

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ**



ИНФОРМАТИКА И КИБЕРНЕТИКА

2(4)

Донецк – 2016

УДК 004.3+004.9+004.2+51.7+519.6+519.7

**ИНФОРМАТИКА И КИБЕРНЕТИКА, № 2(4), – Донецк: ДонНТУ,
2016.**

Представлены материалы по вопросам приоритетных направлений научно-технического обеспечения в области информатики, кибернетики и вычислительной техники.

Материалы предназначены для специалистов народного хозяйства, ученых, преподавателей, аспирантов и студентов высших учебных заведений.

Редакционная коллегия

Главный редактор: Павлыш В.Н., д.т.н., проф.

Зам. глав. ред.: Андрюхин А.И., к.т.н., с.н.с.

Ответственный секретарь: Завадская Т.В., к.т.н., доц.

Члены редакционной коллегии: Аверин Г.В., д.т.н., проф. Аноприенко А.Я., к.т.н., проф.

Григорьев А.В., к.т.н., доц., Грунский И.С., к.т.н., проф., Зинченко Ю.Е., к.т.н., доц.

Звягинцева А.В., к.т.н., доц., Миненко А.С., д.т.н., проф., Скобцов Ю.А., д.т.н., проф.,

Шелепов В.Ю., д.т.н., проф., Карабчевский В.В., к.т.н., доц. Федяев О.И., к.т.н., доц.

Рекомендовано к печати ученым советом ГВУЗ «Донецкий национальный технический университет» Министерства образования и науки ДНР. Протокол № 5 от 24 июня 2016 г.

Журнал зарегистрирован в Министерстве образования и науки ДНР № 307 от 06.08.2015

Контактный адрес редакции

ДонНТУ, ул. Артема, 58, Донецк, 83001

Тел.: +380 (62) 301-08-56. Эл.почта: infcyb.donntu@yandex.ru

Интернет: <http://infcyb.donntu.org>

© Донецкий национальный технический университет
Министерство образования и науки ДНР, 2016

СОДЕРЖАНИЕ

Компьютерный расчет задачи взаимодействия двух жидких гравитирующих тел: спутники Юпитера Андрюхин А.И., Судаков С.Н.	4
Рекурсивные реализации задач динамического программирования Андрюхин А.И., Полетаев В.А.	12
Разработка математической модели процесса выращивания грибов в культивационной камере Береговых Ю.В., Орлов Ю.К., Грынть О.В.	20
Математические аспекты в моделировании движений гиростата с неподвижной точкой Горр Г.В., Илюхин А.А., Ковалев А.М.	26
Применение гибридных FPGA для реализации композиционного устройства управления с идентификацией выходов Ефименко К.Н.	31
Некоторые вопросы математического и компьютерного моделирования в строительстве Конопацкий Е.В.	37
Анализ возможностей и недостатков технологии построения баз знаний в современных САПР Морозова О.В.	43
Автоматизация проектирования технологических схем сушки обогащённого угля. Павлыш В.Н., Тарабаева И.В., Турчанин Г.И.	49
Исследование алгоритмов рекомендательных систем. Чепикова Е.Д., Савкова Е.О., Привалов М.В.	57
Инструментальные средства виртуальной реконструкции утраченных памятников архитектуры. Руденко М.П.	62
Подсистема защиты авторского права в программном обеспечении. Чернышова А.В., Кубашевский Д.В.	68

Компьютерный расчет задачи взаимодействия двух жидких гравитирующих тел: спутники Юпитера

Андрюхин А.И.¹, Судаков С.Н.²

¹Донецкий национальный технический университет,

²Институт прикладной математики и механики
alexandruckin@rambler.ru, sudakov@iamm.su

Андрюхин А.И., Судаков С.Н. Компьютерный расчет задачи взаимодействия двух жидких гравитирующих тел: спутники Юпитера. Получены уравнения движения для задачи двух тел, одно из которых – жидкий эллипсоид переменной вязкости, совершающий однородное вихревое движение, а другое – сферически симметричное твердое тело. Для решения уравнений использован численный метод Рунге-Кутты. В качестве примеров приведены расчеты движения систем с массово-геометрическими параметрами, системы Юпитер-Ио, Юпитер-Ганимед, Юпитер-Европа и Юпитер-Каллисто.

Ключевые слова: жидкий эллипсоид, задача двух тел, Ио, Ганимед, Европа, Каллисто, Юпитер

Введение

Динамике эллипсоидальных масс идеальной несжимаемой жидкости посвящена обширная литература [1–10]. В работе рассмотрена задача о движении двух тел, одно из которых представляет собой жидкий эллипсоид, а второе – сферически симметричное твердое тело (ССТТ). Частицы жидкости притягиваются друг к другу и к ССТТ по закону Ньютона. Жидкость считается несжимаемой и обладающей переменной вязкостью, допускающей ее однородное вихревое движение [6, 7].

Системы координат

Обозначим через $O\xi_1\xi_2\xi_3$ неподвижную систему координат, начало которой совпадает с общим центром масс рассматриваемой механической системы. Через $O_1\eta_1\eta_2\eta_3$ обозначим подвижные прямоугольные оси, начало которых O_1 совпадает с центром масс жидкого эллипсоида, а оси параллельны соответствующим осям системы $O\xi_1\xi_2\xi_3$. Через $O_1x_1x_2x_3$ обозначим систему координат, оси которой являются главными осями жидкого эллипсоида. Соответствующие системы координат представлены на рисунке 1.

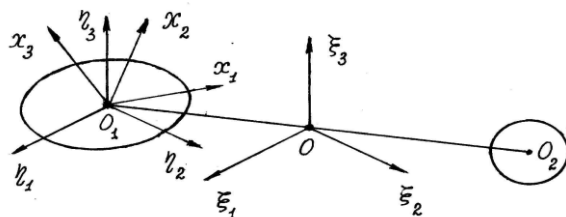


Рисунок 1. – Используемые системы координат

Координаты, скорости и ускорения центра масс жидкости

Положение точки O_1 определим сферическими координатами r_1, φ_1, θ_1 , где θ_1 – угол между плоскостью $O\xi_1\xi_2$ и радиусом-вектором OO_1 ; φ_1 – угол между $O\xi_1$ и проекцией вектора OO_1 на плоскость $O\xi_1\xi_2$.

Тогда координаты точки O_1 в осях $O\xi_1\xi_2\xi_3$ выразятся формулами:

$$\begin{aligned}\xi_1 &= r_1 \cos \varphi_1 \cos \theta_1, \\ \xi_2 &= r_1 \sin \varphi_1 \cos \theta_1, \\ \xi_3 &= r_1 \sin \theta_1.\end{aligned}\quad (1)$$

Дифференцируя формулы (1) по времени t , находим проекции скорости центра масс жидкости (точки O_1) на оси $O\xi_1\xi_2\xi_3$:

$$\begin{aligned}u_1 &= \dot{r}_1 \cos \varphi_1 \cos \theta_1 - \dot{\varphi}_1 r_1 \sin \varphi_1 \cos \theta_1 - \dot{\theta}_1 r_1 \cos \varphi_1 \sin \theta_1, \\ u_2 &= \dot{r}_1 \sin \varphi_1 \cos \theta_1 + \dot{\varphi}_1 r_1 \cos \varphi_1 \cos \theta_1 - \dot{\theta}_1 r_1 \sin \varphi_1 \sin \theta_1, \\ u_3 &= \dot{r}_1 \sin \theta_1 + \dot{\theta}_1 r_1 \cos \theta_1.\end{aligned}\quad (2)$$

Дифференцируя равенства (2) по t , находим проекции ускорения точки O_1 на оси $O\xi_1\xi_2\xi_3$:

$$\begin{aligned}w_1 &= \ddot{r}_1 \cos \varphi_1 \cos \theta_1 - r_1 \ddot{\varphi}_1 \sin \varphi_1 \cos \theta_1 - \\ &\quad r_1 \ddot{\theta}_1 \cos \varphi_1 \sin \theta_1 - r_1 (\dot{\varphi}_1^2 + \dot{\theta}_1^2) \cos \varphi_1 \cos \theta_1 + \\ &\quad + 2\dot{r}_1 \dot{\varphi}_1 \sin \varphi_1 \sin \theta_1 - 2\dot{r}_1 \dot{\theta}_1 \sin \varphi_1 \cos \theta_1 - \\ &\quad - 2\dot{r}_1 \dot{\theta}_1 \cos \varphi_1 \sin \theta_1, \\ w_2 &= \ddot{r}_1 \sin \varphi_1 \cos \theta_1 + r_1 \ddot{\varphi}_1 \cos \varphi_1 \cos \theta_1 - \\ &\quad - r_1 \ddot{\theta}_1 \sin \varphi_1 \sin \theta_1 - r_1 (\dot{\varphi}_1^2 + \dot{\theta}_1^2) \sin \varphi_1 \cos \theta_1 - \\ &\quad - 2\dot{r}_1 \dot{\varphi}_1 \cos \varphi_1 \sin \theta_1 + 2\dot{r}_1 \dot{\theta}_1 \cos \varphi_1 \cos \theta_1 - \\ &\quad - 2\dot{r}_1 \dot{\theta}_1 \sin \varphi_1 \sin \theta_1, \\ w_3 &= \ddot{r}_1 \sin \theta_1 + r_1 \ddot{\theta}_1 \cos \theta_1 - r_1 \dot{\theta}_1^2 \sin \theta_1 + \\ &\quad + 2\dot{r}_1 \dot{\theta}_1 \cos \theta_1.\end{aligned}\quad (3)$$

Положение осей $O_1x_1x_2x_3$ относительно $O_1\eta_1\eta_2\eta_3$ определим углами Эйлера φ, ψ, θ . Матрица перехода $A = (a_{ij})$ от осей $O_1\eta_1\eta_2\eta_3$ к осям $O_1x_1x_2x_3$ имеет следующий вид:

$$\begin{aligned} a_{11} &= \cos\psi \cos\varphi - \sin\psi \cos\theta \sin\varphi \\ a_{12} &= -\cos\psi \sin\varphi - \sin\psi \cos\theta \cos\varphi \\ a_{13} &= \sin\psi \sin\theta \\ a_{21} &= \sin\psi \cos\varphi + \cos\psi \cos\theta \sin\varphi \\ a_{22} &= -\sin\psi \sin\varphi + \cos\psi \cos\theta \cos\varphi \\ a_{23} &= -\cos\psi \sin\theta \\ a_{31} &= \sin\theta \sin\varphi \\ a_{32} &= \sin\theta \cos\varphi \\ a_{33} &= \cos\theta \end{aligned}$$

Используя матрицу перехода A , находим проекции ускорения точки O_1 на оси $O_1x_1x_2x_3$:

$$w_{x_i} = \sum_{j=1}^3 \dot{a}_{ij} w_j, \quad i=1,2,3. \quad (4)$$

Координаты центра масс сферически симметричного твердого тела

Центр масс ССТТ обозначим O_2 . Предполагая, что внешние силы на систему не действуют и центр масс системы неподвижен, получаем соотношение:

$$m_1 r_1 = m_2 r_2, \quad (5)$$

где m_1 – масса жидкого эллипсоида, m_2 – масса ССТТ,

$r_i = |\overline{OO_i}|$, $i=1,2,3$. При движении точки O , O_1 , O_2 будут лежать на одной прямой. Используя (5), находим координаты точки O_2 в осях $O_1\eta_1\eta_2\eta_3$:

$$\begin{aligned} \eta_{21} &= -m^{-1} r_1 \cos\varphi_1 \cos\theta_1, \\ \eta_{22} &= -m^{-1} r_1 \sin\varphi_1 \cos\theta_1, \\ \eta_{23} &= -m^{-1} r_1 \sin\theta_1, \end{aligned} \quad (6)$$

где $m = \frac{m_2}{m_1 + m_2}$. Тогда в осях $O_1x_1x_2x_3$ точка

O_2 будет иметь координаты:

$$x_{2i} = \sum_{j=1}^3 a_{ij} \eta_{2j}, \quad i=1,2,3. \quad (7)$$

Гравитационные силы, действующие на жидкость со стороны сферически симметричного твердого тела

Проекция на оси $O_1x_1x_2x_3$ силы, действующей на единичный объем жидкости со стороны ССТТ, определяются формулами:

$$f_i = G\rho m_2 \frac{\tilde{x}_{2i} - \tilde{x}_i}{r^2 [S]^{3/2}}, \quad i=1,2,3.$$

где $S = (\tilde{x}_{21} - \tilde{x}_1)^2 + (\tilde{x}_{22} - \tilde{x}_2)^2 + (\tilde{x}_{23} - \tilde{x}_3)^2$, G – гравитационная постоянная, ρ – плотность жидкости, $r^2 = x_{21}^2 + x_{22}^2 + x_{23}^2$, $\tilde{x}_{2i} = x_{2i}/r$, $\tilde{x}_i = x_i/r$, x_1, x_2, x_3 – координаты в осях $O_1x_1x_2x_3$ рассматриваемой частицы жидкости.

Линеаризация

В дальнейшем будем рассматривать только те случаи, когда расстояние r между точками O_1 и O_2 настолько велико, что можно провести линеаризацию величин f_i по $\tilde{x}_i$, $i=1,2,3$. Такая линеаризация необходима для существования однородного вихревого движения жидкости и сохранения ее эллипсоидальной формы. После линеаризации будем иметь:

$$f_1 = \frac{G\rho m_2}{r^2} \left(\frac{x_{21}}{r} + \left(\frac{3x_{21}^2}{r^2} - 1 \right) \frac{x_1}{r} + \left(\frac{3x_{21}x_{22}x_2}{r^3} + \frac{3x_{21}x_{23}x_3}{r^3} \right) \right) \quad (123),$$

где символ циклической перестановки индексов (123) не применяется к величине m_2 и к первому индексу величин x_{21}, x_{22}, x_{23} .

Уравнения гидродинамики

Движение жидкости переменной вязкости описывается уравнениями [11]:

$$\begin{aligned} \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} &= -\frac{1}{\rho} \nabla p + \nu(\mathbf{x}, \mathbf{c}) \Delta \mathbf{v} - \mathbf{w} - \boldsymbol{\omega} \times \mathbf{x} - \\ &- \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{x}) - 2\boldsymbol{\omega} \times \mathbf{v} - \nabla \Phi + \frac{1}{\rho} \mathbf{f}, \end{aligned} \quad (8)$$

$$\operatorname{div} \mathbf{v} = 0,$$

где $\mathbf{v} = (v_1, v_2, v_3)$ – скорость движения жидкости относительно осей $O_1x_1x_2x_3$, $\mathbf{x} = (x_1, x_2, x_3)$ – координатный вектор, ρ – плотность жидкости, p – давление, $\boldsymbol{\omega}$ – угловая скорость осей $O_1x_1x_2x_3$, $\mathbf{c} = (c_1, c_2, c_3)$ – полудлины главных осей жидкого эллипсоида, $\mathbf{w} = (w_{x_1}, w_{x_2}, w_{x_3})$ – абсолютное ускорение точки O_1 , $\mathbf{f} = (f_1, f_2, f_3)$.

Кинематическая вязкость $\nu(\mathbf{x}, \mathbf{c})$ определяется выражением

$$\nu = \nu_0 \left(1 - \frac{x_1^2}{c_1^2} - \frac{x_2^2}{c_2^2} - \frac{x_3^2}{c_3^2} \right),$$

где $\nu_0 = \text{const}$.

Граница жидкости в осях $O_1x_1x_2x_3$ задается уравнением:

$$\frac{x_1^2}{c_1^2} + \frac{x_2^2}{c_2^2} + \frac{x_3^2}{c_3^2} = 1.$$

Следовательно, на границе жидкости $\nu = 0$. Компоненты тензора скоростей деформаций жидкости σ имеют вид:

$$\sigma_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial v_j} + \frac{\partial v_j}{\partial v_i} \right), \quad i, j = 1, 2, 3.$$

Потенциал Φ гравитационных сил, порождаемых жидкостью, в ее внутренних точках описывается формулой [2]:

$$\Phi = \pi G (\alpha_1 x_1^2 + \alpha_2 x_2^2 + \alpha_3 x_3^2 - \chi_0),$$

где $\alpha_i = c_1 c_2 c_3 \int_0^\infty \frac{d\lambda}{(c_i^2 + \lambda)D}$, $i = 1, 2, 3$.

$$\chi_0 = c_1 c_2 c_3 \int_0^\infty \frac{d\lambda}{D}, \quad D = [(c_1^2 + \lambda)(c_2^2 + \lambda)(c_3^2 + \lambda)]^{1/2}.$$

Уравнения движения

В случае однородного вихревого движения компоненты скорости жидкости v_1, v_2, v_3 и давление p ищем в виде:

$$v_1 = \frac{\dot{c}_1}{c_1} x_1 - \frac{c_1}{c_2} \omega_3^* x_2 + \frac{c_1}{c_3} \omega_2^* x_3 \quad (123),$$

$$p = -p_0(t) \left(\frac{x_1^2}{c_1^2} + \frac{x_2^2}{c_2^2} + \frac{x_3^2}{c_3^2} - 1 \right), \quad (9)$$

где $\omega_1^*, \omega_2^*, \omega_3^*$ и $p_0(t)$ – неизвестные функции времени t .

Подставляя (9) в уравнения движения жидкости (8), получаем:

$$k_{i0} + k_{i1}x_1 + k_{i2}x_2 + k_{i3}x_3 = 0, \quad i = 1, 2, 3, \quad (10)$$

где $k_{i0} = w_{x_i} - Gm_2 \frac{x_{2i}}{r^3}$, $i = 1, 2, 3$,

$$k_{11} = \frac{\ddot{c}_1}{c_1} - \omega_3^{*2} - \omega_2^{*2} - \frac{2p_0}{\rho c_1^2} + 4v_0 \frac{\dot{c}_1}{c_1^3} - \omega_2^2 - \omega_3^2 -$$

$$- 2 \frac{c_3}{c_1} \omega_2^* \omega_3 - 2 \frac{c_2}{c_1} \omega_3^* \omega_2 + 2\pi G \alpha_1 -$$

$$- \frac{Gm_2}{r^5} (3x_{21}^2 - r^2) \quad (123),$$

$$k_{12} = -\frac{c_1}{c_2} \dot{\omega}_3^* - \frac{2\dot{c}_1}{c_2} \omega_3^* + \frac{c_1}{c_2} \omega_1^* \omega_2^* +$$

$$+ \frac{2v_0}{c_2^2} \frac{c_2^2 - c_1^2}{c_2 c_1} \omega_3^* - \dot{\omega}_3 + \omega_1 \omega_2 + 2 \frac{c_3}{c_2} \omega_1^* \omega_2 -$$

$$- 2 \frac{\dot{c}_2}{c_2} \omega_3 - 3Gm_2 \frac{x_{21}x_{22}}{r^5} \quad (123),$$

$$k_{12} = \frac{c_1}{c_3} \dot{\omega}_2^* + \frac{2\dot{c}_1}{c_3} \omega_2^* + \frac{c_1}{c_3} \omega_1^* \omega_3^* +$$

$$+ \frac{2v_0}{c_3^2} \frac{c_1^2 - c_3^2}{c_1 c_3} \omega_2^* + \dot{\omega}_2 + \omega_1 \omega_3 + \frac{2\dot{c}_3}{c_3} \omega_2 +$$

$$+ \frac{2c_2}{c_3} \omega_1^* \omega_3 - 3Gm_2 \frac{x_{21}x_{23}}{r^5} \quad (123).$$

Равенства (10) должны выполняться при любых значениях координат x_1, x_2, x_3 из области, занимаемой жидкостью. Отсюда следуют уравнения:

$$k_{ij} = 0, \quad i = 1, 2, 3, \quad j = 0, 1, 2, 3, \quad (12)$$

где k_{ij} определены выражениями (11). Из условия несжимаемости следует, что

$$c_1 c_2 c_3 = R^3 = \text{const}. \quad (13)$$

Движение осей $O_1 x_1 x_2 x_3$ относительно $O_1 \eta_1 \eta_2 \eta_3$ описывается кинематическими уравнениями Эйлера, которые запишем в виде:

$$\dot{\varphi} = \omega_3 - (\omega_1 \sin \varphi + \omega_2 \cos \varphi) \text{ctg} \theta,$$

$$\dot{\psi} = (\omega_1 \sin \varphi + \omega_2 \cos \varphi) / \sin \theta,$$

$$\dot{\theta} = \omega_1 \cos \varphi - \omega_2 \sin \varphi, \quad (14)$$

где $\omega_1, \omega_2, \omega_3$ – проекции угловой скорости осей $O_1 x_1 x_2 x_3$ на себя.

Соотношения (12) – (14) представляют собой систему шестнадцати обыкновенных дифференциальных уравнений относительно шестнадцати неизвестных $\omega_i^*, \omega_i, c_i$, $i = 1, 2, 3$, $p_0, \varphi, \psi, \theta, r_1, \varphi_1, \theta_1$ и после задания начальных условий, полностью описывают движение системы.

Уравнения движения в нормальной форме.

Уравнения $k_{23} = 0$, $k_{32} = 0$ (123) представим в виде

$$\frac{c_2}{c_3} \dot{\omega}_1^* + \dot{\omega}_1 = f_{23},$$

$$\frac{c_3}{c_2} \dot{\omega}_1^* + \dot{\omega}_1 = f_{32} \quad (123),$$

где

$$f_{23} = -\frac{2\dot{c}_2}{c_3} \omega_1^* + \frac{c_2}{c_3} \omega_2^* \omega_3^* + \omega_2 \omega_3 + 2 \frac{c_1}{c_3} \omega_2^* \omega_3 -$$

$$- 2 \frac{\dot{c}_3}{c_3} \omega_1 +$$

$$+ \frac{2v_0}{c_3^2} \left(\frac{c_3}{c_2} - \frac{c_2}{c_3} \right) \omega_1^* - 3Gm_2 \frac{x_{22}x_{23}}{r^5} \quad (123),$$

$$f_{32} = -\frac{2\dot{c}_3}{c_2} \omega_1^* - \frac{c_3}{c_2} \omega_2^* \omega_3^* - \omega_2 \omega_3 - 2 \frac{c_1}{c_2} \omega_3^* \omega_2 -$$

$$- 2 \frac{\dot{c}_2}{c_2} \omega_1 -$$

$$-\frac{2\nu_0}{c_2^2}\left(\frac{c_3}{c_2}-\frac{c_2}{c_3}\right)\omega_1^*+3Gm_2\frac{x_{22}x_{23}}{r^5} \quad (123).$$

Разрешая систему (15) относительно ω_i^* , ω_i , $i=1,2,3$, получаем уравнения

$$\begin{aligned} \dot{\omega}_1^* &= \frac{c_2c_3}{c_2^2-c_3^2}(f_{23}-f_{32}), \\ \dot{\omega}_1 &= \frac{c_2^2f_{32}-c_3^2f_{23}}{c_2^2-c_3^2} \quad (17) \end{aligned}$$

Теперь исключим переменные p_0 и $\ddot{c}_3$ из уравнений $k_{ij}=0$, $i=1,2$. Из уравнения $k_{33}=0$ находим

$$\frac{2p_0}{\rho} = \ddot{c}_3c_3 + c_3^2f_{33}, \quad (18)$$

где

$$\begin{aligned} f_{33} &= -\omega_1^{*2} - \omega_2^{*2} - \omega_1^2 - \omega_2^2 - 2\frac{c_2}{c_3}\omega_1^*\omega_1 - 2\frac{c_1}{c_3}\omega_2^*\omega_2 + \\ &+ 4\nu_0\frac{\dot{c}_3}{c_3^3} + 2\pi\rho G\alpha_3 - \frac{Gm_2}{r^5}(3x_{23}^2 - r^2) \quad (123). \end{aligned}$$

Из условия (13) следуют равенства

$$c_3 = R^3c_1^{-1}c_2^{-1}, \quad \dot{c}_3 = -R^3(c_1^{-2}c_2^{-1}\dot{c}_1 + c_1^{-1}c_2^{-2}\dot{c}_2), \quad (19)$$

$$\ddot{c}_3 = \frac{R^3}{c_1c_2}\left(-\frac{\ddot{c}_1}{c_1}-\frac{\ddot{c}_2}{c_2}+\frac{2\dot{c}_1^2}{c_1^2}+\frac{2\dot{c}_1\dot{c}_2}{c_1c_2}+\frac{2\dot{c}_2^2}{c_2^2}\right). \quad (20)$$

Используя (20), представим выражение (18) в виде

$$\frac{2p_0}{\rho} = \frac{R^6}{c_1^2c_2^2}\left(-\frac{\ddot{c}_1}{c_1}-\frac{\ddot{c}_2}{c_2}+\frac{2\dot{c}_1^2}{c_1^2}+\frac{2\dot{c}_1\dot{c}_2}{c_1c_2}+\frac{2\dot{c}_2^2}{c_2^2}\right) \quad (21)$$

$$+ c_3^2f_{33}.$$

Используя (21), представим уравнения $k_{11}=0$ и $k_{22}=0$ в виде

$$\begin{aligned} \left(1+\frac{R^6}{c_1^4c_2^2}\right)\frac{\ddot{c}_1}{c_1}+\frac{R^6}{c_1^4c_2^2}\frac{\ddot{c}_2}{c_2} &= F_1, \\ \frac{R^6}{c_1^2}\frac{\ddot{c}_1}{c_1}+\left(1+\frac{R^6}{c_1^2c_2^4}\right)\frac{\ddot{c}_2}{c_2} &= F_2, \end{aligned} \quad (22)$$

где

$$F_1 = \frac{2R^6}{c_1^4c_2^2}\left(\frac{\dot{c}_1^2}{c_1^2}+\frac{\dot{c}_1\dot{c}_2}{c_1c_2}+\frac{\dot{c}_2^2}{c_2^2}\right)+\frac{c_3^2}{c_1^2}f_{33}-f_{11},$$

$$F_2 = \frac{2R^6}{c_1^2c_2^4}\left(\frac{\dot{c}_1^2}{c_1^2}+\frac{\dot{c}_1\dot{c}_2}{c_1c_2}+\frac{\dot{c}_2^2}{c_2^2}\right)+\frac{c_3^2}{c_2^2}f_{33}-f_{22}, \quad (23)$$

Разрешая систему (22) относительно $\ddot{c}_1/c_1$ и $\ddot{c}_2/c_2$, получаем

$$\begin{aligned} \frac{\ddot{c}_1}{c_1} &= \left[\left(1+\frac{R^6}{c_1^2c_2^4}\right)F_1-\frac{R^6}{c_1^4c_2^2}F_2\right]\Delta^{-1}, \\ \frac{\ddot{c}_2}{c_2} &= \left[\left(1+\frac{R^6}{c_1^4c_2^2}\right)F_2-\frac{R^6}{c_1^2c_2^4}F_1\right]\Delta^{-1}, \end{aligned} \quad (24)$$

Уравнения

$$k_{0i}=0, \quad i=1,2,3$$

записываем в виде

$$w_{x_i} = Gm_2\frac{x_{2i}}{r^3}, \quad i=1,2,3, \quad (25)$$

где

w_{x_i} , $i=1,2,3$ – проекции абсолютного ускорения

точки O_1 на оси $O_1x_1x_2x_3$, которые определяются формулами

$$(3), (4). \text{ Величины } x_{2i}, \quad i=1,2,3$$

определены формулами (7).

Умножая уравнения (25) слева на матрицу A^T , обратную матрице A , приводим их к виду

$$w_i = \frac{Gm_2}{r^3}\sum_{j=1}^3 a_{ij}x_{2j}, \quad i=1,2,3.$$

Учитывая (7), перепишем эти уравнения так

$$w_j = \frac{Gm_2}{r^3}\eta_{2j}, \quad i=1,2,3,$$

где η_{2j} определены формулами (6).

Используя (3) и (6), запишем последние уравнения в виде

$$\ddot{r}_1 \cos \varphi_1 \cos \theta_1 - r_1 \ddot{\varphi}_1 \sin \varphi_1 \cos \theta_1 - r_1 \ddot{\theta}_1 \cos \varphi_1 \sin \theta_1 -$$

$$\begin{aligned}
& -r_1(\dot{\varphi}_1^2 + \dot{\theta}_1^2) \cos \varphi_1 \cos \theta_1 + \\
& 2r_1\dot{\varphi}_1\dot{\theta}_1 \sin \varphi_1 \cos \theta_1 - 2\dot{r}_1\dot{\varphi}_1 \sin \varphi_1 \cos \theta_1 - \\
& -2\dot{r}_1\dot{\theta}_1 \cos \varphi_1 \sin \theta_1 = -\frac{Gm_2}{mr^3} r_1 \cos \varphi_1 \cos \theta_1, \\
& \ddot{r}_1 \sin \varphi_1 \cos \theta_1 + r_1\ddot{\varphi}_1 \cos \varphi_1 \cos \theta_1 - \\
& r_1\ddot{\theta}_1 \sin \varphi_1 \sin \theta_1 - \\
& -r_1(\dot{\varphi}_1^2 + \dot{\theta}_1^2) \sin \varphi_1 \cos \theta_1 - 2r_1\dot{\varphi}_1\dot{\theta}_1 \cos \varphi_1 \sin \theta_1 \\
& + 2\dot{r}_1\dot{\varphi}_1 \cos \varphi_1 \cos \theta_1 - \\
& -2\dot{r}_1\dot{\theta}_1 \sin \varphi_1 \sin \theta_1 = -\frac{Gm_2}{mr^3} r_1 \sin \varphi_1 \cos \theta_1, \\
& \ddot{r}_1 \sin \theta_1 + \ddot{\theta}_1 r_1 \cos \theta_1 - \dot{\theta}_1^2 r_1 \sin \theta_1 + 2\dot{r}_1\dot{\theta}_1 \cos \theta_1 = -\frac{Gm_2}{mr^3} r_1 \sin \theta_1
\end{aligned}$$

Разрешая эти уравнения относительно $\ddot{r}_1, \ddot{\varphi}_1, \ddot{\theta}_1$, приводим их к виду

$$\begin{aligned}
\ddot{r}_1 &= r_1(\dot{\varphi}_1^2 \cos^2 \theta_1 + \dot{\theta}_1^2) - \frac{Gm_2}{mr^3} r_1, \\
\ddot{\varphi}_1 r_1 \cos \theta_1 &= 2r_1\dot{\varphi}_1\dot{\theta}_1 \sin \theta_1 - 2\dot{r}_1\dot{\varphi}_1 \cos \theta_1, \\
(26)
\end{aligned}$$

$$\ddot{\theta}_1 r_1 = -2r_1\dot{\varphi}_1^2 \cos \theta_1 \sin \theta_1 - 2\dot{r}_1\dot{\theta}_1.$$

Уравнения (26) представляют собой уравнения движения в задаче двух тел. Второе из этих уравнений может быть проинтегрировано и представлено в виде

$$\dot{\varphi}_1 r_1^2 \cos^2 \theta_1 = \text{const.}$$

Уравнения (14), (17), (24), (26) разрешены относительно старших производных и представляют собой систему четырнадцати обыкновенных

дифференциальных уравнений относительно четырнадцати неизвестных

$$\varphi, \psi, \theta, \quad \omega_i^*, \omega_i, i=1,2,3, \quad c_1, c_2, r_1, \varphi_1, \theta_1$$

и после задания начальных условий, полностью описывает движение рассматриваемой механической системы.

Для численного решения уравнений (14), (17), (24), (26) вводим

безразмерные переменные $\tau = \frac{t}{T}$,

$$\zeta_i = \frac{c_i}{R}, \quad u_i = \frac{d\zeta_i}{d\tau}, \quad \tilde{\omega}_i^* = \omega_i^* T,$$

$$\tilde{\omega}_i = \omega_i T, \quad i=1,2,3, \quad \tilde{r} = \frac{r_1}{R}, \quad \text{где } T -$$

размерность времени, $R = \sqrt{c_1 c_2 c_3}$.

Для решения уравнений был использован численный метод Рунге-Кутты 4-5.

Единицей времени являются земные сутки. Единицей длины на графиках является 1 метр.

Результаты расчетов представлены на рис.2-

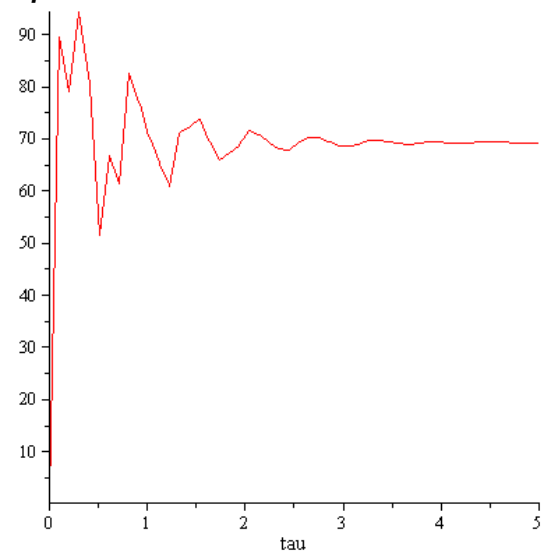


Рисунок 2—Разница между осями для Ио.

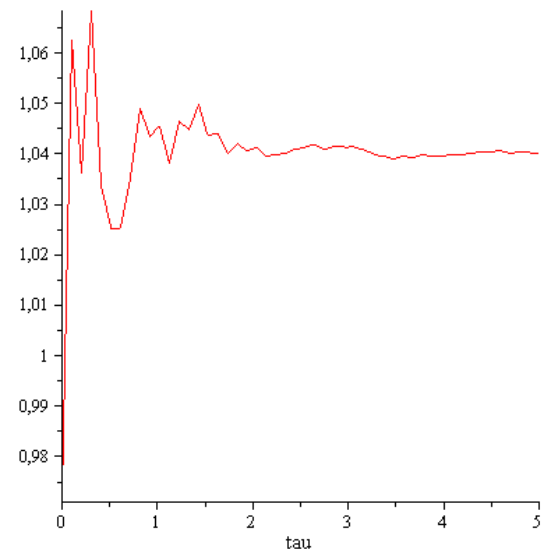


Рисунок 3—Разница между осями для Каллисто.

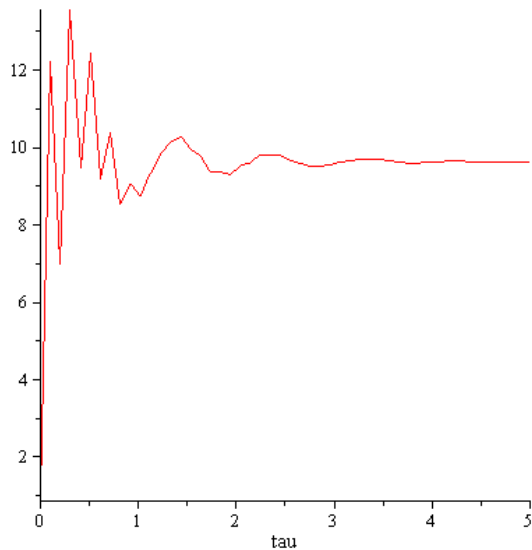


Рисунок 4—Разница между осями для Европы.

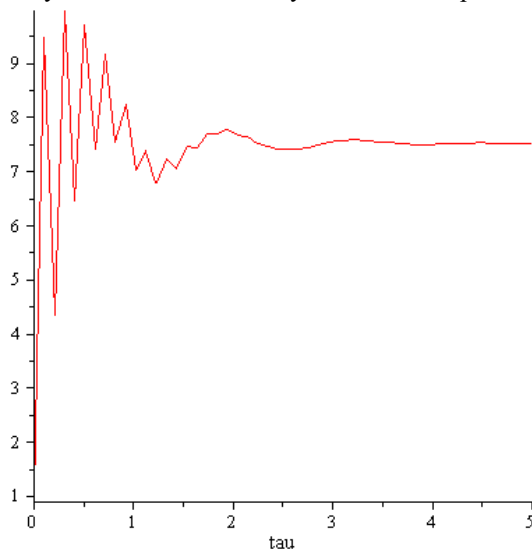


Рисунок 5—Разница между осями для Ганимеда.

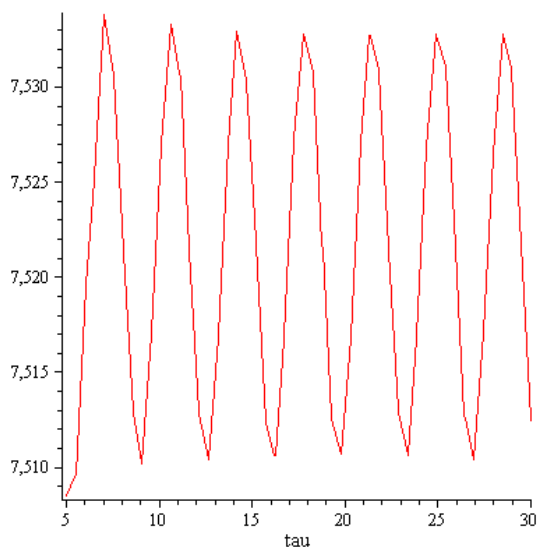


Рисунок 6—Разница между осями для Ганимеда.

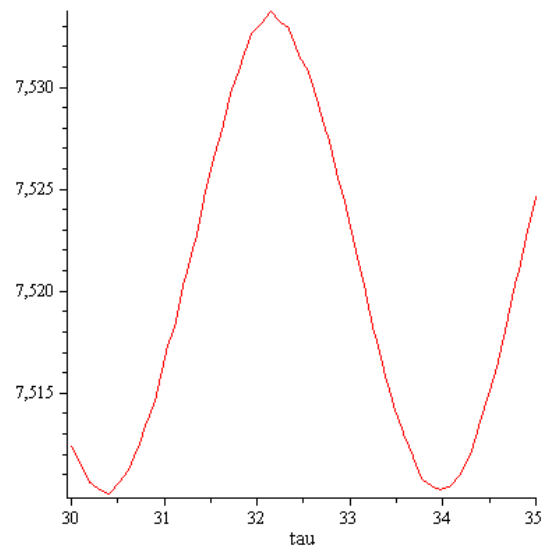


Рисунок 7—Разница между осями для Ганимеда.

Выводы

Решаемая задача является насущной ввиду успешных космических полетов станции Юнона к Юпитеру. На рис. представлены снимки Юпитера и его спутников, которые получены этой станцией.

Расчет моделей достаточно трудоемкий процесс. Построение графика отклонений полюсов за временной промежуток 10 суток занимает примерно 1,5 часа работы процессора I3 модели 4170 с частотой 3,7 ГГц.

Время расчета за более длинный временной период в лучшем случае прямо пропорционально его длине.

Компьютерные расчеты и эксперименты продолжаются в следующих направлениях:

а) увеличение длины рассматриваемого временного интервала;

б) увеличение числа тел, т.е. выполнение компьютерных экспериментов с задачей трех и более тел.

Расчеты производились в пакетах Maple и Mathematica и сравнивались.

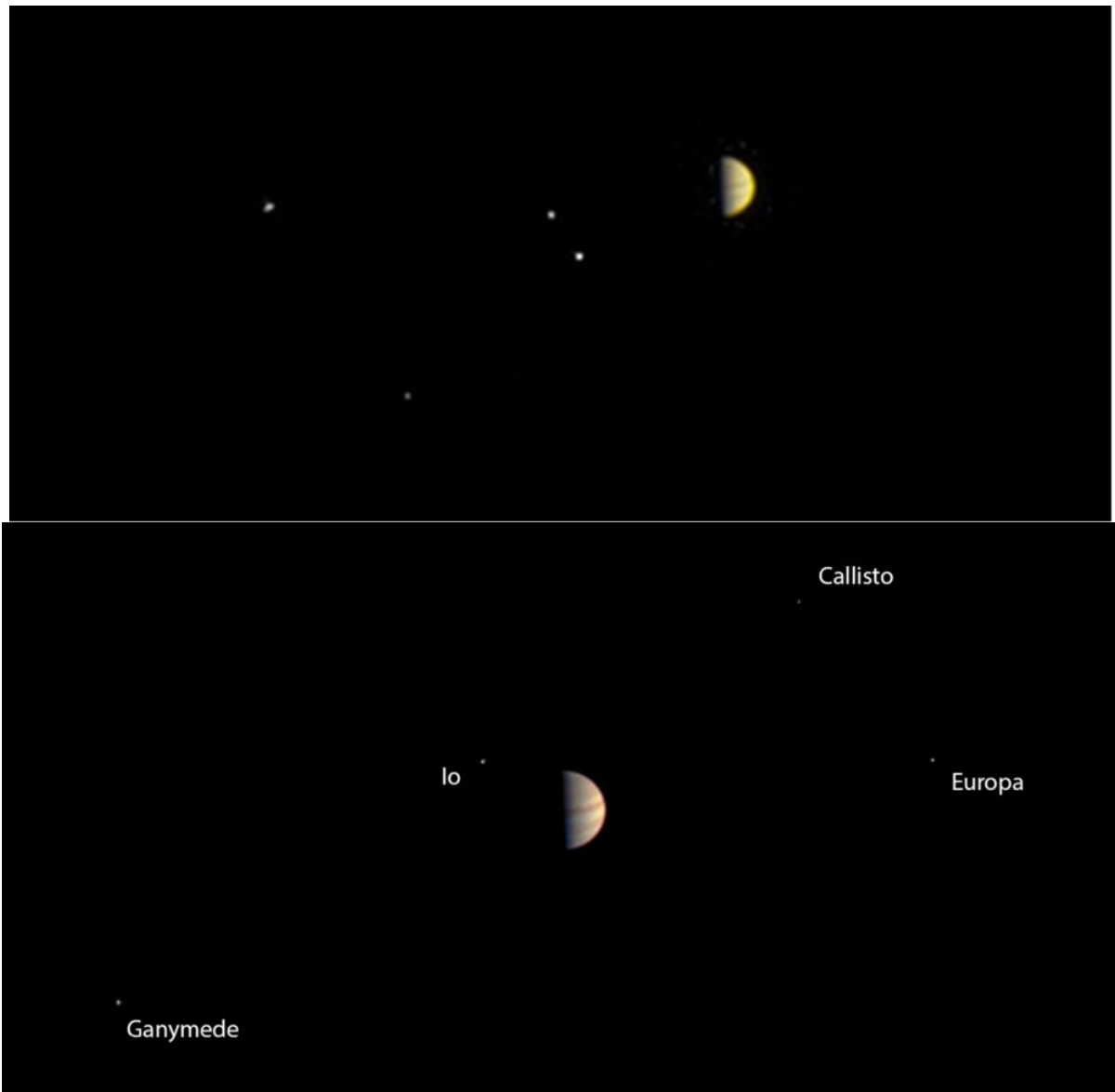


Рисунок 8- Снимок Юпитера и его 4 крупнейших спутников Ио, Европы, Ганимеда и Каллисто из 67 с расстояния а) 10,9 млн километров 21.06.16 б) 6,3 млн километров 29.06.16 (www.nasa.gov) .



Рисунок 9- Снимок Юпитера и его спутников с расстояния 10,9 миллионов километров 21.06.16

Литература

1. Динамика жидких и газовых эллипсоидов / Сборник статей // Ред. А.В. Борисов, И.С. Мамаев. – Москва-Ижевск: НИЦ «Регулярная и хаотическая динамика», 2010. – 364 с.
2. Ламб Г. Гидродинамика. – М.-Л.: Гостехиздат, 1947. – 928 с.
3. Петкевич Е.В. Задача двух жидких тел // Письма в Астрономический журнал. – 1977. – 3. – № 9. – С. 424 – 428.
4. Петкевич Е.В. Уравнения внешней задачи двух тел // Письма в Астрономический журнал. – 1977. – 3. – № 11. – С. 522 – 525.
5. Стеклов В.А. Работы по механике. – М.-Ижевск: ИКИ, 2011. – 492 с.
6. Субботин М.Ф. Курс небесной механики. – Т. 3. – М.: Гостехиздат, 1949. – 280 с.
7. Судаков С.Н. Об уравнениях движения твердого тела с эллипсоидальной полостью, целиком заполненной жидкостью переменной вязкости // Труды ИПММ НАН Украины. – 2000. – 5. – С. 141 – 144.
8. Судаков С.Н. О колебаниях вращающихся жидких гравитирующих эллипсоидов переменной вязкости // Механика твердого тела. – 2002. – Вып. 32. – С. 217 – 226.
9. Chandrasekhar S. Ellipsoidal figures of equilibrium. – New Haven and London: Yale university press, 1969. – 98 p.
10. Ядрицкий В.С. Теория фигур небесных тел. – М.-Ижевск: ИКИ, 2011. – 300 с.
11. Лойцянский Л.Г. Механика жидкости и газа. – М.: Наука, 1973. – 848 с.

Андрюхин А.І., Судаков С.Н. Комп'ютерний розрахунок завдання взаємодії двох рідких гравитуючих тел: супутники Юпітера. Отримано рівняння руху для задачі двох тіл, одне з яких - рідкий еліпсоїд змінної в'язкості, що здійснює однорідне вихровий рух, а інше - сферично симетричне тверде тіло. Для вирішення рівнянь використаний чисельний метод Рунге-Кутта. Для прикладу наведено розрахунки руху систем з масово-геометричними параметрами, системи Юпітер-Іо, Юпітер-Ганімед, Юпітер-Європа і Юпітер-Каллісто.

Ключові слова: рідкий еліпсоїд, завдання двох тіл, Іо, Ганімед, Європа, Калліст, Юпітер

Andryukhin AI, SN Sudakov Computer calculation problem of interaction of two liquid gravitating bodies: the moons of Jupiter. We obtain the equations of motion for the two-body problem, one of which - a liquid ellipsoid of variable viscosity, a homogeneous rotational motion, and the other - a spherically symmetrical solid. To solve the equations used numerical method of Runge-Kutta. As examples, we calculate traffic systems with mass-geometric parameters, the system of Jupiter-Io, Jupiter-Ganymede, Jupiter Europa Jupiter and Callisto.

Keywords: liquid ellipsoid, two-body problem, Io, Ganymede, Europa, Callisto, Jupiter

Стаття поступила в редакцію 20.05.2016
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом

УДК 004.0- 519.854.001

Рекурсивные реализации задач динамического программирования

А.И. Андрухин, к.т.н, В.А. Полетаев, студент.
Донецкий национальный технический университет, г. Донецк,
alexandruckin@rambler.ru

Андрухин А.И. Полетаев В.А. Рекурсивные реализации задач динамического программирования. В работе рассмотрены рекурсивные реализации известных задач, решаемых с помощью метода динамического программирования. Представлены задача оптимального управления, задача оптимального распределения ресурсов, задача оптимальной замены оборудования. Приведен общий алгоритм решения задач динамического программирования. Указаны достоинства и недостатки применения принципа Беллмана и рекурсии в программных реализациях. Приведены результаты компьютерных расчетов. В расчетах использовался пакет Mathematica.

Ключевые слова: динамическое программирование, принцип Беллмана, рекурсия.

Введение

Исследование операций занимается изучением экстремальных задач управления, планированием и разработкой методов их решения. Динамическое программирование, как один из разделов исследования операций, предполагает, что процесс принятия решения и управления может быть разбит на отдельные части или этапы, т.е. является многоэтапным. Одним из основных методов динамического программирования (МДП) является метод функциональных уравнений Р.Беллмана [1–5], который базируется на использовании его же принципа оптимальности.

Рекурсия и ее применение

Ответ на вопрос о нужности использования рекурсии в программировании непрост: с одной стороны, обычно стараются не использовать рекурсию в программах вследствие понижения эффективности и стараются ее искоренять[6]. С другой стороны, если исследуемая проблема является по своей сути рекурсивной, то программная реализация может быть структурно проще и нагляднее. Также можно добавить, что окружающий нас мир является примером глобального использования рекурсии согласно работам[7-8] и др., а рекурсия в современных функциональных языках типа Haskell является единственным способом организации цикла.

Рекурсия применяется там, где она естественно требуется вследствие рекурсивности самой задачи. При этом скорость работы программы имеет малое значение и не нужно решать проблему параллельного функционирования программных модулей реализуемой системы.

Примеры реализации функции Аккермана в [9] показательны в этом плане.

Принцип оптимальности Беллмана

Известный принцип Беллмана состоит в том, что, при любом начальном состоянии на любом шаге и любом управлении, выбранном на этом шаге, последующие управляющие воздействия должны выбираться оптимальными относительно состояния, в котором будет система в конце данного шага, т.е. не они не зависят от предыстории.

Более детально, рассмотрим рис.1, на котором представлена траектория движения управляемого объекта в пространстве переменных состояний $(x_1, \dots, x_i, \dots, x_n)$. Точка x_s делит оптимальную траекторию x_0, x_T на две части, которые обозначены 1 и 2 на рис.1. Принцип Беллмана гласит, что если вся траектория x_0, x_T оптимальна, то оптимален и участок 2.

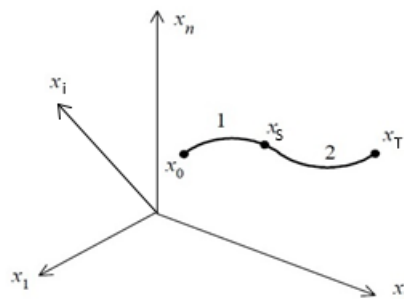


Рисунок 1-Траектория оптимального движения объекта управления

Применение принципа Беллмана зачастую показывает, что выбранное на данном шаге управление не является локально лучшим, но

будет оптимальным по глобальному критерию всего многоэтапного процесса управления. Непрерывный управляемый процесс можно рассматривать как дискретный, условно разбив его на временные отрезки. К сожалению, нет формального метода разбиения всего процесса на этапы и здесь все зависит от умений исследователя.

Алгоритм решения задач МДП

Характерным признаком МДП является небольшое количество подзадач, которые решаются многократно.

Ввиду применения принципа Беллмана следующим отличительным свойством решения задач МДП является оптимальность для подзадач.

Общий подход построения алгоритмов решения задач МДП определим следующими пунктами:

1. Четкая формулировка проблемы.
2. Выполнить разбиение оптимизируемого процесса на этапы (это искусство, а не наука, что указывалось ранее).
3. Написать рекуррентное соотношение, т.е. определить соотношение между глобальным критерием оптимальности всего процесса или задачи и локальными критериями оптимальности этапов или подзадач. В этом пункте необходимо также обеспечить возможность получения характеристик найденного глобального оптимального решения. В чисто практическом плане при программной рекурсивной реализации необходимо явно указывать условие выхода из рекурсии.
4. Найти оптимальное глобальное решение для проблемной задачи или оптимизируемого процесса. При рекурсивном решении используем стратегию решения сверху вниз т.е. берем глобальную задачу, в которой решаем необходимые подзадачи. В противном случае используем стратегию снизу вверх, т.е. решаем сначала элементарные подзадачи, потом на следующей итерации подзадачи, которые требуют результатов решенных подзадач на предыдущей итерации и т.д., пока не будет решена глобальная проблема. Обычно последний подход является более быстродействующим.
5. Решить задачу обратного хода, т.е. получить не только значения критерия оптимальности решения, но и найти само решение,

Задача оптимального управления

Рассмотрим подход к оптимальному управлению дискретной системы согласно [1,10]. Пусть дискретная система описывается с помощью рекуррентных уравнений

$$x(k+1) = f^{k+1}[x(k), u(k)]$$

где $k = 0, 1, \dots, N-1$; $x(0)$ задано.

Целью управления является минимизация функционала

$$J = \sum_{k=0}^{N-1} L[x(k), u(k)]$$

где L – скалярная функция.

Минимальное значение функционала J зависит от начального состояния $x(0)$ (оно задается, т.е. неуправляемо). Обозначим этот минимум через $j_N[x(0)]$.

Справедливы следующие соотношения:

$$\begin{aligned} j_N[x(0)] &= \min_{u(0)} \min_{u(1)} \dots \min_{u(N-1)} \{L[x(0), u(0)] + \\ &L[x(1), u(1)] + \dots + L[x(N-1), u(N-1)]\} = \\ &= \min_{u(0)} \{L[x(0), u(0)]\} + \min_{u(1)} \dots \min_{u(N-1)} \{L[x(1), u(1)] + \dots \\ &+ L[x(N-1), u(N-1)]\} = \\ &= \min_{u(0)} \{L[x(0), u(0)] + j_{N-1}[x(1)]\} \end{aligned}$$

Здесь $j_{N-1}[x(1)]$ – минимальное значение критерия качества для процесса длительностью в $N-1$ шагов и имеющее начальное состояние $x(1)$:

$$\begin{aligned} j_{N-1}[x(1)] &= \min_{u(1)} \{L[x(1), u(1)] \\ &+ j_{N-q-1}[x(2)]\} \end{aligned}$$

Для процесса управления с $N-q$ шагами, имеющего в качестве начального состояния $x(q+1)$ ($1 \leq q \leq N-1$):

$$\begin{aligned} j_{N-1}[x(q)] &= \min_{u(q)} \{L[x(q), u(q)] \\ &+ j_{N-q-1}[x(q+1)]\} \end{aligned}$$

Последнее соотношение является частным вариантом уравнения Беллмана.

Применим эти базовые соотношения к решению стандартной непрерывной задачи.

Пример расчета для задачи оптимального управления.

Пусть траектория объекта управления $x(t)$ удовлетворяет уравнению $\dot{x} = -\dot{x} + bx$.

Необходимо определить минимальное значение функционала

$$\int_{t_0}^{t_1} (nx^2 + x) dt$$

при условиях $t_0=5$, $t_1=7$, $x(t_0)=0$, $x(t_1)=1$, $n=6$, $b=3$.

Число шагов дискретизации возьмем равным 8.

Прямая и рекурсивные программные реализации представлены на рис.2.

Подчеркнем, что достоинством рекурсивной реализации является неизменяемый программный код при изменении числа шагов дискретизации.

Задача оптимального распределения ресурсов

Пусть имеется некоторое количество ресурсов x , которое необходимо распределить между N различными мероприятиями, фирмами, объектами, и т. д. таким образом, чтобы получить максимальную эффективность от выбранного способа распределения.

Примем следующие обозначения: x_i - количество ресурсов, выделенных i -му предприятию,

$R_j(x_i)$ - это функция эффективности использования ресурса x_i j -м мероприятием;

$f_j(x)$ - критерий эффективности использования x -единиц ресурсов первыми j различными мероприятиями.

Построим математическую модель этой оптимизационной задачи:

$$F_n(x) = \max \sum R_j(x_j) \\ \text{при условии, что } \sum x_j = x, \quad x_j > 0, \quad j = 1, n$$

Можно показать, что

$$F_1(x) = R_1(x), \\ F_j(x) = \max (R_j(z) + F_{j-1}(x-z)), j = 2, n$$

Тем самым мы получили рекуррентные функциональные уравнения Беллмана.

Рассмотрим конкретную задачу по распределению средств между мероприятиями.

Пример расчета для задачи оптимального распределения ресурсов.

Совет директоров рассматривает предложения по 4 мероприятиям для увеличения производительности труда.

Для увеличения производительности труда совет директоров выделяет средства в объеме 150 у.е. с дискретностью 25 у.е. Повышение производительности труда зависит от выделенной суммы на каждое мероприятие. Эти характеристики представлены в табл. 1.

Найти распределение средств между мероприятиями, обеспечивающее максимальный прирост производительности труда, причем на одно мероприятие можно выделять не более одного вложения. Рекурсивное программная реализация представлена на рис.3

Таблица 1. Характеристики распределений

Мероприятия	Объем вложений z в у.е.					
	25	50	75	100	125	150
Мероприятие 1 $R_1(x)$	5	9	18	22	29	39
Мероприятие 2 $R_2(x)$	7	12	15	20	34	37
Мероприятие 3 $R_3(x)$	11	20	23	26	32	37
Мероприятие 4 $R_4(x)$	6	15	24	30	36	38

Задача оптимальной стратегии замены оборудования и ее математическая модель

Проблема заключается в построении оптимальной стратегии при замене старых устройств технологического оборудования на новые. Старое оборудование вызывает увеличение затрат на его ремонт и обслуживание, что обуславливает понижение его производительности. Естественно возникает необходимость динамически отслеживать временной момент, когда старое оборудование нужно заменить новым с возможностью продажи первого.

Определение оптимальных временных моментов замены оборудования составляют суть проблемы оптимальной стратегии замены оборудования.

В роли критерия оптимальности при решении этой проблемы в течение определенного временного интервала может выступать

- 1) максимум прибыли от эксплуатации оборудования,
- 2) минимум общих затрат на эксплуатацию, подлежащие минимизации.

Введем следующие обозначения:

t - дискретный момент времени,

$g(t)$ - стоимость продукции, производимой за один год на единице оборудования возраста t лет,
 $u(t)$ - ежегодные затраты на обслуживание и
 $s(t)$ - остаточная стоимость оборудования возраста лет;

p - покупная цена оборудования.

Рассмотрим период n лет, в пределах которого требуется определить оптимальный цикл замены оборудования.

Обозначим через $F_n(t)$ максимальный доход, получаемый от оборудования возраста t лет за оставшиеся n лет цикла использования оборудования при условии оптимальной

```

NMinimize[ $\left\{\frac{\Delta}{2} (bu_0^2 + (nx_1^2 + u_1^2) + (nx_2^2 + u_2^2) + (nx_3^2 + u_3^2) + (nx_4^2 + u_4^2) + (nx_5^2 + u_5^2) + (nx_6^2 + u_6^2) + (nx_7^2 + u_7^2))\right\}$ ,
[численная минимизация]

x1 - Δ b u0 == 0,
x2 - (1 - Δ) x1 - Δ b u1 == 0,
x3 - (1 - Δ) x2 - Δ b u2 == 0,
x4 - (1 - Δ) x3 - Δ b u3 == 0,
x5 - (1 - Δ) x4 - Δ b u4 == 0,
x6 - (1 - Δ) x5 - Δ b u5 == 0,
x7 - (1 - Δ) x6 - Δ b u6 == 0,
1 - (1 - Δ) x7 - Δ b u7 == 0
}, {x1, x2, x3, x4, x5, x6, x7, u0, u1, u2, u3, u4, u5, u6, u7}]

{0.141675, {x1 → 6.63382 × 10-9, x2 → 1.16589 × 10-7, x3 → 1.66934 × 10-6, x4 → 0.0000238801, x5 → 0.000341608,
x6 → 0.00488673, x7 → 0.0699052, u0 → 5.89673 × 10-9, u1 → 9.99495 × 10-8, u2 → 1.41908 × 10-6, u3 → 0.0000202994, u4 → 0.000290385,
u5 → 0.00415398, u6 → 0.0594231, u7 → 0.850053}}

```

Рисунок 2а-Прямое решение задачи оптимального управления, как задачи математического программирования.

```

Δ =  $\frac{t_1 - t_0}{K}$ ;
x[ $\frac{t_1 - t_0}{\Delta}$ ] := 1
x[0] := 0
x[i_] := (1 - Δ) x[i - 1] + Δ b u[i - 1]
u[K - 1] =  $\frac{x[K] - (1 - \Delta) x[K - 1]}{\Delta b}$ ;
S[K] = 0;
S[k_] :=  $\frac{\Delta}{2} (n (x[k])^2 + (u[k])^2) + S[k + 1]$ 

For[i = K - 2, i ≥ 0, i--, u[i] = Minimize[S[i], {u[i]}][[2, 1, 2]]]
[цикл для] [минимизировать]

N[S[0]]
[численное приближение]

Grid[{Table[u_i, {i, 0, K - 1}], Table[N[u[i]], {i, 0, K - 1}]}]
[табл...] [таблица значений] [табл...] [численное приближение]

0.141675

u0 u1 u2 u3 u4 u5 u6 u7
7.21581 × 10-9 9.92173 × 10-8 1.41903 × 10-6 0.0000202994 0.000290385 0.00415398 0.0594231 0.850053

```

Рисунок 2б-Рекурсивное решение задачи оптимального управления, как задачи математического программирования.

стратегии. Возраст оборудования отсчитывается в направлении течения процесса. Считаем, что $t=0$ соответствует варианту использования нового оборудования.

Временные этапы процесса нумеруются в обратном направлении по отношению к ходу процесса и нумерация k этапов и t -возраст оборудования представлена в таб.2.

Таблица 2. Нумерация этапов и возраста оборудования

Этапы- k	n	$n-1$	$n-2$	...	1	0
Возраст- t оборудования	0	1	2	...	$t-1$	t

На каждом этапе $k=1, \dots, n$ необходимо принять решение по замене оборудования.

Критерием для решения является максимизация прибыли.

Функциональные уравнения, основанные на принципе оптимальности, имеют вид:

$$f_1(t) = \max(r(t) - u(t), s(t) - p + r(0) - u(0)) \quad (1)$$

$$f_k(t) = \max(r(t) - u(t) + f(t+1), s(t) - p + r(0) - u(0)) \quad (2)$$

при $k=2, \dots, n$

Уравнение (1) описывает одноэтапный процесс, а уравнение (2) процесс с n -этапами.

Оба уравнения состоят из двух частей: первая часть определяет доход, получаемый при сохранении оборудования; вторая часть — доход, получаемый при замене оборудования на новое.

В уравнении (2) функция $r(t) - u(t)$ есть разность между стоимостью произведенной продукции и эксплуатационными издержками на k -м этапе процесса.

Функция $F_{k-1}(t+1)$ характеризует суммарную прибыль от оставшихся $k-1$ этапов для оборудования, возраст которого в начале реализации этих этапов составляет $t+1$ лет. Вторая часть (2) эксплицируется следующим образом: она определяет чистые издержки по замене оборудования, возраст которого t лет. Функция $r(0)$ определяет доход, получаемый от нового оборудования возраста 0 лет. Предполагаем мгновенный переход от работы на оборудовании возраста t лет к работе на новом оборудовании.

Функция $F_{n-1}(t+1)$ представляет собой доход от оставшихся $k-1$ этапов ($F_0(t)=0$), до начала выполнения которых возраст оборудования составляет один год. Аналогичная экспликация для уравнения для одноэтапного процесса, но в этом случае нет слагаемого вида $F_0(t+1)$, так как k принимает значение 1, 2, ..., n .

Рассматриваемые уравнения (1) и (2) являются рекуррентными функциональными уравнениями, которые позволяют определить величину $F_k(t)$ в зависимости от $F_{k-1}(t+1)$.

Пример расчета задачи оптимальной замены оборудования

Определить оптимальный цикл замены оборудования при следующих исходных данных $p=0, s(t)=0, f(t)=r(t)-u(t)$. Остальные данные, необходимые для расчета определены в рекурсивной программной реализации, которая представлена на рис.4.

Выводы

Необходимо подчеркнуть, что метод динамического программирования предполагает свое применение в условиях неопределенности для исследования и решения задач оптимального управления процессами.

Если считать, что условия неопределенности имеют случайную природу, то необходимо использовать методы теории вероятности и случайных процессов для оптимизации процессов управления. Если же неопределенность связана с активным целенаправленной реакцией внешней среды, то необходимо применять методы теории игр. Последние мы используем и в случае, когда необходимо обеспечить гарантированный результат либо отсеять варианты реализации худших неуправляемых воздействий.

Методы динамического программирования также часто применяют для построения оптимальных алгоритмов определения экстремумов и корней функций.

Однако наряду с универсальностью и малым объемом программного кода (хотя плохо понимаемого и трудно разрабатываемого во многих случаях) есть определенные недостатки.

В первую очередь можно указать, что метод динамического программирования является в сущности простым перебором всевозможных решений, хотя и экономным ввиду использования численных результатов предыдущих этапов на текущем. Поэтому быстрый рост вариантов решения обуславливает важность нейтрализации проклятия размерности, что подчеркивал Р.Беллман.

Для этого возможны следующие основные пути:

1) существенно жертвовать точностью вычислений;

2) определять приближенные управляющие воздействия и соответствующие траектории объекта управления, которые будут являться глобально оптимальными по отношению к малым (локальным) вариациям находимых приближений.

Многочисленные примеры программных реализаций образцов применения МДП на языках C, Java, а также в пакетах Matlab, Maple представлены в [10-13].


```

x = 150;
d = 25;
n = 4;

dom = Table[d i, {i, 1, x/d}];
      |таблица значений

V =  $\begin{pmatrix} 5 & 9 & 18 & 22 & 29 & 39 \\ 7 & 12 & 15 & 20 & 34 & 37 \\ 11 & 20 & 23 & 26 & 32 & 37 \\ 6 & 15 & 24 & 30 & 36 & 38 \end{pmatrix};$ 

Do[Do[R[i, d z] = V[[i, z]], {z, 1, Length[V[[1]]]}], {i, 1, n}]
      |о... |оператор цикла |длина

F[j_, 0] := {0, 0}
F[1, x_] := {R[1, x], x}
F[1, 0] := {0, 0}
F[j_, x_] := SortBy[Table[Join[{R[j, z] + F[j - 1, x - z][[1]]}, F[j - 1, x - z][[2 ;;]]],
      |сортиро... |табли... |соединить
{z}], {z, Select[dom, # <= x &]}], -#[[1]] &][[1]]
      |... |...

res = F[n, x];
Print["Эффективность: ", res[[1]]];
      |печатать

For[i = 2, i <= Length[res], i++, Print["Мероприятие ", i - 1, ": ", res[[i]]]];
      |цикл для |длина |печатать

Эффективность: 51
Мероприятие 1: 0
Мероприятие 2: 25
Мероприятие 3: 50
Мероприятие 4: 75

```

Рисунок 3-Рекурсивная реализация задачи оптимального управления ресурсами.

```

In[12]:= p = 11;
         n = 7;

In[14]:= v =  $\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 12 & 10 & 8 & 6 & 5 & 4 & 3 \\ 0 & 0 & 1 & 1 & 1 & 2 & 2 \\ 12 & 10 & 6 & 4 & 3 & 2 & 1 \end{pmatrix}$ ; (*  $\begin{pmatrix} t \\ r[t] \\ u[t] \\ s[t] \end{pmatrix}$  *)

         f[t_] = 0;
         For[i = 1, i < Length[v[[1]]], i++,
           |цикл ДЛЯ           |длина
           f[v[[1, i]]] = v[[2, i]] - v[[3, i]];
           s[v[[1, i]]] = v[[4, i]];
         ]

In[17]:= f[k_, t_] := Max[f[t] + f[k - 1, t + 1], -p + f[0] + f[k - 1, 1]]
           |максимум

         f[1, t_] := Max[f[t], -p + f[0]]
           |максимум

In[19]:= result = Table[Table[f[k, t], {t, 0, n}], {k, 1, n}];
           |табли... |таблица значений

         Print[Grid[result]]
           |печат... |таблица

12 10 7 5 4 2 1 1
22 17 12 11 11 11 11
29 22 18 18 18 18 18
34 28 25 23 23 23 23
40 35 30 29 29 29 29
47 40 36 36 36 36 36
52 46 43 41 41 41 41

In[21]:= For[i = 1, i ≤ n && result[[n, i]] > -p + f[0] + f[n - 1, 1], i++, 0]
           |цикл ДЛЯ

         Print ["Оптимальный год замены оборудования - ", i - 1]

Оптимальный год замены оборудования - 3

```

Рисунок 4-Рекурсивная реализация задачи оптимальной замены оборудования.

Литература

1. Ф.Л. Черноусько. Динамическое программирование // Соросовский образовательный журнал, 1998, № 2, с. 139-144.
2. Черноусько Ф.Л., Баничук Н.В. Вариационные задачи механики и управления: Численные методы. М.: Наука, 1973.
3. Беллман Р. Динамическое программирование. М.: Изд-во иностранная литература., 1960.
4. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. М.: Наука, 1965.
5. Беллман Р., Калаба Р. Динамическое программирование и современная теория управления. М.: Наука, 1969.
6. Шальто А., Туккель Н., Шамгунов Н. Ханойские башни и автоматы // Программирование. 2002. №8, С.82-90.
7. Д.Хофштадтер. Гедель, Эшер, Бах: эта бесконечная гирлянда. Самара, 2001, 752 с.
8. Анисимов А. В. Рекурсивные преобразователи информации К.: Вища школа, 1987, 225 с.
9. В.Любченко. О борьбе с рекурсией. Мир ПК, 2002, № 11, С.150-152.
10. A. Lew, H. Mauch. Dynamic Programming. A Computational Tool. Springer-Verlag Berlin Heidelberg 2007
11. Applied dynamic programming for optimization of dynamical systems / Rush D. Robinett III ... [et al.], SIAM, Philadelphia, 2005.
12. Moshe Sniedovich, Dynamic Programming: Foundations and Principles. // CRC Press Taylor & Francis Group Second Edition. 2011.
13. Huaguang Zhang, Derong Liu, Yanhong Luo, Ding Wang. Adaptive Dynamic Programming for Control. Algorithms and Stability // Springer-Verlag London, 2013.

Андрюхин А.І. Полетаєв В.А. Рекурсивні реалізації завдань динамічного програмування. В роботі розглянуті рекурсивні реалізації відомих задач вирішуються за допомогою методу динамічного програмування. Представлені завдання оптимального управління, завдання оптимального розподілу ресурсів, завдання оптимальної заміни обладнання. Наведено загальний алгоритм вирішення задач динамічного програмування. Вказані переваги і недоліки застосування принципу Беллмана і рекурсії в програмних реалізаціях. Наведено результати комп'ютерних розрахунків. В розрахунках використовувався пакет Mathematica.

Ключові слова: динамічне програмування, принцип Беллмана, рекурсія.

Andruchin A.I., Poletaev V.A. Recursive implementation of dynamic programming problems. The paper discusses the recursive implementation of the known problems that can be solved using dynamic programming method. The optimal control problem, the problem of optimal allocation of resources, the problem of optimal replacement equipment are presented. General algorithm for solving dynamic programming problems presented. Advantages and disadvantages of the application of the principle of Bellman and recursion are discussed in software implementations. Computing results are shown. Mathematica package was used in the calculations.

Key words: dynamic programming, Bellman principle, recursion.

Стаття поступила в редакцію 20.05.2016
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом

УДК 579.6::004.942

Разработка математической модели процесса выращивания грибов в культивационной камере

Береговых Ю.В., Орлов Ю.К., Грынь О.В.
Донецкий национальный технический университет
ojk1407@gmail.com

Береговых Ю.В., Орлов Ю.К., Грынь О.В. Разработка математической модели процесса выращивания грибов в культивационной камере. В статье приведена разработка динамической математической модели для прогноза характеристик процесса роста грибов в культивационной камере. Для решения задачи был проведен анализ характеристик объекта моделирования, исследованы химико-технологические особенности протекания процесса, сформирована гипотеза о механизме процесса и формализованы математические зависимости модели.

Введение

Вешенка – самый доступный гриб для выращивания. Плоды вешенки содержат до 30% усвояемого белка по сухому весу. Он включает 17 различных аминокислот, в том числе 8 незаменимых. Употребление вешенки снижает высокое кровяное давление, улучшает пищеварение [1].

Производство грибов относится к одному из наиболее перспективных направлений развития пищевой промышленности. Крупные грибные хозяйства используют разнообразные автоматизированные системы для поддержания заданных параметров процесса выращивания грибов. Обзор литературных источников по подобным системам показал, что, несмотря на успешное решение задачи регулирования, эти системы не предполагают решение задач оптимального управления процессом [2,6-9].

Постановка проблемы

Рост конкурентной борьбы на этом рынке повышает требования к уровню эффективности управления процессом, что обуславливает актуальность задачи синтеза системы оптимального управления процессом культивирования грибов с целью увеличения объемов производства.

В данной работе приведена разработка динамической математической модели, позволяющей прогнозировать характеристики процесса выращивания грибов. Решение данной задачи включает:

- анализ характеристик объекта моделирования;
- исследование химико-технологических особенностей протекания процесса на протяжении

всего цикла роста грибов;

- формирование гипотезы о механизме процесса;
- формализацию математических зависимостей.

Объект моделирования

Грибница вешенки обладает активными ферментами для усвоения растительного сырья. За счет высокой скорости роста грибницы вешенка имеет конкурентное преимущество перед плесенью. Поэтому, в отличие от других грибов, вешенка может вырасти даже при не очень качественной термической обработке субстрата. Субстратный блок для культивирования вешенки представляет собой полиэтиленовый мешок плотно набитый древесными опилками или другими растительными материалами [1]. На рисунке 1 показан процесс выращивания вешенки.



Рисунок 1 – Выращивание вешенки

Анализ объекта моделирования содержит исследование устройства культивационной камеры и процесса выращивания грибов.

Культивационная камера схематически представлена на рисунке 2

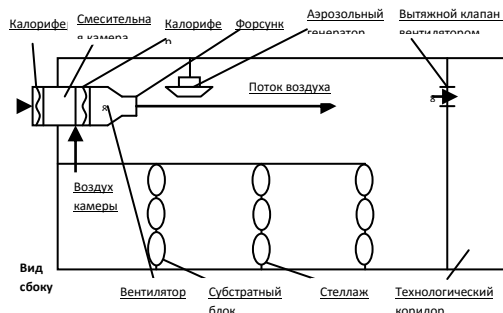


Рисунок 2 – Схематическое представление культивационной камеры

Как видно на рисунке 2, культивационная камера представляет собой закрытое помещение,

которое содержит: стеллажи для подвешивания блоков (три яруса); субстратные блоки (мицелий, субстрат, грибы); аэрозольный генератор; блок предварительной подготовки воздуха окружающей среды (калориферы, смесительная камера, приточный вентилятор); блок вывода воздуха культивационной камеры (вытяжной клапан, вентилятор) [3].

При производстве грибов выделяют четыре глобальные стадии. Первая стадия – стадия застарания субстратных блоков мицелием, вторая – стадия плодоношения и первая волна плодоношения, третья – стадия отдыха между двумя волнами плодоношения, или стадия сна, и последняя – это стадия плодоношения и вторая волна плодоношения (таблица 1).

Таблица 1 – Климатические условия в культивационной камере в процессе роста грибов вешенки в зависимости от фаз развития

№	Стадия роста	Количество дней	Фаза развития	Условия	Примечание
1	Зарастание субстратных блоков мицелием	13	Инкубация	Т воздуха 22-25°C RH 70-75% CO ₂ до 3000-5000 ppm	В период максимального разогрева температура субстрата достигает 30°C
		2	Инициация увлажнением	RH 90-95%	
		1	Образование примордиев	CO ₂ до 2000 ppm	
2	Плодообразование и 1-я волна плодоношения	4	Плодообразование	Т воздуха 14-15°C RH 90-92% CO ₂ до 900-1100 ppm	Для плодообразования не требуется резкого понижения температуры При умеренно низкой температуре качество грибов лучше
		9	Сбор 1-й волны	RH 86-88% CO ₂ 850-900 ppm	
3	Отдых между двумя волнами	5	Нагрев	Т воздуха 20-23°C RH 90-92% CO ₂ до 2000 ppm	Массово появляются примордии
4	Плодообразование и 2-я волна плодоношения	5	Плодообразование	Т воздуха 14-15°C RH 90-92% CO ₂ до 900-1100 ppm	На второй волне температуру воздуха поднимают на 1-2 °C
		10	Сбор 2-й волны	Т воздуха 15-16°C RH 86-88% CO ₂ до 900 ppm	

В таблице приведены, в зависимости от стадии развития грибов, климатические условия для всего цикла роста вешенки (50 дней).

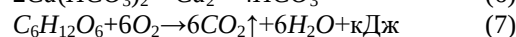
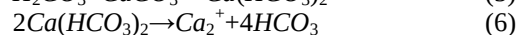
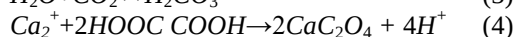
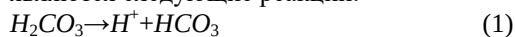
Процесс культивирования грибов возможен при определенных параметрах микроклимата. На

скорость роста грибов и их качество влияют следующие факторы: значения концентрации углекислого газа; значения оптимальной влажности; значения скорости подачи свежего воздуха; значения температуры [4]. Для

выявления механизма влияния указанных факторов на процесс производства необходимо исследовать его химико-биологические особенности.

В результате исследования химико-биологических особенностей протекания процесса культивирования грибов были выявлены основные химические реакции, происходящие в зоне субстратного блока и в зоне плодовых тел [5].

В зоне субстратного блока основными являются следующие реакции:



В зоне плодовых тел (грибов) основными химическими реакциями являются реакции (1), (2) и (3).

По результатам проведенного химико-технологического анализа сформирована гипотеза о механизме процесса.

Зона культивирования грибов разбивается на три псевдооднородные зоны (рисунок 3):

- 1-я зона – зона субстратного блока;
- 2-я зона – зона плодовых тел (грибов);
- 3-я зона – зона кульвационной камеры.

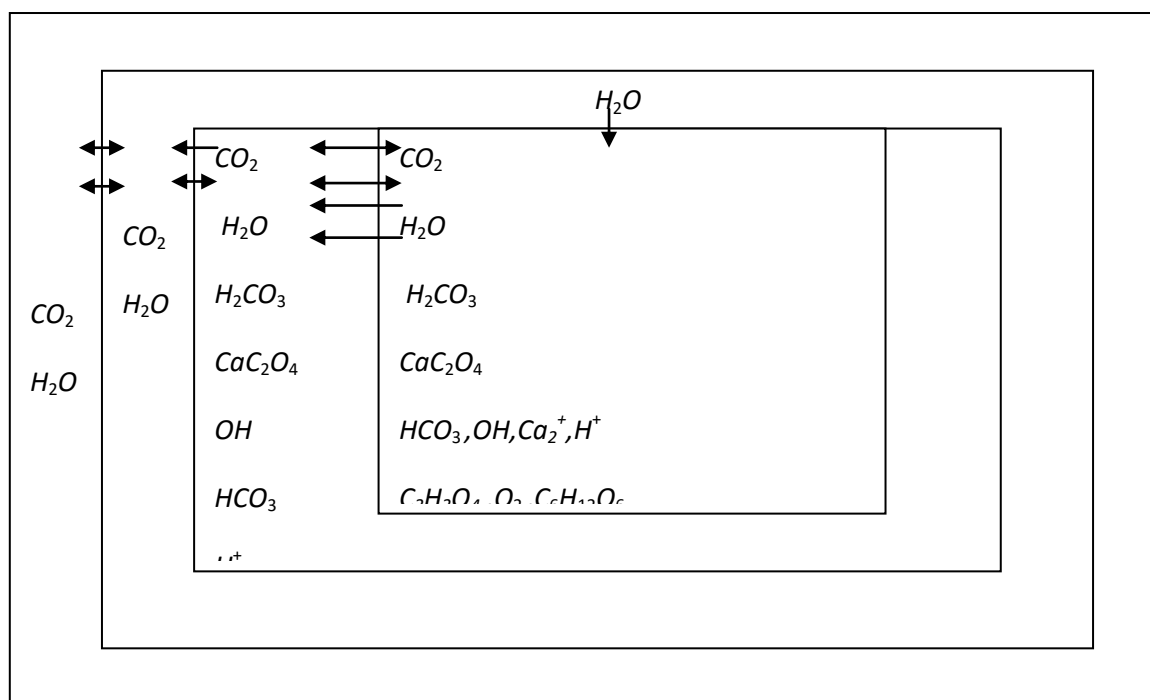


Рисунок 3 – Схематическое представление процесса культивирования грибов вешенки с разбиением на зоны

Для каждой из зон сформирован материальный и тепловой баланс, отображенный схематически на рисунке 3.

Материальный баланс характеризует изменение количества веществ в зоне, а тепловой баланс – изменение количества тепла в зоне.

При формировании гипотезы о механизме процесса приняты допущения: градиенты плотностей внутри каждой из зон отсутствуют, химические реакции протекают лишь в первой и второй зонах, а реакции (4) и (5) не принимаются во внимание из-за их незначительного влияния на процесс культивирования грибов вешенки. В качестве органической добавки в субстрате присутствует только щавелевая кислота.

Разработка математической модели

На основе сформированной гипотезы разработаны уравнения математической модели в виде системы дифференциальных уравнений. Некоторые типовые уравнения приведены ниже.

Изменение количества угольной кислоты H_2CO_3 в зоне 1:

$$\frac{dG_1^{H_2CO_3}}{dt} = \chi_{H_2CO_3}^{CO_2} \cdot g_{13}^{CO_2}(t) - g_{d11}^{H_2CO_3}(t) - \beta_1 F_{1-2} \gamma_1 ([H_2CO_3]_1 - [H_2CO_3]_2) \quad (8)$$

где $\chi_{вещ-во}^{вещ-во}$ – коэффициент стехиометрии;

$g_{13}^{CO_2}(t)$ – количество углекислого газа CO_2 ,

потраченное на реакцию (1.3) (кг/с); $g_{d11}^{H_2CO_3}(t)$ – количество угольной кислоты H_2CO_3 , потраченное на реакцию (1.1) (кг/с); β_i – коэффициент переноса массы из i -ой зоны (м/с); $i = \overline{1,4}$; F_{1-2} – площадь контакта 1-ой и 2-ой зоной (m^2); γ_i – плотность i -й зоны (кг/м³); $[вещество]_i$ – концентрация вещества в i -ой зоне (доли).

Изменение количества тепла в зоне 2:

$$\frac{dQ_2}{dt} = \beta_3 F_{2-3} \gamma_3 C_3 (T_3 - T_2) + K_1 F_{1-2} (T_1 - T_2) - \beta_2 F_{2-3} \gamma_2 C_2 (T_2 - T_3) \quad (9)$$

где C_i – удельная теплоемкость i -ой зоны (Дж/К·кг); T_i – температура i -ой зоны (К); K_i – параметры модели.

Расчет нелинейных зависимостей в правых частях уравнений приведен ниже.

Расчет количества диссоциированной угольной кислоты H_2CO_3 по реакции (3) в зоне i :

$$g_{di3}^{H_2CO_3} = K_2 \exp\left(-\frac{E_{H_2CO_3}}{RT_i}\right) \cdot [H_2CO_3]_i \quad (10)$$

где $E_{вещ-во}$ – энергия активации (Дж/моль); $R = 8,314$ (Дж/(моль·К)) – универсальная газовая постоянная.

Расчет количества углекислого газа CO_2 по реакции (3) в зоне i :

$$g_{i3}^{CO_2} = K_3 \exp\left(-\frac{E_{CO_2}}{RT_i}\right) \cdot [CO_2]_i \cdot [OH]_i \quad (11)$$

Расчет количества глюкозы $C_6H_{12}O_6$ по реакции (7) в зоне 1:

$$g_{17}^{C_6H_{12}O_6} = K_4 \exp\left(-\frac{E_{C_6H_{12}O_6}}{RT_1}\right) \cdot [C_6H_{12}O_6]_1 \cdot [O_2]_1 \quad (12)$$

Расчет концентраций веществ для всех псевдооднородных зон приведен ниже.

Концентрация угольной кислоты H_2CO_3 для i -ой зоны:

$$[H_2CO_3]_i = \frac{G_i^{H_2CO_3}(t)}{G_i(t)} \quad (13)$$

где $G_i^{H_2CO_3}(t)$ – масса угольной кислоты H_2CO_3 в i -ой зоне (кг); $G_i(t)$ – масса всех веществ i -ой зоны (кг).

Концентрация углекислого газа CO_2 для i -

ой зоны:

$$[CO_2]_i = \frac{G_i^{CO_2}(t)}{G_i(t)} \quad (14)$$

где $G_i^{CO_2}(t)$ – масса углекислого газа CO_2 в i -ой зоне (кг).

Концентрация ионов OH для i -ой зоны:

$$[OH]_i = \frac{G_i^{OH}(t)}{G_i(t)} \quad (15)$$

где $G_i^{OH}(t)$ – масса ионов OH в i -ой зоне (кг).

Концентрация глюкозы $C_6H_{12}O_6$ для зоны 1:

$$[C_6H_{12}O_6]_1 = \frac{G_1^{C_6H_{12}O_6}(t)}{G_1(t)} \quad (16)$$

где $G_1^{C_6H_{12}O_6}(t)$ – масса глицерина $C_6H_{12}O_6$ в зоне 1 (кг).

Концентрация кислорода O_2 для зоны 1:

$$[O_2]_1 = \frac{G_1^{O_2}(t)}{G_1(t)} \quad (17)$$

где $G_1^{O_2}(t)$ – масса кислорода O_2 в зоне 1 (кг).

Плотности зон рассчитываются по нижеприведенной формуле.

Плотность i -ой зоны:

$$\gamma_i(t) = \frac{G_i(t)}{V_i(t)} \quad (18)$$

где $V_i(t)$ – это объем i -ой зоны (м³).

Расчет массы зоны:

$$G_i(t) = \sum G_i^{вещ-во}(t) \quad (19)$$

где $\sum G_i^{вещ-во}(t)$ – сумма масс всех веществ в зоне i .

Выводы

Таким образом, разработана структура динамической математической модели для прогноза характеристик процесса роста грибов в культивационной камере. Для её практического применения необходимо провести процедуру параметрической идентификации и сформировать критерии и алгоритм оптимального управления процессом.

Литература

1. <http://keisideas.ru/novye-biznes-idei/vyrashhivanie-gribov-domashnyaya-biznes-ideya.html?/> – Выращивание грибов – домашняя бизнес идея.
2. Дудка И.А., Вассер С.П., Бухало А.С. Промышленное культивирование съедобных грибов. – К.: Наукова думка, 1978. – 264с.
3. Тищенко А.Д. Субстраты для культивирования вешенки. Часть 1 //Школа грибоводства – Донецк: Биотехнологии, 2003. – 60с.
4. Дудка И.А., Вассер С.П. Грибы: Справочник миколога и грибника. – К.: Наукова думка, 1987. – 535с.
5. Тищенко А.Д. Субстраты для культивирования вешенки. Часть 2 //Школа грибоводства – Донецк: Биотехнологии, 2003. – 78с.
6. Рысс А.А., Гурвич Л.И. Автоматическое управление температурным режимом в теплицах. – М.: Агропромиздат, 1986. – 128 с.: ил.
7. Крушель Е.Г. Нелинейные алгоритмы регулирования микроклимата. / Крушель Е.Г., Семенов В.Г., Степанченко И.В., Сургутанов В.В. // Известия ВолгГТУ. – 2007. -№3. – с.89-92.
8. Хребтович А.М., Гарбуз В.М., Иванов В.И. Способ регулирования микроклимата в теплице и система для его осуществления. Патент 1819537 от 28.06.91.
9. Изаков Ф.Я., Попова С.А., Ждан А.Б. Способ автоматического управления температурным режимом в теплице и система для его осуществления. Патент 2128425 от 28.07.94.
10. Суханова Н.В. Разработка математической модели ферментации дрожжей с учетом распределенности параметров. – Воронеж: Воронеж, гос. технол. ин-т, 1995. – 18 с. – Деп. в ВИНТИ 21.02.95., * 493– В 95.

Береговых Ю.В., Орлов Ю.К., Грынъ О.В. Разработка математической модели процесса выращивания грибов в культивационной камере.

В статье приведена разработка динамической математической модели для прогноза характеристик процесса роста грибов в культивационной камере. Для решения задачи был проведен анализ характеристик объекта моделирования, исследованы химико-технологические особенности протекания процесса, сформирована гипотеза о механизме процесса и формализованы математические зависимости модели.

Берегових Ю.В., Орлов Ю.К., Гринь О.В. Розробка математичної моделі процесу вирощування грибів в камері культивування.

В статті наведена розробка динамічної математичної моделі для прогнозу характеристик процесу росту грибів у культивуванні камері. Для рішення задачі був проведений аналіз характеристик об'єкта моделювання, досліджені хіміко-технологічні особливості протікання процесу, сформована гіпотеза про механізм процесу і формалізовані математичні залежності моделі.

Beregovych YU.V., Orlov Yu.K., Gryn O.V. Development of mathematical model of process of cultivation of mushrooms in a kultivatsionny chamber.

In actual development of dynamic mathematical model is resulted for the prognosis of characteristics of process of growth of mushrooms in a cultivating chamber. For the decision of task the analysis of characteristics of object of modeling and research of chemical-technological of flowing of process was conducted, a hypothesis is formed about the mechanism of process and mathematical dependences of model are formalized.

*Статья поступила в редакцию 20.05.2016
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом*

УДК 531.38

Математические аспекты в моделировании движений гиростата с неподвижной точкой

Г.В. Горр¹, А.А. Илюхин², А.М. Ковалев¹1 Государственное учреждение “Институт прикладной математики и механики”, ДНР,
отдел прикладной механики, отдел технической механики;2 Таганрогский государственный педагогический институт, (г. Таганрог, Россия),
кафедра математики
E-mail: kovalev@iamm.su

Горр Г.В., Илюхин А.А., Ковалев А.М. Математические аспекты в моделировании движений гиростата с неподвижной точкой. Доклад посвящен исследованию математических подходов в моделировании сложных механических систем класса “гиростат”. Рассмотрены две задачи о движении гиростата с неподвижной точкой. Первая задача посвящена моделированию движения гиростата с постоянным гиростатическим моментом в поле силы тяжести. Для нее указаны методы интегрирования уравнений движений с помощью первых интегралов. Вторая задача характеризуется переменным гиростатическим моментом, обусловленным неравномерным вращением несомых тел. Указан метод исследования программных (моделируемых) движений, основанный на теории инвариантных соотношений неавтономных дифференциальных уравнений.

Ключевые слова: гиростат, дифференциальные уравнения, интеграл, энергия

Введение

В современном математическом моделировании сложных объектов технических конструкций (роботов, манипуляторов, спутников и гироскопических приборов) широкое применение получила модель системы связанных твердых тел, называемая гиростатом. Характерными особенностями этой модели являются свойства: тело-носитель имеет произвольную форму; несомые тела симметричной формы имеют два режима вращения (для первого режима скорости несомых гироскопов постоянны, для второго режима они зависят от времени).

В докладе рассмотрены оба указанных случая.

В первом случае при математическом моделировании используются уравнения Эйлера–Пуассона и их первые интегралы. Исследованы особенности применения первых интегралов при интегрировании уравнений движения гиростата.

При рассмотрении второго случая, в силу отсутствия у дифференциальных уравнений интеграла энергии, предлагается метод исследования программных движений, основанный на методе инвариантных соотношений неавтономных дифференциальных уравнений.

Применение первых интегралов уравнений движения гиростата с

постоянным гиростатическим моментом

В докладе рассмотрена система обыкновенных дифференциальных уравнений

$$\dot{\mathbf{x}} = (\mathbf{x} + \boldsymbol{\lambda}) \times \mathbf{a}\mathbf{x} + s(\mathbf{e} \times \mathbf{v}), \quad \dot{\mathbf{v}} = \mathbf{v} \times \mathbf{a}\mathbf{x}, \quad (1)$$

которая допускает три первых интеграла

$$\mathbf{v} \cdot \mathbf{v} = 1, \quad (\mathbf{x} + \boldsymbol{\lambda}) \cdot \mathbf{v} = k, \quad (2)$$

$$\mathbf{x} \cdot \mathbf{a}\mathbf{x} - 2s(\mathbf{e} \times \mathbf{v}) = 2E,$$

где k и E – произвольные постоянные.

Переменными задачи являются вектор-функции $\mathbf{x} = \mathbf{x}(t)$, $\mathbf{v} = \mathbf{v}(t)$. Векторы $\mathbf{e}$, $\boldsymbol{\lambda}$, матрица $\mathbf{a} = (a_{ij})$, постоянная s характеризуют распределение масс тела, гиростатический момент, гирационный тензор.

При исследовании редукции уравнений (1) к системе меньшего порядка в ряде статей (см., например, [1]) вместо уравнений Пуассона из (1) используются первые интегралы (обзор результатов по редукции (1), (2) дан в монографии [2]). То есть исходная система (1) заменяется первым уравнением из (1) и интегралами (2). Поэтому актуальна проблема математической эквивалентности системы уравнений (1) системе, состоящей из первого уравнения системы (1) и интегралов (2).

В докладе рассмотрен пример решения, которое удовлетворяет первому уравнению из (1) и соотношениям (2), но не удовлетворяет второму

уравнению из (1). Положим $\lambda = 0$, $a_{12} = 0$, $a_{11} = a_{22}$, $a_{23} = 0$ и параметры решения.

$$\begin{aligned}x_1 &= a_* \cos \sigma, \\x_2 &= a_* \sin \sigma, \\x_3 &= n,\end{aligned}\quad (3)$$

$$v_1 s = a_* a_{13} n \cos \sigma,$$

подчиним условиям

$$\begin{aligned}s^2 &= n^2 a_{13}^2 (n^2 + n a_*^2), \\ks &= n a_{13} (a_*^2 + n^2), \\2E &= a_*^2 a_{11}^2 + a_{33}^2 n^*.\end{aligned}\quad (4)$$

В (3), (4) a_*, n – постоянные, σ – вспомогательная переменная. Непосредственной подстановкой (3), (4) в (1), (2) нетрудно убедиться в том, что решение (3), (4) удовлетворяет первому уравнению из (1), соотношениям (2), а второму уравнению из (1) не удовлетворяет. В статье [3] такие решения названы особыми (или посторонними) и для исследования редукции (1) на первых интегралах (2) вычислены производные соотношений (2) в силу только первого уравнения системы (1):

$$\begin{aligned}\mathbf{e} \cdot (\dot{\mathbf{v}} - \mathbf{v} \times \mathbf{a}\mathbf{x}) &= 0, \\(\mathbf{x} + \lambda) \cdot (\dot{\mathbf{v}} - \mathbf{v} \times \mathbf{a}\mathbf{x}) &= 0, \\\mathbf{v} \cdot (\mathbf{v} - \mathbf{v} \times \mathbf{a}\mathbf{x}) &= 0.\end{aligned}\quad (5)$$

Из уравнений (5) следует, что если векторы $\mathbf{e}$, $\mathbf{x} + \lambda$, $\mathbf{v}$ не являются компланарными, то выполняется второе уравнение из (1). В случае, когда имеет место равенство из (5)

$$(\mathbf{x} + \lambda) \cdot (\mathbf{e} \times \mathbf{v}) = 0, \quad (6)$$

второе уравнение из (1) может не иметь места. То есть построенные решения системы (1) возникают на многообразии (6). Следствием условия (6) является равенство

$$(\mathbf{x} + \lambda)^2 = n^2, \quad (7)$$

которое вытекает из первого уравнения системы (1). В статье [4] показано, что инвариантное соотношение для исходной системы (1) может выполняться только на множестве M интегрального множества N ($M \subset N$), которое описывается прецессиями гиростата [2]. Таким образом, при исследовании уравнений движения гиростата в случае (1) моделирование прецессионных движений гиростата должно осуществляться дополнительно. Это свойство должно учитываться и в моделировании движений гиростата в полях сложной структуры (например, в случае, когда гироскат движется в магнитном, электрическом и ньютоновском полях).

Движение гиростата с переменным гиростатическим моментом

В этом случае движение гиростата описывается неавтономной системой дифференциальных уравнений. Рассмотрим общий случай такой системы:

$$\dot{x}_i = \frac{dx_i}{dt} = X_i(x_1, \dots, x_n, t) \quad (i = \overline{1, n}). \quad (8)$$

В системе (8) t – независимая переменная, $x_1, \dots, x_n$ – неизвестные функции этой переменной, а X_i – функции от $n+1$ переменных, заданные на некотором открытом множестве U пространства размерности $n+1$, в котором координатами являются компоненты вектора $(x_1, \dots, x_n, t)$. Будем предполагать, что функции $X_i(x_1, \dots, x_n, t)$ ($i = \overline{1, n}$) имеют непрерывные частные производные любого порядка.

При рассмотрении неавтономной системы (8) целесообразно преобразовать ее к автономному виду. Полагая $t = x_{n+1}$, уравнения (8) можно записать в виде

$$\dot{u}_i = Y_i(u_1, \dots, u_{n+1}) \quad (i = \overline{1, n+1}), \quad (9)$$

где $Y_i \equiv X_i$, $i = \overline{1, n}$; $Y_{n+1} \equiv 1$. Решению уравнений (8) с начальными условиями $x_i(t_0) = x_i^{(0)}$ будет соответствовать решение уравнений (9) начальными условиями

$$u_i(0) = x_i^{(0)} \quad (i = \overline{1, n}), \quad u_{n+1}(0) = t_0. \quad (10)$$

Это соответствие определяется зависимостью $x_i(t) = u_i(t - t_0)$ ($i = \overline{1, n}$).

Определение 1. Непустое множество $M \subseteq U$ называется инвариантным по отношению к (9), если для любой точки $(x_1^{(0)}, \dots, x_n^{(0)}, t_0)$ из M решение уравнения (9) с начальными условиями (10) удовлетворяет условию $(u_1(t), \dots, u_{n+1}(t)) \in M$ при $t \in (t_1 - t_0, t_2 - t_0)$, где (t_1, t_2) – интервал существования соответствующего решения системы (8).

Рассмотрим программное движение механической системы, которое задано уравнением

$$f(u_1, \dots, u_{n+1}) = 0. \quad (11)$$

Предположим, что функция $f(u_1, \dots, u_{n+1})$ дифференцируема по всем переменным до произвольного порядка, в частности до порядка $n+1$, и

$$\left(\frac{\partial f}{\partial u_1}, \dots, \frac{\partial f}{\partial u_{n+1}} \right) \neq 0 \quad (12)$$

в рассматриваемой области U .

Определение 2. Соотношение (11) называется инвариантным соотношением (ИС)

системы (9), если множество G точек, удовлетворяющих этому соотношению, содержит некоторое инвариантное множество.

Поскольку инвариантное множество по определению не может быть пустым, то и ИС должно быть совместным (т.е. должна существовать по крайней мере одна точка, для которой (4) выполнено). Поставим задачу об исследовании условий существования ИС (11) системы (9).

В докладе для функции (11) построена последовательность функций

$$f^{(1)}(u_1, \dots, u_{n+1}) = f(u_1, \dots, u_{n+1})$$

$$f^{(l)}(u_1, \dots, u_{n+1}) = \sum_{j=1}^{n+1} \frac{\partial f^{(l-1)}(u_1, \dots, u_{n+1})}{\partial u_j} Y_j(u_1, \dots, u_{n+1}), \quad (13)$$

$$l = 2, 3, \dots,$$

члены которой являются производными от соответствующих функций в силу уравнений (9).

Лемма 1. Пусть для уравнений (9) соотношение (11) – инвариантное, и множество G , определяемое им, содержит инвариантное множество M . Тогда для точек множества M должны выполняться уравнения

$$\begin{aligned} f^{(1)}(u_1, \dots, u_{n+1}) &= 0 \\ f^{(2)}(u_1, \dots, u_{n+1}) &= 0, \\ f^{(l)}(u_1, \dots, u_{n+1}) &= 0 \\ \dots \quad \dots \quad \dots \end{aligned} \quad (14)$$

Доказательство. Рассмотрим множество M . Выберем произвольную точку $(x_1^{(0)}, \dots, x_n^{(0)}, t_0)$ (обозначим ее $u^{(0)}$ этого множества и возьмем решение $u_i = u_i(t)$, удовлетворяющее начальным условиям (10)). Поскольку $M \subseteq G$, то в силу определения ИС при подстановке решения в уравнения (11) получим тождество по t :

$$f^{(1)}(u_1(t), \dots, u_{n+1}(t)) = 0. \quad (15)$$

Многократно дифференцируя это тождество по t , получим

$$\begin{aligned} f^{(2)}(u_1(t), \dots, u_{n+1}(t)) &= 0, \\ \dots \\ f^{(l)}(u_1(t), \dots, u_{n+1}(t)) &= 0, \\ \dots \quad \dots \quad \dots \end{aligned} \quad (16)$$

Поскольку тождества (15) и (16) справедливы для всех t , то они верны и при $t=0$, а значит и для точки $u^{(0)}$. В силу же произвольности $u^{(0)}$ получаем, что соотношения (14) выполнены всюду на M .

Отметим, что в общем случае для составления цепочки производных необходимо

потребовать бесконечную дифференцируемость функции $f(u_1, \dots, u_{n+1})$.

В докладе доказана:

Теорема 1. Если в последовательности (13) существует k независимых членов, то независимыми будут и первые k членов последовательности (13).

Из этой теоремы вытекает следствие.

Следствие 1. Если существует непустое множество M , определяемое уравнениями (14), то система (14) равносильна своим первым k уравнениям, где k – максимальное количество независимых функций в последовательности (13).

Пусть поставлена задача о нахождении уравнений, определяющих множество M при заданном ИС (11). Согласно доказанной выше теореме, необходимо построить цепочку производных (13). Затем нужно поэтапно провести исследование зависимости входящих в (11) уравнений. То есть на первом этапе следует провести исследование зависимости производной $f^{(2)}(u_1, \dots, u_{n+1})$ от соотношения (11). Если будут найдены условия, при выполнении которых

$$f^{(2)}(u_1, \dots, u_{n+1}) \Big|_{f^{(1)}(u_1, \dots, u_{n+1})=0} \equiv 0,$$

то множество G является инвариантным.

Очевидно, что при дальнейшем изучении системы (14) необходимо рассмотреть случаи

Движение гиростата с переменным гиростатическим моментом

$$f^{(l)}(u_1, \dots, u_{n+1}) \Big|_{f^{(1)}(u_1, \dots, u_{n+1})=0, \dots, f^{(l-1)}(u_1, \dots, u_{n+1})=0} \equiv 0$$

В докладе изложенные результаты применены в задаче о движении гиростата с переменным гиростатическим моментом. Изучены условия существования равномерных вращений гиростата с переменным гиростатическим моментом под действием силы тяжести:

$$\begin{aligned} A\dot{\omega} &= -L + \lambda \times \omega + A\omega \times \omega + s \times v, \\ \dot{v} &= v \times \omega, \\ \dot{\lambda} &= L. \end{aligned} \quad (17)$$

В уравнениях (17) введены обозначения:

$\omega = (\omega_1, \omega_2, \omega_3)$ – угловая скорость тела-носителя; $v = (v_1, v_2, v_3)$ – единичный вектор, указывающий направление силы тяжести; $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ – гиростатический момент; L – вектор-функция, характеризующая взаимодействие тела-носителя и носимых тел; A – тензор инерции; $s = (s_1, s_2, s_3)$ – вектор, сонаправленный с вектором OC , где O – неподвижная точка, C центр тяжести гиростата.

На основе теоремы 1 найдены условия

существования равномерных вращений гиростата. Данная методика может быть использована в других задачах моделирования движений гиростата.

/D.Kinderlehrer, L. Nirenberg. – Séminaire Brézis, 1975.

Литература

1. Hess W. Über die Eulerschen Bewegungsgleichungen und über eine neue partikuläre Lösung des Problems der Bewegung eines starren schweren Körpers um einen festen Punkt // Math. Ann.– 1890.– В. 37, Н. 2.– С. 153–181.
2. Горп Г.В., Ковалев А.М. Движение гиростата. – Киев: Наукова думка, 2013. – 408 с.
3. Горп Г.В., Илюхин А.А., Харламова Е.И. Об особых решениях одной формы уравнений движения тела, имеющего неподвижную точку // Механика твердого тела. – 1974. – Вып. 6. – С. 3–9.
4. Горп Г.В., Илюхин А.А. Случаи постоянства модуля момента количества движения гиростата // Механика твердого тела. – 1974. – Вып. 6. – С. 9–15.
5. Ковалев А.М., Горп Г.В., Неспирный В.Н. Инвариантные соотношения неавтономных систем дифференциальных уравнений // Доклады НАН Украины. Математика. – 2014, № 2. – С. 13–19.
6. C. Baiocchi: Problèmes à frontière libre et inéquations variationnelles / C. Baiocchi. –C. R. Acad. Sci. Paris, sér. A. – 976 – V. 283. – P.29-32.
7. Duvaut M.G. Solytion of two phases Stefan problem by variational inequality / M.G. Duvaut // In. Proc. Of the Symp. on moving boundary problems. – Oxford, mars, 1974. – P. 25-27.
8. Brézis H. Estimates on the support of solutions of parabolic variational inequalities. III. / A. Friedman // J. Math. – 1976. – V. 20, №1. – P.82-97.
9. Caffarelli L.A. The one-phase Stefan problem and the porous medium equation. Continuity of the solution in n -space dimensions / L.A. Caffarelli, A. Friedman. – USA : Proc. Mat. Acad. Sci, 1978. – V.75. – 2004 p.
10. Kinderlehrer D. The smoothness of the free boundary in the phase Stefan problem

Горр¹ Г.В., Илюхин² А.А., Ковалев¹ А.М.

1 Державна установа "Інститут прикладної математики і механіки", ДНР, відділ прикладної механіки, відділ технічної механіки;

2 Таганрозький державний педагогічний інститут, (м. Таганрог, Росія), кафедра математики
Математические аспекты в моделировании движений гиростата с неподвижной точкой.

Доповідь присвячено дослідженню математичних підходів у моделюванні складних механічних систем класу "гіростат". Розглянуто дві задачі про рух гіростата з нерухомою точкою. Перша задача присвячена моделюванню руху гіростата з постійним гіростатическим моментом у полі сили тяжіння. Для неї вказані методи інтегрування рівнянь рухів з допомогою перших інтегралів. Друга задача характеризується змінним гіростатическим моментом, обумовленим нерівномірним обертанням несомих тіл. Вказаний метод дослідження програмних (модельованих) рухів, заснований на теорії інваріантних співвідношень неавтономних диференціальних рівнянь.

Ключові слова: гіростат, диференціальні рівняння, інтеграл, енергія.

G. V. Gorr¹, A. A. Ilyukhin², A. M. Kovalev¹

1 State institution "Institute of applied mathematics and mechanics", DNR;

2 Taganrog state pedagogical Institute (Taganrog, Russia)

Mathematical aspects of modeling motions of a gyrostat with a fixed point.

The report focuses on the study of mathematical approaches in modeling of complex mechanical systems class "gyrostat". Considered two problems of motion of a gyrostat with a fixed point. The first task deals with the modeling of motion of a gyrostat with constant virostaticski point in the gravity field. For it illustrates the methods of numerical integration of equations of motions using the first integrals. The second task is characterized by a variable gyrostatic momentum, due to the uneven rotation of the carried bodies. This method of research software (simulated) movements based on the theory of invariant ratios of no autonomous differential equations.

Keywords: gyrostat, differential equations, integral energy.

Статья поступила в редакцию 20.05.2016

Рекомендована к публикации д-ром техн. наук В.Н. Павлышом

УДК 004.274

Применение гибридных FPGA для реализации композиционного устройства управления с идентификацией выходов

К.Н. Ефименко

ГОУВПО «Донецкий национальный технический университет»
KN_Efimenko@mail.ru

Ефименко К.Н. Применение гибридных FPGA для реализации композиционного устройства управления с идентификацией выходов. Предлагается метод снижения аппаратных затрат в схеме композиционного микропрограммного устройства управления с идентификацией выходов, ориентированный на технологию гибридных FPGA. Метод основан на использовании трех источников кодов классов псевдоэквивалентных операторных линейных цепей и реализации схемы адресации микрокоманд на программируемых логических матрицах. Такой подход позволяет уменьшить площадь кристалла, занимаемую схемой устройства управления.

Введение

Использование программируемых логических интегральных схем (ПЛИС) вида FPGA (field-programmable gate arrays) [1,2] предоставляет широкие возможности для реализации схем устройств управления (УУ). При этом основу FPGA составляют макроячейки LUT (look-up table), имеющие ограниченное число входов (4-6). Также в настоящее время развивается технология гибридных FPGA [3,4], в состав которых входят LUT элементы, встроенные блоки памяти (EMB, embedded memory blocks) и встроенные программируемые логические матрицы (PLA, programmable logic array). В случае если алгоритм управления некоторой системы представлен линейной граф-схемой алгоритма (ГСА) [5], для реализации схемы УУ может быть использована модель композиционного микропрограммного устройства управления (КМУУ) с идентификацией выходов [6,7]. При этом появляется возможность реализации части схемы на блоках PLA. Это позволяет уменьшить площадь кристалла, занимаемую схемой КМУУ. В настоящей работе предлагается метод усовершенствования КМУУ с идентификацией выходов, основанный на использовании трех источников кодов классов псевдоэквивалентных операторных линейных цепей (ОЛЦ) за счет использования возможностей, представляемых технологией гибридных FPGA. Задачей исследования является разработка метода синтеза КМУУ с идентификацией выходов, позволяющего уменьшить число LUT-элементов в комбинационной части УУ.

КМУУ с идентификацией выходов

Пусть алгоритм управления цифровой системы задан ГСА $\Gamma = \Gamma(B, E)$, представленной

множествами вершин B и соединяющих их дуг E . Тогда $B = \{b_0, b_E\} \cup E_1 \cup E_2$, где b_0 – начальная вершина, b_E – конечная вершина, E_1 – множество операторных вершин и E_2 – множество условных вершин ГСА Γ . В операторных вершинах $b_q \in E_1$ записываются наборы микроопераций $Y(b_q) \subseteq Y$, где $Y = \{y_1, \dots, y_N\}$ – множество микроопераций. В условных вершинах $b_q \in E_2$ записываются элементы множества логических условий $X = \{x_1, \dots, x_L\}$. Введем ряд определений, взятых из [5,8].

Определение 1. Операторной линейной цепью ГСА Γ называется конечная последовательность операторных вершин $\alpha_g = \langle b_{g1}, \dots, b_{gE_g} \rangle$ такая, что для любой пары соседних компонент b_{gi}, b_{gi+1} , где i – номер компоненты кортежа α_g , существует дуга $\langle b_{gi}, b_{gi+1} \rangle \in E$.

Определение 2. Вершина $b_q \in D^g$, где D^g – множество вершин, входящих в ОЛЦ α_g , называется входом ОЛЦ α_g , если существует дуга $\langle b_t, b_q \rangle \in E$, где $b_t \notin D^g$.

Определение 3. Вершина $b_q \in D^g$, называется выходом ОЛЦ α_g , если существует дуга $\langle b_q, b_t \rangle \in E$, где $b_t \notin D^g$.

Определение 4. ОЛЦ α, α' называются псевдоэквивалентными ОЛЦ, если их выходы связаны со входом одной и той же вершины $b_q \in B$.

Пусть для ГСА Γ сформировано множество ОЛЦ $S = \{\alpha, \dots, \alpha_c\}$, определяющее разбиение на множестве E_1 [5,8], и пусть $|E_1| = M$. Поставим в

соответствие каждой вершине $b_q \in E_1$ микрокоманду MI_q с адресом $A(b_q)$, имеющим разрядность

$$R = \lceil \log_2 M \rceil. \quad (1)$$

Используем для адресации микрокоманд переменные $T_r \in T$, где $|T|=R$. Адресация выполняется таким образом, чтобы выполнялось условие

$$A(b_{gi+1}) = A(b_{gi}) + 1, \quad (2)$$

где $b_{gi}, b_{gi+1} \in D^g$ и $\langle b_{gi}, b_{gi+1} \rangle \in E$.

В этом случае ГСА Γ может быть интерпретирована КМУУ с идентификацией выходов (рис. 1), что предлагается в работах [6,7], называемым в дальнейшем КМУУ U_1 .

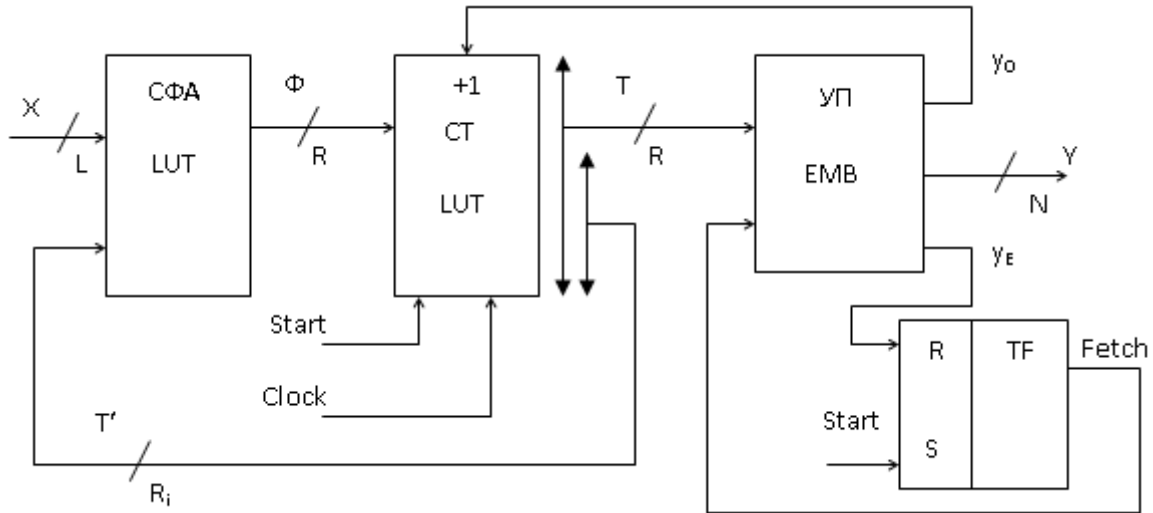


Рисунок 1 – Структурная схема КМУУ U_1

В КМУУ U_1 схема формирования адреса (СФА) реализует систему функций возбуждения триггеров счетчика СТ, определяемую как

$$\Phi = \Phi(T', X). \quad (3)$$

Переменные $T_r \in T$, где $|T|=R$, используются для адресации микрокоманд. Однако для однозначной идентификации ОЛЦ $\alpha_g \in C$ достаточно

$$R_1 = \lceil \log_2 G \rceil. \quad (4)$$

переменных, где $G = |C|$. Таким образом, за счет применения процедуры специальной адресации микрокоманд [6,7] формируется $T' = \{T_1, \dots, T_{R_1}\}$ – множество адресных разрядов, достаточное для однозначной идентификации выходов ОЛЦ $\alpha_g \in C'$, где $C' \subseteq C$ – множество ОЛЦ, выходы которых не связаны с входом вершины b_E .

По сигналу Start счетчик СТ устанавливается в ноль, что соответствует адресу первой микрокоманды реализуемого алгоритма. По сигналу $y_0 = 1$ содержимое СТ увеличивается на единицу, что соответствует режиму (2). Управляющая память (УП) хранит наборы микроопераций $Y(b_q) \subseteq Y$ и переменные y_0 (управление СТ) и y_E (признак окончания алгоритма). Триггер считывания TF формирует сигнал Fetch, разрешающий выборку микрокоманд из УП. При достижении окончания алгоритма формируется переменная $y_E = 1$, что приводит к Fetch = 0 и прекращению выборки из

УП.

Методика синтеза КМУУ U_1 на FPGA предложена в [6,7]. При этом схемы СФА, СТ и TF реализуются на LUT, а схема УП – на встроенных блоках памяти ЕМВ. Основным недостатком КМУУ U_1 является значительное число термов в системе функций (3). Это приводит к увеличению числа LUT элементов и их уровней в схеме СФА. Уменьшение числа термов системе (3) гарантируется в КМУУ с преобразованием адресов микрокоманд в коды классов псевдоэквивалентных ОЛЦ [8]. Однако это связано с введением в схему КМУУ дополнительного блока преобразователя адресов (БПА), который увеличивает аппаратные затраты кристалла. В настоящей работе предлагается метод синтеза КМУУ, позволяющий уменьшить как число термов в системе (3), так и число LUT элементов в схемах БПА и СФА.

Основная идея метода

Найдем разбиение $\tilde{I}_N = \{B_1, \dots, B_I\}$ множества ОЛЦ C на классы псевдоэквивалентных ОЛЦ. Выполним адресацию микрокоманд так, чтобы при выполнении условия (2) максимально возможное число классов $B_i \in \tilde{I}_C$ выражалось одним интервалом R -разрядного булева пространства. Представим разбиение $\tilde{I}_C$ в виде $\tilde{I}_C = \tilde{I}_1 \cup \tilde{I}_2$, где $\tilde{I}_1 \cap \tilde{I}_2 = \emptyset$. Пусть $B_i \in \tilde{I}_1$,

если этот класс представляется более, чем одним интервалом пространства кодирования. Остальные классы принадлежат множеству $\tilde{I}_2$.

Характерной особенностью блоков ЕМВ является фиксированное число выходов $t_F \in \{1, 2, 4, 8, 16, 32, 64\}$. Пусть V_0 означает число ячеек ЕМВ при $t_F = 1$. Тогда число ячеек V при некотором фиксированном значении t_F можно определить как

$$V = \lceil V_0 / t_F \rceil. \quad (5)$$

Для реализации управляющей памяти КМУУ достаточно M ячеек ЕМВ. При этом блок будет иметь

$$t_M = \lceil V_0 / M \rceil \quad (6)$$

выходов. Пусть следующее условие выполняется для некоторой ГСА Γ и блоки ЕМВ используемой микросхемы FPGA

$$t_M > N + 3. \quad (7)$$

Закодируем классы $B_i \in \tilde{I}_1$ двоичными кодами $K(B_i)$ разрядности

$$R_1 = \lceil \log_2 I_1 \rceil, \quad (8)$$

где $I_1 = |\Pi_1|$. Используем для кодирования

переменные $\tau_i \in \tau$, где $|\tau| = R_1$. Пусть $\tilde{I}_1 \neq \emptyset$, $\tilde{I}_2 \neq \emptyset$ и следующее условие выполняется для ЕМВ одновременно с условием (6):

$$t_M < N + 3 + R_1. \quad (9)$$

Очевидно, что часть разрядов кодов $K(B_i)$ для классов $B_i \in \tilde{I}_1$ может быть сформирована схемой УП. Теперь УП формирует R_2 разряда кода $K(B_i)$, а БПА – R_3 разряда:

$$R_2 = t_M - (N + 3); \quad (10)$$

$$R_3 = R_1 - R_2. \quad (11)$$

При этом множество τ может быть представлено в виде $\tau = \tau^1 \cup \tau^2$, где $|\tau^1| = R_3$, $|\tau^2| = R_2$.

Предлагаемая в работе модель КМУУ основана на представлениях $\tilde{I}_C = \tilde{I}_1 \cup \tilde{I}_2$ и $\tau = \tau^1 \cup \tau^2$.

При этом в предлагаемой структуре КМУУ U_2 схема СФА реализуется на блоках PLA (рис. 2).

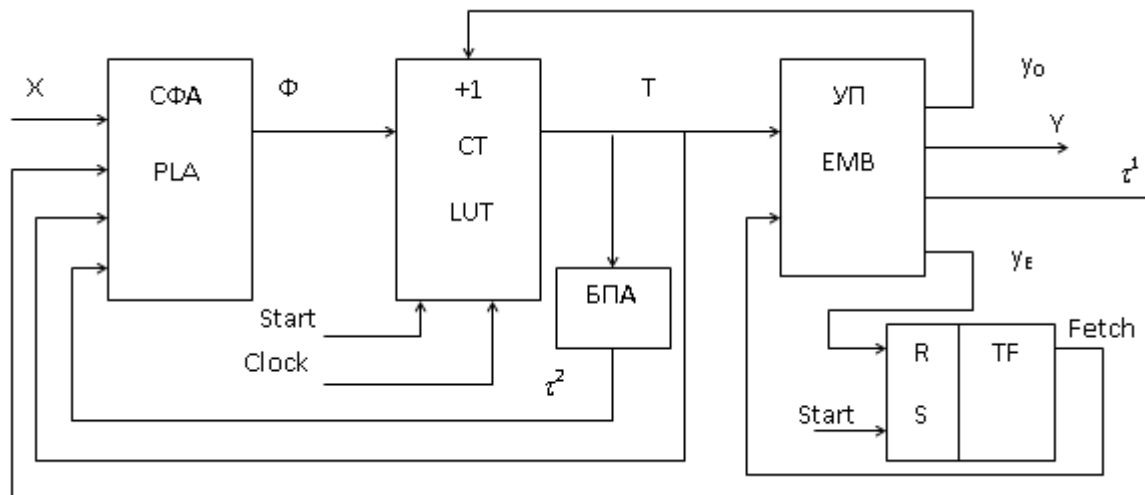


Рисунок 2 – Структурная схема КМУУ U_2

Схема СФА реализует функции Φ в следующем виде:

$$\Phi = \Phi(T, \tau, X). \quad (12)$$

Блок БПА реализует часть разрядов кодов $K(B_i)$, образующих множество τ^2 . Такой подход позволяет уменьшить число термов в системе (3) до минимума. Кроме того, уменьшается сложность блока БПА по сравнению с аналогичным КМУУ из [8]. Недостатком является увеличение числа выходов блока УП. Однако этот блок строится из реконфигурируемых блоков ЕМВ, которые имеют строго определённое число выходов. При этом имеется высокая вероятность наличия неиспользованных выходов [4].

В настоящей работе предлагается метод синтеза

КМУУ U_2 , включающий следующие этапы:

1. Формирование множества ОЛЦ S для ГСА Γ .
2. Формирование разбиения $\tilde{I}_C$ множества S .
3. Оптимальная адресация микрокоманд.
4. Кодирование классов $B_i \in \tilde{I}_1$.
5. Формирование таблицы переходов КМУУ.
6. Формирование таблицы блока преобразователя адреса.
7. Формирование таблицы содержимого управляющей памяти.
8. Реализация схемы КМУУ в заданном элементном базисе.

Пример применения метода

Пусть для некоторой ГСА Γ получено множество

ОЛЦ $C = \{\alpha_1, \dots, \alpha_{12}\}$, где $\alpha_1 = \langle b_1 \rangle$,
 $\alpha_2 = \langle b_2, \dots, b_5 \rangle$, $\alpha_3 = \langle b_6, b_7 \rangle$, $\alpha_4 = \langle b_8 \rangle$,
 $\alpha_5 = \langle b_9, b_{10} \rangle$, $\alpha_6 = \langle b_{11}, b_{12} \rangle$, $\alpha_7 = \langle b_{13}, \dots, b_{16} \rangle$,
 $\alpha_8 = \langle b_{17}, \dots, b_{20} \rangle$, $\alpha_9 = \langle b_{21}, \dots, b_{26} \rangle$, $\alpha_{10} = \langle b_{27}, b_{28} \rangle$,
 $\alpha_{11} = \langle b_{29}, \dots, b_{31} \rangle$, $\alpha_{12} = \langle b_{32} \rangle$. Пусть эти ОЛЦ
могут быть разбиты на $I = 7$ классов, где
 $B_1 = \{\alpha_1, \alpha_2\}$, $B_2 = \{\alpha_3, \alpha_4\}$, $B_3 = \{\alpha_5, \alpha_6\}$,
 $B_4 = \{\alpha_7\}$, $B_5 = \{\alpha_8, \alpha_9\}$, $B_6 = \{\alpha_{10}, \alpha_{11}\}$,
 $B_7 = \{\alpha_{12}\}$. Пусть выход ОЛЦ $\alpha_i \in C$ связан с
входом вершины b_E . Как известно, переходы из
таких ОЛЦ не рассматриваются, так как их
последняя вершина должна включать

$T_1 T_2 T_3$

000 001 010 011 100 101 110 111

$T_4 T_5$

00	b_1	b_5	b_9	b_{13}	b_{17}	b_{21}	b_{25}	b_{29}
01	b_2	b_6	b_{10}	b_{14}	b_{18}	b_{22}	b_{26}	b_{30}
10	b_3	b_7	b_{11}	b_{15}	b_{19}	b_{23}	b_{27}	b_{31}
11	b_4	b_8	b_{12}	b_{16}	b_{20}	b_{24}	b_{28}	b_{32}
	B_1	B_2	B_3	B_4	B_5	B_6	B_7	

Рисунок 3 – Адреса микрокоманд КМУУ $U_2(\Gamma)$

Следующие интервалы могут быть найдены для классов $B_i \in \tilde{I}_C$. Класс B_1 соответствует интервалам 000** и 00100; класс B_2 – интервалам 00101 и 0011*; класс B_3 – интервалу 010**; класс B_4 – интервалу 011**; класс B_5 – интервалам 10*** и 1100*; класс B_6 – интервалам 1101*, 1110* и 11110. Итак, имеем следующие классы для разбиения $\tilde{I}_C$: $\tilde{I}_1 = \{B_1, B_2, B_5, B_6\}$ и $\tilde{I}_2 = \{B_3, B_4\}$. Для кодирования классов $B_i \in \tilde{I}_1$ необходимо $R_1 = 2$ переменных $\tau_r \in \{\tau\}$.

Однако необходим код, позволяющий идентифицировать тот факт, что $B_i \notin \tilde{I}_1$. В этой связи (в данном примере) разрядность R_1 увеличивается до 3. Пусть код 000 соответствует условию $B_i \notin \tilde{I}_1$. Закодируем классы $B_i \in \tilde{I}_1$ следующим образом: $K(B_1) = 001$, $K(B_2) = 010$, $K(B_5) = 100$ и $K(B_6) = 101$.

Для формирования таблицы переходов необходимо построить систему обобщенных формул перехода [8] для классов $B_i \in \tilde{I}_C$. Пусть для классов $B_1, B_3 \in \tilde{I}_C$ получены следующие формулы:

переменную y_E [8].

Выполним адресацию микрокоманд так, чтобы выполнялось условие (2) и максимально возможное число классов представлялось одним обобщенным интервалом R -мерного булева пространства. В рассматриваемом примере $M = 32$, то есть $R = 5$ и $T = \{T_1, \dots, T_5\}$. Один из возможных вариантов оптимальной адресации микрокоманд приведен на рис. 3. Этот рисунок содержит видоизмененную карту Карно, которая достаточна для получения обобщенных интервалов, соответствующих кодам классов $B_i \in \tilde{I}_C$. Символ $U_2(\Gamma)$ означает, что КМУУ U_2 реализуется по ГСА Γ .

$$B_1 \rightarrow x_1 b_6 \vee \overline{x_1} b_9; \quad (13)$$

$$B_3 \rightarrow x_2 b_{13} \vee \overline{x_2} x_3 b_{17} \vee \overline{x_2} x_3 b_6.$$

Фрагмент таблицы переходов КМУУ $U_2(\Gamma)$, соответствующий (13), приведен в табл. 1.

Таблица 1. Фрагмент таблицы переходов КМУУ $U_2(\Gamma)$

B_i	$K(B_i)1$	$K(B_i)2$	b_q	$A(b_q)$	X_h	Φ_h
B_1	001	*****	b_6	00101	$\overline{x_1}$	$D_3 D_5$
			b_9	01000	$\overline{x_1}$	D_2
B_3	000	010**	b_{13}	01100	$\overline{x_2}$	$D_2 D_3$
			b_{17}	10000	$\overline{x_2} x_3$	D_1
			b_6	00101	$\overline{x_2} x_3$	$D_3 D_5$

В столбце $K(B_i)1$ указан код класса $B_i \in \tilde{I}_1$, в столбце $K(B_i)2$ – код класса $B_i \in \tilde{I}_2$. Если рассматриваются переходы из класса $B_i \in \tilde{I}_2$, то в строке записан код 000. В противном случае, код из столбца $K(B_i)2$ игнорируется.

Таблица переходов служит для формирования функций (12). Например, из табл. 1 может быть

получена функция

$$D_2 = \tau_1 \tau_2 \tau_3 x_1 \vee \tau_1 \tau_2 \tau_3 \tau_1 \tau_2 \tau_3 x_2.$$

Остальные этапы синтеза выполняются аналогично их выполнению в [9]. Только этап 7 имеет некоторые отличия. Рассмотрим следующий пример.

Пусть в вершине b_{31} ГСА Г записан набор микроопераций y_5, y_7 . Из предыдущего материала ясно, что вершина b_{31} является выходом ОЛЦ α_1 , которая входит в класс B_6 . Класс B_6 имеет код $K(B_6) = 101$, а вершине b_{31} соответствует адрес 11110 (рис. 3). Таким образом, в ячейку ЕМВ с адресом 11110 должен быть помещен код набора y_5, y_7 и переменные τ_1 и τ_3 .

Итак, переменные $\tau_i \in \{\tau^1\}$ помещаются в ячейки УП, соответствующие выходам ОЛЦ, входящим в классы $B_i \in \tilde{I}_1$. Схема БПА реализуется на LUT элементах. Очевидно, что разбиение множества τ на классы τ^1 и τ^2 надо производить следующим образом. В класс τ^2 помещаются переменные $\tau_i \in \{\tau\}$, которым соответствуют схемы с наименьшим числом LUT элементов. Такой подход позволяет уменьшить аппаратные затраты в схеме БПА. Дальнейшая реализация схемы КМУУ U_2 сводится к реализации системы функций на PLA и УП на ЕМВ. Для решения этой задачи используются стандартные промышленные пакеты. Этот этап выходит за пределы статьи.

Отметим, что реализация предложенного подхода возможна только благодаря большому числу входов PLA. Например, в блоках PLA микросхем АРЕХ20К имеется $S=32$ входа [4]. В общем случае предложенный метод целесообразен при выполнении условия

$$R_1 + R + L \leq S. \quad (14)$$

Анализ стандартных примеров [10] показал, что условие (14) выполняется для 87% МПА из этой библиотеки.

Заключение

Предлагаемый в работе метод уменьшения аппаратных затрат в схеме КМУУ основан на учете особенностей гибридных FPGA, а также наличии классов псевдоэквивалентных ОЛЦ. Использование двух источников кодов классов позволяет гарантированно уменьшить число термов в системе функций возбуждения триггеров счетчика адресов микрокоманд до максимально возможной величины. Если КМУУ с идентификацией выходов рассматривать как автомат Мура, то предлагаемый подход позволяет уменьшить число термов до величины этого параметра у эквивалентного автомата Мили. Кроме того, уменьшается число LUT элементов в схеме преобразователя адреса, так как не все адреса выходов ОЛЦ подлежат преобразованию.

В случае выполнения условия (14) схема СФА реализуется в виде одного блока PLA. При этом существенно уменьшается площадь кристалла, занимаемая схемой СФА. Это во многом объясняется уменьшением числа межсоединений по сравнению с КМУУ U_1 .

Научная новизна предложенного метода заключается в использовании особенностей базиса гибридных FPGA (большой коэффициент объединения по входам блоков PLA) для уменьшения числа LUT элементов в схеме КМУУ. С практической точки зрения метод дает возможность уменьшить площадь кристалла FPGA, занимаемую схемой КМУУ с идентификацией выходов, что позволяет получать схемы, обладающие меньшей стоимостью, чем известные из литературы аналоги.

Дальнейшие исследования могут быть направлены на разработку метода синтеза КМУУ, уменьшающего число блоков PLA в схеме адресации при нарушении условия (14).

Литература

1. Maxfield, S. The Design Warrior's Guide to FPGAs/ S. Maxfield. – Amsterdam: Elsevier, 2004. – 541 pp.
2. Грушвицкий, Р.И. Проектирование систем на микросхемах с программируемой структурой/ Р.И. Грушвицкий, А.Х. Мурсаев, Е.П. Угрюмов. – С-Пб: БХВ-Петербург, 2006. – 736 с.
3. Kaviani, F. The Hybrid Field Programmable Architecture/ F. Kaviani, S. Brown// IEEE Design & Test of Computers, Vol.16, N 4, 1999. – P. 74-83.
4. Sigh, S. Performance Evaluation of Hybrid Reconfigurable Computing Architectures over Symmetrical FPGA/ S. Sigh, R. Singh, V. Bhatia// International Journal of Embedded Systems and Applications, Vol.2, N 3, 2012. – P. 107-116.
5. Barkalov, A. Logic synthesis for compositional microprogram control units/ A. Barkalov, L. Titarenko. – Berlin: Springer, 2008. – 272 pp.
6. Баркалов, А.А. Оптимизация метода синтеза композиционного микропрограммного устройства управления на FPGA/ А.А. Баркалов, Р. Вишневский, К.Н. Ефименко// Искусственный интеллект, №3, 2005. – С. 265-271.
7. Баркалов, А.А. Оптимизация схемы адресации устройства управления с идентификацией выходов/ А.А. Баркалов, К.Н. Ефименко// Наукові праці ДонНТУ. Серія „Проблеми моделювання та автоматизації проектування динамічних систем” (МАП-2007). Випуск 6 (127). – Донецьк: ДонНТУ, 2007. – С.21-29.
8. Баркалов, А.А. Синтез микропрограммных автоматов на заказных и программируемых СБИС/ А.А. Баркалов, Л.А. Титаренко. – Донецк: УНИТЕХ, 2009. – 336 с.
9. Баркалов, А.А. Оптимизация схемы КМУУ с общей памятью/ А.А. Баркалов, Л.А. Титаренко, К.Н. Ефименко, Я.М. Липински// УСИМ, №5, 2011.

– С. 47-52.

44 pp.

10. Yang, S. Logic synthesis and optimization benchmarks user guide. Technical report/ S. Yang. – Microelectronic Center of North Carolina, 1991. –

К.Н. Ефименко. ПРИМЕНЕНИЕ ГИБРИДНЫХ FPGA ДЛЯ РЕАЛИЗАЦИИ КОМПОЗИЦИОННОГО УСТРОЙСТВА УПРАВЛЕНИЯ С ИДЕНТИФИКАЦИЕЙ ВЫХОДОВ

Предлагается метод уменьшения аппаратных затрат в схеме композиционного микропрограммного устройства управления (КМУУ) с идентификацией выходов, ориентированный на технологию гибридных FPGA. Метод основан на использовании трех источников кодов классов псевдоэквивалентных операторных линейных цепей (ПОЛЦ) и реализации схемы адресации микрокоманд на программируемых логических матрицах. Такой подход позволяет уменьшить площадь кристалла, занимаемую схемой устройства управления. Приведен пример применения предложенного метода.

Ключевые слова: КМУУ с идентификацией выходов, ПОЛЦ, гибридные FPGA, логическая схема

К.М. Єфіменко. ЗАСТОСУВАННЯ ГІБРИДНИХ FPGA ДЛЯ РЕАЛІЗАЦІЇ КОМПОЗИЦІЙНОГО ПРИСТРОЮ КЕРУВАННЯ З ІДЕНТИФІКАЦІЄЮ ВИХОДІВ

Пропонується метод зменшення апаратних витрат в схемі композиційного мікропрограмного пристрою керування (КМПК) із ідентифікацією виходів, який орієнтовано на технологію гібридних FPGA. Метод засновано на використанні трьох джерел код класів псевдоеквівалентних операторних лінійних ланцюгів (ПОЛЛ) і реалізації схеми адресації мікрокоманд на програмованих логічних матрицях. Такий підхід дозволяє зменшити площу кристала, яку займає схема пристрою керування. Наведено приклад використання запропонованого методу.

Ключові слова: КМПК із ідентифікацією виходів, ПОЛЛ, гібридні FPGA, логічна схема

K.N. Efimenko. THE USE OF HYBRID FPGA FOR IMPLEMENTING COMPOSITIONAL CONTROL UNIT WITH IDENTIFICATION OF OUTPUTS

A method for reducing the hardware amount in the circuit of compositional microprogram control unit (CMCU) with identification of outputs is proposed oriented on hybrid FPGA technology. The method is based on the use of three sources of codes classes of pseudoequivalent operational linear chains (POLC) and implementing the block of microinstruction addressing with programmable logic arrays. Such approach allows reducing the chip area occupied by the circuit of CMCU. An example of the proposed method application is given.

Key words: CMCU with identification of outputs, POLC, hybrid FPGA, logic circuit

*Статья поступила в редакцию 20.05.2016
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом*

УДК 004.942, 514.18

Некоторые вопросы математического и компьютерного моделирования в строительстве

Е.В. Конопацкий

Донбасская национальная академия строительства и архитектуры
e.v.konopatskiy@donnasa.ru

Конопацкий Е.В. Некоторые вопросы математического и компьютерного моделирования в строительстве. В статье рассмотрены достоинства и недостатки существующих методов математического и компьютерного моделирования, применяемые в строительной отрасли, а также показаны преимущества геометрического моделирования, реализованного в БН-исчислении. Приведены примеры построения двух геометрических моделей процессов зависимости: предела прочности при сжатии мелкозернистого дегтебетона от четырех параметров: вязкости дегтя, концентрации полимера в органическом вяжущем, концентрации активатора поверхности минерального порошка и температуры, а также предела прочности при сжатии образцов газобетона после ТВО от двух параметров: напряженности электростатического поля и длительности электрообработки.

Общая постановка проблемы

В современном строительстве, как на стадии проектирования, так и на стадии эксплуатации, огромную роль играют системы автоматизированного проектирования и расчёта. Все они относятся к системам компьютерного моделирования. Следует отметить, что строительство относится к той отрасли хозяйственной деятельности, где проведение эксперимента с реальной системой, как минимум, не рентабельно, а иногда при строительстве уникальных зданий и сооружений, просто не возможно. Если рассматривать не реальную систему, а её модель, то можно выделить два вида: физическая модель и математическая модель. В свою очередь, математическая модель условно делится на аналитическую и компьютерную. Традиционно аналитическая модель используется для описания достаточно простых систем, когда есть возможность получить точное аналитическое решение. Для решения сложных задач, когда аналитическое решение является сложным и требует большого количества вычислительных ресурсов, гораздо эффективнее проявили себя методы компьютерного моделирования.

В данном случае, под моделью объекта понимается любой другой объект, отдельные свойства которого частично или полностью совпадают со свойствами исходного объекта. Следует отметить, что полученная математическая модель исчерпывающе полной быть не может. Она всегда ограничена и должна соответствовать исключительно целям моделирования, отражая ровно столько свойств исходного объекта, сколько

необходимо для данного конкретного исследования. Поэтому для любых моделей возникает вопрос: насколько она достоверна по отношению к реальной системе? Так адекватность математической модели оценивается по близости результатов расчётов к исходным данным моделирования. В качестве аналитических методов моделирования в строительстве широкое распространение получили методы математической статистики, для проверки эффективности которых, используются соответствующие критерии адекватности (например, критерий Фишера), которые показывают, насколько полученная математическая модель отличается от значений исходных данных. С другой стороны компьютерные системы компьютерного моделирования, используемые в строительстве, основаны в основном на методе конечных элементов, который, при всех его преимуществах, имеет ряд недостатков и ограничений. Например, при разбишке объекта на конечное число элементов, исходный сложный объект аппроксимируется кусочно-непрерывной функцией, что приводит к потере некоторого количества информации об исходном состоянии объекта. На следующем этапе составляется система линейных алгебраических уравнений, для решения которой обычно используются численные методы, которые также содержат некоторую погрешность. Таким образом, имеет место накопление ошибки, которую можно уменьшить при увеличении количества элементов, но здесь ограничивающим критерием служит вычислительная способность компьютерной техники.

Другим перспективным направлением может служить геометрическое моделирование многопараметрических процессов и явлений. В зависимости от данной конкретной задачи его можно отнести как к аналитическому моделированию, так и к компьютерному. На данном этапе геометрическое моделирование используется исключительно для аналитического или компьютерного описания геометрической формы строительных конструкций, а также для визуализации полученных результатов моделирования в виде дуг кривых, их комбинаций (номограмм) и отсеков поверхностей. Между тем во многих случаях, ставя в соответствие любому процессу или явлению геометрический объект, который получен на основе исходного массива данных, можно получать результаты гораздо более высокого качества, чем те которые получены другими способами математического и компьютерного моделирования.

Анализ последних исследований и публикаций

Перед решением задач оптимизации на основе полученной математической модели важно убедиться в её адекватности. Иначе и результат оптимизации можно будет считать сомнительным. Как уже было отмечено выше, достоверность полученных результатов оценивается с помощью исходных данных для моделирования. Среди зарубежных и отечественных учёных, которые исследуют свойства строительных систем, будь то влияние различных факторов на системы строительных конструкций или многокомпонентные системы, которые используются для моделирования физико-механических свойств строительных материалов, получили широкое распространение методы математической статистики, одним из которых является регрессионный анализ. Значительным недостатком регрессионного анализа является слабая его устойчивость к исходным данным. Т.е. с точки зрения адекватности, модели, полученные на основе регрессионного анализа, всегда будут отличаться от исходных данных моделирования. При большом массиве исходных данных для моделирования, когда геометрически мы имеем облако точек, такой подход оправдан, поскольку позволяет оценить характер протекания процесса, но при небольшом количестве исходных данных он даёт достаточно большую погрешность, что и было показано в работе [1].

Для геометрического моделирования процессов и явлений в строительстве используется новый математический аппарат – БН-исчисление (точечное исчисление Балюбы-Найдыша), который позволяет определять геометрические объекты по наперед заданным условиям в пространстве любой размерности. Основным элементом БН-исчисления

[2-3] является точка, которая характеризуется рядом параметров. Количество параметров, которые определяют точку в пространстве, зависит от размерности этого пространства. А любой геометрический объект является организованным множеством точек. Поэтому точечные уравнения, которые определяют геометрический объект в пространстве, справедливы для пространства любой размерности. Эта особенность БН-исчисления даёт возможность представлять геометрические объекты в многомерном пространстве (имеется в виду аффинное многомерное пространство). Исходя из этого, можно сделать вывод, что геометрическая модель, представленная в БН-исчислении, по сути, является организованным множеством точек, которые зависят от нескольких, связанных между собой, текущих параметров. Так однопараметрическим множеством точек в двумерном пространстве является линия, двухпараметрическим в трехмерном пространстве – поверхность, трехпараметрическим в четырехмерном пространстве – гиперповерхность, т.д.

С геометрической точки зрения, для обеспечения адекватности будущей модели, необходимо на стадии моделирования достаточно обеспечить моделирование геометрического объекта с наперед заданными условиями. Геометрически это достигается тем, что моделируемый объект проходит через наперед заданные точки. Т.е. если рассматривается однопараметрический процесс, то ему в соответствие ставится дуга кривой отклика, проходящей через наперед заданные точки. Подробнее о способе моделирования дуг кривых проходящих через наперед заданные точки можно познакомиться в работах автора и его учеников, например [4-5].

Чтобы получить геометрический объект, который соответствует двухпараметрическому процессу, необходимо упорядочить двухпараметрическое множество точек, которые соответствуют исходным данным эксперимента, и получить отсек поверхности отклика. Для реализации этой задачи в БН-исчислении был разработан специальный метод, получивший название метода подвижного симплекса [6-7].

Обобщая выше сказанное, получим, что для моделирования трёхпараметрического процесса, ему в соответствие ставится отсек гиперповерхности отклика, проходящий через наперед заданные точки и так далее. Таким образом, используя метод подвижного симплекса для создания геометрических объектов многомерного пространства по заданным условиям, можно моделировать многопараметрические процессы и явления любой степени сложности. Следует отметить, что конфигурация и способ моделирования геометрического объекта напрямую зависит от исходных данных моделирования.

Формулировка цели статьи

Показать преимущества использования геометрического моделирования процессов и явлений в строительной отрасли.

Основная часть

Рассмотрим пример моделирования зависимости предела прочности при сжатии мелкозернистого дегтебетона, зависящего от четырех параметров: вязкости дегтя, концентрации полимера в органическом вяжущем, концентрации активатора поверхности минерального порошка и температуры. В работе [8] эта была получена регрессионная модель искомой зависимости, анализ которой показал, что отклонения от исходных данных моделирования достигают 50-60%.

Здесь следует остановиться на существующей матрице планирования эксперимента. Дело в том, что при использовании регрессионного анализа значения факторов варьирования кодируется тремя значениями: -1, 0 и +1. С геометрической точки зрения, если речь идёт об однопараметрическом объекте, который соответствует однопараметрическому процессу, то задача сводится к определению дуги кривой проходящей через 3 точки. Точечные уравнения таких дуг, были получены в работе [4]. Следует отметить, что полученные точечные уравнения позволяют интерполировать эти 3 точки с помощью параболической, гиперболической и эллиптической дуги кривой, что, в свою очередь, позволяет подобрать такой вид дуги кривой, который отображает максимально точно и достоверно характер протекания процесса. При этом не зависимо от вида дуги кривой, свойство её прохождения через 3 наперёд заданные точки остаётся неизменным.

Если планируется провести эксперимент с изменением двух параметров, то кодируемых факторов варьирования будет 6: 3 для одного параметра и 3 для второго, а их комбинаций – 9. Т.е. геометрически задача сводится к определению отсека поверхности, которая проходит через 9 точек и определяется двумя параметрами. Обобщая такой подход, получим необходимое количество экспериментальных данных: для трёхпараметрического эксперимента – 27 и т.д.

Так в работе [8] было представлено необходимое количество измерений и натуральные значения варьировемых факторов. Поскольку исследовалась зависимость от 4-х параметров, то количество таких экспериментов было 81. Решая эту задачу с помощью геометрического моделирования в БН-исчислении, была получена последовательность аналитических зависимостей [1, 9] и составлен вычислительный алгоритм, который можно разбить на 4 этапа.

1. Определим 27 направляющих дуг отсеков поверхностей. Для экономии места приведем только первые 9 уравнений опорных дуг, которые соответствуют пределу прочности дёгтебетона при температуре 0°C. Остальные 18 уравнений будут аналогичными только соответствовать пределу прочности дёгтебетона при 20°C и 50°C.

$$\begin{aligned} P_{52}^0 &= A_1^{52}\bar{u}(1-2u) + 4A_2^{52}\bar{u}u + A_3^{52}u(2u-1), \\ Q_{52}^0 &= B_1^{52}\bar{u}(1-2u) + 4B_2^{52}\bar{u}u + B_3^{52}u(2u-1), \\ R_{52}^0 &= C_1^{52}\bar{u}(1-2u) + 4C_2^{52}\bar{u}u + C_3^{52}u(2u-1), \\ P_{130}^0 &= A_1^{130}\bar{u}(1-2u) + 4A_2^{130}\bar{u}u + A_3^{130}u(2u-1), \\ Q_{130}^0 &= B_1^{130}\bar{u}(1-2u) + 4B_2^{130}\bar{u}u + B_3^{130}u(2u-1), \\ R_{130}^0 &= C_1^{130}\bar{u}(1-2u) + 4C_2^{130}\bar{u}u + C_3^{130}u(2u-1), \\ P_{208}^0 &= A_1^{208}\bar{u}(1-2u) + 4A_2^{208}\bar{u}u + A_3^{208}u(2u-1), \\ Q_{208}^0 &= B_1^{208}\bar{u}(1-2u) + 4B_2^{208}\bar{u}u + B_3^{208}u(2u-1), \\ R_{208}^0 &= C_1^{208}\bar{u}(1-2u) + 4C_2^{208}\bar{u}u + C_3^{208}u(2u-1), \end{aligned}$$

где $\bar{u} = 1 - u$ – дополнение до единицы.

2. Определим 9 направляющих дуг отсеков гиперповерхности, которые одновременно являются образующими соответствующих отсеков поверхности, полученных в первом пункте данного алгоритма:

$$\begin{aligned} M_{52}^0 &= P_{52}^0\bar{v}(1-2v) + 4Q_{52}^0\bar{v}v + R_{52}^0v(2v-1), \\ M_{130}^0 &= P_{130}^0\bar{v}(1-2v) + 4Q_{130}^0\bar{v}v + R_{130}^0v(2v-1), \\ M_{208}^0 &= P_{208}^0\bar{v}(1-2v) + 4Q_{208}^0\bar{v}v + R_{208}^0v(2v-1), \\ M_{52}^{20} &= P_{52}^{20}\bar{v}(1-2v) + 4Q_{52}^{20}\bar{v}v + R_{52}^{20}v(2v-1), \\ M_{130}^{20} &= P_{130}^{20}\bar{v}(1-2v) + 4Q_{130}^{20}\bar{v}v + R_{130}^{20}v(2v-1), \\ M_{208}^{20} &= P_{208}^{20}\bar{v}(1-2v) + 4Q_{208}^{20}\bar{v}v + R_{208}^{20}v(2v-1), \\ M_{52}^{50} &= P_{52}^{50}\bar{v}(1-2v) + 4Q_{52}^{50}\bar{v}v + R_{52}^{50}v(2v-1), \\ M_{130}^{50} &= P_{130}^{50}\bar{v}(1-2v) + 4Q_{130}^{50}\bar{v}v + R_{130}^{50}v(2v-1), \\ M_{208}^{50} &= P_{208}^{50}\bar{v}(1-2v) + 4Q_{208}^{50}\bar{v}v + R_{208}^{50}v(2v-1), \end{aligned}$$

где $\bar{v} = 1 - v$.

3. Определим 3 направляющие дуги кривой отсека гиперповерхности четырехмерного пространства, которые одновременно являются образующими соответствующих отсеков гиперповерхности, полученных во втором пункте данного алгоритма:

$$\begin{aligned} N_0 &= M_{52}^0\bar{w}(1-2w) + 4M_{130}^0\bar{w}w + M_{208}^0w(2w-1), \\ N_{20} &= M_{52}^{20}\bar{w}(1-2w) + 4M_{130}^{20}\bar{w}w + M_{208}^{20}w(2w-1), \\ N_{50} &= M_{52}^{50}\bar{w}(1-2w) + 4M_{130}^{50}\bar{w}w + M_{208}^{50}w(2w-1), \end{aligned}$$

где $\bar{w} = 1 - w$.

4. Определим образующую отсека гиперповерхности, которая зависит от четырех параметров u , v , w и t :

$$T = N_0\bar{t}(1-2t) + 4N_{20}\bar{t}t + N_{50}t(2t-1),$$

где $\bar{t} = 1 - t$.

Таким образом, получим геометрический объект, принадлежащий 5-ти мерному пространству, который определяется 4-мя текущими параметрами и проходит через 81 точку, соответствующую исходным данным для моделирования.

Как видно из приведенных точечных уравнений, что имеют однотипную структуру, в которой меняются только точки, которые соответствуют исходным данным, и параметры. В данном случае была использована параболическая интерполяция сложной гиперповерхности, проходящей через 81 точку. Как было отмечено выше, инструментарий БН-исчисления располагает и другими точечными уравнениями дуг кривых, проходящими через наперед заданные точки, что позволяет использовать гиперболические, эллиптические и другие дуги кривых для интерполяции модели процесса. Т.е. фактически, геометрическое моделирование ограничивается исключительно исходными данными моделирования и даёт возможность создавать математические модели процессов и явлений любой степени сложности и с любым количеством переменных параметров. Главное условие – достаточное количество исходных (экспериментальных) данных. Так для 4-х параметрического процесса, пример которого приведен выше, при условии проведения параболической интерполяции необходимо располагать исходными данными в количестве 81, а для 3-х параметрического процесса, описанного параболической интерполяцией, в количестве – 27. С другой стороны, если проведение такого количества экспериментов для получения исходных данных является, по какой либо причине, не целесообразным, то можно использовать линейную интерполяцию, для которой достаточно иметь гораздо меньшее количество экспериментальных данных. Конечно, достоверность такой модели будет значительно ниже. С другой стороны, в предложенной математической модели использовалась существующая матрица планирования, в которой количество экспериментов относительно каждого параметра было равно трём. Таким образом, получилась зависимость необходимого количества исходных данных для моделирования n от количества параметров:

$$n = k^m,$$

где k – количество измерений, полученных для каждого параметра;

m – количество параметров.

Следует отметить, что данная зависимость справедлива исключительно для случая, когда количество экспериментов относительно каждого параметра является величиной постоянной, что достаточно часто встречается при исследовании и математическом моделировании физико-

механических свойств строительных материалов. Но сам аппарат геометрического моделирования, реализованный в БН-исчислении, является универсальным и может создавать математические модели для любого количества экспериментальных измерений и любого количества параметров. При этом рост количества измерений приводит к увеличению порядка аппроксимирующей кривой, а рост количества параметров – к увеличению размерности пространства в котором рассматривается геометрический объект.

Рассмотрим другой пример построения модели процесса зависимости предела прочности при сжатии образцов газобетона после ТВО от двух параметров: напряженности электростатического поля и длительности электрообработки. Поскольку искомый результат моделирования является функцией двух параметров, то геометрически его можно представить в виде отсека поверхности, расположенного в трёхмерном пространстве.

Воспользуемся план-матрицей проведения эксперимента и значениями варьируемых факторов, которые представлены в работе [10]. В соответствии с план-матрицей, получаем 9 различных значений предела прочности при сжатии, которые соответствуют различным комбинациям значений напряженности электростатического поля и длительности электрообработки. Причём оба параметра в работе [10] представлены в кодированном (-1, 0, +1) и в натуральном значении.

Для построения такой поверхности будем использовать методом подвижного симплекса [6-7] и точечное уравнение параболы, проходящей через три наперед заданные точки [9]. Выделим из девяти имеющихся точек, которым соответствуют натуральные значения факторов варьирования три направляющих дуги, которые соответствуют длительности электрообработки 10 мин., 20 мин. и 30 мин. Точечные уравнения направляющих дуг, согласованные с помощью параметра u , будут иметь следующий вид:

$$M_{10} = A_1^{10}\bar{u}(1-2u) + 4A_{1,5}^{10}\bar{u}u + A_2^{10}u(2u-1),$$

$$M_{20} = A_1^{20}\bar{u}(1-2u) + 4A_{1,5}^{20}\bar{u}u + A_2^{20}u(2u-1),$$

$$M_{30} = A_1^{30}\bar{u}(1-2u) + 4A_{1,5}^{30}\bar{u}u + A_2^{30}u(2u-1),$$

где A_i^j – соответствует среднему значению параметра оптимизации при i -й напряженности электростатического поля и j -й длительности электрообработки, которые принимаются в соответствии с план-матрицей.

По этим направляющим параболическим дугам движется симплекс трёх точек M_{10} , M_{20} и M_{30} , в котором зададим образующую дугу параболы с помощью аналогичного точечного уравнения, но уже в зависимости от параметра v :

$$M = M_{10}\bar{v}(1-2v) + 4M_{20}\bar{v}v + M_{30}v(2v-1).$$

В результате получим небольшой вычислительный алгоритм, который определяется последовательностью точечных уравнений зависящих от двух параметров u и v , которые, в свою очередь, определяют поверхность, проходящую через девять наперед заданных точек, соответствующих исходным данным план-матрицы эксперимента. Причём, значению параметра u , который меняется в пределах от 0 до 1, соответствует напряженность электростатического поля, изменяющаяся от 1 кВ/см до 2 кВ/см, а значению параметра v , который также меняется в пределах от 0 до 1, соответствует длительность электрообработки, изменяющаяся от 10 мин. до 30 мин.

Выводы и перспективы последующих исследований

В статье рассмотрены достоинства и недостатки существующих методов математического и компьютерного моделирования, применяемые в строительной отрасли, а также показаны преимущества геометрического моделирования, реализованного в БН-исчислении, что обеспечивает полную адекватность полученных результатов моделирования, поскольку условие соответствия геометрического объекта исходным данным было изначально заложено на стадии моделирования.

Другим преимуществом предложенного метода, является возможность одновременного учёта влияния всех параметров на систему в целом за счёт увеличения размерности пространства, заложенного в основы БН-исчисления, что было не всегда возможно при использовании традиционных методов аналитического моделирования и приводило к необходимости использования компьютерного моделирования, которое, в свою очередь, имеет свои недостатки и ограничения.

Литература

1. Найдыш А.В., Конопацкий Е.В., Бумага А.И. Теоретические основы геометрического моделирования физико-механических свойств асфальтобетонных методами БН-исчисления. // Науковий вісник Мелітопольського державного педагогічного університету ім. Б.Хмельницького. Серія: Математика. Геометрія. Інформатика. – Мелітополь: МДПУ ім. Б. Хмельницького, 2014. – Т.1. – С.111-117.
2. Балюба И.Г. Конструктивная геометрия многообразий в точечном исчислении: диссертация на соискание научной степени доктора технических наук: 05.01.01 / Балюба Иван Григорьевич – Макеевка: МИСИ, 1995. – 227 с.
3. Балюба И.Г., В.М.Найдыш Точечное исчисление: учебное пособие. – Мелітополь: МГПУ им. Б. Хмельницького, 2015. – 236 с.

4. Конопацкий Є.В. Геометричне моделювання алгебраїчних кривих та їх використання при конструюванні поверхонь у точковому численні Балюби-Найдиша. Дис. ... канд. техн. наук: 05.01.01. Конопацький Євген Вікторович. – Мелітополь, 2012. – 164 с.

5. Конопацький Є.В., Бумага А.І., Крисько О.А. Геометричне моделювання поверхні резервуару для зберігання нафтопродуктів з урахуванням недосконалостей методами БН-числення. // Матеріали II-ї Міжнародної науково-практичної конференції студентів, аспірантів та молодих вчених «Прикладна геометрія, дизайн та об'єкти інтелектуальної власності». Вип. 2. – К.: ДІА, 2013. – С.118-122.

6. Давыденко И.П. Конструирование поверхностей пространственных форм методом подвижного симплекса: Дис. ... канд. техн. наук: 05.01.01. Давыденко Иван Петрович. – Макеевка, 2012. – 186 с.

7. Конопацький Є.В., Поліщук В.І. Теоретичні основи точкового визначення поверхонь зі змінним симплексом. // Наукові нотатки. Міжвузівський збірник. Вип. 22. Ч. 2. – Луцьк: ЛДТУ. – 2008. – С.276-281.

8. Ходун В.Н. Дёгтебетоны с комплексно-модифицированной микроструктурой: дис. ... кандидата технических наук: 05.23.05 / Ходун Владимир Николаевич. – Макеевка, 1999. – 146 с.

9. Бумага А.І. Геометрична модель залежності фізико-механічних властивостей асфальтобетону від чотирьох параметрів у БН-численні / Бумага А.І.// Сучасні проблеми моделювання: зб. наук. праць / МДПУ ім. Б.Хмельницького. – Мелітополь: МДПУ ім. Б. Хмельницького, 2014. – Вип. 3. – С.28-33.

10. Мартынова Р.Б. Модифицированный неавтоклавный газополистиролбетон с повышенными физическими и механическими свойствами: дисс.... канд. техн. наук: Макеевка: ДонНАСА, 2012. – 197 с.

Конопацкий Е.В. Некоторые вопросы математического и компьютерного моделирования в строительстве. В статье рассмотрены достоинства и недостатки существующих методов математического и компьютерного моделирования, применяемые в строительной отрасли, а также показаны преимущества геометрического моделирования, реализованного в БН-исчислении. Приведены примеры построения двух геометрических моделей процессов зависимости: предела прочности при сжатии мелкозернистого дегтебетона от четырех параметров: вязкости дегтя, концентрации полимера в органическом вяжущем, концентрации активатора поверхности минерального порошка и температуры, а также предела прочности при сжатии образцов газобетона после ТВО от двух параметров: напряженности электростатического поля и длительности электрообработки.

Konopatskiy E.V. Some questions of mathematical and computer modeling in building. The article examines the advantages and disadvantages of existing methods of mathematical and computer modeling used in the building industry, as well as the advantages of geometric modeling implemented in the BN-calculation. Examples of the construction of two geometric models of processes depending on: the compressive strength of fine tarmacadam on four parameters: viscosity of the tar, the polymer concentration in an organic binder, a surface activator concentration of mineral powder and temperature, as well as the strength of the compressive samples of aerated concrete after TWT on two parameters: electrostatic field strength and duration of the electric treatment.

Статья поступила в редакцию 20.05.2016
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом

Анализ возможностей и недостатков технологии построения баз знаний в современных САПР

О.В. Морозова

Донецкий национальный технический университет, г. Донецк
кафедра программной инженерии
olmalyavka@gmail.com

Морозова О.В. Анализ возможностей и недостатков технологии построения баз знаний в современных САПР. Рассмотрены стандартные инструментальные САПР общего назначения. Так же рассмотрены типичные интеллектуальные системы для создания объектов в конкретной предметной области. Определены достоинства и недостатки этих оболочек с точки зрения наличия механизма создания базы знаний и механизмов логического вывода. Рассмотрена структура и-или дерева. Представлена классификация видов продукции по механизму их срабатывания. Приставлена характеристика формирования модуля знаний. Показана схема взаимосвязи модулей знаний.

Ключевые слова: CATIA, AutoCAD Plant 3D, И-ИЛИ дерево, продукция, посылка, следствие, модуль знаний.

Введение

Современное развитие систем искусственного интеллекта привело к созданию большого количества универсальных программных средств для построения экспертных систем различного назначения, так называемых «оболочек». В основе большинства таких продуктов лежит механизм накопления знаний о предметной области и способов их обработки. Большинство существующих инструментальных оболочек являются оболочками общего назначения, к ним можно отнести такие инструментальные системы как G2, Clips, Prolog, Nexpert Object, Exys Corvid, Jess и т.д. [1]. Все эти системы имеют малую степень адаптации на конкретную предметную область. Инструментальные системы общего назначения состоят из различных компонентов для обработки и представления знаний. В них используются стандартные подходы, такие как фреймы, семантические сети, системы продукции, нечеткие множества, нейронные сети, генетические алгоритмы. По этому, есть актуальные вопросы построения инструментальных систем адаптированных на конкретную предметную область, в концепции «умного эксперта» способного задать методику проектирования без привлечения инженера по знаниям [2].

Достоинства инструментальных средств состоят в, что все базы знаний строятся на основе правил, которые записываются в традиционном и удобном для пользователя виде «если .. то ..», так же есть с возможность работы с нечеткими множествами (Clips), работы в реальном времени, работа с шаблонами и удобные графические интерфейсы для ввода продукции. Но при их рассмотрении были выявлены и существенные недостатки, такие как большие затраты времени на прямой и обратный логический вывод; сложный для понимания пользователя механизм логического вывода, который предполагает возвраты в определенные узлы, с сохранением позиций

нахождения правила в базе знаний, несогласованность между организацией поиска в глубину и деревом поиска; количество этапов логического вывода может определяться экспоненциальной зависимостью от количества базовых фактов.

Данные программные комплексы используют разнообразные специфические способы представления и обработки знаний, основанные на стандартных подходах. Поэтому необходимо рассмотреть формирования единичного модуля знаний и представление продукции в виде и-или дерева на конкретной предметной области.

Цель статьи – провести анализ особенностей существующих интеллектуальных САПР в конкретной предметной области, таких как CATIA, AutoCAD Plant 3D, и выявить в них достоинства и недостатки, изучить существующие структуры и-или дерева, соответствующей специфике метода представления знаний и структуры базы знаний. Привести разработанные виды продукции по механизму срабатывания и метод представления модуля знаний.

Современные интеллектуальные САПР

Современные предприятия для обеспечения получения и накопления информации, и ее дальнейшее использование заставляют создавать базы знаний. CATIA V5 предлагает предприятиям самые современные решения для внедрения процесса накопления и эффективного использования знаний [3]. Система CATIA V5 состоит из нескольких платформ, которые представляют подмножества продуктов, данные продукты в свою очередь находятся на разных уровнях разработчиков (функциональном, производственном, структурном и др.)

Блок по обработке знаний называется KnowledgeEngineering. В данном блоке хранятся модули по обработке знаний, которые обеспечивают спецификацию в процессе создания изделия.

CATIA V5 состоит из многих инструментальных блоков, но для создания интеллектуальной части САПР, отвечает инструментарий «База знаний». В данной части системы CATIA V5 происходит создания нового изделия, расширение его на уровне аналитических и технических процессов, как автоматический просчет параметров, проверка соответствия проектированных правил и проверка ограничений.

Вид правила в системе CATIA V5 (см. рис 1).

**If Материал==‘Золото‘ Цена=1000
else Цена= min(M1,M2,...)**

Рисунок 1- Пример правила CATIA V5

Для составления правил можно использовать любые спецификации из разных разделов процесса создания готового продукта. По различным разделам можно понимать пространственные параметры, логические связи с другими компонентами изделия, а так же можно вводить экономические (стоимость) и технологические (время обработки) ограничения.

Важной особенностью «базы знаний» системы является способность сохранения любого введенного правила, то есть использование готовых продуктов при создании новых из готовых спецификаций.

Важными преимуществами с точки зрения структурного проектирования:

- система отображения структурной сборки;
- быстрая стадия прототипирования;
- возможность анализа детали в пространстве.

Недостатками системы являются:

- слабый модуль создания чертежей;
- долгое обучение для работы в системе;
- разделение системы на много платформ;
- сложное управление системой.

AutoCAD Plant 3D – инструмент для технологического проектирования и создания цифровых прототипов (цифрового моделирования промышленных объектов), а также создания технической документации [4].

Инструментальная система AutoCAD Plant 3D предназначенная для технического проектирования и создания цифровых прототипов, и создание технической документации. Данная инструментальная система на платформе AutoCAD, является самой распространённой на пост советской территории системой проектирования. Ее функциональные возможности позволяют создавать схемы и трехмерные модели объектов. Система Autodesk Navisworks предназначена для нахождения ошибок объекта и получения визуализации объекта. Система позволяет сохранять готовые продукты в разных форматах, таких как DWG, PCF и др. Инструмент 3D-моделирования позволяет создавать двухмерные и изометрические чертежи различной сложности, что увеличивает производительность системы. Организация работы AutoCAD Plant 3D происходит таким образом, что при создании модели объекта происходит и формирование

документации к данной модели, что позволяет обеспечить согласованность данных и связей модели.

Дополнительными модулями системы AutoCAD Plant 3D, является Visual LISP и AutoLISP. Данные модули позволяют расширять технические возможности системы. Условно их можно назвать «базой знаний», так как в них создается некое подобие единичного модуля знаний системы, который расширяет технические спецификации модели. Данные модули можно использовать многократно (использовать как базовый элемент модели), и расширять его технические и функциональные характеристики.

Основные достоинства инструментальной системы:

- база для создания больших расчетно-графических моделей;
 - возможность получения из не полной модели (двух проекций), получение полной модели объекта;
 - любое количество объектов объемной фигуры.
- Недостатки системы AutoCAD Plant 3D:
- сложность привязки информации из базы данных к графическим объектам;
 - системе отсутствует расчетный модуль;
 - предназначена для решения узких задач.

Структура и-или дерева

Одним из способов представлений модели знаний является и-или дерево. И-или дерево это морфологическое множество в виде графа. Граф имеет два вида узлов: и-узлы и или-узлы. И-узлы характеризуются конечное состояние подсистемы и не имеют дочерних узлов. Или-узел является признаком с возможными вариантами выбора одного из узлов. Корневой узел дерева описывает исходную цель. Существует и понятие дочернего узла, для каждого узла дерева соответствует описаниям подцелей. Конечные вершины (листья), не имеющие дочерних узлов, соответствуют описаниям элементарных целей, не разбиваемых на подцели. [5].

Морфологическое и-или дерево включает все структурные идентификаторы объектов, данного класса, получаются путем вырождения ИЛИ-вершин [6].

И-или дерево небольшого множества структурных решений достаточно просто показывает ограничения на введенные входные данные модели.

И-или дерево показывает все структурные особенности модели объекта, но не показывает функциональных зависимостей между ними.

И-ИЛИ дерево как грамматика

Основной структурой, представляющий данные внутри системы есть и-или дерево. На рис. 1 представлена структура деревьев, существующих в инструментальном наборе и разрабатываемых в конкретном примере.

При организации базы знаний необходимо перечислить основные аспекты построения, такие как

структурно-функциональная модель проектирования объекта, как совокупность модулей знаний; задание порядка срабатывания входных продукций разного вида.

Для построения и-или дерева необходимо рассмотреть виды продукций, которые могут быть использованы в инструментальной оболочке. Продукции делятся на виды, по механизму срабатывания посылки [7].

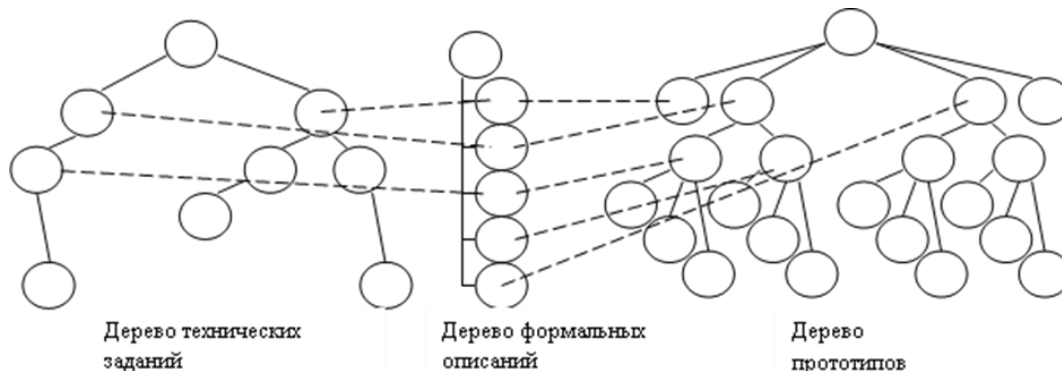


Рисунок 2 – Дерево нахождения формальных решений

Виды продукций по механизму срабатывания:

- проверяющая продукция – это такая продукция посылка, которой, при срабатывании проверяет наличие данной посылки базе знаний;
- не проверяющая продукция – это такая продукция, при которой посылка не проверяется на наличие совпадений в базе знаний;
- спрашивающая продукция – это продукция организовывающая работу с экспертом предметной области, которая в посылке отправляется вопрос;
- не спрашивающая продукция – это такая продукция, при которой посылка на прямую работает с базой знаний, без участия эксперта в предметной области;
- удаляющая продукция – это продукция, при которой наличие посылки означает удаление следствия из базы знаний;
- прибавляющая продукция – это продукция, при которой наличие посылки означает добавление следствия в базу знаний.

Для организации построения дерева формальных решений используется процедура выбора из ряда продукций, при срабатывании которых наиболее быстро заканчивается вывод. Для оптимизации построения дерева решений и удовлетворения принципа монотонности, будем использовать правильный порядок следования продукций. Из этого следует, что при организации базы знаний, новый факт влияет на имеющийся результат данного этапа и неизменности результатов предыдущих этапов вывода, при том итоговый результат должен удовлетворять всем условиям, заданным в фактах [8]. При этом предполагается учитывать, что факты могут иметь необратимые последствия в базе знаний.

Модуль знаний

Модуль знаний описывает множественные структуры некоторого структурного или функционального блока, которые представляются в

виде и-или-дерева и разбиты на отдельные подмодули [5]. Данные подмодули связаны формой представление блока с учетом физической семантики предметной области САПР:

- внешняя граница;
- внутренняя граница как состав подблоков;
- внутренняя среда как состав связей подблоков.

Отдельный модуль знаний задает локальную базу знаний о множестве структур (или функций), принадлежащих к некоторому типу. Отдельное решение носит наименование прототипа. Модуль знаний строится как и-или-дерево с определенными отношениями зависимости (продукциями) над ИЛИ-синтермами, задающими контекстную зависимость между отдельными альтернативами. Цель вывода в базе знаний на уровне отдельного модуля знаний - обеспечение выбора требуемого прототипа (решения) по техническому заданию (ТЗ) как подмножеству значений ИЛИ-синтермов. При этом и-или-дерево есть средство компактной записи множества известных прототипов, механизм порождения гипотез о возможных новых прототипах и как форма обобщения прототипов имеет место на любом уровне представления моделей. и-или-дерево задает множество синтаксически правильных выражений семиотической модели, а прототипы или аксиомы семиотической модели есть основа построения и-или-дерева.

Отношения (продукции) между «ИЛИ» связывают те термы, комбинация которых принадлежит некоторому непустому множеству семантически верных (проверенных) прототипов семиотической модели, имеющих место в и-или-дереве. Продукции, определенные над и-или-деревом, задают правила вывода семиотической модели, которые в совокупности позволяют вычленивать из и-или-дерева семантически верное подмножество семиотической модели, т.е. аксиомы-прототипы.

Прототипы, как модели объектов, имеют следующую иерархию описаний [9]:

1) Внешнее описание прототипа - ТЗ; обеспечивает интерфейс с малоквалифицированным пользователем в процессе выбора необходимого прототипа из множества всех прототипов;

2) Описание на языке внутреннего представления; позволяет описывать модель структур и функций [6]: блоки, массивы блоков, свойства блоков, значения свойств, массивы свойств, связи и т.д.; обеспечивает интерфейс с высококвалифицированным пользователем в процессе ввода (редактирования) моделей;

3) Табличная форма записи и хранения модели; назначение табличной формы записи - внутренне представление прототипов в форме, позволяющей обеспечить оптимальную форму для хранения и преобразования прототипов. Интерфейс с пользователем по табличной форме записи не предусматривается.

Текстовый характер описаний моделей в МЭО позволяет использовать в качестве формы представления и-или-деревьев формальные грамматики, например, в форме нотации Бекуса-Наура (БНФ). Аппарат теоретико-множественных операций (ТМО) над грамматиками может обеспечить:

а) процесс формирования модуля знаний (путем обобщения описаний ряда решений относящихся к общему типу;

б) процесс вывода необходимых решений (путем задания технического задания по ИЛИ-синтермам с использованием продукционных зависимостей структур различных составляющих грамматик).

Процесс создания базы знаний в режиме «умного» эксперта

Специфика извлечения знаний в режиме «умного» эксперта, где эксперт предметной области выступает инженером по знаниям, состоит из следующих понятий:

– эксперт предметной области способен строить единичный модуль знаний;

– модуль знаний представляет собой набор правил, в которых передается опыт разработчика области проектирования;

– представление знаний, то есть набор правил и порядок их следования, выбирается экспертом предметной области;

– база знаний строится из набора введенных модулей знаний.

Подходы в создании базы знаний в режиме «умного» эксперта:

– и-или дерево формируется на первом этапе при подаче в первого модуля знаний в интеллектуальную оболочку, и его дальнейшее изменение, отсечение узлов, при каждом вопросе к пользователю

– поэтапное добавление в и-или дерево нового элемента, путем диалога с пользователем, построение заканчивается при исчерпании всех вопросов к эксперту;

– можно представить еще смешанный подход, в котором происходит и урезание дерева решений, так и добавление в него недостающих элементов.

Недостаток данных подходов – отсутствие идентификации различных структур уровней иерархии. Т.е. пользователь в спрашивающих продукциях оперирует только структурными особенностями. Список прототипов используется не явно как комбинация тех или иных структурных особенностей. Это означает, что пользователь не в состоянии обратиться на каком-то этапе вывода к списку оставшихся прототипов и выбрать среди них нужные ему, минуя последующие этапы выбора структурных особенностей.

Особенности метода организации базы знаний:

– построение базы знаний, как совокупности модулей знаний, описывающих структурную модель объекта проектирования;

– добавление в базу знаний функциональных особенностей модели проектирования;

– использование жесткого порядка ввода продукций разного типа для обеспечения монотонности вывода в базе знаний;

– возможность использовать в продукциях «топологические» свойства, т.е. компоненты модели пространства.

Выводы

В статье был проведен анализ существующих САПР предметной области, таких как САПР, AutoCAD Plant 3D. В них были выявлены достоинства проектирования на структурном уровне, такие как интерфейс отображения структурной модели, большой расчетно-графический модуль, пространственное проектирование любого количества объектов. Из недостатков систем можно назвать сложный интерфейс для пользователя, решение узких задач и отсутствие проектирования объектов на функциональном уровне. Были описаны метод построения структуры и-или дерева. Были приведены виды продукций по механизму их срабатывания. Рассмотрен метод представления модуля знаний. Описан процесс создания базы знаний в режиме «умного» эксперта.

Перспективы дальнейшего развития работы является создание нового подхода к проектированию на структурно-функциональном уровне модели. В котором, и-или дерево будет учитывать иерархическую структуру компонентов предметной области, а так же хранить функциональные зависимости объекта между структурными элементами.

Литература

1. Морозова О.В. Анализ методов построения экспертных систем в продукционных инструментальных оболочках / О.В. Морозова, А.В. Григорьев // Наукові праці ДонНТУ. – Донецьк, 2012. – Вип. 16(204). – С. 180-191.

2. Григорьев А.В. Анализ существующих способов создания интерфейса «языки формальных

спецификаций — проблемно-ориентированные языки» / Григорьев А.В., Морозова О.В. // Сборник научных трудов донецкого национального технического университета. — Донецк, 2011. — № 14. — С. 270 – 275.

3. Голдовский, П. Knowledgeware: как задействовать самые ценные корпоративные активы [Электронный ресурс] / Павел Голдовский, Андрей Бубнов // САПР и графика. - Электрон. журн. - №4'2001. - Режим доступа к журн.: <http://www.sapr.ru/article.aspx?id=7125&iid=291>

4. Работа с рабочими пространствами AutoCAD Plant 3D [Электронный ресурс]. - Электрон. дан. — Режим доступа: <http://docs.autodesk.com/PLNT3D/2012/RUS/filesPUG/GUID-4A4F57DD-F2CC-474E-ADB9-45A9A3089F1-2.htm>

5. Одрин, В.М. Морфологический анализ систем. Построение морфологических таблиц / В.М. Одрин, С.С. Картавов. — К.: Наукова думка, 1977. - 148с.

6. Одрин, В.М. Метод морфологического анализа технических систем / В.М. Одрин. - М.: ВНИИПИ, 1989. — 312 с.

7. Григорьев А.В. Обеспечение монотонности вывода и верификация баз знаний в инструментальной оболочке для создания интеллектуальных надстроек над САПР / А.В. Григорьев, О.В. Малявка // Наукові праці ДонНТУ. — 2010. — Вип. 11(164). — С. 161-164.

8. Толковый словарь по искусственному интеллекту / Аверкин А.Н., Гаазе-Рапопорт М.Г., Поспелов Д.А. — М.: Радио и связь, 1992.

9. Григорьев, А.В. Создание интеллектуальной оболочки по методикам проектирования локальных сетей при использовании проектировщика как эксперта по знаниям / А.В. Григорьев, О.В. Морозова // Наукові праці ДонНТУ, Серія: «Проблеми моделювання та автоматизації проектування». Вип. №1(10)-2(11). — Донецьк, 2012. — С. 144-155.

10. Григорьев А.В. Построение двухсторонних трансляторов в задаче создания интеллектуальных надстроек над проблемно-ориентированными САПР / Григорьев А.В., Морозова О.В. // Сборник трудов XI международной научной конференции им. Т.А. Таран. - Киев: Просвита, 2011. - С. 68-75.

Morozova OV Analysis of possibilities and disadvantages of technology for building knowledge bases in modern CAD systems. CAD tools are considered standard general purpose. Also considered typical intelligent systems to create objects in a particular subject area. Identified the advantages and disadvantages of these membranes in terms of a mechanism creating a knowledge base and the inference engine. The structure of and-or tree. The classification of types of products on the mechanism of their operation. The charge of the formation of the characteristic unit of knowledge. A diagram showing the relationship of knowledge modules.

Морозова О.В. Аналіз можливостей і недоліків технології побудови баз знань в сучасних САПР. Розглянуто стандартні інструментальні САПР загального призначення. Так само розглянуті типові інтелектуальні системи для створення об'єктів в конкретній предметній області. Визначено переваги та недоліки цих оболонок з точки зору наявності механізму створення бази знань і механізмів логічного висновку. Розглянуто структуру та-або дерева. Представлена класифікація видів продукцій за механізмом їх спрацьовування. Приставлена характеристика формування модуля знань. Показана схема взаємозв'язку модулів знань.

*Статья поступила в редакцию 20.05.2016
Рекомендована к публикации к-том техн. наук А.В.Григорьевым*

УДК 622.734.001.57

Автоматизация проектирования технологических схем сушки обогащённого угля

В.Н. Павлыш, И.В. Тарабаева, Г.И. Турчанин
Донецкий национальный технический университет
pavlyshvn@mail.ru

Павлыш В.Н., Тарабаева И.В., Турчанин Г.И. Автоматизация проектирования технологических схем сушки обогащённого угля. Рассматривается задача разработки структуры и принципов функционирования системы автоматизированного проектирования технологических схем сушки обогащённого угля. Показано, что наиболее эффективным методом сушки является способ «кипящего слоя». В основу математического обеспечения системы положена детерминированная математическая модель процесса сушки в виде системы уравнений математической физики.

Введение

В ряде отраслей промышленности для обеспечения требуемого качества конечных продуктов, получаемых в виде влажных сыпучих масс, применяются различные способы и технологические схемы обезвоживания влагосодержащих материалов. Применение способов и схем обезвоживания является завершающей стадией производственного цикла и определяется на стадии проектирования технологии производства. Важнейшее значение этот процесс играет на завершающей стадии обогащения углей [1].

Однако эффективность процесса не всегда в полной мере отвечает поставленным требованиям. Одной из причин этого является то, что на этапе проектирования предварительная оценка схем весьма затруднена ввиду сложности процесса. Применение математического моделирования позволяет улучшить качество принимаемых проектных решений. В этой связи тема работы является актуальной.

Постановка проблемы.

Процесс проектирования включает ряд этапов, в том числе рассмотрение и оценка вариантов проектов, обоснование и верификация принимаемых проектных решений. По мере усложнения технологий, расширения числа возможных вариантов проектов возникает необходимость автоматизации процесса проектирования с применением методов математического моделирования и ЭВМ [2]. С этой целью разрабатывается подсистема автоматизированного проектирования технологических схем, призванная повысить

качество проектов за счет эффективного применения компьютерных технологий для реализации трудоемких рутинных функций.

Анализ выполненных исследований по теме работы.

На основании исследований и разработок, выполненных учеными в данной области, созданы и функционируют на различных предприятиях установки, осуществляющие процесс обезвоживания сыпучих материалов. Вместе с тем, учитывая постоянно изменяющиеся условия эксплуатации, связанные как с дефицитом энергоресурсов, так и с ужесточающимися требованиями к экологической составляющей деятельности промышленных предприятий, продолжаются исследования процессов и технологических схем с целью их дальнейшего совершенствования.

Результатами многочисленных теоретических и экспериментальных исследований обосновано, что наиболее прогрессивным способом является сушка в кипящем слое, подчеркивается его экономичность, более низкая по сравнению с другими способами экологическая вредность, а также высокая степень «моделируемости» [3].

Используя методику, разработанную проф. В.Н. Ткаченко, проведено исследование характеристик плотности распределения частиц по времени пребывания в псевдоожиженном слое методом вычислительного эксперимента [4].

В таблице 1 приведены результаты одного из вычислительных экспериментов. Здесь над разделительной чертой указана величина математического ожидания, под разделительной чертой – среднее квадратическое отклонение случайной величины.

Таблица 1. Вероятностные характеристики температур и степени обезвоживания частиц при различных значениях коэффициента диффузии и законах распределения времени пребывания.

Коэффициент диффузии m^2 / c	С.к.о. времени пребывания (мин)	Температура поверхности частицы (°C)	Среднемассовая температура (°C)	Степень обезвоживания частицы (%)
0,0	0	94,5	62,5	82
0,0001	8,3	95,4 ----- 4,1	72 ----- 5,3	81 ----- 4,4
0,0003	10,8	95,2 ----- 5,9	71 ----- 7,8	80 ----- 5,7
Аппроксимация нормальным законом распределения				
0,0003	8,6	95,2 ----- 7,3	71 ----- 7,5	82 ----- 7,6
0,0005	12,6	95,2 ----- 8,3	91 ----- 7,9	80 ----- 7,5
0,0010	17,5	95 ----- 8,2	90 ----- 10,4	79 ----- 9,9
Режим идеального смещения (экспоненциальный закон)				
∞	40,0	98 ----- 18,2	89 ----- 16,1	68 ----- 13,8

В результате исследования установлено, что математическое ожидание времени пребывания частицы в активной зоне составляет более 11 мин, чем подтверждается, что сушка в «кипящем слое» обеспечивает более длительный контакт материала с сушильным агентом по сравнению с другими схемами, и по этой причине является наиболее эффективной технологией обезвоживания.

Цель работы.

Обоснование структуры и алгоритмов функционирования системы автоматизации проектирования технологических схем сушки влажной сыпучей массы.

Задачи работы:

- разработка общей структуры технологических схем обезвоживания;
- разработка функционально-логической схемы технологии обезвоживания;
- разработка математического обеспечения САПР.

Основное содержание работы.

Классификация основных видов технологии сушки приведена в табл. 2 и на рис. 1.

На рис. 2 приведена функционально-логическая схема системы автоматизированного проектирования (САПр) технологии процесса сушки.

Как показывает практика организации работ по созданию сушильных агрегатов, наиболее перспективным направлением в технологии обезвоживания является сушка в «кипящем слое» [1, 3].

Сушилка «кипящего слоя» позволяет увеличить продолжительность сушки материала во много раз и регулировать ее в очень широких пределах [4]. Этот принцип сушки наиболее приемлем для таких материалов, продолжительность сушки которых исчисляется десятками секунд или минутами. Таким образом, сушка материалов в «кипящем слое» позволяет значительно расширить область применения метода сушки во взвешенном состоянии.

Упрощенная схема аппарата, выполняющего сушку в «кипящем слое», приведена на рис. 3, картина потоков в «псевдооживленном слое» приводится на рис. 4.

В камеру сушилки, снабженной газораспределительной решеткой, помещается

сыпучий материал.

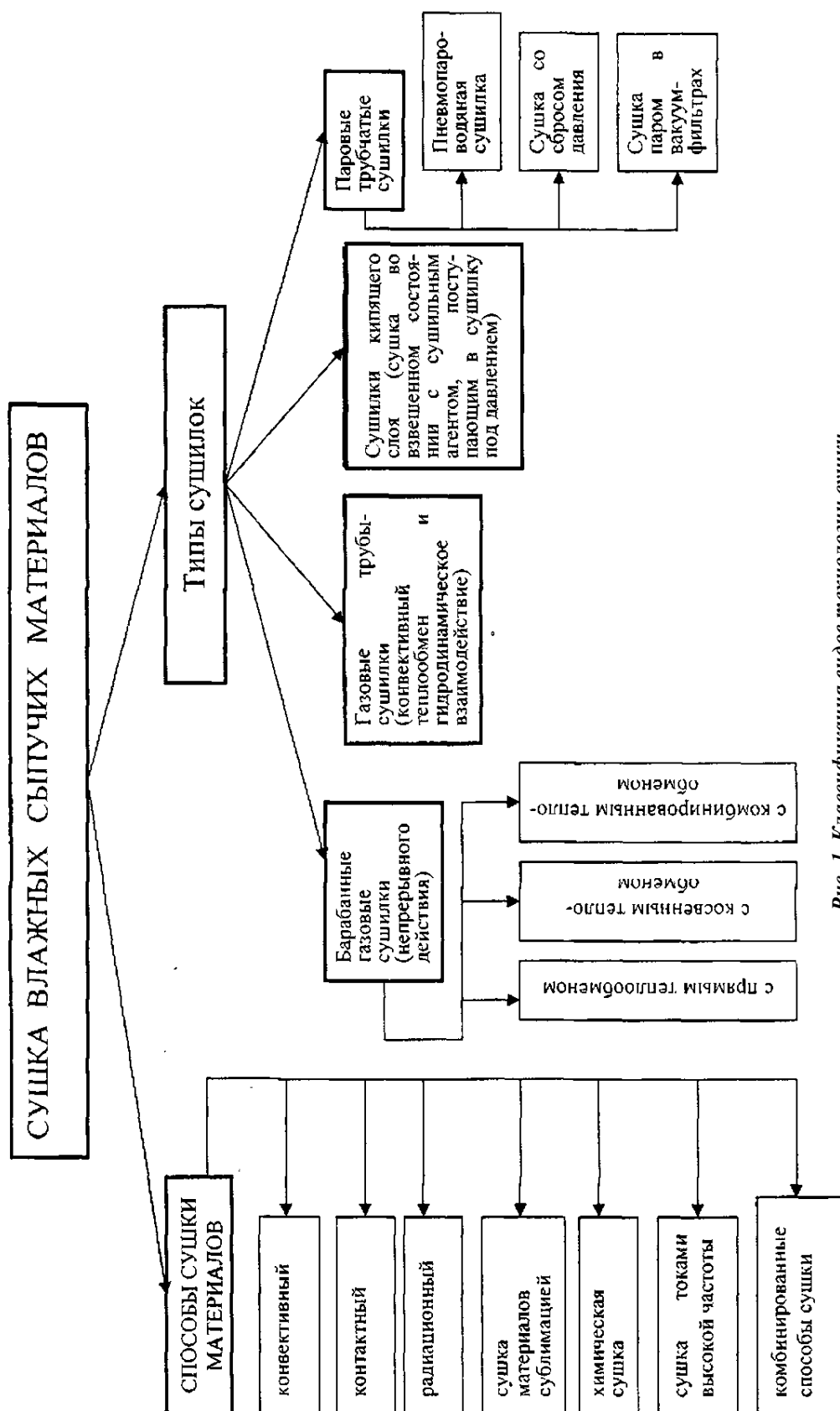


Рис. 1. Классификация видов технологий сушки

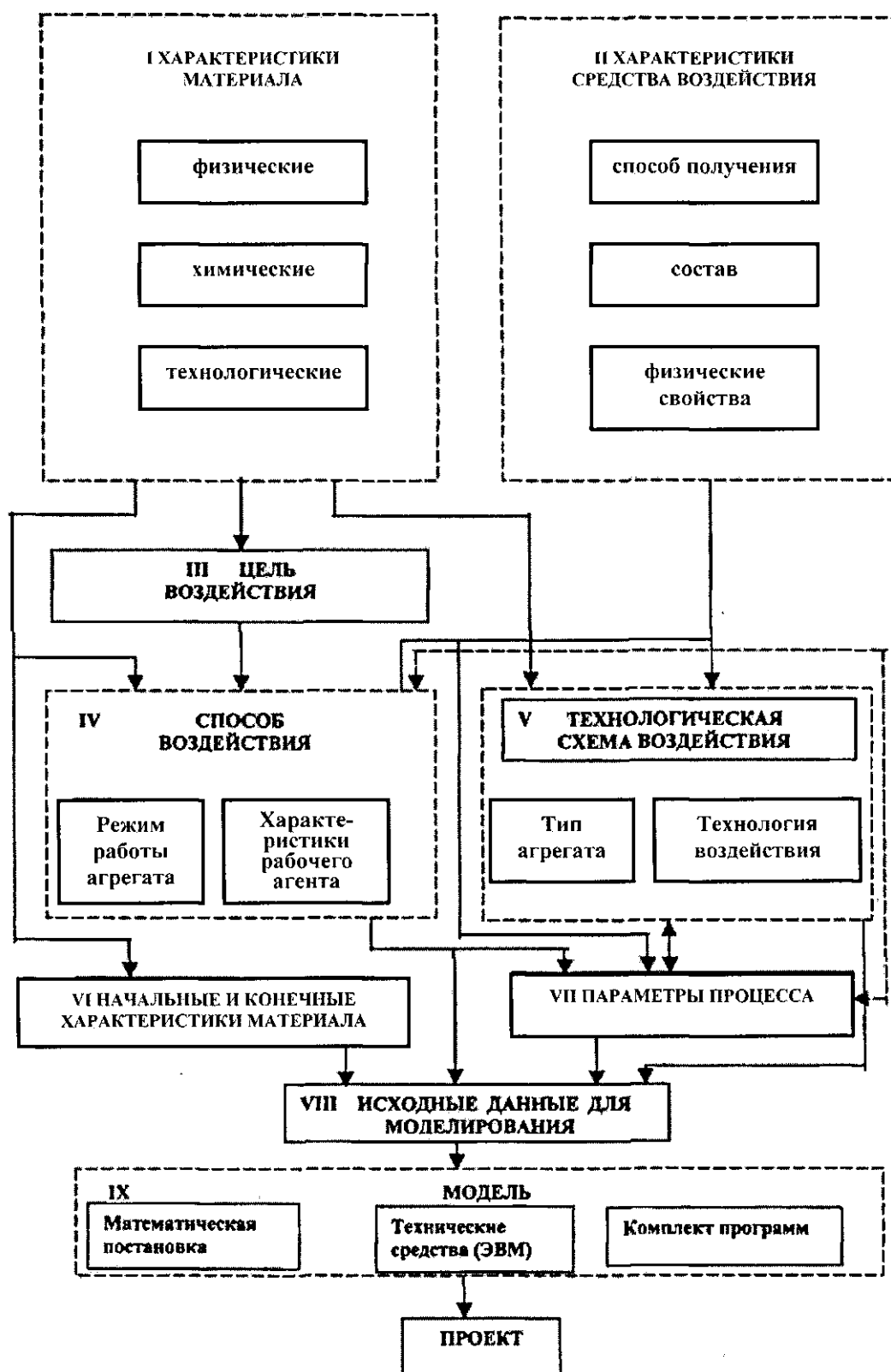


Рис. 2. Функционально-логическая схема системы автоматизированного проектирования технологических схем сушки

Таблица 2. Классификация сушильных установок

Признак классификации	Типы сушилок
По способу подвода тепла к материалу	Конвективные, контактные (сушка на горячих поверхностях), радиационные (сушка инфракрасными лучами), электрические (сушка в электрическом поле)
По давлению в рабочем пространстве	Атмосферные вакуумные
По способу действия	Периодического или непрерывного действия
Сушильный агент	Воздух, топочные газы, смесь воздуха с топочными газами, перегретый пар и инертные газы
По направлению движения сушильного агента относительно материала	С прямотоком, противотоком, перекрестным током и реверсивные
По характеру циркуляции сушильного агента	С естественной и принудительной циркуляцией
По способу нагрева сушильного агента	С паровыми воздухоподогревателями, огневыми воздухоподогревателями, путем смешения с топочными газами, с электронгревом
Схема нагрева сушильного агента	С централизованным подогревом, с подогревом индивидуальными агрегатами и с промежуточным подогревом

Для создания режима локального фонтанирования через газораспределительную решетку в псевдооживленный слой вводится оживающий агент с высокой скоростью. Благодаря этому в слое образуются зоны, в которых частица и среда движутся с более высоким, чем в слое, скоростями, а обмен между этими зонами делает более интенсивными процессы тепло- и массообмена.

Гидродинамическая структура потоков, возникающих при локальном вводе оживающего агента в псевдооживленный слой, указывает на наличие четырех зон перемешивания (рис. 2).

Зона I - фонтан из частиц, движущихся вверх.

Зона II - прирешеточная активная зона.

Равнодействующая сил на частицы в этой зоне направлена в сторону фонтана вследствие интенсивного перемешивания и втягивающей силы фонтана.

Зона III - зона слоя, прилегающего к фонтану и активно питающая фонтан.

Зона IV - наименее активная зона с преимущественным движением вниз за счет обмена с активной прирешеточной зоной II и зоной III.

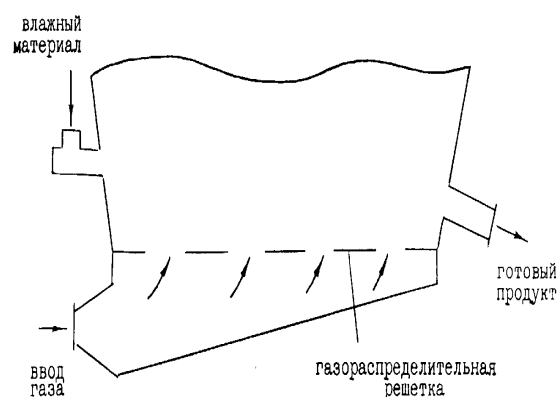


Рис. 3 Схема сушильного аппарата

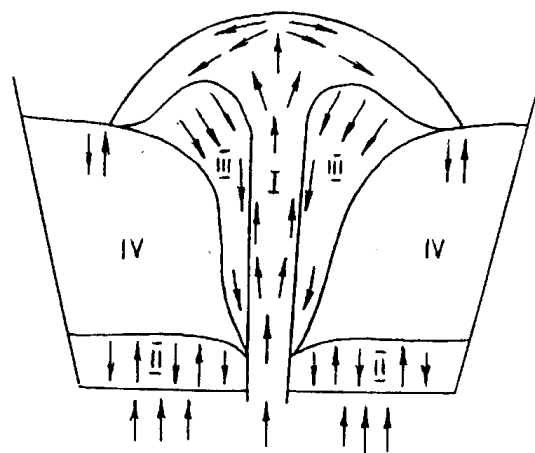


Рис. 4 Схема структуры потоков при наложении на псевдооживающий слой режима фонтанирования

При исследовании процесса выделим три основных величины, изменение которых будет исследоваться: температура, концентрация и скорость влажного сыпучего материала. Имеющийся математический аппарат уравнений в частных производных позволяет моделировать

распределение этих величин по сечению камеры сушиллки, система полученных уравнений составляет основу математического обеспечения разрабатываемой системы проектирования.

Математическая модель распределения температуры строится следующим образом.

В сушильном аппарате требуется определить распределение температуры при заданном количестве узлов (отверстий) на газораспределительной решетке.

Такие задачи относятся к классу краевых задач со смешанными краевыми условиями, а само уравнение есть уравнение эллиптического типа.

Дифференциальное уравнение распределения температуры и краевые условия имеют следующий вид:

$$\omega \cdot c \cdot \rho \frac{\partial u}{\partial x} = k \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \alpha \cdot (u - T_s), \quad (1)$$

Краевые условия (Γ_K – непроницаемая граница, D_K – отверстия в решетке):

$$\begin{aligned} &0 \leq x \leq l \quad \left. \frac{\partial u(x, y)}{\partial y} \right|_{y=h} = 0; \\ \text{при} & \\ &0 \leq y \leq h \quad \left. \frac{\partial u(x, y)}{\partial y} \right|_{x=0} = 0; \quad \left. \frac{\partial u(x, y)}{\partial y} \right|_{x=l} = 0; \\ \text{при} & \\ &(x, y) \in \Gamma_K \quad \frac{\partial u(x, y)}{\partial y} = 0; \\ \text{при} & \\ &(x, y) \in D_K \quad u(x, y) = U, \end{aligned}$$

где U – температура поступающего газа; c – удельная теплоемкость сыпучей среды; ρ – плотность среды; ω – скорость поступающего газа; k – коэффициент теплопроводности среды; α – коэффициент теплообмена; T_s – температура внешнего слоя.

Математическая модель распределения скорости строится на основании следующих соображений. Пусть $u(x, y)$ – продольная, а $v(x, y)$ – поперечная скорость вещества в сушилке. Эти компоненты скорости определяются как решение следующей краевой задачи:

$$\begin{cases} u \cdot \frac{\partial u}{\partial x} + v \cdot \frac{\partial u}{\partial y} = -\frac{dP}{dx} + \frac{\partial^2 u}{\partial y^2}, & (x, y) \in G \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \end{cases} \quad (2)$$

при соответствующих граничных условиях.

Математическая модель поля концентрации в камере сушиллки основывается на следующих положениях. Функция $C(x, y, t)$ является решением конвективного уравнения диффузии

$$\frac{\partial C}{\partial t} = D(C_{xx} + C_{yy}) - v_1(x, y) \frac{\partial C}{\partial x} - v_2(x, y) \frac{\partial C}{\partial y}, \quad (3)$$

где D – коэффициент диффузии;

$v_1(x, y)$, $v_2(x, y)$ – продольная и поперечная компоненты скорости, которые определяются из решения краевой задачи (2).

Таким образом, разработанные детерминированные математические модели основных процессов, происходящих при сушке влажного сыпучего материала с использованием способа «кипящего слоя», адекватно представляют процесс как объект с распределенными параметрами и дают возможность предварительного теоретического исследования и обоснования рациональных технологических параметров при проектировании сушильного оборудования.

Функционирование системы обеспечивается головной программой.

На рис. 5 приведена обобщенная блок-схема алгоритма функционирования головной программы.

Блок 1 содержит информацию о составе и свойствах материала, поступающего на сушку. Информация представляется в виде массива или файла с как можно более полным набором характеристик.

В этом же блоке представляется информация о сушильном агенте, за счет воздействия которого создается эффект «псевдооживления».

В блоке 2 помещается информация о тех характеристиках конечного продукта, которые должны быть обеспечены в результате воздействия. Основные из них – это влажность на выходе и производительность аппарата, однако в процессе исследований могут быть рассмотрены и другие показатели.

Блок 3 – основной расчетный блок. Здесь выполняется численная реализация всех математических моделей, включаемых в предметную область САПр. На данном этапе это математические модели распределения температуры материала в камере сушиллки, скорости частиц и концентрации обрабатываемого материала.

Как результат работы блока получают значения указанных параметров и вычисляются все другие величины, характеризующие показатели работы установки.

В блоке 4 выполняется верификация расчетных значений параметров по заданным значениям показателей, определяющих требования к конечному продукту. Верификация может быть построена как по принципу «консенсуса», когда отклонение хотя бы одного параметра вызывает необходимость коррекции исходных данных и повторения моделирования, так и по интегральному принципу с

использованием, например, метода наименьших квадратов.

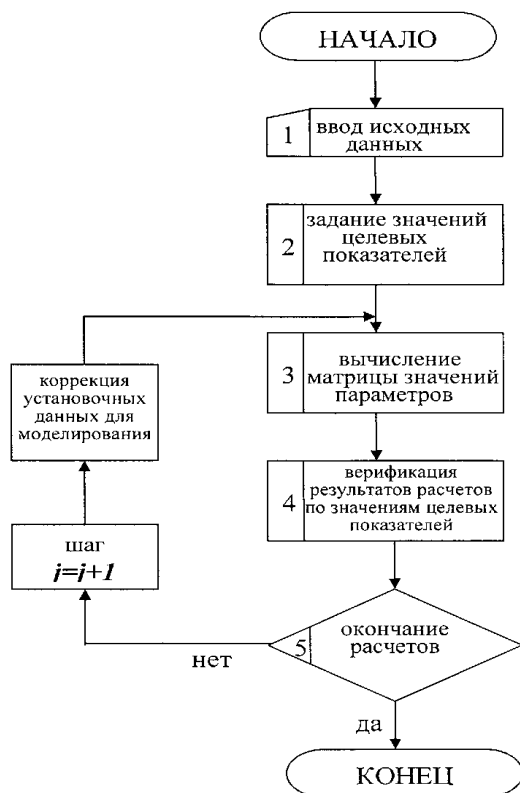


Рис. 5 Блок-схема алгоритма головной программы

Выводы

В результате выполненных исследований разработана структура системы автоматизированного проектирования технологии сушки и алгоритмы

функционирования её подсистем. Обоснован состав математического обеспечения системы, основу которого составляют уравнения математической физики, образующие детерминированную модель процессов. Основным преимуществом системы автоматизированного проектирования является то, что с ее помощью удастся получить достаточно полную информацию о различных вариантах проекта, при этом избежать массы рутинных операций и необходимости проведения значительного количества дорогостоящих промышленных экспериментов.

Литература.

1. Филиппов В.А. Конструкция, расчеты и эксплуатация устройств и оборудования для сушки минерального сырья / В.А. Филиппов – [2-е изд., перераб. и доп.] – М.: «Недра», 1989. – 309 с.: илл., табл.
2. Павлыш В.Н. Развитие теории и совершенствование технологии процессов воздействия на угольные пласты: Монография. - Донецк: РВА ДонНТУ, 2005. - 347с.
3. Ткаченко В.Н. Математическое моделирование, идентификация и управление технологическими процессами тепловой обработки материалов: Монография / В.Н. Ткаченко – Киев: «Наукова думка», 2008. – 243 с.
4. Tkachenko V.N. Research of the probabilistic characteristics of a response time and temperatures of particles in fluidized bed on the basis of computing experiment // Book of abstracts of 9-th conference of the European consortium for mathematics in industry. Copenhagen: Technical university of Denmark, 1996, p. 642 – 643.

Павлыш В.Н., Тарабаева И.В., Турчанин Г.И. Автоматизация проектирования технологических схем сушки обогащённого угля. Рассматривается задача разработки структуры и принципов функционирования системы автоматизированного проектирования технологических схем сушки обогащенного угля. Показано, что наиболее эффективным методом сушки является способ «кипящего слоя». В основу математического обеспечения системы положена детерминированная математическая модель процесса сушки в виде системы уравнений математической физики. Основным преимуществом системы автоматизированного проектирования является то, что с ее помощью удастся получить достаточно полную информацию о различных вариантах проекта, при этом избежать массы рутинных операций и необходимости проведения значительного количества дорогостоящих промышленных экспериментов.

Ключевые слова: процесс, технология, математическая модель, структура, алгоритм.

Pavlysh V.N., Tarabayeva S.V., Turchanin G.I. The automatic projecting of enriched coal drying technological schemes. *The problem of creation of structure and principles of activity of system of automatic projecting of technological schemes of drying of enriched coal is considered. Provided most effective method of drying "boiling layer".. The base of mathematical soft of system is determined mathematical model of process of drying as the system of equations of mathematical physics. The main preference of automatic projecting system is the ability of obtaining enough full information about various variants of project without grate number of routine operations and many industrial experiments*

Key words: process, technology, mathematical model, structure, algorithm.

Статья поступила в редакцию 20.05.2016

Рекомендована к публикации д-ром техн. наук А.С. Миненко

УДК 004.9

Исследование алгоритмов рекомендательных систем

Е.Д. Чепикова, Е.О. Савкова, М.В. Привалов
Донецкий национальный технический университет

Чепикова Е.Д., Савкова Е.О., Привалов М.В. Исследование алгоритмов рекомендательных систем. Выполнен анализ типов рекомендательных систем и алгоритмов использующихся для систем с коллаборативной фильтрацией. Разработана алгоритмическая часть гибридной рекомендательной системы на основе коллаборативной фильтрации. Предложено решение проблемы «холодного старта» для только что зарегистрировавшихся пользователей и для новых фильмов, добавленных в систему.

Постановка проблемы

Сейчас пришло время развития рекомендательных систем (РС). Пользователи дорожат своим свободным временем и хотят потратить его с пользой. Этому способствуют системы рекомендаций, где система сама выбирает и предлагает пользователю музыку, фильмы, книги и прочее. Такие же системы используют магазины, предлагающие товары покупателям. Подбор товаров позволяет найти для пользователя подходящую продукцию и не потерять покупателя.

Экономия времени и стремление к удобству рождает потребность в таких системах. Для разработки математического обеспечения такой системы нужно выполнить следующие этапы:

- выбрать тип разрабатываемой РС;
- сравнить алгоритмы и выбрать наиболее быстрый, дающий более точные результаты (модификацию алгоритма);
- решить проблему подбора рекомендаций для новых пользователей и новых фильмов.

Цель работы

В данной работе целью является исследование типов РС, их алгоритмов с точки зрения скорости решения задачи и точности получаемых результатов.

Постановка задачи исследования

В данной работе необходимо определить последовательность алгоритмов для реализации рекомендательной Web-ориентированной системы, содержащей медиа-контент, а также решить проблему «холодного старта» для пользователей, только что прошедших регистрацию и новых фильмов, добавленных в систему.

Решение задач и результаты исследований

Рассмотрим основные типы РС. Традиционно их разделяют на четыре типа.

Коллаборативная фильтрация

Коллаборативная фильтрация (англ. collaborative filtering) вырабатывает рекомендации, основанные на модели предшествующего поведения пользователя. Эта модель может быть построена исключительно на основе поведения данного пользователя или — что более эффективно — с учетом поведения других пользователей со сходными характеристиками [1].

Плюсы: Теоретически высокая точность. Минусы: Высокий порог входа: не зная ничего об интересах пользователя, рекомендации практически бесполезны, многие пользователи будут просто сразу уходить [2].

РС, основанные на контенте (content-based)

В контентных рекомендательных системах вывод о полезности $u(h,s)$ товара s для потребителя h делается, исходя из полезности (h,s_i) , присвоенной потребителем товарам, сходным с товаром s . Например, в системе, рекомендующей фильмы, для того, чтобы рекомендовать фильмы потребителю h , контентная РС пытается найти сходство между фильмами, высоко оцененными потребителем ранее (общие актеры, режиссеры, жанры и т.д.) И только фильмы, обладающие высокой степенью общности с предпочтениями потребителя, будут рекомендованы [3].

Плюсы: Можно делать рекомендации даже незнакомым пользователям, тем самым вовлекая их в сервис. Возможность рекомендовать те объекты, которые еще не были никем оценены. Минусы: Точность сильно падает, время разработки немного возрастает.

РС, основанные на знаниях (knowledge-based)

Рекомендации, основанные на знаниях о предметной области. Часто предыдущий тип (content-based) определяют как частный случай knowledge-based, где знаниями являются сведения о товаре, но content-based имеет такую широкую распространенность, что имеет смысл выносить его в отдельный тип. Эти самые дополнительные знания позволяют делать рекомендации не основываясь на «похожести» чего-либо, а с более сложными условиями [2]. Плюсы: Возможность исключить ситуацию рекомендации товаров уже неактуальных для данного пользователя. Минусы: Высокая сложность разработки и сбора данных.

Гибридные (hybrid) РС

Комбинируют различные подходы, что позволяет избежать ограничений, свойственных каждой системе.

Проанализировав плюсы и минусы можно сделать выбор в пользу гибридных систем. Для нашей задачи будем использовать метод коллаборативной фильтрации, и предложим своё решение проблемы «холодного старта» для пользователей и новых фильмов.

Рассмотрим основные алгоритмы, которые используются для реализации этого метода.

Алгоритм основанный на коэффициенте корреляции.

В [4] предложено следующее решение задачи коллаборативной фильтрации.

Имеется вектор предпочтений для каждого пользователя (строки матрицы R) и вектор оценок пользователей для каждого продукта (столбцы матрицы R). Прежде всего оставим в этих векторах только те элементы, для которых нам известны значения в обоих векторах, т.е. оставим только те продукты, которые оценили оба пользователя, или только тех пользователей, которые оба оценили данный продукт. В результате нам просто нужно определить, насколько похожи два вектора вещественных чисел. Для этого необходимо подсчитать коэффициент корреляции

$$w_{i,j} = \frac{\sum_a (r_{i,a} - \bar{r}_i)(r_{j,a} - \bar{r}_j)}{\sqrt{\sum_a (r_{i,a} - \bar{r}_i)^2} \sqrt{\sum_a (r_{j,a} - \bar{r}_j)^2}}, \quad (1)$$

где $\bar{r}_i$ — средний рейтинг, выставленный

пользователем i. Иногда пользуются так называемой «косинусной похожестью», используя косинус угла между векторами

$$w_{i,j} = \frac{\sum_a r_{i,a} r_{j,a}}{\sqrt{\sum_a r_{i,a}^2} \sqrt{\sum_a r_{j,a}^2}}. \quad (2)$$

Но для того, чтобы косинус хорошо работал, желательно всё равно сначала вычесть среднее по каждому вектору, так что в реальности это та же самая метрика.

Для того чтобы воспользоваться этими коэффициентами схожести необходимо приблизить новый рейтинг как средний рейтинг данного пользователя плюс отклонения от среднего рейтингов других пользователей, взвешенных этими самыми весами

$$\hat{r}_{i,a} = \bar{r}_i + \frac{\sum_j (r_{j,a} - \bar{r}_j) w_{i,j}}{\sum_j |w_{i,j}|}. \quad (3)$$

Для ускорения поиска ближайших соседей автор работы [4] предлагает воспользоваться k-d-деревьями или локально-чувствительным хешированием (locally sensitive hashing).

Плюсом данного алгоритма является достаточно простая реализация. Однако этот алгоритм является устаревшим, на его основе работали первые РС, сейчас же существуют другие алгоритмы, которые быстрее справляются со своей задачей и дают более точные результаты.

Алгоритм наивных сетей Байеса

Теорема Байеса — одна из основных теорем элементарной теории вероятностей, которая определяет вероятность наступления события в условиях, когда на основе наблюдений известна лишь некоторая частичная информация о событиях.

Согласно [5] условная вероятность $P(A|H)$ события A при условии наступления события H вычисляются по формуле

$$P(A|H) = \frac{P(A,H)}{P(H)}, \quad (4)$$

где P(A) и P(H) — это вероятности каждого события по отдельности, а P(A,H) является совместной вероятностью A и H. Значит, совместную вероятность можно выразить двумя способами (5):

$$P(A,H) = P(A|H) * P(H) = P(H|A) * P(A) \quad (5)$$

Тогда теорема Байеса имеет вид

$$P(A|H) = \frac{P(H|A)P(A)}{P(H)} \quad (6)$$

Перепишем теорему Байеса в других обозначениях

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)} \quad (7)$$

$p(\theta|D)$ — это то, что мы хотим найти,

распределение вероятностей параметров модели

после того, как мы приняли во внимание данные; это называется апостериорной вероятностью. В источнике [6] говорится, что эту вероятность, как правило, напрямую не найти, и здесь как раз и нужна теорема Байеса. $p(D|\theta)$ — это так

называемое правдоподобие, вероятность данных при условии зафиксированных параметров модели; это как раз найти обычно легко, собственно, конструкция модели обычно в том и состоит, чтобы задать функцию правдоподобия. А $p(\theta)$ —

априорная вероятность, она является математической формализацией нашей интуиции о предмете, формализацией того, что мы знали раньше, ещё до всяких экспериментов.

Алгоритм SVD

SVD (Singular Value Decomposition), переводится как сингулярное разложение матрицы. В теореме о сингулярном разложении утверждается, что у любой матрицы A размера $n \times m$ существует разложение в произведение трех матриц: U, D и V^T

$$A = U * D * V^T \quad (8)$$

Матрицы U и V ортогональные, а D — диагональная (хотя и не квадратная). Алгоритм достаточно простой, но позволяет не только предсказывать оценки. С его помощью мы можем по истории пользователей выявлять скрытые признаки объектов и интересы пользователей.

Итак, чтобы предсказать оценку пользователя U для фильма I , мы берем некоторый вектор p_u (набор параметров) для данного пользователя и вектор для данного фильма q_i . Их скалярное произведение и будет нужным нам предсказанием

$$\hat{r}_{ui} = \langle p_u, q_i \rangle \quad (9)$$

Однако мы не можем найти SVD-разложение матрицы, т.к. мы не знаем саму матрицу. Но мы хотим воспользоваться этой идеей и придумать модель предсказания, которая будет работать сходным с SVD образом. Хорошим образом такой модели является модель, описанная в блоге компании Surfingbird [7].

Введем так называемые базовые предикторы $b_{i,a}$, которые складываются из базовых предикторов отдельных пользователей b_i , и отдельных продуктов b_a , а также просто общего среднего рейтинга по базе μ

$$b_{i,a} = \mu + b_i + b_a \quad (10)$$

где μ — средний рейтинг по базе; b_i — средний рейтинг каждого i пользователя; b_a — средний рейтинг каждого a продукта.

Для определения только базовых предикторов необходимо найти такие μ , b_i , b_a , для которых $b_{i,a}$ лучше всего приближают имеющиеся рейтинги. Затем можно будет добавить собственно факторы. Поскольку теперь, когда сделана поправка на базовые предикторы, остатки будут сравнимы между собой, вполне возможно будет получить разумные значения для факторов

$$\hat{r}_{i,a} = \mu + b_i + b_a + v_a^T u_i, \quad (11)$$

где v_a — вектор факторов, представляющий продукт a ; u_i — вектор факторов, представляющий пользователя i .

С учётом (10) формулировка исходной задачи принимает следующий вид: необходимо найти наилучшие предикторы, которые приближают величину $\hat{r}_{i,a}$.

Лучшими будут те предикторы, которые дают минимальную ошибку, определяемую по формуле

$$L(\mu, b_i, b_a, v_a, u_i) = \sum_{(i,a) \in D} (r_{i,a} - \hat{r}_{i,a})^2 = \sum_{(i,a) \in D} (r_{i,a} - \mu - b_i - b_a - v_a^T u_i)^2 \quad (12)$$

Функцию $L(\mu, b_i, b_a, v_a, u_i)$ минимизируем методом градиентного спуска: берем частные производные по каждому аргументу и двигаемся в сторону, обратную направлению этих частных производных. Для компенсации эффекта переобучения добавляется параметр регуляризации. Иными словами, накладывается штраф за слишком большие значения обучаемых переменных. Как показано в работе [8], для этого можно просто добавить в функцию ошибки сумму квадратов всех факторов и предикторов:

$$b_*, q_*, p_* = \arg \min = \sum_{(i,a)} (r_{i,a} - \mu - b_i - b_a - q_a^T p_i)^2 + \lambda (\sum_i b_i^2 + \sum_a b_a^2 + \|q_a\|^2 + \|p_i\|^2), \quad (13)$$

где λ — параметр регуляризации.

Если взять у функции ошибки в формуле (12) частные производные по каждой из оптимизируемых переменных, получим простые правила для градиентного (стохастического) спуска

$$\begin{aligned} b_i &= b_i + \gamma(e_{i,a} - \lambda b_i), \\ b_a &= b_a + \gamma(e_{i,a} - \lambda b_a), \\ q_{a,j} &= q_{a,j} + \gamma(e_{i,a} p_{i,j} - \lambda q_{a,j}), \\ p_{i,j} &= p_{i,j} + \gamma(e_{i,a} p_{i,j} - \lambda p_{i,j}), \end{aligned} \quad (14)$$

для всех j , где $e_{i,a} = r_{i,a} - \hat{r}_{i,a}$ — ошибка на данном тестовом примере, а γ — скорость обучения

Для компенсации эффекта переобучения добавляется параметр регуляризации. Иными словами, накладывается штраф за слишком большие значения обучаемых переменных [9].

Исходя из данных источника [7], SVD является самым быстрым алгоритмом РС. Существуют три его разновидности: SVD, SVD++, timeSVD. При этом SVD немного проигрывает в скорости двум другим. В статье описывалась реализация SVD++, его мы и предлагаем использовать в нашей системе.

Решение проблемы «холодного старта»

Проблема «холодного старта» делится на «холодный старт» для пользователей и «холодный старт» для фильмов. Рассмотрим решение обоих вариантов. Проанализировав источник [10], предлагается собственное решение проблемы.

«Холодный старт» для пользователей. Эта проблема состоит в том, что только что зарегистрировавшимся пользователям нечего порекомендовать, так как они не успели выставить оценок фильмам. Логично, что чем больше оценок пользователь поставит, тем точнее система будет выдавать результаты.

Для решения возьмём набор обычных базовых данных при регистрации – возраст пользователя (из даты его рождения), местоположение пользователя, и его гендерную принадлежность. Система должна осуществить подбор векторов пользователей, которые имеют схожее геоположение (одну и ту же страну, один и тот же регион), такой же пол и входят в те же возрастные рамки. Возрастные рамки предлагается поделить на промежутки: до 12 лет, 12-18, 18-25, 25-40, 40-55, старше 55. После этого будут отобраны фильмы которым данные пользователи поставили максимальные оценки и предложены к просмотру данному, только что вошедшему в систему пользователю. После проставления собственных оценок эти данные будут заменяться результатами алгоритма SDV++.

Проблема «холодного старта» для новых фильмов, добавленных в систему, заключается в том, что у этих фильмов пока ещё нет рейтинговых оценок, следовательно их некому рекомендовать.

Для решения этой проблемы следует провести корреляционно-регрессионный анализ для таких данных, как жанры, актёры и режиссёры. Для рекомендаций новых фильмов нужно выделить те из них, которые наиболее коррелируют по выше названным данным с тем, что предпочитает пользователь. Результаты по подобранным новинкам системы следует выводить перед результатами алгоритма SVD++.

Выводы

В результате проведенного анализа типов рекомендательных систем для применения в Web-ориентированной системе подбора фильмов, был выбран гибридный метод, дающий наиболее точные результаты. Метод включает в себя коллаборативную фильтрацию и собственную

разработку решения проблемы подбора медиа-контента для новых пользователей системы. После исследования алгоритмов коллаборативной фильтрации был выбран алгоритм SVD++, который является модификацией самого быстрого и широко используемого алгоритма SVD, разработанного для разреженных матриц рейтингов. Предложено решение для проблемы «холодного старта» новых пользователей, заключающееся в подборе фильмов пользователям на основе геоположения и возрастной категории, и новых фильмов, заключающееся в подборе наиболее коррелированных по жанру с предпочитаемыми жанрами пользователя.

Направлением дальнейших исследований в данной работе является модификация метода Singular Value Decomposition с целью повышения точности результатов, что планируется достигнуть за счёт усложнённого формирования правил базовых предикторов.

Литература

1. Тим Джонс - Рекомендательные системы: Часть 1. Введение в подходы и алгоритмы [Электронный ресурс] // Режим доступа: <http://www.ibm.com/developerworks/ru/library/os-recommender1/index.html>
2. Рекомендательные системы [Электронный ресурс] // Режим доступа: <http://vas3k.ru/blog/355/>
3. Gediminas Adomavicius, Alexander Tuzhilin – IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6, 2005
4. Блог компании Surfingbird – Рекомендательные системы: user-based и item-based [Электронный ресурс] // Режим доступа: <http://habrahabr.ru/company/surfingbird/blog/139518/>
5. Вентцель Е.С. Теория вероятностей: Учеб. для вузов. — 6-е изд. стер. — М.: Высш. шк., 1999.— 576 с.
6. Блог компании Surfingbird – Рекомендательные системы: теорема Байеса и наивный байесовский классификатор [Электронный ресурс] // Режим доступа: <http://habrahabr.ru/company/surfingbird/blog/15020/>
7. Блог компании Surfingbird – Рекомендательные системы: SVD и базовые предикторы [Электронный ресурс] // Режим доступа: <http://habrahabr.ru/company/surfingbird/blog/140555/>
8. Королева Д.Е., Филиппов М.В. – Анализ алгоритмов обучения коллаборативных рекомендательных систем, 2013
9. Рекомендательные системы: You can (not) advise [Электронный ресурс] // Режим доступа: <http://habrahabr.ru/post/176549>
10. Блог компании Surfingbird – Рекомендательная система: введение в проблему холодного старта [Электронный ресурс] // Режим доступа: <http://habrahabr.ru/company/surfingbird/blog/168733>

Чепикова Е.Д., Савкова Е.О., Привалов М.В. *Исследование алгоритмов рекомендательных систем. Выполнен анализ типов рекомендательных систем и алгоритмов использующихся для систем с коллаборативной фильтрацией. Разработана алгоритмическая часть гибридной рекомендательной системы на основе коллаборативной фильтрации. Предложено решение проблемы «холодного старта» для только что зарегистрировавшихся пользователей и для новых фильмов, добавленных в систему.*

Ключевые слова. Алгоритмы рекомендательных систем, гибридный метод, коллаборативная фильтрация, проблема «холодного старта», применение алгоритма SVD (Singular Value Decomposition) в рекомендательных системах.

Чепікова О.Д., Савкова О.Й., Привалов М.В. *Дослідження алгоритмів рекомендаційних систем. Виконано аналіз типів рекомендаційних систем і алгоритмів, що використовуються для систем з коллаборативною фільтрацією. Розроблено алгоритмічну частину гібридної рекомендаційної системи на основі коллаборативної фільтрації. Запропоновано вирішення проблеми «холодного старту» для користувачів, що щойно зареєструвалися і для нових фільмів, доданих в систему.*

Ключові слова: Алгоритми рекомендаційних систем, гібридний метод, коллаборативна фільтрація, проблема «холодного старту», застосування алгоритму SVD (Singular Value Decomposition) в рекомендаційних системах.

Elena Chepikova, Elena Savkova, Maksim Privalov, Research of recommender systems' algorithms. *Made analysis of the existing types of recommender systems and performed investigation of the algorithms used in the systems which are based on collaborative filtering approach. Developed set of the algorithms designed to be used in hybrid recommender system. Proposed solution of the "cold start problem" that could be applied to initially registered users and new films added to the system.*

Keywords: Algorithms of recommender systems, hybrid method, collaborative filtering, the problem of "cold start", using of the algorithm SVD (Singular Value Decomposition) in recommendation systems.

*Статья поступила в редакцию 20.05.2016
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом*

УДК 004.946

Инструментальные средства виртуальной реконструкции утраченных памятников архитектуры

М.П. Руденко

Донецкий национальный технический университет
m.p.rudenko@gmail.com

Руденко М.П. Инструментальные средства виртуальной реконструкции утраченных памятников архитектуры. Рассматриваются популярные графические приложения для виртуальной реконструкции утраченных памятников архитектуры, используемые в среде архитекторов и археологов. Приводится их классификация по способу построения и визуализации трехмерной модели на следующие типы редакторов: воссоздающие модель утраченного архитектурного памятника с нуля; работающие с моделями, полученными путем лазерного сканирования; и путем фотограмметрии, анализируется качество их работы, на основе анализа выбирается оптимальное графическое приложение для работы с виртуальной реконструкцией утраченных памятников архитектуры.

Введение

Виртуальная реконструкция утраченных памятников архитектуры является новым направлением изучения как истории архитектуры, искусства, археологии, так и всеобщей истории. Использование трехмерного измерения для воссоздания утраченной архитектурной среды является большим вкладом в культурное наследие общества. Не всегда удастся восстановить вживую прежний облик архитектурного здания или комплекса, а то и целого города из-за многих причин, однако трехмерная модель не требует больших затрат, кроме наличия устройства отображения виртуальной информации.

Требования к детализации в визуализации трехмерной модели архитектурного памятника становятся повышенными, поэтому и требования к графическим редакторам, осуществляемым эту визуализацию повышаются. Требования для удобства работы графического редактора с архитектурными памятниками следующие:

- корректная работа с иконографическими материалами (возможность анализа и построения существующей части архитектурного памятника на основе фотографий, рисунков, чертежей);
- корректная работа с самой трехмерной моделью памятника (простота построения модели, отсутствие или минимум артефактов и искажений);
- быстрота и качество работы (минимальные временные затраты и хорошее качество изображения при визуализации);
- большой инструментарий графического редактора (возможность нарисовать или начертить объект, либо часть объекта любой сложности).

Исходя из этих требований, можно условно разделить графические приложения для работы с

утраченными памятниками архитектуры на:

- графические редакторы, которые воссоздают модель утраченного памятника с нуля, используя библиотеки различных архитектурных элементов и карту материалов;
 - графические редакторы, работающие с облаком точек, которое получают при помощи лазерного сканирования или фотограмметрии;
 - графические редакторы, использующие векторное построение на основе фотограмметрии.
- Цель статьи - рассмотреть и проанализировать графические приложения для создания виртуальной реконструкции утраченных памятников архитектуры с целью выбора наиболее оптимального из них.

Обзор научных исследований

Занимаясь исследованием данного вопроса, можно отметить научное достижение Пола Дебевета, исследователя в области компьютерной графики, который, на основании построения основной геометрии архитектурного здания по фотографиям, создал графический редактор «Facade» (рис.1). Данный редактор строит трехмерную модель здания по прямым линиям, которыми маркируются основные грани здания на фотографии (рис.2). «Facade» был позже выкуплен компанией Adobe, которая на его основании создала программу Снопта, немного доработанную. Однако, дальнейшего развития данная программа не получила.

Анализ графических редакторов для виртуальной реконструкции архитектурных и археологических памятников в своей работе проводит Саманта Портер, которая предлагает новый фотограмметрический инструмент для воссоздания трехмерных моделей (рис.3).

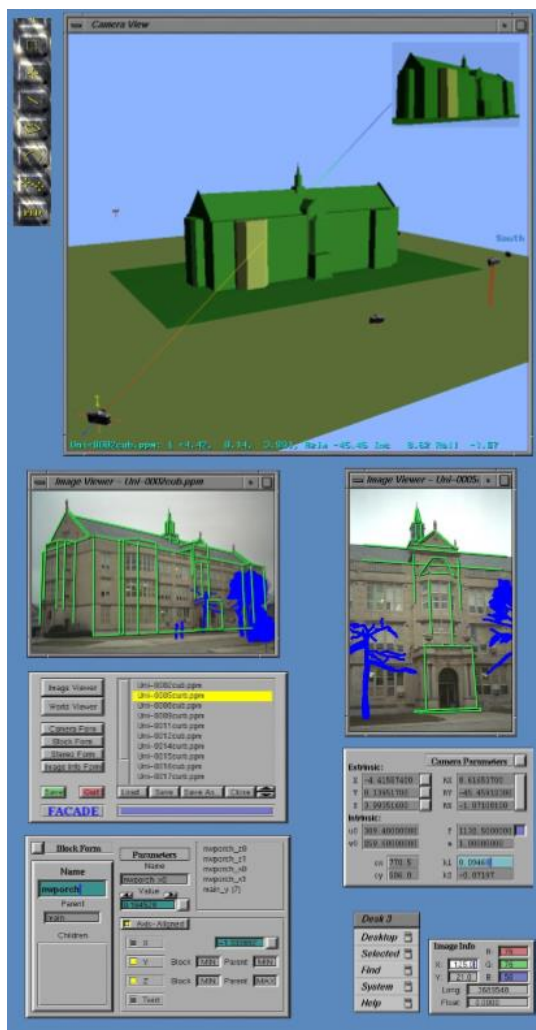


Рисунок 1 – Изображение рабочей области графического редактора, где показана одновременная работа нанесения маркировок на фотоизображения объекта и построение его объемной модели «Facade» [1]

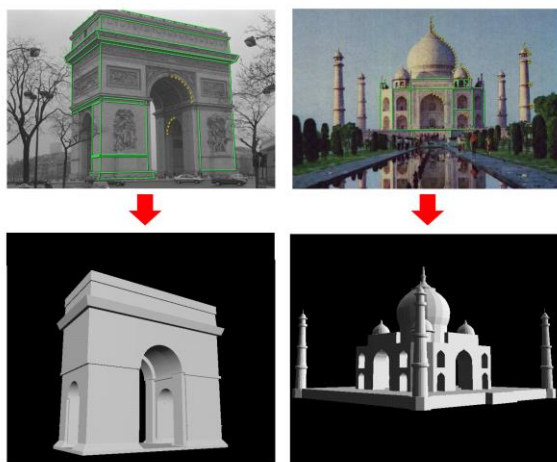


Рисунок 2 – Изображение примеров создания трехмерной модели в редакторе «Facade» [1]

Использование графического редактора AutoCAD с применением встроенного языка

программирования AutoLisp для автоматизации процесса геометрического моделирования утраченных памятников архитектуры предлагает в своей научной работе Марков О.Б. [3].

Проведя анализ этих работ, можно сделать вывод, что исследователи в области архитектуры и археологии стремятся добиться простоты и небольших временных затрат при работе с графическими редакторами. При этом они требуют точности построения чертежей или изображений и хорошего качества визуализации модели утраченного архитектурного памятника.

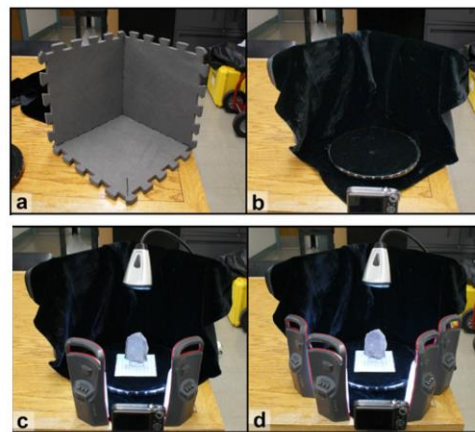


Рисунок 2 – Изображение фотограмметрического устройства для работы в полевых условиях, состоящего из круглой подставки, позволяющей прокручивать объект во время фотографирования.

Такое устройство заменяет использование лазерного сканера для работы с небольшими объектами [2]

Графические приложения, воссоздающие трехмерную модель утраченных памятников архитектуры с нуля

К такому классу графических приложений относятся редакторы компании Autodesk – AutoCAD, 3D Max, Revit [4].

К основным плюсам в работе с этими графическими редакторами можно отнести создание графической модели любого вида сложности, качественную визуализацию, возможность интегрирования форматов друг в друга.

Редактор Revit позволяет воссоздавать архитектурный объект, используя библиотеки архитектурных элементов.

Немаловажным фактором в работе с архитектурным объектом является соотношение масштабов и пропорций.

На высоком уровне находится инструмент визуализации трехмерных моделей с условием освещения и назначения материалов модели.

Возможность автоматизации процесса построения объекта, используя языки программирования, встроенные в данные

редакторы (AutoLisp в AutoCAD и MaxScript в 3D Max) может значительно ускорить построение, применяя интеллектуальное построение.

Графические редакторы компании Autodesk в последнее время все чаще используют BIM технологию, применяющую метод информационного моделирования [5]. Этот процесс позволяет контролировать как процесс проектирования, так и процесс реконструкции архитектурного здания на всех этапах, учитывая не только его объемно-планировочное решение, но и инженерно-конструктивное (рис.3).

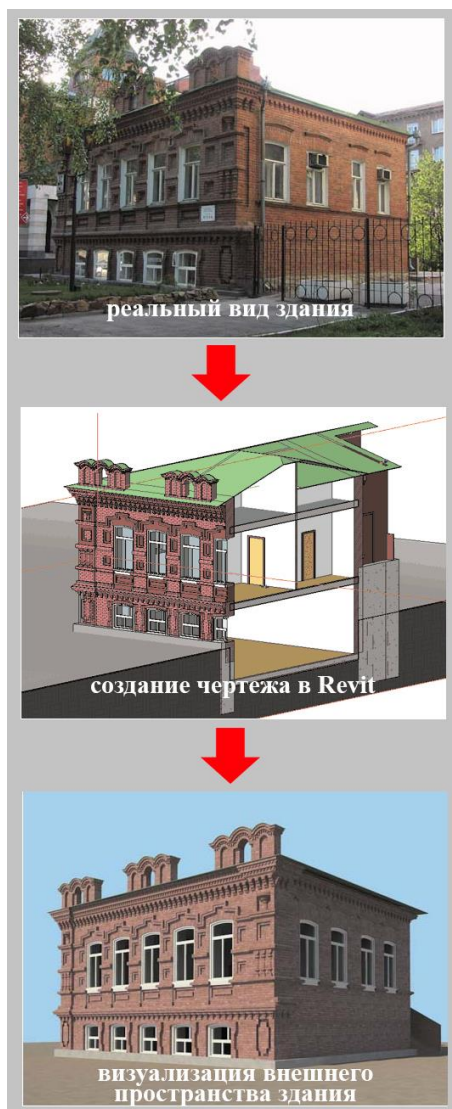


Рисунок 3 – Изображение процесса BIM моделирования объекта [5]

Графические редакторы типа AutoCAD и 3D Max были очень популярны в среде архитекторов до появления лазерного сканирования и разработки новых графических приложений по обработке данных, полученных путем сканирования. Работа с такого типа редакторами представляет сначала сбор информации, и потом построение объекта на

основании собранной информации. Однако, AutoCAD и 3D Max не работают с файлами, полученными в результате лазерного сканирования. А значит, что процесс реконструкции архитектурной модели в трехмерную приходится строить с нуля.

Графические приложения, обрабатывающие изображения, полученные лазерным сканером

В последнее время лазерное сканирование стало одним из популярных методов захвата и обработки данных в среде архитектуры и археологии (рис.4). Лазерное сканирование позволяет захватить часть утраченного архитектурного или археологического памятника со всех сторон с точностью до миллиметра, сохраняя данные в образе «облака точек», то есть, трехмерной модели с множеством полигонов. Данный тип модели можно открыть в графическом редакторе, специально предназначенном для работы с такого типа форматами [7].

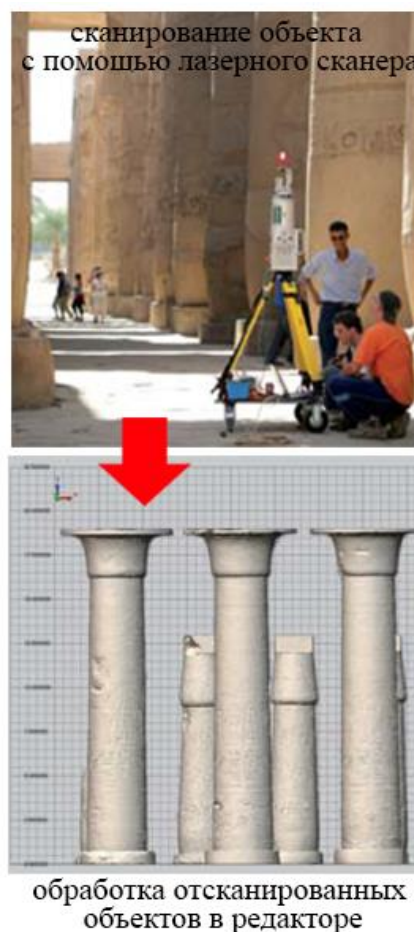


Рисунок 4 – Изображение процесса обработки объекта (колонны гипостильного зала в Карнакском храме), лазерного сканирования [7]

Компании, выполняющие услуги лазерного сканирования, обычно предлагают программных

обеспечения, ими же разработанные. Они мотивируют это тем, что модель, полученная путем лазерного сканера, является тяжелым файлом, поэтому, не каждый графический редактор способен не только обработать такую модель, но и открыть ее. Редакторы, открывающие модель лазерного сканирования – Autodesk Inventor, Rhino, Blender, Meshlab [8].

К основным преимуществам данного лазерного сканирования следует отнести: возможность отсканировать любой объект, возможность быстро получить трехмерную модель.

Однако у такого способа захвата объектов есть много недостатков: дорогостоящее оборудование и наличие обучающего материала, либо опытного инженера по использованию сканера, не всегда точное измерение объекта, особенно с отражающимися поверхностями и поверхностями черного цвета, тяжелый вес самого оборудования.

Дальнейшая обработка отсканированной трехмерной модели также требует специализированного графического приложения, не всегда бесплатного и нового для понимания. Сама модель содержит огромное количество полигонов, с которыми сложно работать, особенно, если модель содержит сложные декоративные или архитектурные элементы. В конечном итоге, обработка и редактирование модели, созданной из «облака точек», занимает много времени. Тяжелый вес трехмерной модели также тормозит процесс обработки и визуализации.

Графические приложения, использующие фотограмметрию для создания трехмерной модели объекта

Фотограмметрия является методом построения виртуальной реконструкции утраченного памятника архитектуры в том случае, когда нет возможности сделать натурные съемки лазерным сканером, либо архитектурный памятник является полностью утраченным и от него остались только старые фотографии, чертежи и рисунки.

Фотограмметрическое построение объекта можно разделить на два подхода:

- обработка изображений, сделанных с натурной съемки во всех ракурсах объекта;
- построение полностью утраченного объекта по нескольким фотографиям, чертежам и снимкам.

С первым подходом удачно справляется графический редактор Agisoft PhotoScan [9]. Принцип работы такой – сделать фотографии объекта со всех ракурсов по направлению 360 градусов и загрузить сделанные фотографии в рабочую область редактора. Agisoft PhotoScan создает «облако точек» по фотографиям, которое похоже на модель, созданную лазерным сканером (рис.5). Однако, данный формат модели весит

меньше, чем отсканированный.

Обработка и редактирование модели, созданной таким методом занимает много времени из-за большого количества точек и полигонов, с которыми приходится работать.

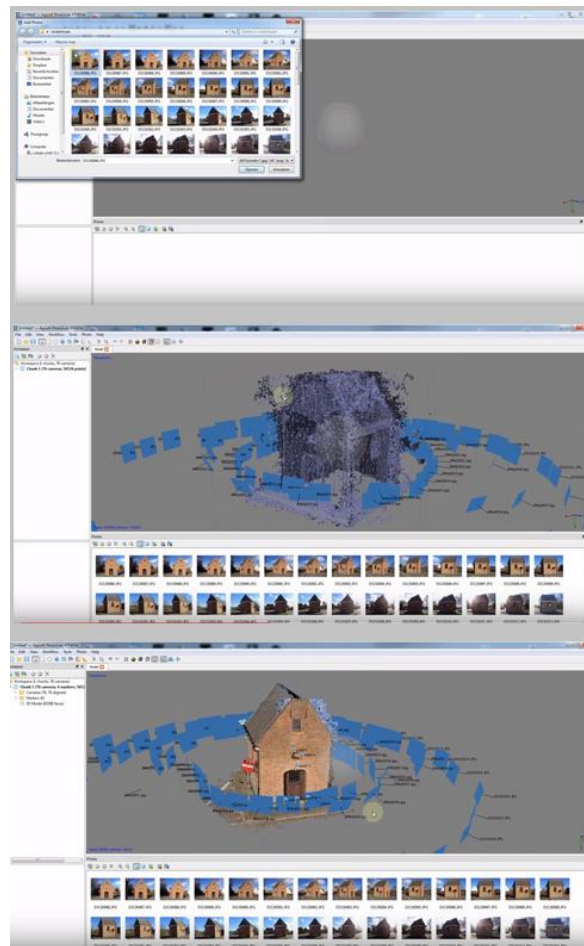


Рисунок 5 – Изображение процесса получения трехмерной модели объекта в редакторе Agisoft PhotoScan путем импортирования в рабочую область фотографий объекта, сделанных по периметру объекта [10]

Второй подход сложнее в выполнении из-за недостатка полной информации по утраченному объекту. Однако построение по существующим фотографиям проще, так как основывается на создании простых геометрических фигур, по которым дальше воссоздается объект. Для построения необходимо загрузить фотографию в рабочую область редактора и промаркировать на ней основные точки и грани, по которым дальше строится объект. С такой задачей справляется графический редактор Scanoma, который просуществовал до 2000 года, однако потом не стал совершенствоваться (рис.6).



Рисунок 6 – Изображение трехмерной модели архитектурного пространства в редакторе Canoma с использованием одной фотографии [12]

Выводы

Классификация графических приложений для виртуальной реконструкции утраченных памятников архитектуры показала, что различные подходы к реконструкции требуют использования определенных графических приложений. Следует также учитывать материальные затраты, связанные с виртуальной реконструкцией, так как существуют редакторы, воссоздающие трехмерную модель архитектурного памятника с использованием минимума специальной техники, как, например, фотограмметрический редактор Agisoft PhotoScan, и не требующие особых навыков в работе, как редакторы, обрабатывающие модель объекта, полученную путем лазерного сканирования. Однако, оптимального графического приложения, которое бы удовлетворяло всем требованиям по виртуальной реконструкции архитектурных памятников, таких как простота в работе, использование нескольких подходов для создания виртуальной реконструкции, работа со всеми форматами трехмерной модели, недорогая цена, нет. Перспективным направлением дальнейшей работы является разработка инструментального средства виртуальной реконструкции на базе графического редактора, который соответствует большинству требований, предъявляемых для создания реконструкции с целью его доработки под соответствующие задачи.

Литература

1. Debevec P.E. Modeling and Rendering Architecture from Photographs. Doctoral dissertation. University of California at Berkeley. 1996
2. Thi Porter S., Roussel M., Soressi M. A Simple Photogrammetry Rig for the Reliable Creation of 3D Artifact Models in the Field. *Advances in Archaeological Practice*, 2016, 4(1), 71-86 doi:10.7183/2326-3768.4.1.71
3. Марков Б.Г. Автоматизация геометрического моделирования утраченных памятников архитектуры по иконографическим материалам: дисс. канд. техн. наук: 05.13.16 / Петрозаводский государственный университет. – Петрозаводск, 2000. – 146 с.
4. Autodesk. Программное обеспечение для 3D-проектирования, дизайна, графики и анимации: autodesk.ru URL: <http://www.autodesk.ru> (19.03.2016)
5. Информационное моделирование зданий: опыт применения в реконструкции и реставрации // Журнал «САПР и графика»: sapr.ru URL: <http://www.sapr.ru/Article.aspx?id=20649> (19.03.2016)
6. 3D сканирование объектов и трехмерное моделирование // Клона: klona.ua URL: <http://klona.ua/3d-skanirovanie.html> (19.03.2016)
7. Chandelier L., Chazaly B., Egels Y., Laroze F., Schelstraete D. Numérisation 3D et déroulé photographique des 134 colonnes de la Grande Salle Hypostyle de Karnak. *Revue XYZ • N° 120 – 3e trimestre 2009*, 33-39.
8. Чем открыть .stl // Чем открыть. Библиотека расширений: chem-otkrit.ru URL: <http://chem-otkrit.ru/format/stl> (19.03.2016)
9. Agisoft: agisoft.com URL: <http://www.agisoft.com/> (20.03.2016)
10. Agisoft – 3D model building // «Youtube» - видеохостинг: https://www.youtube.com/watch?v=FkcmhVFD_38&list=PLAlduMKqE4RBod2ir-lFhjsCmttKOloMz (20/03/2016)
11. Adout Canoma // CANOMA. Quickly create realistic 3D models from photographs: canoma.com URL: <http://www.canoma.com/> (20.03.2016)
12. Big Ben, London - 3D model made from one photograph // «Youtube» - видеохостинг: <https://www.youtube.com/watch?v=Ov39VAYRvmE> (10/05/2016)

Руденко М.П. Инструментальные средства виртуальной реконструкции утраченных памятников архитектуры. Рассматриваются популярные графические приложения для виртуальной реконструкции утраченных памятников архитектуры, используемые в среде архитекторов и археологов. Приводится их классификация по способу построения и визуализации трехмерной модели на следующие типы редакторов: воссоздающие модель утраченного архитектурного памятника с нуля; работающие с моделями, полученными путем лазерного сканирования; и путем фотограмметрии, анализируется качество их работы, на основе анализа выбирается оптимальное графическое приложение для работы с виртуальной реконструкцией утраченных памятников архитектуры.

Статья поступила в редакцию 20.05.2016
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом

УДК 004.056.5

Подсистема защиты авторского права в программном обеспечении

А.В. Чернышова, Д.В. Кубашевский
Донецкий национальный технический университет
alla@pmi.dgtu.donetsk.ua, dehax12@gmail.com

Чернышова А.В., Кубашевский Д.В. Подсистема защиты авторского права в программном обеспечении. В данной работе выполнен анализ существующих механизмов защиты авторского права на программное обеспечение и предложено описание работы программной подсистемы защиты ПО с использованием криптографических алгоритмов шифрования.

Введение

Ежедневно продаётся огромное количество копий программного обеспечения. Каждая такая копия обычно включает в себя систему защиты от копирования и проверки лицензии на приобретённое ПО. Однако зачастую подобные системы защиты ненадёжны, так как их разработке уделяется значительно меньше времени и внимания, чем оригинальному продукту — объекту защиты. Поэтому программное обеспечение, которое имеет такую слабую защиту, обычно подвергается "взлому" "крэкерами". Далее взломанные копии попадают в открытый доступ, где обычные пользователи бесплатно скачивают полностью работоспособное ПО. Таким образом, разработчики ПО несут огромные финансовые потери.

Лицензия на программное обеспечение — это правовой инструмент, определяющий использование и распространение программного обеспечения, защищённого авторским правом. Обычно лицензия на программное обеспечение разрешает получателю использовать одну или несколько копий программы, причём без лицензии такое использование рассматривалось бы в рамках закона как нарушение авторских прав издателя.[1]

На рынке программного обеспечения в настоящий момент активно используется программное обеспечение с открытым кодом, но также большое количество коммерческих программных продуктов.

Сегодня используются такие виды лицензий на программное обеспечение: открытая лицензия (Open Source), бесплатная лицензия (Freeware, GPL, Adware, Postcardware, Donationware, Nagware, begware) условно-бесплатное программное обеспечение (ShareWare, Trial, trialware, Demo, demoware),

коммерческое программное обеспечение (Commercial). [1]

Цель работы — описать подсистему защиты авторского права в программном обеспечении с использованием криптографических алгоритмов шифрования для возможного использования при лицензировании условно-бесплатного или коммерческого программного обеспечения.

Обзор существующих механизмов защиты авторского права на программное обеспечение

Для защиты ПО используется ряд методов, таких как:

- алгоритмы запутывания - используются хаотические переходы в разные части кода, внедрение ложных процедур - "пустышек", холостые циклы, искажение количества реальных параметров процедур ПО, разброс участков кода по разным областям ОЗУ и т.п.;

- алгоритмы мутации - создаются таблицы соответствия операндов - синонимов и замена их друг на друга при каждом запуске программы по определённой схеме или случайным образом, случайные изменения структуры программы;

- алгоритмы компрессии данных - программа упаковывается, а затем распаковывается по мере выполнения;

- алгоритмы шифрования данных - программа шифруется, а затем расшифровывается по мере выполнения;

- вычисление сложных математических выражений в процессе отработки механизма защиты - элементы логики защиты зависят от результата вычисления значения какой-либо формулы или группы формул;

- методы затруднения дизассемблирования - используются различные приёмы, направленные на предотвращение дизассемблирования в

пакетном режиме;

- методы затруднения отладки - используются различные приёмы, направленные на усложнение отладки программы;

- эмуляция процессоров и операционных систем - создаётся виртуальный процессор и/или операционная система;

- нестандартные методы работы с аппаратным обеспечением - модули системы защиты обращаются к аппаратуре. [2]

Практически для каждого популярного ПО можно найти в интернете взломанную рабочую копию. В основном, защитные механизмы делятся на две основные категории: основанные на знании пароля, серийного номера и основанные на владении носителя информации с ключом, файлов программы.

Защита, основанная на знании, является наиболее уязвимой, так как легальные пользователи могут сообщить всем серийный номер, с помощью которого другие пользователи могут запустить программу нелегально. В случае если серийный номер уникально генерируется для каждого пользователя и является функцией его имени, активировать программу одинаковым серийным номером не получится. В этом случае придётся проанализировать алгоритм генерации серийного номера в зависимости от имени пользователя либо избавиться от алгоритма защиты, "пропатчив" исполнимый код программы, то есть, изменив в бинарном коде программы условие, при котором система защиты посчитает ПО активированным.

Более стойкими являются системы защиты второй категории, основанные на владении некоторым носителем, который практически невозможно воссоздать самостоятельно. Но огромным минусом такой защиты является тот факт, что ключевой носитель может быть потерян или испорчен. Тогда пользователю придётся пройти процедуру восстановления такого носителя при условии, если разработчик её вообще поддерживает. [3]

Алгоритм работы подсистемы защиты авторского права программного обеспечения

Предположим, имеется разработанное программное обеспечение, которое нужно защитить от незаконного копирования и дальнейшего использования.

При выборе алгоритма работы подсистемы защиты авторского права программного обеспечения будем считать, что программное обеспечение условно-бесплатное

или коммерческое.

Основная идея подсистемы защиты авторского права ПО состоит в том, что существует некоторый файл, без которого программа не способна выполнить какие-либо важные для пользователя функции (набор функций программы ограничен). Для доступа ко всему набору функций ПО, необходимо получить доступ к дополнительному файлу. Получить расширенные возможности по работе с программой пользователь может лишь предварительно подтвердив свои лицензионные данные — права на использование данного ПО. Таким образом, невозможно будет использовать программу без покупки лицензии на неё. [4]

Пусть защищаемое ПО состоит из запускаемого исполнимого модуля *.exe и вспомогательного модуля *.dll [5], благодаря которому будет возможно работать с полной версией программы, доступной в платной версии. Покупая программу, пользователь получает установочный дистрибутив, который содержит только лишь запускаемый модуль *.exe.

После установки программы, система защиты запрашивает данные о лицензии, например, персональные данные пользователя и соответствующий ключ, которые данный пользователь получил при покупке ПО.

На рисунке 1 изображён процесс получения вспомогательного модуля программы.

Алгоритм получения пользователем вспомогательного модуля программы будет следующим. При установке программного обеспечения и вводе пользователем данных (ФИО, электронный адрес, номер телефона), персональные данные пользователя и некоторая информация об используемом аппаратном обеспечении компьютера (серийный номер процессора, жёсткого диска, MAC-адрес сетевого адаптера), а также ключ, который пользователь получил при покупке, передаётся на сервер фирмы-разработчика устанавливаемого программного обеспечения с использованием защищённого канала связи (использование защищённого протокола передачи данных) [6]. Полученная информация на сервере записывается в базу данных фирмы-разработчика ПО. Таким образом, на сервере хранится вся информация, идентифицирующая пользователя, установившего программный продукт.

Следующим этапом является использование криптографических алгоритмов [7] для шифрования вспомогательного модуля программы *.dll.

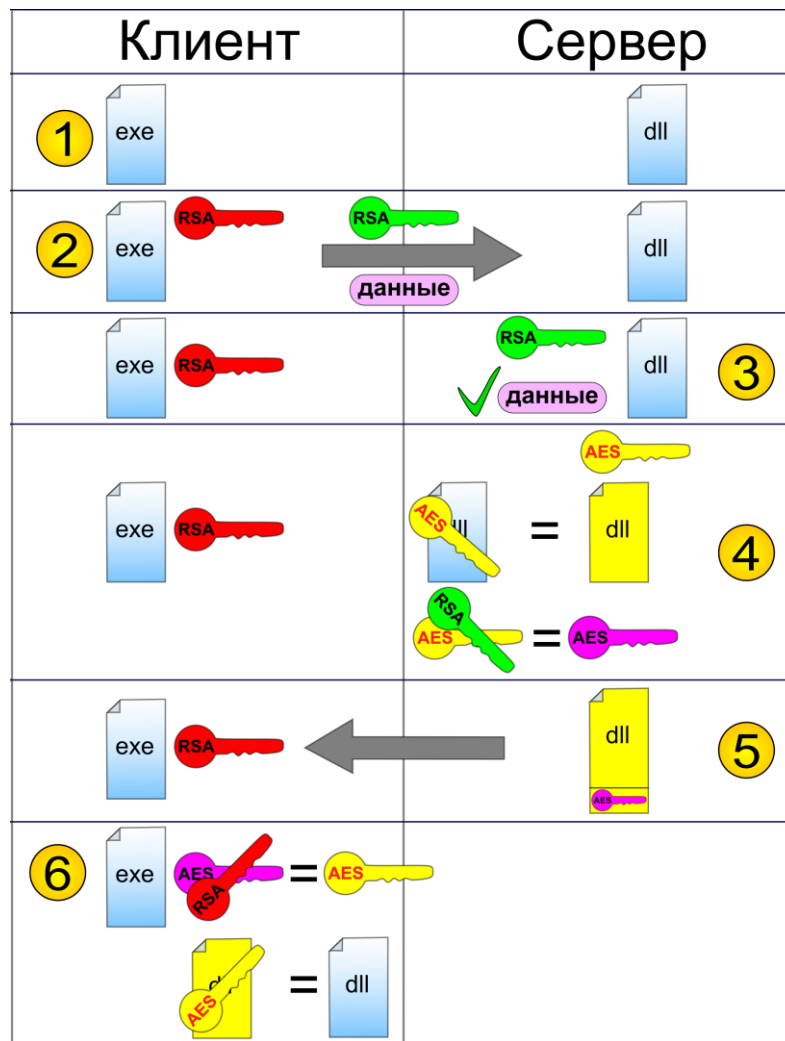


Рисунок 1 — Получение вспомогательного модуля программы с использованием криптографических алгоритмов шифрования

Файлы *.dll можно защитить, используя симметричный алгоритм шифрования (например, AES) [8]. Для генерации уникального ключа симметричного алгоритма, которым будет зашифрован вспомогательный модуль, можно использовать полученную информацию из регистрационной формы пользователя.

Таким образом, вспомогательный модуль программы будет зашифрованным передаваться по сети, храниться на машине пользователя.

При использовании симметричных алгоритмов шифрования для защиты файлов, хранящихся на диске или передаваемых по сети, рекомендуется ключи, которыми шифруются файлы, также защищать. Для защиты ключей симметричных криптографических алгоритмов используют криптографические алгоритмы с открытым ключом, например алгоритм RSA.[7]

Предположим, что на сервере фирмы-разработчика ПО уже сгенерирован ключ симметричного алгоритма шифрования для пользователя, подтверждающего свою

лицензию и ожидающего вспомогательного модуля программы.

Для шифрования самого ключа симметричного алгоритма используем схему шифрования с открытым ключом [7].

При регистрации пользователя в программе должна сгенерироваться пара ключей (открытый и закрытый).[8] В дальнейшем закрытый ключ будет храниться на машине пользователя, открытый ключ вместе с персональными данными пользователя передаётся на сервер фирмы разработчика и также хранится в базе данных. С помощью открытого ключа сервер зашифрует ключ симметричного алгоритма, которым был зашифрован вспомогательный модуль программы, допишет его в зашифрованный вспомогательный модуль (например, в конец файла) и передаст программе пользователя.

При необходимости использования вспомогательного модуля программой сначала будет извлечён зашифрованный открытым ключом ключ симметричного алгоритма, с

помощью закрытого ключа ассиметричного алгоритма будет расшифрован ключ симметричного алгоритма, затем с помощью ключа симметричного алгоритма будет расшифрован сам файл вспомогательного модуля. Программа сообщает об успешной активации лицензии. Теперь, при необходимости обращения к вспомогательному модулю, программа расшифровывает его "на лету", а при завершении работы удаляет расшифрованный файл из памяти.

Достоинства и недостатки подсистемы защиты авторского права программного обеспечения

Описанная подсистема защиты ПО обладает рядом достоинств и недостатков.

Достоинства:

- обычный опытный пользователь не сможет обойти защиту;
- система защиты способна справиться с крэкерами-новичками, а также запутать крэкеров среднего уровня;
- в случае использования эффективного средства обфускации для сокрытия алгоритма подсистемы защиты, а также использования методов виртуализации функций системы защиты, даже опытным крэкерам понадобится немало времени, чтобы исследовать и взломать алгоритм проверки лицензии;
- для получения вспомогательного модуля, для защиты которого используются криптографические алгоритмы, нужно знать ключи и используемые криптографические алгоритмы;
- взлом базы данных сервера фирмы-разработчика ПО, на котором хранится персональная информация о пользователях и их серийных номерах программы, не даст информации обо всех используемых ключах криптосистемы, а значит, не позволит расшифровать вспомогательный модуль программы и использовать незаконно программный продукт;
- для предотвращения использования метода перебора ключей, рекомендуется использовать значительную длину ключей в симметричных и ассиметричных алгоритмах шифрования.

Недостатки:

- без обфускации подсистема защиты не защищена от декомпиляции и дизассемблирования с целью внесения изменений в алгоритм работы функций защиты (патчинг);
- при активации программы необходимо подключение к интернету;
- в случае использования информации об аппаратном обеспечении конечного

пользователя для генерации уникального ключа, которым будет зашифрован вспомогательный модуль, при замене какой-либо аппаратной части необходимо будет сообщать разработчику об этом и проходить снова процедуру активации программы.

Выводы

В ходе выполнения данной работы были рассмотрены и проанализированы существующие механизмы защиты авторского права на программное обеспечение, предложен общий алгоритм работы подсистемы защиты авторского права программного обеспечения, рассмотрена возможность использования криптографических алгоритмов шифрования для защиты вспомогательного модуля программного обеспечения, представлены достоинства и недостатки предлагаемой подсистемы защиты. На данный момент не существует абсолютной программной защиты ПО, которое невозможно было бы взломать, если не учитывать время, затраченное на исследование этой защиты. Однако наиболее эффективными средствами при создании подобного рода систем проверки лицензии являются использование криптографических алгоритмов и обфускация функций подсистемы защиты.

Литература

1. Виды лицензий на программное обеспечение // Information Technology Engineering Projects. [Электронный ресурс]. – Режим доступа: http://www.it-ep.ru/knowledge_base/software_licensing/type_of_software_license/
2. Серада С.А. Оценка эффективности систем защиты программного обеспечения // CITForum. [Электронный ресурс]. – Режим доступа: <http://citforum.ru/security/software/sereda1/>
3. Касперски К., Рокко Е. Искусство дизассемблирования. – СПб.: БХВ-Петербург, 2008. – 896 с.
4. Складов Д. Искусство защиты и взлома информации. – СПб.: БХВ-Петербург, 2004. – 288 с.
5. Работа с библиотеками динамической компоновки (DLL) // Visual C++. [Электронный ресурс]. – Режим доступа: <http://www.softzenware.com/visual/visual12.html>
6. Каналы защищенной передачи данных //Энциклопедия теоретической и прикладной криптографии. [Электронный ресурс] - Режим доступа: <http://cryptowiki.net/>

7. Столлингс В. Криптография и защита сетей. - М.: Издательский дом "Вильямс", 2001 г., - 669 стр.

8. Соколов А.В., Шаньгин В.Ф. Защита информации в распределенных корпоративных сетях и системах. - М.: ДМК Пресс, 2002 г. – 655 стр.

Alla Chernyshova, Denis Kubashevskiy. The Copyright Protection Subsystem in the Software.

In this article, the authors analyzed the existing mechanisms for the copyright protection of software and suggested the description of software protection subsystem which uses cryptographic encryption algorithms.

Keywords: software license, additional module, software protection, symmetric encryption algorithms, asymmetric encryption algorithms

А.В. Чернишова, Д.В. Кубашевський. Підсистема захисту авторського права у програмному забезпеченні. У даній роботі були проаналізовані наявні механізми захисту авторського права на програмне забезпечення та запропонований опис роботи програмної підсистеми захисту ПЗ із використанням криптографічних алгоритмів шифрування.

Ключові слова: ліцензія на програмне забезпечення, додатковий модуль, захист ПО, симетричні алгоритми шифрування, асиметричні алгоритми шифрування

Статья поступила в редакцию 20.05.2016

Рекомендована к публикации д-ром техн. наук В.Н. Павлышом

CONTENT

Andryukhin A.I., S.N. Sudakov. Computer calculation problem of interaction of two liquid gravitating bodies: the moons of Jupiter.....	4
Andruckin A.I., Poletaev V.A. Recursive implementation of dynamic programming problems..	12
Beregovych Yu.V., Orlov Yu.K., Gryn O.V. Development of mathematical model of process of cultivation of mushrooms in a kultivatsionny chamber...	20
G. V. Gorr, A. A. Ilyukhin, A. M. Kovalev. Mathematical aspects of modeling motions of a gyrostat with a fixed point.....	26
K.N. Efimenko. The use of hybrid FPGA for implementing compositional control unit with identification of outputs	31
Konopatskiy E.V. Some questions of mathematical and computer modeling in building.....	37
Morozova O.V. Analysis of possibilities and disadvantages of technology for building knowledge bases in modern CAD systems..	43
Pavlysh V.N., Tarabayeva S.V., Turchanin G.I. The automatic projecting of enriched coal drying technological schemes.....	49
Elena Chepikova, Elena Savkova, Maksim Privalov. Research of recommender systems' algorithms.....	57
Руденко М.П. Инструментальные средства виртуальной реконструкции утраченных памятников архитектуры.....	62
Alla Chernyshova, Denis Kubashevskiy. The Copyright Protection Subsystem in the Software.....	68

Научное издание

ИНФОРМАТИКА И КИБЕРНЕТИКА

Научный журнал

(на русском, украинском, английском языках)

№ 2(4)-2016

Ответственный за выпуск А.И. Андрюхин

Технические редакторы Т.В. Завадская, А.В. Звягинцева

Компьютерная верстка Д.А. Филипишин

Подписано к печати 24.06.2016. Формат 60×84 1/8. Бумага офсетная. Печать ризографическая.
Усл. печ. лист. 10. Уч.- изд. лист. 7,1. Тираж 100 экз.

Адрес редакции: ДНР, 83001, г. Донецк, ул. Артема, 58, ГВУЗ «ДонНТУ», 5-й учебный корпус,
к. 425. Тел.: +38 (062) 301-08-51 E-mail: infcyb.donntu@yandex.ru, URL: <http://infcyb.donntu.org>

Издатель Государственное высшее учебное заведение «Донецкий национальный технический
университет» ДНР, 83001, г. Донецк, ул. Артема, 58. Тел.: +38 (062) 301-08-67, +38 (062) 301-
09-67 Свидетельство о государственной регистрации субъекта издательского дела: серия ДК
№2982 от 21.09.2007

Отпечатано ООО фирма «ДРУК-ИНФО» ЛУН, 83001, г. Донецк, ул. Артема, 58, к. 113, тел.: +38
(062) 335-64-55