задач. Перечислим типичные требования для информационно-управляющих систем:

- безопасность,
- надежность,
- функционирование в реальном масштабе времени,
- непрерывность функционирования,
- учет возможных типов использования системы с точки зрения присутствия оператора в контуре управления (автоматическая/автоматизированная система),
- специфические способы разработки ИУС и ее отладки.

Использование эффективного и удобного языка для описания прикладных задач означает, что разработчики прикладного уровня могут с достаточной степенью эффективности и не испытывая неудобств применить его для решения целевых задач. Невыполнение данного требования создает трудности и взаимное непонимание прикладных и системных разработчиков. Кроме того, создаются дополнительные сложности при анализе элементов системы с помощью специалистов в области прикладных задач, что влечет за собой трудность верификации системы и поиск ошибок в прикладных задачах.

Эффективность на этапе исполнения означает, что при функционировании системы на обеспечение работы механизмов операционной среды затрачиваются приемлемое количество ресурсов.

Под абстрагированием от особенностей вычислительной платформы подразумевается сокрытие деталей реализации того программно-аппаратного комплекса, который используется в целевой системе. Несоблюдение данного требования опасно сосредоточением внимания прикладных разработчиков и экспертов на несущественных деталях системного уровня. Кроме того, оно и ведет к созданию решений, использующих уникальные особенности той или иной вычислительной платформы. Это может затруднить выполнение требования переносимости, которое актуально при работе в современном мире с динамично изменяющейся вычислительной базой. Невозможность переноса существующих наработок на другую

платформу может стать причиной затруднения или даже отказа от их использования. Перенос потребует переработки и последующей проверки, а, значит, и материальных затрат.

Требование масштабируемости в применении к операционной среде означает возможность решать на ее основе задачи различной размерности. При увеличении размерности решаемой прикладной задачи должна существовать возможность эффективно дополнить операционную среду необходимыми особенностями.

И, наконец, требование универсальности и гибкости для внесения новых возможностей подразумевает возможность простого изменения операционной среды при необходимости реализации новых свойств системы. Данное требование можно трактовать как открытость операционной среды для модификации.

Разумеется, удовлетворение всем этим требованиям сразу является невозможным делом. Большинство заявленных требований противоречат друг другу. Поэтому разработчикам приходится всегда искать некоторый разумный компромисс, основываясь на своем видении проблемы. Кроме того, степень соответствия заявленным требованиям может меняться и, как правило, меняется, в ходе более-менее крупного проекта. Часто необходимость удовлетворить одно, более приоритетное требование заставляет разработчиков делать это за счет других параметров системы. К примеру, обнаруженная невозможность обеспечить функционирование системы в реальном масштабе времени может быть разрешена с использованием низкоуровневых, специфичных для данной вычислительной платформы механизмов. Таким образом, следует говорить о наличии в каждой конкретной разработке собственных приоритетов в отношении требований к операционной среде.

Литература

1. Фаулер М., Скотт К. UML в кратком изложении. Применение стандартного языка объектного моделирования: Пер. с англ. М.: Мир, 1999. 191 с., ил.
2. Страуструп Б. Язык программирования C++. 3-е изд. / Пер. с англ. СПб-М.: Невский Диалект - Издательство БИНОМ, 1999. 911 с., ил.

ОПТИМИЗАЦИЯ РАБОТЫ ДИСКОВОЙ ПОДСИСТЕМЫ СЕРВЕРА МУЛЬТИМЕДИА

А.А. Петров, М.С. Косяков, Т.И. Алиев

Введение

Развитие компьютерных технологий сделало возможным хранение и передачу мультимедийной информации, такой как аудиозаписи, видеоролики, голосовые сообщения и т.д. Стали появляться приложения, позволяющие пользователям работать как с обычными текстовыми (или непотоковыми) данными, так и с мультимедийными (потоковыми) данными. При этом характеристики данных различных типов и методы доступа к ним значительно различаются между собой. Так, мультимедийные файлы требуют непрерывности своего воспроизведения. Если в воспроизводимом потоке произойдет задержка очередного фрагмента, передаваемая информация может быть искажена или даже полностью утеряна.

В свою очередь, текстовые файлы не накладывают подобных требований на механизмы доступа и предполагают использование иных дисциплин дискового планирования. Кроме того, мультимедийные данные обычно характеризуются большими требованиями к пропускной способности подсистемы ввода/вывода. Вследствие этого для данных каждого типа различны методы их размещения в адресном пространстве подсистемы ввода/вывода и алгоритмы контроля качества обслуживания [1].

При организации сервера, обеспечивающего хранение и предоставляющего доступ к данным обоих типов, возможны два принципиально различных подхода [2]:

- сервера отдельной архитектуры, выделяющие независимый пул ресурсов (дисковый массив и оперативную память) для данных каждого типа;
- сервера интегрированной архитектуры, предоставляющие свои ресурсы для совместного использования данными обоих типов.

Так как методы организации хранения и доступа к данным конкретного типа хорошо известны [3], сервера отдельной архитектуры просты при разработке и реализации. В этом случае запросы к данным различных типов обслуживаются независимо, и предотвращается их взаимное влияние друг на друга: обслуживание запросов к текстовым данным не влияет на качество воспроизведения мультимедийных объектов. В то же время необходимость единой поддержки различных типов данных с различными требованиями к их воспроизведению приводит к усложнению архитектуры интегрированного сервера. Однако из-за возможности динамического выделения ресурсов данный подход является предпочтительным в случае изменяющейся во времени нагрузки. Кроме того, в данном случае на обслуживание каждого запроса выделяется больше ресурсов, что обеспечивает лучшую производительность сервера. Указанные преимущества позволяют авторам сделать вывод о перспективности такого подхода.

В связи с этим в работе проводится анализ влияния методов размещения данных в адресном пространстве подсистемы ввода/вывода и алгоритмов дискового планирования на производительность сервера интегрированной архитектуры. Кроме того, осуществляется поиск значений функциональных параметров, увеличивающих производительность системы. В данном случае под производительностью системы мы понимаем площадь области допустимых значений числа мультимедийных потоков и интенсивности потока запросов к текстовым данным, которые могут поддерживаться системой в рамках требуемого качества обслуживания.

Описание исследуемой системы

Для размещения мультимедийных данных в адресном пространстве подсистемы ввода/вывода соответствующие мультимедийные объекты разбиваются на фрагменты согласно постоянному шагу времени [3], т.е. каждый фрагмент соответствует фиксированному времени воспроизведения T . Подобная схема разбиения непрерывного потока позволяет реализовать циклический метод доступа к данным, когда в течение каждого цикла с периодом T для каждого из потоков считывается ровно один фрагмент, независимо от различий в степени сжатия и скорости воспроизведения. При этом каждый фрагмент целиком размещается на одном диске, и фрагменты, принадлежащие одному объекту, распределяются в циклическом порядке по всем дискам. Так как степень сжатия мультимедийных данных может меняться, фрагменты различаются по размеру даже в пределах одного и того же объекта. Это приводит к необходимости поддержки со стороны файловой системы функций чтения/записи данных экстендами, а не кластерами фиксированной длины и, как следствие, усложнению ее структуры. Однако, как показано в [4], в отличие от традиционных файловых систем при подобном размещении мультимедийных данных в адресном пространстве подсистемы ввода/вывода достигается лучшая производительность и масштабируемость сервера. Кроме того, такой подход сильно упрощает реализацию дискового планирования и алгоритма контроля допустимости.

Текстовые данные хранятся также экстендами, обслуживание запросов к ним определяется выбранным методом дискового планирования.

С учетом приведенной схемы размещения данных в адресном пространстве подсистемы ввода/вывода в [5, 6] были рассмотрены возможные методы дискового планирования и показано, что за счет выбора смешанного динамического дискового планирования FCFS с дисциплиной обслуживания очереди запросов к диску SCAN возможно добиться максимизации производительности дисковой подсистемы. Согласно указанному алгоритму, назначение на обслуживание запросов к текстовым данным осуществляется модулем контроля допустимости динамически, в зависимости от установленных критериев качества обслуживания. При этом для выбора запросов из очереди используется дисциплина FCFS, позволяющая обеспечить минимальную дисперсию пребывания запросов этого типа в системе ввода/вывода. В отличие от рассмотренного в [2] алгоритма дискового планирования Cello, указанный алгоритм не разбивает цикл обслуживания на подпериоды, соответствующие обслуживанию отдельно запросов к текстовым данным и отдельно – к мультимедийным. Тем самым обеспечивается смешанное обслуживание. В течение цикла запросы обслуживаются независимо от их типа согласно дисциплине SCAN.

Однако вопрос выяснения оптимальных значений параметров данного метода дискового планирования, а именно длительности периода обслуживания T , остался вне поля зрения авторов. В связи с этим далее проводится исследование влияния длительности периода обслуживания T на производительность системы и определяется такое значения данного параметра, которое позволило бы максимизировать производительность исследуемой системы при всех возможных нагрузках.

Параметры модели

Поставленная задача была решена с использованием предложенной в [5] имитационной модели дисковой подсистемы сервера мультимедиа. В модели были реализованы описанные выше методы размещения данных и дискового планирования. В качестве структурных параметров модели использовались параметры,

соответствующие жесткому диску IBM DDYS-T36950 [7]. Нагрузочные параметры описывались следующим образом:

- часть нагрузки, создаваемая запросами к текстовым данным, описывается как пуассоновский поток с интенсивностью поступления запросов λ ;
- часть нагрузки, создаваемая запросами к мультимедийным данным, описывается как замкнутая система с фиксированным мультипрограммным уровнем N ;
- размеры фрагментов запрашиваемых данных описываются случайной величиной, имеющей гамма-распределение, с параметрами, соответствующими распределению размеров html-файлов для запросов к текстовым данным и распределению размеров фрагментов MPEG2-объектов для мультимедийных запросов;
- номера секторов, адресуемых запросами, описываются случайной величиной, имеющей равномерное распределение.

При оценке производительности моделируемой системы использовались стохастические требования к качеству обслуживания запросов:

- вероятность того, что время пребывания запроса к текстовым данным в дисковой подсистеме сервера u_D превысит $u_D^{zap} = 2$ с, должна оставаться менее 0,01: $P(u_D > u_D^{zap}) < 0,01$;
- вероятность того, что количество разрывов g_K в потоке мультимедийных данных превысит значение $0,01K$ на K фрагментов, должна оставаться менее 0,01: $P(g_K > 0,01K) < 0,01$.

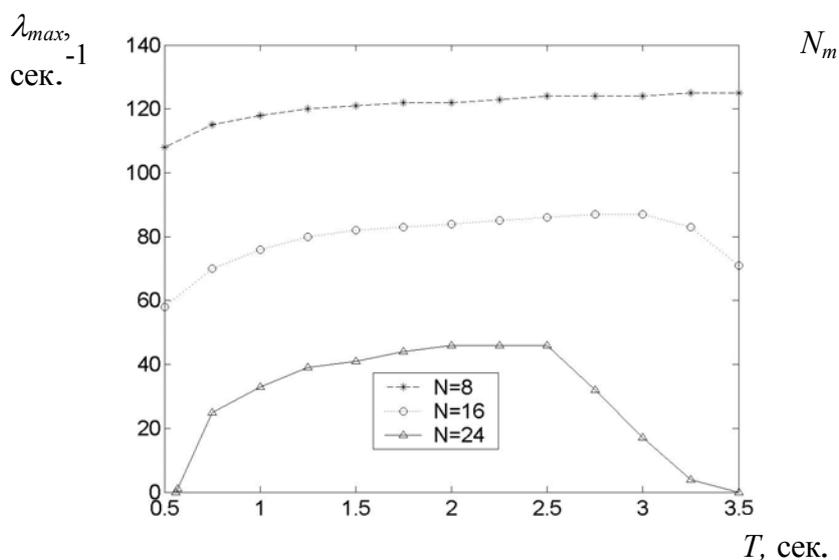
Результаты моделирования

На рис. 1а представлены графики зависимости максимально допустимой интенсивности запросов к текстовым данным λ_{max} , которая может поддерживаться сервером в пределах требуемого качества обслуживания, от длины периода обслуживания T при различных количествах воспроизводимых мультимедийных потоков N . По графикам видно, что с возрастанием T до порогового значения T_0 происходит увеличение максимально допустимой интенсивности запросов к текстовым данным, а при дальнейшем росте T – ее снижение. Кроме того, для графиков, соответствующим различным значениям N , обнаруживаются следующие закономерности:

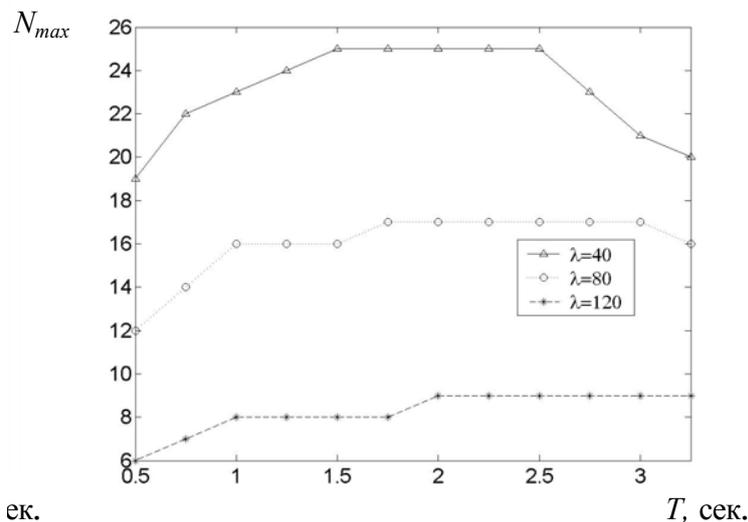
- скорость увеличения и скорость снижения λ_{max} зависят от величины N – чем больше количество воспроизводимых потоков, тем быстрее происходит и возрастание, и убывание максимально допустимой интенсивности запросов к текстовым данным;
- пороговое значение T_0 тем меньше, чем больше мультимедийная составляющая нагрузки на дисковую подсистему.

Такое поведение системы объясняется следующим образом. При увеличении периода обслуживания соответственно увеличиваются размеры запрашиваемых фрагментов мультимедийных данных. Вследствие того, что эти фрагменты располагаются на диске непрерывными экстендами, уменьшаются накладные расходы на позиционирование головок при чтении мультимедиа информации, т.е. увеличивается производительность дисковой подсистемы при чтении мультимедийных данных. При этом проявляется и другой фактор – увеличение времени считывания одного мультимедийного фрагмента ведет к увеличению времени ожидания запросов к текстовым данным и, следовательно, увеличению времени их пребывания в системе. Однако, как показывают исследования, суммарный эффект все же положителен: снижение накладных расходов на позиционирование при обращении к мультимедийным данным позволяет обслужить больше текстовых запросов, поэтому заметного падения качества обслуживания запросов к тестовым данным не происходит.

В результате с ростом периода обслуживания T производительность системы увеличивается.



а

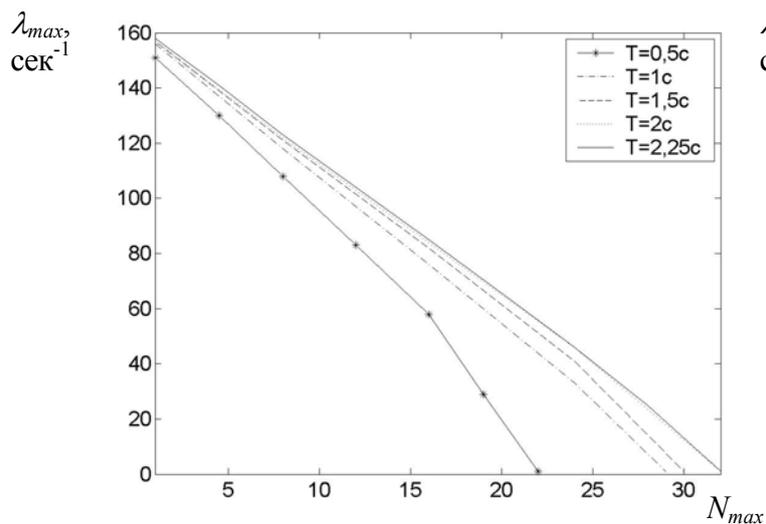


б

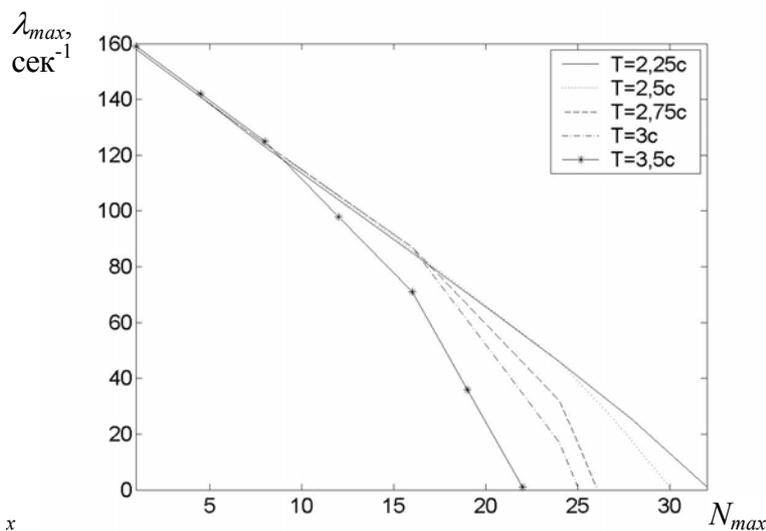
Рис. 1. Зависимость максимально допустимой интенсивности запросов к текстовым данным (а) и максимального количества одновременно воспроизводимых потоков (б) от длительности периода обслуживания

Снижение производительности дисковой подсистемы при достижении периодом обслуживания T некоторой пороговой величины T_0 связано с используемыми функциональными параметрами системы. Выбор набора текстовых запросов из очереди осуществляется в соответствии с дисциплиной FCFS. После этого в течение периода времени T запросы обоих типов обслуживаются в порядке, определяемом дисциплиной SCAN. При этом получается так, что в пределах периода запросы обслуживаются безотносительно времени их поступления, и некоторая доля запросов, пришедших в предыдущем цикле и выбранных на обслуживание в начале текущего цикла, будет обслужена в его конце. Таким образом, при приближении величины периода обслуживания к гарантированному времени пребывания запросов к текстовым данным u_D^{rap} и, тем более, по превышении этого значения доля текстовых запросов, время пребывания которых превысит требуемое вследствие назначения их на

обслуживание в конце периода, будет расти. Кроме того, количество запросов к текстовым данным, обслуживание которых будет приходиться на конец периода, напрямую зависит от величины мультимедийной нагрузки. При большем количестве поддерживаемых мультимедийных потоков вероятность того, что пришедший текстовый запрос будет обслужен в конце периода, больше. Поэтому снижение величины λ_{max} происходит в результате использования указанного дискового планирования: с увеличением продолжительности периода обслуживания T сверх порогового значения T_0 производительность дисковой подсистемы продолжает возрастать, однако качество обслуживания запросов к текстовым данным начинает резко снижаться, что ограничивает возможность дальнейшего увеличения нагрузки.



а

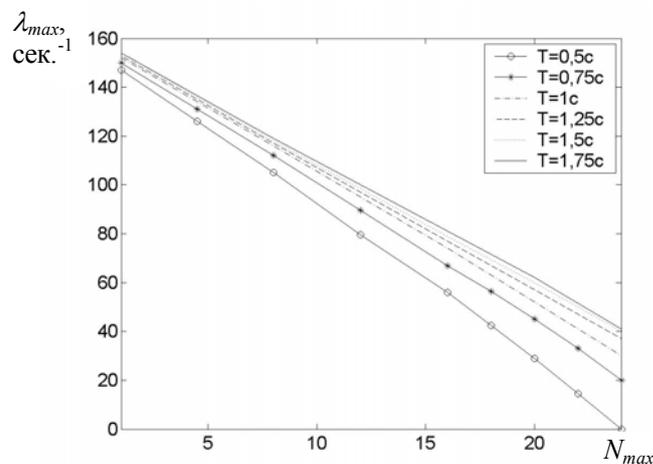


б

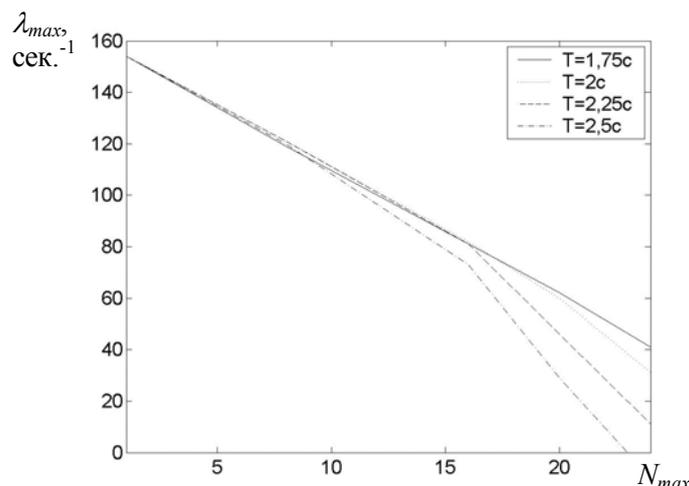
Рис. 2. Области допустимых нагрузок, соответствующие различным значениям периода обслуживания T , для $u_D^{gap}=2$ сек.

На рис. 1б представлены графики зависимости максимального количества мультимедийных потоков N_{max} , которое может поддерживаться сервером в пределах требуемого качества обслуживания, от длины периода обслуживания T при различных интенсивностях запросов к текстовым данным λ . Очевидно, что в целом наблюдаемая картина подтверждает выводы, сделанные на основе графиков, представленных на

рис. 1а. С увеличением значения T происходит возрастание, а затем убывание производительности системы. Причем увеличение и снижение производительности происходит тем быстрее, чем меньше интенсивность запросов к текстовым данным и чем больше количество поддерживаемых мультимедийных потоков. Меньшее значение пороговой продолжительности периода обслуживания T_0 также соответствует большей величине мультимедийной нагрузки.



а



б

Рис. 3. Области допустимых нагрузок, соответствующие различным значениям периода обслуживания T , для $u_D^{rap}=1,5с$

Приведенные выше графические зависимости позволяют установить закономерности влияния длительности периода обслуживания T на производительность дисковой подсистемы, однако определение на их основе оптимальной величины периода обслуживания T_{opt} не является очевидным. Для этого были построены графики, определяющие форму области допустимой нагрузки и соответствующие различным продолжительностям периода обслуживания. На рис. 2а и 2б по оси Ох откладывается количество мультимедийных потоков, а по оси Оу – интенсивность запросов к текстовым данным. Таким образом, все точки, расположенные левее и ниже кривой, соответствуют нагрузкам, которые могут быть поддержаны дисковой подсистемой с требуемым качеством обслуживания; все точки, расположенные правее

и выше кривой, соответствуют нагрузкам, поддержка которых с требуемым качеством обслуживания невозможна.

При построении областей допустимых нагрузок длительность периода обслуживания T изменялась с шагом 0,25 с. В результате было установлено, что при увеличении T до значения $T_{opt} = 2,25$ с происходит расширение (рис. 2а), а по превышении этого значения – сужение области допустимых нагрузок (рис. 2б). Как уже было упомянуто выше, первый указанный эффект объясняется увеличением производительности дисковой подсистемы, а второй – снижением качества обслуживания запросов к текстовым данным. Таким образом, использование $T_{opt} = 2,25$ с в качестве значения длительности периода обслуживания T позволяет максимизировать размер области допустимых нагрузок, а следовательно и производительность системы.

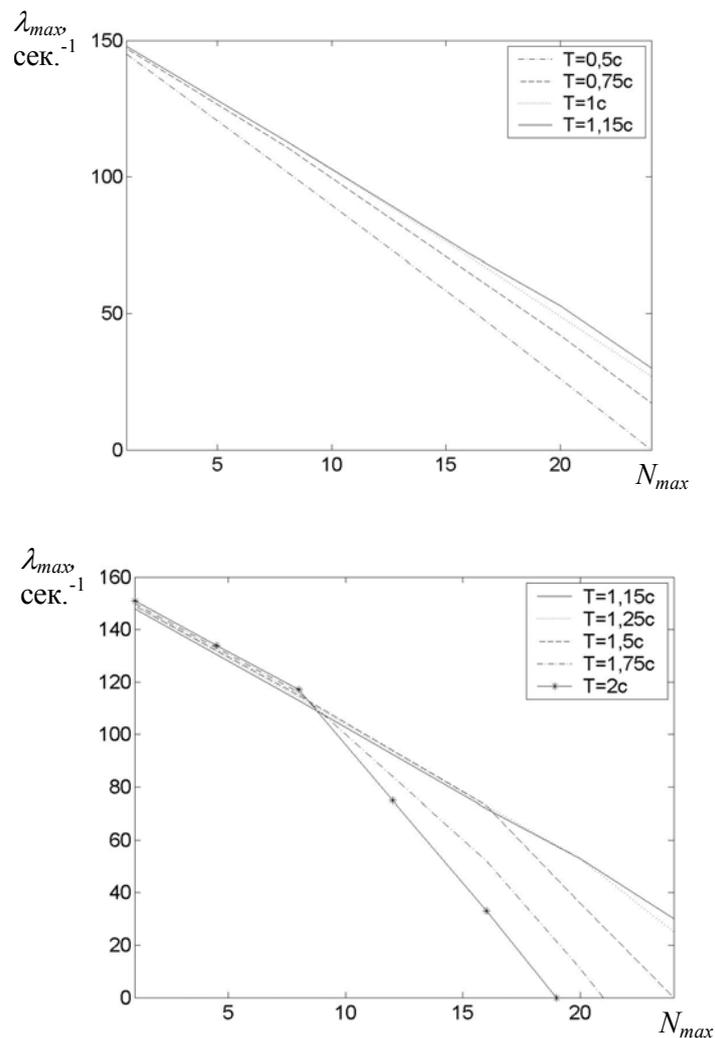


Рис. 4. Области допустимых нагрузок, соответствующие различным значениям периода обслуживания T , для $u_D^{rap} = 1c$

Выявленное значение T_{opt} лишь незначительно превышает требуемое время пребывания запросов к текстовым данным в дисковой подсистеме u_D^{rap} . Это объясняется следующим образом. Запрос гарантированно обслуживается за период T . Следовательно, при значениях T , меньших u_D^{rap} , время обслуживания запроса всегда меньше u_D^{rap} , и, следовательно, время пребывания запроса к текстовым данным u_D может превысить u_D^{rap} только в том случае, если запрос долгое время проведет в

очереди. Если же T больше u_D^{gap} , то даже выбранный из очереди запрос вследствие используемой дисциплины SCAN может не быть обслужен так долго, что его результирующее время пребывания u_D превысит u_D^{gap} .

Для подтверждения существования зависимости между T_{opt} и u_D^{gap} были определены значения оптимальной длительности периода обслуживания T_{opt} , соответствующие $u_D^{\text{gap}} = 1,5$ с (рис. 3) и $u_D^{\text{gap}} = 1$ с (рис. 4).

На рис. 3 изображены кривые, отображающие области допустимых нагрузок, соответствующие различным длительностям периода обслуживания T для $u_D^{\text{gap}} = 1,5$ с, а на рис. 4 – $u_D^{\text{gap}} = 1$ с. По графикам видно, что требуемому времени пребывания запроса к текстовым данным $u_D^{\text{gap}} = 1,5$ с соответствует оптимальное значение периода обслуживания $T_{\text{opt}} = 1,75$ с, а требуемому времени пребывания $u_D^{\text{gap}} = 1$ с – $T_{\text{opt}} = 1,15$ с. Здесь следует сделать вывод, что для любых значений требуемого времени пребывания запросов к текстовым данным u_D^{gap} длительность периода обслуживания T_{opt} , позволяющая максимизировать площадь области допустимых нагрузок или, иными словами, производительность системы, составляет около 115% от значения u_D^{gap} .

Выводы

В работе были проанализированы основные аспекты организации дисковой подсистемы сервера, обеспечивающего доступ к текстовым и мультимедийным данным. А именно, были рассмотрены методы размещения данных в адресном пространстве подсистемы ввода/вывода и дискового планирования, позволяющие максимизировать количество одновременно воспроизводимых мультимедийных потоков и поддерживаемую интенсивность запросов к текстовым данным. Кроме этого, на имитационной модели было проведено исследование влияния длительности периода обслуживания T на производительность системы. Полученные результаты позволили сделать следующие выводы:

- при увеличении длительности периода обслуживания T происходит увеличение производительности дисковой подсистемы сервера, а затем более резкое ее снижение;
- скорость изменения производительности системы зависит от интенсивности мультимедийной нагрузки на сервер, при этом она тем больше, чем больше количество поддерживаемых мультимедийных потоков;
- несмотря на то, что падение производительности объясняется снижением качества обслуживания запросов к текстовым данным, это отражается в первую очередь на количестве поддерживаемых мультимедийных потоков – оно должно быть снижено для обеспечения требуемого качества обслуживания запросов к текстовым данным;
- значение продолжительности периода обслуживания T_{opt} , позволяющее максимизировать площадь области допустимых нагрузок на дисковую подсистему сервера, составляет около 115% от требуемого времени пребывания запросов к текстовым данным u_D^{gap} ;
- при значениях длительности периода обслуживания, близких к оптимальному, область допустимой нагрузки близка к идеальной, т.е. форма ее границы близка к линейной.

Литература

1. D. James Gemmell, Harrick M. Vin, Dilip D. Kandlur, P. Venkat Rangan. Multimedia Storage Servers: A Tutorial and Survey. // IEEE Computer, 1995.

2. P. Shenoy, P. Goyal, H.M. Vin. Architectural Considerations for Next Generation File Systems. // Proceedings of the Seventh ACM Multimedia Conference, Orlando, November 1999.
3. Таненбаум Э. Современные операционные системы. 2-е изд. СПб: Питер, 2002.
4. H.M. Vin, S.S. Rao, and P. Goyal. Optimizing the Placement of Multimedia Objects on Disk Arrays. // Proceedings of the Second IEEE International Conference on Multimedia Computing and Systems, Washington, D.C., May 1995. P. 158–165.
5. Петров А.А. Моделирование дисковой подсистемы мультимедийного сервера при комбинированной нагрузке. Бакалаврская работа. СПбГИТМО, кафедра ВТ, Июнь 2002г.
6. G. Nerjes, P. Muth, M. Paterakis, Y. Romboyannakis, P. Triantafillou, and G. Weikum. Incremental Scheduling of Mixed Workloads in Multimedia Information Servers. // Multimedia Tools and Applications Journal, September 2000.
7. Hard disk drive specification. Ultrastar 36LZX, Models: DDYS-T36950, DDYS-T18350, DDYS-T09170. Revision 2.1. IBM Corp. 9 June 2000.

МЕТРИКИ В СОВРЕМЕННОМ ТЕСТИРОВАНИИ

В. Ю. Рахманов, Б.Д. Тимченко

Рассматривается процесс тестирования в составе современных CASE-средств, с целью выявления, систематизации и оценки используемых метрик.

Введение

В технологиях производства программных продуктов (ПП) тестированию отводится роль основного средства обеспечения и контроля качества продукта. Это проявляется в том, что процессы тестирования все глубже интегрируются в проектные методы, а управление тестированием становится важнейшей составляющей управления проектами.

Для демонстрации данного положения рассмотрим V-модель тестирования, в которой стадии тестирования соотносятся с соответствующими стадиями разработки (рис. 1). В результате мероприятия по поиску дефектов сдвигаются на ранние стадии процесса разработки, что позволяет уменьшить затраты на поиск и исправление дефектов в ПП. На рисунке пунктирные линии со стрелками определяют отношение между видом тестовой деятельности и видом проектной деятельности. Например, верхняя пунктирная линия со стрелками показывает, что цель приемочного тестирования заключается в подтверждении требований, а само приемочное тестирование основано на требованиях [4].

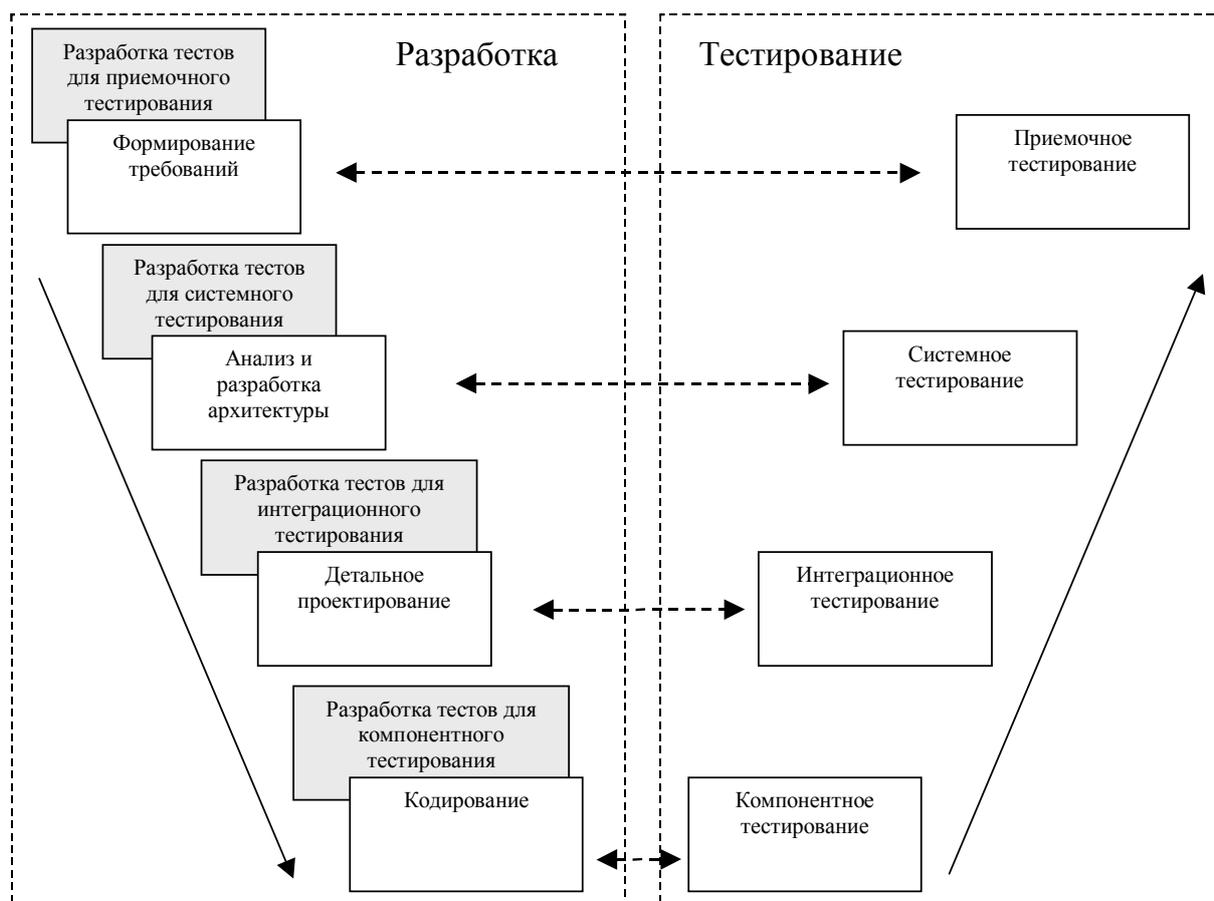


Рис. 1. V-модель тестирования

Для оценки и управления качеством при производстве ПП необходимы метрики, которые должны давать количественные оценки качества конечного продукта. В настоящее время CASE-средства собирают и анализируют некоторые метрики, используемые как для оценки качества ПП, так и для оценки результативности процесса тестирования. К таким средствам, прежде всего, следует отнести Rational Suite компании Rational Software, входящей в состав корпорации IBM с февраля 2003 года, и Together Control Center компании Together Soft, являющейся ныне подразделением Borland. Однако отмечаются проблемы в метрическом обеспечении тестирования, в частности, это касается критериев завершения тестирования. В связи с этим нами предпринят анализ и систематизация метрик, используемых в CASE-средствах.

Метрики тестирования

Множество метрик процесса тестирования в соответствии с [11] будем разделять на два класса: первичные или накопительные метрики и вычисляемые метрики. Накопительные метрики – это числовые значения показателей, полученные на этапе тестирования ПП. Накопительные метрики – это основа анализа тенденций и прогнозирования. К таким метрикам относятся время и стоимость тестирования, количество дефектов. Вычисляемые метрики не могут быть получены непосредственно, а вычисляются на основании первичных метрик. Вычисляемые метрики в большей степени ориентированы на оценку результативности и эффективности тестирования и используются для мониторинга и контроля за процессом тестирования. Примерами таких метрик являются тестовое покрытие, количество дефектов на строку исходного кода, результативность тестирования [8, 9] В табл. 1 представлены основные первичные метрики.

Название метрики	Описание	Примечание
Количество дефектов, найденных на этапе тестирования	Дефекты, найденные на этапе тестирования, позволяют косвенно оценивать квалификацию разработчиков, а также дополнительные затраты, необходимые на исправление и доработку ПП.	Для отслеживания и управления дефектами в ПП используются инструментальные средства – Rational ClearQuest, StarTeam, Bagzilla и др.
Количество дефектов, найденных на этапе эксплуатации	Метрика основана на информации о проблемах, поступивших от пользователей ПП. Служит оценкой качества разработанного программного продукта.	Метрика является апостериорной и используется при оценке качества тестирования
Время тестирования	Метрика оценивает временные затраты на подготовку, выполнение и документирование тестирования. Подготовка тестирования представляет собой деятельность по планированию, разработке тестовых действий и развертыванию тестовой среды. Временные затраты зависят от сложности разрабатываемой системы, степени детальности требований и их документированности, а также опыта тест инженеров. Выполнение тестирования представляет собой тестовые прогоны и документирование результатов тестирования.	Метрика является прямо наблюдаемой и поэтому широко применяется при планировании различных видов тестирования.
Стоимость тестирования	Стоимость тестирования включает в себя затраты на поиск дефектов и амортизацию оборудования для проведения тестирования.	Вместе со стоимостью исправления дефектов метрика часто используется в качестве критерия прекращения тестирования.

Название метрики	Описание	Примечание
Объем тестирования	Для планирования процесса тестирования используется понятие объема тестирования, определенное как планированный тестовый набор, выраженный в количестве разработанных тестов. Эта метрика важна как для планирования, так и впоследствии для вычисления производной метрики, оценивающую полноту тестирования.	В программе Rational Test Manager для планирования тестовых наборов используется план тестирования, который связывает требования к ПП с соответствующими тестовыми случаями.

Таблица 1. Первичные метрики

В отличие от первичных метрик, которые, как упоминалось выше, не применяются для управления тестированием, вычисляемые метрики позволяют оценить результативность процесса. В табл. 2 представлены вычисляемые метрики, используемые в CASE-средствах для оценки результативности (качества) процесса тестирования.

Название метрики	Описание	Примечание
Количество дефектов на строку исходного кода	Общее число дефектов, обнаруженных в программном продукте, в пересчете на количество строк в исходном коде. Данная метрика показывает плотность ошибок в программном продукте.	На основе анализа плотности ошибок в нескольких версиях одного проекта, делают вывод о необходимости продолжения или возможности прекращения работ по тестированию.
Тестовое покрытие (полнота тестирования)	Тестовое покрытие представляет собой отношение планируемого тестового набора к полному тестовому набору. Это важнейший критерий, который имеет отношение к оценке готовности продукции.	Конструктивные методы определения полного тестового набора неизвестны, хотя по этому вопросу имеется большое количество литературы [1, 2]. Практически все, что написано, предполагает структурные методы тестирования, которые позволяют оценить тестовое покрытие операторов, условий и логических путей. К сожалению, это громоздкие процедуры, которые имеют ограниченное применение на практике.
Результативность тестирования: $E = \frac{N_t}{N_t + N_e} \times 100,$ где N_t , N_e – дефекты обнаруженные на этапе тестирования и эксплуатации соответственно.	Для оценки результативности (качества) тестирования используется метрика, основанная на отношении количества дефектов, найденных на этапе тестирования, к общему числу дефектов, найденных на этапе тестирования и эксплуатации.	Эта важная оценка является апостериорной и используется при анализе результатов тестирования для улучшения процесса тестирования.

Таблица 2. Вычисляемые метрики

Из сказанного выше видно, что для оценки готовности ПП в современных CASE-средствах используется метрика, основанная на анализе тестового покрытия. Однако существуют проблемы в определении первичных метрик, необходимых для ее вычисления. Существующие метрики, используемые в CASE-средствах, в основном сводятся к учету дефектов, времени и затрат и поэтому непригодны для оценки

готовности продукта и тем самым не являются критерием завершения тестирования. Несмотря на это, о применении следует сказать более подробно. В зрелых компаниях по разработке ПО, которые используют в качестве средств управления проектами MS Project 2000 или Primavera Enterprise, эти данные составляют статическую основу для планирования проектов и управления ими. Для примера сошлемся на модель качества СММ, где необходимым условием является ведение и анализ статистики проекта, которая иногда называется Knowledge Base. Что касается возможных направлений исследований, то внимание концентрируется на оценке рисков в случае неполного тестирования (т.е. управление тестированием с учетом рисков) и исследовании методологий, основанных на попытках поиска наиболее "важных" тестов, т.е. исследовании методов ранжирования тестов в зависимости от важности (приоритетности) требований.

Литература

1. Майерс Г. Искусство тестирования программ. М.: Финансы и статистика, 1982.
2. Липаев В.В. Тестирование программ. М.: Радио и связь, 1986. 296 с.: ил.
3. Липаев В.В. Обеспечение качества программных средств. Методы и стандарты. М.: Синтег, 2001.
4. Калбертсон Р., Браун К., Кобб Г. Быстрое тестирование. М.: Вильямс, 2002.
5. Evarld E. Mills. Software Metrics. // Software Engineering Institute, 1988
6. Pankaj Jalote. Use of Metrics in High Maturity Organizations. // Software Quality Professional, Mach 2002.
7. Dorothy R. Graham. Measuring the Value of Testing. ESCOM, 2nd April 2001
8. Marnie L. Hutcheson. Software Testing Methods and Metrics. <http://www.ideva.com>
9. Guide of Software Engineering Body of Knowledge. // IEEE Computer Society, May 2001.
10. Testing Metrics – Mercury Interactive,
<http://model.mercuryinteractive.com/quality/metrics/>
11. Rational Unified Process – Rational Software, <http://www.rational.com>

ИСПОЛЬЗОВАНИЕ СТАНДАРТНЫХ БИБЛИОТЕК ВО ВСТРОЕННЫХ СИСТЕМАХ (НА ПРИМЕРЕ СТАНДАРТНОЙ БИБЛИОТЕКИ ЯЗЫКА C++)

В.В. Окулевич, О.Г. Самохвалова

Стандартизация библиотеки языка C++ была предпринята ISO/IEC в августе 1998 года. При этом большое участие в работе в комитете по стандартизации языка принимал сам создатель языка C++ Б. Страуструп. Стандартная библиотека C++ (Standard C++ Library) включает в себя стандартную библиотеку шаблонов (Standard Templates Library, далее STL), потоки ввода/вывода (iostreams), расширенную библиотеку языка C. Стандартная библиотека C++ представляет собой набор спецификаций на достаточно универсальную среду программирования.

Вообще говоря, целью любой библиотеки является создание привычного и удобного контекста для разработчика, накопление и подготовка к повторному использованию удачных решений. Не является секретом, что опытные специалисты практически всегда используют для эффективного решения задач элементы прошлых проверенных наработок. Чем дольше программист решает задачи определенного класса и чем больше он приобретает таких наработок, тем быстрее и качественнее он может решать новые задачи. Хочется отметить, что задачи производства, в том числе и производства программного обеспечения, отличаются от научно-исследовательских задач, в первую очередь, значительно меньшим процентом позволенной новизны [1]. Таким образом, использование предыдущих наработок будет максимально востребовано в организациях, занимающихся производством программного обеспечения для решения четко определенного круга задач. В случае же исследовательских, новаторских проектов количество повторно используемых наработок будет меньше.

Для эффективного использования существующих наработок полезные части, которые могут быть пригодны для новых проектов, должны быть выделены в отдельные библиотеки. При этом можно говорить о библиотеках в виде исходных текстов и библиотеках двоичного уровня. Однако такие собственные библиотеки, создаваемые разработчиками, в большинстве своем обладают рядом недостатков. Они могут быть привязаны к вычислительной платформе и даже к конкретному инструменту разработки (компилятору). При необходимости использования таких наработок в новых условиях приходится довольно много сил тратить на перенос библиотеки (ее портирование). Следует заметить, что в процессе переноса неизбежно вносятся изменения, которые порождают новые ошибки. Кроме того, начинают проявляться незамеченные ранее старые ошибки (например, из-за различия механизмов оптимизации, иного выравнивания данных, другой компоновки).

Возможно, одним из наиболее подходящих способов решения проблемы повторного использования в случае языка C, предшественника языка C++, является использование библиотек, написанных в соответствии со стандартом ANSI C. При этом перенос такой библиотеки в новую среду разработки обходится максимально дешево, иногда совсем без изменений. Платой за такую легкость переноса является отказ от возможностей использовать особенности инструмента или расширения возможностей языка.

Однако при попытках стандартизации "старшего брата" C – языка C++ – возникли новые сложности. При значительном увеличении сложности языка, росте числа сочетаний и богатстве выразительных возможностей возрастает и сложность спецификации всех особенностей языка. Таким образом, мы наблюдаем пример перехода количества в качество: из-за возросшей сложности языка он приобретает новое качество – его практически невозможно полностью специфицировать.

Для решения этой проблемы приходится жертвовать степенью детализации, которая используется в спецификации. Многие теперь отдаются на "откуп" реализации. Таким образом, работа по переносу написанной на С++ библиотеки в новую среду разработки может включать в себя фазу адаптации к инструменту. Впервые обнаруживший необходимость переделки уже готовой программы разработчик обычно бывает весьма озадачен.

Библиотека широко использует механизмы объектно-ориентированной технологии. Кроме того, при спецификации библиотеки широко используется механизм шаблонов. Достоинством этого механизма является удачное сочетание достаточной гибкости и невысокой стоимости на этапе исполнения.

Основные понятия, которые составляют основу библиотеки – это контейнеры, итераторы и алгоритмы. Контейнер – это объект, который содержит множество элементов. Существует несколько различных типов контейнеров. В качестве примеров контейнеров можно привести вектор, список, очередь, ассоциативный массив. Для каждого типа контейнеров существует свой набор разрешенных операций, которые можно выполнять над ним. В этот набор операций включаются те, использование которых не приведет к неэффективному использованию контейнера. Например, для списка элементов вставка нового элемента в середину является сравнительно дешевой операцией. Однако для вектора (массива, в котором к каждому элементу можно обращаться по целочисленному индексу) вставка в середину является весьма дорогой операцией, требующей перемещений других элементов в памяти и изменения их индексов. Поэтому вставка в середину для контейнера список может быть полезной, а для вектора – скорее всего, нет. Таким образом, в набор предоставляемых операций включаются такие, использование которых не будет заведомо неэффективным.

Итераторы являются универсальным способом доступа к элементам контейнера. Набор операций над итераторами позволяет определять работу с элементами контейнера в общем виде, что, в свою очередь, позволяет использовать итераторы при создании алгоритмов. Алгоритмы служат для выполнения действий с элементами контейнера через итераторы, позволяя унифицировать работу с контейнерами разных типов. Важно отметить, что доступ к функциям, специфичным для данного класса, осуществляется через функции-члены класса, в то время как множество общих действий с элементами контейнера (поиск, сравнение, сортировка) выносятся на уровень алгоритмов. Особенности конкретного контейнера (например, обращение по индексу для вектора) локализуются в самом контейнере. Использование этих особенностей может быть полезно для достижения максимальной эффективности.

Кроме того, в стандартную библиотеку входит множество других механизмов, которые позволяют решать часто встречающиеся при разработке приложений проблемы. Например, разработана подсистема потокового ввода-вывода низкого и высокого уровня, при этом предусмотрена мощная система работы с данными различных форматов. Также предложены механизмы для работы с такими структурами данных, которые встречаются при использовании численных методов.

Представляется целесообразным использование библиотеки в следующих условиях. Во-первых, не самым плохим вариантом будет начать использовать ее в тех случаях, когда под рукой нет собственных готовых проблемно-ориентированных библиотек. Также часто приходится начинать разработку в условиях, когда сжатые сроки не позволяют начинать все "с нуля". Зачастую это и не нужно, для многих задач будет совершенно достаточно предусмотренной мощности.

Целесообразно использовать библиотеку в случае решения задач большой размерности. Кроме того, сравнительно легко повторно использовать наработки, реализованные с помощью стандартной библиотеки. Также использование стандартной библиотеки может очень пригодиться, если приходится в течение длительного времени

осуществлять поддержку программы, выпуск новых версий, исправление ошибок. При необходимости модификации программы, использующей временные, специальные решения, через год или два в ней достаточно сложно разобраться. Более того, весьма вероятно, что разбираться в программе придется другому исполнителю. Если же используются, возможно, менее эффективные, но стандартные решения, то в такой программе изменить функциональность или исправить обнаруженную ошибку будет проще.

Следует учитывать, однако, что для эффективного использования мощи библиотеки C++ требуются достаточно квалифицированные программисты. Разработчики должны понимать, что существует достаточно много нюансов при ее применении. Отметим особенности использования STL в случае ограничений размера используемой памяти и времени реакции на события. Такие ограничения характерны в первую очередь для встроенных систем (embedded systems). Однако эти ограничения могут оказаться важными и для многих других изделий класса информационно-управляющих систем.

Ограничение на объем используемой памяти возможно преодолеть следующими образом. Во-первых, следует приблизительно подсчитать, какой объем памяти необходим для функционирования приложения в наихудшем случае. Например, известно, что в используемом в программе контейнере будет содержаться не более чем 100 элементов, при этом не будет элементов размером более 16 байт. С учетом того, что необходим некоторый дополнительный объем памяти для хранения служебной информации контейнера элементов (скажем, по 8 байт на элемент контейнера), получаем $100 \cdot (16 + 8) = 2400$ байт. Размер этой служебной информации контейнера зависит от конкретной реализации среды разработки, его приходится уточнять в каждом конкретном случае.

По умолчанию для выделения памяти элементам, добавляемым в контейнер, используется динамически распределяемая память. При этом необходимо корректно обрабатывать ситуации, когда памяти оказывается недостаточно.

В многозадачных операционных системах возможны ситуации, когда из-за деятельности других задач возможно изменение объема свободной оперативной памяти. Таким образом, приходится учитывать вероятность того, что в какой-то момент (возможно, на второй месяц непрерывной работы системы) при выполнении ранее успешно выполнявшегося запроса задачи о выделении блока динамической памяти система вернет отказ. Чтобы устранить такую возможность, можно выделять память для всех объектов во время инициализации задачи. При этом желательно использовать статические объекты. Другой вариант состоит в том, что библиотеке указывается необходимость использования самостоятельно реализованных программистом так называемых распределителей памяти. Такой распределитель однажды, при запуске, может выделить память необходимого объема, а затем выдает ее участки по мере необходимости.

Что же касается ограничений на время реакции системы, то использование библиотеки не должно создавать особых трудностей. Никаких особенных неожиданностей разработчика здесь, скорее всего, не ждет. Более того, при использовании алгоритмов, представленных в библиотеке, можно обратиться к справочной информации, в которой указана средняя трудоемкость алгоритмов библиотеки.

Для использования языка C++ во встроенных системах был создан проект EC++. Это попытка предложить стандарт языка C++ и библиотеки языка, который, являясь подмножеством (в отличие от CFront 1.2), был бы адаптирован к использованию во встроенных системах. Специальный комитет (Embedded C++ Technical Committee), ведущий работы по этому направлению, был создан в 1995 году в Японии.

Из средств языка C++ в EC++ отсутствуют множественное наследование и виртуальные базовые классы, идентификация типов на этапе выполнения (Run-Time Type Identification), шаблоны, исключения, пространства имен, приведения типов в стиле C++.

Библиотека языка была также сильно изменена. В ней не используются шаблоны и, таким образом, отсутствует STL, нет специальных классов для использования динамической идентификации типов. Кроме того, из библиотеки удалена работа с локализацией, ввод/вывод 16-тиразрядных символов и т.п.

Однако дальнейшее развитие событий показало важность возможности использования шаблонов, пространств имен, исключений, приведения типов в стиле C++ и т.д. Так, пространства имен и приведение типов в стиле C++ объявлены возможными в некоторых реализациях. А в библиотеке-посреднике (Abridged Library) фирмы Dinkum, доступной, например, на платформе Windows CE, кроме упомянутых ранее возможностей, добавлена работа с шаблонами и реализован STL. Вообще же следует помнить, что "каждое свойство C++ вводилось с учетом ограничений по времени выполнения и требуемой памяти" [2].

Поскольку мощность вычислительных устройств со временем увеличивается, то постепенно становится целесообразным использование даже казавшихся недавно дорогими и расточительными инструментов и технологий [1]. Как видно из приведенного примера, это проявляется и в области встроенных систем.

Схожие со стандартной библиотекой C++ средства для решения типовых задач предоставляют такие библиотеки, как MFC, VCL. В этих библиотеках заложены большие функциональные возможности, но следует помнить о том, что они имеют ряд особенностей при использовании. Например, могут использоваться нестандартные правила индексации, или разработчику могут потребоваться дополнительные знания о структуре применяемой библиотеки. Кроме того, данные библиотеки не универсальны, а проблемно-ориентированы. Поэтому такие библиотеки классов, как правило, меньше внимания уделяют вопросам эффективности, что ведет к потенциально менее эффективному решению.

Также следует помнить, что программы, написанные с использованием библиотек классов, специфичных для конкретной среды разработки, будет сложно или вовсе невозможно перенести на другие платформы.

Разработчику полезно помнить о том, что в ранних версиях инструментов C++ они могут столкнуться с первыми версиями реализации стандартной библиотеки. В таких версиях могут подстерегать самые различные сюрпризы, поскольку стандартизация библиотеки C++ произошла сравнительно недавно.

Стандартная библиотека языка C++ является очень эффективным инструментом при ее грамотном использовании. Однако следует помнить о некоторых особенностях ее применения, учитывать существующие ограничения на имеющиеся вычислительные ресурсы и квалификацию разработчиков.

Может быть, самая важная роль, которую играет стандартная библиотека языка C++, состоит в том, чтобы предложить ряд ключевых абстракций, которые ведут к хорошему стилю программирования и помогают разработке простых и понятных программ.

Литература

1. Фокс Д. Программное обеспечение и его разработка. / Пер. с англ. М.: Мир, 1985. – 368 с.
2. Страуструп Б. Язык программирования C++. 3-е изд. / Пер. с англ. СПб-М.: Невский Диалект - Издательство БИНОМ, 1999. 911 с., ил.

ТЕХНОЛОГИИ ПОВТОРНОГО ИСПОЛЬЗОВАНИЯ И РЕИНЖИНИРИНГА И СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

О.Г. Самохвалова

Повторное использование существующих компонентов и создание повторно используемых компонентов являются одной из составляющих деятельности разработчиков, начиная с момента первых попыток усовершенствовать процесс создания ПО. В принципе разработчик, имеющий дело с программным обеспечением, которое создается не на один день, а будет широко применяться, модифицироваться, дорабатываться, всегда имеет дело с технологиями повторного использования и реинжиниринга. Это позволяет ему минимизировать стоимость работ, сократить сроки разработки, улучшить качество сопровождения и т.п.

Тем не менее, несмотря на очевидные выгоды, по ряду сложившихся причин, (например, фрагментарность языков программирования, операционных систем, стандартов и т.д.), немногие компании на сегодня перешли порог 20% так называемого случайного использования ранее созданного кода, хотя при грамотном управлении процессом разработки ПО этот показатель можно довести до 80%.

Современные методы и технологии повторного использования можно разбить на ряд групп в зависимости от различных критериев. Ниже будут приведены наиболее часто используемые классификации.

Методам реинжиниринга в современной теории и практике уделяет значительно меньше внимания. Считается, что система должна изначально разрабатываться с учетом дальнейших изменений. Другими словами, база для реинжиниринга (доработки) должна закладываться в проект изначально. В противном случае может сложиться ситуация, когда доработка системы выльется в ее полную переделку, что фактически можно будет приравнять к разработке с нуля. В дальнейшем в работе в основном будет рассматриваться только это направление как наиболее перспективное.

Все элементы повторного использования в первую очередь можно разделить на коммерческие компоненты и ПО, создаваемое непосредственно разработчиками. Введем для обозначения последнего термин "домашняя заготовка" [6].

Коммерческие компоненты повторного использования разрабатываются как отдельные самостоятельные продукты и ориентированы на то, что они будут в совокупности с другими элементами составлять среду разработки ПО. Сегодня во многих областях распространенным подходом при разработке ПО является максимальное увеличение интеграции с коммерческими компонентами и готовыми продуктами с целью уменьшения процента кода, производимого на заказ. Наибольшего применения сегодня достигли такие продукты, как операционные системы, системы управления базами данных, среды визуальной разработки интерфейсов приложений, библиотеки стандартных функций и прочее.

При разработке систем реального времени и встроенных систем решение об использовании коммерческих компонент должно приниматься еще на стадии анализа и проектирования, поскольку это может значительно изменить архитектуру системы. Часто же использование таких компонент невозможно из-за ограниченности вычислительных ресурсов или невыгодно. Поэтому и сегодня в разработке СВВ и ВС преобладают "домашние заготовки" – компоненты, создаваемые коллективом разработчиков самостоятельно и часто используемые в рамках только данной организации, а то и отдела.

Коммерческим компонентам присущи такие достоинства, как заранее известная стоимость, широко используемые, устоявшиеся технологии, непосредственная доступность, наличие организации, осуществляющей поддержку, аппаратная/программная независимость. К недостаткам относятся частое появление новых версий,

необходимость постоянной оплаты сопровождения, попадание в зависимость от поставщика, потеря эффективности при выполнении, функциональные ограничения, нетривиальность интеграции, отсутствие средств контроля за новыми версиями и сопровождением, функциональная избыточность, за которую приходится платить, недостаточная надежность и стабильность, закрытость кода, несовместимость у различных поставщиков.

"Домашние заготовки" по сравнению с готовыми компонентами предоставляют возможность свободного внесения изменений, дают меньший объем кода, более простую реализацию, большее быстродействие и контроль за разработкой и развитием. Однако они дороги, имеют непредсказуемый срок создания, неопределенную модель сопровождения и часто зависят от платформы.

Практически структуру всего современного ПО можно представить в виде каркаса, в гнездах (специальных ячейках) которого размещаются различные элементы программы, назовем их наполняющими. С этой точки зрения методы повторного использования можно разделить на [3]:

- использующие наполняющие элементы в качестве элементов повторного использования;
- использующие каркас системы в качестве элемента повторного использования;
- сборочные – используют в качестве элементов повторного использования как каркас, так и наполняющие элементы.

Наиболее известным примером применения методов первой группы являются библиотеки подпрограмм. Механизм их неоднократного использования известен уже на протяжении многих десятилетий и в ближайшем будущем, очевидно, не сдаст своих позиций. Так, библиотеки стандартных функций используются во всех высокоуровневых языках программирования. Основной акцент в данном методе делается на повторном использовании кода.

В качестве примера второй технологии можно привести выделение в неизменный каркас работу с объектам Windows. Здесь манипуляции, сопровождающие инициализацию, уничтожение и т.д., выделены в неизменный каркас. Это позволяет программисту сразу приступить непосредственно к работе с изменяющейся частью программы. Данный подход применяется во многих современных инструментах создания программных приложений. Другим примером многократно используемого каркаса может служить организация программ вычислительного эксперимента. Здесь по мере получения новых результатов часто приходится перестраивать вычислительную модель. Поэтому консервативную часть программы выгодно выделить в неизменный каркас, а меняющиеся сделать наполняющими элементами. Таким образом, каркас диктует определенную архитектуру приложения, здесь внимание уделяется повторному использованию дизайна, а не кода.

Сборочные технологии также довольно распространены в настоящее время. Так, например, классы в объектно-ориентированном программировании объединяют в себе важные свойства как наполняющих элементов, так и каркаса. С одной стороны, к методам класса можно обратиться как к подпрограмме для решения задач, с другой стороны, механизм наследования позволяет использовать класс в качестве каркаса для дочерних классов. В принципе, абстрактные классы – не что иное, как каркас.

Еще одним признаком, по которому можно классифицировать методы повторного использования, являются критерии декомпозиции программы (разбивки программы на элементы повторного использования). По этому признаку можно выделить следующие технологии [5]:

- функциональный подход – один элемент отвечает за выполнение одной функции;
- подход, ориентированный на данные – один элемент должен инкапсулировать одну структуру данных;

- подход, ориентированный на события – один элемент должен распознавать и обрабатывать только одно событие;
- проблемно-ориентированный подход – каждый элемент должен соответствовать одной и только одной вещи (проблеме) в мире.

Критерием для классификации методов повторного использования также может служить программный элемент, принимаемый за единицу повторного использования. По этому признаку можно выделить:

- функционально-ориентированный подход,
- объектно-ориентированный подход,
- субъектно-ориентированный подход,
- компонентно-ориентированный подход,
- шаблонно-ориентированный подход.

Функционально-ориентированный подход основан на повторном использовании различных функций. Примером здесь могут служить подпрограммы вычисления синуса или косинуса, включенные в стандартные библиотеки и используемые многими программистами. Разработчики могут также создавать свои функции, процедуры и макросы, составлять из них библиотеки внутреннего пользования и применять их неоднократно в своих проектах [4].

В объектно-ориентированном подходе единица повторного использования – объект. Объект представляет собой структуру, которая объединяет в себе некоторые данные и функции для работы с ними. Для объектов поддерживаются такие механизмы, как наследование, инкапсуляция и полиморфизм. Объект считается по своей природе пассивным элементом, т.е. функции или методы объекта для выполнения своих специфических действий должны вызываться извне [2, 5, 8, 9].

Создатели субъектно-ориентированного подхода утверждают, что данная технология расширяет концепцию объектов для модели параллельных вычислений. Так называемые субъекты инкапсулируют в себе потоки управления и имеют интерфейсы (функции, методы) для взаимодействия с другими субъектами. Субъект считается активным элементом [1].

Идея компонентно-ориентированного подхода в корне отличается от двух предыдущих [2]. Программная компонента – это повторно используемые куски кода и данных в двоичной форме, которые могут быть относительно легко встроены в другие программные компоненты от других производителей. Часто от разработчика скрыта внутренняя реализация этих элементов. Как правило, выделение какой-либо части программы в отдельную компоненту происходит не по функциональному признаку, а с точки зрения удобства программирования, распространения, инсталляции и др. Сегодня данный подход находит все большее применение, например в технологии ActiveX.

Шаблонно-ориентированный подход основан на использовании в качестве многократно применяемых компонентов шаблонов [2]. Смысловыми синонимами данного термина являются паттерны, образцы, эталоны. Шаблон в общем случае представляет собой заготовку для создания какого-либо элемента системы. В процессе создания изделия на основе шаблона происходит насыщение шаблона конкретным содержанием. Как правило, подобный образец содержит в себе набор существенных компонент, ограничений, правил, необходимый для создания законченного объекта на его основе. В принципе шаблон в некотором роде напоминает каркас, но между ними существуют существенные различия: шаблоны более абстрактны, чем каркасы, как архитектурные элементы шаблоны мельче, чем каркасы, шаблоны менее специализированы, чем каркасы.

На данный момент технологии, которые используют тот или иной подход в чистом виде, практически отсутствуют. Так, вместе с объектно-ориентированным

подходом может применяться шаблонно-ориентированный. Более того, приверженность только к одной технологии может снизить эффективность процесса разработки ПО.

Анализируя специфические особенности СРВ, с одной стороны, и существующие технологии повторного использования и реинжиниринга, с другой, можно сделать следующие выводы.

- Большинство современных технологий повторного использования ориентированы на системы общего назначения. Применение их при разработке СРВ часто экономически невыгодно.
- Требования к функционированию систем в масштабе реального времени, а также повышенной отказоустойчивости, надежности и предсказуемости ограничивают набор средств, механизмов и методов повторного использования.
- В число многократно используемых компонентов таких систем должны включаться те, которые умеют работать с временными ограничениями.
- Недопустимо применение компонент – "черных ящиков", код которых скрыт от разработчика, а поведение в критических ситуациях неизвестно.
- Из-за нередкой ограниченности возможностей оборудования вероятны ограничения на размер повторно используемых компонент, а также средств их реализации. Например, программное обеспечение для встроенных систем очень редко пишется на объектно-ориентированных языках высокого уровня.
- Повторно используемые компоненты, описывающие интерфейс с пользователем, могут быть сильно упрощены, а также совмещены с рядом специфических устройств (световые диоды, динамики и пр.)
- Средства симуляции, отладки и тестирования часто необходимо разрабатывать вместе с системой, к ним предъявляется ряд специфических требований, так как в первую очередь необходимо проверить временные и поведенческие характеристики системы.
- Комплекс средств для тестирования можно также создавать ориентированным на компоненты повторного использования.
- Необходимо тщательно выбирать уровень детализации при разбивке системы на компоненты повторного использования; большое число компонент может повлечь за собой увеличение временных затрат на процессы взаимодействия между отдельными частями системы.

Литература

- 1) Lee E.F. Embedded software. // Advances in Computers. Vol. 56. Academic Press, London, 2002.
- 2) Бокс Д. Сущность технологии СОМ. Библиотека программиста. СПб: Питер, 2001.
- 3) Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб: Питер, 2001.
- 4) Горбунов-Посадов М. Облик многократно используемого компонента. // Открытые системы. 1998. №3.
- 5) Дал У., Дейкстра Э., Хоор К. Структурное программирование. М.: Мир, 1975.
- 6) Йордон Э., Аргила К. Структурные модели в объектно-ориентированном анализе и проектировании. М.: Лори, 1999.
- 7) Ройс У. Управление проектами по созданию ПО. М.: Лори, 2002.
- 8) Фридман А.Л. Основы объектно-ориентированной разработки программных систем. М.: Финансы и статистика, 2000.
- 9) Элиенс А. Принципы объектно-ориентированной разработки программ. / 2-е издание. М.: Вильямс, 2002.

**СОЗДАНИЕ ВИРТУАЛЬНОЙ ЛАБОРАТОРИИ ПРОЕКТИРОВАНИЯ
ТЕСТОВОГО ОБЕСПЕЧЕНИЯ****М.М. Зинатуллин****Введение**

Получив повсеместное распространение, Интернет в настоящее время представляет собой унифицированное средство обмена информацией. Его основное предназначение – предоставление возможностей поиска информации в глобальном распределенном хранилище данных. В настоящее время наблюдается расширение сферы использования Интернет. В контексте данной статьи представляет интерес возможность удаленной работы пользователя с системой, решающей задачи конкретной предметной области. Возможность использования данного свойства сети явилась предпосылкой создания виртуальной лаборатории (ВЛ) проектирования тестового обеспечения цифровых схем.

В последнее время термин "виртуальная лаборатория" получает все большее распространение, принимая при этом различную смысловую нагрузку. Проведя анализ существующих систем, можно привести следующую классификацию%

- ВЛ как научная лаборатория – виртуальный аналог традиционного для академической среды научного коллектива. Интернет-технологии позволяют имитировать рабочую среду. ВЛ присущи обычные для исследовательской лаборатории атмосфера коллективного творчества и набор соответствующих технических возможностей;
- ВЛ как общедоступный Интернет-ресурс, продвигающий продукты фирмы на рынок с помощью бесплатного обучения;
- виртуальная обучающая лаборатория – виртуальный аналог комплекса учебных материалов, тренажеров, приборов, позволяющих моделировать реальные естественнонаучные эксперименты с заранее известными результатами.

В нашем случае ВЛ представляет информационную среду, предоставляющую с помощью удаленного доступа возможность проектирования тестового обеспечения. Существующие системы создания тестов по принципу взаимодействия с пользователем бывают двух типов.

Системы первого типа позволяют работать в интерактивном режиме. При этом пользователю предоставляется широкое поле для деятельности. Работа разбивается на большое количество этапов. На каждом этапе формируются подробные отчеты, имеются возможность возврата к предыдущим этапам и т.п.

Системы второго типа работают в пакетном режиме. Здесь, получив задание, система самостоятельно проходит всю последовательность этапов и выдает один подробный комплексный отчет. Специфика же работы в ВЛ такова, что интенсивная работа, как в случае с интерактивным режимом, является затруднительной, что связано с ограничениями пропускной способностью сети, производительности сервера и т.п. С другой стороны, пакетный режим ограничивает влияние пользователя на работы по созданию тестов, что можно отнести к серьезным недостаткам таких систем.

При разработке ВЛ все вышеупомянутые моменты были учтены. Была реализована двухэтапная технология разработки тестового обеспечения [1]. Такой подход позволяет сократить до минимума количество обращений к системе, но сохраняет при этом ведущую роль пользователя в проектировании тестового обеспечения.

Структура виртуальной лаборатории

В качестве инструментального средства разработки программного обеспечения был выбран язык программирования Java. Это обусловлено наличием в языке Java таких важных свойств, как: переносимость программного кода на множество программно-аппаратных платформ, развитость сетевых возможностей, наличие технологии апплетов/сервлетов, встроенные средства для работы с базами данных. Вышеуказанные моменты позволяют писать гибкие, масштабируемые Интернет-приложения, отвечающие требованиям функциональности, удобству и безопасности. В работе также используются следующие программные продукты: ОС Windows, СУБД MySQL v.3.23.54, HTTP-сервер Apache v.1.3.23 в паре с Java-сервером Tomcat v4.1, специализирующимся на выполнении сервлетов.

Структура виртуальной лаборатории представлена на рис.1.

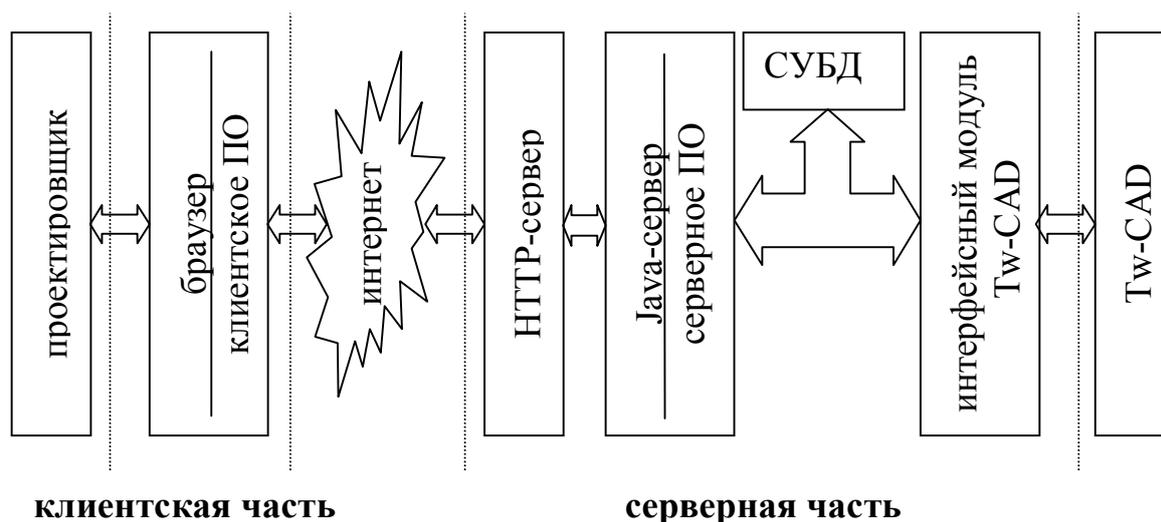


Рис. 1. Структура виртуальной лаборатории

В структуре ВЛ можно выделить 2 основные компоненты, находящиеся в тесной взаимосвязи – клиентскую и серверную.

Проектировщик работает непосредственно с клиентской компонентой, которая представляет собой стандартный WEB-браузер с загружаемым при работе в виртуальной лаборатории клиентским программным обеспечением (ПО). В функции клиентского ПО входит :

- регистрация пользователя;
- аутентификация пользователя;
- построение рабочего пространства пользователя;
- создание проекта и сбор исходных данных по нему;
- формирование таблиц и диаграмм показателей тестопригодности, предоставление результатов в виде текстового отчета;
- выдача сгенерированного теста.

Клиентское ПО реализуется с помощью апплетов языка Java и HTML-страниц.

Серверная компонента является ядром лаборатории. К ее основным составным частям относятся:

1. тандем HTTP-Java сервер;
2. СУБД;
3. интерфейсный модуль Tw-CAD.

Рассмотрим каждую из частей более подробно.

Тандем HTTP – Java сервер. Укрупненная схема тандема HTTP-Java сервер представлена на рис. 2. Все запросы, полученные от клиентского ПО, принимаются HTTP-сервером, который в зависимости от типа запрашиваемой информации определяет, кому обрабатывать запрос. Запрашиваемые данные могут быть 2-х типов: статические и динамические.

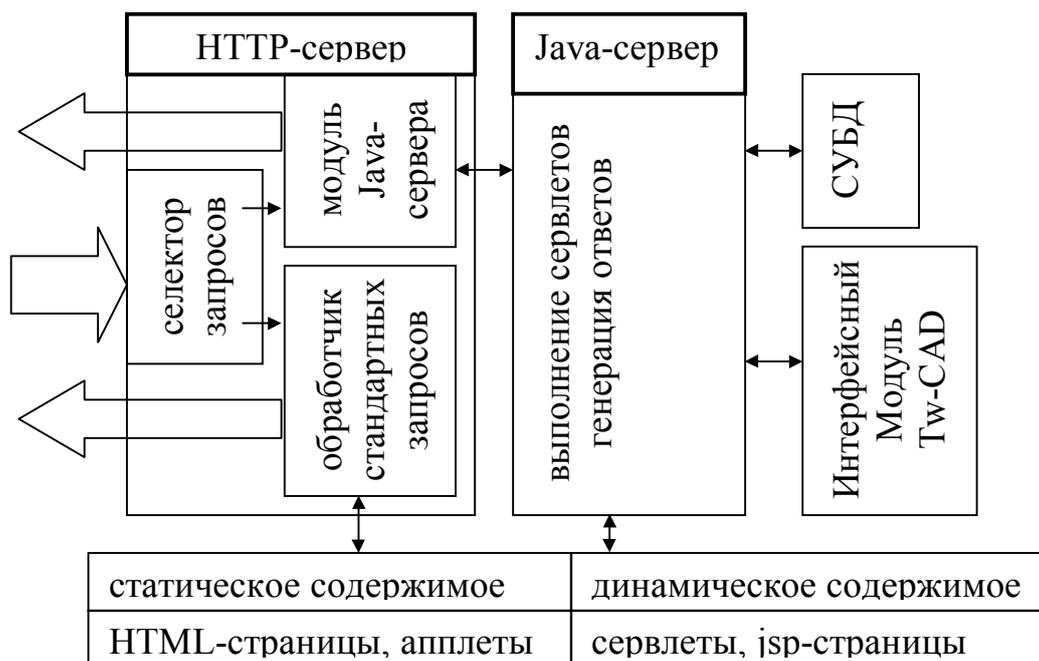


Рис. 2. Укрупненная схема тандема HTTP-Java сервер

К статическим данным относят готовые HTML-страницы, апплеты, рисунки и т.п. Динамические данные формируются в зависимости от каких-либо условий, например, указаний пользователя по созданию проекта, полученных от Tw-CAD оценок тестопригодности и т.п. При поступлении запроса на получение статических данных HTTP сервер самостоятельно его обрабатывает. Если же приходит запрос на получение динамических данных, они передаются модулю Java-сервера. В этом случае запрос пересылается Java-серверу, который загружает на выполнение нужную программу (являющуюся частью серверного ПО). Выполняемая программа (сервлет или jsp-страница) при необходимости может обращаться к СУБД или интерфейсному модулю Tw-CAD. Сгенерированный программой ответ, если таковой присутствует, передается через модуль Java-сервера в HTTP сервер и далее направляется клиентскому ПО.

СУБД. База данных используется для хранения информации, используемой на всем протяжении работы с виртуальной лабораторией. В нее заносятся регистрационные данные каждого пользователя, хранится список проектов и их текущее состояние и т.п. С СУБД взаимодействуют как программы серверного ПО, так и интерфейсный модуль Tw-CAD.

Интерфейсный модуль Tw-CAD взаимодействует непосредственно с Tw-CAD. В качестве исходных данных модуль принимает схему и список работ, которые

необходимо выполнить. После получения результатов модуль формирует отчеты и заносит их в базу данных.

Технология проектирования тестового обеспечения цифровых схем

Целью рассматриваемой технологии является создание теста высокого качества, сгенерированного с умеренными затратами времени. Предлагаемая технология прозрачна для пользователя в силу того, что она объясняет, из чего складывается качество теста. Для того чтобы достичь поставленной цели, Tw-CAD позволяет сгенерировать тест верификации, произвести анализ состоятельности шин и двунаправленных магистралей (АСШДМ), оценить уровень тестопригодности схемы (подсистема DFT – Design For Test), сгенерировать рекомендации по улучшению схемы (подсистема ATRG – Automatic Testable Recommendation Generator).



Рис. 3. Работа пользователя на первом этапе

При верификации для каждого бита (входа, элемента памяти, комбинационного или буферного элемента, выхода схемы) делается попытка вычисления установочной последовательности, что позволяет обнаружить ошибки в разработке схемы. Анализ

состоятельности шин и двунаправленных магистралей позволяет выявить такие аварийные ситуации, как работа двух источников сигнала в шине, а также попытка использования магистрали как на вход, так и на выход одновременно. Тестопригодность схемы оценивается в виде прогноза достижимого качества теста, который рассчитывается как отношение числа бит в зоне тестопригодности к общему числу бит, за вычетом избыточных. При создании модели схемы биты ранжируются по оценкам управляемости, наблюдаемости и тестопригодности. Управляемость характеризует, насколько сложно управлять битом со стороны входов, а наблюдаемость – насколько сложно транспортировать логическое значение бита к выходам. Комплексное представление о зоне уверенной генерации теста дает зона тестопригодности – пересечение хороших зон управляемости и наблюдаемости.

Таким образом, анализируя схему, можно спрогнозировать качество теста по зоне тестопригодности и отчету верификации. Возможна и обратная задача, меняя размер зоны тестопригодности можно влиять на качество теста. В этом и состоит суть данной технологии.

Работа пользователя разбивается на 2 этапа. На первом этапе (см. рис. 3) схема верифицируется, производится АСШДМ и рассчитывается прогноз качества теста. Если данный прогноз подходит, то генерируется тест. В противном случае подсистема ATRG вырабатывает рекомендации по введению точек дополнительного управления (ТДУ) и точек дополнительного наблюдения (ТДН), позволяющих улучшить качество теста. При этом характер рекомендаций таков, что пользователь абсолютно свободен в выборе способа их реализации. В ситуации, когда предлагаемые изменения по каким-либо причинам невыполнимы, возможна генерация теста лишь с тем качеством, которое было спрогнозировано ранее. Если же рекомендации вполне осуществимы и приводят к желаемому прогнозу качества теста, необходимо реализовать полученные рекомендации в исходной схеме. В случае, когда нужный прогноз все еще не получен, прежде чем реализовывать изменения, необходимо доработать схему по протоколу верификации, АСШД, а также устранить избыточность.

В качестве входных данных второго этапа (см. рис. 4) выступают:

- измененная схема пользователя,
- ожидаемое им качество теста,
- процент верификации,
- количество реализованных рекомендаций.

Схема верифицируется, и для нее рассчитывается прогноз качества теста. Если уточненный на данном этапе прогноз совпадает с ожидаемым прогнозом, то генерируется тест. Разработчик получает вместе с тестом в заказанном формате реально полученное значение его качества. В противном случае сличается ожидаемое и уточненное значение процента верификации. Их несовпадение свидетельствует о неполном или ошибочном характере исправлений по протоколу верификации. В данной ситуации генерируется новый протокол верификации, который возвращается разработчику для доработки схемы. Если с верификацией все в порядке, то, скорее всего, проблема заключается в неправильной реализации рекомендаций или реализации изменений в обход полученных от Tw-CAD рекомендаций. Поэтому разработчику дается указание проверить правильность и полноту реализованных изменений. Вне зависимости от того, каков результат второго этапа, разработчик получает в свое распоряжение протокол изменений в схеме. Это позволит ему проконтролировать свои предыдущие действия на наличие в них ошибок. Следует отметить тот факт, что второй этап является заключительным в технологии. Однако по объективным причинам, указанным выше, возможна ситуация, когда разработчик получает вместо теста указания к дальнейшей работе над схемой. Но результат внесенных изменений будет являться входными данными все для того же второго этапа.



Рис. 4. Работа пользователя на втором этапе

Заключение

ВЛ является логическим продолжением работ по созданию технологии разработки тестового обеспечения в рамках Tw-CAD. Использование ВЛ обладает рядом несомненных преимуществ: для работы в лаборатории необходимо лишь подключение к сети Интернет; работа с самой последней версией системы, поддерживаемой и настраиваемой централизованно; удобный веб-интерфейс с возможностью индивидуальной настройки и т.п. Используемая в ВЛ технология учитывает особенности работы в сети. Реализован эффективный двухэтапный подход к проектированию тестового обеспечения, где пользователь принимает активное участие в процессе проектирования.

Литература

1. Звягин В.Ф., Зинатуллин М.М. Технология проектирования тестового обеспечения цифровых схем. // Изв. вузов. Приборостроение. 2003. Т. 46. № 1. С.42–47.

ПРИМЕНЕНИЕ КВАТЕРНИОНОВ ДЛЯ МОДЕЛИРОВАНИЯ ЗЕРКАЛЬНЫХ СИСТЕМ В ДИНАМИКЕ

А.Ю. Королев, А.В. Демин

При проектировании обзорно-поисковых оптических комплексов (ОПК) одной из проблем является проектирование систем, реализующих движение линии визирования в пространстве. Изменение пространственного положения линии визирования может быть реализовано либо путем изменения пространственного положения ОПК, либо за счет подвижных оптических элементов при неподвижных ОПК. При реализации второго пути одной из проблем является оптимизация кинематических и динамических параметров подвижных оптических элементов.

Уравнение, описывающее отражение от плоского зеркала, приведено в [1]. При этом ориентация плоского зеркала в пространстве задается вектором нормали \vec{N} к плоскости зеркала. Падающий луч задается вектором \vec{A}_0 , определяющим направление луча. Таким образом, имеем формулу для отраженного луча \vec{A}_1 :

$$\vec{A}_1 = Q \circ \vec{A}_0, \quad (1)$$

где Q – кватернион отражения, символ \circ означает кватернионное умножение,

$$Q = 1 - \tilde{A}\tilde{N} \circ \vec{A}_0, \quad (2)$$

где $\tilde{A} = -2\vec{N}\vec{A}_0$ – относительная постоянная отклонения в случае отражения.

При расчете по формуле (1) направления векторов нормали и лучей берутся, как показано на рис. 1. Как видно из рисунка, вектор падающего луча направлен противоположно направлению хода луча.

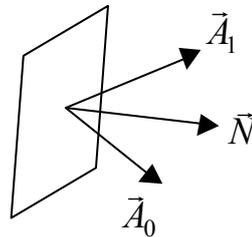


Рис. 1. Направления векторов нормали, падающего и отраженного лучей

При расчете по формуле (1) используются единичные векторы (т.е. их модуль равен 1), если же мы задаем не единичные вектора, то перед их подстановкой в формулу (1) необходимо произвести их нормирование:

$$\vec{A}_{0i} = \left(\frac{a_x}{|\vec{A}_0|}, \frac{a_y}{|\vec{A}_0|}, \frac{a_z}{|\vec{A}_0|} \right), \quad \vec{N}_i = \left(\frac{n_x}{|\vec{N}|}, \frac{n_y}{|\vec{N}|}, \frac{n_z}{|\vec{N}|} \right).$$

Однако приведенная выше формула (1) для расчета реально существующих зеркал практически непригодна, поскольку обладает следующими недостатками:

1. формула не учитывает сторону зеркала, на которую падает луч, т.е. описывает двустороннее зеркало, и отражение возможно как с "лицевой", так и с "обратной" стороны зеркала (рис. 2).
2. формула не учитывает ограниченности размеров зеркала, т.е. описывает отражение от неограниченной зеркальной плоскости и дает отражение даже в том случае, если луч проходит мимо зеркала (рис. 3).

Для устранения указанных недостатков проведем дополнительные расчеты, чтобы выяснить, с какой стороны зеркала луч падает и попадает ли луч на зеркало.

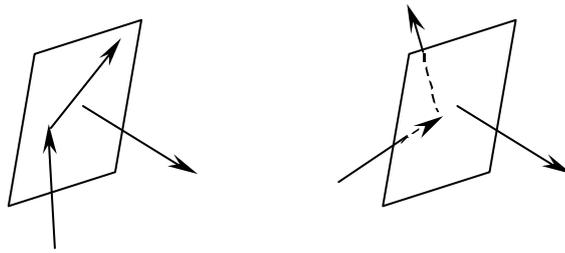


Рис. 2. Возможные случаи отражения

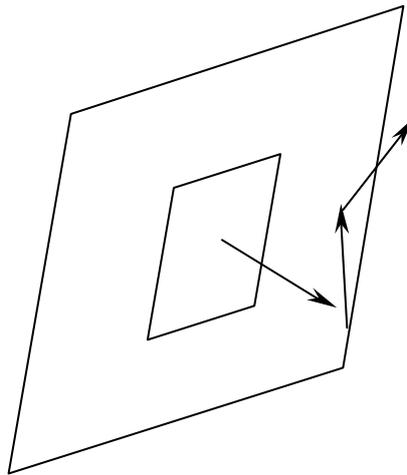


Рис. 3. Отражение от неограниченной плоскости

Чтобы учесть первое ограничение, необходимо найти угол между вектором падающего луча и вектором нормали; если этот угол $< 90^\circ$, то луч падает с "лицевой" стороны, т.е. есть отражение; если угол $> 90^\circ$, то луч падает с "обратной" стороны и, значит, отражение невозможно; если же угол строго равен 90° , то луч лежит в плоскости зеркала (можно считать, что он проходит вдоль поверхности зеркала не испытывая отражения).

Угол между двумя векторами может быть найден по формуле:

$$\cos \alpha = \frac{\vec{A} \cdot \vec{N}}{|\vec{A}| |\vec{N}|}, \quad (3)$$

так как $|\vec{A}| = |\vec{N}| = 1$, то

$$\alpha = \arccos(\vec{A} \cdot \vec{N}). \quad (4)$$

Зеркало представляет собой четырехугольник, для его задания необходимо указать координаты углов. Но такое его задание является избыточным (поскольку для задания плоскости достаточно указать три точки, не лежащих на одной прямой) и, следовательно, потенциальным источником ошибок (ведь всегда можно указать такие четыре точки, которые не будут лежать в одной плоскости). Лучшим вариантом указания плоскости зеркала будет задание координат двух его углов (являющихся концами одного из ребер четырехугольника) и координат центра многоугольника (см. рис. 4).

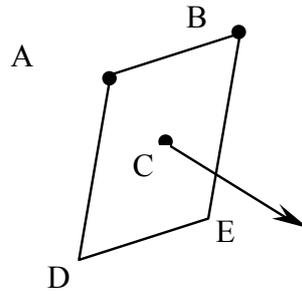


Рис. 4. Точки, описывающие положение зеркала в пространстве

При таком задании однозначно определяется положение зеркала в пространстве, при этом всегда может быть вычислен вектор нормали и координаты оставшихся вершин.

Вектор нормали вычисляем по формуле:

$$\vec{N} = \vec{AC} \times \vec{BC}, \quad (5)$$

при этом после вычисления вектора нормали в общем случае мы получим неединичный вектор, и нужно будет его нормировать.

Падающий луч будем задавать связанным вектором, т.е. вектором, закрепленным в некой точке.

Таким образом, имеем прямую, заданную вектором и точкой, лежащей на прямой (падающий луч), а также плоскость, заданную точкой и вектором нормали (зеркало).

Чтобы определить, пересекает ли луч зеркало, необходимо сначала найти точку пересечения прямой с плоскостью.

Пусть прямая описывается уравнением в координатном виде:

$$\frac{x - x_1}{l} = \frac{y - y_1}{m} = \frac{z - z_1}{n}, \quad (6)$$

где x_1, y_1, z_1 – координаты точки, лежащей на прямой, l, m, n – координаты направляющего вектора, а плоскость описывается уравнением:

$$A(x - x_2) + B(y - y_2) + C(z - z_2) = 0, \quad (7)$$

где x_2, y_2, z_2 – координаты точки, лежащей на плоскости, A, B, C – координаты вектора нормали.

Выразив в формуле (6) y и z через x и подставив эти выражения в (7), найдем координату x точки пересечения прямой и плоскости:

$$x = \frac{Ax_2l + Bx_1m - By_1l + By_2l + Cx_1n - Cz_1l + Cz_2l}{Al + Bm + Cn}. \quad (8)$$

Аналогично для координат y и z (выражая их через две другие координаты) получим:

$$y = \frac{Ay_1l - Ax_1m + Ax_2m + By_2m + Cy_1n - Cz_1m + Cz_2m}{Al + Bm + Cn}, \quad (9)$$

$$z = \frac{Az_1l - Ax_1n + Ax_2n + Bz_2m - By_1n + By_2n + Cz_2n}{Al + Bm + Cn}. \quad (10)$$

Зная координаты точки пересечения прямой с плоскостью, задаваемой вектором нормали зеркала, определим, лежит ли точка внутри зеркала или снаружи. Для этого найдем расстояния от этой точки до каждой из сторон зеркала по формуле (11):

$$d_{AB} = \left(\left\{ [(x_p - x_A)(y_B - y_A) - (y_p - y_A)(x_B - x_A)]^2 + [(y_p - y_A)(z_B - z_A) - (z_p - z_A)(y_B - y_A)]^2 + [(z_p - z_A)(x_B - x_A) - (x_p - x_A)(z_B - z_A)]^2 \right\} / (x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2 \right)^{1/2} \quad (11)$$

(для расчета d_{BC} , d_{AD} , d_{DC} нужно в (11) заменить соответствующие координаты).

Если одно из этих расстояний больше, чем длина соответствующей стороны, то, значит, точка лежит вне четырехугольника (зеркала), т.е. луч не пересекает зеркало (рис. 5).

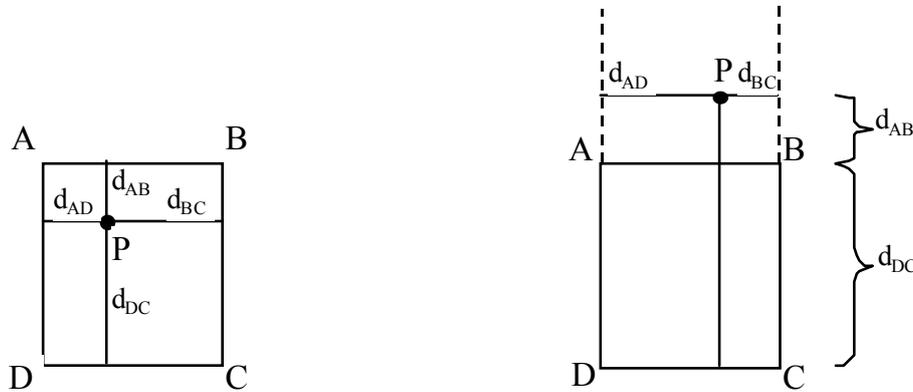


Рис. 5. Расположение точки пересечения относительно зеркала.

По аналогичной методике рассчитывается призма. Однако призма имеет две "работающие" грани. Поэтому необходимо рассчитывать преломление от двух плоскостей. В отличие от зеркала, каждая "рабочая" грань призмы не имеет стороны, поглощающей пучок света. Поэтому нет необходимости учитывать сторону, с которой на грань падает луч. Для преломляющей поверхности относительная постоянная отклонения Γ находится по формуле

$$\Gamma = \sqrt{\left(\frac{n'}{n}\right)^2 - 1 + (\vec{N} \cdot \vec{A}_0)^2} - \vec{N} \cdot \vec{A}_0,$$

где n' – показатель преломления 2-й среды, n – показатель преломления 1-й среды.

Выше было рассмотрено отражение от неподвижного зеркала и преломление от призмы, однако поставленная задача подразумевает вращение зеркала (призмы) вокруг некоторой оси (осей).

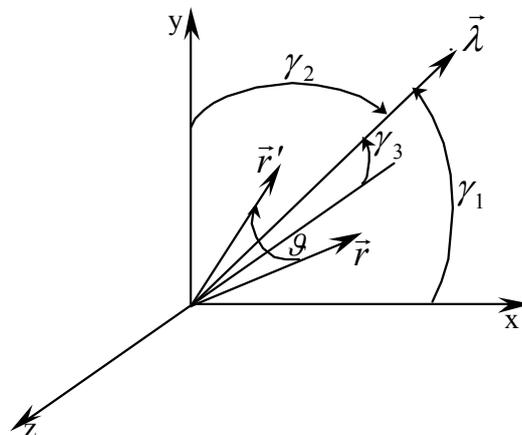


Рис. 6. Поворот вокруг произвольного вектора.

Как показано в [2], вращение вектора вокруг произвольной оси (рис. 6) может быть в кватернионном виде задано как

$$\vec{r}' = \tilde{\Lambda} \circ \vec{r} \circ \Lambda, \quad (12)$$

где \vec{r}' – вектор, полученный вращением вектора \vec{r} вокруг оси $\vec{\lambda}$, Λ – собственный кватернион преобразования вращения, $\tilde{\Lambda}$ – сопряженный кватернион. Кватернион Λ имеет следующий вид:

$$\Lambda = \left(\cos \frac{\vartheta}{2}; \gamma_1 \sin \frac{\vartheta}{2}; \gamma_2 \sin \frac{\vartheta}{2}; \gamma_3 \sin \frac{\vartheta}{2} \right), \quad (13)$$

где $\gamma_1, \gamma_2, \gamma_3$ – направляющие косинусы вектора поворота $\vec{\lambda}$.

В случае поворота вокруг оси, коллинеарной одной из координатных осей, кватернион Λ принимает более простой вид:

1. при повороте вокруг оси абсцисс $\Lambda = \left(\cos \frac{\vartheta}{2}; \sin \frac{\vartheta}{2}; 0; 0 \right);$
2. при повороте вокруг оси ординат $\Lambda = \left(\cos \frac{\vartheta}{2}; 0; \sin \frac{\vartheta}{2}; 0 \right);$
3. при повороте вокруг оси аппликат $\Lambda = \left(\cos \frac{\vartheta}{2}; 0; 0; \sin \frac{\vartheta}{2} \right).$

Перейдем к выводу алгоритма расчета подвижной системы. Вывод произведем на примере системы, состоящей из двух зеркал, подвешенных на подвижном подвесе (рис. 7).

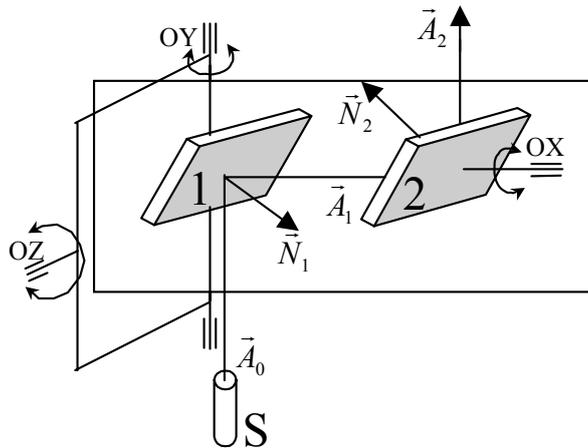


Рис. 7. Моделируемая система

Система может поворачиваться вокруг осей OY и OZ, кроме того, зеркало 2 может независимо поворачиваться вокруг оси OX. На зеркало 1 падает луч света \vec{A}_0 от неподвижного источника S. Необходимо рассчитать направление выходного луча \vec{A}_2 .

Далее для упрощения алгоритма пренебрежем тем фактом, что зеркала имеют ограниченные размеры, и будем считать, что они задаются их единичными векторами нормалей.

Пусть имеется инерциальная система отсчета I, относительно которой задаются все вектора и ведется расчет. Пусть имеется также связанная система отсчета E, оси которой определяют положения осей поворота зеркал, т.е. направление орта e_1 совпадает с направлением оси OX, направление орта e_2 совпадает с направлением оси OY и направление орта e_3 совпадает с направлением оси OZ (см. рис. 8). Положение ортов задается относительно системы отсчета I.

Как видно из рис. 8, повороту системы зеркал вокруг оси OY или OZ соответствует поворот связанной системы отсчета E вокруг орта \vec{e}_2 или \vec{e}_3 , при этом положения векторов нормалей \vec{N}_1 и \vec{N}_2 относительно системы отсчета E остается неизменным, а относительно системы отсчета I меняется (поскольку происходит изменение положения системы отсчета E).

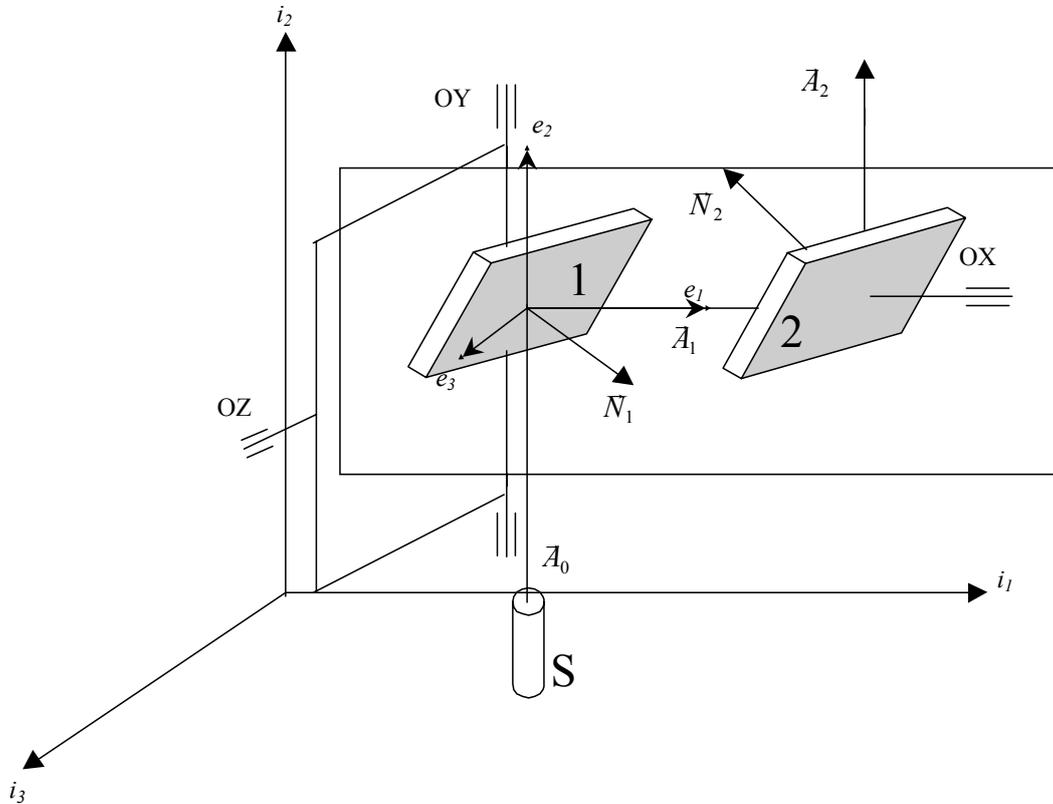


Рис. 8. Ввод систем отсчета

Повороту зеркала 2 вокруг оси OX соответствует поворот вектора нормали \vec{N}_2 вокруг орта \vec{e}_1 , при этом происходит изменение положения вектора нормали как относительно системы отсчета E , так и относительно системы отсчета I .

Перейдем теперь к более строгому – математическому – описанию алгоритма расчета выходного луча в данной системе при ее движениях.

Пусть заданы орты $\vec{e}_1, \vec{e}_2, \vec{e}_3$ связанной системы отсчета E , единичные векторы нормалей \vec{N}_1 и \vec{N}_2 и единичный вектор входящего луча \vec{A}_0 относительно инерциальной системы отсчета I .

Рассмотрим поворот системы зеркал относительно \vec{e}_2 (см. [2]). Найдем новое положение ортов системы отсчета E относительно I :

$$\vec{e}'_k = \Lambda_m^{-1} \circ \vec{e}_k \circ \Lambda_m, \quad (k = 1, 2, 3), \quad m = 2, \quad (14)$$

где Λ_m – кватернион преобразования, Λ_m^{-1} – обратный ему кватернион.

Кватернион преобразования находится по формуле

$$\Lambda_m = \cos \frac{\varphi_m}{2} + \cos \angle(\vec{e}_m, \vec{i}_1) \sin \frac{\varphi_m}{2} i + \cos \angle(\vec{e}_m, \vec{i}_2) \sin \frac{\varphi_m}{2} j + \cos \angle(\vec{e}_m, \vec{i}_3) \sin \frac{\varphi_m}{2} k, \quad (15)$$

где φ_m – угол поворота вокруг \vec{e}_m .

Так как $\|\Lambda_m\| = 1$ (см. [2]), то выражение (14) запишется иначе:

$$\vec{e}'_k = \tilde{\Lambda}_m \circ \vec{e}_k \circ \Lambda_m, (k = 1, 2, 3), m = 2, \quad (16)$$

где $\tilde{\Lambda}_m$ – сопряженный кватернион, и находится как

$$\Lambda_m = \cos \frac{\varphi_m}{2} - \cos \angle(\vec{e}_m \vec{i}_1) \sin \frac{\varphi_m}{2} i - \cos \angle(\vec{e}_m \vec{i}_2) \sin \frac{\varphi_m}{2} j - \cos \angle(\vec{e}_m \vec{i}_3) \sin \frac{\varphi_m}{2} k, \quad (17)$$

Повернув систему отсчета E вокруг \vec{e}_2 , мы должны повернуть векторы нормалей \vec{N}_1 и \vec{N}_2 вокруг этой же оси и на тот же угол (восстановив тем самым исходное положение векторов нормалей относительно связанной системы отсчета E):

$$\begin{aligned} \vec{N}'_1 &= \tilde{\Lambda}_m \circ \vec{N}_1 \circ \Lambda_m, \\ \vec{N}'_2 &= \tilde{\Lambda}_m \circ \vec{N}_2 \circ \Lambda_m, \quad m = 2. \end{aligned} \quad (18)$$

Аналогично происходит поворот системы зеркал относительно \vec{e}_3 :

$$\begin{aligned} \vec{e}'_k &= \tilde{\Lambda}_3 \circ \vec{e}_k \circ \Lambda_3, \quad k = (1, 2, 3), \\ \vec{N}'_1 &= \tilde{\Lambda}_3 \circ \vec{N}_1 \circ \Lambda_3, \\ \vec{N}'_2 &= \tilde{\Lambda}_3 \circ \vec{N}_2 \circ \Lambda_3 \end{aligned} \quad (19)$$

где Λ_3 и $\tilde{\Lambda}_3$ вычисляются соответственно по формулам (15) и (17) при $m = 3$.

Рассмотрим теперь поворот зеркала 2 вокруг \vec{e}_1 . По формулам (15) и (17) находим кватернион преобразования Λ_1 и сопряженный ему кватернион $\tilde{\Lambda}_1$ и новое положение вектора нормали \vec{N}_2 :

$$\vec{N}'_2 = \tilde{\Lambda}_1 \circ \vec{N}_2 \circ \Lambda_1. \quad (20)$$

Как видно, в данном случае положение системы отсчета E относительно I остается неизменным, также не меняется положение вектора нормали \vec{N}_1 .

После совершения необходимых поворотов и получения новых положений векторов нормалей нужно рассчитать выходной луч.

Луч \vec{A}_1 , отраженный от зеркала 1, находим по формуле (1). Для того чтобы найти его отражение от зеркала 2, необходимо сначала найти вектор, направленный противоположно вектору \vec{A}_1 :

$$\vec{A}'_1 = -\vec{A}_1,$$

и, подставив его в формулу (1), найти выходящий луч \vec{A}_2 .

На примере системы, состоящей из двух зеркал, было показано, что каждый элемент при его движении и при движении системы в целом рассчитывается по отдельности, независимо друг от друга. Поэтому рассмотренная методика позволяет производить моделирование зеркально-призмных систем с произвольным числом элементов, необходимо лишь правильно задать оси поворота каждого зеркально-призмного элемента и выбрать расположение подвижной системы координат.

Рассмотрим обобщенный алгоритм расчета произвольной зеркально-призмной системы.

1. Если вокруг оси \vec{e}_x зеркально-призмный элемент (или группа элементов) поворачивается независимо от системы в целом, то его поворот описывается уравнением

$$\vec{N}'_i = \tilde{\Lambda}_x \circ \vec{N}_i \circ \Lambda_x. \quad (21)$$

2. Если вокруг оси \vec{e}_x поворачивается система в целом, то это движение описывается системой уравнений

$$\begin{cases} \vec{e}'_k = \tilde{\Lambda}_x \circ \vec{e}_k \circ \Lambda_x, & k = 1, 2, 3 \\ \vec{N}'_i = \tilde{\Lambda}_x \circ \vec{N}_i \circ \Lambda_x, & i = \overline{1, n} \end{cases} \quad (22)$$

3. Аналогично рассмотренным поворотам в п.1 и п.2, осуществляется повороты вокруг осей \vec{e}_y, \vec{e}_z .

4. Расчет выходного луча ведется после осуществления всех поворотов, при этом применяется рекуррентное соотношение

$$\vec{A}_{i+1} = Q \circ \vec{A}_i, \quad i = \overline{0, n-1}. \quad (23)$$

5. Для расчета динамики используется следующая система уравнений:

$$\begin{cases} M_x = J_x \dot{\omega}_x + (J_z - J_y) \omega_y \omega_z \\ M_y = J_y \dot{\omega}_y + (J_x - J_z) \omega_z \omega_x \\ M_z = J_z \dot{\omega}_z + (J_y - J_x) \omega_x \omega_y, \end{cases} \quad (24)$$

где M_x, M_y, M_z – внешние моменты инерции, прикладываемые к системе для ее поворота, J_x, J_y, J_z – главные центральные моменты инерции группы одновременно поворачивающихся зеркально-призмных элементов, $\omega_x, \omega_y, \omega_z$ – угловые скорости поворота.

Литература

1. Потепун В.Е. О применении кватернионов в геометрической оптике. / Тр. ЛИТМО, 1976, вып. 84. Расчет, конструирование и исследование точности оптических систем.
2. Бранец В.Н., Шмыглевский И.П. Применение кватернионов в задачах ориентации твердого тела. М.: Наука, 1973. 320с.

**ЕДИНЫЙ ИНФОРМАЦИОННЫЙ КОМПЛЕКС ПРЕДПРИЯТИЯ
В ТЕХНОЛОГИЧЕСКОЙ ПОДГОТОВКЕ И ПРОИЗВОДСТВЕ****О.Ю. Дмитриева**

Задачи общей компьютеризации предприятия состоит в том, чтобы обеспечить его структурное построение и функционирование как единого комплекса на основе технических средств автоматизации и информационных технологий. Анализ информационных технологий проектирования и производства показывает [1], что внедрение комплексных автоматизированных систем конструирования (CAD – AutoCAD, Autodesk, Mechanical Desktop), инженерного анализа (CAE – WinFrame, WinStructure, visualNastran), технологической подготовки производства (CAM – Simatron, T-flex, EgeCAM), а также программ, обеспечивающих их взаимодействие (PDM – SmarTeam) [2] выводит предприятие на качественно новый высокий уровень эффективности производства.

В основе их взаимодействия лежит объемная геометрическая модель детали, узла, изделия. В результате продвижения определенного изделия от конструктора к производству модель дополняется такими ее составляющими, как расчетная и сеточная модели, модели технологических процессов обработки, управляющие программы для станков с ЧПУ. Это позволяет заметно сократить время проектирования и подготовки производства за счет почти параллельной работы конструкторов, расчетчиков и технологов с проектируемым изделием. В результате может быть достигнута производительность, превышающая максимально возможную на данный момент. Представляют интерес программы, с которыми могут работать как конструктор, так и технолог. Примером таких программ могут являться Simatron, T-flex, IMS, Cimagrafi и др.

Для обеспечения качественной работы CAD/CAM/CAE составляющих в функции PDM-систем входит:

- организация обмена и предоставления данных в различных CAD/CAM/CAE системах;
- авторизация пользователей и рабочих групп, назначение прав доступа и защита информации;
- организация быстрого поиска, просмотра и аннотирования документов и моделей различных форматов, без загрузки приложений, в которых эта информация создавалась;
- установление последовательности прохождения документов и контроль выполнения;
- ведение распределенных архивов конструкторской, технологической коммерческой информации и др.
- организация информации на физических носителях различного типа, удаленный доступ к информации.

В настоящее время дальнейшая автоматизация производства наукоемких изделий выходит на качественно новый уровень, который находит свое воплощение в концепции CALS-технологий, реализующей информационную поддержку жизненного цикла изделий. Практическая реализация этой концепции связана с формированием

единого информационного пространства, объединяющего все этапы жизненного цикла изделия (проектирование, производство, эксплуатация и утилизация).

Внедрение на предприятии информационных технологий требует решения следующих задач:

- разработка новой корпоративной распределенной информационной системы архитектуры "клиент-сервер";
- составление баз данных и процедур электронного документооборота;
- построение локальных вычислительных сетей, основанных на реальных потребностях и взаимоотношениях;
- выбор программного обеспечения.

При формировании технической стороны внедрения единого информационного комплекса следует обратить внимание на следующие требования, которыми должно обладать вновь поставляемое оборудование: достаточная производительность, аппаратная совместимость, экономичность, возможность использования ранее установленного оборудования, ремонтпригодность и низкая стоимость эксплуатации.

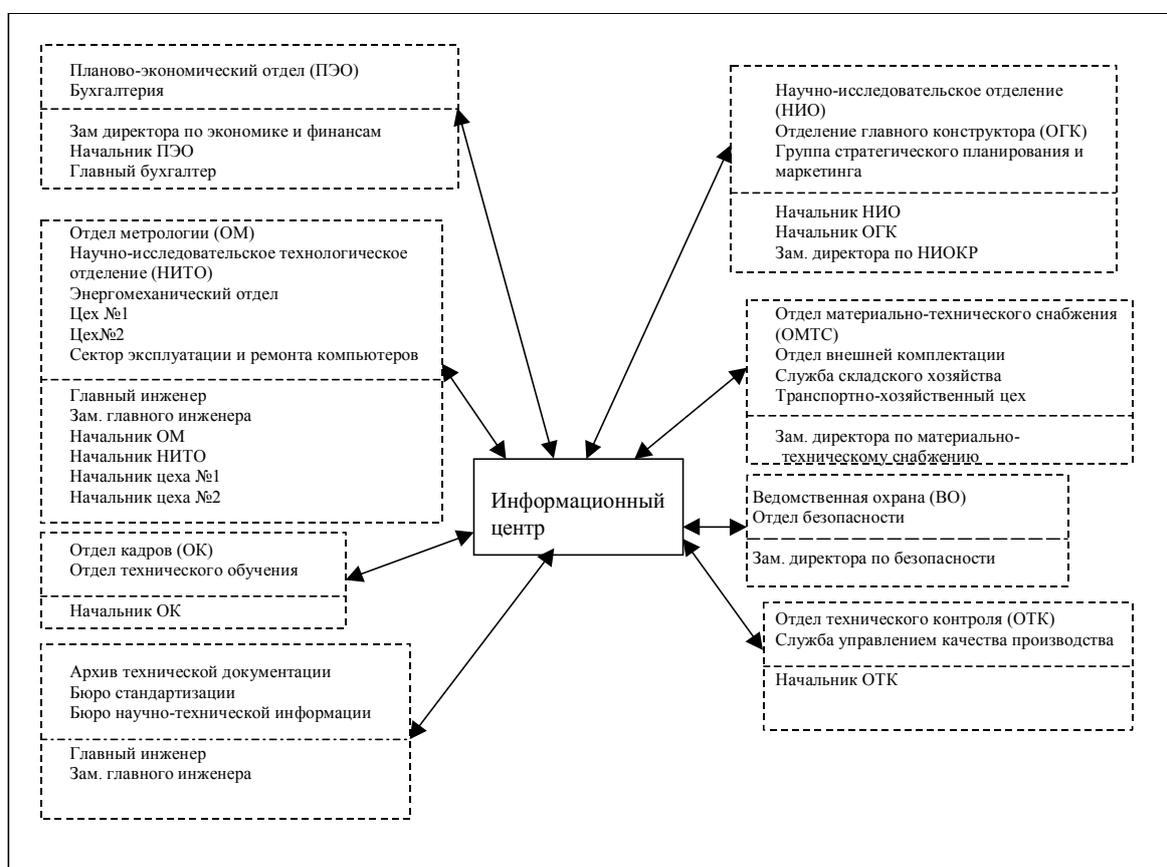


Рис. 1. Схема построения информационного комплекса предприятия

Для выполнения поставленных задач создания компьютеризированного сертифицированного производства модно использовать условную схему построения информационного комплекса предприятия (рис. 1). Ядром комплекса служит информационно-вычислительный центр, построенный на базе файл-сервера (фирмы SUN Microsystem) или рабочих станций, основными функциями которого являются:

- объединение различных по назначению компонент информационного комплекса и осуществление обмена между ними;
- хранение информации;

- управление информационными модулями;
- предоставление вычислительных ресурсов.

Основными компонентами информационного комплекса являются локальные вычислительные сети подразделений различного профиля (КО, ТО, цеха, ЛИМ, ОМ, ПДО, ОНС, ПЭО, ПЗ) и отдельные автоматизированные места (рабочие места руководителей). Каждый компонент является самостоятельной единицей и должен работать независимо от ядра комплекса. Возможно их соединение для обмена внутренней или конфиденциальной информацией.

Программное обеспечение, используемое в информационном комплексе, может быть достаточно разнородным. Поэтому более желательно применение стандартных протоколов для связи между компонентами сети. За базовый принят протокол TCP/IP, который используется в Internet. Таким образом, любой компонент может использовать любое программное обеспечение, но при этом должна быть обеспечена поддержка TCP/IP протокола.

Для защиты информации можно установить зеркальные дисковые подсистемы, дисковые массивы (RAID) или параллельно работающие зеркальные файловые серверы (например NetWareSPRIII)

В решении технических задач, возникающих при создании единого информационного комплекса предприятия, и для обеспечения дальнейшего экономичного расширения, как правило, желательно использовать только стандартные протоколы для обмена информацией и опираться на достижения в области вычислительных сетей ведущих фирм: Microsystems, Novell, Hewlett-Packard, Hitachi, гарантирующие срок морального старения технических и программных средств не менее 5 лет.

Литература

1. Елисеев Ю.С.; Кузнецов И.И.; Алексеева Н.И. Разработка и внедрение элементов CALS-технологий на ММПП "Салют" // Информационные технологии в проектировании и технологии. 2001. №3.
2. Яблочников Е.И. Организация единого информационного пространства технической подготовки производства с использованием PDM SmarTeam. // Информационные технологии в проектировании и технологии. 2001. №3.

РАЗРАБОТКА ТЕХНОЛОГИИ ИЗГОТОВЛЕНИЯ АЛЮМИНИЕВЫХ ОТЛИВОК СЛОЖНЫХ ТОНКОСТЕННЫХ ДЕТАЛЕЙ СПОСОБОМ ВАКУУМНО-ПЛЕНОЧНОЙ ФОРМОВКИ

В.Н. Гераничев

В условиях опытного и мелкосерийного производства часто основным способом изготовления корпусных деталей модулей, рамок, панелей является механическая обработка заготовок из плиты АМг6 и сплава Д16. Некоторые детали выполняются литьем под давлением, дающим хорошее качество поверхности отливки и требующим минимальной механической обработки для получения готовых изделий по сравнению с другими технологическими процессами литья. К недостаткам способа относятся большие затраты и время на подготовку производства из-за высокой трудоемкости проектирования и изготовления литейных форм. Эффективность литья под давлением проявляется при серийности от 100 деталей.

В последнее время повышается интерес к специальным способам литья, сочетающим низкую стоимость подготовки производства с относительно высоким качеством отливок. Как правило, у таких литых заготовок коэффициент использования материала в 2–3 раза больше, чем у деформируемых заготовок, и их применение обеспечивает существенное уменьшение трудоемкости при дальнейшей механической обработке.

Целью работы явилась разработка технологии изготовления тонкостенных деталей способом вакуумно-пленочной формовки (ВПФ) с максимальным приближением по чистоте поверхности и геометрической точности к готовым деталям. Расчеты показывают, что ВПФ целесообразно применять при количестве деталей от 10 до 50 штук.

Технология ВПФ показана на рис. 1 [1].

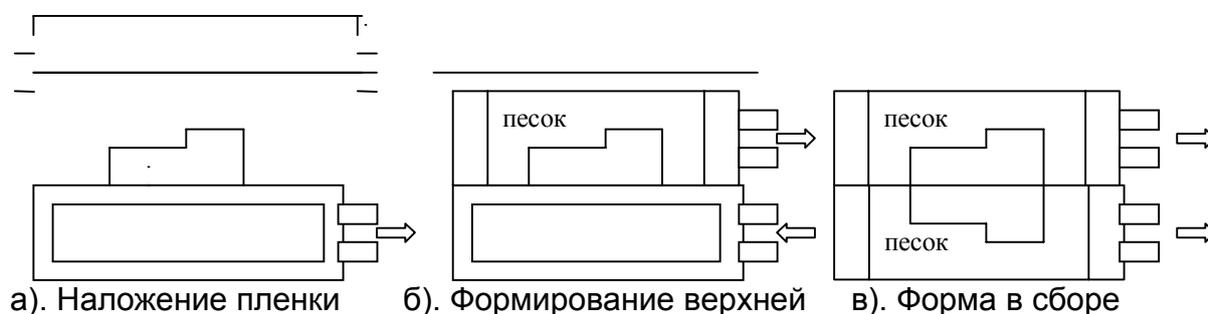


Рис. 1. Схема технологического процесса при вакуумно-пленочной формовке

Над модельной плитой (рис. 1а), полость которой соединяется с вакуумной системой и сообщается с атмосферой через сквозные каналы, устанавливается нагреватель с открытой спиралью и температурой нагрева 700–800 °С. Пленка разогревается под действием нагревателя до пластического состояния и накладывается на плиту с моделью, где за счет созданного разряжения прижимается к поверхностям модели и плиты. Затем на модельную плиту (рис. 2б) устанавливается опока, имеющая двойные стенки, полости которых также соединены с вакуумным насосом. В опоку засыпают сухой песок, уплотняют его на вибростенде и накрывают сверху подогретой пленкой. Полость опоки подключают к вакуумной системе, а модельную плиту отсоединяют. Полуформу снимают с модельной плиты, при этом опока постоянно должна быть соединена с вакуумной системой. Нижняя полуформа изготавливается аналогично верхней. На рис. 1в изображена форма в сборе. Разность давлений в теле и полости формы является основным фактором формирования литейной формы под

вакуумом. Эту разность необходимо поддерживать до конца заливки. Отливка удаляется из формы простым отключением вакуумного насоса.

Главное отличие ВПФ от известных способов литья в разовые песчаные формы заключается в том, что формовочная смесь не содержит связующего, а необходимые прочностные и размерные характеристики литейной формы достигаются путем создания разрежения в объеме формовочной смеси за счет применения герметизирующей пленки. При этом исключаются операции смесеприготовления, затраты на связующие и резко сокращается расход формовочных песков. Процесс легко поддается механизации и автоматизации.

Конструкция модельной оснастки для ВПФ несколько отличается от оснастки для изготовления форм традиционным способом. Для нее характерны небольшие литейные уклоны, которые объясняются тем, что силы трения между моделью и пленкой незначительны, и при вытяжке модели из формы требуются усилия в 20 раз меньшие, чем при обычном литье "в землю". Анализ экспериментальных данных различных источников показывает, что при определении припусков на механическую обработку величину их можно уменьшить на 25–35% по сравнению с предусмотренными ГОСТ 2009–78 и ГОСТ 1855–71 для соответствующего класса точности. Выбор материала модели определяется серийностью производства, требуемой точностью размеров и обеспечением чистоты поверхности отливок. Поскольку модели не контактируют с формовочным составом, у них практически отсутствует износ. Модельная оснастка, сохраняя длительный срок службы, может изготавливаться из недорогих материалов: дерева, пластмасс, гипса.

Была спроектирована и изготовлена деревянная модельная оснастка с припуском 2 мм на механическую обработку для двух тонкостенных деталей – "панель" и "рамка", имеющих толщину стенок 2,6–3,0 мм, и с припуском 5 мм для сложной детали "корпус" с внутренними полостями и толщиной стенок 5 мм. Проектирование оснастки включало в себя:

- выбор положения отливки в литейной форме,
- определение поверхности разъема формы,
- определение мест расположения прибылей,
- расчет и построение литниковой системы,
- расчет стержней,
- проверку спроектированной оснастки.

Большое влияние на физико-механические свойства технического металла в литом состоянии оказывает литниковая система, от которой зависит механизм заполнения расплавом полостей литейной формы и в конечном итоге уровень усадочной пористости в отливках [2]. Формирование пор происходит в период кристаллизации и последующего затвердевания отливки, когда объемная усадка не компенсируется поступлением жидкого металла из литников питающей системы. Последующее охлаждение затвердевшей отливки не оказывает существенного влияния на величину и характер пористости. В настоящее время отсутствуют надежные методы расчета литниковой системы, пригодные для любой отливки. Методики расчета заполняемости форм можно условно разделить на 2 группы: эмпирические, являющиеся результатами статистической обработки экспериментальных и производственных данных, и теоретические, разработанные на основе использования в качестве исходной модели течение расплава в пробах на жидкотекучесть и на основе использования законов гидравлики. При построении литниковых систем были использованы упрощенные модели процессов заполнения формы и процессов, происходящих на стадии формирования отливки. Было принято допущение, что при толщинах стенок отливки более 5 мм трением в полости формы можно пренебречь. Тем не менее, даже эта приближенная оценка позволила получить отливки без геометрических дефектов уже на первых образцах.

Для опробования был выбран сплав АК7 (аналог АЛ9), имеющий хорошую жидкотекучесть. Для изготовления отливок применялась отечественная пленка сэвилен, в качестве формовочного материала – песок марки К 016. Вакуум величиной 0,7 атм создавался водокольцевым вакуумным насосом модели ВВН. Вид опытной отливки детали "рамка" представлен на рис. 2.

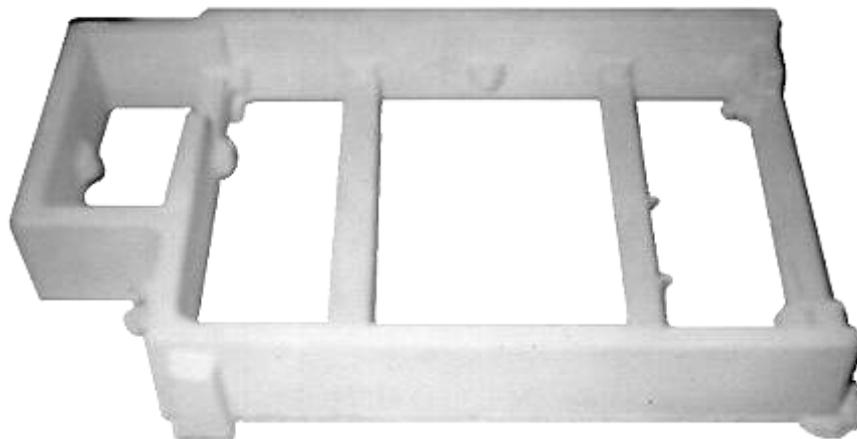


Рис. 2. Опытная отливка детали "рамка"

На рис. 3 приведен внешний вид детали "корпус". При получении отливок детали "корпус" для формирования внутренних полостей применялись стержни.

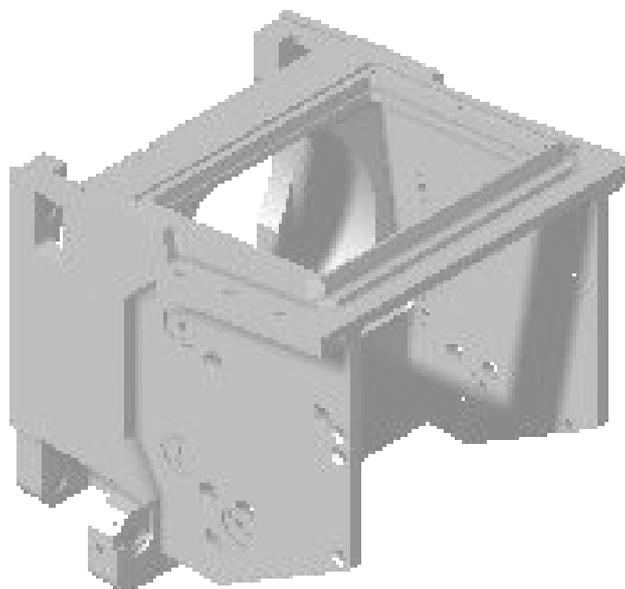


Рис. 3. Схематичный вид детали "корпус"

Получение высоких механических и технологических свойств отливок возможно только из качественного жидкого расплава [3]. Этому было уделено особое внимание. Выплавка опытных образцов производилась в индукционной печи с применением графитосодержащего тигля. Температура в печи находилась в пределах 700–720°С.

Для рафинирования металла использовался гексахлорэтан. Модифицирование осуществлялось лигатурой алюминий – цирконий.

Химический состав и механические свойства опытных плавок приведены в табл. 1. Видно, что наиболее высокие показатели по механическим свойствам – у сплава АК7 после проведения термообработки по режиму Т5. Этот режим использовался в дальнейшем для термообработки получаемых отливок.

Марка сплава	Химический состав, масс. %								Механические свойства		Примечание
	Mg	Cu	Be	Ti	Fe	Si	Mn	Zn	σ_B , МПа	δ , %	
АК7	0,3	0,68	--	--	1,2	7,0	0,26	--	140	1,0	литье в песчано-глиняные формы, сырое состояние
									142	1,1	
									174	1,4	литье по способу ВПФ, сырое состояние
									178	1,2	
									214	1,28	литье по способу ВПФ, режим термообработки Т5
								217	1,25		

Таблица 1. Физико-химические свойства опытных плавок

В первых полученных отливках детали "панель" была обнаружена значительная пористость. Для повышения скорости теплоотвода был применен стальной холодильник, что свело пористость к минимуму. В тепловых узлах отливки "рамка" наблюдались усадочные раковины, которые были устранены добавлением прибылей.

Технические условия на деталь "корпус" предполагают ее герметичность с применением газовой рабочей среды. Вопрос герметичности конструкции тесно связан с пористостью, являющейся функцией характера затвердевания. Статистический анализ брака деталей пневмогидравлических агрегатов и систем показывает, что основным дефектом, вызывающим разгерметизацию, является усадочная пористость.

Используемые в пневмоагрегатостроении литейные сплавы условно можно разделить на две группы.

1. Сплавы, обладающие высокими прочностными показателями, пластическими характеристиками и низкими технологическими свойствами. К ним относятся сплавы типа твердого раствора, например АМ5. Эти сплавы применяют при изготовлении нагруженных деталей, испытывающих ударные и знакопеременные нагрузки.
2. Сплавы, обладающие высокими прочностными показателями, удовлетворительными пластическими характеристиками и высокими технологическими свойствами. Сюда относятся сплавы систем Al-Si, Al-Si-Mg, в частности, АК7. Они обладают высокими литейными свойствами и применяются для высокогерметичных и нагруженных деталей сложной конфигурации. В табл. 2 приведены свойства основных представителей этих групп.

Сплав	Температурный интервал кристаллизации, °С	Жидкотекучесть мм	Линейная усадка, %	Герметичность МПа
АК7	33	360	1,0	19
АМ5	100	205	1,25	7

Таблица 2. Технологические свойства и герметичность сплавов

Большое число работ для предупреждения пористости и получения герметичных отливок рекомендует использование сплавов, содержащих процент эвтектики не менее 60% [4]. На практике в качестве оценочной меры герметичности отливок из

алюминиевых сплавов и применимости их для деталей пневмоагрегатов принята и используется шкала герметичности, представленная в табл. 3.

Марка сплава	Содержание эвтектики в сплаве по основным компонентам, %	Состояние поверхности и толщина стенок	Уровень герметичности в баллах в зависимости от давления, атм			
			5	10	20	30
АК7	60	Литая – 3 мм, механически обработанная – 4 мм	1	1	1	1
АЛ3	40	Литая – 5 мм, механически обработанная – 4 мм	1	1	1	1
АЛ40	25	Литая – 3 мм, механически обработанная – 4 мм	1	1	1	-
АЛ19	19	Литая – 3 мм, механически обработанная – 4 мм	2	3	3	-
			4	4	-	-

Примечание: Балл 1 – нет утечки, Балл 2 – 1–10 мл/мин, Балл 3 – 10–100 мл/мин, Балл 4 – выше 100 мл/мин

Таблица 3. Шкала герметичности алюминиевых литейных сплавов для точного литья

Эти данные были приняты во внимание при выборе алюминиевого сплава для изготовления детали "корпус".

В результате проведенной работы получены опытные образцы отливок деталей "панель", "рамка" и "корпус". На поверхности отливок отсутствуют внешние дефекты. Исследования механических свойств образцов, вырезанных из отливок, показали их соответствие техническим требованиям по ОСТ 90021–79. Размерные точности изготовленных из отливок деталей соответствуют требованиям конструкторской документации.

Разработана технология литья алюминиевого сплава АК7 по способу ВПФ, позволяющая в короткие сроки получать заготовки, приближенные к форме готовых изделий. Литейные заготовки технологичны, обладают хорошей обрабатываемостью. Метод перспективен для применения в точном приборостроении крупногабаритных отливок сложной формы. По результатам анализа литературных источников проведено обоснование использования сплава АК7 при изготовлении пневмогерметичной детали "корпус". Вопрос соответствия опытных образцов корпуса требованиям по герметичности, позволяющим эксплуатировать изделия с обеспечением заданного ресурса, требует дополнительной проработки и проведения пневмоиспытаний имеющихся отливок и деталей.

Литература

1. Рябинкин В.Д., Кузнецов В.П. Оборудование для вакуумно-пленочной формовки. // Литейное производство. 1993. №9.
2. Берг П.П. Основы построения технологических формул для оценки тепловых процессов в литейных формах при затвердевании металла. Сб. статей. М.: Машгиз, 1958.
3. Пирязев В.П. Разработка и исследование высокопрочного алюминиевого сплава и технологии литья сложных тонкостенных деталей. Диссер. канд. техн. наук, Горький, 1977.
4. Эскин Г.И. Точное литье авиационных агрегатов из алюминиевых сплавов. М.: Машиностроение, 1967.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА МОНИТОРИНГА ЭЛЕКТРОМАГНИТНЫХ ПОЛЕЙ

А.Г. Коробейников, М.Ю. Тимошенкова

Сегодня почти каждый человек осознает, что загрязнение окружающей среды достигло огромных размеров и катастрофически прогрессирует. Экологическая опасность вызвана ускоренным развитием различных отраслей промышленности. Обычно основное внимание уделяется загрязнению окружающей среды веществами. Однако в последнее время появились указания на то, что загрязнение среды различного рода физическими полями, в частности, электромагнитным, оказывает не менее существенное влияние на живые организмы, в том числе на человека.

Организму человека свойственны определенные ритмы функционирования – так называемые биоритмы. Их существование определяется работой различных систем организма человека, таких как нервная, кровеносная и другие, с физической точки зрения представляющих собой резонансные системы с высокой электропроводностью, т.е. электрические резонансные контура.

Воздействие на организм внешних переменных магнитных полей на резонансных частотах (частотах биоритмов) приводит к нарушению его нормального функционирования. Люди, живущие в крупных промышленных центрах, практически постоянно находятся под воздействием низкочастотных магнитных полей, создаваемых линиями электропередач как переменного (50 Гц), так и постоянного (линии питания электропоездов, трамваев, метро) тока, мощными промышленными и бытовыми электродвигателями и электроустановками.

Первоначально основное внимание уделялось полям на частоте 50 Гц (60 Гц в США). Гораздо меньше изучены переменные магнитные поля ультранизкочастотного (УНЧ) диапазона 0,001–10 Гц, хотя именно они представляют наибольший интерес по двум причинам:

- магнитная составляющая более опасна, чем электрическая;
- УНЧ магнитная окружающая среда имеет особое биологическое значение, так как соответствует основным физиологическим ритмам организма человека.

Магнитные поля от электротранспорта отличаются от полей, производимых линиями электропередач, большей сложностью, наличием резких скачков интенсивности и компонент в широком диапазоне частот, преимущественно УНЧ. Магнитные поля от электротяги, в отличие от магнитных полей линий электропередач, до последнего времени были мало изучены, так как не существовало специальной измерительной аппаратуры, способной производить непрерывную запись информации и ее хранение. Сейчас такие приборы существуют, например оригинальный трехкомпонентный магнитометр MVC-3DS.

В настоящее время обнаружены УНЧ электромагнитные литосферные эмиссии, регистрируемые на различных стадиях развития очагов сильных землетрясений, в диапазоне частот 0,01–10 Гц. УНЧ электромагнитные литосферные эмиссии достоверно выделяются на фоне геомагнитных пульсаций и техногенных помех. Как правило, первые всплески УНЧ электромагнитных литосферных эмиссий регистрируются за несколько недель до мощного толчка, а за несколько часов до него наблюдается резкое увеличение интенсивности УНЧ электромагнитных литосферных эмиссий.

Биологические предвестники землетрясений давно и хорошо известны. Рыбы, земноводные, птицы и млекопитающие необычно ведут себя: уходит от эпицентра землетрясения, спасая свою жизнь за дни-часы до сильного толчка. Изменяются показатели системы крови у человека. Однако природа биопредвестников до сих пор неизвестна. Существует ряд гипотез, в частности, предположение об электромагнитной

природе агента, воздействующего на живые организмы перед землетрясением. Был проведен ряд экспериментов, в которых показана чувствительность биологических систем к УНЧ электромагнитным полям.

Ультранизкочастотные литосферные эмиссии (УЛЭ), наблюдаемые перед сильными землетрясениями, могут воздействовать на живые организмы, вызывая стрессовую коллективную реакцию популяций или отдельных особей. При этом информационный сигнал может быть закодирован в спектральных и пространственно-временных характеристиках УЛЭ. Поэтому в экспериментах с искусственными УНЧ электромагнитными полями наиболее эффективно проводить воздействие на биоорганизмы модельным сигналом, максимально приближенным к естественному.

УНЧ электромагнитные поля являются биологически активными, и их спектрально-временные характеристики могут быть экологически значимыми для живых организмов. В стрессовых ситуациях действует комплекс причин, но можно предполагать, что как во время магнитной бури, так и перед землетрясением УНЧ электромагнитный сигнал, его спектрально-временная характеристика является носителем информации, воспринимаемой биосистемой.

Таким образом, не исключено, что одной из основных причин, вызывающих необычное поведение живых организмов перед сильными землетрясениями, являются ультранизкочастотные электромагнитные литосферные сигналы, источник которых расположен в районе очаговой зоны готовящегося землетрясения.

Измерения электромагнитных излучений при помощи MVC-3DS проводится для решения многих задач, например:

- оперативного контроля и мониторинга техногенной электромагнитной обстановки в различных районах;
- оперативного контроля появления магнитных бурь;
- проведения поиска так называемых "магнито-спокойных мест";
- проведения геологической разведки;
- предсказания землетрясений.

Для решения каждой задачи необходимо свое программное обеспечение (ПО). Например, при проектировании ПО для мониторинга и цифровой регистрации переменных магнитных, электрических и сейсмических полей необходимо решение следующих задач:

- создание многооконного интерфейса;
- мониторинг в реальном масштабе времени и сбор данных в цифровом виде на жесткий диск IBM-совместимого компьютера или на любой другой накопитель информации (например, магнито-оптический);
- управление магнитометром MVC-2DS;
- графическое отображение на экране монитора измеряемых сигналов в реальном времени;
- запись измеренных сигналов на жесткий диск РС в бинарном формате в реальном масштабе времени;
- конвертирование данных измерений из бинарного формата в ASCII формат.
- автоматическое масштабирование входных сигналов;
- цифровой контроль выходных сигналов магнитных и электрических датчиков;
- продолжительная (в пределах 72 часов) запись данных в файл;
- экспорт ASCII файлов.

Созданное ПО должно функционировать совместно с оборудованием, удовлетворяющим следующим техническим характеристикам:

- разрядность аналого-цифровых преобразователей (АЦП) 24 бита;
- частота дискретизации АЦП 50 отсчетов в секунду (50 Гц);

- коэффициент подавления сигнала встроенным в АЦП цифровым рекурсивным режекторным фильтром на частоте дискретизации АЦП 200 дБ;
- измеряемые величины – три взаимно-ортогональные компоненты магнитного поля (H , D и Z) и любые три компоненты электрического (теллурического) поля;
- тип датчиков магнитных компонент – магнитостатический (*torsion*);
- коэффициент подавления сигнала на частоте 50 Гц аналоговыми фильтрами 30 дБ;
- диапазон измеряемых частот 0–10 Гц;
- диапазон измерений каждого магнитного датчика (± 10 В по выходному напряжению) ± 5000 нТл $\pm 5\%$;
- интегральный уровень шумов для каждой магнитной компоненты (*peak to peak*) в частотном диапазоне 0–10 Гц (центральная частота 1 Гц) с 99% интервалом вероятности не более 0.01 нТл;
- постоянная внутренних калибровочных колец каждого из трех магнитных (H , D и Z) датчиков 170 ± 10 нТл/мА;
- тип датчиков электрических компонент – любые с выходным сопротивлением менее 100 КОм (в данной версии – заземленные электроды, разнесенные на ~ 100 м);
- диапазон измерений для каждой электрической компоненты ± 2.5 В;
- интегральный уровень шумов для каждой электрической компоненты (*peak to peak*) в частотном диапазоне 0–10 Гц (центральная частота 1 Гц) с 99% интервалом вероятности 10 мкВ.

Разработанное ПО использовалось в составе:

интегрированного комплекса для исследования особенностей распространения гидромагнитных волн вдоль земной поверхности и проведения магнитно-теллурического зондирования в субавроральной зоне (п. Лехта);

систем сбора и обработки информации в интегрированном комплексе для мониторинга геомагнитного поля в сейсмоактивных районах Грузии (г. Душети).

ИССЛЕДОВАНИЕ СТРУКТУРНЫХ ДЕФЕКТОВ МЕЖСЛОЙНОЙ ИЗОЛЯЦИИ В КМОП ИС

С.Н. Новиков, А.Е. Семенов, А.М. Скворцов, Е.Г. Фролкова, Р.А. Халецкий

В настоящей работе приведены результаты исследований пористости слоев окислов кремния, полученных термическим окислением поликристаллического кремния ($Si_{п/к}$), осаждаемого на планарную структуру.

Структура содержала изолированные области со сформированными в них областями стоков и истоков n- и p-канальных МОП-транзисторов и подзатворный слой SiO_2 . Осаждение слоя $Si_{п/к}$ производилось газо-фазным методом на установке УНЭС. Толщина слоя составляла 0,5 мкм. В серийной технологии КПОП ИС этот слой используется для формирования затворов МОП-транзисторов и первого слоя коммутации элементов микросхемы. Для снижения сопротивления коммутирующего слоя после нанесения поликремний легируется диффузией фосфора. Далее следует операция травливания фосфорно-силикатного стекла (ФСС) и процесс фотолитографии в поликремнии (формирование затворов и шин коммутации).

В производстве КМОП ИС серии 590 между поликремниевым и алюминиевым слоями коммутации в качестве межслойной изоляции на установке типа "Оксин" в среде аргона газо-фазным методом осаждается слой ФСС толщиной 0,5–0,6 мкм. Эта операция достаточно сложна и не всегда обеспечивает стабильные и повторяемые результаты. Поэтому конечная цель настоящего исследования – определение возможности замены межслойной изоляции на основе ФСС слоем SiO_2 , полученным путем термического окисления поверхности коммутирующего слоя поликремния.

Из приведенных в обзоре [1] методов исследования пористости слоев диэлектриков авторами был выбран метод электрографии по следующим причинам. Он достаточно информативен и нагляден, прост в реализации, а для обработки результатов может быть использована компьютерная графика, существенно упрощающая и ускоряющая подсчет плотности дефектов окисла. Наконец, именно этот метод используется в производстве КМОП ИС на предприятии "Светлана-Полупроводники".

Сущность этого метода заключается в получении картины распределения дефектов на фотобумаге, смоченной раствором гидрохинона и прижатой к подложке, вследствие выделения зародышей серебра в местах протекания тока через дефект. Разновидности метода электрографии и подробное их описание приводится в работах [2–4]. Отличие методики, примененной в настоящей работе, заключается в том, что в площадь электрограммы с помощью компьютерной графики вписывается максимально возможное число квадратов площадью в 1 см^2 . По результатам подсчета пор в каждом квадрате рассчитывается средняя плотность пор на подложке.

На рис.1 показаны структуры двух видов подложек, отличающихся методами их формирования, а в таблице – режимы получения этих структур.

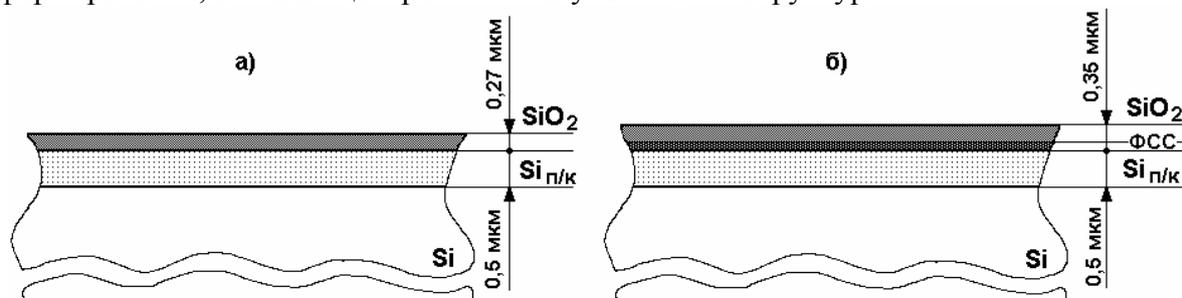


Рис.1. Структура исследованных образцов: а - образец №1; б - образец №2

Для изготовления указанных образцов использовался кремний монокристаллический марки КЭФ7,5(100), используемый для получения контрольных тестовых образцов при изготовлении ИС серии 590. Подготовка подложек и изготовление экспериментальных образцов производились на предприятии "Светлана-Полупроводники" по действующей технологии 590 серии. Вводимое в существующий процесс окисление $Si_{п/к}$ производилось в одном случае после легирования $Si_{п/к}$ фосфором, т.е. при сохранении на его поверхности ФСС, во втором случае – после стравливания ФСС.

Вид образца	Основные технологич. параметры
$Si_{п/к+P}$ затем ФСС стравлено (образец №1)	$h=0,5$ мкм+ легир.Р при $t^{\circ}=860$ °С
$Si_{п/к+P}$ (образец №2)	$h=0,5$ мкм + легир.Р при $t^{\circ}=860$ °С
SiO_2 (образец №1)	$h=0,27$ мкм, $t^{\circ}=1000$ °С, вл. O_2+HCl
SiO_2 (образец №2)	$h=0,35$ мкм, $t^{\circ}=1000$ °С, вл. O_2+HCl

Таблица. Основные режимы изготовления образцов

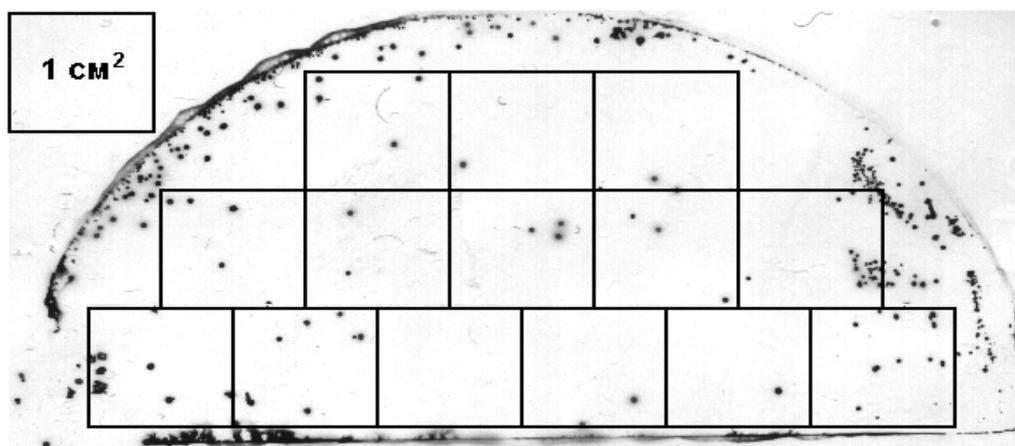


Рис. 2. Электрограмма образца № 1

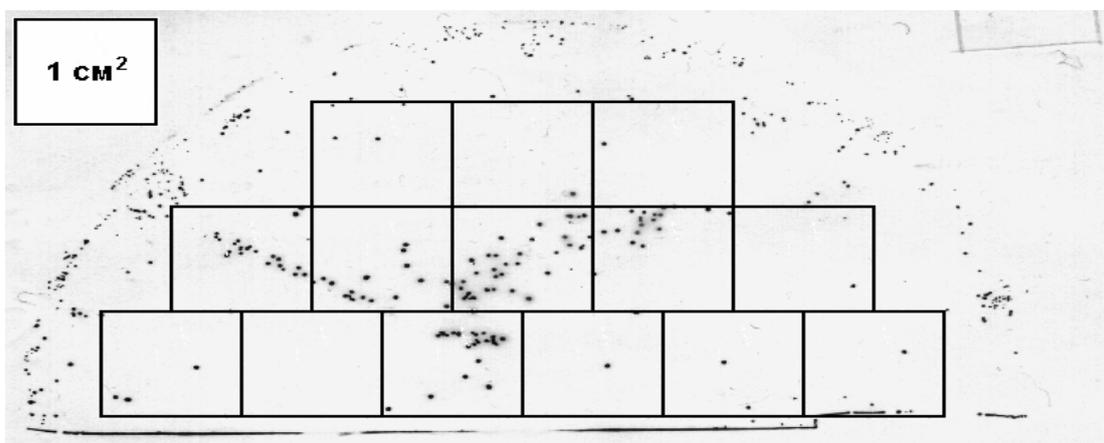


Рис. 3. Электрограмма образца № 2

На рис. 2 и рис. 3 приведены электрограммы образцов с нанесенными на них измерительными квадратами. Расчет среднего количества пор ("проколов") дал следующие результаты. На образце № 1 с окислением $Si_{п/к}$ после снятия ФСС число пор приближается к 5 шт/см². Средняя плотность пор на образце с окислением $Si_{п/к}$, покрытого слоем ФСС (образец № 2), превышает 8 пор/см². Размеры пор на образце № 2 меньше, чем на образце № 1, и лежат в пределах до 5 и до 120 мкм, соответственно.

Таким образом, меньшая дефектность термического окисла кремния обеспечивается при окислении поликристаллического кремния свободного от фосфоросиликатного стекла. Следовательно, такой окисел предпочтительнее для межслойной изоляции шин коммутации. Однако для окончательного решения необходимо провести аналогичные исследования повторно, а также исследовать электрическую прочность полученных окислов.

Литература

1. Концевой Ю.А., Филатов Д.К. Дефекты кремниевых структур и приборов. Часть 2.- М: ЦНИИ "Электроника", 1987. 68 с.
2. Носиков С.В., Пресс Ф.П. Электрографический метод исследования слоев двуокиси кремния и нитрида кремния. // Электронная промышленность. 1971. Вып.1. С. 69–73.
3. Носиков С.В., Иерусалимчик И.Г. Электрохимические процессы в электрографической системе с бромсеребряной бумагой. // Электронная техника. Сер. 2. Полупроводниковые приборы. 1972. Вып.1. С.125–129.
4. Вайнштейн А.Б., Власова Г.В., Королева Г.И. и др. Исследование сплошности тонких диэлектрических пленок методом электрографии. // Электронная техника. Сер. 3. Микроэлектроника. 1974. Вып.2. С. 57–61.

ИССЛЕДОВАНИЕ ПРОБИВНЫХ НАПРЯЖЕНИЙ МЕЖСЛОЙНОЙ ИЗОЛЯЦИИ В КМОП ИС

С.Н. Новиков, А.Е Семенов, А.М. Скворцов, Е.Г. Фролкова, Р.А. Халецкий

В настоящей работе приведены результаты исследований электрической прочности слоев окислов кремния, полученных термическим окислением поликристаллического кремния ($Si_{п/к}$), осаждаемого на планарную структуру. Электрическая прочность является одним из важнейших параметров диэлектрика применяемого для межслойной изоляции.

Структура содержала изолированные области со сформированными в них областями стоков и истоков n- и p-канальных МОП-транзисторов и подзатворный слой SiO_2 . Осаждение слоя $Si_{п/к}$ производилось газо-фазным методом на установке "Изотрон – 2М". Толщина слоя составляла 0,5 мкм. В серийной технологии К-МОП ИС этот слой используется для формирования затворов МОП-транзисторов и первого слоя коммутации элементов микросхемы. Для снижения сопротивления коммутирующего слоя после нанесения поликремний легируется диффузией фосфора. Далее следует операция травливания фосфорно-селикатного стекла (ФСС) и процесс фотолитографии в поликремнии (формирование затворов и шин коммутации).

В производстве К-МОП ИС серии 590 между поликремневым и алюминиевым слоями коммутации в качестве межслойной изоляции на установке типа "Изотрон – 3М" в среде аргона газо-фазным методом осаждается слой SiO_2 толщиной 0,8 мкм. Эта операция достаточно сложна и не всегда обеспечивает стабильные и повторяемые результаты. Поэтому конечная цель настоящего исследования – определение возможности замены межслойной изоляции на основе слоя SiO_2 , осажденного газо-фазным методом, слоем SiO_2 , полученным путем термического окисления поверхности коммутирующего слоя поликремния.

Из приведенных в обзоре [1] методов исследования электрической прочности слоев диэлектриков авторами был выбран метод пробивных напряжений. Сущность этого метода заключается в получении распределения пробивных напряжений окисла по пластине. Для этого производится большое количество измерений в различных точках, распределенных по пластине. Отсчет напряжения пробоя производится на шкале характеристикографа в точке резкого изменения вида вольт-амперной характеристики окисленной структуры.

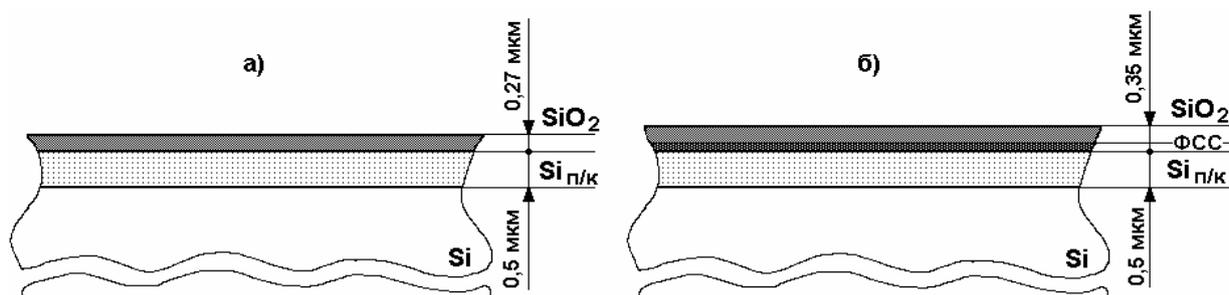


Рис.1. Структура исследованных образцов: а – образец № 1; б - образец № 2

Пробивные напряжения окисленной структуры измерялись на характеристикографе Л2-56. Для этого на пластину устанавливается игольчатый зонд для осуществления электрического контакта. Затем производится последовательное увеличение

напряжения до пробоя слоя SiO₂. Напряжение пробоя определяется по проградуированной шкале характеристики Л2-56.

На рис. 1 показаны структуры двух видов подложек, отличающихся методами их формирования, а в таблице – режимы получения этих структур. Для изготовления указанных образцов использовался кремний монокристаллический марки КЭФ7,5(100), используемый для получения контрольных тестовых образцов при изготовлении ИС серии 590. Подготовка подложек и изготовление экспериментальных образцов производились на предприятии "Светлана-Полупроводники" по действующей технологии 590 серии. Вводимое в существующий процесс окисление Si_{п/к} производилось в одном случае после легирования Si_{п/к} фосфором, т.е. при сохранении на его поверхности ФСС, во втором случае – после стравливания ФСС.

Вид образца	Основные технологич. параметры
Si п/к+P+ затем ФСС стравлено (образец №1)	h=0,5 мкм+ легир.Р при t°=860 °C
Si п/к+P (образец №2)	h=0,5 мкм + легир.Р при t°=860 °C
SiO ₂ (образец №1)	h=0,27 мкм, t°=1000 °C, вл. O ₂ +HCl
SiO ₂ (образец №2)	h=0,35 мкм, t°=1000 °C, вл. O ₂ +HCl

Таблица. Основные режимы изготовления образцов

В результате проведения измерений были получены значения пробивных напряжений распределенных пластине, как уже было указано выше. Для описания полученных результатов измерений пробивных напряжений использовалась величина напряженности поля, соответствующая пробоем окисла. Результаты измерений, для наглядного представления, обрабатывали статистическими методами анализа многократных измерений. Результаты измерений представлены на рис. 2 в виде гистограммы распределения напряженностей поля, соответствующих пробоем в слое окисла.

Напряженность поля вычисляли по формуле: $E = \frac{U_{\text{ПР}}}{D}$, где D – толщина диэлектрика; $U_{\text{ПР}}$ – напряжение пробоя [2].

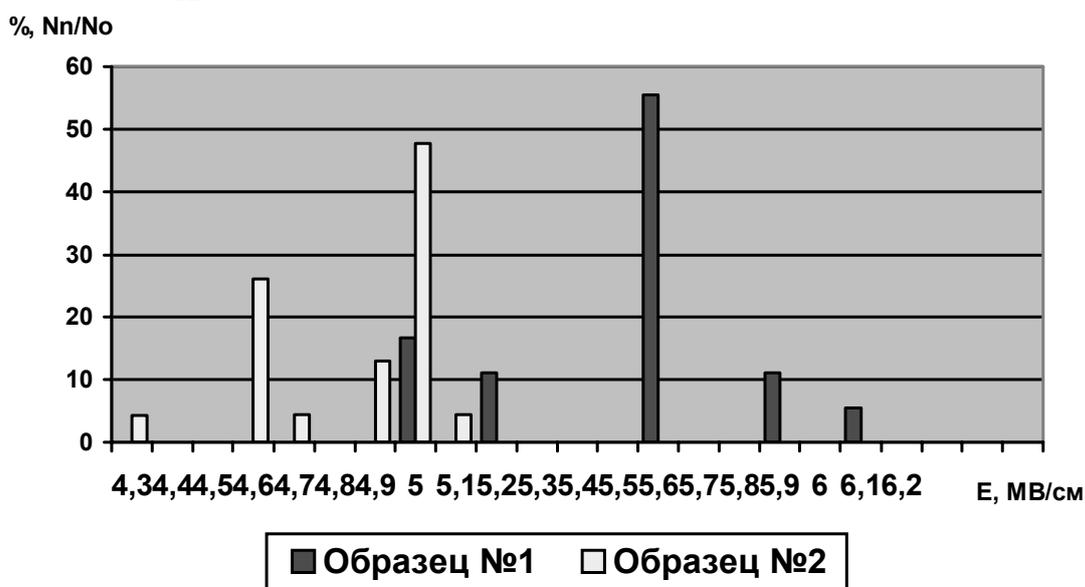


Рис. 2. Гистограмма напряженностей поля, соответствующих пробоем диэлектрика в исследуемых образцах

Как видно из представленной на рис. 2 гистограммы, у образца № 1, окисленного после снятия ФСС, общее распределение напряженностей электрического поля SiO_2 , соответствующих пробею, лежит в пределах от 5,6 МВ/см до 6,1 МВ/см, а у образца №2 – в пределах от 4,6 МВ/см до 5,1 МВ/см.

Основная часть измерений пробивных напряжений SiO_2 , у образца №1 характеризуется значением напряженности поля, соответствующей пробею, равным 5,6 МВ/см (75%) и более, а у образца №2 только 47% достигает значения 5,0 МВ/см.

Таким образом, образец № 1, с окислением $\text{Si}_{п/к}$ после снятия ФСС, имеет лучшие характеристики диэлектрической прочности по сравнению с образцом № 2. Следовательно, параметры изготовления образца № 1, представленные в таблице, могут быть рекомендованы при формировании межслойной изоляции шин коммутации.

Литература

1. Воробьев А.А., Завадовская Е.К. Электрическая прочность твердых диэлектриков. М.: Гостехиздат, 1956. 312с..
2. Кухлинг Х. Справочник по физике. М.: Мир, 1982. 323 с.

**ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС
ДЛЯ УСТАНОВЛЕНИЯ ХАРАКТЕРИСТИК ГАЗОРАЗРЯДНОГО
СВЕЧЕНИЯ "ГРВ-КОМПЛЕКС" И ПЕРСПЕКТИВЫ
ЕГО ИСПОЛЬЗОВАНИЯ ПРИ АНАЛИЗЕ СОСТОЯНИЯ
ОРГАНИЗМОВ И ВЫЯВЛЕНИИ ЭКОЛОГИЧЕСКОГО РИСКА**

**Н.С. Прияткин, К.Г. Коротков, Э.И. Слепян,
Г.С. Авдеева, Т.Б. Дорофеева, Г.О. Пономаренко**

Экологический риск, экологически обусловленные патологические явления и симбиотрофная заболеваемость, во многих отношениях экологически зависимая, – реальность как индустриального, так и постиндустриального общества (Слепян, 2002а), очевидное следствие усложнения, в том числе возрастающей стратификации, общества и возникновения кризисных ситуаций во взаимоотношениях общества и природы. Насущная проблема в связи с этим, требующая объективного и действенного решения, – обоснование и разработка аппаратных технологий, обеспечивающих возможность выявления и краткосрочной дифференциальной численной диагностики изменений, имеющих значение, во-первых, свидетельств возникновения преморбидных (донозологических) отклонений в строении и функционировании организмов (как продуцентов, так и консументов и редуцентов), и, во-вторых, свидетельств появления начальных нарушений, возникающих в физических средах – в воздушном бассейне, в водных массах, в донноосадочном покрове, в почвенном покрове и в скальных массивах экотопов (биотопов). В особенности существенны упомянутые технологии для своевременного выявления свидетельств и предупреждения экологических чрезвычайных ситуаций, экологических катастроф, диверсионных актов экологического терроризма (Слепян 2002б) и ликвидации их последствий, а также при создании, поддержании и увеличении эффективности систем экологической защиты.

Одна из наиболее действенных и, что принципиально, оперативных технологий сверхраннего выявления и диагностики нарушений физических сред обитания и жизнедеятельности организмов – технология установления характеристик газоразрядного свечения [газоразрядная визуализация (ГРВ)], осуществляемая с использованием специального программно-аппаратного комплекса – ГРВ-комплекса (Коротков, 2001 и др. раб.).

Технология газоразрядной визуализации, основанная на эффекте Кирлиан, имеет значение методики (ГРВ-методики), позволяющий оценивать энергетическое (фрактально-энтропийное) состояние абиотических и биотических составляющих экологических систем (Коротков, 2001). Современная аппаратура, применяемая при использовании ГРВ-методики, дает возможность регистрировать и анализировать газоразрядное свечение, индуцированное у организмов и у их структур, в том числе у биологических жидкостей, у природных вод, почв, грунтов, включая и химически загрязненные.

ГРВ-методика использована при анализе состояния пресных вод (водопроводной, загрязненной сточной, минеральной), эфирных масел, урины, жидкости крови, культуральной жидкости дрожжей, растений, беспозвоночных и позвоночных животных – олигохет, рыб, грызунов (Гудакова и соавт., 1988, 1990, Прияткин и соавт., 2003).

Установлено (Прияткин и соавт., 2002, 2003), что при интегральной экспресс-оценке состояния загрязненных вод активного стока, осуществляемой с применением ГРВ-методики, наиболее демонстративно сочетание фрактально-энтропийных параметров одиночных газоразрядных отображений (ГРВ-грамм) и характеристик тренда динамических кривых изменения абсолютной площади засветки, устанавливаемых посредством динамической ГРВ-графии (Коротков и соавт., 2001).

Следует отметить, что ГРВ-методика информативна при определении сравнительной активности жизнедеятельности растений, функционирующих в условиях стресса (Коротков, 2001), возникающего при подверженности растений повреждающим воздействием окружающей среды, пестицидов, возбудителей инфекций и инвазии, и т.д. Подтверждение изложенного, в частности, – специфика газоразрядного свечения тканей коры и древесины стебля однолетних побегов ильмовых (*Ulmus lacvis* Pall., *Ulmus grabra* Huds, *Ulmus scabra* Mill.) до и после колонизации *Graphium ulmi* Schwazz [сумчатая стадия – *Ophiostoma ulmi* (Buism.) Nannf.] – возбудителя трахеомикоза, которое является опаснейшим для ильмовых инфекционным заболеванием (рис. 1).

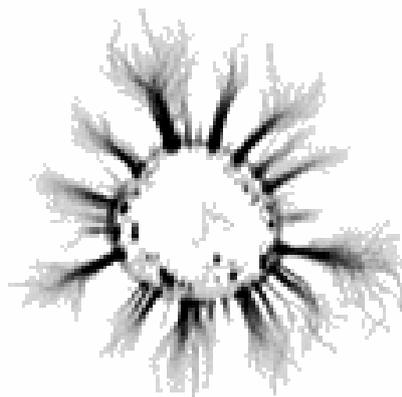


Рис. 1. Отображение газоразрядного свечения коры и древесины стебля однолетнего побега *Ulmus* spp. (поперечный разрез).

Совершенствование ГРВ-методики дает возможность использовать ее в диагностических целях при управлении ростом и развитием однолетних и многолетних растений, осуществлении интегрированной системы их защиты, установлении эффективности мероприятий по увеличению их хемотолерантности, и т.д.

Соответствующая адаптивная ГРВ-методика (Прияткин и соавт., 2002) позволила применить ее для оценки состояния жизнедеятельности красного калифорнийского дождевого червя *Eisenia fetida* (Savigny) – основного вида почвенных олигохет, используемых в целях вермикомпостирования.

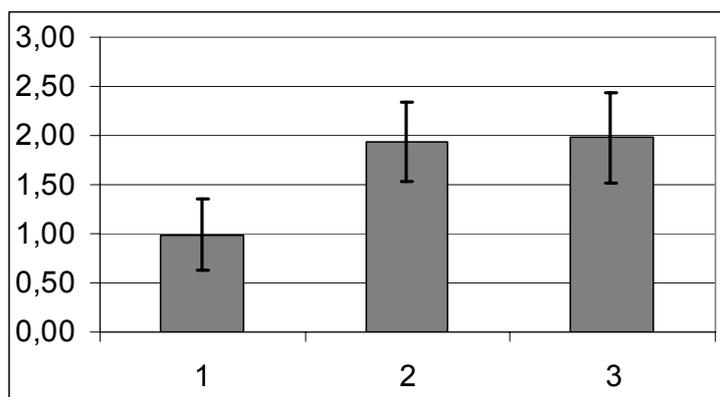


Рис. 2. Нормализованная площадь (в отн. ед.) ГРВ-отображений тела особей *Eisenia fetida* (Savigny) (ГРВ-фотографирование с фильтром), подвергнувшихся повреждающему воздействию: 1 - без воздействия, 2 - после мгновенного механического воздействия (надавливания), 3 - после выдерживания в водной среде 20 минут

Установленные различия ГРВ-параметров (нормализованной площади газоразрядного свечения, энтропии, фрактальности), характеризующие особи, во-первых, принадлежащие преемственным размерно-возрастным меропопуляциям и, во-вторых, подвергающиеся повреждающему механическому воздействию и воздействию водной среды (рис. 2), имеют, в связи с этим, определенное значение для определения каждой из меропопуляций в формировании гумуса.

Есть все основания полагать, что ГРВ-методика может быть использована при решении многих насущных вопросов экологии и ресурсоведения – для целей диагностики химического, механического и биологического загрязнения природных вод и почв, выявления аллелохимического почвоутомления и оценки состояния популяций организмов, для целей мониторинга и индикации, при наблюдении за жизнедеятельностью человека (в частности, в условиях подверженности воздействию ионизирующей радиации, электромагнитного, акустического, светового и вибрационного полей), а также для контроля за качеством продукции растениеводства и животноводства, обеспечивающим продовольственную безопасность.

Литература

1. Гудакова Г. З., Галынкин В.А., Коротков К.Г. Исследование характеристик газоразрядного свечения микробиологических культур. // Журнал. прикладной спектроскопии. 1988. Т. 49. №3. С. 412–417.
2. Дорофеева Т.Б., Тюпина Г.Н. Графиоз ильмовых в Санкт-Петербурге и меры борьбы с ним / Экология большого города. Альманах. Вып. 6. Ред. Х.Г. Якубов. Проблемы содержания зеленых насаждений в условиях Москвы // М.: Прима-М, 2002. С. 57–61.
3. Коротков К.Г. Основы ГРВ-биоэлектрографии. СПб., СПбГИТМО(ТУ), 2001. 360 с.
4. Коротков К.Г., Крыжановский Э.В., Борисова М.Б. Методика исследования объектов методом динамической ГРВ-графии. // Наука. Информация. Сознание. V международный научный конгресс по ГРВ-биоэлектрографии. Тезисы. СПб.: Издательский центр "Университетские Телекоммуникации", 2001. С.83–85.
5. Прияткин Н.С., Авдеева Г.С., Коротков К.Г., Слепян Э.И. Адаптивная методика получения газоразрядных изображений низших беспозвоночных животных методом ГРВ-биоэлектрографии и результаты ее применения при исследовании особей красного калифорнийского червя (*Eisenia fetida*) разных возрастных групп. // Наука. Информация. Сознание. VI Международный научный конгресс по ГРВ биоэлектрографии. Тезисы. СПб: ИВА, 2002. С.94–96.
6. Прияткин Н.С., Коротков К.Г., Слепян Э.И. Перспективы использования метода ГРВ-Биоэлектрографии в целях интегральной квалиметрии вод // Наука. Информация. Сознание. VI Международный научный конгресс по ГРВ биоэлектрографии. Тезисы. СПб: ИВА, 2002. С.92–93.
7. Прияткин Н.С., Коротков К.Г., Слепян Э.И., Бурцева Е.П. Интегральная оценка качества сточных вод предприятий здравоохранения с использованием методики газоразрядной визуализации в аспекте проблемы экологического риска. // Региональная экология. СПб., 2003. В печати.
8. Слепян Э.И. Экологический риск. // Региональная экология. СПб., 2002 а. №1–2(18). С. 62-82.
9. Слепян Э.И. Экологический терроризм // Жизнь и безопасность. СПб., 2002б. №1–2. С.277–283.

ПРИМЕНЕНИЕ САПР P-CAD-2001 ПРИ ПРОЕКТИРОВАНИИ ПЕЧАТНЫХ ПЛАТ

Е.Б. Романова

При разработке радиоэлектронной аппаратуры важным этапом является проектирование печатных плат. От разводки электрических связей зависит качество прохождения сигналов, а также размеры конструкции. Для достижения оптимальных результатов работы в области проектирования необходимо использовать современные средства САПР, обладающие целым рядом принципиально новых возможностей. Современной САПР для проектирования печатных плат является P-CAD-2001.

P-CAD предназначена для сквозного проектирования печатных плат. Она позволяет формировать электрические принципиальные схемы и топологию печатной платы, а также дает возможность оформления конструкторской и технологической документации. Выходные данные используются для вывода информации на различные устройства: принтеры, плоттеры, фотоплоттеры (для изготовления фотошаблонов), сверлильные станки и др.

Система P-CAD позволяет автоматизировать работу на всех этапах проектирования [1].

1. Формирование технического задания, включая:
 - оформление структурных и функциональных схем;
 - оформление текстовой документации;
 - разработку и моделирование электрической принципиальной схемы.
2. Конструирование печатных плат посредством разработки топологии, включая:
 - размещение компонентов на плате;
 - трассировку электрических соединений;
 - выпуск конструкторской документации.
3. Подготовку к производству, включая:
 - формирование файлов пригодных для изготовления;
 - выпуск технологической документации.

САПР P-CAD-2001 является удобным средством проектирования печатных плат. Она является одной из наиболее распространенных в России. Выбор этой САПР обусловлен поддержкой шрифтов TTF в обоих редакторах (Schematic и PCB) и адаптацией к отечественному оборудованию.

До сих пор в некоторых организациях при проектировании печатных плат используют версию P-CAD 4.5. Внедрение новой системы связано с тем, что разработчикам и конструкторам необходимо осваивать новую версию системы проектирования, а также настраивать оборудование для изготовления печатных плат, что приводит к финансовым и временным затратам. Но необходимо понимать, что в конечном итоге будет получен значительный выигрыш во времени за счет преимуществ новой системы и, следовательно, прирост финансов, так как скорость выполнения заказа возрастет и количество выполненных заказов за определенный срок увеличится.

При освоении новой версии САПР зависимость скорости проектирования от времени можно наблюдать на рис. 1.

Стоит отметить, что при освоении каждой новой версии программного продукта время освоения минимально и спад скорости проектирования во время освоения незначительный.

Зависимость скорости проектирования и экономической эффективности проекта прямо пропорциональная.

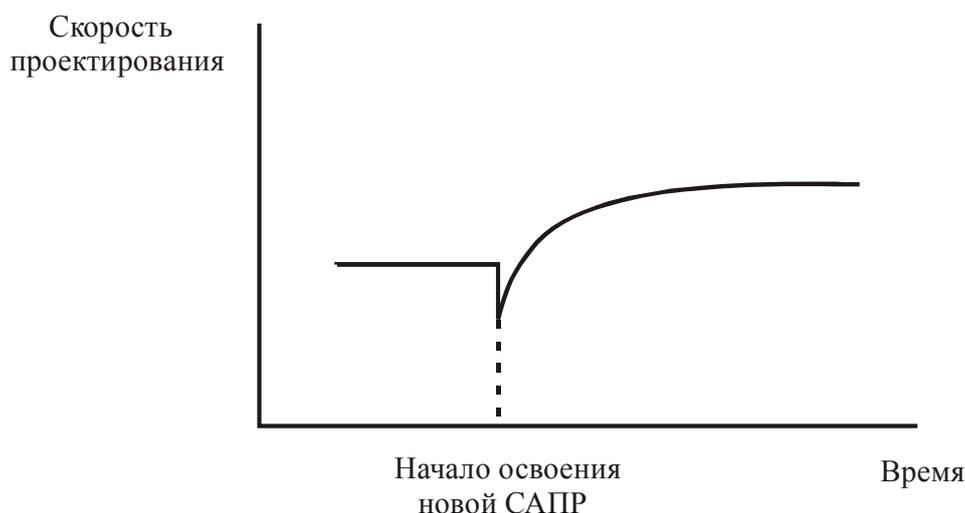


Рис. 1. Зависимость скорости проектирования от времени

Система P-CAD 4.5 работает в среде MS-DOS, а P-CAD-2001 – в Windows. P-CAD-2001 обладает следующими преимуществами перед версией P-CAD 4.5:

- удобный интерфейс;
- более удобное ведение библиотек компонентов;
- проверки на ошибки осуществляются непосредственно по окончании отдельного этапа проектирования, а в P-CAD 4.5 проверки были не на каждом этапе, вследствие чего было сложнее обнаружить ошибку;
- удобство разработки и редактирования на всех этапах;
- добавлены принципиально новые программные приложения, такие как:
 - а) графический редактор Relay, позволяющий предъявлять основные требования к плате со стороны разработчиков и системных конструкторов;
 - б) программный комплекс Design Flow, предназначенный для организации процесса проектирования;
 - в) средство просмотра чертежей схем и печатных плат через Интернет без возможности внесения в них изменения – View Center;
 - г) утилита анализа целостности сигналов;
 - д) некоторые утилиты DBX, позволяющие извлечь данные из файлов схем или плат, провести их обработку, выдать статистику, перекодировать в формат третьих фирм и т.п., например, утилита для контроля высоты компонентов, утилита для проверки зазоров между компонентами, утилита преобразования в формат DXF и др.;
- использование утилит стало более удобным за счет вызова их непосредственно из рабочей программы – графического редактора;
- оптимальная автотрассировка в системе SPECCTRA.

Система SPECCTRA представляет собой принципиально новый класс систем проектирования печатных плат [2]. В основе ее алгоритмов лежит так называемая бессеточная техника трассировки проводников, позволяющая задавать детальные правила для трассировки практически всех типов компонентов с учетом большого числа ограничений, при этом учитывая общие ограничения для всего проекта установок автотрассировки. За счет поддержки сложных иерархических правил проектирования улучшается технологичность изготовления платы. Описание точных геометрических моделей контактных площадок, трасс, дополнительных сквозных отверстий и других образов объектов в бессеточной базе данных позволяет максимально использовать доступное пространство, в результате чего возможно уменьшение количества слоев печатной платы. Бессеточная база данных занимает

гораздо меньше памяти, чем сеточная, используемая в P-CAD 4.5. С помощью команды Route/Fanout возможна трассировка коротких отрезков проводников для выравнивания их стартовых точек. Это позволяет разводить связи, идущие от компонентов, с малым шагом выводов, что в P-CAD 4.5 было невозможно.

Результаты автотрассировки двухслойной печатной платы контроля сигналов в различных версиях P-CAD представлены в таблице.

Параметры	P-CAD 4.5	P-CAD-2001
Количество переходных отверстий	180	142
Суммарная длина проводников	≈ 300 тыс.	≈ 270 тыс.
Образование "петель" проводников	20	отсутствуют
Площадь, занимаемая всеми	≈ 89000 мм ²	≈ 84000 мм ²
Процент автотрассировки	76%	100%

Таблица. Результаты автотрассировки

САПР P-CAD-2001 удачно вписывается в идеи CALS-технологий за счет:

- единого интерфейса,
- возможности преобразования файлов в стандартный формат данных – DXF и PDF,
- полной взаимосвязи данных, полученных на разных этапах проектирования,
- возможности коррекции на любом этапе проектирования с автоматическим обновлением необходимых данных в остальных документах,
- автоматизированного управления данными.

Таким образом, применение САПР P-CAD-2001 для проектирования печатных плат является наиболее актуальным за счет высокой скорости и качества трассировки.

Литература

1. Стешенко В.Б. ACCEL EDA. Технология проектирования печатных плат. М.: Нолидж, 2000.
2. Ёлшин Ю.М. Справочное руководство по работе с подсистемой SPECCTRA в P-CAD 2000. М.: СОЛОН-Р, 2002.

ИССЛЕДОВАНИЕ СТРУКТУР ТИПА $Si_{ПК}-SiO_2-Si$ С ОКИСЛАМИ, ПОЛУЧЕННЫМИ В РАЗЛИЧНЫХ РЕЖИМАХ ОКИСЛЕНИЯ

А.М. Семенов, А.М. Скворцов, Р.А. Халецкий

Как известно, замена алюминиевого затвора в МОП ИС на поликремневый приводит к ряду преимуществ:

- снижению пороговых напряжений;
- уменьшению длины канала за счет самосовмещения затвора;
- повышению стабильности МОП-структур;
- возможности формирования слоя коммутации элементов на основе поликремния.

В настоящей работе приводятся результаты исследования МОП-структур, полученных в различных режимах окисления. В качестве исследуемых объектов использовались МОП-структуры с поликремневым затвором и окислом, полученным при различных условиях окисления. Подложка МОП-структур представляет собой монокристаллический кремний n-типа с удельным сопротивлением $7.5 \text{ Ом}\cdot\text{см}$, ориентированный в плоскости (100). Перед термическим окислением образцы проходили очистку в перекисно-аммиачном растворе и смеси серной кислоты и перекиси водорода.

№ образца	Толщина SiO_2 , мкм	Температура окисления, °C	Время окисления, мин			Наличие добавки Cl
			На 1-ом этапе в сухом кислороде	На 2-ом этапе в парах воды	На 3-ем этапе в сухом кислороде	
1	0,12	950	10	40	10	Да
2		1000	10	18	10	Да
3		1050	-	80	-	Да
4		1150	-	35	-	Да

Таблица 1. Режимы термического окисления образцов

В табл. 1 представлены режимы термического окисления образцов. Образец №1 получен с использованием производственного варианта окисления. В производстве 590 серии КМОП ИМС при получении подзатворного окисла применяют комбинированный метод окисления (сухой-влажный-сухой). Здесь вместо влажного окисления, когда кислород пропускается через борботер с водой, увлажнение кислорода осуществляют следующим образом: вода попадает в смеситель через специальную форсунку, испаряется и поступает вместе с кислородом и хлором в зону реакции, где находятся подложки. Выбор временных режимов окисления производился из условия нормировки по заданному значению толщины окисла.

На рис. 1 представлены характерные высокочастотные вольт-фарадные характеристики полученных образцов. Из анализа полученных кривых следует, что для всех образцов характерна малая плотность поверхностных состояний на межфазной границе $Si-SiO_2$. Следовательно, при таком существенном различии в технологических режимах отсутствует отличие в электрофизическом состоянии границы кремний-окисел. Возможно, это связано с пока недостаточно изученной природой воздействия хлора на формируемую межфазную границу. Можно предположить, что ионы хлора, помимо взаимодействия с атомами щелочных металлов, производят пассивацию поверхности Si , уменьшая заряд на поверхностных состояниях.

Из рис.1 видно, что увеличение температуры до 1000°C для комбинированного окисления не приводит к заметному изменению вольт-фарадных характеристик (ВФХ). Изменение влажной среды окисления на сухую с одновременным увеличением времени окисления и температуры до 1000°C также не приводят к изменению ВФХ. Однако при увеличении температуры окисления в сухом кислороде до 1150°C происходит сдвиг кривой в направлении идеальной расчетной ВФХ. На основании этого факта можно заключить, что структуры с поликремневым затвором (образцы 1–3) имеют равные, достаточно большие фиксированные заряды в окисле, о чем свидетельствует положение соответствующих кривых относительно идеальной ВФХ.

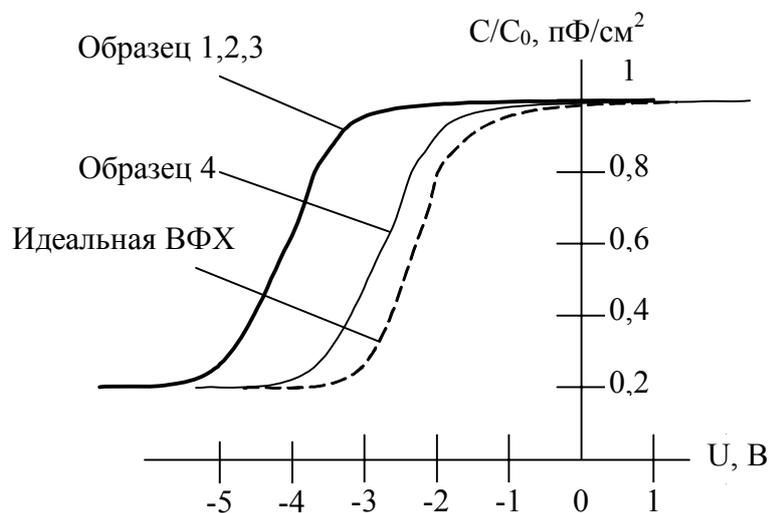


Рис. 1. Характерные экспериментальные вольт-фарадные характеристики образцов

При увеличении температуры сухого окисления окисел получается с меньшим встроенным зарядом. Это свидетельствует о том, что именно температура является преобладающим фактором, определяющим кинетику процесса окисления и параллельного формирования границы Si-SiO₂. Возможно, это связано с еще совсем не изученным поведением хлора в окисле и его влиянием на формирующуюся границу окисел-кремний [1].

Окисление в сухом кислороде при более высокой температуре дает лучшие результаты по сравнению с производственным окислением (кривая 4). Однако такое окисление может сопровождаться дополнительной разгонкой примеси в имплантированном слое, что может существенно сказаться на характеристиках готовой КМОП ИМС.

Для полного понимания данных процессов формирования окисла и межфазной границы Si-SiO₂ необходимы дополнительные исследования.

Литература

1. Санин К.В., Скворцов А.М. Технология БИС микропроцессоров и микро-ЭВМ. Учебное пособие. Л.: ЛИТМО, 1988.

ИССЛЕДОВАНИЕ МОП-СТРУКТУР С ОКИСЛАМИ, ПОЛУЧЕННЫМИ ПРИ РАЗЛИЧНЫХ РЕЖИМАХ ОКИСЛЕНИЯ

А.М. Семенов, А.М. Скворцов, Р.А. Халецкий

Термический оксид кремния представляет собой один из основных материалов, применяющихся в современной микроэлектронике. Он используется в качестве маски при проведении процессов диффузии, выполняет функции диэлектрика в МОП-структурах, изолятора в КСДИ-структурах и многослойной коммутации и др. Особенные жесткие требования к качеству предъявляются к термическому оксиду, применяемому в качестве подзатворного диэлектрика в МОП и КМОП интегральных микросхемах. При термическом окислении кремния одновременно происходит формирование межфазной границы Si-SiO₂. Именно электрофизические параметры окисла и границы интерфейса Si-SiO₂ являются главными факторами, определяющими работу и надежность функционирования КМОП ИМС.

Несмотря на большое количество научных работ, на сегодняшний день отсутствует единая установившаяся теория по физике Si-SiO₂, которая бы полностью объясняла все экспериментальные факты, зачастую противоречащие друг другу.

Как известно, процесс получения подзатворного окисла отличается своими режимами от процесса получения окисла над другими участками кремниевой подложки. Во-первых, подзатворный окисел часто получают в сухом кислороде. Во-вторых, при окислении вводят добавки хлора, чтобы исключить влияние подвижных ионов натрия и калия в окисле на работу КМОП ИМС [1]. Известно также, что использование влажного окисления не рекомендуется для подзатворного окисла из-за худшего, по сравнению с сухим окислением, качества окисла [2]. Несмотря на эти известные факты, применение влажного кислорода для получения подзатворного окисла имеет место в серийном производстве КМОП ИМС 590-ой серии на заводе “Светлана-Полупроводники”.

В настоящей работе приводятся результаты исследования МОП-структур на основе Al-SiO₂-Si, с окислом, полученном в различных режимах окисления. В качестве исследуемых объектов использовались МОП-структуры с алюминием в качестве затвора и окислом, полученным при различных условиях окисления. Подложка МОП-структур представляет собой монокристаллический кремний n-типа с удельным сопротивлением 7.5 Ом·см, ориентированный в плоскости (100). Перед термическим окислением образцы проходили очистку в перекисно-аммиачном растворе и смеси серной кислоты и перекиси водорода.

№ образца	Толщина SiO ₂ , мкм	Температура окисления, °С	Время окисления, мин			Наличие добавки Cl
			На 1-ом этапе в сухом кислороде	На 2-ом этапе во влажном кислороде	На 3-ем этапе в сухом кислороде	
1	0,12	950	10	40	10	Да
2		1000	10	18	10	Да
3		1150	-	35	-	Да

Таблица. Режимы термического окисления образцов

В таблице представлены режимы термического окисления образцов. Образец № 1 представляет собой производственный вариант исполнения. В производстве 590 серии КМОП ИМС используется комбинированный метод окисления при получении

подзатворного окисла (сухой-влажный-сухой). Здесь вместо влажного окисления, когда кислород пропускается через барботер с водой, увлажнение кислорода осуществляют следующим образом: вода попадает в смеситель через специальную форсунку, испаряется и поступает вместе с кислородом и хлором в зону реакции, где находятся подложки. Выбор временных режимов окисления производился из условия нормировки по заданному значению толщины окисла.

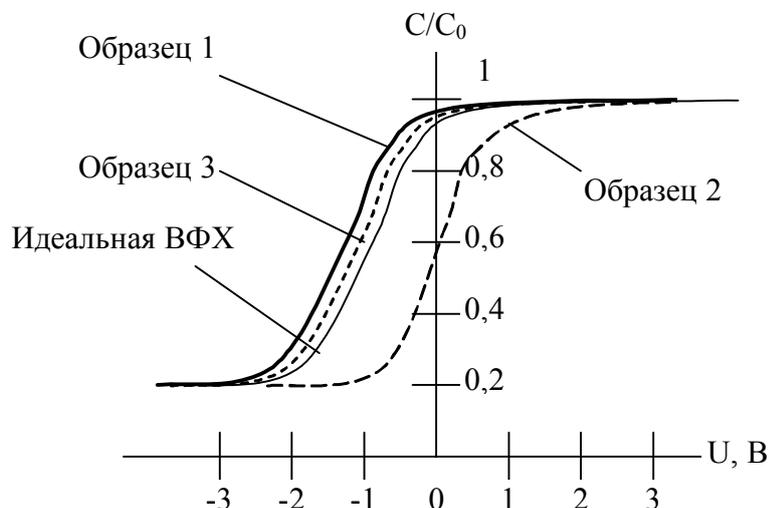


Рис. 1. Характерные экспериментальные вольт-фарадные характеристики образцов

На рис. 1 представлены характерные высокочастотные вольт-фарадные характеристики полученных образцов. Из анализа полученных кривых следует, что для всех образцов характерна малая плотность поверхностных состояний на межфазной границе Si-SiO₂. Следовательно, при таком резком отличии в технологических режимах отсутствует отличие в электрофизическом состоянии границы кремний-окисел. Возможно, это связано с пока неизученной природой воздействия хлора на формируемую межфазную границу. Можно предположить, что ионы хлора, помимо взаимодействия с атомами щелочных металлов, производят внутреннюю пассивацию приграничной области Si-SiO₂, уменьшая заряд на поверхностных состояниях.

При увеличении температуры от 950 °С до 1000 °С происходит сдвиг вольт-фарадной характеристики вправо на 0,8 В по оси напряжений, что эквивалентно избыточному фиксированному отрицательному заряду в окисле. При этом время окисления уменьшилось в два раза (см. таблицу). Можно предположить, что этот отрицательный заряд обусловлен тем, что с увеличением температуры увеличивается коэффициент диффузии гидроксильных -ОН и силанольных Si-OH групп, концентрация которых в условиях окисления в парах воды высока. Происходит увеличение потока этих групп к поверхности кремния, который начинает преобладать над скоростью химической реакции на границе Si-SiO₂. Непрореагировавшие ионы ОН- начинают взаимодействовать с окислом. Происходит формирование окисла с высокой концентрацией немостикового кислорода, которая обусловлена замещением атома кислорода с валентностью -2 в вершине тетраэдра SiO₄ на ион гидроксила с валентностью -1. Таким образом, избыточные гидроксильные и силанольные группы, а также немостиковый кислород в приповерхностной области обуславливают появление отрицательного заряда.

Окисление в сухом кислороде при более высокой температуре дает несколько лучшие результаты по сравнению с производственным окислением (рис. 1). Однако это может сопровождаться дополнительной разгонкой примеси в имплантированном слое, что может существенно сказаться на характеристиках готовой КМОП ИМС.

Таким образом, увеличение температуры при использовании комбинированного окисления существенно снижает качество окисла. Использование сухого окисления позволяет получить окисел с наименьшим значением фиксированного заряда, однако повышается температура окисления. Для полного понимания данных процессов формирования окисла и межфазной границы Si-SiO₂ необходимы дополнительные исследования.

Литература

1. Технология СБИС / Под ред. С. Зи. Т.1. 2. М.: Мир, 1986.
2. Майоров С.А., Скворцов А.М. Технология производства вычислительных машин. М.: Высшая школа, 1973.
3. Пичугин И.Г., Таиров Ю. . Технология полупроводниковых приборов. М.: Высшая школа, 1984.

ОЦЕНКА СУЩЕСТВУЮЩИХ МЕТОДОВ ОТРАБОТКИ И ПРОВЕРКИ КБО ГРАЖДАНСКИХ САМОЛЕТОВ

А.М. Рогачевский

Введение

ОКБ "Электроавтоматика", которое представляет автор, традиционно ведет работы по КБО летательных аппаратов различного назначения в самом широком смысле этого слова: разработка, испытания, поставка опытных образцов, сопровождение эксплуатации, модернизация.

Работы ведутся в соответствии с требованиями ТЗ, в которых в настоящее время значительное внимание уделяется таким вопросам, как финансово-временные затраты на различных этапах жизненного цикла КБО: стоимость и срок появления 1-ого опытного образца, затраты на этапе эксплуатации на проведение технического обслуживания и ремонтов, а также проведение модернизации КБО на протяжении более чем 30-летнего жизненного цикла самолета.

В соответствии с указанными требованиями формулируется следующий критерий разработки КБО: "Минимизация суммарной стоимости КБО, отнесенной ко всему его жизненному циклу с обеспечением возрастающих требований к его эффективности".

Основные способы реализации данного критерия представлены в табл. 1.

№	Этапы жизненного цикла КБО	Основные статьи финансово-временных затрат	Способы минимизации финансово-временных затрат
1	Приобретение	Разработка	Стандартизация и унификация аппаратных и программных модулей в целях использования базовых разработок
		Испытания (стендовые, наземные, летные)	Оптимальные способы и соотношения объемов проверок КБО в стендовых, наземных и летных условиях
		Поставка образца КБО	Унификация компонентной и элементной базы аппаратуры КБО
2	Эксплуатация	Техническое обслуживание и ремонт	Отказоустойчивость КБО, реализуемая путем информационной и аппаратурной избыточности. Унификация компонентной и элементной базы аппаратуры КБО
		Модернизация	Открытая архитектура КБО, реализуемая путем избыточной вычислительной мощности и пропускной способности интерфейсов, стандартизацией и унификацией аппаратных и программных модулей. Загружаемое ПО.

Таблица 1. Критерий разработки КБО - минимизация суммарной стоимости КБО, отнесенной ко всему его жизненному циклу с обеспечением возрастающих требований к его эффективности

Существует два этапа жизненного цикла КБО: приобретение (с подэтапами разработки, испытаний и изготовления опытного образца) и эксплуатация (с подэтапами технического обслуживания, ремонта и модернизации).

Финансово-временные затраты подэтапа разработки могут быть уменьшены стандартизацией и унификацией аппаратных и программных модулей в целях возможности использования базовых разработок, подэтапа испытаний – оптимизацией способов и соотношений объемов проверок в стендовых, наземных и летных условиях.

Стоимость образца КБО может быть уменьшена унификацией компонентной и элементной базы аппаратуры КБО.

На подэтапе технической эксплуатации и ремонта основополагающим является свойство отказоустойчивости КБО, реализуемое путем информационной и аппаратурной избыточности, что позволяет достоверно определить его текущее состояние и реконфигурировать, при необходимости, архитектуру. Важным является также унификация и быстросъемность аппаратуры КБО.

На подэтапе модернизации основополагающим является открытая архитектура КБО, реализованная путем избыточной вычислительной мощности и пропускной способности интерфейсов, реализация загружаемого программного обеспечения, а также унификацией аппаратных и программных модулей.

Назначение и состав КБО

В зависимости от назначения гражданских самолетов (транспортные, грузовые, конвертируемые транспортно-грузовые, для исследования земной поверхности, воздействия на атмосферу, самолеты-салоны и т.д.) требования к КБО также различны. Это либо навигационный комплекс, либо навигационный комплекс с решением дополнительных задач (индикация навигационных параметров; управление общесамолетным оборудованием, индикация и контроль параметров общесамолетного оборудования; выход в точку сброса и управление сбросом грузов; управление аппаратурой воздействия на атмосферу; управление аппаратурой исследования земной поверхности и т.д.).

Однако основное назначение КБО во всех случаях неизменно:

- автоматизированное и ручное самолетовождение с оптимизацией режимов полета по внутренним и зарубежным трассам гражданской авиации;
- управление радиотехническими системами (РТС) в автоматизированном и ручном режимах;
- отображение экипажу пилотажной, навигационной и обзорной информации, а также справочной информации, информации о функционировании КБО, его отказах и неисправностях, рекомендации по парированию отказов и неисправностей;
- контроль технического состояния КБО и, при необходимости, его реконфигурация, а также регистрации, хранение и выдача результатов контроля при техническом обслуживании самолета.

В соответствии с назначением КБО может включать в себя ряд подсистем:

- "Ядро", состоящее из информационно-вычислительной системы (ИВС) и информационно-управляющего поля (ИУП). ИВС организована на базе высокопроизводительных вычислителей, состоящих из унифицированных конструктивных функциональных модулей (КФМ). ИУП комплекса является одной из наиболее важных его частей, так как оно обеспечивает связь пилота с КБО и летательным аппаратом в целом. В состав ИУП входят индикаторы и органы управления. В качестве индикаторов могут быть использованы, например, многофункциональные цветные индикаторы (МФЦИ) с жидкокристаллической панелью, электромеханические индикаторы, индикационная панель пультов управления и индикации (ПУИ). Управляющая часть ИУП может включать клавиатуру ПУИ, кнопочное обрамление МФЦИ и т.д.;
- навигационно-пилотажное оборудование, которое может включать инерциальную систему (ИС), курсовертикаль (КВ), систему воздушной сигналов (СВС) и т.д.;
- радиотехническое оборудование навигации и посадки, которое может включать радиотехническую систему навигации и посадки (РСБН), спутниковую навигационную систему (СНС), доплеровский измеритель скорости и угла сноса

(ДИСС), автоматический радиокompас (АРК), радиовысотомер (РВ), самолетный ответчик (СО) и т.д.;

- систему автоматического управления (САУ);
- общесамолетное оборудование (ОСО);
- радиосвязное оборудование (РС) и т.д.

Часть указанных систем может входить в состав КБО, часть – являться самолетным оборудованием и взаимодействовать с КБО системами.

Вопрос выбора состава КБО определяется не только техническими причинами.

Поколения цифровых КБО ОКБ "Электроавтоматика"

Поколения цифровых КБО ОКБ "Электроавтоматика" представлены в табл. 2.

Поколение КБО	Наименование КБО	Суть КБО		Контроль КБО			
		Назначение	Состав	Автономный входной контроль систем КБО	Стендовый контроль КБО		
					Разрабатываемые имитаторы с аналог. и цифр. интерфейсом	Серийные имитаторы с цифр. интерфейсом	Имитаторы с цифровым и аналоговым интерфейсом на базе ПК
I	ОЛЬХА	НК	Полный	КПА систем	+	+	—
	ПИЖМА	НК					
	ТОПОЛЬ	НК + спец.					
II	МАЛЬВА	НК+ спец.					
	МАК						
	ИРИС						
	ЦИКЛОН						
	ЧЕРНИКА						
III	ЖАСМИН	НК					
	ВСС-85						
IV	ССИ-80	НК + СЕИ + СУОСО	Ядро	ВСК систем	+	—	+
	КВСС-74	НК + СЕИ			—		
	КВСС-334	НК + СЕИ					

Таблица 2. Обзор методов входного контроля и стендовой отработки КБО

К первому поколению могут быть отнесены КБО, появившиеся в начале 70-х годов, установленные на самолетах Як-42, Ил-86 и Ан-72. Они базировались на 16-разрядном бортовом вычислителе типа БЦВМ20, имеющем быстродействие порядка 200 000 оп/с на операциях типа "сложение", емкость ОЗУ – 2 Кб, ПЗУ – 64 Кб.

Ко второму поколению относятся КБО для самолетов Ан-74 и его модификаций, ТУ-134СХ, Ил-24Н и т.д., ядром которых является доработанный 32-разрядный бортовой вычислитель БЦВМ20.

КБО третьего поколения базировались на БЦВМ80 производительностью 800 тыс. оп/с на операциях типа R-R, с емкостью ОЗУ – 32 Кб, ПЗУ – 256 Кб. Они установлены на самолетах ТУ-154М, ТУ-204, Ил-96.

К четвертому поколению относятся КБО для самолетов С-80, Ан-74ТК-300, ТУ-334, находящихся в настоящее время на этапе летных испытаний. Ядро КБО включает ИВС (бортовой 32-разрядный вычислитель БЦВМ90 с производительностью 25 млн. оп/с на операциях типа R-R, емкость ОЗУ – 1 Мб, ПЗУ –

4 МБ и развитый интерфейс) и ИУП (пульт ПУИ с экраном 3"×4" и клавиатурой, а также индикаторы МФЦИ с размером экрана 6"×8" с кнопочным обрамлением). Для самолета ТУ-334 ядром КБО является пульт-вычислитель (ПВ), объединяющий функции высокопроизводительного вычислителя и пульта управления.

Входной контроль систем КБО

Отработка КБО на базе предприятия-разработчика представлена в табл. 2 и проводится в два этапа: входной контроль составных частей КБО и комплексная отработка КБО на стенде.

Входной контроль составных частей КБО первого, второго и третьего поколения проводился с помощью контрольно-проверочной аппаратуры (КПА).

КПА разрабатывалась, изготавливалась и поставлялась совместно с поставкой входящих в КБО систем для их автономной проверки.

Для КБО четвертого поколения КПА в основном отсутствует, поскольку развитая система встроенных средств контроля (ВСК), обеспечивающая полноту контроля не менее 0,95, позволяет отказаться от специальной КПА при проведении входного контроля систем и далее на этапе эксплуатации.

Требование проведения подготовки к полетам и всех видов обслуживания КБО в процессе эксплуатации без применения КПА присутствует в ТЗ на КБО четвертого поколения.

Стендовая отработка КБО

Стенд для отработки КБО включает в свой состав жгутовую самолетную схему (фидер), технологические рамы для установки входящих в КБО систем, набор имитаторов взаимодействующих с КБО систем и дополнительное оборудование (стойки, приборная доска, система электропитания, система обдува и т.д.).

КБО первого, второго и частично третьего поколений включало в свой состав не только "ядро", но и широкий набор систем, в связи с чем стенды их отработки требуют установки всего набора входящих в КБО систем, используя имитаторы (как правило, электромеханические аналоговые и громоздкие цифровые) только взаимодействующего с КБО оборудования.

В качестве примера серийно изготавливаемого имитатора с цифровым интерфейсом, используемого при отработке КБО, может быть приведен задатчик кодовой информации (ЗКИ).

ЗКИ обеспечивает:

- задание и выдачу в КБО информационных слов 32-разрядным биполярным последовательным кодом в соответствии с ГОСТ 18977-79 и РТМ 1495-75 (обеспечивается от 1 до 8 слов в цикле с выдачей в асинхронном режиме);
- индикацию выбранного слова в 32-разрядном двоичном коде.

ЗКИ имеет следующие характеристики:

- выдача последовательного кода – с частотой 50 Кбит/с;
- электропитание ~ 200 В, 400 Гц;
- масса – 15 кг.

Такой подход являлся неудовлетворительным, поскольку требовал существенных финансовых и временных средств на доставку входящих в КБО систем, их распаковку, входной контроль, отработку, упаковку и т.д. Поэтому впоследствии в связи с рядом причин (развитие средств вычислительной техники, ограничения экономического и временного характера и т.д.) подход к определению состава и

отработке КБО был изменен. В состав КБО стали включать только "ядро", состоящее из ИВС и ИУП.

В качестве имитаторов взаимодействующего с КБО оборудования начали использовать персональные компьютеры с установленными модулями сопряжения (шина – внешняя среда) и программами, реализующими статические модели взаимодействующих систем.

Так, например, используются модули сопряжения:

- МД01-24 (с 2-мя выходными и 6-ью входными каналами 32-разрядного биполярного последовательного кода, а также с 4-мя входными и 4-мя выходными разовыми командами обрыв/корпус);
- МД03 (с 6-ью выходными каналами 32-разрядного биполярного последовательного кода),

реализующие цифровой интерфейс в соответствии с ГОСТ 18977-79.

В связи с небольшим количеством свободных слотов в ПК начали использоваться специальные контейнеры, представляющие собой промышленные ПК, допускающие установку до 20 модулей сопряжения. Подобное использование позволяет минимизировать весогабаритные и стоимостные характеристики имитаторов взаимодействующего с КБО оборудования. Исчезает также необходимость реализации локальной сети для централизованного управления системой моделей.

Так, например, используются модули сопряжения:

- PCL-726 (с 6 выходными каналами постоянного или переменного тока, 16 выходными и 16 входными каналами TTL уровня);
- PCL-727 (с 12 выходными каналами постоянного тока, 16 выходными и 16 входными каналами TTL уровня)

и прочие, реализующие цифровой интерфейс в соответствии с ГОСТ 18977–79, а также различные аналоговыми интерфейсами.

Заключение

Анализируя рассмотренные выше способы отработки и проверки КБО гражданских самолетов на базе предприятий-разработчиков, можно предложить следующую технологию проведения данной работы:

- на этапе написания ТЗ на взаимодействующее с КБО оборудование представлять требование его проверки с помощью средств ВСК;
- входной контроль взаимодействующих систем проводить с использованием ВСК;
- при отработке КБО на стенде использовать в качестве имитаторов взаимодействующих систем промышленный ПК с определенными типами и количеством модулей сопряжения, позволяющими реализовать необходимые аналоговые и цифровые интерфейсы. Возможно также использование стандартных ПК с модулями сопряжения, объединенных в локальную вычислительную сеть;
- в качестве моделей взаимодействующих систем использовать математические статические модели, реализующие несколько вариантов наборов входных-выходных параметров каждой системы.

Для повышения качества отработки можно считать перспективным развитие описанной выше технологии отработки КБО, заключающееся в реализации математических динамических моделей взаимодействующих систем, самолета, внешней среды.

ПРОБЛЕМЫ НАДЕЖНОСТИ МАГНИТОУПРАВЛЯЕМЫХ КОНТАКТОВ В СИСТЕМАХ УПРАВЛЕНИЯ

В.Л. Ткалич, Н.В. Жуков

Несмотря на то, что на сегодня предложено множество решений, посвященных повышению срока службы и эффективности устройств автоматики на основе различных упругих элементов, задача повышения надежности таких систем приобретает все большую актуальность. Такие упругие элементы входят в конструкцию большинства известных устройств, приборов и механизмов, и они являются их незаменимыми функциональными частями.

Разработан новый подход к исследованию упругих чувствительных элементов (УЧЭ) герконов, основанный на применении абелевых функций, что позволяет достичь наиболее правильного аналитического описания как самих УЧЭ, так и динамических процессов происходящих в них.

Найдем амплитудно-частотную характеристику (АЧХ) для нелинейного дифференциального уравнения, описывающего динамику плоских УЧЭ. Для этого воспользуемся методом комплексных амплитуд [2]. Положим, что возмущающая сила меняется по закону $F(t) = F_0 \cos \omega t$, и приведем уравнение

$$A \frac{d^2 \varphi}{dt^2} + B_1 \frac{d\varphi}{dt} + B_2 \frac{d\varphi}{dt} \left| \frac{d\varphi}{dt} \right| + C \sin \varphi = F(t)$$

к виду

$$\ddot{y} + 2 \frac{n}{m_\Sigma} \dot{y} + \frac{\mu}{m_\Sigma} \dot{y} |\dot{y}| + p_k^2 y = (F_0 / m_\Sigma) \cos \omega t. \quad (1.1)$$

Введем в рассмотрение комплексную величину y^* , действительная часть которой совпадает с выражением для смещения $\text{Re} y^* = y$. Зависимость возмущающей силы от времени также представим в комплексной форме: $F^*(t) = F_0 \exp(i\omega t)$, так что

$$F(t) = \text{Re } F^*(t) = F_0 \cos \omega t.$$

Действительная часть решения уравнения

$$\ddot{y} + 2 \frac{n}{m_\Sigma} \dot{y} + \frac{\mu}{m_\Sigma} \dot{y} |\dot{y}| + p_k^2 y = (F_0 / m_\Sigma) e^{i\omega t}$$

совпадает с решением уравнения (1.1), так как коэффициенты уравнения являются действительными величинами. Искомое решение запишем в виде

$$y^* = A^* \exp(i\omega t). \quad (1.2)$$

Соответственно,

$$\dot{y}^* = A^* \omega i e^{i\omega t};$$

$$\ddot{y}^* = -A^* \omega^2 e^{i\omega t}; \quad (1.3)$$

$$|\dot{y}^*| = |A^* \omega i e^{i\omega t}| = A^* \omega.$$

Подставив (1.2) в (1.3), получим:

$$A^* (-\omega^2 + 2 \frac{n}{m_\Sigma} \omega i + \frac{\mu}{m_\Sigma} A^* \omega^2 i + p_k^2) = F_0 / m_\Sigma, \quad (1.4)$$

откуда определяется комплексная амплитуда:

$$A^*(\omega) = \frac{F_0}{m_\Sigma \left[p_k^2 - \omega^2 + i \left(2 \frac{n}{m_\Sigma} \omega + \frac{\mu}{m_\Sigma} A^* \omega^2 \right) \right]} \quad (1.5)$$

или, в показательной форме,

$$A_* = A_* \exp(i\omega t),$$

где

$$A(\omega) = \frac{F_0}{m_\Sigma \sqrt{(p_k^2 - \omega^2)^2 + (2\frac{n}{m_\Sigma}\omega + \frac{\mu}{m_\Sigma}A_*\omega^2)^2}}, \quad (1.6)$$

$$\phi = \arctg \frac{2(n/m_\Sigma)\omega + (\mu/m_\Sigma)A_*\omega^2}{p_k^2 - \omega^2}. \quad (1.7)$$

Формулы (1.6)–(1.7) являются искомыми в расчете.

Для построения графиков АЧХ герконов различных типов разработан пакет прикладных программ (ППП) на языке программирования Visual C++ 6.0 Enterprise Edition. При расчете функции $A(\omega, p_k)$ использовался метод половинного деления [1, 3]. Начальными данными в алгоритме расчета АЧХ являются коэффициенты уравнения и требуемый уровень точности вычислений. Разработанные математические модели для АЧХ плоских пружин герконов позволяют выявить влияние конструктивных параметров магнитоуправляемых контактов (МК) на их АЧХ, что очень важно при разработке инженерной методики подбора коэффициента демпфирования данных устройств в зависимости от рабочего диапазона.

Выявлено новое определение инерционных свойств жидкости, выражающееся через коэффициенты присоединенных масс. Знание этих присоединенных масс и моментов инерции очень важно при исследовании жидкости и тела в нестационарном взаимодействии. Влияние нелинейной составляющей жидкостного трения отражается на частотных характеристиках измерительных устройств, работающих в широком частотном и амплитудном диапазонах. К числу таких устройств можно отнести газонаполненные и ртутносмачиваемые герконы. При получении ММ важно учитывать широкий спектр факторов – ориентацию УЧЭ в пространстве, показатели обтекаемости элемента и его геометрию, влияние твердых стенок, влияние отрыва жидкости от поверхности.

Анализ результатов большого числа исследований поверхностей контактных сердечников (КС) с нерегулярными микрорельефами, имеющими случайный стационарный характер, показал, что для улучшения ряда эксплуатационных параметров необходимо добиваться улучшения качества рабочих поверхностей УЧЭ путем регуляризации их микрогеометрии. Способ вибронакатывания позволяет получить оптимальную регулярную микрогеометрию и приповерхностный слой материала УЧЭ с высоким коэффициентом упрочнения.

Анализ сущности и особенностей регулярного микрорельефа (РМР) свидетельствует о том, что практически все недостатки, свойственные шероховатым поверхностям деталей УЧЭ, при создании РМР исключаются.

Литература

1. Андреева Л.Е. Упругие элементы приборов. М.: Машиностроение, 1981. 455 с.
2. Бидерман В.А. Прикладная теория механических колебаний. М.: Высшая школа, 1972. 416 с.
3. Ткалич В.Л., Лобанцев А.В., Галин Ю.В. Построение уточненного аналитического выражения АЧХ газонаполненных герконов. // Тезисы доклада на Всероссийской научной конференции (Computer-Based Conference) "Методы и средства измерений", раздел 11 "Математические модели и численное моделирование измерительных приборов и датчиков". Нижний Новгород, 1 мая 2000 г. С.17.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА МОНИТОРИНГА ЭЛЕКТРОМАГНИТНЫХ ПОЛЕЙ

А.Г. Коробейников, М.Ю. Тимошенкова

Сегодня почти каждый человек осознает, что загрязнение окружающей среды достигло огромных размеров и катастрофически прогрессирует. Экологическая опасность вызвана ускоренным развитием различных отраслей промышленности. Обычно основное внимание уделяется загрязнению окружающей среды веществами. Однако в последнее время появились указания на то, что загрязнение среды различного рода физическими полями, в частности, электромагнитным, оказывает не менее существенное влияние на живые организмы, в том числе на человека.

Организму человека свойственны определенные ритмы функционирования – так называемые биоритмы. Их существование определяется работой различных систем организма человека, таких как нервная, кровеносная и другие, с физической точки зрения представляющих собой резонансные системы с высокой электропроводностью, т.е. электрические резонансные контура.

Воздействие на организм внешних переменных магнитных полей на резонансных частотах (частотах биоритмов) приводит к нарушению его нормального функционирования. Люди, живущие в крупных промышленных центрах, практически постоянно находятся под воздействием низкочастотных магнитных полей, создаваемых линиями электропередач как переменного (50 Гц), так и постоянного (линии питания электропоездов, трамваев, метро) тока, мощными промышленными и бытовыми электродвигателями и электроустановками.

Первоначально основное внимание уделялось полям на частоте 50 Гц (60 Гц в США). Гораздо меньше изучены переменные магнитные поля ультранизкочастотного (УНЧ) диапазона 0,001–10 Гц, хотя именно они представляют наибольший интерес по двум причинам:

- магнитная составляющая более опасна, чем электрическая;
- УНЧ магнитная окружающая среда имеет особое биологическое значение, так как соответствует основным физиологическим ритмам организма человека.

Магнитные поля от электротранспорта отличаются от полей, производимых линиями электропередач, большей сложностью, наличием резких скачков интенсивности и компонент в широком диапазоне частот, преимущественно УНЧ. Магнитные поля от электро тяги, в отличие от магнитных полей линий электропередач, до последнего времени были мало изучены, так как не существовало специальной измерительной аппаратуры, способной производить непрерывную запись информации и ее хранение. Сейчас такие приборы существуют, например оригинальный трехкомпонентный магнитометр MVC-3DS.

В настоящее время обнаружены УНЧ электромагнитные литосферные эмиссии, регистрируемые на различных стадиях развития очагов сильных землетрясений, в диапазоне частот 0,01–10 Гц. УНЧ электромагнитные литосферные эмиссии достоверно выделяются на фоне геомагнитных пульсаций и техногенных помех. Как правило, первые всплески УНЧ электромагнитных литосферных эмиссий регистрируются за несколько недель до мощного толчка, а за несколько часов до него наблюдается резкое увеличение интенсивности УНЧ электромагнитных литосферных эмиссий.

Биологические предвестники землетрясений давно и хорошо известны. Рыбы, земноводные, птицы и млекопитающие необычно ведут себя: уходит от эпицентра землетрясения, спасая свою жизнь за дни-часы до сильного толчка. Изменяются показатели системы крови у человека. Однако природа биопредвестников до сих пор неизвестна. Существует ряд гипотез, в частности, предположение об электромагнитной

природе агента, воздействующего на живые организмы перед землетрясением. Был проведен ряд экспериментов, в которых показана чувствительность биологических систем к УНЧ электромагнитным полям.

Ультранизкочастотные литосферные эмиссии (УЛЭ), наблюдаемые перед сильными землетрясениями, могут воздействовать на живые организмы, вызывая стрессовую коллективную реакцию популяций или отдельных особей. При этом информационный сигнал может быть закодирован в спектральных и пространственно-временных характеристиках УЛЭ. Поэтому в экспериментах с искусственными УНЧ электромагнитными полями наиболее эффективно проводить воздействие на биоорганизмы модельным сигналом, максимально приближенным к естественному.

УНЧ электромагнитные поля являются биологически активными, и их спектрально-временные характеристики могут быть экологически значимыми для живых организмов. В стрессовых ситуациях действует комплекс причин, но можно предполагать, что как во время магнитной бури, так и перед землетрясением УНЧ электромагнитный сигнал, его спектрально-временная характеристика является носителем информации, воспринимаемой биосистемой.

Таким образом, не исключено, что одной из основных причин, вызывающих необычное поведение живых организмов перед сильными землетрясениями, являются ультранизкочастотные электромагнитные литосферные сигналы, источник которых расположен в районе очаговой зоны готовящегося землетрясения.

Измерения электромагнитных излучений при помощи MVC-3DS проводится для решения многих задач, например:

- оперативного контроля и мониторинга техногенной электромагнитной обстановки в различных районах;
- оперативного контроля появления магнитных бурь;
- проведения поиска так называемых "магнито-спокойных мест";
- проведения геологической разведки;
- предсказания землетрясений.

Для решения каждой задачи необходимо свое программное обеспечение (ПО). Например, при проектировании ПО для мониторинга и цифровой регистрации переменных магнитных, электрических и сейсмических полей необходимо решение следующих задач:

- создание многооконного интерфейса;
- мониторинг в реальном масштабе времени и сбор данных в цифровом виде на жесткий диск IBM-совместимого компьютера или на любой другой накопитель информации (например, магнито-оптический);
- управление магнитометром MVC-2DS;
- графическое отображение на экране монитора измеряемых сигналов в реальном времени;
- запись измеренных сигналов на жесткий диск РС в бинарном формате в реальном масштабе времени;
- конвертирование данных измерений из бинарного формата в ASCII формат.
- автоматическое масштабирование входных сигналов;
- цифровой контроль выходных сигналов магнитных и электрических датчиков;
- продолжительная (в пределах 72 часов) запись данных в файл;
- экспорт ASCII файлов.

Созданное ПО должно функционировать совместно с оборудованием, удовлетворяющим следующим техническим характеристикам:

- разрядность аналого-цифровых преобразователей (АЦП) 24 бита;
- частота дискретизации АЦП 50 отсчетов в секунду (50 Гц);

- коэффициент подавления сигнала встроенным в АЦП цифровым рекурсивным режекторным фильтром на частоте дискретизации АЦП 200 дБ;
- измеряемые величины – три взаимно-ортогональные компоненты магнитного поля (H , D и Z) и любые три компоненты электрического (теллурического) поля;
- тип датчиков магнитных компонент – магнитостатический (*torsion*);
- коэффициент подавления сигнала на частоте 50 Гц аналоговыми фильтрами 30 дБ;
- диапазон измеряемых частот 0–10 Гц;
- диапазон измерений каждого магнитного датчика (± 10 В по выходному напряжению) ± 5000 нТл $\pm 5\%$;
- интегральный уровень шумов для каждой магнитной компоненты (*peak to peak*) в частотном диапазоне 0–10 Гц (центральная частота 1 Гц) с 99% интервалом вероятности не более 0.01 нТл;
- постоянная внутренних калибровочных колец каждого из трех магнитных (H , D и Z) датчиков 170 ± 10 нТл/мА;
- тип датчиков электрических компонент – любые с выходным сопротивлением менее 100 КОм (в данной версии – заземленные электроды, разнесенные на ~ 100 м);
- диапазон измерений для каждой электрической компоненты ± 2.5 В;
- интегральный уровень шумов для каждой электрической компоненты (*peak to peak*) в частотном диапазоне 0–10 Гц (центральная частота 1 Гц) с 99% интервалом вероятности 10 мкВ.

Разработанное ПО использовалось в составе:

интегрированного комплекса для исследования особенностей распространения гидромагнитных волн вдоль земной поверхности и проведения магнитно-теллурического зондирования в субавроральной зоне (п. Лехта);

систем сбора и обработки информации в интегрированном комплексе для мониторинга геомагнитного поля в сейсмоактивных районах Грузии (г. Душети).

ИССЛЕДОВАНИЕ СТРУКТУРНЫХ ДЕФЕКТОВ МЕЖСЛОЙНОЙ ИЗОЛЯЦИИ В КМОП ИС

С.Н. Новиков, А.Е. Семенов, А.М. Скворцов, Е.Г. Фролкова, Р.А. Халецкий

В настоящей работе приведены результаты исследований пористости слоев окислов кремния, полученных термическим окислением поликристаллического кремния ($Si_{п/к}$), осаждаемого на планарную структуру.

Структура содержала изолированные области со сформированными в них областями стоков и истоков n- и p-канальных МОП-транзисторов и подзатворный слой SiO_2 . Осаждение слоя $Si_{п/к}$ производилось газо-фазным методом на установке УНЭС. Толщина слоя составляла 0,5 мкм. В серийной технологии КПОП ИС этот слой используется для формирования затворов МОП-транзисторов и первого слоя коммутации элементов микросхемы. Для снижения сопротивления коммутирующего слоя после нанесения поликремний легируется диффузией фосфора. Далее следует операция травливания фосфорно-силикатного стекла (ФСС) и процесс фотолитографии в поликремнии (формирование затворов и шин коммутации).

В производстве КМОП ИС серии 590 между поликремниевым и алюминиевым слоями коммутации в качестве межслойной изоляции на установке типа "Оксин" в среде аргона газо-фазным методом осаждается слой ФСС толщиной 0,5–0,6 мкм. Эта операция достаточно сложна и не всегда обеспечивает стабильные и повторяемые результаты. Поэтому конечная цель настоящего исследования – определение возможности замены межслойной изоляции на основе ФСС слоем SiO_2 , полученным путем термического окисления поверхности коммутирующего слоя поликремния.

Из приведенных в обзоре [1] методов исследования пористости слоев диэлектриков авторами был выбран метод электрографии по следующим причинам. Он достаточно информативен и нагляден, прост в реализации, а для обработки результатов может быть использована компьютерная графика, существенно упрощающая и ускоряющая подсчет плотности дефектов окисла. Наконец, именно этот метод используется в производстве КМОП ИС на предприятии "Светлана-Полупроводники".

Сущность этого метода заключается в получении картины распределения дефектов на фотобумаге, смоченной раствором гидрохинона и прижатой к подложке, вследствие выделения зародышей серебра в местах протекания тока через дефект. Разновидности метода электрографии и подробное их описание приводится в работах [2–4]. Отличие методики, примененной в настоящей работе, заключается в том, что в площадь электрограммы с помощью компьютерной графики вписывается максимально возможное число квадратов площадью в 1 см^2 . По результатам подсчета пор в каждом квадрате рассчитывается средняя плотность пор на подложке.

На рис.1 показаны структуры двух видов подложек, отличающихся методами их формирования, а в таблице – режимы получения этих структур.

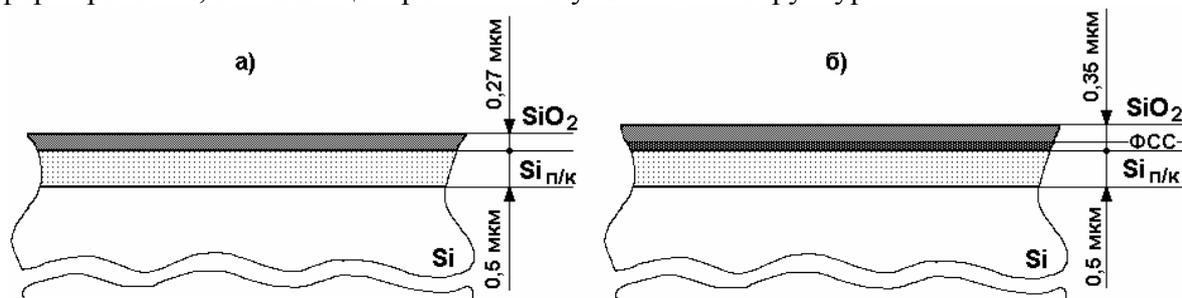


Рис.1. Структура исследованных образцов: а - образец №1; б - образец №2

Для изготовления указанных образцов использовался кремний монокристаллический марки КЭФ7,5(100), используемый для получения контрольных тестовых образцов при изготовлении ИС серии 590. Подготовка подложек и изготовление экспериментальных образцов производились на предприятии "Светлана-Полупроводники" по действующей технологии 590 серии. Вводимое в существующий процесс окисление $Si_{п/к}$ производилось в одном случае после легирования $Si_{п/к}$ фосфором, т.е. при сохранении на его поверхности ФСС, во втором случае – после стравливания ФСС.

Вид образца	Основные технологич. параметры
$Si_{п/к+P}$ затем ФСС стравлено (образец №1)	$h=0,5$ мкм+ легир.Р при $t^{\circ}=860$ °С
$Si_{п/к+P}$ (образец №2)	$h=0,5$ мкм + легир.Р при $t^{\circ}=860$ °С
SiO_2 (образец №1)	$h=0,27$ мкм, $t^{\circ}=1000$ °С, вл. O_2+HCl
SiO_2 (образец №2)	$h=0,35$ мкм, $t^{\circ}=1000$ °С, вл. O_2+HCl

Таблица. Основные режимы изготовления образцов

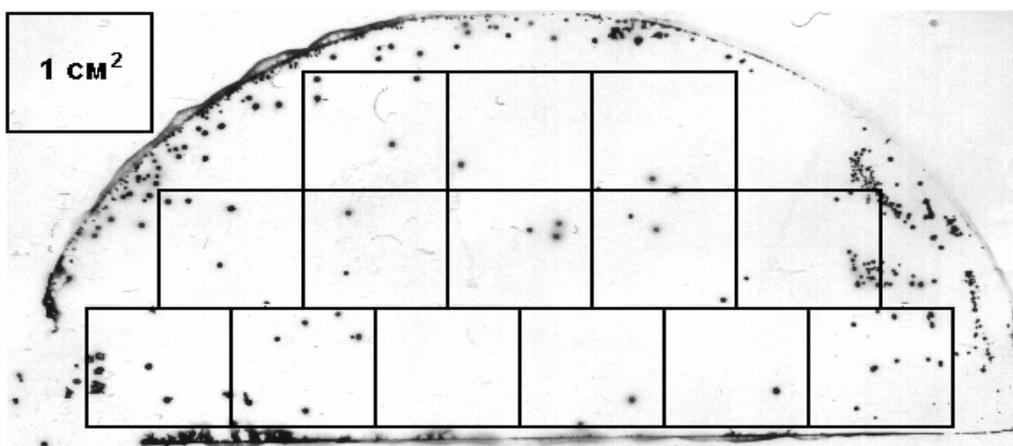


Рис. 2. Электрограмма образца № 1

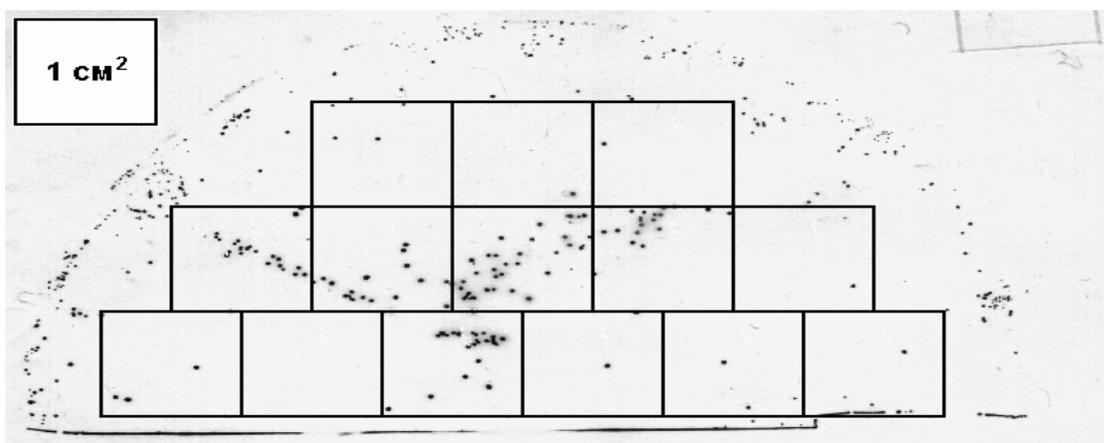


Рис. 3. Электрограмма образца № 2

На рис. 2 и рис. 3 приведены электрограммы образцов с нанесенными на них измерительными квадратами. Расчет среднего количества пор ("проколов") дал следующие результаты. На образце № 1 с окислением $Si_{п/к}$ после снятия ФСС число пор приближается к 5 шт/см². Средняя плотность пор на образце с окислением $Si_{п/к}$, покрытого слоем ФСС (образец № 2), превышает 8 пор/см². Размеры пор на образце № 2 меньше, чем на образце № 1, и лежат в пределах до 5 и до 120 мкм, соответственно.

Таким образом, меньшая дефектность термического окисла кремния обеспечивается при окислении поликристаллического кремния свободного от фосфоросиликатного стекла. Следовательно, такой окисел предпочтительнее для межслойной изоляции шин коммутации. Однако для окончательного решения необходимо провести аналогичные исследования повторно, а также исследовать электрическую прочность полученных окислов.

Литература

1. Концевой Ю.А., Филатов Д.К. Дефекты кремниевых структур и приборов. Часть 2.- М: ЦНИИ "Электроника", 1987. 68 с.
2. Носиков С.В., Пресс Ф.П. Электрографический метод исследования слоев двуокиси кремния и нитрида кремния. // Электронная промышленность. 1971. Вып.1. С. 69–73.
3. Носиков С.В., Иерусалимчик И.Г. Электрохимические процессы в электрографической системе с бромсеребряной бумагой. // Электронная техника. Сер. 2. Полупроводниковые приборы. 1972. Вып.1. С.125–129.
4. Вайнштейн А.Б., Власова Г.В., Королева Г.И. и др. Исследование сплошности тонких диэлектрических пленок методом электрографии. // Электронная техника. Сер. 3. Микроэлектроника. 1974. Вып.2. С. 57–61.

ИССЛЕДОВАНИЕ ПРОБИВНЫХ НАПРЯЖЕНИЙ МЕЖСЛОЙНОЙ ИЗОЛЯЦИИ В КМОП ИС

С.Н. Новиков, А.Е Семенов, А.М. Скворцов, Е.Г. Фролкова, Р.А. Халецкий

В настоящей работе приведены результаты исследований электрической прочности слоев окислов кремния, полученных термическим окислением поликристаллического кремния ($Si_{п/к}$), осаждаемого на планарную структуру. Электрическая прочность является одним из важнейших параметров диэлектрика применяемого для межслойной изоляции.

Структура содержала изолированные области со сформированными в них областями стоков и истоков n- и p-канальных МОП-транзисторов и подзатворный слой SiO_2 . Осаждение слоя $Si_{п/к}$ производилось газо-фазным методом на установке "Изотрон – 2М". Толщина слоя составляла 0,5 мкм. В серийной технологии К-МОП ИС этот слой используется для формирования затворов МОП-транзисторов и первого слоя коммутации элементов микросхемы. Для снижения сопротивления коммутирующего слоя после нанесения поликремний легируется диффузией фосфора. Далее следует операция травли фосфорно-силикатного стекла (ФСС) и процесс фотолитографии в поликремнии (формирование затворов и шин коммутации).

В производстве К-МОП ИС серии 590 между поликремневым и алюминиевым слоями коммутации в качестве межслойной изоляции на установке типа "Изотрон – 3М" в среде аргона газо-фазным методом осаждается слой SiO_2 толщиной 0,8 мкм. Эта операция достаточно сложна и не всегда обеспечивает стабильные и повторяемые результаты. Поэтому конечная цель настоящего исследования – определение возможности замены межслойной изоляции на основе слоя SiO_2 , осажденного газо-фазным методом, слоем SiO_2 , полученным путем термического окисления поверхности коммутирующего слоя поликремния.

Из приведенных в обзоре [1] методов исследования электрической прочности слоев диэлектриков авторами был выбран метод пробивных напряжений. Сущность этого метода заключается в получении распределения пробивных напряжений окисла по пластине. Для этого производится большое количество измерений в различных точках, распределенных по пластине. Отсчет напряжения пробоя производится на шкале характеристического графа в точке резкого изменения вида вольт-амперной характеристики окисленной структуры.

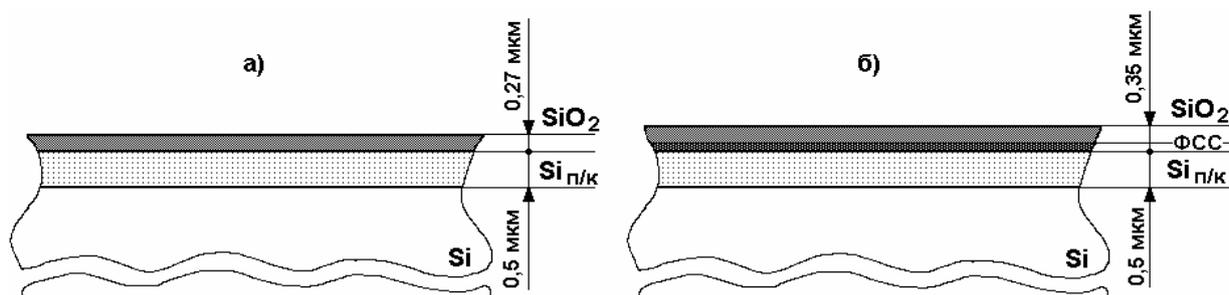


Рис.1. Структура исследованных образцов: а – образец № 1; б - образец № 2

Пробивные напряжения окисленной структуры измерялись на характеристическом графе Л2-56. Для этого на пластину устанавливается игольчатый зонд для осуществления электрического контакта. Затем производится последовательное увеличение

напряжения до пробоя слоя SiO₂. Напряжение пробоя определяется по проградуированной шкале характериографа Л2-56.

На рис. 1 показаны структуры двух видов подложек, отличающихся методами их формирования, а в таблице – режимы получения этих структур. Для изготовления указанных образцов использовался кремний монокристаллический марки КЭФ7,5(100), используемый для получения контрольных тестовых образцов при изготовлении ИС серии 590. Подготовка подложек и изготовление экспериментальных образцов производились на предприятии "Светлана-Полупроводники" по действующей технологии 590 серии. Вводимое в существующий процесс окисление Si_{п/к} производилось в одном случае после легирования Si_{п/к} фосфором, т.е. при сохранении на его поверхности ФСС, во втором случае – после стравливания ФСС.

Вид образца	Основные технологич. параметры
Si п/к+P+ затем ФСС стравлено (образец №1)	h=0,5 мкм+ легир.Р при t°=860 °С
Si п/к+P (образец №2)	h=0,5 мкм + легир.Р при t°=860 °С
SiO ₂ (образец №1)	h=0,27 мкм, t°=1000 °С, вл. O ₂ +HCl
SiO ₂ (образец №2)	h=0,35 мкм, t°=1000 °С, вл. O ₂ +HCl

Таблица. Основные режимы изготовления образцов

В результате проведения измерений были получены значения пробивных напряжений распределенных пластине, как уже было указано выше. Для описания полученных результатов измерений пробивных напряжений использовалась величина напряженности поля, соответствующая пробоем окисла. Результаты измерений, для наглядного представления, обрабатывали статистическими методами анализа многократных измерений. Результаты измерений представлены на рис. 2 в виде гистограммы распределения напряженностей поля, соответствующих пробоем в слое окисла.

Напряженность поля вычисляли по формуле: $E = \frac{U_{\text{ПР}}}{D}$, где D – толщина диэлектрика; $U_{\text{ПР}}$ – напряжение пробоя [2].

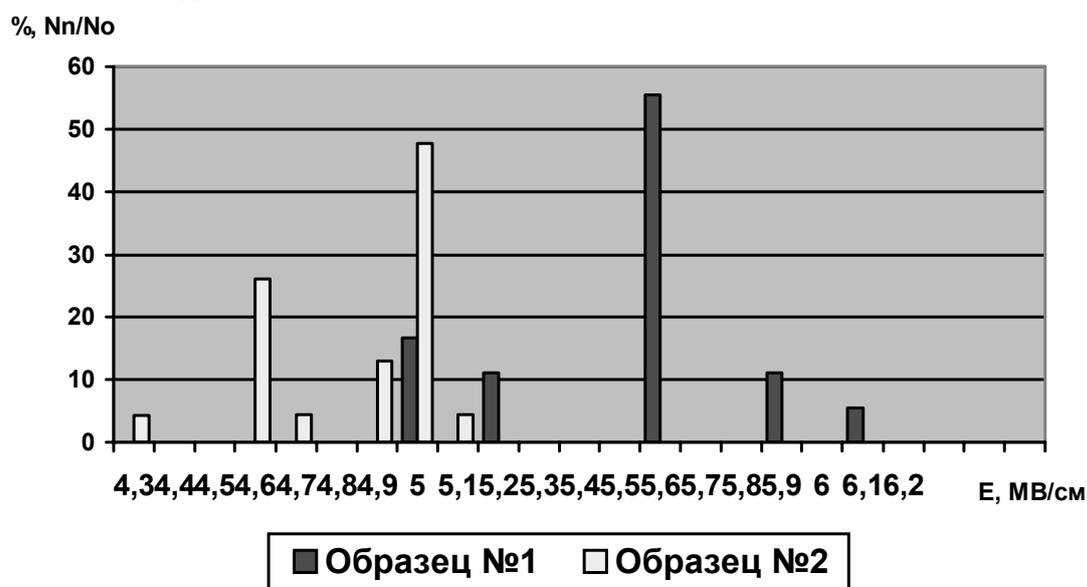


Рис. 2. Гистограмма напряженностей поля, соответствующих пробоем диэлектрика в исследуемых образцах

Как видно из представленной на рис. 2 гистограммы, у образца № 1, окисленного после снятия ФСС, общее распределение напряженностей электрического поля SiO_2 , соответствующих пробею, лежит в пределах от 5,6 МВ/см до 6,1 МВ/см, а у образца №2 – в пределах от 4,6 МВ/см до 5,1 МВ/см.

Основная часть измерений пробивных напряжений SiO_2 , у образца №1 характеризуется значением напряженности поля, соответствующей пробею, равным 5,6 МВ/см (75%) и более, а у образца №2 только 47% достигает значения 5,0 МВ/см.

Таким образом, образец № 1, с окислением $\text{Si}_{п/к}$ после снятия ФСС, имеет лучшие характеристики диэлектрической прочности по сравнению с образцом № 2. Следовательно, параметры изготовления образца № 1, представленные в таблице, могут быть рекомендованы при формировании межслойной изоляции шин коммутации.

Литература

1. Воробьев А.А., Завадовская Е.К. Электрическая прочность твердых диэлектриков. М.: Гостехиздат, 1956. 312с..
2. Кухлинг Х. Справочник по физике. М.: Мир, 1982. 323 с.

**ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС
ДЛЯ УСТАНОВЛЕНИЯ ХАРАКТЕРИСТИК ГАЗОРАЗРЯДНОГО
СВЕЧЕНИЯ "ГРВ-КОМПЛЕКС" И ПЕРСПЕКТИВЫ
ЕГО ИСПОЛЬЗОВАНИЯ ПРИ АНАЛИЗЕ СОСТОЯНИЯ
ОРГАНИЗМОВ И ВЫЯВЛЕНИИ ЭКОЛОГИЧЕСКОГО РИСКА**

**Н.С. Прияткин, К.Г. Коротков, Э.И. Слепян,
Г.С. Авдеева, Т.Б. Дорофеева, Г.О. Пономаренко**

Экологический риск, экологически обусловленные патологические явления и симбиотрофная заболеваемость, во многих отношениях экологически зависимая, – реальность как индустриального, так и постиндустриального общества (Слепян, 2002а), очевидное следствие усложнения, в том числе возрастающей стратификации, общества и возникновения кризисных ситуаций во взаимоотношениях общества и природы. Насущная проблема в связи с этим, требующая объективного и действенного решения, – обоснование и разработка аппаратурных технологий, обеспечивающих возможность выявления и краткосрочной дифференциальной численной диагностики изменений, имеющих значение, во-первых, свидетельств возникновения преморбидных (донозологических) отклонений в строении и функционировании организмов (как продуцентов, так и консументов и редуцентов), и, во-вторых, свидетельств появления начальных нарушений, возникающих в физических средах – в воздушном бассейне, в водных массах, в донноосадочном покрове, в почвенном покрове и в скальных массивах экотопов (биотопов). В особенности существенны упомянутые технологии для своевременного выявления свидетельств и предупреждения экологических чрезвычайных ситуаций, экологических катастроф, диверсионных актов экологического терроризма (Слепян 2002б) и ликвидации их последствий, а также при создании, поддержании и увеличении эффективности систем экологической защиты.

Одна из наиболее действенных и, что принципиально, оперативных технологий сверхраннего выявления и диагностики нарушений физических сред обитания и жизнедеятельности организмов – технология установления характеристик газоразрядного свечения [газоразрядная визуализация (ГРВ)], осуществляемая с использованием специального программно-аппаратурного комплекса – ГРВ-комплекса (Коротков, 2001 и др. раб.).

Технология газоразрядной визуализации, основанная на эффекте Кирлиан, имеет значение методики (ГРВ-методики), позволяющий оценивать энергетическое (фрактально-энтропийное) состояние абиотических и биотических составляющих экологических систем (Коротков, 2001). Современная аппаратура, применяемая при использовании ГРВ-методики, дает возможность регистрировать и анализировать газоразрядное свечение, индуцированное у организмов и у их структур, в том числе у биологических жидкостей, у природных вод, почв, грунтов, включая и химически загрязненные.

ГРВ-методика использована при анализе состояния пресных вод (водопроводной, загрязненной сточной, минеральной), эфирных масел, урины, жидкости крови, культуральной жидкости дрожжей, растений, беспозвоночных и позвоночных животных – олигохет, рыб, грызунов (Гудакова и соавт., 1988, 1990, Прияткин и соавт., 2003).

Установлено (Прияткин и соавт., 2002, 2003), что при интегральной экспресс-оценке состояния загрязненных вод активного стока, осуществляемой с применением ГРВ-методики, наиболее демонстративно сочетание фрактально-энтропийных параметров одиночных газоразрядных отображений (ГРВ-грамм) и характеристик тренда динамических кривых изменения абсолютной площади засветки, устанавливаемых посредством динамической ГРВ-графии (Коротков и соавт., 2001).

Следует отметить, что ГРВ-методика информативна при определении сравнительной активности жизнедеятельности растений, функционирующих в условиях стресса (Коротков, 2001), возникающего при подверженности растений повреждающим воздействием окружающей среды, пестицидов, возбудителей инфекций и инвазии, и т.д. Подтверждение изложенного, в частности, – специфика газоразрядного свечения тканей коры и древесины стебля однолетних побегов ильмовых (*Ulmus lacvis* Pall., *Ulmus grabra* Huds, *Ulmus scabra* Mill.) до и после колонизации *Graphium ulmi* Schwazz [сумчатая стадия – *Ophiostoma ulmi* (Buism.) Nannf.] – возбудителя трахеомикоза, которое является опаснейшим для ильмовых инфекционным заболеванием (рис. 1).

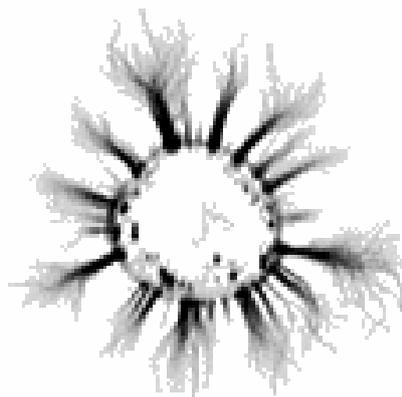


Рис. 1. Отображение газоразрядного свечения коры и древесины стебля однолетнего побега *Ulmus* spp. (поперечный разрез).

Совершенствование ГРВ-методики дает возможность использовать ее в диагностических целях при управлении ростом и развитием однолетних и многолетних растений, осуществлении интегрированной системы их защиты, установлении эффективности мероприятий по увеличению их хемотолерантности, и т.д.

Соответствующая адаптивная ГРВ-методика (Прияткин и соавт., 2002) позволила применить ее для оценки состояния жизнедеятельности красного калифорнийского дождевого червя *Eisenia fetida* (Savigny) – основного вида почвенных олигохет, используемых в целях вермикомпостирования.

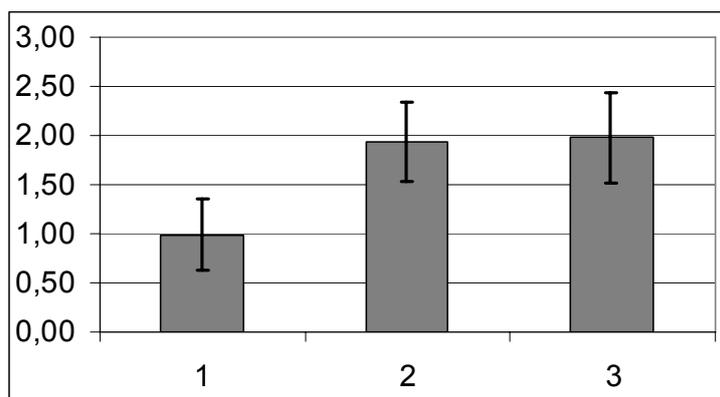


Рис. 2. Нормализованная площадь (в отн. ед.) ГРВ-отображений тела особей *Eisenia fetida* (Savigny) (ГРВ-фотографирование с фильтром), подвергнувшихся повреждающему воздействию: 1 - без воздействия, 2 - после мгновенного механического воздействия (надавливания), 3 - после выдерживания в водной среде 20 минут

Установленные различия ГРВ-параметров (нормализованной площади газоразрядного свечения, энтропии, фрактальности), характеризующие особи, во-первых, принадлежащие преемственным размерно-возрастным меропопуляциям и, во-вторых, подвергающиеся повреждающему механическому воздействию и воздействию водной среды (рис. 2), имеют, в связи с этим, определенное значение для определения каждой из меропопуляций в формировании гумуса.

Есть все основания полагать, что ГРВ-методика может быть использована при решении многих насущных вопросов экологии и ресурсоведения – для целей диагностики химического, механического и биологического загрязнения природных вод и почв, выявления аллелохимического почвоутомления и оценки состояния популяций организмов, для целей мониторинга и индикации, при наблюдении за жизнедеятельностью человека (в частности, в условиях подверженности воздействию ионизирующей радиации, электромагнитного, акустического, светового и вибрационного полей), а также для контроля за качеством продукции растениеводства и животноводства, обеспечивающим продовольственную безопасность.

Литература

1. Гудакова Г. З., Галынкин В.А., Коротков К.Г. Исследование характеристик газоразрядного свечения микробиологических культур. // Журнал. прикладной спектроскопии. 1988. Т. 49. №3. С. 412–417.
2. Дорофеева Т.Б., Тюпина Г.Н. Графиоз ильмовых в Санкт-Петербурге и меры борьбы с ним / Экология большого города. Альманах. Вып. 6. Ред. Х.Г. Якубов. Проблемы содержания зеленых насаждений в условиях Москвы // М.: Прима-М, 2002. С. 57–61.
3. Коротков К.Г. Основы ГРВ-биоэлектрографии. СПб., СПбГИТМО(ТУ), 2001. 360 с.
4. Коротков К.Г., Крыжановский Э.В., Борисова М.Б. Методика исследования объектов методом динамической ГРВ-графии. // Наука. Информация. Сознание. V международный научный конгресс по ГРВ-биоэлектрографии. Тезисы. СПб.: Издательский центр "Университетские Телекоммуникации", 2001. С.83–85.
5. Прияткин Н.С., Авдеева Г.С., Коротков К.Г., Слепян Э.И. Адаптивная методика получения газоразрядных изображений низших беспозвоночных животных методом ГРВ-биоэлектрографии и результаты ее применения при исследовании особей красного калифорнийского червя (*Eisenia fetida*) разных возрастных групп. // Наука. Информация. Сознание. VI Международный научный конгресс по ГРВ биоэлектрографии. Тезисы. СПб: ИВА, 2002. С.94–96.
6. Прияткин Н.С., Коротков К.Г., Слепян Э.И. Перспективы использования метода ГРВ-Биоэлектрографии в целях интегральной квалиметрии вод // Наука. Информация. Сознание. VI Международный научный конгресс по ГРВ биоэлектрографии. Тезисы. СПб: ИВА, 2002. С.92–93.
7. Прияткин Н.С., Коротков К.Г., Слепян Э.И., Бурцева Е.П. Интегральная оценка качества сточных вод предприятий здравоохранения с использованием методики газоразрядной визуализации в аспекте проблемы экологического риска. // Региональная экология. СПб., 2003. В печати.
8. Слепян Э.И. Экологический риск. // Региональная экология. СПб., 2002 а. №1–2(18). С. 62-82.
9. Слепян Э.И. Экологический терроризм // Жизнь и безопасность. СПб., 2002б. №1–2. С.277–283.

ПРИМЕНЕНИЕ САПР P-CAD-2001 ПРИ ПРОЕКТИРОВАНИИ ПЕЧАТНЫХ ПЛАТ

Е.Б. Романова

При разработке радиоэлектронной аппаратуры важным этапом является проектирование печатных плат. От разводки электрических связей зависит качество прохождения сигналов, а также размеры конструкции. Для достижения оптимальных результатов работы в области проектирования необходимо использовать современные средства САПР, обладающие целым рядом принципиально новых возможностей. Современной САПР для проектирования печатных плат является P-CAD-2001.

P-CAD предназначена для сквозного проектирования печатных плат. Она позволяет формировать электрические принципиальные схемы и топологию печатной платы, а также дает возможность оформления конструкторской и технологической документации. Выходные данные используются для вывода информации на различные устройства: принтеры, плоттеры, фотоплоттеры (для изготовления фотошаблонов), сверлильные станки и др.

Система P-CAD позволяет автоматизировать работу на всех этапах проектирования [1].

1. Формирование технического задания, включая:
 - оформление структурных и функциональных схем;
 - оформление текстовой документации;
 - разработку и моделирование электрической принципиальной схемы.
2. Конструирование печатных плат посредством разработки топологии, включая:
 - размещение компонентов на плате;
 - трассировку электрических соединений;
 - выпуск конструкторской документации.
3. Подготовку к производству, включая:
 - формирование файлов пригодных для изготовления;
 - выпуск технологической документации.

САПР P-CAD-2001 является удобным средством проектирования печатных плат. Она является одной из наиболее распространенных в России. Выбор этой САПР обусловлен поддержкой шрифтов TTF в обоих редакторах (Schematic и PCB) и адаптацией к отечественному оборудованию.

До сих пор в некоторых организациях при проектировании печатных плат используют версию P-CAD 4.5. Внедрение новой системы связано с тем, что разработчикам и конструкторам необходимо осваивать новую версию системы проектирования, а также настраивать оборудование для изготовления печатных плат, что приводит к финансовым и временным затратам. Но необходимо понимать, что в конечном итоге будет получен значительный выигрыш во времени за счет преимуществ новой системы и, следовательно, прирост финансов, так как скорость выполнения заказа возрастет и количество выполненных заказов за определенный срок увеличится.

При освоении новой версии САПР зависимость скорости проектирования от времени можно наблюдать на рис. 1.

Стоит отметить, что при освоении каждой новой версии программного продукта время освоения минимально и спад скорости проектирования во время освоения незначительный.

Зависимость скорости проектирования и экономической эффективности проекта прямо пропорциональная.

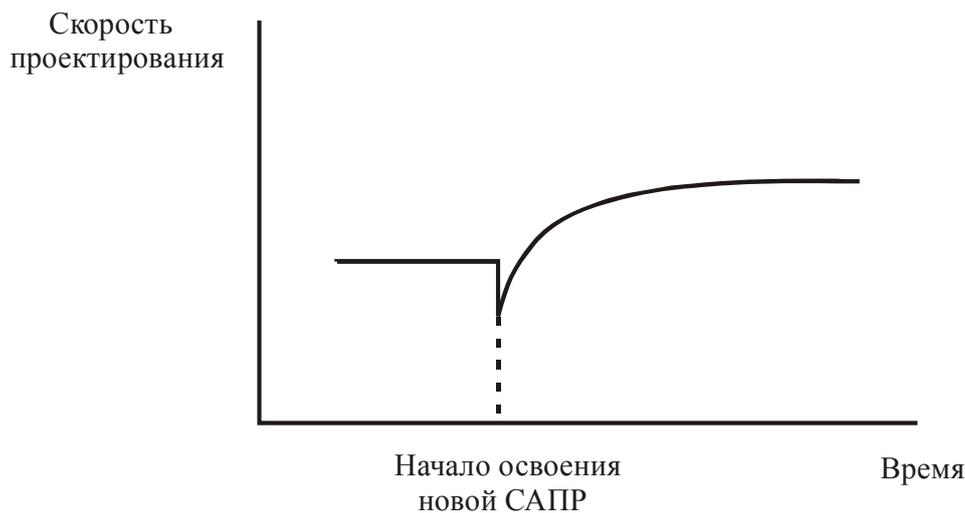


Рис. 1. Зависимость скорости проектирования от времени

Система P-CAD 4.5 работает в среде MS-DOS, а P-CAD-2001 – в Windows. P-CAD-2001 обладает следующими преимуществами перед версией P-CAD 4.5:

- удобный интерфейс;
- более удобное ведение библиотек компонентов;
- проверки на ошибки осуществляются непосредственно по окончании отдельного этапа проектирования, а в P-CAD 4.5 проверки были не на каждом этапе, вследствие чего было сложнее обнаружить ошибку;
- удобство разработки и редактирования на всех этапах;
- добавлены принципиально новые программные приложения, такие как:
 - а) графический редактор Relay, позволяющий предъявлять основные требования к плате со стороны разработчиков и системных конструкторов;
 - б) программный комплекс Design Flow, предназначенный для организации процесса проектирования;
 - в) средство просмотра чертежей схем и печатных плат через Интернет без возможности внесения в них изменения – View Center;
 - г) утилита анализа целостности сигналов;
 - д) некоторые утилиты DBX, позволяющие извлечь данные из файлов схем или плат, провести их обработку, выдать статистику, перекодировать в формат третьих фирм и т.п., например, утилита для контроля высоты компонентов, утилита для проверки зазоров между компонентами, утилита преобразования в формат DXF и др.;
- использование утилит стало более удобным за счет вызова их непосредственно из рабочей программы – графического редактора;
- оптимальная автотрассировка в системе SPECCTRA.

Система SPECCTRA представляет собой принципиально новый класс систем проектирования печатных плат [2]. В основе ее алгоритмов лежит так называемая бессеточная техника трассировки проводников, позволяющая задавать детальные правила для трассировки практически всех типов компонентов с учетом большого числа ограничений, при этом учитывая общие ограничения для всего проекта установок автотрассировки. За счет поддержки сложных иерархических правил проектирования улучшается технологичность изготовления платы. Описание точных геометрических моделей контактных площадок, трасс, дополнительных сквозных отверстий и других образов объектов в бессеточной базе данных позволяет максимально использовать доступное пространство, в результате чего возможно уменьшение количества слоев печатной платы. Бессеточная база данных занимает

гораздо меньше памяти, чем сеточная, используемая в P-CAD 4.5. С помощью команды Route/Fanout возможна трассировка коротких отрезков проводников для выравнивания их стартовых точек. Это позволяет разводить связи, идущие от компонентов, с малым шагом выводов, что в P-CAD 4.5 было невозможно.

Результаты автотрассировки двухслойной печатной платы контроля сигналов в различных версиях P-CAD представлены в таблице.

Параметры	P-CAD 4.5	P-CAD-2001
Количество переходных отверстий	180	142
Суммарная длина проводников	≈ 300 тыс.	≈ 270 тыс.
Образование "петель" проводников	20	отсутствуют
Площадь, занимаемая всеми	≈ 89000 мм ²	≈ 84000 мм ²
Процент автотрассировки	76%	100%

Таблица. Результаты автотрассировки

САПР P-CAD-2001 удачно вписывается в идеи CALS-технологий за счет:

- единого интерфейса,
- возможности преобразования файлов в стандартный формат данных – DXF и PDF,
- полной взаимосвязи данных, полученных на разных этапах проектирования,
- возможности коррекции на любом этапе проектирования с автоматическим обновлением необходимых данных в остальных документах,
- автоматизированного управления данными.

Таким образом, применение САПР P-CAD-2001 для проектирования печатных плат является наиболее актуальным за счет высокой скорости и качества трассировки.

Литература

1. Стешенко В.Б. ACCEL EDA. Технология проектирования печатных плат. М.: Нолидж, 2000.
2. Ёлшин Ю.М. Справочное руководство по работе с подсистемой SPECCTRA в P-CAD 2000. М.: СОЛОН-Р, 2002.

ИССЛЕДОВАНИЕ СТРУКТУР ТИПА $Si_{ПК}-SiO_2-Si$ С ОКИСЛАМИ, ПОЛУЧЕННЫМИ В РАЗЛИЧНЫХ РЕЖИМАХ ОКИСЛЕНИЯ

А.М. Семенов, А.М. Скворцов, Р.А. Халецкий

Как известно, замена алюминиевого затвора в МОП ИС на поликремневый приводит к ряду преимуществ:

- снижению пороговых напряжений;
- уменьшению длины канала за счет самосовмещения затвора;
- повышению стабильности МОП-структур;
- возможности формирования слоя коммутации элементов на основе поликремния.

В настоящей работе приводятся результаты исследования МОП-структур, полученных в различных режимах окисления. В качестве исследуемых объектов использовались МОП-структуры с поликремневым затвором и окислом, полученным при различных условиях окисления. Подложка МОП-структур представляет собой монокристаллический кремний n-типа с удельным сопротивлением 7.5 Ом·см, ориентированный в плоскости (100). Перед термическим окислением образцы проходили очистку в перекисно-аммиачном растворе и смеси серной кислоты и перекиси водорода.

№ образца	Толщина SiO_2 , мкм	Температура окисления, °С	Время окисления, мин			Наличие добавки Cl
			На 1-ом этапе в сухом кислороде	На 2-ом этапе в парах воды	На 3-ем этапе в сухом кислороде	
1	0,12	950	10	40	10	Да
2		1000	10	18	10	Да
3		1050	-	80	-	Да
4		1150	-	35	-	Да

Таблица 1. Режимы термического окисления образцов

В табл. 1 представлены режимы термического окисления образцов. Образец №1 получен с использованием производственного варианта окисления. В производстве 590 серии КМОП ИМС при получении подзатворного окисла применяют комбинированный метод окисления (сухой-влажный-сухой). Здесь вместо влажного окисления, когда кислород пропускается через борботер с водой, увлажнение кислорода осуществляют следующим образом: вода попадает в смеситель через специальную форсунку, испаряется и поступает вместе с кислородом и хлором в зону реакции, где находятся подложки. Выбор временных режимов окисления производился из условия нормировки по заданному значению толщины окисла.

На рис. 1 представлены характерные высокочастотные вольт-фарадные характеристики полученных образцов. Из анализа полученных кривых следует, что для всех образцов характерна малая плотность поверхностных состояний на межфазной границе Si-SiO₂. Следовательно, при таком существенном различии в технологических режимах отсутствует отличие в электрофизическом состоянии границы кремний-окисел. Возможно, это связано с пока недостаточно неизученной природой воздействия хлора на формируемую межфазную границу. Можно предположить, что ионы хлора, помимо взаимодействия с атомами щелочных металлов, производят пассивацию поверхности Si, уменьшая заряд на поверхностных состояниях.

Из рис.1 видно, что увеличение температуры до 1000°C для комбинированного окисления не приводит к заметному изменению вольт-фарадных характеристик (ВФХ). Изменение влажной среды окисления на сухую с одновременным увеличением времени окисления и температуры до 1000°C также не приводят к изменению ВФХ. Однако при увеличении температуры окисления в сухом кислороде до 1150°C происходит сдвиг кривой в направлении идеальной расчетной ВФХ. На основании этого факта можно заключить, что структуры с поликремневым затвором (образцы 1–3) имеют равные, достаточно большие фиксированные заряды в окисле, о чем свидетельствует положение соответствующих кривых относительно идеальной ВФХ.

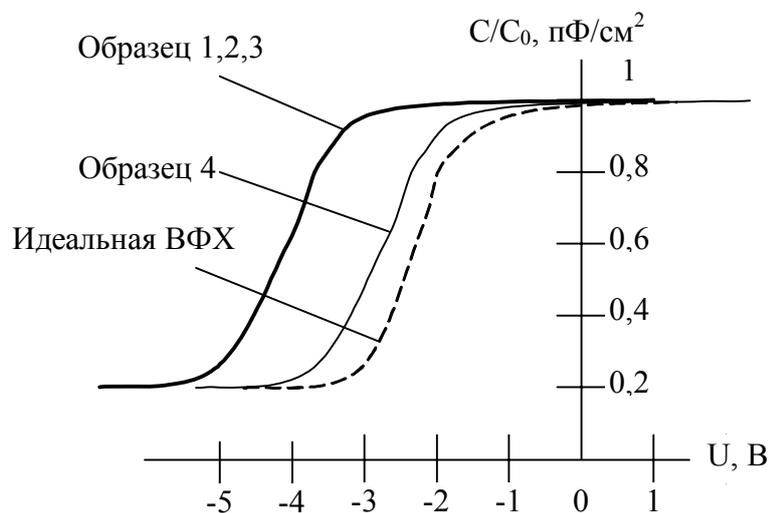


Рис. 1. Характерные экспериментальные вольт-фарадные характеристики образцов

При увеличении температуры сухого окисления окисел получается с меньшим встроенным зарядом. Это свидетельствует о том, что именно температура является преобладающим фактором, определяющим кинетику процесса окисления и параллельного формирования границы Si-SiO₂. Возможно, это связано с еще совсем не изученным поведением хлора в окисле и его влиянием на формирующуюся границу окисел-кремний [1].

Окисление в сухом кислороде при более высокой температуре дает лучшие результаты по сравнению с производственным окислением (кривая 4). Однако такое окисление может сопровождаться дополнительной разгонкой примеси в имплантированном слое, что может существенно сказаться на характеристиках готовой КМОП ИМС.

Для полного понимания данных процессов формирования окисла и межфазной границы Si-SiO₂ необходимы дополнительные исследования.

Литература

1. Санин К.В., Скворцов А.М. Технология БИС микропроцессоров и микро-ЭВМ. Учебное пособие. Л.: ЛИТМО, 1988.

ИССЛЕДОВАНИЕ МОП-СТРУКТУР С ОКИСЛАМИ, ПОЛУЧЕННЫМИ ПРИ РАЗЛИЧНЫХ РЕЖИМАХ ОКИСЛЕНИЯ

А.М. Семенов, А.М. Скворцов, Р.А. Халецкий

Термический оксид кремния представляет собой один из основных материалов, применяющихся в современной микроэлектронике. Он используется в качестве маски при проведении процессов диффузии, выполняет функции диэлектрика в МОП-структурах, изолятора в КСДИ-структурах и многослойной коммутации и др. Особенные жесткие требования к качеству предъявляются к термическому оксиду, применяемому в качестве подзатворного диэлектрика в МОП и КМОП интегральных микросхемах. При термическом окислении кремния одновременно происходит формирование межфазной границы Si-SiO₂. Именно электрофизические параметры окисла и границы интерфейса Si-SiO₂ являются главными факторами, определяющими работу и надежность функционирования КМОП ИМС.

Несмотря на большое количество научных работ, на сегодняшний день отсутствует единая установившаяся теория по физике Si-SiO₂, которая бы полностью объясняла все экспериментальные факты, зачастую противоречащие друг другу.

Как известно, процесс получения подзатворного окисла отличается своими режимами от процесса получения окисла над другими участками кремниевой подложки. Во-первых, подзатворный окисел часто получают в сухом кислороде. Во-вторых, при окислении вводят добавки хлора, чтобы исключить влияние подвижных ионов натрия и калия в окисле на работу КМОП ИМС [1]. Известно также, что использование влажного окисления не рекомендуется для подзатворного окисла из-за худшего, по сравнению с сухим окислением, качества окисла [2]. Несмотря на эти известные факты, применение влажного кислорода для получения подзатворного окисла имеет место в серийном производстве КМОП ИМС 590-ой серии на заводе “Светлана-Полупроводники”.

В настоящей работе приводятся результаты исследования МОП-структур на основе Al-SiO₂-Si, с окислом, полученном в различных режимах окисления. В качестве исследуемых объектов использовались МОП-структуры с алюминием в качестве затвора и окислом, полученным при различных условиях окисления. Подложка МОП-структур представляет собой монокристаллический кремний n-типа с удельным сопротивлением 7.5 Ом·см, ориентированный в плоскости (100). Перед термическим окислением образцы проходили очистку в перекисно-аммиачном растворе и смеси серной кислоты и перекиси водорода.

№ образца	Толщина SiO ₂ , мкм	Температура окисления, °С	Время окисления, мин			Наличие добавки Cl
			На 1-ом этапе в сухом кислороде	На 2-ом этапе во влажном кислороде	На 3-ем этапе в сухом кислороде	
1	0,12	950	10	40	10	Да
2		1000	10	18	10	Да
3		1150	-	35	-	Да

Таблица. Режимы термического окисления образцов

В таблице представлены режимы термического окисления образцов. Образец № 1 представляет собой производственный вариант исполнения. В производстве 590 серии КМОП ИМС используется комбинированный метод окисления при получении

подзатворного окисла (сухой-влажный-сухой). Здесь вместо влажного окисления, когда кислород пропускается через барботер с водой, увлажнение кислорода осуществляют следующим образом: вода попадает в смеситель через специальную форсунку, испаряется и поступает вместе с кислородом и хлором в зону реакции, где находятся подложки. Выбор временных режимов окисления производился из условия нормировки по заданному значению толщины окисла.

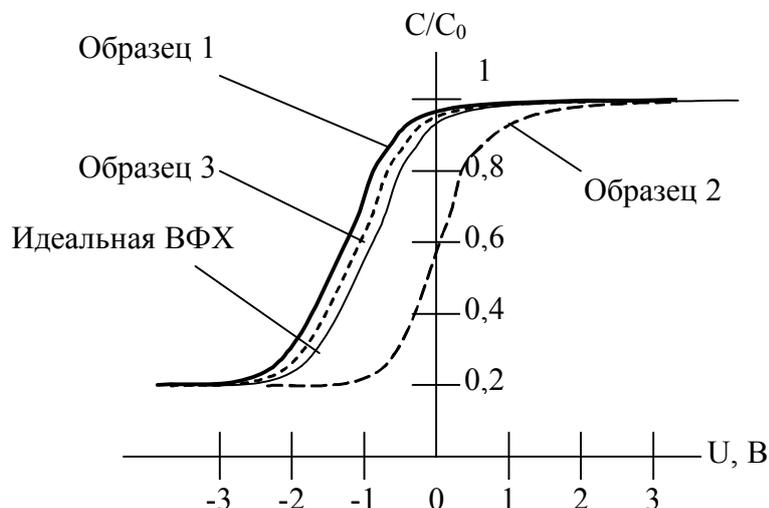


Рис. 1. Характерные экспериментальные вольт-фарадные характеристики образцов

На рис. 1 представлены характерные высокочастотные вольт-фарадные характеристики полученных образцов. Из анализа полученных кривых следует, что для всех образцов характерна малая плотность поверхностных состояний на межфазной границе Si-SiO₂. Следовательно, при таком резком отличии в технологических режимах отсутствует отличие в электрофизическом состоянии границы кремний-окисел. Возможно, это связано с пока неизученной природой воздействия хлора на формируемую межфазную границу. Можно предположить, что ионы хлора, помимо взаимодействия с атомами щелочных металлов, производят внутреннюю пассивацию приграничной области Si-SiO₂, уменьшая заряд на поверхностных состояниях.

При увеличении температуры от 950 °С до 1000 °С происходит сдвиг вольт-фарадной характеристики вправо на 0,8 В по оси напряжений, что эквивалентно избыточному фиксированному отрицательному заряду в окисле. При этом время окисления уменьшилось в два раза (см. таблицу). Можно предположить, что этот отрицательный заряд обусловлен тем, что с увеличением температуры увеличивается коэффициент диффузии гидроксильных -ОН и силанольных Si-OH групп, концентрация которых в условиях окисления в парах воды высока. Происходит увеличение потока этих групп к поверхности кремния, который начинает преобладать над скоростью химической реакции на границе Si-SiO₂. Непрореагировавшие ионы ОН- начинают взаимодействовать с окислом. Происходит формирование окисла с высокой концентрацией немостикового кислорода, которая обусловлена замещением атома кислорода с валентностью -2 в вершине тетраэдра SiO₄ на ион гидроксила с валентностью -1. Таким образом, избыточные гидроксильные и силанольные группы, а также немостиковый кислород в приповерхностной области обуславливают появление отрицательного заряда.

Окисление в сухом кислороде при более высокой температуре дает несколько лучшие результаты по сравнению с производственным окислением (рис. 1). Однако это может сопровождаться дополнительной разгонкой примеси в имплантированном слое, что может существенно сказаться на характеристиках готовой КМОП ИМС.

Таким образом, увеличение температуры при использовании комбинированного окисления существенно снижает качество окисла. Использование сухого окисления позволяет получить окисел с наименьшим значением фиксированного заряда, однако повышается температура окисления. Для полного понимания данных процессов формирования окисла и межфазной границы Si-SiO₂ необходимы дополнительные исследования.

Литература

1. Технология СБИС / Под ред. С. Зи. Т.1. 2. М.: Мир, 1986.
2. Майоров С.А., Скворцов А.М. Технология производства вычислительных машин. М.: Высшая школа, 1973.
3. Пичугин И.Г., Таиров Ю. . Технология полупроводниковых приборов. М.: Высшая школа, 1984.

ОЦЕНКА СУЩЕСТВУЮЩИХ МЕТОДОВ ОТРАБОТКИ И ПРОВЕРКИ КБО ГРАЖДАНСКИХ САМОЛЕТОВ

А.М. Рогачевский

Введение

ОКБ "Электроавтоматика", которое представляет автор, традиционно ведет работы по КБО летательных аппаратов различного назначения в самом широком смысле этого слова: разработка, испытания, поставка опытных образцов, сопровождение эксплуатации, модернизация.

Работы ведутся в соответствии с требованиями ТЗ, в которых в настоящее время значительное внимание уделяется таким вопросам, как финансово-временные затраты на различных этапах жизненного цикла КБО: стоимость и срок появления 1-ого опытного образца, затраты на этапе эксплуатации на проведение технического обслуживания и ремонтов, а также проведение модернизации КБО на протяжении более чем 30-летнего жизненного цикла самолета.

В соответствии с указанными требованиями формулируется следующий критерий разработки КБО: "Минимизация суммарной стоимости КБО, отнесенной ко всему его жизненному циклу с обеспечением возрастающих требований к его эффективности".

Основные способы реализации данного критерия представлены в табл. 1.

№	Этапы жизненного цикла КБО	Основные статьи финансово-временных затрат	Способы минимизации финансово-временных затрат
1	Приобретение	Разработка	Стандартизация и унификация аппаратных и программных модулей в целях использования базовых разработок
		Испытания (стендовые, наземные, летные)	Оптимальные способы и соотношения объемов проверок КБО в стендовых, наземных и летных условиях
		Поставка образца КБО	Унификация компонентной и элементной базы аппаратуры КБО
2	Эксплуатация	Техническое обслуживание и ремонт	Отказоустойчивость КБО, реализуемая путем информационной и аппаратурной избыточности. Унификация компонентной и элементной базы аппаратуры КБО
		Модернизация	Открытая архитектура КБО, реализуемая путем избыточной вычислительной мощности и пропускной способности интерфейсов, стандартизацией и унификацией аппаратных и программных модулей. Загружаемое ПО.

Таблица 1. Критерий разработки КБО - минимизация суммарной стоимости КБО, отнесенной ко всему его жизненному циклу с обеспечением возрастающих требований к его эффективности

Существует два этапа жизненного цикла КБО: приобретение (с подэтапами разработки, испытаний и изготовления опытного образца) и эксплуатация (с подэтапами технического обслуживания, ремонта и модернизации).

Финансово-временные затраты подэтапа разработки могут быть уменьшены стандартизацией и унификацией аппаратных и программных модулей в целях возможности использования базовых разработок, подэтапа испытаний – оптимизацией способов и соотношений объемов проверок в стендовых, наземных и летных условиях.

Стоимость образца КБО может быть уменьшена унификацией компонентной и элементной базы аппаратуры КБО.

На подэтапе технической эксплуатации и ремонта основополагающим является свойство отказоустойчивости КБО, реализуемое путем информационной и аппаратурной избыточности, что позволяет достоверно определить его текущее состояние и реконфигурировать, при необходимости, архитектуру. Важным является также унификация и быстросъемность аппаратуры КБО.

На подэтапе модернизации основополагающим является открытая архитектура КБО, реализованная путем избыточной вычислительной мощности и пропускной способности интерфейсов, реализация загружаемого программного обеспечения, а также унификацией аппаратных и программных модулей.

Назначение и состав КБО

В зависимости от назначения гражданских самолетов (транспортные, грузовые, конвертируемые транспортно-грузовые, для исследования земной поверхности, воздействия на атмосферу, самолеты-салоны и т.д.) требования к КБО также различны. Это либо навигационный комплекс, либо навигационный комплекс с решением дополнительных задач (индикация навигационных параметров; управление общесамолетным оборудованием, индикация и контроль параметров общесамолетного оборудования; выход в точку сброса и управление сбросом грузов; управление аппаратурой воздействия на атмосферу; управление аппаратурой исследования земной поверхности и т.д.).

Однако основное назначение КБО во всех случаях неизменно:

- автоматизированное и ручное самолетовождение с оптимизацией режимов полета по внутренним и зарубежным трассам гражданской авиации;
- управление радиотехническими системами (РТС) в автоматизированном и ручном режимах;
- отображение экипажу пилотажной, навигационной и обзорной информации, а также справочной информации, информации о функционировании КБО, его отказах и неисправностях, рекомендации по парированию отказов и неисправностей;
- контроль технического состояния КБО и, при необходимости, его реконфигурация, а также регистрации, хранение и выдача результатов контроля при техническом обслуживании самолета.

В соответствии с назначением КБО может включать в себя ряд подсистем:

- "Ядро", состоящее из информационно-вычислительной системы (ИВС) и информационно-управляющего поля (ИУП). ИВС организована на базе высокопроизводительных вычислителей, состоящих из унифицированных конструктивных функциональных модулей (КФМ). ИУП комплекса является одной из наиболее важных его частей, так как оно обеспечивает связь пилота с КБО и летательным аппаратом в целом. В состав ИУП входят индикаторы и органы управления. В качестве индикаторов могут быть использованы, например, многофункциональные цветные индикаторы (МФЦИ) с жидкокристаллической панелью, электромеханические индикаторы, индикационная панель пультов управления и индикации (ПУИ). Управляющая часть ИУП может включать клавиатуру ПУИ, кнопочное обрамление МФЦИ и т.д.;
- навигационно-пилотажное оборудование, которое может включать инерциальную систему (ИС), курсовертикаль (КВ), систему воздушной сигналов (СВС) и т.д.;
- радиотехническое оборудование навигации и посадки, которое может включать радиотехническую систему навигации и посадки (РСБН), спутниковую навигационную систему (СНС), доплеровский измеритель скорости и угла сноса

(ДИСС), автоматический радиокompас (АРК), радиовысотомер (РВ), самолетный ответчик (СО) и т.д.;

- систему автоматического управления (САУ);
- общесамолетное оборудование (ОСО);
- радиосвязное оборудование (РС) и т.д.

Часть указанных систем может входить в состав КБО, часть – являться самолетным оборудованием и взаимодействовать с КБО системами.

Вопрос выбора состава КБО определяется не только техническими причинами.

Поколения цифровых КБО ОКБ "Электроавтоматика"

Поколения цифровых КБО ОКБ "Электроавтоматика" представлены в табл. 2.

Поколение КБО	Наименование КБО	Суть КБО		Контроль КБО			
		Назначение	Состав	Автономный входной контроль систем КБО	Стендовый контроль КБО		
					Разрабатываемые имитаторы с аналог. и цифр. интерфейсом	Серийные имитаторы с цифр. интерфейсом	Имитаторы с цифровым и аналоговым интерфейсом на базе ПК
I	ОЛЬХА	НК	Полный	КПА систем	+	+	—
	ПИЖМА	НК					
	ТОПОЛЬ	НК + спец.					
II	МАЛЬВА	НК+ спец.					
	МАК						
	ИРИС						
	ЦИКЛОН						
	ЧЕРНИКА						
III	ЖАСМИН	НК					
	ВСС-85						
IV	ССИ-80	НК + СЕИ + СУОСО	Ядро	ВСК систем	+	—	+
	КВСС-74	НК + СЕИ			—		
	КВСС-334	НК + СЕИ					

Таблица 2. Обзор методов входного контроля и стендовой отработки КБО

К первому поколению могут быть отнесены КБО, появившиеся в начале 70-х годов, установленные на самолетах Як-42, Ил-86 и Ан-72. Они базировались на 16-разрядном бортовом вычислителе типа БЦВМ20, имеющем быстродействие порядка 200 000 оп/с на операциях типа "сложение", емкость ОЗУ – 2 Кб, ПЗУ – 64 Кб.

Ко второму поколению относятся КБО для самолетов Ан-74 и его модификаций, ТУ-134СХ, Ил-24Н и т.д., ядром которых является доработанный 32-разрядный бортовой вычислитель БЦВМ20.

КБО третьего поколения базировались на БЦВМ80 производительностью 800 тыс. оп/с на операциях типа R-R, с емкостью ОЗУ – 32 Кб, ПЗУ – 256 Кб. Они установлены на самолетах ТУ-154М, ТУ-204, Ил-96.

К четвертому поколению относятся КБО для самолетов С-80, Ан-74ТК-300, ТУ-334, находящихся в настоящее время на этапе летных испытаний. Ядро КБО включает ИВС (бортовой 32-разрядный вычислитель БЦВМ90 с производительностью 25 млн. оп/с на операциях типа R-R, емкость ОЗУ – 1 Мб, ПЗУ –

4 МБ и развитый интерфейс) и ИУП (пульт ПУИ с экраном 3"×4" и клавиатурой, а также индикаторы МФЦИ с размером экрана 6"×8" с кнопочным обрамлением). Для самолета ТУ-334 ядром КБО является пульт-вычислитель (ПВ), объединяющий функции высокопроизводительного вычислителя и пульта управления.

Входной контроль систем КБО

Отработка КБО на базе предприятия-разработчика представлена в табл. 2 и проводится в два этапа: входной контроль составных частей КБО и комплексная отработка КБО на стенде.

Входной контроль составных частей КБО первого, второго и третьего поколения проводился с помощью контрольно-проверочной аппаратуры (КПА).

КПА разрабатывалась, изготавливалась и поставлялась совместно с поставкой входящих в КБО систем для их автономной проверки.

Для КБО четвертого поколения КПА в основном отсутствует, поскольку развитая система встроенных средств контроля (ВСК), обеспечивающая полноту контроля не менее 0,95, позволяет отказаться от специальной КПА при проведении входного контроля систем и далее на этапе эксплуатации.

Требование проведения подготовки к полетам и всех видов обслуживания КБО в процессе эксплуатации без применения КПА присутствует в ТЗ на КБО четвертого поколения.

Стендовая отработка КБО

Стенд для отработки КБО включает в свой состав жгутовую самолетную схему (фидер), технологические рамы для установки входящих в КБО систем, набор имитаторов взаимодействующих с КБО систем и дополнительное оборудование (стойки, приборная доска, система электропитания, система обдува и т.д.).

КБО первого, второго и частично третьего поколений включало в свой состав не только "ядро", но и широкий набор систем, в связи с чем стенды их отработки требуют установки всего набора входящих в КБО систем, используя имитаторы (как правило, электромеханические аналоговые и громоздкие цифровые) только взаимодействующего с КБО оборудования.

В качестве примера серийно изготавливаемого имитатора с цифровым интерфейсом, используемого при отработке КБО, может быть приведен задатчик кодовой информации (ЗКИ).

ЗКИ обеспечивает:

- задание и выдачу в КБО информационных слов 32-разрядным биполярным последовательным кодом в соответствии с ГОСТ 18977-79 и РТМ 1495-75 (обеспечивается от 1 до 8 слов в цикле с выдачей в асинхронном режиме);
- индикацию выбранного слова в 32-разрядном двоичном коде.

ЗКИ имеет следующие характеристики:

- выдача последовательного кода – с частотой 50 Кбит/с;
- электропитание ~ 200 В, 400 Гц;
- масса – 15 кг.

Такой подход являлся неудовлетворительным, поскольку требовал существенных финансовых и временных средств на доставку входящих в КБО систем, их распаковку, входной контроль, отработку, упаковку и т.д. Поэтому впоследствии в связи с рядом причин (развитие средств вычислительной техники, ограничения экономического и временного характера и т.д.) подход к определению состава и

отработке КБО был изменен. В состав КБО стали включать только "ядро", состоящее из ИВС и ИУП.

В качестве имитаторов взаимодействующего с КБО оборудования начали использовать персональные компьютеры с установленными модулями сопряжения (шина – внешняя среда) и программами, реализующими статические модели взаимодействующих систем.

Так, например, используются модули сопряжения:

- МД01-24 (с 2-мя выходными и 6-ью входными каналами 32-разрядного биполярного последовательного кода, а также с 4-мя входными и 4-мя выходными разовыми командами обрыв/корпус);
- МД03 (с 6-ью выходными каналами 32-разрядного биполярного последовательного кода),

реализующие цифровой интерфейс в соответствии с ГОСТ 18977-79.

В связи с небольшим количеством свободных слотов в ПК начали использоваться специальные контейнеры, представляющие собой промышленные ПК, допускающие установку до 20 модулей сопряжения. Подобное использование позволяет минимизировать весогабаритные и стоимостные характеристики имитаторов взаимодействующего с КБО оборудования. Исчезает также необходимость реализации локальной сети для централизованного управления системой моделей.

Так, например, используются модули сопряжения:

- PCL-726 (с 6 выходными каналами постоянного или переменного тока, 16 выходными и 16 входными каналами TTL уровня);
- PCL-727 (с 12 выходными каналами постоянного тока, 16 выходными и 16 входными каналами TTL уровня)

и прочие, реализующие цифровой интерфейс в соответствии с ГОСТ 18977–79, а также различные аналоговыми интерфейсами.

Заключение

Анализируя рассмотренные выше способы отработки и проверки КБО гражданских самолетов на базе предприятий-разработчиков, можно предложить следующую технологию проведения данной работы:

- на этапе написания ТЗ на взаимодействующее с КБО оборудование представлять требование его проверки с помощью средств ВСК;
- входной контроль взаимодействующих систем проводить с использованием ВСК;
- при отработке КБО на стенде использовать в качестве имитаторов взаимодействующих систем промышленный ПК с определенными типами и количеством модулей сопряжения, позволяющими реализовать необходимые аналоговые и цифровые интерфейсы. Возможно также использование стандартных ПК с модулями сопряжения, объединенных в локальную вычислительную сеть;
- в качестве моделей взаимодействующих систем использовать математические статические модели, реализующие несколько вариантов наборов входных-выходных параметров каждой системы.

Для повышения качества отработки можно считать перспективным развитие описанной выше технологии отработки КБО, заключающееся в реализации математических динамических моделей взаимодействующих систем, самолета, внешней среды.

ПРОБЛЕМЫ НАДЕЖНОСТИ МАГНИТОУПРАВЛЯЕМЫХ КОНТАКТОВ В СИСТЕМАХ УПРАВЛЕНИЯ

В.Л. Ткалич, Н.В. Жуков

Несмотря на то, что на сегодня предложено множество решений, посвященных повышению срока службы и эффективности устройств автоматики на основе различных упругих элементов, задача повышения надежности таких систем приобретает все большую актуальность. Такие упругие элементы входят в конструкцию большинства известных устройств, приборов и механизмов, и они являются их незаменимыми функциональными частями.

Разработан новый подход к исследованию упругих чувствительных элементов (УЧЭ) герконов, основанный на применении абелевых функций, что позволяет достичь наиболее правильного аналитического описания как самих УЧЭ, так и динамических процессов происходящих в них.

Найдем амплитудно-частотную характеристику (АЧХ) для нелинейного дифференциального уравнения, описывающего динамику плоских УЧЭ. Для этого воспользуемся методом комплексных амплитуд [2]. Положим, что возмущающая сила меняется по закону $F(t) = F_0 \cos \omega t$, и приведем уравнение

$$A \frac{d^2 \varphi}{dt^2} + B_1 \frac{d\varphi}{dt} + B_2 \frac{d\varphi}{dt} \left| \frac{d\varphi}{dt} \right| + C \sin \varphi = F(t)$$

к виду

$$\ddot{y} + 2 \frac{n}{m_\Sigma} \dot{y} + \frac{\mu}{m_\Sigma} \dot{y} |\dot{y}| + p_k^2 y = (F_0 / m_\Sigma) \cos \omega t. \quad (1.1)$$

Введем в рассмотрение комплексную величину y^* , действительная часть которой совпадает с выражением для смещения $\text{Re} y^* = y$. Зависимость возмущающей силы от времени также представим в комплексной форме: $F^*(t) = F_0 \exp(i\omega t)$, так что

$$F(t) = \text{Re } F^*(t) = F_0 \cos \omega t.$$

Действительная часть решения уравнения

$$\ddot{y}^* + 2 \frac{n}{m_\Sigma} \dot{y}^* + \frac{\mu}{m_\Sigma} \dot{y}^* |\dot{y}^*| + p_k^2 y^* = (F_0 / m_\Sigma) e^{i\omega t}$$

совпадает с решением уравнения (1.1), так как коэффициенты уравнения являются действительными величинами. Искомое решение запишем в виде

$$y^* = A^* \exp(i\omega t). \quad (1.2)$$

Соответственно,

$$\begin{aligned} \dot{y}^* &= A^* \omega i e^{i\omega t}; \\ \ddot{y}^* &= -A^* \omega^2 e^{i\omega t}; \end{aligned} \quad (1.3)$$

$$|\dot{y}^*| = |A^* \omega i e^{i\omega t}| = A^* \omega.$$

Подставив (1.2) в (1.3), получим:

$$A^* (-\omega^2 + 2 \frac{n}{m_\Sigma} \omega i + \frac{\mu}{m_\Sigma} A^* \omega^2 i + p_k^2) = F_0 / m_\Sigma, \quad (1.4)$$

откуда определяется комплексная амплитуда:

$$A^*(\omega) = \frac{F_0}{m_\Sigma \left[p_k^2 - \omega^2 + i \left(2 \frac{n}{m_\Sigma} \omega + \frac{\mu}{m_\Sigma} A^* \omega^2 \right) \right]} \quad (1.5)$$

или, в показательной форме,

$$A_* = A_* \exp(i\omega t),$$

где

$$A(\omega) = \frac{F_0}{m_{\Sigma} \sqrt{(p_k^2 - \omega^2)^2 + (2\frac{n}{m_{\Sigma}}\omega + \frac{\mu}{m_{\Sigma}}A_*\omega^2)^2}}, \quad (1.6)$$

$$\phi = \arctg \frac{2(n/m_{\Sigma})\omega + (\mu/m_{\Sigma})A_*\omega^2}{p_k^2 - \omega^2}. \quad (1.7)$$

Формулы (1.6)–(1.7) являются искомыми в расчете.

Для построения графиков АЧХ герконов различных типов разработан пакет прикладных программ (ППП) на языке программирования Visual C++ 6.0 Enterprise Edition. При расчете функции $A(\omega, p_k)$ использовался метод половинного деления [1, 3]. Начальными данными в алгоритме расчета АЧХ являются коэффициенты уравнения и требуемый уровень точности вычислений. Разработанные математические модели для АЧХ плоских пружин герконов позволяют выявить влияние конструктивных параметров магнитоуправляемых контактов (МК) на их АЧХ, что очень важно при разработке инженерной методики подбора коэффициента демпфирования данных устройств в зависимости от рабочего диапазона.

Выявлено новое определение инерционных свойств жидкости, выражающееся через коэффициенты присоединенных масс. Знание этих присоединенных масс и моментов инерции очень важно при исследовании жидкости и тела в нестационарном взаимодействии. Влияние нелинейной составляющей жидкостного трения отражается на частотных характеристиках измерительных устройств, работающих в широком частотном и амплитудном диапазонах. К числу таких устройств можно отнести газонаполненные и ртутносмачиваемые герконы. При получении ММ важно учитывать широкий спектр факторов – ориентацию УЧЭ в пространстве, показатели обтекаемости элемента и его геометрию, влияние твердых стенок, влияние отрыва жидкости от поверхности.

Анализ результатов большого числа исследований поверхностей контактных сердечников (КС) с нерегулярными микрорельефами, имеющими случайный стационарный характер, показал, что для улучшения ряда эксплуатационных параметров необходимо добиваться улучшения качества рабочих поверхностей УЧЭ путем регуляризации их микрогеометрии. Способ вибронакатывания позволяет получить оптимальную регулярную микрогеометрию и приповерхностный слой материала УЧЭ с высоким коэффициентом упрочнения.

Анализ сущности и особенностей регулярного микрорельефа (РМР) свидетельствует о том, что практически все недостатки, свойственные шероховатым поверхностям деталей УЧЭ, при создании РМР исключаются.

Литература

1. Андреева Л.Е. Упругие элементы приборов. М.: Машиностроение, 1981. 455 с.
2. Бидерман В.А. Прикладная теория механических колебаний. М.: Высшая школа, 1972. 416 с.
3. Ткалич В.Л., Лобанцев А.В., Галин Ю.В. Построение уточненного аналитического выражения АЧХ газонаполненных герконов. // Тезисы доклада на Всероссийской научной конференции (Computer-Based Conference) "Методы и средства измерений", раздел 11 "Математические модели и численное моделирование измерительных приборов и датчиков". Нижний Новгород, 1 мая 2000 г. С.17.

СОЗДАНИЕ БИБЛИОТЕКИ КОНЕЧНЫХ ЭЛЕМЕНТОВ В ПРИЛОЖЕНИИ К ПЛАСТИНАМ И МЕМБРАНАМ

В.Л. Ткалич, В.А. Худяков

Прогнозирование пространственно-временного поведения тонкостенных гофрированных оболочек вращения при статическом и динамическом нагружении является одной из перспективных задач как в технике (в частности, в приборостроении), так и в медицине.

Сформирована библиотека конечных элементов тонкостенных упругих чувствительных элементов (УЧЭ), состоящая из конечных оболочек вращения и конечных элементов пластин и мембран. Показано, что при использовании моментной схемы конечных элементов повышается точность расчетов деформаций и напряжений тонкостенных УЧЭ.

При расчете тонкостенных пространственных структур УЧЭ произвольной стереометрии (с переменной толщиной стенки, с произвольным видом гофра, с изменяющимся шагом витка УЧЭ и т.д.) подход к формированию расчетной модели аналогичен. Так, в качестве базовых элементов используются трех- и четырехугольные элементы оболочек Киргоффа-Лява и С.П. Тимошенко, в узлах и по линии сопряжения должны выполняться кинематические условия сопряжения, а функции формы строятся таким образом, чтобы на общей границе отсутствовали разрывы полей перемещений и их производных.

Получены матричные соотношения термоупругости для общего случая напряженных цилиндрических и сильфонных оболочек вращения, выполненных из анизотропных и ортотропных материалов, включающие матрицы жесткости, выраженные через компоненты матрицы податливости.

Стандартный подход к формированию матриц жесткости с использованием системы интерполяционных полиномов при расчете тел сложной формы достаточно часто не обеспечивает удовлетворительной сходимости из-за отсутствия корректного учета смещения конечного элемента как жесткого целого. Все традиционные подходы к улучшению сходимости имели в своей основе более или менее удачные попытки повысить точность аппроксимации для учета жесткого смещения, оставляя в стороне вопрос о влиянии на точность решений жестких смещений элементарных объемов при деформировании. Однако именно этот фактор может оказать определяющее влияние на точность результата.

При расчете конструкций, жесткие повороты частей которых значительны по сравнению с деформациями, погрешности резко возрастают, и точность полученных на основе МКЭ решений значительно меньше, чем при решении задач, в которых жесткий поворот отсутствует.

Следовательно, для учета основных свойств жестких смещений в произвольном конечном элементе следует изменить схему вывода соотношений МКЭ. Суть подхода сводится к отбрасыванию или минимизации некоторых членов разложения деформаций в степенной ряд, реагирующих на жесткие смещения. При этом точные уравнения связи деформации – перемещения заменяются приближенными в результате выполнения следующих формальных процедур:

1. представления функции перемещений в виде отрезков степенных рядов (принятый закон перемещении);
2. подстановки приближенных выражений для перемещений в формулы деформаций и последующего разложения деформаций в ряд Маклорена;
3. повторения предыдущих процедур с дополнением перемещений до бесконечного степенного ряда и сохранения в разложении деформаций в ряд Маклорена только

совпадающих членов для коэффициента разложений, полученных на основе принятого закона перемещений.

Построенные по этим принципам разрешающие уравнения МКЭ позволяют учесть свойства жестких смещений конечных элементов и удовлетворяют требованиям существования, сходимости и устойчивости решений при неограниченном сгущении сетки и невырожденности конечного элемента.

В настоящее время из четырехугольных элементов оболочек наиболее широко применяются модифицированный элемент с пятью степенями свободы в узлах и изопараметрический элемент с большим числом степеней свободы в узлах.

У элемента первого типа в каждом узле вектор обобщенных узловых перемещений имеет пять компонентов – три компоненты вектора перемещений и два угла поворота, т.е.

$$\{U\}^i = [u_1 u_2 u_3 \vartheta_1 \vartheta_2].$$

Углы поворота определяются по формулам

$$\vartheta_1 = \frac{1}{r} \frac{\partial u_3}{\partial \varphi} - \frac{u_2}{R_2}; \vartheta_2 = \frac{\partial u_3}{\partial s} + \frac{u_1}{R_1}.$$

Здесь s, φ – меридиональная и окружная криволинейные координаты срединной поверхности оболочки: R_1, R_2 – главные радиусы кривизны; r – радиус оболочки в сечении, перпендикулярном оси вращения.

В общем случае элемент представляет собой произвольный четырехугольник на срединной поверхности оболочки. Для формирования определяющих характеристик элемента используется нормализованная система координат ξ, η , на которую отображается четырехугольник.

Криволинейные ортогональные координаты s, φ и произвольная осесимметричная поверхность задаются в системе координат ξ, η формулами:

$$\begin{aligned} \varphi &= \sum_{i=1}^4 L_i(\xi, \eta) \varphi_i; \\ r &= \sum_{i=1}^4 H_i(\xi, \eta) r_i + \sum_{i=1}^4 H_{1i}(\xi, \eta) \left(\frac{\partial r}{\partial \xi} \right) + \sum_{i=1}^4 H_{2i}(\xi, \eta) \left(\frac{\partial r}{\partial \eta} \right); \\ z &= \sum_{i=1}^4 H_i(\xi, \eta) z_i + \sum_{i=1}^4 H_{1i}(\xi, \eta) \left(\frac{\partial z}{\partial \xi} \right) + \sum_{i=1}^4 H_{2i}(\xi, \eta) \left(\frac{\partial z}{\partial \eta} \right). \end{aligned}$$

Здесь $L_i(\xi, \eta)$ – билинейные полиномы Лагранжа, $H_i(\xi, \eta), H_{1i}(\xi, \eta), H_{2i}(\xi, \eta)$ – кубические функции Эрмита; S_i, φ_i, r_i, z_i – узловые значения аппроксимируемых функций.

Аппроксимирующие функции имеют вид:

$$L_i = \frac{1}{4} (1 + \xi \xi_i) (1 + \eta \eta_i);$$

$$H_i = \Phi_1(\xi) \Phi_1(\eta);$$

$$H_{1i} = \Phi_1(\eta) \Phi_2(\xi);$$

$$H_{2i} = \Phi_1(\xi) \Phi_2(\eta);$$

$$\Phi_1(\lambda) = \frac{1}{4} (2 + 3\lambda_0 - 3\lambda_{30}); \Phi_2(\lambda) = \frac{1}{4} (\lambda_{30} + \lambda_{20} - \lambda_0 - 1);$$

$$\lambda_0 \rightarrow \xi_0, \eta_0; \xi_0 = \xi \xi_i; \eta_0 = \eta \eta_i; \lambda_i \rightarrow \xi_i, \eta_i$$

где ξ_i, η_i – узловые значения нормализованных координат. Для оболочки с произвольной формой меридиана главные радиусы кривизны R_1, R_2 определяются

согласно известным соотношениям дифференциальной геометрии с учетом того, что $r = r(s)$, $z = z(s)$, $dr/ds = \cos \theta$, $dz/ds = \sin \theta$ (θ – угол между осью вращения z и нормалью к срединной поверхности оболочки).

Использование аппроксимации обеспечивает плавные изменения на границах сопрягаемых элементов радиусов r , R_1 , R_2 . Производные по нормализованным координатам в узловых точках определяются по формуле

$$\left(\frac{\partial q}{\partial \lambda}\right)_i = \left(\frac{\partial q}{\partial s}\right)_i \left(\frac{\partial s}{\partial \lambda}\right), \lambda \rightarrow \xi, \eta, (i=1, \dots, 4).$$

Для оболочек канонической формы можно определять радиусы кривизны и их производные по известным зависимостям.

При формировании общей матрицы жесткости в элементах такого типа необходимо вводить фиктивный угол поворота узлов элемента вокруг нормали для приведения общей матрицы жесткости к одной размерности с матрицей преобразования координат. Для этого вводится шестое узловое перемещение – угол поворота ν_{zi} и блоки матрицы жесткости $[K_{ij}]$ расширяются по закону:

$$k_{ij}^1 = \begin{cases} \gamma Eh \Delta, i = j; \\ -\frac{1}{3} \gamma Eh \Delta, i \neq j. \end{cases}$$

так, что элемент остается самоуравновешенным в локальной системе координат. Изменение коэффициента γ в диапазоне $3 \cdot 10^{-3} \leq \gamma \leq 3 \cdot 10^{-1}$ практически не оказывает влияние на конечный результат.

Недостатками описанного элемента является низкий порядок аппроксимации тангенциальных перемещений и, как следствие, сравнительно низкая точность определения внутренних силовых факторов. К недостаткам также необходимо отнести необходимость введения фиктивного угла поворота ν_{zi} при сборке ансамбля конечных элементов.

В этом смысле более перспективными представляются конечные элементы изопараметрического типа, простейшим из которых является криволинейный четырехугольник с девятью степенями свободы в узле. В общем случае границы элемента могут не совпадать с линиями кривизны. В качестве узловых неизвестных конечного элемента принимаются девять величин – три компоненты вектора перемещения и их первые производные по координатам произвольной криволинейной системы координат

$$\{U\}^T = \left[u_1 \frac{\partial u_1}{\partial \alpha_1} \frac{\partial u_1}{\partial \alpha_2} u_2 \frac{\partial u_2}{\partial \alpha_1} \frac{\partial u_2}{\partial \alpha_2} u_3 \frac{\partial u_3}{\partial \alpha_1} \frac{\partial u_3}{\partial \alpha_2} \right].$$

Здесь u_i ($i = 1, 2, 3$) – составляющие вектора перемещений в локальной системе координат, связанной с координатными линиями.

При сборке ансамбля в узловых точках компоненты вектора перемещений одного из элементов принимаются за основные. Компоненты узлового вектора перемещений примыкающей оболочки выражаются через основные из условий сопряжения оболочек.

С повышением порядка производных в векторе обобщенных узловых перемещений выполняется последовательное повышение порядка точности аппроксимации поля перемещений. Соответственно может быть повышена точность аппроксимации геометрии оболочки.

При выборе конечных элементов стержней, составляющих единый ансамбль с базовыми конечными элементами пластин и оболочек, необходимо обеспечить идентичный выбор вектора обобщенных узловых перемещений и аппроксимирующих функций с тем, чтобы обеспечить выполнение общих требований к конечным элементам.

ИССЛЕДОВАНИЕ НАДЕЖНОСТИ СИЛЬФОННЫХ УПРУГИХ ЭЛЕМЕНТОВ КОММУТАЦИОННОЙ ТЕХНИКИ

Н.А. Рыбакова

Параметры надежности сильфонного упругого элемента – основа для расчета надежности высоковольтных вакуумных и элегазовых выключателей, одним из элементов которых он является. Основной задачей силовых выключателей является отключение токов короткого замыкания и, тем самым, защита электрооборудования от воздействий на него, вызываемых аварийными токами. Силовые выключатели должны справляться с этой задачей с высокой эффективностью и надежностью.

Одним из важных критериев оценки эффективности и надежности электрооборудования является показатель эксплуатационной надежности (ПЭН), который определяется как среднее ожидаемое число лет безаварийной работы. Обратной величиной ПЭН является частота отказов. Как показал многолетний анализ эксплуатационной надежности маломасляных, элегазовых и вакуумных выключателей, ПЭН приблизительно обратно пропорционален числу составных частей коммутационной камеры выключателя [1]. К примеру, ПЭН вакуумного выключателя ЗАН фирмы Siemens составляет около 1500 лет, а ПЭН самих вакуумных дугогасительных камер Siemens (рис.1) – 32000 лет. Частота отказов подвижных деталей, естественно, выше, чем у неподвижных деталей.

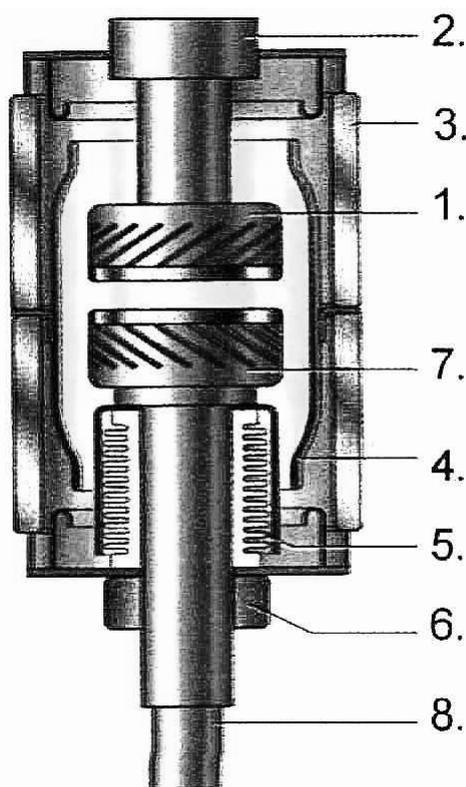


Рис. 1. Вакуумная камера: 1 – неподвижный контакт, 2 – место подключения, 3 – изолятор (керамический), 4 – коммутационная камера, 5 – металлический сильфон, 6 – направляющая шайба, 7 – подвижный контакт, 8 – резьба для присоединения к приводу

Кроме того, число составных частей выключателя влияет на номинальное число его коммутаций, которое является одним из важных характеристик коммутационных аппаратов.

Например, стандартный вакуумный выключатель фирмы Siemens на 12кВ / 31,5кА / 2500А 3АН2 может отключать ток 20кА – 250 раз, 31,5кА – 85 раз; 4,5кА – 10.000 раз; 2500А – 30.000 раз.

Механическая долговечность составляет 60.000 коммутационных циклов. Исходя из этого, вакуумные выключатели хорошо подходят для частых кратковременных отключений в сетях воздушных линий электропередач.

Надежность работы выключателя зависит от надежности работы всех его элементов. В частности, ответственным элементом коммутационной камеры вакуумного выключателя является металлический сиффон.

Сиффоны относятся к невосстанавливаемым элементам, которые после отказа не ремонтируются и подлежат замене. Отказ, проявляющийся в разгерметизации сиффонов вследствие нарушения их механической прочности, характеризует прочностную надежность сиффонов. Частичный отказ ведет к неполной потере работоспособности сиффонов, что выражается в "ухудшении" рабочих параметров сиффонов за допустимые пределы. Частичный отказ характеризует параметрическую надежность.

На стадии проектирования и конструирования показатели надежности трактуют как характеристики вероятностных математических моделей создаваемых объектов. На стадии экспериментальной отработки, испытаний и эксплуатации роль показателей надежности выполняют статистические оценки соответствующих вероятностных характеристик [2].

Долговечность сиффона определяется числом циклов, которые он может выдержать до разрушения. Разрушение сиффона вызывается напряжениями, многократно возникающими в нем на сгибе материала, и зависит от величины этих напряжений.

Циклическая прочность сиффонов (ресурсная характеристика) зависит от условий эксплуатации: рабочего хода, рабочего давления, температуры, рабочей среды, вибрационных нагрузок и т. д. Статистика показывает, что наиболее часто разрушение сиффона при его эксплуатации происходит в районе вершины или впадины гофра, так как именно здесь при долговременной работе появляются усталостные трещины. Для анализа надежности автором построена трехуровневая иерархическая структура, показывающая взаимосвязь параметров сиффонов с характеристиками надежности [3]. Нижний уровень, названный "входные параметры", учитывает геометрию, вид и условия эксплуатации сиффонов. Вторым уровнем являются средства, которые представляют собой расчет механических напряжений по математическому описанию сиффонов и расчет резонансных частот по моделям расчета присоединенных масс. В результате на третьем уровне осуществляется переход к прогнозированию ресурсных характеристик надежности: долговечности, циклической прочности и работоспособности.

Построение такой схемы объясняется тем, что при статических нагружениях ресурсными характеристиками сиффонов являются долговечность и циклическая прочность, основным показателем которых служит наработка до отказа. Наиболее рациональный путь расчета циклической прочности – определение статической связи между циклической прочностью и напряжениями в опасных точках сиффонов. Циклическая прочность зависит от геометрических и эксплуатационных параметров конкретного сиффона, а выявление опасных точек в данной работе осуществляется с помощью математического описания геометрии профиля сиффона.

При динамических условиях эксплуатации сальфонов надежность характеризует работоспособностью сальфонов. В этом случае производят расчет резонансных частот колебаний при гармонических возмущениях без демпфирования для наиболее распространенных схем закрепления сальфонов в приборах. Вопросы надежности впервые были поставлены при расчетах механических систем в связи со статическим истолкованием коэффициентов запаса и допускаемых напряжений. Основная задача теории надежности состоит в оценке показателей надежности систем по известным показателям отдельных элементов.

Для предсказания поведения деталей машин и элементов конструкций необходимо рассматривать процессы деформирования, изнашивания, накопления повреждений и разрушения при переменных нагрузках, температурах и других внешних воздействиях.

У сальфонов можно выделить три типа отказов [5]:

- приработочные, обусловленные засоренностью партии технологическим браком, например, негерметичными сальфонами или сальфонами со скрытыми дефектами;
- внезапные, возникающие вследствие перегрузки сальфонов (при отсутствии в изделии соответствующей защиты);
- износые, обусловленные необратимыми процессами, которые происходят в материале в результате усталостных и релаксационных явлений, естественного старения и др.

В теории надежности отказ трактуют как случайное событие, принимая за один из основных показателей надежности вероятность безотказной работы в течение заданного отрезка времени или в пределах заданной наработки.

Ресурс и срок службы, будучи показателями долговечности, также принадлежат к числу основных понятий теории надежности.

Для оценки долговечности служат показатели:

- средний срок службы (средняя календарная продолжительность эксплуатации изделия до момента возникновения предельного состояния или до списания),
- средний ресурс (средняя наработка изделия до предельного состояния),
- гамма-процентный ресурс (ресурс, который имеет γ процентов изделий).

Гамма-процентный ресурс N_{98} (при $\gamma = 98 \%$) сальфонов при температуре $298 \pm 10\text{K}$ ($25 \pm 10^\circ\text{C}$) определяют по номограммам, приведенным в [4]. На номограммах указана циклическая прочность в циклах для различных схем нагружения сальфонов в зависимости от отношения рабочего хода к максимальному рабочему ходу – горизонтальная ось, рабочего давления к максимальному рабочему давлению – вертикальная ось.

Средний ресурс (средняя циклическая прочность) сальфонов вычисляют по формуле

$$\lg N_{\text{cp}} = 1,25 \lg N_{98}.$$

В книге [6] приведен расчет сальфонов на малоцикловую усталость от периодических смещений, вызванных изменением температуры или перемещениями при эксплуатации. Для гибких элементов, изготовленных из упрочняющихся или разупрочняющихся материалов, максимальные деформации не остаются при заданных смещениях постоянными (хотя меняются несущественно), и условие разрушения принимает вид

$$\int_0^{N_p} \left(e_\alpha - \frac{\sigma_{-1}}{E} \right)^2 dN = \left(\frac{1}{4} \ln \frac{1}{1-\psi} \right)^2,$$

где e_α – деформация нулевого полуцикла, E – предел текучести, N – число циклов, σ_{-1} – предел усталости при базовом числе циклов N_0 , ψ – поперечное сужение при

разрыве, $\ln \frac{1}{1-\psi} = \varepsilon_B$ – истинное удлинение при разрыве (в [6] принято $\psi = 67\%$, $\sigma_{-1} = 24 \cdot 10^5$ Па).

Запасы прочности сильфонов могут определяться по перемещениям для заданного срока службы N ,

$$n_u = \frac{n_{\text{пред}}}{n_{\text{раб}}},$$

либо по числу циклов до разрушения,

$$n_N = \frac{N_{\text{пред}}}{N_{\text{раб}}},$$

для заданных перемещений.

Так, для рассмотренного выше выключателя рабочий ход контактов составляет до 5 мм при сравнительно большем перемещении сильфона. Предельное число циклов до разрушения составляет более 60000 при максимальном числе коммутации выключателя до 30000. Эти механические условия работы сильфона определяют его высокий средний срок службы и, как следствие, величину приведенного выше ПЭН дугогасительных камер.

Литература

1. Евдокунин Г.А., Тилер Г. Современная вакуумная коммутационная техника для сетей среднего напряжения (технические преимущества и эксплуатационные характеристики). СПб: Издательство Сизова М.П., 2002.
2. Болотин В.В. Ресурс машин и конструкций. М.: Машиностроение, 1990.
3. Рыбакова Н.А. Моделирование сильфонов. // Современные технологии: Сборник научных статей / Под ред. Проф. Н.А. Козлова. СПб: СПб ГИТМО (ТУ), 2001. С. 312–319.
4. ГОСТ 21482-76. Сильфоны однослойные измерительные металлические. Технические условия. М.: Издательство стандартов, 1976.
5. Сильфоны. Расчет и проектирование. / Под ред. Л.Е. Андреевой. М.: Машиностроение, 1975.
6. Серенсен С.В., Когаев В.П., Шнейдерович Р.М. Несущая способность и расчет деталей машин на прочность. / Руководство и справочное пособие. М.: Машиностроение, 1975.

МОДЕЛИРОВАНИЕ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ ТЕЛЕКОММУНИКАЦИОННОГО ОБОРУДОВАНИЯ "МАССОВЫЕ ВЫЗОВЫ"

А.П. Ищенко, К.А. Семенов

Основной задачей предприятия, разрабатывающего программное обеспечение, является своевременный выпуск нового продукта с заданной функциональностью и высоким качеством. Для этого оборудование и программное обеспечение, перед тем как быть выпущенным на рынок, проходит множественные проверки. Мировой опыт разработки программного обеспечения показывает, что этап проверки его корректной работы может занимать до половины всего времени разработки. В настоящей работе рассматриваются вопросы проверки качества программных продуктов в области телекоммуникационных сетей с коммутацией пакетов на примере систем автоматизированного тестирования телекоммуникационного оборудования (САТТО) "Массовые вызовы", разработанной и эксплуатируемой на предприятии НТЦ "Протей".

Возрастающая сложность как оборудования, так и управляющего программного обеспечения не позволяет человеку без помощи специального инструментария быстро и полностью проверить работоспособность созданной системы. Это обуславливает необходимость создания систем автоматизированного тестирования.

Существуют различные системы автоматизированного тестирования телекоммуникационного оборудования, работающего по стандарту H.323. Среди них различают:

- системы, которые осуществляют мониторинг за H.323 соединением:
 - полноценный мониторинг соединения;
 - мониторинг служебных сообщений протокола H.323;
- системы, имитирующие узел сети, работающий по стандарту H.323;
- системы создания большого трафика вызовов стандарта H.323.

САТТО относится к системам создания большого потока вызовов для проверки оборудования на стабильность под нагрузкой и получения количественных характеристик производительности системы в целом. В качестве средств реализации системы "Массовые вызовы" были выбраны языки программирования C++ и C#.

САТТО "Массовые вызовы" состоит из трех основных модулей:

- системы нагрузки или "Генератора нагрузки" – модуля генерации абонентской нагрузки;
- автоответчика или "Ответной стороны" – модуля имитации ответной стороны;
- системы обработки данных или "Генератора отчетов" – модуля преобразования статистической информации в удобный для восприятия вид.

Модуль генерации абонентской нагрузки

Управление вызовом. На рис. 1 приведена функциональная схема модуля "Генератора нагрузки". Генератор создает вызовы (звонит) на тестируемую систему. Вызов можно разделить на три логические части:

- запрос на установление соединения с удаленной стороной;
- обмен данными (не только голосовой информацией, но и служебными сообщениями) между сторонами;
- разрыв установленного соединения.

Генератор отрабатывает вызов по заданному сценарию. Под сценарием здесь понимается набор действий, выполнив которые, генератор сделает логически завершенный вызов.

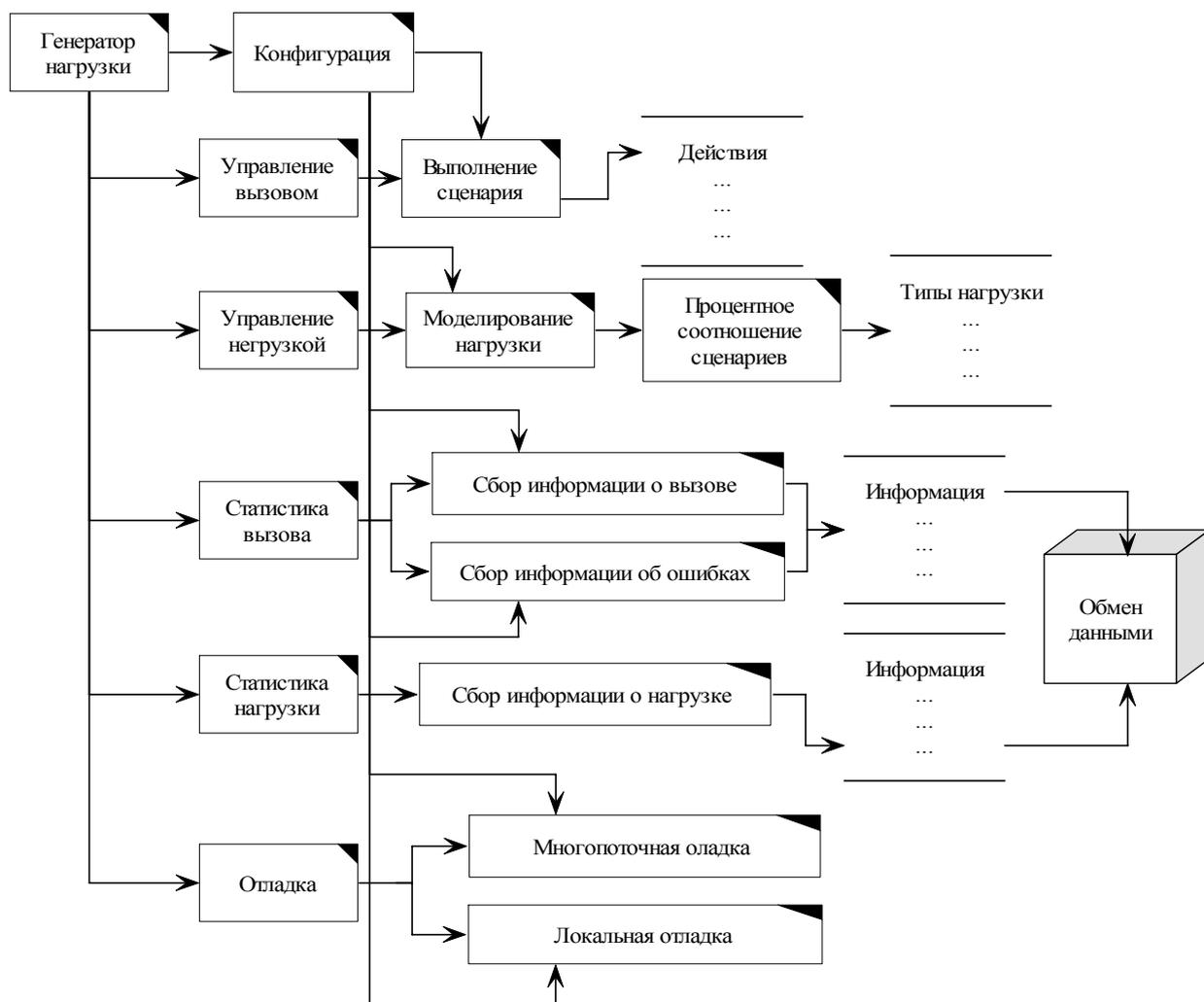


Рис. 1. Функциональная схема модуля "Генератора нагрузки"

Сценарий может содержать следующие действия (в таблицах приведены параметры каждого действия).

- **MAKE CALL** – создать соединение с удаленной стороной (позвонить).

Название параметра	Тип параметра	Описание
TimeOut	Целочисленный	Время ожидания до выполнения действия. Задается начальным значением TimeOutBase и приращением к начальному значению TimeOutStep. Значение таймаута вычисляется следующим образом: $TimeOut = TimeOutBase + TimeOutStep \cdot n$. ¹
CgPN	Целочисленный	(Calling Phone Number) – телефонный номер звонящей стороны. Задается парой параметров CgPNBase и CgPNStep.
CdPN	Целочисленный	(Called Phone Number) – телефонный номер удаленной стороны. Задается парой параметров CdPNBase и CdPNStep. ²
IpAddress	Символьный	IP-адрес удаленной стороны.
IpPort	Символьный	Порт, на который будет сделан вызов

¹ Везде, где используется задание параметра при помощи пары ...Base и ...Step, он вычисляется аналогично.

² В IP-сети каждая сторона идентифицируется IP-адресом. Параметры CgPN и CdPN используются как идентификаторы вызова.

- **WAIT_ANSWER** – ожидание ответа от удаленной стороны. Ответом в IP-сети считается получение сигнала Connect.

Название параметра	Тип параметра	Описание
TimeOut	Целочисленный	Время ожидания ответа. Если ответ не будет получен по истечению этого времени, то сценарий считается ошибочным. Задается парой параметров TimeOutBase и TimeOutStep.
Exclusive	Булевский	

- **WAIT_ALERTING** – ожидание сообщения Alerting. Alerting или сигнал КПВ (контроль посылки вызова) означает, что ответная сторона начала обрабатывать поступивший к ней вызов.

Название параметра	Тип параметра	Описание
TimeOut	Целочисленный	Время ожидания сигнала КПВ. По истечению этого времени, если сигнал КПВ не был получен, сценарий считается ошибочным. Задается парой параметров: TimeOutBase и TimeOutStep.

- **DTMF_RX** – принять DTMF, т.е. ожидать от удаленной стороны передачи DTMF-сигнала.

Название параметра	Тип параметра	Описание
TimeOut	Целочисленный	Время, в течении которого ожидается передача DTMF-сигнала. Если по истечению этого времени сигнал не будет получен, то сценарий считается ошибочным.
DTMFMask	Символьный	Маска для проверки полученного DTMF-сигнала. Если DTMF-сигнал не удовлетворяет условию, заданному в маске, то сценарий считается ошибочным.

- **DTMF_TX** – передать DTMF-сигнал.

Название параметра	Тип параметра	Описание
TimeOut	Целочисленный	Задержка перед передачей DTMF-сигнала.
DTMF	Символьный	Значение DTMF-сигнала. Задается парой DTMFBase и DTMFStep. Могут использоваться любые символы. Если строка DTMF не является числом в шестнадцатичном виде, то параметр DTMFStep игнорируется и посылается DTMF-сигнал со значением DTMFBase; иначе генератор выполняет увеличение DTMF-сигнала на DTMFStep. Пример: DTMFBase = 12A; DTMFStep = 1. В этом случае будет произведено сложение Base со Step; если задать значение DTMFBase = 12A* то сложение производиться не будет.

- **RELEASE_CALL** – разорвать соединение (отбиться).

Название параметра	Тип параметра	Описание
TimeOut	Целочисленный	Задержка перед разрывом соединения.

- **WAIT_RELEASE** – ожидать от удаленной стороны разрыва соединения.

Название параметра	Тип параметра	Описание
TimeOut	Целочисленный	Задается парой TimeOutMin и TimeOutMax. TimeOutMin и TimeOutMax задают промежуток времени ³ (соответственно нижнюю и верхнюю границу промежутка), в течение которого генератор ожидает разрыва от удаленной стороны. Если разрыв произошел раньше TimeOutMin или позже TimeOutMin + TimeOutMax, то сценарий считается ошибочным.

³ TimeOutMin должен быть меньше TimeOutMax.

Действия сценария делятся на обязательные и необязательные. Обязательные действия – это те, которые присутствуют в каждом сценарии. Обязательными является действие MAKE_CALL; если оно не указано, то сценарий вообще не сможет быть выполнен. Генератор должен также корректно следить за окончанием вызова, следовательно, необходимо заканчивать сценарий действиями RELEASE или WAIT_RELEASE.⁴ Остальные действия могут присутствовать по необходимости.

Сценарий задается в конфигурационном файле. Количество сценариев не ограничено. Генератор выполняет их по порядку следования в файле конфигурации, но при этом придерживается заданного процентного соотношения сценариев.

Управление нагрузкой. В функцию управления нагрузкой включаются такие возможности, как:

- создание нагрузки заданной величины (определяемой количеством вызовов в секунду);
- создание нагрузки заданного типа;
- обеспечения заданного процентного соотношения сценариев.

Создание нагрузки заданной величины. Так как телекоммуникационное оборудование призвано обслуживать огромное количество людей, то количество вызовов (трафик), обрабатываемых этим оборудованием, может достигать высоких величин. Следовательно, генератор, для имитации различного рода "стрессовых" ситуаций, должен уметь создавать высокую плотность вызовов, а также уметь создать большое количество одновременно существующих соединений.

Параметры для каждого типа нагрузки задаются из файла конфигурации. Максимальное количество одновременно созданных соединений задается параметром MaxAllocatedHandlers и может быть любым.

Система поддерживает три варианта (или типа) нагрузки:

- линейная нагрузка;
- распределение Пуассона при линейном уровне нагрузки;
- распределение Пуассона при распределении нагрузки волной.

Тип нагрузки, который будет использоваться, задается в конфигурационном файле параметром TrafficType.

Общий вид линейной нагрузки приведен на рис. 2, где I – нагрузка, t – текущее время, ΔT – время работы генератора. Как видно, генератор поддерживает заданный уровень интенсивности вызовов в течение всего времени работы системы. Интервал между созданием вызовов также не меняется в течение всего времени работы системы.

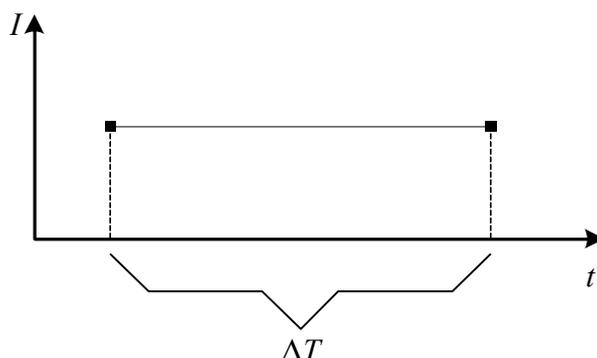


Рис. 2. График линейной нагрузки

Уровень линейной нагрузки указывается параметром Level.

⁴ Если в сценарии не указать, как разрывать соединение, то генератор будет бесконечно ожидать завершения связи от удаленной стороны.

Для реализации варианта нагрузки с распределением Пуассона при линейном уровне система создает нагрузку, наиболее приближенную к реально существующей в ТфОП, так как при реальной эксплуатации системы вызовы поступают со случайным интервалом X , который подчиняется распределению Пуассона:

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda},$$

где λ – параметр распределения. Как показано на рис. 3, при таком распределении интенсивность поступления вызовов будет случайным образом колебаться около величины λ , где I – интенсивность, t – время, λ – уровень интенсивности, R_i – реально генерируемая интенсивность.

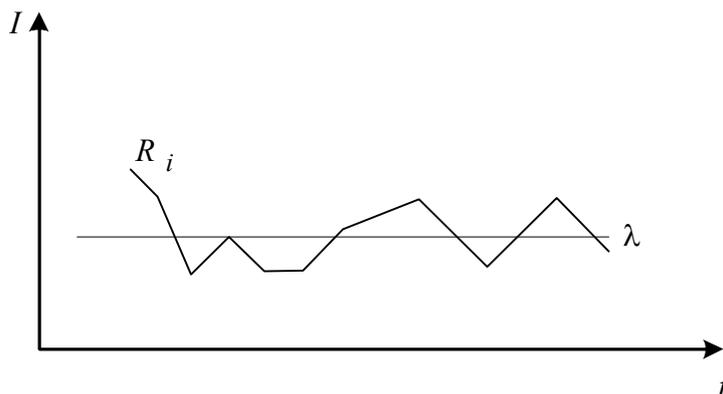


Рис. 3. Распределение Пуассона при линейном уровне нагрузки

Возможны также ситуации, когда нагрузка, в рамках распределения Пуассона, изменяется волнообразно. На рис. 4 изображен график распределения нагрузки волной.

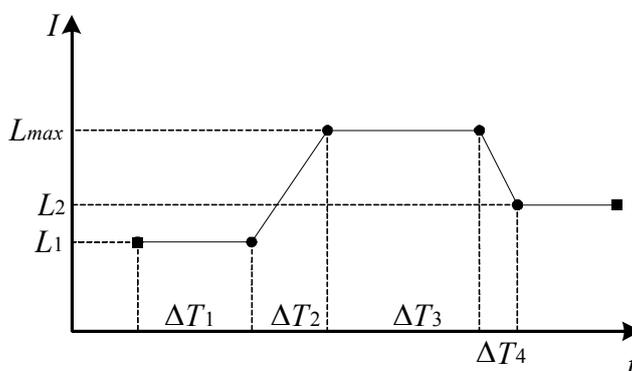


Рис. 4. График распределения нагрузки волной

Для реализации варианта нагрузки с распределением Пуассона при распределении нагрузки волной система моделирует волну нагрузки. Сначала, в течение времени ΔT_1 , уровень интенсивности вызовов равен L_1 . Далее, в течение времени ΔT_2 , происходит возрастание нагрузки до уровня L_{max} . Система поддерживает максимальную нагрузку в течение времени ΔT_3 , после чего, в течение времени ΔT_4 , происходит уменьшение интенсивности нагрузки до уровня L_2 , и этот уровень поддерживается до завершения работы. Под уровнем нагрузки (интенсивностью) понимается значение параметра λ . Увеличение и снижение уровня нагрузки происходит линейно (рис. 4).

Форма волны задается из конфигурационного файла набором параметров.

Название параметра	Тип параметра	Описание
StartLevel	Целочисленный	Начальный уровень интенсивности вызовов (L_1).
StartTimeOut	Целочисленный	Время, в течение которого держится интенсивность, заданная в StartLevel (ΔT_1).
MaxLevel	Целочисленный	Максимальный уровень интенсивности (L_{Max}).
TimeOutUP	Целочисленный	Время, в течение которого увеличивается интенсивность от StartLevel до MaxLevel (ΔT_2).
MaxTimeOut	Целочисленный	В течение этого времени поддерживается уровень интенсивности, равный MaxLevel (ΔT_3).
EndLevel	Целочисленный	Конечный уровень интенсивности (L_2).
TimeOutDown	Целочисленный	Интервал времени, в течение которого уровень интенсивности понижается до EndLevel (ΔT_4).

Статистика вызова. Функция мониторинга вызова включает в себя:

- сбор подробной информации о вызове;
- определение и сигнализация об ошибочных ситуациях.

Для сбора информации о вызове генератор должен предоставлять подробную информацию о каждом вызове и о ходе выполнения сценария. Информация, предоставляемая генератором о вызове, делится на три типа:

- общая информация;
- информация об ошибках;
- статистика в формате "AdvLog".

Общая информация – подробная информация о прохождении сценария, о приеме и послышке служебных сигналов и возникших ошибках с указанием времени каждого события. Информация выводится в файл в формате, указанном на рис. 5.

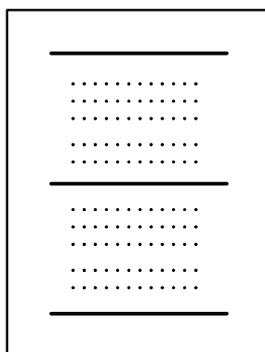


Рис. 5. Формат общей информации в файле

В пределах разделителей⁵ (жирные полосы на рисунке) все записи относятся к одному вызову, что позволяет удобно следить за ходом выполнения сценария и прохождением служебных сигналов.

Формат записи должен быть следующим:

[время] событие

В квадратных скобках указывается время события, которое описывается строкой. Формат времени имеет вид

⁵ Далее набор записей внутри таких разделителей называется блоком.

Ч. М.С. мС ,

где Ч – часы, М – минуты, С – секунды, мС – миллисекунды.

Вызовы в файле могут проходить не по порядку, следовательно, в блоке нужно выводить информацию, идентифицирующую каждый вызов, например CallReference. Ниже приведен упрощенный пример блока:

```
[10.00.00.000] Make call to 192.168.100.10 scenario 1
[10.00.00.010] Wait Answer
[10.00.00.100] Receive answer
[10.00.01.100] Send DTMF
[10.00.02.100] Release Call
[10.00.02.300] Scenario Done
```

Блок должен содержать в себе, помимо указания всех действий сценария, информацию о приеме и передаче следующих служебных сигналов и действий:

- Setup PDU;
- Alerting PDU;
- CapabilitySet PDU;
- Connect PDU;
- Open Receiver Channel;
- Open Sender Channel;
- Close Receiver Channel;
- Close Sender Channel.

Здесь также должны быть указаны ошибки при выполнении действий сценария, например:

```
[10.00.00.000] Make call to 192.168.100.10 scenario 1
[10.00.00.010] Wait Answer
[10.00.01.010] Error: No answer
[10.00.01.110] Release Call
[10.00.01.310] Scenario failure
```

Этот вид информации позволяет:

- удобно и достаточно подробно проследить ход выполнения каждого сценария;
- удобно и достаточно подробно проследить ход выполнения каждого вызова;
- легко определить место возникновения ошибки в ходе выполнения сценария;

Информация об ошибках, возникших в ходе выполнения сценария и о системных ошибках в процессе выполнения вызова, записывается в отдельный файл. Формат записи сообщения об ошибке следующий:

<<Время>> <<Сценарий>> <<Объект⁶>> <<Сообщение об ошибке>>

В сообщении об ошибке должна присутствовать информация, идентифицирующая вызов (CallReference). Ниже приведен пример записей об ошибках:

```
10.33.49.681 1 0 Er2:(Wait Answer) Expire Time Out. T:
ip$localhost/7899
10.33.52.681 1 1 Er2:(Wait Answer) Expire Time Out. T:
ip$localhost/7901
```

Здесь в качестве идентификатора вызова используется CallReference: "ip\$localhost/7899". Вызов идентифицируется числом, записанным после '/', в данном случае это "7899". Это число уникально в течение всего времени работы системы.

Этот вид информации позволяет:

- определить наличие ошибок в процессе работы системы;
- легко определить, какие возникли ошибки во время работы системы;

⁶ Объект – функциональная единица системы, умеющая выполнять сценарий.

- определить, в каком вызове и сценарии, а также в какое время возникла та или иная ошибка.

Статистика в формате "AdvLog" (Advanced Log). Этот тип логов, создаваемых системой, предназначен не для пользователя, а для "Генератора Отчетов". В логе этого типа каждый вызов записан строкой, состоящей из времен – времени старта сценария и времени окончания каждого действия. По данным, записываемым в этом логе, "Генератор отчетов" может строить разнообразные отчеты. Создание этого лога можно включить или выключить для каждого сценария в отдельности. Для каждого сценария лог пишется в отдельный файл с именем:

scen<<номер сценария>>.log

В каждом файле содержатся наборы строк. Каждая строка соответствует одному вызову. В начале каждой строки записывается флаг успешности⁷ или неуспешности выполнения данного сценария. Флаг принимает значение 1, если сценарий был успешен, и значение 0, если неуспешен. В каждой строке, для идентификации конкретного вызова в системе, пишется CgPN и CdPN⁸. Эти значения указываются после разделителя '|'.
 Формат строки в таком файле следующий:

<<1/0>> <<время начала сценария>> <<время окончания действия 1 >> ... <<время
 окончания действия N>> | <<CgPN>> <<CdPN>>

Количество полей в строке зависит от количества действий в сценарии и равно [количество действий]+2. Количество полей в пределах одного сценария постоянно.

Если сценарий неуспешен, то все поля, начиная с действия, где произошла ошибка, и до конца, записываются временем возникновения ошибки. В полях указывается относительное время, т.е. время в миллисекундах, прошедшее с момента старта системы.

Ниже приведены примеры строк успешного и неуспешного выполнения одного сценария:

1 10000 10010 11010 14010 | 111111 222230 – успешный сценарий
 0 10000 10010 **11010 11010** | 111121 222240 – неуспешный сценарий, сбой на втором действии

Этот вид информации при обработке специальной программой (здесь – "Генератор отчетов") позволяет:

- получить процентное соотношение успешных и неуспешных сценариев на любом отрезке времени работы системы;
- получить количественное соотношение успешных и неуспешных сценариев на любом отрезке времени работы системы;
- измерить уровень быстродействия системы;
- измерить пропускную способность канала связи;
- измерить среднюю длительность выполнения одного сценария;
- измерить длительность выполнения любого действия в любом вызове;
- измерить интенсивность вызовов каждого сценария в отдельности;
- измерить длительность любого вызова;
- исследовать удаленную систему на задержки;
- в любом вызове любого сценария определить, на каком действии произошел сбой;
- получить количественную характеристику "сбойности" того или иного действия любого сценария.

⁷ Т.е. не возникло ошибок в ходе выполнения сценария.

⁸ Для идентификации вызова с помощью CgPn и CdPN нужно, чтобы в конфигурации значения CgPNStep и CdPNStep были не равны 0. Тогда при каждом вызове значение этих параметров будут новым.

Статистика нагрузки. Генератор предоставляет информацию о генерируемой нагрузке и о процентном соотношении сценариев. Реально создаваемая нагрузка не всегда может соответствовать заданному в конфигурации уровню. Так как при вызове большой длины⁹ генератор будет для поддержания указанного уровня нагрузки создавать новые вызовы, то в системе начнет возрастать количество открытых соединений (вызовов), и, когда оно достигнет максимально допустимого значения, произойдет спад нагрузки до некоторого уровня. Следовательно, необходимо следить за нагрузкой в режиме реального времени.

В конфигурации интервал времени, через который системе необходимо предоставлять информацию о нагрузке, задается параметром DropInformationInterval, в котором указано время в миллисекундах. Информация записывается в файл info.log с заданным интервалом времени.

Здесь также указывается расчетное процентное соотношение сценариев, которое задается в конфигурации. Параметр ScenarioCount определяет количество сценариев, выполняемых системой. Процентное соотношение задается как

`<<№Scen>> = %`

где № – порядковый номер сценария. Процентные соотношения сценариев в сумме должны равняться 100%.

Пример конфигурации процентного соотношения для трех сценариев:

`ScenarioCount = 3`

`1Scen = 30`

`2Scen = 40`

`3Scen = 30`

Генератор выполняет сценарии таким образом, чтобы процентное соотношение каждого сценария в общем количестве вызовов оставалось заданным.

Генератор предоставляет следующую информацию о нагрузке.

- Сценарии:
 - общее количество вызовов данного сценария;
 - количество успешных вызовов данного сценария;
 - процент успешных вызовов данного сценария от общего количества вызовов данного сценария;
 - количество неуспешных вызовов данного сценария;
 - процент неуспешных вызовов данного сценария от общего количества вызовов данного сценария;
 - процент вызовов данного сценария в общем количестве вызовов всех сценариев.
- Реально создаваемая нагрузка (вызов/секунда)
- Время:
 - время работы системы;
 - текущее время и дата.
- "Last Time Statistic" – статистика за период времени между записями информации в файл:
 - количество удачно выполненных сценариев;
 - количество неудачно выполненных сценариев;
 - расчетная нагрузка, созданная за данный интервал времени (вызов/секунда);
 - количество свободных мест для создания соединений.

Шаблон для предоставления информации следующий:

"№ сценария" "общее количество вызовов" "успешных" "неуспешных" "%"

¹⁰ Имеется в виду длительность вызова во времени.

"№ сценария" "общее количество вызовов" "успешных" "неуспешных" "%"

...

[Сценарии]

"Общее количество вызовов": "количество"

"Успешных вызовов": "количество", "%"

"Неуспешных вызовов": "количество", "%"

"Нагрузка"

[Время]

"Время работы системы"

"Время текущее"

"Дата текущая"

[Last time statistic]

"Успешных вызовов": "количество"

"Неуспешных вызовов": "количество"

"Нагрузка"

"Количество свободных мест для создания соединений"

Эта информация позволяет отследить:

- изменение и уровень реально создаваемой нагрузки;
- количество существующих в системе соединений;
- количество оставшихся соединений;
- реально создаваемое процентное соотношение сценариев.

Отладка. Генератор использует функции трассировки, предоставляемые библиотекой OpenN323. Трассировка позволяет получить подробнейшую информацию о работе системы, отслеживать все передачи данных по сети (прием и отправку служебных сообщений) и между функциями, а также вызовы функций и обработку алгоритмов с указанием потоков исполнения.

Трассировка может отключаться в файле конфигурации. Параметр N323Trace = 0 означает, что трассировка отключена, а если он равен 1, то трассировка будет включена.

В конфигурации также задается уровень трассировки, т.е. степень подробности информации, выдаваемой в отдельный файл N323Trace.log. Уровень имеет значения от 1 до 4 (самый подробный) и задается из конфигурации параметром N323TraceLevel.

Так как функции трассировки выводят информацию еще и о работе самой библиотеки, то генератор может выводить на экран собственную трассировку для отдельного отслеживания собственных алгоритмов. Эта функция также является отключаемой и задается параметром Debug. Значения параметра: 1 – включено, 0 – выключено.

Различные типы трассировок помогают в локализации и устранении ошибок.

Конфигурирование генератора. Конфигурирование генератора осуществляется через специальный файл, записываемый в текстовом режиме в формате профайла. Профайл – это формат, где параметры разделены по секциям, и для получения того или иного значения параметра нужно указать имя секции и имя параметра. По сути, это двумерный массив, итерация по которому выполняется при помощи двух индексов – имени секции и имени параметра. Считывание происходит сверху вниз. Имя секции указывается в квадратных скобках '[' и ']'. Параметры, относящиеся к этой секции, записываются ниже, каждый параметр на отдельной строке. Имена параметров в пределах секции и имена секций должны быть уникальными, например:

```
[Section 1]
```

```
param1 = 10
```

– секция с параметрами

```
param2 = 20
```

```
[Section 2]
```

```
param1 = 30
```

– другая секция с параметрами

```
param2 = 40
```

Имена параметров и названия секций могут быть любыми. Присвоение значения параметров записываются через '='. Например, Level = 3. Формат профайла предусматривает наличие комментариев. Если строка начинается с символа '#' то она игнорируется и считается комментарием.

Конфигурация считывается при запуске генератора. Ниже представлены параметры, задаваемые в файле конфигурации, с пояснениями к ним.

[Common]

MaxAllocatedHandlers – максимальное количество одновременно открытых соединений в системе.

TrafficType – тип генерируемой нагрузки: 'L' – линейная, 'P' – распределение Пуассона с линейной интенсивностью, 'W' – распределение Пуассона с созданием волны интенсивности.

PoissonLambda – значение параметра λ в распределении Пуассона.

Level – уровень линейной интенсивности.

StartLevel

StartTimeOut

MaxLevel

TimeOutUp

MaxTimeOut

EndLevel

TimeOutDown

– параметры, задающие форму волны нагрузки.

ScenarioCount – количество сценариев.

<<№Scen>>¹⁰ – указывает процентное соотношение сценариев; № – номер сценария.

SuspendInterval – задержка перед началом генерации нагрузки.

[Statistics]

DropInformationInterval – временной интервал сброса статистики нагрузки.

[Log]

H323Trace – включить (1)/выключить (0) трассировку H323.

H323TraceLevel – уровень трассировки H323, при выключенной трассировке не используется.

Debug – включить (1)/выключить (0) дополнительный отладочный вывод генератора.

[ScenarioTemplate<номер сценария>]

Количество секций зависит от количества сценариев: каждый сценарий – это отдельная секция. В названии секции указывается номер сценария. Все параметры относятся только к данному сценарию.

Name – название сценария, для работы системы не используется.

AdvLog – включить (1)/выключить (0) для данного сценария запись лога "AdvLog".

IterationsLimit – количество повторов данного сценария. Каждое выполнение сценария (создание вызова с этим сценарием) уменьшает параметр на 1. Как только параметр достигает 0, сценарий перестает выполняться.

Manifold – многообразие сценариев, т.е. количество раз, когда система может увеличивать на Step пару Base-Step любых параметров данного сценария. По достижении заданного количества увеличений параметра происходит сбрасывание значения параметра на начальное значение Base.

ActionsCount – количество действий в сценарии.

<<№ActionType>> – тип действия (указывается код). Номер – номер действия по счету. Например, запись 2ActionType значит, что это второе действие в сценарии.

Соответствие номеров действиям приведено ниже:

1 – MAKE_CALL

¹⁰ Заключение параметра в '<<' '>>' означает, что его название может изменяться в зависимости от других параметров, и количество этих параметров тоже варьируется.

- 2 – WAIT_ANSWER
- 3 – WAIT_ALLERTING
- 5 – DTMF_RX
- 6 – DTMF_TX
- 7 – RELEASE_CALL
- 8 – WAIT_RELEASE

Далее записываются параметры, задающие действие (см. выше):

<№название параметра> = <значение параметра>,

где № – это порядковый номер действия в сценарии. Например, 2TimeOut означает, что параметр TimeOut относится ко второму действию.

Ниже приведен пример файла конфигурации:

```
# Protei_MV_ip configuration file.
# Call generator side.
#
# file :      Config/generator.cfg
# 20.02.2003

[Common]
MaxAllocatedHandlers = 30
TrafficType = 1
PoissonLambda = 3

# number of scenario templates
ScenarioCount = 2

# percents relations
1Scen = 70
2Scen = 30

# wait for signaling start up
SuspendInterval = 3000

[Statistics]
DropInformationInterval = 3000

[Log]
H323Trace = 0
H323TraceLevel =1
Debug =1

[ScenarioTemplate1]
Name = Call, Wait Answer, Receive DTMF, Wait Release
AdvLog = 1
IterationsLimit = 10000
Manifold = 200
ActionsCount = 4

1ActionType = 1
1TimeOutBase = 10
1TimeOutStep = 0
1CgPNBase = 111111
1CgPNStep = 0
1CdPNBase = 111111
1CdPNStep = 0
1IpAddress=192.168.100.27
1IpPort=1800

2ActionType = 2
2TimeOutBase = 1000
```

```

2TimeOutStep = 0
2Exclusive = 0

3ActionType = 5
3TimeOut=5000
3DTMFMask=.(3)

4ActionType = 8
4TimeOutMin = 8000
4TimeOutMax = 12000

[ScenarioTemplate2]
Name = Call, Wait Answer, Release
AdvLog = 1
IterationsLimit = 10000
Manifold = 200
ActionsCount = 3

1ActionType = 1
1TimeOutBase = 10
1TimeOutStep = 0
1CgPNBase = 111111
1CgPNStep = 0
1CdPNBase = 111111
1CdPNStep = 0
1IpAddress=192.168.100.27
1IpPort=1800

2ActionType = 2
2TimeOutBase = 1000
2TimeOutStep = 0
2Exclusive = 0

4ActionType = 7
4TimeOut = 3000

```

В приведенном примере заданы два сценария. Первый сценарий содержит действия: MAKE_CALL, WAIT_ANSWER, DTMF_RX, WAIT_RELEASE. Второй сценарий состоит из: MAKE_CALL, WAIT_ANSWER, RELEASE.

Управление ответом

На рис. 6 приведена функциональная схема "Автоответчика". "Автоответчик" отвечает на входящий вызов, выполняя сценарий, заданный в конфигурации, который выбирается в соответствии с номером CgPN пришедшего вызова. Если для номера CgPN поступившего вызова не задан ни один сценарий, вызов не принимается. Номер CgPN, для которого будет выполнен данный сценарий, задается для каждого сценария отдельно параметром Alias. Сценарием для ответной стороны является набор действий, выполнив которые, "Автоответчик" обслужит поступивший вызов.

Ниже приведены действия, которые может выполнять "Автоответчик".

- **ANSWER** – ответить на вызов, т.е. послать сигнал Connect.

Название параметра	Тип параметра	Описание
TimeOut	Целочисленный	Временная задержка перед ответом.

- **DTMF_TX** – передать DTMF сигнал.

Название параметра	Тип параметра	Описание
TimeOut	Целочисленный	Временная задержка перед передачей DTMF сигнала.
DTMFBase	Символьный	Значение DTMF сигнала, который нужно послать

		удаленной стороне.
--	--	--------------------

- **RELEASE_CALL** – разорвать соединение.

Название параметра	Тип параметра	Описание
TimeOut	Целочисленный	Задержка перед разрывом соединения.

- **WAIT_RELEASE** – ожидать от удаленной стороны разрыва соединения.

Название параметра	Тип параметра	Описание
TimeOut	Целочисленный	Задается парой TimeOutMin и TimeOutMax. TimeOutMin и TimeOutMax задают промежуток времени ¹¹ (соответственно нижнюю и верхнюю границу промежутка) в течении которого "Автоответчик" ожидает разрыва от удаленной стороны. Если разрыв произошел раньше TimeOutMin или позже TimeOutMin + TimeOutMax, то сценарий считается ошибочным.

Статистика ответа. В отличие от "генератора нагрузки", "автоответчик" предоставляет лишь два типа информации о вызове:

- общая информация;
- информация об ошибках.

Оба типа логов, которые создает "Ответная сторона", полностью аналогичны соответствующим типам информации "Генератора".



Рис. 6. Функциональная схема "Автоответчика"

Статистика нагрузки. "Автоответчик" – пассивная сторона и сам не создает нагрузку, следовательно, он должен уметь измерять нагрузку, которая на него приходит. Принцип предоставления информации о нагрузке аналогичен "Генератору". Информация "сбрасывается" через промежуток времени, заданный в конфигурации DropInformationInterval в миллисекундах. Информация, которую предоставляет

¹¹ TimeOutMin должен быть меньше TimeOutMax

"Автоответчик" о нагрузке, – такая же что и у "Генератора", за исключением добавления двух параметров:

- поступившие вызовы;
- принятые вызовы.

Так как "Автоответчик" обслуживает только те вызовы, для которых заданы сценарии, а остальные вызовы не принимает, то в статистике нагрузки указывается общее количество вызовов, поступивших на "Автоответчик" (поступившие вызовы), и количество вызовов, обслуженных ответной стороной (принятые вызовы). В общем виде информация о нагрузке выглядит так:

"№ сценария"	"общее количество вызовов"	"успешных"	"неуспешных"	"%"
"№ сценария"	"общее количество вызовов"	"успешных"	"неуспешных"	"%" ...

[Вызовы]

"Поступившие вызовы": "количество"

"Принятые вызовы": "количество"

[Сценарии]

"Общее количество вызовов": "количество"

"Успешных вызовов": "количество", "%"

"Неуспешных вызовов": "количество", "%"

"Нагрузка"

[Время]

"Время работы системы"

"Время текущее"

"Дата текущая"

[Last time statistic]

"Успешных вызовов": "количество"

"Неуспешных вызовов": "количество"

"Нагрузка"

"Количество свободных мест для создания соединений"

В остальном информация о нагрузке, предоставляемая "Автоответчиком", аналогична информации о нагрузке предоставляемой "Генератором". Система отладки полностью аналогична системе отладки "Генератора нагрузки". Конфигурирование "Автоответчика" происходит аналогично "Генератору". Конфигурация и сценарии хранятся в профайле.

Ниже приведены параметры профайла конфигурации "Автоответчика".

[Common]

MaxAllocatedHandlers – количество одновременно открытых соединений.

IpPort – IP-порт, на котором "Автоответчик" принимает вызовы.

ScenarioCount – количество сценариев.

[Log]

H323Trace – включить(1)/выключить(0) трассировку H323.

H323TraceLevel – уровень трассировки H323. При выключенной трассировке не используется.

Debug – включить (1)/выключить (0) дополнительный отладочный вывод генератора.

[Statistic]

DropInformationInterval – интервал, через который "сбрасывать" статистику о нагрузке.

Далее следуют секции сценариев.

[ScenarioTemplate<номер сценария>] – в названии секции указывается номер сценария.

Все параметры относятся только к данному сценарию.

Name – название сценария, для работы системы не используется.

Alias – номер CgPN, для которого выполняться данный сценарий.

ActionsCount – количество действий в сценарии.

<<№ActionType>> – тип действия. Номер – номер действия по счету.

- 1 – ANSWER
- 2 – DTMF_TX
- 3 – RELEASE_CALL
- 4 – WAIT_RELEASE

Далее, как и в "Генераторе", записываются параметры, задающие действие.

Пример файла конфигурации "Автоответчика" с двумя сценариями представлен ниже.

```
[Common]
MaxAllocatedHandlers = 30
IpPort = 1720
ScenarioCount = 2

[Log]
H323Trace = 0
H323TraceLevel = 1
Debug = 0

[Statistics]
DropInformationInterval = 3000

[ScenarioTemplate1]
Name = Answer, Send DTMF, Wait Release
Alias = 160
ActionsCount = 3

1ActionType = 1
1TimeOut = 100

2ActionType = 2
2TimeOut = 7000
2DTMFBase= 321

3ActionType = 4
3TimeOutMax = 4000
3TimeOutMin = 10000

[ScenarioTemplate2]
Name = Answer, Release
Alias = 163
ActionsCount = 2

1ActionType = 1
1TimeOut = 100

2ActionType = 3
2TimeOut = 4000
```

В конфигурации заданы два сценария: первый будет выполняться для поступившего вызова с CgPN = 160, а второй – для вызова с CgPN = 163. Все остальные вызовы не будут обслуживаться. Первый сценарий состоит из действий ANSWER, DTMF_TX, WAIT_RELEASE. Второй сценарий содержит два действия: ANSWER, RELEASE.

Генератор отчетов

Под отчетом понимается информация, полученная в результате тестирования, обработанная и представленная в удобном виде. "Генератор отчетов" предназначен для обработки данных тестирования и предоставления информации о прошедшем тесте в удобной форме.

На рис. 7 представлена функциональная модель "Генератора отчетов".

Интеграция с O&M системой. Доступ к "генератору отчетов" осуществляется через систему O&M (Operate&Maintains)). Система O&M помогает в управлении проектами, ведет контроль за ходом выполнения проектов и т.д. Управление "генератору отчетов" передается по ссылке, расположенной на Web-странице системы O&M. Кроме того, система O&M передает "генератору отчетов" путь к репозиторию (каталогу) текущего пользователя, откуда и куда последний сможет сохранять и читать шаблоны отчетов.

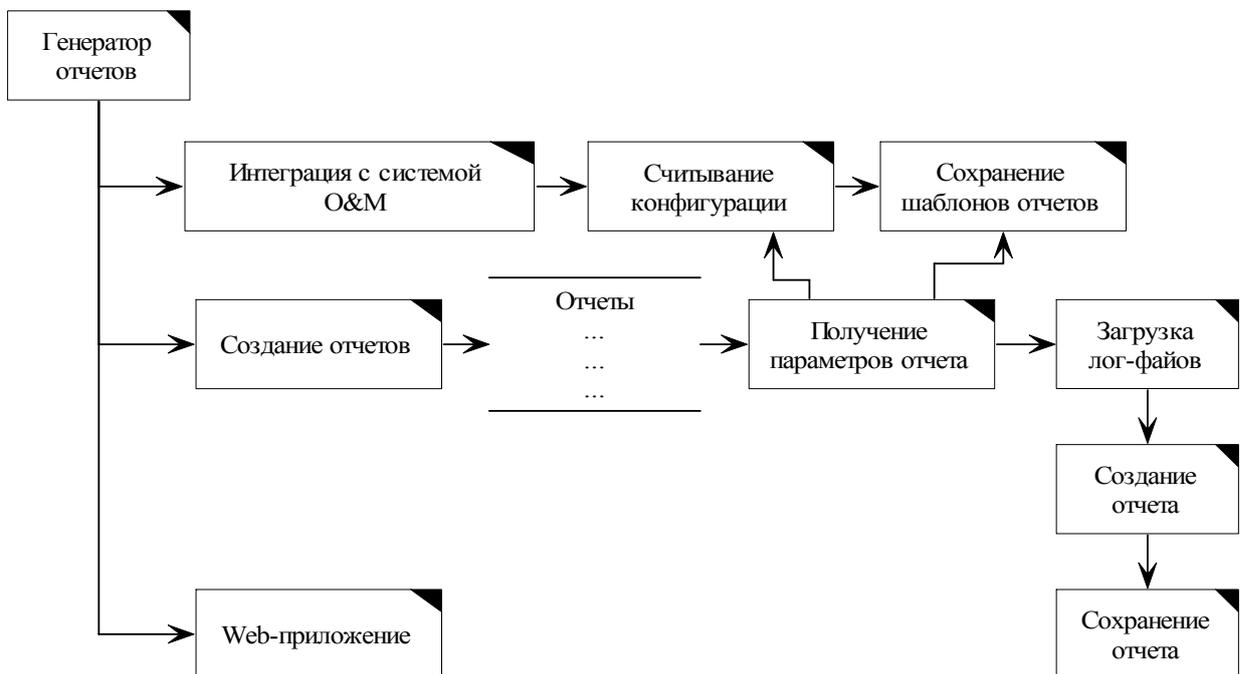


Рис. 7. Функциональная модель "Генератора отчетов"

Отчеты. "Генератор отчетов" анализирует лог-файл в формате "AdvLog" и предоставляет информацию о работе системы в удобной форме. Типы отчетов, которые может создавать система:

- временной интервал между действиями;
- ошибки по сценариям;
- ошибки по действиям;
- количество ошибок.

Отчет представляет собою график, по одной оси которого откладывается интервал времени между действиями, а по другой – абсолютное время работы системы. Отчет строится только по данным успешно выполненных сценариев, неуспешно выполненные сценарии игнорируются. На рис. 8 изображена схема такого отчета.

Параметрами отчета являются:

- сценарий, по которому строится отчет;
- номера действий в сценарии, между которыми вычисляется интервал. Номера задаются по порядку, как они записаны в лог-файле, например:

	1 действие	2 действие	3 действие	4 действие
1	10000	10100	11010	12130

- интервал времени работы системы, на котором строиться отчет.
Система умеет изменять масштаб по обеим осям. Масштаб изменяется заданием начальных и конечных значений каждой оси.
Также система выводит средний интервал между действиями и его дисперсию.



Рис. 8. Зависимость интервала между действиями от времени работы системы

Ошибки по сценариям. Этот вид отчета представляет собой набор таблиц, в которых приводятся количественные характеристики "ошибочности" сценариев. Для каждого сценария создается отдельная таблица, где для каждого действия в отдельности указывается количество сбоев на этом действии и процент от успешных.. Таблица может выглядеть следующим образом:

название сценария		
действие	кол-во сбоев	процент от успешных

Параметрами являются сценарии, по которым строится отчет. Сценарии задаются порядковыми номерами.

Ошибки по действиям. Этот вид отчета аналогичен предыдущему, но вместо сценариев в нем действия, а вместо действий – сценарии. Отчет может строиться для нескольких действий сразу, при этом для каждого действия будет построена отдельная таблица. Таблица в отчете этого типа будет иметь вид:

Название действия		
сценарий	кол-во сбоев	процент от успешных

Параметрами отчета являются действия. Задается действие псевдонимом.

Количество ошибок. Этот отчет представляет собой гистограмму по времени. Высота столбца на гистограмме вычисляется подсчетом количества сбоев на заданном действии в течение некоторого промежутка времени, который определяет ширину столбца. На одном графике возможно размещение гистограмм сразу нескольких действий. На рис. 9 приведен пример такого отчета.

Номера действий и ширина столбца задаются как параметры отчета. Столбцы, обозначающие различные действия, должны различаться по цвету.

Параметры отчета. Система запрашивает некоторые параметры для построения отчета у пользователя, а некоторые считывает сама из конфигурации "генератора нагрузки". После того как параметры заданы, их можно сохранить как шаблон отчета для использования в будущем.

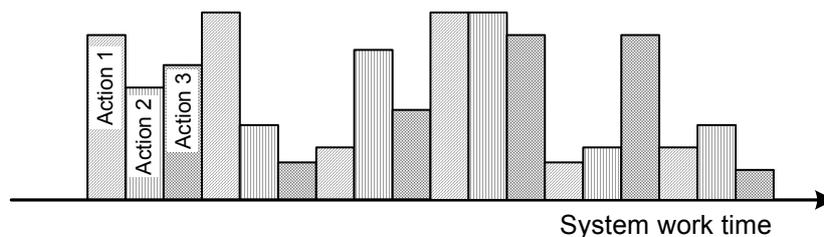


Рис. 9. Пример гистограммы ошибок на отдельном действии в заданный промежуток времени

Считывание лог-файла. Лог-файл "генератор отчетов" считывает по FTP-протоколу с компьютера, на котором был запущен "генератор нагрузки". Кроме того, считывается конфигурация "генератора нагрузки", из которой автоматически извлекаются информация о прошедшем тестировании (например, количество сценариев, которые были выполнены, количество действий в каждом сценарии, названия действий и сценариев и т.п.).

Путь к "месту работы" "генератора нагрузки" задается вместе с параметрами отчета.

Web-приложение. "Генератор отчетов" построен как Web-приложение. Это дает возможность использовать его с любого рабочего места, где есть выход в Интернет или Интранет. Для работы с "генератором отчетов" необходимо лишь наличие Web-браузера, например MS Internet Explorer.

При создании использовался язык программирования платформы dotNET – C# (Си Шарп). Для создания Web-страниц используется технология ASP.NET.

Заключение

Описанная система автоматизированного тестирования телекоммуникационного оборудования "Массовые вызовы" разработана и эксплуатируется на предприятии НТЦ "Протей", где организовано и ведется консультирование по ее адаптации и эксплуатации. Система использовалась при тестировании контакт-центра "Протей-РВ". После внедрения системы количество bug-reports (сообщений об ошибках в программе) сократилось на 40%. До внедрения системы в среднем за месяц было 7–9 bug-reports, при использовании системы "Массовые вызовы" количество bug-reports составило 2–4 в месяц.

Литература

1. Бабушкин М. и др. Web-сервер в действии. СПб.: Питер, 1997.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. 2-е изд./Пер. с англ. М.: "Издательство Бином", СПб: "Невский диалект", 1998.

АЛГОРИТМИЧЕСКОЕ МОДЕЛИРОВАНИЕ ЦИФРОВОГО ФИЛЬТРА

И.Н. Жданов, К.М. Денисов

Реализация цифровых фильтров с помощью микропроцессора или другим аппаратным путем связана с рядом присущих таким системам ограничений, что приводит к отклонению выходного сигнала реальной системы от его идеального значения. Ошибки цифровой фильтрации обусловлены, во-первых, квантованием входных сигналов, во-вторых, квантованием коэффициентов фильтра и, в-третьих, конечной разрядностью операционных устройств, вследствие чего имеет место округление результатов арифметических операций. Целью данной работы является моделирование алгоритма цифрового фильтра, реализованного на микропроцессоре с сохранением той же последовательности и разрядности вычислений. Это позволяет оценить величину ошибки, возникающей в реальных дискретных системах.

В общем случае линейным дискретным фильтром называется дискретная система, удовлетворяющая линейному разностному уравнению

$$y(n) = -\sum_{m=1}^{M-1} a_m y(n-m) + \sum_{k=0}^{K-1} b_k x(n-k), \quad (1)$$

где $x(n)$ и $y(n)$ – соответственно входная и выходная последовательности устройства.

Передаточная функция линейного дискретного фильтра имеет вид

$$H(z) = \frac{\sum_{k=0}^{K-1} b_k z^{-k}}{1 + \sum_{m=1}^{M-1} a_m z^{-m}}, \quad (2)$$

который получается в результате применения z -преобразования к левой и правой частям уравнения (1).

В общем случае линейный дискретный фильтр может быть реализован путем комбинации операций умножения сигнала на константу, алгебраического сложения и задержки сигнала на один интервал дискретизации T .

Под цифровым фильтром понимают дискретный фильтр, описываемый уравнением (1) и реализованный программным путем с помощью микропроцессора или аппаратным путем в виде специализированного цифрового вычислительного устройства, состоящего из элементов памяти (регистров), сумматоров, умножителей и устройств управления. Фильтр может быть реализован различными структурными схемами. Выбранная структура цифрового фильтра, разрядность входных и выходных сигналов, разрядность арифметических устройств влияют на точность работы фильтра и должны выбираться таким образом, чтобы результирующая ошибка цифрового фильтра не превышала допустимой величины.

Другим важным критерием качества реализации цифрового фильтра является его быстродействие, определяемое минимальным временем, необходимым для вычисления одного отсчета выходного сигнала. Очевидно, что это время должно быть не больше периода дискретизации сигналов.

В реальных устройствах, реализующих алгоритмы цифровой обработки сигналов, необходимо учитывать эффекты, обусловленные квантованием входных сигналов и конечной разрядностью всех регистров.

Для анализа эффектов, связанных с конечной разрядностью представления данных, необходимо сделать некоторые предположения относительно статистической независимости различных источников шумов, возникающих в цифровом фильтре. Принимается статистическая модель, основанная на следующих предположениях:

1. любые два отсчета шума от одного и того же источника не коррелированы;
2. любые два источника шума создают некоррелированные шумы;
3. шум каждого из источников не коррелирован с входным сигналом.

Эти предположения значительно упрощают анализ процессов, связанных с шумами квантования в цифровых фильтрах, поскольку делают отдельные источники шума статистически независимыми друг от друга и дают возможность проводить анализ для каждого из них отдельно. Однако принятые предположения справедливы далеко не всегда. Можно привести множество примеров, для которых эти предположения неверны – например, если входной сигнал является постоянным или синусоидальным с частотой, кратной частоте дискретизации. В первом случае все отсчеты ошибки квантования будут одинаковы, а во втором они образуют периодическую последовательность. Таким образом, в обоих случаях выдвинутые предположения неверны.

Эффекты квантования приводят в конечном итоге к погрешностям в выходных сигналах цифровых фильтров, а в некоторых случаях и к неустойчивым режимам работы. В силу принятых допущений выходная ошибка цифрового фильтра вычисляется как суперпозиция ошибок, обусловленных каждым независимым источником.

Выбор оптимальной разрядности квантования входного сигнала определяется необходимой точностью представления информации, заложенной во входном сигнале, наличием в нем предвходящего шума и процедурой, которая применяется для обработки сигнала.

Содержащийся в сигнале шум определяет верхнюю границу числа уровней квантования. Очевидно, нет смысла использовать большое количество разрядов, когда в сигнале содержится большой шум, так как в этом случае с большой точностью будет квантоваться шум, а не сигнал. Достаточно выбрать столько уровней квантования, чтобы вклад шума квантования был мал по сравнению с шумом, содержащимся в сигнале.

С другой стороны, минимально допустимое количество уровней квантования должно обеспечивать желаемое качество выходного сигнала. Ухудшение качества входного сигнала может быть вызвано воздействием неидеальностей на этапе предварительной обработки сигнала (шумы и ограниченные частотные характеристики предварительных масштабирующих усилителей и аналоговых фильтров).

При физической реализации коэффициенты разностного уравнения фильтра хранятся в элементах электронной памяти (запоминающих ячейках), которые имеют ограниченную разрядность. Это означает, что коэффициенты фильтра, также как и входной сигнал, подвергаются квантованию. Квантование коэффициентов фильтра подчиняется тем же закономерностям, что и квантование входного сигнала. В результате квантования коэффициентов фильтра значения полюсов и нулей передаточной функции фильтра в большей или меньшей степени изменяются, что, в свою очередь, приводит к соответствующему изменению частотных характеристик фильтра. Так, квантование коэффициентов фильтра приводит к появлению ошибки

$$\Delta A = A(\omega) - A_d(\omega), \quad (3)$$

где $A(\omega)$ – АЧХ фильтра с неквантованными коэффициентами, $A_d(\omega)$ – АЧХ фильтра с квантованными коэффициентами. Величина $|\Delta A(\omega)|$ не должна превосходить

допустимую величину $|\Delta A(\omega)|_{\max}$, определяемую обычно из условия, чтобы отклонения реальной АЧХ от идеальной были в допустимых пределах.

Различные структуры фильтров имеют различную чувствительность к изменению отдельных коэффициентов. Поэтому универсального метода определения требуемого количества разрядов квантования коэффициентов для всех типов фильтров предложить невозможно. Необходимое количество разрядов в квантованных коэффициентах фильтров можно определить путем вычисления $|\Delta A(\omega)|$ для последовательно возрастающего числа разрядов в кодах коэффициентов до выполнения условия $|\Delta A(\omega)| < |\Delta A(\omega)|_{\max}$.

Возможны и практически применяются и другие методы, в частности методы, основанные на предварительном изучении чувствительности характеристик конкретного типа фильтра к изменениям его коэффициентов.

При реализации алгоритма цифрового фильтра выполняются операции сложения и умножения на коэффициенты. Сложение чисел с фиксированной точкой при разрядности сумматора, не меньшей разрядности представления слагаемых, не приводит к ошибкам округления представления суммы, однако при выполнении операций умножения эта ошибка будет появляться, поэтому при последовательном выполнении операций умножения необходимо ограничивать разрядность произведений, иначе разрядность последующих произведений будет неограниченно возрастать. В результате округления произведений алгоритм фильтра реализуется неточно, и выходной сигнал вычисляется с ошибкой.

Уровень выходного шума фильтра, обусловленного квантованием произведений, сильно зависит от особенностей структуры, выбранной для реализации фильтра. Это объясняется тем, что импульсная характеристика участка фильтра от выхода конкретного множителя до выхода фильтра зависит от применяемой структуры. При выборе структуры фильтра необходимо учитывать влияние ошибок квантования произведений наряду с ошибками квантования коэффициентов.

При суммировании чисел с фиксированной разрядностью возможно возникновение переполнения, когда получившийся результат не помещается в количество разрядов, соответствующее разрядности слагаемых. При возникновении переполнения во избежание нарушения алгоритма функционирования фильтра сумма должна быть ограничена с учетом знака на уровне максимального значения, уместяющегося в заданное количество разрядов результата. При программной реализации фильтра это осуществляется соответствующим ветвлением алгоритма функционирования, а при аппаратной реализации требует включения в схему фильтра специальных устройств анализа переполнения и ограничения суммы с учетом знака. Однако даже реализация указанных средств не решает всех проблем, связанных с переполнением, так как при наличии переполнений фильтр превращается в существенно нелинейное устройство со всеми вытекающими из этого последствиями. Поэтому для нормальной работы фильтра необходима реализация специальных мер, позволяющих избежать появления ситуации переполнения вообще.

Одно из средств для предотвращения переполнения заключается во введении масштабирования, которое сводится к сдвигу вправо (что эквивалентно делению) двоичных кодов слагаемых на всех входах сумматоров. Если исходные слагаемые нормированы на уровне 1.0, то при суммировании двух чисел для исключения возможности переполнения необходимо каждое из слагаемых сдвинуть на один разряд вправо. Такой метод называется автоматическим масштабированием.

Другой способ предотвращения возможности переполнения сводится к масштабированию входных сигналов фильтра или его составных частей. Для этого включают на входах фильтра или на выходах множителей масштабирующие

умножители. При этом, аналогично случаю автоматического масштабирования, возникает шум масштабирования, который снижает отношение сигнал-шум на выходе фильтра. Масштабирующий множитель, как и любой другой множитель в структурной схеме фильтра, является источником шума ошибки квантования, влияние которого на выходной сигнал может быть учтено аналогично шумам других умножителей.

Существующие методы оценки возникающих погрешностей базируются на предположении статистической независимости входного сигнала и отдельных источников шума внутри фильтра, но на практике подобные допущения часто нарушаются, что делает эти оценки весьма условными.

Разработанная математическая модель в пакете MathCad позволяет выявить ошибки, присущие реальным системам, но практически не рассчитываемые аналитически. Это делает возможным оценить величину ошибки в зависимости от разрядности для фильтров различного порядка и реализованных различными структурными схемами, что поможет правильно выбрать вычислительное устройство для реализации цифрового фильтра. Так, например, для фильтра Баттерворта второго порядка расхождение АЧХ при 12-тиразрядном представлении коэффициентов фильтра от АЧХ идеального фильтра представлено на рис. 1. При реализации структурной схемы в прямой форме моделирование реакции фильтра на скачок задающего воздействия дало следующие результаты: при числе разрядов $N=12$ ошибка составляет 0,58% (или точность – 7 разрядов, рис. 2), $N=16$ – 0,03% (11 разрядов), $N=32$ – 0,00016% (19 разрядов). Для фильтра 8-го порядка при $N=12$ – 2,051% (5 разрядов), $N=16$ – 0,1221% (9 разрядов), $N=32$ – 0,0017% (15 разрядов).

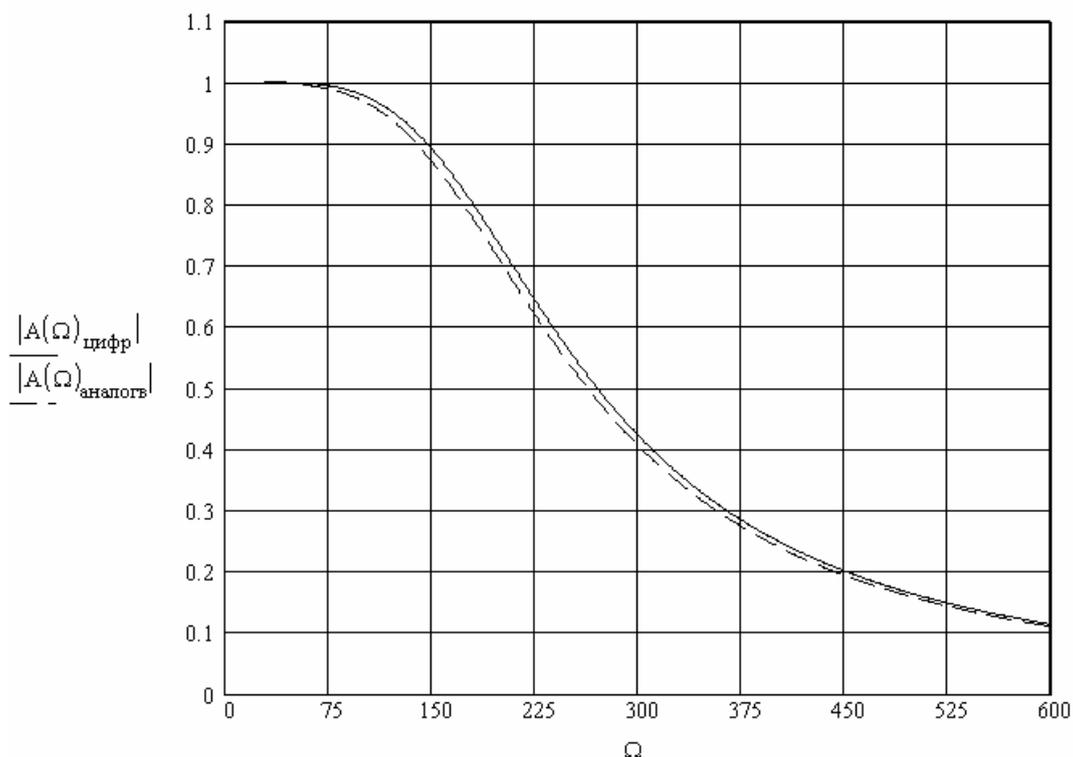


Рис.1. АЧХ фильтра с квантованными коэффициентами и АЧХ идеального фильтра

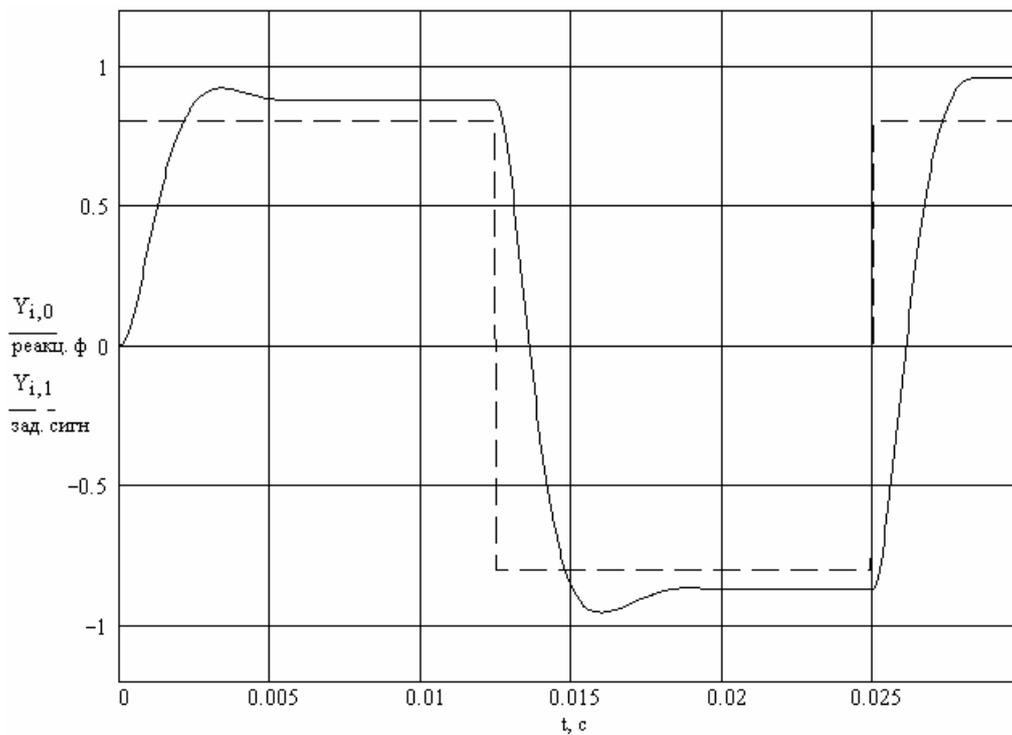


Рис. 2 Реакция фильтра на скачок задающего воздействия

Литература

1. Digital Control System Design. Analog Devices Inc., 1998. Data sheet.
2. Титце У., Шенк К., Полупроводниковая схемотехника: Справочное руководство. / Пер. с нем. М.: Мир, 1982.
3. Гольденберг Л.М., Матюшкин Б.Д., Поляк М.Н. Цифровая обработка сигналов: учеб. пособие для вузов. / 2-изд., перераб. и доп. М.: Радио и связь, 1990. 256 с.: ил.
4. Изерман Р. Цифровые системы управления. М.: Мир, 1984.

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ВЫДЕЛЕНИЯ ГЛАДКОЙ СОСТАВЛЯЮЩЕЙ СИГНАЛА ТОКА В ЭЛЕКТРОПРИВОДЕ

Д.В. Лукичев, А.А. Усольцев

При построении замкнутого электропривода переменного тока часто автономный инвертор напряжения (АИН) используется в режиме формирования тока. Этот способ иллюстрирует рис. 1. Здесь за счет отрицательной обратной связи по току и релейного элемента (РЭ) ток в нагрузке пульсирует около заданного значения. Распределитель импульсов (РИ) управляет моментом включения транзисторов. Амплитуда и частота пульсаций определяются параметрами R , L активно-индуктивной нагрузки и шириной петли гистерезиса Δ релейного элемента. Часто такой способ АИН называют "токовым коридором". При реализации "токового коридора" инвертор представляет собой источник тока, а управление электрической машиной в этом случае относят к частотно-токовому [1].

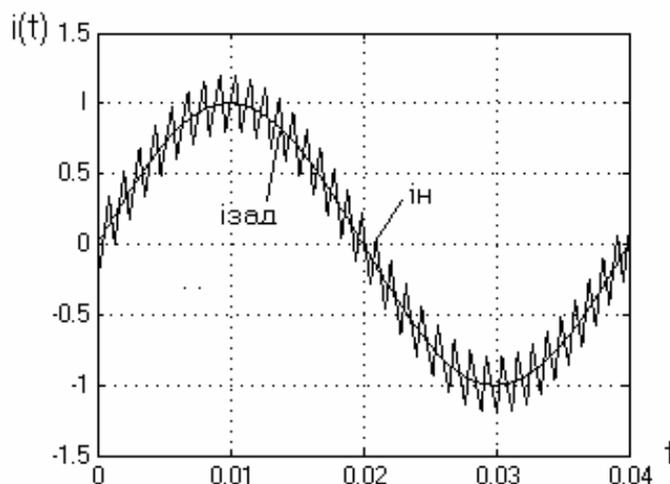
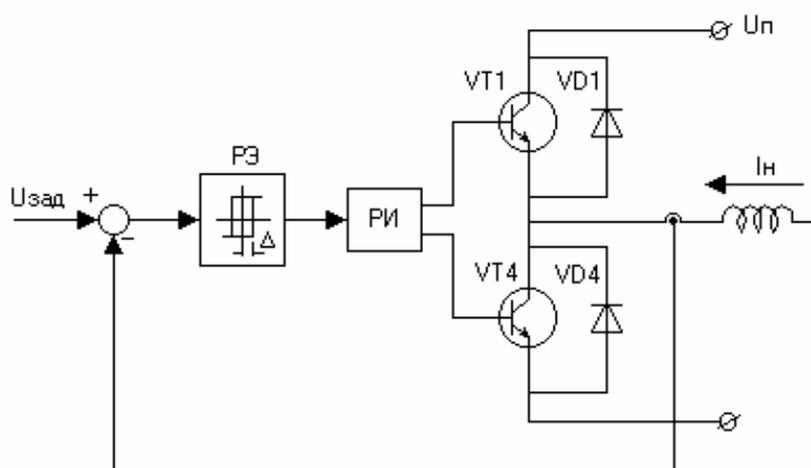


Рис. 1. Реализация "токового коридора" в АИН

Устройства управления инвертором должны реализовать способ, который обеспечил бы удовлетворение двух основных требований, предъявляемых к системе преобразователь-машина переменного тока:

- минимальные потери в двигателе и минимальные пульсации момента, обусловленные воздействием полей первой и высших гармоник;
- минимальные потери в элементах преобразователя.

Эти требования противоречивы, так как для улучшения работы двигателя следует повышать частоту коммутации, а для уменьшения потерь в преобразователе ее следует уменьшать. На практике достаточно, чтобы частота коммутации при всех режимах работы системы приблизительно на порядок превосходила частоту основной гармоники. Важно отметить, что частота коммутаций или так называемая несущая частота изменяется на периоде в довольно широком пределе и зависит от производной синуса в каждый момент времени. В данной работе величина Δ релейного элемента была выбрана из расчета, чтобы максимальная несущая частота не превышала 15кГц. При этом в "токовом коридоре" пульсирующие значения тока находятся в пределах $\pm 20\%$ от задающего сигнала.

Для различных вычислений в системе управления электроприводом необходимо иметь информацию о значении основной гармоники тока в фазе двигателя, которое, вообще говоря, в динамике будет отличаться от задающего сигнала. Эта информация может использоваться, например, в вычислениях потокосцепления электродвигателя. В современных системах для получения гладкой составляющей тока применяются различные фильтры. Но их существенным недостатком является наличие ненулевой фазочастотной характеристики, которая приводит к сдвигу фазы пропускаемого через фильтр сигнала. В данной работе исследуется возможность применения искусственных нейронных сетей (ИНС) для получения гладкой составляющей тока без временной задержки с заданной точностью.

Искусственная нейронная сеть в простейшем случае представляет собой, подобно сети нервных клеток, упорядоченную по слоям совокупность элементов, называемых нейронами [2]. Исследуемая в данной работе сеть имеет архитектуру, приведенную на рис. 2.

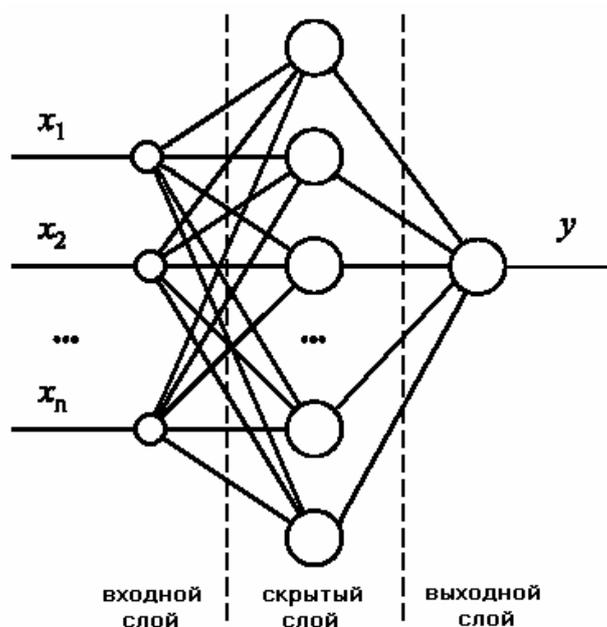


Рис. 2 Схема трехслойной ИНС

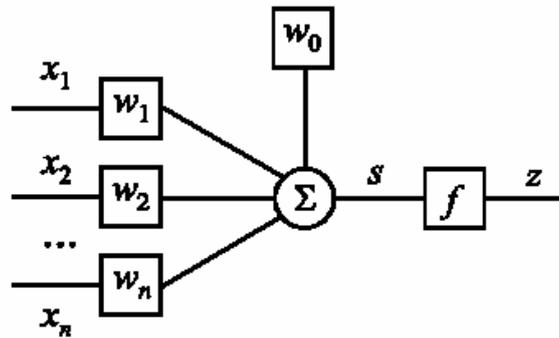


Рис. 3 Модель нейрона

Чтобы подготовить нейронную сеть к работе, ее следует обучить. Процесс обучения сети состоит в подаче на ее вход некоторого тренировочного вектора входных данных и последующей настройке параметров весов связей w_i и смещений w_0 до тех пор, пока сеть не научится с заданной точностью отображать на выходе вектор желаемых (целевых) выходных данных. Для оценки качества работы сети обыкновенно используется целевая функция, представляющая собой сумму квадратов ошибок E по выходам ИНС для всех N обучающих векторов:

$$E = \sum_{i=1}^N (t_i - y_i)^2, \quad (1)$$

где t_i, y_i – i -е значение вектора желаемых и фактических выходов сети, соответственно.

Если удастся так настроить сеть, чтобы величина E не превышала некоторую величину одновременно для всех совокупностей входных сигналов, процесс обучения полагают законченным, после чего параметры сети фиксируют, а саму сеть считают готовой к работе. Благодаря обучению сеть приобретает способность различать не только тренировочные сигналы, предъявляемые в процессе обучения, но также хорошо справляться с любыми другими допустимыми сигналами.

Все эксперименты, описанные в данной работе, проводились с использованием пакетов прикладных программ Neural Network Toolbox и Simulink, функционирующих под управлением ядра системы Matlab 6.0. Была построена нейронная сеть, которая по некоторому количеству предшествующих значений (соответствующих числу входов) выдает на выходе текущее значение гладкой составляющей сигнала. Таким образом, тренировочная входная последовательность состояла из значений пульсирующего токового сигнала, а желаемая последовательность выхода определялась как задающая синусоида. Обучение сети производилось по алгоритму Левенберга-Марквардта. Вектора входов и целей масштабировались так, чтобы их элементы находились в интервале значений $[-1 \ 1]$. Функция активации скрытого слоя – униполярная сигмоидальная, а выходного слоя – линейная. Было показано, что сеть воспроизводит синусоидальный сигнал с заданной точностью только при ее обучении на всем периоде задающего сигнала. Исследовалось влияние количества входов, периода дискретизации и числа нейронов скрытого слоя на ошибку обучения. На рис. 4 приведены графики зависимостей суммы квадратов ошибок от периода дискретизации для сетей, имеющих 3, 5 и 10 входов и 5 нейронов в скрытом слое.

Функция изменения ошибки в зависимости от шага дискретизации имеет два четко выраженных участка. На первом участке с увеличением шага дискретизации ошибка резко уменьшается и достигает первого минимума на частоте дискретизации, при которой число точек, приходящихся на период переключения ключей инвертора, соответствует числу входов нейронной сети. При дальнейшем увеличении шага ошибка

изменяется хаотически, сохраняя тенденцию к уменьшению среднего значения. Неравномерность функции изменения ошибки связана с тем, что при обучении нейронных сигмоидальных сетей, основанном на минимизации значения целевой функции, появляется большое число локальных минимумов, которые не только осложняют процесс обучения, но и представляют собой ловушки на пути к глобальному минимуму, в котором целевая функция принимает минимальное значение.

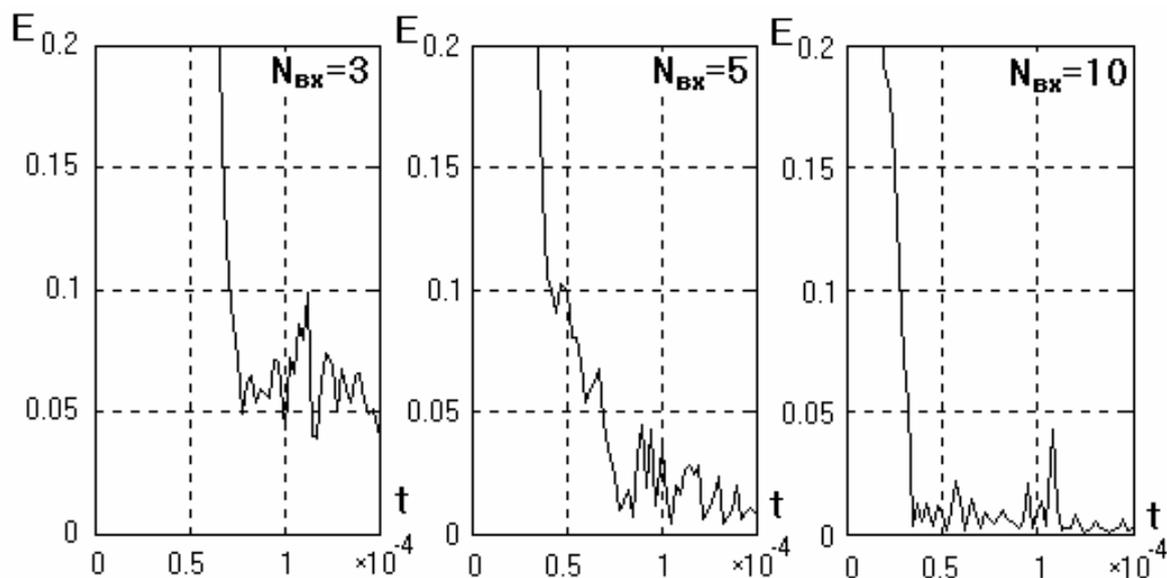


Рис. 4 Ошибка обучения ИНС при 5 нейронах в скрытом слое

На рис. 5 приведены зависимости суммы квадратов ошибок от периода дискретизации для этих же сетей, но при увеличении числа нейронов в скрытом слое до 15.

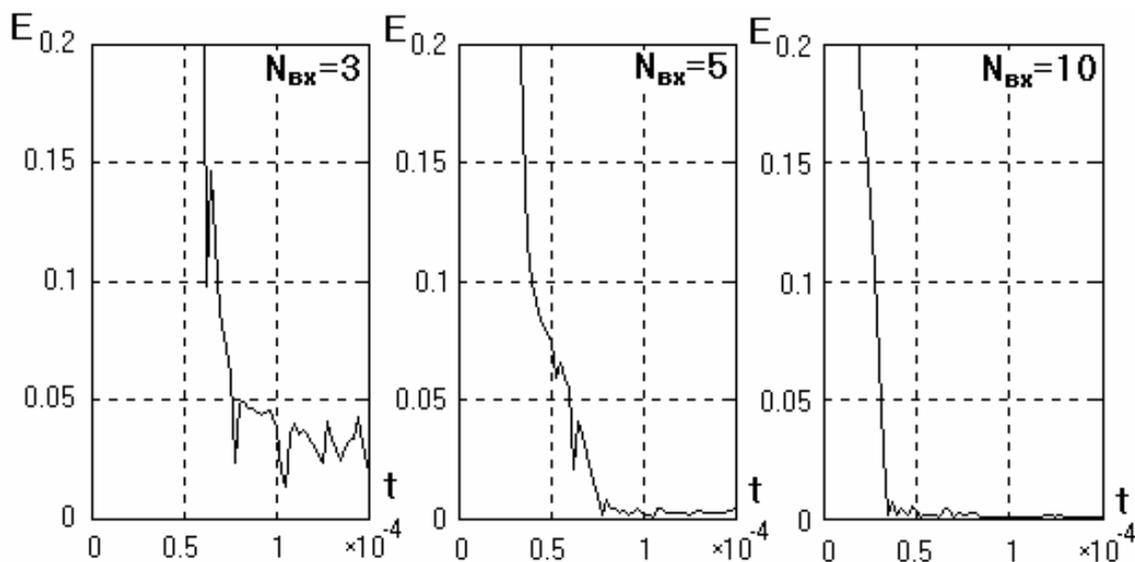


Рис. 5. Ошибка обучения ИНС при 15 нейронах в скрытом слое

Характер функции изменения ошибки в зависимости от шага дискретизации остался прежним, но ошибка значительно уменьшилась. Например, для сети, имеющей 10 входов и 5 нейронов в скрытом слое, в точке первого минимума сумма квадратов

ошибок составляет 0,35%, а при увеличении числа нейронов до 15 ошибка уже 0,14%. Однако увеличение количества нейронов приводит не только к уменьшению ошибки, но и к усложнению архитектуры сети, что делает более затруднительной ее реализацию, а также увеличивает время вычислений.

На рис. 6 приведены диаграммы основной гармонике токового сигнала i_o , а также сигнала i_c , получаемого на выходе обученной нейронной сети.

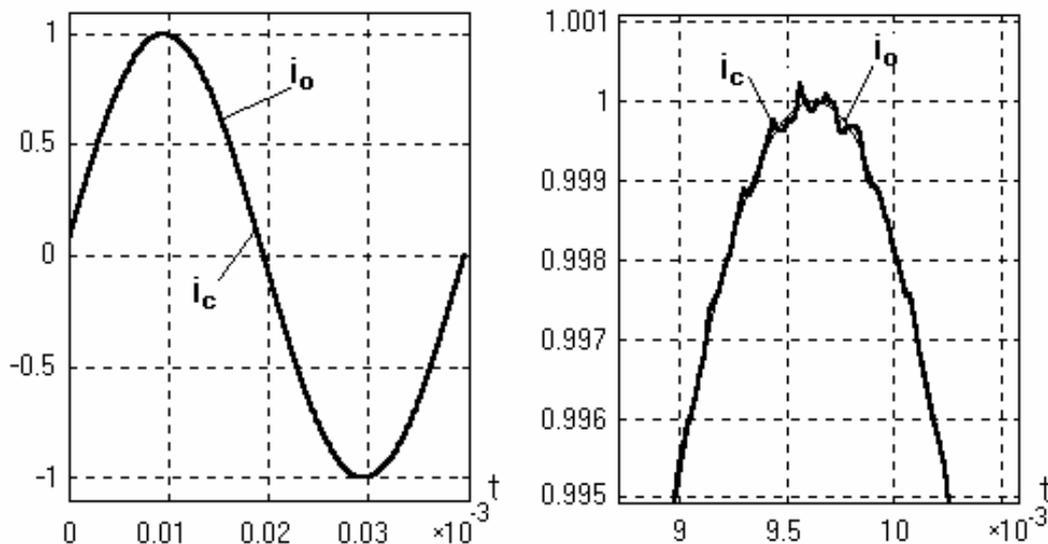


Рис. 6. Сигнал на выходе ИНС и желаемый сигнал

Полученный на выходе ИНС сигнал имеет нулевой фазовый сдвиг относительно основной гармонике обрабатываемого сигнала, что является несомненным преимуществом ИНС по сравнению с традиционными фильтрами.

Выводы

1. Выбор числа входов рассматриваемой сети по условию минимизации ошибки должен производиться, исходя из соотношения $N_{\text{вх}} \geq N_{\text{д}}$, где $N_{\text{вх}}$ – число входов сети, $N_{\text{д}}$ – число точек дискретизации сигнала на периоде коммутации.
2. Показано, что построенная ИНС обрабатывает сигналы без фазового сдвига.

Литература

1. Компьютерное моделирование полупроводниковых систем в Matlab 6.0: Учебное пособие. / С.Г. Герман-Галкин. СПб.: КОРОНА принт, 2001. 320 с., ил.
2. Нейронные сети. Matlab 6.0 / Под общ. ред. к.т.н. В.Г. Потемкина. М.: ДИАЛОГ-МИФИ, 2002. 496 с.

ВЛИЯНИЕ ЧАСТОТЫ ДИСКРЕТИЗАЦИИ НА УСТОЙЧИВОСТЬ ЦИФРОВОЙ СИСТЕМЫ АВТОМАТИЧЕСКОГО РЕГУЛИРОВАНИЯ ТОКА С ШИРОТНО-ИМПУЛЬСНЫМ ПРЕОБРАЗОВАТЕЛЕМ

В.А. Толмачев, Д.В. Осипов

Системы автоматического регулирования (САР) с широтно-импульсными преобразователями (ШИП) постоянного напряжения находят широкое применение в современной преобразовательной технике. Одной из перспективных областей применения САР с ШИП являются источники тока для нестационарного электролиза. Непрерывное повышение требований к статическим и динамическим характеристикам таких источников ставит перед теорией САР новые, все усложняющиеся задачи. К ним в первую очередь относятся задачи синтеза и анализа САР.

Основной задачей синтеза САР тока для нестационарного электролиза является обеспечение в нагрузке импульсов тока требуемой формы. При этом выбор конкретной структуры САР и значений ее параметров связан с решением ряда задач анализа САР, в частности, с задачей анализа ее устойчивости. Эта задача сводится к определению в пространстве параметров САР границ областей, в которых обеспечивается нормальный режим ее работы. Нормальным для САР с ШИП является режим установившихся в ней колебаний с периодом, равным периоду коммутации силовых ключей (СК) ШИП.

При эксплуатации САР или ее настройке возможны как незначительные, так и значительные изменения ее параметров, которые могут привести к нарушению ее устойчивости. В результате нарушения устойчивости в САР возникают так называемые особые (ненормальные) режимы ее работы с периодом колебаний, не равным периоду коммутации СК. Характерными для САР с ШИП особыми режимами работы являются скользящий режим и режим автоколебаний на субгармонических частотах [1, 2].

Современный этап развития преобразовательной техники характеризуется широким внедрением в информационные каналы САР микроконтроллеров (МК), реализующих цифровые алгоритмы управления. Существующие в настоящее время МК объединяют на одном кристалле практически все устройства сопряжения с объектами регулирования, а именно аналого-цифровые преобразователи (АЦП) и широтно-импульсные модуляторы (ШИМ), необходимые для построения замкнутых САР с ШИП [3]. Аналогичная ситуация складывается и при построении энергетических каналов САР. Современные силовые модули ШИП включает в себя, помимо СК, также формирователи импульсов (ФИ), необходимых для управления СК, и различные датчики для организации обратных связей и реализации функций защиты СК [4]. Здесь также необходимо отметить, что построение САР с ШИП на основе МК позволяет исключить скользящий режим ее работы. Но при этом актуальным остается вопрос об устойчивости такой САР к автоколебаниям на субгармонических частотах.

Объектом исследования в данной работе является цифровая одномодульная САР тока, структурная схема которой приведена на рис. 1. САР содержит ШИП и МК. ШИП состоит из ФИ и силового каскада мостового типа на четырех СК (СК1 – СК4) с напряжением питания U на входной диагонали. В выходную диагональ каскада включен датчик тока (ДТ) с коэффициентом передачи $K_{дт}$, сглаживающий дроссель с индуктивностью L , и резистивный элемент с сопротивлением R , учитывающий активные сопротивления ДТ, нагрузки, обмотки дросселя, внутреннее сопротивление источника напряжения питания силового каскада, а также дифференциальные сопротивления открытых СК.

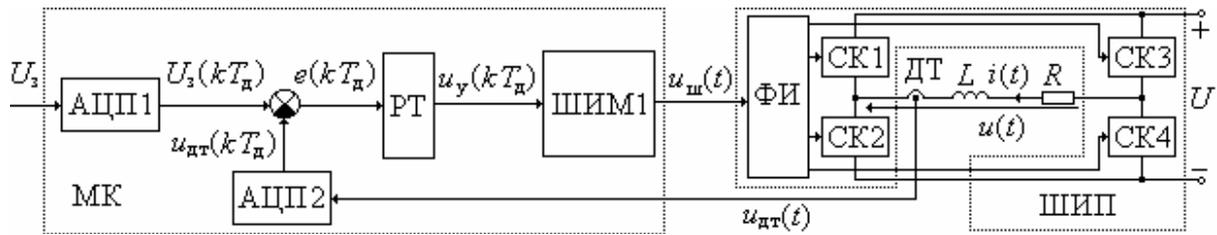


Рис. 1. Структурная схема цифровой САР

МК включает в себя АЦП1, АЦП2, регулятор тока (РТ) и ШИМ первого рода (ШИМ1). АЦП1 преобразует с некоторой частотой дискретизации f_d непрерывный сигнал задания U_3 тока нагрузки $i(t)$ в дискретный сигнал $U_3(kT_d)$, где $T_d = 1/f_d$ – период дискретизации, $k = \overline{0, \infty}$. АЦП2 синхронно с АЦП1 преобразует непрерывный сигнал ДТ $u_{дрт}(t) = K_{дрт}i(t)$ в дискретный сигнал ДТ $u_{дрт}(kT_d)$. РТ преобразует сигнал ошибки $e(kT_d) = U_3(kT_d) - u_{дрт}(kT_d)$ в сигнал управления $u_y(kT_d)$ согласно цифровому пропорционально-интегральному (ПИ) алгоритму управления, заданному следующими разностными уравнениями [5]:

$$u_y(kT_d) = u_n(kT_d) + u_{ин}(kT_d); \quad u_n(kT_d) = K_n e(kT_d); \quad u_{ин}((k+1)T_d) = u_{ин}(kT_d) + K_{ин} e(kT_d), \quad (1)$$

где $u_n(kT_d)$ и $u_{ин}(kT_d)$ – пропорциональная и интегральная составляющие сигнала управления $u_y(kT_d)$; K_n и $K_{ин}$ – коэффициенты передачи пропорционального и интегрального каналов РТ, соответственно.

На выходе ШИМ1 формируется маломощный сигнал $u_{ш}(t)$ в виде однополярных импульсов с амплитудой, равной напряжению питания МК $U_{МК}$ (обычно +5В), частотой следования, равной частоте коммутации СК f_k , и длительностью пауз между ними, пропорциональной значениям сигнала управления $u_y(kT_d)$. При этом осуществляется модуляция как фронтов, так и срезов этих импульсов (двухсторонняя модуляция). ФИ преобразует выходной сигнал ШИМ1 $u_{ш}(t)$ в импульсы, необходимые для управления СК и обеспечения требуемого закона их коммутации. В данном случае обеспечивается несимметричный закон коммутации так, что выходной сигнал ШИП $u(t) = u_{ш}(t)U/U_{МК}$ есть копия сигнала ШИМ1 $u_{ш}(t)$, но с амплитудой U .

Сигнал управления $u_y(kT_d)$ может принимать на периоде коммутации $T_k = 1/f_k$ одно или более значений в зависимости от взаимного соотношения частот f_d и f_k . Далее рассматриваются два случая. В первом случае $f_d = f_k$, и сигнал управления $u_y(kT_d)$ принимает только одно значение на периоде T_k . Во втором случае $f_d = 2f_k$, и сигнал управления $u_y(kT_d)$ принимает два значения на периоде T_k . Временные диаграммы работы исследуемой САР в переходном процессе для обоих случаев приведены на рис. 2. Длительности интервалов времени внутри n -ого периода T_k , в конце которых формируются срезы и фронты импульсов ШИП $u(t)$, обозначены через $t_{с,n}$ и $t_{ф,n}$ соответственно. При $f_d = f_k$ (рис. 2 а) длительности обоих интервалов, как $t_{с,n}$, так и $t_{ф,n}$, пропорциональны значению сигнала управления $u_y(kT_d)$ в начале периода T_k . А при $f_d = 2f_k$ (рис. 2 б) длительность интервала $t_{с,n}$ пропорциональна значению сигнала управления $u_y(kT_d)$ в начале периода T_k , а длительность интервала

$t_{\phi,n}$ – значению сигнала управления $u_y((k+1)T_d)$ в середине периода T_k . В обоих случаях, как при $f_d = f_k$, так и при $f_d = 2f_k$, начало интервала $t_{c,n}$ совпадает с началом периода T_k , а начало интервала $t_{\phi,n}$ совпадает с серединой периода T_k .

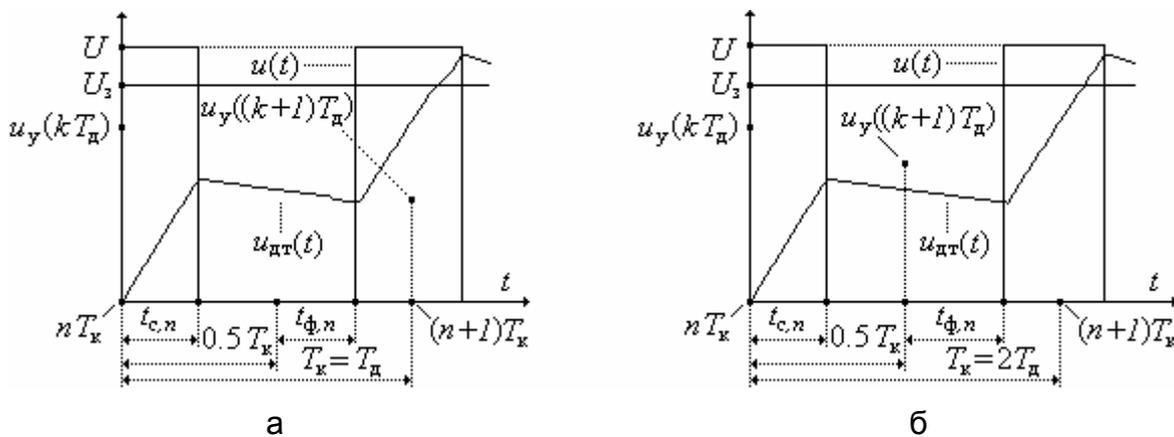


Рис. 2. Временные диаграммы работы цифровой САР:

а – при $f_d = f_k$; б – при $f_d = 2f_k$

Задачей данного исследования является определение в пространстве параметров рассматриваемой САР областей ее устойчивости при $f_d = f_k$ и $f_d = 2f_k$. Для решения поставленной задачи требуется наличие расчетных соотношений, задающих границы указанных областей. Для получения требуемых расчетных соотношений необходимо располагать математическими моделями исследуемой САР.

С точки зрения теории автоматического управления САР с ШИП представляют собой нелинейные дискретные системы. Для анализа устойчивости таких САР широко применяются различные методы линеаризации нелинейных математических моделей САР с дальнейшим применением теории линейных дискретных или непрерывных систем. Но эти методы имеют ряд существенных ограничений и пригодны лишь при решении локальных задач на различных этапах нелинейного анализа. Перспективны в этом отношении методы теории бифуркаций и метод точечных отображений [2]. Поставленная задача решается на основе этих методов и сводится к анализу математических моделей исследуемой САР, заданных в виде точечных отображений.

На этапе формирования математических моделей САР приняты следующие допущения. При рассмотрении процессов в ее энергетических каналах каждый СК представлен последовательным соединением идеального ключа и резистивного элемента, учитывающего дифференциальное сопротивление включенного СК. При этом принято, что сопротивления всех СК равны друг другу. Не учитываются индуктивности и емкости СК, поскольку время затухания переходных процессов, возбуждаемых этими параметрами в моменты переключений СК, составляет незначительную часть периода коммутации СК. Не учитываются нелинейности сглаживающего дросселя и источника напряжения питания U , а их схемы замещения приняты последовательными.

При рассмотрении процессов в информационных каналах САР принято, что сигнал задания U_3 изменяется в начале периода коммутации СК, а АЦП1 преобразует его в сигнал $U_3(kT_d)$, а также АЦП2 преобразует сигнал ДТ $u_{дт}(t)$ в сигнал $u_{дт}(kT_d)$, мгновенно. Разрядности АЦП1, АЦП2 и ШИМ1 приняты бесконечно большими. Не учитывается также нелинейность ДТ и принято, что он безынерционный.

Формирование математических моделей исследуемой САР тока основывается на рассмотрении процессов в ее линейной непрерывной части (ЛНЧ), на вход которой подается сигнал ШИП $u(t)$, а также процессов в РТ и ШИМ1. На каждом n -ом периоде T_k в нормальном режиме работы САР можно выделить три следующих интервала времени, внутри которых ее структура остается постоянной:

1. $nT_k \leq t \leq nT_k + t_{c,n}$, где $u(t) = U$;
2. $nT_k + t_{c,n} < t < (n + 0.5)T_k + t_{\phi,n}$, где $u(t) = 0$;
3. $(n + 0.5)T_k + t_{\phi,n} \leq t \leq (n + 1)T_k$, где $u(t) = U$.

На каждом из этих интервалов времени процессы в ЛНЧ, состоящей в данном случае из дросселя с индуктивностью L и резистивного элемента с сопротивлением R , описываются следующим дифференциальным уравнением состояния ЛНЧ

$$L \frac{di(t)}{dt} + Ri(t) = u(t). \quad (2)$$

Применяя метод припасовывания к решениям уравнения (2) на каждом из этих интервалов, получаем следующее разностное уравнение состояния ЛНЧ

$$i((n + 1)T_k) = d(T_k)i(nT_k) + (1 - d(T_k) + d(T_k - t_{c,n}) - d(0.5T_k - t_{\phi,n}))I, \quad (3)$$

где $d(t) = e^{-tR/L}$; $I = U/R$ – максимально возможное значение тока нагрузки $i(t)$.

Аналогичным образом, рассматривая процессы в РТ на каждом из указанных интервалов и применяя метод припасовывания к уравнениям ПИ-алгоритма управления (1) с учетом уравнения сигнала ошибки $e(kT_d) = U_3(kT_d) - u_{дт}(kT_d)$, получаем следующие разностные уравнения:

$$u_{и}((n + 1)T_k) = u_{и}(nT_k) + K_{и}(U_3 - K_{дт}i(nT_k)); \quad (4)$$

$$u_{у}(nT_k) = u_{и}(nT_k) + K_{п}(U_3 - K_{дт}i(nT_k)); \quad u_{у}((n + 0.5)T_k) = u_{и}(nT_k) + K_{п}(U_3 - K_{дт}i(nT_k)); \quad (5)$$

$$u_{и}((n + 1)T_k) = u_{и}(nT_k) + 2K_{и}U_3 - K_{и}K_{дт}((1 + d(0.5T_k))i(nT_k) + (d(0.5T_k - t_{c,n}) - d(0.5T_k))I); \quad (6)$$

$$u_{у}(nT_k) = u_{и}(nT_k) + K_{п}(U_3 - K_{дт}i(nT_k)); \quad u_{у}((n + 0.5)T_k) = u_{и}(nT_k) + (K_{п} + K_{п})U_3K_{дт} \times \\ \times ((K_{и} + K_{п}d(0.5T_k))i(nT_k) + K_{п}(d(0.5T_k - t_{c,n}) - d(0.5T_k))I), \quad (7)$$

где (4) и (5) – уравнение интегральной составляющей $u_{и}(kT_d)$ сигнала управления $u_{у}(kT_d)$ (уравнение состояния РТ) и уравнения самого сигнала управления $u_{у}(kT_d)$ на выходе РТ в начале и в середине n -ого периода T_k при $f_d = f_k$, когда $k = n$ и $T_d = T_k$ (рис. 2 а), а (6) и (7) – соответственно при $f_d = 2f_k$, когда $k = 2n$ и $T_d = 0.5T_k$ (рис. 2 б).

Исходя из принципа действия ШИМ1, получаем следующие уравнения замыкания САР (уравнения длительностей интервалов $t_{c,n}$ и $t_{\phi,n}$):

$$t_{c,n} = K_{ш}u_{у}(nT_k); \quad t_{\phi,n} = 0.5T_k - K_{ш}u_{у}((n + 0.5)T_k), \quad (8)$$

где $K_{ш} = 0.5T_k/U_{у,max}$ – коэффициент передачи ШИМ1; $U_{у,max}$ – максимальное значение сигнала управления $u_{у}(kT_d)$, соответствующее насыщению ШИП.

Подставляя уравнения (5) в уравнения (8), получаем уравнения замыкания САР при $f_d = f_k$:

$$t_{c,n} = K_{ш}(u_{и}(nT_k) + K_{п}(U_3 - K_{дт}i(nT_k))); \quad t_{\phi,n} = 0.5T_k - K_{ш}(u_{и}(nT_k) + K_{п}(U_3 - K_{дт}i(nT_k))). \quad (9)$$

Аналогично, подставляя уравнения (7) в уравнения (8), получаем уравнения замыкания САР при $f_d = 2f_k$:

$$t_{c,n} = K_{ш}(u_{и}(nT_k) + K_{п}(U_3 - K_{дт}i(nT_k))); t_{ф,n} = 0.5T_k - K_{ш} \times \\ \times (u_{и}(nT_k) + (K_{и} + K_{п})U_3 K_{дт}((K_{и} + K_{п}d(0.5T_k))i(nT_k) + K_{п}(d(0.5T_k - t_{c,n}) - d(0.5T_k))I)). \quad (10)$$

Здесь необходимо отметить, что длительности интервалов $t_{c,n}$ и $t_{ф,n}$ ограничены "снизу", так как они не могут быть отрицательными, и "сверху", в силу возможности насыщения ШИП. Данное обстоятельство учитывается следующими неравенствами:

$$0 \leq t_{c,n} \leq 0.5T_k; \quad 0 \leq t_{ф,n} \leq 0.5T_k. \quad (11)$$

Итак, совокупности выражений (3), (4), (9) и (11) и выражений (3), (6), (10) и (11) являются математическими моделями исследуемой САР соответственно при $f_d = f_k$ и $f_d = 2f_k$ и представляют собой нелинейные двумерные отображения первого порядка.

Дальнейший анализ устойчивости рассматриваемой САР сводится к анализу устойчивости однократной неподвижной точки ее отображения, которая соответствует нормальному режиму ее работы. Для этого необходимы уравнения координат i_N , $u_{и,N}$, $t_{c,N}$ и $t_{ф,N}$ этой точки, а также уравнения матрицы Якоби отображения САР. Матрица Якоби \mathbf{J} отображения является некоторой функцией расширенного вектора состояния САР $\mathbf{X}(nT_k) = [i(nT_k); u_{и}(nT_k); t_{c,n}; t_{ф,n}]^T$ и представляет собой следующую матрицу частных производных

$$\mathbf{J} = f(\mathbf{X}(nT_k)) = \begin{bmatrix} j_{1,1} & j_{1,2} \\ j_{2,1} & j_{2,2} \end{bmatrix} = \begin{bmatrix} \partial i((n+1)T_k)/\partial i(nT_k) & \partial i((n+1)T_k)/\partial u_{и}(nT_k) \\ \partial u_{и}((n+1)T_k)/\partial i(nT_k) & \partial u_{и}((n+1)T_k)/\partial u_{и}(nT_k) \end{bmatrix}, \quad (12)$$

причем уравнения элементов $j_{1,1}$, $j_{1,2}$, $j_{2,1}$ и $j_{2,2}$ этой матрицы могут быть получены путем дифференцирования, сначала по $\partial i(nT_k)$, затем по $\partial u_{и}(nT_k)$, уравнений (3), (4) и (9) при $f_d = f_k$ и уравнений (3), (6) и (10) при $f_d = 2f_k$, соответственно. Этими же уравнениями можно воспользоваться и для вычисления координат неподвижной точки. Для этого в них необходимо произвести следующую замену:

$$i((n+1)T_k) = i(nT_k) = i_N; \quad u_{и}((n+1)T_k) = u_{и}(nT_k) = u_{и,N}; \quad t_{c,n} = t_{c,N}; \quad t_{ф,n} = t_{ф,N} \quad (13)$$

и далее решить их совместно относительно искомым координат i_N , $u_{и,N}$, $t_{c,N}$ и $t_{ф,N}$.

Условия устойчивости неподвижной точки отображения САР заключаются в том, чтобы собственные числа λ_1 и λ_2 матрицы \mathbf{J} , вычисленные в этой точке, т. е. при $i(nT_k) = i_N$, $u_{и}(nT_k) = u_{и,N}$, $t_{c,n} = t_{c,N}$ и $t_{ф,n} = t_{ф,N}$, были по модулю меньше единицы. Представим эти условия в виде следующих неравенств [6]:

$$|\lambda_{1|X=X_N}| < 1; \quad |\lambda_{2|X=X_N}| < 1, \quad (14)$$

где $\mathbf{X}_N = [i_N; u_{и,N}; t_{c,N}; t_{ф,N}]^T$ – вектор координат неподвижной точки.

Перед построением границ областей устойчивости САР необходимо провести предварительный ее синтез. Методика синтеза рассматриваемой САР, исходящая из условия обеспечения ее предельного быстродействия при ограниченной частоте f_k , приведена в работе [5]. В результате синтеза по данной методике получены следующие значения параметров САР: $T_k = 1$ мс, $R = 0.25$ Ом, $L = 1.33$ мГн, $U = 50$ В, $U_3 = 10$ В, $U_{y,max} = 10$ В, $K_{п} = 1$, $K_{и} = 0.158$ и $K_{дт} = 0.2$.

Рассмотрим границы областей устойчивости САР в пространстве ее параметров U_3 , R и $K_{п}$. В качестве бифуркационного параметра примем $K_{п}$. Расчет его граничного значения $K_{п,гр}$, при котором происходит нарушение устойчивости САР, организован следующим образом. При фиксированных значениях параметров U_3 и R и

изменении параметра K_n с определенным шагом на каждом шаге вычисляются координаты неподвижной точки i_N , $u_{и,N}$, $t_{с,N}$ и $t_{ф,N}$, элементы матрицы \mathbf{J} и ее собственные числа в этой точке до тех пор, пока не нарушится какое-либо из неравенств (14). После этого фиксируется номер l наибольшего по модулю собственного числа и находится граничное значение $K_{n,гр}$ как корень (функция root) относительно K_n следующего нелинейного уравнения

$$K_{n,гр} = \text{root}\left(\left|\lambda_l\right|_{\mathbf{x}=\mathbf{x}_N}\right) = 1. \quad (15)$$

Граничные поверхности областей устойчивости САР, рассчитанные при $f_d = f_k$ и $f_d = 2f_k$ по уравнению (15) в диапазонах значений параметра U_3 от 1 В до 15 В и параметра R от 0.05 Ом до 0.35 Ом, приведены на рис. 3. Области устойчивости САР располагаются ниже соответствующих поверхностей. Здесь необходимо отметить, что при значениях K_n , больших его граничного значения $K_{n,гр}$, в указанных диапазонах значений U_3 и R наблюдается классический механизм потери устойчивости САР, а именно бифуркация удвоения периода колебаний, причем только за счет нарушения второго из неравенств (14).

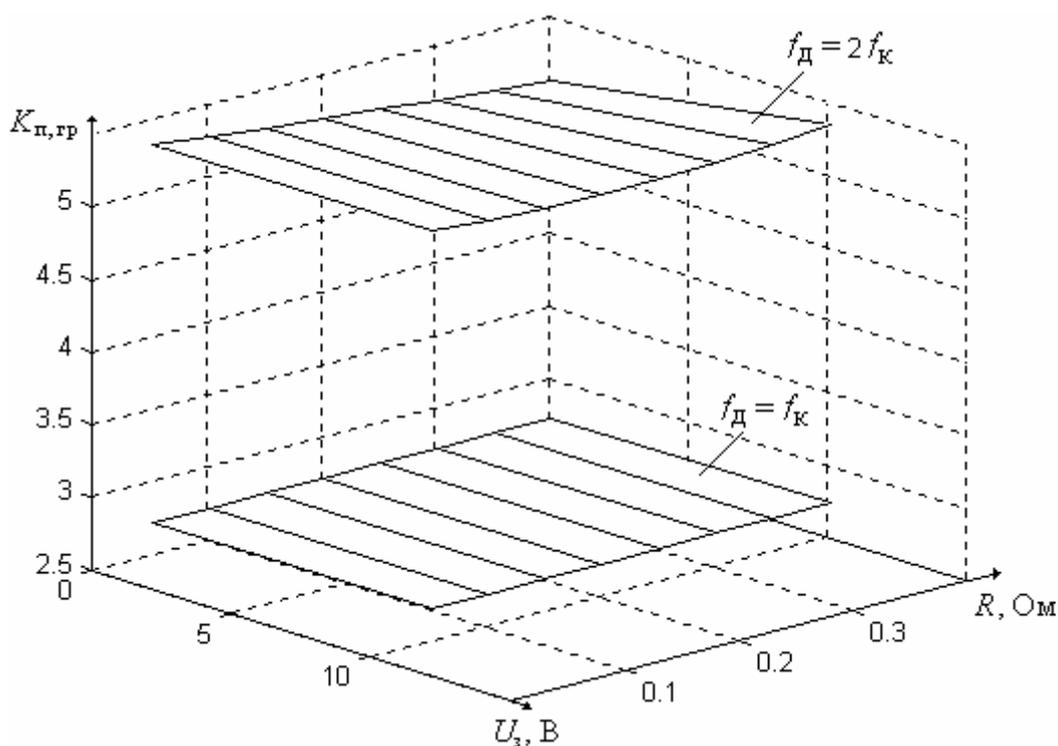


Рис. 3. Граничные поверхности областей устойчивости цифровой САР

Исследование микроструктуры полученных граничных поверхностей показывает, что при $f_d = f_k$ граничное значение $K_{n,гр}$ параметра K_n практически не изменяется, а при $f_d = 2f_k$ претерпевает незначительные изменения при изменении параметров U_3 и R относительно их расчетных значений $R = 0.25$ Ом и $U_3 = 10$ В. Исследование макроструктуры этих поверхностей показывает, что при $f_d = 2f_k$ граничное значение $K_{n,гр}$ параметра K_n примерно в два раза больше, чем при $f_d = f_k$. При этом, как видно из рис. 3, расчетное значение $K_n = 1$ параметра K_n , полученное в результате синтеза,

располагается значительно ниже граничной поверхности при $f_d = f_k$, а, следовательно, и ниже граничной поверхности при $f_d = 2f_k$.

Таким образом, на основе полученных результатов можно сделать следующие выводы:

1. выбор значения коэффициента передачи пропорционального канала регулятора, большего, чем его расчетное значение, получаемое в результате синтеза по методике, приведенной в работе [5], и меньшего, чем его граничное значение, получаемое в результате анализа, позволяет повысить ее быстродействие, не приводя при этом к возникновению в ней автоколебаний на субгармонических частотах в широких диапазонах возможных изменений сопротивления нагрузки и сигнала задания тока нагрузки относительно их расчетных значений;
2. увеличение частоты дискретизации аналого-цифровых преобразователей микроконтроллера вдвое по отношению к частоте коммутации силовых ключей широтно-импульсного преобразователя приводит к значительному расширению областей устойчивости системы, что позволяет еще более повысить быстродействие системы.

Литература

1. Глазенко Т.А., Синицын В.А., Толмачев В.А. Сравнительный анализ динамических характеристик транзисторных широтно-импульсных преобразователей. // Электротехника. 1988. № 3. С. 70–75.
2. Белов Г.А. Динамика импульсных преобразователей. Чебоксары: Изд-во Чуваш. ун-та, 2001. 528 с.
3. Micro Converter ADuC831, 12-Bit ADCs and DACs With Embedded 62 kBytes Flash MCU. Data Sheet. Analog Devices. 2002.
4. A DMOS 3A, 55V, H-Bridge: The LMD18200. Data Sheet. National Semiconductor Corporation. 1999.
5. Гурьянов В.А., Кротенко В.В. Цифровая система управления источника тока, построенного на основе транзисторного ШИП // Научно-технический вестник СПбГИТМО. Выпуск 3. СПб: Изд-во СПбГИТМО (ТУ), 2001. С. 120–125.
6. Охоткин Г.П. Анализ и синтез САР тока с ПИ-регулятором и ШИМ-1 // Информационные технологии в электротехнике и электроэнергетике / Материалы III Всерос. науч.-техн. конф. Чебоксары: Изд-во Чуваш. ун-та, 2000. С. 153–159.

МОДЕЛИРОВАНИЕ ЭЛЕКТРОМАГНИТНЫХ ПРОЦЕССОВ В СИЛОВЫХ ЦЕПЯХ N-МОДУЛЬНЫХ ИМПУЛЬСНЫХ ИСТОЧНИКОВ ТОКА ПРОГРАММИРУЕМОЙ ФОРМЫ

В.А. Толмачев, М.В. Никитина

К числу актуальных проблем электронного приборостроения относится проблема построения систем управления современными источниками электропитания с высокими динамическими показателями. Использование транзисторных широтно-импульсных преобразователей (ШИП) в энергетических подсистемах позволяет существенно поднять как энергетические, так и динамические характеристики источников. Предельные динамические характеристики их в конечном итоге определяются максимально-допустимой частотой коммутации силовых ключей ШИП [1]. В случае ограниченной частоты коммутации повышение предельных динамических показателей источника возможно при использовании в системе управления многомодульных усилительно-преобразовательных устройств (УПУ) с многофазным принципом синхронизации составляющих ШИП, работающих на общую нагрузку. Технические требования и принципы построения таких систем управления рассмотрены в работе [2].

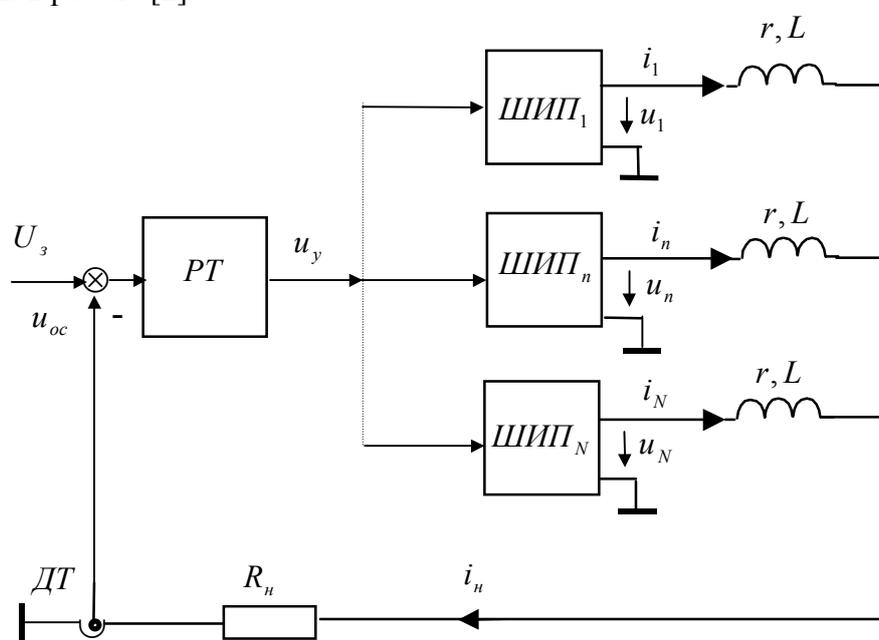


Рис. 1. Структурная схема цепи питания нагрузки

Вопросы выбора структуры системы управления источников с многомодульными УПУ рассмотрены в работе [3]. При незначительном разбросе параметров силовых цепей каналов УПУ может быть выбрана одноконтурная система управления со структурной схемой, представленной на рис. 1. Система содержит N реверсивных ШИП, работающих на общую нагрузку с активным сопротивлением R_n , и один контур регулирования тока нагрузки i_n . Каждый ШИП содержит силовой каскад мостового типа на четырех транзисторных ключах с напряжением питания E_n и широтно-импульсный модулятор (ШИМ), осуществляющий коммутацию силовых ключей с постоянным периодом T_k по несимметричному закону и обеспечивающий род модуляции ШИМ2 и глубину модуляции M ; при многофазном способе синхронизации опорные напряжения u_{on} одинаковой формы, амплитуды и частоты разнятся

начальными фазами, так что начальная фаза n -го ШИП смещена относительно начальной фазы $(n+1)$ -го ШИП на угол, соответствующий временному интервалу T_k / N . Напряжение на нагрузке имеет вид импульсов с амплитудой E_n , относительной длительностью γ , пропорциональной величине входного выходного сигнала u_y ПИ-регулятора тока (РТ), и полярностью, определяемой полярностью последнего. На входе регулятора осуществляется сравнение напряжения U_s , пропорционального заданному значению тока нагрузки I_s , и напряжения u_{oc} , поступающего с датчика тока нагрузки ДТ с коэффициентом передачи k_{om} и пропорционального истинному значению тока нагрузки. В представленной структуре разделительные дроссели каналов с параметрами r , L осуществляют одновременно и сглаживание пульсаций токов каналов i_n .

С позиции теории автоматического управления (ТАУ) система управления является многосвязной, поскольку электромагнитные процессы в каналах взаимосвязаны через нагрузку. Трудности анализа таких систем обусловлены высоким порядком системы дифференциальных уравнений, зависящим от числа модулей N , и наличием в каждом из них нелинейного импульсного звена – ШИП.

Обзор большого числа публикаций, посвященных теории замкнутых систем управления с широтно-импульсной модуляцией сигналов управления, позволяет заключить, что наиболее перспективным и практически применимым на современном этапе является двухэтапный подход, при котором на первом этапе осуществляется предварительный синтез с использованием приближенных непрерывных моделей (линейных или нелинейных), а на втором – уточненный анализ на ПЭВМ результатов предварительного синтеза с использованием имитационных моделей, достаточно точно отражающих как электромагнитные процессы в системе "ШИП-нагрузка", так и дискретный характер сигналов в информационной подсистеме.

Вопросы предварительного синтеза рассматриваемой системы управления детально освещены в работе [4]. Результатами такого синтеза при заданной предельно-допустимой частоте коммутации силовых ключей транзисторных ШИП являются число модулей N энергетической подсистемы источника, параметры N -канального сглаживающего фильтра (L) и параметры ПИ-регулятора (k_I и T_I).

В данной публикации проводится сравнительный анализ двух подходов к построению программ расчета фазовых координат силовых цепей источников с параметрами, определенными на этапе предварительного синтеза.

При первом подходе программа расчета фазовых координат строится на основе численного интегрирования системы уравнений состояния силовой цепи на интервалах постоянства элементов вектора управления в виде выходных напряжений ШИП и определении моментов изменения этих элементов в соответствии с алгоритмом работы ШИП, при втором подходе алгоритмы вычисления фазовых координат строятся на основе аналитических решений поинтервальных уравнений состояния (далее – модели первого и второго типа, соответственно).

Электромагнитные процессы в силовой цепи источника, представленного на рис. 1, на интервалах постоянства выходного напряжения ШИП n -го канала можно описать системой линейных дифференциальных уравнений с постоянными коэффициентами

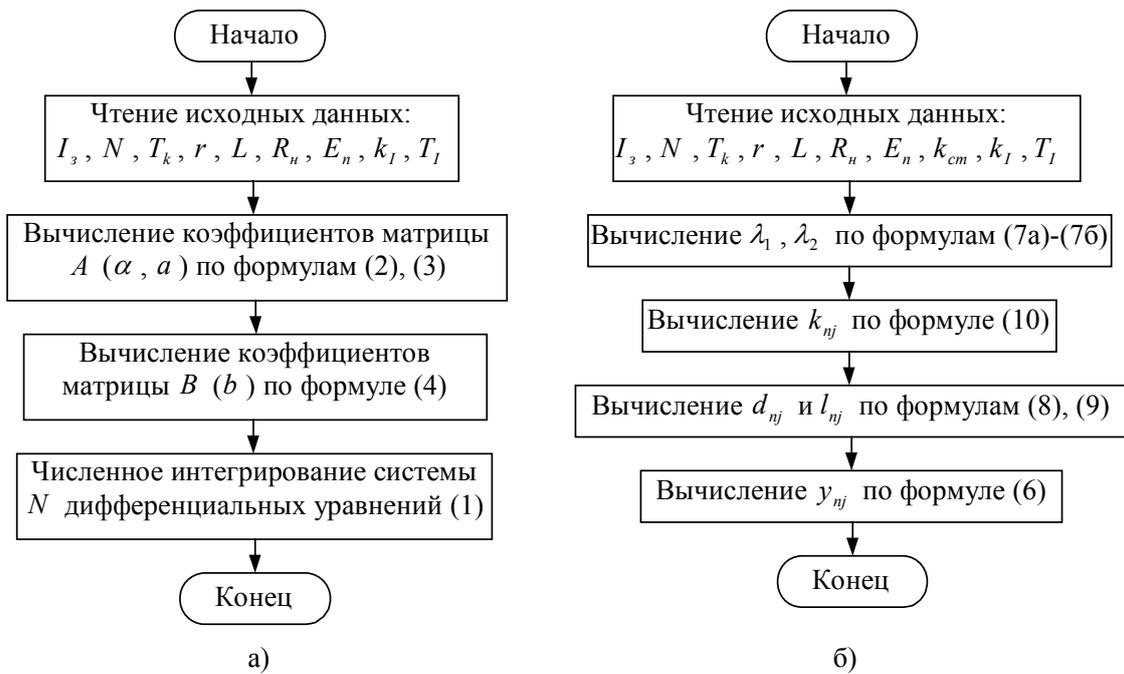


Рис. 2. Блок-схема алгоритма расчета фазовых координат

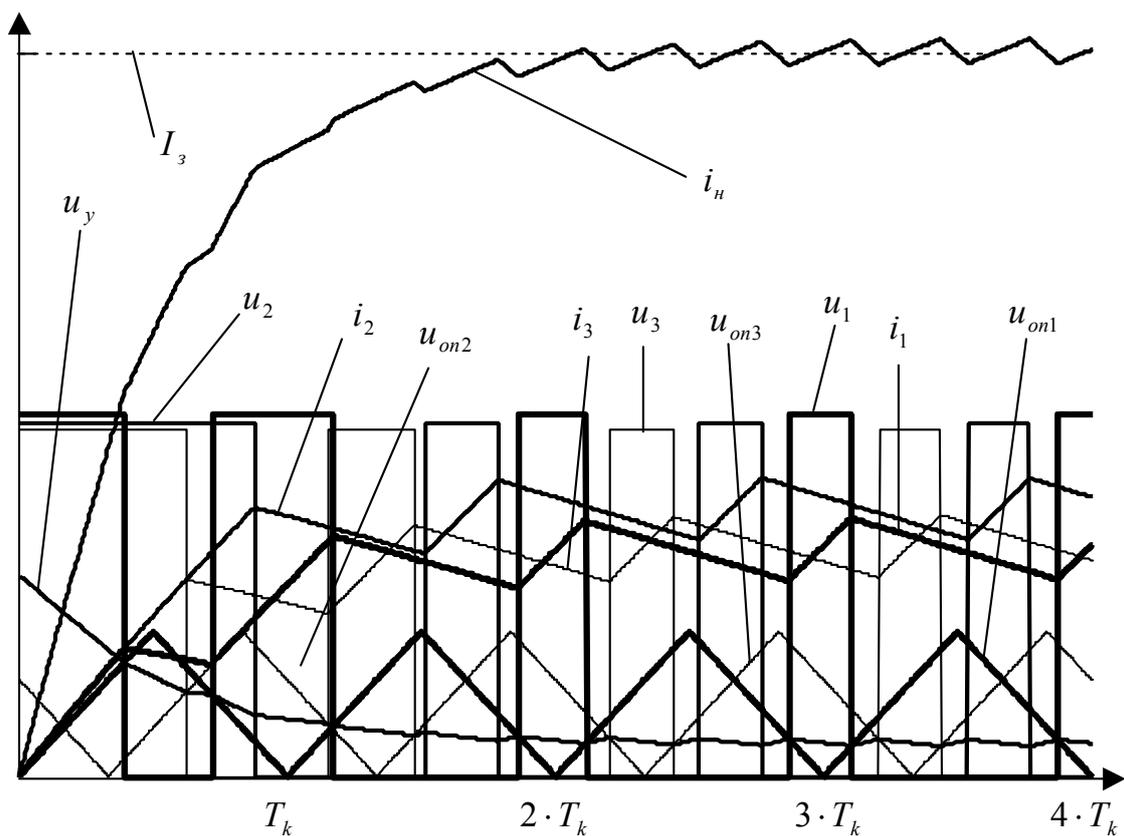


Рис. 3. Электромагнитные процессы в силовых цепях, полученные на модели первого типа

На рис. 2 приведены блок-схемы алгоритмов расчета фазовых координат для моделей первого (рис. 2а) и второго (рис. 2б) типов. Как видно, модели первого типа требуют решения системы дифференциальных уравнений N -го порядка, естественно, что вычислительные затраты и время интегрирования системы уравнений растут

пропорционально числу модулей N , тогда как у моделей второго типа, позволяющих избежать процедуры численного интегрирования системы, вычислительные затраты связаны лишь с определением коэффициентов матриц K , D и L .

На рис. 3 и 4 представлены результаты моделирования электромагнитных процессов в силовых цепях источников с одинаковыми параметрами ($N = 3$, $R_n = 0.1$ Ом, $r = 0.03$ Ом, $L = 25$ мГн, $T_k = 25$ мс, $E_n = 24$ В, $I_s = 50$ А, $T_l = 76$ мс, $k_l = 1.39$), полученные на математических моделях первого (рис. 3) и второго (рис.4) типов в среде MathCad, где i_n – ток нагрузки, I_s – заданное значение тока нагрузки, i_1, i_2, i_3 – токи сглаживающих дросселей каналов, u_y – выходной сигнал регулятора тока, $u_{on1}, u_{on2}, u_{on3}$ – опорные напряжения ШИП каналов, u_1, u_2, u_3 – выходные напряжения ШИП каналов.

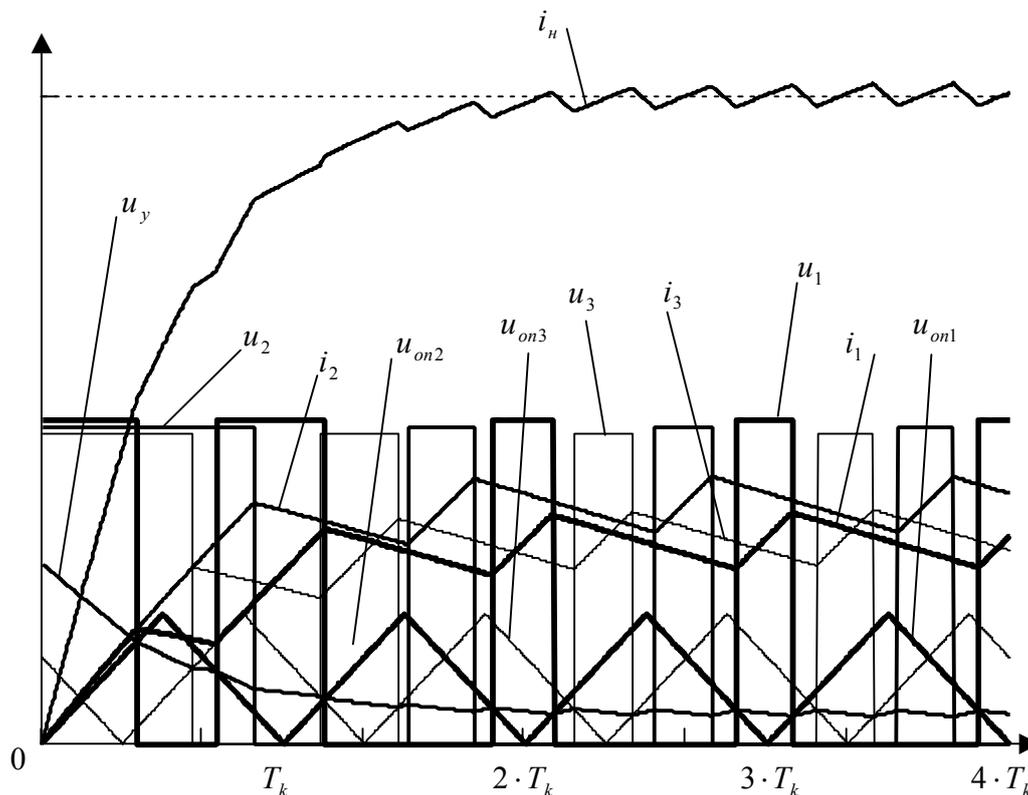


Рис. 4. Электромагнитные процессы в силовых цепях, полученные на модели второго типа

Как видно, результаты моделирования электромагнитных процессов, полученные на моделях обоих типов, совпадают с высокой точностью, при этом программа второго типа в среде MathCad выполняется на порядок быстрее.

Таким образом, для моделирования электромагнитных процессов систем высокого порядка ($N > 2$) целесообразнее применять математические модели второго типа, т.е. построенные на основе аналитических решений уравнений состояния, что позволит существенно снизить вычислительные затраты.

Литература

1. Глазенко Т.А., Сеницын В.А., Толмачев В.А. Сравнительный анализ динамических характеристик транзисторных широтно-импульсных преобразователей // Электротехника. 1988. №3. С.70-75.

2. Сеницын В.А., Толмачев В.А., Томасов В.С. Принципы построения и пути совершенствования технических характеристик мощных источников электропитания с произвольной формой выходного параметра // Изв. вузов. Приборостроение. 1996. Т.39. № 4. С.47–54.
3. Толмачев В.А. Синтез системы управления источника тока с многоканальным питанием нагрузки. // Изв. вузов. Приборостроение. 1995. Т.38. № 11–12. С. 40–43.
4. Толмачев В.А. Синтез системы управления многомодульного источника тока с индуктивным фильтром. // Изв. вузов. Приборостроение. 2001. Т. 44. № 3. С. 16–22.
5. Толмачев В.А. Электромагнитные процессы в силовой цепи программируемого источника электропитания с нагрузкой резистивного характера // Изв. вузов. Приборостроение. 1996. Т. 39. №3. С. 50–54.

АНАЛИЗ ЭЛЕКТРОМАГНИТНЫХ ПРОЦЕССОВ ПРИ ВКЛЮЧЕНИИ АКТИВНЫХ ВЫПРЯМИТЕЛЕЙ НАПРЯЖЕНИЯ В ПИТАЮЩУЮ СЕТЬ

В.С. Томасов, П.А. Борисов

Стремительное развитие электротехнических комплексов и систем на базе устройств полупроводниковой преобразовательной техники, увеличение их мощности оказывают значительное влияние на качество электрической энергии питающей сети. В связи с этим требования к показателям качества таких систем постоянно ужесточаются, вновь вводимые стандарты заставляют разработчиков отказываться от простых выпрямительных схем на входе полупроводниковых преобразователей различных типов, а также регулируемых электроприводов на базе управляемых выпрямителей. О государственной значимости этой проблемы свидетельствует принятие Закона "Об электромагнитной совместимости" и нового российского ГОСТ на качество электрической энергии [1]. Поэтому создание комплексных методик и рекомендаций по выбору принципиальных схем построения энергоподсистем в зависимости от рабочих мощностей силовых установок является актуальной проблемой.

В этой связи в статье основное внимание уделяется силовым схемам, которые позволяют наиболее перспективно решать проблемы повышения энергетической эффективности и экономичности использования электрической энергии в полупроводниковых преобразователях и регулируемых электроприводах как постоянного, так и переменного тока.

Практическое применение чаще всего находит схема выпрямителя, нагрузка которого начинается с емкостного элемента, где конденсатор используется в качестве сглаживающего фильтра. Наиболее эффективным из числа описанных в литературе способов повышения качества электропотребления является способ, основанный на введении в схему неуправляемого выпрямителя с емкостным фильтром одного или нескольких полностью управляемых полупроводниковых приборов, работающих в ключевом режиме с широтно-импульсной модуляцией (ШИМ). Схемы, построенные по такому принципу носят в англоязычной литературе название ШИМ-выпрямителей или активных выпрямителей напряжения (АВН).

Один из возможных способов построения АВН состоит в введении между неуправляемым выпрямителем (однофазным или трехфазным) и фильтром дополнительного силового блока, выполненного по схеме импульсного повышающего регулятора постоянного напряжения, в состав которого входят токоограничивающий дроссель, ключевой прибор и отсекающий диод. Входной ток питающей сети, близкий к синусоидальному, может быть получен при неизменной в течение периода частоты сети длительности включенного состояния ключевого прибора, что существенно упрощает систему управления. На интервале, когда ключ открыт, токоограничивающий дроссель запасает энергию; когда ключ выключен, происходит передача энергии в конденсатор фильтра не только из сети, но и энергии, накопленной в дросселе [2]. Указанные схемы позволяют формировать непрерывный ток питающей сети и находят широкое применение в электротехнических комплексах и системах, не требующих работы нагрузки в генераторном режиме, благодаря простоте их реализации. Однако построение полупроводниковых преобразователей по таким схемам возможно только в случаях, когда не требуется рекуперация энергии в питающую сеть, так как схемы обладают односторонней проводимостью.

Схема АВН, обеспечивающая двухсторонний обмен энергией между питающей сетью и нагрузкой, представлена на рис. 1. По аналогичной схеме строится трехфазный мостовой АВН. Эти схемы по сути представляют собой обращенное относительно зажимов питания и нагрузки включение схем одно- и трехфазного автономного

инвертора напряжения, что в сочетании с импульсно-модуляционными алгоритмами переключения ключей обеспечивает возможность работы как в выпрямительном, так и в инверторном режимах работы.

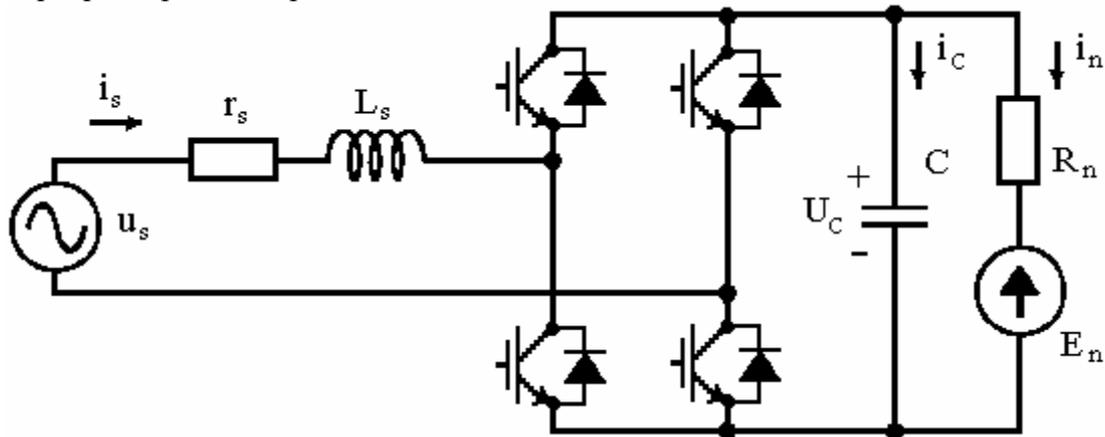


Рис. 1. Принципиальная электрическая схема однофазного мостового АВН

Электромагнитные процессы для однофазной мостовой схемы АВН, характеризующиеся изменением мгновенных значений токов и напряжений, можно описать следующей системой контурных, узловых и элементных уравнений [3]:

$$u_s = r_s \cdot i_s + L_s \cdot \frac{di_s}{dt} + U_C \cdot f_v, \quad (1)$$

$$i_c = C \cdot \frac{dU_C}{dt}, \quad (2)$$

$$i_s \cdot f_v = i_c + i_n, \quad (3)$$

$$U_C = U_n, \quad (4)$$

$$i_n = \frac{U_n - E_n}{R_n}, \quad (5)$$

где $u_s = U_m \cdot \sin(\omega_s \cdot t + \psi_s)$ – мгновенное значение напряжения питающей сети, изменяющееся по гармоническому закону, U_m – амплитуда, $\omega_s = 2 \cdot \pi \cdot f_s$ – угловая частота, ψ_s – начальная фаза, i_s – сетевой ток, r_s – суммарное активное сопротивление продольной ветви АВН, L_s – суммарное индуктивное сопротивление продольной ветви АВН, U_C – выпрямленное напряжение на конденсаторе фильтра, C – емкость конденсатора фильтра, R_n – резистивное сопротивление нагрузки, E_n – противо-ЭДС нагрузки, f_v – коммутационная функция, учитывающая свойства работы диодов и полностью управляемых ключей схемы. Тогда на основании системы уравнений (1)–(5) получим уравнения переменных состояния цепи i_s и U_C :

$$\frac{di_s}{dt} = u_s \cdot \frac{1}{L_s} - \frac{r_s}{L_s} \cdot i_s - U_C \cdot f_v \cdot \frac{1}{L_s}, \quad (6)$$

$$\frac{dU_C}{dt} = \frac{i_s}{C} \cdot f_v - \frac{U_C}{C} \cdot \frac{1}{R_n}. \quad (7)$$

Разработанная в пакете MathCad математическая модель позволила решить с учетом f_v уравнения переменных состояния цепи i_s и U_C (7), (8), описывающие однофазный мостовой АВН. Разработана также математическая модель трехфазного мостового АВН. Управление ключами осуществляется по алгоритму двухсторонней

симметричной ШИМ с частотой f_k , при этом модулирующий сигнал изменяется по синусоидальному закону $U_k = k \cdot U_{mk} \cdot \sin(2 \cdot \pi \cdot f_k \cdot t + \psi_k)$, где k меняется в диапазоне от 0 до 1, U_{mk} – амплитуда опорного сигнала пилообразной формы, ψ_k – начальная фаза модулирующего сигнала, $U_{\text{ШИП}}$ – амплитуда импульсов управления широтно-импульсного преобразователя.

На рис. 2 представлены результаты моделирования электромагнитных процессов. Числовые параметры при моделировании были взяты следующие: $U_m = 311$ В, $f_s = 50$ Гц, $f_k = 5$ кГц, $r_s = 0.5$ Ом, $L_s = 0.47$ мГн, $C = 2000$ мкФ, $R_n = 10$ Ом, $E_n = 0$ В. Демонстрируется возможность двухстороннего обмена энергией между питающей сетью и нагрузкой.

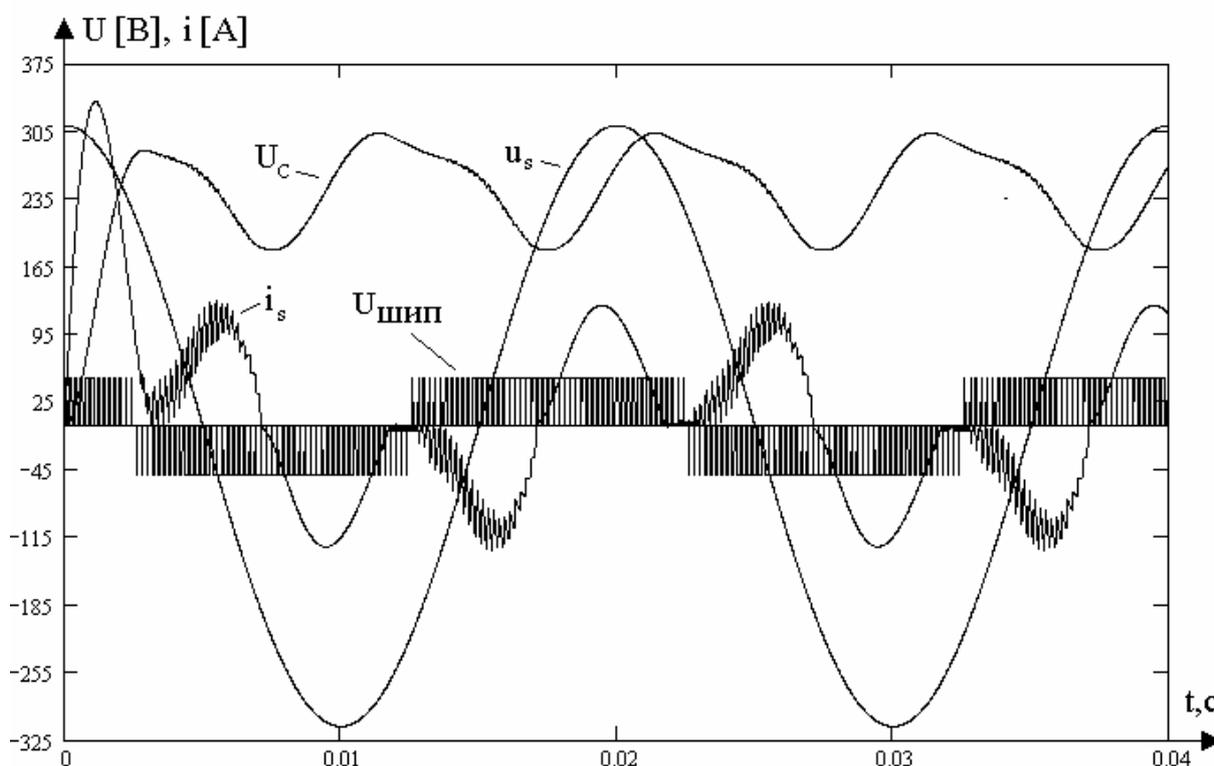


Рис. 2. Переходной процесс включения однофазного мостового АВН на незаряженный конденсатор при $\psi_s = 90^\circ$, $\psi_k = 135^\circ$, $k = 0.9$

Необходимо учитывать, что в литературе чаще всего представлены методики анализа выпрямителей для квазистационарного режима. Вместе с тем очевидно, что наиболее тяжелым режимом работы всей энергоподсистемы является включение ее в питающую сеть, что, как правило, сопровождается увеличением электромагнитных нагрузок и изменением режима работы схемы. Это особенно касается работы выпрямителя с бестрансформаторным входом, поскольку токи вентиля в нем не ограничены сопротивлениями фаз. В большинстве работ пусковой режим рассматривается для начальной фазы напряжения питающей сети $\psi_s = 0$. Однако увеличение электромагнитных нагрузок следует ожидать при пуске с начальной фазой $\psi_s = 90^\circ$ [4]. Также следует отметить тот факт, что предотвратить значительный бросок тока при включении на незаряженный конденсатор в переходном процессе с помощью соответствующего управления ключами за счет регулирования интервалов их

открытого состояния не удастся. Это объясняется тем фактом, что весь пусковой ток заряда конденсатора фильтра протекает через диоды, которые запираются только к моменту спада тока до нуля.

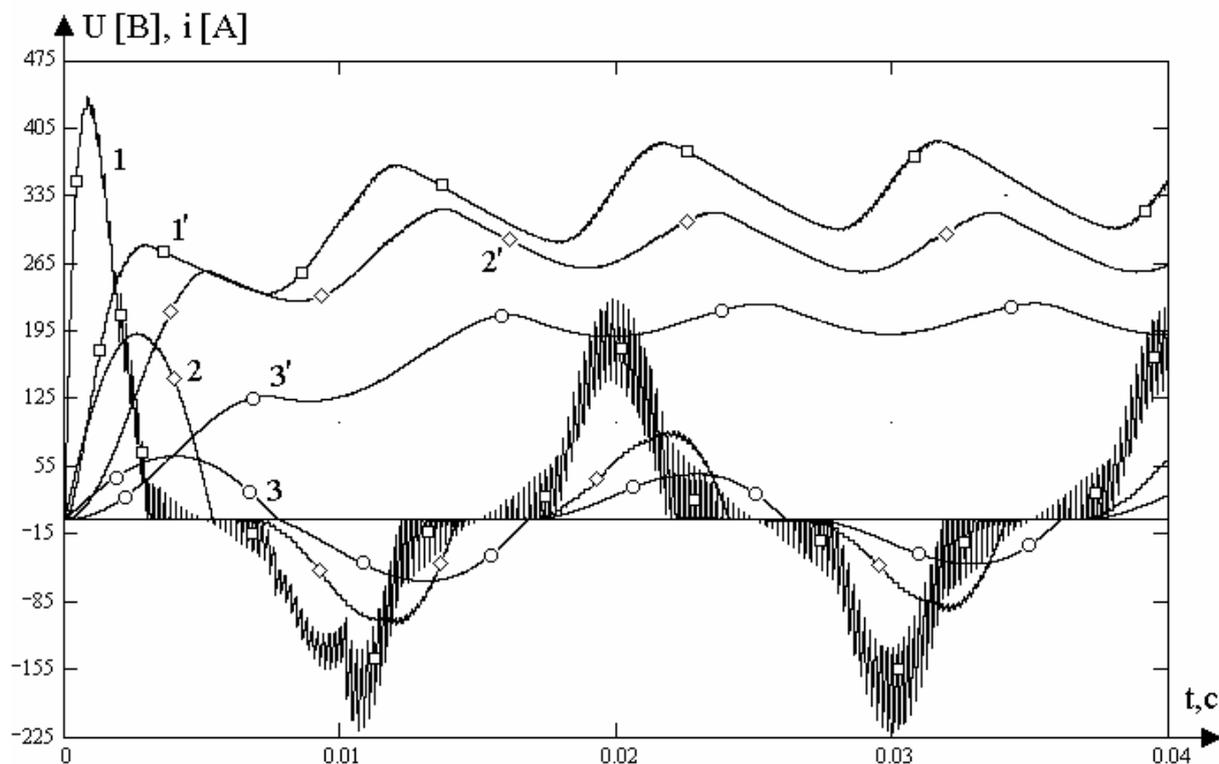


Рис. 3. Сравнительные кривые сетевого тока i_s и напряжения на конденсаторе фильтра U_c для схемы однофазного мостового АВН при различных значениях индуктивности L_s

Из рис. 3 видно что увеличение L_s , как и следовало ожидать, уменьшает бросок пускового тока, однако при этом фазовый сдвиг основной гармоники сетевого тока приближается к $\pi / 2$. Числовые параметры при моделировании были взяты следующие: $U_m = 311$ В, $f_s = 50$ Гц, $f_k = 5$ кГц, $r_s = 0.5$ Ом, $C = 2000$ мкФ, $R_n = 10$ Ом, $E_n = 0$ В, $\psi_s = 90^\circ$, $\psi_k = -90^\circ$, $k = 0.18$, кривым 1 соответствует $L_s = 0.235$ мГн, кривым 2 соответствует $L_s = 2.35$ мГн, кривым 3 соответствует $L_s = 11.75$ мГн (i_s – номер без штриха, U_c – номер со штрихом). АВН работает как импульсный повышающий регулятор постоянного напряжения. Наличие фазового сдвига основной гармоники тока приводит к снижению коэффициента мощности, в силу чего такой режим, приближающийся к режиму непрерывного тока, нежелателен. Увеличение L_s сопровождается также удорожанием всей установки в целом и ухудшением массогабаритных показателей.

Выводы

Выбор параметров схем АВН без учета электромагнитных нагрузок в течение переходных процессов включения может привести к неработоспособности или к существенному завышению установленных мощностей полупроводниковых приборов

и элементов фильтра, а также к ухудшению массогабаритных, энергетических и надежностных показателей качества.

В современных устройствах для устранения этих недостатков возможно применение следующих способов:

- для полупроводниковых преобразователей малых и средних мощностей в продольную ветвь фильтра вводят дополнительный резистор для ограничения зарядного тока конденсатора фильтра. После заряда конденсатора резистор необходимо шунтировать тиристором или полностью управляемым ключом;
- для полупроводниковых преобразователей средних и больших мощностей введение дополнительного дросселя и полностью управляемого ключа в продольную ветвь фильтра позволяет не только ограничить пусковые токи, но и реализовать на его основе понижающий импульсный регулятор постоянного напряжения, таким образом обеспечивая возможность работы как в режиме повышения, так и понижения напряжения;
- для систем регулируемого электропривода, в которых требуется обеспечивать высокую динамику, указанные до этого способы могут приводить к некоторому ухудшению быстродействия привода. Поэтому для таких задач возможны варианты построения силовых схем в которых неуправляемые диоды заменяются тиристорами или полностью управляемыми ключами [5].

Литература

1. ГОСТ 13109-97. Электрическая энергия. Электромагнитная совместимость. Нормы качества электрической энергии в системах электроснабжения общего назначения. М.: Госстандарт, 1998.
2. Лабунцов В.А., Дайжун Ч. Трехфазный выпрямитель с емкостным фильтром и улучшенной кривой потребляемого из сети тока. // Электричество. 1993. № 6. С. 45–48.
3. Ефимов А.А., Шрейнер Р.Т. Активные преобразователи в регулируемых электроприводах переменного тока. / Под общей ред. д-ра техн. наук, проф. Р.Т. Шрейнера. Новоуральск: Изд-во НГТИ, 2001. 250 с.
4. Глазенко Т.А., Томасов В.С. Методика анализа переходных процессов в выпрямителях с RC-фильтрами // Изв. вузов. Приборостроение. 1994. Т. 37. № 11–12. С. 45–53.
5. Зиновьев Г.С. Прямые методы расчета энергетических показателей вентильных преобразователей. Новосибирск: Изд-во Новосиб. ун-та, 1990. 220 с.

НАШИ АВТОРЫ

Авдеева Галина Сергеевна – студентка биолого-почвенного факультета

Алиев Тауфик Измаилович — доктор технических наук, профессор, зав. кафедрой вычислительной техники

Болгаров Иван Сергеевич – студент 5-го курса

Борисов Павел Александрович – аспирант кафедры электротехники и прецизионных электромеханических систем

Васильев Владимир Николаевич – доктор технических наук, профессор, ректор

Гераничев Владимир Николаевич – зам. начальника отдела ОКБ "Электроавтоматика"

Герасимов Дмитрий Николаевич – студент 4-го курса

Гончаров Денис Александрович – аспирант кафедры вычислительной техники

Гришин Михаил Викторович – аспирант кафедры вычислительной техники

Громов Геннадий Юрьевич – старший преподаватель кафедры вычислительной техники

Демин Анатолий Владимирович – доктор технических наук, профессор кафедры информатики и прикладной математики

Денисов Константин Михайлович – старший преподаватель кафедры электротехники и прецизионных электромеханических систем

Дмитриева Ольга Юрьевна – сотрудник ОКБ "Электроавтоматика"

Дорофеева Татьяна Борисовна – кандидат биологических наук, энто-фитопатолог, Станция защиты зеленых насаждений Санкт-Петербурга, Испытательная лаборатория фитосанитарного мониторинга зеленых насаждений.

Дорохин Денис Алексеевич – аспирант кафедры вычислительной техники

Елькин Игорь Владимирович – магистрант кафедры вычислительной техники

Ермыкин Алексей Александрович – аспирант кафедры вычислительной техники

Жданов Иван Николаевич – аспирант кафедры электротехники и прецизионных электромеханических систем

Жуков Николай Владимирович – студент 4-го курса

Зинатуллин Марат Минисагитович – аспирант кафедры информатики и прикладной математики

Ищенко Алексей Петрович – старший преподаватель кафедры компьютерных технологий

Кириллов Владимир Васильевич – кандидат технических наук, профессор кафедры вычислительной техники

Ключев Аркадий Олегович – кандидат технических наук, доцент кафедры вычислительной техники

Ковязин Рустам Раисович – аспирант кафедры вычислительной техники

Кожевникова Лариса Владиславовна – программист кафедры систем управления и информатики

Кокшаров Дмитрий Николаевич – студент 5-го курса

Коробейников Анатолий Григорьевич – доктор технических наук, профессор кафедры проектирования компьютерных систем

Королев Александр Юрьевич – студент 5-го курса

Коротков Константин Георгиевич – доктор технических наук, профессор кафедры проектирования компьютерных систем

Косяков Михаил Сергеевич – аспирант кафедры вычислительной техники

Котов Алексей Владимирович – студент 5-го курса

Кустарев Павел Валерьевич – кандидат технических наук, доцент кафедры вычислительной техники

Лукичев Александр Николаевич – студент 4-го курса

Лукичев Дмитрий Вячеславович – аспирант кафедры электротехники и прецизионных электромеханических систем

Лукьянова Галина Владимировна – аспирантка кафедры систем управления и информатики

Макеев Сергей Анатольевич – аспирант кафедры вычислительной техники

Маковецкая Наталья Андреевна – аспирантка кафедры вычислительной техники

Мигуш Сергей Алексеевич – аспирант кафедры систем управления и информатики

Никитина Мария Владимировна – аспирант кафедры электротехники и прецизионных электромеханических систем

Никифоров Владимир Олегович – доктор технических наук, профессор, проректор ГИТМО

Новиков Сергей Николаевич – студент 6 курса

Ожиганов Александр Аркадьевич – доктор технических наук, профессор кафедры вычислительной техники

Окулевич Владимир Викентьевич – аспирант кафедры вычислительной техники

Осипов Дмитрий Владимирович – аспирант кафедры электротехники и прецизионных электромеханических систем

Патиев Антон Александрович – студент 5-го курса

Петров Александр Александрович – магистрант кафедры вычислительной техники

Платунов Алексей Евгеньевич – кандидат технических наук, доцент кафедры вычислительной техники

Прияткин Николай Сергеевич – аспирант кафедры проектирования компьютерных систем

Рахманов Виктор Юрьевич – аспирант кафедры вычислительной техники

Рогачевский Александр Маркович – аспирант кафедры проектирования компьютерных систем

Романова Ева Борисовна – аспирант кафедры проектирования компьютерных систем

Рыбакова Наталья Анатольевна – аспирант кафедры проектирования компьютерных систем

Самохвалова Ольга Геннадьевна – аспирантка кафедры вычислительной техники Санкт-Петербургского государственного университета

Семенов Александр Евгеньевич – технолог предприятия "Светлана-полупроводники"

Семенов Алексей Михайлович – студент 6 курса

Семенов Кирилл Алексеевич – студент 5 курса

Сергеев Константин Александрович – ассистент кафедры систем управления и информатики

Скатын Андрей Владимирович – магистрант кафедры вычислительной техники

Скворцов Альберт Матвеевич – доктор технических наук, профессор кафедры проектирования компьютерных систем

Слепян Эрик Иосифович – доктор биологических наук, профессор, Национальный комитет экологической безопасности

Тимошенко Мария Юрьевна – аспирантка кафедры проектирования компьютерных систем

Ткалич Вера Леонидовна – доктор технических наук, профессор кафедры проектирования компьютерных систем

Толмачев Валерий Александрович – кандидат технических наук, доцент кафедры электротехники и прецизионных электромеханических систем

Тропченко Александр Ювенальевич – доктор технических наук, профессор кафедры вычислительной техники

Усольцев Александр Анатольевич – кандидат технических наук, доцент кафедры электротехники и прецизионных электромеханических систем

Ушаков Анатолий Владимирович – доктор технических наук, профессор кафедры систем управления и информатики

Фролкова Екатерина Григорьевна – кандидат технических наук, доцент кафедры проектирования компьютерных систем

Халецкий Роман Александрович – аспирант кафедры проектирования компьютерных систем

Худяков Владимир Андреевич – студент 4-го курса

Черемухин Виктор Станиславович – аспирант кафедры вычислительной техники

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ	3
УПРАВЛЕНИЕ И АВТОМАТИКА В ТЕХНИЧЕСКИХ СИСТЕМАХ.....	5
Лукиянова Г.В., Никифоров В.О. Алгоритм компенсации внешних детерминированных возмущений: операторный метод синтеза	5
Мигуш С.А., Герасимов Д.Н., Никифоров В.О. Разработка математической модели инжекторного двигателя	10
Сергеев К.А., Котов А.В., Кокшаров Д.Н. Измерительная система для учебно-исследовательского комплекса "Мобильные роботы"	19
Кожевникова Л.В., Ушаков А.В. Анализ процессов в системах с модуляцией – демодуляцией при внешнем конечномерном воздействии, с использованием кронекеровского модельного представления	25
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ПРОГРАММИРОВАНИЕ	30
Болгаров И.С., Патиев А.А. Резидентное и инструментальное ПО промышленных контроллеров сетевого ввода/вывода.....	30
Васильев В.Н., Громов Г.Ю., Кириллов В.В., Черемухин В.С., Дорохин Д.А., Скатын А.В. Концепция построения интегрированной информационной системы управления региональной системой образования	34
Гончаров Д.А. Алгоритмы выбора соединений в ПЛИС.....	39
Гришин М.В., Ожиганов А.А., Тропченко А.Ю. Сжатие изображений на основе выделения локальных однородных областей	50
Елькин И.В., Кустарев П.В. Модель абстрактных функциональных блоков	55
Ермыкин А.А, Алиев Т.И. Тестирование и анализ распределенных информационных систем, разработанных по технологии J2EE	62
Ключев А.О., Маковецкая Н.А. Проблемы тестирования системного программного обеспечения распределенных информационно-управляющих систем	67
Ковязин Р.Р., Платунов А.Е. Применение технологии OPC.....	71
Косяков М.С., Алиев Т.И. Сравнение архитектур построения серверов мультимедиа	77
Лукичев А.Н. Расширение возможностей лабораторного комплекса SDK-1.1.....	86
Макеев С.А., Алиев Т.И. Оптимизация конфигурируемых параметров маршрутизатора.....	91
Окулевич В.В. Разделение системных и прикладных задач при разработке информационно-управляющих систем	95
Петров А.А., Косяков М.С., Алиев Т.И. Оптимизация работы дисковой подсистемы сервера мультимедиа.....	99
Рахманов В.Ю., Тимченко Б.Д. Метрики в современном тестировании	108

Самохвалова О.Г., Окулевич В.В. Использование стандартных библиотек во встроенных системах (на примере стандартной библиотеки языка C++)	112
Самохвалова О.Г. Технологии повторного использования и реинжиниринга и системы реального времени	116

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ, АВТОМАТИЗАЦИЯ ЛОГИЧЕСКОГО ПРОЕКТИРОВАНИЯ И ЗАЩИТА ИНФОРМАЦИИ 120

Зинатуллин М.М. Создание виртуальной лаборатории проектирования тестового обеспечения	120
Королев А.Ю., Демин А.В. Применение кватернионов для моделирования зеркальных систем в динамике	126

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ, ТЕХНОЛОГИИ ЭЛЕМЕНТОВ И УЗЛОВ КОМПЬЮТЕРНЫХ СИСТЕМ..... 134

Дмитриева О.Ю. Единый информационный комплекс предприятия в технологической подготовке и производстве	134
Гераничев В.Н. Разработка технологии изготовления алюминиевых отливок сложных тонкостенных деталей способом вакуумно-пленочной формовки	137
Коробейников А.Г., Тимошенко М.Ю. Автоматизированная система мониторинга электромагнитных полей	142.
Новиков С.Н., Семенов А.Е., Фролкова Е.Г., Скворцов А.М., Халецкий Р.А. Исследование структурных дефектов межслойной изоляции в К-МОП ИС	145
Новиков С.Н., Семенов А.Е., Фролкова Е.Г., Скворцов А.М., Халецкий Р.А. Исследование пробивных напряжений межслойной изоляции в К-МОП ИС	148
Прияткин Н.С., Коротков К.Г., Слепян Э.И., Авдеева Г.С., Дорофеева Т.Б. Программно-аппаратный комплекс для установления характеристик газоразрядного свечения “ГРВ-комплекс” и перспективы его использования при анализе состояния организмов и выявлении экологического риска	151
Романова Е.Б. Применение САПР P-CAD-2001 при проектировании печатных плат..	154
Семенов А.М., Скворцов А.М., Халецкий Р.А. Исследование структур типа $Si_{ПК}-SiO_2-Si$ с окислами, полученными в различных режимах окисления	157
Семенов А.М., Скворцов А.М., Халецкий Р.А. Исследование МОП-структур с окислами, полученными при различных режимах окисления	159
Рогачевский А.М. Оценка существующих методов обработки и проверки КБО гражданских самолетов	162
Ткалич В.Л., Жуков Н.В. Проблемы надежности магнитоуправляемых контактов в системах управления	167
Ткалич В.Л., Худяков В.А. Создание библиотеки конечных элементов в приложении к пластинам и мембранам	169
Рыбакова Н.А. Исследование надежности сильфонных упругих элементов коммутационной техники	172

ЭЛЕКТРОТЕХНИКА, ЭЛЕКТРОМЕХАНИКА И ЭЛЕКТРОТЕХНОЛОГИИ 176

Денисов К.М., Жданов И.Н. Алгоритмическое моделирование цифрового фильтра ..	176
Лукичев Д.В., Усольцев А.А. Исследование эффективности применения	

искусственных нейронных сетей для выделения гладкой составляющей сигнала тока в электроприводе.....	181
Осипов Д.В., Толмачев В.А. Влияние частоты дискретизации на устойчивость цифровой системы автоматического регулирования тока с широтно-импульсным преобразователем	186
Толмачев В.А., Никитина М.В. Моделирование электромагнитных процессов в силовых цепях n-модульных импульсных источников тока программируемой формы.....	193
Томасов В.С., Борисов П.А. Анализ электромагнитных процессов при включении активных выпрямителей напряжения в питающую сеть	199
НАШИ АВТОРЫ.....	204

Научно-технический вестник СПб ГИТМО (ТУ). Выпуск 10.
Информация и управление в технических системах. / Под ред. Ю.А. Гатчина.
СПб: СПб ГИТМО(ТУ), 2003. 209 с.

**НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
СПб ГИТМО (ТУ)
Выпуск 10
ИНФОРМАЦИЯ И УПРАВЛЕНИЕ В ТЕХНИЧЕСКИХ СИСТЕМАХ**

Редактор Ю.А. Гатчин

Дизайн обложки В.А. Петров
Редакционно-издательский отдел СПб ГИТМО (ТУ)
Зав. РИО Н.Ф. Гусарова
Лицензия ИД № 00408 от 05.11.99.
Подписано в печать 15.07.03.
Заказ 710. Тираж 100 экз.