

Министерство образования и науки Российской Федерации

Российская академия наук

**Рыбинский государственный авиационный технический университет
имени П. А. Соловьева**

Институт системного анализа РАН

**III Всероссийская научная
конференция молодых ученых
с международным участием**

**ТЕОРИЯ И ПРАКТИКА
СИСТЕМНОГО АНАЛИЗА**

**Труды III Всероссийской научной конференции
молодых ученых с международным участием**

ТОМ I

21 – 24 мая 2014 г.

Рыбинск

Теория и практика системного анализа: Труды III Всероссийской научной конференции молодых ученых с международным участием. – Т. I. – Рыбинск: РГАТУ имени П. А. Соловьева, 2014. – 200 с.

Представлены результаты научно-исследовательской работы молодых ученых ведущих научных организаций. Исследования, проводимые по широкому спектру актуальных проблем системного анализа, нацелены на развитие и улучшение современных методов управления, проектирования, анализа сложных систем.

В организации конференции принимали участие:

- 1) Учреждение Российской академии наук Институт системного анализа РАН;
- 2) Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Рыбинский государственный авиационный технический университет имени П. А. Соловьева»;
- 3) Общество с ограниченной ответственностью «Технологии системного анализа» (ООО «ТСА»).

Конференция поддержана: Российским фондом фундаментальных исследований, грант № 14-07-06804; Отделением нанотехнологий и информационных технологий РАН.

Программный комитет

Председатель

Попков Ю. С. – чл.-корр. РАН, директор ИСА РАН

Сопредседатель

Полетаев В. А. – д. т. н., ректор РГАТУ имени П. А. Соловьева

Заместитель председателя

Кожина Т. Д. – д. т. н., проректор по науке и инновациям РГАТУ имени П. А. Соловьева

Члены комитета

Милош Железны – PhD, Западночешский Университет, Чехия
Васил Сгурев – академик, Болгарская академия наук, Болгария
Арлазаров В.Л. – чл.-корр. РАН, ИСА РАН
Осипов Г.С. – д.ф.-м.н., ИСА РАН
Шаров В. Г. – к.ф.-м. н., РГАТУ имени П. А. Соловьева
Паламарь И.Н. – к.т.н., РГАТУ имени П. А. Соловьева
Афанасьев А.П. – д.ф.-м.н., ИППИ РАН
Лившиц В.Н. – д.э.н., ИСА РАН
Якимец В.Н. – д.с.н., ИППИ РАН
Бритков В.Б. – к.ф.-м.н., ИСА РАН

Организационный комитет

Председатель

Тихомиров И. А. – к. т. н., совет молодых ученых ИСА РАН

Заместитель председателя

Булычев А.В. – к.т.н., ИСА РАН

Члены комитета

Жарова Н.Ю.	–	ИСА РАН
Яковлев К. С.	–	к.ф.-м.н., ИСА РАН
Курочкин И.И.	–	к.т.н., ИППИ РАН
Большаков О.С.	–	РГАТУ
Петров А. В.	–	РГАТУ
Тарханов И.А.	–	к.т.н., ИСА РАН
Панов А.И.	–	ИСА РАН
Парадовская Л.И.	–	ООО «ТСА»

СЕКЦИЯ «АВТОМАТИЗАЦИЯ И МОДЕЛИРОВАНИЕ»

ОПТИМИЗАЦИЯ КОРРЕКЦИИ ОКОЛОКРУГОВОЙ ОРБИТЫ ИСЗ ПО ВЕРОЯТНОСТНОМУ КРИТЕРИЮ

В.М. Азанов

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Московский авиационный институт (национальный исследовательский университет)» (МАИ)

г. Москва, E-mail: azanov59@gmail.com

Ю.С. Кан

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Московский авиационный институт (национальный исследовательский университет)» (МАИ)

г. Истра, E-mail: yu_kan@mail.ru

Аннотация

Рассматривается задача коррекции траектории движения искусственного спутника Земли (ИСЗ), вращающегося по околокруговой орбите. Орбита периодически корректируется с помощью корректирующей двигательной установки (КДУ) большой тяги. Математическая модель процесса коррекции представляется в виде дискретной стохастической системы управления с вероятностным терминальным критерием качества. Предполагается, что ошибки отработки расчетной величины корректирующего импульса имеют равномерное распределение. Задача оптимального управления для случая двух шагов по времени аналитически решается с помощью метода динамического программирования.

1. Введение

Задачи оптимального управления по вероятностным критериям качества составляют предмет изучения специального раздела теории стохастического оптимального управления. К числу вероятностных критериев относятся функционал вероятности и функционал квантили. Функционал вероятности представляет собой вероятность непревышения некоторым точностным функционалом заданного допустимого уровня. Сам точностный функционал при этом характеризует точность системы управления, но зависит от траектории стохастической системы. Примером точностного функционала служит терминальный промах системы наведения. В постановке задачи оптимального управления с критерием качества в форме функционала вероятности обычно требуется этот функционал максимизировать.

Функционал квантили является, в некотором смысле, обратной характеристикой по отношению к функционалу вероятности. Физический смысл функционала квантили в том, что он, будучи верхней доверительной границей для точностного функционала, по сути характеризует гарантированную по вероятности точность системы управления. Задача оптимального управления с критерием в форме функционала квантили обычно трактуется как задача минимизации.

Примером указанных выше задач является задача коррекции траектории движения ИСЗ, вращающегося по околосферической орбите [1]. Цель коррекции заключается в превращении орбиты ИСЗ в круговую с заданным периодом, при этом предполагается использовать КДУ большой тяги, позволяющую с малой задержкой осуществлять корректирующие импульсы.

В данной статье рассмотрена задача оптимальной двухимпульсной коррекции движения ИСЗ с помощью двигателя большой тяги в дискретном времени. Критерием является вероятность попадания терминального состояния в интервал фиксированной длины. Случайные ошибки отработки КДУ большой тяги предполагаются равномерно распределенными. Соответствующая задача оптимального управления аналитически решается методом динамического программирования.

2. Постановка задачи

Геостационарные ИСЗ играют важнейшую роль в современных системах связи. Для земного наблюдателя они должны казаться неподвижными. Однако из-за ошибок различной природы (например, ошибок выведения) ИСЗ, который должен быть геостационарным, постепенно смещается, дрейфует по отношению к земному наблюдателю. Для устранения такого дрейфа время от времени орбита спутника подвергается коррекции, чтобы постепенно ликвидировать дрейф ИСЗ. Для коррекции орбиты предполагается использовать корректирующую двигательную установку (КДУ) большой тяги, позволяющую практически мгновенно осуществлять корректирующие импульсы.

Рассмотрим задачу управления с учетом случайных воздействий в дискретном времени $k = \overline{0, N+1}$. В соответствии с [2], введем обозначения: z_k^1 - текущее угловое расстояние между k -ым прохождением через апогей и требуемым положением на круговой орбите; z_k^2 - угловая скорость дрейфа в k -ый момент времени прохождения апогея, измеряемая угловым смещением ИСЗ за один оборот; t_k - количество оборотов по орбите; u_k - величина k -ого корректирующего импульса, пересчитанная в скорость дрейфа; ω_k - случайный коэффициент, $\omega_k \sim R[-\varepsilon, \varepsilon]$, характеризующий неточность отработки расчетной величины корректирующего импульса u_k . Будем рассматривать случай $\varepsilon < 1$. Математическая модель процесса управления ИСЗ имеет вид [1]:

$$\begin{cases} z_{k+1}^1 = z_k^1 + t_k z_k^2, \\ z_{k+1}^2 = z_k^2 + u_k(1 + \omega_k), \end{cases} \quad (1)$$

начальные условия: $z_0^1 = z^1$, $z_0^2 = 0$. Процесс корректирования считается законченным, если после проведения последней ($i = N$) коррекции выполняются следующие терминальные ограничения:

$$\begin{aligned} |z_{N+1}^1| &\leq \varphi, \\ |z_{N+1}^1 + tz_{N+1}^2| &\leq \varphi, \end{aligned}$$

где t - требуемое время пребывания ИСЗ, в окрестности требуемой долготы.

Введем в рассмотрение функционал вероятности:

$$P_\varphi(u(\cdot)) = P(\max\{|z_{N+1}^1|, |z_{N+1}^1 + tz_{N+1}^2|\} \leq \varphi), \quad (2)$$

Поставим задачу нахождения оптимального управления $u_k(\cdot) = (u_0, \dots, u_N)$, максимизирующего функционал (2):

$$P_\varphi(u(\cdot)) \rightarrow \max_{u(\cdot)}. \quad (3)$$

Задача (3) относится к классу задач оптимального управления с вероятностным терминальным критерием. К таким задачам применим метод динамического программирования [1].

3. Синтез оптимального управления

В соответствии с методом динамического программирования, введем функцию выигрыша

$$W_k^\varphi(z^1, z^2) = \sup_{u_k(\cdot), \dots, u_N(\cdot)} P(\max\{|z_{N+1}^1|, |z_{N+1}^1 + tz_{N+1}^2|\} \leq \varphi | z_k^1 = z^1, z_k^2 = z^2).$$

Рассмотрим случай $N=1$. Тогда для $k=1$ функция выигрыша имеет вид

$$W_1^\varphi(z_1^1, z_1^2) = \max_{u_1} M[W_2^\varphi(z_2^1, z_2^2) | z_1^1, z_1^2],$$

$M[\cdot]$ - математическое ожидание. Границные условия:

$$W_2^\varphi(z_2^1, z_2^2) = \begin{cases} 1, & \max\{|z_2^1|, |z_2^1 + tz_2^2|\} \leq \varphi, \\ 0, & \max\{|z_2^1|, |z_2^1 + tz_2^2|\} > \varphi. \end{cases}$$

Оптимальное управление определяется в результате решения конечномерной задачи

$$u_1^* = \arg \max_{u_1} M[W_2^\varphi(z_2^1, z_2^2) | z_1^1, z_1^2].$$

Преобразуем выражение для функции выигрыша на $k=1$ шаге

$$W_1^\varphi(z_1^1, z_1^2) = \max_{u_1} P(\max\{|z_1^1 + t_1 z_1^2|, |z_1^1 + (t+t_1) z_1^2 + tu_1(1+\omega_1)|\} \leq \varphi).$$

Следующая формула устанавливает аналитический вид оптимального управления на $k=1$ шаге.

$$u_1^* = \begin{cases} \frac{-\varphi - (z_1^1 + (t+t_1)z_1^2)}{t(1+\varepsilon)}, & z_1^1 + (t+t_1)z_1^2 \geq \frac{\varphi}{\varepsilon}, \\ \frac{\varphi - (z_1^1 + (t+t_1)z_1^2)}{t(1-\varepsilon)}, & \varphi \leq z_1^1 + (t+t_1)z_1^2 \leq \frac{\varphi}{\varepsilon}, \\ 0, & -\varphi \leq z_1^1 + (t+t_1)z_1^2 \leq \varphi, \\ \frac{-\varphi - (z_1^1 + (t+t_1)z_1^2)}{t(1-\varepsilon)}, & -\frac{\varphi}{\varepsilon} \leq z_1^1 + (t+t_1)z_1^2 \leq -\varphi, \\ \frac{\varphi - (z_1^1 + (t+t_1)z_1^2)}{t(1+\varepsilon)}, & z_1^1 + (t+t_1)z_1^2 \leq -\frac{\varphi}{\varepsilon}. \end{cases} \quad (4)$$

Функция выигрыша на данном шаге определяется выражением

$$W_1^\varphi(z_1^1, z_1^2) = \begin{cases} 1, & \varphi \geq |z_1^1 + (t+t_1)z_1^2| / \varepsilon, \\ \frac{\varphi}{|z_1^1 + (t+t_1)z_1^2|} \frac{\varepsilon+1}{\varepsilon}, & |z_1^1 + t_1 z_1^2| < \varphi < |z_1^1 + (t+t_1)z_1^2| / \varepsilon, \\ 0, & \varphi \leq |z_1^1 + t_1 z_1^2|. \end{cases} \quad (5)$$

Из второй ветви последнего выражения видно, что должно выполняться неравенство

$$\frac{|z_1^1 + t_1 z_1^2|}{|z_1^1 + (t+t_1)z_1^2|} < \varepsilon.$$

Данное неравенство устанавливает следующий факт: при малом ε время пребывания ИСЗ в окрестности требуемой долготы должно быть невероятно большим. Для того, чтобы охарактеризовать найденное оптимальное управление, рассмотрим 3 случая.

Случай 1, когда $|z_1^1 + (t+t_1)z_1^2| \leq \varphi$, не интересен, поскольку требуемая конечная точность выполнена и коррекции проводить вообще не требуется. Однако в данном случае имеет место некая зона нечувствительности $u_1 = 0$. В дальнейшем будет показано, что указанная зона нечувствительности имеет большое значение для нахождения оптимального управления на следующих шагах.

Случай 2, когда выполняется неравенство $\varphi \leq |z_1^1 + (t+t_1)z_1^2| \leq \frac{\varphi}{\varepsilon}$, является основным на $k=1$ шаге. Оптимальное управление в этом случае обеспечивает значение 1 критерию вероятности.

Случай 3, когда $|z_1^1 + (t+t_1)z_1^2| \geq \frac{\varphi}{\varepsilon}$, означает, что требуемая точность не может быть достигнута при проведении одной коррекции, следовательно необходимо проведение дополнительных коррекций.

Далее, действуя в соответствии со схемой динамического программирования, найдем функцию выигрыша на $k=0$ шаге:

$$W_0^\varphi(z_0^1, z_0^2) = \max_{u_0} M[W_1^\varphi(z_1^1, z_1^2) | z_0^1, z_0^2],$$

Введем обозначения

$$z_0 = z_0^1 + (t+t_1+t_0)z_0^2, \quad z_0^* = z_0^1 + (t_1+t_0)z_0^2.$$

Перепишем функцию выигрыша на $k=1$ шаге, преобразуя z_1^1, z_1^2 в соответствии с (1)

$$W_1^\varphi(z_0^1, z_0^2) = \begin{cases} 1, & \varphi \geq |z_0 + (t+t_1)u_0(1+\omega_0)| \varepsilon, \\ \frac{\varphi}{\varphi + |z_0 + (t+t_1)u_0(1+\omega_0)|} \frac{\varepsilon+1}{\varepsilon}, & |\zeta_0^* + t_1 u_0(1+\omega_0)| < \varphi \leq |z_0 + (t+t_1)u_0(1+\omega_0)| \varepsilon, \\ 0, & \varphi \leq |\zeta_0^* + t_1 u_0(1+\omega_0)|. \end{cases} \quad (6)$$

Из первой ветви (6) следует, что оптимальным на интервале $\varphi > |z_0| \varepsilon$ является управление $u_0^* = 0$. Разрешим неравенства в (6) относительно помехи ω_0 , предполагая, что $u_0 < 0$

$$W_1^\varphi(z_0^*, z_0) = \begin{cases} 0, & \omega_0 \geq \frac{-\varphi - z_0^*}{t_1 u_0} - 1, \\ \frac{\varphi}{\varphi + z_0 + (t+t_1)u_0(1+\omega_0)} \frac{\varepsilon+1}{\varepsilon}, & \frac{-\varphi / \varepsilon - z_0}{(t+t_1)u_0} - 1 \leq \omega_0 < \frac{-\varphi - z_0^*}{t_1 u_0} - 1, \\ 1, & \frac{\varphi / \varepsilon - z_0}{(t+t_1)u_0} - 1 < \omega_0 < \frac{-\varphi / \varepsilon - z_0}{(t+t_1)u_0} - 1, \\ \frac{\varphi}{\varphi - z_0 - (t+t_1)u_0(1+\omega_0)} \frac{\varepsilon+1}{\varepsilon}, & \frac{\varphi - z_0^*}{t_1 u_0} - 1 < \omega_0 \leq \frac{\varphi / \varepsilon - z_0}{(t+t_1)u_0} - 1, \\ 0, & \omega_0 \leq \frac{\varphi - z_0^*}{t_1 u_0} - 1. \end{cases} \quad (7)$$

В случае $u_0 > 0$, выражение (6) примет вид:

$$W_1^\varphi(z_0^*, z_0) = \begin{cases} 0, & \omega_0 \geq \frac{\varphi - z_0^*}{t_1 u_0} - 1, \\ \frac{\varphi}{\varphi + z_0 + (t + t_1)u_0(1 + \omega_0)} \frac{\varepsilon + 1}{\varepsilon}, & \frac{\varphi / \varepsilon - z_0}{(t + t_1)u_0} - 1 \leq \omega_0 < \frac{\varphi - z_0^*}{t_1 u_0} - 1, \\ 1, & \frac{-\varphi / \varepsilon - z_0}{(t + t_1)u_0} - 1 < \omega_0 < \frac{\varphi / \varepsilon - z_0}{(t + t_1)u_0} - 1, \\ \frac{\varphi}{\varphi - z_0 - (t + t_1)u_0(1 + \omega_0)} \frac{\varepsilon + 1}{\varepsilon}, & \frac{-\varphi - z_0^*}{t_1 u_0} - 1 < \omega_0 \leq \frac{-\varphi / \varepsilon - z_0}{(t + t_1)u_0} - 1, \\ 0, & \omega_0 \leq \frac{-\varphi - z_0^*}{t_1 u_0} - 1. \end{cases}$$

Введем обозначения

$$\begin{aligned} M^- &= \frac{-\varphi - z_0^*}{t_1 u_0} - 1, \quad M^+ = \frac{\varphi - z_0^*}{t_1 u_0} - 1, \quad N^- = \frac{-\varphi / \varepsilon - z_0}{(t + t_1)u_0} - 1, \quad N^+ = \frac{\varphi / \varepsilon - z_0}{(t + t_1)u_0} - 1, \\ f_{z_0}^\varphi(u_0) &= \frac{1}{(t + t_1)u_0} \frac{\varphi(\varepsilon + 1)}{2\varepsilon^2} \ln \left[\frac{\varphi + z_0 + (t + t_1)u_0(\varepsilon + 1)}{(\varphi + \varphi / \varepsilon)} \right], \\ g_{z_0}^\varphi(u_0) &= \frac{1}{(t + t_1)u_0} \frac{\varphi(\varepsilon + 1)}{2\varepsilon^2} \ln \left[\frac{\varphi - z_0 - (t + t_1)u_0(1 - \varepsilon)}{(\varphi + \varphi / \varepsilon)} \right], \\ h_{z_0}^\varphi(u_0) &= \frac{1}{(t + t_1)u_0} \frac{\varphi(\varepsilon + 1)}{2\varepsilon^2} \ln \left[\frac{\varphi + z_0 - (t + t_1)u_0(\varphi + z_0^*)/t_1}{(\varphi - \varphi / \varepsilon)} \right], \\ i_{z_0}^\varphi(u_0) &= \frac{1}{(t + t_1)u_0} \frac{\varphi(\varepsilon + 1)}{2\varepsilon^2} \ln \left[\frac{\varphi - z_0 - (t + t_1)u_0(\varphi - z_0^*)/t_1}{(\varphi - \varphi / \varepsilon)} \right], \end{aligned}$$

где $M^+ \leq N^+ \leq N^- \leq M^-$, если $u_0 < 0$, или $M^+ \geq N^+ \geq N^- \geq M^-$, если $u_0 > 0$. Математическое ожидание функции (6) определяется выражением

$$M[W_1^\varphi(z_0^*, z_0)] = \begin{cases} 1, & M^+ \leq -\varepsilon, N^+ \leq -\varepsilon, M^- \geq \varepsilon, N^+ \geq \varepsilon, \\ \frac{1}{(t+t_1)u_0} \frac{\varphi}{\varepsilon^2} + f_z^\varphi(u_0) + g_z^\varphi(u_0), & M^+ \leq -\varepsilon, N^+ \geq -\varepsilon, M^- \geq \varepsilon, N^- \leq \varepsilon, \\ \frac{1}{2\varepsilon} \left(\frac{\varphi/\varepsilon + z_0}{(t+t_1)u_0} - \frac{\varphi + z_0^*}{t_1 u_0} \right) + h_z^\varphi(u_0) + i_z^\varphi(u_0), & M^+ \geq -\varepsilon, M^- \leq \varepsilon, \\ h_z^\varphi(u_0) + \frac{1}{2\varepsilon} \left(\frac{-\varphi/\varepsilon - z_0}{(t+t_1)u_0} - 1 + \varepsilon \right), & N^- \geq -\varepsilon, M^- \leq \varepsilon, \\ \frac{\varphi(\varepsilon+1)}{2\varepsilon^2(t+t_1)u_0} \ln \left[\frac{\varphi + z_0 - (t+t_1)u_0(\varphi + z_0^*)/t_1}{\varphi + z_0 + (t+t_1)u_0(1-\varepsilon)} \right], & N^- \leq -\varepsilon, -\varepsilon \leq M^- \leq \varepsilon, \\ h_z^\varphi(u_0) + g_z^\varphi(u_0) + \frac{1}{2\varepsilon} \left(\frac{\varphi/\varepsilon + z_0}{(t+t_1)u_0} - \frac{\varphi + z_0^*}{t_1 u_0} \right), & N^+ \geq -\varepsilon, M^- \leq \varepsilon, M^+ \leq -\varepsilon, \\ f_z^\varphi(u_0) + i_z^\varphi(u_0) + \frac{1}{2\varepsilon} \left(\frac{\varphi/\varepsilon + z_0}{(t+t_1)u_0} - \frac{\varphi + z_0^*}{t_1 u_0} \right), & N^- \leq \varepsilon, M^- \geq \varepsilon, M^+ \geq -\varepsilon, \\ i_z^\varphi(u_0) + \frac{1}{2\varepsilon} \left(\varepsilon - \frac{\varphi/\varepsilon - z_0}{(t+t_1)u_0} + 1 \right), & M^+ \geq -\varepsilon, N^- \geq \varepsilon, N^+ \leq \varepsilon, \\ \frac{\varphi(\varepsilon+1)}{2\varepsilon^2(t+t_1)u_0} \ln \left[\frac{\varphi - z_0 - (t+t_1)u_0(\varphi - z_0^*)/t_1}{\varphi - z_0 - (t+t_1)u_0(1+\varepsilon)} \right], & -\varepsilon \leq M^+ \leq \varepsilon, N^+ \geq \varepsilon, \\ f_z^\varphi(u_0) + \frac{1}{2\varepsilon} \left(\frac{-\varphi/\varepsilon - z_0}{(t+t_1)u_0} - 1 + \varepsilon \right), & M^- \geq \varepsilon, -\varepsilon \leq N^- \leq \varepsilon, \\ g_z^\varphi(u_0) + \frac{1}{2\varepsilon} \left(\varepsilon - \frac{\varphi/\varepsilon - z_0}{(t+t_1)u_0} + 1 \right), & M^+ \leq -\varepsilon, -\varepsilon \leq N^+ \leq \varepsilon, \\ \frac{\varphi(\varepsilon+1)}{2\varepsilon^2(t+t_1)u_0} \ln \left[\frac{\varphi - z_0 - (t+t_1)u_0(1-\varepsilon)}{\varphi - z_0 - (t+t_1)u_0(1+\varepsilon)} \right], & M^+ \leq -\varepsilon, N^+ \geq \varepsilon, \\ \frac{\varphi(\varepsilon+1)}{2\varepsilon^2(t+t_1)u_0} \ln \left[\frac{\varphi + z_0 + (t+t_1)u_0(1+\varepsilon)}{\varphi + z_0 + (t+t_1)u_0(1-\varepsilon)} \right], & N^- \leq -\varepsilon, M^- \geq \varepsilon. \end{cases} \quad (8)$$

Оптимальное управление на $k=0$ шаге определяется из условия

$$u_0^* = \arg \max_{u_0} M[W_1^\varphi(z_0^*, z_0) | z_0^*, z_0].$$

Отметим, что несмотря на сложную структуру функции (8), определить оптимальное управление на некоторых отрезках нетрудно. Это связано со строгой монотонностью функций в соответствующих ветвях. Если $u_0 < 0$, функции в указанных случаях строго возрастают по u_0 и оптимальными являются управления, определяющие правые концы областей допустимых управлений. Области допустимых управлений, в свою очередь, формируются из ограничений в (8). Ниже приведено найденное оптимальное управление для ряда частных случаев:

$$u_0^* = \begin{cases} \frac{-\varphi / \varepsilon - z_0}{(t + t_1)(1 + \varepsilon)}, & z_0 \geq \frac{\varphi}{\varepsilon^2}, \\ \frac{\varphi / \varepsilon - z_0}{(t + t_1)(1 - \varepsilon)}, & \frac{\varphi}{\varepsilon} \leq z_0 \leq \frac{\varphi}{\varepsilon^2}, \\ 0, & -\frac{\varphi}{\varepsilon} \leq z_0 \leq \frac{\varphi}{\varepsilon}, \\ \frac{-\varphi / \varepsilon - z_0}{(t + t_1)(1 - \varepsilon)}, & -\frac{\varphi}{\varepsilon^2} \leq z_0 \leq -\frac{\varphi}{\varepsilon} \\ \frac{\varphi / \varepsilon - z_0}{(t + t_1)(1 + \varepsilon)}, & z_0 \leq -\frac{\varphi}{\varepsilon^2}. \end{cases} \quad (9)$$

Из последнего выражения видно, что зона нечувствительности уменьшилась в ε раз относительно последней коррекции. Вторая и четвертая ветви (9) являются основными расчетными случаями на $k=0$ шаге, на соответствующем отрезке указанное оптимальное управление обращает вероятностный критерий в 1. Первая и четвертая ветви (9) также могут обращать вероятностный критерий в 1, однако лишь в некоторых точках. Однако исключая такие точки, они не гарантируют достижение некоторой точности, что свидетельствует об необходимости проведения дополнительных корректирующих воздействий.

4. Заключение

Рассмотрена задача оптимальной двухимпульсной коррекции околосферической орбиты ИСЗ с помощью двигателя большой тяги по критерию вероятности в дискретном времени. С помощью метода динамического программирования найдено оптимальное управление, решающее указанную задачу.

Литература

- Лебедев А.А., Красильщиков М.Н., Малышев В.В. Оптимальное управление движением космических летательных аппаратов. М.: Машиностроение, 1974.
- Малышев В.В., Кибзун А.И. Анализ и синтез высокоточного управления летательными аппаратами. М.: Машиностроение, 1987.

СИСТЕМНЫЕ ИССЛЕДОВАНИЯ ПОЛИМЕРНЫХ ПОКРЫТИЙ МЕТАЛЛА С ЦЕЛЬЮ УПРАВЛЕНИЯ ИХ ПРОЧНОСТЬЮ

Баруздина О.С.

Череповецкий государственный университет

г. Череповец, olgabaruzdina91@mail.ru

Максимова О.Г.

Череповецкий государственный университет

г. Череповец, og62@mail.ru

Аннотация

В данной работе использована решеточная модель сложной системы «полимер + металл». Для исследования полимерных покрытий металла с целью управления их прочностью использованы методы имитационного моделирования. Рассчитано влияние металлической поверхности и качество ее обработки на величину параметра дальнего ориентационного порядка полимерной системы. Новизна работы заключается в том, что рассмотрено влияние двойного электрического слоя и приведены методы моделирования такой системы.

1. Введение

На современном этапе развития науки и техники наиболее часто и с высокой степенью эффективности в промышленности и в быту находят применение изделия с покрытиями, как класс слоистых композиционных материалов. Высокая эффективность применения покрытий достигается только при единственно – правильном выборе конструктивных параметров поверхностных слоев, отвечающих заданным условиям эксплуатации, а также разработке оптимальных (в плане достижения требуемого комплекса свойств) методов формирования таких слоев.

Первоначально лакокрасочный материал представляет собой систему мономеров, затем, по мере выпаривания растворителя, производится их ориентация, прилипание к поверхности металла и последующая полимеризация. При исследовании каждого из этих этапов существует различные модели, например, модели Рауза, Зимма или Каргина – Слонимского[6] для описания свойств полимерной системы, модель Желе – для изучения поверхности металла. Отсутствие комплексного подхода к процессам управления формированием покрытий позволяет решать только частные задачи, например, определение адгезионных характеристик металлов без применения в каком-либо технологическом процессе или создание систем автоматического контроля нанесения покрытия без учета его химического строения. Поэтому в проведении системных исследований полимерных покрытий стального листа играет важную роль создание их обобщенной модели, которая отображает реальную рассматриваемую систему и позволяет прогнозировать и исследовать различные характеристики (прочность, адгезию и др.).

В роли модели рассматривается отображение реальной системы, имеющей определенное соответствие, и позволяющее прогнозировать и исследовать характеристики (параметр порядка, коэффициент прилипания), определяющие взаимодействие системы с внешней средой [1]. Для исследования объемных свойств полимерных покрытий применена трехмерная решеточная модель Гейзенберга с ориентационными анизотропными взаимодействиями. Влияние металла рассматривалось в форме существования двойного электрического слоя, то есть в потенциальную энергию полимерной системы введено дополнительное слагаемое, зависящее от расстояния до межфазной границы.

Моделирование проводилось с помощью метода Монте-Карло с использованием программного комплекса MK-ORDER. Такие макроскопические характеристики полимерного материала, как параметр порядка, теплоемкость, восприимчивость, коэффициент прилипания, поверхностная энергия в зависимости от жесткости цепи, количества растворителя и температурного режима без учета влияния металла рассматривались ранее в работах [7-8].

2. Ориентационная модель полимерной системы

На рис.1.*a* приведено изображение контактного слоя между полимером и сталью, полученное с помощью сканирующего электронного микроскопа [13], из которого видно, что его структура состоит из ориентационно упорядоченных волокон, перпендикулярных поверхности стального листа. Поэтому для исследования таких полимерных покрытий металлического листа можно использовать решеточную модель (рис.1.*b*), в которой $N_2 N_3$ цепей, каждая из которых состоит из N_1 кинетических единиц (звеньев, ротаторов, субцепей), образуют трехмерную упорядоченную “квазирешетку” $(n_1, n_2, n_3) \equiv \vec{n}$.

Предполагается, что потенциал ориентационных взаимодействий $H_{\vec{n}, \vec{m}}$ между кинетическими единицами полимерных цепей, расположенными в узлах \vec{n} и \vec{m} решетки, зависит только от их взаимной ориентации, т.е. пространственного угла $\Phi_{\vec{n}, \vec{m}}$ между их осями. Вид этого потенциала зависит от химического строения полимерного вещества. Например, звенья макромолекул поливинилиденфторида (ПВДФ) имеют дипольный момент, направленный перпендикулярно оству цепи и связанный с электроотрицательным фтором и электроположительным водородом в составе боковых групп [3]. Поэтому их взаимодействия в подобных макромолекулярных системах можно описать с помощью потенциала дипольного типа [11]:

$$H_{\vec{n}, \vec{m}} = -K_i \cos \Phi_{\vec{n}, \vec{m}} = -K_i (u_{\vec{n}} u_{\vec{m}} + v_{\vec{n}} v_{\vec{m}} + w_{\vec{n}} w_{\vec{m}})/l^2, \quad (i = 1, 2, 3), \quad (2)$$

где $\Phi_{\vec{n}, \vec{m}}$ – пространственный угол между звеньями, расположенными в узлах $\vec{n} = (n_1, n_2, n_3)$ и $\vec{m} = (m_1, m_2, m_3)$, а $u_{\vec{n}}, v_{\vec{n}}, w_{\vec{n}}, u_{\vec{m}}, v_{\vec{m}}, w_{\vec{m}}$ – декартовы проекции на оси координат векторов, направленных вдоль соответствующих кинетических единиц цепей (рис.1.*b*), l - длина ротатора.

Значение индекса i в формуле (2) зависит от взаимного пространственного расположения звеньев цепей. Энергетическая константа K_1 вдоль продольного криволинейного направления n_1 решетки описывает внутрицепные ориентационные взаимодействия и связана с жесткостью цепи на изгиб [9], определяющей персистентную длину и средний косинус угла между соседними звеньями изолированной цепи. Константы K_2 и K_3 характеризуют ориентационные взаимодействия между соседними звеньями разных цепей в “поперечных” направлениях n_2 и n_3 “квазирешетки” [9, 10]. Значения этих констант также определяется конкретной химической структурой взаимодействующих полимерных цепей. Для систем, состоящих из полимерных цепей с полярными группами, константы K_2 и K_3 можно оценить из мультипольного разложения [4] энергии взаимодействия двух одинаковых диполей с моментом p , находящихся в звеньях соседних цепей:

$$K_2 \approx K_3 = \frac{1}{4\pi\epsilon_0} \frac{p^2}{r^3},$$

где r – среднее расстояние между соседними цепями, ϵ_0 – диэлектрическая постоянная. Оценка энергетических констант для полимерных систем приведена в работах [9, 10].

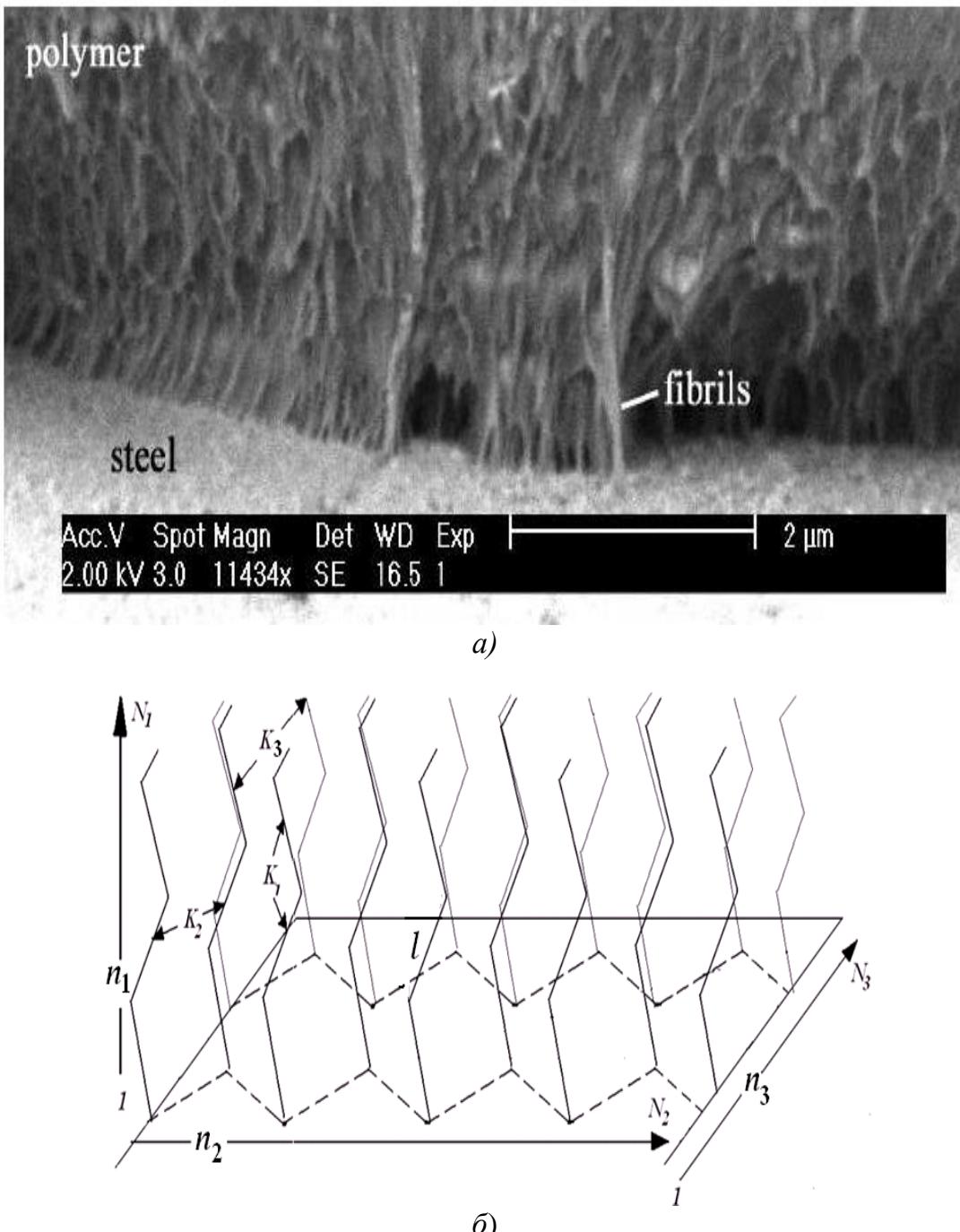


Рис.1 – Изображение контактного слоя между полимером и сталью, полученное с помощью сканирующего электронного микроскопа [13] (а). Пространственная многоцепочная модель жестких звеньев полимерных цепей с ориентационными взаимодействиями (б).

3. Учет взаимодействия металла с полимером

Для описания взаимодействия полимера с металлом используется модель двойного электрического слоя, то есть в потенциальную энергию полимерной системы введено дополнительное слагаемое, зависящее от расстояния до межфазной границы.

Как известно, возможны три механизма образования двойного электрического слоя (ДЭС) [12]:

- 1) За счет перехода ионов или электронов из одной фазы в другую (поверхностная ионизация),
- 2) Благодаря адсорбции соединений – примесей водных сред ,
- 3) За счет определенного ориентирования молекул сопряженных фаз в результате их взаимодействия.

Величина потенциала ДЭС (рис.2) в значительной мере связана со значением pH среды и химическими свойствами полимерного покрытия.

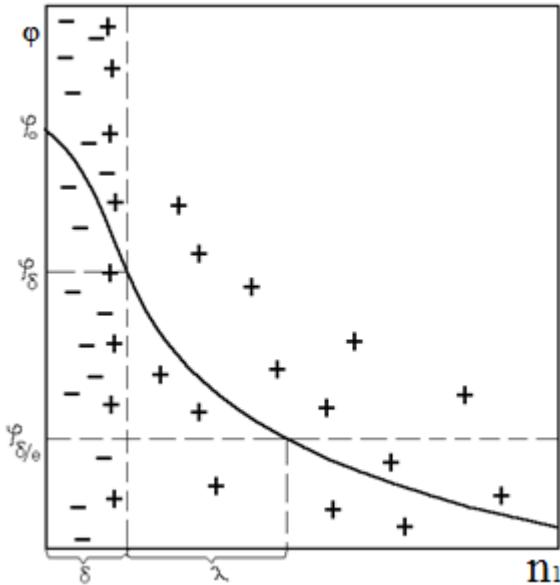


Рис.2 – Строение двойного электрического слоя и изменение в нем потенциала (ϕ).

Современная теория строения ДЭС [12], логически развивающая представления Гельмгольца, Штерна, Гуи и Чепмена, рассматривает слой в виде двух частей (рис. 2). Одна его часть находится непосредственно у межфазной поверхности и образует адсорбционный слой (слой Гельмгольца) толщиной δ , соответствующий диаметру гидратированных ионов, находящихся в нем. Остальная часть противоионов находится в диффузной части толщиной λ , зависящей от свойства и состава раствора и коллоидной системы. Значение потенциала в слое Гельмгольца при удалении от потенциалообразующих ионов снижается линейно от φ_0 до потенциала диффузного слоя φ_δ , а затем изменяется по экспоненте. Толщина диффузного слоя λ соответствует расстоянию, на котором потенциал диффузного слоя снижается в e раз.

Согласно уравнению Гуи – Чепмена, внутри диффузационного слоя потенциал можно рассчитать как:

$$\varphi = \varphi_\delta \cdot e^{-\lambda n_1} \quad (1),$$

где λ – толщина диффузационного слоя, т.е. количество слоев, при котором φ уменьшается в e раз.

4. Результаты моделирования.

При исследовании данной системы процесс ориентации полимерного вещества воспроизведен на компьютере с помощью его математической модели. Получить требуемые результаты другими способами крайне затруднительно, например, данная модель даже в отсутствии подложки в настоящем не имеет аналитического решения.

Поэтому для решения поставленных задач проведены численные эксперименты с разработанной моделью. Прежде всего, имитация представляет собой численный метод проведения на ЭВМ экспериментов с математическими моделями, описывающими поведение сложной системы «полимер+ металл» в течение заданного или формируемого периода времени. Имитируя различные реальные ситуации на модели, а именно исследование полимерных покрытий металла, прежде всего получаем возможность решения таких задач, как оценка прочности покрытий, сравнение вариантов структурных схем, определение степени влияния изменений параметров системы (параметр дальнего порядка, поверхностная энергия) и начальных условий на показатель эффективности системы [1].

Моделирование проводилось с помощью метода Монте-Карло с использованием программного комплекса MK-ORDER. Анализ сходимости метода Монте – Карло при проведении компьютерного эксперимента показал, что достаточно 10^4 шагов испытаний. Можно рассчитать многие макроскопические характеристики полимерного материала, в данной работе рассчитаны параметр порядка в зависимости от температуры и жесткости цепи (рис.3). На рис.3. приведена зависимость параметра порядка от величины приведенной температуры $k_B T / K_1$, при различных значениях потенциала ϕ , где T -абсолютная температура, K_1 - константа, описанная выше, κ_B - постоянная Больцмана ($1.38 \cdot 10^{-23} \frac{\text{Дж}}{K}$).

В полимерах с полярными молекулами (например, ПВДФ, поливинилхлорид (ПВХ) и др.) параметр дальнего ориентационного порядка определяется как средний косинус угла между ротаторами и направлением их преимущественного направления (директора):

$$\mu = \langle \cos \varphi_{i,j,k} \rangle \quad (2)$$

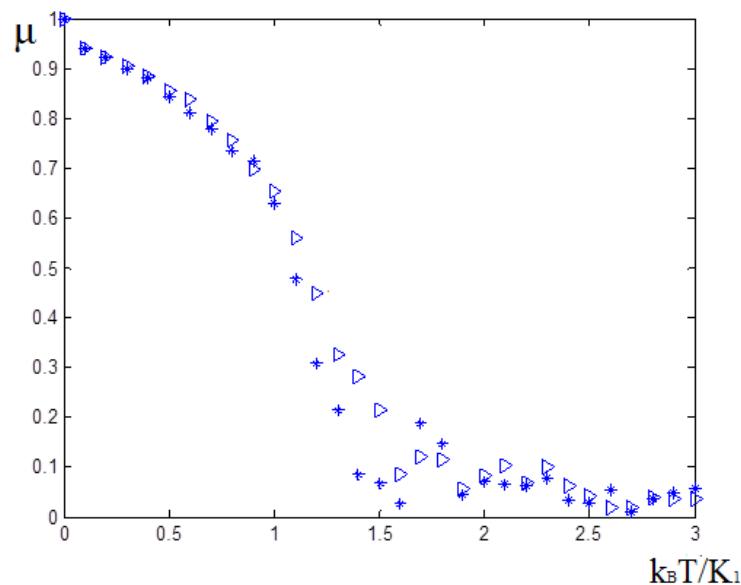


Рис.3 – Зависимость параметра дальнего порядка μ , рассчитанного по формуле (2) от величины $k_B T / K_1$, при $\phi_0 / K_1 = 0$ (*), $\phi_0 / K_1 = 1$ (>)

Как видно из рис.3, при определенной температуре в полимерной системе наблюдается фазовый переход типа «порядок-беспорядок». Нахождение полимера вблизи металлической поверхности приводит к сдвигу точки фазового перехода, кроме того, этот переход «размывается».

Рассмотрим теперь, как на данную зависимость влияет значение толщины диффузационного слоя:

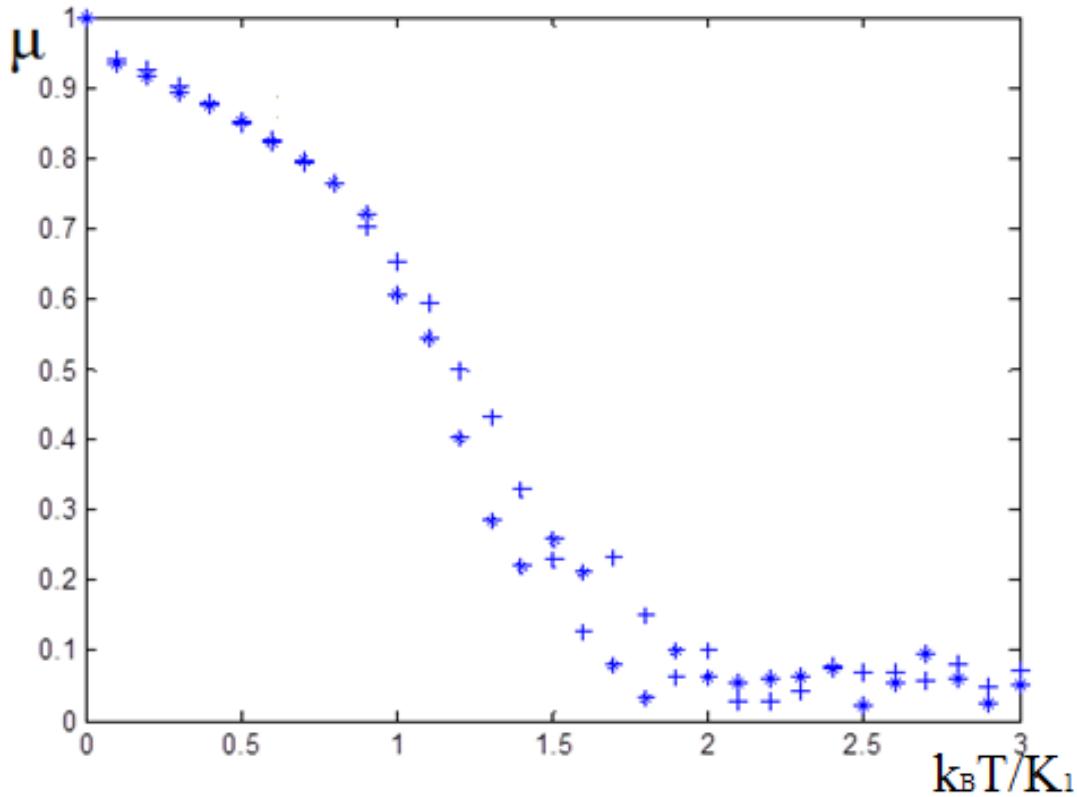


Рис.4 – Зависимость параметра дальнего порядка μ , рассчитанного по формуле (2) от величины $k_B T / K_1$, при $\lambda=0.01(*)$, $\lambda=0.001(+)$

Таким образом, при большом значении λ на фазовый переход влияет только нижний слой. Из этого следует что, переход наступает при более высоких температурах, что также подтверждается теорией Ландау, включение внешнего электрического поля приводит к увеличению температуры фазового перехода и к его размытию.

Толщина диффузационного слоя зависит от восприимчивости, значения которой приведены в ряде работ [2].

5. Заключение

Под термином «имитационное моделирование» обычно подразумевают вычисление значений некоторых характеристик развивающегося во времени процесса путем воспроизведения течения этого процесса на компьютере.

Прочность полимерного покрытия определяется параметром порядка [5], так как ориентированный полимер в 10-15 раз прочнее неориентированного. Поэтому, зависимости, приведенные на рис. 3 и 4, показывают, что увеличение двойного электрического слоя металла приводит к увеличению прочности покрытия. Для увеличения потенциала двойного электрического слоя, который определяется значением pH – фактора поверхности металла, его необходимо обрабатывать специальным

шелочным раствором с наибольшим значением рН.

Литература

1. Антонов А.В. Системный анализ./А.В.Антонов// Учеб. для вузов.- М.: Высш. шк., 2004. – С.101-102,131-132.
2. Баруздина О.С. Современные теории структурных фазовых переходов в сегнетоэлектрических системах. Материалы IX научно - практической конференции «Новости передовой науки»/О.С. Баруздина. - Т.52. Физика. София. «Бял ГРАД-БГ», Болгария, 2013.
3. Блинов Л.М., Фридкин В.М., Палто С.П. и др. Двумерные сегнетоэлектрики./ Л.М, Блинов , В.М. Фридкин, С.П. Палто и др// Успехи физических наук. 2000 г., Т. 170, 13.
4. Бэкингем Э. Основы теории межмолекулярных сил. Применение к малым молекулам. [ред.] Б. Пюльман. Межмолекулярные взаимодействия: от двухатомных молекул до биополимеров/Э.Бэкингем/ М. : Мир, 1981, стр. 9–98.
5. Гуль, В. Е. Структура и механические свойства полимеров / В. Е. Гуль, В. Н. Кулезнев. – М.: Высшая школа, 1972.
6. Иванов В.А., Рабинович А.Л., Хохлов А.Р. Методы компьютерного моделирования для исследования полимеров и биополимеров/ В.А. Иванов, А.Л. Рабинович, А.Р. Хохлов. М.: Либроком, 2009.
7. Максимова О.Г., Максимов А.В., Ершов Е.В., Петрова Т. О., Варфоломеев И.А.Модели, методы и алгоритмы оптимального управления формированием поверхностных слоев и пленочных полимерных покрытий металлического листа/О.Г. Максимова, А.В.Максимов, Е.В.Ершов, Т.О.Петрова, И.А.Варфоломеев//Коллективная монография. – Череповец: ЧГУ,2013 г.
8. Максимова О.Г, Максимов А.В, Баруздина О.С, Диордийчук Д.В. Влияние температурного режима сушильных печей на качество полимерных покрытий/ О.Г.Максимова, А.В.Максимов, О.С.Баруздина, Д.В.Диордийчук// Вестник Череповецкого государственного университета.№4(53)Т.3. Череповец. 2013. - С.12-15.
9. Максимов А.В., Максимова О.Г. Описание ориентационного порядка в двумерных полимерных системах с помощью анизотропной модели Вакса – Ларкина/А.В. Максимов, О.Г. Максимова// Высокомолек. соед. А. 2003 г., Т. 45, № 9, стр. 1476-1468.
10. Максимов А.В. Теория упорядочения трехмерных полимерных систем с локальными ориентационно–деформационными взаимодействиями/А.В. Максимов//Высокомолек. соед. А. 2007 г., Т. 49, № 5, стр. 891–904.
11. Петрова Т.О., Максимова О. Г., Герасимов Р.А., Максимов А.В. Применение аналитических и компьютерных методов моделирования систем с ориентационными взаимодействиями /Т.О.Петрова, О.Г.Максимова, Р.А. Герасимов, А.В. Максимов//. ФТТ. 2012 г., Т. 54, № 5, стр. 883-884.
12. Савицкая Т.А. Коллоидная химия: строение двойного электрического слоя, получение и устойчивость дисперсных систем. Пособие для студентов химического факультета/Т.А. Савицкая , Д.А. Еотиков, Т.А. Шичкова – Минск:БГУ, 2011. –С. 5-7
13. Van den Bosch, M. J. An improved description of the exponential Xu and Needleman cohesive zone law for mixedmode decohesion / M. J. Van den Bosch, P. J. G. Schreurs and M. G. D. Geers // Eng. Frac. Mech. – 2007. – Vol. 73. – P. 1220–1234.

ОТБОР ЗНАЧИМЫХ ФАКТОРОВ ПРИ ПОСТРОЕНИИ НЕЙРО-НЕЧЕТКОЙ МОДЕЛИ УПРАВЛЕНИЯ ПРОЦЕССОМ УСКОРЕННОГО ОХЛАЖДЕНИЯ ЛИСТА

Д.В. Богачев

Череповецкий государственный университет,
г. Череповец, bogachev-d@yandex.ru

И.А. Варфоломеев

Череповецкий государственный университет,

г. Череповец, igor.varf@gmail.com

Е.В. Ершов

Череповецкий государственный университет,

г. Череповец, eve@chsu.ru

Аннотация

В работе проводится анализ проблемы выбора значимых факторов при формировании нейро-нечетких моделей сложных технологических процессов на примере системы управления установкой ускоренного охлаждения листа. Рассматривается один из подходов к решению задачи автоматического исследования данных с использованием технологии «box-counting». В докладе также приведены результаты практического применения разработанной модели на стане 5000 ЛПЦ-3 ОАО «Северсталь».

1. Введение

Повышение качественных показателей выпускаемой продукции и увеличение прибыли являются неотъемлемыми целями любого промышленного производства. Автоматизация технологических процессов позволяет достичь поставленных задач и становится важным этапом формирования производственного цикла. При этом необходимым является всестороннее изучение рассматриваемых объектов для создания эффективной системы управления. Большинство технологических процессов на предприятиях характеризуются нестационарностью, существенной нелинейностью, сложной взаимозависимостью параметров, наличием изменяющихся постоянных времени, ошибок измерения параметров. Поэтому не представляется возможным учесть все особенности рассматриваемых объектов. В таком случае традиционные методы построения моделей (использование конечно-разностных схем, экспериментально-статистические зависимости) не всегда приводят к желаемым результатам в силу принимаемых допущений.

В подобных ситуациях возможным является использование интеллектуальных методов, наиболее популярными из которых являются нейронные сети и нечеткая логика. Оба подхода являются мощными средствами проектирования, имеющими свои сильные и слабые стороны. Сочетание этих двух алгоритмов в единую систему является перспективным путем развития интеллектуальных методов. Гибридные технологии нейро-нечеткого моделирования обладают преимуществами обоих алгоритмов, а также лишены их недостатков. Они позволяют автоматизировать процесс формирования структуры нечеткой системы, представленной в параметрической форме, за счет применения алгоритмов обучения искусственных нейронных сетей. Однако использование нейро-нечетких методов сталкивается с необходимостью выбора значимых входных переменных, так как каждый новый фактор значительно усложняет структуру готовой модели. Решение данной проблемы на примере конкретного технологического процесса составляет цель настоящей работы.

2. Описание технологического процесса

Для повышения конкурентоспособности на мировом и внутреннем рынках металлургические предприятия активно внедряют в процесс производства сортового проката новейшие технологии и агрегаты, обеспечивающие высокое качество готовой

продукции и снижающие экономические затраты. Одним из способов совершенствования контролируемой прокатки является использование процесса ускоренного охлаждения металла. Применение данной процедуры позволяет получить на выходе из стана прокат с требуемой температурой. При этом ускоренное охлаждение является перспективной технологией термомеханической обработки низколегированных сталей, которая дает возможность достичь мелкозернистой структуры металла, обеспечивающей особые прочностные свойства, и увеличить производительность стана [7].

Основными задачами при охлаждении являются исключение коробления полосы из-за неравномерного ее охлаждения и возможность регулирования скорости охлаждения металла для задания механических свойств готовой продукции.

В настоящее время существуют различные способы охлаждения проката: охлаждение в баках, струйное охлаждение, охлаждение воздуховоздушной смесью, камерное охлаждение [3, 8]. Наиболее известным и широко изученным является способ струйного охлаждения. На использующих его установках охладитель подается на верхнюю и нижнюю поверхность листа в виде плоских потоков. Данная технология обеспечивает равномерное охлаждение проката. При этом расходы охладителя, поступающего снизу и сверху, задаются независимо друг от друга. Как правило, установка имеет несколько секций охлаждения с раздельно регулируемым расходом. Дополнительно каждая секция может быть разбита на отдельные зоны для более тонкого управления.

Оператору установки ускоренного охлаждения или в систему АСУ ТП агрегата до начала обработки партии поступает информация о прокате, требуемой технологии обработки (режиме), конечной температуре охлаждения и т.д. Оператор или АСУ ТП производит выбор количества включаемых секций, расчет расходов охладителя по зонам секций, выполняет предварительный расчет скорости движения листов через установку.

Организация управления ускоренным охлаждением сталкивается с рядом трудностей, связанных с большой нестационарностью протекающих процессов теплообмена (скорости охлаждения могут достигать нескольких сотен градусов в секунду). Поэтому в данном случае оправданным является использование нейро-нечетких методов моделирования.

3. Нейро-нечеткий модуль управления

При определенных условиях нечеткая система может быть представлена в форме многослойной сети с прямым распространением сигнала. Рассмотрим структуру блока нечеткого вывода (рис. 1) [10].

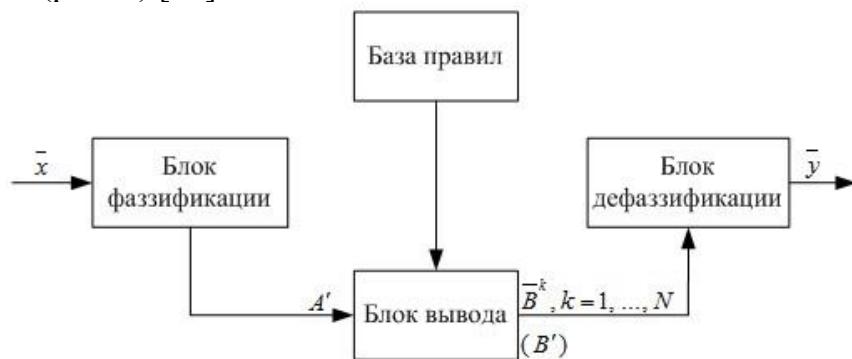


Рис. 1 – Структура модуля нечеткого управления

Конкретное значение $x = (x_1, x_2, \dots, x_n)^T \in X$ входного сигнала модуля нечеткого управления подлежит операции фазификации (ФАЗ), в результате которой ему будет сопоставлено нечеткое множество A' . Это нечеткое множество подается на вход блока выработки решения. На выходе блока нечеткого вывода (НВ) по обобщенному нечеткому правилу *modus ponens* получаем одно нечеткое множество B' , либо несколько нечетких множеств \bar{B}^k . Далее полученные нечеткие множества с помощью определенных методов (дефазификация по среднему центру, метод центра тяжести) отображаются в четкое значение \bar{y} в блоке дефазификации (ДЕФ).

Знания, составляющие основу корректного функционирования модуля нечеткого управления, записываются в виде нечетких правил в лингвистической форме [10]:

$$\begin{aligned} R^k : & IF(x_1 \text{ это } A_1^k \text{ AND } x_2 \text{ это } A_2^k \text{ AND } \dots \text{ AND } x_n \text{ это } A_n^k) \\ & THEN(y_1 \text{ это } B_1^k \text{ AND } y_2 \text{ это } B_2^k \text{ AND } \dots \text{ AND } y_m \text{ это } B_m^k), \end{aligned} \quad (1)$$

где $A_1^k, \dots, A_n^k, B_1^k, \dots, B_m^k$ – нечеткие множества; $k = 1, \dots, n$; x_1, x_2, \dots, x_n – входные переменные модели; y_1, y_2, \dots, y_m – выходные переменные модели.

Один из наиболее известных и часто применяемых способов реализации нечетких систем задается формулой [10]:

$$\bar{y} = \frac{\sum_{k=1}^N \bar{y}^k \sigma^k \left(\prod_{i=1}^n \exp \left[-\left(\frac{\bar{x}_i - \bar{x}_i^k}{\sigma_i^k} \right)^2 \right] \right)}{\sum_{k=1}^N \sigma^k \left(\prod_{i=1}^n \exp \left[-\left(\frac{\bar{x}_i - \bar{x}_i^k}{\sigma_i^k} \right)^2 \right] \right)}, \quad (2)$$

где N – количество нечетких правил, n – количество входных переменных, \bar{x}_i – конкретное значение входного сигнала, \bar{y} – конкретное значение выходного сигнала. Здесь в качестве функций принадлежности входных и выходных переменных используются непрерывно дифференцируемые функции Гаусса (удобны для вычисления производных в алгоритме обратного распространения ошибки), определяемые по формулам:

$$\begin{aligned} \mu_{A_i^k}(\bar{x}_i) &= \exp \left[-\left(\frac{\bar{x}_i - \bar{x}_i^k}{\sigma_i^k} \right)^2 \right], \\ \mu_{B^k}(\bar{y}) &= \exp \left[-\left(\frac{\bar{y} - \bar{y}^k}{\sigma^k} \right)^2 \right], \end{aligned}$$

где \bar{x}_i^k , \bar{y}^k – центры, а σ_i^k , σ^k – коэффициенты растяжения (сжатия) гауссовских кривых для входных и выходных переменных соответственно.

Теперь каждый элемент формулы (1) можно задать в форме функционального

блока, что после соответствующего объединения позволяет создать нейроподобную многослойную сеть. А поскольку алгоритм обратного распространения ошибки можно обобщить на любую сеть с прямым распространением сигнала, то данный модуль нечеткого управления можно обучать также как и обычную нейронную сеть. При этом параметры и весовые коэффициенты будут модифицироваться в процессе обучения, что позволит улучшать подбор нечетких множеств.

Основные трудности использования полученной структуры применительно к построению модели управления установкой ускоренного охлаждения возникают в ходе построения нечетких правил и задания начальных значений параметров функций принадлежности. Данная проблема решается с помощью методов, основанных на самоорганизации. Применительно к функциям принадлежности это означает такое размещение их центров, чтобы они охватывали только те области входных и выходных пространств, в которых находятся данные. При этом требуется избавить оператора, осуществляющего расчет параметров процесса охлаждения, от задания количества термов для всех входных и выходных переменных. Исходя из этого, актуальной представляется задача построения нечетких правил и формирования функций принадлежности на основе обучающих данных.

Для разбиения пространства каждой переменной в работе был использован метод субтрактивной кластеризации [2, 13]. Для синтеза базы правил применялся так называемый алгоритм на основе конкуренции (competitive learning algorithm) [2, 10, 14]. Использование указанных технологий позволило автоматизировать процесс построения нейро-нечеткой структуры.

Описанный метод построения нейро-нечеткой модели был протестирован на примере прогнозирования динамики индекса РТС. Алгоритм прогнозирования подразумевает тот факт, что каждое последующее значение рассчитывается на основе четырех предыдущих. При этом использовались исходные данные индекса РТС с 30.11.2012 г. до 29.04.2013 г. На рисунке 2 представлен график обучения нейро-нечеткой модели по итогам 500 эпох.

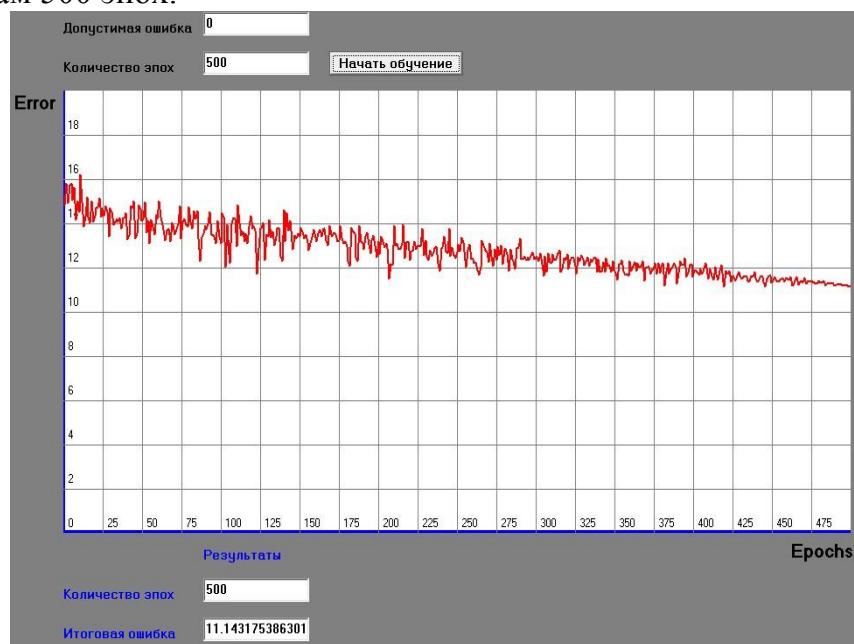


Рис. 2 – Результаты обучения модуля для расчета индекса РТС

Итоговая ошибка составляет 11.14. Для сравнения на рисунке 3 показан график обучения ANFIS-модели пакета MATLAB с использованием алгоритма субтрактивной кластеризации на том же наборе обучающих данных.

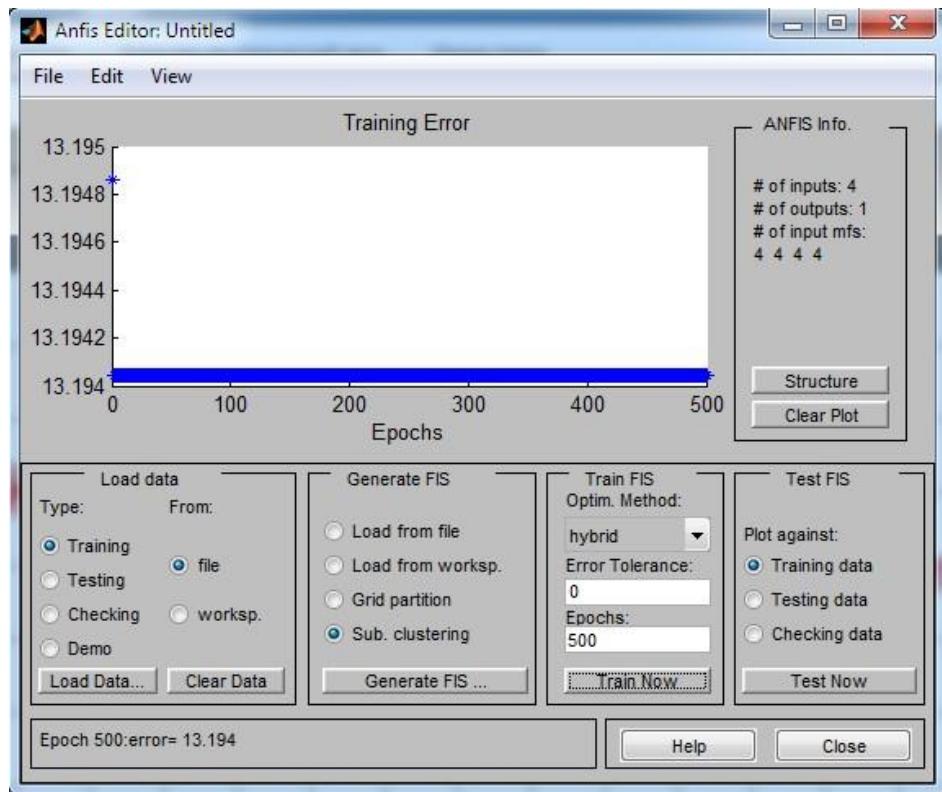


Рис. 3 – Результаты обучения ANFIS-модели с алгоритмом субтрактивной кластеризации для расчета индекса РТС

Как видим, итоговая ошибка составляет 13.19. При этом данное значение достигается за первые две эпохи и далее обучение не происходит.

4. Отбор значимых факторов модели

Максимальное число правил (1) модели экспоненциально зависит от числа входных переменных и числа термов для каждого входа [9]. Поэтому процесс построения базы правил стремительно усложняется с возрастанием количества входов и количества нечетких множеств для каждого из них. В литературе данное явление называется «проклятием размерности» [12]. Использование нейроподобной структуры модели упрощает синтез базы правил. Однако вместе с ростом числа правил увеличивается количество весовых коэффициентов модели. В этом случае резко ухудшаются обучающие способности нейро-нечетких сетей. При использовании субтрактивной кластеризации для разбиения пространства каждого входа отсутствует возможность повлиять на количество термов переменных. Поэтому единственным способом регулирования степени обучаемости модели является уменьшение числа входных переменных, т.е. определение значимых факторов.

Понижение размерности входного пространства – сложный и чрезвычайно важный элемент автоматического исследования данных. Исключение малозначащих факторов требуется не только при использовании нейро-нечеткой структуры, но и является

необходимым этапом предварительной обработки данных.

При моделировании процесса охлаждения в качестве входных параметров могут выступать следующие показатели: начальная температура проката; требуемая итоговая температура; габариты листа; температура охладителя; промежуток времени между окончанием прокатки и началом охлаждения; марка стали; параметры прокатки.

В качестве основного параметра управления (выходной переменной) обычно используется скорость движения проката или расход охладителя.

Наиболее остро проблема выбора входных переменных возникает в случае управления установкой за счет регулирования скорости рольгангов. В такой ситуации параметры расхода охладителя становятся ключевыми влияющими факторами. При этом их число может достигать нескольких десятков в зависимости от имеющегося числа секций охлаждения. Использование такого количества переменных в нейро-нечеткой модели не представляется возможным. В тоже время методы определения значимых факторов, выполняющие отбор нескольких переменных из общего числа, здесь не применимы, так как необходимым является сохранение информации обо всех охлаждающих зонах. Для решения данной проблемы использовался метод главных компонент, позволяющий уменьшить количество входных переменных с наименьшей потерей информации [1, 4]. При этом обработке данным алгоритмом подвергались только параметры, отвечающие за расход охладителя.

Часто в случае установления линейного характера зависимости можно оценить значимость входов построением линейных регрессионных моделей. Однако управляющие параметры установки ускоренного охлаждения проката (скорость рольгангов, расход охладителя) имеют сложные взаимосвязи с входным пространством. Применение же нелинейных регрессионных моделей затруднено необходимостью явного определения характера нелинейности еще до проведения анализа.

Одним из эффективных методов выбора значимых факторов, обладающим возможностью установления нелинейных зависимостей, является метод «box-counting» [5]. Данный алгоритм основывается на определении меры предсказуемости выходов, обеспечиваемой данным набором входных переменных. Согласно положениям теории информации, мерой предсказуемости случайной величины, является ее энтропия. Энтропия максимальна, когда все значения фактора равновероятны. В методике «box-counting» энтропия приближенно оценивается по набору чисел заполнения ячеек, на которые разбивается интервал ее возможных значений. Чем больше энтропия переменной, тем менее предсказуемо ее значение. Когда все значения примеров сосредоточены в одной ячейке – их энтропия равна нулю, т.к. положение данных определено. Равномерному заполнению ячеек соответствует максимальная энтропия – наибольший разброс возможных значений переменной. Предсказуемость случайного вектора Y , обеспечиваемая знанием другой случайной величины X , определяется кросс-энтропией, которая рассчитывается по формуле:

$$I(X, Y) = \log_2 \frac{N_x N_y}{N_{xy}}, \quad (3)$$

где N_x – число проекций ячеек на пространство X , N_y – характерный разброс данных по оси Y , N_{xy} – число клеток, в которых содержится точка с координатами (X_i, Y_i) ,

Чем больше кросс-энтропия, тем больше определенности вносит знание значения X в предсказание значения переменной Y . Для нахождения оптимальной комбинации входов воспользуемся наиболее простым методом, использующим значения кросс-

энтропии отдельно для каждого входа. Тогда алгоритм уменьшения размерности будет выглядеть следующим образом [11]:

- 1) Нормируем значения переменных.
- 2) Выбираем единицу дискретности. Она будет определять длины ячеек, на которые разбиваются интервалы значений переменных. Рассчитываем $N_{X_1}, N_{X_2}, \dots, N_Y$.
- 3) Вычисляем кросс-энтропию по формуле (3) для каждого фактора.
- 4) Упорядочиваем рассчитанные значения кросс-энтропии, переменные с наименьшими показателями отбрасываем.

5. Моделирование процесса управления установкой ускоренного охлаждения листа

Описанный метод построения интеллектуального модуля управления был реализован программно в среде Borland Delphi 7 и протестирован на примере установки контролируемого охлаждения (УКО) стана 5000 ЛПЦ-3 ОАО «Северсталь» [6]. Основным параметром управления являлась скорость движения проката. В качестве входных переменных использовались следующие данные: начальная температура проката; требуемая итоговая температура; длина, ширина, толщина листа; температура охладителя; промежуток времени между окончанием прокатки и началом охлаждения; время прокатки; время подстуживания листа на воздухе; выполнение кантовки при подстуживании. При этом модель строилась применительно к каждой марки стали. Структура полученного интеллектуального модуля представлена на рисунке 4. В зависимости от используемой выборки ретроспективных данных для конкретной марки стали методом «box-counting» формировался определенный набор значащих факторов. Метод главных компонент позволил сократить число параметров, отвечающих за расход охладителя, до 3 – 5 переменных. Для алгоритма «box-counting» принято решение отбирать 4 значимых фактора.

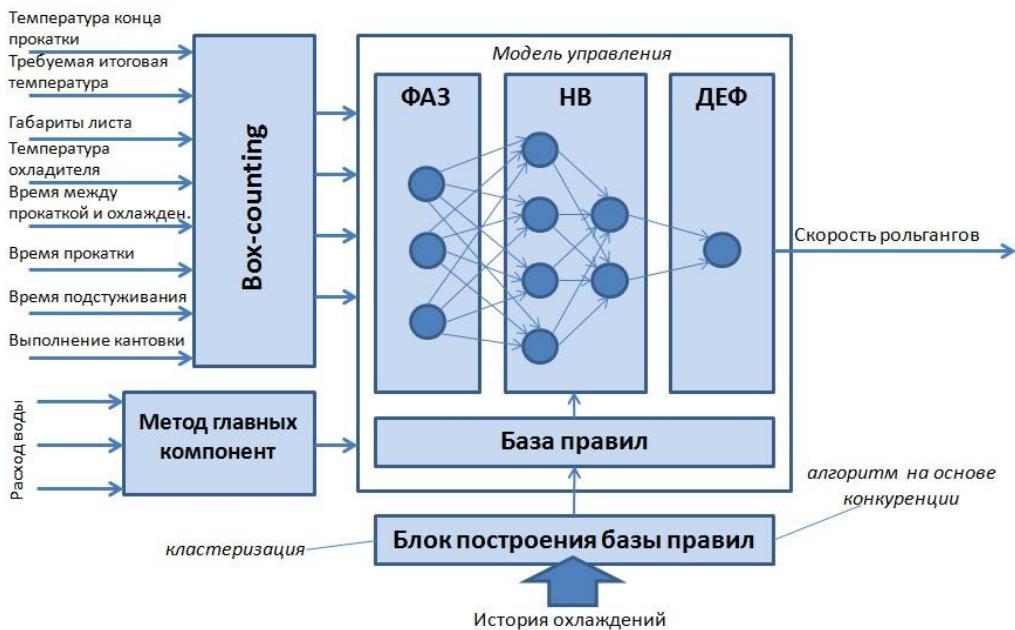


Рис. 4 – Структурная схема модели УКО

6. Заключение

Разработка описанных этапов позволила решить проблему выбора значимых факторов, используемых в нейро-нечеткой модели системы управления установкой ускоренного охлаждения. Программное обеспечение, реализующее расчетный модуль, может быть использовано оператором установки в режиме поддержки принятия решения при управлении процессом охлаждения. При этом применение метода «box-counting», инвариантного к виду зависимости параметров, позволило автоматизировать процесс выбора значимых факторов модели, тем самым избавив от необходимости аналитического исследования условий охлаждения проката.

Литература

1. Айвазян С.А., Бухштабер В.М., Енюков И.С., Мешалкин Л.Д. Прикладная статистика: Классификация и снижение размерности. – М.: Финансы и статистика, 1989. – 607 с.
2. Богачев Д.В., Ерилов Е.В., Варфоломеев И.А. Оптимизация нейро-нечеткой модели управления технологическими процессами в металлургии // Вестник Череповецкого государственного университета, 2013 №1 Т.2, С. 10–14.
3. Бровкин В.Л., Анурова Т.В., Радченко Ю.Н. и др. Анализ существующих технологий ускоренного охлаждения сортового проката и его влияние на структуру и механические свойства металла // Металлургическая теплотехника, 2010, вып. 2 (17). С. 14–22.
4. Дубров А.М., Мхитарян В.С., Трошин Л.И. Многомерные статистические методы и основы эконометрики – М., Московский государственный университет экономики, статистики и информатики. 2003. – 79 с.
5. Ежов А.А., Шумской С.А. Нейрокомпьютинг и его применения в экономике и бизнесе. – М.: МИФИ, 1998. – 224 с.
6. Липунов Ю.И., Траянов Г.Г., Эйсмондт К.Ю. Разработка систем регулируемого охлаждения и технологий термоупрочнения // Сталь. 2010. № 3. С. 96–99.
7. Морозов, Ю.Д, Эфрон Л.И. Стали для труб магистральных трубопроводов: состояние и тенденции развития // Металлург, 2006, № 5, С. 53–57.
8. Минаев А.А. Совмещенные металлургические процессы: Монография / Минаев А. А. – Донецк: Технопарк ДонГТУ УНИТЕХ, 2008. – 552 с.
9. Пегат А. Нечеткое моделирование и управление / пер. с англ. – 2-е изд. – М.: БИНОМ. Лаборатория знаний, 2013. – 798 с.
10. Рутковская Д., Пилинский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: пер. с польск. И.Д. Рудинского. – М.: Горячая линия – Телеком, 2006. – 452 с.
11. Снитюк В.Е. Прогнозирование. Модели, методы, алгоритмы: учебное пособие. – К.: «Маклаут», 2008. – 364 с.
12. Brown M., Bossley K. M., Mills D. J., Harris C. J. High dimensional neurofuzzy systems: overcoming the curse of dimensionality. Proceedings of the International Conference. FUZZ-IEEE/IFES'95. Yokohama, Japan, 1995, pp. 2139-2146.
13. Chiu. S., An Efficient Method for Extracting Fuzzy Classification Rules from High Dimensional Data // Advanced Computational Intelligence, vol. 1, no. 1, 1997.
14. Lin C.-T., Lee G. C. S., Neural-network-based fuzzy logic control and decision system // IEEE Transactions on Computers, December 1991, vol. 40, nr 12, 1320-1336.

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ УПРАВЛЕНИЯ ПРОЦЕССОМ СУШКИ ПОЛИМЕРНОГО ПОКРЫТИЯ НА ПОВЕРХНОСТИ МЕТАЛЛИЧЕСКОЙ ПОЛОСЫ

Варфоломеев Игорь Андреевич, к.т.н.

ФГБОУ ВПО Череповецкий государственный университет
г. Череповец, igor.varf@gmail.com

Ершов Евгений Валентинович, д.т.н., профессор

ФГБОУ ВПО Череповецкий государственный университет
г. Череповец, eve@chsu.ru

Виноградова Людмила Николаевна, к.т.н., доцент

ФГБОУ ВПО Череповецкий государственный университет
г. Череповец, Invinogradova@bk.ru

Трифанова Елена Владимировна, магистрант

ФГБОУ ВПО Череповецкий государственный университет
г. Череповец, helen.trif@gmail.com

Аннотация

В статье представлен способ повышения эффективности окраски оцинкованного металла на этапе производства как механизм увеличения коррозионной стойкости конечной продукции. На основе моделей решеточного газа и адсорбции Ленгмюра осуществлено моделирование процесса адгезии полимерного покрытия при покраске металлической полосы по технологии «Coil Coating». Дано описание структур интеллектуальных моделей оптимизации и управления процессом сушки полимерного покрытия. Представлены результаты применения полученных программных модулей.

1. Введение

Антикоррозийная защита может быть достигнута несколькими способами, но самый доступный и наиболее эффективный способ – это нанесение антикоррозийных красок, эмалей и грунтов. Нанесение полимерного покрытия на этапе производства металла увеличивает антикоррозийные характеристики продукции в связи с минимизацией времени взаимодействия металла с внешними агрессивными компонентами окружающей среды. Лакокрасочный материал (ЛКМ) представляет собой барьерное покрытие, которое при правильном использовании, обеспечивает достаточную защиту стали от коррозии. Однако антикоррозионная краска не всегда выполняет свою защитную функцию в полной мере. Правильное нанесение покрытия на поверхность металла, обеспечивающее наилучшее прилипание является одним из механизмов повышения степени коррозионной стойкости.

Метод «Coil Coating» является современным методом проведения окрасочных работ, когда нанесение полимерных покрытий на полосу оцинкованного металла происходит в поточных автоматизированных линиях. В этом процессе металлические полосы рулонного проката подвергаются специальной обработке с последующим нанесением лакокрасочного материала с помощью валковых машин и сушкой покрытия в печных установках.

Для повышения конкурентоспособности предприятиям, выпускающим окрашенный

металл необходимо снижать издержки производства при одновременном улучшении качества продукции. Процесс сушки полимерного покрытия является энергозатратным, так как при этом используются печные установки. Кроме того, брак, полученный на данном этапе, является наиболее дорогостоящим ввиду того, что сушка ЛКМ является заключительной стадией производства предварительно окрашенного проката. Сокращения энергозатрат и повышения уровня адгезии, а, следовательно, и улучшения антисорбционных свойств металла с покрытием можно достичь за счёт оптимального подбора технико-технологических параметров процесса полимеризации. По этой причине повышение эффективности процесса сушки полимерного покрытия за счёт реализации оптимального управления является актуальной научно-технической задачей.

2. Моделирование процесса сушки полимерного покрытия

Для решения задачи оптимального управления на первом этапе строится математическая модель процесса, описывающая поведение объекта под влиянием управляющих воздействий. В связи с тем, что параметры процесса сушки распределены в пространстве и времени то для его описания необходимо использовать дифференциальные уравнения в частных производных [1]. На этапе моделирования необходимо уделить особое внимание адгезии ЛКМ, так как процесс сушки в значительной степени влияет на прилипание полимерного покрытия. Правильная сушка покрытия позволяет увеличить коррозионную стойкость окрашенного металла.

Распространенной теорией описания динамики полимерных молекул является модель Рауза, согласно которой полимерная структура представлена частицами, связанными в цепь [2]. Если частица в результате теплового движения сталкивается с поверхностью, она может либо зеркально отразиться от нее, либо оказаться крепко связанной с поверхностью, то есть стать адсорбированной. Возможно также обратное явление десорбции, когда адсорбированная частица, имеющая достаточную кинетическую энергию, покинет поверхность. В случае физической адсорбции звено полимерной цепи (адатом) связывается с поверхностью при помощи слабых связей сил Ван-дер-Ваальса, которые характеризуются энергией E_p . При более высоких температурах происходит обмен электронов между адсорбированной частицей и поверхностью, способствующий образованию сильной химической связи между ними, и о частице говорят, что она хемосорбирована [2]. Каждый адатом теперь оказывается в гораздо более глубокой потенциальной яме E_c . Чтобы попасть в эту яму, частица должна преодолеть энергетический барьер E_b . Величина $E_c + E_b$ определяет также работу выхода $A_{\text{вых}}$. На рис. 1 показан схематический график зависимости потенциальной энергии частицы E от расстояния до плоской поверхности r .

В качестве отправной точки использованы модели решеточного газа и адсорбции Ленгмюра, которая основывается на предположениях, что адсорбция ограничена толщиной однослойного покрытия, все адсорбционные узлы решетки эквивалентны и только одна частица может занять адсорбционный узел [2].

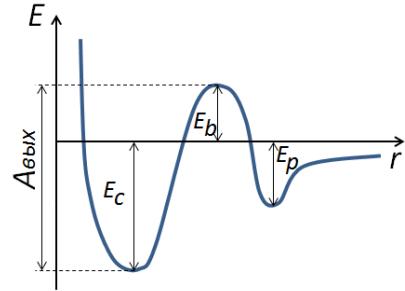


Рис. 1 – Зависимость потенциальной энергии частицы от расстояния до поверхности для случая хемосорбции

В этом случае адсорбционному слою соответствует двумерная модель решеточного газа, являющаяся одной из классических моделей статистической механики. В данной работе поверхность полосы представляется в виде двумерной прямоугольной решетки, которая содержит N узлов (адсорбционных мест) вдоль оси X и M узлов вдоль оси Y . Тогда положение узла характеризуется двумя числами i и j , а его занятость – числом $n_{i,j}$, которое равно нулю, если адсорбционное место свободно, и единице в противном случае. Энергия взаимодействия между узлами решетки (звеньями полимерных цепей) в модели определяется по формуле 1 [2,3].

$$E = \sum_{i,j=1}^{N,M} K n_{i,j} n_{i+1,j} + \sum_{i,j=1}^{N,M} K n_{i,j} n_{i,j+1}, \quad (1)$$

где K – константа взаимодействия. При расчетах по формуле 1 учитывается взаимодействие только между ближайшими узлами.

В связи с тем, что хемосорбция может происходить только при достаточно высокой температуре (частицы должны преодолеть энергетический барьер E_b), вероятность того, что звено полимерной цепи будет захвачено поверхностью металлического листа при его столкновении с поверхностью, можно аппроксимировать формулой 2 [2].

$$P = P_{\max} \left(1 - \frac{1}{1 + \left(\frac{T}{T_c} \right)^g} \right), \quad (2)$$

где T – температура поверхности, T_c – температура, при которой возникает хемосорбция, P_{\max} – максимальная вероятность захвата звена полимерной цепи при высоких значениях температуры, g – параметр, определяемый материалом полимера и типом грунтовки металла.

В данной работе в начальный момент времени ($t = 0$) принималось, что для всех узлов решетки адсорбции не было, т.е. $n_{i,j} = 0$. Расчеты производились с помощью компьютерного моделирования методом Монте-Карло на основе алгоритма Метрополиса, генерирующего случайный процесс, состоящий из набора конфигураций системы [3]. Количество шагов Монте-Карло было выбрано равным 20000.

Далее было произведено вычисление относительной степени заполнения поверхности металла звеньями полимерных цепей θ для различных температурных режимов (3).

$$\theta = \frac{\sum_{i,j=1}^{N,M} n_{i,j}}{NM} \quad (3)$$

На рис. 2б показаны результаты вычислений величины θ при различных температурных режимах $T(t)$, значения которых показаны на рис. 2а.

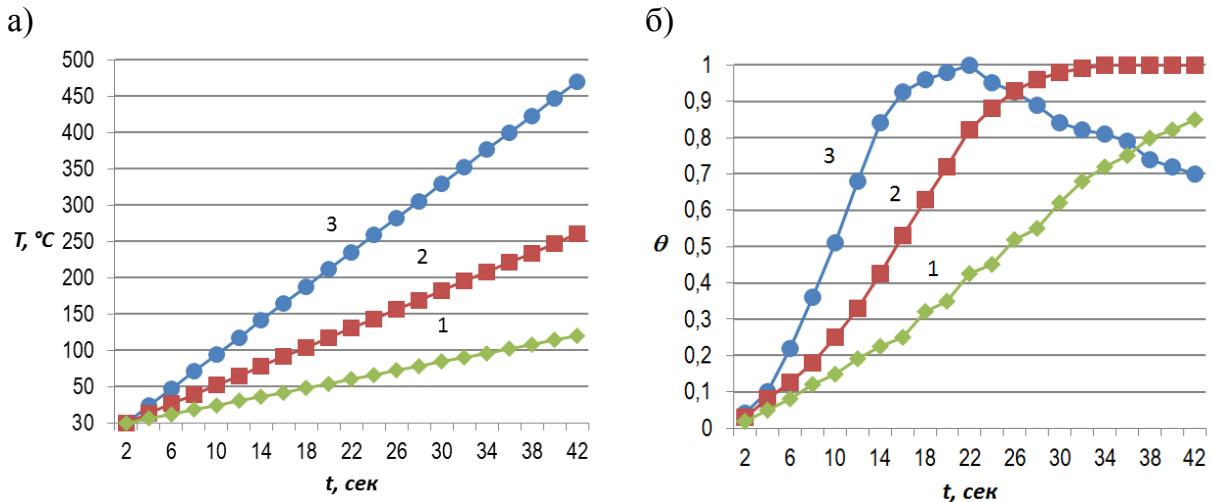


Рис. 2. а) Изменение температуры стального листа от времени б) Относительная степень заполнения поверхности металла звеньями полимерных цепей при соответствующем режиме изменения температуры

При низких значениях температуры (кривая 1 на рис. 2а и 2б) прилипание слабое. При очень высоких температурах (кривая 3) прилипание возникает, но впоследствии уменьшается. В этом случае после окончания сушки полимерного покрытия оно отстанет от поверхности, что является основной причиной появления дефектов и снижения антикоррозийных свойств продукции. Это связано с тем, что выполняется работа необходимая для разрушения адгезионной связи. Такую работу также называют адгезионной прочностью. Распределение температур на поверхности стального листа, определяемое кривой 2, является оптимальным для максимального прилипания [3].

Повышение температуры и продолжительности нагревания до определенного предела благоприятствует адгезии. Но в случае протекания деструктивных процессов в материале пленки адгезионная прочность снижается. Поэтому для каждого покрытия существует определенный оптимальный температурный режим его формирования, выбор которого является важной технологической задачей [4].

Процесс сушки полимерного покрытия нанесённого по технологии «Coil Coating» заключается в нагреве полосы до температуры полимеризации ЛКМ. Таким образом, необходимо построить модель рассчитывающую температуру поверхности металла в различных точках печи. Температура полосы зависит от температур 7 печных зон, скорости линии, толщины, материала и начальной температуры полосы.

При построении моделей процессов теплообмена между объектом и печной установкой принято считать, что передача тепла осуществляется за счёт радиационно-конвективного теплообмена. В рассматриваемом случае отсутствует необходимость учитывать передачу тепла по толщине полосы в связи с тем, что адгезия является поверхностным явлением.

Теплопередача внутри печи может осуществляться как за счет конвекции газа, так и за счет теплового излучения. Поэтому в данной работе использовалось дифференциальное уравнение 4 [3].

$$c\rho h \frac{\partial T}{\partial t} = k(T_{sr} - T) + \sigma\eta(T_{sr}^4 - T^4), \quad (4)$$

где T и T_{sr} – температура стального листа и температура печи соответственно, c – удельная теплоёмкость стали, ρ – плотность стали, h – толщина металла, σ –

постоянная Стефана–Больцмана, k – коэффициент теплоотдачи при конвекции, η – коэффициент рассеивания энергии при излучении. В правой части формулы 4 первая составляющая характеризует плотность потока тепла, поглощаемого сталью за счет излучения, вторая – конвекции.

В качестве примера на рис. 3 показано распределение температуры внутри печи T_{sr} как заданные дискретные величины в центре каждой из семи зон.

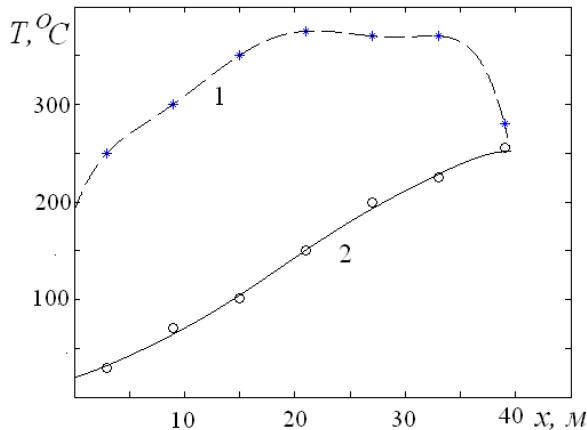


Рис. 3 – Функция зависимости температуры поверхности металлического листа от температур печных зон

Для более точного решения дифференциального уравнения численными методами нужно знать температуру как непрерывную функцию от координаты x печи. Для функции $T_{sr}(x)$ была произведена интерполяция с помощью интерполяционного полинома Лагранжа (кривая 1). Заданные значения температуры зон печи, на основании которых производилась интерполяция, обозначены символом «*». Температуры стального листа в различных точках печи, полученные в результате компьютерного моделирования, представлены на рисунке кривой 2. Экспериментальные значения температур поверхности листа показаны на рисунке символом «о».

Коэффициенты в дифференциальном уравнении 4 определялись из сравнения полученных численных расчетов с экспериментальными данными. При сопоставлении результатов моделирования с экспериментальными данными оказалось, что максимальное совпадение наблюдается в случае, когда 68 % потока тепловой энергии, поглощаемого стальным листом, приходится на излучение. На рис. 3 изображена зависимость температуры стального листа от координаты печи, вычисленная при решении указанного дифференциального уравнения (кривая 2) с подобранными коэффициентами.

При увеличении скорости движения полосы или изменении толщины листа нарушается тепловой режим, а, следовательно, и уменьшается прилипание. В этом случае необходимо изменять температуру внутри печи.

Построенная в рамках этого этапа модель позволяет подобрать температуры 7-ми печных зон ($T_{z1} \dots T_{z7}$) в зависимости от свойств металла (ρ, c), толщины полосы (h), скорости линии (V) и начальной температуры полосы ($T_{нач.}$), которые обеспечивают выходную поверхностную температуру металла ($T_{пов.вых.}$) попадающую в интервал температур полимеризации наносимой краски ($T_{пол1} \dots T_{пол2}$). Совокупность параметров,

перечисленных выше, в производстве принято называть рецептом. Однако набор подобранных таким образом рецептов нельзя считать оптимальным, т.к. в модели не учитываются критерии качества.

3. Оптимизация технологических параметров процесса сушки полимерного покрытия

На следующем этапе к математической модели добавляется формулировка цели управления, выраженная через критерии качества управления. Первым критерием является попадание выходной температуры поверхности металла в интервал температур полимеризации ЛКМ. Именно этот критерий обеспечивает максимальный уровень адгезии и наилучшую коррозионную стойкость металла (критерий качества). Вторым критерием является уменьшение расхода газа за счет снижения температур печных зон (критерий экономичности). Таким образом, видно, что оптимизация технологических параметров по перечисленным критериям осуществляется исключительно за счёт регулирования температур печных зон. Кроме того, стоит отметить, что задача усложняется многоокритериальностью оптимизации. Однако очевидным является то, что критерий качества более значимый, чем критерий экономичности.

Для получения оптимальных значений температур печных зон использовался генетический алгоритм (ГА), осуществляющий поисковую оптимизацию. Он реализует эвристический поиск, на основе случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, напоминающих биологическую эволюцию, для которой характерны такие методы как наследование, мутация, отбор и кроссинговер. Показателем оптимальности полученных значений является функция приспособленности, которая и реализует критерий оптимальности. По аналогии с естественным отбором – «Выживает самая приспособленная особь» [5]. Выбор критериальной функции зависит от задачи. Для нашего случая функция критерия представлена в выражении 5.

$$\text{Оценка} = \frac{k}{\sum_{i=1}^7 T_{3_i}}, \text{ где } k = \begin{cases} 1, & \text{если } T_{\text{пов. вых.}} \in [T_{\text{пол.1}}; T_{\text{пол.2}}] \\ 0, & \text{если } T_{\text{пов. вых.}} \notin [T_{\text{пол.1}}; T_{\text{пол.2}}] \end{cases} \quad (5)$$

Чем выше значение оценки полученной с помощью выражения 5, тем более оптimalен набор температур печных зон. Как видно из выражения 5, значение оценки будет минимальным в случае, когда выходная температура поверхности металла не попадает в интервал температур полимеризации ЛКМ (критерий качества). Модель оптимизации температур печных зон показана на рис. 4.

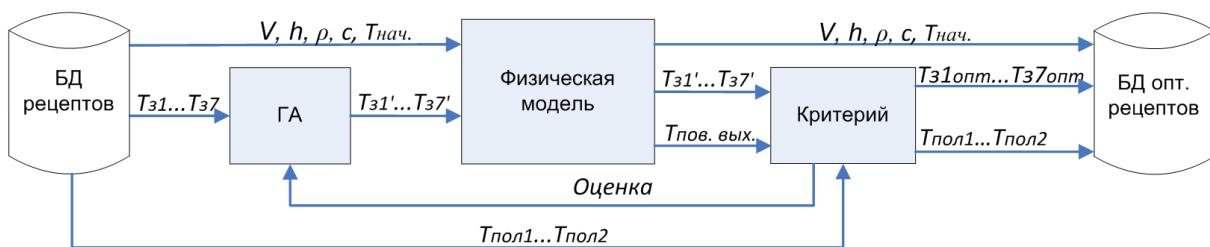


Рис. 4 – Модель многоокритериальной оптимизации температур

Из базы данных выбирается нужный рецепт, температуры печных зон которого передаются в блок генетического алгоритма. В этом блоке генерируется изменённый

набор температур печных зон ($T_{z1} \dots T_{z7}$), который передается в физическую модель. Интервал допустимых температур полимеризации ЛКМ ($T_{пол1} \dots T_{пол2}$) сразу передается в блок критерия, так как он необходим исключительно для оценки оптимальности. Оставшиеся параметры рецепта, передаются в физическую модель без изменения. В этом блоке вычисляется поверхностная температура полосы в момент выхода из печи ($T_{пов.вых.}$). В блоке критерия с помощью выражения 5 вычисляется оценка оптимальности. Генетический алгоритм реализует итеративное вычисление, поэтому оценка, полученная с помощью функции критерия, используется на следующем шаге для генерации нового более оптимального набора температур печных зон.

Выполнение данного этапа позволило осуществить подбор оптимальных технологических параметров по таким критериям как качество прилипания ЛКМ и расход газа. Однако модуль оптимизации рецептов не может использоваться в режиме реального времени в связи с тем, что использование генетического алгоритма требует значительных временных затрат. Кроме того, затруднено использование данного модуля для новых видов сортамента. В связи с этим решено использовать модель интеллектуального управления, обучение которой осуществляется на основе оптимизированных рецептов, полученных на этом этапе.

4. Интеллектуальное управление печью сушки полимерного покрытия

Задача управления заключается в подборе температур 7 печных зон, позволяющих нагреть поверхность металлической полосы с полимерным покрытием до температуры полимеризации при заданных значениях скорости линии, толщины, начальной температуры и материала полосы.

Печь сушки полимерного покрытия является многосвязным объектом. Это обусловлено тем, что на значение каждой из 7 температур печных зон оказывают влияние множество параметров, причем влияния на температуры различных зон отличаются. В данных условиях большое значение приобретает процесс определения структуры и идентификация модели для описания процесса со сложными и нестационарными параметрическими связями. В качестве механизма интеллектуального управления многосвязными тепловыми объектами в металлургии широко используются нейро-нечёткие модели, которые представляют собой отображение алгоритмов нечеткого логического вывода в форме многослойной сети с прямым распространением сигнала [1,5,6,7].

При использовании нейро-нечётких методов исходная модель декомпозируется на подмодели, количество которых равно количеству выходных параметров. В каждую подмодель на вход подается весь вектор входных параметров, однако коэффициенты влияния определённого входного параметра на выходной будут различны [5]. Структура модели управления процессом сушки ЛКМ представлена на рис. 5.

Нечёткость модели управления подразумевает разделение интервала возможных значений каждого входного и выходного параметра на лингвистические переменные. Лингвистической называется переменная, значениями которой являются слова или предложения естественного или искусственного языка. Например, начальная температура металлической полосы может меняться в диапазоне от 20 °C до 40 °C. Тогда данный интервал может быть разбит на следующие лингвистические переменные: «холодный» (20 °C – 30 °C), «тёплый» (27 °C – 36 °C), «горячий» (34 °C – 40 °C). Например, температура 29 °C принадлежит двум лингвистическим множествам, но значения совместимости с ними будут различны. Такое разбиение интервала

допустимых значений характерно для человеческого мышления ориентированного на категории. Количество лингвистических переменных (кластеров) зависит от задачи [5].

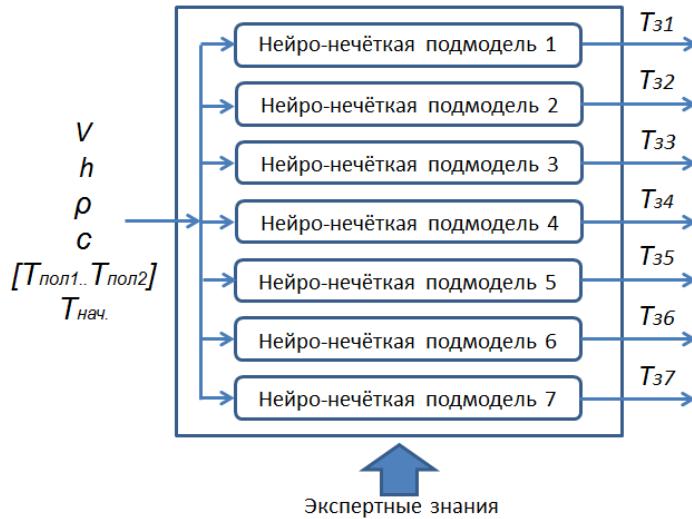


Рис. 5 – Модель интеллектуального управления процессом сушки ЛКМ

Нейро-нечёткая модель осуществляет вывод результатов на основе правил записанных с помощью лингвистических переменных, которые составляет эксперт, основываясь на своих знаниях о предметной области. Правило в лингвистической форме имеет вид, показанный в выражении 6.

$$\begin{aligned} \text{Правило: если } & (x_1 \text{ это } A_1 \text{ И } x_2 \text{ это } A_2 \text{ И } \dots \text{ И } x_n \text{ это } A_n) \\ & \text{то } (y_1 \text{ это } B_1 \text{ И } y_2 \text{ это } B_2 \text{ И } \dots \text{ И } y_m \text{ это } B_m), \end{aligned} \quad (6)$$

Правило состоит из двух частей – условия (*если*) и вывода правила (*то*). В условии x_i – входной параметр (например, начальная температура металлической полосы), A_i – лингвистическое значение входного параметра (например, «тёплый»), y_i – выходной параметр, B_i – лингвистическое значение выходной параметра, n, m – количество входных и выходных переменных соответственно.

В отсутствии эксперта или невозможности формализации правил для их построения можно использовать обучающие данные [8]. В данном случае обучающими данными являются оптимальные рецепты, полученные на предыдущем этапе. При обучении нейро-нечёткой модели полезным было бы избавление специалиста от задания кластеров для всех входных и выходных переменных. По этой причине для разбиения пространства каждой переменной использовался метод субтрактивной кластеризации, не требующий задания количества и границ кластеров [7].

В качестве операции нечёткой импликации для k -го правила используется операция умножения (правило Ларсена), показанную в формуле

$$\mu_{A^k \rightarrow B^k}(x, y) = \mu_{A^k}(x) \cdot \mu_{B^k}(y).$$

Декартово произведение нечётких множеств определяется по формуле

$$\mu_{A^k}(x) = \mu_{A_1^k \times \dots \times A_n^k}(x) = \mu_{A_1^k}(x) \cdot \dots \cdot \mu_{A_n^k}(x),$$

где n – количество термов. Т-норму представим произведением вида

$$\sup_{x \in X} \left\{ \mu_{A'}(x) * \mu_{A \rightarrow B}(x, y) \right\} = \sup_{x \in X} \left\{ \mu_{A'}(x) \cdot \mu_{A \rightarrow B}(x, y) \right\},$$

где A' – нечёткое множество, полученное в результате процедуры фузификации.

Для введения нечёткости используется операция типа синглетон.

$$\mu_{A'}(x) = \begin{cases} 1, & \text{если } x = \bar{x} \\ 0, & \text{если } x \neq \bar{x} \end{cases}.$$

Рассмотрим задачу отображения нечётких множеств \bar{B} в единственное значение $\bar{y} \in Y$. Такое отображение реализуется в отдельном блоке. Значение \bar{y} определяется с применением дефузификации по среднему центру:

$$\bar{y} = \frac{\sum_{k=1}^N \mu_{\bar{B}^k}(\bar{y}) \bar{y}^k}{\sum_{k=1}^N \mu_{\bar{B}^k}(\bar{y}^k)}$$

где \bar{y}^k – центр нечеткого множества B^k , т. е. точка, в которой $\mu_{B^k}(y)$ достигает максимального значения. Тогда нечёткий вывод будет определяться как

$$\bar{y} = \frac{\sum_{k=1}^N \bar{y}^k \left(\prod_{i=1}^n \mu_{A_i^k}(\bar{x}_i) \right)}{\sum_{k=1}^N \left(\prod_{i=1}^n \mu_{A_i^k}(\bar{x}_i) \right)},$$

где N – количество нечётких правил, n – количество входных переменных, \bar{x}_i – конкретное значение входного сигнала, \bar{y} – конкретное значение выходного сигнала. В качестве функций принадлежности используются функции Гаусса, определяемые по формуле

$$\mu_{A_i^k}(\bar{x}_i) = \exp \left[- \left(\frac{\bar{x}_i - \bar{x}_i^k}{\sigma_i^k} \right)^2 \right],$$

где \bar{x}_i^k – центр, а σ_i^k – коэффициент растяжения (сжатия) гауссовской кривой. Значения этих параметров корректируются в процессе обучения нейро-нечёткой системы. Тогда модуль нечёткого управления принимает окончательный вид, представленный в выражении

$$\bar{y} = \frac{\sum_{k=1}^N \bar{y}^k \left(\prod_{i=1}^n \exp \left[- \left(\frac{\bar{x}_i - \bar{x}_i^k}{\sigma_i^k} \right)^2 \right] \right)}{\sum_{k=1}^N \left(\prod_{i=1}^n \exp \left[- \left(\frac{\bar{x}_i - \bar{x}_i^k}{\sigma_i^k} \right)^2 \right] \right)}.$$

Данное выражение представляет собой один из наиболее известных и часто применяемых способов реализации нечётких систем [5]. Теперь каждый элемент этой формулы можно задать в форме функционального блока, что после соответствующего объединения позволяет создать нейроподобную многослойную сеть. Структура нейро-нечёткого модуля представлена на рисунке 6.

Элементы, обозначенные символом Π (мультипликаторы), перемножают все входные сигналы, элементы, обозначенные символом Σ (сумматоры) – суммируют их, а элемент $\frac{a}{b}$ делит один сигнал на другой. Чёрные поименованные точки, размещенные на связях, обозначают веса этих связей. Элементы Слоя 1, осуществляющего фуззификацию (ФУЗ), реализуют функцию Гаусса с параметрами \bar{x}_i^k и σ_i^k . Они будут

модифицироваться в процессе обучения, что позволит уточнить подбор нечётких множеств. В Слой 1 поступают входные сигналы \bar{x}_i , а на его выходе формируются значения функции принадлежности для этих сигналов, т.е. $\mu_{\bar{A}_i^k}(\bar{x}_i)$. На выходе Слоя 2 формируется результат нечёткого вывода (НВ) в виде значения функции принадлежности $\mu_{\bar{B}^k}(\bar{y}^k)$. Количество элементов этого слоя равно количеству правил N . Слой 4 представляет собой реализацию блока дефузификации (ДЕФУЗ). Веса связей интерпретируются как центры функций принадлежности нечётких множеств B^k . Эти веса также будут модифицироваться в процессе обучения. На выходе Слоя 4 формируется «чёткое» выходное значение модуля управления \bar{y} .

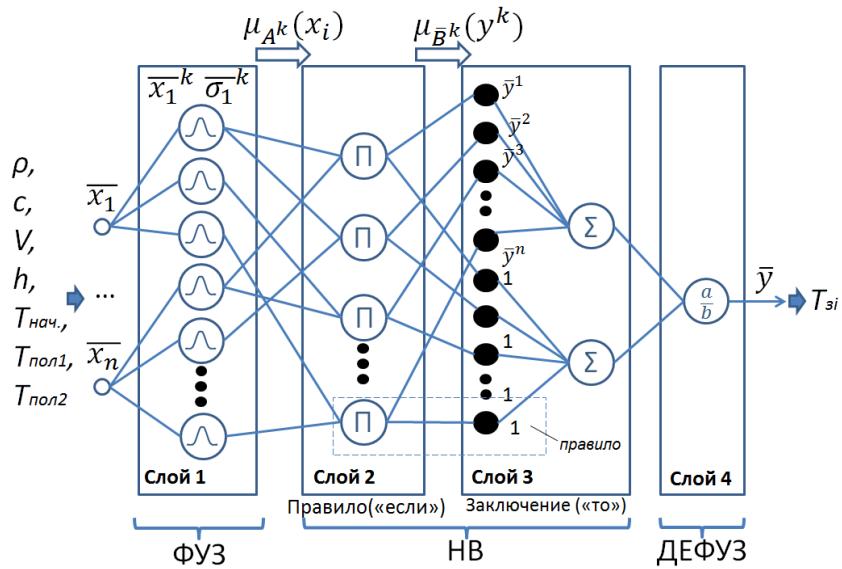


Рис. 6 – Структура нейро-нечёткого модуля

На рисунке 7 представлен график обучения подмодели для расчета температуры в первой зоне. В верхней части окна указывается количество эпох и допустимая ошибка, которая гарантирует необходимый нагрев полосы. Под эпохой понимается обход всего набора обучающих данных. Допустимая ошибка же равна среднему отклонению выходного значения модели от эталонного за одну эпоху.

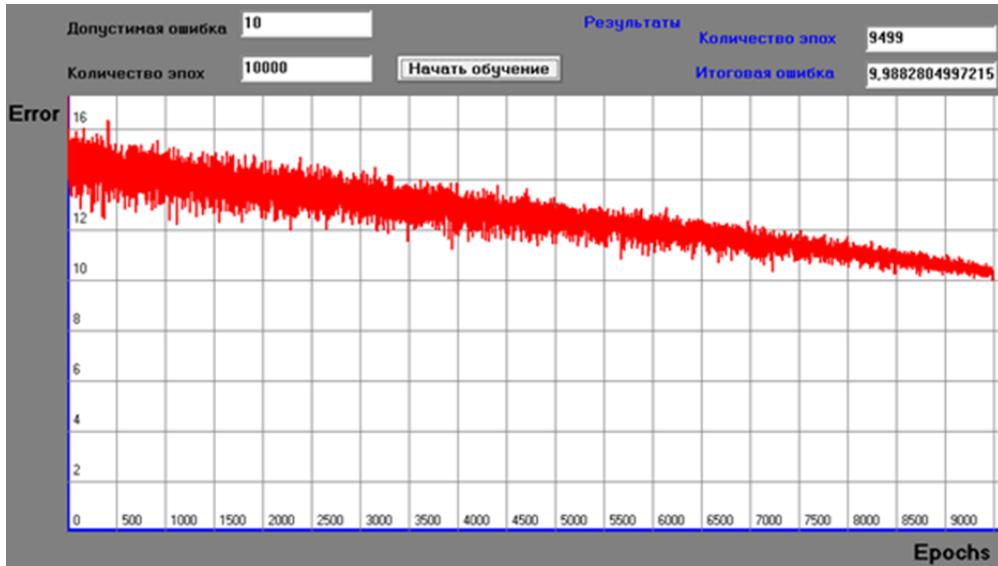


Рис. 7 – Результаты тестирования модели управления процессом сушки полимерного покрытия

Анализ результатов тестирования модели позволяет сделать вывод о её достаточной степени точности, т.к. значение ошибки выходного параметра меньше допустимого отклонения. Аналогичным способом была осуществлена настройка параметров для остальных 6 подмоделей, что сделало возможным использование модели в системе управления процессом сушки полимерного покрытия.

5. Результаты

Использование полученной модели сушки полимерного покрытия позволило увеличить точность прогнозирования поверхностной температуры металлической полосы на выходе из печи на 3-6%. Применение интеллектуальной системы управления печью сушки, построенной на основе оптимизированных рецептов, показало возможность снижения температур печных зон для существующих рецептов в среднем на 5-10 °С. Это эквивалентно снижению расхода газа на 1-2 %.

Определение качества адгезии проводилось с помощью испытания «Адгезия по методу РН», рекомендованного Ассоциацией переработчиков стали с покрытием. Для определения качества сцепления со стальной основой использовался адгезиметр. Более точное попадание поверхностной температуры полосы в диапазон рекомендуемых температур полимеризации позволило снизить количество бракованной продукции, связанной с отслоением покрытия и низкой коррозионной стойкостью, на 1,3 % согласно ГОСТ Р 52146. Это особенно актуально в условиях повышенной конкуренции на рынке продукции металлургического производства с высокой добавленной стоимостью.

Одним из брендов российской металлургии является «Стальной шёлк». Это оцинкованный металлопрокат с полимерным покрытием, специально разработанный ОАО «Северсталь» для изготовления кровельных материалов. Основным отличительным преимуществом данной продукции является наличие гарантии долговечности. В независимой лаборатории были проведены испытания, которые подтвердили срок службы «Стального шёлка» в течение 25 лет без появления признаков коррозии и сквозной коррозии [9]. Конкурентное преимущество данной продукции было достигнуто за счёт снижения издержек, а также совершенствования и повышения эффективности технологии нанесения и сушки покрытия на поверхности оцинкованного

металла.

Литература

1. Растрогин Л.А. Современные принципы управления сложными объектами. – М.: Сов. радио, 1980. – 232 с.
2. Жданов В.П. Элементарные физико-химические процессы на поверхности. Новосибирск: Наука, 1988. 320 с.
3. Варфоломеев И.А., Максимова О.Г., Ершов Е.В., Максимов А.В., Виноградова Л.Н. Моделирование процесса адгезии полимерного покрытия при покраске металлической полосы по технологии «Coil Coating» // Производство проката, 2013 №4, С. 26-30.
4. Ляхович А. М., Сюгаев А. В., Лялина Н. В., Решетников С. М. Защитные свойства полимерных пленок, полученных из бензола в низкотемпературной плазме на железе // Коррозия: материалы, защита, 2012 №12, С. 38-44.
5. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: пер. с польск. И.Д. Рудинского. – М.: Горячая линия – Телеком, 2006. – 452 с.
6. Богачев Д.В., Ершов Е.В., Варфоломеев И.А. Оптимизация нейро-нечёткой модели управления технологическими процессами в металлургии // Вестник Череповецкого государственного университета, 2013 №1, Т.2, С. 10-14.
7. Ершов Е.В., Бажинов А.Н. Прогноз потребления электроэнергии как средство повышения эффективности металлургического производства // Металлург – 2011, № 11 – С. 34-37.
8. Виноградова Л.Н., Ершов Е.В. Способ сжатия экспериментальных данных фрактальными распределениями для прогнозирования хода непрерывных металлургических процессов // Производство проката. – 2010, № 3. – С. 35 – 38.
9. <http://www.severstal.com/rus/media/news/document10402.phtml>

МЕТОД РЕШЕНИЯ ЗАДАЧИ КВАНТИЛЬНОЙ ОПТИМИЗАЦИИ С БИЛИНЕЙНОЙ ФУНКЦИЕЙ ПОТЕРЬ

Васильева С.Н.

Московский Авиационный Институт (Национальный исследовательский университет)
г. Москва, E-mail: taryaturunen@inbox.ru
Кан Ю.С.

Московский Авиационный Институт (Национальный исследовательский университет)
г. Истра, E-mail: yu_kan@mail.ru

Аннотация

В данном докладе представлен численный метод решения задач квантильной оптимизации с билинейной функцией потерь, основанный на аппроксимации ядра вероятностной меры в пространстве реализаций вектора случайных параметров выпуклым многогранником.

1. Введение

Задачи оптимизации функции квантили (задачи квантильной оптимизации) относятся к интенсивно развивающейся в настоящее время теории стохастического программирования с вероятностными критериями, современное состояние которой отражено в монографии [1]. Функция квантили определяется как квантиль заданного уровня для вероятностного распределения некоторой функции потерь, зависящей от вектора оптимизируемых переменных и вектора случайных параметров, малые значения

которой наиболее желательны для лица, принимающего решения. С точки зрения системного анализа и исследования операций функция квантили характеризует гарантированный по вероятности результат принимаемого решения. Поэтому оптимационные модели с квантильным критерием используются в приложениях для учета риска или требований надежности при наличии неопределенностей стохастической природы. Обычно функция квантили подлежит минимизации.

В теории стохастического программирования задачи с билинейными функциями потерь (линейными по случайным параметрам и по вектору оптимизируемых переменных) занимают особое место, поскольку порождены экономической проблематикой – задачами оптимизации портфельных инвестиций с учетом риска.

В докладе предлагается новый численный метод решения задач квантильной оптимизации с билинейными функциями потерь. Метод основан на результате из [1, 2], суть которого в том, что при выполнении некоторых требований регулярности к совместному распределению случайных параметров функция квантили равна максимуму функции потерь на ядре вероятностной меры. Ядро определяется как пересечение всех выпуклых и замкнутых доверительных множеств в пространстве реализаций вектора случайных параметров. Поэтому исходная задача квантильной оптимизации оказывается эквивалентной минимаксной задаче, в которой минимум берется по оптимизируемым переменным, а максимум – по реализациям случайных параметров на ядре. К сожалению, к настоящему времени методы построения ядра не развиты, что затрудняет использование этого результата для синтеза алгоритмов минимизации функции квантили с применением инструментария решения минимаксных задач. В докладе предлагается способ построения сколь угодно точных аппроксимаций ядра в регулярном случае с помощью выпуклых многогранников. Использование этих аппроксимаций вместо ядра в указанной минимаксной задаче приводит к минимаксной задаче, сводящейся к задаче линейного программирования большой размерности.

2. Постановка задачи

Пусть η - случайная величина с функцией распределения $F(x) = P(\eta \leq x)$. Квантиль распределения этой случайной величины для заданного уровня доверительной вероятности $\alpha \in (0,1)$ определяется выражением $[\eta]_\alpha = \min\{x : F(x) \geq \alpha\}$.

Рассматривается задача минимизации функции квантили с линейными ограничениями на u

$$\varphi_\alpha(u) = [u^T \xi]_\alpha \rightarrow \min_{u \in U} \quad (1)$$

где u - вектор оптимизируемых переменных, U – выпуклый многогранник, ξ - вектор случайных параметров с известным законом распределения.

3. Метод решения

Определение 1 [1]. Пусть ξ – n -мерный случайный вектор. Борелевское множество $S \in R^n$ называется α -доверительным, если $P(\xi \in S) \geq \alpha$.

Определение 2 [1]. Пусть ξ – n -мерный случайный вектор. Пересечение всех выпуклых и замкнутых α -доверительных множеств в R^n называется α -ядром вероятностной меры (распределения ξ).

Очевидно, что α -ядро является выпуклым компактным множеством. В [1] установлено, что α -ядро K_α может быть определено как пересечение всех замкнутых α -доверительных полупространств:

$$K_\alpha = \bigcap_{\|c\|=1} \left\{ x : c^T x \leq [c^T \xi]_\alpha \right\}. \quad (2)$$

Определение 3 [1]. α -ядро K_α называется регулярным, если всякое замкнутое полупространство, содержащее это ядро, автоматически является α -доверительным.

Достаточное условия регулярности α -ядра можно найти в [1]. В частности, известно, что K_α регулярно для невырожденного многомерного нормального распределения вектора ξ .

Теорема 1 [1]. Если случайный вектор ξ имеет регулярное α -ядро K_α , то

$$[u^T \xi]_\alpha = \max_{x \in K_\alpha} u^T x. \quad (3)$$

Предлагаемый метод основан на построении внешней аппроксимации $V_{\alpha n}$ α -ядра K_α , основанной на формуле (2), где $V_{\alpha n}$ является пересечением конечного множества α -доверительных полупространств, сгенерированных с помощью предлагаемого ниже алгоритма.

При аппроксимации ядра многогранником его грани задаются нормалами к ним, которые, в свою очередь, являются радиус-векторами точек, лежащих на единичной n -мерной сфере. При этом необходимо достаточно равномерно распределить точки по поверхности сферы. Для этого возьмем n -мерный куб, описанный вокруг сферы. Наложим на его поверхность равномерную сетку, плотность которой определяется числом разбиений каждой грани. Спроектируем узлы полученной сетки на сферу. Радиус-векторы к этим точкам задают направления нормалей к граням α -доверительных полупространств. Чем чаще сетка разбиения, тем ближе полученное решение к решению задачи (1). Приведем алгоритм построения множества таких точек.

Алгоритм:

1. Зададим начальные параметры:

n – размерность задачи;

N – число точек разбиения на ребре n -мерного куба;

$m = N^{n-1}$ – число узлов сетки на каждой грани n -мерного куба;

k – номер пары противоположных граней n -мерного куба ($k = \overline{1, n}$). На первой итерации $k = 1$.

Матрица $A \in R^{n \times m}$ служит для записи координат точек разбиения фиксированной грани по столбцам.

2. Зафиксируем грань куба, положив k -ю координату всех точек этой грани равной -1 .

$$A(k, j) = -1, \quad j = \overline{1, m}.$$

Осуществим обход всех точек выбранной грани. За начало обхода примем точку с координатами

$$A(i, 1) = -1, \quad i = \overline{1, n}, \quad i \neq k.$$

На первой итерации номер столбца матрицы A равен $s = 1$. Номер рабочей строки равен $f = n$, если $k \neq n$, и $f = n - 1$ иначе. Таким образом изменять координаты начинаем с конца. Перейдем к п. 3.

3. Если сумма элемента матрицы $A(f, s)$ и длины отрезка разбиения $\frac{2}{N-1}$ не превосходит единицу, то есть выполнено неравенство

$$A(f, s) + \frac{2}{N-1} \leq 1,$$

тогда координаты следующей точки получаются из координат предыдущей путем прибавления к f -ой координате длины отрезка разбиения.

$$\begin{aligned} A(i, s+1) &= A(i, s), \quad i = \overline{1, n}, \quad i \neq k, \\ A(f, s+1) &= A(f, s+1) + \frac{2}{N-1}. \end{aligned}$$

Перейдем к нахождению координат следующей точки $s = s+1$. Перейдем к п. 3. Если же исходное предположение не выполнено, тогда перейдем к п. 4.

4. Если еще не достигнут конец массива по строкам, то есть $f > 1$ при $k > 1$, или $f > 2$ при $k = 1$, тогда перейти к п. 5, иначе перейти к п. 7.

5. Если $f - 1 = k$, то $f = f - 1$. Если сумма элемента матрицы $A(f-1, s)$ и длины отрезка разбиения $\frac{2}{N-1}$ не превосходит единицу, то есть выполнено неравенство

$$A(f-1, s) + \frac{2}{N-1} \leq 1,$$

тогда координаты новой точки с первой по $f-2$ совпадают с координатами предыдущей. Координата $f-1$ точки s равна значению той же координаты предыдущей точки, увеличенной на длину отрезка разбиения $\frac{2}{N-1}$, остальные координаты положим равными -1 . Координата под номером k остается без изменений.

$$\begin{aligned} A(i, s+1) &= A(i, s), \quad i = \overline{1, f-2}, \quad i \neq k, \\ A(f-1, s+1) &= A(f-1, s) + \frac{2}{N-1}, \\ A(i, s+1) &= -1, \quad i = \overline{f, n-1}, \quad i \neq k. \end{aligned}$$

Перейдем к нахождению координат следующей точки $s = s+1$, номер рабочей строки равен $f = n$, если $k \neq n$, иначе $f = n-1$. Перейти к п. 3.

Если второе предположение не выполнено, перейти к п. 6.

6. Перейдем к предыдущей строке. Если $f-1 \neq k$, то $f = f-1$, иначе $f = f-2$. Перейти к п. 4.

7. Заметим, что матрица A для противоположной грани n -мерного куба отличается от полученной лишь k -ой строкой. Обозначим её A' . И определим её элементы следующим образом:

$$\begin{aligned} A &= A', \\ A'(k, j) &= 1, \quad j = \overline{1, n}, \quad j \neq k. \end{aligned}$$

Перейдем к п. 8.

8. Составим матрицу $B_k = [A : A'] \in R^{n \times 2m}$. Если $k < n$, то положим $k = k+1$ и перейдем к п. 2. Иначе перейдем к п. 9.

9. Составим массив K координат точек сетки на n -мерном кубе путем слияния матриц B_i , $i = \overline{1, n}$:

$$K = [B_1 : B_2 : \dots : B_n].$$

Перейдем к п. 10.

10. Для проекции точек с куба на сферу достаточно пронормировать координаты каждой точки в смысле евклидовой нормы. Для этого определим нормирующий множитель для каждой точки, как корень из суммы квадратов её координат:

$$T_j = \sqrt{\sum_{i=1}^n K_{ij}^2}, \quad j = \overline{1, 2nm}.$$

Каждую из координат j -го столбца разделим на соответствующий элемент T_j , тем самым получим координаты точек на единичной n -мерной сфере:

$$C(i, j) = K(i, j)/T_j.$$

В результате работы алгоритма строится выпуклый многогранник $V_{\alpha n}$ с вершинами v_j , $j = \overline{1, 2nm}$.

Пусть C_j – j -ый столбец матрицы C . Тогда

$$V_{\alpha n} = \bigcap_{j=1}^{2mn} \left\{ x : C_j^T x \leq [C_j^T \xi]_\alpha \right\}. \quad (4)$$

В формуле (4) квантиль $[C_j^T \xi]_\alpha$ в некоторых случаях, например для нормального распределения, может быть вычислена аналитически для любого вектора C_j . В тех же случаях, когда это не удается, для оценки этой величины можно использовать выборочную квантиль, генерируя искусственную или используя готовую выборку случайных значений ξ .

Поскольку $K_\alpha \subset V_{\alpha n}$, то в соответствии с формулой (3)

$$[u^T \xi]_\alpha \leq \max_{x \in V_{\alpha n}} u^T x. \quad (5)$$

При этом для регулярного α -ядра аппроксимация $V_{\alpha n}$ может быть сделана сколь угодно точной путем сгущения сети точек на единичном кубе.

В правой части (5) записана задача линейного программирования, поэтому

$$\max_{x \in V_{\alpha n}} u^T x = \max_j u^T v_j.$$

Поскольку аппроксимация ядра $V_{\alpha n}$ является более широким множеством, чем само ядро K_α , то с использованием аппроксимации строится верхняя оценка функции квантили:

$$\varphi_\alpha(u) \leq \max_{x \in V_{\alpha n}} u^T x.$$

В результате задача (1) сводится к минимаксной задаче

$$\max_j u^T v_j \rightarrow \min_{u \in U}. \quad (6)$$

Преобразуем задачу (6) в эквивалентную задачу линейного программирования:

$$y \rightarrow \min_{u \in U}, \\ u^T v_j \leq y, \quad j = \overline{1, m}.$$

Эффективность предложенного метода иллюстрируется на примере решения задач оптимизации инвестиционного портфеля с учетом риска.

Рассмотрим случай нормального распределения доходностей финансовых инструментов. Тогда известно [1], что оптимальный портфель по квантильному критерию, учитывающему риск, может быть найден в результате последовательного решения задач квадратичного программирования.

Подход, предложенный в настоящем докладе, сводит ту же самую задачу к задаче линейного программирования большой размерности (с большим числом ограничений).

Тестовые расчеты показали, что решения, полученные с использованием этих двух подходов, сближаются при сгущении сети точек на кубе.

4. Заключение

В данной работе был предложен метод решения задач квантильной оптимизации с билинейной функцией потерь при помощи аппроксимации ядра вероятностной меры многогранником, полученным при пересечении конечного числа доверительных полупространств. Это позволяет аппроксимировать исходную задачу, с помощью задачи линейного программирования. Решение тестовых примеров свидетельствует о работоспособности предложенного метода.

Литература

1. Кибзун А.И., Кан Ю.С. Задачи стохастического программирования с вероятностными критериями. – М.: ФИЗМАТЛИТ, 2009. – 372 с.
2. Kibzun A. I., Kan Yu. S. Stochastic Programming Problems with Probability and Quantile Functions. – Chichester: Wiley, 1996.

ИССЛЕДОВАНИЕ ПРИМЕНИМОСТИ НЕЙРОСЕТЕВОГО ОПТИМИЗАТОРА ПАРАМЕТРОВ ПИД-РЕГУЛЯТОРА ДЛЯ УПРАВЛЕНИЯ НАГРЕВАТЕЛЬНЫМ ОБЪЕКТОМ

Глушченко Антон Игоревич

Старооскольский технологический институт им А.А. Угарова (филиал) ФГАОУ ВПО
Национальный исследовательский технологический институт "МИСиС"

г. Старый Оскол, strondutt@mail.ru
Полещенко Дмитрий Александрович

Старооскольский технологический институт им А.А. Угарова (филиал) ФГАОУ ВПО
Национальный исследовательский технологический институт "МИСиС"
г. Старый Оскол, po-dima@yandex.ru
Еременко Юрий Иванович

Старооскольский технологический институт им А.А. Угарова (филиал) ФГАОУ ВПО
Национальный исследовательский технологический институт "МИСиС"
г. Старый Оскол, po-dima@yandex.ru

Аннотация

В статье рассмотрены вопросы реализации схемы управления с ПИД-регулятором,

коэффициенты которого в реальном времени подбираются с помощью нейросетевого оптимизатора. Для такого оптимизатора была выбрана трехслойная нейронная сеть прямого распространения, обучаемая оперативно с помощью алгоритма обратного распространения ошибки. Алгоритм был модифицирован рядом условий, позволяющих учесть особенности работы с нагревательными объектами управления. Работа системы была проверена на муфельной электронагревательной печи, работающей в различных режимах изменения задания и загрузки. Так как динамика объекта при нагреве и охлаждении различна, потребовалось разработать оптимизатор, содержащий две нейронные сети, отвечающих за подбор коэффициентов в соответствующих режимах.

1. Введение

На сегодняшний день достаточно активно ведутся исследовательские работы в области совершенствования систем управления технологическими процессами. Эта проблема весьма актуальна, так как большинство реальных объектов управления (ОУ) обладают нелинейными характеристиками, которые изменяются в процессе эксплуатации, в то время как управление ОУ ведется, в большинстве случаев, с помощью линейных ПИД-регуляторов. Коэффициенты таких регуляторов часто подобраны оптимально для конкретного состояния объекта, однако при смене его состояния (например, изменении задания по температуре, загрузке печи) данные значения коэффициентов уже не позволяют получать требуемого качества переходных процессов, что приводит к необоснованному увеличению энергозатрат на выполнение регламентов технологических процессов. Особенно данная проблема актуальна для металлургической отрасли ввиду наличия там большого числа энергоемких потребителей, которые можно объединить в класс тепловых объектов, например, таких как — печи нагрева литой заготовки, дуговые печи и пр. Для них принудительное охлаждение невозможно, ввиду чего наличие даже небольшого перерегулирования требует длительного времени на охлаждение [1].

Одним из возможных решений данной проблемы является построение адаптивных систем управления технологическими процессами [2], автоматически подбирающих коэффициенты ПИД-регулятора. Всю совокупность методов построения таких систем можно разбить на две группы. Первая группа — это классические методы, такие как метод Циглера-Никольса [3], частотный метод А.Г. Александрова [4], метод, заложенный в основу адаптивных ПИД-регуляторов фирмы Siemens [1]. Вторая группа — это методы на основе методологии искусственного интеллекта. Анализ некоторых из них предложен авторами в [5].

Анализ детерминированных методов автоматической настройки ПИД-регулятора показывает, что подобные методы не находят широкого применения в промышленности из-за своей сложности и необходимости постоянной и точной оценки модели ОУ.

Существует также другой подход, позволяющий решить рассматриваемую проблему за счет применения методов искусственного интеллекта. Одно из возможных направлений предложил Сигеру Омату в [6]. Он в своих исследованиях использовал методологию нейронных сетей (НС). Это связано с тем, что НС обладают нелинейными свойствами и способностью к обучению, что придает адаптивные свойства нейросетевым системам управления. В [6] было предложено несколько схем нейроуправления.

Экспериментально было установлено [7-9], что наиболее реализуемой с практической точки зрения является схема управления на основе автономной

коэффициентов ПИД-регулятора K_P , K_I , K_D с помощью НС. Такая схема позволяет учитывать нелинейные свойства объекта, не внося значительных изменений в существующую схему управления, что существенно снижает стоимость внедрения такой интеллектуальной надстройки (рис.1).



Рис.1 – Схема нейросетевого управления с самонастройкой.

В качестве предмета исследований была выбрана схема, указанная выше. Осуществлена ее практическая реализация для реального технологического объекта. Выполнена проверка работоспособности схемы для различных режимов изменения задания и загрузки. Показана необходимость использования двух нейронных сетей для управления нагревательными объектами.

2. Реализация системы управления с нейросетевым оптимизатором

В качестве ОУ была выбрана муфельная электронагревательная печь типа СНОЛ-1,6.2,5.1/11-И4. Реализованная система управления муфельной электропечью, функциональная схема которой представлена на рис.2, состояла из трех уровней управления.

На нижнем уровне находились термопара, измеряющая температуру рабочей зоны печи, и реле, посредством симисторов включающее нагревательный элемент.

Управляющий уровень (Simatic S7-300) осуществлял непосредственное управление муфельной электропечью. Контроллер получал данные от термопары через термопреобразователь, производил их обработку и отправлял на рабочую станцию, от которой он получал значение управляющего воздействия. В контроллере производилась широтно-импульсная модуляция управляющего воздействия и его выдача на исполнительный механизм включения нагревательного элемента.

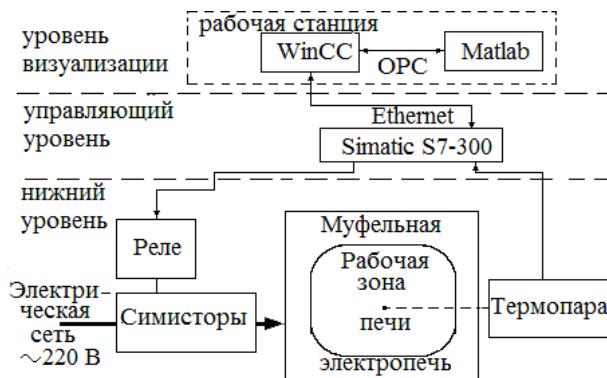


Рис.2 – Функциональная схема системы управления электрической печью.

На уровне визуализации находилась рабочая станция с системой визуализации WinCC и пакетом Matlab, в котором реализован нейросетевой оптимизатор параметров

ПИД-регулятора. WinCC и Matlab взаимодействовали между собой через OPC-интерфейс посредством тегов. Схема управления, реализованная в пакете Matlab, приведена на рис.3. WinCC передавала данные о температуре (*temper*) и уставку (*zadanie*) по температуре в Matlab. Запущенная в Matlab НС формировала значение управляющего воздействия (*upravl*), которое через WinCC поступало в контроллер. *Group1*, *group2* – это OPC блоки для соответственно записи и чтения значений в тегах WinCC.

В Matlab реализован ПИД-регулятор (PID) с нейронадстройкой (см. блок *S-function neuC_PID*, рис.3) – нейросетевой оптимизатор. U_i , U_p , U_d , $Upid$ – сигналы управления на выходе i -, p -, d -каналов и всего ПИД-регулятора соответственно. Блок *satur* использован для ограничения сигнала управления с ПИД-регулятора до диапазона [0;100] единиц. U – управляющее воздействие, подаваемое на ОУ.

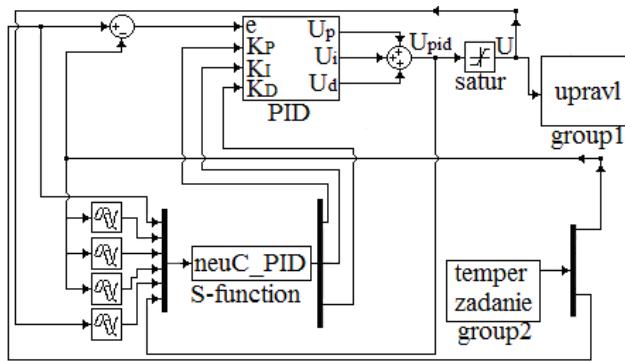


Рис.3 – Схема управления электрической печью с нейросетевым оптимизатором.

Для первого опыта график смены заданий выглядел следующим образом: $590^{\circ}\text{C} \rightarrow 640^{\circ}\text{C} \rightarrow 505^{\circ}\text{C} \rightarrow 590^{\circ}\text{C}$ и т.д. Для смены уставки температура должна удерживаться в 5% окрестности задания в течение 30 минут. Для второго опыта был использован следующий график заданий: $300^{\circ}\text{C} \rightarrow 330^{\circ}\text{C} \rightarrow 240^{\circ}\text{C} \rightarrow 300^{\circ}\text{C}$ и т.д.

Для рассматриваемой пустой печи с помощью идентифицированной ранее модели были подобраны коэффициенты ПИД-регулятора. Для перехода $505^{\circ}\text{C} \rightarrow 590^{\circ}\text{C}$: $K_p=0.6$, $K_i=6.976 \cdot 10^{-4}$, $K_D=0$. Для перехода $240^{\circ}\text{C} \rightarrow 300^{\circ}\text{C}$: $K_p=0.37$, $K_i=2.66 \cdot 10^{-4}$, $K_D=0$. Т.е. был выбран ПИ-регулятор, поскольку Д-канал существенно подвержен влиянию помех.

До начала каждого из экспериментов создавалась необученная НС. Она имела трехслойную структуру. Количество и вид входов НС были определены экспериментальным путем. Во входном слое расположено 5 нейронов (будут описаны ниже), в выходном слое – три нейрона (отвечают за коэффициенты ПИД-регулятора K_p , K_i , K_D). Экспериментально было установлено, что в данном случае 15 нейронов в скрытом слое является оптимальным числом. В скрытом слое была использована функция активации гиперболический тангенс, в выходном – линейная функция.

Нейронадстройка над ПИД-регулятором была реализована при помощи средств пакета Matlab – это S-функция *neuC_PID*. Она имеет шесть входов (первые пять совпадают со входами НС): текущее задание по температуре; задержанный на 1 секунду, на 20 секунд и на 40 секунд сигнал по выходу ОУ; задержанный на 1 секунду сигнал управления после ограничителя; задержанный на 1 секунду сигнал управления до ограничителя. Выбор времени задержки сигналов был произведен на основе оценки соотношения объема рабочего пространства печи к ее мощности. Внутри функции

реализован алгоритм оперативного обучения описанной НС, представляющий собой алгоритм обратного распространения ошибки [6], модифицированный рядом условий. Рассмотрим эти условия подробнее.

3. Особенности реализации метода обучения нейронной сети

Основные дополнения, внесенные в схему С.Омату: 1) разработан метод формирования начального состояния НС; 2) разработаны правила изменения скорости обучения НС; 3) разработаны правила обучения выходов НС, формирующих p -канал и i -канал; 4) определено число эпох обучения НС на каждом такте работы схемы. Данные модификации частично описаны в [10]. Рассмотрим детально пункт 1 и 3.

3.1. Формирование начального состояния работы НС

В [6] предлагается первоначально автономно обучать НС на данных об ОУ. Однако в реальных условиях производства в большинстве случаев известен лишь набор коэффициентов K_P , K_I , K_D , записанных в ПИД-регуляторе. Для автономного обучения этих данных недостаточно. Как показали опыты, использование необученной сети в режиме оперативного обучения также не дает требуемых результатов, поскольку в этом случае система довольно часто не сходится к заданию. Это связано с тем, что в начальный момент времени, в большинстве случаев, значения выходов НС оказываются слишком неадекватны для данной системы (особенно для интегрального канала). Поэтому необходимо в начальный момент времени задавать «начальную точку» для работы НС.

Эта проблема в данной работе решалась следующим образом: весовые коэффициенты выходного слоя НС задавались таким образом, чтобы в момент запуска системы выход НС совпал с известными коэффициентами ПИД-регулятора. Применение линейных функций активации на выходном слое позволило добиться этого путем решения системы линейных уравнений (1), где неизвестными выступали весовые коэффициенты выходного слоя. Смещения выходного слоя приравнивались к нулю.

Для того, чтобы построить подобную систему уравнений первые 15 секунд нейросетевой оптимизатор вычислял значения выхода с нейронов скрытого слоя сети HO [1x15]. Эти вектора формировали матрицу HOm [15x15], которая использовалась как известные значения в уравнениях. Также как и значения коэффициентов ПИД-регулятора (приведено для первого опыта) Kp_m [15x1] = {0.6, 0.6, ..., 0.6}, Ki_m [15x1] = { $6.976 \cdot 10^{-4}$, $6.976 \cdot 10^{-4}$, ..., $6.976 \cdot 10^{-4}$ }, Kd_m [15x1] = {0, 0, ..., 0}.

$$\begin{cases} HOm \cdot W^{(2)}(:,1) = Kp_m \\ HOm \cdot W^{(2)}(:,2) = Ki_m, \\ HOm \cdot W^{(2)}(:,3) = Kd_m \end{cases} \quad (1)$$

где $W^{(2)}(:,i)$ - это i -й столбец матрицы весовых коэффициентов между скрытым и выходным слоем. Размер матрицы $W^{(2)}$ составлял [15x3].

Подобные действия позволили в момент передачи управления коэффициентами K_P , K_I , K_D НС получать на ее выходе не случайную величину, а значения коэффициентов, которые до этого использовались для управления объектом. Затем включалось оперативное обучение НС, посредством которого и изменялись K_P , K_I , K_D .

3.2. Модификации для обучения нейрона, формирующего K_I

Для влияния на качество переходного процесса за счет варьирования скоростью прироста интегральной составляющей регулятора в алгоритм управления был внесен ряд

условий. Причем при очередной смене задания определялось, находится ли система в стабильном состоянии, то есть скорость изменения температуры должна быть менее 0.5 градуса за последние 2 такта работы схемы (1 такт = 20 секундам), а текущее рассогласование – менее 3% от разницы между текущим заданием и предыдущим. Если система стабильна, то текущее задание и сигнал управления сохранялись в специальный массив. Такие воздействия принимались как эталонные для соответствующих заданий.

В течение любого переходного процесса при первом пересечении задания фиксировалось значение сигнала управления в i -канале. Если это значение меньше эталонного, то принималось решение об увеличении K_I , а если больше – то об уменьшении.

Ускорение обучения i -канала в сторону увеличения начиналось при соблюдении следующих условий: переходный процесс уже достиг своего первого максимума после пересечения задания, скорость изменения температуры за последние 2 такта работы схемы менее 0.5 градуса, а текущее рассогласование больше 3% от разницы между текущим заданием и предыдущим. Ускорение обучения i -канала в сторону уменьшения начиналось при соблюдении следующих условий: скорость изменения температуры за последние 2 такта работы схемы более 0.5 градуса, а текущее рассогласование больше 3% от разницы между текущим заданием и предыдущим. Скорость обучения в обоих случаях возрастала в 100 раз.

3.3. Модификации для обучения нейрона, формирующего K_P

В алгоритм управления был внесен ряд условий, варьирующих скорость обучения нейрона, формирующего коэффициент пропорциональной составляющей регулятора, для влияния на качество переходного процесса. Если после пересечения задания перерегулирование составляло более 5% разницы между предшествующим заданием и текущим и не было принято решение об уменьшении значения K_I (см. п.3.2), а модуль производной температуры за 20 секунд превышал 2% разницы между предшествующим заданием и текущим, то производилось увеличение скорости обучения р-канала в 100 раз. Вызванное этим уменьшение значения K_P позволяло в ходе следующих переходных процессов снизить перерегулирование.

Кроме того, если при смене уставки с меньшей на большую в процессе нагревания в промежутке от 10% до 80% завершенности переходного процесса скорость нарастания сигнала температуры падала ниже 0.5°C за 20 секунд (прекращался рост температуры), то принималось решение об увеличении скорости обучения нейрона, формирующего коэффициент p -канала, в 100 раз. Это позволило избегать ситуации, когда коэффициент K_P недостаточен для вывода объекта на задание.

4. Результаты экспериментов

В рамках исследования были проведены два эксперимента, каждый для двух систем управления — для системы с нейросетевым оптимизатором и для обычного ПИД-регулятора. Первоначально использовалась пустая печь для отслеживания следующего графика заданий: $590^{\circ}\text{C} \rightarrow 640^{\circ}\text{C} \rightarrow 505^{\circ}\text{C} \rightarrow 590^{\circ}\text{C}$ (первый опыт) и $300^{\circ}\text{C} \rightarrow 330^{\circ}\text{C} \rightarrow 240^{\circ}\text{C} \rightarrow 300^{\circ}\text{C}$ (второй опыт) (11 смен задания). Затем печь загружалась литой заготовкой, объем которой составлял примерно 30% объема печи, и для нее повторялся указанный график заданий (вновь 11 смен заданий). Затем печь разгружалась, и для нее также повторялся указанный график заданий (11 смен заданий). Каждый переходный процесс считался завершенным, если после входа в 5% окрестность задания проходило 30 минут. Затем происходила смена задания. Для этого был

использован счетчик времени, обнуляемый в случае, если выходной сигнал температуры печи выходил за пределы окрестности. Нейросетевой оптимизатор начинал эксперимент с известными значениями коэффициентов ПИД-регулятора, указанными ранее как оптимальные для пустой печи.

Результаты экспериментов, полученные с использованием обычного ПИД-регулятора для первого варианта графика заданий, приведены на рис.4. Качество переходных процессов для системы управления с коэффициентами, подобранными для пустой печи, существенно падает (до 15 % по перерегулированию) при работе на загруженной печи. Время опыта - 34,4 часа.

Результаты экспериментов, полученные с помощью нейросетевого оптимизатора параметров ПИД-регулятора, приведены на рис.5. Время опыта - 27,72 часа. Это на 19% быстрее, чем время опыта для ПИД-регулятора. На рис. 6 приведены графики изменения коэффициентов K_P и K_I ПИД-регулятора нейросетевым оптимизатором. Нейросетевой оптимизатор перенастроил коэффициенты на новый режим работы объекта (загруженная печь), что позволило сохранить качество переходных процессов практически на уровне, наблюдавшемся до загрузки печи. После разгрузки нейросетевой оптимизатор вернул исходные коэффициенты для незагруженной печи (см. рис. 6). Это заняло некоторое время, поэтому сразу после разгрузки наблюдается ухудшение качества переходных процессов.

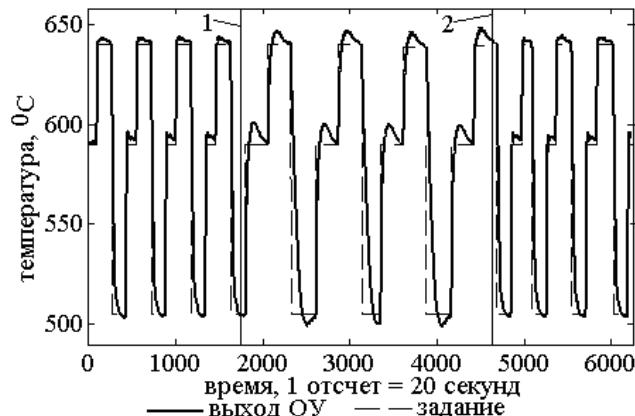


Рис. 4 – Выход ОУ, управляемого ПИД-регулятором с коэффициентами $K_P = 0.6$, $K_I = 6.976 \cdot 10^{-4}$ (1 - момент загрузки печи, 2 - момент разгрузки).

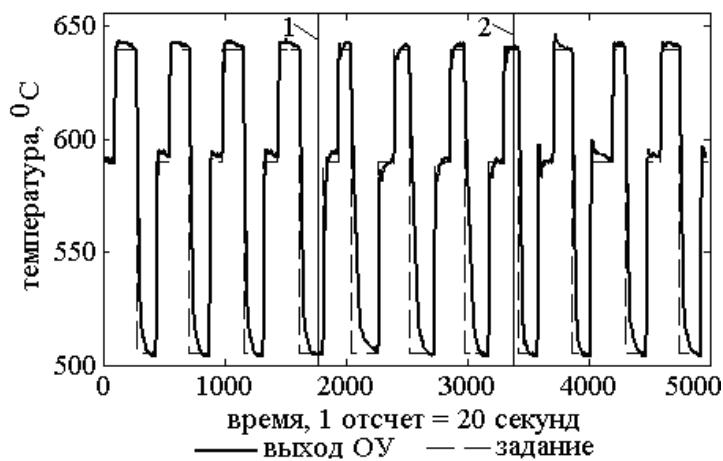


Рис. 5 – Выход ОУ, управляемого нейросетевым оптимизатором (1 - момент загрузки печи, 2 - момент разгрузки).

Второй опыт предполагал использование следующего графика заданий: $300^{\circ}\text{C} \rightarrow 330^{\circ}\text{C} \rightarrow 240^{\circ}\text{C} \rightarrow 300^{\circ}\text{C}$. Коэффициенты ПИД-регулятора были настроены на переход $240^{\circ}\text{C} \rightarrow 300^{\circ}\text{C}$. Результаты экспериментов с обычным регулятором приведены на рис.7.

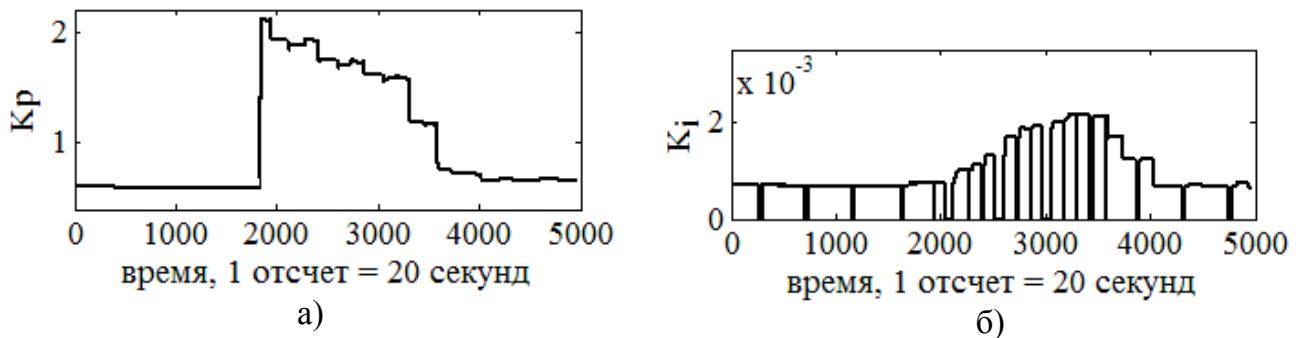


Рис. 6 – График изменения коэффициентов KP (а) и KI (б).

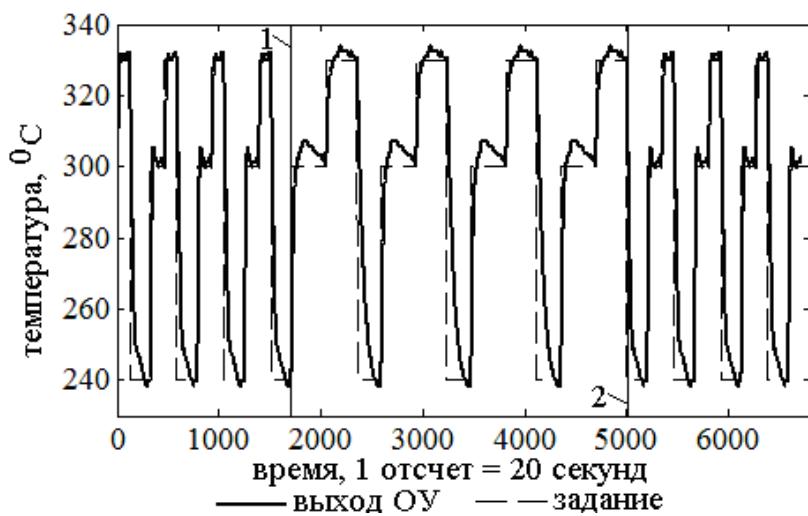


Рис. 7 – Выход ОУ, управляемого ПИД-регулятором с коэффициентами $K_p=0.37$, $K_i=2.66 \cdot 10^{-4}$ (1 - момент загрузки печи, 2 - момент разгрузки).

Следует отметить, что даже при незагруженной печи коэффициенты регулятора позволяют получать приемлемые результаты только для процесса нагревания. При охлаждении наблюдается затянутый по времени переходный процесс, вызванный недостаточным значением коэффициента K_I . После загрузки наблюдается обратная картина - процесс охлаждения проходит достаточно качественно, однако процессы нагревания проходят длительное время и, хотя перерегулирование при этом составляет лишь порядка 10%, недостаточное значение коэффициента K_P приводит к насыщению i-канала, которое затем преодолевается достаточно значительное время. Время опыта - 37,33 часа.

Нейросетевой оптимизатор, использовавшийся в первой серии опытов, был применен и для нового графика заданий. Однако, при проведении эксперимента даже до момента загрузки печи возникла следующая проблема. При охлаждении печи нейросетевой оптимизатор определяет недостаточность коэффициента K_I и производит его увеличение (рис.8 и 9). Однако подобное увеличение приводит к существенному

перерегулированию (17%) на следующем же нагревании $240^{\circ}\text{C} \rightarrow 300^{\circ}\text{C}$. Затем, как видно из рис.9 б, система производит уменьшение K_I и переход $300^{\circ}\text{C} \rightarrow 330^{\circ}\text{C}$ проходит нормально. Однако, на следующем же охлаждении проблема вновь повторяется. Таким образом, очевидно, что оптимизатор в данной конфигурации не способен эффективно управлять подбором коэффициентов ПИД-регулятора при некотором чередовании процессов нагрева и охлаждения.

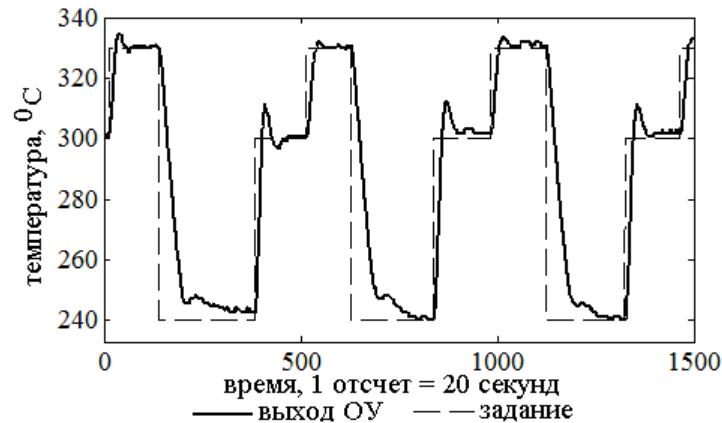


Рис. 8. - Выход ОУ (пустая печь), управляемого нейросетевым оптимизатором для графика заданий $3000^{\circ}\text{C} \rightarrow 3300^{\circ}\text{C} \rightarrow 2400^{\circ}\text{C} \rightarrow 3000^{\circ}\text{C}$.

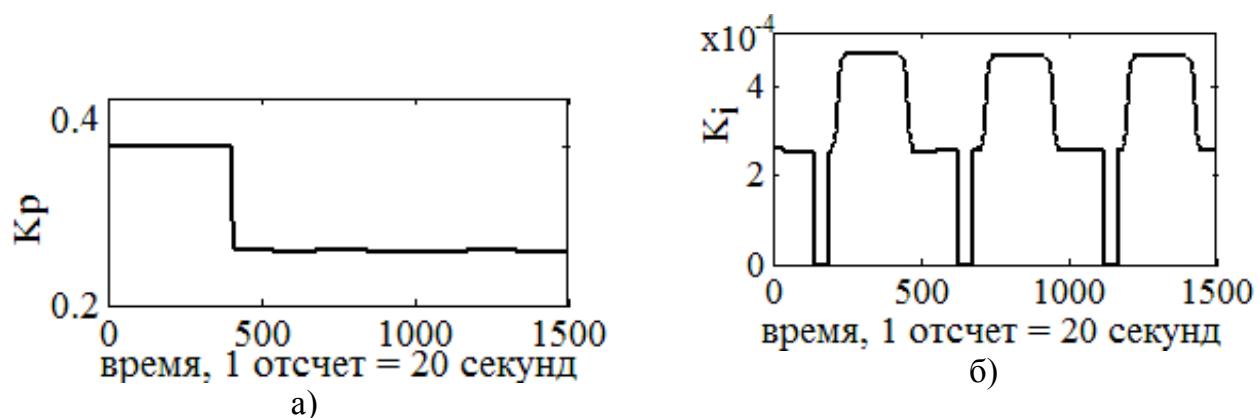


Рис. 9 – График изменения коэффициентов KP (а) и KI (б).

В связи с вышеуказанными проблемами было предложено реализовать нейросетевой оптимизатор как две отдельные нейронные сети, которые пользуются одним и тем же алгоритмом обучения и набором правил, описанных в п.3. Переключение между нейронными сетями производится в момент очередной смены задания. При этом определяется тип переходного процесса (нагрев или охлаждение) и производится выбор соответствующей сети. Начинают свою работу обе сети с одинаковых коэффициентов регулятора $K_P=0.37$, $K_I=2.66 \cdot 10^{-4}$. Их начальное состояние формируется в соответствии с методом, описанным в п.3.1. Предложенный подход позволил решить указанную выше проблему. На рис. 10, 11 приведены графики переходных процессов и изменения коэффициентов регулятора для пустой печи.

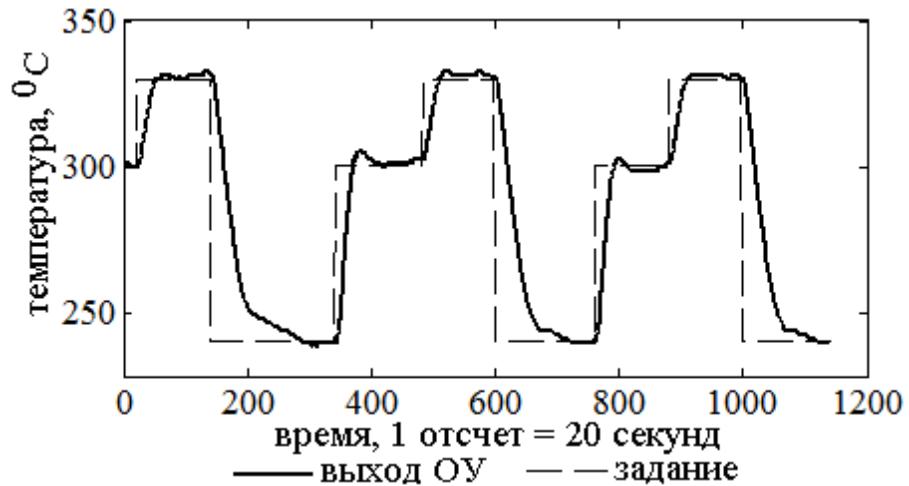


Рис. 10 – Выход ОУ (пустая печь), управляемого нейросетевым оптимизатором с двумя нейронными сетями для картины заданий 3000С→3300С→2400С→3000С.

Первая нейронная сеть, отвечающая за нагрев, не производит изменения коэффициентов регулятора, поскольку нагревание производится в соответствии с требованиями регламента. В свою очередь, нейронная сеть, отвечающая за охлаждение, производит увеличение значения коэффициента K_I , что приводит к улучшению качества переходных процессов при охлаждении, но никак не влияет на процессы нагрева.

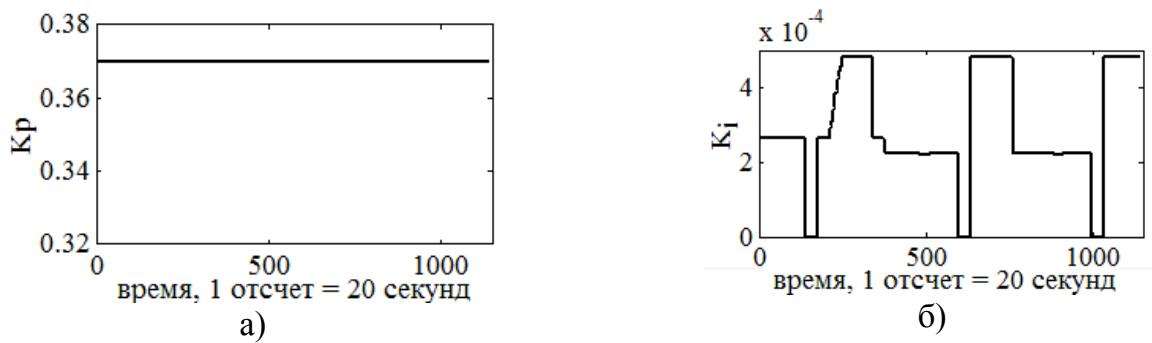


Рис. 11 – График изменения коэффициентов КР (а) и КI (б).

Дальнейший опыт с загрузкой и разгрузкой печи проводился с использованием нейросетевого оптимизатора, содержащего две нейронных сети. Результаты его работы приведены на рис.12,13.

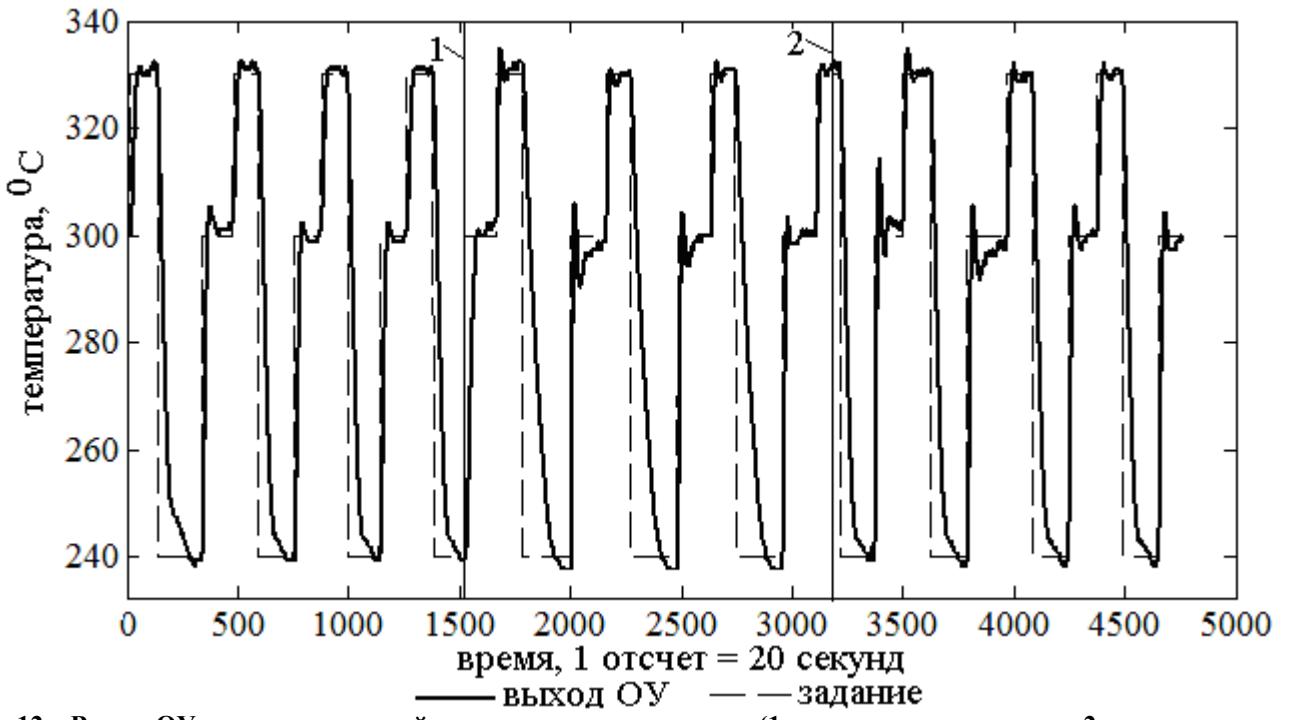


Рис. 12 – Выход ОУ, управляемого нейросетевым оптимизатором (1 - момент загрузки печи, 2 - момент разгрузки).

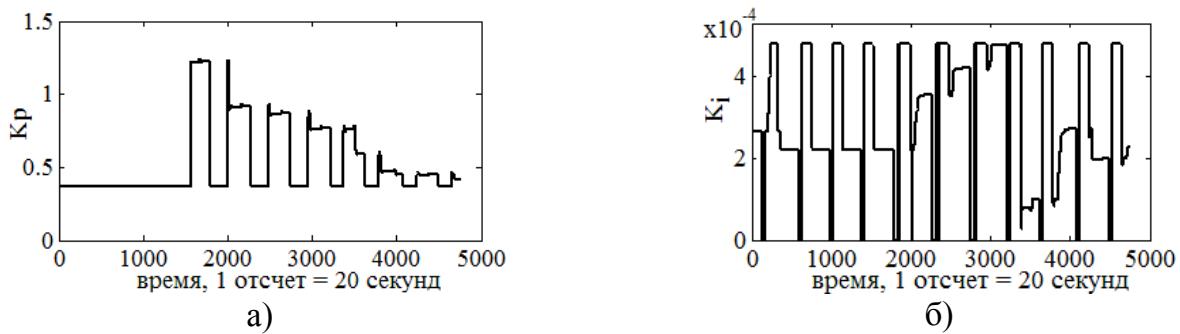


Рис. 13 – График изменения коэффициентов KP (а) и KI (б).

Время опыта составило 26,47 часа. Это на 29% быстрее, чем время опыта для ПИД-регулятора. В соответствии с приведенными графиками можно сделать вывод о том, что нейросетевой оптимизатор в конфигурации с двумя сетями перенастроил коэффициенты на новый режим работы печи (загруженное состояние). После разгрузки нейросетевой оптимизатор вернул исходные коэффициенты для незагруженной печи (см. рис.12). Это заняло некоторое время, поэтому сразу после разгрузки наблюдается ухудшение качества переходных процессов.

5. Заключение

Предлагаемые усовершенствования, внесенные в схему реализации нейросетевого оптимизатора параметров ПИД-регулятора, позволили обеспечить стабильную работу НС и ее обучаемость в контуре управления в реальном масштабе времени. Результаты проведенных экспериментов показывают, что в случае если для нагрева и охлаждения печи необходимы различные наборы коэффициентов, то нейросетевой оптимизатор целесообразнее строить на двух нейронных сетях, использующих один и тот же принцип

обучения и базу условий на выбор скорости обучения отдельных нейронов. Ввиду того, что на выполнение графика задания в обоих опытах система управления с нейросетевым оптимизатором затратила меньше времени (в среднем, на 24%), можно сделать вывод о том, что использование подобного оптимизатора позволило бы интенсифицировать производство на реальном объекте управления, что принесло бы ощутимый экономический эффект.

Литература

1. Pfeiffer B.-M. Towards «plug and control»: self-tuning temperature controller for PLC // International journal of Adaptive Control and Signal Processing. 2000. No.14. P.519-532.
2. Astrom K. J., Hagglund T., Hang C. C., Ho W. K. Automatic tuning and adaptation for PID controllers. A survey. // IFAC J. Control Eng. Practice. 1993. Vol.1, No.4. P.699–714.
3. Ziegler, J., Nichols, N. Optimum settings for automatic controllers // Trans. ASME. 1942.No.65. P.759–768.
4. Alexandrov A.G., Palenov M.V Self-tuning PID-I controller // Proceedings of the 18th IFAC World Congress. Milano, Italy, 2011. P. 3635-3640.
5. Еременко Ю.И., Полещенко Д.А., Глущенко А.И. Анализ методов реализации схемы нейросетевого управления с самонастройкой // Приборы и системы. Управление. Контроль. Диагностика. 2012. №6. С.50-55.
6. Omatsu S., Khalid M., Yusof R. Neuro-Control and its Applications. London: Springer, 1995. 255 p.
7. Chen J., Huang T. Applying neural networks to on-line updated PID controllers for nonlinear process control // Journal of Process Control. 2004. No.14. P.211-230.
8. Fang M., Zhuo Y., Lee Z. The application of the self-tuning neural network PID controller on the ship roll reduction in random waves // Ocean Engineering. 2010. No.37. P.529–538.
9. Reyes J., Astorga C., Adam M., Guerrero G. Bounded neuro-control position regulation for a geared DC motor // Engineering Applications of Artificial Intelligence. 2010. No.23. P.1398–1407.
10. Еременко, Ю.И., Полещенко, Д.А., Глущенко, А.И. К вопросу о совершенствовании схемы ПИД-нейрорегулятора с самонастройкой // Системы управления и информационные технологии. - 2012. - №4. - С.27-31.

ОЦЕНИВАНИЕ ПАРАМЕТРОВ ЛИНЕЙНЫХ МНОГОМЕРНЫХ ПО ВХОДУ ARX СИСТЕМ НЕЦЕЛОГО ПОРЯДКА С ПОМЕХОЙ В ВЫХОДНОМ СИГНАЛЕ

Иванов Д.В.

Самарский государственный университет путей сообщения
г. Самара, dvi85@list.ru
Ширинов И.Р.

Самарский государственный университет путей сообщения
г. Самара, shirinov-89@mail.ru

Аннотация

Предложен алгоритм, являющийся обобщением метода наименьших квадратов, который позволяет получать сильно состоятельные оценки параметров линейных ARX (Auto Regressive model with External input) систем при наличии помех наблюдения в выходном сигнале в условиях отсутствия информации о законе распределения помех.

Ключевые слова: параметрическая идентификация, метод наименьших квадратов, разность дробного порядка, разностностные уравнения.

1. Введение

Аппарат дробного математического анализа имеет множество приложений таких как теория вязкоупругости, теория хаоса, фракталы, для описания диэлектрических материалов, электрохимических процессов, траффики в компьютерных сетях. Поэтому развитие методов идентификации систем дробного порядка является актуальной задачей.

Производные и разности дробного порядка зависят от всех предыдущих значений временного ряда поэтому идентификация моделей дробного порядка более сложна в сравнении с моделями целого порядка. Обзор методов идентификации моделей дробного порядка представлен в [1]. В настоящее время активно развиваются методы нелинейного оценивания параметров динамических систем [2,3].

В [4] предложен метод оценивания параметров авторегрессии нецелого порядка с помехой в выходном сигнале, в [5,6] метод оценивания динамической системы с помехой в выходном сигнале, обобщение метода на динамическую систему нецелого порядка с ошибками в переменных приведено в [7].

В данной статье дано обобщение предложенного в [8] алгоритма идентификации на случай многомерной по входу линейной динамической системы дробного порядка. Доказана сильная состоятельность оценок многомерных по входу линейных динамических систем дробного порядка при помехе класса мартингал-разность.

2. Постановка задачи

Рассмотрим линейную ARX систему дробного порядка, описываемую следующими стохастическими уравнениями с дискретным временем $i = \dots -1, 0, 1, \dots$:

$$z_i = \sum_{m=1}^r b_0^{(m)} \Delta^{\alpha_m} z_{i-1} + \sum_{k=1}^d \sum_{m=1}^{r_k^{(k)}} a_0^{(mk)} \Delta^{\beta_m^{(k)}} x_i^{(k)} + \zeta_i, \quad y_i = z_i + \xi_i, \quad (1)$$

где $0 < \alpha_1 \dots < \alpha_r$, $0 < \beta_1^{(k)} \dots < \beta_{r_k^{(k)}}^{(k)}$, $\Gamma(\alpha) = \int_0^\infty e^{-t} t^{\alpha-1} dt$, $k = \overline{1, d}$

$$\Delta^{\alpha_m} z_i = \sum_{j=0}^i (-1)^j \binom{\alpha_m}{j} z_{i-j}, \quad \Delta^{\beta_m^{(k)}} x_i^{(k)} = \sum_{j=0}^i (-1)^j \binom{\beta_m^{(k)}}{j} x_{i-j}^{(k)}, \quad \binom{\alpha_m}{j} = \frac{\Gamma(\alpha_m + 1)}{\Gamma(j+1)\Gamma(\alpha_m - j + 1)},$$

$$\binom{\beta_m^{(k)}}{j} = \frac{\Gamma(\beta_m^{(k)} + 1)}{\Gamma(j+1)\Gamma(\beta_m^{(k)} - j + 1)},$$

z_i , y_i – ненаблюдаемая и наблюдаемая выходные переменные;

$x_i^{(k)}$ – наблюдаемая входная переменная; ξ_i – помеха наблюдения в выходном сигнале;

ζ_i – ошибка в уравнении;

Предположим, что выполняются следующие условия:

1. Множество \tilde{B} , которому априорно принадлежат истинные значения устойчивой, наблюдаемой и управляемой билинейной динамической системы является компактом.

2. Помехи $\{\xi_i\}$, $\{\zeta_i\}$ статистически независимые последовательности. $\{\xi_i\}, \{\zeta_i\}$ – стационарные в совокупности в узком смысле последовательности независимых случайных векторов с $E\{\xi_i\} = 0, E\{\xi_i^2\} = \sigma_\xi^2 > 0, E\{\zeta_i\} = 0, E\{\zeta_i^2\} = \sigma_\zeta^2 > 0$, и для некоторых постоянных π_ξ и π_ζ : $|\xi_i| < \pi_\xi$ и $|\zeta_i| < \pi_\zeta$ почти наверное (п.н.), где E – оператор математического ожидания.

3. Входной сигнал x_i является случайным процессом с $E(x_i) = 0$, $E(x_i^2) = \sigma_x^2 < \infty$ и истинные значения параметров $b_0, a_0^{(1)}, \dots, a_0^{(d)}$ удовлетворяют условию

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \left(\begin{pmatrix} (\varphi_z^{(i)})^T & | & (\varphi_x^{(i)})^T & | & \dots & | & (\varphi_x^{(id)})^T \end{pmatrix}^T \right)^T \left(\begin{pmatrix} (\varphi_z^{(i)})^T & | & (\varphi_x^{(i)})^T & | & \dots & | & (\varphi_x^{(id)})^T \end{pmatrix} \right) = H \quad \text{п.н.},$$

$$\varphi_z^{(i)} = \left(\sum_{j=0}^i (-1)^j \binom{\alpha_1}{j} z_{i-j-1}, \dots, \sum_{j=0}^i (-1)^j \binom{\alpha_r}{j} z_{i-j-1} \right)^T,$$

$$\varphi_x^{(ik)} = \left(\sum_{j=0}^i (-1)^j \binom{\beta_1^{(k)}}{j} x_{i-j}^{(k)}, \dots, \sum_{j=0}^i (-1)^j \binom{\beta_{r_1^{(k)}}^{(k)}}{j} x_{i-j}^{(k)} \right)^T,$$

причем H существует, ограничена и положительно определена.

4. $\{x_i^{(k)}\}$ статистически не зависит от $\{\xi_i\}, \{\zeta_i\}$.

5. Априорно известно отношение $\gamma = \frac{\sigma_\zeta^2}{\sigma_\xi^2}$.

Требуется определять оценки неизвестных коэффициентов динамической системы описываемой уравнением (1) по наблюдаемым последовательностям $y_i, x_i^{(k)}$ при известных порядках $r, r_1, \alpha_m, \beta_m^{(k)}$.

3. Критерий для оценивания параметров

Система может быть записана как линейная регрессия

$$y_i = \varphi_i^T \theta + \varepsilon_i,$$

(2)

$$\text{где } \varphi_i = \left(\begin{pmatrix} (\varphi_y^{(i)})^T & | & (\varphi_x^{(i)})^T \end{pmatrix}^T \right)^T, \varphi_y^{(i)} = \left(\sum_{j=0}^i (-1)^j \binom{\alpha_1}{j} y_{i-j-1}, \dots, \sum_{j=0}^i (-1)^j \binom{\alpha_r}{j} y_{i-j-1} \right)^T, \theta_0 = (b_0^T \mid a_0^T)^T,$$

$$b_0 = (b_0^{(1)}, \dots, b_0^{(r)})^T, a_0 = (a_0^{(1)} \mid \dots \mid a_0^{(d)})^T, a_0^{(k)} = (a_0^{(0k)}, \dots, a_0^{(r_1^{(k)} k)})^T, \varepsilon_i = \zeta_i + \xi_i - b_0^T \varphi_\xi^{(i)},$$

$$\varphi_\xi^{(i)} = \left(\sum_{j=0}^i (-1)^j \binom{\alpha_1}{j} \xi_{i-j-1}, \dots, \sum_{j=0}^i (-1)^j \binom{\alpha_r}{j} \xi_{i-j-1} \right)^T$$

Из предположения (2) следует, что обобщенная ошибка имеет нулевое среднее значение, а из предположения (3) - что ее локальная дисперсия с вероятностью 1 будет равна:

$$\bar{\sigma}_\varepsilon^2 = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N E((\varepsilon_i(b_0, i))^2) = \bar{\sigma}_\xi^2 (1 + \gamma + b_0^T H_\alpha b_0) = \bar{\sigma}_\xi^2 \omega(b_0).$$

$$\text{где } \bar{\sigma}_\xi^2 = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \xi_i^2, \quad \bar{\sigma}_\zeta^2 = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \zeta_i^2,$$

$$H_\alpha = \lim_{N \rightarrow \infty} \frac{1}{N} E \left[\sum_{i=1}^N \varphi_\xi^{(i)} (\varphi_\xi^{(i)})^T \right] = \begin{pmatrix} h_\alpha^{(11)} & \dots & h_\alpha^{(r1)} \\ \vdots & \ddots & \vdots \\ h_\alpha^{(1r)} & \dots & h_\alpha^{(rr)} \end{pmatrix},$$

$$\begin{aligned}
h_{\alpha}^{(mn)} &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=0}^{N-1} E \left(\sum_{j=0}^i (-1)^j \binom{\alpha_m}{j} \xi_{i-j-1} \cdot \sum_{j=0}^i (-1)^j \binom{\alpha_n}{j} \xi_{i-j-1} \right) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{j=0}^{N-1} \binom{\alpha_m}{j} \binom{\alpha_n}{j} \frac{N-j}{N} \\
&= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{j=0}^{N-1} \binom{\alpha_m}{j} \binom{\alpha_n}{j} \frac{N-j}{N}, \quad m=1, r, n=1, r.
\end{aligned}$$

Тогда определим оценку $\hat{\theta}(N)$ неизвестных параметров θ из условия минимума суммы взвешенных квадратов обобщённых ошибок $\varepsilon_i^2(b_0, a_0^{(1)}, \dots, a_0^{(d)}, i)$ с весом $\omega(b_0)$, т.е.

$$\min_{\theta \in \tilde{\mathcal{B}}} \sum_{i=1}^N \frac{(y_i - \varphi_i^T \theta)^2}{1 + \gamma + b^T H_\alpha b}, \quad (3)$$

Имеет место, следующая теорема:

Теорема 1. Пусть некоторый случайный процесс $\{y_i, i = \dots -1, 0, 1, \dots\}$ описывается уравнением (1) с начальными нулевыми условиями и выполняются предположения 1-5. Тогда оценка $\hat{\theta}(N)$, определяемая выражением (3) с вероятностью 1 при $N \rightarrow \infty$, существует, единственная и является сильно состоятельной оценкой, т.е.

$$\hat{\theta}(N) \xrightarrow[N \rightarrow \infty]{\text{П.Н.}} \theta_0.$$

Доказательство.

Определим

$$\begin{aligned}
&\frac{1}{N} U_N(b, a^{(1)}, \dots, a^{(d)}) = \\
&= N^{-1} \sum_{i=1}^N \left(z_i + \xi_i - (\varphi_z^{(i)} + \varphi_\xi^{(i)})^T b - \sum_{k=1}^d (\varphi_x^{(ik)})^T a^{(k)} \right)^2 = \\
&= N^{-1} \sum_{i=1}^N \left(\xi_i + \zeta_i + (\varphi_z^{(i)})^T b_0 + \sum_{k=1}^d (\varphi_x^{(ik)})^T a_0^{(k)} - (\varphi_z^{(i)} + \varphi_\xi^{(i)})^T b - \sum_{k=1}^d (\varphi_x^{(ik)})^T a^{(k)} \right)^2 = \\
&= N^{-1} \sum_{i=1}^N \left(\xi_i + \zeta_i - (\varphi_z^{(i)})^T \tilde{b} - \sum_{k=1}^d (\varphi_x^{(ik)})^T \tilde{a}^{(k)} - (\varphi_\xi^{(i)})^T b \right)^2 = \\
&= v_1 + v_2 + v_3; \\
v_1 &= N^{-1} \sum_{i=1}^N \left(\xi_i^2 + \zeta_i^2 + b^T \varphi_\xi^{(i)} (\varphi_\xi^{(i)})^T b - 2\xi_i (\varphi_\xi^{(i)})^T b - 2\zeta_i (\varphi_\xi^{(i)})^T b \right), \\
(4) \quad v_2 &= \frac{1}{N} \sum_{i=1}^N \tilde{\theta}^T \left(\begin{array}{c|c|c|c} (\varphi_z^{(i)})^T & (\varphi_x^{(i1)})^T & \dots & (\varphi_x^{(id)})^T \end{array} \right)^T \left(\begin{array}{c|c|c|c} (\varphi_z^{(i)})^T & (\varphi_x^{(i1)})^T & \dots & (\varphi_x^{(id)})^T \end{array} \right) \tilde{\theta}, \\
v_3 &= 2N^{-1} \sum_{i=1}^N \left(-\xi_i (\varphi_z^{(i)})^T \tilde{b} - \sum_{k=1}^d \xi_i (\varphi_x^{(ik)})^T \tilde{a}^{(k)} + b^T \varphi_\xi^{(i)} (\varphi_z^{(i)})^T \tilde{b} - \zeta_i (\varphi_z^{(i)})^T \tilde{b} - \sum_{k=1}^d \zeta_i (\varphi_x^{(ik)})^T \tilde{a}^{(k)} + \sum_{k=1}^d b^T \varphi_\xi^{(i)} (\varphi_x^{(ik)})^T \tilde{a}^{(k)} \right)
\end{aligned}$$

где $\tilde{b} = b - b_0$, $\tilde{a}^{(k)} = a^{(k)} - a_0^{(k)}$, $\tilde{\theta} = \theta - \theta_0$.

Применив лемму [9] для случайных процессов, получаем, что

$$v_1 \xrightarrow[N \rightarrow \infty]{\text{П.Н.}} \bar{\sigma}_\xi^2 (1 + \gamma + b^T H_\alpha b),$$

так как из 3

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \left(\begin{pmatrix} \varphi_z^{(i)} \\ \varphi_x^{(i)} \end{pmatrix}^T \mid \begin{pmatrix} \varphi_x^{(i1)} \\ \varphi_x^{(id)} \end{pmatrix}^T \mid \dots \mid \begin{pmatrix} \varphi_z^{(id)} \\ \varphi_x^{(id)} \end{pmatrix}^T \right)^T \left(\begin{pmatrix} \varphi_z^{(i)} \\ \varphi_x^{(i)} \end{pmatrix}^T \mid \begin{pmatrix} \varphi_x^{(i1)} \\ \varphi_x^{(id)} \end{pmatrix}^T \mid \dots \mid \begin{pmatrix} \varphi_z^{(id)} \\ \varphi_x^{(id)} \end{pmatrix}^T \right) = H \text{ п.н.,}$$

to $v_2 \xrightarrow[N \rightarrow \infty]{\text{P.H.}} \tilde{\theta}^T H \tilde{\theta}$.

Первые два слагаемых в сумме v_3 в силу условий 2, 3, 4 удовлетворяют условиям леммы [10] и, следовательно:

$$N^{-1} \sum_{i=1}^N \xi_i \begin{pmatrix} \varphi_z^{(i)} \\ \varphi_x^{(i)} \end{pmatrix}^T \tilde{b} \xrightarrow[N \rightarrow \infty]{\text{P.H.}} 0 \text{ п.н., } \forall \theta \in \tilde{B}.$$

Заметим, что

$$N^{-1} \sum_{i=1}^N b^T \varphi_\xi^{(i)} \begin{pmatrix} \varphi_z^{(i)} \\ \varphi_x^{(i)} \end{pmatrix}^T \tilde{b} = \frac{1}{N} \sum_{i=1}^N b^T H_{\xi z} \tilde{b},$$

$$H_{\xi z} = \lim_{N \rightarrow \infty} \frac{1}{N} E \left[\sum_{i=1}^N \varphi_\xi^{(i)} \begin{pmatrix} \varphi_z^{(i)} \\ \varphi_x^{(i)} \end{pmatrix}^T \right] = \begin{pmatrix} h_{\xi z}^{(11)} & h_{\xi z}^{(21)} & \dots & h_{\xi z}^{(r1)} \\ h_{\xi z}^{(21)} & h_{\xi z}^{(22)} & \dots & h_{\xi z}^{(r1)} \\ \vdots & \vdots & \ddots & \vdots \\ h_{\xi z}^{(1r)} & h_{\xi z}^{(2r)} & \dots & h_{\xi z}^{(rr)} \end{pmatrix},$$

$$h_\xi^{(mn)} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=0}^{N-1} E \left(\sum_{j=0}^i (-1)^j \binom{\alpha_m}{j} \xi_{i-j-1} \cdot \sum_{j=0}^i (-1)^j \binom{\alpha_n}{j} z_{i-j-1} \right), m = \overline{1, r}, n = \overline{1, r},$$

таким образом, (4) можно представить в виде r^2 слагаемых, каждое из которых в силу предположений 2-3 по лемме [10] сходится к нулю.

Можно доказать, что и все остальные слагаемые в v_3 сходятся к нулю с вероятностью 1 при $N \rightarrow \infty$. Таким образом, имеем: $v_3 \xrightarrow[N \rightarrow \infty]{\text{P.H.}} 0$.

Окончательно имеем

$$N^{-1} U_N(b, a^{(1)}, \dots, a^{(d)}) \xrightarrow[N \rightarrow \infty]{\text{P.H.}} \bar{\sigma}_\zeta^2 + \bar{\sigma}_\xi^2 + \tilde{\theta}^T H \tilde{\theta} + \bar{\sigma}_\xi^2 b^T H_\alpha b \equiv \bar{U}(b, a^{(1)}, \dots, a^{(d)}) \quad (5)$$

Покажем, что решение задачи

$$\min \omega^{-1}(b) \bar{U}(b, a^{(1)}, \dots, a^{(d)}), \theta \in \tilde{B} \quad (6)$$

существует и достигается в единственной точке $\theta = \theta_0$, т.е.

$$\min_{\theta \in \tilde{B}} \omega^{-1}(b) \bar{U}(b, a^{(1)}, \dots, a^{(d)}) = \frac{\bar{U}(b_0, a_0^{(1)}, \dots, a_0^{(d)})}{\omega(b_0)} \quad (7)$$

Для этого вместе с критерием (3) рассмотрим функцию

$$V(b, a^{(1)}, \dots, a^{(d)}, \lambda) = \bar{U}(b, a^{(1)}, \dots, a^{(d)}) - \lambda \omega(b), \quad (8)$$

$$V(\lambda) = \min_{\theta \in \tilde{B}} V(b, a^{(1)}, \dots, a^{(d)}, \lambda).$$

Тогда (8) равно

$$V(b, a^{(1)}, \dots, a^{(d)}, \lambda) = \bar{\sigma}_\xi^2 + \bar{\sigma}_\zeta^2 - (1+\gamma)\lambda + \begin{pmatrix} b_0 \\ a_0^{(1)} \\ \vdots \\ a_0^{(d)} \end{pmatrix}^T H \begin{pmatrix} b_0 \\ a_0^{(1)} \\ \vdots \\ a_0^{(d)} \end{pmatrix} -$$

$$- 2 \begin{pmatrix} b_0 \\ a_0^{(1)} \\ \vdots \\ a_0^{(d)} \end{pmatrix}^T H \begin{pmatrix} b \\ a^{(1)} \\ \vdots \\ a^{(d)} \end{pmatrix} + \begin{pmatrix} b \\ a^{(1)} \\ \vdots \\ a^{(d)} \end{pmatrix}^T \begin{pmatrix} H_{yy} - \lambda H_\alpha & \cdots & H_{zx^{(d-1)}} & H_{zx^{(d)}} \\ \vdots & H_{xx}^{(1)} & \cdots & H_{x^{(1)}x^{(d)}} \\ (H_{zx^{(d-1)}})^T & \vdots & \ddots & \vdots \\ (H_{zx^{(d)}})^T & (H_{x^{(1)}x^{(d)}})^T & \cdots & H_{xx}^{(d)} \end{pmatrix} \begin{pmatrix} b \\ a^{(1)} \\ \vdots \\ a^{(d)} \end{pmatrix},$$

где $\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \Phi_y^{(i)} (\Phi_y^{(i)})^T = H_{yy}$.

Дифференцируя $V(b, a^{(1)}, \dots, a^{(d)}, \lambda)$ по $b, a^{(1)}, \dots, a^{(d)}$ и приравнивая производные к нулю, находим

$$\begin{pmatrix} b(\lambda) \\ a^{(1)}(\lambda) \\ \vdots \\ a^{(d)}(\lambda) \end{pmatrix} = \begin{pmatrix} H_{yy} - \lambda H_\alpha & \cdots & H_{zx^{(d-1)}} & H_{zx^{(d)}} \\ \vdots & H_{xx}^{(1)} & \cdots & H_{x^{(1)}x^{(d)}} \\ (H_{zx^{(d-1)}})^T & \vdots & \ddots & \vdots \\ (H_{zx^{(d)}})^T & (H_{x^{(1)}x^{(d)}})^T & \cdots & H_{xx}^{(d)} \end{pmatrix}^{-1} H \begin{pmatrix} b_0 \\ a_0^{(1)} \\ \vdots \\ a_0^{(d)} \end{pmatrix}, \quad (9)$$

и тогда

$$V(\lambda) = \bar{\sigma}_\xi^2 + \bar{\sigma}_\zeta^2 - (1+\gamma) + \begin{pmatrix} b_0 \\ a_0^{(1)} \\ \vdots \\ a_0^{(d)} \end{pmatrix}^T H \begin{pmatrix} b_0 \\ a_0^{(1)} \\ \vdots \\ a_0^{(d)} \end{pmatrix} -$$

$$- \left(H \begin{pmatrix} b_0 \\ a_0^{(1)} \\ \vdots \\ a_0^{(d)} \end{pmatrix} \right)^T \begin{pmatrix} H_{yy} - \lambda H_\alpha & \cdots & H_{zx^{(d-1)}} & H_{zx^{(d)}} \\ \vdots & H_{xx}^{(1)} & \cdots & H_{x^{(1)}x^{(d)}} \\ (H_{zx^{(d-1)}})^T & \vdots & \ddots & \vdots \\ (H_{zx^{(d)}})^T & (H_{x^{(1)}x^{(d)}})^T & \cdots & H_{xx}^{(d)} \end{pmatrix}^{-1} H \begin{pmatrix} b_0 \\ a_0^{(1)} \\ \vdots \\ a_0^{(d)} \end{pmatrix}$$

функция $V(\lambda)$ на интервале $(-\infty, \lambda_{\min} + \bar{\sigma}_\xi^2)$ непрерывна, где λ_{\min} – наименьшее собственное число регулярного пучка квадратичных форм, то есть наименьший корень уравнения:

$$\det \left(H_{yy} - \begin{pmatrix} H_{zx^{(1)}} \\ \vdots \\ H_{zx^{(d)}} \end{pmatrix}^T \begin{pmatrix} H_{xx}^{(1)} & \cdots & H_{x^{(1)}x^{(d)}} \\ \vdots & \ddots & \vdots \\ (H_{x^{(1)}x^{(d)}})^T & \cdots & H_{xx}^{(d)} \end{pmatrix}^{-1} \begin{pmatrix} (H_{zx^{(1)}})^T \\ \vdots \\ (H_{zx^{(d)}})^T \end{pmatrix} - \lambda H_\alpha \right) = 0 \quad (10)$$

Легко показать, что уравнение $V(\lambda)=0$ имеет не более одного корня λ_1 на $(-\infty, \lambda_{\min} + \bar{\sigma}_\xi^2)$, так как на этом интервале функция $V(\lambda)$ непрерывна и

$$\dot{V}(\lambda) = -(\bar{\sigma}_\xi^2 + \bar{\sigma}_\zeta^2 + b^T(\lambda) H_\xi b(\lambda)) < -1, \lambda \in (-\infty, \lambda_{\min} + \bar{\sigma}_\xi^2)$$

Непосредственной подстановкой $\lambda_1 = \bar{\sigma}_\xi^2$ в уравнение $V(\lambda) = 0$ можно убедиться, что $\lambda_1 = \bar{\sigma}_\xi^2$ является корнем уравнения единственным на этом интервале. Тогда из (10) непосредственно следует справедливость (7).

Выражение (3) можно привести к виду

$$\min_{\theta \in \tilde{B}} \omega^{-1}(b) U_N(b, a^{(1)}, \dots, a^{(d)}) = \min_{\theta \in \tilde{B}} \frac{(Y - \Phi^T \theta)^T (Y - \Phi^T \theta)}{1 + \theta^T D_\alpha(N) \theta},$$

где $Y = (y_1, \dots, y_N)^T$, $D_\xi(N) = \begin{pmatrix} H_\alpha(N)/(1+\gamma) & 0_{r \times (r_1^{(1)} + \dots + r_1^{(d)})} \\ 0_{(r_1^{(1)} + \dots + r_1^{(d)}) \times r} & 0_{(r_1^{(1)} + \dots + r_1^{(d)}) \times (r_1^{(1)} + \dots + r_1^{(d)})} \end{pmatrix}$,

$$\Phi = (\Phi_y \mid \Phi_w^{(1)} \mid \dots \mid \Phi_w^{(d)}) = \begin{pmatrix} (\varphi_y^{(0)})^T & (\varphi_w^{(01)})^T & \dots & (\varphi_w^{(0d)})^T \\ \vdots & \vdots & \ddots & \vdots \\ (\varphi_y^{(N-1)})^T & (\varphi_w^{(N-11)})^T & \dots & (\varphi_w^{(N-11)})^T \end{pmatrix},$$

Введём новый вектор переменных $(1 \mid \theta)^T = \bar{\theta}$ и матрицу $\bar{\Phi} = (-Y \mid \Phi)$. Имеем для (3):

$$\min_{\theta \in \tilde{B}} \frac{\bar{\theta}^T \bar{\Phi}^T \bar{\Phi} \bar{\theta}}{\bar{\theta}^T \bar{D}_\alpha(N) \bar{\theta}}, \quad \bar{D}_\alpha(N) = \begin{pmatrix} 1 & 0 \\ 0 & D_\alpha(N) \end{pmatrix}, \quad \bar{\Phi}^T \bar{\Phi} > 0.$$

Аналогично (9) можно получить (для конечной выборки объёма N):

$$V_N(\lambda) = Y^T Y - \lambda - (\Phi^T Y)^T \tilde{H}^{-1} \Phi^T Y,$$

$$\tilde{H} = \begin{pmatrix} \Phi_y^T \Phi_y - \lambda H_\alpha(N)/(1+\gamma) & \Phi_y^T \Phi_x \\ \Phi_x^T \Phi_y & \Phi_x^T \Phi_x \end{pmatrix},$$

и $V_N(\lambda) = 0$ имеет свойства, аналогичные $V(\lambda) = 0$.

(11)

Однако, нахождение корня $V_N(\lambda) = 0$ можно записать в следующей форме: $\hat{\lambda}_1(N) = \lambda_{\min}(N) [\bar{\Phi}^T \bar{\Phi} \bar{D}_\alpha]$ - минимальное характеристическое число пучка квадратичных форм, определяемых $\bar{\Phi}^T \bar{\Phi}$ и \bar{D}_α . Однако $\bar{D}_\alpha \geq 0$, поэтому рассмотрим

$$\lambda_{\min}(N) [\bar{\Phi}^T \bar{\Phi} - \lambda \bar{D}_\alpha] = \frac{1}{\lambda_{\max}(N) [\bar{\Phi}^T \bar{\Phi}]^{-1} \bar{D}_\alpha].$$

Известно [11], что

$$\begin{aligned} \frac{1}{N} \frac{1}{\lambda_{\max}(N) [\bar{\Phi}^T \bar{\Phi}]^{-1} \bar{D}_\alpha] &\xrightarrow[N \rightarrow \infty]{\text{П.Н.}} \frac{1}{\lambda_{\max} \left[\left(\lim_{N \rightarrow \infty} \frac{1}{N} \bar{\Phi}^T \bar{\Phi} \right)^{-1} \bar{D}_\alpha \right]} = \\ &= \lambda_{\min} \left[\left(\lim_{N \rightarrow \infty} \frac{1}{N} \bar{\Phi}^T \bar{\Phi} \right) \bar{D}_\alpha^{-1} \right]. \end{aligned}$$

Так как нахождение $\lambda_{\min} \left[\left(\lim_{N \rightarrow \infty} \frac{1}{N} \bar{\Phi}^T \bar{\Phi} \right) \bar{D}_\alpha^{-1} \right]$ можно представить как определение корня уравнения $V(\lambda) = 0$, то

$$\frac{1}{N} \hat{\lambda}_1(N) \xrightarrow[N \rightarrow \infty]{\text{П.Н.}} \lambda_1.$$

Далее, неизвестные параметры можно определить из решения системы следующих линейных уравнений:

$$\tilde{H}\theta = \Phi^T Y \quad (12)$$

Тогда, очевидно, получаем

$$\frac{1}{N} \tilde{H}\theta - \frac{1}{N} \Phi^T Y \xrightarrow[N \rightarrow \infty]{\text{П.Н.}} \begin{pmatrix} H_{yy} - \lambda H_\alpha / (1 + \gamma) & H_{zx} \\ (H_{zx})^T & H_{xx} \end{pmatrix} \begin{pmatrix} b \\ a \end{pmatrix} - H \begin{pmatrix} b_0 \\ a_0 \end{pmatrix} = 0.$$

Из единственности решений (11), (12) и последнего выражения следует, что

$$\hat{\theta}(N) \xrightarrow[N \rightarrow \infty]{\text{П.Н.}} \theta_0.$$

4. Численный метод оценивания параметров

В общем случае вычисление оценок является задачей минимизации отношения двух квадратичных форм (3) и применение прямых методов (градиентного, метода Ньютона и т.д.) для определения глобального минимума сопряжено с известными трудностями, особенно при идентификации динамических систем большой размерности.

В работе был разработан численный метод, который позволяет:

1. ответить на вопрос существует ли единственная оценка $\hat{b}(N, \hat{\lambda}(i)), \hat{a}^{(1)}(N, \hat{\lambda}(i)), \dots, \hat{a}^{(d)}(N, \hat{\lambda}(i))$;
2. определить начальное приближение, гарантирующее сходимость итерационного процесса к единственной оценке $\hat{b}(N, \hat{\lambda}(i)), \hat{a}^{(1)}(N, \hat{\lambda}(i)), \dots, \hat{a}^{(d)}(N, \hat{\lambda}(i))$;
3. вычислить с любой наперед заданной точностью оценку $\hat{b}(N, \hat{\lambda}(i)), \hat{a}^{(1)}(N, \hat{\lambda}(i)), \dots, \hat{a}^{(d)}(N, \hat{\lambda}(i))$.

Доказательство утверждений аналогично доказательству приведенному в [12].

Пусть последовательность $\{\hat{\lambda}'(i)\}$ определяется следующим алгоритмом:

Шаг 0. $\hat{\lambda}'(0) = 0$;

Шаг 1. $\hat{\lambda}'(i) = \frac{(\lambda_{\min} + \hat{\lambda}'(i-1))}{2}$, где $\lambda_{\min}(N)$ - минимальное собственное число

регулярного пучка форм, то есть наименьший корень уравнения

$$\det \left(\Phi_y^T \Phi_y - \Phi_y^T \Phi_x (\Phi_x^T \Phi_x)^{-1} (\Phi_y^T \Phi_x)^T - \lambda H_\alpha \right) = 0, \quad (13)$$

где $\Phi = (\Phi_y \mid \Phi_x)$, $\Phi_x = (\Phi_x^{(1)} \mid \dots \mid \Phi_x^{(d)})$,

$$\Phi_y = \left((\varphi_y^{(0)})^T, \dots, (\varphi_y^{(N-1)})^T \right)^T, \quad \Phi_x^{(k)} = \left((\varphi_x^{(0k)})^T, \dots, (\varphi_x^{(N-1k)})^T \right)^T.$$

Шаг 2. Вычислить $\hat{b}(N, \hat{\lambda}(i)), \hat{a}^{(1)}(N, \hat{\lambda}(i)), \dots, \hat{a}^{(d)}(N, \hat{\lambda}(i))$ из системы линейных уравнений

$$\begin{pmatrix} \hat{b}(N, \hat{\lambda}(i)) \\ \hat{a}^{(1)}(N, \hat{\lambda}(i)) \\ \vdots \\ \hat{a}^{(d)}(N, \hat{\lambda}(i)) \end{pmatrix} = \begin{pmatrix} \Phi_y^T \Phi_y - \hat{\lambda}(i) H_\alpha(N) & \Phi_y^T \Phi_x^{(1)} & \dots & \Phi_y^T \Phi_x^{(d)} \\ (\Phi_x^{(1)})^T \Phi_y & (\Phi_x^{(1)})^T \Phi_x^{(1)} & \dots & (\Phi_x^{(1)})^T \Phi_x^{(d)} \\ \vdots & \vdots & \ddots & \vdots \\ (\Phi_x^{(d)})^T \Phi_y & (\Phi_x^{(d)})^T \Phi_x^{(1)} & \dots & (\Phi_x^{(d)})^T \Phi_x^{(d)} \end{pmatrix}^{-1} \begin{pmatrix} \Phi_y^T Y \\ (\Phi_x^{(1)})^T Y \\ \vdots \\ (\Phi_x^{(d)})^T Y \end{pmatrix}. \quad (14)$$

где $Y = (y_1, \dots, y_N)^T$.

Шаг 3. Вычислить

$$V_N(\lambda) = Y^T Y - \hat{\lambda}'(i) - \gamma \hat{\lambda}'(i) - \left(\Phi_y^T Y \hat{b}(N, \hat{\lambda}(i)) + \sum_{k=1}^d (\Phi_x^{(k)})^T Y \hat{a}^{(k)}(N, \hat{\lambda}(i)) \right),$$

Шаг 4. Проверить условие $V_N(\hat{\lambda}'(i)) \leq 0$.

Тогда если уравнение $V_N(\hat{\lambda}'(i)) = 0$ имеет корень $\hat{\lambda}'_l(N) \in [0, \lambda_{\min}(N)]$, то последовательность $\hat{\lambda}'(0), \hat{\lambda}'(1), \dots, \hat{\lambda}'(0)$ - конечна и $\lambda(0) \in [\hat{\lambda}'_l(N), \lambda_{\min}(N)]$, в противном случае последовательность бесконечна.

Этот алгоритм позволяет определить начальное приближение $\hat{\lambda}(0)$, необходимое для дальнейшего применения метода Ньютона или определить, что корень $\hat{\lambda}'_l(N)$ не существует.

Пусть существуют $\hat{\lambda}(0) \in [\hat{\lambda}'_l(N), \lambda_{\min}(N)]$, тогда $\lim_{i \rightarrow \infty} \hat{\lambda}(i) = \hat{\lambda}'_l(N)$, $\lim_{i \rightarrow \infty} \hat{b}(N, \hat{\lambda}(i)) = \hat{b}(N)$, $\lim_{i \rightarrow \infty} \hat{a}^{(k)}(N, \hat{\lambda}(i)) = \hat{a}^{(k)}(N)$, где $\hat{\lambda}(i)$, $\hat{b}(i, \hat{\lambda}(i))$ и $\hat{a}^{(k)}(i, \hat{\lambda}(i))$, определяются совместно со следующим алгоритмом:

Шаг 1. Вычислить $\hat{b}(N, \hat{\lambda}(i)), \hat{a}^{(1)}(N, \hat{\lambda}(i)), \dots, \hat{a}^{(d)}(N, \hat{\lambda}(i))$ из системы уравнений (14);

Шаг 2. Вычислить

$$\begin{aligned} \hat{\lambda}(i+1) = & \left(1 + \gamma + [\hat{b}(N, \hat{\lambda}(i))]^T H_\alpha(N) \hat{b}(N, \hat{\lambda}(i)) \right)^{-1} \times \left(Y^T Y + \hat{\lambda}(i) [\hat{b}(N, \hat{\lambda}(i))]^T H_\alpha(N) \hat{b}(N, \hat{\lambda}(i)) \right) - \\ & - \left(\Phi_y^T Y \hat{b}(N, \hat{\lambda}(i)) + \sum_{k=1}^d (\Phi_x^{(k)})^T Y \hat{a}^{(k)}(N, \hat{\lambda}(i)) \right). \end{aligned}$$

Шаг 3. Перейти к шагу 1. Процесс вычисления заканчивается, если выполняется условие

$$\frac{\|V_N(\hat{\lambda}(i+1)) - V_N(\hat{\lambda}(i))\|}{\|V_N(\hat{\lambda}(i+1))\|} \leq \delta,$$

где δ - априорно заданная точность оценок.

Это утверждение непосредственно вытекает из метода Ньютона:

$$\hat{\lambda}(i+1) = \hat{\lambda}(i) - \frac{V_N(\hat{\lambda}(i))}{V_N'(\hat{\lambda}(i))}.$$

5. Результаты моделирования

Предложенный в предыдущем пункте алгоритм был реализован в Matlab. Динамическая система описывается уравнениями:

$$z_i = 0.5\Delta^{0.9} z_{i-1} - 0.2\Delta^{1.7} z_{i-1} + \Delta^{0.7} x_i^{(1)} - 0.2\Delta^{1.4} x_i^{(1)} + 0.4\Delta^{0.6} x_i^{(2)} - 0.5\Delta^{1.2} x_i^{(2)} + 0.4\Delta^{0.6} x_i^{(3)} - 0.5\Delta^{1.1} x_i^{(3)} + \zeta_i.$$

$$y_i = z_i + \xi_i,$$

число входов модели $d=3$, порядок модели $r = r_1^{(1)} = r_1^{(2)} = r_1^{(3)} = 2$.

Входные сигналы описываются уравнениями:

$$x_i^{(j)} - 0.5 \cdot x_{i-3}^{(j)} = \zeta_i^{(j)} - 0.2 \cdot \zeta_{i-1}^{(j)} - 0.75 \cdot \zeta_{i-2}^{(j)} + \zeta_{i-4}^{(j)},$$

где $\zeta_i^{(j)}$ - белый шум.

Предложенный в статье нелинейный метод наименьших квадратов, сравнивался с методом наименьших квадратов (МНК). Алгоритмы сравнивались по следующим характеристикам:

$$\text{относительной погрешностью параметров: } \delta\theta = \|\hat{\theta} - \theta_0\| / \|(\theta_0)\| \cdot 100\%,$$

$$\text{относительной погрешностью моделирования: } \delta z = \|\hat{z} - z\| / \|z\| \cdot 100\%,$$

где $z = [z_1, \dots, z_N]^T$ – вектор выходной ненаблюдаемой переменной,

$\hat{z} = [\hat{z}_1, \dots, \hat{z}_N]^T$ – оценка вектора выходной ненаблюдаемой переменной, полученная с помощью модели.

Количество наблюдений $N = 2000$.

В таблице 1 приведены средние значения и среднеквадратические отклонения относительных погрешностей, рассчитанные по 50 процедурам оценивания, для различных отношений “шум-сигнал”.

Таблица 1.

σ_ξ / σ_z	σ_ζ / σ_x	Относительные погрешности	МНК (среднее±СКО), %	Критерий (3), (среднее±СКО), %
0.2	0.2	δz , %	4.11±0.84	2.77±0.65
		$\delta\theta$, %	9.00±2.614	5.78±2.52
0.5	0.5	δz , %	10.21±1.96	5.86±1.66
		$\delta\theta$, %	24.26±5.81	16.15±6.71
0.75	0.75	δz , %	15.80±2.29	8.42±2.60
		$\delta\theta$, %	34.88±7.65	22.72±9.40

Как видно из таблицы предложенный алгоритм дает наименьшие относительные погрешности, как для оценивания параметров, так и для моделирования.

6. Заключение

В работе предложен алгоритм для оценивания параметров линейных ARX систем с помехой наблюдения. В среде MATLAB создано программное обеспечение, результаты моделирования подтверждают эффективность работы алгоритма. Дальнейшее направление исследований может быть направлено на обобщение предложенного алгоритма на случай более сложных моделей шума.

Литература

1. Malti R., Aoun M., Sabatier J., Oustaloup A. Tutorial on system identification using fractional differentiation models.” In 14th IFAC Symposium on System Identification (SYSID’2006), Newcastle, Australia, pp. 606-611.
2. Кацюба О.А. Теория идентификации стохастических динамических систем в условиях неопределенности: монография. Самара: СамГУПС, 2008. 119с.
3. Иванов Д.В. Рекуррентное оценивание параметров динамических систем. Модели с ошибками в переменных. Saarbrucken: LAP LAMBERT Academic Publishing GmbH. 2011. 136с.
4. Иванов Д.В. Идентификация авторегрессии нецелого порядка с помехой в выходном сигнале Междисциплинарные исследования в области математического моделирования и информатики.

- Материалы научно-практической internet-конференции. 18-19 июня 2013 г./ отв. ред. Ю.С. Нагорнов. Ульяновск: SIMJET, 2013. С. 64-67.
5. Иванов Д.В. Идентификация линейных динамических систем нецелого порядка с помехой в выходном сигнале// Вестник Тамбовского университета. Серия: Естественные и технические науки. 2013. Т. 18. № 5-2. С. 2534-2536.
 6. Ivanov D.V. Identification discrete fractional order linear dynamic systems with output-error//Proceedings International Siberian Conference on Control and Communications (SIBCON'2013). Krasnoyarsk: Siberian Federal University. Russia, Krasnoyarsk, September 12-13, 2013. IEEE Catalog Number: CFP13794-CDR.
 7. Ivanov D.V. Identification discrete fractional order linear dynamic systems with errors-in-variables Proceedings of IEEE East-West Design & Test Symposium (EWDTs'2013), Rostov-on-Don, Russia, September 27 – 30, 2013, pp. 374-377.
 8. Иванов Д.В., Кацюба О.А. О Состоятельности оценок параметров ARX систем дробного порядка с помехой в выходном сигнале// Стохастическая оптимизация в информатике. 2013. Т. 9. № 2. С. 21-32.
 9. Кацюба О.А., Жданов А.И. Особенности применения МНК для оценивания линейных разностных операторов в задачах идентификации объектов управления //Автоматика и телемеханика.– 1979. - № 8. - С.86-90.
 10. Кацюба О.А., Жданов А.И. Идентификация методом наименьших квадратов параметров уравнений авторегрессии с аддитивными ошибками измерений // Автоматика и телемеханика. – 1982. - № 2. – С.29-38.
 11. Stoica P., Soderstrom T. Bias correction in least – squares identification // Int. J. Control. 1982. V. 35. № 3. Р. 449-457.
 12. Иванов Д.В. Численный алгоритм оценивания параметров линейных динамических систем дробного порядка с помехой в выходном сигнале// Эвристические алгоритмы и распределённые вычисления. 2014. Т. 1, № 1. С. 53-63.

СИНТЕЗ ДИНАМИЧЕСКОГО РЕГУЛЯТОРА СОСТОЯНИЯ С КОНТУРОМ НЕЙРОСЕТЕВОЙ АДАПТАЦИИ

А.С. Михайлов

ФГБОУ ВПО «Костромской государственный технологический университет»
г. Кострома, lelik-electronic@yandex.ru

Б.А. Староверов

ФГБОУ ВПО «Костромской государственный технологический университет»
г. Кострома, sba44@mail.ru

Аннотация

Предлагается гибридный метод нейроуправления нестационарным объектом с использованием динамического регулятора состояния и контура адаптации на основе модуля нейросетевой настройки параметров. Проводится структурный синтез системы автоматического управления с регулятором указанного типа. Рассматривается численный пример для объекта второго порядка. Оценивается соответствие переходного процесса в системе эталонному при изменении параметров объекта управления.

1. Введение

Современная теория автоматического управления характеризуется ориентацией на использование инновационных подходов при решении задач управления технологическими объектами и процессами. Одним из таких подходов является применение искусственных нейронных сетей (ИНС) в структуре регуляторов,

идентификаторов, адаптаторов и т.п. В настоящее время разработан ряд методов нейроуправления, предложены типовые схемы САУ на их основе. Классификация существующих методов нейроуправления приводится в [1].

Анализ публикаций по нейроуправлению и нейрорегуляторам за последние годы показывает, что наиболее часто встречающимся является метод нейроуправления, при котором управляющее воздействие для объекта управления (ОУ) формируется непосредственно на выходе ИНС. Примерами использования такого подхода к проектированию нейрорегуляторов являются работа [2], а также некоторые работы, рассмотренные в [3].

Несмотря на кажущуюся простоту и элегантность, данный метод имеет существенные недостатки. К ним прежде всего относятся:

- сложности, возникающие при анализе устойчивости нелинейных САУ на базе нейрорегуляторов;
- сложная структура схем нейроуправления;
- сложности при формировании обучающей выборки и длительность процедуры обучения (особенно при on-line обучении);
- отсутствие общих методик выбора архитектуры ИНС, используемой в качестве нейрорегулятора;
- низкая робастность системы с нейрорегулятором такого типа.

Указанные недостатки могут быть устранены при использовании гибридных систем, являющихся компромиссом между «чисто» нейросетевыми регуляторами и традиционными структурами САУ. Такие системы строятся как типовые регуляторы, параметры которых подстраиваются адаптирующими контурами в зависимости от изменения параметров объекта или режимов работы. При этом адаптирующие контуры проектируются на базе ИНС. Уже имеются примеры построения ПИД-регуляторов с нейросетевой настройкой [4], нейросетевых параметрических идентификаторов ОУ с известной структурой [5], а также применения таких идентификаторов при настройке полиномиальных регуляторов, представляющих собой сумму обратных связей по производным от выходной величины ОУ [6].

Однако, несмотря на достаточно большое число публикаций по рассматриваемой проблематике, в настоящее время практически не исследуются вопросы синтеза регуляторов состояния (РС) на основе ИНС. Вместе с тем данное направление исследований представляется особенно перспективным в случае сочетания типовых динамических РС с нейросетевыми адаптирующими контурами для управления нестационарными ОУ. По сравнению с регуляторами других типов РС имеют определенные преимущества поскольку обеспечивают возможность получения теоретически любого заданного переходного процесса, а также имеют простую структуру, поскольку представляют собой сумму жестких обратных связей по внутренним координатам объекта управления. Однако при использовании динамического РС с параметрически нестационарными ОУ для обеспечения качественного управления необходимо обеспечивать автоматическую настройку параметров регулятора. Эта задача в свою очередь может быть решена с помощью контура нейросетевой параметрической адаптации. В данном случае ко всем преимуществам РС добавляется еще одно – высокая робастность.

2. Постановка задачи

Задан нестационарный ОУ, имеющий передаточную функцию следующего вида:

$$W(p) = \frac{b_m \cdot p^m + b_{m-1} \cdot p^{m-1} + b_{m-2} \cdot p^{m-2} + \dots + b_1 \cdot p + b_0}{p^n + a_{n-1} \cdot p^{n-1} + a_{n-2} \cdot p^{n-2} + \dots + a_1 \cdot p + a_0}, \quad (1)$$

где m – порядок полинома числителя;

n – порядок полинома знаменателя;

b_m, \dots, b_0 – коэффициенты полинома числителя, принадлежащие множеству

$$B = \{b_i \mid b_{i \min} \leq b_i \leq b_{i \max}\}, \quad i = 0 \div m;$$

a_{n-1}, \dots, a_0 – коэффициенты полинома знаменателя, принадлежащие множеству

$$A = \{a_j \mid a_{j \min} \leq a_j \leq a_{j \max}\}, \quad j = 0 \div (n-1).$$

Пусть данный ОУ при определенных значениях коэффициентов полиномов числителя и знаменателя имеет переходную характеристику, представленную на рис. 1 (кривая 1). Очевидно, что данная переходная характеристика будет изменяться в некоторых пределах при изменении параметров ОУ или режимов его работы.

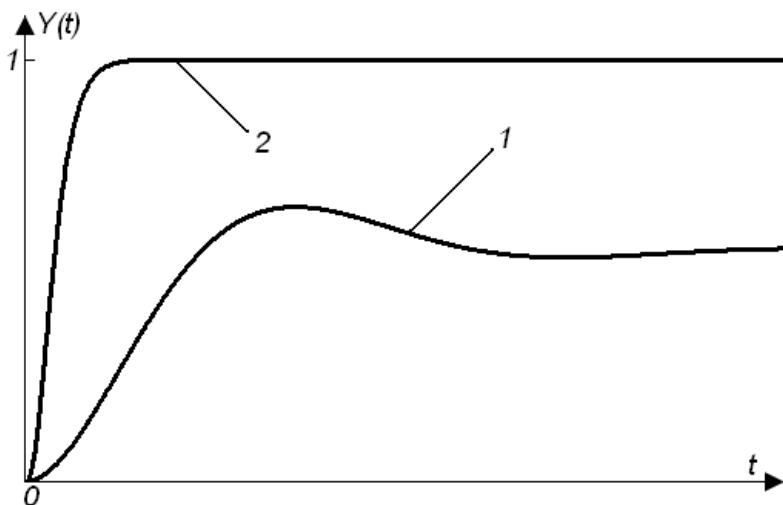


Рис. 1 – Переходная характеристика ОУ (1) и график желаемого переходного процесса (2).

Требуется провести структурный синтез регулятора состояния, обеспечивающего определенный вид переходного процесса в замкнутой системе во всем диапазоне изменения параметров ОУ (рис. 1, кривая 2). Для исключения негативного влияния нестационарности ОУ на переходный процесс в замкнутой САУ необходимо ввести в структуру регулятора контур нейросетевой параметрической адаптации.

3. Структурный синтез динамического регулятора состояния с контуром нейросетевой адаптации

Как известно из [7, 8], система управления с РС должна включать в себя:

- объект управления;
- модель ОУ в пространстве состояний для вычисления координат состояния;
- регулятор состояния, представляющий собой сумму жестких обратных связей по координатам состояния ОУ;
- наблюдатель состояния (НС);
- интегратор в основном контуре управления для обеспечения астатизма в установившихся режимах.

На рис. 2 приведена структурная схема САУ с астатическим динамическим РС для объекта n -го порядка. В качестве формы представления модели ОУ в пространстве состояний здесь и далее используется каноническая форма наблюдаемости (КФН).

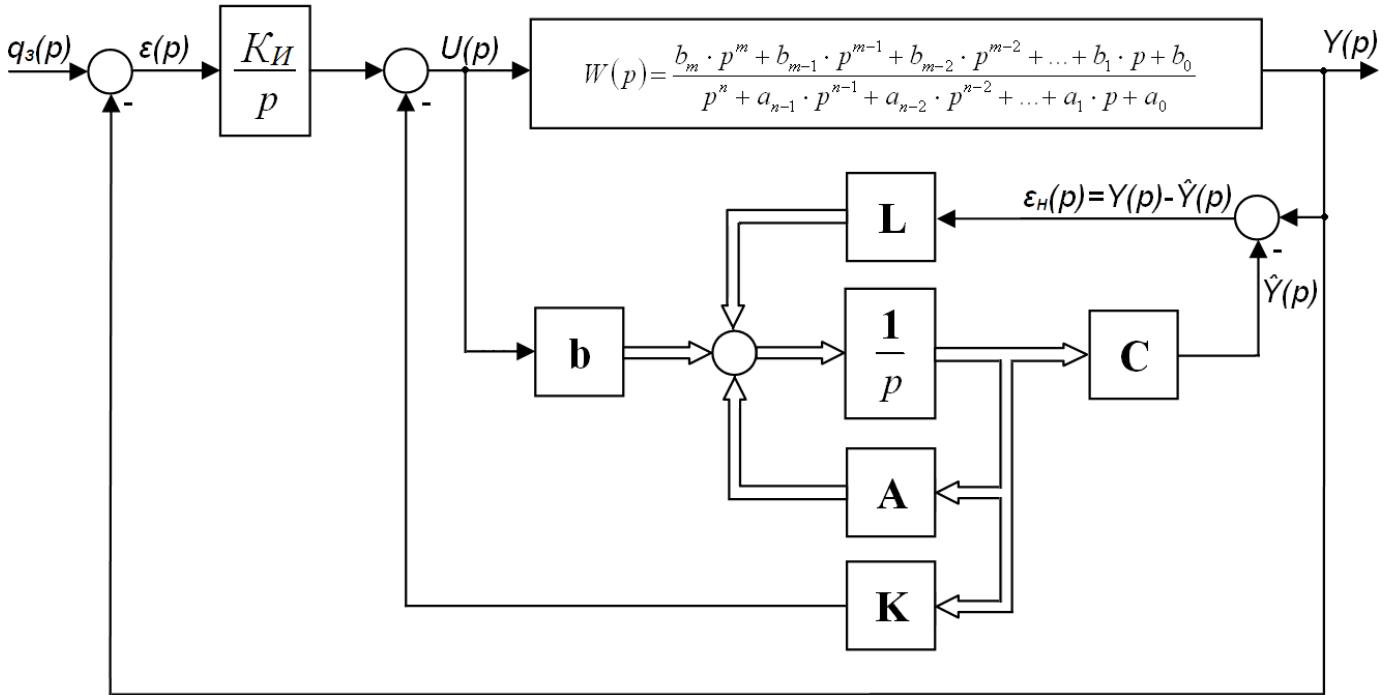


Рис. 2 – САУ с астатическим динамическим РС для объекта n -го порядка.

На рис. 2 приняты следующие обозначения сигналов: $q_3(p)$ – сигнал задатчика, $U(p)$ – управляющее воздействие с выхода регулятора, $Y(p)$ – сигнал с выхода САУ, $\hat{Y}(p)$ – сигнал с выхода модели ОУ, $\varepsilon_H(p)$ – ошибка наблюдения, $\varepsilon(p)$ – ошибка управления.

Кроме того, в соответствии с [9, 10] на схеме обозначены:

K_И – коэффициент добротности интегратора в основном контуре управления;

A – переходная (собственная) матрица ОУ;

b – входная матрица;

C – выходная матрица;

K – матрица-строка коэффициентов обратных связей по координатам состояния ОУ;

L – матрица коэффициентов подстройки НС.

При использовании контура нейросетевой адаптации в составе динамического РС для обеспечения качественного управления параметрически нестационарными ОУ автоматической настройке подлежат следующие параметры:

- элементы собственной матрицы ОУ **A** (коэффициенты полинома знаменателя передаточной функции ОУ);

- элементы входной матрицы **b** (коэффициенты полинома числителя передаточной функции ОУ);

- элементы матрицы-строки коэффициентов обратных связей по координатам состояния ОУ **K**;

- элементы матрицы коэффициентов подстройки НС **L**;

- коэффициент добротности интегратора в основном контуре управления **K_И**.

На рис. 3 приведена структурная схема САУ с астатическим динамическим РС и контуром адаптации на основе модуля нейросетевой настройки для объекта n-го порядка.

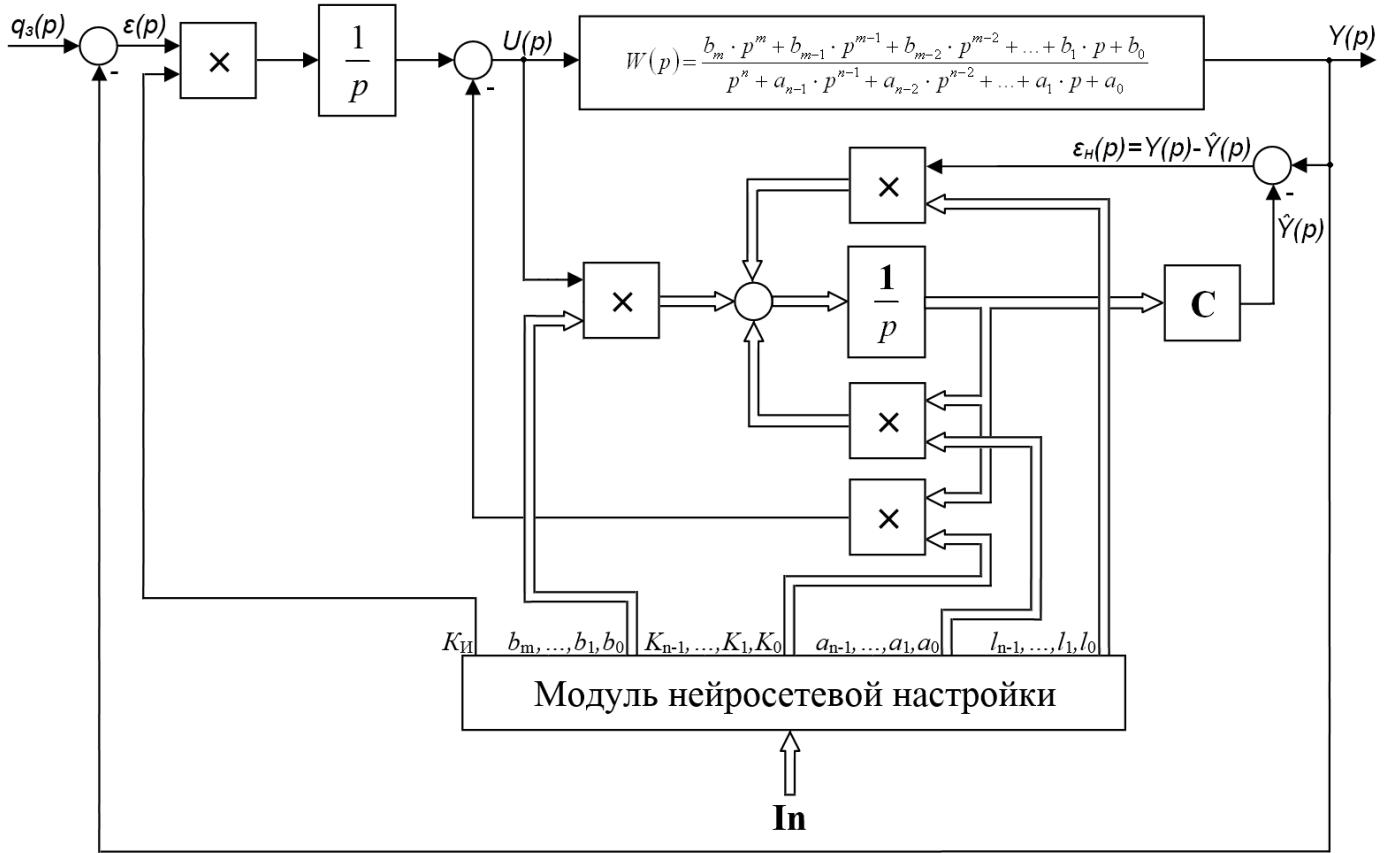


Рис. 3 – САУ с астатическим динамическим РС и контуром адаптации на основе модуля нейросетевой настройки для объекта n-го порядка

Модуль нейросетевой настройки, получая на входе вектор In , формирует на выходе вектор $\text{Out} = [K_H; b_m; \dots; b_1; b_0; K_{n-1}; \dots; K_1; K_0; a_{n-1}; \dots; a_1; a_0; l_{n-1}; \dots; l_1; l_0]$. Сигналы с выхода модуля нейросетевой настройки поступают на умножители, введенные в структуру САУ вместо безынерционных усилительных звеньев. Таким образом, осуществляется настройка параметров динамического РС по некоторым исходным данным, которые содержит вектор In .

Очевидно, что общее число параметров динамического РС, подлежащих адаптации, достаточно велико. Например, для ОУ второго порядка при отсутствии нулей передаточной функции ($m=0$) оно равно 8. В общем случае число подстраиваемых параметров динамического РС v определяется следующей зависимостью:

$$v = 3 \cdot n + m + 2. \quad (2)$$

Из выше изложенного становится очевидным, что для правильной работы контура адаптации динамического РС в основу принципа функционирования модуля нейросетевой настройки должна быть положена однозначная зависимость между данными, которые содержит входной вектор In , и параметрами динамического РС, которые выдает выходной вектор Out . Поэтому в качестве данных для вектора In можно использовать переходные либо частотные характеристики ОУ.

Также необходимо отметить, что принцип работы синтезированной САУ с астатическим динамическим РС и контуром адаптации на основе модуля нейросетевой настройки предполагает разделение во времени процесса настройки динамического РС и процесса управления, поэтому процесс настройки динамического РС не влияет на устойчивость замкнутой системы управления.

4. Параметрический синтез динамического регулятора состояния с контуром нейросетевой адаптации для нестационарного объекта второго порядка

Пусть задан ОУ второго порядка с передаточной функцией:

$$W(p) = \frac{K_p}{T^2 \cdot p^2 + 2 \cdot \rho \cdot T \cdot p + 1} = \frac{b_0}{p^2 + a_1 \cdot p + a_0}, \quad (3)$$

где K_p – коэффициент передачи ОУ;

T – постоянная времени ОУ;

ρ – показатель колебательности.

Пусть диапазоны изменения параметров ОУ находятся в следующих пределах: $K_p \in [0,5;1]$, $T \in [0,158;0,632]$, $\rho = const$ или, соответственно, $b_0 \in [1,25;40]$, $a_1 \in [1,5;6]$ и $a_0 \in [2,5;40]$.

Необходимо обеспечить апериодический переходный процесс с длительностью $t_{pp} = 0,5$ с во всем диапазоне изменения параметров ОУ с использованием астатического динамического РС и контура адаптации на основе модуля нейросетевой настройки. Настройку параметров РС необходимо проводить по амплитудно-частотным характеристикам (АЧХ) ОУ.

На рис. 4 приведена структурная схема САУ с астатическим динамическим РС и контуром адаптации на основе модуля нейросетевой настройки для объекта 2-го порядка.

Обучающая выборка для модуля нейросетевой настройки в данном случае формируется следующим образом:

- формируются векторы отсчетов АЧХ ОУ $\mathbf{In}_1, \mathbf{In}_2, \dots, \mathbf{In}_i$;
- каждому вектору отсчетов АЧХ ОУ ставится в соответствие вектор параметров $\mathbf{Out}_i = [K_{H_i}; b_{0i}; K_{l_i}; a_{l_i}; l_{l_i}; K_{0i}; a_{0i}; l_{0i}]$, определяемых аналитически, причем при расчете коэффициентов K_{H_i} , K_{l_i} и K_{0i} в качестве желаемого характеристического уравнения выбирается биномиальный полином третьего порядка, а при расчете коэффициентов l_{l_i} и l_{0i} – биномиальный полином второго порядка, кроме того принимается, что переходный процесс в контуре наблюдателя должен протекать в 3 раза быстрее, чем в основном контуре управления.

Однако на начальном этапе синтеза динамического РС с контуром адаптации в наличии имеются, как правило, лишь некоторые разреженные экспериментальные данные, полученные из переходных и/или частотных характеристик ОУ в различных режимах его работы. Очевидно, что этих данных недостаточно для того, чтобы обучить модуль нейросетевой настройки. Чтобы дополнить обучающую выборку, необходимо на основании экспериментальных данных получить динамическую модель ОУ в виде передаточной функции (3) и новые наборы исходных данных получать уже с этой модели.

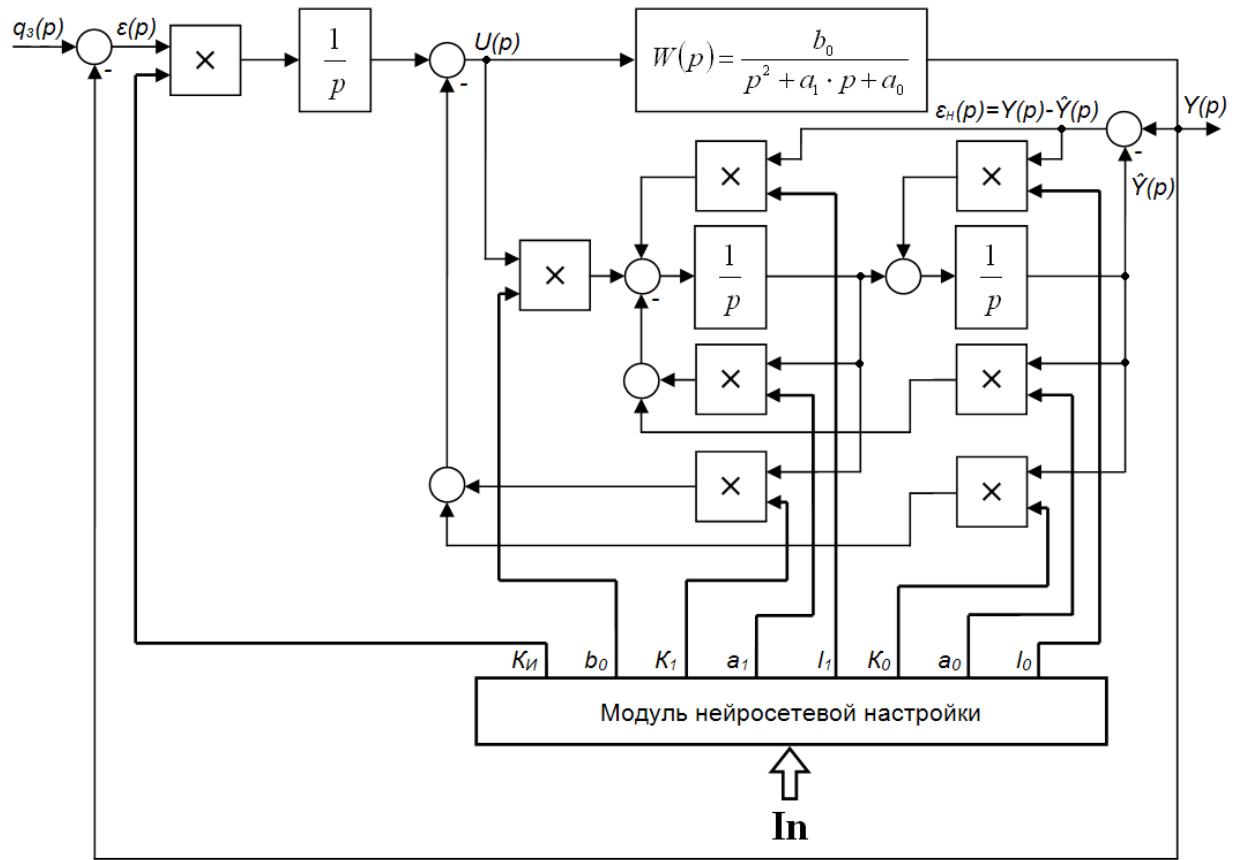


Рис. 4 – САУ с астатическим динамическим РС и и контуром адаптации на основе модуля нейросетевой настройки для объекта 2-го порядка.

На рис. 5 в пространстве варьируемых параметров ОУ показаны наборы исходных данных, входящих в обучающую выборку.

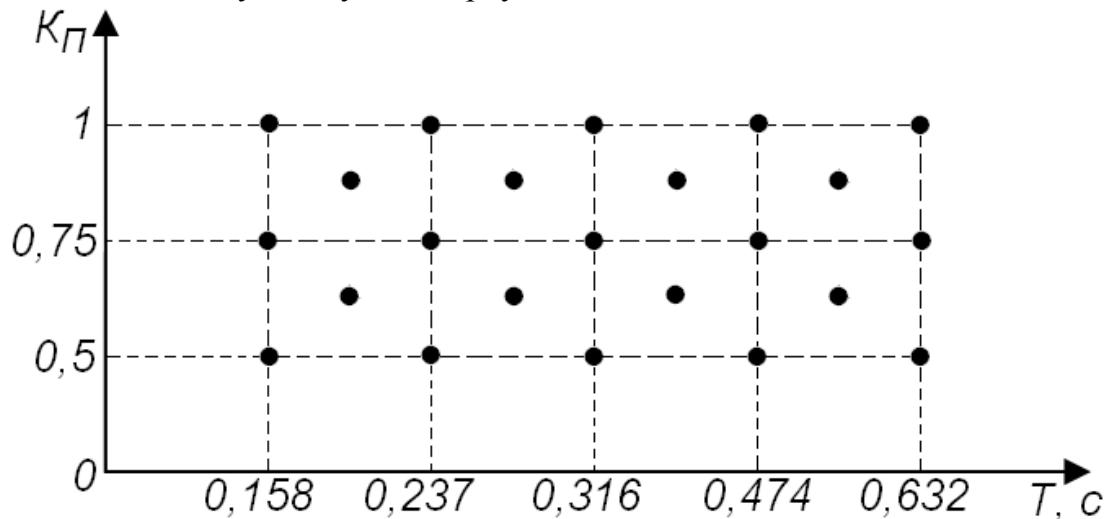


Рис. 5 – Наборы исходных данных, входящих в обучающую выборку, в пространстве варьируемых параметров ОУ.

Здесь по осям абсцисс и ординат отложены параметры ОУ, меняющиеся в различных режимах его работы. Точками отмечены комбинации параметров ОУ, при которых необходимо иметь АЧХ ОУ для получения исходных данных, входящих в

обучающую выборку.

Задача определения оптимального шага изменения параметров по осям абсцисс и ординат с точки зрения представительности обучающей выборки на сегодняшний день аналитически не решена, поэтому предлагается использовать метод половинного деления относительно базовых значений параметров ОУ $K_{\Pi\delta}$ и T_δ , выбранных как середины интервалов изменения параметров $K_\Pi \in [K_{\Pi\min}; K_{\Pi\max}]$ и $T \in [T_{\min}; T_{\max}]$.

На рис. 6 в качестве примера приведена АЧХ ОУ для базовых значений параметров ОУ $K_{\Pi\delta} = 0,75$; $T_\delta = 0,316$ с; $\rho_\delta = 0,475$.

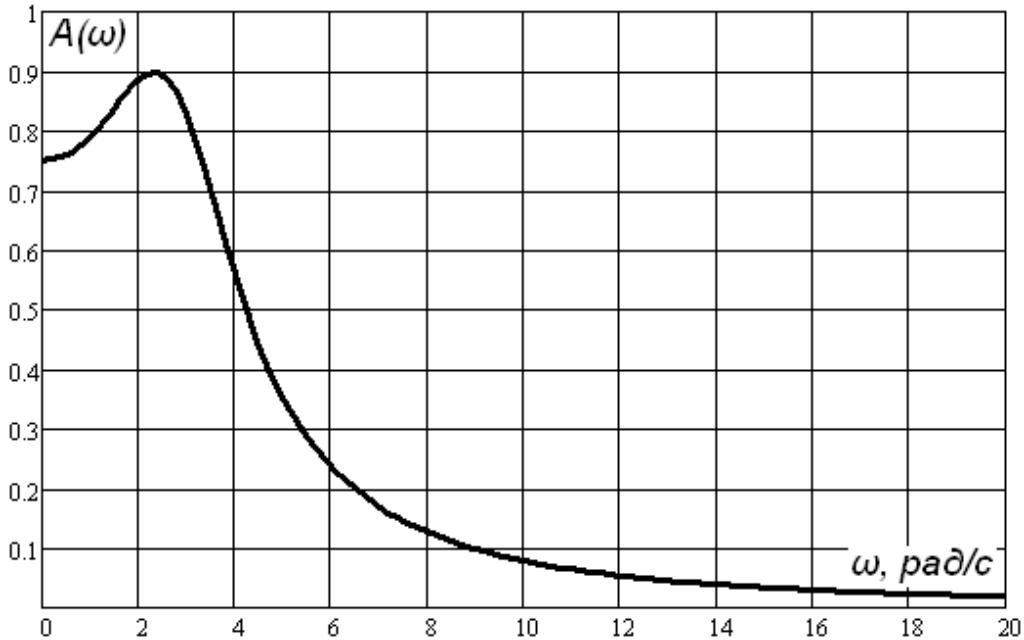


Рис. 6 – Амплитудно-частотная характеристика объекта управления при базовых значениях параметров $K_{\Pi\delta} = 0,75$; $T_\delta = 0,316$ с; $\rho_\delta = 0,475$.

В результате серии численных экспериментов определено, что наилучшие результаты работы в качестве модуля нейросетевой настройки дает радиальная базисная нейронная сеть с нулевой ошибкой. Нейронная сеть имеет такие веса и смещения, что ее выходы точно равны целям, т.е. имеет место нулевая погрешность на обучающем множестве, а количество нейронов радиального базисного слоя равно количеству примеров в обучающей выборке. Более подробно архитектура нейронных сетей данного класса приводится в [11]. Обучающая выборка сформирована из 23 АЧХ ОУ по 100 точек в каждой. АЧХ ОУ табулировались в интервале частот от 0 рад/с до 20 рад/с с шагом 0,2 рад/с.

На рис. 7 приведены графики переходных процессов в САУ нестационарным ОУ с астатическим динамическим РС и контуром адаптации на основе модуля нейросетевой настройки для одного из примеров тестовой выборки.

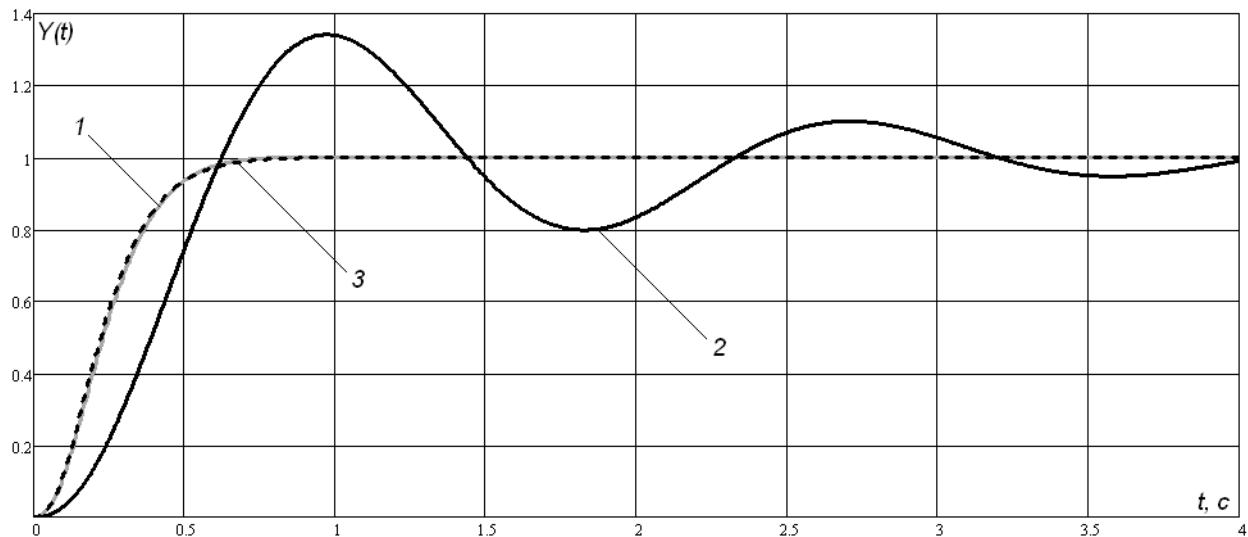


Рис. 7 – Графики переходных процессов в САУ нестационарным ОУ с динамическим РС и модулем нейросетевой настройки: 1 – для базовых значений параметров ОУ при аналитическом расчете параметров динамического РС; 2 – при изменении параметров ОУ до значений $K_{\Pi} = 0,55$; $T = 0,6$ с и неизмененных значениях параметров динамического РС; 3 – при включении контура параметрической адаптации в виде модуля нейросетевой настройки параметров.

Эталонный переходный процесс для базовых значений параметров ОУ $K_{\Pi_0} = 0,75$; $T_0 = 0,316$ с; $\rho_0 = 0,475$ при аналитическом расчете параметров динамического РС представлен кривой 1. При изменении параметров объекта до значений $K_{\Pi} = 0,55$; $T = 0,6$ с; $\rho = 0,475$ и неизмененных значениях параметров динамического РС качество переходного процесса значительно ухудшается (кривая 2). При включении в структуру САУ контура адаптации в виде модуля нейросетевой настройки параметров, точки обучения которого в пространстве варьируемых параметров ОУ не совпадают с точкой, соответствующей измененным параметрам, переходный процесс практически совпадает с эталонным (кривая 3). Очевидно, что для осуществления процесса адаптации необходимо предварительно экспериментальным путем получить АЧХ ОУ с изменившимися параметрами.

5. Заключение

Предложенный алгоритм синтеза динамического регулятора состояния с контуром адаптации позволяет настраивать динамический РС при изменении параметров объекта управления, предварительно обучив модуль нейросетевой настройки по экспериментальным данным. Рассмотренная структура системы автоматического управления с динамическим РС и модулем нейросетевой настройки имеет высокие адаптивные свойства и перспективна для управления нестационарными объектами. Дальнейшим развитием теории и практики гибридных нейросетевых систем управления с регуляторами состояния является определение максимального шага изменения параметров нестационарного объекта при формировании обучающей выборки для модуля нейросетевой настройки, а также синтез модуля нейросетевой идентификации, позволяющего автоматически определять порядок и параметры модели объекта управления в пространстве состояний.

Литература

1. Михайлов А.С. Актуальные проблемы применения нейрорегуляторов в системах автоматического управления // Материалы международной научно-технической конференции, приуроченной к 50-летию МРТИ–БГУИР. – Минск: БГУИР. – 2014. – Ч. 1. – С. 434–435.
2. Буянкин В. М., Пантиухин Д. В. Синтез интегрального пропорционального нейрорегулятора для управления электроприводом // Известия ТРГУ. – 2006. – № 3. – С. 115–121.
3. Михайлов А.С., Староверов Б.А. Проблемы и перспективы использования искусственных нейронных сетей для идентификации и диагностики технических объектов // Вестник ИГЭУ. – 2013. – №3. – С. 64–68.
4. Омату С., Халид М., Юсоф Р. Нейроуправление и его приложения: пер. с англ. – М.: ИПРЖР, 2000. – 272 с.
5. Анисимов А.А., Горячев М.Н. Идентификация электромеханических систем с использованием искусственной нейронной сети // Вестник ИГЭУ. – 2008. – №3. – С. 55–58.
6. Анисимов А.А., Тарапыкин С.В. Автоматическая настройка полиномиальных регуляторов электромеханических систем с использованием искусственной нейронной сети // Мехатроника, автоматизация, управление. – 2008. – №8. – С.13–18.
7. Изерман Р. Цифровые системы управления: Пер. с англ. – М.: Мир, 1984. – 541 с.
8. Кузовков Н.Т. Модальное управление и наблюдающие устройства. – М.: Машиностроение, 1976. – 184 с.
9. Староверов Б.А. Цифровые системы автоматического управления техническими объектами: Учебное пособие. – Кострома: Изд-во Костромского государственного технологического ун-та, 2005. – 102 с.
10. Староверов Б.А. Микропроцессорное управление электромеханическими системами: Учебное пособие. – Иваново: ИвГУ, 1986. – 80 с.
11. Медведев В.С., Потемкин В.Г. Нейронные сети. MATLAB 6 / Под общ. ред. к.т.н. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2002. – 496 с.

ИМИТАЦИОННАЯ МОДЕЛЬ СУШКИ ПОЛИУРЕТАНОВОГО ПОКРЫТИЯ.

Проуторов Е.В.

ФГБОУ ВПО «Череповецкий государственный университет»,
г. Череповец, proutorov@gmail.com
Максимова О.Г.

ФГБОУ ВПО «Череповецкий государственный университет»
г. Череповец, og62@mail.ru
Максимов А.В.

ФГБОУ ВПО «Череповецкий государственный университет»,
г. Череповец, a_v_maximov@mail.ru
Диордийчук Д.В.

цех покрытий металла №2 ОАО «СеверСталь».
г. Череповец, dxdordiychuk@severstal.com

Аннотация

Разработана модель сушки полиуретанового покрытия, на основе многоцепной модели полимерной системы с ориентационными взаимодействиями. Разработан алгоритм для имитационного моделирования методом Монте-Карло. Изучен процесс формирования структуры полиуретанового покрытия листового металлопроката.

Исследован процесс полимеризации в слое. Приведены методы управления прочностью покрытия.

1. Введение

В настоящее время возросла востребованность в листовом металлопрокате с различными лакокрасочными полимерными покрытиями, к качеству которых предъявляются очень серьезные требования с точки зрения сопротивляемости воздействию агрессивной внешней среды. Поэтому постоянно разрабатываются новые виды покрытий (в том числе полиуретановых), специализированных для различных применений.

Разнообразие механических свойств различных типов полиуретана обеспечивает большие возможности для разработок этого класса материалов с заданными свойствами поверхности. Свойства полиуретановых покрытий зависят от способов его нанесения и химической модификации [1]. При формировании поверхностных слоев в этих покрытиях большое значение имеют процессы целетоды системного направленного теплового воздействия с целью придания покрытию требуемого комплекса геометрических (шероховатость, волнистость, дефекты поверхности и др.) и физико-механических свойств (прочность, твердость, пластичность, износостойкость, цвет, блеск).

Для производства листового металла с полимерным покрытием практическое значение имеет детальное изучение явлений адгезии и ориентационного упорядочивания в поверхностном слое покрытия на уровне наноструктур. Учитывая полученные при исследовании закономерности, можно существенно улучшить качество выпускаемой продукции за счет получения более стойкого покрытия при одновременном удешевлении производства.

Качество покрытия определяется, прежде всего, величиной силы адгезии и коэффициента прилипания, а также его прочностью, которое зависит от степени ориентационной упорядоченности макромолекул в слое. Зависимость коэффициента прилипания от температурного режима сушки покрытия изучена в работе [2].

Широкое применение при решении практических задач в настоящее время стали находить математические методы, так как любые методы системного анализа опираются на математическое описание тех или иных фактов [3]. Управление сложными техническими системами требует проведения исследований из различных областей современной науки, то есть необходим комплексный подход. В данной работе рассмотрена модель производства полиуретанового покрытия, созданного на стыке нескольких наук - математики, физики, химии, металлургии, программирования и др.

2. Модель

Термопластичный полиуретан состоит из блок-сополимерных молекул с чередующимися мягкими и твердыми сегментами. Свойства эластомера ему придает сочетание мягких эластичных сегментов с высокой степенью эластичности и низкой температурой кристаллизации с одной стороны и твердыми кристаллизирующими сегментами с высокой точкой возгорания с другой стороны. Поэтому, благодаря смене участков мягких сегментов на группировку твердых сегментов такие свойства полиуретана, как твердость, стойкость, жесткость, эластичность, имеют широкий диапазон варьирования, в зависимости от длины участков цепи между уретановыми группировками, а также межмолекулярной структуры.



Рис.1 – Структурная схема полиуретана.

Для изучения свойств подобных веществ обычно используют модели одиночной полимерной цепи с учетом деформации, но без учета внутрецепных взаимодействий [4]. В работе [5] использована многоцепная модель с учетом внутрецепных взаимодействий, но без учета деформации и изменения длины мягкого сегмента. В данной работе применена модель, в которой учитываются и деформационные, и межцепные взаимодействия.

Моделирование процесса сушки полиуретанового покрытия в данной работе произведено на основе решеточной модели полимерной системы (рис.2) с помощью метода Монте-Карло.

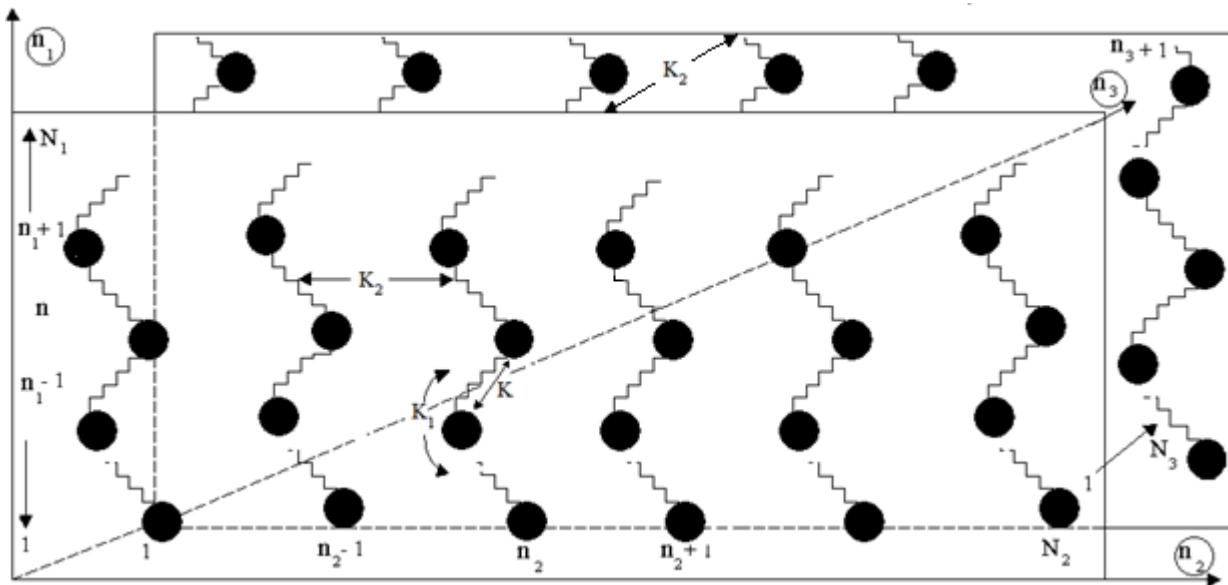


Рис. 2 – Многоцепная решеточная модель системы из гибких КЕ с локальными ориентационными и деформационными взаимодействиями

Для данной многоцепной модели предполагается, что эффективная потенциальная энергия внутри – и межцепных ориентационно – деформационных взаимодействий между соседними кинетическими единицами (КЕ) полиуретановых цепей представляется в форме:

$$\begin{aligned}
H \left\{ \vec{S}_{\vec{n}} \right\} = & \frac{1}{2} K \sum_{\vec{n}} (S_{\vec{n}} - l_0)^2 - K_1 \sum_{|\vec{n}_1 - \vec{m}_1| = 1} \sum_{|\vec{n}_2 - \vec{m}_2| = 1} \sum_{|\vec{n}_3 - \vec{m}_3| = 1} \vec{S}_{\vec{n}} \vec{S}_{\vec{m}} - \\
& - K_2 \left[\sum_{|\vec{n}_1 - \vec{m}_1| = 1} \sum_{|\vec{n}_2 - \vec{m}_2| = 1} \sum_{|\vec{n}_3 - \vec{m}_3| = 1} \vec{S}_{\vec{n}} \vec{S}_{\vec{m}} + \sum_{|\vec{n}_1 - \vec{m}_1| = 1} \sum_{|\vec{n}_2 - \vec{m}_2| = 1} \sum_{|\vec{n}_3 - \vec{m}_3| = 1} \vec{S}_{\vec{n}} \vec{S}_{\vec{m}} \right]
\end{aligned} \tag{1}$$

Первая сумма в выражении (1) описывает взаимодействия, обусловленные кинематической связью КЕ в цепи, которые характеризуются энергетической константой K [6], [7]. Величина l_0 является равновесным значением длины КЕ цепи. Значения параметров K и l_0 зависят от химической модификации полиуретана и определяются средней длиной участка цепи между уретановыми группировками. Жесткость цепей на изгиб учитывается (в рамках моделей Херста – Харриса и др. [8] – [10]) константой взаимодействия K_1 , которую можно определить из условий совпадения статистических характеристик изолированной цепи с квазиупругой потенциальной энергией при $K_2 = 0$ с параметрами более “реалистической” модели цепи, например цепи, характеризуемой жесткими КЕ (звеньями) длиной l и деформируемыми углами между ними [11]. Третье слагаемое в выражении (1) описывает энергию локальных межцепных взаимодействий дипольного типа. Значение константы K_2 описывает взаимодействия с другими цепями [6], [7], которую можно оценить из мультипольного разложения [12] энергии взаимодействия двух одинаковых диполей с моментом u находящихся в звеньях соседних цепей:

$$K_2 = \frac{1}{4\pi\varepsilon_0} \frac{u^2}{r^3}, \tag{2}$$

где r – среднее расстояние между соседними цепями, ε_0 – диэлектрическая постоянная. Подробно метод определения констант с помощью сопоставления теоретических расчетов с экспериментальными данными приведен в работе [13]. Отрицательные знаки у констант K_2 и K_1 [14], выбраны для того, чтобы параллельная ориентация ближайших КЕ (как в одной цепи, так и в соседних цепях) была бы предпочтительнее, чем антипараллельная. В разработанной модели не вводятся силы, препятствующие взаимным трансляционным смещениям соседних узлов разных цепей.

При моделировании конечной системы необходимо задать граничные условия. Для минимизации эффекта конечности системы вдоль осей n_2 и n_3 (рис.2) были выбраны периодические граничные условия. В направлении вдоль нормали к поверхности стального листа (n_1) граничные условия заданы особым способом. В ячейках с $n_1 = 1$ полимерные цепи взаимодействуют с металлом. Поэтому КЕ в этих ячейках могут быть свободными или закрепленными ($\theta_{1,n_2,n_3} = 0$) с вероятностью, равной коэффициенту прилипания, метод определения которого приведен в работе [2]. В ячейках с $n_1 = N_1$ введены свободные граничные условия.

Так как прочность ориентированного полимера в 10-15 раз выше неориентированного [15], в данной работе были изучены температурные зависимости параметра дальнего ориентационного порядка $\mu(T)$, в качестве которого была выбрана средняя проекция КЕ цепей полиуретана вдоль преимущественного их направления и рассчитывалась по формуле:

$$\mu = \sum_{\vec{n}} \vec{S}_{\vec{n}} / \sum_{\vec{n}} S_{\vec{n}} \quad (3)$$

Модель способна описывать полиуретановые покрытия в области упорядоченности. Также, может использоваться в других полимерных системах, молекулы которых не имеют разветвлений. Основная трудность заключается в нахождении констант взаимодействий, которые определяются из сопоставления с экспериментом.

3. Алгоритм

Алгоритм был реализован следующим образом:

1. Формирование начальной конфигурации произвольно ориентированных кинетических единиц (КЕ) в решетке. Расчеты проводились в стандартной сферической системе координат, где ориентация КЕ характеризуется двумя углами: $\theta_{\vec{n}}$ и $\varphi_{\vec{n}}$. Предполагалось, что величина сферического угла $\varphi_{\vec{n}}$ имеет равномерное распределение в промежутке $[0; 2\pi]$. Так как координатные θ -линии сферической системы координат имеют различную длину, в данной работе равномерное распределение выбрано не для величины угла $\theta_{\vec{n}}$, а для значения функции $\cos \theta_{\vec{n}}$. Длина вектора $S_{\vec{n}}$ имеет нормальное распределение с математическим ожиданием l_0 .
2. На каждом шаге Монте – Карло (n_{MK}) случайно выбиралась одна из ячеек решетки, с помощью формулы (1) определялись энергия E_0 находящейся в ней КЕ. Учитывались взаимодействия только с соседними КЕ.

3. Далее случайным образом изменялась конфигурация ячейки решетки и вычислялась новая энергия конфигурации E , которая может быть как меньше (E_1), так и больше (E_2) первоначальной E_0 . Данный этап можно проиллюстрировать с помощью рис. 3. Если система на первом шаге цепи Маркова находится в конфигурационном состоянии, обладающем энергией E_0 , то на втором шаге цепи она с определенной долей вероятности (W) может оказаться в конфигурационном состоянии, обладающем энергией E_1 или E_2 . Таким образом, если происходит понижение энергии системы, то ей приписывается новое конфигурационное состояние. В противном случае производится генерация случайного числа в интервале $(0; 1)$ и производится его сравнение с величиной больцмановского

фактора, вычисляемого по формуле $\exp\left(-\frac{\Delta E}{k_B T}\right)$. Если она оказывается больше

сгенерированного числа, то система переходит в состояние с более высокой энергией (рис.3). Если же этого не происходит, то ансамблю КЕ приписывается исходное конфигурационное состояние.

Так как в методе Монте-Карло сходимость достаточно медленная, проводились исследование количества шагов Монте-Карло (n_{MK}) необходимых для приведения системы в состояние равновесия. В данной работе применялось $n_{MK} = 10^4$, при числе частиц $N=10^3$.

Таким образом, при достаточно большом значении числа шагов Монте-Карло система приходит в состояние, соответствующее заданному значению температуры и энергетических и геометрических параметров модели в соответствии с химическим

строением и модификацией полиуретана. Полученная конфигурация позволяет определить средние практически всех термодинамических величин (параметр порядка, теплоемкость, восприимчивость и др.).

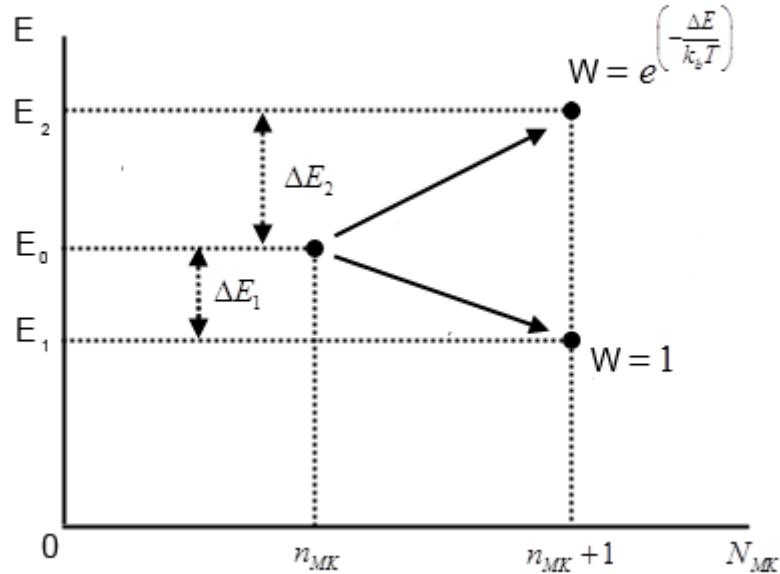


Рис. 3 – Изменения энергии системы в зависимости от номера шага в цепи Маркова

4. Результаты моделирования

Лакокрасочные материалы в момент их нанесения на листовой металлопрокат содержат растворители, которые испаряются в процессе их сушки. Скорость испарения на первой стадии можно определить из уравнения Кнудсена [16]:

$$\Omega = kP \sqrt{\frac{2\pi RT}{M}} \quad (4)$$

Где Ω – количество растворителя, испарившегося в единицу времени с единицы поверхности, k – коэффициент пропорциональности, p - давление насыщенного пара, R – газовая постоянная, T – температура, M – молярная масса.

При расчетах предполагалось, что давление насыщенного пара определяется по формуле:

$$P = P_0 e^{\frac{M\lambda}{R} \left(\frac{1}{T_0} - \frac{1}{T} \right)} \quad (5)$$

где λ – удельная теплота испарения, P_0 – давление насыщенного пара растворителя при температуре $T_0 = 300$ К.

Поскольку при движении внутри сушильной печи металлического листа с нанесенным сырьем покрытием толщиной h_1 , изменяется температура покрытия, то его

толщина уменьшается со скоростью W , вплоть до толщины сухого слоя h_2 . Испарение растворителя сопровождается изменением объема материала, в основном за счет толщины поверхностного слоя, что сказывается на структуре и свойствах образующегося покрытия. Полимерная молекула длиной L в среднем занимает объем $V = r^2 L$, где r – среднее расстояние между макромолекулами. Поэтому на единице поверхности стали в полимерном слое толщиной h , находится $n = \frac{h}{r^2 L}$ длинных молекул. Отсюда следует, что при уменьшении толщины слоя расстояние между молекулами уменьшается по закону $r \propto \sqrt{h}$. Поэтому на стадии сушки увеличивается константа межмолекулярного взаимодействия по закону $K_2 \propto h^{-3/2}$.

Увеличение температуры, с одной стороны ускоряет выпаривание растворителя, что способствует увеличению межцепных взаимодействий и, следовательно, способствует ориентационному упорядочению, с другой стороны увеличивает энтропию, что приводит к разупорядочению. Результаты моделирования показали, что переход рассматриваемой системы в упорядоченное состояние происходит скачком.

Для компьютерного моделирования изучена модификация полиуретана, изготовленного фирмой «Becker» и применяемого в ЦПМ-2 ОАО «Северсталь». При расчетах применялись следующие энергетические константы полимерных цепей: $K_1 = 10^{-20}$ Дж, $K = 3 \cdot 10^{-21}$ Дж. Принималось, что первоначально в сыром слое покрытия, константа $K_2 = 10^{-21}$ Дж. Полученные результаты согласуются с экспериментом в пределах погрешности.

На рис. 4а приведены зависимости температуры стального листа от его координаты x в сушильной печи. На рис. 4б. приведены рассчитанные зависимости параметра ориентационного порядка полиуретанового покрытия при соответствующих температурных режимах, изображенных на рис. 4а. Из этих рисунков видно, что при температурном режиме, соответствующей кривой 2 на рис. 4а возникает наибольшее упорядочение полимерных цепей (кривая 2 на рис. 4б).

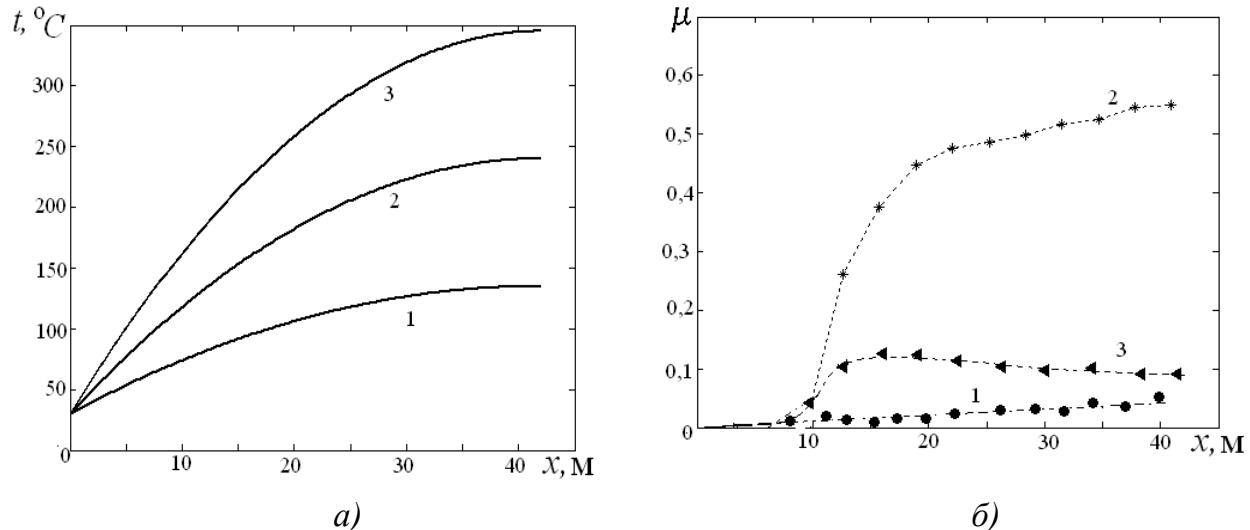


Рис. 4 – Зависимости дальнего ориентационного порядка полимерной системы от координаты сушильной печи (б) при различных режимах нагрева стального листа (а)

Таким образом, при выпаривании растворителя межмолекулярные связи становятся

сильнее, увеличивается плотность упаковки, система становится упорядоченной. При низкой температуре стального листа (кривая 1 на рис.4а) растворитель не успевает полностью испариться, что не дает сформироваться качественному покрытию (рис.4б). При высоких температурах (кривые 3 на рис.4а и 4б) за счет увеличения сегментальной подвижности происходит увеличение степени разупорядоченности молекул полимера.

5. Заключение

Как известно, предсказательные способности у имитационной модели выше, чем у аналитической и они достаточно просты в модифицировании, за счет легкого изменения входных данных. В рамках созданной модели авторами разработано, программное обеспечение, позволяющее предсказывать свойства покрытия при изменении режима сушильной печи. Результаты, полученные в данной работе с помощью разработанного авторами программного обеспечения, позволяют сформулировать вполне определенные рекомендации для технологического процесса нанесения полиуретанового покрытия на металлический лист, а также подобрать оптимальные технологические параметры для сушки полимерного покрытия.

Литература

1. Iskender Yilgor. Thermoplastic Polyurethanes – an Overview of the Synthetic Approaches and Chemical Structure Morphology-Property Relation. Functional polymer coatings for the fabrication of cell culture substrates/ Iskender Yilgor //International conference «Polymers advanced technologies». Berlin. 2013.
2. Варфоломеев И.А. Моделирование процесса адгезии полимерного покрытия при покраске металлической полосы по технологии Coil Coating. Производство проката, 2013г. №4. С. 26-30.
3. Антонов А.В., Моисеев Н.Н. Математические задачи системного анализа. М.: Книжный дом «ЛИБРОКОМ», 2012.-488 с.
4. Rouse, P. E. A Theory of Linear Viscoelastic Properties of Dilute Solutions of Ceiling polymers. J. Chem.Phys. – 1953. – Vol. 21. – № 7. – P. 1272–1280.
5. Максимов А. В. Теория равновесных и динамических свойств полимерных систем с ориентационным порядком: дис. ... д-ра. физ.-мат. наук – СПб.: ИВС РАН, 2010.
6. Kuznetsov D. V., Sung W. A green's-function perturbation theory for nonuniform semiflexible polymers: Phases and their transitions near attracting surfaces. J. Chem. Phys. – 1997. – Vol. 107. – P. 4729–4740.
7. Kuznetsov D. V., Sung W. Semiflexible polymers near attracting surfaces Macromolecules. – 1998. – Vol. 31. – P. 2679–2682.
8. Максимов А. В. и др. Сопоставление ориентационных корреляционных функций для двумерных полимерных систем с межцепными ориентационными взаимодействиями и полимерных систем с объемными взаимодействиями. Высокомолекулярные соединения. А. – 1999. – Т. 41. – № 7. – С. 1130–1137.
9. Harris R. A., Hearst I. E. On polymer dynamics. J. Chem. Phys. – 1966. – Vol. 44. – № 7. – P. 2595–2602.
10. Onishi T., Yamamoto M. Relaxation Spectrum of Chain Higt Polymeric Substances . J. Phys. Soc. Japan. – 1958. – Vol. 12. – № 12. – P. 1439–1443.
11. Максимов А. В. Теория упорядочения трехмерных полимерных систем с локальными ориентационно-деформационными взаимодействиями. Высокомолекулярные соединения. А. – 2007. – Т. 49. – № 5. – С. 891–904.
12. Бэкингем Э. Основы теории межмолекулярных сил. Применение к малым молекулам. [ред.] Б. Пюльман. Межмолекулярные взаимодействия: от двухатомных молекул до биополимеров. М.: Мир, 1981, стр. 9–98.
13. Maksimov A.V., Val'kova T.A., Maksimova O.G. Orientational Order and Birefringence in Nano-Layers of Polymer Films Ferroelectrics, 2013. Vol.: 444, Issue: 01, P. 162–167
14. Бекстер, Р. Точно решаемые задачи в статистической механике.– М.: Мир, 1985. 486 с.
15. Гуль В. Е., Кулезнев В. Н. Структура и механические свойства полимеров. – М.: Высшая школа, 1972. 367 с.

16. Максимова О.Г., Варфоломеев И.А., Ершов Е.В., Максимов А.В., Виноградова Л.Н. Применение компьютерного моделирования для управления процессом сушки лакокрасочного покрытия на поверхности металлического листа. Вестник Череповецкого государственного университета, 2012 г. Т.2. № 4 (43). С.14-16.

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССА ВОЗДЕЙСТВИЯ ЗАТЕКАЮЩИХ ВОЗДУШНЫХ ПОТОКОВ

Соболь В.Р.

Московский авиационный институт (национальный исследовательский университет)
г. Москва, vitsobol@mail.ru

Аннотация

В статье рассмотрена идентификация статистической модели возмущений, возникающих из-за затекания воздушных потоков и флуктуаций давления в нише в фюзеляже самолета. Методика основана на комбинации методов спектрального, регрессионного и корреляционного анализа. Предложен вариант математической модели процесса колебания давления. Представлены результаты идентификации модели временных рядов, содержащих значения давления в нескольких точках.

1. Введение

В настоящий момент широкое распространение получили исследования воздействия воздушных потоков, затекающих в выемки в фюзеляже летательного аппарата (ЛА), на динамику летательного аппарата, а также на системы и устройства, расположенные в этих выемках [1]. К этим задачам относится исследование затекания воздушных потоков в люки бомбардировщиков, в ниши стоек шасси и так далее. Вследствие затекания воздушного потока в нише возникают турбулентные потоки, приводящие к колебанию давления в фюзеляже. Колебания давления вызывают вибрации, которые могут помешать правильной работе установленных устройств, а в худшем случае — к их разрушению. Таким образом, определение влияния затекающего потока на процесс изменения давления является важной задачей при проектировании и эксплуатации ЛА.

Исследование влияния атмосферных возмущений может быть проведено по результатам летных испытаний, в ходе которых данные о давлении и других характеристиках непрерывно снимаются с датчиков, установленных на стенках ниши и на конструкции устройств, если они есть. Однако проведение одного такого эксперимента является чрезвычайно дорогостоящим мероприятием. При этом из-за неудачного выбора мест установки датчиков полученные данные могут оказаться малоинформативными, а увеличение объема измерительной аппаратуры может привести к искажению перераспределения воздушных масс в нише.

Задача моделирования процессов перераспределения воздушных масс в заданном объеме решается на основе современного математического и программно-алгоритмического обеспечения, разработанного для численного решения уравнений газодинамики [2]. Применение этих средств позволяет провести полноценное имитационное моделирование, в ходе которого определяются поле скоростей, распределения давления и плотности в достаточно большом объеме, содержащем нишу,

ее содержимое и часть внешней среды. Однако численное решение уравнений газодинамики требует огромных вычислительных ресурсов и длительного времени (например, моделирование процессов, протекающих в течение только одной секунды, занимает сутки машинного времени и десятки терабайт памяти для записи всех временных слоев). По этой причине имитационное моделирование, основанное на численном решении уравнений газовой динамики, не обладает должной гибкостью и оперативностью получения результатов. Для решения этой проблемы предлагается использовать методы статистического моделирования для описания процесса изменения давления в небольшом количестве опорных точек. При этом результаты численного решения уравнений газодинамики рассматриваются как статистическая информация, по которой требуется идентифицировать модель случайного процесса, описывающего эволюцию давления.

2. Газодинамическая модель

Подавляющее большинство исследований в данной области основывается на применении газодинамической модели. Она позволяет смоделировать изменение важнейших параметров воздушного потока в любой наперед заданной точке выемки. В качестве математической модели принята система двумерных осредненных по Фавру—Рейнольдсу нестационарных уравнений Навье—Стокса с соответствующими граничными условиями.

Если предположить, что имеет место осесимметричное, нестационарное сжимаемое течение вязкого, калорически совершенного газа, то указанные уравнения записываются в следующем виде:

$$\frac{\partial(rQ)}{\partial t} + \frac{\partial(rF_{in})}{\partial x} + \frac{\partial(rG_{in})}{\partial y} = \frac{\partial(rF_v)}{\partial x} + \frac{\partial(rG_v)}{\partial y} + rS, \quad (1)$$

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho rw \\ E \\ \rho k \\ \rho \varepsilon \end{bmatrix}, \quad F_{in}(Q) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho urw \\ (E + p)u \\ \rho uk \\ \rho u \varepsilon \end{bmatrix}, \quad G_{in}(Q) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vrw \\ (E + p)v \\ \rho vk \\ \rho v \varepsilon \end{bmatrix}, \quad (2)$$

$$F_v(Q) = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ r\tau_{x\varphi} \\ q_x + u\tau_{xx} + v\tau_{xy} + w\tau_{x\varphi} \\ (\mu + \frac{\mu_t}{\sigma_k}) \frac{\partial k}{\partial x} \\ (\mu + \frac{\mu_t}{\sigma_k}) \frac{\partial \varepsilon}{\partial x} \end{bmatrix}, \quad G_v(Q) = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ r\tau_{y\varphi} \\ q_y + u\tau_{yx} + v\tau_{yy} + w\tau_{y\varphi} \\ (\mu + \frac{\mu_t}{\sigma_k}) \frac{\partial k}{\partial y} \\ (\mu + \frac{\mu_t}{\sigma_k}) \frac{\partial \varepsilon}{\partial y} \end{bmatrix}, \quad (3)$$

где ρ , p , — плотность и давление в единице объема соответственно, u , v , w — компоненты вектора скорости \vec{V} в цилиндрической системе координат (x, y, φ) ,

$\vec{F} = (f_x, f_y, f_z)$ — плотность распределения объемных сил (объемная сила), k — кинетическая энергия турбулентности, ε — скорость диссипации кинетической энергии турбулентности, e — внутренняя энергия единицы объема, S — вектор источниковых членов, равный

$$S = \begin{bmatrix} 0 \\ \rho f_x \\ \rho f_y \\ \rho f_z \\ \rho \vec{F} \cdot \vec{V} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \alpha \frac{p + \rho w^2 - \tau_{\varphi\varphi}}{r} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ P_k - \rho \varepsilon + S_k \\ c_{\varepsilon 1} P_k \frac{\varepsilon}{k} - c_{\varepsilon 2} \rho \frac{\varepsilon^2}{k} + S_\varepsilon \end{bmatrix},$$

$$E = \rho e + 0.5 \rho (u^2 + v^2 + w^2) + \rho k \text{ - полная энергия в единице объема.}$$

В случае цилиндрической системы координат система (1)-(3) является квазитрехмерной.

При этом, w много меньше u и v , и при отсутствии закрутки потока $w=0$, $r=1$.

Параметр α обеспечивает возможность записи соотношений одновременно в разных системах координат (см. [3]). В случае плоского течения и декартовой системы координат $\alpha = 0$, и $\alpha = 1$ в случае осесимметричного течения и цилиндрической системы координат.

В приведенных выше уравнениях использованы компоненты тензора вязких напряжений $\tau_{xx}, \tau_{yy}, \tau_{xy}, \tau_{x\varphi}, \tau_{y\varphi}, \tau_{\varphi\varphi}$ и компоненты вектора теплового потока q_x, q_y

$$\tau_{xx} = 2(\mu + \mu_t) \frac{\partial u}{\partial x} - \frac{2}{3} (\mu + \mu_t) \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \alpha \frac{v}{r} \right), \quad \tau_{xy} = (\mu + \mu_t) \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right),$$

$$\tau_{yy} = 2(\mu + \mu_t) \frac{\partial v}{\partial y} - \frac{2}{3} (\mu + \mu_t) \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \alpha \frac{v}{r} \right), \quad \tau_{y\varphi} = (\mu + \mu_t) \left(\frac{\partial w}{\partial y} - \frac{w}{r} \right),$$

$$\tau_{\varphi\varphi} = 2(\mu + \mu_t) \frac{v}{r} - \frac{2}{3} (\mu + \mu_t) \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \alpha \frac{v}{r} \right), \quad \tau_{x\varphi} = (\mu + \mu_t) \frac{\partial w}{\partial x},$$

$$q_x = (k + k_T) \frac{\partial T}{\partial x}, \quad q_y = (k + k_T) \frac{\partial T}{\partial y}$$

Система (1)-(3) дополняется уравнением, записанным с учетом калорического уравнения состояния политропного газа

$$p = (\gamma - 1)e,$$

где γ отношение удельных теплоемкостей и термического уравнения состояния $p = \rho RT$. R — газовая постоянная, T — абсолютная температура.

Кроме того, μ и μ_t — коэффициенты динамической ламинарной и турбулентной вязкости, k и k_t — соответствующие коэффициенты теплопроводности, P_k — порождение кинетической энергии турбулентности

$$P_k = \mu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_m}{\partial x_m} \right) \frac{\partial u_i}{\partial x_j} - \frac{2}{3} \rho k \frac{\partial u_m}{\partial x_m}$$

S_k и S_ε — дополнительные источниковые члены, которые вводятся в некоторых вариантах так называемой k - ε модели турбулентности:

$$\mu_t = c_\mu \rho k^2 / \varepsilon$$

Константы, используемые в стандартной k - ε модели, являются безразмерными величинами. Их значения приведены в следующей таблице:

$c_{\varepsilon 1}$	$c_{\varepsilon 2}$	C_μ	σ_k	σ_ε
1.44	1.92	0.09	1.0	1.3

Таблица 1 – значения констант

В ходе исследования производился расчет изменения давления в 25 точках, расположенных внутри прямоугольной ниши длиной 0.16м и глубиной 0.035м. Число Маха было принято равным $M=0.85$, скорость набегающего потока $u=M*340=289$ м/с. Процесс моделируется в течение $T = 0.15$ с. Расположение точек представлено на рисунке 1.

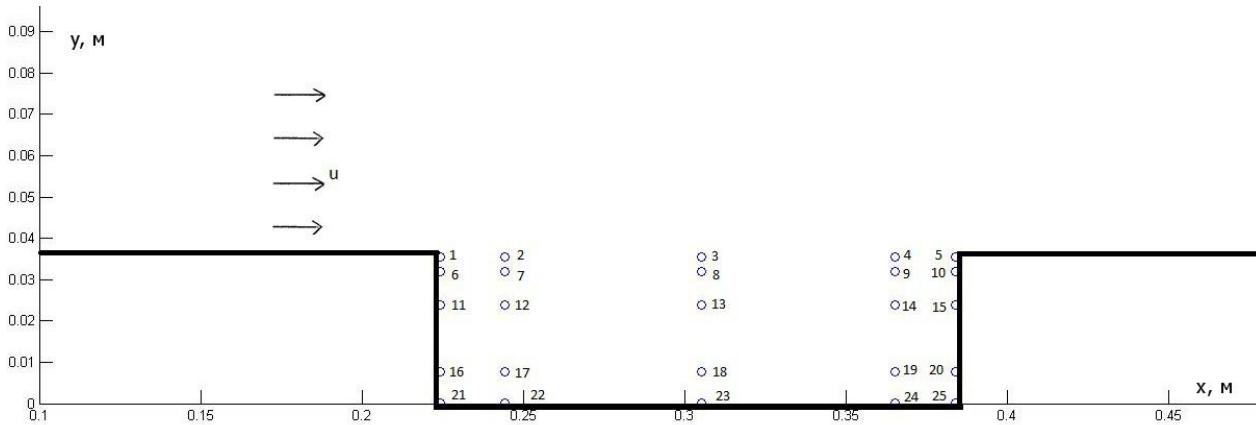


Рис. 1 – Расположение точек

Для решения системы уравнений (1)-(3) используется код собственной разработки. Описание используемых алгоритмов и результаты тестов представлены в работах [4]-[7].

3. Стохастическая модель

Временные ряды значения давления в точках выемки, получаемых в ходе решения системы уравнений Навье-Стокса, используется в качестве статистической информации. Визуальный анализ временных рядов позволяет выявить периодичность в колебаниях давлений в рассматриваемых точках (см. рис.2).

Анализ спектра ковариационной матрицы на основе метода MUSIC (Multiple signal classification) [8] позволил выделить преобладающие частоты, однако применение методов регрессионного анализа позволило убедительно отвергнуть гипотезу о наличии гармонических колебаний в имеющихся временных рядах (увеличение числа оцениваемых гармоник до 30 не изменило справедливость этого вывода).

Отметим, что скачки давления, имеющие максимальную амплитуду, происходят с относительно небольшой частотой. Очевидно, что именно эти колебания могут оказывать наиболее существенное воздействие на объекты, расположенные внутри ниши в фюзеляже ЛА. Для описания низкочастотных колебаний давления необходимо предложить иную математическую модель. Для описания периодических скачков

(провалов) давления с частотой φ относительно среднего давления p_0 предлагается использовать следующую модель:

$$p_\varphi(t) = p_0 + A_\varphi \left(e^{(\pm|\sin(\varphi t - \varphi_0)|^{q_\varphi})} - 1 \right). \quad (4)$$

Таким образом, для описания временного ряда предлагается использовать следующую модель:

$$p(t) = m_p + \sum_{i=1}^N A_{\varphi_i} \left(e^{(\pm|\sin(\varphi_i t - \varphi_{0i})|^{q_i})} - 1 \right) + \xi(t), \quad (5)$$

где $\xi(t)$ – случайный процесс с независимыми приращениями, m_p – математическое ожидание $p(t)$.

Поясним выбор данной зависимости. Экспонента обеспечивает достаточно быстрый рост функций $p_\varphi(t)$ в окрестности точек «пиков» $t_\varphi = \frac{(2k+1)\pi - \varphi_0}{\varphi}$. За счет выбора параметра A_φ можно варьировать амплитуду скачков, а за счет выбора параметра q_φ – их ширину.

4. Идентификация модели

В силу того, что колебания не являются стационарными, для их исследования исходный временной ряд можно разбить на несколько промежутков, на которых исходный процесс можно считать стационарным. Поскольку описываемая модель в первую очередь учитывает колебания максимальной амплитуды, для определения доминирующей частоты на каждом из промежутков предлагается использовать оценку ковариационной функции. В силу стационарности процесса ковариационная функция является функцией одного аргумента, равного разности между моментами времени.

$$t^* = \operatorname{argmax}$$

Определив точку максимума оценки ковариационной функции $\widehat{\text{cov}}(t)$, можно определить доминирующую частоту колебаний. Поскольку выборка имеет ограниченный объем, оценка ковариационной функции строится на интервале $[0; L]$, где $L < T$,

$$\widehat{\text{cov}}(t) = \sum_{i=0}^{N-L} (p(i) - m_p) * \frac{p(i+t) - m_p}{N-L}, \quad t = 0..L.$$

Таким образом оценки значений ковариационной функции получаются равноточными, т.е. вычисляются по равному числу слагаемых. Доминирующая частота f^* определяется однозначно по t^* (с учетом модуля в (4)) как

$$f^* = \frac{\pi}{t^*}.$$

Величина φ_0 определяется из условия

$$\varphi_0 = \operatorname{argmax}_\varphi \sum_{t_i < T} p^2(t_i), \quad t_i = \frac{\pi - 2\varphi}{2f^*}.$$

Величина q_φ определяется путем перебора на сетке целочисленных значений, границы которой определяются эмпирическим путем.

На каждом из промежутков предлагается использовать следующий алгоритм:

1. Положить $i = 1$, задать $\varepsilon > 0$, $p_0(t) = p(t)$.
2. Найти доминирующую частоту f_i по оценке ковариационной функции, построенной по остаткам временного ряда $p_{i-1}(t)$.

3. С выбранным шагом на отрезке $[-\pi; \pi]$ найти величину φ_0 .
4. Для каждого допустимого значения q с помощью метода наименьших квадратов оценить амплитуду колебаний на частоте f_i .
5. Вычислить остатки $p_i(t) = p_{i-1}(t) - p_{f_i}(t)$ и норму вектора остатков $\|p_i(t)\|$.
Положить $i = i + 1$.
6. Повторять шаги 2-5 до достижения сходимости. В качестве критерия остановки можно использовать малость относительного изменения сумм квадратов остатков

$$\frac{\|p_{i-1}(t)\| - \|p_i(t)\|}{\|p_i(t)\|} < \varepsilon$$

5. Результаты расчетов

На рисунке 2 представлены исходный временной ряд и его остатки в точке №1 на промежутке времени от 0 до 0.075 секунд за одну итерацию алгоритма.

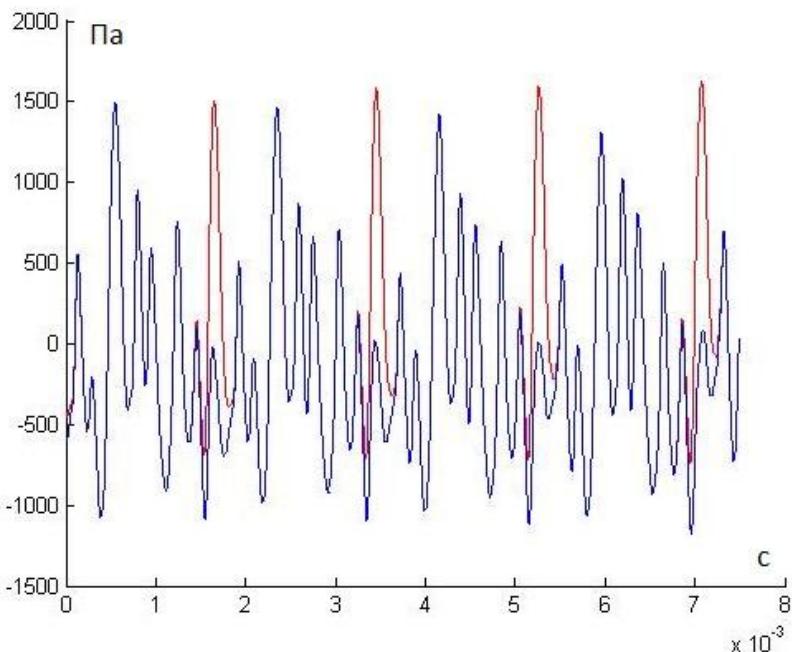


Рис. 2 – Флюктуации давления в точке 1 и остатки временного ряда (синий)

Доминирующая частота $f_1 = 1743,6\text{ Гц}$, амплитуда параметра q принято равным 50.

Значение

Окончательный результат работы алгоритма в 1й точке представлен на рисунке 3.

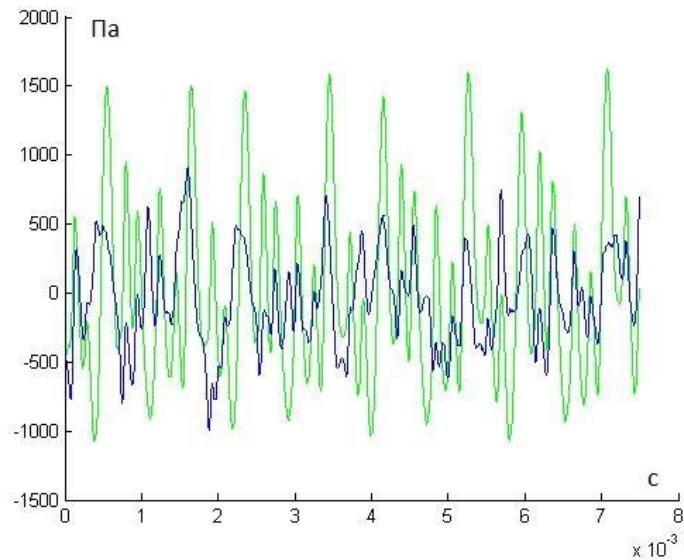


Рис. 3 – Исходный временной ряд (зеленый) и остатки

Как видно из рисунка, алгоритм удовлетворительно описывает периодические скачки и провалы значения давления в точке. Временной ряд остатков уже не имеет ярко выраженной периодической структуры.

Анализ частот, выделяемых в ходе работы алгоритма, показывает, что во всей области существует единый набор доминирующих частот. Это позволяет в дальнейшем использовать алгоритм при построении пространственной модели временных рядов. Приведем пример расчета параметров модели для частоты $f_1 = 1743,6\text{Гц}$ во всех точках области:

№	1	2	3	4	5	6	7	8	9
A_φ (Па)	924,7	1065,4	-1441	1838,2	-538,1	1092,8	1023,6	-1583,3	1957,6
φ_0 (рад)	-2,24	-2,23	0,244	-2,047	0,634	-2,233	-2,232	0,243	-2,044

Таблица 2 – результаты расчетов

10	11	12	13	14	15	16	17	18	19
-1749	983,4	854,2	1504,6	2084,8	2347,7	902,1	713,1	1189,1	1742,9
0,642	-	-2,237	-0,532	-2,041	-2,063	-2,23	-	-0,529	-2,047

Продолжение таблицы 2

20	21	22	23	24	25
1826,2	889,2	699,2	1137,3	1653,9	2016,9
-2,067	-2,23	-2,224	-0,529	-2,045	-2,068

Один из возможных вариантов модификации метода заключается в переходе к модели с кусочно-линейными амплитудами колебаний, вместо кусочно-постоянных, либо к более сложным полиномиальным моделям. Коэффициенты модели по-прежнему будут оцениваться с помощью метода наименьших квадратов.

6. Заключение

Представленные в работе результаты могут быть в дальнейшем использованы для

построения и исследования пространственной модели процесса изменения давления в нише в фюзеляже ЛА, на основе методов корреляционного анализа. Построение пространственной модели позволит, в свою очередь, исследовать такие параметры, как, например, момент силы, прикладываемый к стенкам выемки и к расположенным внутри устройствам. Переход при исследовании от газодинамической модели к стохастической позволит существенно сократить время, необходимое для проведения расчетов, а также снизит требования к объему оперативной памяти. Анализ доминирующих частот колебаний давления позволит сделать вывод о возможности возникновения резонансных явлений в рассматриваемой области.

Литература

1. K Atvars, K Knowles, S A Ritchie, N J Lawson «Experimental and computational investigation of an ‘open’ transonic cavity flow» //Proceedings of The Institution of Mechanical Engineers Part G-journal of Aerospace Engineering - PROC INST MECH ENG G-J A E 01/2009; 223(4):357-368.
2. Куликовский А.Г., Погорелов Н.В., Семенов А.Ю. Математические вопросы численного решения гиперболических систем уравнений. М.: ФИЗМАТЛИТ, 2001.
3. Пирумов У.Г. Обратная задача теории сопла М: Машиностроение. 1988
4. Глушко Г.С., Иванов И.Э., Крюков И.А. Метод расчета турбулентных сверхзвуковых течений. Математическое моделирование, 2009, т. 21, № 12, с.103 - 121.
5. Глушко Г.С., Иванов И.Э., Крюков И.А., Терехов И.В. Численное моделирование отрывных течений в соплах. Препринт № 815 Институт проблем механики РАН, М. 2006, 40 с.
6. Глушко Г.С., Иванов И.Э., Крюков И.А. Расчет сверхзвуковых турбулентных течений. Препринт № 793 Институт проблем механики РАН, М. 2006, 36 с.
7. Иванов И.Э., Крюков И.А. Квазимонотонный метод повышенного порядка точности для расчета внутренних и струйных течений невязкого газа. Математическое моделирование, том 8, N 6, 1996, 47-55.
8. Vaseghi S.V. Advanced digital signal processing and noise reduction. 2nd ed. Chichester, Wiley, 2000

ВОЗМОЖНОСТЬ ПРИМЕНЕНИЯ ОДНОТИПНЫХ МОДЕЛЕЙ ПРИ ИССЛЕДОВАНИИ СИСТЕМ, НАХОДЯЩИХСЯ НА РАЗНЫХ УРОВНЯХ ОРГАНИЗАЦИИ

Соловьева Я.А.

Череповецкий государственный университет

г. Череповец, soloviova.ya.a@gmail.com

Максимова О.Г.

Череповецкий государственный университет

г. Череповец, og62@mail.ru

Максимов А.В.

Череповецкий государственный университет

г. Череповец, a_v_maximov@mail.ru

Аннотация

В данной работе приведены примеры использования одной модели при исследовании систем, находящихся на различных уровнях организации. В качестве

примера рассмотрены базовые модели ферромагнетизма и тепловых структур. Показано, что существует ряд систем, которые при моделировании эти системы наследуют свойства у базовых моделей. Большинство результатов, полученных в работе, прошли экспериментальную проверку.

1. Введение

Одной из проблем, с которой сталкиваются почти всегда при проведении системного анализа, является проблема эксперимента в системе или над системой. Но довольно часто это связано с материальными затратами, а в ряде случаев запрещено моральными законами или законами безопасности.

Существуют классы различных систем, которые могут быть описаны одной моделью, только в различных терминах и переменных. Классическим примером этого является модель колебательных процессов, которая применяется в физике, химии, биологии, социальных науках и др. Допустим, что базовая система класса имеет ряд свойств, которые можно описать и математически, и с помощью опытных данных. Можно предположить, что другие системы этого класса наследует те же свойства, которые невозможно заранее определить с помощью эксперимента. В данной работе приведены два таких класса систем, которые могут наследовать свойства друг у друга.

2. Применение модели ферромагнетизма для исследования сегнетоэлектрических и полимерных систем

В настоящее время одним из ведущих разделов физики конденсированного состояния является физика сегнетоэлектриков, жидких кристаллов (ЖК) и других систем с различными типами ориентационного порядка.

В моделях ферромагнетиков [1, 2, 3] с каждым узлом $\vec{c} = (n, m, k)$ кристаллической решетки (рис.1) связан магнитный момент или спиновая переменная $\vec{S}_{\vec{c}}$. Полная энергия системы имеет вид:

$$E = -\frac{1}{2} \sum_{\vec{c}} \sum'_{\vec{c}'} K_{\vec{c}, \vec{c}'} \vec{S}_{\vec{c}} \vec{S}_{\vec{c}'} - \vec{h} \sum_{\vec{c}} \vec{S}_{\vec{c}}, \quad (1)$$

где вектор \vec{h} характеризует вектор напряженности внешнего магнитного поля, а параметры $K_{\vec{c}, \vec{c}'}$ определяют энергию взаимодействия спинов в различных направлениях решетки (n, m, k) . Если значения $K_{\vec{c}, \vec{c}'} > 0$, то данная решетка является моделью ферромагнетика, если $K_{\vec{c}, \vec{c}'} < 0$ – то моделью антиферромагнетика. При изучении этих явлений (ферромагнетизма и антиферромагнетизма) обычно используют приближение ближайших соседей, в котором пренебрегают взаимодействиями удаленных спинов. В этом приближении энергия взаимодействий (1) принимает вид:

$$E = -K_1 \sum_{n, m, k} \vec{S}_{n, m, k} \vec{S}_{n+1, m, k} - K_2 \sum_{n, m, k} \vec{S}_{n, m, k} \vec{S}_{n, m+1, k} - K_3 \sum_{n, m, k} \vec{S}_{n, m, k} \vec{S}_{n, m, k+1} - \vec{h} \sum_{n, m, k} \vec{S}_{n, m, k}.$$

С помощью точных и приближенных вычислений с использованием сферического приближения, методов компьютерного моделирования (метод Монте-Карло) и натурных экспериментов было показано, что при низких температурах взаимодействия между

спинами приводят к появлению магнитного порядка (намагченности), при высоких температурах система переходит в разупорядоченное, изотропное состояние. Кроме того, доказано, что в ферромагнетике существует фазовый переход типа «порядок - беспорядок».

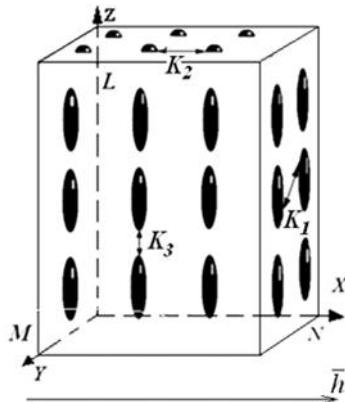


Рис. 1 – 3d-модель упорядоченной системы роторов (модель ферромагнетизма).

Было обнаружено, что особенности физико-химических характеристик и фазовых переходов в сегнетоэлектрических, жидкокристаллических и полимерных системах можно описать с помощью модели ферромагнетизма. Различие состоит в виде потенциала (1) и названий терминов (табл.1). Ранее такие изменения вызывали достаточно большие сложности для аналитических вычислений, но использование ЭВМ при исследовании сложных систем позволило их преодолеть.

Таблица 1.

Интерпретация величин в модели ферромагнетика

	Ферромагнетики	Сегнетоэлектрики	Жидкие кристаллы
Кинетическая единица (S)	Магнитный момент	Дипольный момент	Квадрупольный момент
Внешнее воздействие (h)	Магнитное поле	Электрическое поле	Электромагнитное поле
Реагирующий параметр (μ)	Вектор намагченности	Вектор поляризации	Фактор Цветкова

На рис. 2 приведены температурные зависимости диэлектрической проницаемости сегнетоэлектрической системы $SrTiO_3 - BiScO_3$, полученные экспериментально в работе [4].

В рамках модели ферромагнетизма с учетом таблицы 1 автором было проведено компьютерное моделирование методом Монте-Карло с использованием алгоритма метрополиса. Метод Монте-Карло стохастический, поэтому результаты моделирования уже имеют вероятностный характер. Для определения погрешности были вычислены среднее математическое ожидание и дисперсия. Расчеты показали относительную погрешность около 3%. Идентификация параметров для каждой из моделей проводилась с помощью сопоставления результатов по трем точкам.

Сравнение этих зависимостей показало, что они согласуются друг с другом в пределах погрешности моделирования. Поэтому можно предположить, что в подобных сегнетоэлектрических системах также существует фазовый переход второго рода типа «порядок - беспорядок».

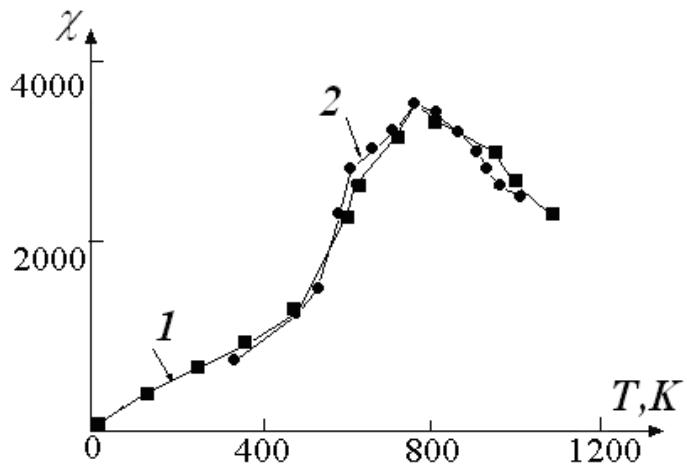


Рис. 2 – Зависимости диэлектрической проницаемости ($SrTiO_3$ - $BiScO_3$) χ от температуры.

1 – результаты численного моделирования, 2 – данные эксперимента [4].

Механизмы структурных превращений в одноосно-вытянутых пленках поливинилденфторида (ПВДФ) обычно исследуют с помощью молекулярной спектроскопии в полях высокой напряженности. На рис. 3 приведены экспериментальные зависимости коэффициенты поглощения [5] и вычисленные с помощью метода Монте – Карло в рамках модели ферромагнетизма [6]. Как видно, кривые имеют вид «баттерфляй» и результаты согласуются в пределах погрешности моделирования.

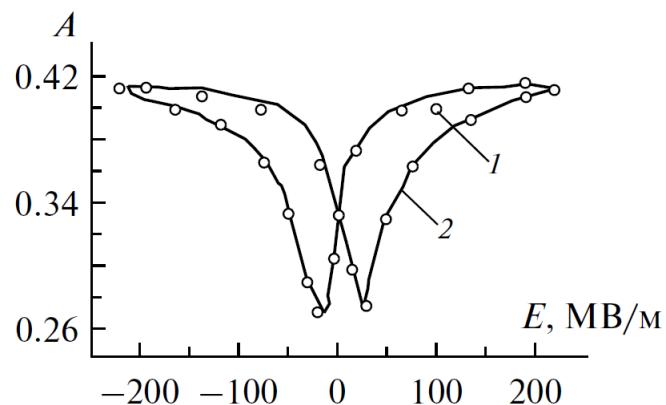


Рис. 3 – Полевая зависимость поглощения ИК-излучения в пленке ПВДФ: линия – результат моделирования, о – экспериментальные данные [5].

3. Модель тепловых структур

Модель тепловых структур, первоначально представлявшаяся как упрощенная модель для исследования ряда процессов в физике плазмы и теории управляемого термоядерного синтеза может быть использована и для феноменологического описания

распространения информации в научном сообществе или развития экономических структур (см. Таблицу 2).

Нелинейный источник $Q(T)$, отражает положительную обратную связь, и диссипативный процесс, нелинейность которого определяется коэффициентом $k(T)$ в уравнении в частных производных второго порядка для функции двух переменных $T(t, x)$:

$$\partial T(x, t) / \partial t = \partial / \partial x \{k[T(x, t)] \partial T(x, t) / \partial x\} + Q[T(x, t)], \quad (2)$$

где переменная t – время, x – некоторая обобщенная координата, смысл которой указан в Таблице 1. Функции $k(T)$ и $Q(T)$ в уравнении (2) имеют степенной вид:

$$k(T) = k_0 T^\sigma, \quad Q(T) = q_0 T^\beta. \quad (3)$$

В данной работе использован простейший вариант модели тепловых структур [6], в котором *источник* не зависит от времени. Показатели степени в уравнениях (3) $\beta > 0$ и $\sigma > 0$ в теории информации определяют, например, скорость производства новой информации от ее источника и эффективность ее распространения в информационной среде соответственно. Эти показатели и коэффициенты $k_0 > 0$ и $q_0 > 0$ предполагаются фиксированными в модели, их можно определить из сравнения с экспериментальными данными, из временного развития конкретного типа структур.

Таблица 2
Интерпретация величин в модели тепловых структур

Величина	Физика	Теория информации	Экономика
T	температура	плотность информации	плотность бизнес-структур
Q – источник	тепловой энергии	информации	инвестиций
k - коэффициент	теплопроводность и	информационной проводимости общества	экономической проводимости общества
x - координата	плазменного шнура	удаленность агентов информации	удаленность бизнес-партнеров

Рассмотрим поведение параметров данной модели на примере распространения научных знаний. При соотношении параметров $\beta > \sigma + 1$ в рассматриваемой модели решение растет в режиме с обострением, оставаясь локализованным, однако его полуширина сокращается. Это так называемый *LS*-режим с обострением [7] – режим локализации и одновременного роста интенсивности процесса во все более узкой области вблизи максимума этого процесса. Он соответствует тому, что научные исследования развиваются настолько быстро, что новое понимание оказывается сосредоточенным в рамках одной научной “элитной” супершколы. Типичный пример в теоретической физике – рождение квантовой механики, ключевые результаты в которой были получены несколькими учеными - гигантами (Гейзенбергом, Шредингером, Дираком).

Когда нелинейность в модели (3) очень велика ($\beta > \sigma + 3$), может существовать *HS*-режим без обострения [7] – режим, при котором происходит снижение интенсивности

процессов, размывания структур их от центра их образования. При этом процессы могут идти в виде волн даже с уменьшающейся амплитудой. В этом случае работы в данном научном направлении могут быть очень перспективны. От конкретных результатов проблема сначала переходит на уровень, когда идеи могут “вить в воздухе”, но никак не могут найти применения. Но эта ситуация неустойчива, и группа энтузиастов, даже небольшой “продвинутый” студенческий проект, может радикально изменить ситуацию, в этом случае возникает быстрый процесс в *LS*- режиме с обострением [7].

Для двух пространственно-распределенных систем [7, 8] уравнения (2) и (3) следует дополнить начальными данными, например, $T(x, 0) = T_1(x)$ при $a_1 < x < b_1$ и $T(x, 0) = T_2(x)$ при $a_2 < x < b_2$, которые означают, что существует значительная часть сообщества, которая не располагает информацией о данном научном направлении (в экономике – с неразвитой инфраструктурой), для которой $T_0(x) = 0$ при $x > b_k$ и $x < a_k$; $k=1, 2$.

В работе при постановке задачи предполагалось, что обобщенная координата x принимает значение в интервале: $-\infty < x < \infty$. В этом случае краевые (границные) условия обычно не указываются, как в классических задачах о колебаниях бесконечной струны, распространении тепла в бесконечном стержне и др. Однако для численного решения дифференциального уравнения (2) в общем случае следует использовать краевые (границные) условия на конечном нормированном промежутке.

На рис. 4 представлены профили функции $T(x, t)$, рассчитанные в различные моменты времени (в условных единицах) на основе использования численных методов решения. Происходящие в этом случае сценарии кардинально зависят от соотношения между скоростью производства новой информации и эффективностью ее распространения (или, в терминах модели (табл.2), от соотношения показателей β и σ соответственно).

Когда информационная (или экономическая) проводимость общества равна нулю ($k_0 = 0$) и источник информации (инвестиций) является постоянным ($\beta = 0$) даже при наличии двух начальных структур информация всегда равномерно распределяется по всему обществу, и далее только монотонно растет с течением времени. В этом режиме, который можно назвать “застоем” [8], сами структуры все же немного выделяются в информационном (или экономическом) обществе, но заметного количественного роста их не происходит (рис. 4, а).

В линейной модели ($\beta = 1$) при $k_0 = 0$ развитие структур происходит независимо и локализовано, как друг от друга, так и от всего общества (рис. 4, б). Сохранение формы и полуширины профиля свидетельствует о появлении организации и возникновении диссипативной структуры. Упорядоченность такого типа называют нестационарной, в отличие от традиционных стационарных (консервативных), не меняющихся со временем структур. Такое решение уравнения (2) описывает процесс коэволюции [7, 8]. Смысл такого решения прост: в определенных школах быстро развивается теория, математический аппарат или технология, которая успешно осваивается одной или несколькими “элитными” группами специалистов, работающих в этой области, и не выходит за рамки их круга. В нелинейном случае ($\beta > 1$) “элитные” структуры становятся квазистационарными, их профили практически “замирают”, и они не практически не влияют на развитие всего общества (рис. 4, в).

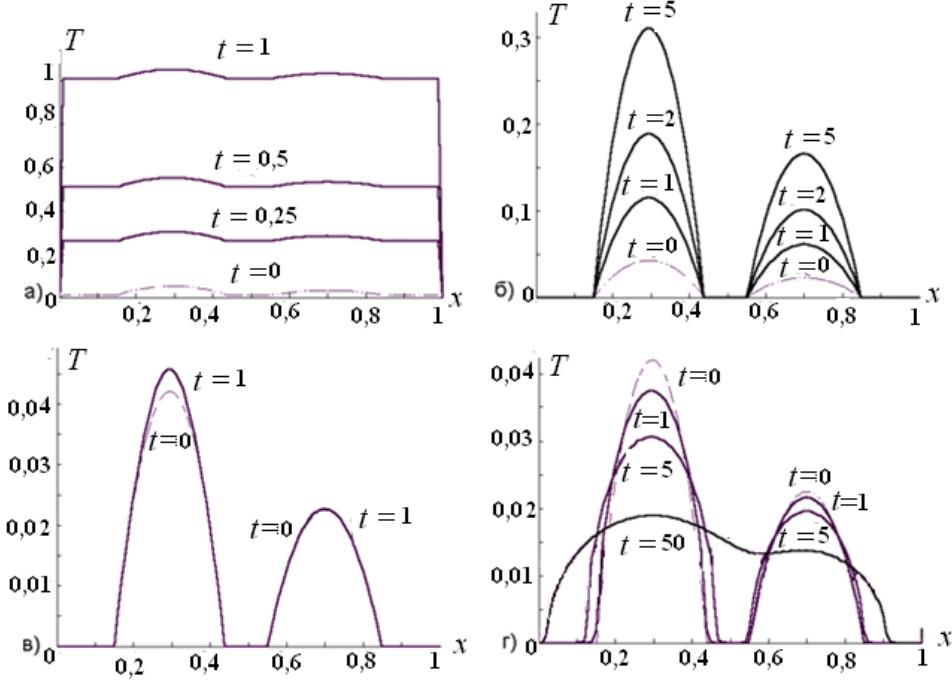


Рис. 4 – Сценарии эволюции: а) “застоя” ($k_0 = 0, \beta = 0, q_0 = 1, \sigma = 2$), б) “элиты” (нестационарный режим: $k_0 = 0, \beta = 1, q_0 = 1, \sigma = 2$), в) “элиты” (квазистационарный режим: $k_0 = 0, \beta = 4, q_0 = 1, \sigma = 2$), г) “НТР” ($k_0 = 1, \beta = 4, q_0 = 3, \sigma = 2$).

В случае, когда информационная (или экономическая) проводимость общества не равна нулю, и нелинейность в модели (2.1) очень велика ($\beta \geq \sigma + 2$), в науке и технике осуществляется сценарий научно-технической революции (НТР) [8]: изначально сформировавшиеся глобальные структуры (например, первые университеты) порождают в обществе более мелкие локальные научные школы (рис. 4, г). В компьютерном бизнесе в России 90-е годы прошлого века хороший пример – возникновение по городам страны салонов по распродаже компьютеров - сателлитов крупных фирм из Москвы и Санкт-Петербурга, теряющих со временем свою “элитарность” и сверхприбыль.

4. Заключение

Управление сложными системами обычно требует объединения усилий специалистов различных профилей. Анализ экспериментальных данных позволяет заключить, что различие между системами разной природы не столь резко, как это интуитивно представляется. Первом этапом исследования сложных систем является построение модели, которое должно опираться на опытный материал. При исследовании систем одного класса возможен синергетический подход, в котором единая модель или система уравнений при корректной трактовке входящих в него величин можно применить при описании различных явлений, что позволяет исследовать и управлять различными системами в условиях нехватки информации, ограниченных ресурсов и времени.

Литература

1. *Дайсон Ф., Монтролл Э., Кац М., Фишиер М.* Устойчивость и фазовые переходы. М.: Мир, 1973.
2. *Гинсбург В.Л.* Фазовые переходы в сегнетоэлектриках (несколько исторических замечаний) // *Успехи физических наук*. 2001. Т. 171. № 10. – С. 1091–1097.
3. *Mermin N., Wagner H.* Absence of ferromagnetism or antiferromagnetism in one- or two-dimensional isotropic Heisenberg models // *Phys.Rev.Lett.* 1966. V. 17. – P. 1133-1136.
4. *Иванов О.Н., Даньшина Е.П.* Релаксорные свойства системы SrTiO₃- BiScO₃ // Тезисы докладов XIX Всероссийской конференции по физике сегнетоэлектриков. М.: МИРЭА. 2011.
5. *Кочервинский В.В.* Особенности структурных превращений в сегнетоэлектрических полимерах на основе поливинилиденфторида в полях высокой напряженности // Высокомолекулярные соединения. Серия С, 2008. Т. 50. № 7. – С. 1407-1440.
6. *О. Г. Максимова, Т.О. Петрова, А. В. Максимов.* Моделирование процессов переполяризации в сегнето- и антисегнетоэлектрических полимерных системах // Известия РАН. Серия физическая. 2013. Т. 77. № 8, с. 1161–1163
7. *Капица С.П., Курдюмов С. П., Малинецкий Г.Г.* Синергетика и прогнозы будущего. М.: Наука, 1997.
8. *Maximov A.V.* Physics as a basis of the synergetic approach in natural sciences. //TheAbstracts of 12-th Conf. of Europ. Phys. Soc. “Trend in Physics”, Budapest, 2002, p. 222.

СЕКЦИЯ «ПРИКЛАДНЫЕ АСПЕКТЫ СИСТЕМНОГО АНАЛИЗА»

МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ПРОГРАММ УПРАВЛЕНИЯ РАСПРЕДЕЛЕННЫМИ МИКРОКОНТРОЛЛЕРНЫМИ СИСТЕМАМИ

Большаков Олег Сергеевич

РГАТУ имени П. А. Соловьева,
г. Рыбинск, E-mail: bolsh.os@gmail.com

Петров Александр Викторович

ООО “НПП САТЭК плюс”
г. Рыбинск, E-mail: gmdidro@gmail.com

Аннотация

В работе рассматриваются особенности разработки программ для специализированных микроконтроллерных систем как подвида встраиваемых систем. Рассматриваются недостатки существующих подходов к программированию специализированных микроконтроллерных систем. Предлагается четырехуровневая модель представления программ управления распределенными микроконтроллерными системами, на базе которой разрабатывается интегрированная среда разработки программ для микроконтроллерных систем, которая позволяет повысить уровень программирования, генерировать реализации высокогородневых пользовательских протоколов, описывать кроссплатформенные программы с генерацией программного кода для различных моделей микроконтроллеров на различные языки и под различные существующие компиляторы.

1. Введение

Одной из современных тенденций развития возможностей вычислительных устройств является создание встраиваемых систем и сетей (embedded systems & networks). Встраиваемые системы и сети принято определять как специализированные микропроцессорные системы, непосредственно взаимодействующие с объектом контроля или управления и объединенные с ним конструктивно [1].

Проектирование встраиваемых систем имеет следующие особенности [2, 3]:

- специализированность задач проектирования, уникальность систем;
- ограничения по габаритам и массе аппаратуры;
- ограничения по энергопотреблению (часто используется автономное питание);
- ограничения по условиям эксплуатации;
- дополнительные требования надежности системы, прогнозируемого времени реакции на определенные события.

В связи со спецификой применения встраиваемые системы очень разнообразны как с точки зрения выполняемой функциональности, так и с точки зрения используемых аппаратных и программных платформ, и, следовательно, методов, применяемых для проектирования таких систем [1]. Несмотря на большое количество готовых программных и аппаратных решений, разработчик каждый раз создает

специализированную систему, и поэтому в процессе проектирования ей приходится анализировать все уровни ее организации.

Среди аппаратных платформ, на которых основываются встраиваемые системы, выделяют системы, базирующиеся на микроконтроллерах - однокристальных микросхемах, включающих в себя микропроцессор, оперативное запоминающее устройство, постоянное запоминающее устройство, периферийные блоки управления [4]. Достоинством такой вычислительной платформы является баланс эффективности, универсальности аппаратуры и стоимости всего решения: разработчик выбирает готовое аппаратное решение с фиксированным набором периферийных блоков управления, потребляющих малое количество энергии.

2. Специализированные микроконтроллерные системы как особый вид встраиваемых систем

При проведении классификации микроконтроллеров можно выделить два их типа по степени специализированности под решаемые задачи:

- 1) общеселевые микроконтроллеры (и микропроцессоры в виде отдельных микросхем), используемые в многофункциональных мобильных системах (смартфонах и мобильных компьютерах);
- 2) специализированные микроконтроллеры для генерации сигналов определенных форм и реализации определенного набора интерфейсов.

К первому типу микроконтроллеров относятся 32-битные микроконтроллеры (иногда в этих же целях используются 32- и 64-битные микропроцессоры как отдельные микросхемы). Наиболее популярными представителями данного типа микросхем являются микроконтроллеры на базе архитектуры микропроцессора ARM. Ко второму типу устройств относятся энергоэффективные 8- и 16-битные микроконтроллеры (например, микроконтроллеры семейств AVR, PIC, STM, MCS 51, MSP430), а также некоторые 32-битные микроконтроллеры (например, ARM Cortex-M0, ARM Cortex-M0+). На устройства, в состав которых входят микроконтроллеры первого типа, для реализации многофункциональности устанавливают операционные системы общего назначения: Linux (в том числе Android), iOS, Window Phone и другие. Многие из устройств на базе данного типа микроконтроллеров представляют собой мобильные компьютеры и предназначены для выполнения широкого круга задач в разнообразных сферах деятельности человека. Операционные системы в таких устройствах управляют приложениями и различными видами ресурсов, что обеспечивает максимальную гибкость и расширяемость функциональности при наличии достаточно универсального набора интерфейсов (USB, Bluetooth, Wi-Fi) и датчиков (микрофон, сенсор экрана, гироскоп, акселерометр, компас, цифровая камера, GPS, датчик света, датчик уровня заряда аккумулятора). В данном виде систем часто используются различного рода виртуальные машины и в целом процесс проектирования программного обеспечения для них схож с процессом проектирования для общеселевых компьютерных систем - используются те же методологии и схожие технологии.

Совершенно иначе обстоит дело со вторым типом микроконтроллеров, которые специализированы под узкий круг решаемых задач. На базе таких микроконтроллеров обычно создают либо автономные устройства, либо такие микроконтроллеры используются как посредники, передавая информацию от датчиков до центрального устройства системы, которым может быть устройство с микроконтроллером первого типа. Так, например, специализированные микроконтроллеры входят в состав

мобильных систем для работы с периферией мобильного устройства - клавиатурой, сенсором, графическим дисплеем и т. д. Особенностью специализированных микроконтроллеров является то, что они обладают на несколько порядков меньшими объемами памяти и частотой процессора, чем микроконтроллеры первого типа. Фактически задачей такого микроконтроллера является генерация определенных видов сигналов для управления внешним оборудованием: либо сигналов специальной формы (широко-импульсная модуляция, тактирование, генерация аналогового сигнала), либо общение по одному из стандартизованных низкоуровневых протоколов (SPI, RS-232, TWI, USB, CAN и других). Генерация сигналов может осуществляться аппаратно с использованием периферийной системы, режимы и состояние которой задаются периферийными регистрами. Особенностью микроконтроллеров является наличие эффективных режимов энергосбережения: различные режимы сна и варианты пробуждения из них, или использование асинхронной событийной системы, позволяющей передавать между периферийными модулями события без участия центрального процессора, который при работе на высокой частоте потребляет много энергии.

В таблице 1 приведено сравнение различных архитектур микроконтроллеров. Сравнение производится по основным видам ресурсов: количество памяти, битность регистров, количество оперативной памяти и максимальная частота процессора. Как видно из таблицы, 32-битные микропроцессоры и микроконтроллеры ARM по количеству ресурсов на несколько порядков превосходят энергоэффективные 16- и 8-битные микроконтроллеры. Именно это и позволяет использовать на данных контроллерах мобильные операционные системы общего назначения.

Таблица 1

Сравнение особенностей архитектур микроконтроллеров

<i>Архитектуры</i>	ARM	PIC, MSP430	AVR, PIC, STM8
<i>Битность</i>	32, 64	16	8
<i>ОЗУ, порядок</i>	Внешняя DRAM,	< 100 Кб	< 10 Кб
<i>Частота</i>	500МГц..2 ГГц	20..70 МГц	< 20 МГц
<i>Использование ОС общего назначения</i>	возможно	невозможно	невозможно

3. Недостатки существующих технологий для разработки программ для специализированных микроконтроллеров

Выделим два важных аспекта технологий разработки программ:

- 1) предоставляемый уровень абстракции;
- 2) предоставляемый уровень динамичности.

Под предоставляемым уровнем абстракции будем понимать набор слоев абстракции, на которых пользователь может описывать программу. Низший слой абстракции называется низкоуровневым, высший слой абстракции называется высокоуровневым. Под предоставляемым уровнем динамичности будем понимать набор возможностей по изменению состояния программы и ее функциональности в процессе ее выполнения. Под состоянием программы понимается количество и структура выделенной памяти, а также хранимые в ней значения. Под функциональностью программы понимается набор исполняемых ею алгоритмов.

Как известно, возможность использования высших слоев абстракции является преимуществом любой технологии разработки программ, однако улучшение этого показателя зачастую коррелирует со снижением эффективности разрабатываемой программы, что связано с накладными расходами на обеспечение высокого уровня. То же касается и аспекта динамичности: высокие показатели данного аспекта используемой технологии позволяют делать гибкие, легко изменяемые программы как до выполнения, так и во время. В то же время при повышении уровня динамичности программы в общем случае так же уменьшается её эффективность, что связано либо с дополнительными проверками, необходимыми для корректности изменения параметров программы, либо с использованием дополнительных ресурсов (используется большее количество памяти). При проектировании программного обеспечения разработчику приходится делать выбор технологий согласно выделенным аспектам: таким образом у него возникает задача выбора технологии с приемлемой высокоуровневостью технологии, динамичностью и эффективностью получаемых программ.

На данный момент существует множество инструментов для программирования специализированных микроконтроллеров: Atmel Studio, CodeVisionAVR, WinAVR, AtmanAVR, IAR Embedded Workbench, MicroC/Pascal/Basic PRO for AVR/PIC, ICC AVR, AVR Eclipse Plugin, Algorithm Builder, BASCOM-AVR, LDMicro, MPLAB IDE, Proton Development Suite, Gimpel Software PC-Lint, Flowcode, MicroCode Studio, Pic Micro Pascal, Code Composer Studio, ST Visual Develop IDE и другие [5, 6, 7]. Несмотря на большое разнообразие инструментов для программирования микроконтроллеров, набор возможных технологий является скучным: большинство из существующих подходов к программированию специализированных микроконтроллеров предполагают разработку программ на языке С. Реже используется С++ с ограничениями и в редких случаях другие текстовые языки; существуют реализации графических языков (LD, блок-схемы).

Рассмотрим перечисленные технологии программирования микроконтроллеров с точки зрения выделенных ранее аспектов.

- 1) предоставляемый уровень абстракции

Существующие технологии, предполагающие разработку на языках С, Pascal, языке блок-схем и т. п., нельзя назвать высокоуровневыми, т. к. в рамках этих технологий программист традиционно пишет программу для микроконтроллера, настраивая биты периферийных регистров устройства и считывая с них значения. Существуют следующие подходы, которые частично решают проблему высокоуровневости для

данной технологии:

а) использование метода восходящего проектирования для вложения функций в обобщенные функции-агрегаты, имеющие большее количество параметров, чем исходные функции;

б) создание генераторов инициализационного кода по настройке периферии микроконтроллера.

Первый подход наиболее распространен при разработке библиотек кода и неизбежно приводит к избыточности, т. к. при увеличении количества параметров, в коде функций появляются избыточные проверки для принятия решений о вызове тех или иных низкоуровневых функций. Проблему избыточности пытаются решать средствами условной компиляции, в результате чего код таких библиотечных функций становится крайне сложным для анализа и расширения, что связано с неприспособленностью используемых языков предоставлять возможности для повышения степени абстракции с учетом статических свойств программ.

Второй подход заключается в том, что пользователь задает настройки периферии и с помощью генератора получает исходный код, который выполняет эту инициализацию. В данном случае повышение уровня абстракции происходит только при задании настроек периферии и не распространяется на всю программу. Тем не менее это позволяет немного повысить уровень разработки части программ: для создания генераторов используются либо наборы диалоговых окон (программная система CodeVisionAVR), либо используются графические редакторы, схематически показывающие связь пользовательских настроек с непосредственной работой аппаратной части микроконтроллера и специализированные калькуляторы для расчета задаваемых числовых параметров (инструмент QtouchComposer в составе среды Atmel Studio для микроконтроллеров AVR [8], STM32Cube для микроконтроллеров STM32 [9], инструмент Grace в составе Code Composer Studio для микроконтроллеров MSP430 [10]).

Как видно, оба подхода не решают проблему комплексно: первый подход сильно усложняет построение и использование программ, лишает их прозрачности и гибкости. Второй подход упрощает лишь часть работы пользователя, никак не изменяя уровня абстракции при написании основной части программы управления микроконтроллерной системой.

2) предоставляемый уровень динамичности

В рассматриваемых технологиях существует возможность динамического управления памятью (выделение блоков памяти во время исполнения программы). Тем не менее, данная возможность в специализированных микроконтроллерах используется редко. При изучении особенностей программ для специализированных микроконтроллеров становится очевидным важное их свойство, которое обуславливает как архитектуру их аппаратуры, так и ограничения используемых технологий - статичность. Статичность программ для микроконтроллеров в сравнении с программами для компьютеров обуславливается особенностями микроконтроллерных систем:

- специализированные микроконтроллерные системы управления являются функционально распределенными, т. е. еще до разработки программы для устройства известно, какие функции оно будет выполнять, и набор этих функций не меняется в процессе работы устройства;

- ограничения по габаритам, массе и энергопотреблению часто вынуждают использовать автономное питание, малое количество оперативной памяти, низкие частоты микропроцессора - такие процессы, как динамическое управление памятью,

обработка исключений становятся избыточными, и поэтому обычно память под переменные в программе выделяется статически, т. е. на этапе компиляции / линковки модулей;

- микроконтроллеры предназначены для использования в системах управления и не требуют серьезных вычислений - для ресурсоемких вычислений, требующих больших объемов памяти, существуют более мощные вычислители (цифровые сигнальные процессоры, ПЛИС).

- в случае реализации систем с динамическими связями (сенсорные сети и т.д.), реализация динамических связей осуществляется за счет заранее известного протокола (что тоже позволяет выделять память статически).

Все вышеприведенные причины статичности микроконтроллерных систем управления отражаются и на архитектуре, по которой построена аппаратная часть микроконтроллеров. Так энергоэффективные микроконтроллеры (AVR, PIC, STM8) построены на базе Гарвардской архитектуры (в случае PIC - расширенной Гарвардской архитектуры), что предполагает раздельное хранение данных и инструкций программ, причем у программы отсутствует возможность полиморфизма (изменение набора инструкций) в режиме ее исполнения (только на этапе "прошивки" микроконтроллера программой).

Именно из-за свойства статичности программ управления для микроконтроллерных систем в данной области не используются технологии из компьютерных систем или используются их усеченные аналоги, позволяющие реализовывать статические программы, что нивелирует в таких средствах разработки те достоинства, которые изначально были в них заложены. Примером использования технологий из области компьютерных систем может являться технология объектно-ориентированного программирования на базе языка C++: ситуация с использованием данной технологии такова, что существующие компиляторы под специализированные микроконтроллеры не позволяют использовать все возможности объектно-ориентированного языка, поскольку динамическое инстанцирование и удаление объектов, хранение таблиц методов, поиск метода при его вызове, использование механизма исключений являются крайне ресурсоемкими, и как следствие, избыточными процессами для специализированных микроконтроллеров. Именно по этой причине из реализаций C++ для, например, микроконтроллеров AVR, исключены операторы для динамического инстанцирования и удаления объектов. C++ в данном случае фактически используется только с той целью, чтобы привнести в язык модульность на уровне статических классов, которой нет в языке С. По этим же причинам в специализированных микроконтроллерах не используется технология .Net Micro Framework [11], содержащая интерпретатор, среду для выполнения объектно-ориентированной программы, механизмы обработки исключений, сборщик мусора.

К недостаткам технологий в области программирования микроконтроллерных систем стоит отнести не только отсутствие подходов, предоставляющих адекватную реализацию программ в аспектах высокогенерности и одновременно статичности программ, но также следующие проблемы существующего инструментария, отстающего от инструментария в области компьютерных систем:

- у разработчиков отсутствует возможность отлаживать программу на распределенной системе (или ее модели);

- используемые языки программирования не содержат конструкций для организации параллельного исполнения процессов; отсутствуют стеки протоколов и в то

же время отсутствуют какие бы то ни было технологии, упрощающие создание протоколов.

Отсутствие подходов, позволяющих реализовывать эффективные программы для специализированных микроконтроллерных систем с точки зрения высокоуровневости и одновременно статичности программ приводят к выводам о необходимости разработки нового подхода, которые бы смог повысить уровень и вместе с тем предоставить адекватные средства для реализации статичных программ.

4. Модель представления программ управления распределенными микроконтроллерными системами

Ниже представляется новый подход к разработке программ для специализированных микроконтроллеров, предполагающий описание программы в рамках новой многоуровневой модели, низший слой которой соответствуют уровню абстракции языка С при программировании микроконтроллеров. Предлагаемая модель имеет архитектуру, представленную на рисунке 1.

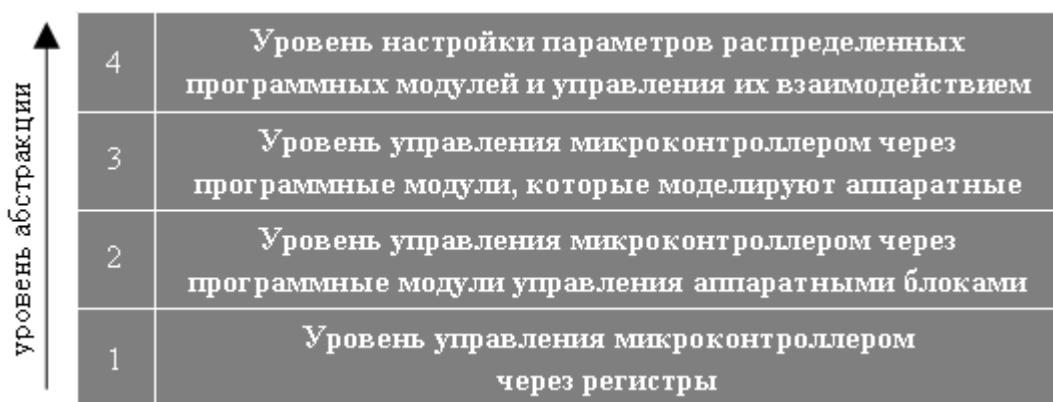


Рис. 1 – Архитектура предлагаемой модели представления программ управления

Предлагается выделить следующие уровни представления программы управления распределенной системой в порядке увеличения степени абстракции:

- 1) уровень управления микроконтроллером через регистры;
- 2) уровень управления микроконтроллером через программные модули управления аппаратными блоками микроконтроллера;
- 3) уровень управления микроконтроллером через программные модули, которые моделируют аппаратные модули микроконтроллера;
- 4) уровень настройки параметров распределенных программных модулей управления взаимодействием программных модулей и настройки их параметров.

В качестве средства, позволяющего описать программу в рамках данной модели предлагается новый высокоуровневый язык, являющийся надстройкой над языком С, и позволяющий описывать программы в рамках разрабатываемой интегрированной среды программирования.

Поясним значение каждого уровня на примере несложной системы: два микроконтроллера взаимодействуют друг с другом по беспроводному каналу связи (через радиомодули). К одному из них подключен переменный резистор, к другому - светодиод. Необходимо разработать программы для микроконтроллеров так, чтобы при изменении напряжения на потенциометре, менялась яркость светодиода,

подключенного к другому микроконтроллеру (рис. 2).

На данном уровне programma для устройства описывается в виде инструкций по настройке периферийных регистров - определенных ячеек памяти микроконтроллера. Настройки периферийных регистров интерпретируются электронной схемой устройства и тем самым реализуется аппаратное выполнение программы. Описание функций тех или иных регистров обычно производится в технической документации к программируемым устройствам. К типичным инструментам для программирования микроконтроллеров на этом уровне относятся языки программирования assembler, C. Для управления микроконтроллером программист записывает данные в память периферийных регистров и считывает данные из нее. Часть из этих регистров позволяет напрямую задавать состояния и считывать их с pinов микроконтроллера, остальные необходимы для изменения внутренних режимов работы устройства. Пример программного кода на языке C в логике состояния регистров, позволяющего задать настройки периферийного интерфейсного блока UART и передать два байта (3,4) приведен на рисунке 3.



Рис. 2 – Пример микроконтроллерной схемы

1) Уровень управления микроконтроллером через регистры

```
/* настройка USART */
UBRRH=00;
UBRRL=25;//установка делителя частоты на скорость 19200 бит/с
UCSRB=(1<<RXEN) | (1<<TXEN); //включение приемника и передатчика
UCSRC=(1<<URSEL) | (1<<USBS) | (3<<UCSZ0); //формат кадра данных:

//слово данных 8 бит, 2 стоп-бита
/* Отправка по USART пакета из двух байт: 3, 4 */
while ( !( UCSRA & (1<<UDRE)) ) ; // ждать, пока не будет готов буфер отправки
UDR = 3; // положить данные в буфер для отправки

while ( !( UCSRA & (1<<UDRE)) ) ; // ждать, пока не будет готов буфер отправки
UDR = 4; // положить данные в буфер для отправки
```

Рис. 3 – Программный код работы с USART на уровне регистров

2) Уровень управления микроконтроллером через программные модули управления аппаратными блоками микроконтроллера

На данном уровне микроконтроллер представляется как набор аппаратных блоков,

выполняющих аппаратные процессы, программа управляет этими процессами через соответствующие программные модули: задачи и пакеты, в которых выполняется настройка регистров. Большинство аппаратных блоков предназначены для реализации интерфейсов, т. е. стандартизованных в области микроконтроллерных систем протоколов канального уровня (UART, SPI, TWI, CAN, USB и т. д.) для передачи данных между программируемым контроллером и внешним оборудованием, подключенным к нему, кроме того блоки обеспечивают генерацию определенных форм сигналов и передачу данных по более простым цифровым и аналоговым интерфейсам. К числу таких блоков относятся таймеры, счетчики, ШИМ-блоки, АЦП, аналоговые компараторы, блоки организации внешних прерываний и т. д.. Вместе с тем, существуют аппаратные блоки, предназначенные строго для внутреннего использования программой микроконтроллера (некоторые таймеры / счетчики) и не выдающие сигналов для внешнего оборудования.

В рамках разрабатываемой интегрированной среды программирования существует библиотека программных модулей управления блоками, в которой описываются шаблоны модулей, для которых необходимо задать конкретные значения для возможность использования данных модулей, т. е. произвести инстанцирование модуля по шаблону. На рисунке 4 приведен пример программного кода на высокочувствительном языке программирования, в котором производится инстанцирование модуля управления аппаратным блоком UART из шаблона, с помощью этого модуля затем отправляется сообщение по одноименному интерфейсу микроконтроллера.

3) Уровень управления микроконтроллером через программные модули, которые моделируют аппаратные модули микроконтроллера

Данный уровень позволяет абстрагироваться от природы процессов за счет добавления виртуальных блоков, т. е. блоков, логика которых полностью реализуется программными модулями с возможностью использования модулей управления аппаратными блоками. Управление регистрами в данном случае используется как внутренняя реализация блока и служит для осуществления непосредственного считывания и вывода цифровых сигналов с ножек микроконтроллера. Кроме того, на этом уровне управления возможна иерархическая компоновка программных модулей, т. е. создание программных модулей, которые используют другие модули. Поэтому на этом уровне появляются модули, управляющие внешним подключенным к микроконтроллеру оборудованием, что выводит степень абстракции за рамки микроконтроллера. На рисунке 5 показан пример высокочувствительного кода, в котором происходит инстанцирование модуля управления виртуальными блоком RadioModuleControl, который управляет радиомодулем по интерфейсу UART (т. е. при использовании модуля управления аппаратным блоком UART).

4) Уровень управления взаимодействием программных модулей и настройки их параметров

На данном уровне распределенная система управления представляется как набор программных модулей (задач и пакетов), непосредственно взаимодействующих друг с другом, что позволяет абстрагироваться от участвующей в обмене информацией аппаратуры и низкоуровневых протоколов, по которым производится взаимодействие.

```
#init_template UART
{
    baud_rate = 19200;
    transmit = true;
```

```

receive = true;
word_size = UART.WORS_SIZES.bit8;
stop_bits = UART.STOP_BITS.TWO
}

// установить радио модуль в режим готовности
UART.TransmitBytes( 0x06, 0x00 );
UART.TransmitBytes( 0x05, 0x00 );
UART.TransmitBytes( RF23B_PWRSTATE_REGISTR, RF23B_PWRSTATE_READY );
UART.TransmitBytes( 0x3E, 2 ); // установить количество передаваемых байт (2)
UART.TransmitBytes( 0x7f, 5 ); // отправить байт "5"
UART.TransmitBytes( 0x7f, 6 ); // отправить байт "6"

// начать беспроводную передачу байтов
UART.TransmitBytes(RF23B_PWRSTATE_REGISTR, RF23B_PWRSTATE_TX);
UART.TransmitBytes(RF23B_INTERRUPT_REGISTR, RF23B_PACKET_SENT_INTERRUPT);

```

Рис. 4 – Программный код по работе с UART через модуль управления аппаратным блоком

```

#init_template UART
{
    <...>
}

#init_template RadioModuleControl
{
    UARTblock = UART; // задать UART в качестве используемого интерфейсного блока
}
RadioModuleControl.EnqueueBytes(5, 6); // добавить в очередь байты "5", "6"
RadioModuleControl.StartTransmit(); // начать беспроводную передачу
                                    // пакета данных

```

Рис. 5 – Программный код для отправки сообщения радиомодулю через модуль управления виртуальным блоком

Задачи выполняются параллельно относительно друг друга. Возможно как асинхронное взаимодействие между задачами, так и синхронное. Синхронное взаимодействие осуществляется при использовании механизма “рандеву” [12].

Также на данном уровне происходит абстрагирование от свойств пассивности / активности оборудования в схеме: можно считать, что каждое устройство может быть активным и любой из его модулей может инициировать отправку сообщений модулю другого устройства.

На рисунке 6 приведен пример на высокоуровневом языке, на котором одной строчкой кода настраивается автоматическая отсылка сообщения OnWithBrightness модуля управления светодиодом при каждой оцифровке значения, производимого модулем управления потенциометром.

```

// задание параметров шаблона модуля управления потенциометром
#initblock PotentiometerControl
{
    pin = [PORTB, 2]; // настройка вывода, через который

```

```

        // подключен потенциометр
    }

    <...> // задание параметров шаблонов модулей управления UART,
    //RadioModuleControl

    // автоматическое изменение яркости светодиода
    // при изменении значения потенциометра
    PotentiometerControl.OnValueRecieved += mcu2.LedControl.OnWithBrightness;

```

Рис. 6 - Программный код настройки автоматического изменения яркости светодиода при каждой оцифровке значения с потенциометра

5. Заключение

В работе рассмотрены особенности разработки программ для специализированных микроконтроллерных систем как подвида встраиваемых систем. Рассмотрены недостатки существующих подходов к программированию специализированных микроконтроллерных систем, не способных предоставлять одновременно высокий уровень разработки программ и реализовывать необходимые для эффективности статические элементы. В работе предложена новая модель представления программ управления распределенными микроконтроллерными системами. На основе предложенной модели разрабатывается инструментальная среда разработки программ для микроконтроллерных систем, использование которой дает следующие преимущества:

- повышается уровень программирования с уровня регистров до уровня распределенных по устройствам модулей управления - для этого используется язык программирования высокого уровня и графические редакторы-конфигураторы схемы оборудования;
- появляется возможность естественно представить в программе параллелизм распределенных процессов и упростить их синхронизацию;
- производится абстрагирование от коммуникационных протоколов, по которым производится взаимодействие программных модулей: системой производится генерация протоколов на основе настроек интерфейсных блоков;
- появляется возможность описывать кросс-платформенные программы управления с возможность получения кода для нужной модели микроконтроллера на нужном языке и для нужного компилятора.

Литература

1. Платунов А.Е., Постников Н.П. Высокоуровневое проектирование встраиваемых систем. – СПб.: НИУ ИТМО, 2011. – 121 с.
2. Белых А. А. Унификация архитектур однокристальных микроконтроллеров и ее применение для разработки программного обеспечения встраиваемых систем: дис. ... канд техн. наук: 05.13.15. — М., 2006. — 176 с.
3. Платунов А.Е. Встраиваемые системы управления // Control Engineering Россия, №1 (43), 2013.
4. Gridling G., Bettina W. Introduction to microcontrollers. — Vienna University of Technology, Institute of Computer Engineering, Embedded Computing Systems Group. — Vienna, 2006.
5. Atmel Studio [Electronic resource]. – Режим доступа: <http://www.atmel.com/ru/ru/tools/atmelstudio.aspx>, свободный. – Загл. с экрана.
6. Code Composer Studio (CCStudio) Integrated Development Environment (IDE) v5 [Electronic resource]. – Режим доступа: <http://www.ti.com/tool/ccstudio>, свободный. – Загл. с экрана.

7. ST Visual develop IDE for developing ST7 and STM8 applications [Electronic resource]. – Режим доступа: <http://www.st.com/web/en/catalog/tools/FM147/CL1794/SC1807/SS1747/PF210567>, свободный. – Загл. с экрана.
8. QTouch Composer — полная интеграция [Electronic resource]. – Режим доступа: http://www.atmel.com/ru/ru/Microsite/atmel_studio6/qtouch_tools.aspx, свободный. – Загл. с экрана.
9. TM32Cube initialization code generator (UM1718) [Electronic resource]. – Режим доступа: http://www.st.com/web/catalog/tools/FM147/CL1794/SC961/SS1743/PF259242?icmp=stm32cubemxpron_pr-stm32cubef2_apr2014, свободный. – Загл. с экрана.
10. Grace – Graphical Peripheral Configuration Tool [Electronic resource]. – Режим доступа: <http://www.ti.com/tool/grace>, свободный. – Загл. с экрана.
11. .Net Micro Framework [Electronic resource]. – Режим доступа: <http://www.netmf.com/>, свободный. – Загл. с экрана.
12. Деревянко А.С., Солошук М.Н. Операционные системы: Учебное пособие. - Харьков: НТУ "ХПИ", 2002. - 573с.

ИССЛЕДОВАНИЕ, РАЗРАБОТКА И МОДЕРНИЗАЦИЯ ПРОТОКОЛОВ ОБЕСПЕЧЕНИЯ КОНФИДЕНЦИАЛЬНОСТИ В МНОГОСТОРОННЕМ КЛАСТЕРНОМ АНАЛИЗЕ¹

Вашкевич Алексей Вячеславович

ФГБОУ ВПО «Сибирский государственный аэрокосмический университет им.ак.

М.Ф. Решетнёва»

Красноярск, alex23-5@yandex.ru

Жуков Вадим Геннадьевич

ФГБОУ ВПО «Сибирский государственный аэрокосмический университет им.ак.

М.Ф. Решетнёва»

Красноярск, vadimzhukov@mail.ru

Аннотация

В работе рассматриваются актуальные проблемы в разработке и применении протоколов конфиденциальных вычислений в кластерном анализе. Представлен обзор существующих решений конфиденциального кластерного анализа при различных секционированиях данных, а также предложены пути устранения их недостатков, в частности, указаны алгоритмы кластерного анализа и требования, которым должны соответствовать криптографические протоколы, основанные на них. Приведён пример модернизации реального протокола обеспечения конфиденциальности в кластерном анализе k-means. Сформулированы дальнейшие направления действий по разработке и модернизации протоколов кластерного анализа, сохраняющего конфиденциальность входных данных.

1. Введение

Основная цель кластерного анализа – выделить в исходных многомерных данных такие однородные подмножества, чтобы объекты внутри групп были похожи друг на друга, а объекты из разных групп – не похожи. Под «похожестью» понимается близость

¹ Работа поддержана грантом Президента молодым кандидатам наук, договор № 14.124.13.473-МК от 04.02.2013.

объектов в многомерном пространстве признаков, и тогда задача сводится к выделению в этом пространстве естественных скоплений объектов [1]. Вследствие этого кластерный анализ применяется в исследованиях практически любого рода, в частности, в интернет-технологиях, торговле, промышленном производстве, медицине, банковском деле, страховом бизнесе, информационной безопасности [2]. В таких исследованиях анализируемые данные могут быть распределены между несколькими организациями и при этом относиться к конфиденциальной информации, разглашение которой может привести к финансовым убыткам, недополученной прибыли и/или репутационным потерям.

Для обеспечения конфиденциальности информации в подобных распределённых исследованиях могут применяться услуги какой-либо доверенной сторонней организации, однако наличие такой организации не всегда возможно и допустимо [3]. По этой причине разрабатывались различные протоколы конфиденциальных вычислений для простых функций, использующие т.н. криптографические примитивы – относительно простые криптографические преобразования. Данные примитивы «встраиваются» в этапы алгоритма анализа данных, не изменяя исходный алгоритм, но защищая данные в нём от раскрытия другими участниками анализа.

Применяемых методов кластерного анализа достаточно много, выбор конкретного метода зачастую зависит от структуры анализируемых данных [4]. Поэтому нельзя применить протоколы конфиденциальных вычислений только к одному алгоритму кластеризации, требуется обеспечить конфиденциальность вычислений в наиболее популярных алгоритмах. Таким образом, формирование протоколов, сохраняющих конфиденциальность данных при кластеризации, является актуальной научно-практической задачей.

2. Особенности конфиденциальной кластеризации

В алгоритмах конфиденциальной кластеризации традиционно используется модель с т.н. «получестными» (semi-honest) участниками [5]. Получестные участники следуют правилам протокола, но вольны использовать то, что они видят в ходе выполнения протокола, для попытки раскрытия конфиденциальных данных других участников. При этом рассматривается возможность сговора (т.е. объединения знаний) некоторых из участников.

Модель с получестными участниками проще и больше отвечает реалиям проведения распределённых исследований. Предполагается, что получить результат анализа участникам важнее, чем узнать данные других участников. Следует заметить, что даже единственный злоумышленник из числа участников, предоставляющий для анализа неверную информацию, может сорвать получение верных результатов. Однако злоумышленник (или несколько злоумышленников) всё равно не должен узнать конфиденциальную информацию других участников, так как она всё равно не должна быть получена исходя из результата и хода выполнения протокола.

Криптографические примитивы, использующиеся в протоколах конфиденциальной кластеризации, условно можно разделить на два типа [6]: основанные, на гомоморфном шифровании (шифровании, позволяющем сохранять математические операции); основанные на случайных числах. В первых конфиденциальность базируется на сложности вычисления обратной функции к функции шифрования, во вторых – в невозможности разрешить систему уравнений с большим количеством неизвестных. Независимо от природы криптографических примитивов, они должны быть

корректными с математической точки зрения, т.е. приводить к таким же результатам анализа, как если бы их не было вообще.

При кластерном анализе, проводимом несколькими участниками, данные между ними могут быть распределены по-разному. Так, при горизонтальном секционировании разные участники анализа хранят данные о разных объектах, а при вертикальном секционировании – разные измерения (свойства, признаки) одних и тех же объектов. Также существует арбитральное (произвольное) секционирование, где данные между участниками распределены в произвольном порядке, однако оно распространено значительно реже. Соответственно, для защиты данных в кластерном анализе необходимо разработать два отдельных протокола и для горизонтального, и для вертикального секционирования. Протокол для произвольного секционирования, разумеется, можно применять и при горизонтальном, и при вертикальном, однако такой протокол сложнее в разработке и применении, более требователен к ресурсам.

При различных видах секционирования требуется защищать разные этапы кластерного анализа. Например, при вертикальном секционировании участники не должны узнать значения измерений объектов, хранимых другими участниками. Защищаются этапы, связанные с вычислением расстояний между объектами, где приходится суммировать разные измерения с помощью заданной метрики (Евклида, Манхэттенских кварталов и др.). При горизонтальном же секционировании вычисления расстояний, как правило, не нарушают конфиденциальность, т.к. происходят локально, а защищаются чаще всего только шаги, связанные с подсчётом средних значений в кластере, медиан и т.п., то есть когда происходят вычисления на основании объектов данных, хранимых у разных участников.

К особенностям проведения конфиденциальной кластеризации следует отнести:

- большие объёмы данных (за счет объединения баз данных для анализа несколькими участниками);
- необходимость в передаче данных по сети.

Из-за данных особенностей к протоколам предъявляются требования (разумеется, не в ущерб конфиденциальности) к как можно меньшим вычислительным и коммуникационным затратам. Поэтому в частности протоколы для произвольного секционирования мало распространены, т.к. не только сложны, но и требуют большого количества математических операций и переданных пакетов данных.

Для определения текущего положения вещей в сфере конфиденциальной кластеризации следует рассмотреть существующие решения по обеспечению конфиденциальности в наиболее популярных протоколах кластерного анализа.

3. Анализ и модернизация существующих решений по обеспечению конфиденциальности в кластерном анализе

Одним из наиболее распространённых алгоритмов кластеризации является метод k-means, что обусловлено простотой алгоритма. Вероятно, поэтому он имеет наибольшее число модификаций для обеспечения конфиденциальности данных в нём [6, 7]. Есть решения задачи конфиденциальной кластеризации k-means даже для произвольного секционирования и для любого количества участников. Однако у большинства работ есть определённые недостатки. Один из самых распространённых – наличие т.н. «особых», доверенных участников, которые выполняют некоторые дополнительные операции с данными и ни в коем случае не должны сговариваться друг с другом. Однако использование доверенных участников, по сути, сводит на нет одну из целей создания

протоколов конфиденциальной кластеризации – уход от схемы с «третьей доверенной стороной».

Более того, в ходе исследования, проведённого ранее [7], были обнаружены математические неточности алгоритмов. Так, в ходе протокола могли получаться неверные результаты, а также раскрывались данные одного из участников анализа. Также при сговоре всего двух участников раскрывались данные участника, находящегося в протоколе «между ними». Впрочем, в ходе исследования эти и другие выявленные недостатки были устранены. Подробнее это описано в следующем разделе.

Метод кластеризации c-means является некоторым усложнением k-means, добавляющим в исходный алгоритм «нечёткость» в определении ближайшего кластера. Вместо однозначной принадлежности объекта данных к некоторому кластеру теперь используется матрица принадлежности, где значения ячеек заполнены от 0 до 1, однако это никак принципиально не влияет на криптографические примитивы, применяемые для обеспечения конфиденциальности.

На рисунке 1 представлена общая схема встраивания криптографических примитивов в протоколы кластерного анализа, родственных с k-means и имеющих общую структуру:



Рис. 1 – Схема защиты шагов протокола кластерного анализа, подобного k-means.

Стоит отметить, что если условие останова определяется не отсутствием изменения принадлежности точек к кластерам, а сравнением минимизируемой величины с пороговым значением, то этот шаг также должен защищаться при вертикальном секционировании.

Алгоритмы семейства k-means являются частным случаем EM-алгоритма кластерного анализа. Данный алгоритм так же, как и алгоритм c-means, приводит к нечёткой кластеризации, однако нечёткость вызвана применением аппарата теории вероятности. Решение задачи конфиденциальной кластеризации данного метода есть только для горизонтального секционирования [8, 9]. Вероятно, это связано со сложностью подбора существующих и создания новых криптографических примитивов для вычислений с несколькими переменными, применяемых в данном алгоритме.

Алгоритм кластерного анализа k-medoids является похожим на k-means, однако использует не центры кластеров, которые генерируются и высчитываются в ходе протокола и ничего в реальности не значат, а т.н. «медоиды», которые являются

реальными объектами данных. В таком случае при вертикальном секционировании данных ничего принципиально по сравнению с k-means не меняется (каждый участник знает только свои измерения объектов данных), но при горизонтальном секционировании участники не должны знать характеристики медоида, принадлежащего лишь одному из участников. Протокол, сохраняющий конфиденциальность при горизонтальном секционировании, в открытых источниках найден не был. Сложность в создании такого протокола в том, что участнику-владельцу медоида необходимо каким-либо образом передать координаты медоида, а в открытом виде это делать нельзя.

Кластеризация методом k-ближайших соседей (k-nearest) имеет аналогичные особенности, поскольку в ней также высчитываются расстояния до реальных объектов данных. Для горизонтального секционирования существуют решения [10], однако в данном случае раскрывается часть информации о количестве объектов данных (пусть и обезличенных, т.е. без идентификатора) и их примерном расположении в пространстве.

Такие же особенности (простота встраивания примитивов при вертикальном секционировании и сложность при горизонтальном) будут у всех методов кластеризации, где для совместных расчётов в горизонтальном секционировании применяются координаты реальных объектов данных. Для кластеризации методом BIRCH (один из иерархических методов кластеризации) был придуман алгоритм для произвольного секционирования данных [11], однако он был разработан только для двух участников. Для многих алгоритмов кластеризации, например, CURE, FOREL и других, имеющих те же особенности, протоколы обеспечения конфиденциальности не разработаны даже для вертикального секционирования.

Самоорганизующие карты [12] являются методом кластеризации, основанном на нейронных сетях, и достаточно сильно отличаются от большинства других методов. Найден один протокол обеспечения конфиденциальности при вертикальном секционировании только для двоих участников [13]. При горизонтальном секционировании для обеспечения конфиденциальности необходимо совместно хранить части значения нейронов, не зная при этом настоящего значения, т.к. протокол построен на нахождении наиболее близкого нейрона для каждого объекта данных.

Существуют и другие методы кластеризации данных, однако они в целом менее распространены и к ним так или иначе можно применить те же криптографические примитивы или указать те же проблемы, что и для рассмотренных алгоритмов.

В таблице 1 указаны общие моменты, характерные для существующих решений и будущих разработок алгоритмов, предназначенных для обеспечения конфиденциальности в кластерном анализе.

Таблица 1
Сводная таблица эффективности применения криптографических примитивов.

Алгоритмы	Примеры	Вертикальное секционирование	Горизонтальное секционирование
Алгоритмы, в которых требуются вычисления между реальными и виртуальными	k-means, c-means, и т.п.	Существующие примитивы обеспечивают конфиденциальность и при вертикальном, и при горизонтальном секционировании. Требуется применить их ко всему семейству алгоритмов, а также стремиться к уменьшению	

объектами. Пересчёт виртуальных объектов происходит после всех вычислений в итерации		вычислительных и коммуникационных затрат	
Алгоритмы, в которых требуются вычисления между реальными и виртуальными объектами. Пересчёт виртуальных объектов происходит после каждого из вычислений	само-организующиеся карты и иные алгоритмы, основанные на нейронах	Существующие примитивы обеспечивают конфиденциальность только для двух участников. Требуется модернизировать их для любого количества участников	Решения отсутствуют. Предварительно: невозможно полностью обеспечить конфиденциальность, требуется применить примитивы, минимально раскрывающие информацию (и только косвенную)
Алгоритмы, в которых требуются вычисления только между реальными объектами	иерархические алгоритмы (BIRCH, CURE и др.), FOREL, k-nearest и др.	Решения либо отсутствуют, либо разработаны для малого количества участников. Требуется создать их и стремиться к уменьшению вычислительных и коммуникационных затрат	Решения либо отсутствуют, либо раскрывают информацию. Предварительно: невозможно полностью обеспечить конфиденциальность, требуется применить примитивы, минимально раскрывающие информацию (и только косвенную)

Следует также отметить следующие моменты, присущие протоколам конфиденциальной кластеризации:

- для большинства рассмотренных алгоритмов на настоящий момент существуют обеспечивающие конфиденциальность решения только при каком-либо одном секционировании и/или для определённого (а не произвольного) количества участников;
- даже разработанные протоколы обеспечения конфиденциальности могут иметь уязвимости и неточности, которые следует устраниТЬ. Математический аппарат обязательно должен быть уточнён.

4. Пример модернизации существующего протокола конфиденциальной кластеризации

Шаг 1. Определение «необходимой» степени раскрытия конфиденциальности для того, чтобы было возможно провести кластерный анализ.

При проведении конфиденциальных многосторонних вычислений неизбежно раскрытие той или иной косвенной информации о данных, хранимых остальными

участниками. Это происходит из-за того, что всем участникам известны свои собственные данные, общие конечные результаты анализа, а также некоторые сведения о промежуточных результатах анализа. Так, при вертикальном секционировании неизбежно раскрытие следующей информации: если участник имеет два объекта с одинаковыми значениями по измерениями, а объекты относятся к разным кластерам, то у других участников в совокупности эти объекты обладают различными значениями [14]. Такая информация некритична, однако в зависимости от методов анализа и криптографических примитивов степень раскрытия информации может варьироваться вплоть до существенных величин. Например, могут быть раскрыты данные некоторых (пусть и не всех) участников, что уже не допустимо.

Шаг 2. Выбор или разработка схемы с отсутствием «доверенных» участников, которые обладают большими сведениями, чем остальные. Так, для модернизации протокола обеспечения конфиденциальности в k-means при вертикальном секционировании в предшествующих работах за основу была взята [15] схема, предложенная Саметом и соавторами [16], поскольку в ней нет «доверенных» участников, в отличие от [17, 18].

Шаг 3. Подбор или создание криптографических примитивов, подходящих под нужные математические операции алгоритма, или изучение схем работы уже применяемых криптографических примитивов, поиск их недостатков. В решении Самета применялись примитивы «Безопасное скалярное произведение» (для двух участников) и «Безопасная сумма» (для трёх и более участников).

В «Безопасном скалярном произведении» на основе конфиденциальных данных формировались двухмерные вектора, которые при перемножении давали нужный результат. Однако анализ криптографического примитива показал [16], что при двухмерных векторах один из участников получает два уравнения с двумя неизвестными, решив которые, он узнаёт данные другого участника:

$$\begin{aligned} y_1 u_1 + y_2 u_2 &= \bar{y} \cdot \bar{u}, \\ y_1 v_1 + y_2 v_2 &= \bar{y} \cdot \bar{v}, \end{aligned}$$

где вектора \bar{u} и \bar{v} сгенерированы ранее самим участником, а результаты скалярных произведений $\bar{y} \cdot \bar{u}$ и $\bar{y} \cdot \bar{v}$ получены от участника-жертвы. Поэтому в модернизации было предложено [16] изменить исходные вектора, увеличив их размерность. Для увеличения размерности в уравнение второй участник добавляет случайное число z :

$$\left(\frac{d_1}{l_1}; \frac{1}{l_1} \right) \cdot (1; d_2) \Rightarrow \left(\frac{d_1 + z}{l_1}; \frac{-z}{l_1}; \frac{1}{l_1} \right) \cdot (1; 1; d_2).$$

Вследствие этого, в два уравнения, получаемые одним из участников, добавится ещё один неизвестный аргумент:

$$\begin{aligned} y_1 u_1 + y_2 u_2 + y_3 u_3 &= \bar{y} \cdot \bar{u}, \\ y_1 v_1 + y_2 v_2 + y_3 v_3 &= \bar{y} \cdot \bar{v}. \end{aligned}$$

Теперь чтобы раскрыть данные участника-жертвы, из двух уравнений с тремя неизвестными предстоит найти $y_1 + y_2$ и y_3 . Таким образом, при соблюдении некоторых условий (в частности, равенства $u_1 = u_2$ и $v_1 = v_2$ не должны выполняться одновременно), данные обоих участников не будут раскрыты.

«Безопасная сумма» (схема изображена на рисунке 2) используется с той же целью

(нахождение ближайшего кластера к точке), что и «Безопасное скалярное произведение», однако для большего количества участников. Основным недостатком является то, что если сговорятся участники $N-1$ и $N+1$, то они узнают данные участника N .

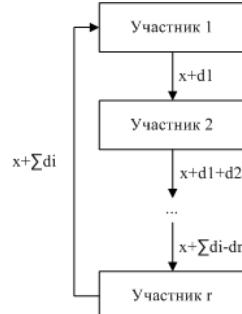


Рис. 2 – Схема криптографического примитива «Безопасная сумма»

Для обеспечения конфиденциальности в предыдущих исследованиях [14] было предложено использовать усовершенствованное «Безопасное скалярное произведение» (т.е. с трёхмерными векторами) для нескольких участников (пример схемы представлена на рисунке 3). Таким образом, решение не раскрывает никакую дополнительную информацию, в отличие от оригинальной работы.

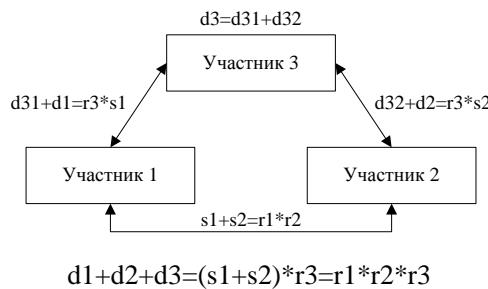


Рис. 3 – Использование «Безопасного скалярного произведения» для трёх участников

Шаг 4. Уменьшение коммуникационных и вычислительных затрат. В рассматриваемом алгоритме k-means «Безопасное скалярное произведение» требуется взаимодействие каждого участника с каждым. Вследствие этого при больших объёмах данных и большом количестве участников протокол будет значительно медленнее той же «Безопасной суммы». Требовалось подобрать или разработать алгоритм с большей скоростью работы.

С этой целью в ходе исследований протокол «Безопасная сумма» был усовершенствован [7]. Основным недостатком протокола, как уже было сказано, является опасность сговора соседей участника-жертвы. За основу для усовершенствования была взята идея суммирования данных по нескольким маршрутам так, чтобы соседи не повторялись.

Теорема. Пусть r – количество участников. Тогда определим семейство операций $+N$ на множестве $\{1, 2, \dots, r\}$ такое, что любой элемент из $Z_{r+1} \setminus \{0\}$ после ровно r применений операции из семейства $+N$ вернется сам в себя. Для этого все множители числа N должны быть взаимно простыми со всеми множителями числа r

(либо $N = 1$).

Если взять часть операций из этого семейства таких, что $N < \frac{r}{2}$, то можно составить такие маршруты суммирования, в которых соседи будут каждый раз меняться. Обозначим количество таких операций из $+N$ как n . Тогда максимальное допустимое количество сговорившихся участников, которое все равно обеспечит сохранение конфиденциальности данных, будет равно $2n - 1$, поскольку у участника $2n$ соседей по протоколу. Также можно заметить, что при простых r допустимое количество сговорившихся участников максимально и равно $r - 2$.

В таблице 2 приведён пример построения маршрутов суммирования. 7 – простое число, поэтому из чисел, меньших $7/2$, взаимно прост со всеми. 8 – составное число, из чисел, меньших $8/2$, взаимно прост только с 1 и 3.

Таблица 2

Примеры маршрутов суммирования.

Количество участников	7	8
$N < r/2$	1, 2, 3	1, 3
Маршруты	1-2-3-4-5-6-7 (+1)	1-2-3-4-5-6-7-8 (+1)
	1-3-5-7-2-4-6 (+2)	1-4-7-2-5-8-3-6 (+3)
	1-4-7-3-6-2-5 (+3)	–
Допустимо сговорившихся	3 маршрута * 2 – 1 = 5	2 маршрута * 2 – 1 = 4

Таким образом, требуется большее количество сговорившихся участников для раскрытия чьей-либо конфиденциальной информации. Количество маршрутов зависит от количества взаимно простых чисел (меньше $N/2$ и включая единицу) с общим числом участников. Начиная с 7 участников протокол даёт лучшие результаты по обеспечению конфиденциальности, чем оригинальная «Безопасная сумма».

Таблица 3

Зависимость количества маршрутов от общего количества участников.

Участников	Маршрутов	Допустимо сговорившихся участников
5	2	3
6	1	1
7	3	5
8	2	3
9	3	5
10	2	3
11	5	9

Шаг 5. Сравнительный анализ исходных и разработанных/модернизированных протоколов. Следует сравнить максимально допустимо количество сговорившихся участников в протоколах, а также затраты на выполнение протокола, в частности, сетевой трафик (данные о сетевом трафике приведены для определения ближайшего

кластера из двух для одного объекта данных на одной итерации).

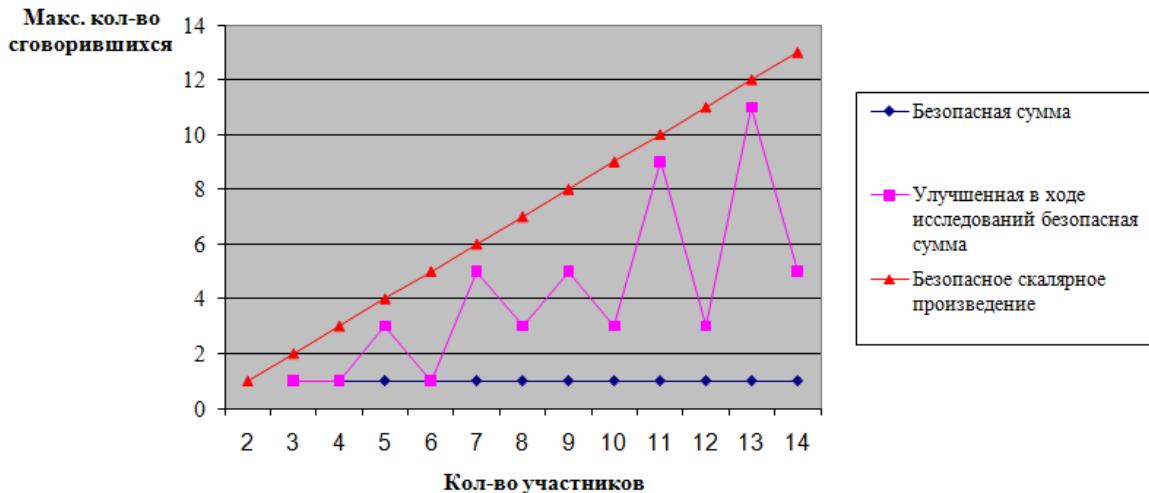


Рис. 4 – Зависимость допустимого числа говорившихся участников от общего числа участников

Таким образом, протокол обеспечения конфиденциальности считается улучшенным, т.к. приводит к обеспечению конфиденциальности и к приемлемым коммуникационным затратам как при малом количестве участников (корректно применяемое «Безопасное скалярное произведение»), так и при большом количестве участников (улучшенная «Безопасная сумма»).

5. Заключение

В настоящее время есть множество протоколов кластеризации, применяемых в целях анализа данных. Поскольку анализируемые данные могут относиться к конфиденциальной информации, следует использовать криптографические протоколы, позволяющие одновременно проводить анализ данных и защищать их от раскрытия другими участниками анализа.

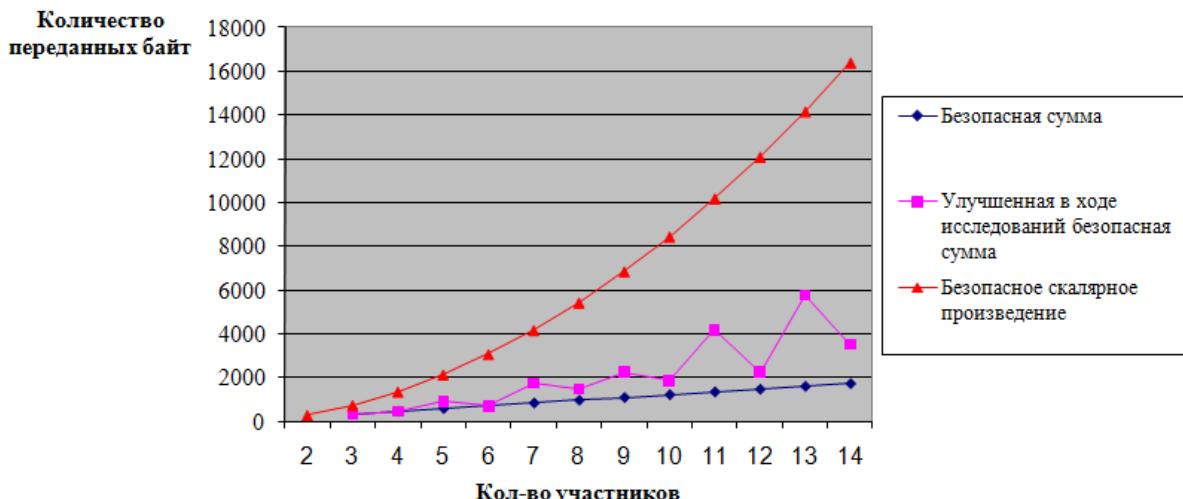


Рис. 5. – Зависимость сетевого трафика от общего числа участников

Существующие протоколы не всегда и не для всех секционирований данных

обеспечивают требуемую конфиденциальность. Более того, протоколы не всегда могут давать корректный результат, а также могут порождать большое количество коммуникационных и вычислительных затрат. Таким образом, в дальнейших исследованиях, посвящённым методам конфиденциальной кластеризации, следует проверить корректность существующих протоколов, разработать протоколы для пока ещё не защищённых алгоритмов кластеризации и решить проблему обеспечения конфиденциальности при горизонтальном секционировании. При этом необходимо применять криптографические примитивы, которые при равной степени обеспечения конфиденциальности приводят к наименьшим вычислительным и коммуникационным затратам.

Литература

1. Мандель И.Д. Кластерный анализ. – М.: Финансы и статистика. 1988. – 176 с.
2. Матвеев В.Г., Дмитриевский Б.С., Ляпин Н.Р. Информационные системы интеллектуального анализа. – М.: Машиностроение, 2008. – 92 с.
3. Шутый Р.С. Рандомизированные протоколы, применяемые для выполнения конфиденциальных многосторонних вычислений в компьютерных сетях. – Санкт-Петербург, 2009. – 170 с.
4. Нейский И.М. Классификация и сравнение методов кластеризации // Интеллектуальные технологии и системы. Сборник учебно-методических работ и статей аспирантов и студентов. – М.: НОК «CLAIM», 2006. – Выпуск 8. – С. 130-142.
5. Vaidya J., Clifton C. Privacy-preserving k-means clustering over vertically partitioned data // Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. – Washington, 2003. – P. 206-215.
6. Meskine F., Bahloul S.N. Privacy preserving k-means clustering: a survey research // International Arab Journal of Information Technology, 2012. – Vol. 9(2) – P. 194-200.
7. Vashkevich A.V., Zhukov V.G. Privacy-preserving protocol over Vertically Partitioned Data in Multiparty K-means Clustering // Middle-East Journal of Scientific Research, 2013. – Vol. 17(7) – P. 992-997.
8. Clifton C., Kantarciooglu M., Vaidya J., Lin X., Zhu M.Y. Tools for privacy preserving distributed data mining // ACM SIGKDD Explorations Newsletter, 2003. – Vol. 4(2) – P. 28-34.
9. Lin X., Clifton C., Zhu M.Y. Privacy-preserving clustering with distributed EM mixture modeling // Knowledge and Information Systems, 2005. – Vol. 8(1) – P. 68-81.
10. Shaneck M., Kim Y., Kumar V. Privacy preserving nearest neighbor search // Machine Learning in Cyber Trust, 2009. – P. 247-276.
11. Prasad P.K., Rangan C.P. Privacy Preserving BIRCH Algorithm for Clustering over Arbitrarily Partitioned Databases // Proceedings of the 3rd international conference on Advanced Data Mining and Applications, 2007. – P. 146-157.
12. Зиновьев А.Ю. Визуализация многомерных данных. – Красноярск: Изд. Красноярского государственного технического университета, 2000. – 180 с.
13. Han S., Ng W.K. Privacy-Preserving Self-Organizing Map // Data Warehousing and Knowledge Discovery: Lecture Notes in Computer Science, 2007 – Volume 4654, P. 428-437.
14. Жуков В.Г., Вашкевич А.В. Конфиденциальный кластерный анализ при вертикальном секционировании данных // Материалы 13-й международной научной конференции «Информационная безопасность», Таганрог, 2013. – С. 191-198.
15. Жуков В.Г., Ващекевич А.В. Модифицированное безопасное скалярное произведение в конфиденциальном кластерном анализе k-means с вертикальным секционированием данных // Вестник Сибирского государственного аэрокосмического университета, 2013. – Выпуск 4(50). – С. 18-22.
16. Samet S., Miri A., Orosco-Barbosa L. Privacy-Preserving K-Means Clustering in Multi-Party Environment // Proceedings of International Conference on Security and Cryptography, Barcelona, Spain, 2007. – P. 381-385.
17. Vaidya J., Clifton C. Privacy-Preserving K-Means Clustering Over Vertically Partitioned Data // Proceedings of the 9th ACM SIGKDD International Conference on Knowledge discovery and Data Mining, Washington, USA, 2003. – P. 206-215.

18. Doganay M., Pederson T., Saygin Y., Savas E., Levi A. Distributed Privacy Preserving Clustering with Additive Secret Sharing // Proceedings of the International Workshop on Privacy and Anonymity in Information Society, 2008. – P. 6003-6011.

РАЗРАБОТКА МНОГОПОЛЬЗОВАТЕЛЬСКОГО WEB-СЕРВИСА ИНФОРМАЦИОННОЙ ПОДДЕРЖКИ РАДИОЛОГИЧЕСКОГО МОНИТОРИНГА

А.В. Вицентий

Институт информатики и математического моделирования Кольского НЦ РАН,
Кольский филиал ПетрГУ
184205, г. Апатиты

Аннотация

В качестве практического приложения организации фрагмента региональной системы информационной поддержки экологического мониторинга рассматривается задача разработки многопользовательского web-сервиса для информационной поддержки радиологического мониторинга и визуального анализа данных. В общем виде приводятся схема работы, архитектура и пользовательский интерфейс сервиса. Кратко описываются основные возможности web-сервиса с точки зрения конечного пользователя. Web-сервис построен на основе данных радиологического мониторинга по Мурманской области.

1. Введение

Основы экологической политики России закреплены в Конституции Российской Федерации (РФ), Федеральном законе “Об охране окружающей среды”, указах Президента РФ “О Концепции перехода Российской Федерации к устойчивому развитию” и “О Концепции национальной безопасности Российской Федерации”, в Экологической доктрине РФ, и других нормативно-правовых актах. В соответствии с основными понятиями федерального закона "Об охране окружающей среды", государственный экологический мониторинг (ГЭМ) (государственный мониторинг окружающей среды) это комплексные наблюдения за состоянием окружающей среды, в том числе компонентов природной среды, естественных экологических систем, за происходящими в них процессами, явлениями, оценка и прогноз изменений состояния окружающей среды [1]. Мониторинг осуществляется посредством создания и обеспечения функционирования наблюдательных сетей и информационных ресурсов в рамках подсистем единой системы ГЭМ, а также создания и эксплуатации государственного фонда данных. Однако, такие сети и ресурсы, принадлежат различным службам и на практике сильно разобщены и не скординированы. Поэтому задача подготовки оценок, прогнозов, критериев альтернатив выбора управлеченческих решений на базе имеющихся в регионе ведомственных данных становится, в общем случае, неопределенной [2].

На сегодняшний день, единая система ГЭМ включает в себя 15 различных подсистем, среди которых присутствует и подсистема государственного мониторинга радиационной обстановки на территории РФ. Кроме того, Комитетом промышленного

развития экологии и природопользования Мурманской области (по данным мониторинга которой выполнена эта работа) разработана Концепция функционирования и развития сети особо охраняемых природных территорий (ООПТ) Мурманской области до 2018 года и на перспективу до 2038 года, целью которой является сохранение и восстановление естественных экосистем на территории Мурманской области, поддержание экологического равновесия и выявление закономерностей естественного развития природных комплексов и их компонентов.

Неотъемлемой частью данной Концепции является Схема развития и размещения ООПТ в Мурманской области, которой предусматривается до 2038 года организовать и реорганизовать более 40 новых ООПТ, что позволит довести площади, занимаемые ООПТ до 16,4% от площади области. Также правительством Мурманской области поддержана инициатива общественных экологических организаций о создании новой ООПТ федерального значения - национального парка «Хибины» на территории Хибинского горного массива (территория четырех муниципальных образований - г. Апатиты, Кировск, Оленегорск с подведомственными территориями, муниципальный район «Ловозерский район»). Целью создания является сохранение уникального природного комплекса, повышение туристической привлекательности региона и развитие соответствующей инфраструктуры [3].

Таким образом, задача применения новых подходов в области оценки радиологического загрязнения, использования новых методов получения данных и разработки средств мониторинга и прогнозирования радиационной обстановки на территории РФ является крайне актуальной.

2. Экологический мониторинг и источники данных ДЗЗ

В общем виде процесс экологического мониторинга можно представить в виде последовательности: окружающая среда → измерение параметров различными подсистемами мониторинга → сбор и передача информации → обработка и представление данных (формирование обобщенных оценок) → прогнозирование. Система экологического мониторинга предназначена для обслуживания систем управления качеством окружающей среды (система управления). В системе управления можно также выделить три подсистемы: принятие решения (специально уполномоченный государственный орган), управление выполнением решения (администрация предприятий и т.п.), выполнение решения с помощью различных технических или иных средств. В хорологическом аспекте обычно выделяют системы (или подсистемы) детального, локального, регионального, национального и глобального уровней мониторинга [4, 5].

Обычно научно обоснованный мониторинг осуществляется в соответствии с программой мониторинга окружающей среды, которая должна включать в себя общие цели организации, конкретные стратегии его проведения и механизмы реализации. Ключевыми элементами программы мониторинга окружающей среды являются [2]: перечень объектов, находящихся под контролем с их строгой территориальной привязкой; перечень показателей контроля и допустимых областей их изменения; временные масштабы - периодичность отбора проб, частота и время представления данных. Кроме того, в приложении в программе мониторинга должны присутствовать схемы, карты, таблицы с указанием места, даты и метода отбора проб и представления данных.

Первые автоматические системы слежения за параметрами внешней среды были

созданы в рамках военных и космических программ. Одним из основных источников данных для экологического мониторинга являются материалы дистанционного зондирования Земли (ДЗЗ). В общем случае, они объединяют данные, получаемые как с космических аппаратов, так и с приборов авиационного базирования. Аэрокосмические (дистанционные) методы экологического мониторинга включают систему наблюдения при помощи самолетных, аэростатных средств, спутников и спутниковых систем, а также систему обработки данных дистанционного зондирования[6]. Для решения задач космического экологического мониторинга можно опираться на данные, полученные как на отечественных аппаратах (спутники типа «Метеор», «Океан» и «Ресурс»), так и на американских спутниках серий NOAA, Landsat и SPOT.

В своей работе мы опирались на данные, получаемые с космических аппаратов (КА) программы Landsat. Программа Landsat - наиболее продолжительный на сегодняшний день проект по получению спутниковых фотоснимков планеты Земля. Первый из спутников в рамках программы был запущен в 1972; последний - Landsat 8 - 11 февраля 2013.

Landsat 8 - американский спутник дистанционного зондирования Земли, создан совместно NASA и USGS. Аппарат ведет съемку в 11 различных диапазонах с помощью OLI (Operational Land Imager) и TIRS (Thermal InfraRed Sensor) инструментов. OLI работает в 9 спектральных диапазонах, семь из которых близки к тем, которые использовались в более ранних инструментах Thematic Mapper (TM) и Enhanced Thematic Mapper Plus (ETM+) с предыдущих спутников Landsat, за счет чего обеспечивается преемственность и совместимость с ранее накопленным массивом данных. Добавлено два новых диапазона, канал 1 (темно-синий и фиолетовый) для изучения прибрежных вод и аэрозолей и канал 9 (ближний ИК) для упрощения поиска облаков на снимках.

Инструмент TIRS предназначен для получения изображений в дальнем ИК, он использует тот же принцип получения изображений pushbroom, что и OLI, и также имеет полосу обзора в 185 километров. Получение изображений происходит в двух каналах, 10 и 11, которые, совместно, работают в том же диапазоне, что и канал TIR на более ранних спутниках программы Landsat [7-9]. Landsat 8 получает изображения в видимом диапазоне волн, в ближнем ИК и в дальнем ИК, с разрешением снимков от 15 до 100 метров на точку. Производится съемка суши и полярных регионов. В сутки снимается порядка 400 сцен.

Основные научные задачи Landsat 8 [10]: сбор и сохранение многоспектральных изображений среднего разрешения (30 метров на точку); сохранение геометрии, калибровки, покрытия, спектральных характеристик, качества изображений и доступности данных на уровне, аналогичном предыдущим спутникам программы Landsat; бесплатное распространение изображений, полученных с помощью Landsat 8.

3. Разработка многопользовательской системы информационной поддержки радиологического мониторинга (по данным Мурманской области)

Развитие визуальной интерпретации многомерных данных и ГИС-технологий в области экологического мониторинга связано, в частности, с тем, что человеку с его ограниченным трехмерным пространственным воображением сложно, а в большинстве случаев невозможно, анализировать и давать обобщенные оценки многомерным объектам. Одним из первых опытов комплексного анализа пространственно

распределенной экологической информации явились поэтапные работы Института экологии Волжского бассейна РАН на примере данных о состоянии экосистем города Тольятти, Самарской области и всего Волжского бассейна, выполняемые с 1989 г. [11-14]. Эколого-экономическая информация по Волжскому бассейну, собиралась в виде различного рода карт распределения тех или иных параметров.

Что касается радиологического мониторинга в Мурманской области, можно сказать, что она является одним из самых потенциально опасных регионов Российской Федерации с точки зрения ядерной и радиационной безопасности. Здесь находятся Кольская АЭС, атомные ледоколы, военные подводные и надводные корабли с ядерными энергетическими установками, объекты временного хранения отработавшего ядерного топлива, большой пункт временного хранения реакторных отсеков, предприятия по обслуживанию и ремонту атомных судов. Радиационную обстановку вокруг этих объектов контролируют разные ведомства. Например, за фоном вокруг Кольской АЭС наблюдают как специалисты самой станции, так и Межрегиональное управление № 118 Федерального медико-биологического агентства России [15].

Создание систем непрерывного мониторинга радиационной обстановки и доступность данных - это требование времени и необходимость [16]. Важную роль в эффективности функционирования таких систем играет применение современных информационных технологий и использование новых источников данных, примером которых могут быть данные, полученные с помощью дистанционного зондирования Земли (ДЗЗ) из космоса - прежде всего спутниковые снимки.

Такие системы направлены на удовлетворение потребности специалистов различных предметных областей в получении оперативной информации радиологического мониторинга. А внедрение информационных продуктов, полученных на основе данных ДЗЗ и компьютерных методов картографирования и анализа, в свою очередь, позволяет улучшить качество обработки данных и более адекватно отражать состояние сложных биосоциоэкономических систем. В данной работе в качестве примера представлен мониторинг радиологической обстановки по пяти видам измерений: почва, вода, аэрозоли, осадки и пища. Данные измерений хранятся в соответствующих базах данных и визуализируются средствами разработанной системы.

Для визуального анализа результатов радиологического мониторинга специалист через интуитивно понятный веб-интерфейс получает возможность выбрать на интерактивной карте необходимые точки, в которых происходили измерения, и обозначить тот период времени, данные за который его интересуют. Эта информация визуализируется в виде графиков, отображающих значения измерений за выбранный период. Анализируя экстремумы графиков, специалист определяет области на карте, космические снимки которых он хотел бы проанализировать дополнительно, а также примерные даты получения снимков и прочие характеристики.

В процессе практической реализации система информационной поддержки радиологических измерений (СИПРИ) была разделена на три логических блока (Рис. 1.), взаимодействие которых организовано посредством протоколов HTTP и FTP. Комплекс тематической обработки данных и средства анализа данных и решения предметно-ориентированных задач предоставляются специалисту в виде готовых web-сервисов, имеющих интуитивный интерфейс, что обеспечивает более эффективное их использование. Реализация блоков, отвечающих за доступ и первичную обработку данных, а также физическая организация хранения данных скрыты от пользователя.

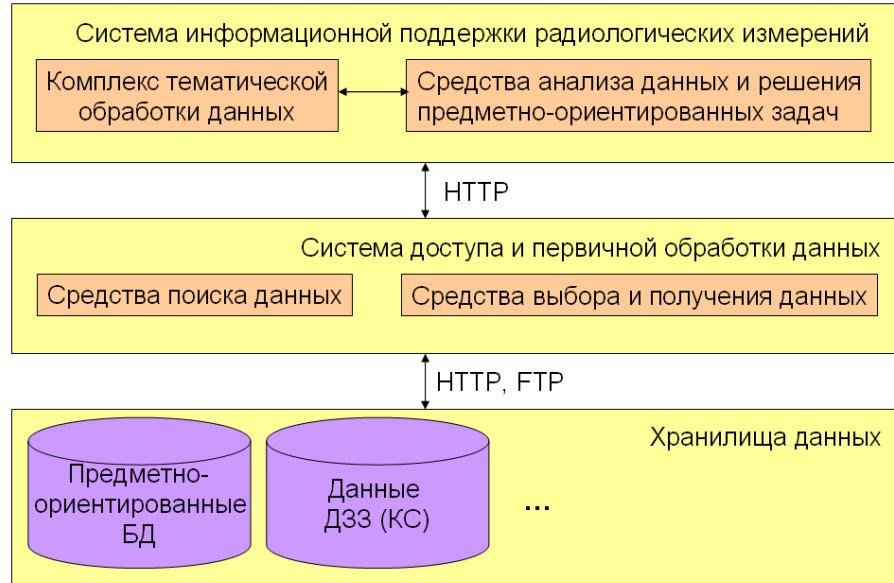


Рис. 1 – Основные блоки СИПРИ.

Архитектурно СИПРИ представлена двумя уровнями – уровнем представления, обеспечивающем интерфейс пользователю и прикладным уровнем, реализующим вычислительные сервисы (рис. 2)

Пример интерфейса, отображающего наложение спутникового снимка космического аппарата Landsat 8 [17] представлен на рисунке 3. Однако архитектура приложения предоставляет возможность подключения и других источников космических снимков исследуемых территорий.

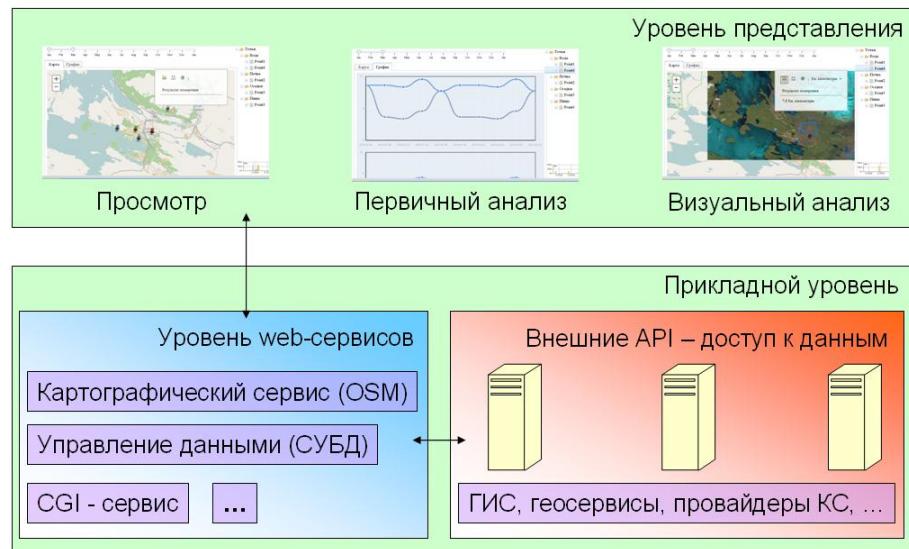


Рис. 2 – Архитектура СИПРИ.

Первичная навигация для выбора необходимых точек мониторинга, а также картографическая подложка обеспечивается средствами технологии OpenStreetMap [18],

что соответствует современным тенденциям в области разработки предметно-ориентированных геопорталов

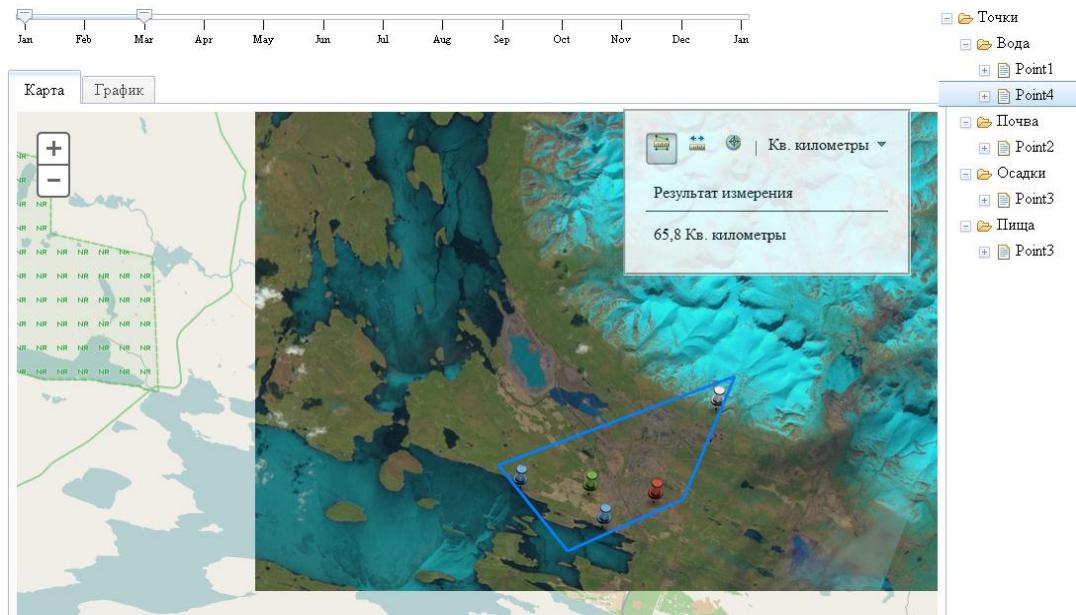


Рис. 3 – Пример интерфейса пользователя.

4. Заключение

Таким образом, разрабатываемая система информационной поддержки радиологических измерений позволяет специалистам на основе визуального анализа полученных данных дистанционного зондирования, сделать предположение о причинах повышения уровня радиации и спрогнозировать дальнейшие изменения.

В настоящее время ведется работа по включению в систему модуля реализующего технологию динамического когнитивного картографирования биосоциоэкономических систем с использованием результатов дистанционного зондирования Земли, что поможет существенно расширить ее функциональные возможности и позволит использовать ее совместно с другими прикладными технологиями информационной поддержки управления развитием компонентов пространственно-организованных биосоциоэкономических систем.

Литература

1. Федеральный закон от 10.01.2002 N 7-ФЗ (ред. от 12.03.2014) "Об охране окружающей среды" - "Российская газета", N 6, 12.01.2002.
2. Сюткин В. М. Экологический мониторинг административного региона (концепция, методы, практика на примере Кировской области). - Киров: ВГПУ, 1999. - 232 с.
3. Информация и предложения Комитета промышленного развития экологии и природопользования Мурманской области. - Интернет-конференция по теме «Проблемы регулирования земельных отношений в районах Крайнего Севера и приравненных к ним местностях» (март - май 2011 года). – Режим доступа: <http://www.severcom.ru/kit/abc/id315.html>.
4. Израэль Ю. А. Экология и контроль состояния природной среды. - Л.: Гидрометеоиздат, 1979, - 376 с.
5. Израэль Ю.А Глобальная система наблюдений. Прогноз и оценка окружающей природной среды. Основы мониторинга. - Метеорология и гидрология. 1974, № 7. - С.3-8.
6. Виноградов Б. В. Аэрокосмический мониторинг экосистем. - М.: Наука, 1984. - 320 с.

7. Fact Sheet - LDCM Earth Image Collection Satellite. Orbital Sciences Corporation. Режим доступа: http://www.orbital.com/newsinfo/publications/LDCM_Fact.pdf.
8. LDCM Spacecraft. NASA. Режим доступа: http://www.nasa.gov/mission_pages/landsat/spacecraft/index.html.
9. United Launch Alliance Successfully Launches Second NASA Payload in Just 12 Days. United Launch Alliance (11 February 2013). Режим доступа: <http://www.ulalaunch.com/site/pages/News.shtml#/129/>.
10. U.S. Geological Survey Landsat Data Continuity Mission. Rolla Publishing Service Center (July 2012). Режим доступа: <http://pubs.usgs.gov/fs/2012/3066/fs2012-3066.pdf>.
11. Моисеенкова Т.А., Шитиков В.К. Принципы организации регионального банка эколого-экономической информации // Моделирование процессов экологического развития (М., ВНИИСИ АН СССР). 1989. № 7. С. 110-117.
12. Розенберг Г.С. К построению системы концепций современной экологии // Журн. общ. биологии. 1991. Т. 52. № 3. С. 422-440.
13. Розенберг Г.С., Краснощеков Г.П., Шитиков В.К. К созданию пространственно-распределенной базы эколого-экономических данных бассейна крупной реки (на примере Волжского бассейна) // Вопросы экологии и охраны природы в лесостепной и степной зонах: Межвед. сб. науч. тр. - Самара: Изд-во "Сам. ун-т", 1995. С. 8-15.
14. Розенберг Г.С., Краснощеков Г.П. Волжский бассейн: экологическая ситуация и пути рационального природопользования - Тольятти: ИЭВБ РАН, 1996. - 249 с.
15. Межрегиональное управление № 118 Федерального медико-биологического агентства России. – Режим доступа: <http://www.ru118.ru/>.
16. В Мурманске обсудили проблему мониторинга радиационной обстановки. – Режим доступа: http://bellona.ru/articles_ru/articles_2012/1337334623.04.
17. Landsat Science. – Режим доступа: <http://landsat.gsfc.nasa.gov/>.
18. Openstreetmap. – Режим доступа: <http://www.openstreetmap.org>.

СИСТЕМА РЕЦЕНЗИРОВАНИЯ НАУЧНОГО КОНТЕНТА

Воронина Варвара Валентиновна

ЗАО «РосИнтернет технологии», г. Рыбинск, vvoronina@ritech.ru

Пучков Виталий Викторович

ЗАО «РосИнтернет технологии», г. Рыбинск, vpuchkov@ritech.ru

Аннотация

В работе рассматриваются проблемы рецензирования научных текстов в рамках подготовки докладов научных конференций. Рассматриваются соответствующие российские и зарубежные системы, существующие на данный момент, приводятся их достоинства и недостатки. Большое внимание уделяется описанию разработанной системы рецензирования научных текстов, делается вывод о её преимуществах.

1. Введение

Рецензирование — процедура рассмотрения научных статей и монографий учёными-специалистами в той же области. Цель рецензирования до публикации — удостовериться в точности и достоверности изложения и в необходимых случаях добиться от автора следования стандартам, принятым в конкретной области или науке в целом. Публикации работ, не прошедших рецензирования, часто воспринимаются с недоверчивостью профессионалами во многих областях.

Рецензирование является необходимой составляющей любой систематической обработки научного контента, например, допуск докладов на конференцию, отбор статей

для печати в номере научного журнала и т.д. В данной статье мы будем рассматривать организацию рецензирования докладов научных конференций. Речь пойдет об автоматизации этого процесса и упрощении взаимодействия автора и рецензента.

2. Обзор аналогов

За рубежом проведение научных конференций сопровождается, как правило, регистрацией в таких системах, как EasyChair [1], Peerage of Science [2] и EDAS [3]. Данные веб-системы включают в себя все необходимое для проведения конференций, в том числе, и средства для работы с докладами участников, позволяющие автоматизировать процесс рецензирования заявленных на конференцию статей.

Следует отметить, что недостатками данных систем являются:

1. отсутствие возможности построения произвольных настраиваемых веб-форм рецензирования для базовых бесплатных версий;

2. избыточное количество функционала, большая часть которого остается невостребованной для рецензируемых работ, но затрудняет доступ к действительно необходимым функциям (оценка адекватности рецензий другими рецензентами, статистика по работе системы и экспорт данных в различные форматы и др.);

3. данные системы являются англоязычными, что затрудняет их использование в России для широкой аудитории авторов.

Ввиду перечисленных недостатков, данные системы почти не привлекаются для организации процессов проведения конференций в России.

Среди русскоязычных систем наиболее близко к решению проблемы подходит RAE Conference System [4]. Данная система включает в себя следующие возможности: управление контентом (информация, представленная в текстовом, графическом, аудио и видео форматах, и необходимая участникам конференции), регистрация, размещение докладов в системе, формирование списка участников. Вопрос автоматизации рецензирования докладов данной системой не затрагивается.

Подготовка докладов, как для зарубежных, так и для российских конференций, часто проходит в формате общения организаторов, участников и рецензентов по электронной почте. Организаторы распределяют доклады по рецензентам, а те, в свою очередь, направляют рецензии авторам и организаторам по почте. Этот процесс требует внимания и отнимает у всех его участников много времени, здесь возможны потери и задержка данных в связи с ошибками пересылки.

Также встречаются попытки автоматизировать хотя бы прием докладов. Участники конференции регистрируются на сайте и подают доклады через специальную форму, но рецензирование в таких системах по-прежнему осуществляется через ручную почтовую рассылку.

Таким образом, перечисленные особенности данной области разработки обуславливают необходимость в создании новой системы рецензирования научного контента.

Проведенный анализ аналогов, а так же статей, посвященных описанию и исследованию процесса рецензирования научных работ [5, 6], позволил выделить следующую необходимую функциональность разрабатываемой системы:

- регистрация и построение списка пользователей с присвоением им соответствующих ролей, управление пользователями и их ролями;

- загрузка статей с помощью настраиваемых форм, построение списка статей, фильтрация, версионность (управление версиями загружаемых статей);
- рецензирование с применением настраиваемых форм;
- возможность изменять параметры рецензирования (количество рецензий на одну статью, зависимость статуса статьи от оценки рецензента, критерии оценки статьи рецензентом и т.д.);
- возможность как ручного, так и автоматического закрепления статей за рецензентами;
- возможность менять лингвистические настройки системы (использование нескольких языков).

3. Система рецензирования докладов и публикаций

Система рецензирования научного контента – это компонент, подключаемый к любому веб-порталу и осуществляющий прием научных текстов (доклады конференций, статьи для научных журналов), их обработку в зависимости от поставленных условий (научная новизна, совпадение тематики с заявленной, уровень грамотности и связности текста и др.)

Система значительно упрощает процесс рецензирования докладов, включает в себя несколько настраиваемых параметров и достаточно легко масштабируется на любой уровень научной конференции, а также на любой базовый веб-портал.

В процессе рецензирования участвуют пользователи солями **Автор**, **Рецензент** и **Администратор**.

Автор добавляет свою статью в систему, загружая файл через специальную форму (см. рис. 1).

Новая статья

* - поля, обязательные для заполнения.

[Правила оформления статьи](#)

[Порядок рецензирования статьи](#)

Название статьи: *

Тематика статьи: *



Аннотация: *



Соавторы:



Ключевые слова: *



Файл со статьей: *

[Обзор...](#)

Файл не выбран.

[Отмена](#)

[Отправить](#)

Рис.1 – Форма загрузки статьи.

При добавлении статьи **Автором**, статья принимает статус «**Ожидает рецензирования**». Этот статус виден и **Автору** и **Рецензентам** в их **Личных кабинетах**. **Автор** может удалять статьи в корзину и загружать новые версии статей.

По решению программного комитета статус статьи может поменяться на «**Принять**», «**Отклонить**», либо «**Принять с обязательным исправлением**».

Затем к статье прикрепляются **Рецензенты**. Прикрепить **Рецензентов** можно с помощью автораспределения, либо вручную. Автораспределение **Рецензентов** – это специальная возможность системы, позволяющая за каждым докладом случайным образом закрепить заданное количество **Рецензентов** так, чтобы всем **Рецензентам** системы досталось оптимальное количество докладов.

Рецензенты оставляют свои рецензии через специальную форму рецензирования (см. рис.2).

НОВАЯ РЕЦЕНЗИЯ

*- поля, обязательные для заполнения.

[Порядок рецензирования статьи](#)

Внимание! Вариант ответа "Нет" на любой из первых двух вопросов автоматически отклоняет статью, не зависимо от других оценок.

Соответствие тематике:*	<input type="text" value="Нет"/>
Научная новизна и значимость материала:*	<input type="text" value="Нет"/>
Уровень русского языка, грамотность:*	<input type="text" value="Не соответствует принятому уровню"/>
Соответствие требованиям к оформлению:*	<input type="text" value="Нет"/>
Оценка содержания статьи:*	<input type="text" value="Категорически не принимается"/>
Рецензия:*	<input style="height: 300px; width: 100%;" type="text"/>

Рис.2 – Форма рецензирования докладов.

После того, как первый **Рецензент** оставил рецензию на статью (или сохранил драфт рецензии), статус статьи становится «**На рецензировании**». Драфт рецензии – это черновик, т.е. некоторая часть рецензии, которую рецензент может в системе сохранить, а в последующем дополнить. Драфт не считается рецензией. Данный статус виден **Автору** и **Рецензентам**. На данном этапе **Автор** может удалять статьи в корзину.

После того, как на статью написали количество рецензий, необходимое для анализа (на данный момент установлена планка в две рецензии), идёт автоматическая проверка. По результатам проверки автоматически выставляется один из трёх статусов:

- «Рекомендуется принять с обязательным исправлением»,
- «Рекомендуется принять»,
- «Рекомендуется отклонить».

Эти статусы видны Администратору, а также Рецензентам. У Автора в это время остаётся статус «На рецензировании».

Далее проходит заседание программного комитета, после которого выносятся решения:

- «Принять»,
- «Отклонить»,
- «Принять с обязательным исправлением».

Эти статусы видны Автору и Рецензентам. Автору приходит письмо с извещением о смене статуса его статьи на один из трех указанных. Если статус работы «Принять», то Автор не имеет возможности заменить либо удалить статью. Статьи со статусом «Отклонить» Автор может удалять в корзину.

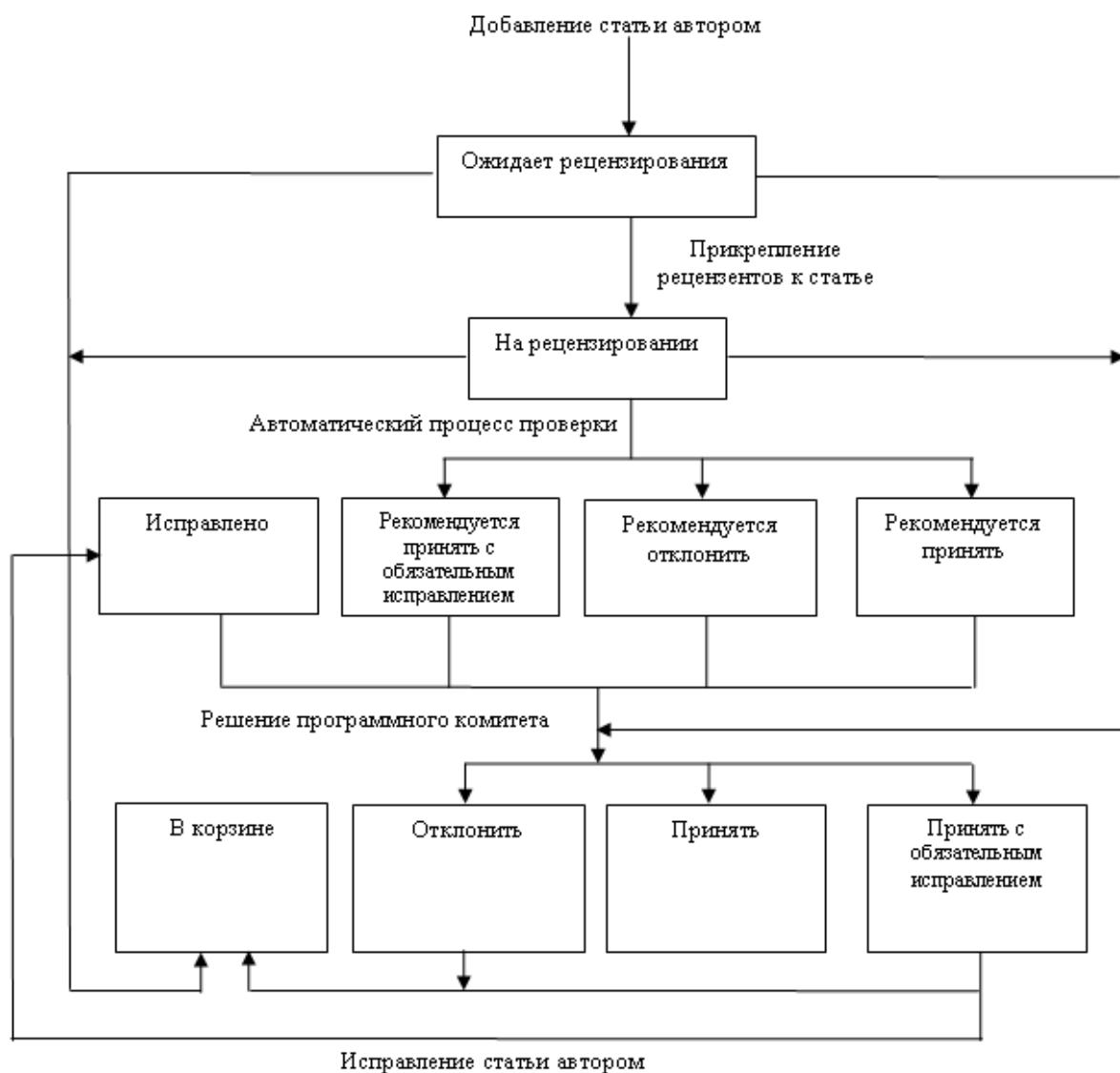


Рис.3 – Общая схема процесса рецензирования.

Общая схема процесса рецензирования и смены статуса статьи в системе представлена на рис.3.

Автор, получив письмо с уведомлением о том, что статус его статьи сменился на

«Принять с обязательным исправлением», может либо удалить ее в корзину, либо исправить статью, затем загрузить новую версию (**Рецензентам** не приходит письмо о загрузке новой версии статьи).

При этом статус статьи меняется на **«Исправлено»**. **Администратору** отправляется письмо о смене статуса соответствующей статьи на **«Исправлено»**. Статус виден **Автору**, а также **Администратору** системы. У **Рецензента** так и остается статус **«Принять с обязательным исправлением»**.

После этого программный комитет ещё раз рассматривает статью и выносит окончательное решение, которое видят все. **Автор** оповещается об этом автоматически.

Необходимо отметить, что система сама не принимает окончательных решений. Выводы о докладе, сделанные системой на основании полученных рецензий, носят рекомендательный характер и всегда могут быть оспорены программным комитетом конференции, как в сторону повышения, так и в сторону понижения результата. Статусы **«Принять»**, **«Отклонить»** и **«Принять с обязательным исправлением»** выставляет именно программный комитет конференции, таким образом, все участники процесса рецензирования застрахованы от ошибки и излишнего субъективизма оценки.

4. Заключение

Первая версия системы рецензирования докладов была разработана специалистами компании «РосИнтернет технологии» несколько лет назад, и с тех пор претерпела значительные изменения, продиктованные сферой применения. Система была проверена в ходе более 10 научных конференций, проведение которых значительно упростила и систематизировала. Среди них такие конференции, как «Системный анализ и информационные технологии» (2009, 2011, 2013), «Математическая теория систем» (2009), «Прикладные проблемы управления макросистемами» (2012, 2014), «Конференция по искусственному интеллекту» (2008, 2010, 2014), «Теория и практика системного анализа» (2010, 2012, 2014). Данная конференция ТПСА-2014 также проводится с использованием системы рецензирования докладов.

Уникальность данной системы состоит в том, что она достаточно полно учитывает специфику проведения российских конференций. Планируется расширение системы рецензирования для ее использования при подготовке выпусков научных журналов и сборников публикаций.

Литература

1. EasyChair <http://www.easychair.org>
2. Peerage of Science <http://www.peerageofscience.org>
3. EDAS <https://edas.info/index.php>
4. RAE Conference System <http://www.csrae.ru>
5. Bornmann L., Daniel H. D. The manuscript reviewing process: Empirical research on review requests, review sequences, and decision rules in peer review // Library & Information Science Research. 2010. Vol. 32. № 1. Pp. 5-12.
6. Papagelis M., Plexousakis D., Nikolaou P. N. CONFIOUS: Managing the electronic submission and reviewing process of scientific conferences // Web Information Systems Engineering-WISE 2005. – Springer Berlin Heidelberg, 2005. Pp. 711-720.

МОДЕЛЬ ФОРМЫ И СОДЕРЖАНИЯ ГРУППОВЫХ ПРЕДСТАВЛЕНИЙ ДОКУМЕНТОВ

Евсеев А.С.

Институт системного анализа РАН
г. Москва, andse-t610@yandex.ru

Аннотация

В работе рассматриваются вопросы представления и внутренней структуры наборов взаимосвязанных документов. Данные вопросы предлагается решать с помощью универсального представления инверсного входа. Рассмотрены частные варианты использования данного подхода.

1. Вопросы группового представления документов

Задача представления одиночного документа проработана в *теории документного интерфейса* [3]. Однако часто возникает задача представления набора взаимосвязанных документов.

Групповое представление документов – отображение набора взаимосвязанных документов.

В данной работе мы хотим исследовать и описать способ представления одного класса наборов документов в виде экранных форм.

В общем случае **набор документов** – объект, содержащий ссылки на документы, а также дополнительную информацию, описывающую взаимосвязи между ними.

Простейшим набором документов является коллекция документов. Количество документов в коллекции может быть велико, поэтому для удобства навигации по ней можно произвести *группировку* её элементов.

Группировка набора документов – разбиение множества документов на именованные подклассы на основании некоторого критерия.

Для более чёткого понимания рассматриваемой задачи введём понятие документа.

2. Понятие документа

Понятие документа не имеет общепринятого определения. В самом широком смысле документ – это любой материальный носитель, на котором так или иначе зафиксирована информация **с целью её передачи во времени и пространстве**.

В рамках работы мы принимаем следующее определение:

Документ – это структурированный текст [1].

Тексты составляются на естественном языке. Всякий текст несёт некоторую информацию об упоминаемых в нём объектах, их характеристиках, взаимоотношениях, то есть имеет определённые *смысл*.

С целью упрощения понимания *смысла* текста производится его *структурирование* – разбиение на законченные блоки, которые в некоторой степени независимы друг от друга. Например, текст этой работы разбит на подзаголовки, те в свою очередь разбиты на абзацы, которые разбиты на предложения и т.д.

Вычислительная техника не способна понимать тексты так, как это делает человек, поэтому для эффективной обработки информации требуется гораздо более глубокое структурирование, чем, например, разбиение на абзацы.

Для хранения объекта в информационной системе в нём выделяются значимые атрибуты. Такое описание объекта с явно выделенными атрибутами может быть представлено на языке XML [10], что удовлетворяет нашему определению документа (Рис. 1).

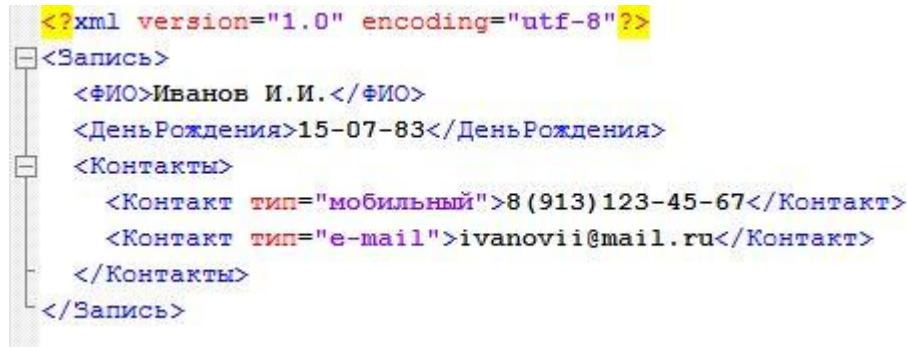


Рис. 1 – Документ, описывающий запись из записной книжки

Понятие документа порождает следующие составляющие его понятия: **содержание** и **представление** [1, 3] документа.

Содержание — информация, которую несёт документ [1].

Так или иначе содержание документа имеет некоторую структуру (ведь мы считаем, что документ это структурированный текст), для формального описания которой необходимо использовать одну из **моделей данных** (реляционную, иерархическую и т.д.). Описание *структур* документа в терминах выбранной модели данных - **модель содержания**.

Представление (форма) — способ отображения данных документа [1, 3].

Иными словами, *представление* есть визуальный образ документа. С целью возможности интерактивной работы документы часто представляют в виде экранных форм [5, 6]. Содержание документа может взимно-однозначно отображено в элементы управления экранной формы, что обеспечивает возможность интерактивной работы с документом при помощи формы [5]

Описание представления в рамках некоторого строго заданного набора инструментов — **модель представления**.

В качестве модели данных, в терминах которой описывается *модель содержания*, мы будем использовать модель данных СУБД xNiKa [4].

В этой модели данных структура документа описывается при помощи следующих блоков (см. Рис. 2) [2]:

- структура (полукруг);
- массив (прямоугольник);
- реквизит – неделимое данное (строка, число);
- ссылка.

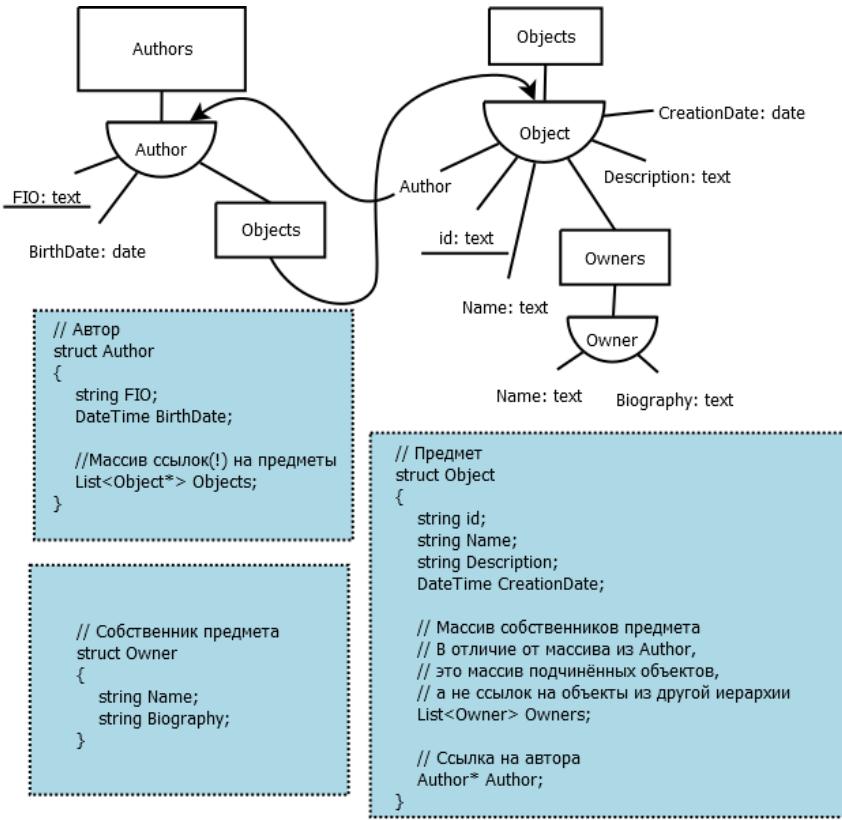


Рис. 2 – Модель данных xNika и соответствующий С-подобный псевдокод

На Рис. 2 приведены модели содержания двух взаимосвязанных документов: Author и Object. Ключём документа Author является реквизит FIO, а документа Object – id. Все документы **типа** Author содержатся в массиве Authors, а все документы **типа** Object – в массиве Objects. Такого рода *корневые* массивы будем называть **коллекциями**.

Коллекция – набор всех документов *одного типа*, упорядоченный по ключу.

3. Представление группы документов с помощью инверсного входа

В общем случае набор документов с группировками представляет из себя **инверсный вход**.

Инверсный вход – цепочка вложенных массивов, в которой:

- Каждый массив строится на основе группировки набора документов по определённому признаку. Значение признака становится ключом элемента массива.
- Последний массив содержит ссылки на документы.

Вообще говоря, на каждом уровне возможен сбор дополнительной информации, а критерий формирования уровня может быть отличным от проверки значения реквизита.

На Рис. 3.представлен набор документов «Указатель_МестоРождения». На основе данных реквизита «МестоРождения» документа «Анкета» с использованием внешней функции выделения страны, области и района, произведена группировка анкет по странам, затем по областям, затем по районам. Каждый уровень инверсного входа содержит ссылки на документы, что позволяет перейти к просмотру анкет, не доходя до нижнего уровня (например, можно получить все анкеты, относящиеся к определённой стране). Также инверсный вход позволяет получить информацию о группировках:

список всех стран с главами государств, для каждой страны – области с количеством жителей и т.д.

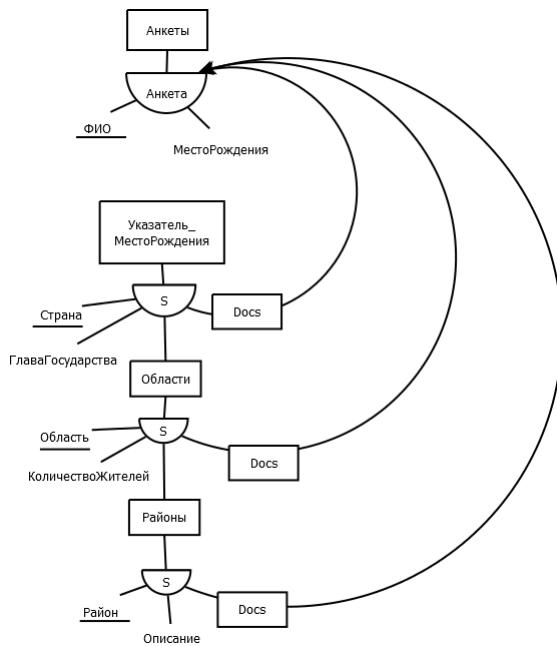


Рис. 3 – Группировка анкет по географическому признаку

Итак, в рамках данной работы *модель содержания* набора документов – структура *инверсного входа*. Существует и другие типы наборов документов, не сводимые к инверсному входу, но в рамках данной работы мы ограничимся указанным случаем.

В рассмотренной нами литературе под представлением набора документов подразумевается представление в виде различных графов отношений [7, 8]. Такое представление позволяет хорошо понять взаимосвязи документов, но, как нам кажется, усложняет получение информации о единичном документе набора и не всегда целесообразно. В данной работе мы сфокусируемся на более привычном способе представления – в виде экранных форм.

При отображении набора документов снова возникает задача представления единичного документа. Однако требования к представлению другие [9]:

- Уникальность – представление должно однозначно идентифицировать документ в наборе.
- Компактность – представление не должно занимать много места, чтобы можно было видеть как можно больше документов.
- Информативность – представление должно нести как можно больше информации о документе.
- Обоснование – иногда должна быть представлена информация, объясняющая, почему документ попал в данный набор/группировку.

Вопрос об общем алгоритме отображения единичного документа в наборе оставим открытым. Ограничимся лишь некоторыми рекомендациями:

- Желательно снабдить представление документа изображением описываемого им объекта.
- Не стоит выводить технические данные, не значимые для пользователя в данном контексте (например, не всегда нужно выводить инвентарный номер объекта).
- Должна быть явно выделена информация, идентифицирующая документ (например, название)
- Как правило, представление документа в наборе не предусматривает его редактирование.

При описании *модели представления* набора документов мы будем использовать следующие базовые элементы: таблица, список, дерево, простые элементы (текстовая метка, кнопка).

Опишем разработанный нами универсальный способ представления инверсного входа на примере инверсного входа с Рис. 3 (см. Рис. 4).

Инверсный вход подразумевает навигационный доступ – положение определяется уровнем в иерархии массивов и ключом элемента массива. Находясь на определённой позиции (например, в элементе массива «Страны» с ключом «Россия»), можно получить следующую информацию:

- список подчинённых группировок (список областей);
- список документов, относящихся к текущей группировке (список анкет, относящихся к России);
- дополнительную информацию группировки (сведения о главе государства).

В зависимости от ситуации, некоторые пункты списка могут отсутствовать.

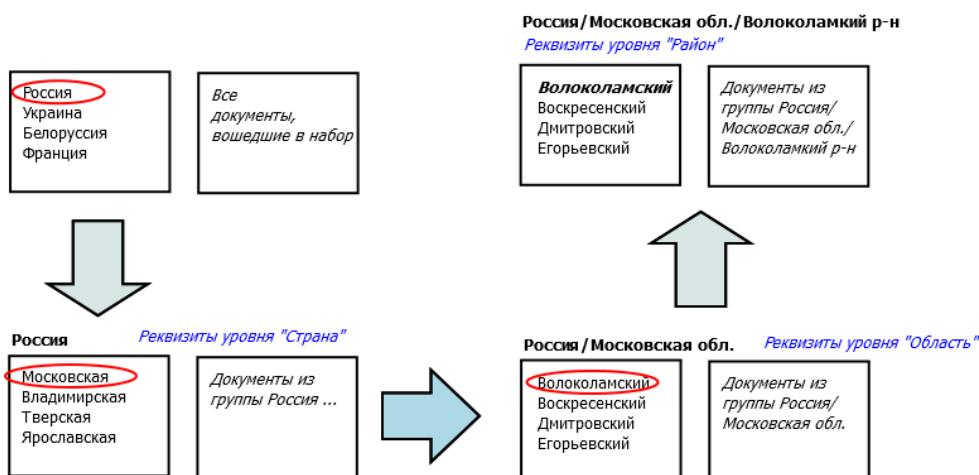


Рис. 4 – Универсальное представление инверсного входа

Универсальное представление (Рис. 4) состоит из следующих частей:

- Адресная строка - последовательность ключей пройденных элементов (например **«Россия/Московская обл.»**). Каждый ключ размещается на кнопке, активация которой приводит к переходу в соответствующее место инверсного входа.
- Реквизиты группировки – сведения, специфичные для текущей группировки (текст синего цвета на Рис. 4). Отображаются в виде текстовых меток.

- Список подчинённых группировок (левый прямоугольник на Рис. 4).
- Список документов, относящихся к текущей группировке (правый прямоугольник на Рис. 4).

В начальном состоянии отображается список корневых группировок (страны), а также список всех документов набора. При выборе определённой группировки (страны):

- название выбранной группировки добавляется к адресной строке;
- в списке подчинённых группировок отображаются группировки, подчинённые выбранной (области);
- в списке документов отображаются документы, относящиеся к выбранной группировке (анкеты из России).

Если группировка не содержит подчинённых группировок, при выборе элемента список подчинённых группировок не обновляется (лишь выделяется выбранный элемент) – обновляется лишь список документов.

Адресная строка позволяет двигаться в обратном направлении снизу вверх (возвращаться к пройденным узлам).

Если на каком-то уровне нет списка документов, соответствующих группировке, то список документов не отображается (например, если бы массив стран не содержал ссылок на анкеты, то при выборе «Россия» список документов стал бы невидим, так как в нём нечего отображать).

Для отображения списка подчинённых группировок и списка документов можно воспользоваться таблицей или списком, так как это одномерные массивы.

4. Частные случаи групповых представлений

Рассмотрим частные случаи групповых представлений.

Инверсный вход является *однородным* если все его уровни имеют одинаковую структуру.

В простейшем случае, каждая группировка должна иметь единственный реквизит – свой ключ. В этом случае вместо отображения списка подчинённых группировок можно отобразить сразу все группировки в виде дерева (Рис. 5). В общем случае группировки *однородного инверсного входа* могут быть отражены в виде дерева, соединённого с таблицей. При этом к дереву на Рис. 5 справа присоединяются дополнительные колонки, соответствующие реквизитам группировок.

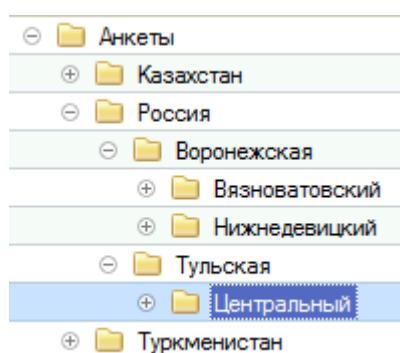


Рис. 5 – Дерево группировок

Если инверсный вход содержит ссылки на документы только на последнем уровне, а группировки не содержат никаких реквизитов, кроме ключа, то весь инверсный вход может быть отображён в виде дерева, соединённого с таблицей (Рис. 6).

ФИО	Год поступления	Дата рождения
Студенты		
Исторический		
И1-1		
Еремин Е.Ф.	01.09.2010	01.03.1993
Панин П.К.	01.09.2010	01.01.1993
И1-2		
Маврова Л.А.	01.09.2009	04.05.1993
Математический		
М1-1		
Иванов И.И.	01.09.2008	01.02.1991
Петров П.П.	01.09.2008	12.12.1990
М1-2		
Ивлева И.В.	01.09.2008	01.01.1991
Сидоров С.С.	01.09.2008	10.10.2008

Рис. 6 – Список студентов, сгруппированный по факультету и группе

5. Заключение

В работе рассмотрена задача представления *инверсного входа* в виде экранной формы.

В качестве базовых понятий были использованы понятия из *теории документного интерфейса*: *документ*, *модель содержания*, *модель представления*, а также модель данных СУБД xNika.

В качестве модели содержания выступает структура *инверсного входа*, также выделены и описаны базовые элементы, при помощи которых описывается *модель представления*. Приведён универсальный способ представления *инверсного входа*, даны рекомендации по представлению единичного документа в этом случае. Также исследованы некоторые частные случаи реализации в существующих информационных системах.

Литература

1. Емельянов Н.Е. Проблемы автоматизации документирования (Единый подход к автоматизации программирования обработки документов): Препринт/ВНИИСИ. - М., 1986 — С. 5 — 14.
2. Емельянов Н.Е., Щелкачева И.И. Проектирование баз данных в среде СУБД ИНЕС (методические материалы): Препринт/ВНИИСИ. - М., 1986 — С 13 — 26.
3. Емельянов Н.Е. Теоретический анализ документного интерфейса: Препринт. - М. : Всесоюзный научно-исследовательский институт системных исследований. —1987.
4. Богданов А. С. и др. НИКА-технология построения информационных систем //Организационное управление и искусственный интеллект/Сб. трудов ИСА РАН. Под ред. члена-корр. РАН Арлазарова ВЛ и д. т. н. проф. Емельянова НЕ-М.: Едиториал УРСС. – 2003. – С. 52-67.
5. Романов Б.Л. Представление структурированных информационных объектов в видеэлектронных форм // Управление информационными потоками: Сборник трудов ИСА РАН

6. Костенко К. И., Левицкий Б. Е., Назаренко С. В. Профессиональные электронные энциклопедии в системе VEDA. – 2000. - модель представления документа.
7. Carey M., Heesch D., Rüger S. Info Navigator: A visualization tool for document searching and browsing. – 2003.
8. Борисов В. Игнатенко А. Системы трехмерной визуализации данных <http://cgm.computergraphics.ru/content/view/115>
9. Olsen K. A. et al. Visualization of a document collection with implicit and explicit links //Scandinavian Journal of Information Systems. – 1993. – Т. 5. – С. 79-95.
10. XML (The Extensible Markup Language) - <http://www.w3.org/TR/xml/>

ПРИМЕРЫ ЭФФЕКТИВНОГО ИСПОЛЬЗОВАНИЯ ДВУХУРОВНЕВОЙ ДИНАМИЧЕСКОЙ МАРШРУТИЗАЦИИ В ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЯХ

Гринберг Яков Рувимович

Центр распределенных вычислений, Институт проблем передачи информации им.
Харкевича РАН, 117309, г. Москва,
greenjak@isa.ru

Курочкин Илья Ильич

Центр распределенных вычислений, Институт проблем передачи информации им.
Харкевича РАН, 117309, г. Москва,
qurochkin@gmail.com

Корх Артем Владимирович

Центр распределенных вычислений, Институт проблем передачи информации им.
Харкевича РАН, 117309, г. Москва,
artkorkh@gmail.com

Аннотация

Данная работа продолжает исследование задачи заполнения сетей потоками с помощью последовательных алгоритмов. Для уменьшения вычислительных затрат этих алгоритмов была введена парадигма двухуровневой маршрутизации, предполагающая выделение на графе телекоммуникационной сети специального подграфа - подграфа суперузлов. Маршрутизацией на первом уровне в сети будет построение пути для потока от источника до ближайшего суперузла, и от суперузла до стока. Второй уровень маршрутизации – построение пути для передачи потока в подграфе суперузлов, с помощью известных последовательных алгоритмов. Предыдущие работы по двухуровневой маршрутизации показали, что, несмотря на ускорение работы алгоритмов заполнения сети потоками, наблюдается существенное уменьшение величины суммарного потока, передаваемого через сеть. В этой работе будут приведены примеры топологий сетей, когда уменьшение величины суммарного потока, получаемого двухуровневыми алгоритмами будет несущественно.

Ключевые слова: последовательные алгоритмы, динамическая маршрутизация, двухуровневая маршрутизация, суперузлы.

1. Введение

Представленная работа является продолжением серии исследований алгоритмов последовательного заполнения сетей потоками от возникающих в них заявок [1][2], а также методов увеличения эффективности последовательных алгоритмов, посредством введения двухуровневой маршрутизации [3][4][5].

Напомним постановку задачи: телекоммуникационная сеть описывается неориентированным графом G , каждому ребру которого поставлено в соответствие натуральное число, называемое пропускной способностью, и выделены вершины, являющиеся источниками и стоками. В сеть последовательно во времени поступают заявки на прокладку пути между источником и стоком для единичного потока. Соответственно, основной проблемой маршрутизации является прокладка потока так, чтобы оставшийся ресурс сети был как можно больше – в этом случае можно добиться максимизации величины переданного через сеть суммарного потока. Для удовлетворения заявки необходимо найти для единичного потока путь из источника в сток. Если же такого пути не существует, то заявка сбрасывается. Заявки имеют конечное время жизни, то есть, если заявка была удовлетворена, то спустя некоторое время ее единичный поток исчезнет из сети, увеличив остаточные пропускные способности каналов связи.

Для решения задачи удовлетворения заявок были разработаны последовательные алгоритмы маршрутизации, работающие по принципу поиска пути минимальной стоимости, а именно, перед определением пути потока для каждого ребра графа сети вычисляется ее дополнительная характеристика – ее "стоимость" или "метрика". Разница между последовательными алгоритмами заключается в способах назначения этих "метрик". Алгоритмы, в которых "стоимость" ребра графа сети вычислялась с учетом ее принадлежности к минимальным разрезам между парами источников и стоков, образуют класс *минимально-разрезных последовательных алгоритмов*. Назначение "стоимости" ребер графа сети, учитывающее остаточную пропускную способность, дает класс *дуговых последовательных алгоритмов*. Дополнительно, был предложен *простой алгоритм*, который выбирает кратчайший путь между источником и стоком, без учета ресурса сети.

Данные алгоритмы разрабатывались с ориентацией на непрерывный рост объемов передаваемых данных в телекоммуникационных сетях (в частности, идет увеличение объемов мультимедийных данных), и решали задачу нахождения оптимальных каналов передачи. Однако при значительном увеличении размеров сетей эффективность работы последовательных алгоритмов падает, где под эффективностью следует понимать: величину суммарного потока, предаваемого через сеть при обработке всей последовательности заявок, поступающих в сеть (данное требование равнозначно увеличению числа удовлетворенных заявок); временные затраты на применение последовательных алгоритмов для расчета пути потока от поступившей заявки. В последнем случае увеличение эффективности алгоритмов предполагает уменьшение вычислительных затрат на них. Для этого используется декомпозиция последовательных алгоритмов, путем введения двухуровневой маршрутизации.

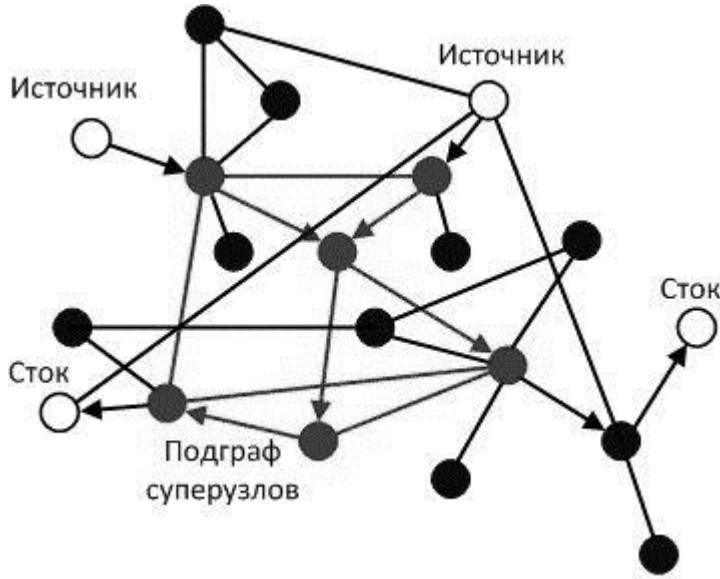


Рис. 1 – Схема двухуровневой маршрутизации в сети.

Для выделения суперузлов была предложена весовая характеристика (также называемой "разгруженностью"), как мера способности вершины графа сети принять и передать поток. По результатам работ [3][4] вид весовой характеристики был

$$\text{следующим: } U(p) = \lambda_1 * \left(\frac{\text{rescap}(p)}{\max_p(\text{rescap}(p))} \right) + \lambda_2 * \left(\frac{d(p)}{\max_p(d(p))} \right), \text{ где } \text{rescap}(p) - \text{сумма}$$

остаточных пропускных способностей ребер, инцидентных вершине p , $d(p)$ - степень рассматриваемой вершины, обе эти величины нормированы на максимальные значения этих величин, встречающиеся в сети. λ_1 и λ_2 - нормирующие коэффициенты, определяющие решающий вклад степени или остаточной пропускной способности вершины. Выделение суперузлов производилось с помощью метода агрегации данных сети [6][7][4][5], позволяющего каждой вершине в сети «самостоятельно» получать гистограмму зависимости числа вершин в сети от их величины их весовых характеристик и определять порог U_{Limit} вхождения вершины во множество суперузлов в зависимости от их требуемого количества.

В работах [5][8] были представлены результаты численных экспериментов по заполнению сетей с кластерной и стохастической топологией. Кратко опишем параметры проведенных экспериментов. В сетях с кластерной топологией количество вершин колебалось от 30 до 45, плотность заполнения сети ребрами колебалась от 10% до 20%. В сетях со стохастической топологией количество вершин колебалось от 30 до 45, плотность заполнения сети ребрами колебалась от 10% до 20%. В качестве исходных одноуровневых алгоритмов вычисления путей для потоков в сетях были использованы *простой* (*Simple*), *субоптимальный минимально-разрезный* (*EmpCut*) и *субоптимальный дуговой* (*EmpEdge*) алгоритмы. Таким образом, удалось охватить все перечисленные выше способы назначения "метрик" ребрам графов телекоммуникационных сетей. Анализ численных экспериментов преследовал две цели:

1. Сравнение времени работы одноуровневых алгоритмов маршрутизации и их двухуровневых модификаций.

2. Сравнение суммарного потока, проложенного через сеть за время работы имитационной модели для одно- и двухуровневых реализаций алгоритмов последовательного заполнения сети потоками последовательных заявок.

В таблице ниже представлены результаты применения двухуровневых последовательных алгоритмов маршрутизации и сравнение их результатов с одноуровневыми последовательными алгоритмами. Величина "доля потока" показывает какую долю $R = \frac{\sum F^2}{\sum F^1}$ от суммарного потока, получаемого одноуровневыми последовательными алгоритмами, прокладывают их двухуровневые модификации. Под суммарным потоком следует понимать все число всех удовлетворенных заявок, возникших в сети за время проведения численного эксперимента. Соответственно, доля потока, получаемого двухуровневыми алгоритмами, это отношения числа заявок, удовлетворенных двухуровневыми алгоритмами за все время проведения численного эксперимента, к суммарному числу заявок, удовлетворенных одноуровневыми последовательными алгоритмами, причем последовательность поступающих в сеть заявок в обоих случаях одинакова. "Ускорение" же обозначает уменьшение времени работы моделирования процесса поступления заявок в сеть и вычисления соответствующих им потоков.

Таблица 1

	Сети с кластерной топологией				Сети со стохастической топологией		
	Simple	EmpCut	EmpEdge		Simple	EmpCut	EmpEdge
Доля потока, % (среднее знач.)	92.51	91.90	91.92		90.42	88.73	88.70
Доля потока, % (дисперсия знач.)	4.29	4.09	4.09		3.81	3.54	3.43
Ускорение, раз (среднее знач.)	1.05	17.9	2.9		1.15	22.5	4.0
Ускорение, раз (дисперсия знач.)	0.04	9.9	0.5		0.09	8.8	1.3

Результаты численных экспериментов позволяют заключить, что несмотря на значительное ускорение маршрутизации, а именно, вычисления "стоимостей" ребер графа сети, наблюдается уменьшение, в среднем на 10%, величины суммарного потока. Тем не менее, в следующем разделе будут приведены примеры таких топологий сетей,

что уменьшение суммарного потока, получаемого двухуровневыми последовательными алгоритмами маршрутизации будет минимальным.

2. Организация двухуровневой маршрутизации

В данной работе рассматривалась еще один вид топологий телекоммуникационных сетей, который мы будем называть "древовидной" топологией. Данная топология предполагает существование некоторого "ядра", представляющего собой связный подграф вершин и "ветвей" - подграфов вершин, связанных с какой-либо вершиной из "ядра"; при этом ребра, соединяющие вершины различных "ветвей" либо отсутствуют, либо их число мало. Такой тип топологий может описывать автономные системы, где "ядро" представляет собой подсеть административных узлов, шлюзов и граничных маршрутизаторов различных автономных систем. На рисунке ниже представлен пример такой "древовидной" сети.

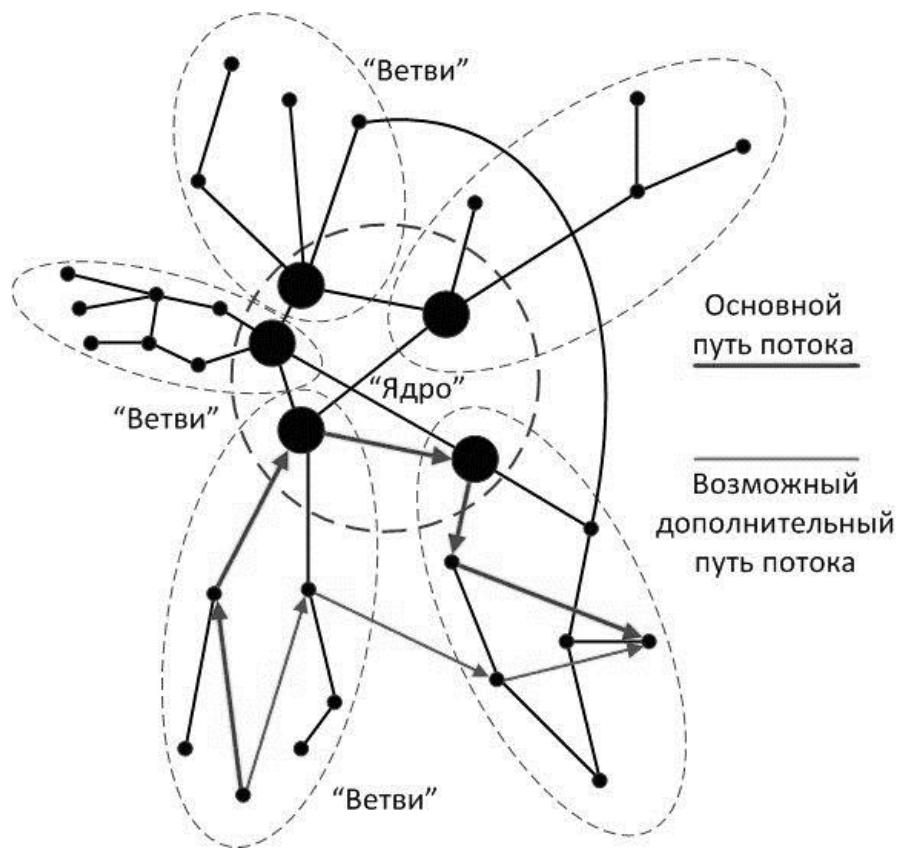


Рис. 2 – Пример сети с "древовидной" топологией.

Сети, работающие в рамках одной автономной системы, описываются упомянутыми "ветвями". Такие сети характеризуются очень большим числом вершин, при этом, следует предполагать, что существенная часть трафика проходит через узлы "ядра". Алгоритм построения графа таких сетей выглядит следующим образом (более подробное его описание можно прочитать здесь [9]):

1. В самом начале работы алгоритма генерации графа сети берется N вершин, которые произвольным образом располагаются на поверхности ограниченным размером $n \times m$.

2. Для каждой вершины определяется ее будущая степень (число инцидентных ей вершин). Данный момент является основным, поскольку позволяет заранее обозначить те вершины, которые будут представлять административные узлы автономных систем, промежуточные узлы передачи данных (маршрутизаторы) и обычные клиентские узлы.

3. Для каждой вершины определяется с какими вершинами она может быть соединена. Выбор инцидентных вершин производится с помощью некоторого априорного вероятностного закона или, например, пропорционально расстоянию между вершинами.

Существует модификация данного алгоритма, заключающаяся в следующем:

1. Для вершин со степенью больше двух строится оствное дерево.

2. Вершины со степенью один последовательно присоединяются к этому дереву.

3. Если остаются вершины, чья действительная степень меньше номинальной, то они попарно соединяются дополнительными ребрами.

Таким образом были сгенерированы четыре семейства сетей описание которых будет представлено ниже:

Таблица 2

Описание семейств сетей, использовавшихся в численном моделировании маршрутизации потоков, последовательно во времени возникающих в сети.

Название семейства	Число сетей в семействе	Основные свойства
stoh200	50	Сети с древовидной топологией. "Ядро" сети представлено графом со стохастической топологией. Число вершин равно 200. Число вершин центральной части равно 21. Плотность заполнения дугами равна 40% в центрально части, и 4% для периферийных вершин. Длина очереди заявок принимала значения от 32000 до 60150 единиц времени. Число пар корреспондирующих вершин (полюсов) равнялось 210. Время жизни заявок распределено по нормальному распределению в промежутке от 10 до 7500 единиц с математическим ожиданием в 200 единиц. Пропускная способность дуг сети центральной части равномерно распределена в диапазоне от 800 до 1000, пропускная способность периферийных дуг распределена от 200 до 600.
stoh100_10	60	Сети с древовидной топологией. "Ядро" сети представлено графом со стохастической топологией. Число вершин равно 100. Число вершин центральной части равно 10. Плотность заполнения дугами равняется от 40% до 70% в центрально части сети. Длина очереди заявок принимала значения от 46620 до 52380 заявок. Число пар корреспондирующих вершин (полюсов) равнялось 45. Время жизни заявок распределено поциальному распределению в промежутке от 10 до 1000 единиц с математическим ожиданием в 350 единиц. Пропускная способность дуг сети центральной части равномерно распределена в диапазоне от 100 до 300, пропускная способность периферийных дуг распределена от 1

		до 100.
stoh100_20	60	Сети с древовидной топологией. "Ядро" сети представлено графом со стохастической топологией. Число вершин равно 100. Число вершин центральной части равно 20. Плотность заполнения дугами равняется от 40% до 70% в центрально части сети. Длина очереди заявок принимала значения от 50160 до 68020 заявок. Число пар полюсов равнялось 190. Время жизни заявок распределено по нормальному распределению в промежутке от 10 до 1000 единиц с математическим ожиданием в 350 единиц. Пропускная способность дуг сети центральной части равномерно распределена в диапазоне от 100 до 300, пропускная способность периферийных дуг распределена от 1 до 100.
clust100_10	60	Сети с древовидной топологией. "Ядро" сети представлено графом с кластерной топологией. Число вершин равно 100. Число вершин центральной части равно 10. Плотность заполнения дугами равняется 20% в центрально части сети. Длина очереди заявок принимала значения от 43380 до 50040 заявок. Число пар корреспондирующих вершин (полюсов) равнялось 45. Время жизни заявок распределено поциальному распределению в промежутке от 10 до 1000 единиц с математическим ожиданием в 350 единиц. Пропускная способность дуг сети центральной части равномерно распределена в диапазоне от 100 до 300, пропускная способность периферийных дуг распределена от 1 до 100.

Перечислим еще раз типы численных экспериментов по заполнению сетей потоками:

1. Сравнение времени работы одноуровневых алгоритмов последовательного заполнения сети и их двухуровневых модификаций (**эксперимент 1**).

2. Сравнение суммарного количества потока, пропущенного через сеть за все время работы имитационной модели для одноуровневых и двухуровневых реализаций алгоритмов маршрутизации потоков продуктов последовательных заявок (**эксперимент 2**).

Несмотря на очевидный характер результатов в **эксперименте 1**, а именно ускорение работы алгоритмов последовательного заполнения на меньшем множестве вершин, представляет интерес зависимость ускорения работы алгоритмов маршрутизации от типа способа назначения "метрик" ребрам графа (простой, дуговой или минимально-разрезный).

Во **эксперименте 2** наиболее приемлемым считается такой результат, где разница между суммарным потоком, проложенным одноуровневым и двухуровневыми алгоритмами минимальна.

3. Результаты численных экспериментов

Приведем в общей таблице результаты по **эксперименту 1** и **эксперименту 2**.

Таблица 3

Результаты численных экспериментов.

	Сети семейства stoh100_10				Сети семейства stoh100_20		
	Simple	EmpCut	EmpEdge		Simple	EmpCut	EmpEdge
Доля потока, % (среднее знач.)	99.7	99.4	99.5		99.3	99.4	99.6
Доля потока, % (дисперсия знач.)	0.01	0.01	0.01		0.01	0.01	0.01
Ускорение, раз (среднее знач.)	1.5	5.2	2.9		1.25	3.1	1.8
Ускорение, раз (дисперсия знач.)	0.14	1.1	0.2		0.1	0.45	0.14
Сети семейства clust100_10			Сети семейства stoh200				
	Simple	EmpCut	EmpEdge		Simple	EmpCut	EmpEdge
Доля потока, % (среднее знач.)	99.5	99.4	99.5		95.9	96.0	96.1
Доля потока, % (дисперсия знач.)	0.01	0.01	0.01		0.03	0.02	0.03
Ускорение, раз (среднее знач.)	1.5	6.3	2.7		2.0	39.7	12.04
Ускорение, раз (дисперсия знач.)	0.1	1.3	0.2		0.01	9.38	1.03

Отметим, что ввиду большого размера сетей семейства **stoh200** и достаточно высокой плотности ребер в них, проведение численный экспериментов по заполнению их потоками продуктов от последовательно поступающих заявок с помощью одноуровневого последовательного субоптимального минимально-разрезного и субоптимального дугового алгоритмов оказалось затруднительным и весьма продолжительным по времени. Поэтому в приведенных выше результатах соответствующие результаты давались, исходя из аналогичных результатов простого алгоритма.

Приведем рисунки, иллюстрирующие ускорение работы алгоритмов маршрутизации в зависимости от плотности дуг в их "ядрах".

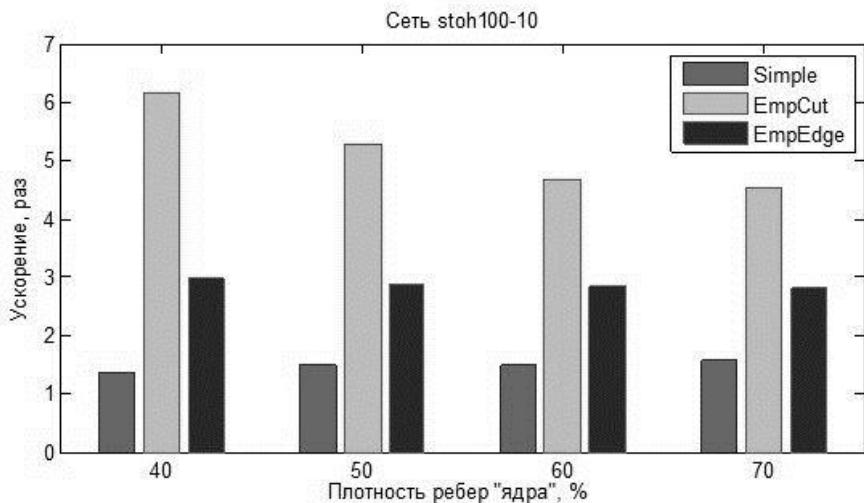


Рис. 3 – Ускорение работы двухуровневых последовательных алгоритмов маршрутизации на сетях семейства stoh100_10.

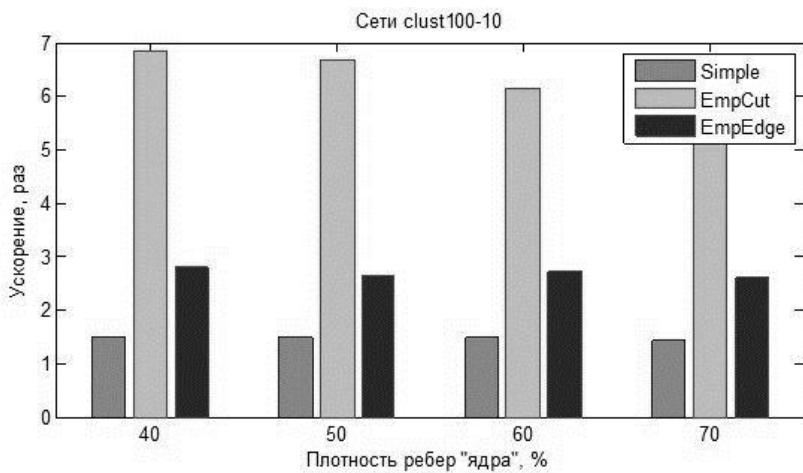


Рис. 4 – Ускорение работы двухуровневых последовательных алгоритмов маршрутизации на сетях семейства clust100_10.

Приведенные результаты отчетливо показывают, что использование двухуровневых последовательных алгоритмов маршрутизации на сетях с древовидной топологией позволяет получить суммарный поток, составляющий 95% - 99% от потока полученного аналогичными одноуровневыми последовательными алгоритмами. При этом, наибольшее ускорение наблюдается опять же у двухуровневых субоптимальных минимально-разрезных алгоритмов (от 3 до 6 раз). Наименьшее ускорение работы наблюдается у двухуровневого простого алгоритма (порядка 1.5 раз)

4. Заключение

В данной работе были разобраны случаи использования двухуровневой маршрутизации, когда не происходит значительного уменьшения суммарного потока, передаваемого через телекоммуникационную сеть. Успешные результаты достигаются на сетях с древовидной топологией, т.е. сетях, которые описываются графом с некоторым "ядром", представляющим собой связный подграф вершин и "ветвями" -

подграфами вершин, связанных с какой-либо вершиной из "ядра". В случае применения двухуровневых последовательных алгоритмов на таких сетях удается получить уменьшение суммарного потока не более чем на 5% от величины потока, получаемого одноуровневыми последовательными алгоритмами (простым, минимально-разрезными и дуговыми). При этом сохраняется выигрыш в ускорении работы алгоритмов маршрутизации.

Полученный инструментарий алгоритмов (одно- и двухуровневые последовательные алгоритмы) позволяет осуществлять выбор стратегии вычисления путей для потоков в телекоммуникационных сетях. Например, если в тестовом режиме выяснится, что величина суммарного потока, получаемого с помощью двухуровневых алгоритмов, учитывающих ресурс сети, равен или превышает суммарный поток простого одноуровневого алгоритма, то с большой вероятностью можно считать, что сеть имеет древовидную топологию, и наиболее целесообразно использовать двухуровневую парадигму маршрутизации. В противном случае и при сильной загруженности каналов связи можно сделать вывод, что использование одноуровневых последовательных алгоритмов предпочтительнее.

Литература

1. Гринберг Я.Р., Курочкин И.И. Математическое моделирование динамического последовательного заполнения сетей потоками связи / Проблемы вычислений в распределенной среде: Труды ИСА РАН // Под ред. С.В. Емельянова, А.П. Афанасьева – Т.46 - М.: КРАСАНД, 2009, с.233-258.
2. Гринберг Я.Р., Курочкин И.И. Анализ результатов численного эксперимента по последовательному заполнению сетей со стохастической топологией // Проблемы вычислений в распределенной среде: распределенные приложения, коммуникационные системы, математические модели и оптимизация: Сборник трудов ИСА РАН / Под ред. А.П. Афанасьева – Т.25 - М.: КомКнига, 2006, с.99-128.
3. Гринберг Я.Р., Курочкин И.И., Корх А.В. "Алгоритм кластеризации сетей передачи данных" // журнал "Информационные технологии и вычислительные системы", Т. 3, с. 18-30, 2012 г.
4. Гринберг Я.Р., Курочкин И.И., Корх А.В., Альгулиев Р.М., Гашимов М.А. "Применение метода агрегации данных в задаче нахождения узлов с дополнительной функциональностью" // журнал "Информационные технологии и вычислительные системы", Т. 2, 2013 -100 с. (с. 27-39)
5. Афанасьев А.П., Гринберг Я.Р., Курочкин И.И., Корх А.В. «Моделирование двухуровневой маршрутизации в задаче последовательного заполнения сети потоками продуктов» // Труды ИСА РАН. Т. 64, №4, 2013. (с. 26-35)
6. J. Sacha, J. Dowling, R. Cunningham, R. Meier. Using aggregation for adaptive superpeer discovery on the gradient topology // In Proceedings of 2nd IEEE International Workshop on Self-Managed Networks, Systems and Services, pages 77-90, 2006.
7. M. Jelasity, A. Montresor, and O. Babaoglu. Gossip-based aggregation in large dynamic networks // ACM Transactions on Computer Systems, vol. 23, p. 219-252, August 2005.
8. Курочкин И.И., Гринберг Я.Р., Корх А.В. "Результаты сравнительного анализа алгоритмов одноуровневой и двухуровневой динамической маршрутизации" // Пятая международная конференция «Системный анализ и информационные технологии» САИТ-2013 19-25 сентября 2013 г., г. Красноярск, Россия: Труды конференции. Т.2 – Красноярск: ИВМ СО РАН, 2013. –435 с. (с.344-349)
9. C. Jin, Q. Chen, S. Jamin. Inet: Internet Topology Generator // White paper for "Inet Generator" software, Department of EECS, University of Michigan, 2000.

ИСПОЛЬЗОВАНИЕ ВОЗМОЖНОСТЕЙ ПРОЦЕССОРОВ IA-32 ДЛЯ СОЗДАНИЯ ПРОСТОЙ ПЕРЕНОСИМОЙ РЕАЛИЗАЦИИ СИНХРОНИЗАЦИИ ПОТОКОВ В МНОГОПОТОЧНЫХ ПРИЛОЖЕНИЯХ НА ЯЗЫКАХ С И С++ ДЛЯ СИММЕТРИЧНЫХ МНОГОПРОЦЕССОРНЫХ АРХИТЕКТУР.

Альбертьян А.М.

Институт системного анализа РАН

117312, г. Москва, assa@isa.ru

Курочкин И.И.

Институт проблем передачи информации им. А.А. Харкевича РАН
127994, г. Москва, qurochkin@gmail.com

Аннотация

В работе рассматриваются особенности реализации библиотеки для синхронизации потоков в многопоточных приложениях. Рассмотрены предпосылки для создания библиотеки, а также обосновывается ее реализация для SMP-систем, которые получили распространение в настоящее время. Приводится сравнение с существующими аналогами и обсуждаются особенности измерения производительности многопоточных приложений. Указываются возможные сферы и методы применения предложенного решения.

1. Введение

Начиная с двухтысячных годов, появление технологий, подобных Intel® Hyper-Threading (HT), а также массовое распространение многоядерных процессоров, объединяющих несколько процессорных ядер в одном корпусе, привело к тому, что большая часть используемых на данный момент компьютерных систем является многопроцессорными. Большинство современных высокопроизводительных многопроцессорных компьютерных систем на аппаратном уровне представляют собой реализацию симметричной многопроцессорной архитектуры (SMP). SMP представляет собой архитектуру многопроцессорных компьютеров, в которой два или более идентичных процессора подключены к используемой совместно общей памяти, имеют полный доступ ко всем устройствам ввода-вывода и выполняют в системе одни и те же функции под управлением единой операционной системы. К SMP системам относится большая часть используемого на данный момент серверного оборудования, узлов вычислительных кластеров, высокопроизводительных рабочих станций, персональных компьютеров и т.д.

2. Методика

С точки зрения операционной системы процессоры в SMP системах являются равнозначными и независимо исполняют каждый свой поток инструкций. В отличие от классической однопоточной модели программирования, когда программа выполняет обработку данных последовательно, в соответствии с заданным алгоритмом, современное программное обеспечение может задействовать возможности

многопроцессорных систем, например, для оптимизации производительности, и создавать несколько одновременно исполняемых потоков обработки данных. Следовательно, исполняемые одновременно в определенный момент времени потоки могут принадлежать как различным, так и одному и тому же приложению, и работать, в том числе, с одним и тем же набором объектов (данных). В этом случае, для получения детерминированного результата работы приложения, необходимо обеспечение синхронизации работы создаваемых им потоков. Общий смысл синхронизации сводится к обеспечению правильного порядка взаимодействия нескольких одновременно исполняемых потоков команд с совместно используемыми объектами внутри приложения, модуля или библиотеки. Нарушение данной последовательности может привести к искажению критически важных данных и нарушению функционирования приложения, в том числе к появлению крайне трудно диагностируемых сбоев в процессе работы приложения на различном аппаратном обеспечении.

Рассматриваемая библиотека была реализована в ходе работы над расширением стандартной библиотеки шаблонов C++ (C++ STL) для высокопроизводительной работы с мульти множествами [7]. В результате перехода к многопоточной обработке, возникла необходимость в создании переносимого метода синхронизации потоков для многопоточных приложений. В данной библиотеке используются возможности процессоров с системой команд Intel Architecture 32-bit (IA-32) для реализации непосредственно в коде основной программы (*inline*) объектов синхронизации для многопоточных приложений.

Целью написания собственной реализации объектов синхронизации было создание производительного, простого и легко переносимого метода синхронизации потоков, который по возможности не зависел бы от используемой операционной системы и конкретных версий используемого компилятора, библиотек и т.п. А также создание более эффективных объектов синхронизации, с точки зрения приходящихся на каждое использование объекта системных вызовов и обеспечения максимальной непрерывности общего потока выполнения команд. Результатом этого является увеличение общей производительности многопоточного приложения по сравнению с использованием системных средств синхронизации или синхронизации средствами стандартных библиотек. Также возможно использование полученных в процессе работы над библиотекой примитивов для реализации прочих техник межпоточной синхронизации, таких например, как пользовательские блокировки (*spin-lock*), неблокирующие (*lock-free*) структуры данных и т.д.

3. Особенности реализации

Точное измерение производительности многопоточных приложений является нетривиальной задачей по причине сильного влияния на скорость работы многопоточного приложения таких факторов, как контекст самой задачи – количество одновременно исполняющихся потоков, интенсивность обращений к защищенному объектом синхронизации ресурсу и т.п., характеристик целевого оборудования – количество ядер (процессоров) в системе, их производительности, используемой операционной системы и пр. Не говоря уже о том, что на производительность полученного исполняемого кода очень сильно влияет то, какой компилятор использовался и какие опции оптимизации были заданы. По относительным оценкам, в сравнении, например, со стандартным объектом синхронизации ОС Windows критическая секция (*critical section*), при использовании компилятора OpenWatcom 1.9 и

количество потоков равному числу потоков исполнения используемого процессора Intel Xeon e5-1650 (12 потоков), при интенсивности обращений к объекту каждым потоком около 1000 раз в секунду, выигрыш в среднем времени ожидания освобождения объекта составляет около 3 раз. При этом разброс значений времени ожидания меньше, а максимальная задержка уменьшается примерно в 4 раза. Также продемонстрированное время освобождения объекта синхронизации, которое значительно менее подвержено изменению под влиянием вышеуказанных факторов, меньше времени освобождения системного объекта не менее, чем в 3-4 раза.

Для реализации стандартных объектов синхронизации, таких как семафор (semaphore), мьютекс (mutex) или критическая секция (critical section) преимущественно используется, при наличии поддержки со стороны используемых средств разработки (в первую очередь компилятора), встраиваемый (inline) ассемблер для архитектуры IA-32. Для полноценного использования данной реализации, на текущий момент, требуется использование совместимого компилятора языка C или C++ (Microsoft Visual C/C++, Intel C/C++, Open Watcom C/C++, GNU C/C++, Borland C/C++, Symantec/Digital Mars C/C++ и т.д.) на поддерживающей платформе (Microsoft Windows NT, OS/2, различные версии Unix, Linux, Mac OS X, QNX и т.д.)

В этом случае для обеспечения синхронизации потоков многопоточного приложения используется эффективная реализация, использующая непосредственно команды процессоров с архитектурой IA-32, такие как lock xadd, xchg, lock cmpxchg, lock cmpxchg8b, pause и т.п., для выполнения атомарных операций над счетчиками, идентификатором текущего потока и выполнения циклов ожидания (spin-wait loops). При использовании библиотеки на не полностью поддерживающей платформе, производится попытка обеспечения максимальной функциональности объектов синхронизации с подробной диагностикой необходимой доработки и/или изменения конфигурации используемых средств разработки для обеспечения работоспособности проекта.

4. Заключение

Предложенное решение может быть использовано в проектах, использующих принципы структурного или объектно-ориентированного программирования (ООП) на языках С/С++ для компьютеров на базе процессоров с архитектурой IA-32 для создания эффективных многопоточных приложений для современных операционных систем.

Литература

1. Аммераль Л. STL для программистов на С++. – М.: ДМК, 1999. – 240 с.
2. Джосьюти Н. С++. Стандартная библиотека. – СПб.: Питер, 2003. – 736 с.
3. Вандевурд Д., Джосаттис Н.М. Шаблоны С++: справочник разработчика. – М.: Вильямс, 2003. – 544 с.
4. Страуструп Б. Язык программирования С++. Специальное издание. Пер. с англ. – М.: Издательство Бином, 2011. – 1136 с.
5. ISO/IEC 14882. Programming languages – С++. – ISO, 1998. – 776 р.
6. ISO/IEC 9899. Programming languages – С. – Second edition. – ISO, 1999. – 554 р.
7. Петровский А.Б. Пространства множеств и мультимножеств. – М.: Едиториал УРСС, 2003. – 248 с.
8. Руссинович М., Солomon Д. Внутреннее устройство Microsoft Windows: Windows Server 2003, Windows XP и Windows 2000. Мастер-класс. / Пер. с англ. – 4-е изд. – М.: Издательство «Русская Редакция»; СПб.: Питер, 2008. – 992 с.
9. Танненбаум Э. Современные операционные системы. – 3-е изд. – СПб.: Питер, 2010. – 1120 с.

10. Intel® 64 and IA-32 Architectures Software Developer's Manual. Volume 2 (2A, 2B & 2C): Instruction Set Reference, A-Z. – Intel, 2014. – 1479 p.
11. Intel® 64 and IA-32 Architectures Software Developer's Manual. Volume 3 (3A, 3B & 3C): System Programming Guide. – Intel, 2014. – 1446 p.
12. Intel® 64 and IA-32 Architectures Optimization Reference Manual. – Intel, 2014. – 670 p.
13. MSDN Library. System Services. Synchronization. – Microsoft, 2014. <http://msdn.microsoft.com>

ON TILING FOR HETEROGENEOUS SYSTEMS DURING MECHANICAL CODE PARALLELIZATION

Lebedev A.

Rybinsk State Aviation Technical University, Rybinsk, Russia
tementy@gmail.com

Abstract

A heuristic approach addressing workload distribution problems during mechanical code parallelization for heterogeneous systems is discussed. A pair of multicore CPU and CUDA-capable GPU is considered as target compute platform. The goal of minimizing overall compute time is achieved via greedy scheme of minimizing time cost for every time step through dynamic load balancing between CPU cores and GPU accelerator. Transformed for heterogeneous execution LU-decomposition kernel illustrates significant speedup due to offload acceleration of its parts.

1. Introduction

Large number of processing elements per one chip is a common feature of modern microarchitectures. Many of them are represented by highly-parallel processors: graphics processing units (GPUs), many integrated core (Intel MIC) and universal multicore processors (CPUs). Writing an efficient code to expose compute capabilities of parallel processor often leads to significant challenges. The task become more complex when targeting heterogeneous computing systems. A very attractive approach addressing this issue is automatic parallelization (ideally, without any programmers effort).

Many compute-intensive applications often spend most of their execution time in nested loops. This is particularly common in scientific and engineering applications. The polyhedral model provides a powerful abstraction to reason about transformations on such loop nests consigned for locality and parallelization optimizations.

Polyhedron model methods are used commonly in static compilation scheme [1, 2, 3, 4, 5]: the code written in high-level language like C or FORTRAN is optimized once (for data locality, parallel execution) and runs on target architecture. The main idea of polyhedral transformations is to reorder operations of the source program (transform the iteration space) in order to maximize parallelism alongside with cache usage improving which makes the program run faster.

All the transformations in polyhedron model are done after dependence analysis. It is reasonable to highlight two major phases: space-time mapping and tiling. The first one is for computing two functions for every dependence-related operation: schedule – to designate the logical order of computations defined by data dependencies, and placement – to map every logical operation to particular virtual processor. Tiling process follows after space-time

mapping for further partitioning of target index space: it is necessary to specify for physical processors which points of virtual processor space they should be mapped to.

Lots of tiling techniques are intended for uniform physical processor space trying to minimize computation time via reducing communication costs, but there are only few ones designed specifically for heterogeneous systems with accelerators. Recent works [6, 7] discuss only static approach to determine optimal workload distribution based on reducing communication volume. Such problem is worth attention, because there are algorithms that could be efficiently parallelized for heterogeneous execution directly in tiling domain.

We start from polyhedron model overview and continue with LU-decomposition example as an illustrative sketch showing possibility of dynamic load balancing between CPU cores and GPU accelerator.

2. Polyhedral framework basics

The polytope model [8] is a computational model for sequential or parallel programs: a program is modelled by a polytope, a finite convex set of some dimensionality with flat surfaces. The data dependencies between the points (computations) in the polytope must be regular and local. Dependences are determined precisely through dataflow analysis, but we consider all dependences including anti (write-after-read), output (write-after-write) and input (read-after-read) dependences. These properties required of the model (convexity, flat surfaces, regularity of the dependences) restrict the set of programs that can be modeled (linear programs). A set of, say, r perfectly nested loops with a constant-time loop statement, with bounds that are linear expressions in the indexes of the enclosing loops and in the problem size, and with certain additional restrictions on the use of the loop indexes can be represented by a polytope embedded in Z^r (each loop defines the extent of the polytope in one dimension).

The Data Dependence Graph (DDG) is a directed multi-graph with each vertex representing a statement, and an edge, $e \in E$, from node S_i to S_j representing a polyhedral dependence from a dynamic instance of S_i to one of S_j : it is characterized by a polyhedron, P_e , called the dependence polyhedron that captures the exact dependence information corresponding to e . The dependence polyhedron is in the sum of the dimensionalities of the source and target statement's polyhedra (with dimensions for program parameters as well). Let s represent the source iteration and t be the target iteration pertaining to a dependence edge e . It is possible to express the source iteration as an affine function of the target iteration, i.e., to find the last conflicting access. This affine function is also known as the h -transformation, and will be represented by h_e for a dependence edge e . Hence, $s = h_e(t)$. The equalities corresponding to the h -transformation are a part of the dependence polyhedron and can be used to reduce its dimensionality. Let S_1, S_2, \dots, S_n be the statements of the program. A one-dimensional affine transform for statement S_k is defined by:

$$\varphi_{S_k}(i) = [c_1, \dots, c_{m_{S_k}}](i) + c_0 \quad (1)$$

φ_{S_k} can also be called an affine hyperplane, or a scattering function when dealing with the code generator. A multi-dimensional affine transformation for a statement is represented by a matrix with each row being an affine hyperplane.

When modelling sequential execution, the dimensions are scanned, one at a time, to enumerate the points in the polytope. For a parallel execution, the polytope that represents the source program – we call it the source polytope – is segmented into time slices, sets of points that can be executed concurrently. The parallelization methods based on the polytope model address the problem of this segmentation. It can be formulated as an affine mapping that

transforms the source polytope into a target polytope that contains the same points, but in a new coordinate system in which some dimensions are strictly temporal, i.e., scanning along them enumerates time, and the others are strictly spatial, i.e., scanning along them enumerates space.

A schedule θ of a program is a function which maps every operation to an integer vector that represents logical time. For every statement S , its schedule θ_S can be represented by a $t \times d$ matrix Θ_S where t is the number of time dimensions, and d is the dimensionality of the index set of S plus the number of symbolic parameters. Affine schedule:

$$\theta_S(i, n) = \Theta_S \begin{pmatrix} i \\ n \end{pmatrix} \quad (2)$$

Analogously to the schedule, a computation placement π is a function which maps every operation to an integer vector that represents a virtual processor. With the same affinity constraints, the placement of a statement S , π_S , can be represented by a $p \times d$ matrix Π_S , where p is the number of processor dimensions, and d is as above. Computational placement:

$$\pi_S(i, n) = \Pi_S \begin{pmatrix} i \\ n \end{pmatrix} \quad (3)$$

When switching from the model back to the code level, we must scan the target (i.e., space-time mapped) index sets, i.e., we must enumerate all points inside. For this purpose, we again want to represent the target index sets as polyhedra.

3. LU-decomposition analysis

Consider two variants of simple LU-decomposition kernel: naive parallelization with OpenMP and optimization for reuse distance with modern PLUTO [4] algorithm:

C version	PLUTO, C version without tiling
<pre> for (int k = 0; k < N; k++) { for (int l = k + 1; l < N; l++) A[l * N + k] /= A[k * N + k]; for (int i = k + 1; i < N; i++) for (int j = k + 1; j < N; j++) A[i * N + j] -= A[i * N + k] * A[k * N + j]; } </pre>	<pre> int t1, t2, t3; int lbp, ubp; if (N >= 2) { for (t1 = 1; t1 <= 2 * N - 3; t1++) { lbp=ceild(t1 + 1, 2); ubp=min(t1, N - 1); </pre>
Naive OpenMP C version	<pre> #pragma omp parallel for private(t2,t3) for (t2 = lbp; t2 <= ubp; t2++) { a[t2][t1 - t2] /= a[t1 - t2][t1 - t2]; for (t3 = t1 - t2 + 1; t3 <= N - 1; t3++) { a[t2][t3] -= a[t2][t1 - t2] * a[t1 - t2][t3]; } } </pre>
<pre> for (int k = 0; k < N; k++) { #pragma omp parallel for for (int l = k + 1; l < N; l++) A[l * N + k] /= A[k * N + k]; #pragma omp parallel for for (int i = k + 1; i < N; i++) for (int j = k + 1; j < N; j++) A[i * N + j] -= A[i * N + k] * A[k * N + j]; } </pre>	

As we stated earlier [9], naive OpenMP version outperforms generated by PLUTO on Intel Xeon CPU and Intel Xeon Phi. This is because PLUTO minimizes data reuse distance such that every time step (iteration of t_1 -loop) is very cache efficient, while most of them haven't enough parallelism which causes hardware stalling. Figure 1 illustrates amount of computations for every time step with $N = 16$. «Operation» stands for t_3 -loop body execution, «line» stands for whole t_3 -loop execution with arbitrary bounds.

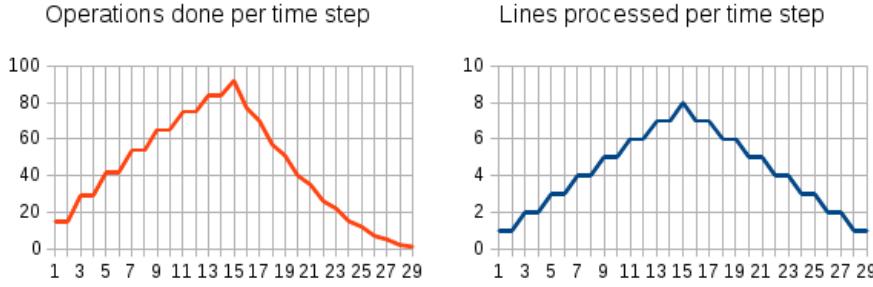


Fig. 1 – Uneven workload distribution w.r.t. time steps for PLUTO with $N = 16$

On the contrary, naive OpenMP version provides fine-grained parallelism with $i-j$ loop nest. Let k and j iterate through time dimensions, l and i iterate through space dimension. A valid placement function for all operations is one which gives row number of processed matrix element. This is Forward-Communication-Only placement, which guarantees freedom of tiling [10].

$i-j$ loop nest is the region of interest. It spends most of time while executing time step k . Since placement is FCO, we can freely map any virtual processor to any physical processor here. Physical processor now stands not only for CPU core, but for GPU-accelerator as well. A portion of computations (say, α of $N-k-1$ matrix rows) should be done by accelerator, if reasonable. We also should leave one CPU core for GPU management host thread. Rest $N-k-1-\alpha$ matrix rows should be processed by free CPU cores.

Let $T_{GPU} = dep_{cost}^{input} + \alpha \cdot row_{cost}^{GPU}$ be the approximated GPU execution time depending linearly from α , where dep_{cost}^{input} stands for input dependencies communication overhead and row_{cost}^{GPU} is average row processing cost including communication and computation. All the parameters we suggest to estimate empirically.

Let $T_{CPU} = \frac{N - k - 1 - \alpha - probed}{NP - 1} \cdot row_{cost}^{CPU}$ be the approximated CPU execution time

depending linearly from α , where NP is the number of CPU cores, row_{cost}^{CPU} is average row processing cost (to be estimated empirically), $probed$ is the number of rows processed while estimating time parameters.

Total execution time we are going to minimize is $T = \max(T_{GPU}, T_{CPU})$. The minimum is determined by intersection point of lines defined by approximations. It means that we look for the point α for which $T_{GPU} = T_{CPU}$ holds. Usage of the accelerator is reasonable if and only if $\alpha > 1$.

$$\alpha = \frac{(N - k - 1 - probed) \cdot row_{cost}^{CPU} - (NP - 1) \cdot dep_{cost}^{input}}{row_{cost}^{GPU} (NP - 1) + row_{cost}^{CPU}} \quad (4)$$

4. Experiments

We use server with 6-core Intel Xeon X5650 and NVidia Tesla M2050. Matrix to be factorized is square matrix of order 2^{15} of doubles. We decided to estimate time parameters every 100 iterations of k -loop and calculate alpha respectively. For last iterations of k -loop (which induce rows width smaller than 2048) we reject accelerator resources and stay only with CPU cores – small amounts of work are well handled with CPU cores and usage of the accelerator gives imperceptible speedup.

We collected values of alpha during run time and tried to approximate the regression with quadratic polynomial: $-3.1 \cdot 10^{-7} \cdot k^2 - 0.0027643 \cdot k + 48.92687353$. Figure 2 illustrates percentage of computations offloaded to accelerator for every time step.

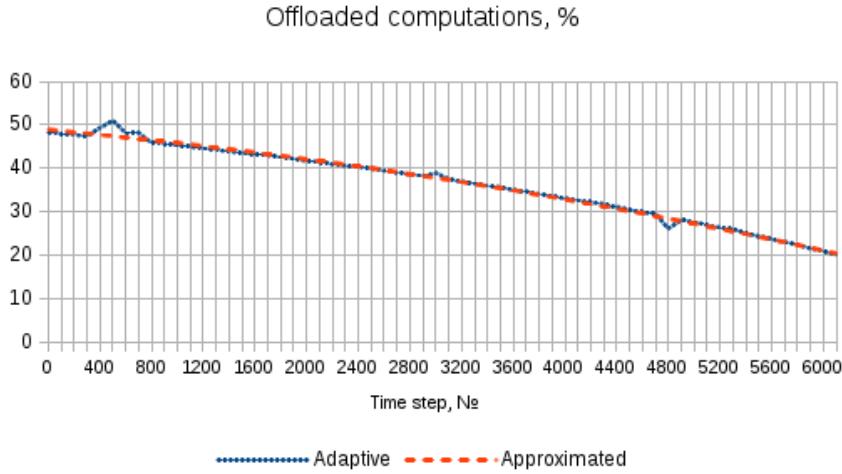


Fig. 2 – Percentage of offloaded computations for every time step

Table 1 shows time (given in seconds) of LU-kernel execution for adaptive, approximated and CPU-only case. As we can see, adaptive parallelization outperforms other solutions. Surprisingly, but approximated solution without runtime estimations works slightly slower than adaptive one. Consequently, adaptive solution is preferable in case of lightweight runtime estimations.

Table 1

Results of LU-kernel benchmarks

CPU + GPU, adaptive	CPU + GPU, approximated	CPU, 6 threads
271	291	396

5. Conclusion

A heuristic mentioned here could be generalized in the following way: choose space-time mapping uniformly exposing fine-grained parallelism w.r.t. most of time steps (if possible), then determine representative benchmark tile size and estimate time parameters in order to compute α value for every time step. The code generation phase [11] should be followed by instrumentation: runtime estimations and dynamic workload distribution should be injected into target program.

References

1. Irigoin, François, Pierre Jouvelot, and Rémi Triolet. Semantical interprocedural parallelization: An overview of the PIPS project. Proceedings of the 5th international conference on Supercomputing. ACM, 1991.
2. Lee, Sang-Ik, Troy A. Johnson, and Rudolf Eigenmann. Cetus – an extensible compiler infrastructure for source-to-source transformation. Languages and Compilers for Parallel Computing. Springer Berlin Heidelberg, 2004. 539-553.

3. Griebl, Martin, and Christian Lengauer. The loop parallelizer LooPo. Proc. Sixth Workshop on Compilers for Parallel Computers. Vol. 21. Konferenzen des Forschungszentrums Jülich, 1996.
4. Bondhugula, Uday, et al. Automatic transformations for communication-minimized parallelization and locality optimization in the polyhedral model. Compiler Construction. Springer Berlin Heidelberg, 2008.
5. Grosser, Tobias, et al. Polly-Polyhedral optimization in LLVM. Proceedings of the First International Workshop on Polyhedral Compilation Techniques (IMPACT). Vol. 2011. 2011.
6. Dathathri, Roshan, et al. Generating efficient data movement code for heterogeneous architectures with distributed-memory. Parallel Architectures and Compilation Techniques (PACT), 2013 22nd International Conference on. IEEE, 2013.
7. U. Bondhugula, Automatic distributed memory code generation using the polyhedral framework. Indian Institute of Science, Tech. Rep. IISc-CSA-TR-2011-3, 2011.
8. Christian Lengauer. Loop Parallelization in the Polytope Model. In Eike Best, editor, CONCUR'93, number 715 in Lecture Notes in Computer Science, pages 398–416. Springer-Verlag, 1993.
9. Lebedev A., Khachumov V. Offload acceleration of scientific calculations within .NET assemblies. Proceedings of the Third International Conference on High Performance Computing. National Technical University of Ukraine «Kyiv Polytechnic Institute», 2013.
10. Martin Griebl. Automatic Parallelization of Loop Programs for Distributed Memory Architectures. habilitation thesis, Department of Mathematics and Informatics, University of Passau, 2004.
11. Bastoul, Cedric. Code generation in the polyhedral model is easier than you think. Proceedings of the 13th International Conference on Parallel Architectures and Compilation Techniques. IEEE Computer Society, 2004.

МНОГОКРИТЕРИАЛЬНАЯ ОПТИМИЗАЦИЯ В МУЛЬТИАГЕНТНОЙ СИСТЕМЕ

Малтугуева Г.С.

Институт динамики систем и теории управления СО РАН
г. Иркутск, gama@icc.ru
Юрин А.Ю.

Институт динамики систем и теории управления СО РАН
г. Иркутск, iskander@icc.ru

Аннотация

В работе исследован вопрос применимости методов многокритериальной оптимизации в мультиагентной среде для получения согласованных (компромиссных) решений. В соответствии со своими ролями и компетенциями интеллектуальные агенты способны выполнять определенные действия (операции), так агенты-эксперты оценивают альтернативы в пределах своей компетенции, а на основе полученных оценок агент-координатор решает задачу выбора. Предложен новый метод принятия решений при многих критериях, основанный на преобразовании количественно заданных оценок в обобщенные ранжировки и применении нескольких принципов согласования в составе КПП-метода.

1. Введение

Современный уровень развития исследований, осуществляемых с помощью агентов в различных предметных областях, обуславливает необходимость применения специализированных математических методов для повышения эффективности их взаимодействия. Каждый агент в системе может выполнять определенные задачи,

соответствующие его роли, при этом в целом мультиагентная система ориентирована на достижение общей цели.

Рассмотрим мультиагентную систему (МАС), состоящую из агента-руководителя, агента-координатора (диспетчера) и агентов-экспертов (у каждого единственная компетенция). Руководитель задает цель исследования, определяет ограничения на возможные варианты действий, и передает управляющую информацию координатору, который распределяет задания между экспертами согласно их компетенций. Эксперты, оценив все возможные альтернативы, передают свои оценки координатору. На основании полученных оценок и с учетом ограничений координатор формирует результат и предоставляет его руководителю. Руководитель, оценив соответствие полученного результата входным данным, принимает решение (одобрение или доработка результата).

В настоящее время для бесконфликтного согласования интересов (сведений) в МАС применяются следующие методы – переговоров, уступок, эволюционного согласования. Все они носят итеративный характер и предусматривают постоянный обмен сообщениями между всеми агентами. Соответственно, процесс согласования может оказаться длительным. Исследуем вопрос применимости процедур голосования в качестве метода многокритериальной оптимизации для согласования информации в мультиагентной системе.

2. Методы многокритериальной оптимизации

В настоящий момент в теории многокритериальной оптимизации разработаны разнообразные методы принятия решений при многих критериях [1-6], приведем основные понятия из данной области.

Альтернатива – вариант действия. Для принятия решения необходимо иметь в рассмотрении не менее двух альтернатив. Альтернативы могут быть как зависимыми, так и независимыми. Они могут быть заданы при постановке задачи (априори), а могут появляться в процессе принятия решения. Множество альтернатив обозначим через $A = \{A_1, A_2, \dots, A_n\}$, где n – число альтернатив [1-6].

Критерий – некоторая выделенная особенность, характеризующая альтернативы (признак, свойство, атрибут). Критерии должны быть заданы априори, их количество определяется участниками процесса принятия решений. По числу критериев принято различать однокритериальный и многокритериальный выбор. Каждому критерию соответствует собственная шкала измерений (шкала наименований, ранговая, интервальная, шкала отношений), согласно которой оценивается каждая из альтернатив [1-6].

Множество Парето – совокупность альтернатив, не уступающих друг другу одновременно по всем критериальным оценкам. Подобные альтернативы ещё принято называть несравнимыми. Таким образом, особая роль множества Парето заключается в том, что решение, лучшее с точки зрения лица, принимающего решение, обычно видится в множестве Парето-оптимальных [1-6].

Существует множество методов многокритериальной оптимизации, позволяющих обрабатывать разнородную информацию, например: метод анализа иерархий, метод АРАМИС, метод МАСКА, метод уступок, методы свертки. Методы свертки не всегда однозначны, т.к. требуют назначения коэффициентов. В работах [7-8] ведется дискуссия о корректности и метода анализа иерархий, методы АРАМИС и МАСКА предназначены для оценки качественно-количественных оценок [3,9]. Наиболее распространены

методами бесконфликтного согласования интересов, применяемыми в МАС, являются методы переговоров и уступок, которые с помощью последовательного ознакомления агентов с оценками других агентов приводят к компромиссному решению. Однако вопросы размера устанавливаемых уступок и изменений в переговорном множестве вызывают дискуссии и их обсуждение может занять много времени. Кроме того, применение данных методов требует изменения критериальных оценок и постоянного обмена сообщениями, что накладывает определенные требования на каналы связи между агентами.

3. Принципы голосования в мультиагентной системе

Рассмотрим мультиагентную систему, агентов в которой можно разделить согласно их ролям и решаемым задачам (таблица 1).

Таблица 1

Роли и задачи агентов в МАС.

Роль	Задачи
Руководитель	постановка цели
	формирование ограничений на возможные варианты, значения по некоторым критериям
	установление приоритетов для критериев
	принятие окончательного решения
Координатор	формирование множества альтернатив
	информирование экспертов и руководителя
	согласование информации от экспертов
Эксперт	оценивание альтернатив в рамках своей компетенции (критерия)

Цель деятельности МАС определяется агентом-руководителем и окончательное решение также принимает он. Однако для оценки сложных объектов требуется привлечение агентов-экспертов, обладающих специализированными знаниями в определенной предметной области (компетенция). Для обеспечения эффективности взаимодействия (избежание и разрешение конфликтов, сокращение количества взаимодействий с руководителем) в системе присутствует агент-координатор, который осуществляет сбор и обработку информации, поступающей от агентов-экспертов. Последние, в свою очередь, согласно своей компетенции способны оценить все возможные альтернативы. В большинстве случаев результат оценивания представляет собой количественную оценку определенного параметра альтернатив [10-11].

Для обработки количественной информации на практике широко используются методы свертки (аддитивной, мультипликативной), главного критерия, однако зачастую их адекватность и обоснованность вызывают сомнение. Поэтому предложено построить обобщенные ранжировки на основе количественных оценок, предоставленных агентами-экспертами.

Обобщенная ранжировка – упорядочение всех рассматриваемых альтернатив с указанием равнозначности: $R_j = A_{j,i_1}Q_{i_1}^j \dots Q_{i_{n-1}}^j A_{j,i_n}$, где $j = \overline{1, m}$, $Q_i^j \in \{\succ, \approx\}$, \succ, \approx – отношения строгого предпочтения и эквивалентности соответственно. Каждой ранжировке соответствует некоторый элемент из \mathfrak{R}_n ($R_j \in \mathfrak{R}_n$), где \mathfrak{R}_n – множество всех возможных

упорядочений n альтернатив, например $\mathfrak{R}_1 = \{A_1\}$, $\mathfrak{R}_2 = \{A_1 \succ A_2, A_1 \approx A_2, A_2 \succ A_1\}$, $\mathfrak{R}_3 = \{A_1 \succ A_2 \succ A_3, A_1 \succ A_2 \approx A_3, A_1 \succ A_3 \succ A_2, A_1 \succ A_3 \approx A_2, A_2 \succ A_1 \approx A_3, A_2 \succ A_3 \succ A_1, A_3 \succ A_1 \succ A_2, A_3 \succ A_1 \approx A_2, A_3 \succ A_2 \succ A_1, A_3 \succ A_2 \approx A_1, A_1 \approx A_2 \approx A_3\}$ и т.д. [12-13]

В качестве способа обработки обобщенных ранжировок, выражающих оценки всех альтернатив, предлагается использовать методы построение отношения группового предпочтения, в частности КПП-метод, основанный на последовательном применении различных принципов согласования (Кондорсе, Борда, Симпсона, Доджсона, Нансона, Кумбса, Фишберна, Коупленда, КМ-метод). Схема работы КПП-метода приведена на рисунке 1.

Преобразование критериальных оценок агентов-экспертов в обобщенные ранжировки и применение к ним принципов согласования позволит построить компромиссное решение независимо от входных данных, не потребует привлечения дополнительных ресурсов на разрешение конфликтов, сократит количество сообщений между агентами в МАС.

Предложенный метод многокритериальной оптимизации в мультиагентной системе применим в ходе:

- оценки деятельности организации за истекший период и т.п.;
- взаимодействия автономных необитаемых подводных аппаратов в ходе проведения поисково-исследовательских работ;
- диагностики состояния сложных технических объектов.

4. Заключение

К мультиагентной системе, включающей агента-руководителя, агента-координатора и агентов-экспертов, применимы методы многокритериальной оптимизации, т.к. они предназначены для обработки информации, представленной в виде как количественных, так и качественных оценок. В работе предложен новый подход к согласованию количественно заданных критериальных оценок, который основан на их преобразовании в обобщенные ранжировки и применении КПП-метода (Комплексное Применение Принципов согласования)

Работа выполнена при частичной финансовой поддержке гранта РФФИ №14-01-31321 «Разработка методов принятия согласованных решений в мультиагентных системах».

Литература

1. Айзerman M.A., Алекскеров Ф. Т. Выбор вариантов основы теории. М.: Наука. – 1990. – 240 с.
2. Ларичев О.И. Теория и методы принятия решений. М.: «Логос». – 2000. – 295 с.
3. Петровский А.Б. Теория принятия решений. М.: Издательский центр «Академия». – 2009. – 400 с.
4. Подиновский В.В., Ногин В.Д.. Парето-оптимальные решения многокритериальных задач. – М.: Физматлит. – 2007. – 256 с.
5. Лотов А.В., Поспелова И.И.. Конспект лекций по теории и методам многокритериальной оптимизации. М.:Изд. ВМиК МГУ. – 2006. – 130 с.
6. Черноруцкий И.Г. Методы принятия решений. СПб.: БХВ-Петербург. – 2005. – 416 с.
7. Митихин В.Г. Об одном контрпримере для метода анализа иерархий // Проблемы управления. – 2012. – №3. – С. 7-79.
8. Подиновский В. В., Подиновская О. В. Еще раз о некорректности метода анализа иерархий // Автоматика и телемеханика. – 2012. – № 4. – С.75–78.

9. Петровский А.Б.. Метод «МАСКА» для групповой экспертной классификации многопризнаковых объектов // Доклады Академии наук. – 2010. – Т. 431. – № 1. – С. 27-31.
10. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. М.: Финансы и статистика. – 2006. – 424 с.
11. Карелин В.П., Протасов В.И, Квадратичный подход к формированию группы экспертов и определению их компетентностей при организации коллективного интеллекта // Вестник Таганрогского университета управления и экономики. – 2012. – №2(16). – С. 46-50.
12. Малтугуева Г.С., Котлов Ю.В. Алгоритм группового выбора при описании индивидуальных предпочтений в виде ранжировок // Вестник ТГУ. – 2004. №9 (II). С. 44 – 47.
13. Малтугуева Г.С., Юрин А.Ю. Метод поддержки принятия решений в малых группах. // Вестник Бурятского государственного университета. Математика, информатика. – 2012 – № 1. – С. 26 – 34.

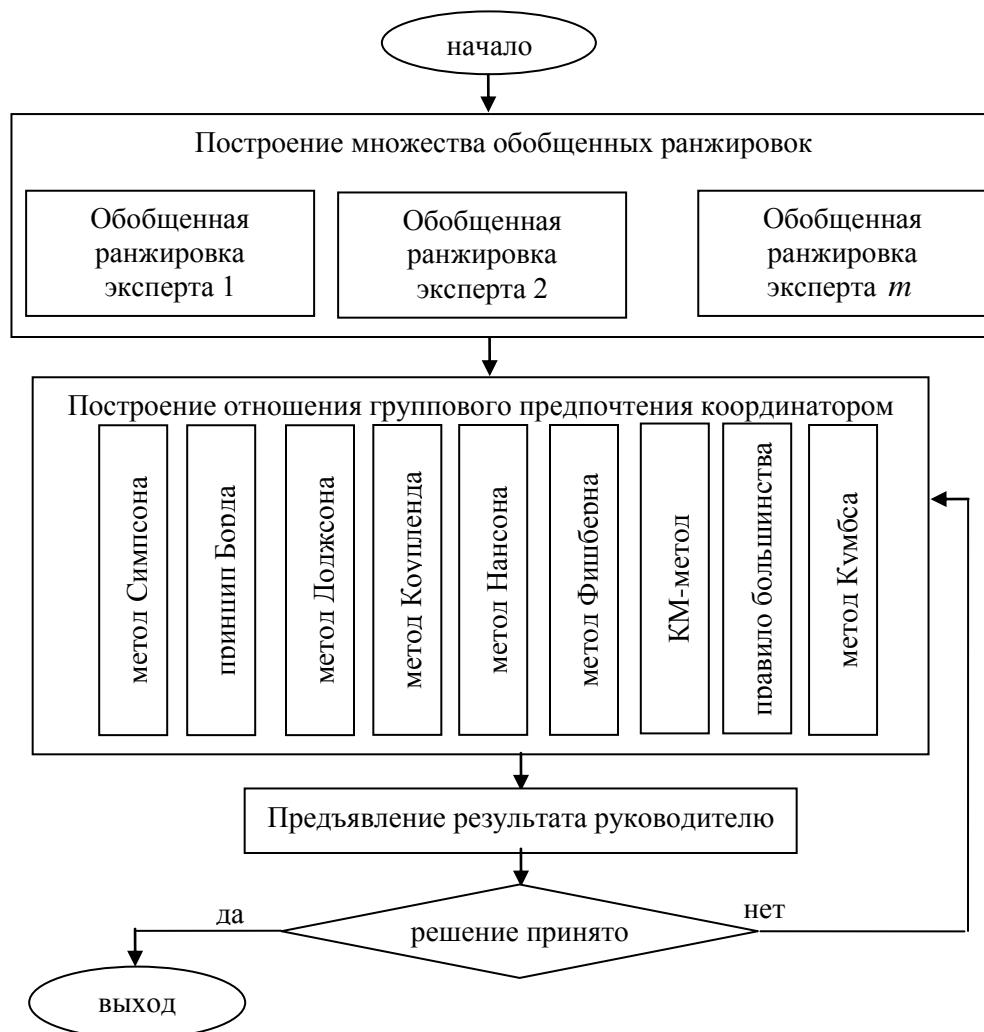


Рис. 1 – Схема работы КПП-метода.

О РЕШЕНИИ ЗАДАЧИ СТРУКТУРНО-ПАРАМЕТРИЧЕСКОГО СИНТЕЗА СЕТИ СТАНДАРТА IEEE 802.11X¹

Н. Ю. Паротькин

Сибирский государственный аэрокосмический университет
имени академика М. Ф. Решетнева, г. Красноярск, 660014, NYParotkin@yandex.ru
В. Г. Жуков

Сибирский государственный аэрокосмический университет
имени академика М. Ф. Решетнева, г. Красноярск, 660014, vadimzhukov@mail.ru

Аннотация

Рассматривается решение задачи структурно-параметрического синтеза беспроводных сетей для обеспечения автоматизированного поиска их оптимальных параметров, обеспечивающих наименьшее отклонение зоны покрытия сети от заданной пользователем. Приводиться формальное описание задачи оптимизации и метода ее решения. Так же рассмотрена программная система, реализующая комбинацию алгоритма оптимизации и модели зоны покрытия, с помощью которой были получены данные сравнительного тестирования с зоной покрытия реальной беспроводной сети.

1. Введение

Беспроводные локально-вычислительные сети (БЛВС) IEEE 802.11x сегодня имеют обширное распространение и применение. Практически все мобильные устройства оснащаются модулем доступа к ним. Это позволяет иметь малозаметный высокопроизводительный инструмент для неправомерной эксплуатации ресурсов сети. Кроме того самостоятельно развернуть данный тип внутриофисной сети не составит затруднения практически для любого человека, благодаря низкой стоимости базового коммуникационного оборудования начального уровня и наличию интегрированных мастеров для его общей настройки. Но их использование часто снижает общую эффективность сети и ее безопасность, поскольку пользователь не задумывается о дополнительных аспектах, влияющих на зону покрытия.

Поскольку частотный диапазон 2,4 ГГц в России является регулируемым и для него действует строгий порядок выделения и назначения частот, то для законной эксплуатации беспроводного оборудования, работающего в этих диапазонах, без его дополнительной регистрации в надзирающих органах необходимо, чтобы оно удовлетворяло определенным ограничениям по характеристикам и способу эксплуатации.

Согласно [1 – 3] разрешено использовать оборудование беспроводных локальных сетей малого радиуса действия стандартов 802.11b/g/n внутри закрытых помещений в диапазоне частот 2400-2483,5, 5150-5350 и 5650-6425 МГц с допустимой мощностью излучения передатчика не более 100 мВт и 200 мВт соответственно без оформления отдельных решений ГКРЧ (Государственной комиссии по радиочастотам) и разрешений на использование радиочастот или радиочастотных каналов при выполнении следующих условий:

¹ Работа выполнена при финансовой поддержке РФФИ, соглашение № 14-07-31036\14 от 24.02.2014 г.

- соответствия технических характеристик, условий использования и типов устройств малого радиуса действия основным техническим характеристикам, условиям использования и типам, указанным в приложениях к [2, 3];
- устройства малого радиуса действия не должны создавать недопустимых помех и не должны требовать защиты от помех со стороны радиоэлектронных средств, работающих в соответствии с «Таблицей распределения полос частот между радиослужбами Российской Федерации»;
- регистрации устройств малого радиуса действия в установленном в Российской Федерации порядке.

Устройство малого радиуса действия – это техническое средство, предназначенное для передачи и/или приёма радиоволн на короткие расстояния. Условие применения устройств малого радиуса действия внутри закрытых помещений предусматривает дополнительное ослабление радиосигнала, вносимое конструкциями помещений, от указанных устройств в направлении других радиоэлектронных средств (РЭС). Несоблюдение вышеперечисленных требований при эксплуатации внутриофисных БЛВС влечет административную ответственность согласно статьям 13.3 и 13.4 кодекса РФ об административных правонарушениях, выражющуюся в конфискации оборудования и наложении штрафа.

Кроме правового аспекта при эксплуатации внутриофисных беспроводных сетей необходимо уделять особое внимание защите данного сегмента корпоративной сети, т.к. она является сетью с низким уровнем доверия, поскольку более уязвима с точки зрения информационной безопасности из-за природы носителя информации, являющегося радиоволной, свободно распространяющейся в пространстве. Следовательно, БЛВС имеют наибольший риск несанкционированного подключения и перехвата трафика. Обычно для решения данной проблемы используют методы сетевой аутентификации клиентов беспроводной сети и криптографической защиты трафика (например, IEEE 802.11i, 802.11x). Для увеличения эффективности данных методов защиты целесообразно на этапе проектирования беспроводной сети уделить внимание вопросу ограничения зоны распространения радиосигнала. Это позволяет решить две задачи: снизить вероятность обнаружения беспроводной сети и уменьшить расстояние, с которого злоумышленник может провести активные или пассивные информационные атаки. Кроме того уменьшения зоны покрытия для общедоступных сетей доступа к Интернет, к примеру пределами кафе или кинотеатра, позволит выделить более высокоскоростной канал посетителям, а не посторонним лицам. Следовательно, при проектировании и развертывании БЛВС необходимо обязательно учитывать вопросы минимизации уровня сигнала вне периметра помещения путем оптимизации зоны покрытия для повышения уровня информационной безопасности, соблюдения требований законодательства и повышения привлекательности организации.

2. Задача оценки зоны покрытия БЛВС

При рассмотрении задачи выбора и размещения сетевого оборудования для внутриофисных сетей возникают две противоположные по сути подзадачи: уменьшения мощности сигнала вне помещения и его максимизация внутри помещения для обеспечения наибольшей скорости доступа к сети. Совместное решение этих подзадач позволяет говорить об оптимальном размещении и настройке сетевого оборудования

БЛВС, выраженное в описании зоны покрытия, конфигурация которой будет совпадать с заданной пользователем. Ее оценка возможна двумя методами. Во-первых, использование эмпирических моделей, предполагающих проведение реальных замеров при различных конфигурациях оборудования. Во-вторых, создание теоретических моделей распространения сигнала с выбором и проверкой наилучшей конфигурации из перечня всего доступного оборудования. Эти подходы имеют как преимущества так и недостатки. Первый предполагает высокие материальные и временные затраты с точным получением конечного результата, учитывая все особенности взаимодействующих реальных объектов. Второй же дает приблизительный результат с максимальной экономией времени и материальных ресурсов, который при необходимости может быть уточнен с помощью первого метода путем изменения настроек оборудования. Следовательно, второй подход является предпочтительным при создании средства для решения задачи проектирования БЛВС.

Зона покрытия и скорость доступа к внутриофисной БЛВС, т.е. ее итоговые характеристики с точки зрения пользователя, определяются техническими характеристиками используемого сетевого оборудования, местом его размещения и загруженностью радиочастотного диапазона. Учет радио обстановки осуществляется автоматически сетевым оборудованием, т.е. точка доступа самостоятельно может выбирать наименее загруженный канал, а контролеры беспроводных точек доступа согласовывать их выбор и обеспечивать роуминг клиентов между ними для равномерного распределения нагрузки. Следовательно, пользователю необходимо определить только физические параметры оборудования, изменение которых затруднено после введения сети в эксплуатацию:

- 1) модель точки доступа и места её размещения;
- 2) модели сетевых карт (уровень чувствительности приемника);
- 3) мощность сигнала активного сетевого оборудования (зависит от модели точки доступа и беспроводных сетевых карт);
- 4) выбор антенн с определенной диаграммой направленности и их ориентация в пространстве.

В итоге получается 8 независимых переменных, оптимальное соотношение которых должно определить заданную скорость работы беспроводной сети внутри помещения и минимальную мощность сигнала вне его.

Задаче построения модели зоны покрытия беспроводных сетей посвящено значительное количество работ, среди которых работы российских и зарубежных ученых: В. М. Вишневского, С. Портного, И. Шахновича, G. Balbo, S. C. Bruell, L. Fratta, L. Kleinrock, M. Olivetty, H. Takagi, S.C. Borst, O.J. Boxma. В них в основном рассматривается описание зоны покрытия для территорий с размерами от городского квартала, при этом некоторые могут быть использованы для описания внутриофисных беспроводных сетей. Примером таких моделей может послужить методы геометрической оптики (метод мнимых источников, запуска лучей) или модель представленная в работе [4]. Их применение в автоматизированных средствах затруднено, поскольку требует больших накладных расходов на единичное вычисление характеристик сети и введение ограничений на зону покрытия. Чаще всего в них применяется минимизация площадей зон перекрытия от различных точек доступа и затухание сигнала при прохождении сквозь препятствие, как это реализовано в Ekahau Site Survey и AirMagnet Planner.

Другая модель для покрытия внутриофисной БЛВС, позволяющая задавать зону

распространения сигнала и подходящей проведения поиска оптимального решения на стационарном компьютере, предложена в [5]. Она позволяет, учитывая характеристики помещения и применяемого оборудования, строить зоны покрытия БЛВС различной формы, максимально удовлетворяющие требованиям пользователя.

3. Алгоритм оценки зоны покрытия сигнала

В основе предлагаемого алгоритма лежит метод определения мощности сигнала БЛВС на определенном расстоянии от точки доступа. Она оценивается по формуле (1):

$$P_{i,\partial BmBm} = P_{t,\partial BmBm} + G_{t,\partial B} - (43 + 20(\lg D + \lg F) + L_{\partial B}), \quad (1)$$

где $P_{t,\partial BmBm}$ – мощность передатчика точки доступа; $G_{t,\partial B}$ – коэффициент усиления передающей антенны в направлении точки расчета мощности; D – расстояние до точки (км); F – центральная частота канала, на котором работает система связи ($MГц$); $L_{\partial B}$ – падение мощности сигнала при прохождении через препятствия. В формуле (1) вводится компенсация возможных помех в канале за счет введения дополнительных постоянных потерь, выражаемых числом 43 dB . Параметр $L_{\partial B}$ определяется видом и толщиной материала, который необходимо преодолеть сигналу по прямой линии от точки, в которой вычисляется мощность сигнала, до передающего устройства. Примерная величина ослабления сигнала в децибелах для различных типов препятствий представлена в таблице 1 и получена на практике в ходе исследований, проводимых в рамках программы «У.М.Н.И.К.».

Для определения возможных препятствий на пути сигнала использовался метод геометрической оптики. Для этого выбирается наименьшая длина между X и Y координатами и используется её обратная величина в качестве приращения по обеим координатам. После каждой итерации из мощности сигнала, подсчитанного в точке, вычитается ослабление сигнала для координат, полученных на данном шаге.

Оценка скорости доступа к сети в данной точке (V_j) осуществляется по величине, получаемой в результате сложения P_i для данной точки и наибольшего коэффициента усиления для принимающей антенны (G_r). Полученное значение сравнивается с возможными значениями чувствительности приёмника в сетевой карте. Для каждой скорости приёмник имеет определённую чувствительность (P_{jr}). Для небольших скоростей чувствительность наименьшая, для высоких намного выше. Данная информация указывается производителем в спецификации оборудования. Следовательно, скорость доступа к сети в i точке для данной сетевой карты можно оценить по формуле (2):

$$V_j : \max_j P_{jr} \leq P_i + G_r. \quad (2)$$

Для интегральной оценки зоны покрытия БЛВС от одной точки доступа предлагается использовать следующее выражение, позволяющее достичь максимальной скорости доступа к сети с минимизацией мощности сигнала вне требуемой территории (формула 3).

Таблица 1

Зависимость затухания сигнала от среды распространения

Название	Толщина (м)	Коэффициент (дБ)
Железобетонная несущая стена	0,4	25
Железобетонная несущая стена	0,3	21
Железобетонная несущая стена	0,15	15
Межкомнатная гипсовая стена	0,1	1
Кирпичная стена	0,6	14
Кирпичная стена	0,4	9
Кирпичная стена	0,1	4
Металлическая дверь	-	30
Деревянная дверь	-	2
Двойная деревянная дверь со стеклом	-	6
Стеклопакет без тонировки	-	3
Стеклопакет с тонировкой	-	7
Двойное окно в деревянной раме	-	6
Фасадная плитка	-	2

$$F(X) = \sum_{i=1}^N (P_i - T) / T - r^2 \sum_{i=1}^R e^{V-V_i}, \quad (3)$$

где $F(X)$ – функция оценки; N – количество точек, где мощность сигнала не должна превышать заданного порога; P_i – мощность сигнала в точке, рассчитываемая по формуле (1); T – максимальный допустимый уровень мощности сигнала, задаётся пользователем; R – количество точек помещения, местоположение которых определяется выделением областей на плане пользователем, где необходим доступ к сети с заданной скоростью; r – коэффициент, определяющий критичность уменьшения скорости передачи относительно заданной, задаётся пользователем (рекомендованное значение 0,4); V – требуемая скорость передачи данных для сетевой карты, изменяется дискретно, имеет всего 9 значений от 0 Мбит/с до 54 Мбит/с, т.е. задаётся множество $\{1;2;\dots;8;9\}$; V_i , – предполагаемая скорость доступа в i -ой точке при данных настройках сети, определяется на основании чувствительности приемника выбранной модели сетевой карты и мощности сигнала в точке по формуле (2). Вектор X будет иметь следующие измерения: x_1 – номер модели точки доступа, влияет на значение параметра $P_{t,\text{дБмВт}}$ из формулы (1); x_2 и x_3 – первая и вторая координата места размещения точки доступа, определяют параметр D из формулы (1); x_4 – уровень мощности сигнала точки доступа, влияет на значение параметра $P_{t,\text{дБмВт}}$ из формулы (1); x_5 – номер модели антенны точки доступа, определяет значение параметра $G_{t,\text{дБ}}$ из формулы (1); x_6 – угол поворота антенны точки доступа, влияет на значение параметра $G_{t,\text{дБ}}$ из формулы (1); x_7 – номер модели сетевой карты, определяет значение параметра P_{jr} из формулы (2); x_8 – номер модели антенны сетевой карты, определяет значение параметра G_r из формулы (2).

Переменная, соответствующая модели оборудования, представляется номинальной шкалой, значения которой соответствуют записям в таблице базы данных, содержащей характеристики конкретной модели оборудования, необходимые для вычисления

целевой функции. В бинарной строке она представляется порядковым номером записи. Кодирование угла поворота антенны осуществляется указанием одного из 24 секторов (длина дуги соответственно равна 15°). На середину 12 сектора должен приходиться главный лепесток диаграммы направленности. На текущий момент вектор X описывается битовой строкой состоящей из 33 бит, т.е. данный вектор может принимать порядка $8,6 \cdot 10^9$ значений. Единичное вычисление функции (3) занимает 0,001 с на одном ядре процессора с тактовой частотой 3,2 ГГц.

4. Решение задачи оптимизации функции оценки зоны покрытия сигнала

Нахождение экстремума функции (3) позволит найти оптимальную конфигурацию БЛВС для данного помещения, что позволяет создать на основе решения данной задачи оптимизации автоматизированное программное средство для структурно-параметрического синтеза беспроводной сети. Следовательно, необходимо дать формальную постановку задачи оптимизации (формула 4).

$$F(X) = \sum_{i=1}^N (P_i - T)/T - r^2 \sum_{i=1}^R e^{V-V_i} \rightarrow \max , \quad (4)$$

$$\begin{cases} V_i \geq V_{\min} \\ C_t + C_{Gt} + nC_r + nC_{Gr} \leq C \end{cases}$$

где V_{\min} – минимально допустимая скорость доступа к сети; C_t , C_{Gt} , C_r , C_{Gr} – стоимость точки доступа, ее антенны, сетевой карты и ее антенны; C – бюджет на внедрение сети. Кроме того значения измерений вектора X дополнительно ограничены исходя из возможных мест размещения оборудования, совместимости моделей антенн и точек доступа и сетевых карт, ограниченного перечня сетевого оборудования и т.д.

Таким образом, задача структурно-параметрического синтеза беспроводных локальных сетей является задачей условной оптимизации с алгоритмически заданной целевой функцией и ограничениями на множестве разношкольных переменных, принадлежащих множеству натуральных чисел. Следовательно, в силу достаточной ресурсоемкости единичного вычисления целевой функции, зависимости ее ландшафта от характеристик помещения и большой размерности вектора X , то для решения данной задачи оптимизации в данной постановке целесообразно использовать эвристические методы оптимизации, например, генетические алгоритмы. В качестве такового был применен дифференцированный генетический алгоритм [6], поскольку применение стандартного генетического алгоритма в силу разношкольности переменных не дало требуемого уровня качества поиска решения. Он обладает следующими отличительными особенностями, повышающими его эффективность:

1) Применение системы кодирования признаков, при которой соседние значения фенотипа отличаются меньшим количеством позиций в генотипе, в идеале значением одного бита, например, коды Грэя или Джонсона.

2) Применение генератора псевдослучайных чисел, обладающего хорошими статистическими характеристиками для повышения качества генерации случайных последовательностей чисел, например, BBS-генератор (алгоритм Блюма-Блюма-Шуба).

3) Применение значения длины хромосомы для определения шага сетки в пространстве поиска, так как при решении большинства практических задач идет поиск аргументов целевой функции, а не ее значений и при таком определении шага всегда осуществляется поиск не конкретного значения точки оптимума, а ее окрестности с некоторым малым радиусом Δx .

4) Применение параметра, отражающего время жизни каждого решения для реализации механизма информационного взаимодействия между подсистемами, препятствующего смешиванию всей информации. Значение данного параметра является определяющим при принятии решения о переходе информации, закодированной в решении, из оперативной в консервативную подсистему.

5) Информационные объемы оперативной и консервативной подсистем являются переменной величиной, тесно связанной с условиями среды. Поэтому, в процессе работы алгоритма, это соотношение должно меняться по некоторому закону, аналогично изменению окружающей среды.

6) В формировании новых решений могут участвовать только пары (s_k^i, s_o^j) , при этом решение s_k^i выбирается случайным образом из S_k , а решение s_o^j по турнирной схеме из S_o .

7) Применение различных стратегий формирования нового поколения для оперативной и консервативной подсистем. С учетом сохранения своих специализаций для решений оперативной подсистемы стратегия аутбридинга, а для консервативной – инбридинг.

8) Применение в качестве критерия остановки работы алгоритма комплексный показатель, который учитывает динамику следующих параметров: дисперсия генотипов и фенотипов оперативной и консервативной субпопуляций, соотношение $\frac{S_k}{S_o}$.

Таким образом, блок-схема дифференцированного генетического алгоритма представлена на рисунке 1.

Для решения задачи условной оптимизации было реализовано сочетание метода динамических штрафов и следующих правил отбора индивидов в субпопуляцию S_k :

1) удовлетворение наибольшему количеству критериев;

2) наибольшее значение параметра T_{life} .

Решение поставленной задачи структурно-параметрического синтеза сети путем оптимизации функции (3) дифференцированным генетическим алгоритмом было реализовано в виде программного обеспечения, которое на основании плана помещения с указанием мест размещения рабочих мест, материалов стен и уровня допустимого сигнала вне помещения предоставляло пользователю параметры БЛВС с зоной покрытия, соответствующей его требованиям.

5. Программная система решения задачи структурно-параметрического синтеза сети стандарта IEEE 802.11x

На основе выше представленных данных было разработано программное средство, выполняющее на основе введённого плана помещения, загружаемого перечня оборудования, установок желаемых уровней мощности сигнала вне помещений и требуемых зон покрытия сигнала структурно-параметрический синтез беспроводной сети Wi-Fi, осуществляя подбор оптимальной комплектации оборудования, его настроек и размещения для данных условий [6]. Структурная схема программной системы приводится на рисунке 2.

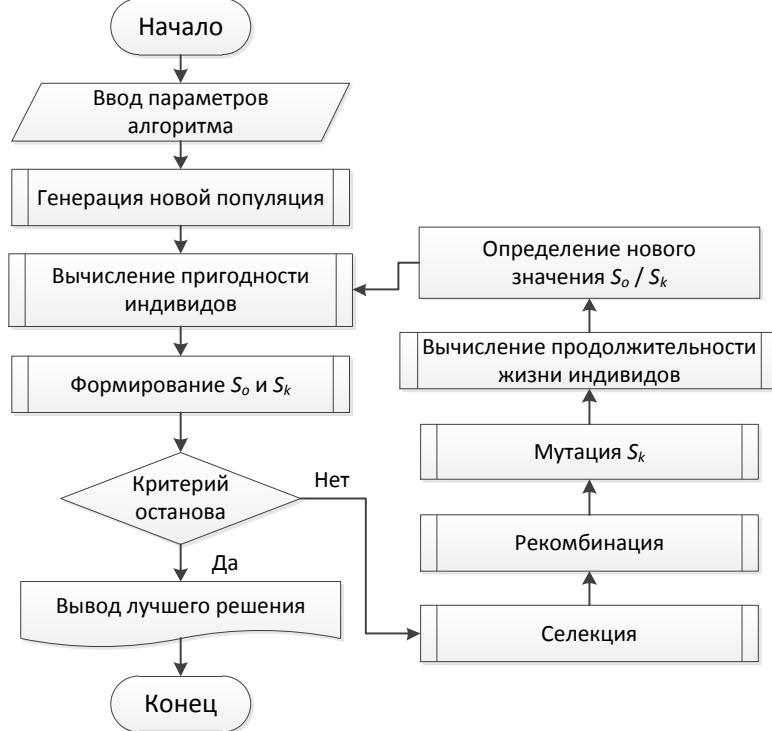


Рис. 1 – Блок-схема дифференцированного генетического алгоритма

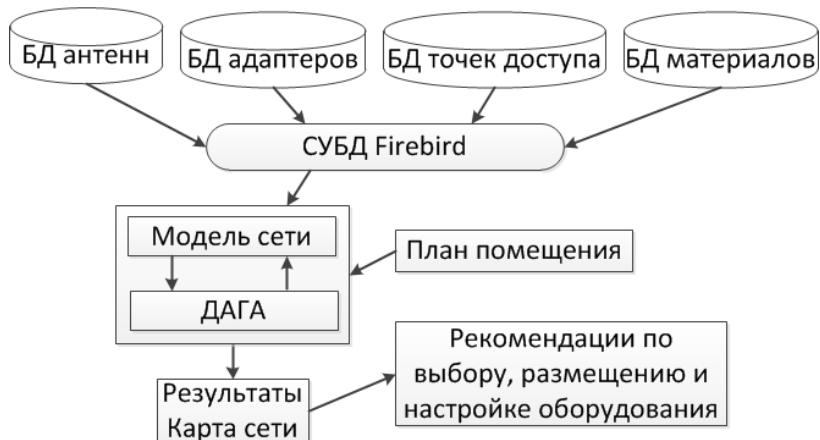


Рис. 2 – Структурная схема программной системы

Для работы программного средства необходимо загрузить базы данных точек доступа, сетевых карт и антенн, в которых указываются параметры оборудования, необходимые для расчёта сети, например, мощность излучаемого сигнала, чувствительность приёмника, диаграмма направленности антенн. Для тестирования программного средства были использованы характеристики сертифицированного сетевого оборудования фирм, наиболее распространенных на российском рынке: D-Link, TP-Link, Cisco, HP, ASUS. Для хранения данных используется СУБД Firebird 2.5 embedded. Информация хранится в файле базы данных wifi.fbd, расположенным в директории с программой. Для его редактирования применяется утилита Admin.exe с графическим пользовательским интерфейсом, позволяющим вносить изменения и дополнения в базу данных (рисунок 3).

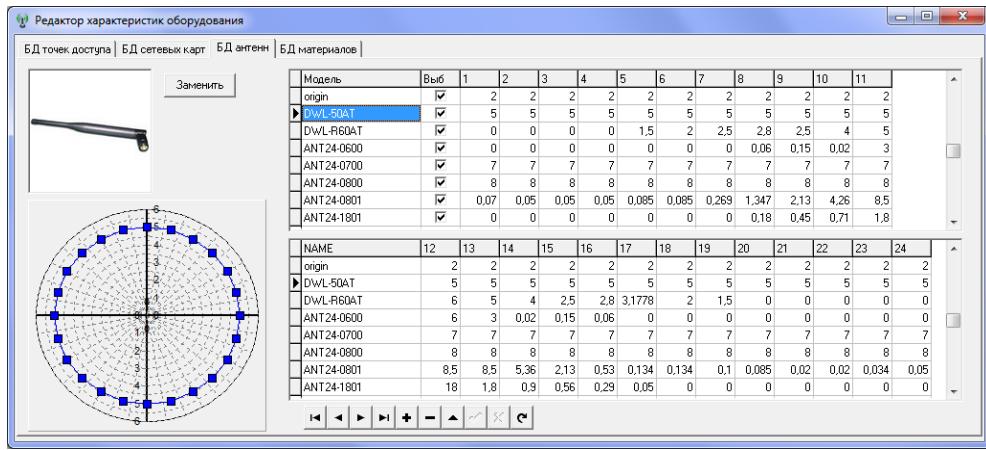


Рис. 3 – Экранная форма программы для редактирования базы данных

Пользовательский интерфейс программной системы аналогичен подобного рода программам (рисунок 4). В нем основную часть занимает окно для ввода плана помещения, в котором предполагается развернуть БЛВС. Данный процесс аналогичен работе с графическим редактором, т.е. при выборе справа определенного типа материала необходимо провести линию, соответствующую конструкции в помещении, созданной из него. С помощью выделения областей помещения пользователь имеет возможность указывать места, в которых доступ к сети необходим или запрещен. В программе присутствует возможность задания масштаба, максимально возможной мощности сигнала вне помещения и минимально необходимой скорости доступа к сети.

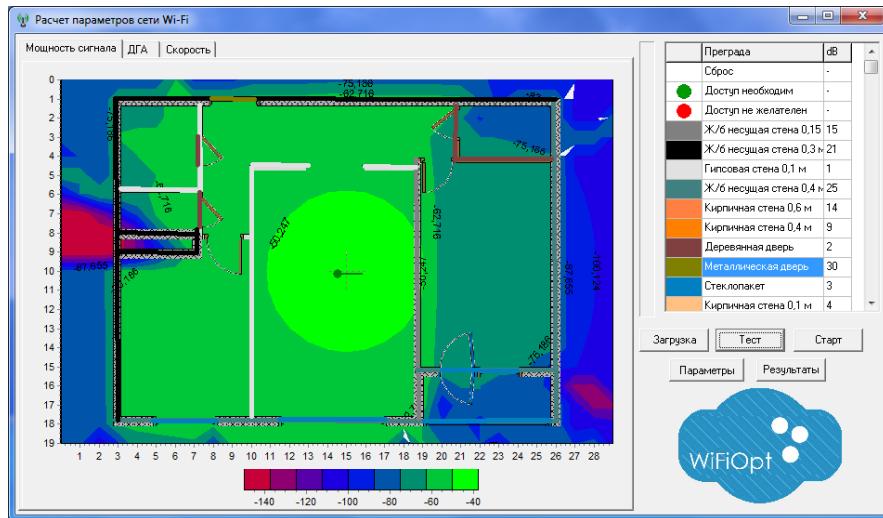


Рис. 4 – Экранная форма программной системы «Расчет параметров сети Wi-Fi»

Кнопки управления, расположенные на форме (рисунок 4), позволяют загружать данные характеристик оборудования, вызывать окна для настройки параметров дифференцированного генетического алгоритма. Кнопка «Тест» позволяет рассчитать характеристики сети и значение целевой функции, при этом параметры оборудования, место его размещения и модель указываются в диалоговом окне «Параметры оборудования». После окончания расчета параметров в окно «Отчет» выводится полная спецификация использованного оборудования и параметры его установки.

Кнопка «Старт» позволяет запускать алгоритм нахождения оптимальных параметров сети, т.е. нахождения генетическим алгоритмом максимума целевой функции. Параметром завершения поиска экстремума является достижение числа поколений, указанного в параметрах дифференцированного ГА. В процессе работы алгоритма на график выводится статистическая информация по его работе.

По окончанию работы программной системы пользователю представляется полный отчет о моделях оборудования, месте его размещения и рекомендуемых настройках. Так же на план помещения накладывается карта мощности радиосигнала БЛВС (рисунок 4) в dBm при данных настройках и предполагаемая скорость доступа к ней. Местоположение и направление антенны точки доступа указывается символом  . Осью поворота является утолщение на линии, отмеченное зеленой точкой.

6. Проверка корректности прогнозируемых данных

Для оценки адекватности созданной модели необходимо сравнить мощность сигнала в точке, предполагаемую моделью сети и полученную экспериментально на основе замера мощности сигнала от реального оборудования. Для всех измерений в качестве точки доступа использовался беспроводной маршрутизатор D-Link DIR-300, как наиболее распространённый в домашнем сегменте рынка, с всенаправленной антенной, имеющей коэффициент усиления 2 dBi , и расположенный в примерном центре помещения. Его мощность передатчика была установлена в 3 dBm для более наглядного отображения картины затухания сигнала. В качестве приемника сигнала использовался адаптер D-Link DWA-126 с чувствительностью -62 dBm и внешней всенаправленной антенной с коэффициентом усиления 3 dBi . Для замера мощности сигнала использовалось программное обеспечение MetaGeeks inSSIDer.

Измерения проводились для 3 тестовых помещений. Для первого помещения результаты моделирования сети с выше обозначенными параметрами приведены на рисунке 4. В общей сложности для каждого помещения было проведено не менее 400 замеров. Отклонение рассчитывалось как взвешенная разница между реальной и предполагаемой мощностью к реальной, выраженная в процентах (рисунок 5).

По итогам всех исследований были получены следующие данные средней абсолютной ошибки моделирования для помещений:

- помещение № 1 – 4%;
- помещение № 2 – 8%;
- помещение № 3 – 8%.

7. Заключение

По результатам сравнения реальных данных и полученных в результате моделирования можно сделать следующие выводы:

- наличие небольшого количества точек, где мощность сигнала значительно превышает прогнозируемую обусловлено отражением радиоволн от поверхностей материалов и тем самым в точку измерения они приходят с большей мощностью, чем предсказывалось;

-3%	0%	-3%	-4%	-3%	4%	-2%	-5%	7%	-4%	-2%	-2%	2%	9%	-1%	-4%	7%	-2%	3%	-1%	9%	9%	3%	0%	-2%	0%	-3%	-2%	7%	0%	
0%	7%	-3%	7%	7%	2%	1%	1%	8%	5%	7%	3%	-3%	3%	1%	-2%	-5%	-1%	2%	-1%	-1%	-5%	9%	1%	2%	5%	-4%	-2%	7%	-4%	
4%	2%	8%	8%	-2%	5%	7%	5%	-5%	2%	-5%	0%	9%	4%	1%	9%	6%	-5%	8%	-5%	5%	-5%	2%	7%	-1%	2%	7%	3%	4%	1%	
-3%	-4%	9%	3%	8%	7%	5%	-1%	8%	-5%	7%	4%	-1%	-5%	2%	-4%	4%	3%	7%	6%	7%	-1%	3%	7%	5%	6%	1%	4%	4%	-1%	
0%	1%	4%	4%	-4%	1%	0%	2%	6%	6%	-3%	15%	4%	3%	7%	-5%	4%	0%	7%	1%	-3%	3%	5%	0%	1%	7%	3%	1%	8%	5%	
5%	-5%	-1%	-3%	7%	5%	-1%	7%	-5%	-5%	3%	0%	7%	0%	9%	3%	-5%	-3%	9%	1%	-1%	7%	3%	5%	4%	1%	-2%	-2%	5%		
9%	-4%	6%	0%	1%	8%	4%	-5%	5%	-1%	1%	6%	7%	9%	1%	-4%	7%	-2%	-3%	8%	-5%	-4%	-4%	-1%	2%	7%	7%	-1%	-3%	2%	
3%	7%	6%	0%	1%	-4%	2%	7%	7%	-3%	4%	-2%	-4%	2%	3%	0%	-3%	2%	8%	-5%	2%	-5%	5%	15%	3%	-1%	0%	-1%	1%	0%	
7%	-1%	9%	4%	0%	5%	-2%	4%	4%	1%	6%	-4%	0%	0%	-2%	2%	-4%	3%	1%	3%	-3%	8%	15%	0%	0%	7%	6%	-1%	-5%	4%	8%
1%	-5%	5%	7%	6%	15%	2%	3%	5%	5%	6%	7%	0%	2%	0%	3%	1%	-3%	6%	3%	0%	-3%	-4%	7%	-4%	3%	7%	0%	6%	8%	
3%	7%	0%	7%	7%	6%	-4%	6%	1%	7%	0%	-1%	0%	0%	-2%	-1%	0%	-1%	9%	1%	3%	-4%	4%	2%	5%	-3%	-3%	-5%	-4%	-3%	
9%	8%	6%	6%	15%	6%	8%	3%	-4%	8%	7%	-2%	1%	0%	-3%	-2%	0%	-2%	-4%	4%	9%	8%	7%	-3%	-5%	7%	5%	6%	-3%	6%	
8%	1%	1%	15%	7%	7%	7%	6%	5%	6%	5%	-4%	2%	3%	3%	0%	-1%	1%	-4%	-3%	0%	7%	8%	-1%	8%	4%	-4%	-2%	8%	9%	
-1%	9%	-4%	-3%	-3%	3%	1%	15%	-1%	3%	8%	7%	3%	-2%	0%	1%	2%	4%	-4%	1%	-1%	3%	5%	7%	-3%	-3%	-5%	3%	-2%	1%	
7%	4%	4%	3%	4%	3%	6%	0%	7%	-3%	7%	0%	1%	0%	0%	-1%	-2%	3%	5%	7%	1%	7%	5%	1%	7%	9%	3%	3%	6%	-5%	
2%	-5%	-1%	0%	7%	15%	7%	0%	5%	4%	6%	8%	5%	-4%	1%	-1%	-2%	7%	7%	7%	2%	7%	-5%	8%	8%	6%	-1%	7%	2%	1%	9%
7%	-5%	5%	-5%	5%	7%	3%	-4%	-5%	3%	6%	-5%	5%	6%	8%	6%	1%	8%	-3%	-2%	3%	-5%	8%	3%	2%	1%	2%	-1%	6%	-4%	
-3%	4%	7%	3%	5%	2%	0%	2%	7%	2%	7%	2%	-1%	-4%	-4%	0%	7%	7%	8%	1%	1%	-1%	0%	3%	4%	-1%	0%	7%	7%	9%	
-3%	3%	8%	9%	3%	-1%	-1%	2%	4%	-3%	-1%	0%	8%	6%	3%	1%	-4%	-5%	-4%	-4%	3%	1%	7%	1%	9%	6%	3%	-2%	-3%	1%	
7%	7%	3%	3%	-5%	-2%	7%	8%	-4%	-3%	7%	0%	1%	-1%	0%	-2%	4%	7%	-4%	2%	-1%	0%	-1%	1%	8%	3%	-5%	2%	6%	-2%	

Рис. 5 – Ошибки в прогнозировании мощности сигнала для помещения № 1

- наличие значительной разницы в мощности сигнала между двумя соседними точками обуславливается неоднородностью преград и разными путями распространения радиоволн;
- большее значение ошибки для кирпичных зданий определяется большей толщиной внешних стен и, следовательно, большей неоднородностью;
- для больших помещений становится заметно влияние других источников радиосигнала, вносящих помехи в измеряемый сигнал.

Разработанное программное средство способно рассчитать параметры внутриофисной беспроводной сети Wi-Fi с необходимой точностью для одной точки доступа, при этом соблюдая ограничения, устанавливаемые на мощность сигнала вне помещения и на скорость передачи данных в определённых точках внутри него, поэтому можно говорить об успешном решении задачи структурно-параметрического синтеза сети стандарта IEEE 802.11x в соответствии с требованиями законодательства и безопасности. Это позволяет его применять при построении моделей внутриофисных беспроводных сетей стандарта IEEE 802.11g автоматизированным способом с возможностью количественно оценить степень соответствия итоговой сети предъявляемым требованиям и ограничениям. В дальнейшем данная модель будет расширена на стандарт IEEE 802.11n и сеть беспроводных точек доступа, работающих согласованно.

Литература

1. Решение ГКРЧ от 20 декабря 2011 г. № 11-13-07-1 «О внесении изменений в решение ГКРЧ от 7 мая 2007 г. № 07-20-03-001 «О выделении полос радиочастот устройствам малого радиуса действия»
2. Решение ГКРЧ от 7 мая 2007 г. № 07-20-03-001 «О выделении полос радиочастот устройствам малого радиуса действия»
3. Решение ГКРЧ от 15 июля 2010 г. № 10-07-02 «Об использовании полос радиочастот 5150-5350 МГц и 5650-6425 МГц радиоэлектронными средствами фиксированного беспроводного доступа»
4. Косилов, Н.А. Модели и алгоритмы пространственной организации беспроводных широкополосных сетей : автореф. дис. канд. техн. наук / Н.А. Косилов ; Московский гос. институт электроники и математики. – Москва, 2012. – 20 с.

5. Жуков, В.Г. Автоматизированная система проектирования беспроводной сети IEEE 802.11/ В.Г. Жуков, Н.Ю. Паротькин // Международный журнал «Программные продукты и системы», – Тверь: ЗАО НИИ «Центрпрограммсистем», 2012. – Выпуск № 3.– С. 202 – 207.
6. Паротькин, Н.Ю. Дифференцированный генетический алгоритм решения сложных задач оптимизации : дисс. ... канд. техн. наук / Н.Ю. Паротькин – Красноярск: СибГАУ, 2013. – 109 с.

РЕАЛИЗАЦИЯ АГЕНТА ДОСТУПА К ВЫЧИСЛИТЕЛЬНЫМ РЕСУРСАМ ДЛЯ ПЛАТФОРМЫ EVEREST

Рубцов А.О

Институт проблем передачи информации РАН

Москва, rubtsov493@gmail.com

Сухорослов О.В.

Институт проблем передачи информации РАН

Москва, oleg.sukhoroslov@gmail.com

Аннотация

Данная работа посвящена разработке программного компонента облачной платформы Everest, осуществляющего интеграцию в платформу различного рода вычислительных ресурсов. Данный компонент, называемый агентом, выполняясь на стороне вычислительного ресурса, должен обеспечить связь с платформой и выполнение заданий, сформированных и переданных ему от сервисов платформы. В работе описаны протокол взаимодействия агента и платформы, архитектура и реализация агента, а также особенности интеграции агента с различными типами ресурсов.

1. Введение

Облачная платформа Everest [1] реализует модель «платформа как сервис», заключающуюся в предоставлении интегрированной платформы для разработки, тестирования, развертывания и поддержки вычислительных приложений и сервисов. В основе платформы лежит сервис-ориентированная архитектура, в рамках которой вычислительное приложение представляется в виде доступного по сети сервиса.

Everest позволяет обеспечить возможность интеграции приложений друг с другом и внешними приложениями на уровне программных интерфейсов соответствующих сервисов. Пользователи платформы могут самостоятельно создавать на основе имеющихся приложений новые вычислительные сервисы и предоставлять доступ к ним другим пользователям. Это позволяет раскрыть имеющийся потенциал сервис-ориентированного подхода для научной коллаборации и публикации результатов исследований. Платформа доступна через веб-браузер, без необходимости установки дополнительного ПО на компьютер пользователя.

Важной особенностью платформы Everest является то, что, вместо предоставления собственной вычислительной инфраструктуры для выполнения приложений, платформа предлагает разработчикам сервисов готовые средства для интеграции с внешними вычислительными ресурсами. Это позволяет избежать затрат, связанных с созданием соответствующей инфраструктуры, и максимально задействовать потенциал имеющихся в распоряжении пользователей вычислительных ресурсов.

В данной работе рассматривается реализация механизма интеграции платформы Everest с вычислительными ресурсами. В основе предлагаемого подхода лежит использование специального программного компонента (т. н. агента), выполняющегося на стороне ресурса, под полным контролем его владельца, и играющего роль посредника между платформой и ресурсом.

Основными функциями агента являются выполнение вычислительных задач на ресурсе (включая загрузку входных данных, запуск задачи, контроль выполнения и передачу результатов) и предоставление информации о характеристиках и текущем состоянии ресурса, которая используется при планировании задач на стороне платформы.

В отличие от подходов, основанных на использовании существующих механизмов удаленного доступа к ресурсам (например, протокола SSH), использование агента позволяет избежать передачи платформе реквизитов для прямого доступа к ресурсу и связанных с этим рисков, реализовать контроль владельца над тем, как используется ресурс, а также обеспечить подключение ресурсов, изначально недоступных по сети.

В разделе 2 описан протокол взаимодействия агента и платформы Everest. В разделе 3 рассматриваются архитектура и принципы реализации агента. Раздел 4 посвящен интеграции агента с различными типами вычислительных ресурсов. В Заключении формулируются выводы и планы дальнейшего развития реализации агента.

2. Протокол взаимодействия агента и Everest

Для взаимодействия агента и платформы Everest используются протоколы HTTP(S) и WebSocket [2]. Выбор данных протоколов обусловлен в частности тем, что они позволяют обеспечить двустороннее взаимодействие между агентом и платформой даже в ситуациях, когда агент выполняется на машине, находящейся за межсетевым экраном. В общем случае достаточно, чтобы HTTP-сервер платформы был доступен с машины агента.

При запуске агент устанавливает шифрованное WebSocket-соединение с сервером платформы, образующее двусторонний канал для обмена сообщениями между агентом и сервером. Рассматриваемый далее протокол взаимодействия агента и платформы определяет типы и форматы сообщений, которыми они могут обмениваться, безотносительно того, каким образом реализован агент.

Непосредственно после установления соединения с сервером агент должен пройти аутентификацию, путем отправки серверу текстового сообщения, начинающегося со строки «AUTN» и содержащего секретный код, идентифицирующий ресурс. Данный код владелец ресурса получает при регистрации ресурса в Everest. Аутентификация позволяет однозначно идентифицировать ресурс, а также предотвратить его несанкционированное использование и «подмену» другими пользователями платформы. Аутентификация платформы на ресурсе не производится.

После успешной аутентификации агент может принимать от сервера управляющие команды, оформленные в виде сообщений следующих типов:

- Назначение задачи (TASK_SUBMIT);
- Отмена задачи (TASK_CANCEL);
- Агент также может отправлять следующие информационные сообщения серверу:
- Текущее состояние ресурса (HEARTBEAT);
- Уведомление об изменении состояния задачи (TASK_STATE).

Все упомянутые сообщения, передаваемые через WebSocket-соединение, имеют текстовый формат и относительно небольшой размер. Передача файлов с данными (входных и выходных файлов задач) между агентом и платформой Everest осуществляется через протокол HTTP.

При потере соединения с сервером агент предпринимает попытки заново установить соединение и пройти аутентификацию.

Таким образом, протокол взаимодействия агента и сервера включает пять типов сообщений. Подробное описание данных типов сообщений приведено в таблице ниже.

Формат сообщения	Направление	Описание
AUTH token resource_info	Агент → сервер	Аутентификация агента на сервере. Если аутентификация не прошла, сервер закрывает соединение с агентом token - секретный код, используемый для идентификации ресурса, resource_info - информация о ресурсе в виде строки в формате JSON [3]
TASK_SUBMIT task_descr	Сервер → агент	Назначение задачи ресурсу task_descr — описание задачи в формате JSON
TASK_CANCEL task_id	Сервер → агент	Отмена задания task_id - идентификатор задачи, указанный в task_descr
TASK_STATE task_id state info	Агент → сервер	Уведомление об изменении состояния задачи. task_id — идентификатор задачи state — состояние задачи info — строка содержащая дополнительные данные о состоянии задания, например причина перехода в состояние FAILED
HEARTBEAT resource_info	Агент → сервер	Отправляется агентом с заданной периодичностью, по умолчанию 10 секунд.

Атрибуты структуры TASK_DESCR, описывающей вычислительную задачу, приведены в таблице ниже.

Атрибут	Описание
id	уникальный идентификатор задачи
user	идентификатор пользователя (на стороне сервера), от имени которого запускается задача
service	идентификатор сервиса (на стороне сервера), вызов которого инициировал запуск задачи
command	команда, которую необходимо запустить для выполнения задачи

stage-in	массив URL архивов с входными данными задачи
stage-out:	
files	массив имен выходных файлов задачи (с поддержкой шаблонов), которые необходимо передать на сервер после окончания выполнения задачи
destination	URL, по которому необходимо загрузить архив с выходными файлами

Возможные значения параметра STATE, отражающего состояние задачи, приведены в таблице ниже.

Состояние	Описание
STAGED_IN	Входные данные задачи загружены
QUEUE	Задача находится в очереди агента или передана ресурсу и ожидает исполнения в его внутренней очереди.
RUNNING	Задача выполняется
COMPLETED	Выполнение задачи завершено
STAGED_OUT	Выходные данные задачи загружены на сервер
DONE	Выполнение задачи завершено успешно
FAILED	Выполнение задачи завершено с ошибкой
CANCELLED	Выполнение задачи отменено

В строке RESOURCE_INFO передается следующая информация:

- Тип ресурса;
- Информация о системе в формате Unix-команды «uname»;
- Максимальное количество задач, одновременно выполняемых агентом;
- Полное количество слотов на ресурсе;
- Количество свободных слотов на ресурсе;
- Полное количество задач, обрабатываемых агентом в данный момент;
- Количество задач, выполняющихся на ресурсе в данный момент.

Данная информация используется платформой для мониторинга состояния ресурсов и планирования выполнения задач.

3. Архитектура и реализация агента

Основными требованиями к реализации агента являются:

- Простота развертывания и использования агента на ресурсе;
- Функционирование за межсетевым экраном;
- Защита ресурса от несанкционированного использования и запуска вредоносного ПО;
- Выполнение вычислительных задач на ресурсе по запросу сервера;
- Сбор и передача информации о ресурсе;
- Отказоустойчивость.

Агент реализован на языке Python с поддержкой версий 2.6-2.8. Данный выбор обусловлен кроссплатформенностью языка Python, его повсеместной доступностью и наличием на многих ресурсах. Для установки агента достаточно скопировать его в какую-либо директорию, указать желаемые значения конфигурационных параметров в соответствующем файле и запустить скрипт start.sh. Права суперпользователя при этом не требуются. Богатая стандартная библиотека языка позволяет свести к минимуму сторонние зависимости. Единственной необходимой сторонней библиотекой является ws4py[4], используемая для установления соединения по протоколу WebSocket. Всё это обеспечивает легкость установки и использования агента на ресурсе.

Разработанный протокол позволяет агенту функционировать, имея в распоряжении только исходящую коннективность по стандартному порту HTTP или HTTPS. Дополнительные средства обхода межсетевых экранов агенту не требуются.

Агент реализован в парадигме объектно-ориентированного программирования, с учетом специфики реализации многопоточных приложений на языке Python. Архитектура агента изображена на рисунке 1.

Синие прямоугольники на рисунке обозначают потоки и процессы агента. Серыми прямоугольниками обозначены внешние программы, с которыми взаимодействует агент. Зеленый и розовый прямоугольники — объекты межпроцессного взаимодействия.

Синими стрелками изображены сообщения, передаваемые по протоколу WebSocket, зелеными — внутренние сообщения, содержащие информацию о задаче, розовыми — внутренние сообщения, содержащие информацию о состоянии запущенной задачи, серыми — все прочие внутренние сообщения.

Потоки TaskWorker, TaskManager и Sender исполняют одноименные методы класса AgentProtocolClient, который является наследником класса WebSocketClient, предоставленного библиотекой ws4py.

Поток AgentProtocolClient является основным потоком отвечающим за WebSocket-соединение, именно в нем происходит инициализация приложения, считывание конфигурации из файла, авторизация на сервере при запуске (метод opened), ожидание поступающих от сервера сообщений и их обработка (метод received_message).

При поступлении сообщения TASK_SUBMIT, AgentProtocolClient помещает задание в очередь TaskQueue, за состоянием которой следит поток TaskManager. Этот поток, если количество заданий не превышает заданное конфигурацией агента максимальное, извлекает это задание из очереди и создает для него новый поток TaskWorker.

Поток TaskWorker, принимает задание и выполняет все необходимые действия для его исполнения:

- 1) Скачивает и распаковывает входные данные с сервера;
- 2) Проверяет задание на соответствие политике ресурса и запускает его (метод submit соответствующего TaskHandler'a);
- 3) Отслеживает ход исполнения задания (метод get_state соответствующего TaskHandler'a);
- 4) Проверяет наличие всех требуемых выходных данных, упаковывает их и отправляет на сервер.

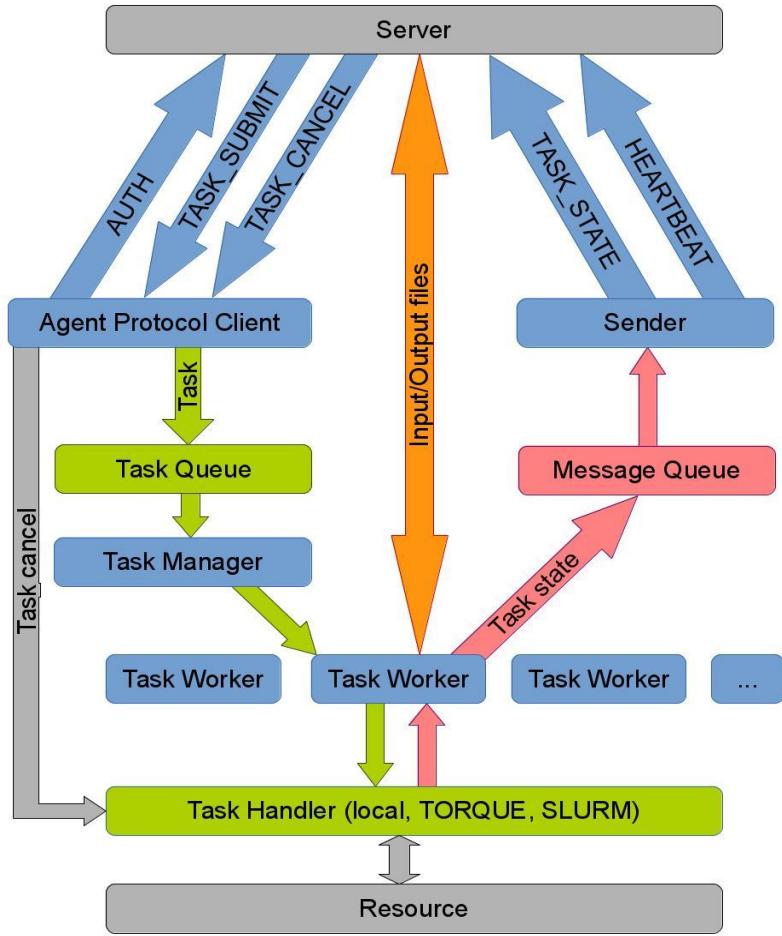


Рис. 1 – Архитектура агента.

В процессе работы исполнение каждого шага сопровождается помещением сообщения с соответствующим шагу STATE-значением в очередь MessageQueue.

Все сообщения поступающие в очередь извлекаются из неё потоком Sender и отправляются на сервер. Данный функционал был вынесен в отдельный поток, поскольку в классе `WebSocketClient` нет встроенной защиты от одновременного доступа к ресурсам, и в случае одновременной отправки сообщения двумя и более потоками поведение клиента непредсказуемо, и возможны повреждение и потеря передаваемых данных. Чтобы этого не произошло, все исходящие сообщения агрегируются в очередь `MessageQueue` и отправляются по одному. Помимо этого Sender собирает информацию о системе для передачи её в HEARTBEAT сообщениях и исполняет роль сторожевого таймера, отслеживая возможность передачи сообщений и реинициализируя `WebSocket`-соединение (метод `Reconnect`) в случае необходимости, что обеспечивает отказоустойчивость приложения при нарушениях связи с сервером.

Реализации абстрактного класса `TaskHandler`, обеспечивают потокобезопасные версии следующих методов:

- `submit` — метод, отвечающий за преобразование задачи в формат, понятный ресурсу, и её запуск. Возвращает некий идентификатор, присвоенный задаче ресурсом (например, PID, для `LocalTaskHandler`);
- `get_state` — возвращает состояние задачи, принимает на вход идентификатор, полученный при вызове метода `submit`;
- `cancel` — отменяет выполнение задачи, принимает на вход идентификатор, полученный при вызове метода `submit`;
- `get_slots` — возвращает максимально допустимое количество одновременно выполняемых задач на ресурсе (используется для получения информации в `RESOURCE_INFO`);
- `get_task_stat` — возвращает дополнительную информацию о выполненной задаче, такую, например, как время выполнения, использованные ресурсы и т. п. (Передается в сообщении `TASK_STATE` и может быть использована на сервере для учета потребленных пользователем ресурсов).

Основным средством обеспечения безопасности является список разрешенных команд (так называемый «белый список»). Команда каждой входящей задачи проверяется на соответствие списку регулярных выражений, которые позволяют максимально гибко настроить не только допустимые приложения, но и допустимые аргументы для каждого из них. Помимо этого запуск заданий реализован без поддержки оболочки командной строки, что делает невозможным атаки типа `shell injection`. Все операции с файлами реализованы таким образом, чтобы ограничить возможность задания считать или записать данные вне обозначенной для него рабочей директории. Таким образом, возможность запустить вредоносную команду через данный агент присутствует только в том случае, если данная команда в явном виде присутствует в списке разрешенных команд.

4. Интеграция с различными типами ресурсов

В настоящий момент агент поддерживает работу с тремя типами ресурсов, каждому из которых соответствует отдельная реализация описанного ранее класса `TaskHandler`.

`LocalTaskHandler` представляет собой простую систему запуска задач на том компьютере, на котором запущен агент. Предназначен для работы в Unix-подобных системах и дополнительно к основному объекту реализует оболочку над процессом, которая запускает процесс, возвращает присвоенный ему операционной системой PID, перехватывает стандартные потоки вывода команды и направляет их в файлы. После завершения задачи возвращает код завершения, что позволяет установить корректность выполнения программы.

`TorqueTaskHandler` представляет собой адаптер для запуска задач на вычислительном кластере под управлением системы очередей TORQUE [5]. Запуск задач, отслеживание их состояния и отмена осуществляются с помощью стандартных команд TORQUE (`qsub`, `qstat`, `qdel` соответственно). В качестве идентификатора задачи используется идентификатор, полученный от системы TORQUE в результате выполнения команды `qsub`. В настоящий момент это единственный `TaskHandler` возвращающий полезную информацию при вызове метода `get_task_stat`. В силу

особенностей реализации пока невозможно указывать желаемое количество ядер для приложения, запускаемого на кластере (невозможно средствами платформы установить значения nodes и ppn передаваемые команде qsub), что делает агент неэффективным для запуска параллельных приложений.

SlurmTaskHandler предназначен для запуска задач на вычислительном кластере под управлением системы очередей SLURM [6]. Запуск задач, отслеживание их состояния и отмена осуществляются с помощью стандартных команд SLURM (sbatch, scontrol, scancel соответственно). В качестве идентификатора задачи используется идентификатор, полученный от системы SLURM в результате выполнения команды sbatch. В силу особенностей реализации пока невозможно указывать желаемое количество ядер для приложения, запускаемого на кластере, что делает агент неэффективным для запуска параллельных приложений.

5. Заключение

Реализован рабочий прототип агента доступа к вычислительным ресурсам в рамках облачной платформы Everest, удовлетворяющий основным требованиям по функциональности и безопасности.

В настоящий момент агент используется для проверки выполнения студентами практических заданий по курсу «Параллельные и распределенные вычисления». Для этого на кластере развернуты программы тестирования, компилирующие и запускающие решения студентов на тестовых наборах данных. Студенты передают свои решения на ресурс через веб-интерфейс платформы, получают результаты работы и, если эти результаты удовлетворительны, передают ссылку на них преподавателю, который выставляет оценку. Результат работы программы хранится на платформе, что не позволяет его подменить или исправить. Это позволяет тестировать программы студентов быстро, надежно и удобно, как для студентов, так и для преподавателей.

В дальнейшем планируется расширять функционал агента в сторону более широкого взаимодействия с сервером, включая передачу информации о загрузке ресурса, трансляцию выходных данных задачи на сервер во время выполнения задачи и получение дополнительных параметров, специфичных для различных типов ресурсов.

Кроме этого планируется также реализовать поддержку других типов ресурсов, таких как грид-инфраструктуры, ресурсы под управлением ОС Windows и пристаивающие персональные компьютеры.

Литература

1. Sukhoroslov O., Afanasiev A.. Everest: A Cloud Platform for Computational Web Services // 4th International Conference on Cloud Computing and Services Science (CLOSER 2014)
2. Fette, I., Melnikov A., "The WebSocket Protocol", RFC 6455, December 2011.
3. Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, March 2014.
4. ws4py by Sylvain Hellegouarch, <https://github.com/Lawouach/WebSocket-for-Python>
5. TORQUE Resource Manager at Adaptive Computing, Inc. <http://www.adaptivecomputing.com/products/open-source/torque/>
6. Yoo, A. B., Jette, M. A., & Grondona, M. (2003, January). SLURM: Simple linux utility for resource management. In *Job Scheduling Strategies for Parallel Processing* (pp. 44-60). Springer Berlin Heidelberg.

МАТЕМАТИЧЕСКИЙ АППАРАТ И АРХИТЕКТУРА СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ МАИ CLASS.NET

Джумурат А.С.

Московский авиационный институт (национальный исследовательский университет),
Москва

Аннотация

В статье рассматривается архитектура системы дистанционного обучения МАИ CLASS.NET [1] и математический аппарат, который обеспечивает функциональные возможности системы. Формулируются основные идеи адаптивного компьютерного тестирования и отражение этих идей в математическом обеспечении СДО МАИ CLASS.NET. Описаны связи между программными компонентами системы и математическими моделями, которые им соответствуют. Сформулированы прикладные задачи, которые возникают в процессе дистанционного обучения, а так же описаны пути их решения в рамках архитектуры системы.

1. Введение

Данная статья посвящена проблеме разработки и использования математических моделей для развития систем дистанционного обучения (СДО). СДО — это программный комплекс, обеспечивающий удалённый доступ к учебно-методическим материалам и прохождение онлайн форм контроля для. Одна из важных задач, которая стоит перед разработчиками СДО - выбор подходящей архитектуры системы, которая позволила бы применять такие математические модели на практике без ущерба для производительности и надёжности системы.

Основной задачей образовательного процесса является подбор тестовых заданий для различных форм контроля. Выбор задач должен основываться на индивидуальных способностях каждого обучающегося. Очевидно, что если задания теста слишком простые, то обучающийся решит их без приложения усилий, а слишком сложные задания с большой вероятностью будут оставлены без ответа, так как студент с большой вероятностью пропустит их во время прохождения теста. В обоих случаях задачи имеют низкий развивающий потенциал и не способствуют повышению уровня подготовки обучающегося.

Одним из путей решения этой проблемы является развитие методов адаптивного компьютерного тестирования (АКТ). АКТ — это алгоритмы и методы формирования тестов, основанные на индивидуальных способностях обучающегося. АКТ предполагает разработку и развитие математических моделей, которые на основании истории об успешных и неуспешных ответов на предыдущие задания, позволяют определить уровень подготовки студента и исходя из этой информации выбрать для дальнейшего тестирования те задачи из базы, которые наиболее подходят данному обучающемуся.

2. Архитектура системы

С архитектурной точки зрения СДО МАИ CLASS.NET представляет собой распределённое клиент-серверное приложение. Распределённая архитектура системы обеспечивает возможность обработки большого количества запросов от пользователей, так как нагрузку можно равномерно распределять между кластером из серверов. Так же

снижению нагрузки на каждый сервер системы способствует тот факт, что благодаря клиент-серверной архитектуре часть вычислений производится на стороне клиента.

Архитектура системы представлена на рисунке:

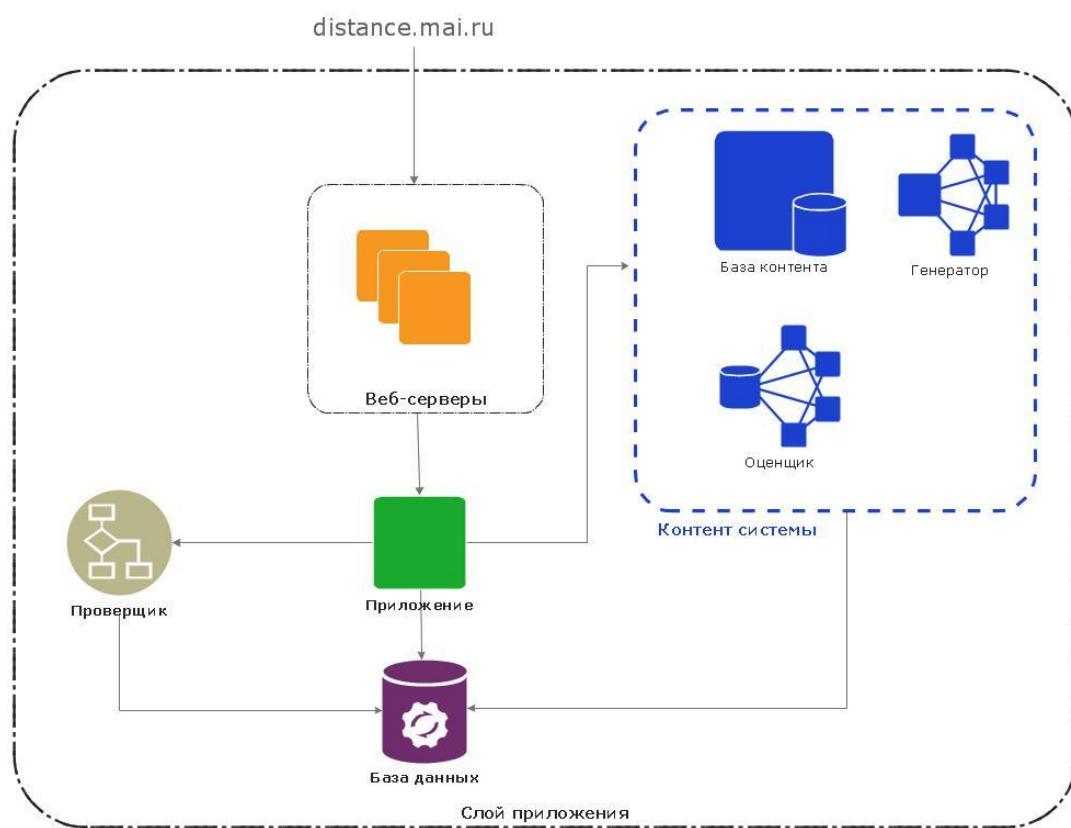


Рис. 1 – Архитектура СДО МА CLASS.NET.

Схема системы дистанционного обучения CLASS.NET отражает устройство серверной части приложения (на стороне клиента происходит генерация форм ввода ответа и элементов интерфейса). В системе разделяется формирование статического и динамического контента. Статические элементы страницы (CSS-файлы, js-файлы, изображения и т.д.) хранятся на стороннем сервере. Это снижает нагрузку на систему генерации динамического контента – заданий, теоретических материалов курсов и статистической информации пользователя (на схеме генерация динамического контента происходит в структурном элементе «Слой приложения»). Слой приложения включает базу данных пользователей (используется СУБД MySQL), которая содержит информацию о решённых задачах и идентификационные данные пользователя. База контента представляет собой хранилище XML файлов с учебно-методическими материалами: задачами и теоретическим материалом – это позволяет разделить контент между несколькими серверами для снижения нагрузки. Слой приложения содержит следующие компоненты: «Проверщик», «Генератор» и «Оценщик». Рассмотрим подробнее каждый компонент системы.

«Проверщик» - программный компонент, который производит обработку ответов пользователя. В СДО применяются различные формы ввода ответов («текст»,

«одиночный выбор», «матрица» и т.д.), используя разнообразные алгоритмы для проверки ответа. В ряде задач (например, нахождение обратной матрицы), правильный ответ на задания не хранится в системе, а вычисляется на основании данных задачи – это исключает возможность получить готовые ответы при взломе системы. Вычисление правильного ответа в этом случае как раз и производит «Проверщик». По результатам правильных и неправильных ответов всех пользователей системы формируется рейтинг каждого пользователя [2].

«Генератор» - программный компонент системы, который по информации из базы контента формулирует условия задачи и формирует теоретический материал курса путём преобразования XML файлов из базы контента в HTML-документы для отображения в браузере. СДО МАИ CLASS.NET производит адаптацию задач системы в том числе с использованием случайных параметров. Подстановку реализации случайных параметров в условие задачи производит «Генератор». Также «Генератор» производит объединение задач, которые содержатся в базе контента, в индивидуальные варианты, предназначенные для проведения рубежного и итогового контроля [4].

«Оценщик» представляет собой программную реализацию математических алгоритмов оценки сложности заданий, уровня знаний пользователя [3] и других алгоритмов адаптивного компьютерного тестирования на основании статистической информации о действиях пользователей, накопленной в реляционной базе данных.

3. Математическое обеспечение

Архитектура системы позволяет применять в СДО идеи адаптивного компьютерного тестирования. Адаптация системы дистанционного обучения происходит путём обработки статистической информации о работе пользователей, которая хранится в базе данных: количество решённых задач из числа предложенных, время ответа на задания и т.д. На основании анализа статистической информации разработаны алгоритмы для решения ряда задач дистанционного обучения: автоматическое формирование рейтинга студентов [2], оценивание уровня подготовки каждого студента [3], формирование набора индивидуальных заданий из общей базы заданий [4]. В настоящее время ведётся работа над созданием семейства алгоритмов, которые используют для адаптации системы прогнозное время ответа студента на задачу.

Одним из факторов эффективного процесса обучения является ограничение общего времени на тест. Очевидно, что скорость ответа пользователя представляет собой результат сложных когнитивных процессов и является важным источником информации для адаптации системы. Если никак не ограничивать время тестирования, то два студента с одинаковым уровнем подготовки наберут равное количество баллов. Однако обычно время теста ограничено – поэтому более «медленный» студент не всегда успевает выполнить все задания, что приводит к низкой оценке. Таким образом, построение математической модели времени ответа является одной из важных задач АКТ. В связи с этим было принято решение о разработке семейства алгоритмов адаптации СДО CLASS.NET, основанных на статистической информации о времени, которое студенты затрачивают для ответа на задания во время тестирования.

В качестве математической модели времени ответа была выбрана модель, предложенная голландским учёным Вимом ван дер Линденом (Wim van der Linden)[5]. Согласно этой модели, логарифм времени ответа T_{ij} пользователя j на задачу i имеет нормальное распределение.

$$\ln T_{ij} = \mu + \beta_i + \tau_j + \varepsilon_{ij};$$

$$\sum_{i=1}^K \beta_i = 0, \sum_{j=1}^M \tau_j = 0; \varepsilon_{ij} \sim N(0, \sigma^2), \quad i = 1, \dots, K; j = 1, \dots, M$$

где

μ – временной параметр, общий по всем задачам в базе задач и всем пользователям;

β_i – временной параметр, индивидуальный для задачи i ;

τ_j – временной параметра, индивидуальный для пользователя j ;

ε_{ij} – случайное отклонение;

σ^2 – дисперсия случайного отклонения;

M – количество пользователей, зарегистрированных в СДО;

K – общее количество задач в базе.

Эта математическая модель позволяет строить прогноз времени, которое затратит пользователь для ответа на предложенную задачу теста. Логарифм времени ответа является случайной величиной и включает несколько структурных компонентов: общий параметр μ , индивидуальные параметры задачи и студента (β_i и τ_j) и случайное отклонение, которое имеет нормальное распределение. Параметры модели являются неизвестными и подлежат оценке, например, методом наименьших квадратов. Для оценки параметров используется статистическая информация, накопленная СДО.

Одним из важных приложений этой модели является генерация вариантов заданной сложности в условиях ограниченных по времени тестов. Обычно проведение текущего и рубежного контроля ограничено по времени – в связи с этим возникает необходимость подбора задач для тестирования таким образом, чтобы индивидуальное задание имело фиксированный уровень сложности и пользователь при этом смог бы уложиться в отведённое на тест время. Эта проблема приводит к следующей задаче стохастического программирования с вероятностным ограничением:

$$X_* = \underset{x}{\operatorname{Argmin}} |c - w^T x|$$

при наборе ограничений

$$e_n^T x = k$$

$$c - w^T x \leq \varepsilon$$

$$c - w^T x \geq -\varepsilon$$

$$P(T^T x \leq T_{\text{общ}}) \geq \alpha$$

$$x_i \in \{0, 1\}, i = 1, \dots, n$$

В приведённой задаче n – общее количество задач в базе, $e_n \in R^n$ – вектор из единиц размерности n , k – требуемое число задач в индивидуальном варианте, $w \in R^n$ – вектор, в котором w_i – сложность задачи i , c – требуемая суммарная сложность индивидуального задания, ε – допустимые границы изменения сложности, $T \in R^n$ – вектор, где t_i – случайная величина, которая является временем ответа на задачу i и α – заранее выбранный уровень доверительной вероятности. Так как логарифм времени ответа имеет нормальное распределение, то само время ответа имеет логнормальное распределение. Вероятностное ограничение представляет собой сумму случайных величин с известным распределением и для решения полученной задачи применяются известные методы линейного стохастического программирования – например, обобщённый минимаксный подход.

Предложенный алгоритм формирования индивидуальных заданий станет частью программного компонента «Генератор» и позволит формировать индивидуальные задания в условиях ограниченных по времени тестов. Приведённая задача формирует индивидуальный вариант для конкретного пользователя системы, однако её можно решать и для группы студентов – в этом случае будет сформирован вариант, который все студенты группы выполнят в течении заранее заданного времени. Такой вариант можно использовать для проведения оффлайн форм контроля (рубежного или итогового).

4. Заключение

Система дистанционного обучения включает алгоритмы управления учебно-методическими материалами, основанные на парадигме адаптивного компьютерного тестирования. Эффективную реализацию алгоритмов проверки правильности заданий, алгоритмов генерирования контента и алгоритмов оценивания уровня студентов обеспечивает распределённая клиент-серверная архитектура системы.

Распределённая архитектура системы позволяет поддерживать процесс одновременной работы большого количества пользователей – преподавателей и студентов, т.к. результирующая нагрузка на систему распределяется между несколькими серверами кластера. Кроме генерации учебного контента СДО производит обработку большого количества данных о пользователях, что дополнительно увеличивает нагрузку на сервер – в связи с этим часть функционала реализована на стороне клиента: отображение элементов интерфейса, генерация форм ввода ответа и т.д.

Приведённая математическая модель времени ответа пользователя используется для адаптации системы дистанционного обучения на основании обработки статистической информации о работе реальных пользователей в системе. В статье приведены пример прикладной задачи, которая решается с использованием данной модели: построение ограниченных по времени тестов заданной сложности. Применение математических методов для адаптации систем дистанционного обучения позволяет учитывать индивидуальные особенности пользователей и на основании этой информации проводить процесс обучения более эффективно.

Литература

1. СДО МАИ CLASS.NET distance.mai.ru/demo
2. Кизун А.И. Панарин С.И. Формирование интегрального рейтинга с помощью статистической обработки результатов тестов // Автоматика и Телемеханика, 2012, № 6, сс. 119-139.
3. Кизун А.И., Иноземцев А.О. Оценивание уровней сложности тестов на основе метода максимального правдоподобия // Автоматика и телемеханика, 2014, №4.
4. Наумов А.В., Иноземцев А.О. Алгоритм формирования индивидуальных заданий в системах дистанционного обучения // Вестник компьютерных и информационных технологий, 2013, №6, сс. 46-51.
5. Wim J. van der Linden, Using Response Times for Item Selection in Adaptive Testing // Journal of Educational and Behavioral Statistics, 2008, Vol. 33. № 7, сс. 5-20

ОБ УПРАВЛЕНИИ РАЗГРАНИЧЕНИЕМ ДОСТУПА В «ЕВФРАТ»

Тарханов И.А.

Институт системного анализа РАН

Москва, ivant@cs.isa.ru

Седов Д.А.

ООО «Когнитивные технологии»

Москва, d_sedov@cognitive.ru

Аннотация

В статье рассмотрены проблемы разграничения доступа в системах электронного документооборота (далее СЭД) и предложен метод решения этих проблем, основанный на логическом управлении доступом с помощью политик безопасности. Данный метод реализован в системе СЭД «ЕВФРАТ» и обеспечивает гибкое управление правами независимо от формальной модели разграничения доступа.

1. Проблема разграничения доступа в СЭД

В современной компьютерной безопасности существует ряд устоявшихся классические моделей разграничения доступа: дискретная, мандатная и ролевая модели [1,2, 7]. На практике системы разграничения доступом в СЭД это не классические системы в чистом виде, а некоторое их сочетание друг с другом. Классические модели недостаточно гибки, и не зависят от бизнес событий, происходящих в системе, что в контексте СЭД имеет очень важное значение. Приведём несколько типовых ситуаций:

- Необходимость временного наследования прав субъектов. Назначение временных заместителей пользователям, которые уходят в отпуск, с наследованием всех прав до определённого момента.
- Изменение набора прав у объекта в зависимости от состояний его обработки. Например, некоторое множество документов находится на контроле, и к ним применяется один и тот же набор правил доступа, при снятии с контроля другой, при возвращении на контроль третий.
- Наследование прав доступа к объекту по иерархии объектов. Например, доступ к изменению версии файлов наследуется от основного файла, т.е. есть у автора файла и у контролёра документа и отсутствует у автора другого файла.

Ещё одним существенным минусом классических моделей при применении их в современных сложных системах, какими являются СЭД, является обязательное определение "типовых" действий с объектами защиты [2, 3, 4, 6, 7, 8]. Это создание, уничтожение, чтение и модификация объекта. Часто отдельно выделяется изменение прав доступа к объекту. В СЭД таких действий намного больше. Более того, системы такого класса характеризуются большой вариативностью бизнес логики для разных заказчиков [5]. Тем самым, конечное множество действий с объектами защиты невозможно определить заранее. Анализ безопасности СЭД ЕВФРАТ выявил более 30-ти типов защищаемых объектов и более 100 типов состояний субъектов, влияющих на разграничение прав (ролей, групп доступа, системных привилегий и т.д.)

В современной теории активно развиваются семейство формальных моделей

безопасности на основе моделей логического управления доступом (ДП моделей), которые в той или иной степени обеспечивают решение поставленных прикладных проблем [2]. Формальная запись таких моделей очень громоздка и их основное предназначение — это выявление возможных угроз с помощью теоретического анализа данной модели. Для решения проблем СЭД необходимы гибкие инструменты управления доступом с помощью политик безопасности [6, 9, 10]. В данной статье приводиться модель политики безопасности в СЭД ЕВФРАТ и особенности её реализации.

2. Политика безопасности в СЭД

Определим политику безопасности, используя стандартные определения из теории информационной безопасности [2] и используя модульный подход для управлением политиками [10].

Политикой безопасности при модульном подходе определяется возможность выполнения субъектом s метода a над защищаемым объектом o . В системах СЭД под защищаемым объектом, как правило, понимается атрибуты документа, бизнес-процесс по документу, приложенные к нему файлы. Под субъектами — пользователи системы, внешние сервисы и системы, т.е. то, что может установить связь с программной средой защищаемых объектов, изолированной слоем безопасности.

Правилом авторизации назовём тройку (s, o, a) , где $s \in S, o \in O, a \in A$. S, O, A , соответственно, множества всех субъектов, объектов и методов объектов, вызываемых субъектами.

Политикой доступа назовём конечный набор базовых правил авторизации. Ясно, что правила в одной политике не должны быть противоречивы друг другу и применяется строго декларированный подход в определении правил. Иными словами, если правило доступа конкретного метода объекта к субъекту не определено, то доступ запрещен.

Помимо этого, для описания логических правил в политиках используются бинарные операторы над элементами множеств O, A, S : $(s \text{ op } s_o), (o \text{ op } o_o), (a \text{ op } a_o)$. К примеру, сравнение атрибутов документа друг с другом $(o \geq o_o)$. Для извлечения простых свойств и преобразований используются предикаты $p(s), p(o), p(a)$. Примеры предикатов — извлечения назначенных на s ролей в случае ролевого доступа, или получения метки конкретной метки при дискретном доступе для a .

Далее для описания гибких проверок внутри правил авторизации необходимо ввести понятие правила, как логической функции:

$$\varphi(s, o, a): L_s, L_o, L_a \rightarrow \begin{cases} \text{true} \\ \text{false} \end{cases},$$

где L_s, L_o, L_a это все возможные наборы бинарных операторов и предикатов над соответствующими объектами. Результатом правила всегда будет ответ, возможен доступ или нет.

Представив политики в виде набора правил, мы сможем обеспечить гибкость для конкретных наборов (s, o, a) . С помощью политики удобно описывать одно состояние системы для определённого множества документов. В системе, в которой для разных типов документов нужно использовать разные схемы доступа, необходимо применение не одной, а наборов политик. Следовательно, возникает вопрос, применения сразу нескольких политик и определения базовых операций с политиками. В зарубежной литературе используется около 9 таких операций [10]. Проведённый авторами анализ реальных проблем при разграничении доступа в СЭД ЕВФРАТ показал, что достаточно

4-х:

$P \wedge a_0$

Сложение (+). Формально, это сложение всех правил двух политик.

$\llbracket [P]_1 + P_2 \rrbracket = \llbracket [P]_1 \rrbracket \cup \llbracket [P]_2 \rrbracket$

На практике, полезно произвести декомпозицию одной громоздкой политики на части, разделив их по описанию методов защищаемых объектов. Далее собирается общая политика сложением всех политик.

Наследование → . По аналогии с объектно-ориентированным программированием использует наследование одной политикой всех свойств родительской с возможностью добавления своих. В современных реализациях СЭД существует множество иерархических защищаемых объектов, наследующих права доступа у родителей. Приведём характерный сценарий. Что бы получить право редактирование свойства задачи, нужно сначала проверить права редактирования документа, затем процесса, запущенного по этому документу, затем задачи в этом процессе и после этого уже проверить непосредственно право редактирования свойства.

Перезапись o. Заменяет часть правил из P_1 , которые определяются множеством правил P_3 , на аналогичные из P_2 .

$o(P_1, P_2, P_3) = ((P_1 \setminus P_3) \cup (P_2 \cap P_3))$

На практике в системах управления доступом в СЭД, эта ситуация возникает при появлении дополнительной логики. Если в процессе запущено согласование со специальным свойством «утверждение», то исполнитель этой деятельности наследует все права, которой обладает группа утверждающих. В противном случае, он ограничен стандартным набором прав, как исполнитель согласования.

Ограничение ^ . Ограничение доступа по политике, согласно определённому методу доступа a.

$P \wedge a_1 \mathbf{0} = P \setminus \{s, o, a_1 l .. l \in A, l \neq o\}$

Например, при снятии документа с контроля необходимо любые действия с ним и связанные с ним объекты запретить, оставив только права на чтения.

Далее покажем, как описанные политики и операции над ним реализованы в системе разграничения доступом в СЭД ЕВФРАТ.

3. Реализация управления доступом в СЭД «ЕВФРАТ»

В СЭД «ЕВФРАТ» для описания правил политики используется формат xml.

```
<object name="document" description="Описание объекта документ">
    <actions>
        <action name="action1" description="Описание action"
priority="significant">
            roleAdmin() // имеют право все администраторы
        <action name="action2" description="Описание action2"
priority="important">
            if (confirmed) then // если объект утверждён
                (acl[DeleteConfirmed]) // необходимо явное право удаления
            else
                (true) // иначе выполнять действия могут все
    </actions>
```

</object>

В приведённом примере объект *document* имеет методы *action1* и *action2*, а в блоке *<actions>* описание логики доступа к этим методом. В частности, для *action1* используется проверка роли субъекта – обладает ли он ролью администратора по модели ролевого доступа. У *action2* более сложное условие вычисление условий доступа. Проверяется, обладает ли объект свойством “утверждено”, если да, то проверяется, обладает ли субъект явным дискреционным правом *DeleteConfirmed*, иначе доступ на выполнение метода предоставляется всем.

- Операция объединения политик реализуется объединением содержимого файлов xml разных политик перед непосредственно проверкой.
- Операция наследования реализуется вложением xml тэгов *<object>* в родительские или использованием ключевого атрибута *inherit* у *<object>*.
- Операция перезаписи реализуется атрибутом *override* тэга *<action>* у конкретного метода.
- Операция ограничение реализуется с помощью логических условий, либо применением атрибута *restrict* тэга *<object>*. Пример, *<object name=“document” restrict=“read[not control]”>*. Здесь при снятии с контроля объекта документ, остаётся только право на метод *read*.

Объекты бизнес логики веб-сервера «ЕВФРАТ» находятся внутри изолированной программной среды, для которой реализуется проверка доступа на каждый запрашиваемый метод к защищённому объекту. В процессе проверке идёт обращение к политикам, которые привязаны к определённому документу или множеству документов.

Гибкость условий обеспечивается логикой проверок и поддержкой 4-х операций над политиками. Для интеграции классических моделей разграничения доступа при описании условий проверок используются специальные предикаты.

- *Role()* – проверяет назначен ли субъект на роль или входит в специальную группу доступа. Используется для интеграции в описанную модель ролевой модели.
- *Acl[<право_доступа>]* - проверяет у текущего субъекта наличие именнованого права доступа к объекту по дискреционной модели.
- *Level[<название_уровня_доступа>]* – проверка наличия определённого уровня безопасности у субъекта для доступа к защищаемому объекту по мандатной модели доступа.

4. Заключение

Описанный в статье подход к логическому управлению доступом реализован в СЭД «ЕВФРАТ», которая внедрена более чем в 100 государственных и коммерческих организациях России и стран СНГ. Полученный при внедрении опыт позволил обобщить все специфические для систем электронного документооборота и управления делопроизводством проблемы разграничения доступа в рамках описанного подхода. Достоинством подхода является его универсальность при встраивании в СЭД с любой моделью разграничения доступа.

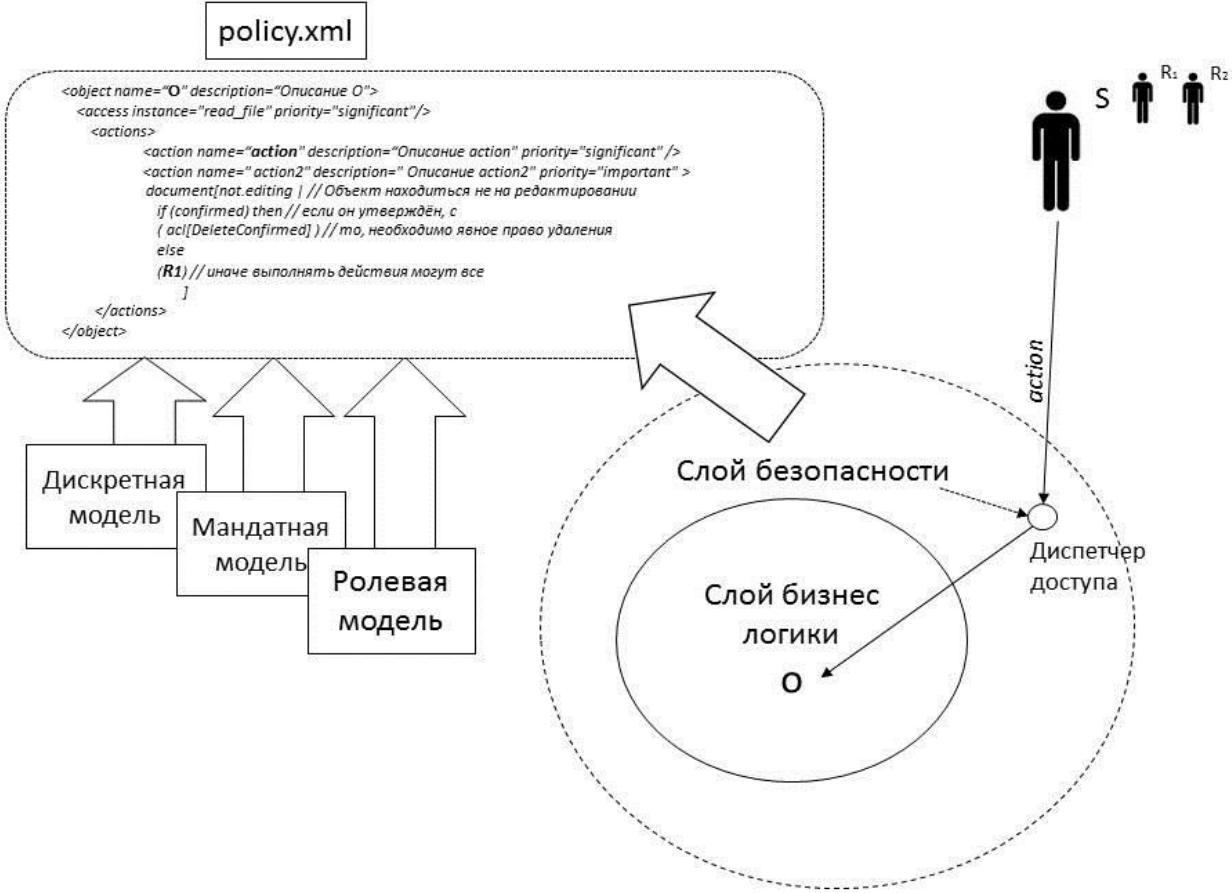


Рис. 1 – Управление разграничением доступа в «ЕВФРАТ».

В будущем предполагается провести анализ достаточности описанных операций над политиками для самых распространённых моделей управления доступом и реализовать пользовательский интерфейс для создания описаний политик.

Литература

1. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации. Руководящий документ ФСТЭК от 30 марта 1992 года / ФСТЭК. — 1992.
2. Девягин П.Н. Модели безопасности компьютерных систем: Учеб. пособие для студ. высш. учеб. заведений. М.: Издательский центр «Академия», 2005.
3. Даниленко А. Ю. Защита данных в сложных информационных системах. Труды ИСА РАН, 2007. Т. 29
4. Стефанцов Д. А. Реализация политик безопасности в компьютерных системах с помощью аспектно-ориентированного программирования. Прикладная дискретная математика Выпуск № 1 / 2008
5. Лухин П.А., Романов Б.Л., Слободецкий Д.Я., Тарханов И.А. Быстрая кастомизация программных продуктов на основе коробочного решения. Труды Института системного анализа Российской академии наук. 2009. Т. 45. С. 95-106.
6. E. Bertino, B. Catania, E. Ferrari, P. Perlasca. A logical framework for reasoning about access control models. // ACM Trans. Inf. Syst. Secur., vol. 6, 2003. — pp. 71-127.
7. K. J. Biba. Integrity Considerations for Secure Computer Systems. — MTR-3153, The Mitre Corporation, April 1977.

8. R. Chandramouli, R. Sandhu. Role Based Access Control Features in Commercial Database Management Systems. // Proceedings of 21st National Information Systems Security Conference,' October 6-9, 1998, Crystal City, Virginia.
9. M. Koch, F. Parisi-Presicce. Describing Policies with Graph Constraints and Rules. // In Proc. ICGT02. Springer-Verlag, 2002. - pp. 223-238.
10. P. Bonatti, S. de Capitani di Vimercati, and P. Samarati. A Modular Approach to Composing Access Control Policies. In Proc. of 7th ACM Conference on Computer and Communication Security, pages 164–173. ACM, 2000.
11. R. Sandhu. Role-Based Access Control. // Advances in Computers, Vol.46. — Academic Press 1998.

О НЕОБХОДИМОСТИ И СПОСОБЕ УЛУЧШЕНИЯ ПРОЦЕССА СТАБИЛИЗАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Югов Александр Сергеевич

НИУ ВШЭ - Пермь

Г. Пермь, Yugov@mail.ru

Аннотация

С увеличением сложности программных продуктов увеличивается количество ошибок, которые они содержат. Ошибки могут нести как практически незаметные потери участникам рынка, так и иметь катастрофические последствия.

Интерес к данной сфере обуславливается еще и стремлением компаний к увеличению качества создаваемых программных продуктов в условиях острой конкуренции.

Навигатор процесса стабилизации предназначен для управления процессом стабилизации (комплексного тестирования) сложного программного продукта (ПП). Он позволяет определить наилучший (по заданным критериям) порядок тестирования и оценить «степень проверенности» продукта. Тем самым служит инструментом повышения качества производимого ПО.

1. Введение

Программное обеспечение стало неотъемлемой частью нашего существования. Фактически провал или успех наших повседневных дел в той или иной степени определяется корректностью функционирования ПО, что ставит современное общество в зависимость от программных продуктов.

Более того, количество разрабатываемых программ с каждым годом неизменно увеличивается. По оценкам IDC¹ (аналитическая фирма, специализирующаяся на исследованиях рынка ИТ) объем российского рынка разработки заказного ПО составил в 2011 г. – 1,61 млрд.долл., в 2012 г. – 1,73 млрд.долл. В 2013 - 2017 годы совокупный среднегодовой рост рынка оценивается аналитиками IDC примерно в 6%.

В такой ситуации предъявляются повышенные требования к качеству ПО. Один из возможных способов улучшения качества ПО – совершенствование процесса тестирования (в частности, с учетом конфигурации будущей среды исполнения программы).

¹ IDC Russia Vertical Markets 2013-2017 IT Spending Forecast

В данной работе рассматривается сфера тестирования ПО и приводится описание идеи создания инструмента, который, во-первых, является инструментом составления плана тестирования комплексных программных продуктов, во-вторых, инструментом для обучения тестированию.

2. Стоимость ошибки

Зачем нужно тестировать ПО? Для компании, создающей программные продукты, ошибка в программе – это экономические потери:

1) расходы компании на тестирование и отладку программы. Расходы такого типа поддаются приблизительной оценке;

2) убытки, которые несет компания от того, что ошибка не была найдена вовремя. В большинстве случаев оценить потери от ошибок такого типа невозможно.

В первом случае, если программа содержала ошибку, и эта ошибка была обнаружена в процессе тестирования, стоимость ошибки вычисляется как суммарные затраты следующих составляющих [1]

1) оплата времени работы специалиста по тестированию;

2) оплата времени работы программиста, который реализовал модуль, содержащий ошибку, а затем исправил ее;

3) Стоимость реализации «неправильного» модуля;

4) Стоимость тестирования «неправильного» модуля;

5) Стоимость исправления ошибки и проблем, появившихся из-за ее наличия.

Очевидно, слагаемые поддаются приблизительной оценке.

Если ошибка была обнаружена пользователем, то к стоимости, вычисляемой по формуле предыдущего случая, могут прибавиться другие убытки, например:

1) время службы поддержки;

2) компенсации пользователю понесенных затрат;

3) иски против компании;

4) утраченная потенциальная оплата услуг компании ушедшими пользователями и пользователями, которые могли бы стать клиентами компании, но не стали ими из-за ухудшения репутации.

Такие убытки, как правило, не поддаются объективной оценке.

Но от «неправильных» программ страдают не только производители, но и пользователи, а иногда даже люди, которые не имеют совершенно никакого отношения к программе.

Часто в качестве примера ПО с наиболее губительными последствиями эксплуатации приводятся катастрофа Ariane 5 и инциденты с Therac-25. Ariane 5 – ракета-носитель, запуск которой был произведен 4 июня 1996 года. Полет продлился неполные 40 сек., после чего произошел взрыв [2].

Из-за ошибок в программной части медицинского комплекса Therac-25 минимум шесть человек получили передозировки радиации, некоторые пациенты получили дозы в десятки тысяч рад. Как минимум двое умерли непосредственно от передозировок. [3]

В истории РФ также были примеры катастрофических последствий ошибок в программном обеспечении. Например, 5 декабря 2010 года три спутника, критически важные для завершения составления группировки российской навигационной системы ГЛОНАСС – упали в Тихий океан недалеко от Гавайских островов вскоре после их запуска ракетой «Протон-М». Финансовые потери оцениваются в 4 миллиарда рублей. В результате расследования виной аварии была признана ошибка в программировании,

которая привела к тому, что в ракету залили неправильное количество топлива. [4]

Такие глобальные катастрофы являются редкими, чаще потери от ошибок носят исключительно финансовый характер.

Однако, важна не только цена одной ошибки, но и их количество (именно эти показатели составляют понятие стоимости ошибки). Согласно отчету компании WorkSoft: «2013 Trends in Automated Testing» [5] почти половина (45,1%) ошибок остаются незамеченными специалистами по тестированию.

Более того, из 594 респондентов, участвовавших в исследовании Worksoft, 204 компаний (34,3%) указали, что в 2013 году планируется увеличение инвестиций в автоматизацию тестирования и обеспечение качества. Только четверть респондентов ответили, что не планируют дополнительных инвестиций. Остальные ответившие не знают о планах в отношении инвестиций в автоматизацию тестирования.

Учитывая большое число не определившихся с выбором (41, 1%), можно предположить, что некоторые из этих компаний всё же планируют дополнительные инвестиции. Это значит, что настояще число фирм, планирующих увеличение инвестиций в автоматизацию тестирования в 2013 году, скорее всего больше 34%. [6]

Результаты данного исследования показывают, что текущий уровень тестирования не соответствует требованиям рынка. Компании все больше инвестируют в развитие надежности продуктов, а значит, и в развитие тестирования.

3. Рынок инструментов тестирования

Рынок инструментов тестирования, в основном, состоит из систем автоматического тестирования.

На рынке присутствуют программные продукты трех видов:

- 1) коммерческий продукт (WinRunner, Rational Robot, SilkTest, TestComplete и др.);
- 2) бесплатные и условно-бесплатные инструменты (freeware, shareware);
- 3) собственные утилиты компаний (написанные в ходе работы над проектами).

Рассмотрим недостатки и преимущества продуктов этих видов.

Коммерческие продукты:

Преимущества:

1. В автоматизированных тестах можно использовать функциональность (модули, процедуры, куски кода), которая идет в поставке продукта. В случае, если для тестирования приложения необходимо задействовать подобные методы, этот код не придется писать самому.

2. Как правило, такие инструменты поставляются со своей собственной средой разработки, которая предоставляет хорошие возможности для написания и отладки автотестов.

Недостатки:

1. Высокая стоимость таких продуктов.
2. Нет доступа к исходному коду.

2.1 Если в инструменте обнаруживается ошибка, то его исправление силами поставщика может занять существенное время.

2.2. Нет возможности оперативно настраивать продукт под нужды компании.

3. Зачастую коммерческие продукты навязывают свои подходы к тестированию приложений - в соответствии с той моделью, по которой построен данный инструмент. Такой подход не всегда приемлем для отдельно взятых приложений в силу их специфики, устоявшихся практик, человеческих и машинных ресурсов.

Собственные инструменты:

Преимущества:

1. Иногда такие инструменты являются побочным результатом создания приложения, особенно в случае TDD (достались бесплатно).

2. О них известно “от и до”, их достаточно легко использовать.

3. Доступен исходный код.

Недостатки:

1. Функциональность таких утилит может быть недостаточной для организации автоматизированных тестов.

2. Доработка до полноценной системы тестирования и её поддержка может быть дорогостоящей.

Бесплатные и условно-бесплатные инструменты:

Преимущества:

1. Разумная стоимость владения. Цена shareware-продукта зачастую невысока. В случае freeware продукт предлагается бесплатно.

2. Просьбы об исправлении багов и расширении функциональности, как правило, находят понимание и оперативный отклик - порой все так же бесплатно либо за скромную плату. В последнем случае выделенные деньги напрямую расходуются на решение тех задач, с которыми столкнулся пользователь инструмента, а не на разработку расширений, которые ему не нужны.

3. За частую у подобных утилит доступен исходный код.

Недостатки:

1. Функциональность таких продуктов может оказаться недостаточной для решения поставленных задач.

2. Существует риск остаться без поддержки со стороны разработчика продукта.

Безусловно, список преимуществ и недостатков представленных подходов далеко не полный и допускает исключения. Скажем, коммерческие продукты могут иметь разумную цену и быть достаточно гибкими, а собственные инструменты могут оказаться достаточно надежными. У коммерческого продукта может быть неприемлемо продолжительный период освоения, а условно-бесплатный продукт может навязывать свой язык программирования. Бывает и так, что бесплатный продукт оказывается более эффективным и удобным в использовании, чем коммерческие продукты. [7]

Сравнение наиболее распространенных на рынке продуктов представлено в таблице 1.

Как видно из сравнительной таблицы, инструменты для тестирования покрывают такие области как функциональное и нагрузочное тестирование, отслеживают качество кода, предлагают способы управления наборами тестов.

Однако, далеко не всегда получается провести абсолютно полное тестирование. В таком случае необходимо выделять, на каких элементах сосредоточиться с целью проведения тщательной проверки, а какие можно (пока) оставить, т.е. провести минимальную проверку.

Таблица 1

Сравнение продуктов автоматизированного тестирования

Разработчик	Продукт	Функциональное тестирование	Нагрузочное тестирование	Качество кода	Управление тестами
IBM	Rational Robot	+	+	+	+
Borland	SilkTest	+	+	-	+
AutomatedQA	TestComplete	+	+	-	+
HP	WinRunner	+	+	+	+
<i>Open-source</i>		Abbot, Selenium, Watir	Grinder, Jmeter, OpenSTA	GCT, NCover, Cobertura	FitNesse, TestLink

4. Навигатор процесса стабилизации

Предлагаемое решение повышения эффективности процесса тестирования – создание навигатора процесса стабилизации (НПС). НПС предназначен для управления процессом стабилизации (комплексного тестирования) сложного программного продукта (ПП). Он позволяет определить наилучший (по заданным критериям) порядок тестирования и оценить «степень проверенности» продукта.

Ранее основным понятием было тестирование. Понятие «стабилизация» отличается от понятия «тестирование» тем, что при стабилизации основное внимание фокусируется на эксплуатации создаваемого программного продукта в реалистичной модели производственной среды.

Основная идея – представить тестируемый программный продукт в виде многомерного пространства («program product space» – PPS). Измерения пространства характеризуют многообразие применения стабилизируемой программы: измерение «Функции», измерение «Операционные системы», измерение «Сервис паки», измерение «Форматы хранения данных», измерения «Производительность», «Нагрузка», «Интерфейс», «Юзабилити», «Документация», «Требования к процессору», «Требования к памяти», «Требования к месту на диске» и т.д. Возможно деление на более мелкие измерения или наоборот их укрупнение («Кроссплатформенность», «Конфигурация аппаратуры»). Конкретный набор измерений уточняется для каждого конкретного продукта. Точки этого пространства задают некоторые конкретные ситуации, которые должны быть проверены.

Каждая точка пространства характеризуется рядом параметров:

- 1) частота использования (для отделения часто используемых частей программы от редко используемых);
- 2) обязательность использования (для выделения обязательных сценариев);
- 3) стоимость исправления ошибки, связанной с этим компонентом;
- 4) стоимость неисправленной ошибки для пользователя (стоимость проявления ошибки на этапе эксплуатации программы);
- 5) важность исправления ошибки с точки зрения текущей версии программы

(можно ли оставить эту ошибку неисправленной до следующей версии);

6) оценка вероятности ошибки по данному измерению или для данного значения (возможна на основе предыдущего опыта работы; может быть грубой: низкая – средняя – высокая);

7) оценка серьезности ошибки (для отделения фатальной ошибки в работе программы от орфографической в руководстве пользователя).

Значение каждого параметра лежит в некотором диапазоне (например, от 0 до 10). Диапазон может быть настраиваемым для каждого продукта.

Оценки параметров для разных измерений и значений по тому или иному измерению задает пользователь. Параметры, не указанные пользователем, будут иметь некие стандартные значения (0 или 1 в зависимости от аддитивности или мультипликативности итоговой оценки). Возможно изменение стандартного значения по желанию пользователя.

По значениям параметров вычисляется приоритетная оценка каждой точки «пространства программы» (program space). Формула для вычислений предлагается по умолчанию (взвешенная сумма или взвешенное произведение), но может быть модифицирована пользователем (изменены веса значений параметров или задана совершенно иная формула).

После этого Навигатор «прокладывает маршрут тестирования» по пространству программы: определяет наборы наиболее приоритетных «точек», которые должны быть проверены в первую очередь.

«Точки» должны быть сгруппированы с точки зрения удобства тестирования так, чтобы переход от одной точки к другой требовал минимальных усилий: все «точки» одной операционной системы (чтобы не приходилось несколько раз переустанавливать ОС), все точки одной аппаратной конфигурации (чтобы не приходилось несколько раз менять аппаратуру) и т.д. Измерения для группирования могут быть взяты стандартные или уточнены пользователем.

В процессе тестирования пользователь вводит в Навигатор информацию об уже проверенных «точках». При этом НПС визуализирует маршруты внутри гиперпространства. В любой момент может быть вычислена некоторая интегральная оценка оттестированной части программы и не оттестированной (процент проверенности программы).

В Навигатор может быть введена информация об обнаруженных ошибках и о точках, в которых были обнаружены ошибки. Навигатор запоминает эту информацию для дальнейшего анализа.

Навигатор поддерживает такое понятие как «тест»: может манипулировать с объектами класса «тест» и их атрибутами. Возможен анализ тестов. Визуализация части пространства, покрытого данным тестом и группой тестов.

Таким образом, Навигатор представляет интерес для всех разработчиков программного обеспечения от программистов-одиночек до крупнейших фирм. Отличия будут в масштабах применения, в объемах и сложности PPS.

НПС применим для разработки программного обеспечения любого класса и размера, поскольку любое программное обеспечение нуждается в тестировании.

5. Применение идеи навигатора в процессе обучения

Ранее речь шла о НПС, как об инструменте для составления плана стабилизации программных продуктов. Однако, те же идеи можно применить к образовательному

процессу, а именно, к процессу обучения тестирования.

Основное отличие состоит в том, что НПС, как инструмент планирования, прокладывает путь и помогает специалистам по тестированию проверять программные продукты наиболее оптимальным образом. А НПС, как обучающий инструмент, следит за действием студента и сравнивает действия «подопечного» с оптимальными. Кроме того, если в первом случае НПС рассчитывает возможное нахождение ошибки в том или ином месте, то во втором случае НПС точно знает, где расположена ошибка (она была заранее внесена умышленно).

Реализация НПС в качестве обучающего инструмента позволит проверить основные идеи на практике, отладить НПС на задачах меньшего объема, чем крупные промышленные комплексы. В дальнейшем, когда обучающая программа будет готова, можно будет переходить к реализации инструмента планирования.

На текущий момент завершена деловая игра в стабилизацию, в которой оперирование шло «бумажным» прототипом будущего НПС. «Бумажный» прототип использовался для определения того, что должно быть включено в НПС в первую очередь, реализацию чего можно немного отложить, от каких идей лучше отказаться (или пока отказаться) и т.д.

6. Краткое описание игры в стабилизацию.

Студент играет роль специалиста по тестированию.

Для проведения деловой игры были подготовлены гиперкуб с заранее расставленными ошибками и техническое задание (не лучшего качества), описывающее программный комплекс для некоего холдинга. Цель работы студентов – обнаружить как можно большее количество ошибок.

Специалист по тестированию (студент) тестирует программу: подготавливает конфигурацию, прогоняет набор тестов, получает некоторый ответ. После чего тестер должен задокументировать полученные результаты, проанализировать и, если была обнаружена ошибка, «программисту» (реально программиста в игре не существует). На построение и прогон тестов тратится модельное время. Это позволит вести работу «во времени», считать количество ошибок, обнаруженных каждый день, отслеживать их увеличение или уменьшение и т.п.

Через какое-то время (модельное) программист передает тестирующему новую версию программы, в которой некоторые ошибки он попытался исправить. Попытка не обязательно удачна. Эти же самые ошибки повторяться уже не должны, но могут возникнуть другие.

На процесс тестирования отведено фиксированное количество модельного времени.

В какой-то момент тестер принимает решение об окончании тестирования. Причиной может быть исчерпание отведенного на тестирование времени, или исправление всех найденных ошибок.

7. Итоги игры в стабилизацию.

С точки зрения реализации НПС основным итогом можно считать то, что представленная модель представления программного продукта в виде гиперкуба была опробована и признана пригодной для дальнейшего использования.

С точки зрения образовательного процесса можно выделить несколько итогов:

студенты на практике поняли, что такое тестирование и как надо действовать при тестировании программ;

студенты на практике ощутили, к чему приводят допущенные в ТЗ неточности спецификации программного продукта;

как признаются сами студенты, такие занятия в виде деловых игр запоминаются лучше, чем обычные лекции, а значит, после сдачи экзамена по курсу полученные знания не будут забыты.

8. Заключение

Исследования по теме «Навигатор процесса стабилизации» (НПС) относятся к архиактуальной теме – повышение качества создаваемых программных продуктов.

НПС предназначен для управления процессом стабилизации сложного программного продукта. Он позволяет определить наилучший порядок тестирования и оценить «степень проверенности» продукта.

НПС предлагает представление всего многообразия потребностей в тестировании в виде многомерного пространства. Измерения PPS характеризуют многообразие применений стабилизируемого ПП: «Функции», «Платформы», «Производительность», «Интерфейс», «Юзабилити», «Документация» и т.д. Точки этого пространства задают некоторые конкретные ситуации, которые должны быть проверены. Процесс тестирования представляется в виде перебора точек PPS.

НПС, как инструмент планирования, представляет интерес для всех разработчиков программного обеспечения от программистов-одиночек до крупнейших фирм. Отличия будут в масштабах применения, в объемах и сложности PPS.

НПС, как инструмент обучения, интересен в образовательном процессе, т.к. позволяет студентам столкнуться с проблемами, которые «поджидают» их при работе над реальными проектами.

Литература

1. Савин Р. Тестирование Дот Ком, или Пособие по жестокому обращению с багами в интернет-стартапах. — М.: Дело, 2007. — 312 с.
2. Аджиев В. «Мифы о безопасном ПО: уроки знаменитых катастроф» [Электронный ресурс]. Режим доступа: <http://www.osp.ru/os/1998/06/179592/>.
3. N. Leveson, C. Turner "An Investigation of the Therac-25 Accidents", - Computer, Vol.26, N.7, July 1993, p. 18-41.
4. Цена программной ошибки. [Электронный ресурс]. Режим доступа: <http://www.cusoft.ru/error.php>.
5. 2013 Trends in Automated Testing For Enterprise Systems. Market Research Report [online] Available: <http://www.worksoft.com/files/resources/Worksoft-Research-Report-2013-Trends-in-Automated-Testing.pdf>
6. Тренды в автоматизированном тестировании в 2013 году — часть 3/3 [Электронный ресурс]. Режим доступа: <http://qapl.net/тренды-в-тестировании-3/>.
7. Автоматизация тестирования: выбор инструмента. [Электронный ресурс]. Режим доступа: <http://blog.openquality.ru/tool-choice/>.

Содержание

СЕКЦИЯ «АВТОМАТИЗАЦИЯ И МОДЕЛИРОВАНИЕ».....	5
В.М. Азанов, Ю.С. Кан. Оптимизация коррекции околокруговой орбиты ИСЗ по вероятностному критерию	5
О.С. Баруздина, О.Г.Максимова. Системные исследования полимерных покрытий металла с целью управления их прочностью	11
Д.В. Богачев, И.А. Варфоломеев, Е.В. Ершов. Отбор значимых факторов при построении нейро-нечеткой модели управления процессом ускоренного охлаждения листа.....	18
И.А. Варфоломеев, Е.В. Ершов, Л.Н. Виноградова, Е.В. Трифанова. Повышение эффективности управления процессом сушки полимерного покрытия на поверхности металлической полосы	27
С.Н. Васильева, Ю.С. Кан. Метод решения задачи квантильной оптимизации с билинейной функцией потерь	38
А.И. Глушченко, Д.А. Полещенко, Ю.И. Еременко. Исследование применимости нейросетевого оптимизатора параметров ПИД-регулятора для управления нагревательным объектом	43
Д.В. Иванов, И.Р. Ширинов. Оценивание параметров линейных многомерных по входу ARX систем нецелого порядка с помехой в выходном сигнале	54
А.С. Михайлов, Б.А. Староверов. Синтез динамического регулятора состояния с контуром нейросетевой адаптации	64
Е.В. Проуторов, О.Г. Максимова, А.В. Максимов, Д.В. Диордийчук. Имитационная модель сушки полиуретанового покрытия	73
В.Р. Соболь. Математическое моделирование процесса воздействия затекающих воздушных потоков	81
Я.А. Соловьева, О.Г. Максимова, А.В.Максимов. Возможность применения однотипных моделей при исследовании систем, находящихся на разных уровнях организации	88
СЕКЦИЯ «ПРИКЛАДНЫЕ АСПЕКТЫ СИСТЕМНОГО АНАЛИЗА»	96
О.С. Большаков, А.В. Петров. Модель представления программ управления распределенными микроконтроллерными системами.....	96
А.В. Вашкевич, В.Г. Жуков. Исследование, разработка и модернизация протоколов обеспечения конфиденциальности в многостороннем кластерном анализе.....	107
А.В. Вицентий. Разработка многопользовательского WEB-сервиса информационной поддержки радиологического мониторинга	118
В.В. Воронина, В.В. Пучков. Система рецензирования научного контента	124
А.С. Евсеев. Модель формы и содержания групповых представлений документов.....	131

Я.Р. Гринберг, И.И. Курочкин, А.В. Корх. Примеры эффективного использования двухуровневой динамической маршрутизации в телекоммуникационных сетях.....	138
А.М. Альбертьян, И.И. Курочкин. Использование возможностей процессоров IA-32 для создания простой переносимой реализации синхронизации потоков в многопоточных приложениях на языках С и С++ для симметричных многопроцессорных архитектур.....	148
A. Lebedev. On tiling for heterogeneous systems during mechanical code parallelization.....	151
Г.С. Малтугуева, А.Ю. Юрин. Многокритериальная оптимизация в мультиагентной системе	156
Н.Ю. Паролькин, В. Г. Жуков. О решении задачи структурно-параметрического синтеза сети стандарта ieee 802.11X.....	161
А.О. Рубцов, О.В. Сухорослов. Реализация агента доступа к вычислительным ресурсам для платформы Everest	172
А.С. Джумурат. Математический аппарат и архитектура системы дистанционного обучения МАИ CLASS.NET	180
И.А. Тарханов, Д.А. Седов. Об управлении разграничением доступа в «ЕВФРАТ».....	185
А.С. Югов. О необходимости и способе улучшения процесса стабилизации программного обеспечения.....	190

Научное издание

III Всероссийская научная
конференция молодых ученых
с международным участием

ТЕОРИЯ И ПРАКТИКА
СИСТЕМНОГО АНАЛИЗА

Труды III Всероссийской научной конференции
молодых ученых с международным участием

Том I

Зав. РИО М. А. Салкова

Подписано в печать 12.05.2014 г.
Формат 60×84 1/8. Уч.-изд.л. 18,8. Тираж 83. Заказ 94

Рыбинский государственный авиационный технический университет имени П. А. Соловьева
(РГАТУ имени П. А. Соловьева)
Адрес редакции: 152934, г. Рыбинск, ул. Пушкина, 53
Отпечатано в множительной лаборатории РГАТУ имени П. А. Соловьева
152934, г. Рыбинск, ул. Пушкина, 53