

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

---



ПОБЕДИТЕЛЬ КОНКУРСА ИННОВАЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ ВУЗОВ

# **НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК**

**САНКТ-ПЕТЕРБУРГСКОГО ГОСУДАРСТВЕННОГО  
УНИВЕРСИТЕТА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,  
МЕХАНИКИ И ОПТИКИ**

**сентябрь-октябрь 2008**

**№ 57**

**МЕХАТРОНИКА, ТЕХНОЛОГИИ,  
СИСТЕМЫ АВТОМАТИЗИРОВАННОГО  
ПРОЕКТИРОВАНИЯ**



## ГЛАВНЫЙ РЕДАКТОР

д.т.н., профессор В.О. Никифоров

## РЕДАКЦИОННАЯ КОЛЛЕГИЯ

д.т.н., доцент А.А. Бобцов, д.т.н. А.В. Бухановский,  
д.т.н., профессор В.А. Валетов, д.ф.-м.н., ст.н.с. Т.А. Вартамян,  
д.т.н. М.А. Ган, д.т.н., профессор Ю.А. Гатчин, д.т.н., профессор А.В. Демин,  
к.т.н., доцент Н.С. Кармановский (заместитель главного редактора),  
д.ф.-м.н., профессор Ю.Л. Колесников, д.ф.-м.н., профессор С.А. Козлов,  
д.т.н., профессор А.Г. Коробейников, д.т.н., профессор В.В. Курейчик,  
д.т.н., доцент Л.С. Лисицына, к.т.н., доцент В.Г. Мельников,  
д.т.н., профессор Ю.И. Нечаев, д.т.н., профессор Н.В. Никоноров,  
д.т.н., профессор А.А. Ожиганов, д.ф.-м.н., ст.н.с. Е.Ю. Перлин,  
д.т.н., профессор И.Г. Сидоркина, д.т.н. О.А. Степанов,  
д.т.н., профессор В.Л. Ткалич, д.т.н., профессор А.А. Шалыто

Секретарь – Г.О. Котелкова

Редактор – к.т.н., ст.н.с. Н.Ф. Гусарова

Адрес: 197101, Санкт-Петербург, Кронверкский пр., 49, СПбГУ ИТМО

Телефон: (812) 233 12 70

Факс: (812) 233 12 70 (с пометкой: для редакции  
Научно-технического вестника)

<http://books.ifmo.ru/ntv>

E-mail: [karmanov@mail.ifmo.ru](mailto:karmanov@mail.ifmo.ru)

---

Подписано к печати 20.10.2008 Тираж 350 экз. Заказ № 5(57)

---

Отпечатано в учреждении «Университетские телекоммуникации»  
Адрес: 197101, Санкт-Петербург, Кронверкский пр., 49

---

Журнал зарегистрирован Федеральной службой по надзору в сфере связи  
и массовых коммуникаций, свидетельство ПИ № ФС77-33466 от 10.10.2008 г.

© Санкт-Петербургский государственный университет  
информационных технологий, механики и оптики, 2008

УДК 615.478.5

**ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ТЕПЛООБМЕНА  
В ТВЕРДОТЕЛЬНОМ ТЕРМОСТАТИРУЮЩЕМ УСТРОЙСТВЕ**

О.В. Белова, М.А. Корнеева, Д.А. Мустафина

Представлены результаты расчетных исследований теплового состояния термостатирующего устройства на термоэлектрических элементах Пельтье. Для проведения исследований был разработан метод расчета в динамическом режиме, включающий расчет на основе одномерной модели, уточненный на основе трехмерной модели. Метод позволяет решать различные задачи совершенствования конструктивных параметров и разработки новых устройств подобного рода.

**Ключевые слова:** полимеразная цепная реакция, теплопроводность, термоэлектрические элементы Пельтье, радиатор, численное моделирование.

**Введение**

Термостатирующие устройства широко применяются в технике для поддержания заданной температуры в течение определенного промежутка времени. Иногда, помимо термостатирования объекта, необходимо изменять его тепловое состояние в соответствии с требуемым законом. Тогда возникает задача разработки методики, которая позволяла бы реализовывать программы управления с необходимыми скоростями нагрева/охлаждения и точностью поддержания температур.

Конструктивные схемы термостатирующих устройств в зависимости от используемого рабочего тела (газ, жидкость или твердое тело) можно разделить на пневматические, гидравлические и твердотельные. В данной работе рассматривается твердотельное термостатирующее устройство на термоэлектрических элементах Пельтье (рис. 1) для проведения медико-биологических исследований. Такой прибор обладает возможностью расширения рабочего диапазона температур и удобством эксплуатации. Пробирки с образцами 1 объемом 0,2 мл устанавливаются на пластине-держателе 2, содержащей 96 лунок под пробирки и цилиндрические углубления для снижения массы и инерционности детали. Подвод и отвод теплоты осуществляется с помощью термоэлектрических элементов Пельтье 3. Производится дополнительное конвективное охлаждение системы с помощью радиатора 4.

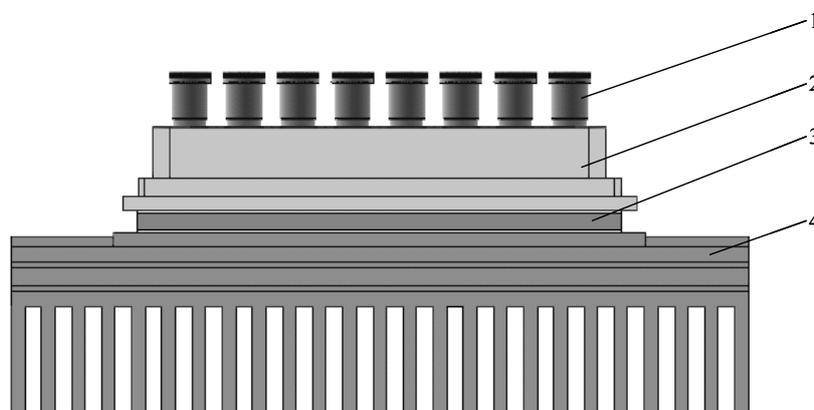


Рис. 1. Твердотельное термостатирующее устройство: 1 – пробирки с образцами; 2 – пластина-держатель; 3 – термоэлектрические элементы Пельтье; 4 – воздушный радиатор

Устройство работает по циклу, примерный вид которого представлен на рис. 2. Цикл состоит из четырех периодически сменяющих друг друга режимов: 1 – нагрев; 2 – охлаждение; 3 – выдержка при верхней температуре; 4 – выдержка при нижней температуре. В соответствии с каждым режимом изменяются величина и направление вектора плотности тока в термоэлементах. Закон изменения вектора плотности тока представляет закон управления термостатирующим устройством. Подбором этого закона можно добиться получения требуемой точности поддержания температур и скоростей нагрева/охлаждения.

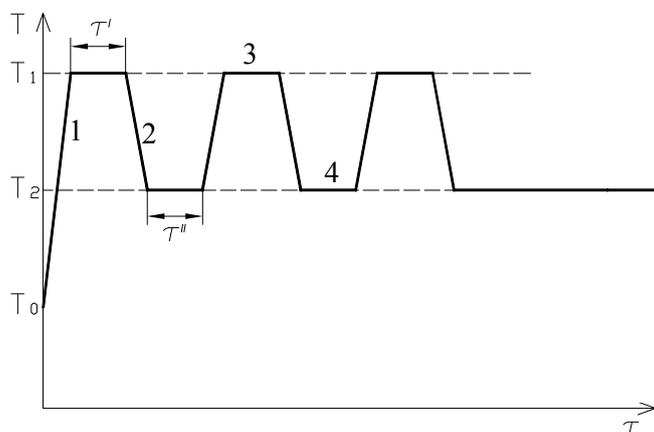


Рис. 2. Режим работы амплификатора: 1 – нагрев до  $T_1$ , 2 – охлаждение до  $T_2$ , 3 – выдержка при  $T_1$ , 4 – выдержка при  $T_2$

Известные методы расчета подобных термостатирующих устройств [1, 2] исследуют только статические тепловые состояния, что не позволяет эффективно рассмотреть тепловое состояние в динамическом режиме работы и оценить неоднородность теплового поля отдельных элементов устройства.

Целью работы является разработка метода расчета твердотельных термостатирующих устройств в динамическом режиме для проведения оценок влияния различных параметров и определения величины неоднородности элементов конструкции, а также закона управления.

### Математическая модель

При моделировании предполагалось, что рабочей температурой устройства является температура пластины-держателя в реперной точке; все материалы элементов конструкции однородны по составу и изотропны по свойствам; контакты между элементами идеальны; боковые поверхности теплоизолированы. Пробирки с образцами в рассмотрение не принимались.

Теплофизические свойства материалов, которые считаются постоянными, приведены в табл. 1. Теплопроводность полупроводника, а также некоторые его термоэлектрические свойства (удельное сопротивление и коэффициент Зеебека) являются функциями от температуры и приведены в [3].

Схема расчетной области изображена на рис. 3. Уравнение нестационарной теплопроводности для теплового блока записывается следующим образом:

$$\frac{\partial}{\partial \tau} (\rho_i \cdot c_{pi} \cdot T_i) - \text{div} (\lambda_i \cdot \text{grad} T) = S_{Ti}, \quad i=1, \dots, 5, \quad (1)$$

где  $T = T(\Omega_i, \tau)$ , К – температура;  $c_{pi}$ , Дж/(кг·К) – теплоемкость;  $\rho_i$ , кг/м<sup>3</sup> – плотность;  $\lambda_i$ , Вт/(м·К) – коэффициент теплопроводности;  $\tau$ , с – время;  $S_{Ti}$ , Вт/м<sup>3</sup> – источниковый член, учитывающий источники и стоки теплоты.

	Наименование	Материал	Теплопроводность, Вт/(м·К)	Теплоемкость, Дж/(кг·К)	Плотность, кг/м <sup>3</sup>
1	Пластина-держатель	Алюминий	200	953	2750
2	Подложка	Поликор Al <sub>2</sub> O <sub>3</sub>	32	79.04	3970
3	Спай	Медь	360	410	8900
4	Полупроводник	Bi <sub>2</sub> Te <sub>3</sub>	$f(T)$	390	7860
5	Радиатор	Алюминий	200	953	2750
6	-	Воздух	0,025	1006	1.205

Таблица 1. Теплофизические свойства материалов

Начальное условие:

$$T(\Omega_i)|_{\tau=0} = T_0. \quad (2)$$

Граничные условия:

$$(-\lambda_1 \cdot \text{grad } T, \vec{n})|_{\Gamma_1} = \alpha_1 \cdot (T_1 - T_{f1})|_{\Gamma_1}, \quad (3)$$

$$(-\lambda_5 \cdot \text{grad } T, \vec{n})|_{\Gamma_5} = \alpha_5 \cdot (T_5 - T_{f5})|_{\Gamma_5}, \quad (4)$$

где  $T_f$ , К – температура окружающей среды;  $\alpha_i$ , Вт/(м<sup>2</sup>·К) – средний по поверхности коэффициент теплоотдачи воздуха со стороны соответствующего элемента;  $\vec{n}$  – единичный вектор внешней нормали. На остальных поверхностях устройства тепловые потоки отсутствуют. Источниковые члены в уравнении теплопроводности (1) записываются в виде [4]

$$S_{Ti} = S_{Ci} + S_{Pi} \cdot T, \text{ Вт/м}^3, \quad (6)$$

но для пластины-держателя ( $\Omega_1$ ), радиатора ( $\Omega_5$ ) и подложек ( $\Omega_2$ ) – в виде

$$S_{Ti} = 0, i=1, 2, 5, \quad (7)$$

а для полупроводника ( $\Omega_4$ ) и спаев ( $\Omega_3$ ) объемная плотность мощности тепловыделения источника является функцией величины силы тока и определяется эффектами, возникающими в термоэлементах, т.е. эффектами Пельтье, Томсона и Джоуля [5].

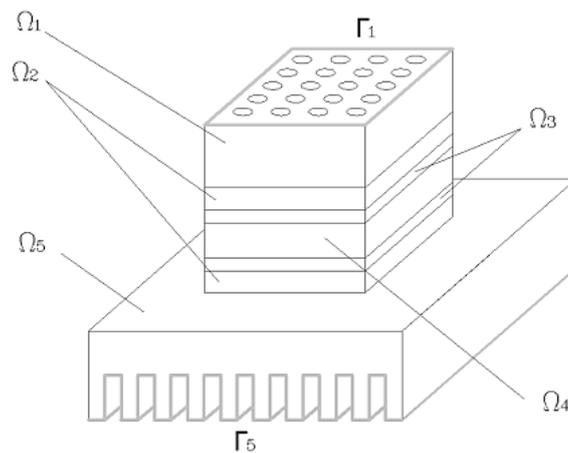


Рис. 3. Схема расчетной области:  $\Omega_1$  – пластина-держатель;  $\Omega_2$  – керамические подложки;  $\Omega_3$  – медные спаи;  $\Omega_4$  – полупроводник;  $\Omega_5$  – радиатор

### Особенности моделирования термоэлементов Пельтье

Эффект Пельтье, возникающий в спае, заключается в том, что при пропускании постоянного тока через термоэлемент в месте контакта двух металлов выделяется или

поглощается некоторое количество теплоты. Выделение или поглощение теплоты зависит от направления тока: для горячего спая

$$Q_p = e \cdot T \cdot j_0 \cdot S_0, \text{ Вт}, \quad (8)$$

для холодного спая

$$Q_p = -e \cdot T \cdot j_0 \cdot S_0, \text{ Вт}, \quad (9)$$

где  $e = e^p + |e^n|$ , В/К – коэффициент Зеебека ( $e^n$ , В/К – коэффициент Зеебека для  $n$ -ветви;  $e^p$ , В/К – коэффициент Зеебека для  $p$ -ветви);  $T$ , К – температура соответствующего спая;  $j_0$ , А/м<sup>2</sup> – величина плотности тока;  $S_0$ , м<sup>2</sup> – площадь сечения ветви. Таким образом, в объеме полупроводника высотой  $H_4$  возникает градиент температур, который при пропускании тока приводит к появлению эффекта Томсона:

$$Q_T = \tau_T (\vec{j}_0, \text{grad } T) \cdot H_4 \cdot S_0, \text{ Вт}, \quad (10)$$

где  $\tau_T = T \left( \frac{\partial e}{\partial T} \right)$ , В/К – коэффициент Томсона. Кроме того, в объеме полупроводника выделяется теплота под воздействием эффекта Джоуля:

$$Q_J = j_0^2 \cdot r \cdot H_4 \cdot S_0, \text{ Вт}, \quad (11)$$

где удельное электрическое сопротивление ветвей термоэлемента складывается из сопротивления  $n$ - ( $r^n$ , Ом/м) и  $p$ -ветвей ( $r^p$ , Ом/м):

$$r = (r^n + r^p), \text{ Ом/м}. \quad (12)$$

При моделировании процессов использовалась эффективная модель, для которой считается, что выделение и поглощение теплоты происходит по всему объему соответствующего слоя. Тогда источники пересчитываются для полученного эффективного объема. Таким образом, слагаемые, входящие в состав источников члена для спая, примут следующий вид:

$$S_{C3} = \begin{cases} 0, & \text{для холодного спая;} \\ \frac{N \cdot S_0}{\tilde{V}_3} \cdot e \cdot j_0 \cdot T, & \text{Вт/м}^3, \text{ для горячего спая,} \end{cases} \quad (13)$$

$$S_{P3} = \begin{cases} -\frac{N \cdot S_0}{\tilde{V}_3} \cdot e \cdot j_0, & \text{Вт/(К} \cdot \text{м}^3), \text{ для холодного спая;} \\ 0, & \text{для горячего спая,} \end{cases}$$

где  $\tilde{V}_3$ , м<sup>3</sup> – эффективный объем спая;  $N$  – число ветвей в нагревательном элементе. Тепловые источники и стоки, возникающие в объеме полупроводника ( $\tilde{V}_4$ , м<sup>3</sup> – эффективный объем полупроводника), имеют вид

$$S_{C3} = \begin{cases} \frac{N \cdot S_0 \cdot H_4}{\tilde{V}_4} \cdot j^2 \cdot r, & \text{Вт/м}^3, \text{ если } \left( \frac{\partial e}{\partial T} \right) \cdot \frac{\partial T}{\partial z} > 0; \\ \frac{N \cdot S_0 \cdot H_4}{\tilde{V}_4} \cdot \left[ j^2 \cdot r + \left( \frac{\partial e}{\partial T} \right) \cdot \frac{\partial T}{\partial z} \cdot j \cdot T_4 \right], & \text{Вт/м}^3, \text{ если } \left( \frac{\partial e}{\partial T} \right) \cdot \frac{\partial T}{\partial z} < 0, \end{cases} \quad (14)$$

$$S_{P3} = \begin{cases} \frac{N \cdot S_0 \cdot H_4}{\tilde{V}_4} \cdot \left( \frac{\partial e}{\partial T} \right) \cdot \frac{\partial T}{\partial z} \cdot j, & \text{Вт/(м}^3 \cdot \text{К)}, \text{ если } \left( \frac{\partial e}{\partial T} \right) \cdot \frac{\partial T}{\partial z} > 0; \\ 0, & \text{если } \left( \frac{\partial e}{\partial T} \right) \cdot \frac{\partial T}{\partial z} < 0. \end{cases}$$

## Решение задачи в одномерной постановке

Для сокращения затрат времени работоспособность системы охлаждения исследовалась в одномерной постановке (рис. 4). Для этого вводились эффективные коэффициенты, характеризующие такие параметры радиатора, как его размеры и поверхность теплообмена. Этими коэффициентами являются эффективный коэффициент теплоотдачи<sup>1</sup>, показывающий изменение площади радиатора при переходе к одномерной модели,

$$\alpha^* = \alpha(\text{Re}) \cdot \frac{F_p}{F_{1d}}, \text{ Вт/ (м}^2 \cdot \text{К)}, \quad (14)$$

и коэффициент преобразования объема

$$K_5 = \frac{V_5}{\tilde{V}_5}, \quad (15)$$

где  $\tilde{V}_5$ , кг – приведенный объем радиатора;  $V_5$ , кг – реальный объем радиатора. Тогда плотность этого элемента конструкции можно представить в виде эффективной плотности  $\tilde{\rho}_5$ , кг/м<sup>3</sup>:

$$\tilde{\rho}_5 = K_5 \cdot \rho_5, \quad (16)$$

где  $\rho_5$ , кг/м<sup>3</sup> – плотность материала радиатора.

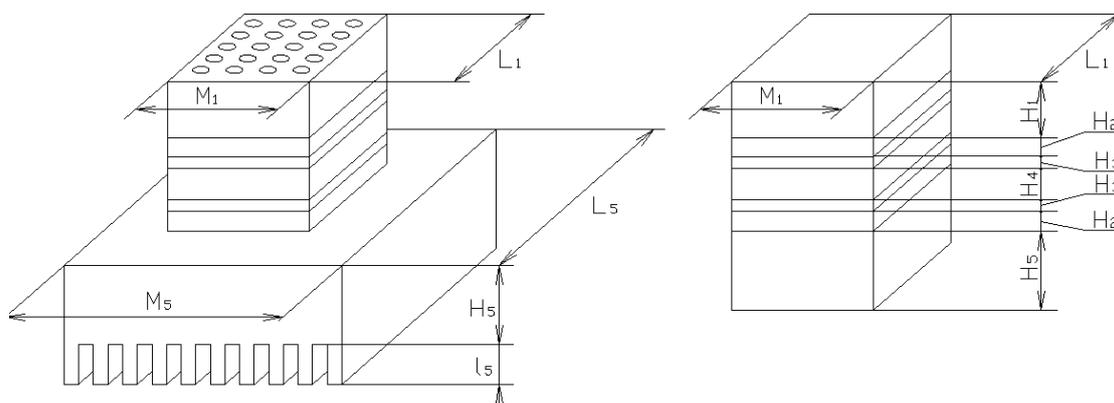


Рис. 4. Построение одномерной модели

Кроме того, для учета изменения геометрии пластины-держателя был введен коэффициент преобразования его объема, и плотность элемента рассчитывалась по следующей формуле:

$$\tilde{\rho}_1 = K_1 \cdot \rho_1, \quad (17)$$

где  $\rho_1$ , кг/м<sup>3</sup> – плотность пластины-держателя;  $K_1 = \frac{V_1}{\tilde{V}_1}$  – коэффициент преобразования объема,  $\tilde{V}_1$ , кг – приведенный объем пластины-держателя;  $V_1$ , кг – реальный объем. Для определения закона изменения вектора плотности тока в термоэлементах, или закона управления, необходимо решить обратную задачу, где неизвестным является источник член  $S_{Ti}(\vec{j}_0)$  ( $i=3, 4$ ).

Обратная задача теплопроводности в рассматриваемом случае выражается следующим функционалом [7]:

<sup>1</sup> Использование эффективного коэффициента теплоотдачи возможно при малых числах Био; в данном

случае [6]  $Bi = \frac{\alpha \cdot l_5}{\lambda_5} = \frac{10,64 \cdot 0,031}{200} = 0,00165 \ll 1$ .

$$\varphi(\tau, \vec{j}_0) = \int_0^t |T(\tau, \vec{j}_0) - T_m(\tau)| d\tau, \quad (17)$$

где  $T_m(\tau)$ , К – температура в реперной точке, равная соответственно температуре верхней или нижней выдержки;  $T(\tau, \vec{j}_0)$ , К – температура, получаемая при решении уравнения теплопроводности. На каждом  $k$ -м шаге по времени имеем:

$$F(\tau_k, \vec{j}_0) = |T(\tau_k, \vec{j}_0) - T_m(\tau_k)|. \quad (18)$$

Таким образом, решение обратной задачи сводится к отысканию минимального значения функции  $F(\tau_k, \vec{j}_0)$ , которым в данном случае является ноль, тогда можно перейти к уравнению вида

$$f(\tau_k, \vec{j}_0) = T(\tau_k, \vec{j}_0) - T_m(\tau_k) = 0. \quad (19)$$

Для решения уравнения (19) использовался метод половинного деления [8]. Для этого необходимо выбрать диапазон изменения плотности тока  $[j_{\min}, j_{\max}]$ , на котором функция непрерывна, причем

$$F(j_{\min}) < 0, \quad (20)$$

$$F(j_{\max}) > 0. \quad (21)$$

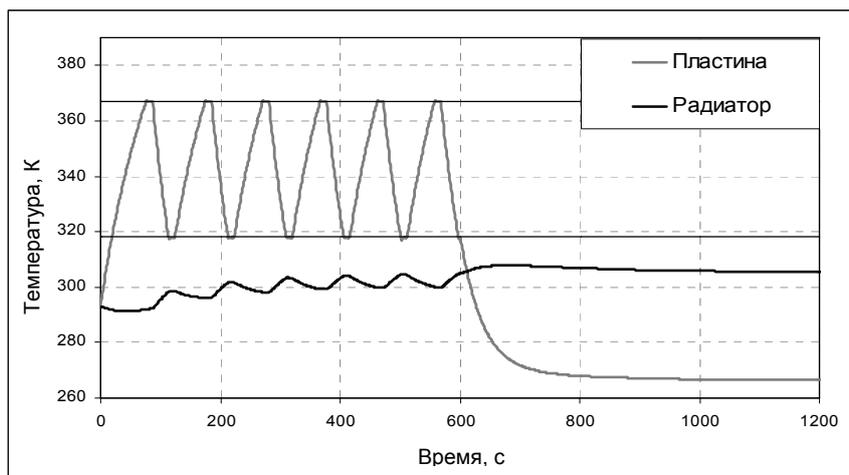
Если условие (20) не выполняется, то плотность тока равна  $j_0 = j_{\min}$ , и уравнение (19) не решается. Аналогично для условия (21), тогда плотность тока равна  $j_0 = j_{\max}$ . В таком случае температура становится больше или меньше температуры выдержки на величину, большую, чем погрешность вычисления  $\varepsilon$ . Это отклонение определяет точность поддержания температур. Отрезок  $[j_{\min}, j_{\max}]$  был выбран из возможностей термоэлементов:  $j_{\max}$  – максимальная плотность тока, которую можно подать на термоэлементы,  $j_{\min}$  – минимальная плотность тока, с которой способна работать установка без прекращения подачи тока.

Величина	Обозначение	Значение	Размерность
Граничные и начальные условия			
Коэффициент теплоотдачи со стороны пластины-держателя	$\alpha_1$	5	Вт/(м <sup>2</sup> ·К)
Температура окружающей среды со стороны пластины-держателя	$T_{f1}$	308	К
Температура окружающей среды со стороны радиатора	$T_{f5}$	308	К
Начальное условие	$T_0$	293	К
Параметры циклирования			
Верхняя температура выдержки	$T_1$	367	К
Нижняя температура выдержки	$T_2$	318	К
Время выдержки	$\tau' = \tau''$	10	с
Геометрические параметры			
Коэффициент преобразования объема пластины-держателя	$K_1$	0.6215	-
Эффективный объем спая	$\tilde{V}_3$	$3,18 \cdot 10^{-6}$	м <sup>3</sup>
Эффективный объем термоэлемента	$\tilde{V}_4$	$15,94 \cdot 10^{-6}$	м <sup>3</sup>

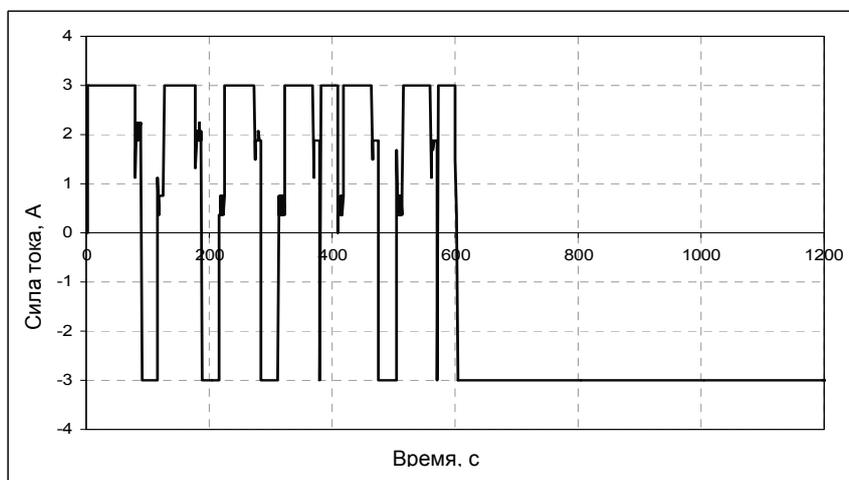
Таблица 2. Исходные данные для моделирования

Для проведения численного исследования использовалась программа, написанная на языке FORTRAN для решения одномерных нестационарных уравнений теплопроводности [9] методом контрольного объема [4]. Цель моделирования – показать рабо-

тоспособность разработанного метода. Исходные данные для проведения моделирования представлены в табл. 2.



а



б

Рис. 5. Результаты расчета термостатирующего устройства: а) изменение температуры пластины-держателя и радиатора по времени; б) изменение силы тока во времени

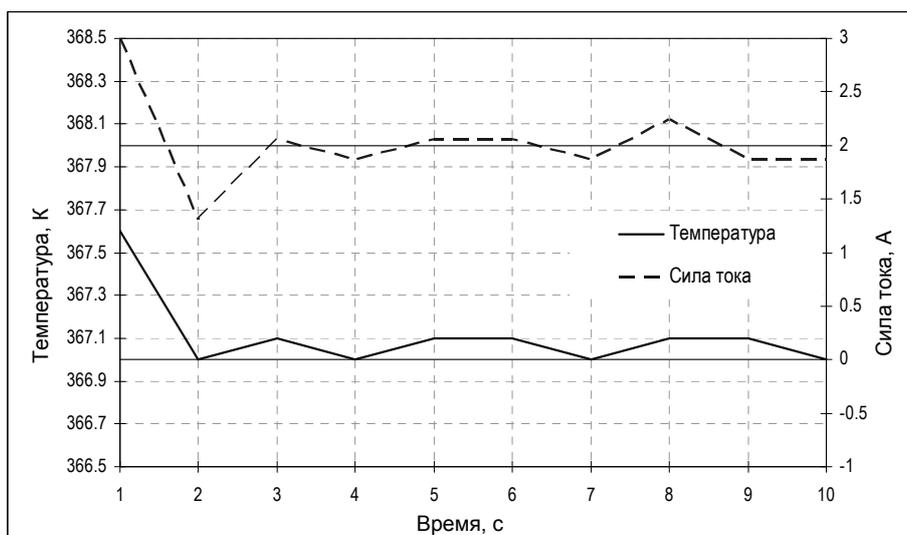


Рис. 6. Изменение силы тока и температуры во время выдержки

Задача решалась с шагом по времени 1 с. Время расчета – 1200 с, время циклирования – 600 с (после чего происходит захлаживание, т.е. ток направляется таким образом, что верхний спай охлаждается, охлаждая пластину-держатель). Изменение температуры во времени приведено на рис. 5, а. Из рисунка видно, что за 600 с проходит 4 цикла. Скорость нагрева равна  $2\text{ }^{\circ}\text{C}/\text{с}$ , скорость охлаждения –  $1,5\text{ }^{\circ}\text{C}/\text{с}$ . Общий закон управления термоэлектрическими элементами представлен на рис. 5, б (для удобства изображен закон изменения силы тока, а не плотности).

На рис. 6 более подробно показаны закон управления для наиболее нагруженного цикла во время режима выдержки, а также график изменения температуры, соответствующий этому режиму. Точность выдержки при этом составила 0,1 К.

### Проведение трехмерного моделирования

Благодаря большому количеству команд встроенного препроцессора в ПК STAR-CD [10] возможно автоматизированное создание компьютерной модели, для чего основные параметры модели записываются в отдельный файл (INPUT\_DATA.inp). В дальнейшем это позволяет без значительных усилий со стороны пользователя формировать подобные расчетные модели, а также проводить необходимую настройку сетки расчетной области и задавать значения изменяемых параметров модели (свойства материалов). Кроме того, в среде ПК STAR-CD пишутся скрипты, составляющие программное обеспечение. Скрипты дополняются подпрограммами, написанными на языке FORTRAN. С помощью этих подпрограмм задаются источниковые члены в уравнении энергии и свойства полупроводника в зависимости от температуры.

Созданию сетки предшествовало построение твердотельной модели с помощью пакета PRO-E wildfire 2.0. Модель термостатирующего устройства представлена на рис. 7.

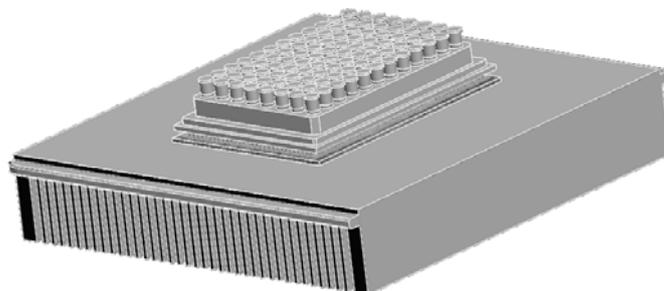


Рис. 7. Твердотельная модель термостатирующего устройства

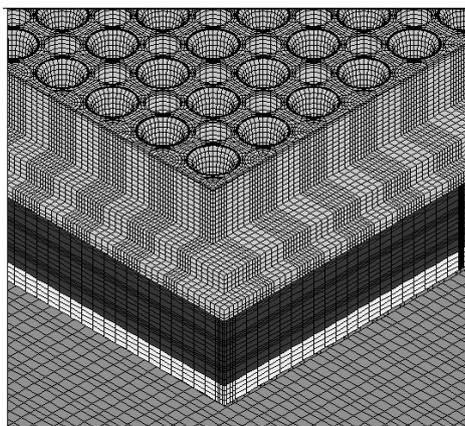


Рис. 8. Фрагмент сетки термостатирующего устройства

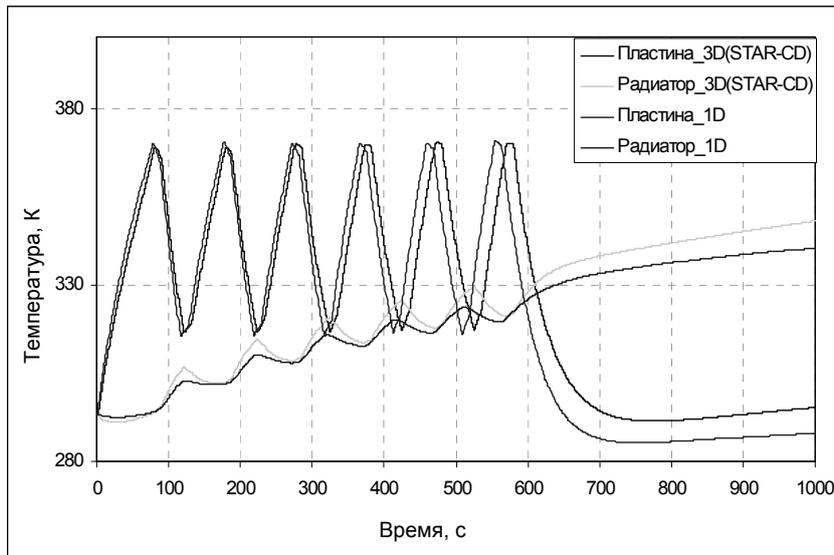


Рис. 9. Изменение температуры пластины-держателя и радиатора по времени

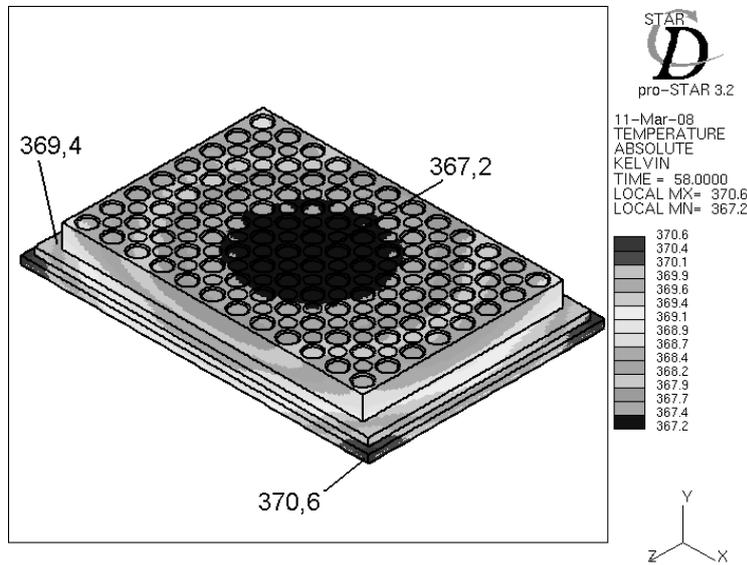


Рис. 10. Температурное поле пластины-держателя в режиме выдержки при 367 К в момент времени 80 с

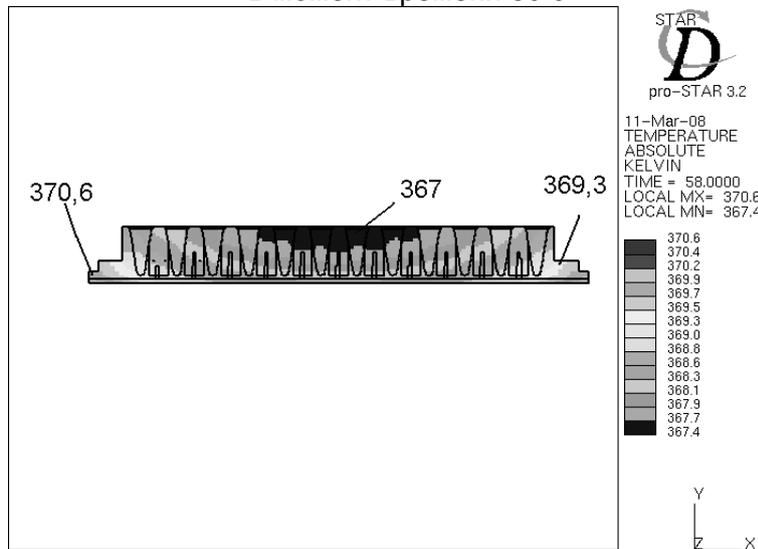


Рис. 11. Разрез пластины-держателя

На рис. 8 представлен фрагмент сетки расчетной области. Сначала строились сетки отдельно для каждого элемента, а затем при помощи специальных интерфейсов они объединялись в общую неструктурированную сетку размером около 1 000 000 контрольных объемов.

Для термостатирующего устройства проведено трехмерное моделирование теплообмена при указанных выше параметрах решения (см. табл. 2). Коэффициент теплоотдачи со стороны радиатора принимался равным  $\alpha_s = 10,64 \text{ Вт}/(\text{м}^2 \cdot \text{К})$  [6]. Закон изменения силы тока в термоэлементах задавался предварительно.

На рис. 9 представлены результаты моделирования в виде циклов изменения температуры пластины-держателя и радиатора по времени (использовались значения температуры в выбранных точках пластины-держателя). Как видно из графиков, результаты одно- и трехмерного расчетов находятся в хорошем соответствии, что подтверждает правомерность использования одномерной модели для проведения серии вариационных вычислений. Расхождение по времени не превышает 3 % (20 с), по температуре – 2 % (8 К). Эти расхождения связаны с влиянием особенностей конструкции держателя и радиатора.

На рис. 10 приведено поле температур пластины-держателя во время режима выдержки при температуре 367 К. Как показывает рисунок, неравномерность температурного поля составляет около 1,5 К. Температура центра выше температуры выдержки, так как реперная точка располагается в центре пластины-держателя.

## Выводы

Разработан метод расчета твердотельных термостатирующих устройств в динамическом режиме работы. Метод включает в себя, во-первых, программное обеспечение на языке FORTRAN для проведения одномерных оптимизационных расчетов, и, во-вторых, модель для исследования трехмерного температурного поля, созданную в ПК STAR-CD. В рамках одномерного расчета решена обратная задача для определения закона изменения силы тока в термоэлементах во время режимов выдержки.

Проведено численное моделирование процессов теплообмена в тепловом блоке амплификатора ДНК в одномерной и трехмерной постановках. Результаты расчета находятся в хорошем соответствии – разница не превысила 3 %. Получено трехмерное температурное поле пластины-держателя. Неоднородность температуры по пластине-держателю не превышает 1,5 К, что соответствует реальным данным для подобного рода устройств.

## Литература

1. Чернышев А.В. Создание теории рабочих процессов, методов расчета и разработка оборудования для ПЦР-диагностики: Автореф. дис. ... докт. техн. наук. – М., 2000. – 32 с.
2. Белова О.В. Разработка метода расчета и исследование прецизионных устройств нагрева и охлаждения: Автореф. дис. ... докт. техн. наук. – М., 2000. – 15 с.
3. Чернышев А.В., Белова О.В. Разработка, расчет и проектирование пневмоэлектро-механического и электровакуумного лабораторного оборудования. – М.: Изд-во МГТУ им. Н.Э. Баумана, 1998. – 40 с.
4. Патанкар С. Численные методы решения задач теплообмена и динамики жидкости. – М.: Энергоатомиздат, 1984. – 152 с.
5. Сакун И.А. Холодильные машины. – Л.: Машиностроение, 1985. – 510 с.
6. Себеси Т., Брэдшоу П. Конвективный теплообмен. Физические основы и вычислительные методы / Пер. с англ. – М.: Мир, 1987. – 592 с.
7. Алифанов О.М. Обратные задачи теплообмена. – М.: Машиностроение, 1988. – 280 с.
8. Калиткин Н.Н. Численные методы. – М.: Наука, 1978. – 512 с.

9. Куракин А.А., Скибин А.П., Югов В.П. Программный комплекс для расчета теплообмена в пограничных слоях. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2000. – 32 с.
10. Methodology STAR-CD version 3.24 CD adapco group 2004.

*Белова Ольга Владимировна*

— Московский государственный технический университет им. Н.Э. Баумана, кандидат технических наук, доцент, [ovbelova@yandex.ru](mailto:ovbelova@yandex.ru)

*Корнеева Мария Александровна*

— Московский государственный технический университет им. Н.Э. Баумана, студентка, [makorneeva@mail.ru](mailto:makorneeva@mail.ru)

*Мустафина Дарья Александровна*

— Московский государственный технический университет им. Н.Э. Баумана, студентка, [dmustafina@yandex.ru](mailto:dmustafina@yandex.ru)

**УДК 629.3.052.6**

## **СИСТЕМА ЛОКАЛЬНОЙ НАВИГАЦИИ ДЛЯ НАЗЕМНЫХ МОБИЛЬНЫХ РОБОТОВ**

**В.А. Черноножкин, С.А. Половко**

Предлагается проект системы локальной навигации, в дальнейшем – составляющей универсального навигационного комплекса для наземных мобильных роботов. Система учитывает особенности движения мобильных роботов, что позволит решать навигационные задачи в этой области наиболее эффективно.

**Ключевые слова:** инерциальная навигационная система, локальная навигация, автономные мобильные роботы, обработка информации.

### **Введение**

Мобильные робототехнические системы применяются сегодня в самых разных отраслях. Чем обширнее область их применения в промышленности, в военных и спасательных приложениях, в медицине и в быту, тем жестче становятся требования к их исполнению для конкретных задач. Одни из самых актуальных таких требований относятся к автономности робота и его навигационным возможностям [1].

Навигация в робототехнике имеет ряд существенных особенностей, не позволяющих использовать навигационное оборудование для иных объектов (автомобилей, кораблей, самолетов и пр.) с достаточной эффективностью. Поэтому каждый раз, обеспечивая навигационными возможностями мобильный робот, приходится применять несколько различных навигационных средств, заново решая задачи комплексирования и обработки информации, учета особенностей кинематики и динамики объекта и т.д. Таким образом, необходимо создание сложного универсального навигационного комплекса, готового к применению на мобильных роботах любых размеров, комплектации и назначения.

В данной работе предлагается проект системы локальной навигации (ЛН) для наземных мобильных роботов – первый шаг к созданию такого комплекса.

### **1. Локальная навигация мобильных роботов**

Говоря о навигации автономных мобильных роботов, выделяют два основных типа систем – глобальной навигации и ЛН. Основная задача систем глобальной навигации – формирование плана перемещения на основании заданной карты. Системы ЛН планируют и контролируют выполнение маневров, составляющих движение по сформированному маршруту.

ЛН – схема навигации автономных мобильных устройств, отвечающая за определение относительных координат объекта преимущественно в течение короткого промежутка времени. Основные задачи ЛН:

- (1) определение относительных декартовых координат и путевого угла мобильного робота, а также его скорости;
- (2) планирование и контроль выполнения роботом маневров, составляющих движение по сформированному маршруту;
- (3) корректирующая поддержка схем глобальной навигации для определения абсолютных координат аппарата.

Типичными примерами использования систем ЛН являются контроль следования по безопасной траектории в ограниченных пространствах и навигация робота в условиях недоступности данных с внешних источников навигации (например, при отказе системы позиционной коррекции на базе GPS). В связи с этим выдвигаются следующие требования при разработке таких систем:

- (1) возможность определения углов ориентации мобильного робота (угла курса, крена и тангажа), а также скорости его движения для формирования адекватной траектории;
- (2) непрерывная выдача навигационной информации;
- (3) высокая точность вычисления координат робота в течение коротких промежутков времени;
- (4) автономность работы системы;
- (5) малые габариты и энергопотребление;
- (6) устойчивость к помехам и вибрациям.

Известно, что ЛН объекта может быть реализована как с помощью активных (инерциальная навигация, системы технического зрения), так и с помощью пассивных схем навигации (спутниковая, по радиомаякам и маркерам) [2]. Однако в свете установленных требований очевидными преимуществами перед остальными вариантами обладает именно инерциальная навигация – метод определения координат, скорости и угловой ориентации объекта на основе измерения и интегрирования его ускорения.

Сегодня инерциальные навигационные системы (ИНС) недороги и устроены довольно просто. В них для решения задач навигации используется некоторое количество инерциальных чувствительных элементов (ЧЭ) – гироскопов и акселерометров, дающих информацию о параметрах движения объекта. ИНС автономны, на их работе не сказываются погодные условия, они не поддаются радиоэлектронному подавлению и обеспечивают скрытность (не генерируют электромагнитного излучения, выдающего присутствие объекта-носителя).

Недостатками ИНС являются необходимость начальной настройки (выставки) по скорости и пространственному положению, а также накопление ошибки определения навигационных параметров объекта со временем. Однако в случае применения ИНС для решения задач ЛН мобильной робототехнической платформы у нее достаточно времени на инициализацию и начальную выставку, поскольку многие системы мобильного робота требуют проведения аналогичных процедур. Что же касается накопления ошибки, то сегодня активно используются различные методы коррекции навигационных данных ИНС с помощью вспомогательных средств (одометров, спутниковых навигационных систем и т.д.). В таком случае система уже перестает быть автономной – такое объединение нескольких систем для решения навигационных задач называется комплексированием. Для оптимального использования данных, поступающих от всех этих систем, нужно, чтобы тщательно учитывались характеристики и погрешности всех навигационных средств. Оптимальное комплексирование данных от разных источников обеспечивает обобщенный фильтр Калмана (ОФК), который также применяется почти во всех ИНС для отслеживания и уменьшения накопленной ошибки.

## 2. Система локальной навигации

### 2.1. Структурная схема

На рис. 1 представлена структура системы ЛН, разработанной с учетом указанных требований. Рассмотрим подробно схему работы системы ЛН.

Чтобы получать информацию обо всех шести степенях свободы мобильного робота, используются шесть ЧЭ – три гироскопа ( $\Gamma_x, \Gamma_y, \Gamma_z$ ) и три акселерометра ( $A_x, A_y, A_z$ ), которые дают исчерпывающую информацию об его угловых и линейных движениях. По причине возможного автономного применения системы только в течение коротких промежутков времени требования к точностным характеристикам гироскопов и акселерометров отодвигаются на задний план требованиями к аппаратуре автономных объектов – к массогабаритным и ценовым характеристикам, надежности и устойчивости к ударам и вибрациям и т.п. В связи с этим для реализации системы рекомендуются микромеханические чувствительные элементы [3–5].

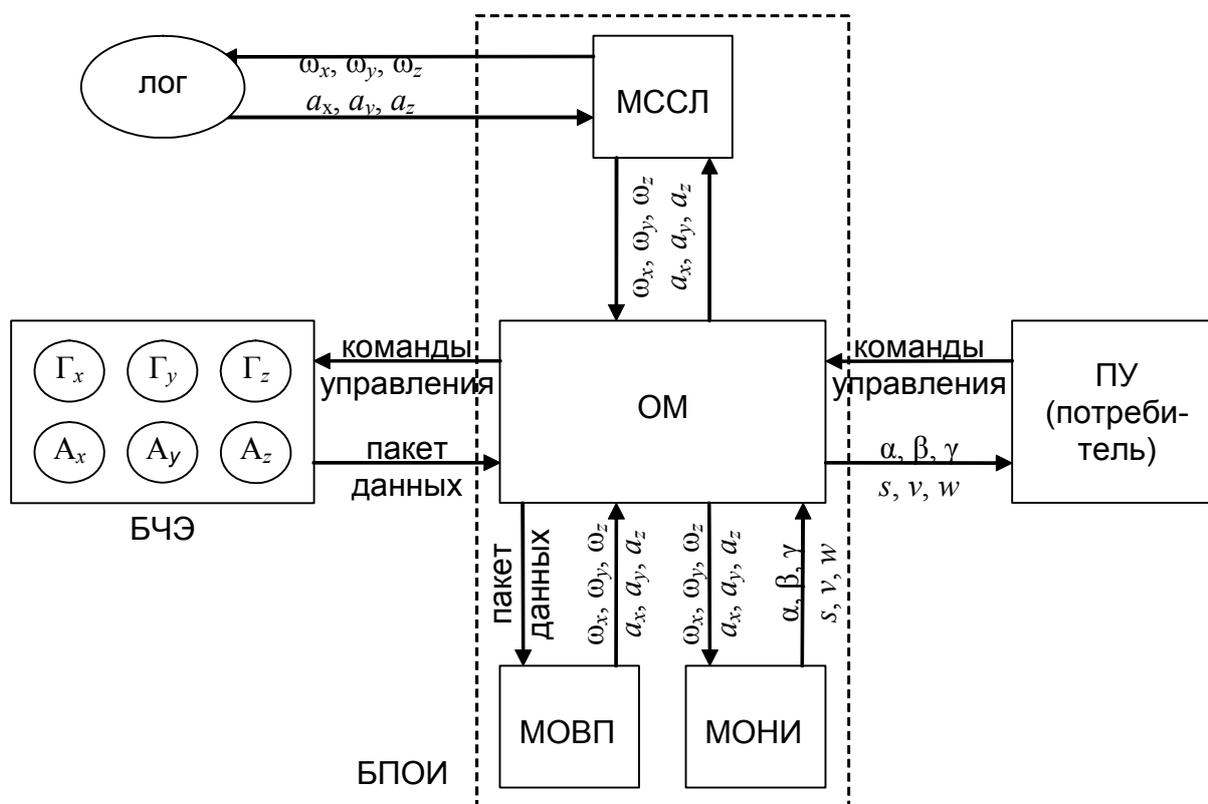


Рис. 1. Структурная схема системы локальной навигации

Блок чувствительных элементов (БЧЭ) выдает пакет данных, содержащий информацию об угловых скоростях и кажущихся ускорениях системы по каждой из осей связанной системы координат (ССК), на бортовой компьютер. Пакет через основной модуль (ОМ) блока программной обработки информации (БПОИ), осуществляющий взаимодействие остальных модулей, а также пульта управления (ПУ) и аппаратуры, поступает в модуль обработки входящих пакетов (МОВП).

МОВП организует проверку целостности пакета данных, разбиение его на составляющие и выдачу обратно в ОМ полезной информации об угловых скоростях и кажущихся ускорениях системы ( $\omega_x, \omega_y, \omega_z, a_x, a_y, a_z$ ). Оттуда она перенаправляется в модуль определения навигационной информации (МОНИ), где проходит процесс фильтрации и вычисления навигационных параметров объекта-носителя – курсового угла, тангажа, крена, пройденного пути, скорости и ускорения ( $\alpha, \beta, \gamma, s, v, w$ ).

Полученная навигационная информация в зависимости от настроек БПОИ, а также команд управления с ПУ может быть выдана в различной форме (графики данных, траекторий, числовые значения и т. д.) на ПУ. Исходная же информация с БЧЭ в обработанном МОВП виде может быть сохранена в лог-файл через модуль сохранения/считывания логов (МССЛ) для последующего воспроизведения и анализа эксперимента.

## 2.2. Математическая модель

Как уже упоминалось, при движении колесных мобильных роботов существует ряд особенностей, влияющих на определение навигационных параметров:

- (1) отсутствие бокового сноса;
- (2) малые диапазоны изменения углов крена и тангажа робота за короткие промежутки времени;
- (3) независимость угловой скорости вокруг вертикальной оси робота от угловых скоростей вокруг других осей;
- (4) малая скорость движения робота.

С учетом всех этих особенностей составлена математическая модель мобильного робота, использующего разрабатываемую систему ЛН автономно. С помощью схемы (рис. 2), определяющей положение связанной системы координат (ССК) относительно подвижной траекторной системы координат (ПТСК), получены уравнения, определяющие соотношение навигационных параметров робота с измеряемыми ЧЭ системы величинами:

$$(1) \text{ для определения угловой ориентации –} \begin{cases} \dot{\alpha} = \omega_z, \\ \dot{\beta} = \omega_x \cdot \cos \gamma + \omega_z \cdot \sin \gamma, \\ \dot{\gamma} = \omega_y, \end{cases} \quad (1)$$

где  $\alpha$ ,  $\beta$ ,  $\gamma$  – курсовой угол, тангаж и угол крена,  $\omega_x$ ,  $\omega_y$ ,  $\omega_z$  – измеряемые в ССК угловые скорости мобильного робота;

$$(2) \text{ для определения составляющих ускорения –} \begin{cases} w_x = a_x \cdot \cos \gamma + a_z \cdot \sin \gamma, \\ w_y = a_x \cdot \sin \beta \sin \gamma + a_y \cdot \cos \beta - a_z \cdot \sin \beta \cos \gamma, \\ w_z + g = -a_x \cdot \cos \beta \sin \gamma + a_y \cdot \sin \beta + a_z \cdot \cos \beta \cos \gamma, \end{cases} \quad (2)$$

где  $w_x$ ,  $w_y$ ,  $w_z$  – проекции истинного ускорения на оси ПТСК,  $a_x$ ,  $a_y$ ,  $a_z$  – измеряемые в ССК составляющие кажущегося ускорения мобильного робота,  $g$  – ускорение свободного падения;

$$(3) \text{ для определения скорости и перемещения –} \begin{cases} v = v_0 + \int w_y dt, \\ s = s_0 + \int v dt, \end{cases} \quad (3)$$

где  $v$ ,  $s$  – скорость и перемещение мобильного робота. Чтобы учесть такую особенность применения инерциальных навигационных систем, как накопление ошибок интегрирования [6], математическая модель системы дополнена уравнениями, связывающими измеряемые значения угловых скоростей и составляющих кажущегося ускорения объекта с их истинными значениями:

$$\begin{cases} \omega_j^{\text{ИЗМ}} = \omega_j + \varepsilon_j \\ a_j^{\text{ИЗМ}} = a_j + \psi_j \end{cases}, \quad (4)$$

где  $\omega_j^{\text{ИЗМ}}$ ,  $a_j^{\text{ИЗМ}}$  – значения угловых скоростей и составляющих кажущегося ускорения мобильного робота, определяемые гироскопами и акселерометрами,  $\omega_j$ ,  $a_j$  – их истинные значения,  $\varepsilon_j$ ,  $\psi_j$  – ошибки измерения,  $j = \{x, y, z\}$ .

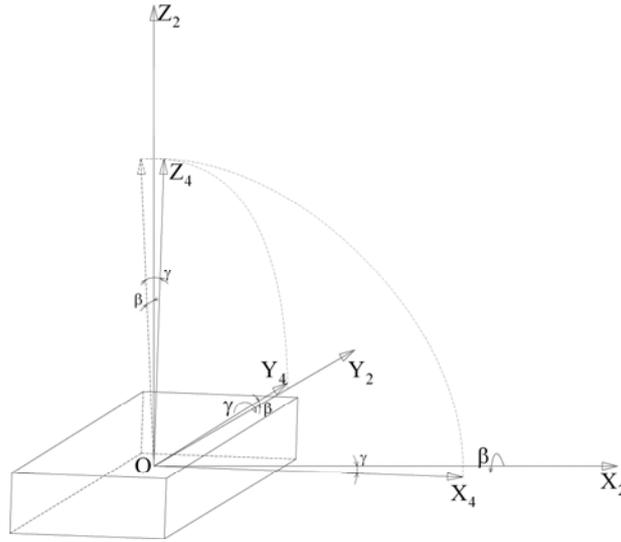


Рис. 2. Схема вращения связанной системы координат  $X_4Y_4Z_4$  относительно подвижной траекторной системы координат  $X_2Y_2Z_2$

### 2.3. Алгоритм комплексной обработки информации

Поскольку все измеряемые и наблюдаемые величины связаны между собой нелинейно, для определения вектора состояния мобильного робота по вектору измерений навигационных датчиков в каждый момент времени используется алгоритм нелинейной фильтрации – обобщенный фильтр Калмана (ОФК) [7]. В этом случае задача оценивания вектора состояния  $x_i$  сводится к фильтрации марковской последовательности  $x_i = \Phi_i(x_{i-1}) + w_i$  по проводимым в каждый  $i$ -й момент времени измерениям  $y_i = s_i(x_i) + v_i$  при гауссовском характере порождающих шумов  $w_i$  и ошибок измерения  $v_i$ . Здесь функция  $\Phi_i(x_{i-1})$  характеризует динамику изменения вектора состояния, а функция  $s_i(x_i)$  – связь вектора измерений  $y_i$  с ним.

ОФК основан на гауссовской аппроксимации апостериорной плотности, для нахождения параметров которой применяется разложение в ряд Тейлора функций  $\Phi_i(x_{i-1})$  и  $s_i(x_i)$ :

$$\Phi_i(x_{i-1}) \approx \Phi_i(x_{n1}) + \frac{\partial \Phi_i(x_{n1})}{\partial x_{i-1}^T} (x_{i-1} - x_{n1}), \quad (5)$$

$$s_i(x_i) \approx s_i(x_{n2}) + \frac{\partial s_i(x_{n2})}{\partial x_i^T} (x_i - x_{n2}), \quad (6)$$

где  $x_{n1}$ ,  $x_{n2}$  – точки линеаризации. В качестве точек линеаризации в (5) и (6) используются оценка вектора состояния предыдущего шага, а также апостериорная оценка текущего шага:  $x_{n1} = \hat{x}_{i-1}$ ,  $x_{n2} = \hat{x}_{i/i-1}$ .

Таким образом, математический аппарат одной итерации алгоритма ОФК заключается в последовательном определении следующих величин:

(1) апостериорной оценки вектора состояния

$$\hat{x}_{i/i-1} = \Phi_i(x_{n1}) + \frac{\partial \Phi_i(x_{n1})}{\partial x_{i-1}^T} (\hat{x}_{i-1} - x_{n1}); \quad (7)$$

(2) матрицы ковариаций апостериорной оценки вектора состояния

$$P_{i/i-1}(x_{n1}) = \frac{\partial \Phi_i(x_{n1})}{\partial x_{i-1}^T} P_{i-1} \left( \frac{\partial \Phi_i(x_{n1})}{\partial x_{i-1}^T} \right)^T + Q_i, \quad (8)$$

где  $Q_i$  – матрица, характеризующая интенсивность порождающих шумов;

(3) матрицы ковариаций текущей оценки вектора состояния

$$P_i(x_{n1}, x_{n2}) = \left( (P_{i/i-1}(x_{n1}))^{-1} + \left( \frac{\partial s_i(x_{n2})}{\partial x_i^T} \right)^T R_i \frac{\partial s_i(x_{n2})}{\partial x_i^T} \right)^{-1}; \quad (9)$$

(4) коэффициента усиления фильтра Калмана

$$K_i(x_{n1}, x_{n2}) = P_i(x_{n1}, x_{n2}) \left( \frac{\partial s_i(x_{n2})}{\partial x_i^T} \right)^T R_i, \quad (10)$$

где  $R_i$  – матрица, характеризующая интенсивность шумов измерений;

(5) текущей оценки вектора состояния

$$\mathbf{x}_i = \mathbf{x}_{i/i-1} + K_i(x_{n1}, x_{n2}) \left( y_i - s_i(x_{n2}) - \frac{\partial s_i(x_{n2})}{\partial x_i^T} (\mathbf{x}_{i/i-1} - x_{n2}) \right). \quad (11)$$

На основе математической модели системы ЛН определен вид используемых в (7)–(11) матриц и векторов. Если форма вектора измерений очевидна –

$$y_i = (\omega_{xi}, \omega_{yi}, \omega_{zi}, a_{xi}, a_{yi}, a_{zi})^T,$$

то 21-мерный вектор состояния

$$x_i = (\alpha_i, \beta_i, \gamma_i, \dot{\alpha}_i, \dot{\beta}_i, \dot{\gamma}_i, \ddot{\alpha}_i, \ddot{\beta}_i, \ddot{\gamma}_i, \varepsilon_{xi}, \varepsilon_{yi}, \varepsilon_{zi}, w_{xi}, w_{yi}, w_{zi}, \dot{w}_{xi}, \dot{w}_{yi}, \dot{w}_{zi}, \psi_{xi}, \psi_{yi}, \psi_{zi})^T$$

должен включать в себя, помимо величин, участвующих в уравнениях (1)–(4), скорости изменения фильтруемых величин  $\dot{\alpha}$ ,  $\dot{\beta}$ ,  $\dot{\gamma}$  и  $\dot{w}_x$ ,  $\dot{w}_y$ ,  $\dot{w}_z$ , так как они необходимы при определении матрицы динамики  $\Phi_i(x_{i-1})$ . Поскольку информацию об этих параметрах брать неоткуда, они моделируются с помощью порождающих шумов в виде марковских стационарных процессов [8].

Опуская громоздкие выкладки, приведем окончательный вид матриц:

(1) динамики системы

$$\Phi_i(\mathbf{x}_{i-1}) = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & E_\alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & E_\beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & E_\gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & E_{w_x} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & E_{w_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & E_{w_z} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \mathbf{x}_{i-1}, \quad (12)$$

где  $E_j = \exp\left(-\frac{\Delta t}{T_j}\right)$ ,  $T_j$  – постоянные времени моделирования скоростей изменения фильтруемых величин, характеризующие возможную частоту их изменения,  $\Delta t = t_i - t_{i-1}$  – время дискретизации;

(2) связи вектора измерений с вектором состояния системы

$$s_i(\mathcal{X}_{i/i-1}) = \begin{bmatrix} \frac{\dot{\beta}_{i/i-1} - \dot{\alpha}_{i/i-1} \cdot \sin \gamma_{i/i-1}}{\cos \gamma_{i/i-1}} + \varepsilon_{xi/i-1} \\ \dot{\gamma}_{i/i-1} + \varepsilon_{yi/i-1} \\ \dot{\alpha}_{i/i-1} + \varepsilon_{zi/i-1} \\ w_{xi/i-1} \cdot \cos \gamma_{i/i-1} + (w_{yi/i-1} \cdot \sin \beta_{i/i-1} - (w_{zi/i-1} + g) \cdot \cos \beta_{i/i-1}) \cdot \sin \gamma_{i/i-1} + \psi_{xi/i-1} \\ w_{yi/i-1} \cdot \cos \beta_{i/i-1} + (w_{zi/i-1} + g) \cdot \sin \beta_{i/i-1} + \psi_{yi/i-1} \\ w_{xi/i-1} \cdot \sin \gamma_{i/i-1} - (w_{yi/i-1} \cdot \sin \beta_{i/i-1} - (w_{zi/i-1} + g) \cdot \cos \beta_{i/i-1}) \cdot \cos \gamma_{i/i-1} + \psi_{zi/i-1} \end{bmatrix}; \quad (13)$$

(3) ковариаций порождающих шумов  $w_i$

$$Q_i = \text{diag}[0, 0, (1 - E_\alpha^2)Q_\alpha, 0, 0, (1 - E_\beta^2)Q_\beta, 0, 0, 0, (1 - E_\gamma^2)Q_\gamma, 0, 0, 0, 0, 0, 0, (1 - E_{w_x}^2)Q_{w_x}, 0, (1 - E_{w_y}^2)Q_{w_y}, 0, (1 - E_{w_z}^2)Q_{w_z}, 0, 0, 0, 0], \quad (14)$$

где  $Q_j$  – интенсивности белых шумов, используемых при моделировании скоростей изменения фильтруемых величин;

(4) ковариаций шумов измерения  $v_i$

$$R_i = \text{diag}\left[N_\alpha, N_\beta, N_\gamma, N_{w_x}, N_{w_y}, N_{w_z}\right], \quad (15)$$

где  $N_j$  – параметры, характеризующие интенсивность шумов измерения.

Таким образом, можно составить перечень параметров, которые необходимо определить или задать, чтобы стало возможным использование полученных матриц (12)–(15) в алгоритме фильтрации ОФК:

- (1) начальные значения вектора состояния  $x_0$  и матрицы ковариаций  $P_0$ ;
- (2) значение времени дискретизации  $\Delta t$ ;
- (3) значения постоянных времени моделируемых экспоненциально-коррелированных процессов  $T_j$ ;
- (4) величины, характеризующие интенсивность порождающих шумов  $Q_j$  и шумов измерений  $N_j$ .

Оптимальные значения этих параметров будут подобраны при испытаниях физической модели.

### 3. Физическая модель системы локальной навигации

#### 3.1. Описание модели

Для экспериментальной настройки параметров системы ЛН, а также проверки составленного алгоритма на точность вычисления навигационных параметров мобильного робота реализована физическая модель системы. В качестве БЧЭ физической модели системы ЛН используется БЧЭ малогабаритной интегрированной навигационной системы (МИНС) «Компанав-2» фирмы «ТеКнол» (г. Москва). Эта система предназначена для определения координат местоположения и углов ориентации объекта и представляет собой комбинацию блока микроэлектромеханических датчиков движения (три гироскопа и три акселерометра) и приемника спутниковой навигационной системы со встроенной антенной. Таким образом, ее использование без последнего аналогично автономному применению разработанной системы ЛН.

Для выполнения функций блока программной обработки информации (БПОИ, рис. 1) разработано программное обеспечение, имеющее удобный для наблюдения экспериментов интерфейс и обладающее следующими функциями в соответствии со структурной схемой системы ЛН:

- (1) получение и обработка пакетов данных с БЧЭ;
- (2) вычисление навигационных параметров объекта, на котором установлена модель, с помощью реализованного алгоритма обработки информации;
- (3) сохранение параметров движения объекта, полученных с БЧЭ во время эксперимента, в лог-файл для их последующего воспроизведения, обработки и анализа результатов.

### 3.2. Анализ результатов испытаний модели

С помощью описанной модели проведен ряд экспериментальных исследований, типичные результаты которых можно наблюдать на рис. 3, 4, позволивших выявить следующие характеристики разработанной системы ЛН.

- (1) Траектория объекта, построенная на основе вычисленных навигационных параметров, обладает конфигурацией, идентичной реально пройденному маршруту (адекватная работа системы ЛН в качестве курсоуказателя).
- (2) Погрешность определения местоположения объекта системой ЛН в автономном режиме составляет при прямолинейном движении  $0,012t^2$  (1,2 м за 10 с), при маневрировании  $0,026t^2$  (2,6 м за 10 с).

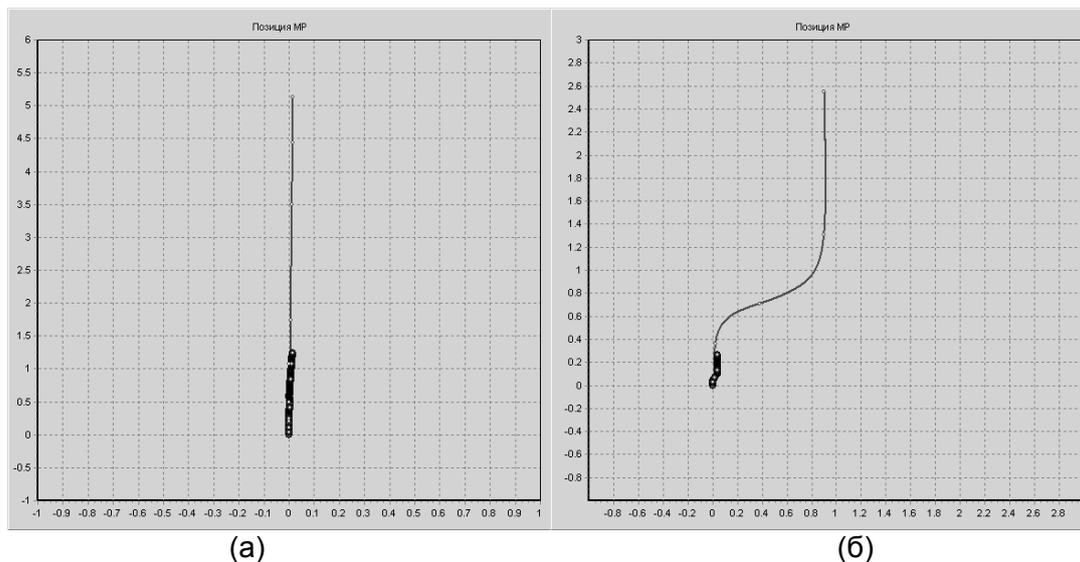


Рис. 3. Траектории объекта, полученные с помощью физической модели (темные кривые), в сравнении с реальными (светлые кривые) при: (а) – прямолинейном движении, (б) – маневрировании

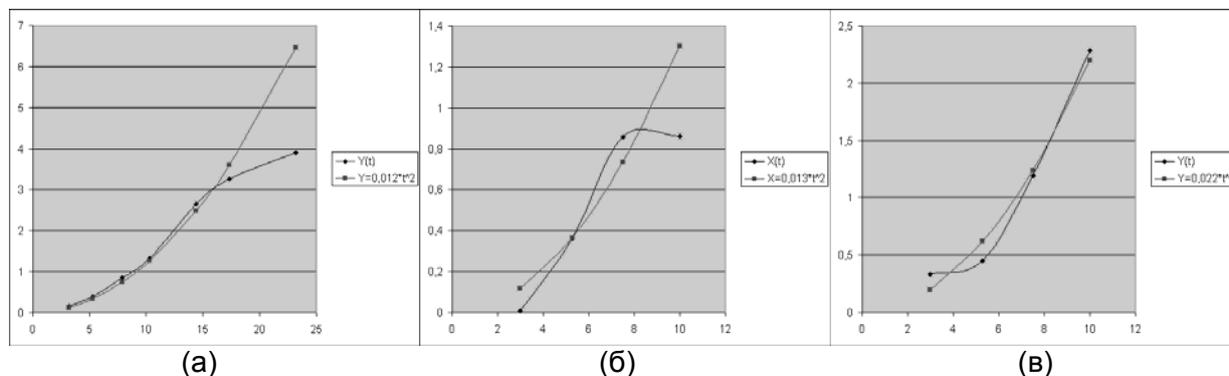


Рис. 4. Зависимости ошибок определения координат объекта от времени (светлые кривые) и функции, их аппроксимирующие (темные кривые), при: (а) – прямолинейном движении, (б), (в) – маневрировании

- (3) Обработка углов ориентации объекта производится с точностью  $0,1-0,3^\circ$  (для крена и тангажа),  $2-3^\circ$  (для угла курса).
- (4) Результаты работы модели системы ЛН по определению углов ориентации объекта аналогичны выдаваемым МИНС «Компанав-2», а в некоторых случаях за счет учета особенностей навигации мобильных роботов показывают большую точность.

На основании перечисленных характеристик модели можно сделать следующие выводы относительно применения разработанной системы ЛН.

- (1) Автономное применение возможно в помещениях в течение коротких промежутков времени (5–7 с), а также на открытых пространствах, где обеспечиваемая точность сравнима с точностью спутниковой навигации, но в течение не более 1 мин.
- (2) Применение в составе навигационного комплекса допустимо и даже имеет некоторые преимущества по сравнению с использованием известных аналогов.

### **Заключение**

По причине возрастающих с каждым годом требований к автономности мобильных робототехнических средств существует необходимость создания сложного навигационного комплекса для наземных мобильных роботов, позволяющего расширить их функциональные возможности по ориентации на местности, составлению карт, проходу по безопасным маршрутам и т.п. В данной работе предложен проект системы локальной навигации – одной из основных составляющих такого навигационного комплекса.

Результаты испытаний физической модели системы подтверждают возможность ее использования в составе навигационного комплекса наземного мобильного робота. Обеспечиваемая системой точность определения местоположения объекта, а также углов его ориентации позволяет применять разработанную систему как автономно для контроля маневров, составляющих движения объекта по безопасному маршруту, в течение коротких промежутков времени, так и в комплексе с другими навигационными средствами (GPS, одометры, дальнометры и т. п.).

В дальнейшем планируется усовершенствование математической модели, алгоритма обработки информации и программного обеспечения, осуществляющего вычисление навигационных параметров объекта, с целью повышения точности системы, а также последующая реализация разработанной системы и интегрирование ее в навигационный комплекс для наземных мобильных роботов.

### **Литература**

1. Бобровский С. Навигация мобильных роботов (в 3 ч.). Ч. 1 // PC Week/RE. – 2004. – № 9. – С. 52.
2. Бобровский С. Навигация мобильных роботов (в 3 ч.). Ч. 2 // PC Week/RE. – 2004. – № 10. – С. 53.
3. Мартыненко Ю.Г. Тенденции развития современной гироскопии // Соровский образовательный журнал. – 1997. – № 11. – С. 120–127.
4. Волович А., Волович Г. Интегральные акселерометры // Компоненты и технологии. Online версия. – Режим доступа: [http://www.compitech.ru/html.cgi/arhiv/02\\_01/stat\\_66.htm](http://www.compitech.ru/html.cgi/arhiv/02_01/stat_66.htm) (19 окт. 2000).
5. Мартыненко Ю.Г. Инерциальная навигация // Соровский образовательный журнал. – 1998. – № 8. – С. 102–108.
6. Бранец В.Н., Шмыглевский И.П. Введение в теорию бесплатформенных инерциальных навигационных систем. – М.: Наука. Гл. ред. физ.-мат. лит., 1992. – 280 с.

7. Степанов О.А. Применение теории нелинейной фильтрации в задачах обработки навигационной информации. – 3-е изд. – СПб: ГНЦ РФ ЦНИИ «Электроприбор», 2003. – 370 с.
8. Городецкий А.Я. Информационные системы. Вероятностные модели и статистические решения: Учеб. пособие. – СПб: Изд-во СПбГПУ, 2003. – 326 с.

*Черноножкин Василий Александрович*

— Центральный научно-исследовательский и опытно-конструкторский институт робототехники и технической кибернетики, аспирант, xou@list.ru

*Половко Сергей Анатольевич*

— Центральный научно-исследовательский и опытно-конструкторский институт робототехники и технической кибернетики, кандидат технических наук, зам. директора, eus@rtc.ru

**УДК 621:658.516.011.56**

## **ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ РАБОТЫ ПРОИЗВОДСТВЕННЫХ ЛИНИЙ МНОГОНОМЕНКЛАТУРНОГО МАШИНОСТРОИТЕЛЬНОГО ПРОИЗВОДСТВА**

**А.А. Баранов, А.Р. Денисов, М.Г. Левин**

В статье приведено описание модуля «Группирование», отвечающего за решение задач технологической унификации проектируемых изделий в автоматизированной системе конструкторско-технологической подготовки производства (КТПП), и обобщенный алгоритм распределения изготавливаемых изделий (балансировки) по производственным линиям.

**Ключевые слова:** календарное планирование, сети Петри, эволюционный алгоритм.

### **Введение**

Одной из тенденций развития современного производства является переход к средне- и мелкосерийности. В этих условиях ключевой проблемой становится повышение гибкости производственной системы, в частности, максимально возможное сокращение сроков подготовки производства и эффективная организация производственного процесса [1]. Однако при решении указанных задач возникает проблема рационального представления результатов технологического проектирования. Так, в соответствии с ЕСТД технологический процесс (ТП) записывается на маршрутной карте в виде линейной последовательности технологических операций (ТО) – маршрута обработки [2]. Такая форма представления позволяет упростить планирование ТП, однако в случае изменения производственных условий (изменение объемов выпуска или номенклатуры выпускаемых изделий, изменение структуры производственных линий (ПЛ) и т.п.) может снизить эффективность производства. Решить данную проблему можно через представление ТП в виде графа следования ТО (сетевого плана ТП) [3, стр. 86], что позволит повысить гибкость производственной системы, так как в этом случае маршрут обработки будет формироваться применительно к существующим в данный момент производственным условиям. Так, например, на основании сетевого плана ТП изготовления женского головного убора (рис. 1) могут быть сформированы следующие маршруты обработки: 1→2→3→4→5→6→7; 1→2→4→3→5→6→7; 1→2→4→5→3→6→7; 1→4→2→3→5→6→7; 1→4→2→5→3→6→7; 1→4→5→2→3→6→7. Однако такой подход требует более сложных алгоритмов планирования. Особенно остро проблема планирования стоит в многономенклатурном производстве, когда необходимо обеспе-

чить одновременное изготовление нескольких типов изделий заданных объемов выпуска на существующих ПЛ (балансировка ПЛ).

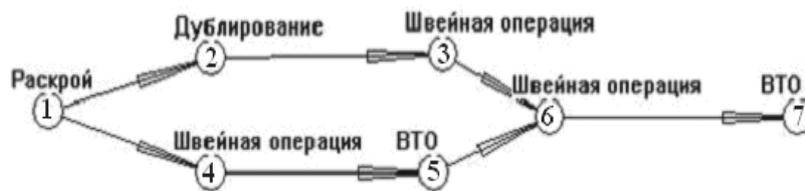


Рис. 1. Сетевой план ТП изготовления женского головного убора

В общем случае задачу балансировки ПЛ для всего множества обрабатываемых изделий  $D$  можно представить в следующем виде:

$$\forall i = \overline{1, |D|} \left( \langle G_{\text{ТП}_i}, \text{Кол}_i \rangle \xRightarrow{\text{ПЛ}} \left\{ \langle \text{ТП}_{ij}, \text{Кпл}_{ij} \rangle \mid j = \overline{1, |\text{ПЛ}|} \right\} \right),$$

где  $G_{\text{ТП}_i}$  – сетевой план ТП изготовления  $i$ -го изделия;  $\text{Кол}_i$  – размер партии для  $i$ -го изделия; ПЛ – множество ПЛ предприятия;  $\text{ТП}_{ij}$  – маршрут обработки  $i$ -го изделия на  $j$ -м ПЛ;  $\text{Кпл}_{ij}$  – объем выпуска  $i$ -го изделий на  $j$ -м ПЛ,

$$\forall i = \overline{1, |D|} \left( \text{Кол}_i = \sum_{j=1, |\text{ПЛ}|} \text{Кпл}_{ij} \right).$$

При анализе маршрутов обработки необходимо учитывать ограничения, накладываемые структурой ПЛ предприятия, которые могут быть связаны с отсутствием в составе ПЛ оборудования, необходимого для выполнения ТО изготовления изделий, а также с расположением оборудования внутри ПЛ. Во втором случае структура ПЛ может исключить некоторые варианты маршрутов обработки одного изделия. Так, например, при развертывании сетевого плана ТП (рис. 1) на ПЛ (рис. 2) недопустимыми являются следующие маршруты:  $1 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7$ ;  $1 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 6 \rightarrow 7$ ;  $1 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 7$ . Это связано с тем, что структура данной ПЛ предполагает первоначальное выполнение операций раскроя и дублирования.

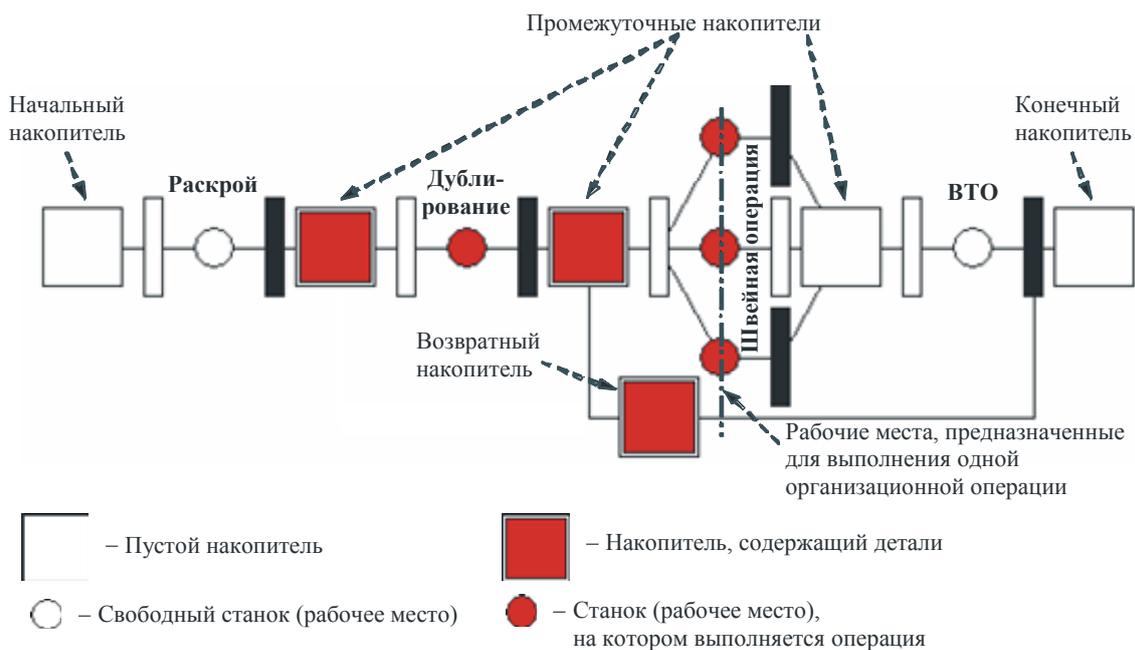


Рис. 2. Пример модели производственной линии в виде сети Петри

Для определения возможности применения маршрута обработки изделия для ПЛ может быть использован алгоритм [3, стр. 88], в соответствии с которым для сетевого плана ТП и структуры ПЛ формируются матрицы достижимости  $M_{тп}$  и  $M_{пл}$ , в которых

$$M_{ij} = \begin{cases} 0, & \text{если вершина } a_j \text{ анализируемого графа достижима из вершины } a_i; \\ 1, & \text{в противном случае.} \end{cases}$$

Далее формируется результирующая матрица  $M_r$ ,

$$M_{r_{ij}} = M_{тп_{ii}} \cdot M_{пл_{ii}} \cdot M_{тп_{jj}} \cdot M_{пл_{jj}} \cdot (M_{тп_{ij}} \oplus M_{пл_{ij}}).$$

Если полученная матрица является «нулевой», то анализируемый маршрут обработки допустим для данной ПЛ, в противном случае – недопустим. Результаты анализа маршрутов  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$  и  $1 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7$  на соответствие ПЛ (рис. 2) представлены на рис. 3. По результатам балансировки ПЛ определяется их загрузка:

$$Z_j = \left\{ \left\langle D_i, TP_{ij}, K_{пл_{ij}} \right\rangle \middle| i = \overline{1, |D|} \right\} \middle| j = \overline{1, |ПЛ|}.$$

При оценке эффективности полученного распределения изделий в первую очередь необходимо учитывать равномерность загрузки ПЛ. В этом случае в качестве критерия эффективности целесообразно использовать время работы предприятия при изготовлении заданных объемов выпуска:

$$T_1 = \sup_{j=1, |ПЛ|} (T(Z_j)) \rightarrow \min,$$

где  $T(Z_j)$  – время работы  $j$ -й линии при выполнении заданных для нее объемов выпуска.

		Маршрут $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$														Маршрут $1 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7$							
		M <sub>тп</sub>							M <sub>пл</sub>							M <sub>r</sub>							
		1	2	3	4	5	6	7	Р	Д	Ш	Ш	В	Ш	В	Р	Ш	Д	Ш	В	Ш	В	
Р	1	1	0	0	0	0	0	0	Р	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Д	2	1	1	0	0	0	0	0	Д	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Ш	3	1	1	1	0	0	0	0	Ш	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Ш	4	1	1	1	1	0	0	0	Ш	1	1	0	0	0	0	0	0	0	0	0	0	0	0
В	5	1	1	1	1	1	0	0	В	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Ш	6	1	1	1	1	1	1	0	Ш	1	1	0	0	0	0	0	0	0	0	0	0	0	0
В	7	1	1	1	1	1	1	1	В	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Р	1	1	0	0	0	0	0	0	Р	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Ш	4	1	1	0	0	0	0	0	Ш	1	0	1	0	0	0	0	0	0	0	0	0	0	0
Д	2	1	1	1	0	0	0	0	Д	1	0	1	0	0	0	0	0	<b>1</b>	0	0	0	0	0
Ш	3	1	1	1	1	0	0	0	Ш	1	0	1	0	0	0	0	0	0	0	0	0	0	0
В	5	1	1	1	1	1	0	0	В	1	0	1	0	0	0	0	0	0	0	0	0	0	0
Ш	6	1	1	1	1	1	1	0	Ш	1	0	1	0	0	0	0	0	0	0	0	0	0	0
В	7	1	1	1	1	1	1	1	В	1	0	1	0	0	0	0	0	0	0	0	0	0	0

Рис. 3. Примеры анализа применимости маршрутов обработки для ПЛ

### Структура модуля «Группирование»

Для решения задач технологической унификации интегрированная система конструкторско-технологической подготовки производства [4] должна быть дополнена соответствующим модулем «Группирование», который отвечает за группирование изделий для их совместного изготовления на многономенклатурных производственных линиях (задача балансировки), нормирование времени изготовления заданной партии изделий,

а также за планирование загрузки производственных линий при изготовлении заданных партий изделий. Для реализации модуля использован аппарат сетей Петри.

Модуль «Группирование» позволяет моделировать производственный процесс как в режиме поточного, так и в режиме бережливого производства (*Lean Manufacturing*) [5]. Эти подходы к организации производственного процесса имеют следующие принципиальные отличия.

1. Если при поточном производстве одно рабочее место чаще всего характеризуется только одним закрепленным за ним видом оборудования, то в бережливом производстве возможно закрепление нескольких видов оборудования. При этом выполняемые на одном рабочем месте технологические операции (ТО) объединяются в одну организационную (ОО).

2. В режиме поточного производства изделие последовательно проходит через все накопители – от начального до конечного, при этом возможна организация циклов через возвратные накопители (рис. 2). В режиме бережливого производства каждая ОО имеет два накопителя (для необработанных и обработанных изделий), при этом после выполнения одной операции изделия перемещаются непосредственно на входной накопитель следующей выполняемой для них операции. Однако такое перемещение возможно только в том случае, если накопитель для необработанных деталей второй операции пуст. В таком режиме порядок перемещения изделий по производственной линии произволен, а возвратные операции отсутствуют.

Модуль «Группирование» позволяет учесть следующие аспекты:

- совмещенные ОО для нескольких производственных линий. Для облегчения визуального восприятия каждая совмещаемая ОО обозначается цветом, отличным от цвета других ОО;
- операции сборки, после выполнения которых множество маркеров полуфабрикатов заменяются на маркер полученного узла, что существенно упрощает визуальный анализ результатов моделирования;
- ОО, при выполнении которых одновременно обрабатываются несколько изделий.

Структура классов, отвечающих за описание производственного процесса, представлена на рис. 4.

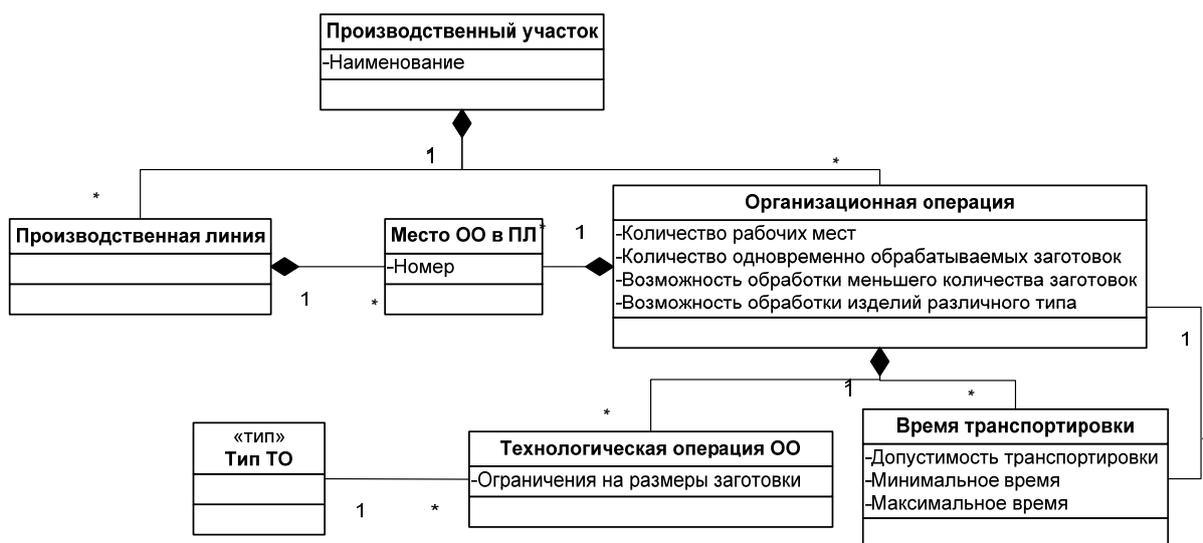


Рис. 4. Структура классов, отвечающих за описание производственного процесса

Для решения задачи балансировки необходимо учитывать возможность изготовления изделий на существующих производственных линиях. Для этого в модуле «Группирование» встроены средства описания ТП изготовления изделия в виде сетевых

го плана (рис. 1), где дуга А→В означает, что операция А должна предшествовать операции В. Структура классов, отвечающих за описание обрабатываемых изделий, представлена на рис. 5.

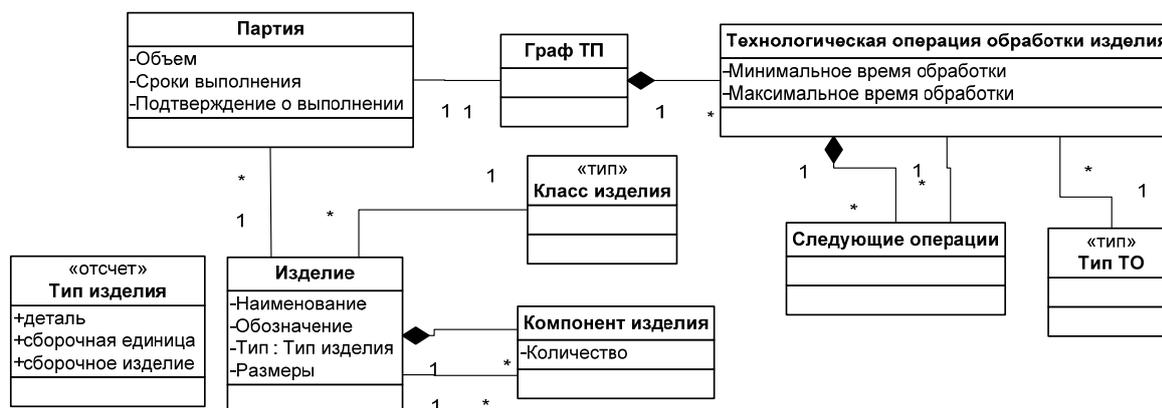


Рис. 5. Структура классов, отвечающих за описание обрабатываемых изделий

### Алгоритм распределения изделий по производственным линиям

В основу решения задачи балансировки нескольких производственных линий положен алгоритм, основанный на принципах эволюционной оптимизации.

*Шаг 1. Определение соответствий (выполнимости) технологических операций изготовления изделий и организационных операций производственных линий.* В результате выполнения этого шага для каждого сочетания «тип изделия – производственная линия» формируется матрица выполнимости, в которой в качестве столбцов выступают организационные операции производственной линии, а в качестве строк – технологические операции изготовления изделия. В каждую ячейку матрицы ставится 1, если технологическая операция может быть выполнена на рабочем месте соответствующей организационной операции, и 0 – в противном случае.

*Шаг 2. Определение всех возможных вариантов изготовления изделий.* На основе матриц выполнимости формируются все варианты изготовления изделий на каждой производственной линии. Каждый вариант проверяется на возможность его выполнения по алгоритму, приведенному в [6]. Если какой-либо вариант не может быть выполнен, то он удаляется, а оставшиеся варианты получают собственные идентификаторы.

*Шаг 3. Первоначальное формирование особей эволюционного алгоритма.* Случайным образом формируются несколько различных особей (вариантов распределения изделий по производственным линиям), где каждая особь имеет вид, представленный в таблице. При этом учитывается, что часть генов особей не будет использоваться вследствие невозможности изготовления некоторых изделий на отдельных производственных линиях.

	Линия 1		Линия 2		...	Линия M	
	№ варианта	Кол-во	№ варианта	Кол-во		№ варианта	Кол-во
Изделие 1					...		
Изделие 2							
...							
Изделие N							

Таблица. Структура хромосомы особи балансировки производственной линии

*Шаг 4. Определение максимального времени работы производственных линий для каждой особи (распределения).* Данный показатель определяет критерия «выживаемости» особей, минимум которой и будет означать наилучшее распределение изделий по производственным линиям. Вследствие наличия в модели стохастической составляющей

шей (рис. 4) необходимо проводить имитационное моделирование работы производственной системы при одном и том же распределении несколько раз, а в качестве времени работы производственных линий брать среднее время для всех экспериментов.

*Шаг 5. Циклическое улучшение полученных популяций методом эволюционной оптимизации.* При выполнении данного этапа применяются следующие виды оптимизации:

- мутация 1 – изменение варианта изготовления одного типа изделия на одной производственной линии;
- мутация 2 – перемещение одного изделия с одной линии на другую; при этом проверяется возможность такого перемещения и, если оно невозможно, новая особь считается «мертворожденной»;
- кроссинговер – обмен между двумя особями распределений одного типа изделия по производственным линиям.

### **Анализ конструкторско-технологического сходства изделий**

При выполнении шага 5 алгоритма в качестве критерия «выживаемости» используется максимальное время работы производственных линий. Это позволяет добиться наилучших показателей технологической унификации изделий. Однако при этом может сложиться такая ситуация, что на одну производственную линию попадут изделия, имеющие различные конструктивные решения. Это приведет к усложнению используемой оснастки и увеличению времени переналадки оборудования. Чтобы минимизировать такие последствия, необходимо на поздних этапах эволюционного алгоритма в качестве критерия «выживаемости» использовать конструкторско-технологическое сходство группируемых изделий на основе формирования множеств наивысшей плотности в пространстве их признаков [7]. При этом критерий «максимальное время работы производственных линий» принимает вид ограничения, например: ухудшение достигнутого на ранних этапах эволюционного алгоритма значения не должно превышать 10%.

Формирование пространства признаков изделий осуществляется по алгоритму [6]. Выделение же подмножеств в пространстве признаков группируемых изделий целесообразно проводить с использованием алгоритмов кластерного анализа, удовлетворяющих следующим условиям:

- 1) малое время работы алгоритма;
- 2) универсальность алгоритма (возможность использования алгоритма на любом множестве кластеризуемых объектов);
- 3) равномерность распределения объектов по формируемым кластерам;
- 4) независимость результатов группирования от начальных характеристик кластеров.

Проведенный сравнительный анализ [7] показал, что наилучшими с точки зрения этих критериев являются алгоритмы К-средних и искусственные нейронные сети Кохонена (ИНСК) при обучении методом «нейронного газа». Дальнейшие эксперименты показали, что существенно улучшить качество кластеризации можно через применение следующих приемов.

- *Равномерное первоначальное распределение центров кластеров.* Такой прием позволяет минимизировать зависимость результатов кластеризации от начального расположения центров кластеров (методы К-средних и ИНСК с обучением методом WTA). Для равномерного распределения все записи массива данных необходимо упорядочить по всем характеризующим их признакам. Сначала все записи сортируются по наиболее значимому признаку. Если несколько записей имеют одинаковое значение данного параметра, то они сортируются в соответствии со вторым по важности параметром, и т.д. Далее в массиве выбираются записи, определяющие начальное положение центров кластеризации, причем выбранные записи должны быть расположены в отсортированном массиве равномерно.

- *Комбинированный метод кластеризации.* В этом случае кластеризация осуществляется в два этапа – первичная кластеризация, например, с использованием ИНСК, и последующее уточнение расположения центров кластеров методом К-средних.

При проведении эксперимента было выявлено, что наилучшие результаты кластеризации достигаются при использовании комбинированного метода на основе нейронного газа и К-средних, что позволяет рекомендовать его в качестве базового метода кластерного анализа. Однако для ряда случаев лучшее качество кластеризации было достигнуто через использование метода К-средних при начальном равномерном распределении центров кластеров. Соответственно, в настройках модуля «Группирование» пользователь имеет право как самостоятельно определить метод кластеризации, используемый при анализе конструкторско-технологического сходства группируемых изделий, так и отказаться от проведения кластеризации, если необходимо добиться наилучших показателей технологической унификации.

### Заключение

Использование модуля «Группирование» позволяет достичь хорошего (близкого к оптимальному) распределения обрабатываемых изделий по производственным линиям. При этом пользователь может оценить полученную загрузку линий через время изготовления изделий на указанных производственных линиях, загрузку и время простоя оборудования внутри линий, время нахождения полуфабрикатов в накопителях, время работы линий и т.п.

### Литература

1. Беляева О.П. Организационные методы повышения гибкости производственных систем: Дис. ... канд. техн. наук / КГУ им. Н.А. Некрасова. – Кострома, 2006. – 126 с.
2. ГОСТ 3.1102-81 ЕСТД. Стадии разработки и виды документов.
3. Технологическая подготовка гибких производственных систем / С.П. Митрофанов, Д.Д. Куликов и др.; Под общ. ред. С.П. Митрофанова. – Л.: Машиностроение, 1987. – 352 с.
4. Денисов А.Р. Принципы конструкторско-технологического проектирования в условиях мелкосерийного машиностроительного производства // Изв. вузов. Приборостроение. – 2007. – Т. 50. – № 12. – С. 56–60.
5. Вумек Д.П., Джонс Д. Бережливое производство: Как избавиться от потерь и добиться процветания вашей компании. – 2-е изд. – М.: Альпина Бизнес Букс, 2005. – 473 с.
6. Денисов А.Р. Автоматизация группирования объектов машиностроительного производства: Дис. ... канд. техн. наук / БГТУ – Брянск, 2001. – 231 с.
7. Денисов А.Р. Сравнение методов кластерного анализа // Информационные технологии. – 2007. – №3. – С. 7–9.

*Баранов Андрей Александрович*

— Костромской государственный технологический университет, аспирант, andrewbaranov@rambler.ru

*Денисов Артем Руфимович*

— Костромской государственный университет им. Н.А. Некрасова, к.т.н., доцент, заведующий кафедрой, iptema@yandex.ru

*Левин Михаил Григорьевич*

— Костромской государственный университет им. Н.А. Некрасова, д.т.н., профессор, iptema@yandex.ru

УДК 620.178

## ИСПОЛЬЗОВАНИЕ ВЕЙВЛЕТ-АНАЛИЗА ДЛЯ ОБРАБОТКИ ПРОФИЛОГРАММ

Д.В. Ершов, Д.В. Запатрин, Н.А. Крылов, В.В. Люлин, В.М. Мусалимов

В работе рассмотрено устройство для экспериментального исследования процессов трения. Сопоставлены динамические характеристики трибопроцессов с микропрофилограммами, установлена корреляция между ними.

**Ключевые слова:** устройство, процесс трения, динамические характеристики, микропрофилограмма.

### Введение

В настоящее время актуальной остается задача более детальных исследований трибологических взаимодействий. Установка «Трибал-2» регистрирует перемещения контробразцов, закрепленных на платформах [1]. Опытные образцы испытываемой трибопары осуществляют возвратно-поступательные движения относительно друг друга. Система позволяет получить максимально точные данные о динамике фрикционного взаимодействия, а использование пакетов Identification Matlab позволяет произвести идентификацию динамики процесса изнашивания. Прибор позволяет в непрерывном режиме времени получать динамические характеристики узла трения, отслеживая их эволюцию.

### Методы исследования

В работе проводится исследование процесса изменения рельефа поверхности и, следовательно, качества поверхности изготовленных образцов. Государственный стандарт на шероховатость поверхности устанавливает единый подход к определению величины шероховатости. Основой для этого является профиль шероховатости и его параметры. Для решения поставленной задачи был проведен эксперимент на установке «Трибал-2» с обеспечением трехточечного взаимодействия образцов латуни ЛС – 59-1 ГОСТ 15527–2004. Некоторые физические свойства материала образца представлены в табл. 1.

Модуль упругости нормальный	105000 МПа
Модуль сдвига	45000 МПа
Относительное сужение	5%
Плотность	8500 кг/м <sup>3</sup>
Предел текучести	410 МПа
Твердость по Бриннелю	160 НВ
Температурный коэффициент линейного расширения	0,000206 1/К

Таблица 1. Физические свойства материала ЛС – 59-1 ГОСТ 15527 – 2004

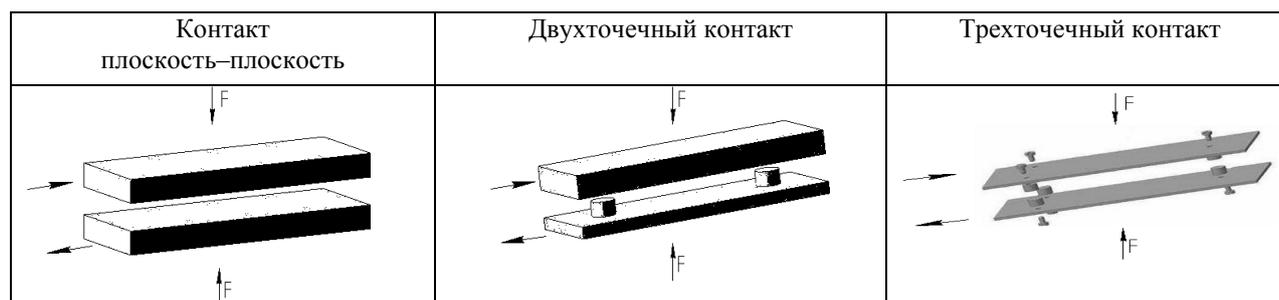


Таблица 2. Схемы контактного взаимодействия трущихся поверхностей

Трехточечный контакт позволил повысить точность измерений по сравнению с процессом трения плоскость–плоскость, а также при двухточечном взаимодействии [2]. Эволюция экспериментов по контактному взаимодействию представлена в табл. 2. Время проведения опыта составило три часа при жестком нагружении контртел трибопар. В результате взаимодействия изменилось качество поверхности образцов. Изображение контактирующей поверхности, полученное с помощью экспериментальной модели микровизора «ЛОМО», представлено на рис. 1.



Рис. 1. Изображение рельефа поверхности: А – первоначальная поверхность образца; Б – поверхность после трения

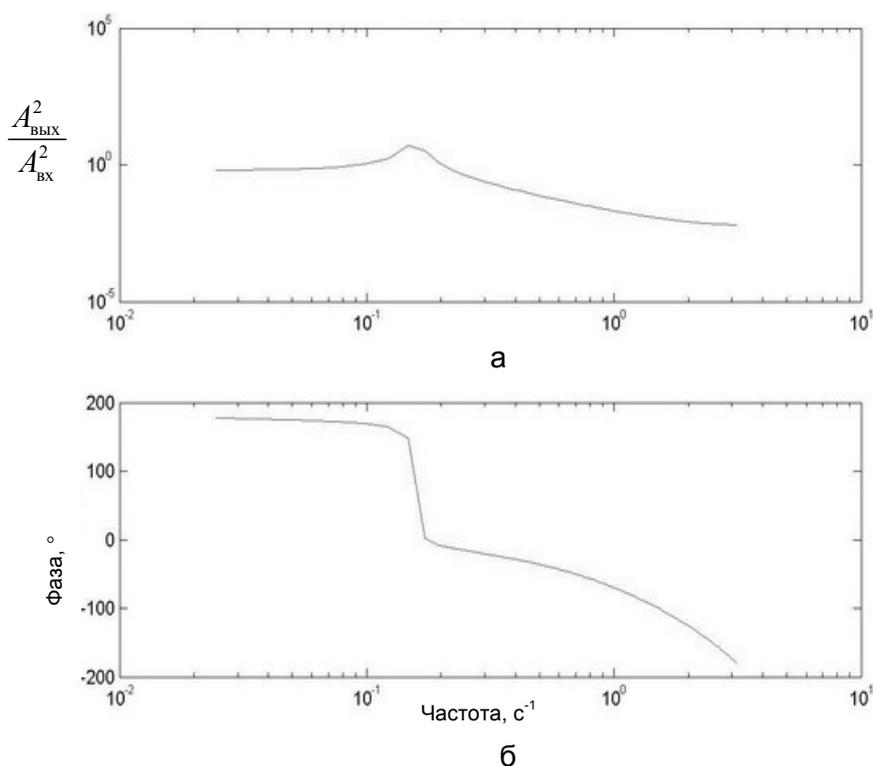


Рис. 2. Амплитудно-частотная (а) и фазо-частотная (б) характеристики фрикционного взаимодействия:  $A_{ВХ}$  – амплитуда входного сигнала,  $A_{ВЫХ}$  – амплитуда выходного сигнала

Далее проводились микроскопический и профилометрический анализы. Кроме того, для детальной оценки изменения профиля поверхности был использован вейвлет-фрактальный анализ [3]. На рис. 2 представлены полученные в пакете Ident программы

Matlab амплитудно-частотные и фазо-частотные характеристики фрикционного взаимодействия.

Микропрофилирование при помощи профилометра PS1 дало качество поверхности, соответствующее параметрам, представленным в табл. 3 и 4.

№ образца, параметр	1	2	3	4	5	6	7 (первоначальная поверхность)
<i>Ra</i>	1,418	2,251	2,589	0,449	2,858	3,059	0,226
<i>Rz</i>	6,256	8,965	12,378	2,447	12,264	13,634	1,233
<i>Rq</i>	1,835	2,859	3,366	0,696	3,613	3,859	0,290
<i>Rt</i>	9,650	14,680	16,380	5,020	18,460	18,050	1,690

Таблица 3. Параметры шероховатости в направлении, перпендикулярном движению контртел, мкм

№ образца, параметр	1	2	3	4	5	6	7 (первоначальная поверхность)
<i>Ra</i>	0,274	0,627	0,748	0,630	1,141	1,160	0,226
<i>Rz</i>	1,447	3,502	4,525	2,950	6,620	5,677	1,233
<i>Rq</i>	0,382	1,001	1,651	0,814	1,825	2,420	0,290
<i>Rt</i>	2,610	7,270	13,650	4,550	13,480	16,100	1,690

Таблица 4. Параметры шероховатости в направлении, параллельном движению образцов, мкм

Интегрально оценить процесс взаимодействия контробразцов можно, используя преобразования Фурье и переход к энергетическим спектрам, как показано на рис. 3 и 4. Очевидно, что в случае снятия профилограммы в направлении, перпендикулярном движению образцов, локальная энергия при одинаковых частотах отличается на порядок.

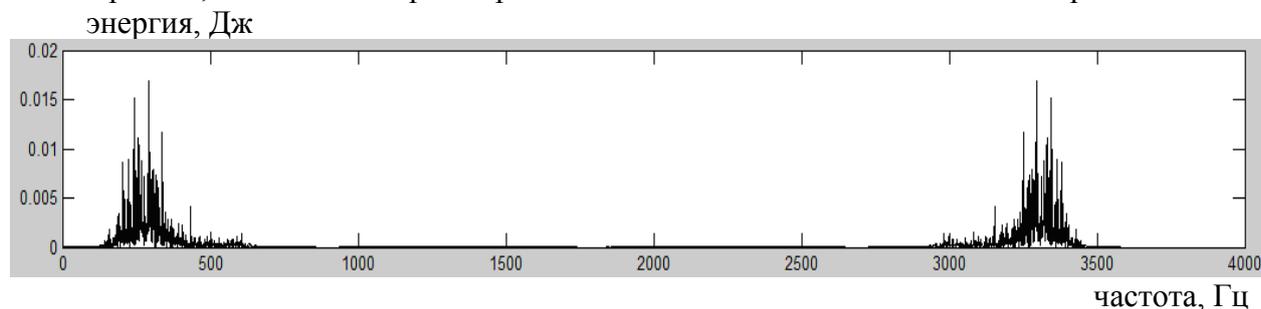


Рис. 3. Энергетическая характеристика поверхности образцов после токарной обработки (в радиальном направлении движения профилометра)

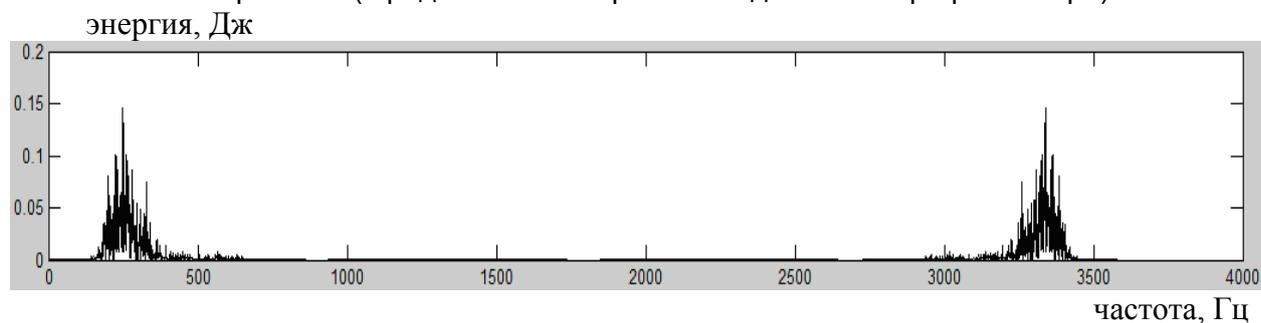


Рис. 4. Энергетическая характеристика поверхности образцов после проведения испытания (при перпендикулярном направлении движения профилометра)

### Заключение

В данной работе был описан эксперимент с использованием трехточечного контакта. Проведено исследование процесса изменения качества поверхности образцов. В статье показаны частоты, которые определяют возможность использования не только характеристик  $Ra$  и  $Rz$ , но и более углубленных параметров шероховатостей. Также здесь сопоставлены динамические характеристики процесса трения с микропрофилограммами и установлена корреляция «эволюция динамической системы – эволюция качества трущихся поверхностей» между ними.

В заключение отметим, что установка «Трибал-2» постоянно модернизируется, и в настоящее время среди подобных трибологических устройств обладает рядом преимуществ: имеет малые габариты; позволяет исследовать характеристики как жестких, так и гибких материалов (металлы, стекло, полимеры и т.д.); использует достижения компьютерных технологий, поддерживает обработку сигналов в режиме реального времени.

## Литература

1. Мусалимов В.М., Валетов В.А. Динамика фрикционного взаимодействия. – СПб: СПбГУ ИТМО, 2006. – 191 с.
2. Бураков В.С., Кerpелева С.Ю. Динамические характеристики и нанощероховатость трущихся поверхностей // Научно-технический вестник СПбГУ ИТМО. – 2002. – № 16.
3. Дьяконов В.П. Вейвлеты. От теории к практике. – 2-е изд., перераб. и доп. – М.: СОЛОН-Пресс, 2004. – 400 с.
4. Труды сессий международной научной школы «Фундаментальные и прикладные проблемы теории точности процессов, машин, приборов и систем». Часть 2 / Под редакцией д.т.н., проф. В.М. Мусалимова и к.т.н., проф. Б.П. Падуна. – СПб: ИП-Маш РАН, 2003. – 172 с.
5. Курбатова Е.А. MATLAB 7. Самоучитель. – М.: Издательский дом «Вильямс», 2006. – 256 с.

*Ершов Денис Владимирович*

Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, Denis071200@mail.ru

*Запатрин Дмитрий Валерьевич*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, zapatrin@front.ru

*Крылов Николай Александрович*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, elmic@narod.ru

*Люлин Виталий Викторович*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, 283\_a@mail.ru

*Мусалимов Виктор Михайлович*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, musvm@yandex.ru

УДК 531.7; 539.3; 535

## МЕТОД ФАЗОВОГО СДВИГА ДЛЯ ИЗМЕРЕНИЯ СТОЙКОСТИ ВОЛОКОННО-ОПТИЧЕСКИХ КАБЕЛЕЙ К ПРОДОЛЬНОЙ РАСТЯГИВАЮЩЕЙ НАГРУЗКЕ

Н.Ю. Овсеев, В.М. Мусалимов

В работе показана связь надежности волоконно-оптического кабеля (ВОК) и его стойкости к продольной растягивающей механической нагрузке. Основным критерием надежности является срок службы кабеля. В работе исследованы механизмы, влияющие на долговечность кабеля. Показано, что механическая продольная нагрузка, прикладываемая к кабелю связи, на практике является наиболее значимым фактором, сокращающим срок службы ВОК. Указаны способы минимизации погрешностей, модернизирована схема метода фазового сдвига. Методика и схема реализованы в действующем испытательном стенде.

**Ключевые слова:** кабель, волокно, растяжение, испытания, фаза, надежность.

### Прогнозирование надежности оптического кабеля

Основным критерием надежности оптического кабеля (ОК) является его срок службы. Согласно нормативным требованиям, срок службы ОК должен быть не менее 25 лет. В конце 1980-х гг. российские кабельные заводы (например, ЗАО «Севкабель-Оптик» и др.) приступили к освоению технологии производства оптического кабеля. Исходя из многолетнего опыта производства электрического кабеля, многие конструкторские и технологические задачи производства оптического кабеля были решены достаточно легко – например, выбор полиэтилена для внешней оболочки кабеля, технология скрутки бронировочного слоя из повива стальной проволоки и т.п. Однако оптическое волокно (ОВ) имеет ряд принципиальных отличий от электрической медной жилы. Оптическое волокно является значительно более чувствительной к механическим воздействиям средой распространения сигнала, чем медная жила. Именно этот факт является решающим при прогнозировании долгосрочной надежности ОК.

По величине механической прочности кварцевое стекло, из которого изготавливаются волокна для оптической связи, сравнимо со сталью. Это природное свойство кварцевых волокон, но из-за несовершенства технологий производства и условий эксплуатации на практике волокно является хрупким и чувствительным к механическим воздействиям, таким, например, как растяжение или кручение.

Прочность оптического волокна определяется в основном наличием поверхностных трещин, которые произвольно располагаются вдоль всей его длины. Величина прочности зависит от длины образца ОВ, скорости приложения нагрузки и условий окружающей среды. Если к волокну не прикладывать нагрузок, то его механические и оптические свойства остаются неизменными, даже если на волокно воздействуют влага, вода, кислоты большой концентрации или другие химические вещества. Если к волокну прикладывать механическое напряжение, то трещины на его поверхности могут увеличиваться. Таким образом, в волокнах, которые подвергаются растягивающему напряжению, кручению или перегибам, имеет место статическая усталость, что со временем приводит к ухудшению их прочностных характеристик или обрыву. Особенно быстро процесс образования трещин и разрушения ОВ происходит в присутствии гидроксильных групп (ОН), поэтому при проектировании защиты ОВ от влаги уделяется большое внимание.

Каждый производитель ОВ (компании Corning Cable Systems, Fujikura и др.) проводит следующий proof-test: ОВ в течение 1 с подвергается растягивающим усилиям, приводящим к его удлинению на 1%, если не происходит обрыв волокна, то оно при-

знается годным. Основываясь на теории разрушения хрупких тел, можно сказать, что если к ОВ, которое прошло proof-test, будут приложены в 3 раза меньшие по величине нагрузки (удлинение на 0,33 %), то срок службы такого волокна составляет 25 лет [1]. Таким образом, по удлинению можно судить о сроке службы и, следовательно, о надежности. Однако, как показывает практика, недопустимое удлинение ОВ (0,33 %) при приложении растягивающей нагрузки к ОК наступает раньше, чем появляется прирост затухания в ОВ, и раньше, чем начинает разрушаться конструкция кабеля.

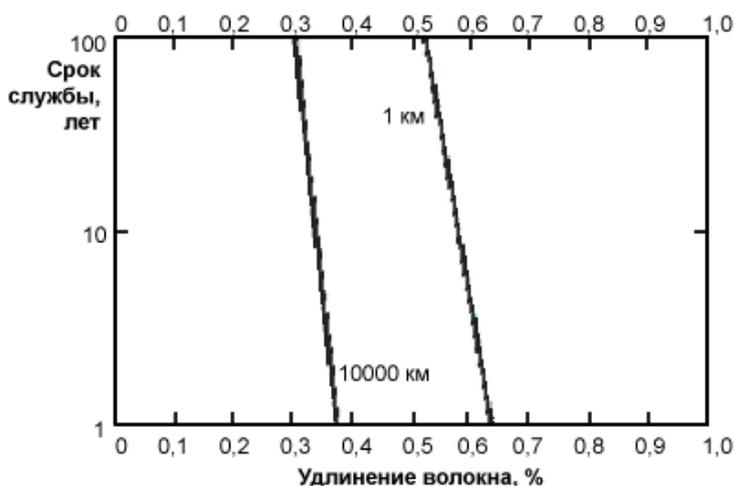


Рис. 1. Зависимость срока службы кабеля от удлинения волокна (по данным фирмы Fujikura, Япония)

Характер зависимости срока службы волокна от его натяжения проиллюстрирован на рис. 1, где натяжение выражено в единицах продольного удлинения. Левая кривая соответствует длине волокна 10000 км, а правая – 1 км, так что между ними располагаются практически все возможные длины волокон в линиях связи. Можно выделить три диапазона напряжений: безопасные – до 0,3 %, недопустимые – более 0,6 % и промежуточные, требующие дополнительного анализа. Хотя в приведенных оценках не учитывался ряд факторов, таких как микрповреждения волокна или влияние влаги, из графика видно, что незначительное увеличение натяжения волокна может приводить к многократному уменьшению его срока службы. Исходя из этого, для прогнозирования долгосрочной надежности ОК при его проектировании и производстве требуется с большой точностью (0,01 %) измерять величину удлинения волокна при приложении растягивающего усилия к ОК. Рассмотрим методы, с помощью которых можно проводить такое измерение.

### Метод контроля затухания

Простейший метод определения допустимой нагрузки для кабеля сводится к следующему: кабель растягивают с помощью силовой установки и при этом постоянно фиксируют прирост затухания в ОВ. Соответствующая экспериментальная зависимость представлена на рис. 2. По уровню затухания принимают решение о том, какую нагрузку может выдержать данная конструкция кабеля. Для стандартного одномодового волокна значение затухания принимается равным 0,22 дБ/км. В приведенном примере (рис. 2), исходя из данной методики оценки кабеля, допустимая нагрузка равна 4,2 кН.

При использовании этой методики можно пойти от обратного: растянуть ОК с усилием, на которое он рассчитан (допустим, 8 кН), и с помощью простейших тестеров определить затухание в ОВ. Если затухание находится в пределах нормы, то кабель считается годным. Преимуществами методики являются простота в использовании и

низкая стоимость оборудования. Однако здесь невозможно получить информацию об удлинении ОВ, а следовательно, сделать долгосрочный прогноз надежности ОК.

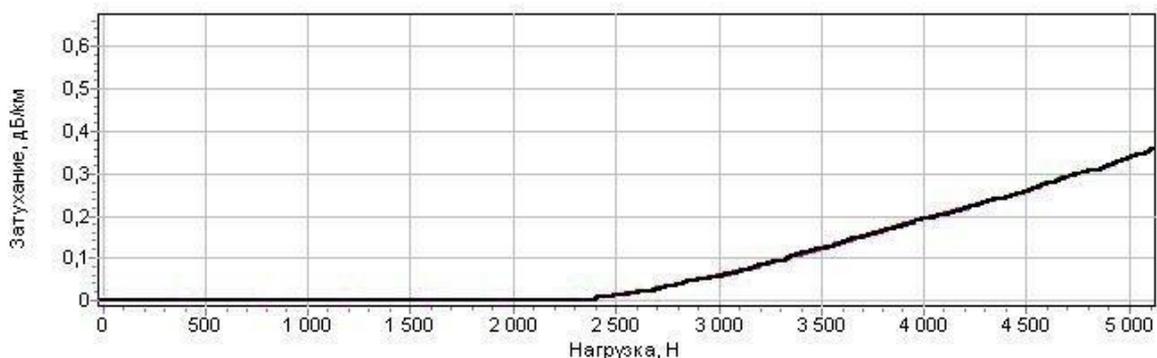


Рис. 2. Зависимость прироста затухания от растягивающей нагрузки

### Метод бриллюэновской рефлектометрии

В оптическом волокне имеются неоднородности, часть из которых обусловлена тепловыми колебаниями атомов. Так, подобно инфракрасному тепловому электромагнитному излучению, в стекле всегда присутствуют гиперзвуковые волны. Рассеяние света на вызванных этими волнами подвижных неоднородностях показателя преломления называется бриллюэновским рассеянием.

Особенностью бриллюэновского рассеяния (например, в отличие от рэлеевского) является то, что вызывающие его неоднородности движутся. В этом случае вследствие эффекта Доплера оптическая частота рассеянного сигнала будет отличаться от частоты лазера. Более того, бриллюэновский сдвиг частот  $\Delta\nu_B$  пропорционален скорости звука и зависит от натяжения волокна, подобно тому, как натяжение струны меняет ее тон. Поэтому, измерив распределение величины  $\Delta\nu_B$  вдоль волокна, можно понять картину распределения напряжений в нем. Для этого прибор должен совмещать возможности рефлектометра и оптического спектрального анализатора.

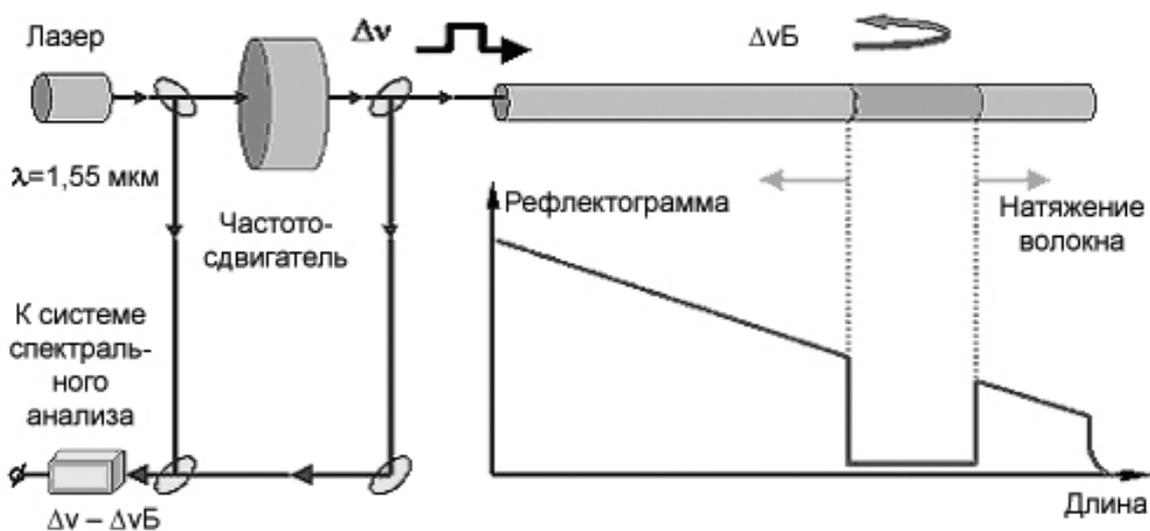


Рис. 3. Схема работы бриллюэновского рефлектометра

Метод анализа кабелей оптической связи с помощью бриллюэновской рефлектометрии имеет большой потенциал. Его главным достоинством является возможность

детально исследовать распределение натяжения волокна в кабеле (все альтернативные методы фиксируют интегральный эффект по всей длине волокна).

Данный метод не получил широкого применения (в России всего несколько таких рефлектометров) по экономической причине – стоимость такого прибора колеблется в пределах 300 000 eur. Кроме этого, расшифровка рефлектограмм, полученных с помощью рефлектометра Бриллюэна, требует высокой квалификации оператора (специалист уровня ведущего инженера), что также увеличивает общую стоимость реализации метода. На данный момент бриллюэновская рефлектометрия в России используется скорее для лабораторий, нежели для кабельных заводов или монтажно-эксплуатационных бригад.

Метод регистрирует натяжение волокна менее 0,05 %, что является практическим пределом точности абсолютных измерений в волокне с неизвестным показателем преломления. При измерениях с известным показателем преломления (для сравнения берется значение при отсутствии механических напряжений) метод бриллюэновской рефлектометрии имеет точность 0,001 %.

### Метод фазового сдвига

При использовании метода фазового сдвига измерение изменения длины ОВ проводится не непосредственно, а путем определения времени распространения оптического сигнала в ОВ. Время распространения сигнала в ОВ рассчитывается по изменению фазы этого сигнала.

Метод реализуется следующим образом. В простейшей схеме (рис. 4) излучение оптического передатчика ОПд, в состав которого входят источник излучения (лазерный диод или светодиод), схема управления его током накачки и схема поддержания температуры, модулируется синусоидальным сигналом генератора с частотой  $f_{\text{мод}}$ . Оптический сигнал ОПд проходит через исследуемое ОВ и поступает на оптический приемник ОПр. Фаза сигнала, с которой был принят ОПр, сравнивается с помощью фазометра с фазой сигнала генератора, который выполняет роль опорного сигнала. По данным о задержке сигнала в исследуемом волокне можно сделать выводы об удлинении ОВ при приложении к ОК растягивающей нагрузки.



Рис. 4. Простейшая схема метода фазового сдвига

Напряжение сигнала на выходе оптического приемника  $U_{\text{ОПр}}$  можно записать как

$$U_{\text{ОПр}} = U_0 \cdot (1 + m \cdot \cos(\omega(t - \tau_{\text{ОВ}} - \tau') + \varphi_0)),$$

где  $U_0$  – среднее значение напряжения,  $m$  – глубина модуляции,  $\omega = 2 \cdot \pi \cdot f$ ,  $\tau_{\text{ОВ}}$  – задержка сигнала в исследуемом ОВ,  $\tau'$  – задержка сигнала в ОПд, ОПр и соединительных ОВ,  $\varphi_0$  – начальная фаза. Измеренное фазометром значение разности фаз равно

$$\Phi = \varphi_{\text{ОПр}} - \varphi_{\text{Г}} = -\omega\tau_{\text{ОВ}} - \omega\tau' + \varphi_0 - \varphi_{\text{Г}},$$

где  $\varphi_{\text{ОПр}}$  и  $\varphi_{\Gamma}$  – фазы сигналов ОПр и генератора соответственно. Поэтому задержка сигнала в исследуемом ОВ составляет

$$\tau_{\text{ОВ}} = -(\Phi - (\varphi_0 + \omega\tau' - \varphi_{\Gamma})) / \omega.$$

Величина  $\varphi_0 + \omega\tau' - \varphi_{\Gamma}$  зависит только от параметров передающей и приемной части устройства и может быть определена с помощью короткого ОВ с известной длиной и показателем преломления. Необходимая точность измерения относительного удлинения ОВ составляет 0,01% от длины испытуемого участка волокна. Так, при длине испытуемого ОВ, равной 50 м, необходимо регистрировать удлинение 5 мм.

### Учет погрешности из-за изменения показателя преломления при удлинении оптического волокна

Изменение длины ОВ является величиной, определяемой при испытаниях. Удлинение ОВ вызывает увеличение времени прохождения сигнала, однако при этом изменяется (уменьшается) показатель преломления ОВ. Поэтому для получения правильного значения изменения длины ОВ результаты измерения должны быть скорректированы. Время распространения сигнала  $t_1$  в ОВ при нормальных условиях равно

$$t_1 = \frac{L \cdot n}{c},$$

где  $L$  – длина волокна,  $n$  – показатель преломления,  $c$  – скорость света. Если в результате воздействия нагрузки одновременно изменяются длина ОВ и показатель преломления, то время распространения становится равным

$$t_2 = \frac{(L + \Delta L)(n + \Delta n)}{c} \cong \frac{L \cdot n + L \cdot \Delta n + \Delta L \cdot n}{c} = \frac{L \cdot n}{c} \cdot \left(1 + \frac{\Delta n}{n} + \frac{\Delta L}{L}\right),$$

где  $\Delta L$ ,  $\Delta n$  – изменение длины ОВ и показателя преломления (предполагается, что они малы по сравнению с  $L$  и  $n$ ). Отсюда следует связь между изменением задержки и изменением длины ОВ:

$$\Delta t = t_2 - t_1 = \frac{L \cdot n}{c} \cdot \left(\frac{\Delta n}{n} + \frac{\Delta L}{L}\right) = \frac{\Delta L \cdot n}{c} \cdot \left(1 + \frac{\Delta n/n}{\Delta L/L}\right), \quad (1)$$

$$\Delta L = \frac{\Delta t \cdot c}{n} \cdot \left(1 + \frac{\Delta n/n}{\Delta L/L}\right)^{-1}.$$

Последнее выражение можно представить в виде

$$\Delta L = \frac{\Delta t \cdot c}{n} \cdot K_s,$$

$$K_s = \left(1 + \frac{\Delta n/n}{\Delta L/L}\right)^{-1}.$$

Поправочный коэффициент  $K_s$  учитывает изменение показателя преломления в результате действия растягивающей нагрузки. Оценку величины коэффициента  $K_s$  можно получить, используя данные [2, 3]. Согласно закону Гука, относительное удлинение ОВ равно

$$\frac{\Delta L}{L} = \frac{\sigma}{E},$$

где  $\sigma$  (Н/мм<sup>2</sup>) – нагрузка, приложенная к ОВ;  $E = 72000$  Н/мм<sup>2</sup> – модуль Юнга кварцевого стекла. Согласно [2], изменение показателя преломления при растяжении равно

$$n = x_s \cdot \sigma,$$

где коэффициент пропорциональности  $x_s = -3,48 \cdot 10^{-6}$  (Н/мм<sup>2</sup>)<sup>-1</sup>.

Подставляя числовые значения в формулы, получим  $K_s = 1,206$ . В работе [3] приведены экспериментальные данные значения поправочного коэффициента  $K_s$ , который

равен 1,26–1,30 для малых значений удлинения волокна ( $< 0,6\%$ ) и 1,22–1,24 при растяжении волокна на 2–3%. Основываясь на этих данных, проводится математическая корректировка результатов испытаний при компьютерной обработке.

### Опорный канал. Учет погрешности из-за влияния температуры

Суть метода состоит в том, что разница фаз между опорным сигналом и сигналом, прошедшим по исследуемому волокну, фиксируется до и после приложения нагрузки к кабелю. По изменению разницы фаз в этих двух ситуациях делается вывод об изменении времени распространения сигнала при нагружении и, далее, об удлинении исследуемого ОВ. В схеме, которая представлена на рис. 4, опорный сигнал от генератора напрямую идет на фазометр, поэтому он изолирован от воздействия внешних климатических факторов (температуры, влажности и т.д.). Исследуемое ОВ находится на испытательном стенде, где условия тестирования не идеальны, поэтому в течение проведения испытания температура ОВ может меняться.

Во время испытаний ОВ на растяжение под воздействием механической нагрузки обязательно должны учитываться температурные условия проведения эксперимента. Изменение температуры окружающей среды вызывает дополнительное удлинение ОВ вследствие линейного теплового расширения и изменение показателя преломления. При этом необходимо учитывать, что часто растяжению подвергается относительно небольшой отрезок ОК (обычно 50–100 м), а изменение температуры воздействует на всю строительную длину ОК (от 2 до 8 км). Для компенсации влияния температуры необходимо в опорный и измерительные каналы измерителя удлинения включать ОВ одного типа и одинаковой длины. Поскольку точно выровнять эти длины, использовать одинаковый тип кабеля и одинаковую тару на практике достаточно сложно и трудоемко, обычно для опорного канала выбирают «средний» тип кабеля (данная классификация условна, например, в нашем случае был выбран кабель марки СКО-ДПС-016А-06-А04×4-7,0) и «среднюю» длину порядка 4 км. Эти допущения оставляют возможность появления погрешности, поэтому проводят дополнительную математическую корректировку результатов.

Влияние температуры на изменение времени распространения сигнала в ОВ можно, по аналогии с (1), записать в виде

$$\Delta t = \frac{L \cdot n}{c} \cdot \left( \frac{\Delta n_T}{n} + \frac{\Delta L_T}{L} \right) = \frac{L \cdot n}{c} \cdot (x_T + \alpha) \cdot \Delta T,$$

где  $L$  – полная длина ОК (или ОВ), подключенного к измерителю удлинения,  $L_T$  и  $\Delta L_T$  – изменение длины ОК (ОВ) и показателя преломления, вызванное изменением температуры окружающей среды на величину  $\Delta T$ ;  $x_T$  – коэффициент пропорциональности при изменении показателя преломления, для ОВ марки SMF-28™ компании Corning Cable Systems  $x_T = 12 \cdot 10^{-6} (\text{°C})^{-1}$ ;  $\alpha = 5,5 \cdot 10^{-7} (\text{K})^{-1}$  – коэффициент линейного теплового расширения ОВ.

Для оценки значения  $\Delta t$  предположим, что температура 1 км ОК (ОВ) увеличилась на 1° С, тогда из последнего выражения получим, что  $\Delta t = 67$  пс. Если 50-метровый участок этого кабеля подвергается испытанию на растяжение и его удлинение составило 0,1 %, то с учетом коэффициента  $K_s$  из формул можно получить  $\Delta t = 206$  пс. Из полученных значений видно, что изменение температуры в процессе проведения испытаний может существенно исказить их результаты.

Цифровой термометр вводится в схему испытательного стенда, чтобы отслеживать изменение температуры окружающей среды  $\Delta T$ . Данные от цифрового термометра поступают на блок сопряжения с компьютером. Далее проводится математическая корректировка результатов.

### Модернизация схемы метода фазового сдвига

Проведенный анализ позволяет модифицировать схему метода фазового сдвига, представленную на рис. 4, до уровня рис. 5. Схема состоит из следующих элементов:

- ОПд – оптический передатчик, в котором находится лазерный диод, схема управления его током накачки и схема температурной стабилизации;
- генератор Г1, формирующий синусоидальный сигнал для модуляции лазерного диода;
- оптический разветвитель ОР;
- два оптических приемника, каждый из которых содержит лавинный фотодиод (ЛФД1 и ЛФД2) и усилители низкой частоты (У1 и У2);
- генератор Г2, формирующий синусоидальный сигнал для модуляции напряжения смещения лавинных фотодиодов;
- аналого-цифровой преобразователь АЦП, который переводит полученные данные в цифровой вид;
- процессор, предназначенный для управления работой блоков схемы и первичной обработки сигналов, поступающих с АЦП.

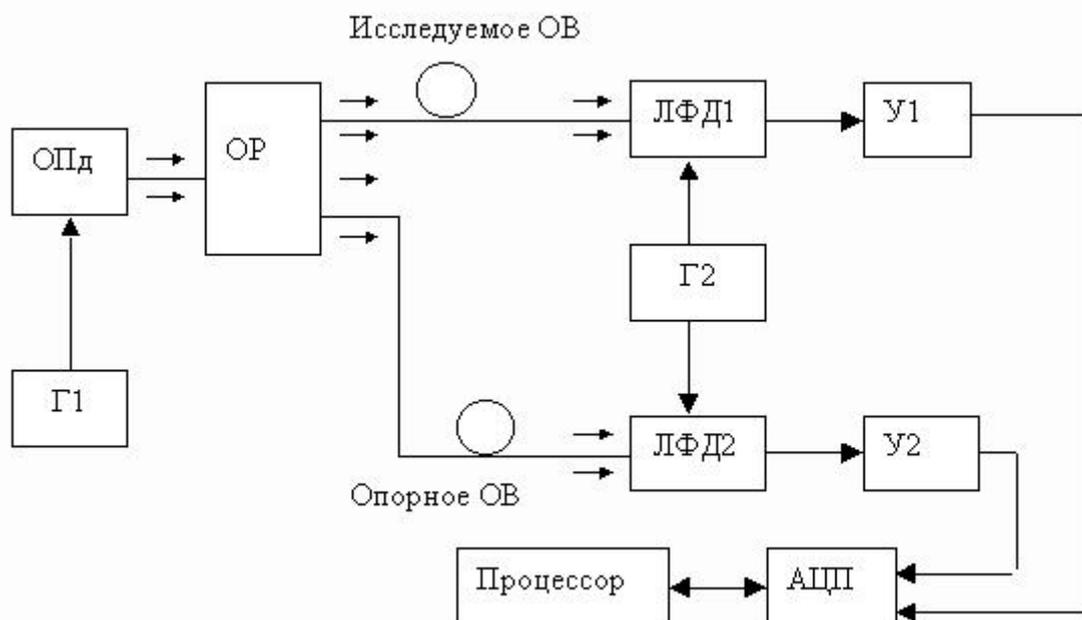


Рис. 5. Модернизированная схема метода фазового сдвига

ОПд генерирует оптический синусоидальный сигнал с частотой  $f_1$  генератора Г1. Оптический разветвитель ОР делит его на два сигнала, которые через исследуемое и опорное ОВ поступают на лавинные фотодиоды ЛФД1 и ЛФД2. Напряжение смещения лавинных фотодиодов модулируется сигналом генератора Г2 с частотой  $f_2$ , близкой к частоте  $f_1$ :

$$\Delta f = f_1 - f_2 \ll f_1.$$

Коэффициент усиления ЛФД1 и ЛФД2 зависит от напряжения смещения, поэтому при его модуляции лавинный фотодиод выполняет роль смесителя сигналов, и в спектре его выходного тока будут присутствовать комбинации частот  $f_1$  и  $f_2$ , в том числе сигнал с разностной частотой  $\Delta f$ . Этот сигнал выделяется на нагрузке фотодиодов и затем усиливается усилителями У1 и У2. Поскольку фаза оптического высокочастотного сигнала, прошедшего через исследуемое или опорное ОВ, переносится непосредственно в фазу низкочастотных сигналов, то разность фаз  $\Psi$  между выходами низкочастотных усилителей У1 и У2 равна

$$\Psi = -\omega_1 \cdot (\tau_1 - \tau_2) + \Delta\varphi,$$

где  $\tau_1$  и  $\tau_2$  – задержки сигналов в исследуемом и опорном ОВ соответственно,  $\omega_1 = 2 \cdot \pi \cdot f_1$ ,  $\Delta\varphi$  – дополнительная разность фаз, обусловленная неидентичностью лавинных фотодиодов и усилителей (значения  $\Delta\varphi$  и  $\tau_2$  определяются при калибровке устройства на коротком ОВ). Лавинные фотодиоды ЛФД1 и ЛФД2 используются в качестве смесителей сигналов, что дает возможность применить низкочастотный оптический приемник и существенно увеличить отношение сигнал/шум. При этом отсутствует дополнительный внешний оптический модулятор и упрощается схема устройства.

### Заключение

Срок службы волоконно-оптического кабеля является основным критерием надежности и составляет 25 лет по нормативной документации. Проведенные в работе теоретические исследования показали, что для прогнозирования этого параметра наиболее оптимальным по соотношению цена/точность измерений является метод фазового сдвига. Метод не только подходит для обеспечения выходного технического контроля при производстве кабеля, но и способен предоставить обратную связь проектировщику кабеля, например, по такому параметру, как избыточная длина волокна в модуле.

Внедрение данного метода на производстве ЗАО «Севкабель-Оптик» (г. Санкт-Петербург) показало правильность сделанных выводов. Достигнута заданная точность измерений (0,01 % относительного удлинения) и относительная низкая стоимость реализации проекта (по сравнению с методом бриллюэновской рефлектометрии).

Модернизированный с учетом погрешностей метод фазового сдвига включен в типичные испытания на длительно допустимую растягивающую нагрузку.

### Литература

1. Листвин А.В., Листвин В.Н. Рефлектометрия оптических волокон. – М.: ЛЕСАРпт, 2005. – 208 с.
2. Williams D.H., Carr J.J., Saikkonen S.L. Single-mode optical fiber index of refraction dependence on product parameters, tensile stress and temperature. – Corning Cable Systems Applications Note, 2001.
3. Abe K., Yoshida K., Daneshvar O., Cait J.J. Photo-elastic correction factor for fiber strain measurements in a cable under tensile load // Journal of lightwave technology. – 1995. – V. 13. – № 1.
4. Портнов Э. Л. Оптические кабели связи. Конструкции и характеристики. – М.: Горячая Линия – Телеком, 2002. – 232 с.
5. Раннев Г.Г., Тарасенко А.П. Методы и средства измерений. – М.: Академия, 2008. – 336 с.

**Овсеев Никита Юрьевич**

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, ovseev@yandex.ru

**Мусалимов Виктор Михайлович**

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, musvm@yandex.ru

УДК 531.746

## АЗИМУТАЛЬНАЯ ВЫСТАВКА ГИРОИНКЛИНОМЕТРОВ ДЛЯ СКВАЖИН ПРОИЗВОЛЬНОЙ ОРИЕНТАЦИИ С ИСПОЛЬЗОВАНИЕМ GPS-КОМПАСА

В.С. Первовский, Я.И. Биндер

Анализируются различные варианты начальной выставки гироинклинометра – от автономного гирокомпасирования до внешнего курсоуказания с применением спутниковых навигационных приборов. Рассматривается возможность осуществления начальной выставки с помощью GPS-компаса, предлагается конструктивное решение для привязки баз приборов.

**Ключевые слова:** азимутальная выставка, гироинклинометр, GPS-компас.

### Введение

Сфера приложения современных задач подземной навигации постоянно расширяется. При этом повышаются требования не только к точности, но и к пространственной частоте определения угловых параметров скважины. Если еще в недавнем прошлом в большинстве случаев было достаточно точечного съема с интервалом 10–50 м, то в настоящий момент все чаще требуется информация с дискретностью до 5 см. В основе подобных требований лежит желание заказчика узнать высокочастотную составляющую биений и изгибов скважины. Эта информация необходима для определения вертикальной глубины скважины, для обсаживания, установки отклонителя и прочих операций. Подобную дискретность может обеспечить только непрерывный режим съема параметров ориентации скважины, для реализации которого необходимо обеспечить работу измерительного модуля в режиме гироазимута. Очевидно, что в силу конструктивных и эксплуатационных ограничений невозможно в подземном приборе реализовать платформенный гироазимут. В то же время реализация гироскопа направления в бесплатформенном исполнении не вызывает особенных сложностей. Но при любой реализации гироскопа направления неизбежно придется решать задачу начальной выставки, от точности решения которой напрямую будет зависеть точность всех последующих измерений [1].

### Алгоритмы определения навигационных параметров

Общие алгоритмы работы бесплатформенных систем общеизвестны, поэтому введем лишь необходимые понятия и соотношения:  $ENh$  – горизонтная система координат с географической ориентацией осей и началом в точке старта;  $ox_1y_1z_1$  – связанная с корпусом инклинометра система координат, развернутая относительно осей  $ENh$  на угол азимута  $A$ , зенитный угол  $\theta$  и угол  $\varphi$  поворота корпуса инерциального измерительного модуля (ИИМ) вокруг продольной оси  $z_1$ . Измерительный модуль обеспечивает вычисление элементов матрицы направляющих косинусов  $C_h^1$ , определяющей взаимную ориентацию трехгранников  $ENh$  и  $ox_1y_1z_1$ .

$$C_h^1 = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} =$$

$$= N \begin{bmatrix} \cos A \cos \varphi + \sin A \cos \theta \sin \varphi & -\cos A \sin \varphi + \sin A \cos \theta \cos \varphi & -\sin A \sin \theta \\ -\sin A \cos \varphi + \cos A \cos \theta \sin \varphi & \sin A \sin \varphi + \cos A \cos \theta \cos \varphi & -\cos A \sin \theta \\ h \begin{bmatrix} \sin \theta \sin \varphi & \sin \theta \cos \varphi & \cos \theta \end{bmatrix} \end{bmatrix},$$

Выражения для угловых скоростей горизонтного трехгранника имеют вид

$$\omega_E = -\dot{\varphi} = -\frac{V_N}{R_\varphi},$$

$$\omega_N = \dot{\lambda}_* \cos \varphi = \Omega \cos \varphi + \frac{V_E}{R_\lambda},$$

$$\omega_h = \dot{\lambda}_* \sin \varphi = \Omega \sin \varphi + \frac{V_E}{R_\lambda} \operatorname{tg} \varphi.$$

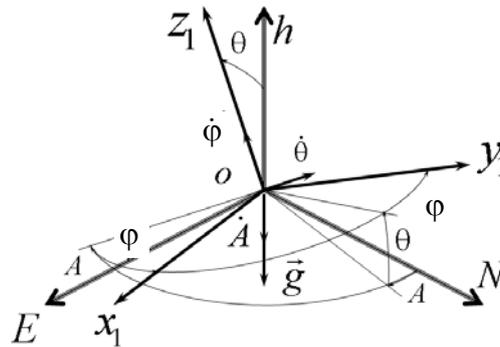


Рис. 1. Системы координат

Кинематические уравнения могут быть получены из соотношений

$$\dot{C}_h^1 = C_h^1 \check{\omega}_1 - \check{\omega}_h C_h^1,$$

где

$$\check{\omega}_h = \begin{bmatrix} 0 & -\omega_h & \omega_N \\ \omega_h & 0 & -\omega_E \\ -\omega_N & \omega_E & 0 \end{bmatrix}, \quad \check{\omega}_1 = \begin{bmatrix} 0 & -\omega_{z1} & \omega_{y1} \\ \omega_{z1} & 0 & -\omega_{x1} \\ -\omega_{y1} & \omega_{x1} & 0 \end{bmatrix}$$

– кососимметрические матрицы, соответствующие векторам угловой скорости вращения

горизонтного  $\check{\omega}_h = N \begin{bmatrix} \omega_E \\ \omega_N \\ \omega_h \end{bmatrix}$  и связанного с корпусом ИИМ  $\check{\omega}_1 = y_1 \begin{bmatrix} \omega_{x1} \\ \omega_{y1} \\ \omega_{z1} \end{bmatrix}$  трехгран-

ников. Очевидно, что для решения уравнения Пуассона необходимо иметь информацию о начальной ориентации инклинометра. От точности этой информации будет напрямую зависеть точность всех последующих измерений. Понятно, что никакая апостериорная обработка не сможет понизить погрешность начальной выставки, поэтому она должна осуществляться с максимально возможной точностью.

## Начальная выставка

С точки зрения точности начальной выставки не представляет никакой сложности определение ориентации прибора относительно горизонтальной плоскости, и, как правило, эта задача успешно решается автономно с помощью акселерометров, которые входят в состав прибора. Определение же ориентации относительно плоскости меридиана места является гораздо более сложной задачей.

Начальная выставка по азимуту может быть осуществлена либо автономно – в режиме компасирования, либо с помощью внешнего курсоуказания. Понятно, что для автономного высокоточного компасирования необходимо иметь в составе прибора точные гироскопы. В то же время чувствительные элементы, устанавливаемые на прибор, должны быть достаточно робастны, так как инклинометр в процессе эксплуатации подвергается сильным механическим воздействиям. Например, во время спуска в обсаженную скважину не исключены удары о стыки обсадных труб, и избежать этого можно, только уменьшая скорость спуска, что в ряде случаев совершенно неприемлемо, так как в современных условиях час простоя буровой установки может обходиться в десятки тысяч долларов. Таким образом, использование в составе прибора высокоточных гироскопов приводит не только к повышению себестоимости прибора за счет более дорогой элементной базы, но и к резкому росту затрат на проведение измерений [2].

Совершенно очевидно, что высокая точность нужна только в момент начальной выставки, во время определения азимута, далее хранение угла в гироазимутальном режиме может быть реализовано на относительно грубых гироскопах. И здесь высокая скорость спуска уже не является однозначно негативным фактором, скорее наоборот: чем быстрее мы спускаемся, тем меньше времени затратим на прохождение скважины, а следовательно, накопим меньшую погрешность. Таким образом, очевиден вывод о необходимости использования внешнего курсоуказания, тогда высокоточный прибор, который обеспечивает азимутальную выставку, будет работать только на поверхности, а измерительный модуль, построенный на грубых робастных гироскопах, отправится вглубь, навстречу ударам о стыки обсадных труб, высокой температуре и прочим неблагоприятным факторам.

## Внешнее курсоуказание

Теперь необходимо выбрать способ внешнего курсоуказания. Здесь нельзя не обратить внимания на стремительный прогресс навигационной аппаратуры, построенной на базе спутниковых навигационных систем GPS и ГЛОНАСС. Очевидно, что прямое использование этих систем для целей подземной навигации невозможно или, во всяком случае, крайне ограничено в силу высокой экранирующей способности земной коры. Хотя существуют варианты применения спутниковой навигационной аппаратуры в системах контроля состояния коммунальных трубопроводов, но, ввиду серьезных эксплуатационных ограничений подобных систем, вряд ли они в ближайшее время получат ощутимое развитие. С другой стороны, для задач начальной выставки спутниковая навигационная аппаратура, с ее беспрецедентно высокими точностными характеристиками, может оказаться как нельзя кстати.

Принцип интерферометрического метода измерения пространственной ориентации для радионавигационных угломерных систем был описан еще в 1955 г. В.Б. Пестряковым. Угловое положение базы в пространстве по сигналам СНС может быть определено на основе измерения разности хода сигналов между двумя антеннами, расположенными на концах базы. Фазовый сдвиг сигнала, принимаемого на две разнесенные антенны, и косинус угла между базой и направлением на спутник связаны выражением

$$\cos \alpha = \frac{\lambda \varphi}{2\pi B}, \quad (1)$$

где  $\lambda$  – длина волны сигнала НКА,  $\varphi$  – фазовый сдвиг,  $B$  – длина базы,  $\alpha$  – угол между вектором-базой и вектором-направлением на НКА. Направляющие косинусы вектора-базы могут быть определены из уравнения на основе скалярного произведения векторов:

$$B \cos \alpha = k_x x + k_y y + k_z z, \quad (2)$$

где  $k_x, k_y, k_z$  – направляющие косинусы вектора-направления на НКА,  $x, y, z$  – координаты вектора-базы. С учетом (1) уравнение (2) можно записать в виде

$$\frac{\lambda \varphi}{2\pi} = \Phi = k_x x + k_y y + k_z z, \quad (3)$$

где  $\Phi$  – фазовый сдвиг, выраженный в единицах длины, который является разностью хода сигналов НКА между антеннами.

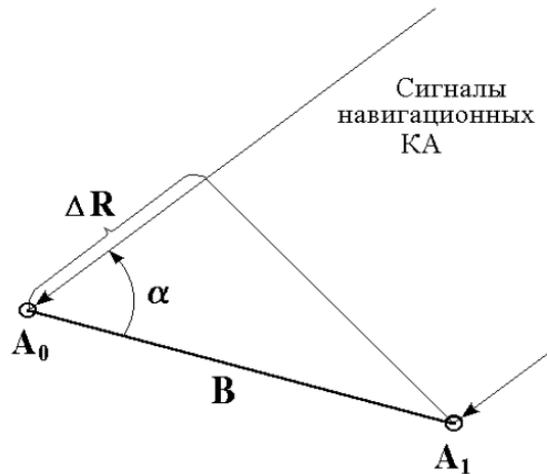


Рис. 2. Измерение ориентации вектора-базы

В общем случае, как видно из (3), число неизвестных ( $x, y, z$ ) равно трем. Для однозначного определения всех неизвестных необходимы три уравнения. Учитывая, что координаты вектора-базы взаимосвязаны, при известной длине базы систему уравнений можно представить в виде

$$\begin{cases} k_{x1}x + k_{y1}y + k_{z1}z = D_1, \\ k_{x2}x + k_{y2}y + k_{z2}z = D_2, \\ x^2 + y^2 + z^2 = D^2. \end{cases}$$

Для нахождения положения вектора-базы в пространстве требуется измерить фазовые сдвиги сигналов двух НКА [3].

При создании угломерной радионавигационной аппаратуры, работающей по сигналам СНС ГЛОНАСС/GPS, основной проблемой является разрешение фазовой неоднозначности. При этом желательно, чтобы алгоритмы разрешения фазовой неоднозначности были одномоментными, т.е. разрешение фазовой неоднозначности должно производиться по результатам измерения параметров сигналов НКА в любой момент времени. Такими методами являются переборные методы, в которых перебираются все возможные значения фазовых сдвигов сигналов НКА, а решение выбирается по критерию максимального правдоподобия. Для решения этой задачи необходима избыточность фазовых измерений, когда число измерений фазы больше числа неизвестных параметров. Это достигается наблюдением значительного числа спутников и использованием второго частотного канала [4]. Эти задачи уже решены, и на данный момент на

рынке представлено довольно много спутниковых навигационных приборов для определения ориентации, полной по трем осям или только по курсу – так называемых GPS-компасов. Эти приборы обладают приемлемой точностью и способны выдавать курс с погрешностью, не превышающей четверти градуса [5].

### Привязка баз приборов

Для осуществления начальной выставки необходимо обеспечить жесткую привязку баз скважного прибора и GPS-компаса. В качестве базовой линии гироинклинометра может выступать ось прибора. Для GPS-компаса в роли такой линии выступает ось, проходящая через фазовые центры антенн. Так как фазовые центры антенн, как правило, не совпадают с точками крепления антенн, то обязательным условием является обеспечение параллельности продольных осей антенн. Естественно, что одним из возможных решений было бы размещение антенн прямо на корпусе гироинклинометра. В этом случае он органично выступал бы в качестве общей базы.

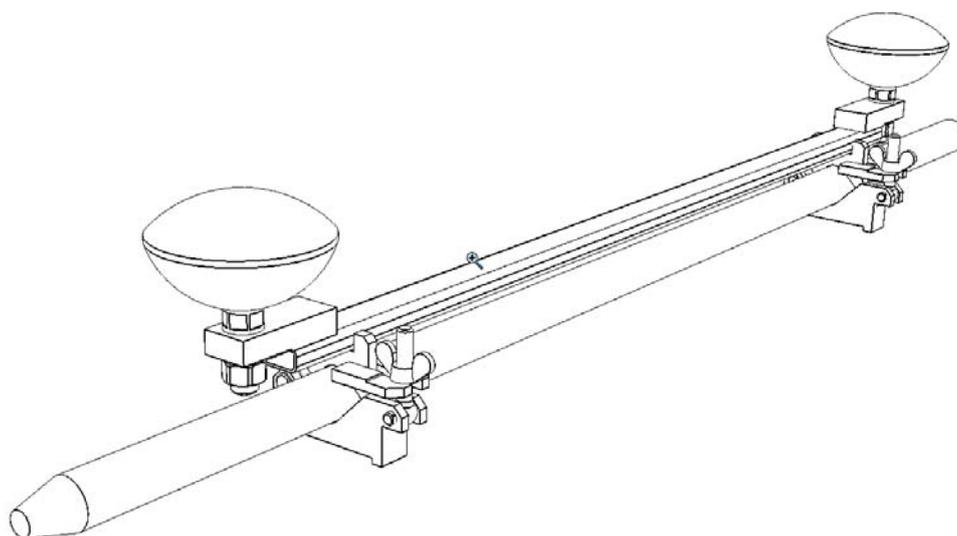


Рис. 3. Кронштейн для сопряжения баз гироинклинометра и GPS-компаса

Но от подобного решения пришлось отказаться в силу того, что корпус прибора в процессе эксплуатации подвергается значительным нагрузкам, в том числе повышенному гидростатическому давлению, поэтому нельзя утончать стенки корпуса. В то же время внешний диаметр прибора ограничен, и развить стенки также нельзя. В результате подробного изучения проблемы было принято решение обеспечить единую базу с помощью трубки прямоугольного сечения, к которой должны крепиться антенны и подземный прибор. Антенны с помощью резьбового соединения крепятся к специальным привалочным плоскостям. Плоскости выполнены в виде монолитных стальных параллелепипедов, приваренных к базовой трубке. Крепление инклинометра обеспечивается призматическими поверхностями, фиксация осуществляется с помощью накладных винтов с «барашками», таким образом, достигается простота конструкции, скорость фиксации и ограничение силы воздействия на прибор. Конструкция оснащена опорами для размещения на горизонтальной поверхности, возможно применение отдельных креплений для установки прибора на другие поверхности. Например, рассматривался вариант подвеса прибора на внешнюю стену буровой платформы.

### Заключение

В рамках работы, в ответ на вызовы современных задач подземной навигации, были рассмотрены некоторые варианты начальной выставки гироскопических инклинометров. Было показано, что для автономной азимутальной выставки в режиме компасирования необходимо применение высокоточных дорогостоящих чувствительных элементов, которые не только удорожают себестоимость конечного изделия, но и резко снижают его эксплуатационные характеристики. В то же время было продемонстрировано, что спутниковые навигационные системы в своем развитии достигли необходимых точностей определения ориентации. Принцип фазовых измерений, предложенный еще в середине прошлого века, позволил построить высокоточные спутниковые системы определения ориентации. В качестве источника внешнего курсоуказания было предложено воспользоваться существующими на данный момент приборами класса GPS-компас. Представлен вариант конструктивного решения, позволяющего связать базы приборов.

Следует отметить, что применение высокоточного внешнего курсоуказания значительно расширяет возможности использования современных инклинометров. Открываются широкие перспективы применения бесплатформенных гироскопических инклинометров в областях, весьма далеких от добывающей промышленности – исторически сложившейся сферы применения инклинометров. Так, в качестве возможного варианта применения новой компоновки инклинометра можно предположить контрольные измерения хладагентных скважин метрополитена. Ранее подобные скважины выпадали из поля зрения инклинометрии в силу очень высоких требований к точности их измерения.

### Литература

1. Калинин А.Г., Кульчицкий В.В. Естественное и искусственное искривление скважин: Учебное пособие для вузов. – Москва-Ижевск: НИЦ «Регулярная и хаотическая динамика»; Институт компьютерных исследований, 2006. – 640 с.
2. Биндер Я.И. Аналитическое компасирование в инклинометрии скважин малого диаметра // Гироскопия и навигация. – 2003. – № 2(41). – С. 38–46.
3. Пестряков В.Б. Радионавигационные угломерные системы. – М.: Госэнергоиздат, 1955. – 304 с.
4. Патент РФ № 2141118. Способ угловой ориентации объектов в пространстве / Фатеев Ю.Л., Чмых М.К. Оpubл. 1999. Бюл. № 31.
5. Tans Vector: Проспект фирмы Trimble, 2004.

*Первовский Владимир Сергеевич*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, [rwladimir@ya.ru](mailto:rwladimir@ya.ru)

*Биндер Яков Исаакович*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, [mail@elmech.ru](mailto:mail@elmech.ru)

**УДК 629.1.033**

## **ПРОГРАММНОЕ ДВИЖЕНИЕ ДВУНОГОГО ШАГАЮЩЕГО РОБОТА В САГИТТАЛЬНОЙ ПЛОСКОСТИ**

**Р.А. Алексеев, Ю.П. Котельников**

Рассматривается комфортабельное движение корпуса двуногого робота, для которого заданы в декартовых координатах программные траектории таза и стоп. Предлагается аналитический расчет задающих воздействий на приводы исполнительного механизма с использованием решения обратных задач кинематики.

**Ключевые слова:** двуногий робот, комфортабельное движение, программные траектории, обратная задача кинематики, точка нуль-момента.

## Введение

Для получения задающих воздействий (ЗВ) на приводы исполнительного механизма (ИМ), осуществляющего комфортабельное движение [1, 2] корпуса двуногого шагающего робота (ДШР), необходимо задать в декартовых координатах либо согласованные (в смысле физической реализуемости кинематической схемой) траектории всех звеньев механизма [3], либо программные траектории части звеньев механизма [1,4–6]. Вычисление на их основании ЗВ составляет обратную задачу кинематики (ОЗК), решение которой неоднозначно для механизма с большим числом звеньев [5, 7–9]. В работе предложено аналитическое решение ОЗК методом разделения общей задачи на составляющие ее частные ОЗК в двухзвенниках, для которых известен закон движения в декартовых координатах, с последующим нахождением ЗВ по теореме косинуса для треугольников, образованных этими двухзвенниками.

## Постановка задачи

Для заданной кинематической схемы ДШР (рис. 1) необходимо обеспечить комфортабельное в сагиттальной плоскости движение корпуса, т.е. движение таза с постоянной скоростью ( $V_T$ ) на постоянной высоте ( $h_0$ ) над опорной поверхностью. Рассмотрим решение поставленной задачи для случая, когда в декартовых координатах задано программное движение корпуса и маховой стопы и следует определить программные задающие воздействия на приводы ИМ, обеспечивающие комфортабельное движение корпуса.

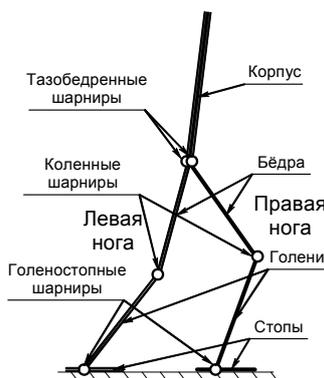


Рис. 1. Кинематическая схема двуногого шагающего робота

## Фазы и параметры движения ДШР

Пусть движение ДШР [4, 10, 11] имеет следующие фазы (см. рис. 2):  
а (одноопорная) – перемещение корпуса и правой ноги с опорой на левую ногу;  
б (двухопорная) – перенос тяжести корпуса с левой ноги на правую ногу с опорой на обе ноги;  
в (одноопорная) – перемещение корпуса и левой ноги с опорой на правую ногу;  
г (двухопорная) – перенос тяжести корпуса с правой ноги на левую ногу с опорой на обе ноги.

Параметрами, определяющими походку робота [7, 12], являются (рис. 2):  $h_0$  – желаемая высота движения таза (тазобедренных суставов),  $l_{ш}$  – длина одного шага,  $l_a$  – протяженность носочной части стопы (от голеностопа до носка),  $l_b$  – протяженность пяточной части стопы (от голеностопа до пятки),  $s$  – начальный фазовый сдвиг таза (относительно голеностопа),  $\Delta h$  – высота подъема стопы над грунтом.

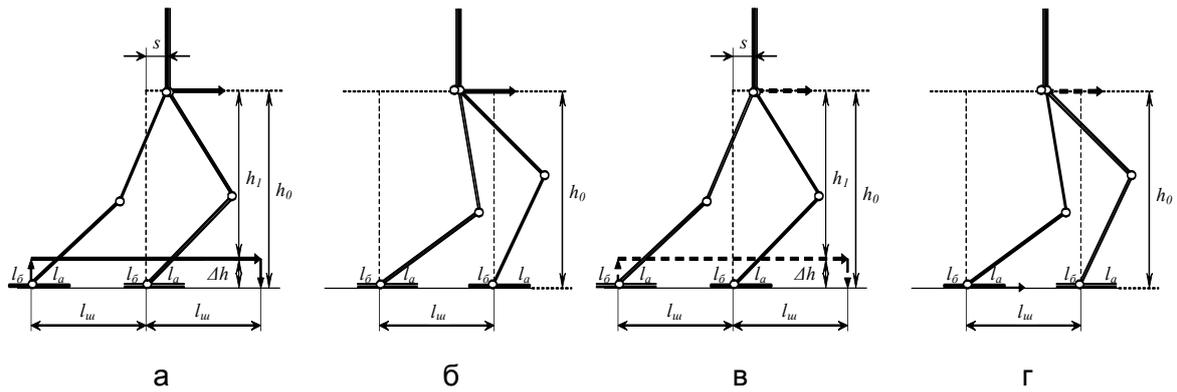


Рис. 2. Фазы движения ДШР

Траектория корпуса, также как и определяющие походку робота траектории стоп, зададим в декартовых координатах. Из всего разнообразия рассмотрены следующие траектории стоп (рис. 3): П-образная траектория (рис. 3, а), содержащая последовательный подъем (подфазы А1/В1), перенос (подфазы А2/В2) и опускание (подфазы А3/В3) стопы и, совмещающие подъем и опускание с горизонтальным переносом, Л-образная (рис. 3, б) и дугообразная (рис. 3, в) траектории, различия между которыми определяются изменением скоростей во времени.

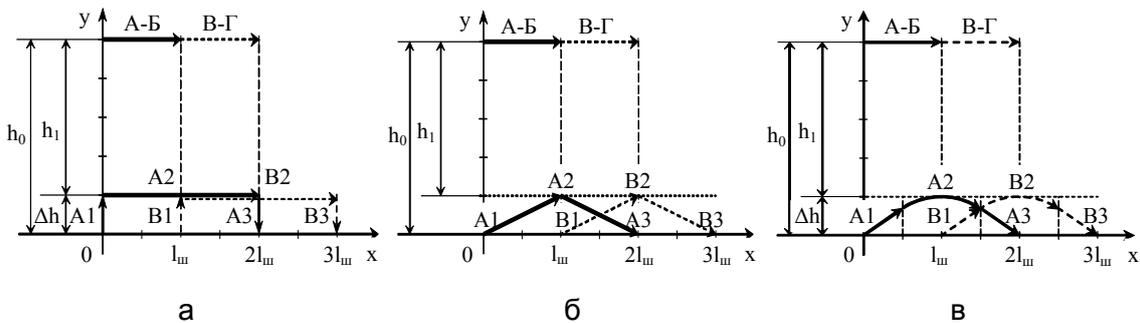


Рис. 3. Траектории таза и стоп. Сплошными линиями показаны траектории (правой стопы и таза) при шаге правой ногой, пунктирными линиями – траектории (левой стопы и таза) при шаге левой ногой

### Системы координат

Для математического описания ДШР введены (рис. 4) следующие декартовы правые системы координат (СК): СК  $O'_{0n}x'_{0n}y'_{0n}$ , (далее СКп), связанная с правой стопой ДШР, СК  $O_0x_0y_0$ , (далее СК0), связанная с корпусом ДШР, а СК  $O'_{0n}x'_{0n}y'_{0n}$ , (далее СКл), связанная с левой стопой ДШР.

СКп и СКл позволяют рассматривать ИМ в фазах ходьбы А-Б и В-Г.

Вектора обобщенных координат  $\Theta'_n = [\theta'_{1n} \theta'_{2n} \theta'_{3n} \theta'_{4n} \theta'_{5n} \theta'_{6n}]^T$ , где индекс «н» принимает значения «п» или «л», однозначно определяют в СКп или СКл расположение отдельных звеньев (корпус, бедра, голени, стопы) и конфигурацию ДШР в целом.

СК0 введена в работе для пересчета обобщенных координат из СКп и СКл с целью получения вектора  $\Theta = [\theta_{1п} \theta_{1л} \theta_{2п} \theta_{2л} \theta_{3п} \theta_{3л}]^T$ , определяющего относительные углы поворота звеньев. Координаты в СК0 однозначно определяют положение механизма при условии, что одна из стоп (опорная) прилегает к опорной поверхности.

Обобщенные координаты в СК0 можно однозначно получить [4, 10, 11], используя следующие выражения:

а) из СКп:

б) из СКл:

$$\theta_{3л} = \pi - \theta'_{1л}, \quad (1.1)$$

$$\theta_{2л} = -\theta'_{2л}, \quad (1.2)$$

$$\theta_{1л} = -\theta'_{3л}, \quad (1.3)$$

$$\theta_0 = \theta'_{1л} + \theta'_{2л} + \theta'_{3л} - \pi, \quad (1.4)$$

$$\theta_{1л} = \theta'_{4л} - \pi, \quad (1.5)$$

$$\theta_{2л} = \theta'_{5л}, \quad (1.6)$$

$$\theta_{3л} = \theta'_{6л}; \quad (1.7)$$

$$\theta_{3л} = \pi - \theta'_{1л}, \quad (2.1)$$

$$\theta_{2л} = -\theta'_{2л}, \quad (2.2)$$

$$\theta_{1л} = -\theta'_{3л}, \quad (2.3)$$

$$\theta_0 = \theta'_{1л} + \theta'_{2л} + \theta'_{3л} - \pi, \quad (2.4)$$

$$\theta_{1л} = \theta'_{4л} - \pi, \quad (2.5)$$

$$\theta_{2л} = \theta'_{5л}, \quad (2.6)$$

$$\theta_{3л} = \theta'_{6л}. \quad (2.7)$$

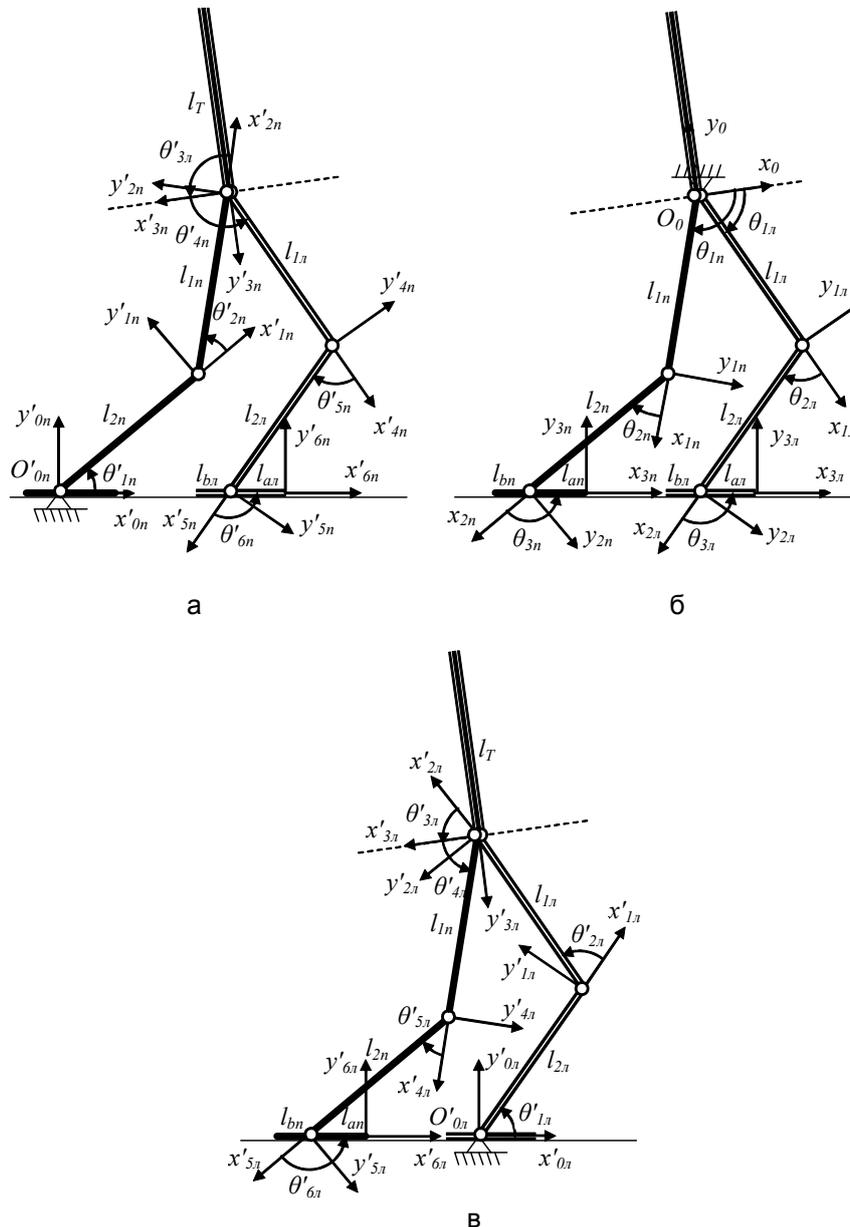


Рис. 4. Системы координат, используемые для математического описания ДШР:  
а – СКп, б – СК0, в – СКл

### Условие кинематической реализуемости походки ДШР

Программное движение корпуса, бедер, голеней, стоп при заданной кинематической схеме и параметрах ДШР возможно только в случае, если выполняются условия кинематической реализуемости походки.

Исходя из геометрических параметров ИМ и параметров ходьбы ДШР, получено кинематическое условие реализуемости походки (рис. 5), имеющее вид

$$(l_{\text{ш}} + s)^2 + h_0^2 \leq (l_1 + l_2)^2, \quad (3)$$

где  $l_1$  и  $l_2$  – длины бедер и голени. Обращение неравенства (4) в равенство позволяет определить наибольшую длину шага ( $l_{\text{ш. max}}$ ) при фиксированных  $s$  и  $h_0$ :

$$l_{\text{ш. max}} = \sqrt{(l_1 + l_2)^2 - h_0^2} - s. \quad (4)$$

### Условие отсутствия неуправляемого движения ДШР

Потребуем, чтобы походка ДШР в сагиттальной плоскости была устойчивой, другими словами, потребуем, чтобы во всех фазах движения стопы опорной ноги не отрывалась носком или пяткой от опорной поверхности под действием моментов сил тяжести звеньев. В противном случае ДШР будет совершать неуправляемое движение, обусловленное моментами сил тяжести звеньев, действующих относительно точки  $O'_{0н}$  – начала координат опорной ноги (см. рис. 4, а, в), которые можно определить по формуле

$$M_i^{(0)} = \sum_{k=i}^7 \left( G_k \cdot \left( \sum_{l=i}^{k-1} \left( l_l \cdot \cos \left( \sum_{j=1}^l \theta_j \right) \right) + l_{ck} \cdot \cos \left( \sum_{j=1}^k \theta_j \right) \right) \right), \quad (5)$$

где  $G_i$  – вес  $i$ -го звена,  $l_{ci}$  – расстояние от  $i$ -го сочленения до  $(i+1)$ -го сочленения,  $l_{ci}$  – расстояние от  $i$ -го сочленения до центра масс  $i$ -го звена,  $\theta_i$  – угол поворота  $i$ -го звена относительно  $(i-1)$ -го звена. Как известно, во всех фазах движения [7, 13] существует точка нуль-момента ( $\tau_{0М}$ ) в которой совокупная реакция опоры (грунта) уравновешивает все силы тяжести звеньев ИМ:

$$R_{\Sigma} = \sum_{i=0}^7 G_i^{(0)}. \quad (6)$$

Из уравнения баланса моментов относительно точки  $O'_{0н}$ ,

$$R_{\Sigma} \cdot x_{\tau_{0М}} = \sum_{i=0}^n M_i^{(0)}, \quad (7)$$

можно определить  $x_{\tau_{0М}}$  – продольную координату точки нуль-момента.

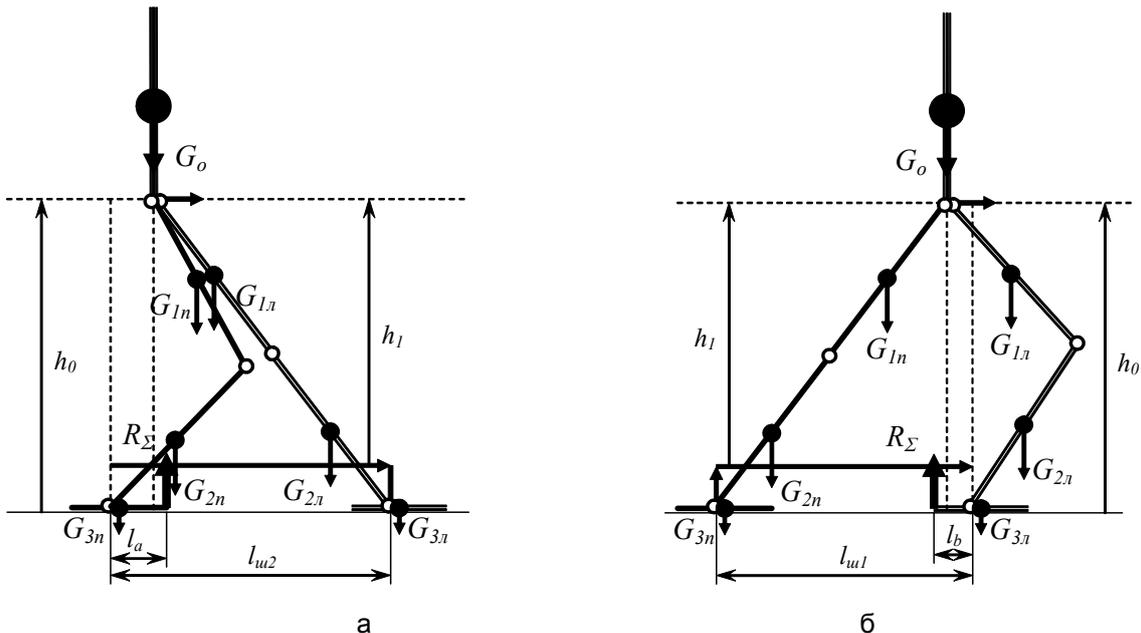


Рис. 5. Силы, действующие на ДШР, при сменах фаз опоры: а – постановка ступни на грунт, б – отрыв ступни от грунта

Условием отсутствия неуправляемого движения ДШР (рис. 5) под действием статических моментов (6) является невыход т0м в одноопорных фазах движения за пределы опорной стопы,

$$l_b \leq x_{т0м} \leq l_a, \quad (8a)$$

а в двухопорных фазах движения – за пределы обеих стоп и промежутка между ними,

$$l_b \leq x_{т0м} \leq l_{ш} + l_a. \quad (8б)$$

Тогда на стыках фаз можно записать условия устойчивого отрыва стопы,

$$l_b \leq x_{т0м}, \quad (9a)$$

и устойчивой постановки стопы,

$$x_{т0м} \leq l_a. \quad (9б)$$

Полученные условия должны быть выполнены при расчете программных траекторий движения ДШР.

### Расчет программных траекторий в декартовых координатах

Движение ДШР с заданной траекторией таза определяется траекториями движения ног. В работе рассматривается решение поставленной задачи путем задания желаемых траекторий таза и стоп в декартовых координатах с последующим определением желаемых обобщенных координат через ОЗК. Для выработки желаемых траекторий движения стоп и таза необходимо формировать для шага правой ногой (фазы А–Б) траектории правой стопы  $(x'_{5л}(t), y'_{5л}(t))$  и таза  $(x'_{2л}(t), y'_{2л}(t))$  в СКл, а для шага левой ногой (фазы В–Г) – траектории левой стопы  $(x'_{5п}(t), y'_{5п}(t))$  и таза  $(x'_{2п}(t), y'_{2п}(t))$  в СКп. Дополнительным условием при ходьбе является поддержание перемещаемой стопы всегда в горизонтальном положении, что соответствует выполнению условия  $\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5 + \theta_6 = 2\pi$  (рис. 4).

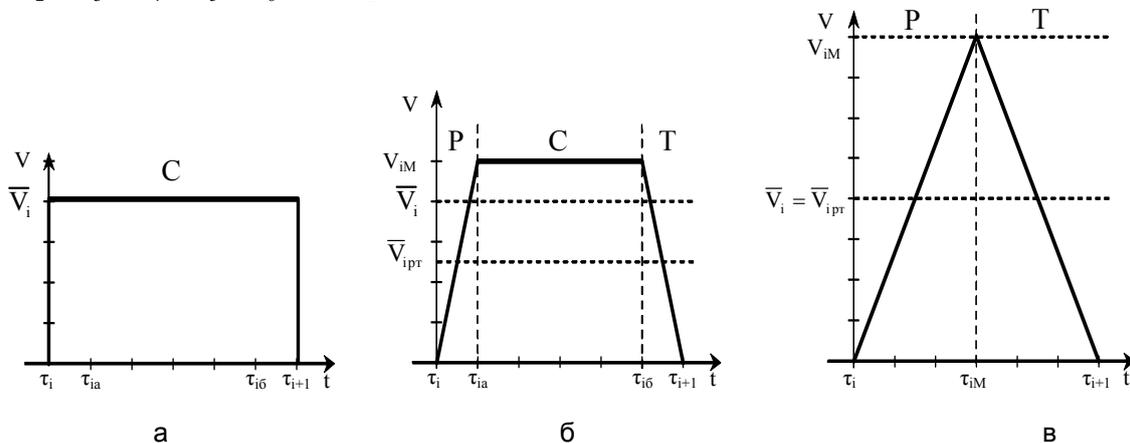


Рис. 6. Графики изменения линейной скорости, используемые в работе:  
 а – постоянное, б – трапециидальное, в – треугольное. Этапы движения:  
 P – разгон, C – стабильное движение, T – торможение

воздействие	постоянное		трапециидальное		треугольное	
наивысшая скорость	$V_{iM} = \bar{V}_i$		$V_{iM} = \frac{\bar{V}_i}{\sigma_i}$		$V_{iM} = 2 \cdot \bar{V}_i$	
время	этапы	скорость	этапы	скорость	этапы	скорость

$\tau_i \leq t < \tau_{ia}$	«С»	$V_{x2M}^* = V_{iM}$	«р»	$V_i^* = \frac{V_{iM}}{\tau_{ia} - \tau_i} (t - \tau_i)$	«р»	$V_i^* = \frac{V_{iM}}{\tau_{ia} - \tau_{iM}} (t - \tau_i)$
$\tau_{ia} \leq t < \tau_{iM}$			«с»	$V_i^* = V_{iM}$		
$\tau_{iM} \leq t < \tau_{i\delta}$			«Т»	$V_i^* = \frac{V_{iM}}{\tau_{i+1} - \tau_{i\delta}} (\tau_{i+1} - t)$	$V_i^* = \frac{V_{iM}}{\tau_{i+1} - \tau_{iM}} (\tau_{i+1} - t)$	
$\tau_{i\delta} \leq t < \tau_{i+1}$						

Таблица 1. Этапы движения и желаемая линейная скорость на них при разных видах воздействий

В работе исследовано движение ДШР при разных заданных траекториях стопы маховой ноги: П-образной, Л-образной и дугообразной (рис. 3). При этом рассматривались желаемые движения с постоянными, трапециидальными и треугольными графиками изменения линейной скорости (рис. 6) при одинаковом времени выполнения заданного движения, из чего следует, что средняя линейная скорость  $\bar{V}_i$  постоянна (табл. 1), где  $i$  – номер задаваемого движения (1 – подъем стопы, 2 – перенос стопы, 3 – опускание стопы),  $\bar{V}_i^*$  – требуемая средняя составляющая линейной скорости движения звена,  $V_{iM}^*$  – максимальная величина составляющей линейной скорости движения звена, а  $V_i^*(t)$  – текущая составляющая линейной скорости движения звена.

### Решение ОЗК и исходное положение робота

По заданным желаемым траекториям таза и маховой стопы ДШР при неподвижной опорной стопе требуется определить задающие воздействия на шесть приводов ИМ, т.е.  $\theta_{in}(t)$ ,  $\theta_{ln}(t)$  при  $i = 1, 2, 3$ . Задача упрощается, если из СКп или СКл перейти в СК0 и получить желаемые траектории голеностопов  $(x'_{2n}{}^*(t); y'_{2n}{}^*(t))$ ,  $(x'_{2i}{}^*(t); y'_{2i}{}^*(t))$  относительно начала координат  $O_0$  (таза ДШР). Таким образом, ОЗК для всего робота распадается на две независимые локальные ОЗК для двузвенников (правой и левой ног).

Используя теорему косинуса для треугольников  $O_0O_1O_2$  и  $O_3O_4O_5$ , можно получить ЗВ на приводы ДШР (желаемые углы взаимной ориентации звеньев):

$$\theta_{2n}^* = -\arccos \frac{l_{1n}^2 + l_{2n}^2 - x_{2n}^{*2} + y_{2n}^{*2}}{2l_{1n}l_{2n}}, \quad (10.1)$$

$$\theta_{1n}^* = -\arccos \frac{x_{2n}^* \cdot (l_{1n} + l_{2n} \cos \theta_{2n}) + y_{2n}^* \cdot l_{2n} \sin \theta_{2n}}{l_{1n}^2 + l_{2n}^2 + 2l_{1n}l_{2n} \cos \theta_{2n}}, \quad (10.2)$$

$$\theta_{3n} = -\theta_{1n} - \theta_{2n}, \quad (10.3)$$

где  $n = п$  для правой ноги и  $n = л$  для левой ноги.

1°. Следует отметить, что знак минус перед функцией арккосинуса соответствует движениям «коленом вперед», что снимает неоднозначность решения ОЗК.

2°. Решив ОЗК при желаемом исходном положении робота, получаем обобщенные координаты исходного положения, которые предварительно должны быть заложены в исследуемую симуляционную модель ИМ.

Полученные в результате компьютерного моделирования в программном продукте Matlab-Simulink [4, 10, 11] программные траектории обобщенных координат, обеспечивающие комфортабельное движение ДШР, приведены в табл. 2. Здесь программные обобщенные координаты обозначены следующим образом: «квадрат» –  $\theta_{1п}^*$  (правый тазобедренный), «треугольник» –  $\theta_{2п}^*$  (правый коленный), «круг» –  $\theta_{1л}^*$  (левый тазобедренный), «ромб» –  $\theta_{2л}^*$  (левый коленный).

В двух правых столбцах приведены последовательно все подфазы движения в цикле ходьбы: А1/В1 – подъем правой/левой ступни, А2/В2 – перенос правой/левой ступни, А3/В3 – опускание правой/левой ступни, Б/Г – перемещение корпуса, р – разгон, с – стабильное движение, т – торможение.

Анализ траекторий относительного движения звеньев (табл. 2) показал, что траектории изменения обобщенных координат могут быть описаны полиномами времени не выше второго порядка. Следовательно, приводы отдельных степеней подвижности должны обладать астатизмом по отношению ко входному воздействию как минимум третьего порядка [14]. Тогда в точке постановки стопы на опорную плоскость будет обеспечена нулевая ошибка позиционирования.

Вид желаемых траекторий движения	Графики задающих воздействий на приводы ДШР	Стадии движения при шаге правой ноги	Стадии движения при шаге левой ноги
П-образная траектория стопы с постоянными скоростями и одновременным перемещением корпуса с постоянной скоростью		<p>А1с+Бс А2с+Бс А3с+Бс Бс</p>	<p>В1с+ Гс В2с+ Гс В3с+ Гс Гс</p>
П-образная траектория стопы с трапецидальными скоростями и одновременным перемещением корпуса с постоянной скоростью		<p>А1р+Бс А1с+Бс А1т+Бс А2р+Бс А2с+Бс А2т+Бс А3р+Бс А3с+Бс А3т+Бс Бс</p>	<p>В1р+ Гс В1с+ Гс В1т+ Гс В2р+ Гс В2с+ Гс В2т+ Гс В3р+ Гс В3с+ Гс В3т+ Гс Гс</p>
Л-образная траектория стопы с постоянными скоростями и одновременным перемещением корпуса с постоянной скоростью		<p>А1с+А2с+Бс А3с+А2с+Бс Бс</p>	<p>В1с+В2с+Гс В3с+В2с+Гс Гс</p>

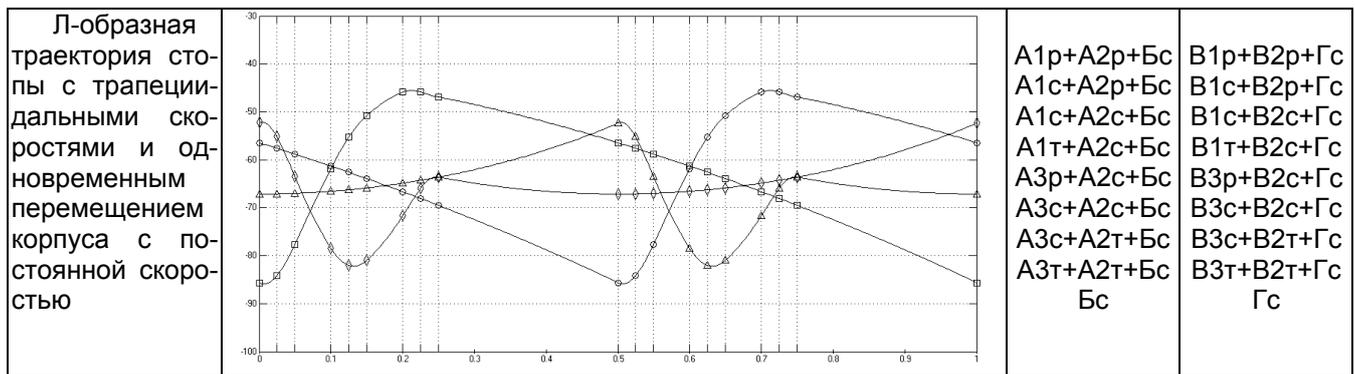


Табл. 3. Виды желаемых траекторий в обобщенных координатах при разных траекториях стоп ДШР

### Заключение

В работе приведены аналитические выражения для вычисления программных траекторий движения таза и стоп в декартовых координатах и получения на их основе задающих воздействий на приводы ДШР. Рассмотрены разные виды программных траекторий корпуса и стоп ДШР, обеспечивающих комфортабельное движение корпуса робота. Приведен анализ полученных при этом задающих воздействий. Даются рекомендации по синтезу системы управления локальными приводами исполнительного механизма.

### Литература

1. Белецкий В.В., Бербюк В.Е. Нелинейная модель двуногого шагающего аппарата, снабженного управляемыми стопами. – М.: Наука, 1982.
2. Chevallereau C., Abba G., Aoustin Y., Plestan F., Westervelt E. R., Canudas-de-Wit C. and Grizzle J.W. RABBIT: A Testbed for Advanced Control Theory // IEEE Control Systems Magazine. – October, 2003. – Vol. 23. – № 5. – P. 57–79.
3. Формальский А.М. Перемещение антропоморфных механизмов. – М.: Наука, 1982.
4. Алексеев Р.А., Котельников Ю.П. Расчет задающих воздействий для двуногого робота // Проблемы машиноведения и машиностроения: Межвузовский сборник. – Выпуск 37. – СЗТУ, 2007. – С. 147–155.
5. Медведев В.С., Лесков А.Г., Ющенко А.С. Системы управления манипуляционных роботов. – М.: Наука, 1978.
6. Смалюк А.Ф. Модель устойчивой пассивной ходьбы // Математическое моделирование деформируемого твердого тела: Сборник статей / Под ред. О.Л. Шведа. – Минск: ИТК НАН Белоруссии, 1999.
7. Белецкий В.В. Двуногая ходьба. Модельные задачи динамики и управления. – М.: Наука, 1984.
8. Бербюк В.Е. Динамика и оптимизация робототехнических систем. – К.: Наукова думка, 1989.
9. Fujimoto Yasutaka, Obata Satoshi, Kawamura Atsuo Robust Biped Walking with Active Interaction Control between Foot and Ground // Proc. of IEEE Int. Conf. on Robotics and Automation. – Leuven, Belgium. – May 1998. – P. 2030–2035.
10. Алексеев Р.А., Мирошник И.В. Разработка алгоритма ходьбы двуногого робота // Научно-технический вестник СПбГУ ИТМО. – 2006. – Вып. 28. – С. 123–132.
11. Алексеев Р.А. Моделирование циклических процессов при передвижении двуногого робота // Научно-технический вестник СПбГУ ИТМО. – 2006. – Вып. 33. – С. 35–47.

12. Чигарев А.В., Михасев Г.И. Биомеханика: Учебное пособие. – Минск: УП Техно-принт, 2004.
13. Вукобратович М.К. Шагающие роботы и антропоморфные механизмы. – М.: Мир, 1976.
14. Бессекерский В.А., Попов Е.П. Теория систем автоматического регулирования. – М.: Наука, 1972.

*Алексеев Ростислав Александрович*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, RostAlexeev@mail.ru

*Котельников Юрий Петрович*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, kotel@mail.ifmo.ru

УДК 621.317.7.027.3; 621.319.027.3

## ПОСТРОЕНИЕ ВЫСОКОВОЛЬТНОГО МОДУЛЯТОРА С НАНОСЕКУНДНЫМ ФРОНТОМ ДЛЯ УПРАВЛЕНИЯ ЭЛЕКТРООПТИЧЕСКИМ ЗАТВОРОМ В СОСТАВЕ ТВЕРДОТЕЛЬНОГО ЛАЗЕРА

Д.С. Терновский, В.В. Тогатов

Рассмотрен режим сверхбыстрого включения МОП-транзистора, при которых время переключения прибора не превышает единиц наносекунд. Дано объяснение механизма сверхбыстрого переключения. Приведены результаты экспериментов, подтверждающих механизм сверхбыстрого переключения. Рассмотрена возможность построения высоковольтного модулятора с наносекундным фронтом на основе эффекта сверхбыстрого переключения МОП-транзистора. Приведены структурная схема и характеристики разработанного высоковольтного импульсного модулятора для управления электрооптическими затворами с амплитудой импульса до 6 кВ и фронтом, не превышающим 5 нс.

**Ключевые слова:** высоковольтная техника, импульсная техника, техника наносекундного диапазона, электрооптический затвор.

### Введение

При формировании высоковольтных импульсов напряжения с наносекундным фронтом используются электронные коммутаторы, построенные на основе различных физических механизмов [1–3]. В последнее время появилась информация о разработке высоковольтных модуляторов с наносекундным фронтом, построенных на основе высоковольтных МОП-транзисторов [4]. Известно, что типовые времена переключения этих транзисторов составляют десятки наносекунд, в то время как фронт импульсов напряжения на выходе модуляторов не превышает единиц наносекунд. В известной нам литературе объяснения механизма такого быстрого переключения МОП-транзисторов не дано.

Данная статья посвящена анализу режима сверхбыстрого переключения МОП-транзистора, его реализации при включении полевого прибора и построению на его основе высоковольтного модулятора с наносекундным фронтом.

### Анализ режима сверхбыстрого включения

Исследуется процесс включения МОП-транзистора в схеме, приведенной на рис. 1. Силовая часть схемы, помимо исследуемого транзистора  $T_2$ , включает сопротивление нагрузки  $R_3$  и высоковольтный источник  $V_H$ . Схема дополнена элементами  $L_C$  и  $L_H$ , учитывающими индуктивности выводов транзистора и токоподводящих дорожек в цепях стока и истока, а также емкостями  $C_{3и}$  и  $C_{3с}$ . Схема управления состоит из стандартного драйвера  $M_1$  и усилительного каскада на  $p$ -канальном транзисторе  $T_1$ . Схема дополнена цепью отрицательного смещения, подаваемого на затвор  $T_2$ . Она включает регулируемый источник  $U_{OBR}$  и резистор  $R_2$ .

На вход исследуемого транзистора  $T_2$  со стока  $T_1$  подаются прямоугольные импульсы напряжения с амплитудой  $U_{ПР}$ . Так как сопротивление канала  $T_1$  равно 0,02 Ом, а импульсный ток стока, идущий на перезаряд емкостей  $C_{3и}$  и  $C_{3с}$ , может достигать 50 А, то источник входного напряжения в первом приближении можно рассматривать как идеальный источник э.д.с. На этапе задержки емкость  $C_{3и}$ , заряженная к моменту включения до обратного напряжения  $U_{OBR}$ , резонансно перезаряжается через индуктивность в цепи истока  $L_H$ . Если напряжение, подаваемое в цепь затвора, равно  $U_{ПР}$  и со-

противление в цепи затвора близко к нулю, то ток истока в момент окончания задержки ( $t = t_3$ ) оказывается равным

$$i_{и}(t_3) = \sqrt{\frac{C_{зи}}{L_{и}} \left[ (U_{пр} + U_{обр})^2 - (U_{пр} - U_{пор})^2 \right]}. \quad (1)$$

Здесь  $U_{пор}$  – пороговое напряжение, определяющее момент окончания задержки  $u_{зи}(t_3) = U_{пор}$ . Оценим величину  $i_{и}(t_3)$ , если в качестве коммутирующего МОП-транзистора используется IRFBE30 со следующими параметрами: крутизна  $s = 3$  А/В, пороговое напряжение  $U_{пор} = 4$  В, емкость затвор-исток  $C_{зи} = 1,3$  нФ и индуктивность в цепи истока  $L_{и} = 5$  нГн. При максимальном размахе входного напряжения  $U_{пр} = U_{обр} = 20$  В величина  $i_{и}(t_3)$  оказывается равной 18,7 А. Этот ток резко возрастает, если индуктивность в цепи истока  $L_{и}$  становится ниже 5 нГн. Как будет показано ниже, увеличение  $i_{и}(t_3)$  принципиально важно для реализации режима сверхбыстрого включения МОП-транзистора.

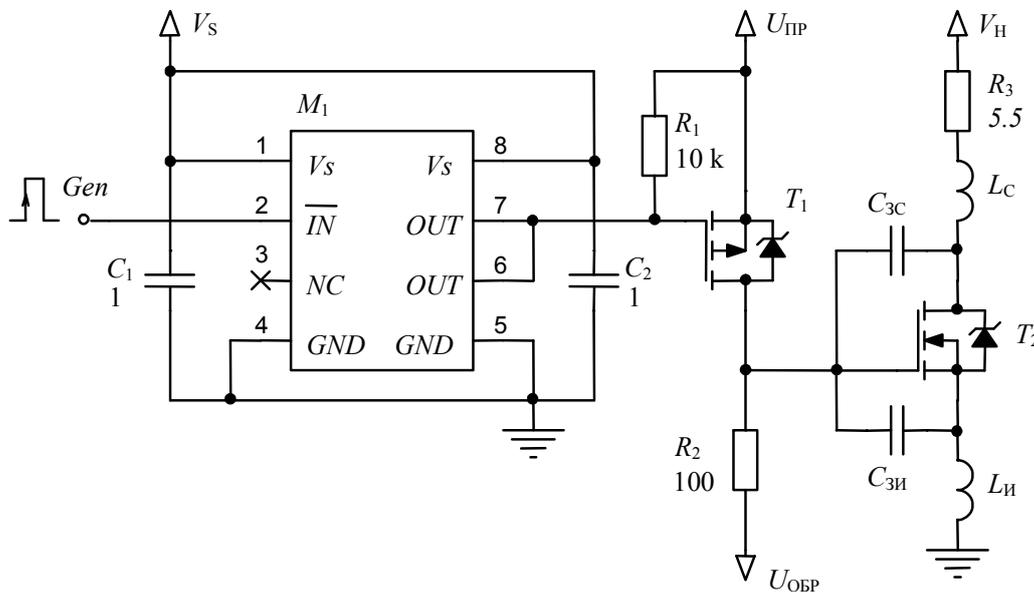


Рис. 1. Схема для исследования процесса включения МОП-транзистора:  $M_1$  – MIC4421ABM,  $T_1$  – IRF7416,  $T_2$  – IRFBE30

После окончания этапа задержки начинается рост тока стока. Как и в работе [5], при анализе процесса включения будем использовать кусочно-линейную аппроксимацию передаточной характеристики МОП-транзистора. Согласно этой аппроксимации, ток стока равен

$$i_c = \begin{cases} 0 & \text{при } U_{зи} \leq U_{пор} \\ s(U_{зи} - U_{пор}) & \text{при } U_{зи} > U_{пор} \end{cases}, \quad (2)$$

где  $s$  – крутизна транзистора,  $U_{зи}$  и  $U_{пор}$  – напряжение затвор-исток и пороговое, соответственно. Можно показать, что при сформулированных допущениях процесс нарастания тока стока при включении МОП-транзистора описывается выражением

$$i_c(t) = I_{пр} \left( 1 - e^{-\frac{t}{sL_{и}}} \right) + i_{и}(t_3) e^{-\frac{t}{sL_{и}}}. \quad (3)$$

Здесь  $I_{пр} = s(U_{пр} - U_{пор})$  – установившееся значение тока стока в активном режиме,  $i_{и}(t_3)$  определено в (1). При выводе уравнения (3) за начало отсчета  $t = 0$  принят момент окончания этапа задержки.

Из выражения (3) следует, во-первых, что рост тока стока при включении происходит с постоянной времени  $\tau = sL_{и}$ . Во-вторых, при  $t = 0$  ток стока скачком увеличивается до значения  $i_{и}(t_3)$ . Этот результат является принципиальным, так как определяет режим сверхбыстрого включения полевого прибора. Физический смысл такого режима заключается в том, что до тех пор, пока ток стока не достигнет величины  $i_{и}(t_3)$ , отрицательная обратная связь, обусловленная индуктивностью  $L_{и}$ , в приборе отсутствует. При этом рост тока стока вплоть до значения  $i_{и}(t_3)$  не превышает единиц наносекунд и определяется темпом заряда емкости  $C_{зи}$  большим током  $i_{и}(t_3)$ . В дальнейшем рост тока стока осуществляется в соответствии с уравнением (3) с постоянной включения  $\tau = sL_{и}$ . В наибольшей степени режиму сверхбыстрого включения отвечают МОП-транзисторы с малой индуктивностью в цепи истока, имеющие отрицательное смещение в цепи затвора в момент включения. Отрицательное смещение увеличивает время заряда емкости  $C_{зи}$  до напряжения  $U_{пор}$ . При этом ток в цепи истока к моменту окончания этапа задержки  $i_{и}(t_3)$  соответственно возрастает.

Очевидно, что режим сверхбыстрого включения может быть реализован только при условии незначительной индуктивности в цепи стока.

Возможность реализации режима сверхбыстрого включения полевого прибора подтверждена нами прямыми экспериментами. На рис. 2 приведены три осциллограммы включения высоковольтного транзистора IRFBE30 на омическую нагрузку 5,5 Ом при напряжении источника питания 400 В. Осциллограмма 1 соответствует включению транзистора прямоугольным импульсом затворного напряжения с амплитудой 10 В без предварительного обратного смещения на затворе. Осциллограммы 2 и 3 соответствуют включению транзистора, при котором импульс затворного напряжения с той же амплитудой подавался на затвор, предварительно смещенный обратным напряжением 10 В и 20 В соответственно. Каждая из трех осциллограмм имеет два явно выраженных участка. Первый – режим сверхбыстрого включения, не превышающий 3 нс, второй – установление стационарного состояния с постоянной  $sL_{и}$ . Зависимость тока истока в момент окончания задержки  $i_{и}(t_3)$  от обратного напряжения  $U_{обр}$  дается выражением (1). При указанных выше значениях параметров транзистора IRFBE30 величины тока  $i_{и}(t_3)$ , рассчитанные по формуле (1) для трех значений обратного напряжения 0, 10, 20 В, оказались равными 4,2, 9,75 и 15 А. Соответствующие экспериментальные значения  $i_{и}(t_3)$ , найденные из осциллограмм на рис. 2, составили 7,2, 14,4 и 21,6 А. Отмеченное различие связано с тем, что время переключения транзистора в этом режиме хоть и мало, но конечно. Поэтому к моменту, когда ток стока достигает тока истока, последний успевает существенно возрасти относительно своего значения в момент окончания задержки.

На рис. 3 приведены три осциллограммы включения того же транзистора IRFBE30 на нагрузку 43 Ом при напряжении источника 400 В и прямом затворном напряжении  $U_{пр} = 10$  В. Так как крутизна этого транзистора составляет  $s = 3$  А/В, то он заведомо включался с заходом в режим насыщения. При этом ограничение тока стока осуществлялось на уровне 9 А. Осциллограмма 1 снималась при отсутствии напряжения обратного смещения в цепи затвора, а осциллограммы 2 и 3 – при наличии предварительного обратного смещения 10 и 20 В соответственно. Как и в предшествующем случае (рис. 2), расчетные значения тока  $i_{и}(t_3)$ , соответствующие обратным напряжениям в цепи затвора 0, 10 и 20 В, равны 4,2, 9,75 и 15 А. Два последних значения тока превосходят ток, определяемый внешней цепью и равный 9 А. Поэтому на осциллограммах 2 и 3 режим сверхбыстрого включения реализуется вплоть до ограничения тока. При этом время включения составляет 2–3 нс. На осциллограмме 1 эксперименталь-

ное значение тока  $i_H(t_3)$  равно 6 А, что близко к расчетному значению. После достижения этой величины темп роста тока резко снижается, и дальнейшее изменение тока вплоть до ограничения осуществляется в соответствии с постоянной установления  $\tau = sL_H$ . Таким образом, рассмотренные экспериментальные зависимости тока стока на рис. 2 и 3 хорошо укладываются в концепцию сверхбыстрого включения транзистора.

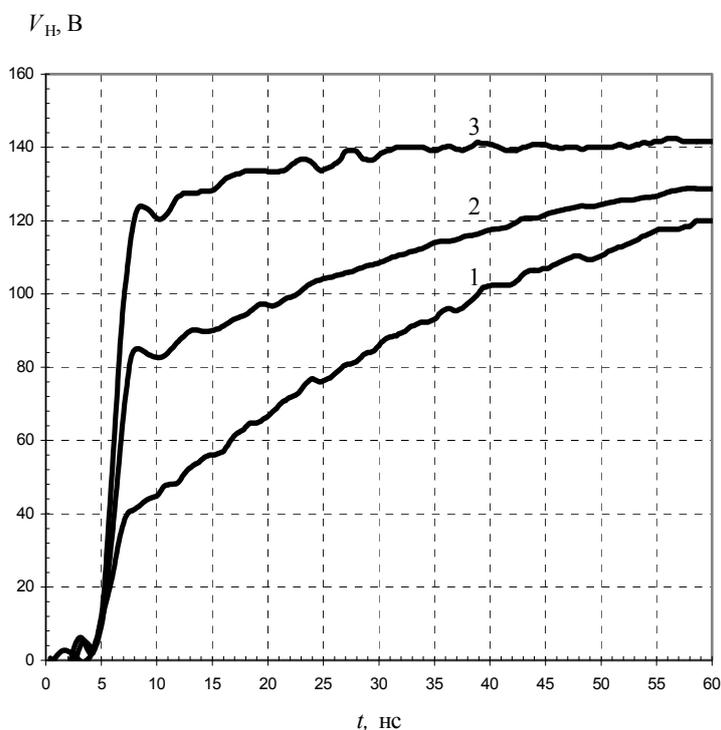


Рис. 2. Осциллограмма напряжения при включении транзистора IRFBE30 на резистивную нагрузку  $R_3 = 5,5$  Ом,  $V_H = 400$  В,  $U_{ПР} = 10$  В; 1 –  $U_{ОБР} = 0$  В; 2 –  $U_{ОБР} = 10$  В; 3 –  $U_{ОБР} = 20$  В

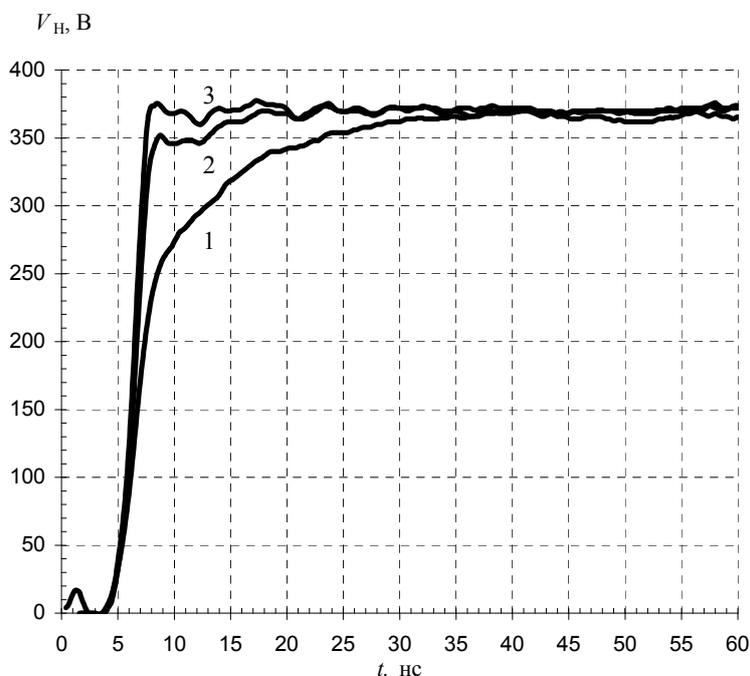


Рис. 3. Осциллограмма напряжения при включении транзистора IRFBE30 на резистивную нагрузку  $R_3 = 43$  Ом,  $V_H = 400$  В,  $U_{ПР} = 10$  В; 1 –  $U_{ОБР} = 0$  В; 2 –  $U_{ОБР} = 10$  В; 3 –  $U_{ОБР} = 20$  В.

В рамках общепринятых представлений длительность процесса включения определяется зарядом собственных емкостей транзистора. В этом смысле при подаче в цепь затвора предварительного обратного смещения время включения должно увеличиться, так как при включении происходит дополнительный перезаряд емкостей транзистора. Однако в режиме сверхбыстрого включения, согласно сказанному выше, имеет место обратная картина, что и подтверждается результатами прямых экспериментов.

Смысл режима сверхбыстрого включения можно пояснить и иначе. Максимальная скорость роста тока в полевом транзисторе ограничена величиной

$$\left(\frac{di}{dt}\right)_{\max} = \frac{U_{зи} - U_{пор}}{L_{и}}$$

В кремниевых МОП-транзисторах  $U_{зи} \leq 20$  В, так как при больших напряжениях возможен пробой диэлектрика (оксида). Если принять  $U_{зи} - U_{пор} = 15$  В, а  $L_{и} = 10$  нГн, то  $(di/dt)_{\max} = 1,5$  А/нс. Режим сверхбыстрого включения позволяет обойти это ограничение за счет предварительной накачки тока в индуктивности истока, осуществляемой по цепи затвора.

### Схема модулятора

Режим сверхбыстрого включения использован нами при создании высоковольтных модуляторов с наносекундными фронтами для управления электрооптическими затворами. Импульсы напряжения, формируемые этими модуляторами, при амплитуде 2–6 кВ характеризуются длительностью фронта 2–4 нс [6].

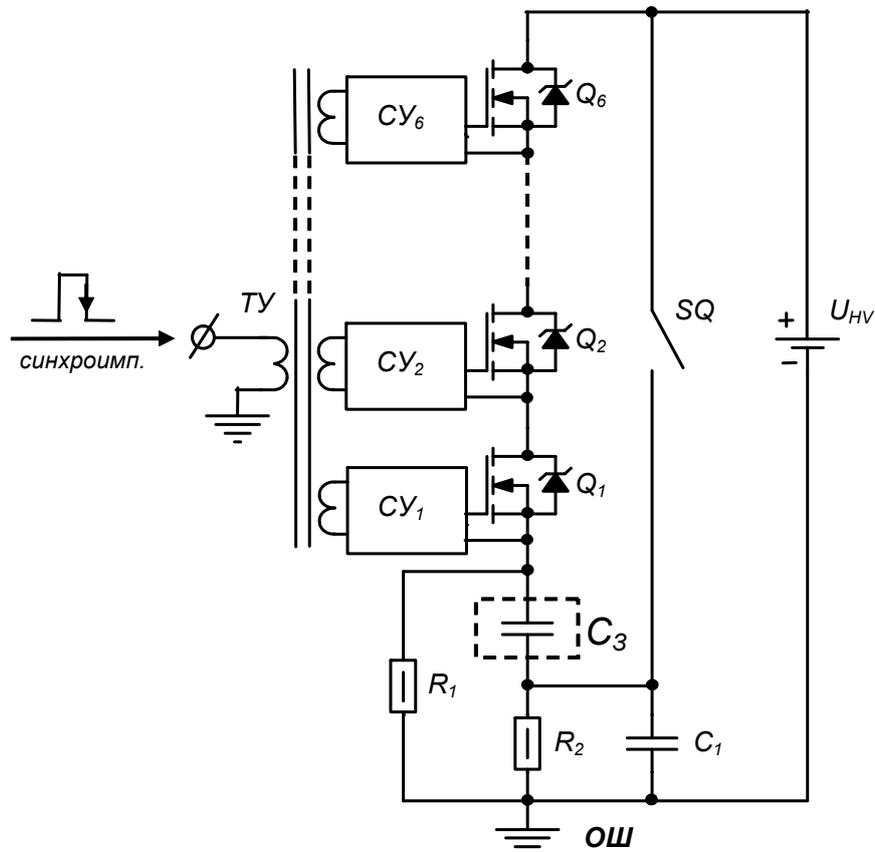


Рис. 4. Схема модулятора

Схема модулятора представлена на рис. 4. Нагрузкой является емкость электрооптического затвора  $C_3$ . Модулятор включает в себя источник высокого напряжения

$U_{HV}$ . Функции коммутирующего элемента выполняет последовательное соединение высоковольтных МОП-транзисторов  $Q_1-Q_6$ . Каждый транзистор имеет независимую схему управления  $CV_1-CV_6$ . Подача синхроимпульсов осуществляется с помощью трансформатора управления ТУ.

В паузе между импульсами высокое напряжение блокируется транзисторами  $Q_1-Q_6$ , напряжение на затворе  $C_3$  равно нулю. При подаче синхроимпульса на вход модулятора трансформатор управления ТУ передает команду на включение каждой схеме управления  $CV$ . Схемы управления одновременно включают высоковольтные МОП-транзисторы в режиме сверхбыстрого включения. Это позволяет зарядить емкость электрооптического затвора  $C_3$  большим импульсным током за единицы наносекунд. Ток заряда замыкается по контуру  $(+)U_{HV}-Q_6-Q_1-C_3-C_1-OШ((-)U_{HV})$ . Емкость конденсатора  $C_1$  много больше емкости затвора  $C_3$ .

В схеме предусмотрена подача уставки высокого напряжения при помощи выключателя  $SQ$ . Этот выключатель также обеспечивает работу модулятора в режимах «switch-on» (с положительным фронтом) и «switch-off» (с отрицательным фронтом).

В отличие от существующих, разработанный модулятор выполнен в виде единой печатной платы, на которой расположен и высоковольтный источник. Для работы в режиме «switch-off» на плате предусмотрен разъем, к которому подключается дополнительный высоковольтный источник. Этот источник разработан нами и может поставляться в комплекте с модулятором. Модулятор (рис. 5) включает следующие основные компоненты: быстродействующий электронный коммутатор, блок накопительных конденсаторов, заряжаемых в паузе между импульсами, управляющие драйверы, трансформатор, с помощью которого осуществляется запуск модулятора, высоковольтный источник питания для формирования выходных импульсов в режиме «switch-on» и низковольтный – для цепей управления. Модулятор управляется внешними синхроимпульсами с частотой до 10 кГц. Регулировка амплитуды выходного импульса осуществляется либо встроенным потенциометром, либо дистанционно через входной разъем. Для контроля состояния платы используется светодиодная индикация, а при отсутствии высокого напряжения генерируется логический сигнал «Error». Питание платы осуществляется от низковольтного источника 5–30 В.

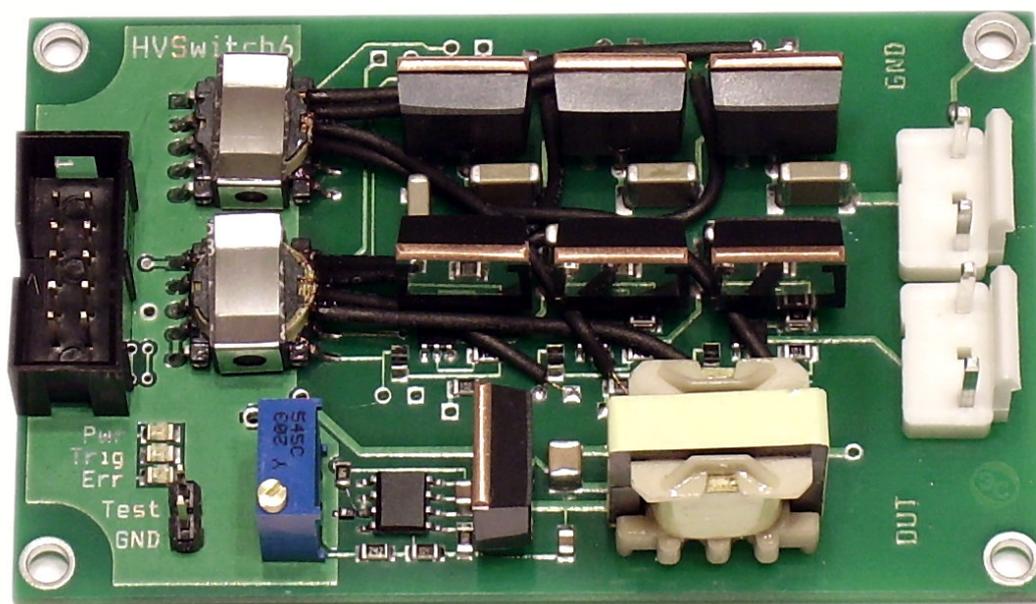


Рис. 5. Электрическая плата модулятора

Импульс напряжения, формируемый модулятором, приведен на рис. 6. Длительность фронта составляет не более 3 нс при амплитуде 4 кВ.

Проведены испытания разработанного высоковольтного импульсного модулятора в составе лазерного излучателя, установленного в технологический комплекс, предназначенный для объемной лазерной маркировки внутри прозрачных диэлектриков. В качестве нагрузки использовался электрооптический затвор типа *QDN3* («*VITRO LASER*» *GmbH*) на кристалле *BBO*, работающий на поперечном электрооптическом эффекте. Емкость затвора составляла 6 пФ, рабочее четвертьволновое напряжение 3,6 кВ. Излучатель представлял собой одномодовый лазер на основе кристалла *Nd3+:YVO4* с продольной импульсной диодной накачкой. Импульсы накачки с энергией до 7 мДж имели длительность 90 нс и частоту повторения до 2,5 кГц.

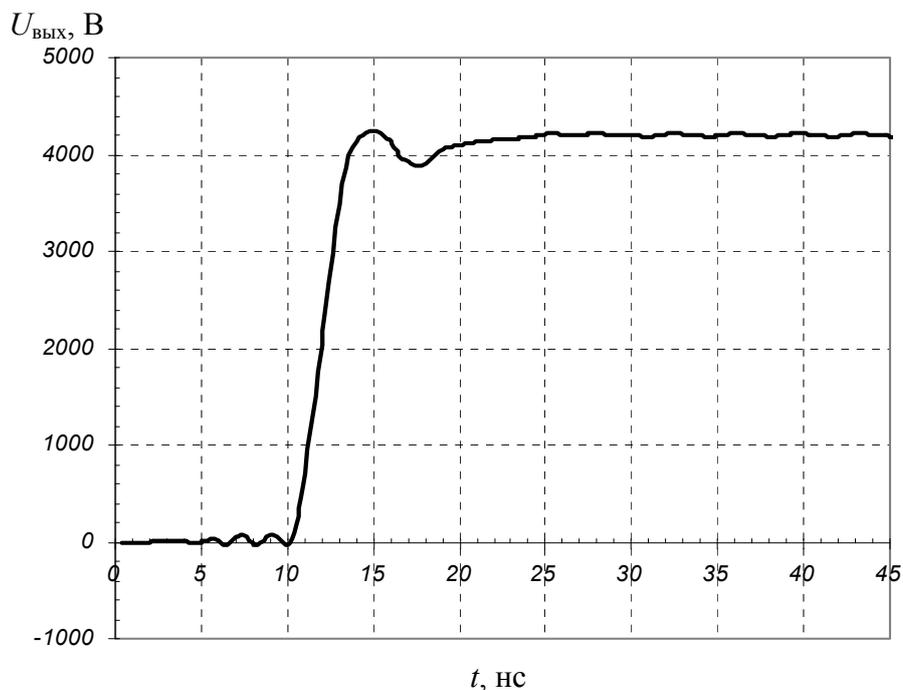


Рис. 6. Оциллограмма импульса напряжения, формируемого высоковольтным импульсным модулятором

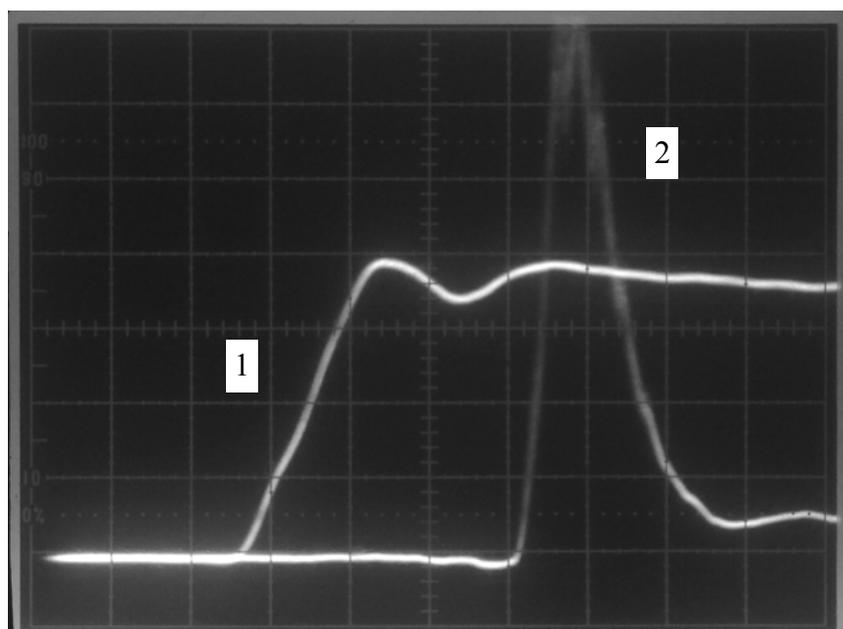


Рис. 7. Оциллограмма импульсов напряжения (1) на затворе и лазерного излучения (2). Масштаб: напряжение на затворе – 1кВ на деление, время – 5 нс на деление

По окончании импульса накачки с импульсного модулятора на затвор подавался открывающий импульс длительностью до 200 нс. Время развития генерации находилось в пределах 30–60 нс. В результате на выходе излучателя генерировались световые импульсы на длине волны 1064 нм с энергией до 1 мДж. Отношение энергетической эффективности работы лазера в режиме модуляции добротности к соответствующему значению эффективности в режиме свободной генерации составило величину 76 %. Длительность импульса при максимальной энергии накачки составила 3,8 нс (рис. 7).

Нестабильность энергии от импульса к импульсу не превышала 2,5 %. Девиация изменений средней мощности излучения при частоте повторения 2 кГц по измерению в течение 16 часов непрерывной работы составила  $\pm 1,6\%$ , что практически совпало со значением девиации в режиме свободной генерации. В результате проведения ресурсных испытаний не выявлено каких-либо изменений в работоспособности модулятора в течение  $10^9$  импульсов.

#### Основные технические характеристики модулятора

– амплитуда выходного импульса .....	2–6 кВ
– стабильность амплитуды .....	5 %
– максимальный ток в импульсе .....	30 А
– максимальная емкость нагрузки .....	100 пФ
– длительность фронта .....	3–8 нс
– длительность плоской части импульса .....	0.2–2 мкс
– время спада .....	0.5 мкс
– максимальная частота повторения импульсов .....	10 кГц
– амплитуда входного синхроимпульса .....	3–15 В
– напряжение источника питания платы .....	5–30 В
– потребляемая мощность .....	2 Вт
– рабочий температурный диапазон .....	(–40)–(+60)°С
– габаритные размеры.....	50×80×24 мм <sup>3</sup>

#### Заключение

1. Показано, что основным фактором, ограничивающим быстродействие МОП-транзисторов, является индуктивность в цепи истока транзистора.

2. Сформулирован режим сверхбыстрого включения МОП-транзистора, позволяющий обойти существующее ограничение по скорости переключения транзистора за счет предварительной накачки тока в индуктивности цепи истока.

3. На основе режима сверхбыстрого включения разработана схема и конструкция высоковольтного модулятора с наносекундным фронтом для управления электрооптическим затвором в составе твердотельных лазеров.

4. Разработанный модулятор используется на фирме *VITRO LASER* (Германия) в составе технологического комплекса для 3D-гравировки в прозрачных диэлектриках и в Институте общей физики академии наук (ИОФАН) (г. Москва.).

#### Литература

1. Аристов Ю.В., Воронков В.Б., Грехов И.В., Козлов А.К., Коротков С.В. Мощный полупроводниковый переключатель высоковольтных импульсов с наносекундным фронтом нарастания // ПТЭ. – 2007. – № 2. – С. 87–90.
2. Грехов И.В., Ефанов В.М., Кардо-Сысоев А.Ф., Шендерей С.В. Формирование киловольтных наносекундных перепадов напряжения дрейфовыми диодами с резким восстановлением // ПТЭ. – 1984. – № 5. – С. 103–105.

3. Грехов И.В., Ефанов В.М., Кардо-Сысоев А.Ф., Шендерей С.В. Мощный полупроводниковый генератор наносекундных импульсов // ПТЭ. – 1986. – № 1. – С. 93–94.
4. Behlke Power Electronics (GmbH). Fast high voltage solid-state switches. – Режим доступа: <http://www.behlke.de>, своб.
5. Clemente S., Pelly B.R., Insidori L. Силовые полупроводниковые приборы. – Воронеж, 1995. – С. 195–215.
6. Тогатов В.В., Гнатюк П.А., Терновский Д.С. Высоковольтный импульсный модулятор с наносекундным фронтом // ПТЭ. – 2007. – № 6. – С. 134–135.

*Терновский Дмитрий Сергеевич*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, dm-ternovsky@mail.ru

*Тогатов Вячеслав Вячеславович*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, dm-ternovsky@mail.ru

**УДК 536.6**

## **ПАРАМЕТРИЧЕСКАЯ ИДЕНТИФИКАЦИЯ ТЕПЛОвого ПОТОКА, ВХОДЯЩЕГО В ОДНОМЕРНЫЙ ТЕПЛОМЕР, С УТОЧНЕНИЕМ КОЭФФИЦИЕНТА ТЕПЛОПРОВОДНОСТИ**

**К.В. Кириллов, Н.В. Пилипенко**

Исследованы возможности применения метода параметрической идентификации для одновременного восстановления теплового потока и уточнения теплопроводности материала тепломера. Представлено описание математических моделей и результаты тестирования разработанной по ним программы.

**Ключевые слова:** приемники теплового потока, дифференциально-разностные модели, обратные задачи теплопроводности, параметрическая идентификация, фильтр Калмана.

### **Введение**

Теплофизические характеристики (ТФХ) материала тепломера (в дальнейшем для краткости ПТП – приемника теплового потока) существенно влияют на результаты восстановления тепловых потоков. В нестационарной теплотометрии, как правило, для получения информации о значениях ТФХ используются известные справочные данные, составление которых является отдельной задачей экспериментальной теплофизики. Однако значения ТФХ существенно различаются в зависимости от технологии получения сплавов, а для неоднородных ПТП величины ТФХ известны лишь приблизительно.

Для преодоления указанных проблем вместо традиционно применяемой прямой градуировки ПТП на специальных стендах предлагается использовать методы комбинированных обратных задач теплопроводности (ОЗТ) [1, 2]. Комбинированная ОЗТ предполагает постановку коэффициентной ОЗТ по определению ТФХ материала ПТП и объединение ее с граничной ОЗТ по восстановлению входящего в ПТП теплового потока.

При постановке коэффициентной ОЗТ рассматривается случай оптимального оценивания только коэффициента теплопроводности, где в качестве обязательной параметризации применяется его кусочно-постоянная аппроксимация, а при постановке граничной ОЗТ – кусочно-линейная сплайн-аппроксимация теплового потока.

В работе описан метод решения комбинированной ОЗТ с использованием фильтра Калмана (ФК) по искомым параметрам, приведены результаты восстановления тепловых потоков и уточнения ТФХ для тепломеров из различных материалов, оценены погрешности решения ОЗТ и даны рекомендации по использованию предлагаемого метода.

## Математические модели теплопереноса и измерений

В качестве математической модели для описания одномерного теплопереноса в ПТП различных типов применяются дифференциально-разностные модели (ДРМ), подробно описанные в работах [1, 2], которые в векторно-матричной форме для линейных стационарных систем обыкновенных дифференциальных уравнений (СОДУ) имеют вид

$$\frac{d}{d\tau} \vec{T}(\tau) = F \cdot \vec{T}(\tau) + G \cdot \vec{U}(\tau), \quad (1)$$

где  $\vec{T}(\tau)$  и  $\vec{U}(\tau)$  – векторы состояния и управления;  $F$  и  $G$  – матрицы обратных связей и управления. Общее решение СОДУ (1) имеет следующий вид:

$$\vec{T}(\tau) = \Phi(\tau, \tau_0) \cdot \vec{T}(\tau_0) + \int_{\tau_0}^{\tau} \Phi(\tau, \theta) \cdot G(\theta) \cdot \vec{U}(\theta) d\theta, \quad (2)$$

где  $\Phi(\tau, \tau_0) = \exp(F(\tau - \tau_0))$  – переходная матрица состояния (матрица Коши) системы;  $\tau_0$  – начальный момент времени. Для программной реализации решения (2) вводится дискретное время  $\tau_k = k\Delta\tau$ , а также дискретные векторы  $\vec{T}_k = \vec{T}(\tau_k)$  и  $\vec{U}_k = \vec{U}(\tau_k)$ . Тогда дискретная переходная матрица  $\Phi = \Phi_{k+1,k} = \Phi(\tau_{k+1}, \tau_k)$  может быть вычислена с требуемой точностью путем суммирования необходимого числа членов следующего бесконечного ряда:

$$\Phi = I + F\Delta\tau + \frac{1}{2!} F^2 \Delta\tau^2 + \dots + \frac{1}{m!} F^m \Delta\tau^m + \dots, \quad (3)$$

где  $I$  – единичная матрица. Решением прямой задачи теплопроводности (ПЗТ) в этом случае является последовательное применение для каждого момента времени следующей известной формулы расчета  $\vec{T}_{k+1}$  по значениям  $\Phi$  и  $\vec{T}_k$ :

$$\vec{T}_{k+1} = \Phi \cdot \vec{T}_k + \frac{1}{2} (I + \Phi) \cdot G \cdot \vec{U}_k \cdot \Delta\tau. \quad (4)$$

Нелинейному теплопереносу в ПТП соответствует нелинейная СОДУ, которую можно записать в следующей векторно-матричной форме:

$$\frac{d}{dt} \vec{T}(\tau) = \vec{f}(\vec{T}(\tau), \vec{U}(\tau)), \quad (5)$$

где  $\vec{f}(\vec{T}(\tau), \vec{U}(\tau))$  – нелинейная вектор-функция, составляющими которого являются правые части уравнений теплового баланса относительно производной температуры по времени. Для решения прямых задач динамики нелинейного теплопереноса в ПТП на основе их ДРМ предлагается использовать известный метод рекуррентной (последовательной) линеаризации правой части уравнения (5).

Метод рекуррентной линеаризации заключается в том, что при вычислении вектора состояния  $\vec{T}_{k+1}$  для  $k+1$  момента времени по формуле (4) выполняется линеаризация правой части уравнения (5) к виду (1). Для этого значения ТФХ материала в уравнениях теплового баланса необходимо отнести к уже рассчитанным температурам  $t_{ik}$  каждого блока. Таким образом, при определении  $\vec{T}_{k+1}$  по формуле (4) переходная матрица  $\Phi = \Phi_{k+1,k}$  вычисляется по формуле (3) через линеаризованную указанным способом матрицу обратной связи  $F = F_{k+1}$ .

Для учета измерительной схемы ПТП и сведений о характере и величинах случайных погрешностей в измерениях температуры используется следующая модель измерений:

$$\bar{Y}_k = H \cdot \bar{T}_k + \bar{\varepsilon}_k, \quad (6)$$

где  $\bar{Y}_k$  и  $\bar{\varepsilon}_k$  – векторы измерений и случайных погрешностей,  $H$  – матрица измерений.

### Параметрическая идентификация

В работах [1, 2] показана целесообразность использования метода параметрической идентификации для решения ОЗТ, так как последний удовлетворяет общепринятым требованиям устойчивости и сходимости вычислительных процедур, точности конечных результатов, универсальности, простоты программной реализации и др. Сущность же метода сводится к предварительной параметризации задачи и последующему нахождению оптимальной несмещенной оценки вектора искомых параметров системы, дающей минимум нормы вектора невязки между измеренными в опыте температурами и прогнозами измерений температуры, рассчитанными по модели. Для получения оценок используется рекуррентная вычислительная процедура цифрового ФК по искомым параметрам.

Под параметризацией ОЗТ понимается априорная функциональная аппроксимация подлежащего восстановлению теплового потока и коэффициента теплопроводности на всем интервале измерений.

При постановке граничной ОЗТ используется кусочно-линейная аппроксимация, где в качестве системы базисных функций применяются  $B$ -сплайны 1-го порядка. Тогда на  $z$ -ом участке аппроксимации значение теплового потока находится по следующей формуле:

$$q_z = q_{a,z} \cdot Sp_{z-1}^{(1)} + q_{b,z} \cdot Sp_z^{(1)},$$

где  $q_{a,z}$  и  $q_{b,z}$  – значения теплового потока на левой и правой границах участка соответственно;  $Sp_{z-1}^{(1)}$  и  $Sp_z^{(1)}$  –  $B$ -сплайны.

Величины  $q_{a,z}$  и  $q_{b,z}$  на каждом  $z$ -ом участке объединяются в вектор искомых параметров  $\bar{Q}_{qz} = [q_{a,z} \quad q_{b,z}]^T$ . Благодаря сквозной  $B$ -сплайн-аппроксимации на всем интервале измерения на границах сплайнов имеют место равенства  $q_{a,z} = q_{b,z-1}$ ;  $q_{a,z+1} = q_{b,z}$ ; ... Вычисления по мере поступления измерений перемещаются последовательно по всем сплайнам и восстанавливают весь набор искомых значений теплового потока.

При постановке коэффициентной ОЗТ ограничимся случаем оптимального оценивания только коэффициента теплопроводности  $\lambda(T)$  материала ПТП, а в качестве обязательной параметризации  $\lambda(T)$  воспользуемся его кусочно-постоянной аппроксимацией  $\lambda_z = const$  на каждом из  $z$  участков. Такая параметризация незначительно повышает сложность решения комбинированной ОЗТ и при этом, как показали наши исследования, существенно снижает уровень погрешности результатов от ошибок в  $\lambda$ . Таким образом, вектор искомых параметров величины  $\lambda$  имеет вид  $\bar{Q}_{\lambda z} = \lambda_z$ , а само  $\lambda_z$  может быть отнесено к средней на  $z$ -м интервале температуре ПТП  $T_{z,\Sigma}$ .

Заметим, что принципиально возможны другие более точные параметры параметризации  $\lambda(T)$ , например, кусочно-линейный, когда  $\bar{Q}_{\lambda z} = [\lambda_{a,z} \quad \lambda_{b,z}]^T$ .

Таким образом, комбинированная ОЗТ заключается в определении на каждом  $z$ -м участке оптимальных обобщенных оценок вектора искомых параметров

$$\bar{Q}_z = [\bar{Q}_{qz} \quad \bar{Q}_{\lambda z}]^T = [q_{a,z} \quad q_{b,z} \quad \lambda_z]^T. \quad (7)$$

## Фильтр Калмана по искомым параметрам

Нелинейный дискретный ФК по искомым параметрам основан на зависимости модельного вектора измерений  $\bar{Y}(\bar{Q})$  от вектора искомых параметров  $\bar{Q}$  (7), что формально позволяет записать:

$$\bar{Y}(\bar{Q}) = [\bar{y}_{j,k}(\bar{Q})]_{j=1}^m.$$

Составляющие  $\bar{y}_{j,k}(\bar{Q})$  – это модельные температуры или их перепады в  $m$  точках ПТП, соответствующие вектору измерений  $\bar{Y}_k$  в  $k$ -й момент времени. В общем случае – для нелинейной модели (5) ПТП – они являются нелинейными функциями от  $\bar{Q}$ .

Тогда модель ПТП в искомых параметрах  $\bar{Q}$ , учитывая условие  $\dot{\bar{Q}} = const$ , может быть записана в виде

$$\dot{\bar{Q}} = 0, \tag{8}$$

а модель измерений в ПТП – в виде

$$\bar{Y}_k = \bar{Y}_k(\bar{Q}_0) + \bar{\varepsilon}_k. \tag{9}$$

Уравнение (9) обозначает нелинейную зависимость между  $\bar{Y}_k$  и  $\bar{Q}_0$ , где  $\bar{Q}_0$  – истинное значение вектора искомых параметров.

К модели (8)–(9) может быть применен алгоритм дискретного нелинейного ФК, позволяющий получать рекуррентные оценки  $\bar{Q}_{k+1}$  вектора искомых параметров  $\bar{Q}$  и ковариационную матрицу  $P_{k+1}$  их ошибок по найденным на предыдущем  $k$ -ом шаге  $\bar{Q}_k$ ,  $P_k$  и известному вектору измерений  $\bar{Y}_{k+1}$ . Алгоритм имеет следующий вид:

$$K_{k+1} = P_k \cdot \bar{H}_{k+1}^T \cdot (\bar{H}_{k+1} P_k \bar{H}_{k+1}^T + N)^{-1}; \tag{10}$$

$$\bar{\mathcal{Q}}_{k+1} = \bar{\mathcal{Q}}_k + K_{k+1} \cdot \left( \bar{Y}_{k+1} - \bar{Y}_{k+1}(\bar{\mathcal{Q}}_k) \right); \tag{11}$$

$$P_{k+1} = P_k - K_{k+1} \bar{H}_{k+1} P_k, \tag{12}$$

где  $K$  – весовая матрица;  $P$  – ковариационная матрица ошибок оценок;  $\bar{H}_{k+1}$  – матрица функций чувствительности;  $N$  – ковариационная матрица случайных погрешностей измерений;  $\bar{Y}_{k+1}(\bar{\mathcal{Q}}_k)$  – модельный вектор измерения, рассчитываемый по модели теплопереноса в ПТП для момента времени  $k+1$  с использованием предыдущей оценки  $\bar{\mathcal{Q}}_k$  вектора  $\bar{Q}$ .

Матрица функций чувствительности  $\bar{H}_{k+1}$  имеет следующий вид:

$$\bar{H}_{k+1} = \frac{\partial \bar{Y}_k}{\partial \bar{Q}} \Big|_{\bar{Q} = \bar{\mathcal{Q}}_k} = \begin{bmatrix} \frac{\partial y_{1,k}(\bar{Q})}{\partial q_a} & \frac{\partial y_{1,k}(\bar{Q})}{\partial q_b} & \frac{\partial y_{1,k}(\bar{Q})}{\partial \lambda} \\ \vdots & \vdots & \vdots \\ \frac{\partial y_{m,k}(\bar{Q})}{\partial q_a} & \frac{\partial y_{m,k}(\bar{Q})}{\partial q_b} & \frac{\partial y_{m,k}(\bar{Q})}{\partial \lambda} \end{bmatrix}.$$

Здесь  $\left. \frac{\partial y_{j,k}(\bar{Q})}{\partial q_a} \right|_{\bar{Q} = \bar{Q}_k}$ ,  $\left. \frac{\partial y_{j,k}(\bar{Q})}{\partial q_b} \right|_{\bar{Q} = \bar{Q}_k}$  и  $\left. \frac{\partial y_{j,k}(\bar{Q})}{\partial \lambda} \right|_{\bar{Q} = \bar{Q}_k}$  – функции чувстви-

тельности  $j$ -го измерения к искомому параметру  $q_a$ ,  $q_b$  и  $\lambda$  в  $k+1$  момент времени.

### Описание программы Kalman Filter

Рекуррентная последовательная процедура ФК по искомым параметрам (10)-(12) была реализована авторами в форме программы Kalman Filter на языке высокого уровня Scilab, который предназначен для выполнения инженерных и научных вычислений.

Основными преимуществами пакета Scilab являются поддержка большого числа математических функций, широкие возможности редактирования графиков, наличие инструментов для создания визуальных приложений, бесплатность (лицензия Scilab позволяет свободное распространение). Версию пакета для операционных систем Windows, Linux и ряда других можно получить по адресу <http://www.scilab.org>.

Главное окно разработанной программы представлено на рис. 1.

Sensor properties			Thermal conductivity		
Total height, m :	0.004		Number of splines :	2	
Number of blocks :	11		Temperature, C		Value, W/m*K
Thermal capacity, J/kg*K :	840		0		0.76
Density, kg/m3 :	2500		50		0.76
Time and initial conditions			0		0
Total time, s :	3		0		0
Time interval, s :	0.01		0		0
Initial temperature, C :	0		0		0
Boundary condition from top			0		0
<input type="checkbox"/> Heat-transfer c., W/m2*K :	0		0		0
<input type="checkbox"/> Linear	a :	10000	0		0
<input checked="" type="checkbox"/> Harmonic	b :	10000			
	w :	10	Measurements		
Boundary condition from bottom			Number of measurements : 2		
<input type="checkbox"/> Heat-transfer c., W/m2*K :	0		Measurement matrix		
<input checked="" type="checkbox"/> Linear	a :	0	[1,0,0,0,0,0,0,0,0,0,0]		
<input type="checkbox"/> Harmonic	b :	0	[0,1,0,0,0,0,0,0,0,0,0]		
	w :	0	[0,0,0,1,0,0,0,0,0,0,0]		
Inverse problem			[0,0,0,0,1,0,0,0,0,0,0]		
Spline Length :	10		[0,0,0,0,0,1,0,0,0,0,0]		
Number of expansion terms :	10		[0,0,0,0,0,0,1,0,0,0,0]		
Initial estimates	Increments, %	Initial covariances	[0,0,0,0,0,0,0,1,0,0,0]		
5000	1	1e+12	[0,0,0,0,0,0,0,0,1,0,0]		
5000	1	1e+12	[0,0,0,0,0,0,0,0,0,1,0]		
0.38	1	10000	[0,0,0,0,0,0,0,0,0,0,1]		
Solve			Close		

Рис. 1. Ввод исходных данных в главном окне программы

### Результаты исследований

В качестве иллюстрации для градиентного однородного ПТП типа вспомогательной стенки приведем результаты математического моделирования метода параметрической идентификации. Рассмотрим различные ТФХ материала ПТП, представленные в следующей таблице.

Материал	Теплопроводность $\lambda$ , Вт/(м·К)	Теплоемкость $c$ , Дж/(кг·К)	Плотность $\rho$ , кг/м <sup>3</sup>	Толщина $h$ , м
Стекло (ПТП-1)	0,76	840	2500	5,0·10 <sup>-3</sup>
Германий (ПТП-2)	15	310	5300	
Алюминий (ПТП-3)	210	900	2700	

Таблица. Физические свойства и геометрические параметры ПТП

Для всех рассмотренных ниже случаев моделирования можно выделить следующие общие положения:

- начальное распределение температур по толщине образца принималось известным;
- число блоков разбиения ПТП по толщине  $n=11$ ;
- длина участка сплайн-аппроксимации  $\Delta_z = 10 \cdot \Delta\tau$ ;
- измерялись температуры поверхности  $t_1$  и второго блока  $t_2$ ;
- выбирался гармонический или линейный закон изменения теплового потока;
- начальные оценки вектора искомым параметров  $\xi_0$  задавались в два раза меньше эталонных;
- начальное значение ковариационной матрицы ошибок оценок  $P_0$  задавалось в виде

$$P_0 = \begin{bmatrix} 10^{11} & 0 & 0 \\ 0 & 10^{11} & 0 \\ 0 & 0 & 10^5 \end{bmatrix}.$$

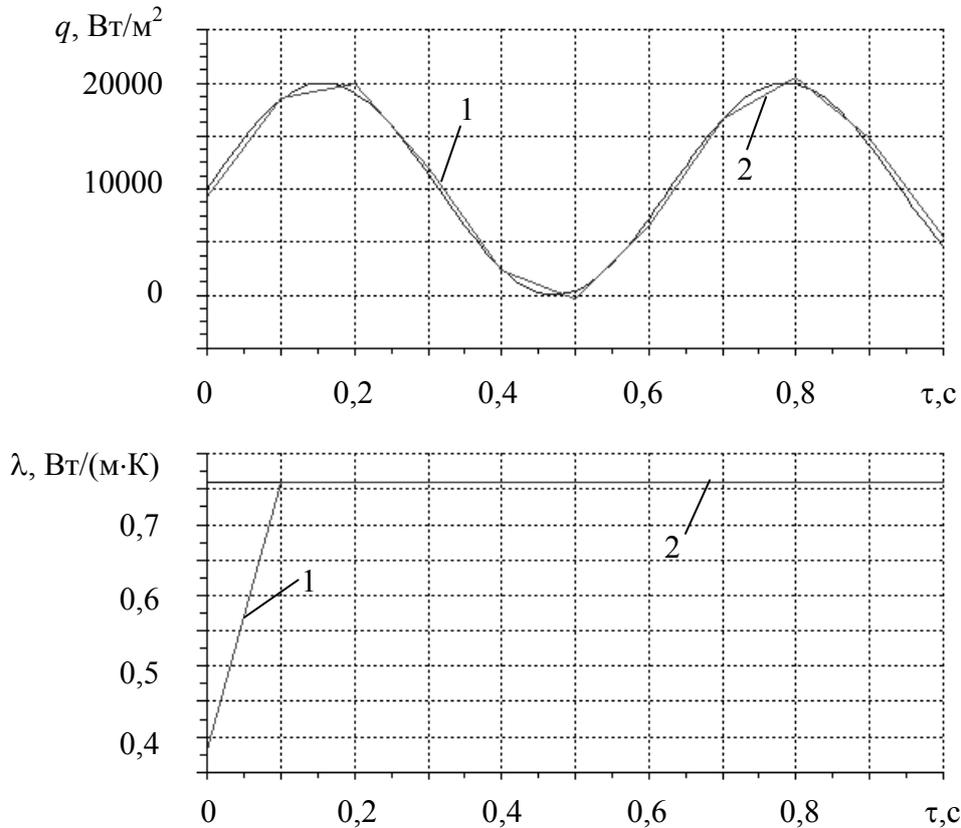


Рис. 2. Эталонные (1) и восстановленные (2) значения теплового потока и коэффициента теплопроводности для ПТП-1

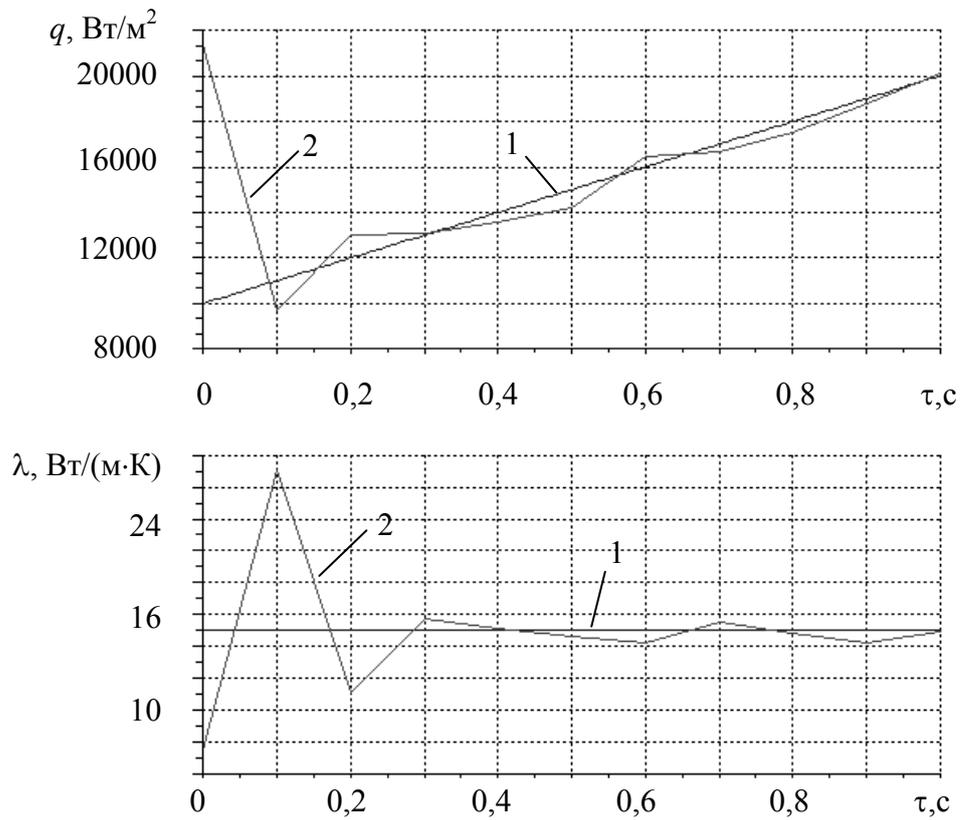


Рис. 3. Эталонные (1) и восстановленные (2) значения теплового потока и коэффициента теплопроводности для ПТП-2

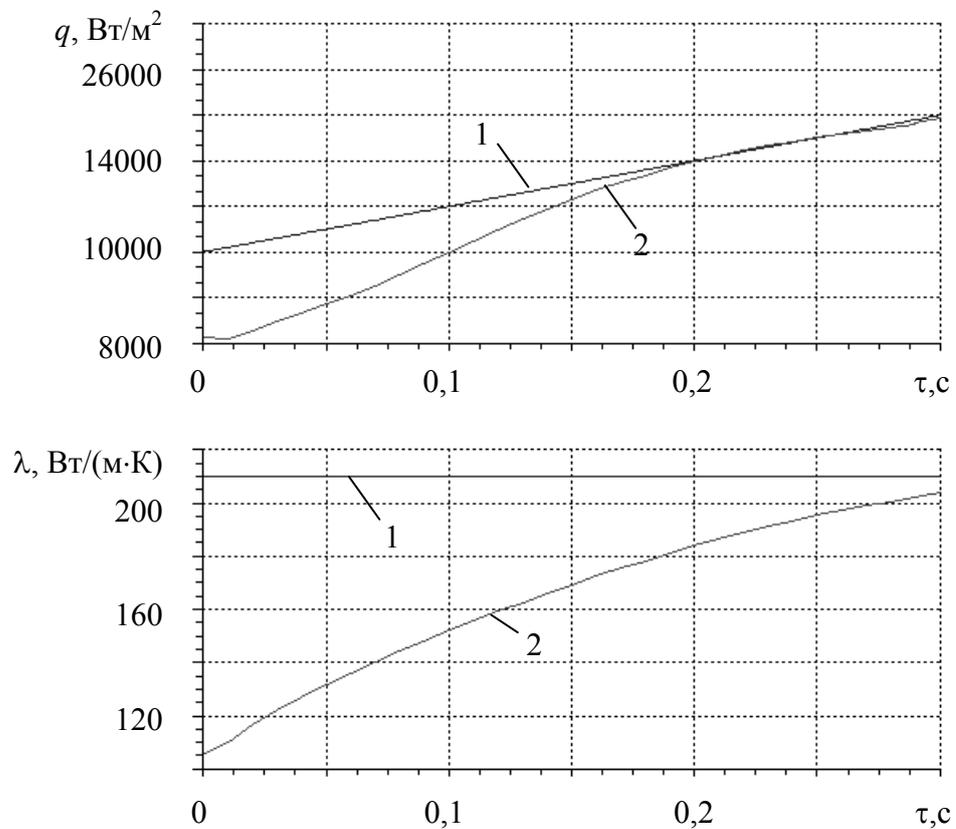


Рис. 4. Эталонные (1) и восстановленные (2) значения теплового потока и коэффициента теплопроводности для ПТП-3

Результаты параметрической идентификации теплового потока и коэффициента теплопроводности для ПТП-1 при  $\Delta\tau=0,01$  с представлены на рис. 2. Результаты расчетов для ПТП-2 при  $\Delta\tau=0,01$  с и уровне шумов в измерениях  $\sigma=0,05$  °С представлены на рис. 3. Результаты для ПТП-3 при  $\Delta\tau=0,001$  с представлены на рис. 4.

Как видно из рис. 2–4, во всех случаях наблюдалась устойчивая сходимость процедуры идентификации: в результате тепловые потоки и коэффициент теплопроводности хорошо уточняются уже на первом участке сплайн-аппроксимации (рис. 2), либо на третьем (рис. 3), либо на 30-м (рис. 4).

### Заключение

В работе описаны математические модели (ММ) линейного и нелинейного теплопереноса в ПТП на основе ДРМ. Предложен метод рекуррентной последовательной линеаризации для решения ПЗТ на основе таких ММ.

Составлена стратегия оценивания для одновременного восстановления теплового потока и теплопроводности на основе метода параметрической идентификации. Адаптирован и реализован в форме программы на языке Scilab нелинейный ФК по искомым параметрам для получения вышеперечисленных оценок.

Математическим моделированием доказана эффективность разработанных алгоритмов параметрической идентификации и показана их практическая применимость при экспериментально-расчетном определении тепловых потоков и теплопроводности.

### Литература

1. Пилипенко Н.В. Методы параметрической идентификации в нестационарной теплометрии (ч. 1) //Изв. вузов. Приборостроение. – 2003. – № 8. – С. 50–54.
2. Пилипенко Н.В. Методы параметрической идентификации в нестационарной теплометрии (ч. 2) //Изв. вузов. Приборостроение. – 2003. – № 10. – С. 67–71.

*Кириллов Кирилл Валерьевич*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, kirill.kirillov@gmail.com

*Пилипенко Николай Васильевич*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, pilipenko@grv.ifmo.ru

УДК 004.4'22

## СИСТЕМА КОДОГЕНЕРАЦИИ THORNADO И ЕЕ ИСПОЛЬЗОВАНИЕ ДЛЯ СОЗДАНИЯ БИЗНЕС-ПРИЛОЖЕНИЙ Д.В. Деев, Ю.С. Окуловский, В.Ю. Попов, В.П. Часовских

Описана система порождающего программирования Thornado, основанная на использовании языков разметки, скриптовых языков и объектно-ориентированного программирования. Приведен пример использования этой системы для создания бизнес-приложений. Исследованы вопросы организации разработки программного обеспечения с применением этой системы.

**Ключевые слова:** разработка программного обеспечения, порождающее программирование, кодогенерация.

### Введение

В теории информации традиционно выделяются два основных объекта – поток данных и поток команд. Исторически считалось, что поток команд является основным,

а поток данных – вспомогательным. С развитием технологий программирования взаимодействие между данными и командами становилось сложнее. Кроме того, появился новый объект – поток спецификации, описывающий команды и данные. Согласно современному взгляду на программирование, этот поток является основным.

Широко распространены три подхода к взаимодействию программы и спецификации: порождение спецификации для существующей программы; проверка соответствия программы спецификации (верификация, [1]); порождение программы по спецификации (порождающее программирование [2]). Третий подход является на сегодняшний день наиболее обсуждаемым и изучаемым.

В случае порождающего программирования система принимает спецификацию на некотором формальном языке и генерирует исходный код приложения, заранее удовлетворяющий спецификации. На сегодняшний день существует множество систем порождающего программирования. Например, это Windows Forms Designer [3], входящий в стандартную поставку Microsoft Visual Studio, который позволяет порождать исходный код графического интерфейса пользователя по визуальному описанию. Более сложным примером является IBM RAD [4], способный сгенерировать исходный код бизнес-приложения. Кроме того, существует множество небольших кодогенераторов, разрабатываемых внутри предприятий для отдельных проектов. Все эти решения обычно имеют некоторые общие недостатки. Так, кодогенераторы обычно созданы для генерирования конкретного типа кода. Вследствие этого идеи, применяемые для генерирования бизнес-приложений, не могут быть непосредственно перенесены на генерирование, например, синтаксических анализаторов. Архитектура генераторов часто не предполагает их расширения, вследствие чего генератор не может быть скорректирован для более полного соответствия разрабатываемому проекту. Далее, спецификация, используемая для генерирования кода, не может быть интерпретирована вне генератора. Из-за этого проблема документирования кода не решена, и вместе с формальной спецификацией для генератора приходится писать также спецификацию на естественном языке. Наконец, часто сгенерированный код не может быть исправлен вручную, так как изменения будут либо потеряны после повторной генерации, либо сделают генератор неработоспособным. Для больших и сложных систем возникает также проблема большой цены использования, связанная с высокой ценой продукта и высокими затратами на дополнительное обучение сотрудников.

В данной статье описывается разрабатываемая система Thornado. Эта система предполагается свободной от приведенных недостатков. Система основывается на хорошо известных технологиях языков разметки, скриптовых языков и объектно-ориентированных технологий. Поэтому работа с системой не требует существенного дополнительного обучения программистов. Система Thornado является расширяемой и может быть приспособлена для нужд конкретного проекта. Кроме того, разработана технология расширения функциональности сгенерированного кода, которая не конфликтует с кодогенератором. С помощью Thornado возможно генерирование кода приложения любого типа и на любом языке программирования. Наконец, документы, используемые системой для генерации кода, могут быть преобразованы в спецификацию кода на естественном языке. В первой части приводятся общие принципы работы этой системы. Во второй части описывается применение системы Thornado для генерирования бизнес-приложений.

### **Общие принципы**

Система кодогенерации Thornado строится на использовании трех технологий программирования: языков разметки, скриптовых языков и объектно-ориентированного программирования. На языке разметки происходит составление спецификации про-

граммного кода, который должен быть сгенерирован. Затем при помощи скриптовых языков эта спецификация транслируется в исходный код программы. Использование принципов объектно-ориентированного программирования позволяет расширять функциональность сгенерированного кода и встраивать его в приложения.

### Язык Thorn

В качестве специального языка разметки для порождающего программирования использован язык Thorn (ранее TH, [5, 6]). Документ Thorn состоит из команд, вложенных одна в другую. Общее представление о синтаксисе языка Thorn дает следующий пример.

```
\table[border=1 width=100%] {  
  \tr{ \td{a11} \td{a12} }  
  \tr{ \td{a21} \td{a22} }  
}
```

Приведенный документ Thorn в некотором смысле эквивалентен следующему документу HTML:

```
<table border="1" width="100%">  
  <td> a11 </td> <td> a12 </td> </tr>  
  <td> a21 </td> <td> a22 </td> </tr>  
</table>
```

В отличие от HTML, синтаксис Thorn может быть существенно сокращен при помощи нескольких правил. Для каждой команды определена последовательность параметров (параметрами являются border и width в примере), и, если значения параметров следуют этой последовательности, то имена параметров могут быть опущены. Кроме того, для команды можно определить естественное вложение: например, для команды tr естественно находиться внутри команды table, а для команды td – внутри tr. Если команды вложены в естественном порядке, то фигурные скобки также могут быть опущены. Используя эти возможности сокращения, документ из примера может быть представлен в следующем виде.

```
\table[1 100%]  
\tr \td a11 \td a12  
\tr \td a21 \td a22
```

Основное различие между языком Thorn и другими распространенными языками разметки обеспечивается компилятором. Каждая команда ассоциируется с некоторой программой, которая может быть написана на любом языке программирования (как правило, используется Perl). При разборе текста компилятор собирает все текстовые данные, связанные с командой. После этого он запускает ассоциированную программу, передавая ей текстовые данные как входные параметры. Программа возвращает строку, которая подставляется на место команды. После того, как все команды выполнены, трансляция документа завершается. Программы могут обмениваться между собой данными через глобальные переменные.

Программы, ассоциированные с командами, сохраняются в библиотеках, которые отделены от документа Thorn. Один и тот же документ Thorn может быть транслирован разными библиотеками, т.е. с различным значением команд. Например, различные библиотеки могут производить код для разных языков программирования. Кроме того, может быть создана специальная библиотека, транслирующая документ Thorn в спецификацию на естественном языке.

Также Thorn является расширяемым языком за счет независимости библиотек. Допустим, мы используем библиотеку А для генерирования кода, и команды из А работают с некоторыми глобальными переменными. Предположим, что требуется породить новый код, который невозможен для библиотеки А. Мы можем написать библиотеку В,

которая использует глобальные переменные, определяемые командами из А. Таким образом, нет необходимости переписывать эти переменные. Кроме того, библиотека А может по-прежнему работать без библиотеки В. Подобная идеология позволяет создавать «диалекты» имеющихся больших библиотек для каждого проекта.

Язык Thorn с набором библиотек можно рассматривать как domain-specific language (DSL) [7]. Фактически, мы создаем DSL для наиболее распространенных типов кода – пользовательских интерфейсов, сохранения данных в файлах и базах данных и т.д. В нашем случае все DSL основаны на одной и той же технологии, что позволяет объединять данные из документов, написанных на различных DSL.

### Скрипты для создания кода

В этом параграфе мы опишем преобразование документа Thorn в программный код. Рассмотрим следующий документ Thorn.

```
\property[int Count] {Count of elements}
Предположим, он должен быть трансформирован в следующий программный код:
///

```

В этом случае команда `property` принимает три параметра: тип свойства, его имя и описание. Скрипт, который преобразует эти данные в выходную строку, очевиден. Однако решение, когда каждая команда создает часть программного кода, не всегда удобно. К примеру, если несколько команд используют один и тот же параметр, он должен быть написан в каждой из команд. Эта проблема решается с помощью глобальных переменных. Другая проблема состоит в том, что при большом количестве параметров команды трудно запомнить их наизусть, и это замедляет набор документа с клавиатуры. В этом случае команду следует разложить в набор других команд.

Мы видим, что необходим выбор оптимальной архитектуры команд. По нашим наблюдениям, описание с помощью многомерных отношений наиболее подходит для порождающего программирования. Отношение в данном случае есть набор записей; в каждой записи определены некоторые поля. Отношение может задаваться матрицей, в которой строки соответствуют записям, а столбцы – полям. В большинстве случаев генератор, транслирующий описание отношения в программный код, представим в виде следующей функции:

$$p(A) = h \begin{pmatrix} g_1(f_1(A_1), f_1(A_2), \dots, f_1(A_m)) \\ g_2(f_2(A_1), f_2(A_2), \dots, f_2(A_m)) \\ \dots \\ g_n(f_n(A_1), f_n(A_2), \dots, f_n(A_m)) \end{pmatrix}.$$

Здесь  $h$  производит программный код. При генерации объектно-ориентированных программ  $h$  производит один класс. Функции  $g_1, \dots, g_n$  производят части кода, обычно методы внутри класса. Функции  $f_1, \dots, f_n$  производят части метода, соответствующие одной записи отношения.  $A_1, \dots, A_m$  являются строками матрицы отношения. Часто  $g_1, \dots, g_n$  являются функциями конкатенации, т.е.  $g(x_1, \dots, x_m) = x_1 \cdot \dots \cdot x_m$ , где точка обозначает конкатенацию строк. Функция  $h$  часто может быть представлена в следующем виде:  $h(x_1, \dots, x_m) = a_0 \cdot x_1 \cdot a_1 \cdot \dots \cdot x_n \cdot a_n$ , где  $a_0, \dots, a_n$  – константы.

Генераторы, представимые таким образом, будем называть функциональными. Этот подход к генераторам реализован в специальном модуле Perl. Функции этого модуля позволяют транслировать документ Thorn в отношение, записанное в глобальных переменных, и затем определять функциональные генераторы для преобразования этого отношения в программный код. Этот модуль заметно облегчает разработку кодогенераторов.

Описанная архитектура кодогенератора часто оказывается достаточной, однако бывают случаи, когда необходима полная функциональность языков Thorn и Perl. Так, если одна запись может воздействовать на код, сгенерированный для другой записи, то функциональный генератор оказывается неспособным обработать эту ситуацию. Другой пример – описание пользовательских меню. Пользовательское меню является деревом, и естественно описывать его именно как дерево, а не отношение.

### **Расширение функциональности**

Одной из главных проблем многих существующих систем порождающего программирования является расширение функциональности сгенерированного кода. Любая корректировка кода будет потеряна после повторной генерации, поэтому для расширения функциональности требуется использовать другие методы. В объектно-ориентированном программировании это возможно за счет использования цепочек наследования.

В цепочке наследования каждый класс наследует предыдущий. Типичная структура цепочки следующая. Первый класс является шаблоном, определяющим некоторые абстрактные и виртуальные методы. Второй класс генерируется автоматически, он перегружает эти методы и заполняет их основной функциональностью. Третий класс является классом доводки. Он пишется вручную и содержит всю функциональность, которую не может обеспечить генератор. Класс доводки также перегружает виртуальные методы, но вызывает их из базового класса для использования функциональности сгенерированного кода. Благодаря этой схеме сгенерированный код и код, написанный вручную, находятся в разных файлах и не влияют друг на друга.

В более сложных случаях возможны дополнительные классы в цепочках наследования. Пусть генератор А производит код, а генератор В производит класс доводки. Эти генераторы не могут быть объединены, потому что А может выполняться и без участия В. В этом случае В генерирует третий класс в цепочке наследования, а четвертый класс в этой цепочке содержит усовершенствования, которые не может обеспечить В.

### **Разработка бизнес-приложений**

Под бизнес-приложением мы понимаем программу, которая:

- собирает данные о бизнес-процессе у сотрудников и хранит их в базе данных;
- обрабатывает данные из базы данных и производит отчеты и документы;
- выполняет расчеты для оптимизации бизнес-процесса.

Хорошо известны крупные бизнес-приложения, такие как 1С. Однако покупка и в особенности адаптация подобных приложений к конкретному бизнес-процессу могут оказаться слишком затратными. Поэтому часто выбирается путь написания бизнес-приложения под нужды конкретного предприятия. При этом бизнес-приложения имеют схожую структуру, что позволяет использовать для их написания порождающее программирование.

При разработке системы кодогенерации для создания бизнес-приложений мы руководствовались следующими принципами.

- 1) Система кодогенерации порождает не все приложение, но типовые, достаточно большие его части.

- 2) Система кодогенерации должна оставаться простой и компактной.
- 3) Разработка спецификаций для кодогенераторов не должна требовать квалификации программиста.

Первое условие связано с тем, что написание спецификации для, например, алгоритмов оптимизации бизнес-процесса требует существенной выразительной возможности языка спецификации. Это приводит к неоправданному усложнению кодогенератора, поскольку становится проще написать исходный код на существующем языке программирования, чем разработать и освоить фактически новый язык. Второе условие связано с тем, что желательно создание простого ядра кодогенератора, подходящего для типичной функциональности приложения. Для обеспечения специфической функциональности не следует усложнять базовый кодогенератор. Вместо этого лучше использовать расширение функциональности сгенерированного кода методом цепочек наследования или разработать диалект кодогенератора для конкретного программного продукта. Наконец, третье требование обусловлено тем, что составление спецификации происходит в процессе ознакомления с бизнес-процессами предприятия. Желательно, чтобы это ознакомление осуществлялось не программистом, а специалистом в области бизнес-процессов, который обычно не имеет квалификации программиста.

В качестве языка программирования был выбран язык C# [8]. Однако система кодогенерации может быть переведена на любой другой язык программирования в случае необходимости.

В соответствии с приведенными принципами были разработаны кодогенераторы для трех типичных подсистем бизнес-приложений: подсистемы ввода и хранения данных, подсистемы вывода документов и подсистемы пользовательского меню.

### **Подсистема ввода данных**

Для примера рассмотрим описание данных сотрудника некоторой фирмы. Это описание состоит из перечисления полей, для каждого поля определены некоторые свойства. Описание поля имеет следующий формат:

```
\field[<тип поля> <имя поля>]
  \desc <описание>
  \default <значение по умолчанию>
  \io <объект ввода-вывода>
  \gui <тип GUI>
```

Соответственно, структура данных определяется последовательностью описания полей:

```
\dataSet
  \field[string FIO]
    \desc Фамилия, Имя, Отчество
  \field[DateTime Birthdate]
  ...
  \field[double Salary]
  ...
  ...
```

На основании такого описания генерируются структуры данных. Так, генерируется класс Employee, поля и свойства которого соответствуют полям описания данных следующим образом.

- Тип поля и свойства класса определяются значением <тип поля>.
- Имя поля класса является значением <имя поля>, переведенным в нижний регистр.
- Имя свойства совпадает с <имя поля>.
- В конструкторе класса полю присваивается <значение по умолчанию>.
- Комментарий к полю определяется значением <описание>.

Помимо этого, дополнительные команды Thonp позволяют настроить модификаторы доступа, наличие в свойстве геттера и сеттера и т.д.

Кроме класса Employee, может быть сгенерирован класс EmployeeArray, являющийся коллекцией объектов типа Employee. Возможны различные модификации этой коллекции: заполнение ее предопределенными объектами в конструкторе, ограничение доступа к методам добавления и удаления элементов и т.д.

После генерации структур данных можно обеспечить их методами ввода и вывода в источники данных – системный реестр, файлы или SQL-базы данных. Для этого в основном используется ассоциированный с полем объект ввода-вывода, который специфицирован в примере командой \io. Этот объект обеспечивает перевод поля в текстовый вид и восстановление поля из текстового вида.

В платформе .NET каждый класс определяет метод ToString, переводящий объект в строку, а большинство встроенных типов имеют статический метод Parse, восстанавливающий объект из строки. Поэтому во многих случаях функциональность объекта ввода-вывода тривиальна. Однако введение объекта ввода-вывода может обеспечить различное представление поля для, например, файлов и базы данных. Кроме того, с помощью объектов ввода-вывода можно сохранять массивы, объекты пользовательского интерфейса и другие объекты, приведение которых в строковый тип не обеспечено платформой.

Для сохранения объекта Employee создается класс EmployeeTextIO. Его статические методы принимают объект Employee и генерируют файл вида

```
[Employee]
FIO=Иванов Иван Иванович
Birthdate=01.01.1981
```

Аналогично генерируется класс EmployeeRegistryIO, который создает ветвь системного реестра Windows, ключи которого хранят информацию об объекте. Также может быть сгенерирован класс EmployeeSqlIO. Этот класс обеспечивает получение списка EmployeeArray из базы данных, обновление строки, соответствующей некоторому объекту Employee, и т.д.

Рассмотрим генерирование пользовательского интерфейса для ввода и вывода сгенерированных структур данных. Под пользовательским интерфейсом мы понимаем форму, содержащую элементы управления, заполнение которых данными позволяет полностью описать объект структуры данных. В полной формулировке проблемы форма должна ассоциировать с полями стандартные элементы управления – текстовые поля, выпадающие списки, флажки и т.д. Эта задача не была решена полностью, хотя некоторые подходы рассмотрены в [9]. Нами была решена эта проблема в более узкой формулировке. Все интерфейсы ввода вывода имеют табличную форму. Так, может быть сгенерирована «вертикальная» таблица, позволяющая описать один объект. В этой таблице два столбца, в первом столбце перечислены описания полей, а во втором – значения. Определение <тип GUI> указывает, каким образом происходит ввод значения поля – например, вводом текста, выбором из выпадающего списка и т.д. Восстановление значения поля из строкового представления осуществляется с помощью объекта ввода-вывода.

Кроме «вертикальных», возможны также горизонтальные таблицы, где поля расположены в столбцах, а объекты – в строках. Такие таблицы позволяют редактировать массивы объектов.

Кроме простых интерфейсов пользователя, возможны также составные интерфейсы, например:

- переключающиеся вкладки, где на каждой из вкладок расположена таблица. Такой интерфейс позволяет одновременно определять несколько объектов. Это используется, например, при создании диалогового окна настроек;

– управляющий компонент списка, содержащий объекты. При выборе объекта его поля загружаются в таблицу с возможностью редактирования.

В составных интерфейсах используется интерфейс страниц данных. Страница данных – это управляющий компонент, который позволяет считывать значения полей из объекта, записывать введенные значения в структуру данных, проверять корректность введенных данных, проверять наличие изменений и т.д. Все генерируемые таблицы являются страницами данных, что позволяет объединять их в сложные интерфейсы. Более того, каждый сложный интерфейс сам является страницей данных, что позволяет, например, встроить переключающиеся вкладки в окно с компонентом списка.

Табличные интерфейсы также имеют преимущества перед обычными при создании автоматизированных рабочих мест. Пусть к данным одной и той же таблицы обращается несколько сотрудников, но одни сотрудники имеют доступ к полю Birthdate, а другие – к полю Salary. Поле FIO все могут просматривать, но не могут редактировать. Пусть EmployeeTable – класс сгенерированной таблицы для сотрудников. Из него выводятся дополнительные классы EmployeeTableForDate и EmployeeTableForSalary. В первой таблице скрывается строка, соответствующая Salary, во второй – строка, соответствующая Birthdate. Поле FIO запрещается для редактирования в обеих. Таким образом, коррекция кода с использованием цепочек наследования способна определить уровни доступа к данным.

Для объединения данных нескольких структур в одном интерфейсе используются структуры данных, которые описывают все необходимые поля. В случае SQL таблиц, такая структура загружает данные из представления SQL.

### **Подсистема вывода документов**

Подсистема вывода документов обеспечивает создание приказов, договоров, приложений, актов и других документов сопровождения бизнес-процесса. Подсистема вывода документов имеет свои особенности. Так, формат документов часто меняется в связи с изменениями законодательства. Кроме того, документы обычно имеют достаточно громоздкий вид, если они оформлены, например, в формате HTML. Соответственно, такой же вид имеет исходный код подсистемы. Наконец, верстка требует скорее навыков верстальщика, чем программиста, поэтому необходимо обеспечить возможность совместной работы как минимум двух специалистов.

Поэтому код подсистемы желательно изолировать от других исходных кодов программы. Следует также запретить его ручную правку, но обеспечить возможность быстро заменять исходный код в случае изменения формата документов.

Подсистема вывода документов формируется непосредственно из образцов документа, предоставленных заказчиком. Подобные документы обычно содержат множество недостатков в форматировании: выполненное пробелами центрирование, различные начертания и размеры шрифтов и т.д. В первую очередь требуется обеспечить документам стандартное форматирование. Для оформления документов стандартом является ГОСТ Р6.30-2003 [10].

В качестве языка форматирования документов был выбран язык HTML 4.01 [11]. Все необходимые способы форматирования могут быть выражены на этом языке. Документы HTML 4.01 также открываются большинством современных текстовых редакторов – Microsoft Word, Open Office, и др. Для упрощения ввода документов HTML 4.01 используется язык Thonp. Для обеспечения стандартного форматирования создана библиотека CSS-стилей, а также набор шаблонных команд Thonp, вставляющих распространенные штампы – например, штамп даты.

После стандартизации форматирования документа он перерабатывается в шаблон. Все участки документа, которые являются переменными, заменяются специальными последовательностями символов, например:

- последовательность `$Name$` обозначает, что на ее место будет подставлена строковая переменная `Name`;
- последовательность `!Name$<текст>$` обозначает, что на ее место будет подставлен `<текст>`, если булева переменная `Name` истинна.

Текстовая переменная может иметь сложную структуру. Пусть в документ вставляется таблица с переменным числом строк. Тогда вставляемая переменная (внутренность таблицы) должна быть конкатенацией строк таблицы, а каждая строка сама по себе является шаблоном. В этом случае создается один шаблон для документа, содержащего таблицу, и второй – для строки таблицы.

После создания шаблона создается описание всех используемых в шаблоне переменных. Это описание составляется на языке `Thorn` и определяет описания переменных на естественном языке, их допустимые значения и т.д. После этого специальная библиотека команд `Thorn` создает исходный код классов, статические методы которых производят документы. Эти классы добавляются в проект и вызываются из программы.

Такой подход позволяет распределить обязанности между всеми участниками разработки подсистемы вывода документов. Менеджер проекта анализирует документы заказчика, выделяет переменные и описывает их на языке `Thorn`. Верстальщик обеспечивает правильное форматирование документа. На основании их работы генерируются модули вывода документов, которые затем используются программистами.

### **Подсистема пользовательского меню**

Третьей важной подсистемой бизнес-приложения является подсистема пользовательского меню. Эта система дублируется в приложении в следующих пользовательских интерфейсах:

- главное меню программы;
- контекстное меню окна;
- панель инструментов, содержащая графическую репрезентацию команд меню.

Разработанная нами система библиотек позволяет описать единую структуру меню в виде дерева или леса (набора непересекающихся деревьев). Для каждого пункта меню указывается название, заголовок, иконка, горячая клавиша и так далее. По этим описаниям для любого набора узлов может быть сгенерировано контекстное меню, панель инструментов, главное меню, а также вспомогательные интерфейсы, которые позволяют автоматически соединить меню с обработчиками команд.

### **Организация разработки бизнес-приложения**

Использование системы `Thornado` позволяет внедрить профессиональную схему разработки бизнес-приложений даже при работе в небольших коллективах над небольшими проектами. В разработке приложений принимают участие следующие специалисты: программист шаблонов, программист приложения, верстальщик и менеджер по работе с заказчиком.

Программист шаблонов разрабатывает новые шаблоны и стереотипные библиотеки `Thornado`, не занимаясь отдельными приложениями. Он должен являться программистом высокой квалификации. Он может также выполнять функции программиста приложений в малой организации.

Программист приложений создает конкретное приложение на базе шаблонов и библиотек `Thornado`, используя спецификации. Он должен являться программистом, однако для его работы достаточно средней квалификации.

Верстальщик обеспечивает правильное форматирование документов. Он может не обладать квалификацией программиста, но должен владеть языком `HTML` или другим языком разметки текста.

Менеджер по работе с заказчиком собирает данные о бизнес-процессах организации-заказчика, составляет спецификацию продукта, описывает на естественном языке те части приложения, которые не могут быть сгенерированы автоматически. Он должен обладать представлением об используемых шаблонах, быть опытным пользователем приложений и разбираться в бизнес-процессах. Менеджер занимается составлением технического задания, тестированием, внедрением и технической поддержкой. Таким образом, основная нагрузка при разработке приложений переходит к менеджеру. При этом менеджер может не обладать квалификацией программиста.

При работе над приложением между участниками разработки приложений возникают следующие информационные потоки, представленные на рис. 1.

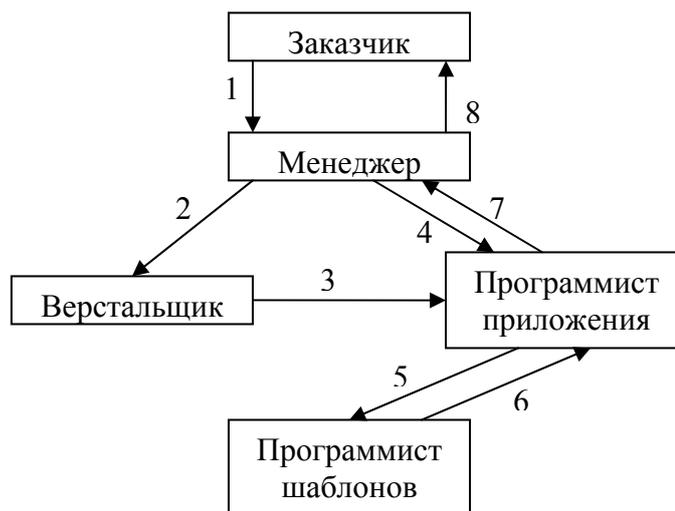


Рис. 1. Диаграмма информационных потоков между участниками разработки приложений

На рисунке приняты следующие цифровые обозначения: 1 – постановка задачи, все материалы для работы: документы, примеры данных и т.д.; 2 – документы для верстки с указанием переменных; 3 – сверстанные документы; 4 – описание данных, описания переменных документов, неформальные описания бизнес-процессов; 5 – запросы на добавление или изменение шаблонов или библиотек кодогенерации; 6 – шаблоны и библиотеки кодогенерации; 7 – программа; 8 – внедрение программы, тестирование и написание руководства пользователя.

## Литература

1. Barnett M., Campbell C., Grieskamp W., Gurevich Y., Nachmanson L., Schulte W., Tillmann N., Veanes M. Specifications in the development process: An ASML demonstration // Specification and Verification of Component-Based Systems, 2006.
2. Czarnecki K., Eisenecker U.W. Generative Programming: Methods, Tools, and Applications. – Addison-Wesley, 2000.
3. Microsoft® Windows Forms Designer. – Режим доступа: [http://msdn2.microsoft.com/enus/library/e06hs424\(VS.80\).aspx](http://msdn2.microsoft.com/enus/library/e06hs424(VS.80).aspx), своб.
4. IBM® Rational Application Developer for WebSphere Software. – Режим доступа: <http://www-306.ibm.com/software/awdtools/developer/application/>, своб.
5. Окуловский Ю.С. Язык TH. // Труды 37-й региональной конф. в Кунгуре. – Екатеринбург, 2006.
6. Окуловский Ю.С. Спецификация языка Thorn. – Режим доступа: <http://cs.usu.edu.ru/langs/th>, своб.

7. Van Deursen A., Klint P., Visserm J. Domain-Specific Languages: an annotated bibliography // ACM SIGPLAN Notices. – 2000. – V. 35. – № 6.
8. Microsoft® C# Language Specification. – Режим доступа: <http://msdn2.microsoft.com/en-us/vcsharp/aa336809.aspx>, своб.
9. Деев Д.В. Исследование структур распространенных графических интерфейсов пользователя // Международная конференция «Информационно-математические технологии в экономике, технике и образовании», тезисы конференции, Екатеринбург, 2007.
10. ГОСТ Р 6.30 – 2003. Унифицированная система организационно-распорядительной документации. Требования к оформлению документов. – Введ. 03.03.2003. – М.: Госстандарт России : Изд-во стандартов, 2003.
11. World Wide Web Consortium. HTML 4.0 Specification. – Режим доступа: <http://www.w3.org/TR/html401/>, своб.

*Деев Дмитрий Васильевич*

— Уральский государственный лесотехнический университет, аспирант, [deyeff@gmail.com](mailto:deyeff@gmail.com)

*Окуловский Юрий Сергеевич*

— Уральский государственный университет им. А.М. Горького, аспирант, [yuri.okulovsky@gmail.com](mailto:yuri.okulovsky@gmail.com)

*Попов Владимир Юрьевич*

Уральский государственный университет им. А.М. Горького, доктор технических наук, профессор, [Vladimir.Popov@usu.ru](mailto:Vladimir.Popov@usu.ru)

*Частовских Виктор Петрович*

Уральский государственный лесотехнический университет, доктор технических наук, профессор, [vip@usfeu.ru](mailto:vip@usfeu.ru)

**УДК 004.75**

## **МНОГОАГЕНТНЫЕ СИСТЕМЫ ИДЕНТИФИКАЦИИ С ИСПОЛЬЗОВАНИЕМ СЕМАНТИЧЕСКОГО СПОСОБА РАСПРЕДЕЛЕНИЯ МОДЕЛИ ФУНКЦИОНИРОВАНИЯ ОБЪЕКТА**

**А.И. Кальянова, А.В. Демин**

В статье рассмотрена архитектура многоагентного комплекса идентификации технических объектов и проведены исследования применения многоагентного подхода с семантическим способом распределения моделей функционирования объекта.

**Ключевые слова:** многоагентные системы, идентификация, модели функционирования.

### **Введение**

Развитие технических объектов, таких как распределенные вычислительные системы, системы связи, специальная техника и т.п., происходит в направлении функционального и структурного усложнения, а также интеграции различных по типу устройств (механических, электрических, электронных и пр.), связанных друг с другом и оказывающих большое влияние на свое окружение, в один технический объект. В связи с этим возникает задача создания эффективной системы идентификации технических объектов на соответствие требуемым функциональным и параметрическим показателям.

Учитывая, что любой технический объект состоит из большого количества разнообразных элементов, связанных друг с другом и оказывающих большое влияние на свое окружение, необходимо разработать математическую модель объекта, рассматривающую объект как сложную систему, состоящую из элементов и связей между ними.

Рассмотрен многоагентный подход к идентификации технических объектов на основе моделей устройств с применением семантического распределения модели функционирования объекта между агентами. Разработана архитектура многоагентного диагностического комплекса и проведены исследования применения многоагентного подхода для семантического распределения моделей поведения устройства.

### Многоагентный подход в системах идентификации

Формулируется следующая задача идентификации. Имеется структурированный объект, проявляющий неисправное поведение. Объект состоит из компонентов. Есть структура связей между компонентами и возможность снимать с них показания на входах и выходах. Необходимо с минимальными затратами ресурсов определить, какие из компонентов неисправны.

Для реализации данного подхода используются многоагентные системы идентификации.

Решение задачи идентификации состоит из двух этапов. В первом идентифицируется множество возможных различий модели и идентифицируемого объекта. Во втором осуществляется сбор дополнительной информации в виде наблюдений, которая улучшает множество возможных различий модели и объекта до тех пор, пока они не сведутся к точному отражению фактических отличий.

Предполагается, что различия модели и неисправного объекта не наблюдаемы непосредственно. Вместо этого все нарушения предположений об исправности компонентов должны быть выведены косвенно из поведенческих наблюдений [1].

Принцип идентификации на основе модели объекта представлен на рис. 1.

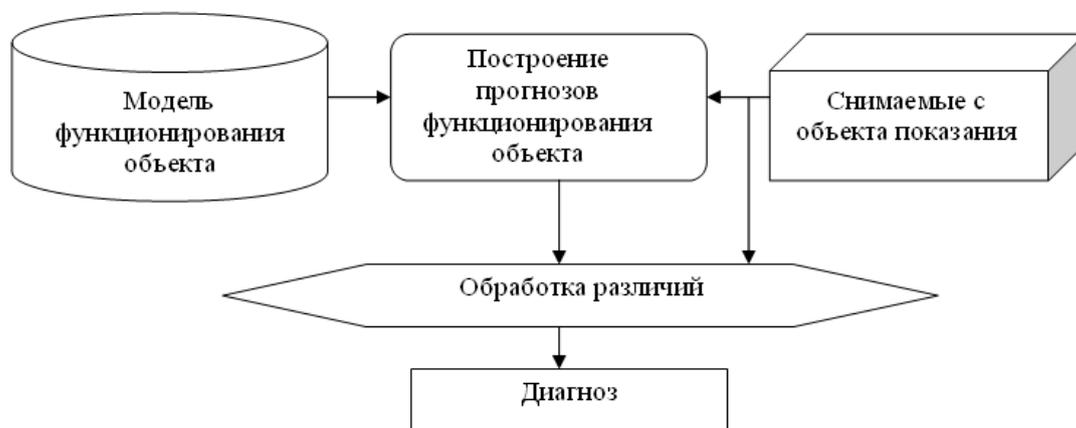


Рис. 1. Принцип идентификации на основе модели объекта

Основные достоинства подхода, основанного на модели устройства:

- идентификация множественных неисправностей;
- работа с непредсказуемыми вариантами;
- возможность идентификации новых объектов;
- независимость точности описания моделей компонентов от опыта эксперта;
- многократное использование моделей;
- детальность объяснений.

Есть несколько причин, вследствие которых использование единого идентификационного механизма может быть невыгодно. Прежде всего, если объект имеет большие размеры и физически распределен в пространстве, то при централизованной идентификации может наблюдаться нехватка времени для вычислений и получении всех наблю-

дений. Во-вторых, если структура объекта является динамической, то изменения могут происходить слишком быстро, из-за чего значения измерений будут быстро терять актуальность. Наконец, иногда существование полной модели просто нежелательно из соображений безопасности. Примеры таких объектов – современные телекоммуникационные сети, динамически конфигурируемые роботизированные системы и т.д. Для подобных систем решением может являться идентификация с помощью многоагентного подхода [2].

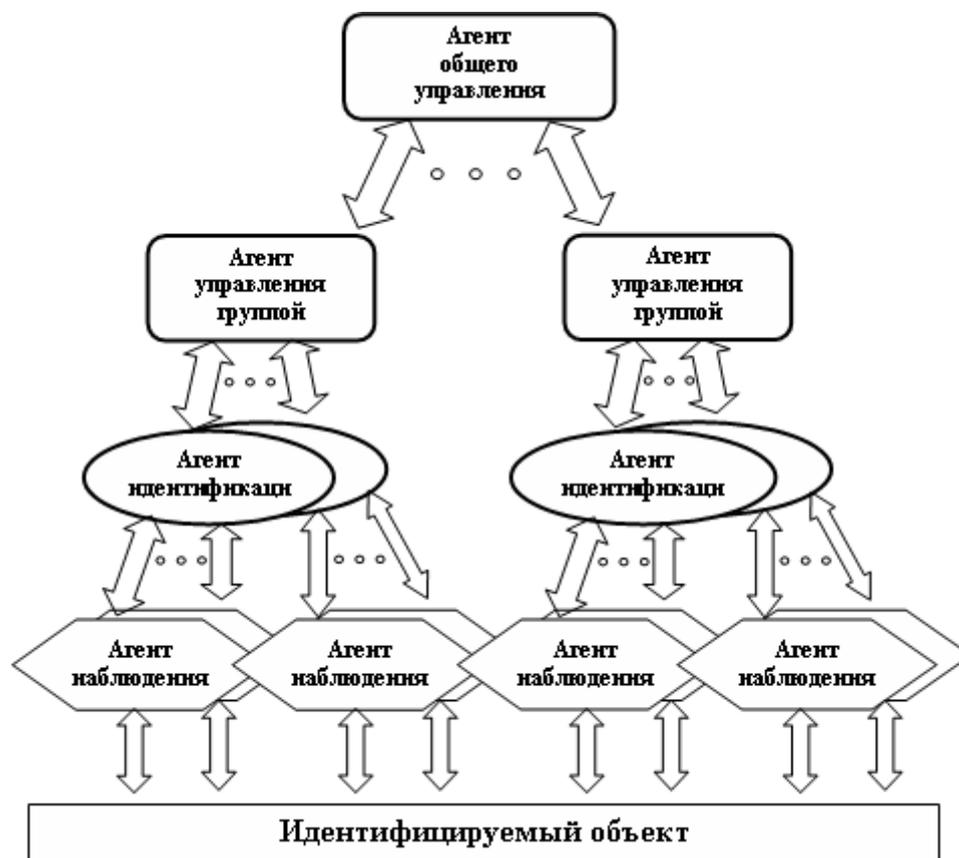


Рис. 2. Архитектура многоагентной системы идентификации

Для построения многоагентной системы идентификации на основе модели устройства используется следующая архитектура (рис. 2). Нижний уровень представляют агенты наблюдения, которые обеспечивают снятие показаний с идентифицируемого объекта. С ними взаимодействуют агенты идентификации, занимающиеся обработкой получаемой информации. Обработкой и сведением воедино сведений, поступающих от агентов идентификации, занимаются агенты управления группами агентов идентификации, с которыми взаимодействует агент общего управления, формирующий итоговое представление об идентифицируемом объекте.

После запуска каждый из агентов автономен и действует исходя из своей цели, имеющейся в наличии информации и сообщений, получаемых из внешней среды, других агентов или пользователя.

Каждый агент идентификации включает в себя модуль взаимодействия, базу знаний о своей части объекта и модуль идентификации. Целью агента идентификации является разрешение противоречий, обнаруженных при сравнении прогнозов поведения компонентов объекта с результатами показаний, снятых с него непосредственно. Для достижения своей цели агент идентификации осуществляет построение прогнозов поведения компонентов объекта, выявление противоречий, возникающих при различии прогнозов

со снимаемыми показаниями, и вычисление количества информации, которое может быть получено при снятии дополнительных показаний в соответствующих точках связи компонентов объекта. Полученные оценки количества информации, получаемого при снятии показаний в той или иной точке устройства, агент идентификации передает соответствующим агентам наблюдения. Агенту управления группой передаются данные о результатах, полученных на основе анализа состояния идентифицируемого объекта, и осуществляется обмен информацией с другими агентами идентификации [3].

Агент наблюдения содержит модуль взаимодействия, блок обработки оценок агента идентификации и модуль снятия показаний с объекта. Целью агента наблюдения является предоставление агенту идентификации результата снятия показания в той точке объекта из возможных, для которой оценка количества полученной информации максимальна. Для этой цели агент наблюдения производит обработку оценок, полученных от агента идентификации, и осуществляет снятие показаний в выбранном месте объекта. Полученный результат передается агенту идентификации.

Агент управления включает в себя модуль взаимодействия и модуль обработки полученных от агентов идентификации запросов и информации. Целью агента управления является получение целостной картины о состоянии части объекта, идентифицируемой связанными с ним агентами идентификации. Для этой цели агент управления решает вопросы взаимодействия между агентами идентификации своей группы и групп других агентов управления, а также поддерживает непротиворечивость получаемой информации. Агент общего управления осуществляет взаимодействие между агентами управления группами и сводит воедино всю информацию о состоянии идентифицируемого объекта.

Основными трудностями многоагентного подхода в идентификации является проблема сотрудничества и согласования работы агентов идентификации. Агент идентификации знает только своего агента управления и не имеет информации о прочих агентах идентификации системы. Таким образом, он не обладает знаниями других агентов идентификации и поэтому способен делать прогнозы только о поведении своей части идентифицируемого объекта. Это делается с целью снижения сложности внутренней структуры агентов идентификации. Взаимодействие и связь между агентами идентификации обеспечиваются через агентов управлениями. Агент управления группы знает всех агентов идентификации, принадлежащих его группе. В его задачу входит обмен информацией, связь между агентами группы и вычисление метки вершины, общей для нескольких агентов идентификации. Агент управления группы имеет информацию только о состоянии своей группы, в то время как агент общего управления владеет информацией обо всей системе в целом.

Когда для одной и той же точки объекта спрогнозированы значения несколькими агентами идентификации, возникает проблема сведения этой информации воедино. Если агенты идентификации принадлежат одной группе, то агент управления группы самостоятельно вычисляет метку общей вершины и передает ее агентам идентификации, которые в ней нуждаются, а также, возможно, агенту общего управления. Вычисление метки вершины состоит в объединении меток, вычисленных агентами идентификации, и удалении противоречий.

До проведения цельной идентификации всего объекта агенты должны идентифицировать свою локальную часть системы, исходя из имеющихся в наличии знаний. Важный вопрос – могут ли они это сделать независимо друг от друга. Ответ на данный вопрос напрямую зависит от способа распределения знаний между агентами. Рассмотрим семантический способ распределения модели устройства между агентами.

В случае семантического распределения модель поведения объекта делится между агентами согласно типу составляющих ее частей, т.е. каждый из агентов анализирует различные части объекта на основе данной ему части модели (подмодели). При этом

каждая из подмоделей описывает поведение соответствующей ей части объекта независимо от остальных подмоделей поведения объекта других агентов. Независимость подразумевает, что нет никаких соединений между различными подмоделями поведения или известны значения во всех точках этих соединений. Примером данного распределения модели объекта могут служить отдельные модели электрического и термодинамического поведения устройства. Другим образцом подобного распределения являются отдельные модели блоков устройства с известными значениями в местах их связи друг с другом. Семантическое распределение в приведенных выше условиях означает, что каждый из агентов, идентифицирующих свою часть объекта, может работать независимо от других, обладая всей необходимой информацией. Таким образом, для получения информации о состоянии всего объекта в целом достаточно произвести объединение сведений, поступивших от каждого из агентов по отдельности, с последующим удалением избыточной информации.

Для начала рассмотрим эффективность применения многоагентного подхода при пространственном распределении модели объекта между агентами. Введем следующие обозначения:  $N$  – количество компонентов объекта;  $Ag_i$  –  $i$ -ый агент идентификации;  $N_i$  – количество компонентов объекта, идентифицируемых  $Ag_i$ ;  $IT$  – время идентификации. Время идентификации неисправного объекта напрямую зависит от числа его компонентов, поэтому время идентификации неисправности без использования многоагентного подхода можно обозначить как  $IT_{single}(N)$ .

При использовании многоагентного подхода модель объекта делится на части, идентификацией которых занимаются соответствующие агенты. Предполагаем, что все агенты могут выполнять свои функции параллельно. Для получения информации о прогнозах значений в общих для нескольких агентов точках модели объекта и сведении воедино результатов работы агентов идентификации используются агенты управления. Частота, с которой у агентов идентификации возникает необходимость получать информацию о прогнозах значений в общих точках модели, зависит от структуры связей между компонентами объекта и распределения его модели между агентами.

Заметим, что основные затраты ресурсов приходятся на работу агентов идентификации, в то время как в процессе обмена информацией у агентов управления нет необходимости производить какие-либо ресурсоемкие вычисления. Соответственно, затраты ресурсов на коммуникацию между агентами значительно меньше ресурсов, необходимых для работы агентов идентификации. Еще меньше затраты ресурсов, требуемых на передачу агентам идентификации значений измерений, снимаемых непосредственно с объекта. Эти затраты ресурсов, которые также зависят от количества компонентов объекта, обозначим как  $NR(N)$ .

Рассмотрим идеальный вариант, когда в процессе определения неисправности у агентов идентификации нет необходимости в информации о прогнозах значений в общих точках модели объекта. С учетом того, что все агенты могут функционировать параллельно друг с другом, время идентификации объекта будет соответствовать времени работы агента, идентифицирующую свою часть объекта дольше других:

$$IT_{Multiagent}^{Best}(N) = \max_i (IT_{Ag_i}(N_i)) + NR(N).$$

Рассмотрим наихудший вариант, когда каждому агенту идентификации для проведения каждого шага работы необходимо получать прогнозы значений общих точек, получаемые другими агентами. В этом случае каждый из агентов идентификации отправляет запрос на получение значения прогноза в общей точке модели объекта и может продолжить свои вычисления только после того, как другой агент идентификации, вычисляющий данные значения прогнозов, закончит свою работу и пришлет запрашиваемый результат. Таким образом, для данного варианта все агенты идентификации бу-

дуг выполнять свои вычисления последовательно друг за другом, и итоговое время идентификации неисправности будет превышать время работы без использования многоагентного подхода на величину накладных расходов, требуемых на коммуникацию между агентами:

$$IT_{Multiagent}^{Worst}(N) = \sum_i (IT_{Ag_i}(N_i)) + NR(N) = IT_{Single}(N) + NR(N).$$

Стоит заметить, что настолько плохой вариант может происходить редко, так как в идентифицируемой агентом модели объекта, как правило, всегда может быть обнаружена часть, для обработки которой не требуется получение значений прогнозов в общих точках модели объекта.

Исходя из полученных формул, время  $IT_{Multiagent}(N)$ , затрачиваемое на идентификацию объекта, можно выразить следующим образом:

$$\max_i (IT_{Ag_i}(N_i)) + NR(N) \leq IT_{Multiagent}(N) \leq IT_{Single}(N) + NR(N).$$

Теперь оценим эффективность применения многоагентного подхода к идентификации при семантическом распределении модели объекта.

Семантическое распределение модели объекта подразумевает, что каждая из подмоделей описывает поведение соответствующей ей части объекта независимо от остальных подмоделей поведения объекта, обрабатываемых другими агентами. Благодаря этому каждый из агентов, идентифицирующих свою часть объекта, может работать независимо от других, обладая всей необходимой информацией, и для получения информации о состоянии всего объекта в целом достаточно произвести объединение сведений, поступивших от каждого из агентов.

Проведенные тесты показали, что затраты ресурсов агентов управления при семантическом распределении модели объекта ниже затрат ресурсов агентов управления, работающих в условиях пространственного распределения, вследствие отсутствия необходимости поддерживать взаимодействие агентов идентификации. Это означает, что ситуация аналогична идеальному варианту при пространственном распределении модели объекта, т.е. все агенты могут идентифицировать свои части объекта параллельно, и время идентификации будет соответствовать времени работы агента, идентифицирующего свою часть объекта дольше остальных:

$$DT_{Multiagent}^{Best}(N) = \max_i (DT_{Ag_i}(N_i)) + NR(N).$$

Одним из выгодных подходов использования многоагентной идентификации с семантическим распределением модели является идентификация объектов, состоящих из большого количества блоков, для которых можно локально осуществлять поиск неисправности без использования единой модели поведения всего объекта.

## Заключение

В статье рассмотрено применение многоагентного подхода к идентификации технических объектов на основе моделей объектов с использованием семантического способа распределения модели функционирования объекта между агентами, в котором модель поведения объекта распределяется между агентами согласно типу составляющих его частей. Проведенные тесты показали, что использование многоагентного подхода в идентификации на основе моделей объекта позволяет повысить эффективность идентификационного комплекса благодаря снижению затрат ресурсов за счет распараллеливания работы идентификационных механизмов и нахождения локальных неисправностей использованием лишь части агентов. При этом эффективность многоагентного подхода напрямую зависит от структуры объекта и того, каким образом произведено распределение его модели функционирования по агентам идентификации.

## Литература

1. Достоверный и правдоподобный вывод в интеллектуальных системах / Под ред. Вагина В.Н. и Поспелова Д.А. – М.: ФИЗМАТЛИТ, 2004. – 704 с.
2. Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – М.: Эдиториал УРСС, 2002. – 352 с.
3. Поспелов Д.А. Многоагентные системы – настоящее и будущее // Информационные технологии и вычислительные системы. – 1998. – № 1. – С.14–21.

*Кальянова Арина Игоревна*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, Kalyanova-a@mail.ru

*Демин Анатолий Владимирович*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, dav\_60@mail.ru

УДК 004.4'22

**ПРИМЕНЕНИЕ ПРИНЦИПОВ MDD И АСПЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ К РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, СВЯЗАННОГО С ФОРМАЛЬНЫМИ ГРАММАТИКАМИ****А.А. Бреслав, И.Ю. Попов**

Современные средства разработки грамматик и связанных с ними приложений не удовлетворяют принципам инженерии программного обеспечения (ПО): различные аспекты системы смешиваются, нарушается модульность и разграничение уровней абстракции. В настоящей работе представлена концепция средства разработки, призванного решить эти проблемы, используя принципы модельно-ориентированной разработки и аспектно-ориентированного программирования.

**Ключевые слова:** грамматика, MDD, модульность, повторное использование.

**Введение**

Разрабатывая программные (и не только программные) системы, необходимо следовать определенным правилам, чтобы совладать со сложностью, присущей решаемой задаче [1]. Речь идет о поддержании на должном уровне так называемых *факторов внутреннего качества* [2], которые в значительной мере обеспечивают уровень внешнего качества: корректность, стабильность, удобство в использовании и т. д. К факторам внутреннего качества относят модульность и повторное использование модулей, разделение уровней абстракции (слоев) в системе, расширяемость, понятность и читаемость кода.

В настоящей работе мы обратимся к тем программным системам, которые зависят от формальных языков и их грамматик (в работе [3] для таких систем вводится название *grammarware*). К этому классу относятся, например, компиляторы, интерпретаторы, средства статического анализа программного кода, генераторы кода, средства трансформации программ и другие инструменты, используемые для работы с программным кодом. Кроме того, многие приложения используют собственные «малые» языки для файлов конфигурации (в последнее время для этой цели используют XML, роль грамматики для которого играют XSD или DTD-определения). В последние годы стали популярны модельно-ориентированные подходы [4], а с ними и предметно-ориентированные языки (с текстовым или графическим синтаксисом), ставшие инструментами уже не только для программистов, но и для конечных пользователей [5]. Все это многообразие приложений страдает от одной серьезной проблемы: внутреннее качество оставляет желать много лучшего в той части, где речь идет о грамматиках и связанных с ними артефактах разработки. Хорошо развитые в других областях программирования подходы и инструменты, обеспечивающие поддержание на должном уровне факторов внутреннего качества системы, попросту не применяются для *grammarware* [3]. Написанные вручную или с использованием генераторов синтаксические анализаторы и другие средства обработки языков, как правило, монолитны (разбиение на модули редко заходит дальше разделения лексического и синтаксического анализа), т.е. происходит смещение различных аспектов системы (и уровней абстракции). Как следствие, такие программы трудно читать, поддерживать и развивать, а использовать такой код повторно почти невозможно.

Справедливости ради стоит отметить, что в современных средствах автоматизации разработки *grammarware* предпринимаются попытки решить означенные проблемы. Так, генератор синтаксических анализаторов SableCC разделяет определение грам-

матики и семантические действия, используя схему трансляции [7], система ANTLR предоставляет возможность повторного использования ряда артефактов за счет наследования грамматик и анализаторов деревьев (Tree Parsers) [6]. Кроме того, существуют системы, строящие на выходе готовые приложения (xText [8], TCS [9]), полностью генерируемые, а потому не подлежащие ни чтению, ни расширению вручную. Однако этих достижений недостаточно: традиционные генераторы, как бы гибки они ни были, решают низкоуровневую задачу построения синтаксического анализатора по грамматике определенного класса, что существенно ограничивает возможности разработчика в смысле разделения системы на модули и слои, а системам, строящим готовые приложения, не хватает гибкости – они используются в основном для прототипирования.

В настоящей работе мы представляем концепцию инструментальной платформы, которая призвана обеспечить разработчиков grammarware аппаратом, позволяющим поддерживать внутреннее качество их продуктов на должном уровне. Мы стремимся обеспечить разделение различных аспектов системы и различных уровней абстракции, а также облегчить эволюционные изменения как системы в целом, так и отдельных модулей. Предлагаемый подход основан на современных концепциях разработки ПО – MDD (Model Driven Development [4]) и аспектно-ориентированном [12] и генеративном программировании [13]. В нашей системе контекстно-свободная грамматика задается в EBNF [14] без каких-либо априорных ограничений на класс грамматики. Определение грамматики может быть дополнено метаданными произвольной структуры. Элементы грамматики, выбираемые специализированными запросами, становятся входными данными для различных преобразований. Результатами преобразований являются всевозможные артефакты разработки.

В следующих разделах дается подробное описание подхода, после чего представлен пример его использования для создания расширяемого анализатора арифметических выражений.

### Определение грамматики

Как уже было сказано, грамматика в системе задается в расширенной форме Бэкуса–Наура (Extended BNF, EBNF [14]), причем мы будем использовать нотацию, проиллюстрированную ниже:

```
Rule -> NontermSpec '->' {SymbolSpec} ';' ;
```

Имена нетерминальных символов начинаются с заглавной буквы, простые терминалы задаются в одинарных кавычках. Вот еще один пример:

```
AttributeList -> Attribute {separator Attribute} ;
```

Именованные терминалы называются со строчной буквы, а знак '|' обозначает альтернативу. Прочие обозначения будут поясняться по мере надобности.

Согласно идеологии MDD, мы рассматриваем определение грамматики не как текст, а как *модель*, т.е. набор объектов, связанных ассоциациями [15]. Далее мы будем говорить об *объектах* или *элементах* грамматики, имея в виду элементы модели, т.е. смысловые единицы описания грамматики – нетерминалы, терминалы, правила, отдельные альтернативы и т.д. Априори на грамматику не накладывается никаких ограничений, т.е. она может быть, например, неоднозначной, если это устраивает разработчика. Сама по себе система работает с грамматикой как с моделью и не пользуется ее семантикой.

### Преобразования грамматики

Располагая определением синтаксической структуры языка и соответствующими инструментами для работы с ним, разработчик может сделать очень многое. Например, можно автоматически преобразовать исходное определение к классу LL(\*) и сгенерировать определение синтаксического анализатора для ANTLR.

Кроме собственно синтаксической структуры языка, разработчик может извлекать из грамматики и другие виды информации, например, построить Java-классы вершин AST разрабатываемого языка и внедрить в генерируемое ANTLR-определение код построения AST. Существует масса других полезных приложений, значительная часть кода которых может быть получена автоматически на основе грамматик: трансляция одного языка в другой, определение правил форматирования кода, подсчет различных метрик и статический анализ кода и т.д. Наша задача – предоставить необходимый инструментарий для подобных операций с грамматиками. Все преобразования задаются с помощью специализированных языков, являющихся частью нашей системы. Каждое преобразование может быть повторно использовано.

## Виды преобразований

Преобразования (или *трансформации*) являются основными модулями при работе в рамках MDD. Фактически MDD состоит в том, чтобы запротоколировать весь процесс разработки в виде трансформаций, чтобы иметь возможность автоматически выполнить его от начала до конца в любой момент, если это понадобится, например, если изменятся входные данные или одна из трансформаций.

Для работы с грамматиками в рамках предлагаемого подхода будут использоваться следующие виды преобразований:

- построение новой грамматики (или расширение существующей);
- построение экземпляра некоторой метамодели [17];
- генерация произвольного текста (например, кода на языке программирования);
- генерация определения грамматики на входном языке какого-либо программного средства (например, ANTLR или xText).

Вне зависимости от вида преобразования разработчику потребуются несколько основных возможностей:

- обработка набора объектов, выбранного по некоторому критерию;
- проверка корректности входных данных и механизм сообщения об ошибках;
- взаимодействие различных преобразований.

Первая задача решается с помощью специализированного языка запросов, позволяющего определять критерии для выборки объектов из грамматики. Всякое преобразование работает с набором объектов, выбранным каким-то запросом. Подробнее язык запросов будет описан в следующем разделе.

Проверить корректность входных данных можно также с помощью языка запросов. Для очень широкого круга проверок вполне достаточно двух форм задания условия: «множество объектов, выбранное запросом, пусто», т.е. в грамматике нет нужных объектов, или, наоборот, «множество, выбранное по этому запросу, не пусто», т.е. в грамматике существуют нежелательные объекты. Для построения произвольных моделей и текста мы используем стандартные средства – вариацию QVT [19], позволяющую просто задать значения всех свойств каждого объекта, и язык текстовых шаблонов StringTemplate [20]; для краткости мы не описываем здесь технические детали, связанные с ними.

Более подробно следует остановиться на средствах построения грамматик. Итак, некоторый запрос выбрал из исходной грамматики набор объектов (символов, правил или частей правил). Какие инструменты понадобятся разработчику трансформации, чтобы построить новую грамматику?

Основной операцией будет создание нового правила:

```
rule RuleDefinition -> 'rule' Nonterminal '->' RuleRightSide;
```

Эта строка создает новое правило и те символы в грамматике, которые входят в его определение. Кроме создания нового правила, нужно еще добавлять альтернативы к существующим правилам. В нашей нотации это делается так:

```
rule RuleDefinition |> 'rule' Nonterminal '|>' RuleRightSide;
```

Символ «|>» означает, что то, что стоит от него справа, будет добавлено к определению нетерминала из левой части правила.

Это все, что нам понадобится в рамках данной работы. Теперь перейдем к определению языка запросов.

## Язык запросов

Задача языка запросов – предоставить инструментарий для выбора объектов из грамматики по всевозможным критериям. К таким критериям относятся как индивидуальные свойства объектов (например, имя символа), так и структурный контекст – свойства фрагмента грамматики, частью которого является объект. Примеры типичных запросов:

1. нетерминал, имя которого оканчивается на «Expression»;
2. правило, в правой части которого стоит ровно один символ;
3. символ, стоящий в каком-либо правиле между двумя нетерминалами.

Начнем с простых свойств выбираемых объектов. Нам понадобится выбирать символы грамматики по имени и типу (нетерминал, терминал). Для этих целей в языке запросов имеются следующие инструменты.

Запрос, представляющий из себя просто регулярное выражение для строки символов, с операторами «\*» (любое количество любых символов) и «.» (один любой символ), определяет шаблон поиска по имени. Для ограничения типа символа грамматики используются следующие обозначения (в квадратных скобках указывается ограничение на имя):

- `#symbol[Plus]` – любой символ (терминал или нетерминал) с именем «Plus»;
- `#nonterm[Expr*]` – нетерминал, имя которого начинается на «Expr»;
- `#term[* .num]` – терминал, имя которого заканчивается на «num», но содержит еще хотя бы один символ.

Перейдем к структурному контексту. Положение того или иного символа в правиле и вид этого правила задается регулярным выражением на символах грамматики. В таких выражениях целесообразно использовать переменные:

1.  $N \rightarrow x \mid N x$  – в точности такое правило: « $N \rightarrow x \mid N x$ »;
2.  $N: \#nonterm \rightarrow x \mid N x$  – правила такого вида, где в первой альтернативе стоит символ с именем «x», а во второй – он же, предваренный нетерминалом из левой части правила;
3.  $\#nonterm \rightarrow \#term^*$  – правила, в правой части которых стоит цепочка терминалов.

Нужно заметить, что по практическим соображениям очень часто необходимо рассматривать не правила целиком, а лишь отдельные альтернативы. При этом имеются в виду альтернативы верхнего уровня, т.е. те знаки «|», непосредственным родителем которых в дереве разбора определения грамматики будет само правило (символ «->»). Для решения этой проблемы мы вводим новые обозначения. Знаком «||» мы разделяем альтернативы верхнего уровня. В дополнение к «||» введем еще одно обозначение: кроме «->», можно использовать «|>», обозначающий, что имеется в виду не правило целиком, а какие-то его альтернативы верхнего уровня. Например, выражение « $N \rightarrow x \mid N x$ » не допускает правило « $N \rightarrow x \mid N x \mid x y$ », а то же выражение со знаком «|>» (« $N |> x \mid N x$ ») допускает.

Также нам понадобится оператор, которому соответствуют все альтернативы данного правила, кроме указанных. Для этой цели мы используем обозначение «{ }».

На самом деле концепция запросов и преобразований сродни идеям, применяемым в аспектно-ориентированном программировании [12]: запросы аналогичны точкам встраивания (pointcuts), а преобразования – встраиваемому коду (advice).

На этом мы остановимся, не затрагивая дальнейшие тонкости, связанные с языком запросов.

## Использование запросов и преобразований

Всякое преобразование работает с результатами некоторого запроса, который указывается перед телом преобразования:

```
map [запрос] {  
  // тело преобразования  
}
```

Запрос определяет переменные, которые могут быть использованы в теле преобразования:

```
map [#nonterm |> #nonterm op:#term #nonterm] {  
  rule BinOp |> op;  
}
```

Это преобразование строит правило, в правой части которого перечислены все терминалы, стоящие между двумя нетерминалами в какой-либо альтернативе в исходной грамматике. Тело преобразования выполняется для каждого значения переменной `op`, выбранного по запросу. Таким образом, если в исходной грамматике были, например, такие правила:

```
S -> S '+' M || M;  
M -> M '*' F || F;
```

то в результате применения нашего преобразования получится правило

```
BinOp -> '+' || '*';
```

Взаимодействие различных преобразований в нашей системе обеспечивается с помощью *глобального репозитория*. Задача состоит в том, чтобы каждое преобразование могло обращаться к объектам, созданным другими преобразованиями (или тем же преобразованием, но вызванным ранее с другими аргументами), причем желательно, чтобы не было необходимости связывать зависимые преобразования статически. В частности, для нашего примера важно не создавать многочисленные копии нетерминала `BinOp`, определяемые по-разному, а, напротив, добавлять альтернативы к определению одного и того же нетерминала. Во многих системах для этих целей применяются журналы (*traces* [19]), хранящие информацию о том, какой объект является результатом какого преобразования, однако нам такой подход представляется не очень удобным. Мы делаем так: всякий создаваемый объект помещается в глобальный репозиторий и может быть извлечен оттуда по значениям ключевых полей. Например, если мы хотим получить ранее созданный символ с именем «`BinOp`» (или создать его, если он не был создан ранее), нам достаточно написать следующее:

```
BinOp[name="BinOp"]
```

Поиск осуществляется по двум параметрам: классу объекта (*символ*, определяется контекстом описания переменной) и значению атрибута `name` (все атрибуты для поиска и их значения указываются в квадратных скобках после имени переменной). Поскольку имя (атрибут `name`) – наиболее распространенный первичный ключ, мы предоставляем сокращенный синтаксис для случая, когда поиск ведется по нему:

```
BinOp
```

Эта строка делает в точности то же, что и предыдущая, но имя переменной теперь само по себе является ключом для поиска.

## Метаданные

В большинстве существующих средств разработки `grammarware` входной язык не ограничивается простым определением грамматики, в него встраиваются различные расширения – приоритеты и ассоциативность операций, семантические акции или схемы трансляции в генераторах компиляторов, синтезируемые и наследуемые атрибуты, правила форматирования кода и т.д. Фактически каждое средство разработки определяет собственный вид *метаданных*, которыми дополняется грамматика. Для каждого кон-

кретного средства разработки метаданные имеют свой собственный вид и семантику: они встроены во входной язык и обрабатываются определенным фиксированным способом. Например, семантические акции в ANTLR пишутся на языке Java с небольшими расширениями, заключаются в фигурные скобки и помещаются в левых частях правил.

Нашей системе не требуется расширять грамматику для какой-то конкретной цели, ведь все, что она умеет делать – предоставлять разработчику возможность обрабатывать грамматику так, как ему хочется. Разработчику будет удобно писать трансформации, если во входной грамматике, кроме синтаксической структуры языка, будут содержаться еще какие-то данные. Так, например, разработчик, создающий трансформацию, которая по входной грамматике строит форматировщик кода (code formatter), мог бы предполагать, что синтаксические конструкции, играющие роль блоков, внутри которых нужно увеличивать отступ, каким-то образом отмечены. Разрешим сопоставлять любому элементу грамматики набор метаданных, с которым априори не ассоциирована никакая семантика – семантику задают трансформации, которые будут применяться к такой грамматике.

Итак, каждому элементу можно приписать некий набор атрибутов, у каждого из которых есть имя и, возможно, значение. Поддерживаются следующие типы значений: целое число, строка, идентификатор (просто слово), набор атрибутов. Атрибуты и их значения указываются в квадратных скобках после имени (или обозначения) символа:

```
'+' : [grName="Plus", priority=Low]
```

Здесь значение атрибута grName – строка «Plus», а значение атрибута priority – идентификатор Low. Для удобства мы введем группы атрибутов, к которым можно многократно обращаться по имени. Описываются группы следующим образом:

```
GROUP_NAME = [attr1=value1, attr2=value2, ...];
```

а использовать их можно так:

```
symbol : GROUP_NAME
```

Метаданные могут использоваться в запросах для уточнения критериев поиска. Кроме того, преобразования, строящие на выходе грамматику, могут добавлять метаданные к ее элементам:

```
rule Op[name=Plus*, priority:=Low] |> '+';
```

Заметим, что в приведенном примере атрибут name используется для поиска (к нему применяется знак «=»), а атрибут priority назначается (используется знак «:=»).

## Использование предложенного подхода

Мы постараемся продемонстрировать преимущества описанного подхода на довольно простом примере – анализаторе арифметических выражений.

Язык выражений задается следующей грамматикой:

```
Expression -> num | '('Expression')' | Expression Operation Expression;
Operation -> '+' | '*' | '-' | '/' | '^';
```

Здесь num – терминал, обозначающий числовой литерал (мы опускаем лексические определения для краткости). Как видно, эта грамматика неоднозначна: не учитываются приоритеты и ассоциативность операций. Вместо того чтобы вручную устранить неоднозначность, напишем преобразование, которое сделает это. В конечном итоге наше преобразование будет работать для любой грамматики, т.е. будет повторно используемым модулем.

Поскольку из исходной грамматики приоритеты и ассоциативность операций извлечь невозможно, мы воспользуемся метаданными, чтобы передать эту информацию нашему преобразованию. Сделаем три макроопределения:

```
PLUS = [grName = 'Plus'; assoc=left; arg='Plus'];
MULT = [grName = 'Mult'; assoc=left; super='Plus'; arg='Pow'];
POW = [grName = 'Pow'; assoc=right; super='Mult'];
```

Здесь атрибут `grName` фиксирует название группы операций, `assoc` – ассоциативность, `super` – группу операций с приоритетом, меньшим на единицу, `arg` – группу операций с приоритетом, большим на единицу.

Преобразуем нашу грамматику так, чтобы сопоставить знакам операций соответствующие группы атрибутов:

```
map [Expression -> def:{*}] {
  rule Expression -> def;
}
map [Operation -> {*}] {
  rule Operation -> '+' : PLUS | '-' : PLUS | '*' : MULT | '/' : MULT
                  | '^' : POW ;
}
```

Теперь напишем преобразование, строящее по исходной грамматике с такими метаданными однозначную грамматику. Наши операции должны выстроиться по возрастанию приоритета, причем в самом конце этой «цепочки» будут уже не бинарные операции, а числа и выражения в скобках. Для начала выделим для этих конечных элементов отдельное правило:

```
map [Expression -> atoms:{*} || Expression Op Expression] {
  rule Atom -> atoms;
}
```

Вот результат применения этого преобразования:

```
Atom -> num | '(' Expression ')';
```

Теперь сгруппируем знаки операций по приоритетам (т.е. по значениям атрибута `grName`):

```
map [Operation |> op:#symbol] {
  rule Op[op.grName + 'Op'] |> op;
}
```

Таким образом, мы получим следующий результат:

```
PlusOp -> '+' | '-' ;
MultOp -> '*' | '/' ;
PowOp -> '^' ;
```

Теперь обратимся к общему виду правил для операций, которые должны у нас получиться:

$$S \rightarrow A \mid A \text{ Op } S; \tag{1}$$

для правоассоциативных операций, и

$$S \rightarrow A \mid S \text{ Op } A; \tag{2}$$

для левоассоциативных. Заметим, что первая альтернатива в обоих случаях одинакова, причем именно по этой альтернативе операции выстраиваются в цепочку:

```
Expression -> Plus -> Mult -> Pow -> Atom
```

Напишем преобразование, которое «строит эту цепочку»:

```
map [Operation |> op:#symbol] {
  rule Super[op.super || 'Expression'] -> Operation[op.grName];
}
```

Входными данными являются все символы, стоящие в правой части определения нетерминала `Operation`, т.е. все знаки операций. В этом преобразовании использовано ранее не упоминавшееся обозначение

```
Super[op.super || 'Expression']
```

Здесь знак «||» обозначает «ленивое или» в следующем смысле: если значение `op.super` определено, то выражение «`op.super || 'Expression'`» имеет значение `op.super`, а если нет – то значение `'Expression'`. Таким образом, для операций «Pow» будет построено правило

```
Mult -> Pow ;
```

поскольку в группе атрибутов POW есть атрибут `super` и его значение – `'Mult'`, а для «Plus» будет построено правило

```
Expression -> Plus ;
```

поскольку в соответствующей группе нет атрибута `super`.

Осталось создать вторые альтернативы в правилах (1) и (2). Они различаются в зависимости от ассоциативности операции:

```
map [op:#symbol[assoc=left]] {
  rule Operation[op.grName] -> Arg[op.arg || 'Factor']
    | Operation Op[op.grName + 'Op'] Arg;
}
map [op:#symbol[assoc=right]] {
  rule Operation[op.grName] -> Arg[op.arg || 'Factor']
    | Arg Op[op.grName + 'Op'] Operation;
}
```

Параметры запроса позволяют различать значения атрибута `assoc` – для каждого из них мы строим свой тип альтернативы. Эти преобразования создают по две альтернативы в каждом правиле, но дублирующиеся альтернативы автоматически отбрасываются. В конечном итоге мы получим однозначную грамматику:

```
Expression -> Plus ;
PlusOp -> '+' | '-' ;
Plus -> Mult | Plus PlusOp Mult ;
MultOp -> '*' | '/' ;
Mult -> Pow | Mult MultOp Pow ;
PowOp -> '^' ;
Pow -> Atom | Atom PowOp Pow ;
Atom -> num | '('Expression')' ;
```

Хотелось бы еще избавиться от левой рекурсии. Создадим для этого повторно используемое преобразование, не завязанное на структуру исходной грамматики:

```
map [N:#nonterm -> N a:* || b:{*}] {
  rule R[N.grName + 'Rest'] -> a R | #empty;
  rule N -> b R;
}
```

В итоге мы получим однозначную грамматику без левой рекурсии. Заметим сразу, что добавление к этой грамматике новых операций – очень простая задача: достаточно лишь дописать их в исходную грамматику и сопоставить правильные метаданные, все остальное произойдет автоматически.

Теперь обратимся к другим видам преобразований. Для начала извлечем из нашей грамматики классы для вершин синтаксического дерева. Базовые классы «Вершина», «Бинарная операция» и «Число» мы опишем вручную, а всех наследников создадим автоматически. Вот базовые классы (описание дается на языке `Emfatic` [21]):

```
abstract class Expression {}
abstract class BinaryExpr extends Expression {
  val left : Expression;
  val right : Expression;
}
class Num extends Expression {
  attr value : int;
}
```

Теперь напишем преобразование, строящее подклассы класса `BinaryExpr`:

```
map [Operation -> op:#term] {
  new EClass {
    name = op.grName;
    super += BinaryExpr;
  }
}
```

Это все, что нужно для создания всех подклассов `BinaryExpr` (в нашем примере их было бы пять – по одному на каждую операцию). Теперь осталось построить входное описание для какого-нибудь генератора синтаксических анализаторов. Для примера возьмем ANTLR. Нам нужно построить семантические акции, создающие объекты наших классов. Напомним, для генерации текста мы используем `StringTemplate`: все, что

находится внутри двойных угловых скобок – шаблон для генерации текста. Вот первое преобразование:

```
map [* |> sym:#nonterm] <<
  return <sym.name>;
>>
```

Для правил, в правой части которых стоит ровно один нетерминал, достаточно вернуть его. Для правил, в правой части которых стоит ровно один терминал (это операции), вернем новый объект:

```
map [Operation |> op:#term] <<
  return new <op.grName>();
>>
```

Когда операция встречается в инфиксной форме, запишем данные в ее атрибуты:

```
map [* |> left:#symbol op:#symbol right:#symbol] <<
  BinaryExpr result = <op.name>;
  op.setLeft(<left.name>);
  op.setRight(<right.name>);
  return result;
>>
```

Приведенный пример демонстрирует исключительную расширяемость, которая достигается за счет применения предлагаемого подхода. Изменения грамматики автоматически учитываются преобразованиями, что делает разработку гораздо проще. Система перестала быть монолитной: вся информация распределена по преобразованиям, причем ее можно повторно использовать. Это происходит благодаря гибкости модельно-ориентированной разработки и разделению различных аспектов системы.

## Заключение

Данная работа является представлением нового подхода к разработке *grammarware*, основанного на MDD и использующего принципы аспектно-ориентированного программирования для разделения различных аспектов системы.

В работе показано, как предлагаемый подход может изменить процесс разработки ПО, связанного с грамматиками: система становится более модульной, ее легче модифицировать и расширять. Сам процесс разработки сохраняется для повторного использования в виде преобразований.

Наш подход основывается на новых инструментальных средствах (предметно-ориентированных языках, описанных в данной работе), а также на изменении самого понятия разработки: артефакты не модифицируются вручную, а генерируются с помощью указанных инструментов.

Описанные принципы предстоит развивать и уточнять прежде, чем их смогут использовать разработчики промышленных систем, но уже сейчас понятно, что большинство проблем, описанных в работе [3], может быть решено в рамках предлагаемого подхода.

## Литература

1. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. – М.: Бином, 2001. – 560 с.
2. Мейер Б. Объектно-ориентированное конструирование программных систем / Пер. с англ. – М.: Русская редакция, 2005. – 1232 с.
3. Klint P., Lammel R., Verhoef C. Toward an engineering discipline for GRAMMARWARE // ACM Transactions on Software Engineering Methodology. – 2005. – Vol. 14:3. – P. 331–380.

4. Atkinson C., Kuhne T. Model-driven development: a metamodeling foundation // IEEE Software. – 2003. – Vol. 20:5. – P. 36–41.
5. Ledeczi A., Bakay A., Maroti M., Volgyesi P., Nordstrom G., Sprinkle J., Karsai, G. Composing domain-specific design environments // Computer. – 2001. – Vol. 34:11. – P. 44–51.
6. Parr T. The definitive ANTLR reference. – The Pragmatic Bookshelf, 2007.
7. Gagnon E., Hedren L. SableCC, an Object-Oriented Compiler Framework // Technology of Object-Oriented Languages and Systems, 1998.
8. Efftinge S. oAW xText: A framework for textual DSLs // Eclipse Summit 2006 Workshop: Modeling Symposium, 2006.
9. Jouault F., Bezivin J., Kurtev I. TCS: a DSL for the specification of textual concrete syntaxes in model engineering // Proceedings of the 5th Int. conf. on generative programming and component engineering, 2006.
10. Klint P. A Meta-Environment for generating programming environments // ACM Transactions on Software Engineering and Methodology. – 1993. – Vol. 2:2. – P. 176–201.
11. Clavel M., Duran F., Eker S., Lincoln P., Marti-Oliet N., Meseguer J., Talcott C. The Maude 2.0 System // Rewriting Techniques and Applications in Lecture Notes in Computer Science. – 2003. – Vol. 2706. – P. 76–87.
12. Kiczales G., Lamping J., Lopes C., Hugunin J., Hilsdale E., Boyapati C. Aspect-oriented programming. – Xerox Corporation, 2002.
13. Чарнецки К., Айзенекер У. Порождающее программирование: методы, инструменты, применение. – СПб: Питер, 2005. – 731 с.
14. Extended Backus-Naur Form / ISO/IEC 14977 : 1996(E) – ISO, 1996.
15. Фаулер М. UML. Основы. – М.: Символ-Плюс, 2006. – 192 с.
16. Ахо А., Сети Р., Ульман Дж. Компиляторы: принципы, технологии, инструменты. – М.: Вильямс, 2003. – 768 с.
17. Clark T., Sammut P., Willans J. Applied metamodeling: a foundation for language driven development. – Ceteva, 2008.
18. Steinberg D., Budinsky F., Paternostro M., Merks E. EMF: Eclipse Modeling Framework. 2<sup>nd</sup> ed. – Addison Wesley, 2007.
19. Jouault F., Kurtev I. On the architectural alignment of ATL and QVT // Proceedings of the 2006 ACM symp. on applied computing, 2006.
20. Parr T. Enforcing strict model-view separation in template engines // Proceedings of the 13th international conference on World Wide Web, 2004.
21. Daly C. Emfatic Language for EMF Development – IBM alphaWorks, 2004. – Режим доступа: <http://www.alphaworks.ibm.com/tech/emfatic>, своб.

**Бреслав Андрей Андреевич**

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, [abreslav@gmail.com](mailto:abreslav@gmail.com)

**Попов Игорь Юрьевич**

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор физ.-мат. наук, профессор, [popov@mail.ifmo.ru](mailto:popov@mail.ifmo.ru)

## ОСОБЕННОСТИ ПРОЕКТОВ С ОТКРЫТОЙ ПРОИЗВОДСТВЕННОЙ ДОКУМЕНТАЦИЕЙ НА ПРИМЕРЕ ОДНОПЛАТНОГО КОМПЬЮТЕРА

П.А. Косенков, Ю.А. Гатчин

Рассмотрена модель создания аппаратных решений, особенностью которой является изначальная доступность всей проектной документации на проектируемое устройство, на примере проекта открытого одноплатного компьютера. Показаны особенности проектирования электронных устройств с моментальной публикацией промежуточных вариантов в общедоступном виде, а также соответствующие механизмы. Рассматривается возможность использования репозитория Subversion с интеграцией программы по автоматической генерации документации. Такой подход позволяет в автоматическом режиме параллельно с созданием проекта создавать документацию, которая может служить обучающим материалом, а в дальнейшем развивается до полноценного коммерческого продукта. Подход позволяет не только экономить ресурсы, но и продвигать разработку, используя открытость как конкурентное преимущество.

**Ключевые слова:** открытая документация, проект, разработка, подход.

### Введение

Сфера разработки и производства радиоэлектронных устройств, в том числе и вычислительной техники, постоянно наращивает темпы. Это касается не только производственных мощностей, но и этапов разработки устройства. Этому способствует как совершенствование автоматизированных средств разработки, так и методов, позволяющих экономить ресурсы и применять уже готовые, отработанные и проверенные решения, требующие минимальных изменений для получения готовой разработки. Один из таких методов – применение в производственном цикле открытых разработок программного и аппаратного обеспечения.

### 1. Модели создания продукции

В мировой практике разработок новых устройств существует тенденция к использованию уже готовых открытых программных разработок. Такой подход обеспечивает экономию на этапе подготовки как программного, так и аппаратного обеспечения и сокращает производственный цикл на данном этапе создания системы, а привлечение сообщества программистов после выпуска устройства позволяет поддерживать актуальность разработки неопределенно долгое время благодаря личной заинтересованности каждого в совершенствовании конечного продукта.

В работе рассматривается модель разработки устройства с открытой исходной документацией. Открытыми в проекте являются как исходные программные коды, так и чертежи печатной платы. В разработке устройства есть неизменный набор этапов:

1. разработка,
2. производство готового устройства,
3. получение потребителем устройства и составление отзывов на него,
4. обратная связь (ОС) с разработкой через службы сервиса и поддержки.

Рассмотрим особенности их реализации в зависимости от принятой модели создания продукции.

#### 1.1. Закрытая модель

Эту модель использует большинство производителей и разработчиков электронных устройств. Ее особенностью является линейность всех этапов, причем каждый требует довольно серьезных денежных расходов. На рис. 1 представлена схема такого подхода. Производственная документация оседает у участников производственно цикла, оставляя

пользователю только инструкцию по пользованию. Это продиктовано, в первую очередь, довольно жесткой конкуренцией и исторически сложившимися патентными отношениями. Обратная связь в таком подходе реализуется с большой задержкой и начинает работать только после выхода продукта на рынок, а иногда вообще отсутствует.

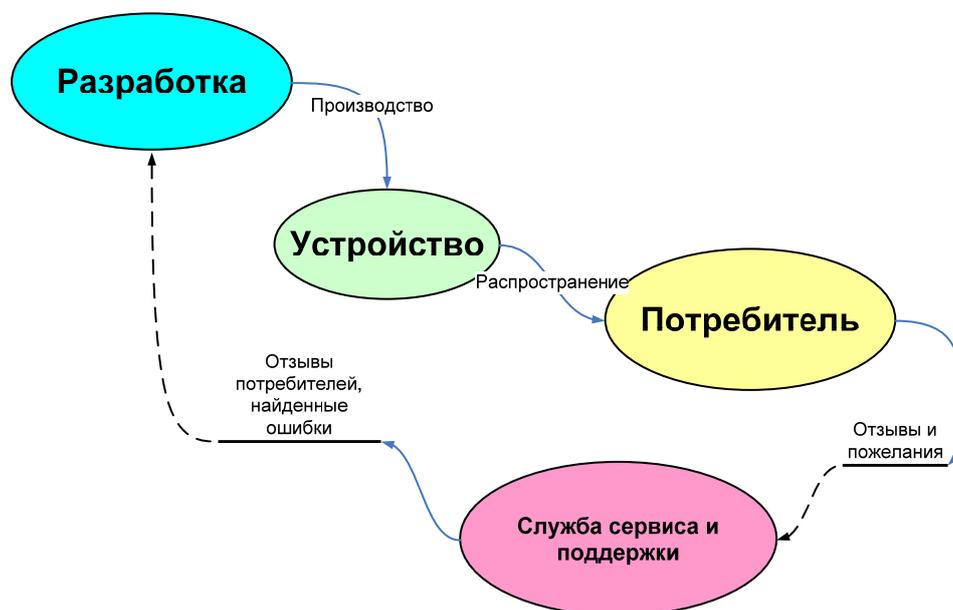


Рис.1. Пример реализации закрытой разработки

## 1.2. Открытая модель

В модели с применением открытых технологий образуются дополнительные обратные связи благодаря сообществу разработчиков, принимающих участие в разработке и модернизации устройства, в первую очередь по собственной инициативе. На рис. 2 показан пример модели, посвященный распространяемому продукту.

Основное отличие открытой модели заключается в том, что производственная документация находится в открытом доступе. С одной стороны, все технические преимущества, которые в предыдущей модели давали защиту от конкурентов, тут доступны всем, в том числе и самим конкурентам. Но, с другой стороны, такая модель позволяет организовать сообщество вокруг выпускаемого устройства – сообщество не просто пользователей, но и инженеров, которые могут на определенном этапе помочь с поиском и исправлением ошибок и добавлением новой функциональности. Такой подход позволит минимизировать издержки при проектировании устройства и последующей его поддержке, а в результате разработки останется сформированная база инженерных и кадровых ресурсов в виде наработанных программно-аппаратных решений и сообщества разработчиков, которые принимали участие в создании устройства.

Как показывает опыт, большинство участников сообщества – энтузиасты, желающие дополнить и/или переделать устройство под свои нужды, а процесс наращивания функционала довольно хаотичен. Чтобы направить инженерную деятельность в нужное русло, можно применять стимулирование, обозначив задачу и объявив поощрение за ее выполнение. Постепенно сообщество будет набирать опыт в виде описаний решений, исходных кодов и другой документации на устройство, который можно будет использовать в будущих разработках. Стоит заметить, что данное сообщество будет не только работать в инженерном плане, но и стимулировать распространение устройства, привлекая новых участников обширной документацией и расширяемой функциональностью устройства. Таким образом, использование открытых технологий позволяет существенно повлиять на этап разработки и обеспечить более продуктивную ОС.

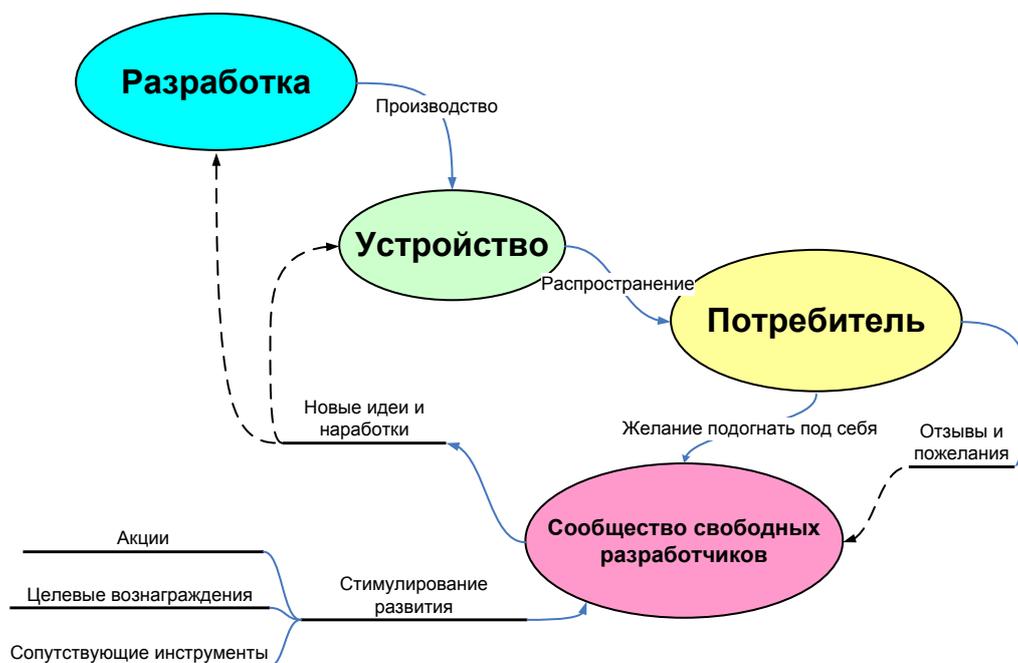


Рис. 2. Пример реализации открытой разработки

Отметим, что все вышеописанное применимо только к устройствам, подразумевающим довольно обширные возможности. В качестве примера был инициализирован проект открытого одноплатного компьютера на базе ARM (Advanced RISC Machine)-архитектуры.

## 2. Проект одноплатного компьютера на базе архитектуры ARM

В рамках открытого проекта [1] была начата работа над одноплатным компьютером на базе микропроцессора с архитектурой ARM9 (at91rm9200) [2]. В дальнейшем проект получил развитие на новом процессоре (at91sam9260), который имеет более совершенную архитектуру, что позволило реализовать ряд дополнительных возможностей.

### 2.1. Плата на базе микропроцессора архитектуры ARM

Общая функциональная схема компьютера представлена на рис. 3, где показаны все основные блоки и линии передачи информации между ними.

Ниже приведен перечень компонентов, использованных в системе:

- процессор ARM9 200MIPS, Atmel AT91SAM9260;
- оперативная память SDRAM 64МБ/128 МБ (две микросхемы по 16 бит), разрядность шины памяти 32 бита;
- встроенный NAND Flash на 256, 512 или 1024 Мб (Samsung K9F2G08U0A-P, возможна установка двух чипов);
- часы реального времени (Integrated RTC/Alarm and 64 Kb FRAM) на I2C;
- USB host 1 порт, 12 Мбит/с;
- USB device 1 порт, 12 Мбит/с;
- RS-232 (debug) + COM0(Порт модемный) + COM1(RS232) +1 RS485;
- Ethernet 10/100 (Micrel KS8721BL);
- IDE порт 40 pin (на шине памяти) или CF в IDE mode (3,3 TTL уровни на выход, совместимы с 5 В на вход, на шине – двунаправленные буферы с малой емкостью);





будет относительно долгим (сейчас проект занимает около 50 Мб), однако впоследствии будет происходить только обновление изменений.

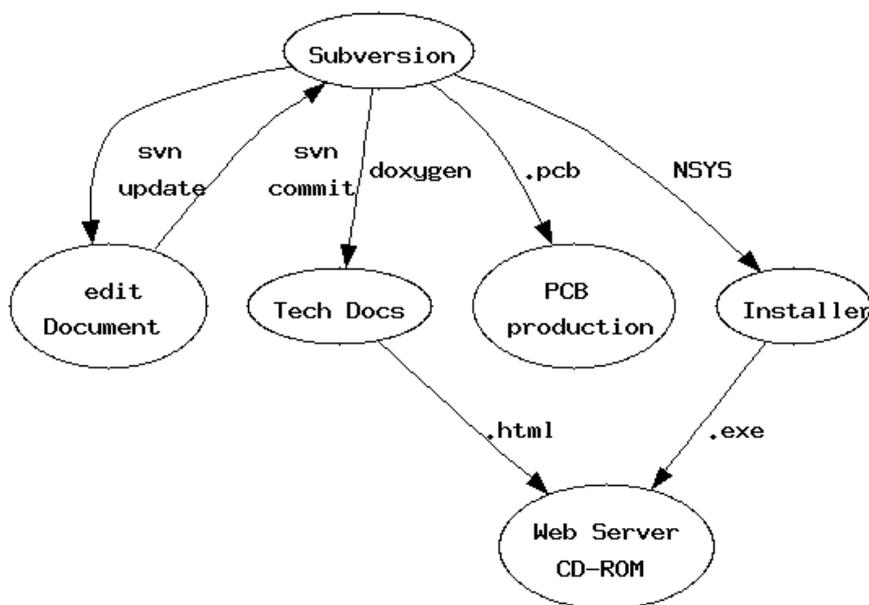


Рис. 5. Механизм работы над проектом

Для полноценной работы целесообразно дополнительно установить программы doxygen (средство автоматического формирования документации) и Graphviz (средство автоматической генерации блок-схем и графов). После установки этих компонентов можно будет создавать локальную документацию в виде html-страниц из исходных файлов проекта. Такая документация будет иметь структурированный вид по разделам и дополнительную справку по программному коду.

### Заключение

Представленный подход к созданию устройств на данный момент является одним из перспективных и интересных как для коммерческого направления, так и для образовательных целей. Несомненно, метод разработки устройств с открытой производственной документацией требует проработки. Именно этому вопросу автор статьи планирует посвятить будущие изыскания. В частности, проект одноплатного компьютера на базе ARM-архитектуры является пилотным проектом, на котором будут отработаны механизмы группового проектирования и создания документации, отладки предсерийных образцов и способы управления сообществом.

Описанный подход хорошо вписывается в учебную программу кафедры проектирования компьютерных систем при СПбГУ ИТМО. Совместно со студентами и преподавателями кафедры планируется запустить несколько подобных проектов в качестве учебного задания и для развития идеи открытой производственной документации.

### Литература

1. Открытый проект одноплатного компьютера на базе ARM микропроцессора. – Режим доступа: at91rm9200 <http://electronix.ru/forum/index.php?showforum=139>, своб.
2. Косенков П.А., Терентьев А.О. Особенности проектирования современных встраиваемых электронно-вычислительных систем и разработка плат для прототипирования // Научно-технический вестник СПбГУ ИТМО. – 2007. – Вып. 40. – С. 16–21.

3. Открытый одноплатный компьютер. – Режим доступа: <http://groups.google.com/group/arm9fpga-evolution-board?hl=ru>, своб.
4. Документация по файловому репозиторию Subversion. – Режим доступа: <http://svnbook.red-bean.com/>, своб.
5. Описание принципов работы файлового репозитория Subversion. – Режим доступа: <http://linux.fopf.mipt.ru/files/other/Subversion/ru/svn.basic.html>, своб.

*Косенков Павел Александрович*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, [microtrigger@gmail.com](mailto:microtrigger@gmail.com)

*Гатчин Юрий Арменакович*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, [gatchin@mail.ifmo.ru](mailto:gatchin@mail.ifmo.ru)

**УДК 535.417**

## **РАСЧЕТ ПОЛЯРИЗАЦИОННЫХ ЭЛЕМЕНТОВ УПРАВЛЕНИЯ В ИССЛЕДОВАНИЯХ МАССОПЕРЕНОСА СПЛОШНЫХ СРЕД**

**З.Г. Симоненко, О.Л. Студеникин, О.В. Елисеев**

Работа посвящена вопросам анализа расчета поляризационных элементов для приборов поляризационной интерферометрии с элементами нуль-эллипсометрии.

**Ключевые слова:** поляризационные элементы, моделирование, процесс переноса масс, эллипсометрия.

### **Введение**

Положения теории алгоритмизации в механике сплошных сред (МСС) были созданы для формализации и разработки общей методики построения математической модели. Разработка алгоритма такой модели позволяет создать программу, состоящую из подпрограмм для отдельных элементов прибора, которые можно заменить или скорректировать по результатам физического эксперимента более точными аналогами.

В оптико-физических исследованиях физическим носителем информации является электромагнитное излучение, состояние поляризации которого определяется ориентацией вектора напряженности электрического поля в плоскости волнового фронта и изменяется при взаимодействии со средой [1]. Наиболее действенными методами исследований состояния поляризации являются эллипсометрия пропускания и отражения поляризованных световых волн.

Приборы поляризационной интерферометрии с элементами нуль-эллипсометрии, применяемые в механике сплошных сред для изучения процесса массопереноса жидких бинарных систем, описаны в работе [2]. Наиболее высокие требования в этих приборах предъявляются к элементам управления ориентацией поляризованных составляющих интенсивности лазерного излучения – поляризаторам (анализаторам), в том числе отражающим, преломляющим, с двухлучепреломлением, дихроичным поляроидам, фазосдвигающим элементам (компенсаторам). Для их моделирования используются такие математические методы, как сферы Пуанкаре, векторы Стокса, Мюллера, Джонса [3].

Целью данной работы является анализ расчетов существующих математических моделей поляризационных элементов, используемых при изучении процесса массопереноса сплошных сред.

## Общие соотношения

При распространении монохроматической однородной бегущей плоской волны вдоль положительного направления оси  $z$  единичные векторы  $\tilde{u}_1$  и  $\tilde{u}_2$  параллельны положительным направлениям осей  $x$  и  $y$ , и электрический вектор такой волны описывается соотношением

$$E(z, t) = [\tilde{E}_x \cos(\omega t - \frac{2\pi}{\lambda} z + \delta_x)]\tilde{x} + [\tilde{E}_y \cos(\omega t - \frac{2\pi}{\lambda} z + \delta_y)]\tilde{y}, \quad (1)$$

где  $E_x$  и  $E_y$  – амплитуды линейных гармонических колебаний компонент электрического поля вдоль осей  $x$  и  $y$ ,  $\delta_x$  и  $\delta_y$  – фазы колебаний, а  $\tilde{x}$  и  $\tilde{y}$  – единичные векторы, соответствующие направлениям осей  $x$  и  $y$ . Описание воздействия поляризирующей оптической системы на проходящий свет выражают посредством связи вектора Джонса  $E_0$  на выходе оптической системы с вектором Джонса  $E_1$  на ее входе:

$$\begin{aligned} E_{0x} &= T_{11}E_{1x} + T_{12}E_{1y}, \\ E_{0y} &= T_{21}E_{1x} + T_{22}E_{1y}, \end{aligned} \quad (2)$$

где

$$T = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \quad (3)$$

– матрица Джонса для оптической системы, ее элементы  $T_{ij}$  – комплексные числа. Для определения состояния поляризации на выходе оптической системы получают вектор Джонса падающей волны и матрицу Джонса исследуемой оптической системы.

## Эллипсометрия пропускания

Простейшей матрицей Джонса является матрица, описывающая свободное распространение плоской световой волны на расстояние  $d$ . Матричное преобразование вектора Джонса описывается соотношением

$$\begin{bmatrix} E_{ox} \\ E_{oy} \end{bmatrix} = \begin{bmatrix} e^{-j2\pi d/\lambda} & 0 \\ 0 & e^{-j2\pi d/\lambda} \end{bmatrix} \begin{bmatrix} E_{ix} \\ E_{iy} \end{bmatrix}. \quad (4)$$

Фаза волны с длиной  $\lambda$  в вакууме изменяется на величину  $2\pi nd/\lambda$ . Здесь поляризационным элементом, *изотропным фазосдвигающим устройством* или *фазовой пластинкой* является плоскопараллельный слой с показателем преломления  $n$ , через который распространяется плоская волна [4].

В случае, когда волна распространяется не через изотропную, а через одноосную линейную двулучепреломляющую среду на расстояние  $d$ , описанный участок среды толщиной  $d$  действует как *линейное фазосдвигающее устройство*. Фаза компонент электрического вектора вдоль двух выбранных направлений, параллельного и перпендикулярного оптической оси, изменяется как  $2\pi n_e d/\lambda$  и  $2\pi n_o d/\lambda$ . Матрица Джонса для такого линейного фазосдвигающего устройства имеет вид

$$T^1 = \begin{bmatrix} e^{j\delta_1} & 0 \\ 0 & e^{j\delta_2} \end{bmatrix}, \quad (5)$$

где

$$\delta = 2\pi \frac{(n_e - n_o)d}{\lambda}, \quad (6)$$

$n_e$ ,  $n_o$  – необыкновенный и обыкновенный показатели преломления среды,  $d$  – толщина среды,  $\lambda$  – длина падающей волны. Обратная матрица Джонса будет иметь вид

$$T^1 = \begin{bmatrix} e^{j2\pi n_e d/\lambda} & 0 \\ 0 & e^{j2\pi n_0 d/\lambda} \end{bmatrix}, \quad (7)$$

а дифференциальная матрица распространения вектора Джонса –

$$N(z) = \begin{bmatrix} -j2\pi n_e/\lambda & 0 \\ 0 & -j2\pi n_0/\lambda \end{bmatrix}. \quad (8)$$

Подставляя значения элементов этой матрицы, можно получить искомое состояние поляризации выходящей волны. На практике для получения линейного фазосдвигающего устройства изготавливается плоскопараллельная пластинка из кристалла, в котором оптическая ось параллельна входной и выходной грани. В рамках НИР «Разработка неразрушающих бесконтактных оптических методов исследования стереометрии внутренних структурных дефектов элементной базы микроэлектроники и микросенсорной техники» была создана программа, моделирующая изменение состояния поляризованного света при прохождении через линейную двулучепреломляющую пластину, с использованием Turbo C 3.0 и Borland C++ 5.02 [5].

Проекция вектора Джонса на старые и новые оси при повороте системы координат относительно оси  $z$ , вдоль которой распространяется световая волна, связаны между собой следующими соотношениями:

$$\begin{aligned} E_{x'} &= \cos(\alpha)E_x + \sin(\alpha)E_y, \\ E_{y'} &= -\sin(\alpha)E_x + \cos(\alpha)E_y. \end{aligned} \quad (9)$$

В матричной форме это соотношение будет выглядеть следующим образом:

$$\begin{bmatrix} E_{x'} \\ E_{y'} \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} E_x \\ E_y \end{bmatrix} \quad (10)$$

или

$$E' = R(\alpha)E. \quad (11)$$

Преобразование матрицы Джонса при повороте системы координат относительно оси  $z$  выглядит следующим образом:

$$E = \frac{1}{\chi}. \quad (12)$$

Умножая матрицу Джонса на матрицу поворота  $R(\alpha)$ , получаем матрицу

$$T = \begin{bmatrix} \cos^2(\alpha e^{j\frac{\delta}{2}}) + \sin^2(\alpha e^{-j\frac{\delta}{2}}) & 2j \sin(\alpha) \cos(\alpha) \sin(\frac{\delta}{2}) \\ 2j \sin(\alpha) \cos(\alpha) \sin(\frac{\delta}{2}) & \sin^2(\alpha e^{j\frac{\delta}{2}}) + \cos^2(\alpha e^{-j\frac{\delta}{2}}) \end{bmatrix}, \quad (13)$$

где  $\delta$  – разность фаз, вносимая пластиной. В рамках НИР «Интегрированная система автоматизированного производства оптических материалов» была создана программа, написанная с использованием Borland C++5.0, моделирующая работу линейных фазосдвигающих пластинок  $(\lambda/2)$  и  $(\lambda/4)$  [6].

В одноосной линейной дихроичной среде затухание линейно поляризованной волны, распространяющейся перпендикулярно оптической оси (поглощения), зависит от направления колебаний поперечного электрического поля относительно оптической оси. Вводятся «необыкновенный»  $k_e$  и «обыкновенный»  $k_o$  коэффициенты поглощения среды для света, линейно поляризованного параллельно и перпендикулярно оптической оси. Величина  $(k_e - k_o)$  является мерой дихроизма среды.

Для участка среды толщиной  $d$  матрица Джонса имеет вид

$$T = e^{-j2\pi dn/\lambda} \begin{bmatrix} e^{-j\pi dk_o/\lambda} & 0 \\ 0 & e^{-j\pi dk_o/\lambda} \end{bmatrix}, \quad (14)$$

где внешний скалярный множитель описывает изотропную задержку по фазе. В наиболее общем случае плоскопараллельная пластинка из одноосного линейного двулучепреломляющего и дихроичного материала, в котором оптические оси двулучепреломления и дихроизма совпадают и параллельны внешним плоским граням, действует как *линейно-дихроичное фазосдвигающее устройство*. Матрица Джонса такого устройства имеет вид

$$T = \begin{bmatrix} e^{-j2\pi dn_e} e^{-j\pi dk_e/\lambda} & 0 \\ 0 & e^{-j2\pi dn_o} e^{-j\pi dk_o/\lambda} \end{bmatrix}. \quad (15)$$

Если волна распространяется перпендикулярно поверхности пластинки, а ось  $x$  системы координат перпендикулярна оптической оси, матрица Джонса для такого устройства имеет вид

$$T = \begin{bmatrix} e^{-j2\pi dn_e} e^{-j\pi dk_e/\lambda} & 0 \\ 0 & e^{-j2\pi dn_o} e^{-j\pi dk_o/\lambda} \end{bmatrix}. \quad (16)$$

При наличии нескольких оптических устройств последовательно перемножаются все матрицы Джонса этих устройств, а затем матрица  $T_{\text{комб}}$  умножается на вектор Джонса. Из получившегося вектора Джонса  $E_0$  находят параметры эллипса поляризации:

$$A = \sqrt{|E_x|^2 + |E_y|^2}, \quad (17)$$

$$\theta = \frac{E_y/E_x \times \cos(\delta_y - \delta_x)}{1 - E_y/E_x}, \quad (18)$$

$$\varepsilon = \frac{E_y/E_x \times \sin(\delta_y - \delta_x)}{1 + E_y/E_x}. \quad (19)$$

В рамках НИР «Разработка неразрушающих бесконтактных оптических методов исследования стереометрии внутренних структурных дефектов элементной базы микроэлектроники и микросенсорной техники» построена математическая модель взаимодействия световой волны с дихроичной пластинкой, создана программа с использованием Borland C++ [7].

*Интерференционно-поляризационные фильтры* (ИПФ) находят большое применение в связи с дороговизной поляризационных призм. В ИПФ оси поляризатора и анализатора устанавливают параллельно, оптическая ось кристаллической пластинки составляет с ними угол  $45^\circ$ ,  $\beta - \alpha = 0$ . В результате интерференции лучей после прохождения пластинки  $K$  пропускание ее описывается формулой

$$T = \frac{J}{J_0} = \cos^2\left(\pi \frac{(n_e - n_o)d}{\lambda}\right), \quad (21)$$

где  $d$  – толщина пластинки  $K$ .

Сложный узкополосный ИПФ составлен из стопы  $N$  одинаково ориентированных двулучепреломляющих плоскопараллельных кристаллических пластин, толщина которых возрастает в геометрической прогрессии:  $d, 2d, \dots, 2^{N-1}d$ . Пропускание сложного ИПФ вычисляется как произведение пропусканий соответствующих простых ИПФ:

$$T = \prod_{k=1}^N T_k = \prod_{k=1}^N \cos^2\left(\frac{2^{(N-1)}(n_e - n_o)d\pi}{\lambda}\right). \quad (22)$$

Результирующая интенсивность всех лучей описывается формулой

$$J = \frac{J_0 \sin^2(2^{(N)} \delta_1)}{2^{(N)} \sin^2(\delta_1)}, \quad (24)$$

где  $J_0$  – интенсивность падающего света, число минимумов определяется условием  $n = 2^{(N)} - 1$ , а вторичные максимумы создают паразитный фон, который влияет на выходную световую волну. В работе [8], выполненной в рамках НИР «Разработка неразрушающих бесконтактных оптических методов исследования стереометрии внутренних структурных дефектов элементной базы микроэлектроники и микросенсорной техники» исследован сложный ИПФ и представлена программа для его расчета с использованием Borland Turbo C 3.0.

Оценки систематических погрешностей поляризационных элементов для измерительных схем приборов для изучения процесса массопереноса жидких бинарных систем [2] представлены в табл. 1.

Оптический элемент	Тип погрешности оптического элемента	Тип схемы		
		№1	№2	№3
Пластина из исландского шпата	Наклон, с	2	2	2
	Клиновидность, с	2	4	4
	Температурная нестабильность, °С	1	1	1
	Разнотолщинность ( $\times 10^{-4}$ ), м	1	2	2
Четвертьволновая пластина ( $\lambda/4$ )	Наклон, с	2	2	3
	Разворот к пучку, с	2	2	3
	Отклонение анализатора в паре ( $\lambda/4-A$ ), с	1	1	1
Полуволновая пластина ( $\lambda/2$ )	Наклон, с	1	1	2
	Разворот к пучку, с	1	1	2
	Отклонение анализатора в паре ( $\lambda/2-A$ ), с	-	-	2

Таблица 1. Систематические погрешности поляризационных элементов

### Эллипсометрия отражения

Результат взаимодействия падающей волны с границей раздела проявляется в изменении амплитудного коэффициента отражения, и при изменении фазы отраженной волны при коэффициенте отражения, равном единице, между компонентами отраженной волны возникает разность фаз эллиптически поляризованной отраженной волны. Зависимость разности фаз составляющих отраженной волны от угла падения при  $\delta_p = 2\alpha$  имеет вид [9]

$$\delta_s = \arctg \left[ \frac{\sin 2\alpha}{\left\{ \left( \frac{n_1}{n_2} \right) \cdot \sin(\alpha) \right\}^2 - \left\{ \left( \frac{n_2}{n_1} \right) \cdot \cos(\alpha) \right\}^2} \right]. \quad (25)$$

Основным фактором, вносящим вклад в обратное отражение, является френелевское отражение  $R_F$  вследствие воздушного зазора  $S$  между торцами волокон. Общее выражение для коэффициента отражения имеет вид

$$R_F = \frac{(n_1^2 - n^2)^2 \sin^2 \left( \frac{2\pi n S}{\lambda} \right)}{4n_1^2 n^2 + (n_1^2 - n^2)^2 \sin^2 \left( \frac{2\pi n S}{\lambda} \right)}, \quad (26)$$

где  $\lambda$  – длина волны лазерного излучения,  $n_1$  – показатель преломления сердцевинны волокна,  $n$  – показатель преломления среды в зазоре. Непараллельность торцов приводит к разным значениям зазора  $S$  для разных участков сердцевинны, а потери на отражение определяются как

$$b_F = -10 \lg \bar{R}_F = -10 \lg \frac{(n_1 - n)^2}{(n_1 + n)^2}. \quad (27)$$

Расчет зависимостей программно реализован средствами Borland C++5.0 и Turbo C++3.0 [10]. В большинстве рассмотренных систем оптического моделирования [11, 12] используется язык C++, который обеспечивает высокую эффективность в вычислительных приложениях.

Работа выполнена в рамках НИР «Разработка методов исследования наноструктурированных оптических материалов».

### Заключение

Рассмотрены расчеты существующих (в том числе предложенных ранее авторами) моделей поляризационных элементов приборов, применяемых в эллипсометрии пропускания и отражения для целей изучения процесса массопереноса в жидких средах. Для проверки правомерности применения математических описаний приведена сводная таблица с результатами расчетов этого класса задач, которая подтверждает адекватность и достоинства рассмотренных моделей. Показаны новые возможности применения практических расчетов поляризационных элементов управления данного типа для решения задач автоматизированного контроля процесса массопереноса сплошных сред.

### Литература

1. Москалев В.А. Теоретические основы оптико-физических исследований. – Л: Машиностроение, 1987. – 318 с.
2. Симоненко З.Г., Ткалич В.Л. Разработка информационно-измерительной системы неразрушающего контроля параметров массопереноса в жидкой среде с границей раздела. – СПб: СПбГУ ИТМО, 2006. – 120 с.
3. Аззам Р., Башара Н. Эллипсометрия и поляризованный свет. – М.: Мир, 1981.
4. Панков Э.Д., Коротаев В.В. Поляризационные угломеры. – М.: Недра, 1992.
5. Симоненко З.Г., Бандура А.С. Расчет линейного фазосдвигающего устройства // Диагностика и функциональный контроль качества оптических материалов. – СПб: СПбГУ ИТМО, 2004. – С. 199–204.
6. Симоненко З.Г., Скобелин А.А. Расчет линейной двулучепреломляющей пластинки // Диагностика и функциональный контроль качества оптических материалов. – СПб: СПбГУ ИТМО, 2004. – С. 205–211.
7. Симоненко З.Г., Антипин Д.А. Расчет линейной дихроичной пластины // Диагностика и функциональный контроль качества оптических материалов. – СПб: СПбГУ ИТМО, 2004. – С. 219–229.
8. Симоненко З.Г., Лобанов В.А. Расчет интерференционно-поляризационного светофильтра // Диагностика и функциональный контроль качества оптических материалов. – СПб: СПбГУ ИТМО, 2004. – С. 192–198.
9. Симоненко З.Г., Студеникин О.Л. Расчет параметров эллиптически поляризованной волны отражающей среды для исследования параметров однородного волновода // Научно-технический вестник СПб ГУ ИТМО. – 2006. – Вып. 29. – С. 260–262.
10. Симоненко З.Г., Ваняев В.Н. Расчет оптических параметров отражающей среды методами эллипсометрии // Диагностика и функциональный контроль качества оптических материалов. – СПб: СПбГУ ИТМО, 2004. – С. 212–218.
11. Волобой А.Г., Галактионов В.А., Ершов С.В., Жданов Д.Д. Оптические элементы как средство расширения функциональности программ оптического моделирования

- // Труды 16-ой Междунар. конф. по компьютерной графике и ее приложениям – Графикон-2006. – Россия, Новосибирск, 1–5 июля 2006 г. – С. 182–191.
12. Симоненко З.Г., Ткалич В.Л. Использование программ численного решения некоторых задач эллипсометрии в учебном процессе // Труды конф. «Оптика и образование». – СПб, 16–17 октября 2002 г.

*Симоненко Зинаида Григорьевна*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, ZGSim@yandex.ru

*Студеникин Олег Леонидович*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, инженер, oleg-studenikin@yandex.ru

*Елисеев Олег Валерьевич*

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, ov1983@yandex.ru

**УДК 004.89; 681.3**

## **ИНФОРМАЦИОННОЕ СОПРОВОЖДЕНИЕ ОБРАБОТКИ ЗАЯВОК В СЛУЖБЕ ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ**

**Н.Ф. Гусарова, Р.В. Иванов, А.Е. Михайленко**

Рассмотрены проблемы организации службы технической поддержки компьютерного оборудования в фирмах среднего масштаба (или отдельных подразделениях крупных компаний). Описывается разработанная структура хранения данных для информационной системы сопровождения приема и обработки заявок.

**Ключевые слова:** служба технической поддержки, обработка заявки, CMDB, техническая диагностика.

### **Введение**

Массовая компьютеризация бизнес-процессов, актуальная как для крупных компаний, так и для небольших фирм, порождает, помимо очевидных плюсов, и целый ряд проблем. Предметом рассмотрения в данной статье являются проблемы, связанные с организацией службы технической поддержки (СТП) компьютерного оборудования в фирмах среднего масштаба (или отдельных подразделениях крупных компаний). Производится анализ существующих подходов к организации СТП с учетом факторов неопределенности, характерных для таких фирм, описывается разработанная авторами структура хранения данных для информационной системы сопровождения приема и обработки заявок в такой СТП.

### **Анализ подходов к организации СТП**

Будем называть совокупность пользователей, эксплуатируемого ими компьютерного оборудования и поддерживающей инфраструктуры модерируемой системой (МС). Как показывает практика, при массовой компьютеризации бизнеса для МС характерны разноплановый контингент пользователей с различными квалификацией и опытом работы на конкретном оборудовании, а также широкий и постоянно обновляемый модельный ряд эксплуатируемых устройств и программных продуктов.

Рассматривая МС как сложную техническую систему (СТС) [1–3], правомерно ставить задачу построения СТП как задачу технической диагностики [4, 5]. Традиционной [1, 4] для таких задач является теоретико-множественная трактовка системы как отношения на множествах, т.е. в виде кортежа

$$D = \langle T, X, Y, Z, F, L \rangle. \quad (1)$$

Здесь  $T$  – множество моментов времени, в которые наблюдается система;  $X, Y$  – множество входных и выходных сигналов, соответственно;  $Z$  – множество состояний системы, а операторы переходов  $F$  и выходов  $L$  реализуют отображения

$$F: T \times X \times Z \rightarrow Z; \quad (2)$$

$$L: T \times X \times Z \rightarrow Y. \quad (3)$$

Для получения конструктивных результатов вводятся следующие допущения:

- (1.а) наблюдаемость МС – существование отображения (3), которое при фиксированных  $\tau \in T$  (здесь  $\mathcal{F} = \{\tau\} \subset T$ ) и  $\mathbf{x}(t) \in X$  обеспечивает возможность на основе известного выходного процесса  $\mathbf{y}(t) \in Y$  определить неизвестные состояния объекта  $\mathbf{z}(t) \in Z$ ;
- (1.б) агрегирование – возможность разбиения бесконечного, в общем случае, множества состояний  $Z$  МС на конечное и небольшое число классов  $E$  – множество заданных видов технического состояния (ТС).

При этих допущениях задача технической диагностики МС моделируется обобщенной диаграммой

$$\begin{array}{ccccc} T \times X \times Z & \xrightarrow{L} & Y & \xrightarrow{\eta} & E & \xrightarrow{\psi} & S \\ & & \searrow \Theta & & \downarrow \kappa & \nearrow \xi & \\ & & & & Y/Q & & \end{array} \quad (4)$$

Здесь

$$\eta: Y \rightarrow E \quad (5)$$

– отношение идентификации, которое содержательно состоит из отношения классификации, т.е. разбиения (факторизации) множества  $Y$  на ряд непересекающихся классов (например, состояния, вызванные дефектом  $i$ -й подсистемы),

$$\Theta: Y \rightarrow Y/Q, \quad (6)$$

и соотнесения полученного фактор-множества  $Y/Q$  с множеством  $E$ , т.е.

$$\kappa: E \rightarrow Y/Q, \quad (7)$$

$$\psi: E \rightarrow S \quad (8)$$

– отношение, учитывающее вероятностные характеристики возможных ошибок при контроле, а  $\xi$  – отображение, уточняющее сформированное фактор-множество  $Y/Q$  по результатам контроля. Задачи типа (4) решаются сравнительно легко, в том числе даже аналитически, в рамках сформулированных выше допущений (1.а) и (1.б) и при условии априорно известных отображений (6) и (7).

Альтернативу подходу технической диагностики предоставляет подход библиотеки ИТЛ (англ. Information Technology Infrastructure Library) [6], в которой деятельность СТП рассматривается как составная часть управления ИТ-инфраструктурой предприятия [7, 8]. Базовым понятием здесь является инцидент – любое событие, не являющееся частью стандартных операций по предоставлению услуги, которое привело или может привести к снижению качества этой услуги. Согласно ИТЛ, для управления инцидентами в состав системы управления ИТ-инфраструктуры предприятия должны входить:

- (2.а) база данных управления конфигурациями (Configuration Management Database, CMDB) – репозиторий технологических данных об объектах ИТ-инфраструктуры, их взаимосвязях и истории изменения, сущностями которой являются, в частности, ИТ-сервисы, компьютерное оборудование, программное обеспечение, ролевые функции акторов МС, используемые формы документации;
- (2.б) база знаний по проблемам/известным ошибкам, описывающая способ распознавания инцидентов, имеющиеся решения и обходные пути;
- (2.в) система регистрации, отслеживания и мониторинга инцидентов.

Очевидно, что компоненты (2.а)–(2.в) на концептуальном уровне призваны обеспечить механизм ведения метаданных об ИТ-инфраструктуре и построить интегрированную платформу управления МС. Однако на технологическом уровне такие обобщенные

системы оказываются чрезвычайно сложными, и, как показывает анализ решений ведущих компаний (Hewlett-Packard [9], IBM [10], Microsoft [11]), концепция ИТП практически реализуется с существенными ограничениями, в том числе:

- (3.а) поддерживаемые ИТ-сервисы конфигурируются с точностью до терминальных сущностей CMDB (1), и отклонения от предписанной конфигурации не допускаются;
- (3.б) на всех системных уровнях (организационном, программном, аппаратном) пресекается активность акторов МС по внесению изменений в конфигурацию CMDB, а также по несанкционированному выходу во внешние сети.

По существу, тем самым реализуются рассмотренные выше допущения наблюдаемости (1.а) и агрегирования (1.б) МС, характерные для систем технической диагностики. Однако упомянутые условия и допущения для многих МС оказываются неправомерными, что связано, в первую очередь, с системной сложностью МС. Наиболее важные проявления системной сложности МС по их генезису можно разделить на две группы – системная сложность технологической среды и системная сложность человеческого фактора (поведения акторов) в составе МС. Рассмотрим их подробнее.

*Системная сложность программных продуктов как технологической среды* была констатирована еще в 1975 г. [12]. Факторы, обуславливающие эту сложность, специфичны для каждого программного продукта и для конкретной конфигурации МС, причем с развитием ИТ их номенклатура постоянно растет. Приведем некоторые примеры таких факторов:

- у программных продуктов число возможных состояний на порядки величин превышает число состояний компьютеров [12];
- подавляющее большинство компьютеров включены в сети, т.е. работают в режиме открытой системы;
- процессы, происходящие в среде проприетарных программных продуктов ниже уровня API, недоступны для контроля внешнему пользователю;
- многие действия, выполняемые пользователями в технологической среде, являются принципиально необратимыми [13].

В результате отображение  $L$  (3) теряет инъективность, а МС в целом становится не вполне наблюдаемой, т.е. допущение (1.а) не выполняется.

*Системная сложность поведения акторов в составе МС* является ответом на неполную наблюдаемость технологической среды, в силу которой полная и объективная интерпретация имеющих место технологических ситуаций актору недоступна. Для работы в такой среде каждый актер строит свою систему интерпретаций, которая зависит от текущего контекста, квалификации и предыдущего опыта актора, «встроена в его культурную матрицу» [12] и во многом формируется на уровне глубинных знаний и навыков [14], т.е. является неявной (имплицитной) даже для него самого. Например, типичная фраза пользователя «Не работает Интернет» может соотноситься с огромным спектром возможных технических состояний МС. В результате отношение классификации (6) реализуется не на уровне разбиения, а только на уровне покрытия (с возможностью пересечений), отношение (7) перестает быть взаимно однозначным и, соответственно, диаграмма (4) в целом также теряет однозначность. Кроме того, для многих МС требования (3.а)–(3.б) могут оказаться неприемлемыми по экономическим или организационным соображениям.

Выходом из создавшегося положения могло бы быть использование в СТП только специалистов с высокой квалификацией и большим опытом работы, которые уже сформировали для себя (чаще всего на имплицитном уровне) спектр отображений (6) и (7), покрывающий все (или большинство) ТС конкретной МС. Однако сложность такого решения очевидна (хотя бы из экономических соображений). Таким образом, при организации СТП компьютерного оборудования в фирмах среднего масштаба возникает следующая задача: создать систему класса экспертной (или системы поддержки принятия решений) для информационного сопровождения работы СТП, которая бы акку-

мулировала опыт квалифицированных специалистов и на этой базе позволила осуществлять большинство операций ТП специалистам более низкой квалификации и/или даже (в определенных пределах) самим акторам МС.

### Разработка структуры CMDB для СТП

Приведенные соображения позволяют сформулировать требования к структуре CMDB для СТП с учетом характерных для фирм факторов неопределенности:

- (4.а) логическая модель объектов ИТ-инфраструктуры, их взаимосвязей и истории изменения должна быть максимально обобщенной (нейтральной) относительно текущих технологий их использования;
- (4.б) структура хранения данных должна быть удобна для использования в качестве декларативной части базы знаний создаваемой ИС СТП и в то же время легко адаптироваться к возможным изменениям процедурной части ИС СТП (механизма логического вывода), отражающим специфику поддерживаемого бизнес-процесса;
- (4.в) структура хранения данных должна быть адаптируемой к текущим изменениям политики ограничений и уровня неопределенностей, принятых в конкретном бизнес-процессе.

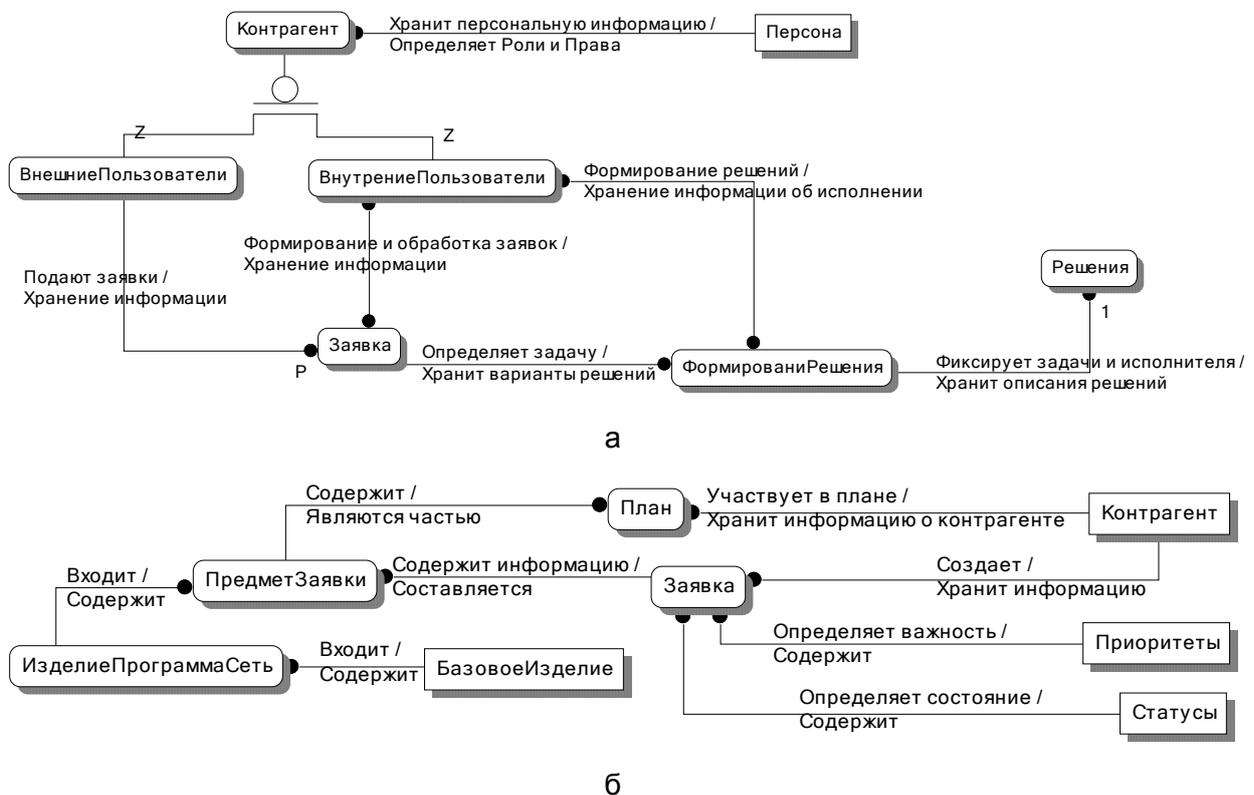


Рис. 1. а – ER-диаграмма базовых сущностей относительно акторов, работающих с заявками; б – ER-диаграмма базовых сущностей, связанных с жизненным циклом заявки

В соответствии с указанными требованиями разработана структура CMDB. На рис. 1, а, б, представлена логическая структура CMDB верхнего уровня, на рис. 2 – инфологическая модель CMDB разрабатываемой системы.

Требование (4.а) обеспечивается (рис. 1, а, рис. 2) за счет возможности задания любой Персоне, с одной стороны, любого статуса и/или Должности, с другой стороны, Роли. При этом поддерживается иерархия должностей и ролей и их динамическая смена в процессе функционирования ИС СТП. Кроме того, система хранения данных о сущностях ИзделиеПрограммаСеть и БазовоеИзделие (рис. 1, б) поддерживает как ие-

рархию, так и принципиальные изменения номенклатуры эксплуатируемых устройств и программных продуктов.

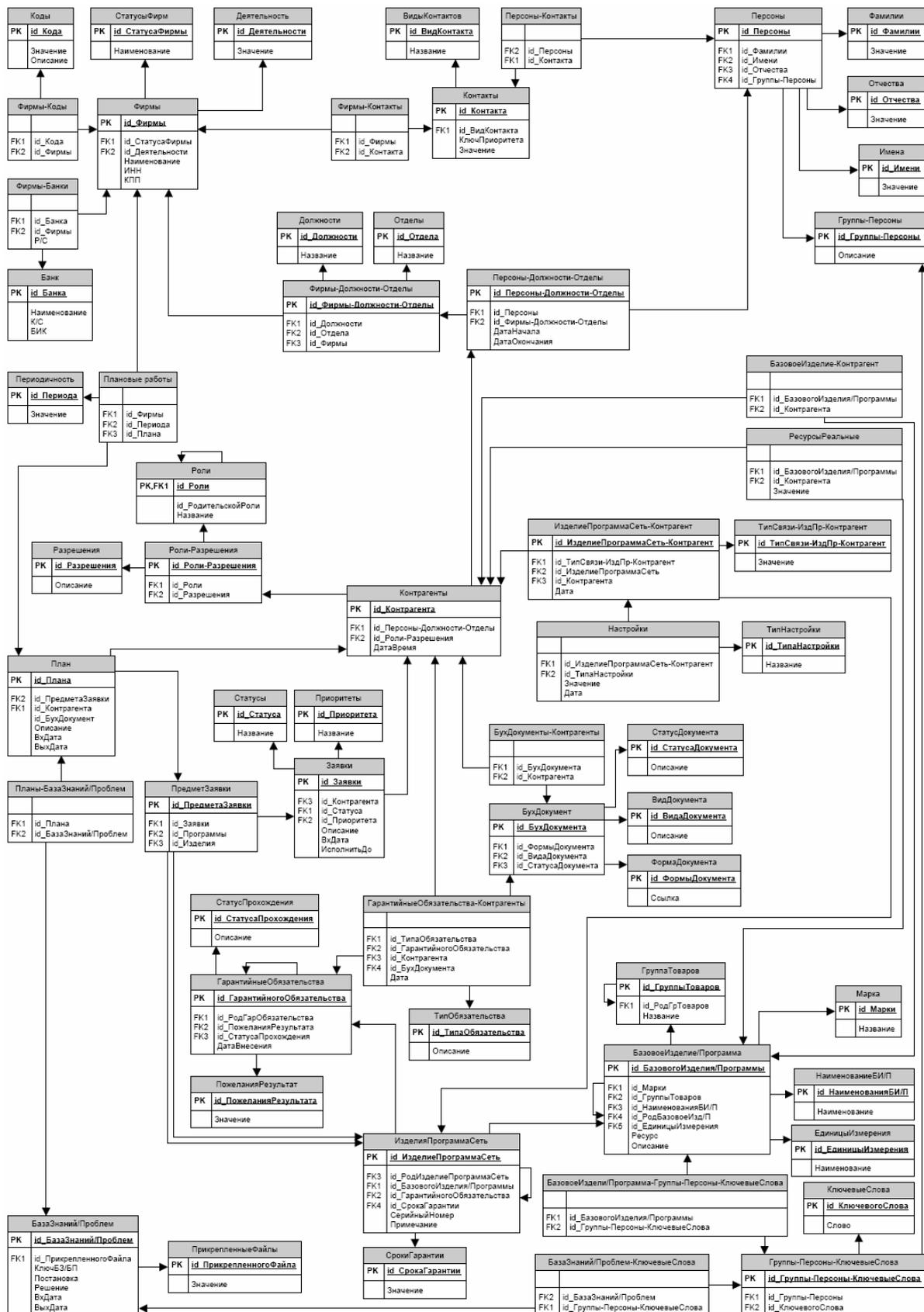


Рис. 2. Инфологическая модель базы данных

Рассмотрим обеспечение требования (4.б). В структуре CMDB выделена декларативная часть, обеспечивающая хранение информации о Персонах, Фирмах, изделиях, программах, сетевых настройках (Изделие/Программа/Сеть) (рис. 2). Процедурная компонента ИС СТП поддерживается частью CMDB, в которую входят такие сущности, как Заявка, Предмет/Заявки, План, База/Знаний/проблем, Ключевые/Слова, причем эта часть CMDB может функционировать и модифицироваться независимо от декларативной части. Кроме того, структура CMDB поддерживает различные типы связей, в том числе иерархические, ассоциативные и каузальные [15].

Адаптация структуры хранения данных (требование 4.в) обеспечивается в CMDB посредством различных механизмов. За счет введения сущности Контрагент парируются текущие изменения политики ограничений и изменений структурных связей с привязкой по времени. Адаптация CMDB по уровню неопределенности бизнес-процесса и первоначальной информации о нем производится за счет разделения декларативной и процедурной частей ИС СТП (см. выше). Кроме того, предусмотрена возможность удобного динамического расширения номенклатуры характеристик сущности Базовое-Изделие/Программа на различных иерархических уровнях.

В качестве задач дальнейшего исследования авторы видят:

- изучение возможных эвристических методов и механизмов обработки заявок на младших уровнях поддержки СТП [16];
- обоснование механизмов и критических параметров заполнения CMDB;
- формирование технологий портирования CMDB на другие бизнес-процессы;
- расширение структуры CMDB для поддержки полного цикла обработки заявки, в частности, в финансовом аспекте.

### **Заключение**

Рассмотрены проблемы, связанные с организацией СТП компьютерного оборудования в фирмах среднего масштаба (или отдельных подразделениях крупных компаний). Выявлены характерные для таких фирм факторы неопределенности системной и организационной природы. Показана ограниченность существующих подходов к организации таких СТП, описывается разработанная авторами структура хранения данных для информационной системы сопровождения приема и обработки заявок в подобных СТП. Поставлены задачи, требующие дальнейшей разработки.

### **Литература**

1. Охтилев М.Ю., Соколов Б.В., Юсупов Р.А. Интеллектуальные технологии мониторинга и управления структурной динамикой сложных технических объектов. – М.: Наука, 2006. – 410 с.
2. Перегудов Ф.И., Тарасенко Ф.П. Введение в системный анализ. – М.: Высш. шк., 1989. – 367 с. ил.
3. Иванов Р.В., Маятин А.В., Михайленко А.Е. Моделирование процесса обработки заявок в службе технической поддержки сложных технических систем // Научно-технический вестник СПбГУ ИТМО. – 2007. – Вып. 44. – С. 268–274.
4. Дмитриев А.К., Мальцев П.А. Основы теории построения и контроля сложных систем. – Л.: Энергоатомиздат, 1988. – 192 с.
5. Иванов Ю.П., Никитин В.Г., Чернов В.Ю. Контроль и диагностика измерительно-вычислительных комплексов. – СПб: СПбГУАП, 2004. – 98 с.
6. The Official ITIL ® Website [Electronic resource]. – Electronic data. – APM Group Ltd., cop. 2007-2008. \– Режим доступа: <http://www.itil-officialsite.com/>, своб.

7. Service Support Book – ITIL® Version 2. – London: Office of Government Commerce (OGC): TSO (The Stationery Office), 2000. – 312 p.
8. Service Delivery Book – ITIL® Version 2. – London: Office of Government Commerce (OGC): TSO (The Stationery Office), 2001. – 382 p.
9. Колесов А. ИТSM и эффективность обслуживания информационных систем предприятий. – Режим доступа: <http://www.bytemag.ru/?ID=602758>, своб.
10. Система управления ИТ-услугами BMC Remedy IT Service Management. – Режим доступа: <http://www.bmc.com/ru>, своб.
11. MOF – Microsoft Operations Framework. – Режим доступа: <http://www.itsmportal.ru/articles/it-control/2003-12-15%2000:00:00-26.html/>, своб.
12. Брукс Ф. Мифический человеко-месяц, или Как создаются программные системы. – М.: Символ-Плюс, 2006. – 304 с.
13. Гусарова Н.Ф., Маятин А.В., Смирнов Ф.А. Обратные задачи в компьютеризированных технологических средах // Научно-технический вестник СПбГУ ИТМО. – 2007. – Вып. 44. – С. 284–294.
14. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб: Питер, 2000. – 384 с.
15. Джексон П. Введение в экспертные системы. – М.: Вильямс, 2001. – 624 с.
16. Иванов Р.В., Михайленко А.Е., Гусарова Н.Ф. ИТSM в ИТIL – структурно-образующий подход к проектированию, внедрению и управлению ИТ-системами класса Help (Service) Desk // Научно-технический вестник СПбГУ ИТМО. – 2008. – Вып. 48. – С. 217–226.

***Гусарова Наталия Федоровна***

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, [natfed@list.ru](mailto:natfed@list.ru)

***Иванов Роман Владимирович***

Санкт-Петербургский государственный университет информационных технологий, механики и оптики, ассистент, [roman@nevsky30.ru](mailto:roman@nevsky30.ru)

***Михайленко Алексей Евгеньевич***

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, [alexey@mikhaylenko.net](mailto:alexey@mikhaylenko.net)

## NUMERICAL SIMULATION OF HEAT TRANSFER PROCESSES IN SOLID STATE THERMOSTATIC SYSTEM

**O. Belova (ovbelova@yandex.ru), M. Korneeva (makorneeva@mail.ru),  
D. Mustafina (DMustafina@yandex.ru)**

**Keywords:** polymerase chain reaction, heat conductivity, Peltier thermopiles, radiator, numerical simulation.

The results of numerical investigation of thermostatic system based on Peltier thermopiles are presented. In order to conduct research in dynamic regime of work the numerical method was developed. This method includes one-dimensional model for quick calculations and three-dimensional more precise model for investigation of temperature fields. This method can be used to solve different types of problems dedicated to upgrading of construction and designing of new devices.

## LOCAL NAVIGATION SYSTEM FOR SURFACE MOBILE ROBOTS

**V. Chernonozhkin (xou@list.ru), S. Polovko (eus@rtc.ru)**

**Keywords:** inertial navigation system, local navigation, autonomous mobile robots, information processing.

The local navigation system, a part of navigation complex for surface mobile robots, is presented in this article. Features of mobile robots' motion are taken into account in this system to solve navigation problems more effectively than when using navigation systems for other types of vehicles.

## SIMULATION MODELLING OF MULTINOMENCLATURE MACHINE-BUILDING PRODUCT LINES OPERATION

**A. Baranov (andrewbaranov@rambler.ru), A. Denisov (iptema@yandex.ru)**

**Keywords:** product lines scheduling, Petri nets, evolutionary algorithm.

The description of the unit "Grouping" is given in the article responsible for problem solving of designing products technological unification in CAD/CAPP/CAM/CAE system and generalized balancing product lines algorithm is resulted.

## USE OF THE WAVELET-ANALYSIS FOR PROFILOGRAM PROCESSING

**D. Ershov (Denis071200@mail.ru), D. Zapatrin (zapatrin@front.ru),  
N. Krilov (elmic@narod.ru), V. Lulin (283\_a@mail.ru)**

**Keywords:** device, process of friction, dynamic characteristics, profilogram.

The device for experimental research of friction processes is considered in the article. Dynamic characteristics of friction processes have been compared with microprofilograms, and correlation between them has been established.

## PHASE SHIFT METHOD FOR DURABILITY MEASUREMENTS OF FIBER OPTIC CABLES UNDER LONGITUDINAL TENSILE LOAD

**N. Ovseev (ovseev@yandex.ru), V. Musalimov (musvm@ya.ru)**

**Keywords:** cable, fiber, tension, proving, phase, durability.

The subject of interest in this article is reliability of fiber optic cables under longitudinal tensile load. The main reliability criterion of optical cable is life time. According to standards it is 25 years. Reasons of durability changing were explored in this work. It is shown that mechanical longitudinal load is practically the main factor that reduces life time of fiber optic cable. Ways of decreasing the level of errors and updating the traditional scheme of the phase shift method are given. The method and installation scheme were realized in a real test bed.

## AZIMUTHAL ALIGNMENT OF GYROINKLINOMETERS FOR ARBITRARY ORIENTATION WELLS WITH GPS-COMPASS

**V. Pervovskiy (pvladimir@ya.ru), J. Binder (mail@elmech.ru)**

**Keywords:** azimuthal alignment, gyroinklinometer, GPS-compass.

Various variants of an initial azimuthal alignment of gyroinklinometer from independent gyrocompassing up to external course pointing with application of satellite navigating devices are analyzed. Lacks of existing methods of an initial alignment and an opportunity of their elimination are shown at transition on external course pointing. The opportunity of realization of an initial alignment by means of GPS-compasses is considered and the constructive decision for device bases binding is offered.

## **PROGRAMMING MOVEMENT OF BIPEDAL WALKING MACHINE IN SAGITTAL PLANE**

**R. Alekseev (RostAlexeev@mail.ru), Yu. Kotel'nikov (kotel@mail.ifmo.ru)**

**Keywords:** bipedal walking machine, comfortable movement, program paths, inverse kinematics problem, zero point.

The article deals with the comfortable bipedal robot movement with given program paths for pelvis and feet in rectangular coordinates. Analytical calculation of master control for actuator with an application of inverse kinematics problems solving is given.

## **HIGH-VOLTAGE MODULATOR CONSTRUCTION WITH NANOSECOND FRONT FOR THE ELECTRO-OPTICAL SHUTTER MANAGEMENT AS A PART OF THE SOLID-STATE LASER**

**D. Ternovsky (dm-ternovsky@mail.ru), V. Togatov (dm-ternovsky@mail.ru)**

**Keywords:** the high-voltage techniques, the pulse techniques, techniques of a nanosecond range, electro optical shutter.

The mode of ultra fast switch-on of the MOSFET at which the device switching time does not exceed nanoseconds is considered. The explanation of the ultra fast switching mechanism is offered. Results of the experiments confirming the ultra fast switching mechanism are presented. Construction possibility of the high-voltage modulator with nanosecond front on the basis of ultra fast switching effect of the MOSFET is considered. Block diagram and characteristics of the developed high-voltage pulse modulator for electro optical shutters management with an impulse amplitude of 6 kV and front time up to 5ns are presented.

## **PARAMETER IDENTIFICATION OF HEAT FLOW IN 1D- CALORIMETER WITH HEAT-CONDUCTION COEFFICIENT SPECIFICATION**

**K. Kirillov (kirill.kirillov@gmail.com), N. Pilipenko (pilipenko@grv.ifmo.ru)**

**Keywords:** heat flow receivers, difference-differential models, inverse heat-conduction problems, parameter identification, Kalman filter.

The article investigates application possibilities of parameter identification method for simultaneous heat flow recovery and heat-conduction coefficient specification of calorimeter material. Mathematical models description and testing results of the designed program are given.

## **CODE GENERATION SYSTEM THORNADO AND ITS APPLICATION TO BUSINESS-SOFTWARE DEVELOPMENT**

**D. Deyev (deyeff@gmail.com), Y. Okulovsky (yuri.okulovsky@gmail.com)**

**Keywords:** software development, generative programming, code generation.

We consider the system of generative programming Thornado. It is based on markup languages, script languages and object-oriented programming. The application example of this system in business-software development is shown. Arrangement of software development with Thornado system is discussed.

## **MULTI AGENT IDENTIFICATION SYSTEMS WITH SEMANTIC DISTRIBUTION OF THE OBJECT OPERATING MODEL**

**A. Kal'yanova (Kalyanova-a@mail.ru), A. Dyomin (dav\_60@mail.ru)**

**Keywords:** multi agent systems, identification, operating models

Multi agent system architecture of engineering objects identification is considered in the article, and researches of multi agent approach application with semantic distribution of the object operating models are given.

## **APPLICATION OF MDD AND AOP PRINCIPLES TO GRAMMARWARE DEVELOPMENT**

**A. Breslav (abreslav@gmail.com), I. Popov (popov@mail.ifmo.ru)**

**Keywords:** grammar, grammarware, MDD, modularity, reuse.

Contemporary tools for grammars and grammarware developing do not satisfy the requirements of OO-software engineering: different system aspects are mixed up, modularity and separation of abstraction levels are violated. A concept of a tool aimed to solve these problems using the principles from Model-Driven Development and Aspect-Oriented Programming is presented in this paper.

## **FEATURES OF OPEN PRODUCTION DOCUMENTATION PROJECTS ON THE SINGLE BOARD COMPUTER EXAMPLE**

**P. Kosenkov (microtrigger@gmail.com), Y. Gatchin (gatchin@mail.ifmo.ru)**

**Keywords:** open documentation, project, development, approach.

The given work is devoted to the hardware creation model. The characteristic feature of the model is project documentation availability shown on the open single board computer example. Electronic devices design peculiarities with instant publication of intermediate variants in open type are shown. This feature allows increasing quality and decreasing founding and time of electronic device engineering. "Subversion" repository application is considered with integration of an automatic documentation generation program. Such approach gives the possibility to create documentation automatically and at the same time with the project design. This method allows not only to save resources but to promote the project using its openness as a competitive advantage.

## **POLARIZED CONTROL ELEMENTS ESTIMATION IN CONTINUUM MASS TRANSFER**

**Z. Simonenko (ZGSim@yandex.ru), O. Studenikin (oleg-studenikin@yandex.ru),  
O. Eliseev (ov1983@yandex.ru)**

**Keywords:** polarized elements, modeling, mass transfer process, ellipsometry.

The article is devoted to polarized elements estimation for polarized interferometer devices with zero-ellipsometry elements.

## **INFORMATION MAINTENANCE OF REQUEST PROCESSING IN TECHNICAL SUPPORT SERVICE**

**N. Gusarova (natfed@list.ru), R. Ivanov (roman@nevsky30.ru),  
A. Mikhaylenko (alexey@mikhaylenko.net)**

**Keywords:** technical support service, request processing, technical diagnostics, CMDB.

The problems of technical support service of computer devices in middle size companies (or separate departments of large companies) are considered. Developed data storage structure for information maintenance system of request registration and processing is described.

# СОДЕРЖАНИЕ

<b>1. МЕХАТРОНИКА</b> .....	<b>3</b>
<b>Белова О.В., Корнеева М.А., Мустафина Д.А.</b> Численное моделирование процессов теплообмена в твердотельном термостатирующем устройстве.....	3
<b>Черноножкин В.А., Половко С.А.</b> Система локальной навигации для наземных мобильных роботов .....	13
<b>Баранов А.А., Денисов А.Р., Левин М.Г.</b> Имитационное моделирование работы производственных линий многономенклатурного машиностроительного производства .....	22
<b>Ершов Д.В., Запатрин Д.В., Крылов Н.А., Люлин В.В., Мусалимов В.М.</b> Использование вейвлет-анализа для обработки профилограмм .....	29
<b>Овсеев Н.Ю., Мусалимов В.М.</b> Метод фазового сдвига для измерения стойкости волоконно-оптических кабелей к продольной растягивающей нагрузке .....	33
<b>Перовский В.С., Биндер Я.И.</b> Азимутальная выставка гироинклинометров для скважин произвольной ориентации с использованием GPS-компаса.....	41
<b>Алексеев Р.А., Котельников Ю.П.</b> Программное движение двуногого шагающего робота в сагиттальной плоскости.....	46
<b>2. ТЕХНОЛОГИИ</b> .....	<b>55</b>
<b>Терновский Д.С., Тогатов В.В.</b> Построение высоковольтного модулятора с наносекундным фронтом для управления электрооптическим затвором в составе твердотельного лазера.....	55
<b>Кириллов К.В., Пилипенко Н.В.</b> Параметрическая идентификация теплового потока, входящего в одномерный тепломер, с уточнением коэффициента теплопроводности.....	63
<b>Деев Д.В., Окуловский Ю.С., Попов В.Ю., Часовских В.П.</b> Система кодогенерации Thornado и ее использование для создания бизнес-приложений.....	70
<b>Кальянова А.И., Демин А.В.</b> Многоагентные системы идентификации с использованием семантического способа распределения модели функционирования объекта .....	80
<b>3. СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ</b> .....	<b>87</b>
<b>Бреслав А.А., Попов И.Ю.</b> Применение принципов MDD и аспектно-ориентированного программирования к разработке ПО, связанного с формальными грамматиками.....	87
<b>Косенков П.А., Гатчин Ю.А.</b> Особенности проектов с открытой производственной документацией на примере одноплатного компьютера.....	97
<b>Симоненко З.Г., Студеникин О.Л., Елисеев О.В.</b> Расчет поляризационных элементов управления в исследованиях массопереноса сплошных сред.....	103
<b>Гусарова Н.Ф., Иванов Р.В., Михайленко А.Ю.</b> Информационное сопровождение обработки заявок в службе технической поддержки.....	109
<b>SUMMARY</b> .....	<b>116</b>