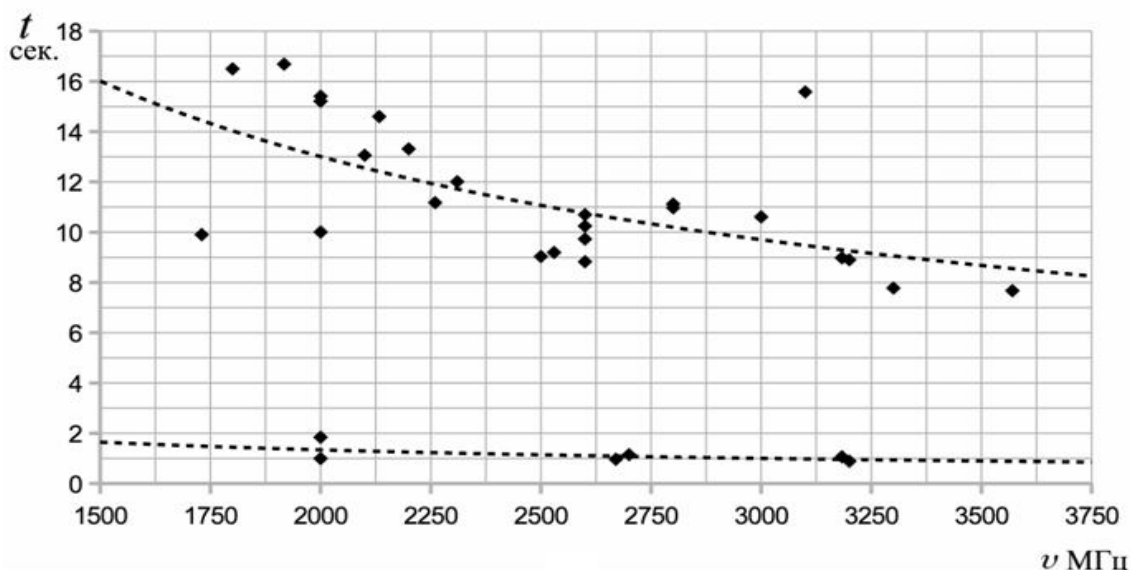


случае является многозначной.

Пунктиром выделены графики функции степенной регрессии.



**Рис. 1.** График зависимости времени выполнения от частоты процессора.

По графику визуально можно выделить две группы данных. Первая группа  $g_1 \in G_1$  лежит в промежутке  $\bar{t}_1 \in [7.674, 15.211]$ , а вторая  $g_2 \in G_1$  – в  $\bar{t}_2 \in [0.893, 1.152]$ . Найдём функции степенной регрессии для каждой группы и наложим их графики на полученные данные. Для группы  $g_1$  функция имеет вид:  $R_1(\nu) = 3167.82\nu^{-0.72}$ ; для  $R_2(\nu) = 312.09\nu^{-0.72}$ . С помощью этих функций, можно грубо представить зависимость времени выполнения приложения от частоты процессора, и в общем, подтвердить зависимость, но образование 2-х разных групп данных ставит под сомнение исследуемую гипотезу.

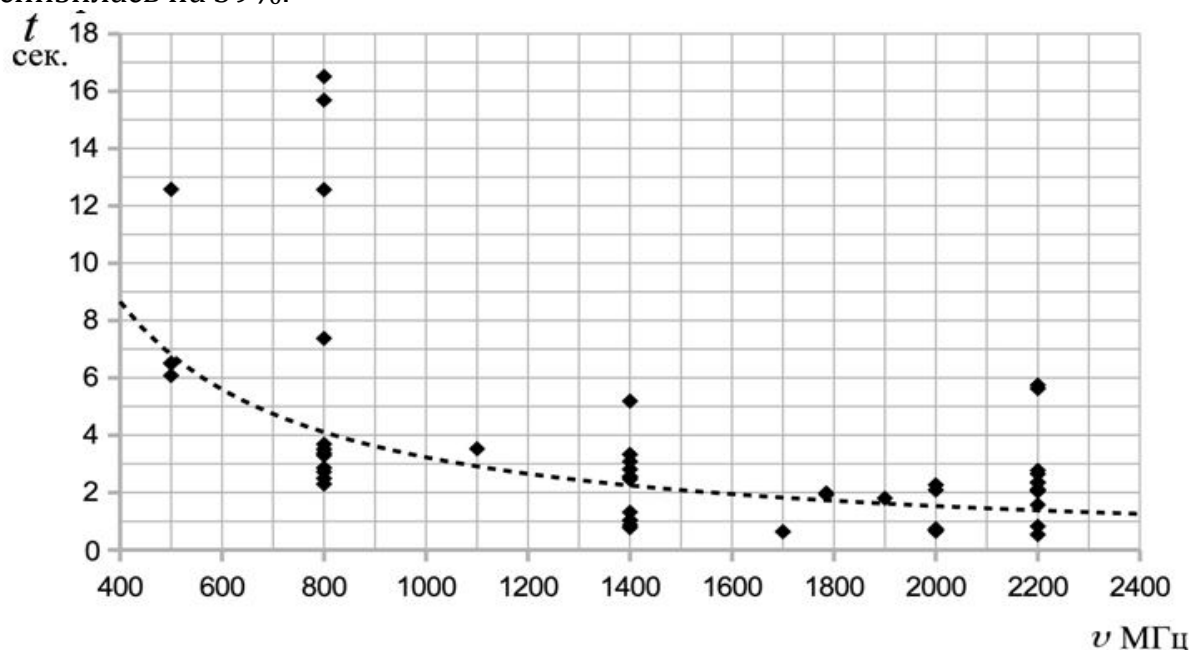
Причина образования группы  $g_2$  проста: некоторые участники эксперимента использовали новую версию компилятора.

Чтобы более полно учитывать факторы, влияющие на работу приложений, можно воспользоваться дополнительными тестовыми приложениями, собирающими информацию о производительности отдельных частей компьютера. Например, в качестве оценочных задач для набора тестов SPEC'92 для оценки производительности коммерческих систем используются: задача сжатия информации, задача упрощения булевой функции, проецирование лучей через оптическую систему и прочие [3]. Таким образом, замеряется скорость работы с целыми, дробными, логическими переменными, работа с памятью и внешними ЗУ. Тестовые приложения позволяют учитывать не только аппаратную составляющую компьютера, но и программную. Время их работы и будем считать *относительным параметром*. Совокупность относительных параметров позволяет снизить степень неопределённости, поскольку множество неучтенных факторов влияют на работу не только контрольного приложения, но и тестовых. Это значит, что найти

компьютеры с одинаковым временем выполнения тестового приложения сложнее, чем с одинаковым процессором.

Основываясь на вышесказанном, построим **гипотезу 2**: *использование относительных параметров позволяет получить более точную оценку.* Соответственно, ожидаемую зависимость можно сформулировать следующим образом: *чем меньше время выполнения тестового (замерочного) приложения, тем меньше прогнозируемое время контрольного.*

Проверим эффективность применения относительных параметров для оценивания времени выполнения приложения. Согласно условию эксперимента имеется ограниченное число компьютеров разной мощности и задача численного интегрирования (шаг, функция и промежуток не меняются). Программа компилируется на разных компьютерах разными компиляторами и опциями и запускается несколько раз. Для сравнения, были найдены величины среднеквадратичной ошибки для функций степенной регрессии, построенные для  $G_2 = \{(v_n, v_n)\}_{n=1}^{66}$  ( $R_{G_2}(v) \approx 543360116.14v^{-1.075}$ , рис.2) и  $G_3 = \{(T3_n, t_n)\}_{n=1}^{66}$  ( $R_{G_3}(T3) \approx 21.823 \cdot [T3]^{0.734}$ ), где  $T3$  – время выполнения тестовой программы по замеру скорости проведения операций над целыми числами. В результате получено:  $E_{R_{G_2}} = 4409360.4$  нс, в то время как  $E_{R_{G_3}} = 2686117.52$  нс, погрешность снизилась на 39%.



**Рис. 2.** График зависимости времени выполнения от частоты процессора. Пунктиром выделены графики функции степенной регрессии.

Проверим **гипотезу 3**: повышение размерности функции отображения  $f$  за счёт увеличения числа тестовых приложений, позволяет повысить точность прогноза. Предположим, что существует некоторая

функция

$$Time(T1, T2, T3, T4, T5, T6, T7) = t.$$

Для поиска сложной многомерной функции воспользуемся многослойным перцептроном [4]. Он обладает свойством универсальной аппроксимации и способен решать задачи интерполяции. Его скрытые слои выделяют статистические зависимости высокого порядка. Это позволяет применять функцию *Time* для большего количества компьютеров, чем то, которое было использовано при обучении нейросети.

### 3. Вычислительный эксперимент

Пусть имеется 7 тестовых приложений для замера относительной мощности компьютера и некая контрольная задача (выше упомянутая задача интегрирования; область интегрирования, функция и шаг считаются постоянными). Тестовые приложения измеряют скорость работы генератора случайных чисел (T1), скорость выделения/заполнения/очистки большого участка памяти (T2), выполнения арифметических операций с различными типами данных (T3 – T6), скорость вычисления сложной функции (T7). Они скомпилированы тем же компилятором и с теми же опциями, что и контрольная задача. Имеется 17 компьютеров, на которых проведены замеры тестовых приложений и контрольной задачи (всего 66 замеров, в среднем  $\approx 3.88$  замера на компьютер).

В таблице 1 приведена часть полученных данных (в наносекундах):

**Табл. 1. Результаты замеров (в нс.)**

T1	T2	T3	T4	T5	T6	T7	R
1258446	835494	6164579	8897774	1292405 8	6126578	7553222	5746288
1258446	835494	6164579	8897774	1292405 8	6126578	7553222	5632924
988033	68416	4234466	4243164	4148505	4135107	6459890	1576530
517369	193213	2031133	2047022	2048443	2062096	3082111	535604
517369	193213	2031133	2047022	2048443	2062096	3082111	533015
517369	193213	2031133	2047022	2048443	2062096	3082111	826273
6499735	10789480	29580576	4089941 9	4270878 6	3098283 2	2644260 8	6496516
699185	276030	2651103	2652607	2651999	2653769	3589381	635932
...	...	...	...	...	...	...	...

Таким образом, обучающая выборка имеет вид  $\{(T1_i, T2_i, \dots, Tn_i, R)\}_{i=1,66}^{n=1,7}$ , где  $Tn$  – независимая входная переменная, (в данном случае, время выполнения тестового приложения),  $R$  – зависимая выходная переменная (время выполнения контрольного приложения).

Используем полученные данные как обучающую выборку для нейросети с семью входными, пятью скрытыми и одним выходным нейронами с сигмоидальными активационными функциями. Для обучения используем метод Левенберга-Марквардта [5-7], поскольку более эффективен, в отличие от классического метода обратного распространения ошибки. В процессе обучения редко достигается

некоторое минимальное значение ошибки (если оно очень мало), поскольку справедливо утверждение 1, поэтому используется ограничение на количество итераций, чтобы гарантировать завершение обучения за разумный срок. В этом эксперименте проведено 2000 итераций обучения. Величина среднеквадратичной ошибки получилась  $E \approx 953311$  нс.

Пусть необходимо сделать прогноз времени выполнения контрольной задачи для нового компьютера. Оценим его мощность с помощью тестовых программ и найдём T1 – T7. После подачи данных на вход нейросети был получен прогноз  $t = 2055935$  нс. Реальное время выполнения приложения составило  $t' = 2299611$  нс. Для сравнения:

$$R_{G_2}(1600) \approx 1948593 \text{ нс.}, R_{G_3}(3589236) \approx 1411130 \text{ нс.}$$

В ходе многих экспериментов, точность оценок не всегда была удовлетворительной. Метод обучения Левенберга-Марквардта очень чувствителен к начальным значениям весовых коэффициентов. Сказывается недостаточность исходных данных для многомерной функции. Из этого следует, что для повышения точности прогнозирования необходимо постоянно обеспечивать нейросеть новыми данными и применять методы очищения данных от шумов.

В ходе различных экспериментов ставилась задача получения максимально точной оценки времени выполнения линейного алгоритма. Благодаря исследованиям была установлена эффективность определения мощности компьютера через различные оценочные тесты и была проанализирована их взаимосвязь с временем выполнения реальных приложений.

#### **4. Вывод**

В результате исследований получен способ достаточно достоверного прогнозирования времени выполнения линейного алгоритма на широком спектре оборудования. Для этого была подобрана совокупность параметров, формирующих историческую базу данных и учитывающую как программную, так и аппаратную составляющую вычислительного устройства. Высокая адаптивность системы позволяет применять её в динамичных средах, таких как сети добровольных вычислений.

В статье предложен способ обработки данных исторической базы, направленный на получение прогнозов с помощью нейронных сетей.

Для подтверждающего эксперимента были собраны данные с различного оборудования и проведены сравнительные вычисления.

#### **Литература**

1. Gibbons R. A historical application profiler for use by parallel schedulers // In Job Scheduling Strategies for Parallel Processing. Springer Verlag, 1997. P. 58–77.
2. Resource selection using execution and queue wait time predictions: Rep.; Executor: Warren Smith, Parkson Wong: 2002.
3. Conte T. M., Menezes K. N. P., Sathaye S. W. A technique to determine power-efficient, high-performance superscalar processors // Proceedings of the 28<sup>th</sup> Annual Hawaii International



Conference on System Science / Computer Architecture Research Laboratory, Department of Electrical and Computer Engineering University of South Carolina. Columbia, South Carolina 29208, 1995.

4. Хайкин С. Задачи обучения // Нейронные сети. М.: Издательский дом «Вильямс», 2006. С. 56.

5. Wikipedia. Levenberg-marquardt algorithm — wikipedia, the free encyclopedia. 2011. [Online, accesse 5-April-2011]. URL: [http://en.wikipedia.org/wiki/Lavanberg-Marquardt\\_algorithm](http://en.wikipedia.org/wiki/Lavanberg-Marquardt_algorithm).

6. Sousa C. Neural network learning by the levenberg-marquardt algorithm with Bayesian regularization (part 1). 2009. URL: [http://crsouza.blogspot.com/2009/11/neural-network-learning-by-levenberg\\_18.html](http://crsouza.blogspot.com/2009/11/neural-network-learning-by-levenberg_18.html).

7. Sousa C. Neural network learning by the levenberg-marquardt algorithm with Bayesian regularization (part 2). 2009. URL: <http://crsouza.blogspot.com/2009/11/neural-network-learning-by-levenberg.html>.

## **Алгоритм построения оптимального эйлера покрытия для многосвязного графа**

Задачи нахождения маршрутов, удовлетворяющих определенным ограничениям, появились из конкретных практических ситуаций. Например, в задачах раскроя листового материала плоский граф является моделью раскройного плана, а маршрут, покрывающий все ребра, определяет траекторию режущего инструмента.

Моделью раскройного листа будем считать плоскость  $S$ , моделью раскройного плана – плоский граф  $G$  с внешней гранью  $f_0$  на плоскости  $S$ . Для любой части графа  $J \subseteq G$  (части траектории движения режущего инструмента) обозначим через  $\text{Int}(J)$  теоретико-множественное объединение его внутренних граней (объединение всех связных компонент множества  $S \setminus J$ , не содержащих внешней грани). Тогда  $\text{Int}(J)$  можно интерпретировать как отрезанную от листа часть. Множества вершин, ребер и граней графа  $J$  будем обозначать через  $V(J)$ ,  $E(J)$  и  $F(J)$  соответственно, а число элементов множества  $M$  – через  $|M|$ . При этом требуется, чтобы отрезанная от листа часть не требовала дополнительных разрезов. В общем случае раскройный план представляет многосвязный граф, состоящий из вложенных компонент связности. Задача состоит в построении покрытия графа ребрами, учитывающего ограничения, наложенные практической задачей. Автором разработаны алгоритмы построения такого покрытия для связного графа [1, 2], а также алгоритмы нахождения допустимого эйлера покрытия с упорядоченным охватыванием для многосвязного графа [3]. В данной работе модифицируем алгоритм, изложенный в [2, 3], на случай многосвязного графа с вложенными компонентами связности.

В соответствии с [1] будем говорить, что цепь  $C = v_1 e_1 v_2 e_2 \dots v_k$  в плоском графе  $G$  имеет **упорядоченное охватывание**, если для любой его начальной части  $C_l = v_1 e_1 v_2 e_2 \dots e_l$ ,  $l \leq (|E|)$  выполнено условие  $\text{Int}(C_l) \cap E = \emptyset$ .

Также будем говорить, что последовательность реберно-непересекающихся цепей

$$C^0 = v^0 e_1^0 v_1^0 e_2^0 \dots e_{k_0}^0 v_{k_0}^0, C^1 = v^1 e_1^1 v_1^1 e_2^1 \dots e_{k_1}^1 v_{k_1}^1, \dots, C^{n-1} = v^{n-1} e_1^{n-1} v_1^{n-1} e_2^{n-1} \dots e_{k_{n-1}}^{n-1} v_{k_{n-1}}^{n-1}$$

с упорядоченным охватыванием, покрывающая граф  $G$  и такая, что

$$(\forall m: m < n) \left( \bigcup_{l=0}^{m-1} \text{Int}(C^l) \right) \cap \left( \bigcup_{l=m}^{n-1} (C^l) \right) = \emptyset$$

является **покрытием с упорядоченным охватыванием**.

Построение покрытия графа с упорядоченным охватыванием решает поставленную задачу раскроя. Наибольший интерес представляют

покрытия с минимальным числом цепей, поскольку переход от одной цепи к другой соответствует холостому проходу режущего инструмента.

Минимальную по мощности последовательность реберно-непересекающихся цепей с упорядоченным охватыванием в плоском графе  $G$  будем называть **эйлеровым покрытием с упорядоченным охватыванием**.

Существование эйлеровых циклов с упорядоченным охватыванием в плоских эйлеровых графах доказано в работах [4, 5]. Рекурсивные алгоритмы построения таких циклов представлены в работе [4]. Проблеме построения маршрута с упорядоченным охватыванием в плоском графе произвольного вида посвящена работа [6], в которой разработан алгоритм построения таких маршрутов, имеющий вычислительную сложность не более  $O(|E|^2)$ . В работе [7] предложен эффективный алгоритм построения циклов с упорядоченным охватыванием в плоских эйлеровых графах, имеющий вычислительную сложность  $O(|E| \cdot \log_2 |V|)$ . Построению покрытий плоских связных графов последовательностью цепей с упорядоченным охватыванием посвящены статьи [2, 6]. Алгоритм нахождения допустимой цепи, покрывающей ребра многосвязного графа предложен в [3].

Рассмотрим результат, доказанный в [2] и дающий возможность построения последовательности цепей с упорядоченным охватыванием с любым наперед заданным множеством дополнительных ребер  $M$ , образующим паросочетание на множестве вершин нечетной степени  $V_{\text{odd}}$ .

**Теорема.** Пусть  $G = (V, E)$  топологический плоский граф на плоскости  $S$ , не имеющий висячих вершин. Для любого множества  $M$ , являющегося паросочетанием на множестве  $V_{\text{odd}}$  (состоящего из  $2n$  вершин нечетной степени графа  $G$ ), и такого, что  $M : (M \cap S) \setminus V = \emptyset$  существует эйлеров цикл  $C = v_1 e_1 v_2 e_2 \dots e_n v_1$ ,  $n = |E| + |M|$ , для любой начальной части которого  $C = v_1 e_1 v_2 e_2 \dots v_l$ ,  $l \leq |E| + |M|$ , выполнено условие  $\text{Int}(C_l) \cap E = \emptyset$ .

**Доказательство** теоремы конструктивно, в ней предложен алгоритм M-Cover для построения искомого покрытия.

При описании алгоритма используется понятие **уровня вложенности ребра**  $e : kmark(e)$ .

Уровнем вложенности ребра  $e$  плоского топологического графа  $G(V, E)$  будем называть значение функции  $kmark(e)$  определяемой рекурсивно:

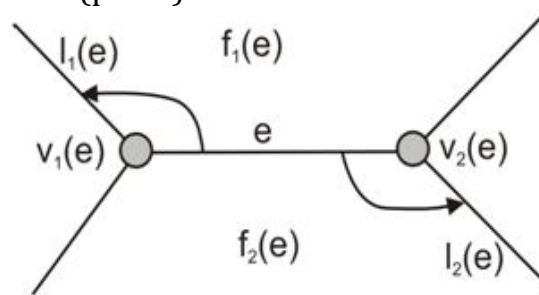
- все ребра, инцидентные внешней грани  $f_0$  графа  $G(V, E)$  образуют множество ребер  $E_1 = \{e \in E : e \subset f_0\}$  с уровнем вложенности  $(\forall e \in E_1)(kmark(e) = 1)$ ;
- ребра с уровнем вложенности 1 в графе

$$G_k \left( V, E \setminus \left( \bigcup_{l=1}^{k-1} E_l \right) \right), \quad k \geq 2$$

составляют множество  $E_k$  ребер с уровнем вложенности  $k$  в исходном графе  $G$ , т. е.  $(\forall e \in E_k)(kmark(e) = k)$ .

С точностью до гомеоморфизма представление любого плоского

графа  $G = (V, E)$  возможно с помощью задания для каждого ребра  $e$  функций  $v_1(e)$ ,  $v_2(e)$ ,  $l_1(e)$ ,  $l_2(e)$ ,  $f_1(e)$ ,  $f_2(e)$  (рис.1).



**Рис.1.** Функции, кодирующие ребра графа

Функции  $v_k(e)$ ,  $k=1,2$  представляют вершины, инцидентные ребру  $e$ ,  $f_k(e)$ ,  $k=1,2$  – грани, которые находятся слева при движении по ребру  $e$  от вершины  $v_k(e)$  к  $v_{3-k}(e)$ , и  $l_k(e)$ ,  $k=1,2$  – ребра, инцидентные граням  $f_k(e)$ .

Более того, алгоритм M-Cover использует две функции из алгоритма OrderedEnclosingCover, отвечающего за построение цепи с упорядоченным охватыванием.

Рассмотренный в работе [6] алгоритм OrderedEnclosingCover состоит из трех основных частей: инициализация, упорядочение и формирование.

На этапе **инициализации** необходимо определить начальные значения всех переменных (найти начальные вершину и ребро, все ребра считать непомеченными, а все списки – пустыми).

Функциональное назначение этапа **упорядочения** состоит в:

- определении значения  $kmark(e)$  для каждого ребра;
- формировании для каждой вершины списка инцидентных ребер, упорядоченных по убыванию значения  $kmark(e)$ .

После сортировке ребер по убыванию значения  $kmark(e)$  осуществляется переход к третьему этапу работы алгоритма – **формированию**, где осуществляется:

- выбор вершины нечетной степени с максимальным значением  $kmark(e)$ ;
- построение цепи, начинающейся в этой вершине, и заканчивающуюся в другой вершине нечетной степени;
- удаление концевых вершин цепи из списка вершин нечетной степени.

Если после выполнения этапа формирования остались ребра, не принадлежащие ни одной из цепей (ребра, ограничивающие внешнюю грань) – построить цикл, состоящий из этих ребер.

Приведем алгоритм M-Cover построения допустимого эйлера покрытия графа цепями с упорядоченным охватыванием [2].

### **Алгоритм M-Cover**

#### **Входные данные:**

- плоский граф  $G$ ;
- паросочетание  $M$  на множестве вершин нечетной степени  $V_{odd}$ .

**Выходные данные:**  $C_j, j=1, \dots, V_{\text{odd}}/2$  – эйлерово покрытие графа  $G$  цепями с упорядоченным охватыванием.

**Шаг 1.** Выполнить процедуры **Инициализация** и **Упорядочение** (т.е. Шаг 1 и Шаг 2 алгоритма `OderedEnclosingCover` [5]). Положить  $j=1$ . Объявить вершину  $v_0$ , смежную внешней грани, текущей, положить  $v_0^j = v_0$ .

**Шаг 2.** Выполнять построение цепи  $C_j$  с помощью процедуры **Формирование** [5], используя вершину  $v_0$  в качестве начальной. Пусть вершина  $v_0$  является конечной вершиной построенной цепи. Если  $v_0 \in W_{\text{odd}}$ , то перейти на шаг 6; иначе перейти на шаг 3.

**Шаг 3.** Если вершина  $v_0$  является тупиковой, то перейти на шаг 5, иначе перейти на шаг 4.

**Шаг 4.** Если  $\text{mark}(v_1) < \text{mark}(v_0)$ , то положить  $v_0 = v_1$  и перейти на шаг 2 (продолжить построение цепи  $C_j$  из текущей вершины).

**Шаг 5.** Найти  $v_1: (v_0, v_1) \in M$ . Закончить построение текущей цепи:  $v_1^j = v_0$ ,  $j = j+1$ ,  $V_{\text{odd}} = V_{\text{odd}} \setminus \{v_0, v_1\}$ ,  $M = M \setminus \{(v_0, v_1)\}$  принять  $v_0^j = v_1$  за текущую вершину следующей цепи и перейти на шаг 2 (начать построение новой цепи  $C_j$  из вершины  $v_0 = v_0^j$ ).

**Шаг 6.** Останов.

#### **Конец алгоритма M-Cover**

Вычислительная сложность алгоритма M-Cover не превосходит величины  $O(|E| \cdot \log_2 |V|)$ .

Для построения оптимального покрытия достаточно в качестве  $M$  взять кратчайшее паросочетание на множестве  $V_{\text{odd}}$  [2].

#### **Алгоритм OptimalCover**

Входные данные:

плоский граф  $G$ , представленный списком ребер с заданными на них функциями  $v_k(e), l_k(e), f_k(e), k=1, 2$ .

**Выходные данные:**  $C_j, j=1, \dots, |V_{\text{odd}}|/2$ , – покрытие графа  $G$  цепями с упорядоченным охватыванием.

**Шаг 1.** Найти кратчайшее паросочетание  $M$  на множестве  $V_{\text{odd}}$ .

**Шаг 2.** Выполнить алгоритм M-Cover для графа  $G$  и паросочетания  $M$ .

**Шаг 3.** Останов.

#### **Конец алгоритма OptimalCover**

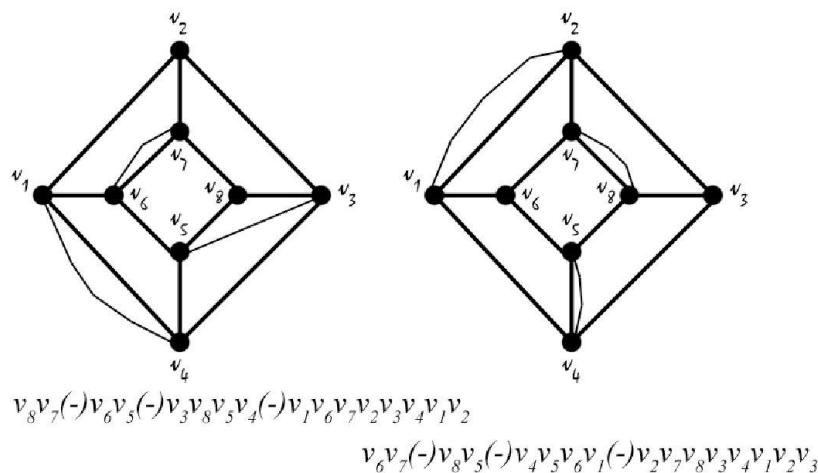
Сложность алгоритма OptimalCover не превосходит величины  $O(|V|^3)$  (т.е. сложности построения паросочетания).

Пример сравнения результатов работы алгоритмов `OderedEnclosingCover` и M-Cover приведен на рис.2.

В случае многосвязного графа можно применить следующий допустимый алгоритм [3].

В алгоритме будет использовано понятие **уровня вложенности компоненты связности**. Под уровнем вложенности компоненты

связности будем понимать минимальный уровень вложенности ребер этой компоненты связности.



**Рис.2.** Результат работы алгоритма *OrderedEnclosingCover* (слева) и *M-Cover* (справа)

### Алгоритм **MultiComponentCover**

**Входные данные:** плоский граф  $G$ .

**Выходные данные:**  $C_j^s$ ,  $j=1, \dots, |V_{\text{odd}}|/2$ , – покрытие графа  $G$  цепями с упорядоченным охватыванием,  $s=1, 2, \dots$  – номер компоненты связности.

**Шаг 1.** Выявить множество  $S$  всех компонент связности графа  $G$  и  $\forall s \in S$  найти уровень вложенности  $K(s)$ .

**Шаг 2.** Упорядочить элементы множества  $S$  в порядке убывания значений  $K(s)$ .

**Шаг 3.** Пусть  $i(s)$  – порядковый номер компоненты связности  $s$ , а  $s(i)$  – компонента связности с порядковым номером  $i$ . Для  $1 \leq i \leq |S|$  выполнить алгоритм *OptimalCover* и получить  $C_j^i$  – покрытие  $i$ -ой компоненты связности цепями с упорядоченным охватыванием.

**Шаг 4.** Останов.

**Конец алгоритма **MultiComponentCover****

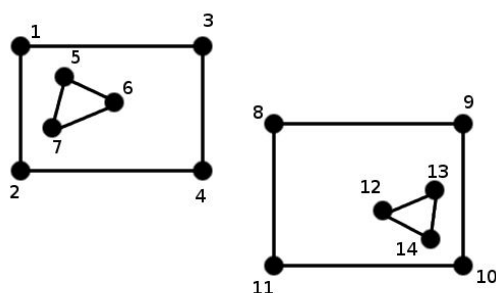
Очевидно, что вычислительная сложность данного алгоритма составляет величину, не превышающую  $O(|V|^2)$ .

Такой алгоритм позволяет построить только допустимое покрытие с указанными свойствами. Что касается оптимальной длины дополнительных построений, то в данном случае компоненты связности можно упорядочить не только по уровням вложенности. Например, их можно подразделить на так называемые «вложенные объединения».

**Вложенным объединением**  $n$  будем называть семейство компонент связности  $S_n$  плоского графа, в котором компонента связности с уровнем вложенности  $k$  содержит в себе (охватывает) компоненты связности с уровнем вложенности больше  $k$ .

Например, на рис.3 изображено четыре компоненты связности,

образующих два вложенных объединения.



**Рис. 3.** Пример вложенных объединений

Введем ограничение: невозможно переключиться на построение покрытия в другом вложенном объединении, пока имеются еще не пройденные ребра в текущем.

Если модифицировать алгоритм MultiComponentCover с учетом введенных замечаний, будет получен рекурсивный алгоритм для обхода всех вложенных компонент связности, фиктивные ребра с оптимальной длиной между которыми будет невозможно построить за полиномиальное время, т.к. задача сведется к поиску решения задачи коммивояжера.

Однако для задачи построения оптимального покрытия с упорядоченным охватыванием в плоском графе возможно построить нерекурсивный алгоритм, который находит решение за полиномиальное время.

### **Алгоритм OptimalMultiComponent**

**Входные данные:** плоский граф  $G$ .

**Выходные данные:**  $C_j^s$ ,  $j=1, \dots, |V_{\text{odd}}|/2$ , – покрытие графа  $G$  цепями с упорядоченным охватыванием,  $s=1, 2, \dots$  – номер компоненты связности.

**Шаг 1.** Выявить множество  $S$  всех компонент связности графа  $G$ .

**Шаг 2.** Построить полный абстрактный граф  $\mathcal{B}$ , вершинами которого являются компоненты связности  $S$  графа  $G$ , а длины ребер равны расстоянию между ближайшими вершинами компонент связности.

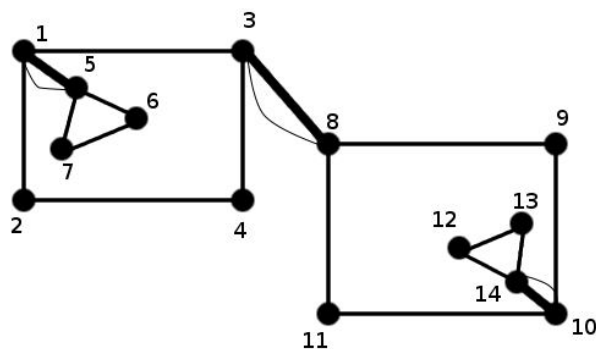
**Шаг 3.** Найти остовное дерево минимального веса  $T(\mathcal{B})$  в  $\mathcal{B}$ .

**Шаг 4.** Добавить ребра найденного остовного дерева в граф  $G$ :  $G_{\mathcal{B}} = G \cup T(\mathcal{B})$ .

Например, для графа, приведенного на рис.3, будут добавлены ребра  $\{1;5\}$ ,  $\{3;8\}$  и  $\{10;14\}$  (рис.4).

**Шаг 5.** Выполнить алгоритм OptimalCover для графа  $G_{\mathcal{B}}$  (например, на рис.4 представлены дополнительные построения, выполненные алгоритмом OptimalCover для рассмотренного графа).

**Конец алгоритма OptimalMultiComponent**



**Рис. 4.** Добавление ребер, соединяющих компоненты связности, и ребер, делающих граф эйлеровым

Предложенный алгоритм строит покрытие цепями с упорядоченным охватыванием за полиномиальное время.

Для графа, приведенного в качестве примера, таким покрытием будет следующим (приведена последовательность цепей, состоящих из ребер исходного графа, в порядке их обхода, здесь верхний индекс – номер компоненты связности, нижний индекс – номер цепи для данной компоненты связности):

$$C_1^1 = \{5, 6, 7, 5\}, C_1^2 = \{1, 3\}, C_1^3 = \{8, 9, 10\}, C_1^4 = \{14, 13, 12, 14\}, \\ C_2^1 = \{10, 11, 8\}, C_2^2 = \{3, 4, 2\}.$$

Шаг 1 алгоритма представляет собой волновой алгоритм, который решает свою задачу за время  $O(|V|^2)$ . Действия, выполняемые на шаге 2, могут быть выполнены также за время  $O(|V|^2)$ . Известно, что решение задачи на шаге 3 алгоритма может быть получено также за полиномиальное время  $O(|V|^2)$  либо при использовании логарифмических сортировок эту величину можно уменьшить до  $O(|V| \log_2 |V|)$ . Шаг 4 имеет сложность  $O(1)$ . Выше упоминалось, что сложность алгоритма **OptimalCover** (шаг 5) не превосходит величины  $O(|V|^3)$ . Таким образом, суммарная сложность алгоритма **OptimalMultiComponent** представляет величину  $O(|V|^3)$ .

Алгоритм **OptimalMultiComponent** позволяет получить не большую длину дополнительных построений, по сравнению с алгоритмом **MultiComponentCover**. Это объясняется тем, что в данном алгоритме переход осуществляется к ближайшей возможной компоненте связности, а не к ближайшей возможной компоненте связности максимального уровня вложенности.

*Работа поддержана грантом РФФИ 10-07-96002-р\_урал\_а.*

## Литература

1. Panyukova, T. Cover with Ordered Enclosing for Flat Graphs // Electronic Notes in Discrete Mathematics, 2007. No. 28. P.17–24.



2. Панюкова, Т.А. Оптимальные эйлеровы покрытия для плоских графов // Дискретный анализ и исследование операций. 2011. Т.18, №2. С. 64–74.
3. Панюкова Т.А. Эйлерово покрытие с упорядоченным охватыванием для многосвязного графа. // Материалы 3-й международной конференции «Математическое моделирование, оптимизация и информационные технологии». – Кишнев: Evrica, 2012. С.429–438.
4. Panioukova T.A. Algorithms for Construction of Ordered Enclosing Traces in Planar Eulerian Graphs. / Panioukova T.A., Panyukov A.V. // The International Workshop on Computer Science and Information Technologies' 2003, Proceedings of Workshop, Ufa, September 16–18, 2003. Volume 1, Ufa State Technical University, 2003. P. 134–138.
5. Панюкова Т.А. Построение маршрутов с упорядоченным охватыванием в плоских графах // Труды 36-й Региональной молодежной конференции «Проблемы теоретической и прикладной математики». Екатеринбург: УрО РАН, 2005. С. 61–66.
6. Панюкова Т.А. Обходы с упорядоченным охватыванием в плоских графах / Панюкова Т.А. // Дискретный анализ и исследование операций, 2006. Сер. 2. Т.13, №2. С. 31–43.
7. Panyukov, A.V. The Algorithm for Tracing of Flat Euler Cycles with Ordered Enclosing / A.V. Panyukov, T.A. Panioukova // Proceedings of Chelyabinsk Scientific Center, 2000. – \#4(9). P.18–22.

### ***Расстояние между лингвистическими шкалами***

*Неопределенность* – неотъемлемая составляющая процесса принятия решений, управления, прогнозирования. Ее уровень обуславливает степень структурированности прикладной задачи [1]. Интерпретация неопределенности может быть различной (рис.1).

*Лингвистическая неопределенность* – неопределенность, возникающая из-за расплывчатости и неоднозначности словесных выражений. Данная неопределенность возникает в тех случаях, когда в процедуре оценивания участвуют эксперты, которые вынуждены оперировать конечным числом слов и составленных из них фраз определенной структуры для описания характеристик альтернатив. Можно выделить такие типы лингвистической неопределенности, как *неопределенность смысла фраз* и *неопределенность значений слов*. Первый тип неопределенности учитывается с помощью теории формальных грамматик, а второй – теорией нечетких множеств. Нечеткость информации обусловлена наличием понятий и отношений с нестрогими границами, а также высказываний с многозначной шкалой истинности. Она имеет следующие проявления:

- нечеткость как следствие субъективности эксперта;
- нечеткость как неясность в процессах мышления и умозаключения;
- нечеткость описания или представления на естественном языке или максимально приближенному к нему.

Теория нечетких множеств предоставляет схему решения проблем, в которых субъективное суждение или оценка играют центральную и значительную роль при учете факторов неясности и нечеткости.

Лингвистическое оценивание является наиболее адекватным методом получения экспертных оценок в случае, когда показатели имеют качественную природу. Данный подход предполагает, что в рамках оценочной процедуры каждый эксперт использует индивидуальную лингвистическую шкалу, мощность которой зависит от его способности различать градации неопределенности при формировании качественных оценок альтернатив. Проблема формирования согласованного группового решения в данном случае может быть решена путем унификации – процедуры, предполагающей переход к единой *универсальной лингвистической шкале*, выбор которой осуществляется неоднозначно [2, 3]. Естественным требованием к универсальной лингвистической шкале

является минимальная рассогласованность с индивидуальными лингвистическими шкалами экспертов [4].

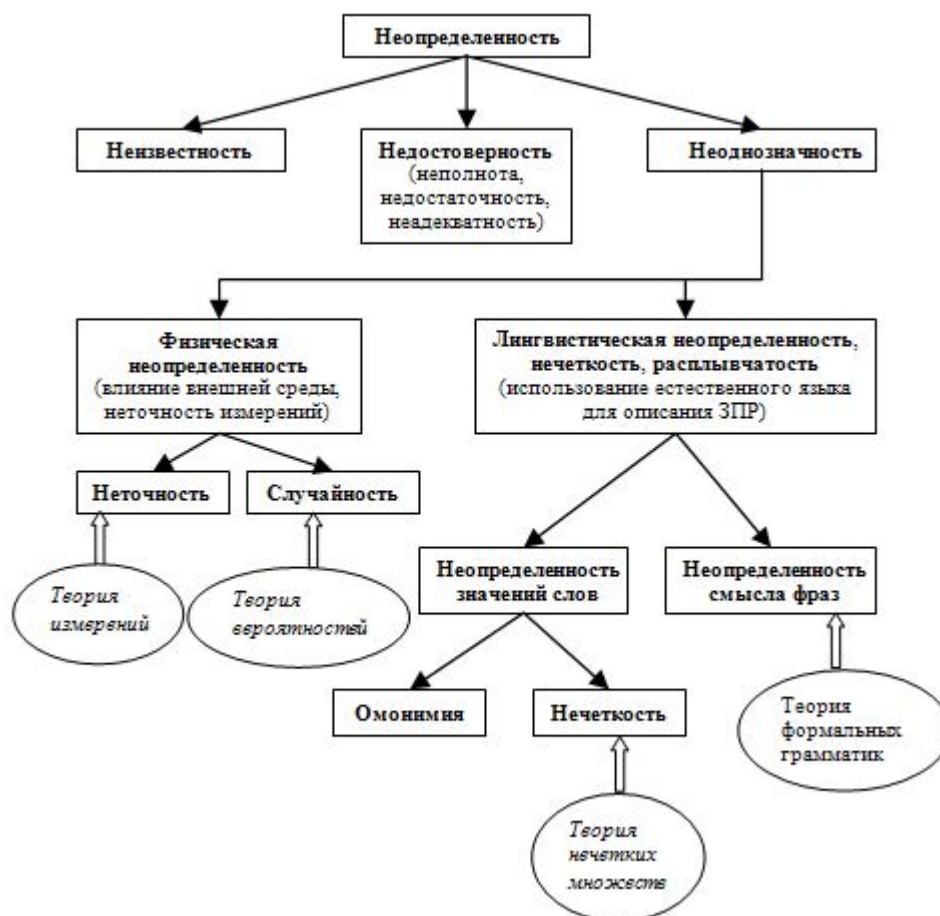


Рис. 1. Интерпретация неопределенности

При разработке алгоритма для вычисления степени рассогласованности лингвистической шкалы возникает необходимость определения расстояния между индивидуальными лингвистическими шкалами. На данный момент не разработан математический аппарат для расчета расстояния между лингвистическими шкалами, что обуславливает актуальность выбранной темы работы.

### **Основные понятия для представления лингвистической информации**

Пусть  $U$  - некоторое множество элементов (универсальное множество),  $u$  - элемент множества  $U$ , и  $\mu_A: U \rightarrow [0,1]$ .

**Определение 1.** Нечетким множеством  $A$  множества  $U$  называется множество упорядоченных пар

$$A = \{(u, \mu_A(u)) \mid u \in U\}$$

где  $\mu_A(u)$  - функция принадлежности, определяющая степень принадлежности с которой элемент  $u$  принадлежит нечеткому подмножеству  $A$  [5].

К функции принадлежности нечеткой величины  $A$  предъявляется ряд требований. Среди наиболее существенных – следующие:

- $\mu_A(u)$  непрерывна (т.е. в результате небольшого изменения значения аргумента значение функции также мало изменится);
- функция принадлежности нечеткой величины является нормальной, т.е.  $\sup_{x \in R} \{\mu_A(x)\} = 1$  (это означает, что существует хотя бы один элемент  $x \in R$ , для которого свойство, соответствующее нечеткому множеству  $A$ , выполнено в полной мере);
- $\mu_A(u)$  выпукла, т.е.

$$\forall x_1, x_2 \ (x_1 < x_2) \ \forall \gamma \in [0, 1]$$

$$\mu_A(\gamma \cdot x_1 + (1 - \gamma) \cdot x_2) \geq \min\{\mu_A(x_1), \mu_A(x_2)\}.$$

Понятие нечеткого множества обеспечивает возможность математического представления качественных оценок, выражаемых людьми в форме лингвистических значений и нечетких чисел, в основе которых лежит понятие нечеткой переменной.

*Определение 2. Нечеткая переменная задается тройкой*

$$\langle \alpha, U, A \rangle,$$

где  $\alpha$  – наименование нечеткой переменной;  $U$  – область определения (универсальное множество);  $A$  – нечеткое множество на  $U$  с функцией принадлежности  $\mu_A(u)$  ( $\mu_A: U \rightarrow [0, 1]$  – определяет степень принадлежности элемента  $x$  к множеству  $A$  или, иными словами, степень выполнения свойства  $A$  для элементов из  $U$ ), описывающее ограничения на возможные значения нечеткой переменной  $\alpha$  (её семантику).

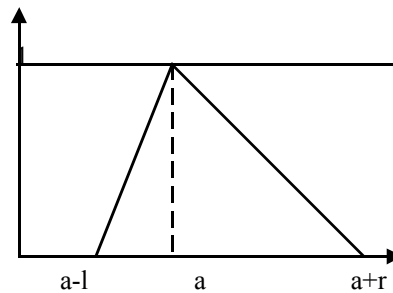
*Определение 3. Нечеткая величина* – это нечеткая переменная, определенная на множестве действительных чисел  $R$ . Нечеткую величину и соответствующее ей нечеткое множество обозначают одной и той же переменной.

*Определение 4. Нечетким числом  $A$  называется нечеткая величина  $A$ , функция принадлежности  $\mu_A(u)$  которой является выпуклой и унимодальной на  $R$ . Нечеткое число с модальным значением  $a$  можно рассмотреть как нечеткое значение высказывания « $x$  приблизительно равно  $a$ ».*

Частные случаи нечеткого числа, которые получили свое название по виду их функции принадлежности.

*Определение 5. Треугольным нечетким числом (рис. 2) с центром в точке  $a$ , левой шириной  $l > 0$  и правой шириной  $r > 0$  называется нечеткое множество  $A$  с треугольной функцией принадлежности  $\mu_A(u)$ :*

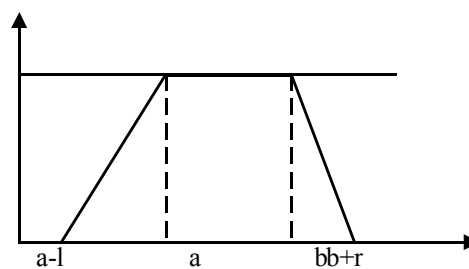
$$\mu_A(x) = \begin{cases} 1 - \frac{a-x}{l}, & \text{если } a-l \leq x < a; \\ 1, & \text{если } x = a; \\ 1 - \frac{x-a}{r}, & \text{если } a < x \leq a+r; \\ 0, & \text{иначе.} \end{cases}$$



**Рис. 2.** Функция принадлежности треугольного нечеткого числа  $A$

**Определение 6.** Трапецевидным нечетким числом (рис. 3) с отрезком толерантности  $[a, b]$ , левой шириной  $l$  и правой шириной  $r$  называется нечеткое множество  $A$  с трапецевидной функцией принадлежности вида:

$$\mu_A(x) = \begin{cases} 1 - \frac{a-x}{l}, & \text{если } a-l \leq x < a; \\ 1, & \text{если } a \leq x \leq b; \\ 1 - \frac{x-b}{r}, & \text{если } b < x \leq b+r; \\ 0, & \text{иначе.} \end{cases}$$



**Рис. 3.** Функция принадлежности трапецевидного нечеткого числа  $A$

**Определение 7.** Лингвистическая переменная задается кортежем [6]

$$\langle \beta, Term, U, G, M \rangle,$$

где  $\beta$  – название переменной;  $Term = \{t_k\}_{k=1,2,\dots}$  – терм-множество или множество значений переменной  $\beta$ , причем каждое из таких значений является нечеткой переменной  $t_k$ , заданной на универсальном множестве  $U$

числовой или нечисловой природы;  $G$  – синтаксическое правило, порождающее новые названия значений лингвистической переменной  $\beta$ ;  $M$  – семантическое правило, которое ставит в соответствие каждой нечеткой переменной ее смысл, т.е. нечеткое подмножество универсального множества  $U$ .

**Определение 8. Семантическим пространством** называется лингвистическая переменная с фиксированным терм-множеством, т.е. четверка

$$S = \langle \beta, Term, U, M \rangle.$$

Иными словами, семантическое пространство – это набор нечетких переменных.

**Определение 9. Лингвистической шкалой**  $S$  называется конечное линейно упорядоченное множество термов  $S = \{s_i\}_{i=1, \dots, T}$ , удовлетворяющим следующим условиям:

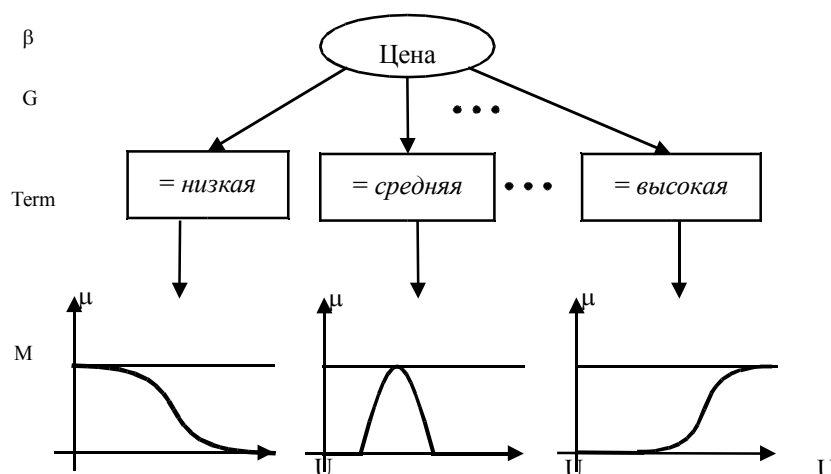
- если  $i < j$ , то  $s_i$  предшествует  $s_j$  ( $s_i \prec s_j$ );
- отрицание терма определяется правилом  $Neg(s_i) = s_{T-i}$ ;
- пусть  $s_i \prec s_j$ , тогда объединение (дизъюнкция, связка «или») термов определяется правилом

$$s_i \vee s_j = \max\{s_i, s_j\} = s_j;$$

- пусть  $s_i \prec s_j$ , тогда пересечение (конъюнкция, связка «и») термов определяется правилом

$$s_i \wedge s_j = \min\{s_i, s_j\} = s_i.$$

На рис. 4 представлено терм-множество лингвистической переменной *Цена*.



**Рис. 4.** Терм-множество лингвистической переменной *Цена*

### **Расстояние между лингвистическими шкалами**

Так как термы лингвистической шкалы  $s_i \in S$ ,  $i = \overline{1, T}$  являются нечеткими переменными, т.е. по сути, каждому терму ставится в соответствие функция принадлежности  $\mu_{s_i} : U \rightarrow [0, 1]$ , то для определения

близости термов лингвистической шкалы можно использовать известные функции расстояния для соответствующих нечетких множеств.

**Определение 10.** Под расстоянием между термами лингвистической шкалы  $s_i, s_j \in S$  понимается функция  $d: S \times S \rightarrow R^+$ , такая, что

1.  $d(s_i, s_j) \geq 0$  и  $d(s_i, s_j) = 0 \Leftrightarrow s_i = s_j$ , т.е.  $\mu_{s_i}(u) = \mu_{s_j}(u) \quad \forall u \in U$ ;
2.  $d(s_i, s_j) = d(s_j, s_i)$ ;
3.  $d(s_i, s_j) \leq d(s_j, s_k) * d(s_k, s_i)$ ,

где  $*$  - некоторая операция, связанная с функцией расстояния и обеспечивающая выполнение свойства транзитивности.

Широко распространенные функции расстояния представлены в следующей табл. 1 [7, 2].

**Табл. 1. Примеры функций расстояния**

Название расстояния	Тип множества $U$	Формула расстояния
Расстояние Хемминга	Конечное универсальное множество $U$ ( $ U =n$ )	$\rho(s_i, s_j) = \sum_{k=1}^n  \mu_{s_i}(u_k) - \mu_{s_j}(u_k) $
Относительное расстояние Хемминга	Конечное универсальное множество $U$ ( $ U =n$ )	$\tilde{\rho}(s_i, s_j) = \frac{1}{n} \sum_{k=1}^n  \mu_{s_i}(u_k) - \mu_{s_j}(u_k) $
Расстояние Хемминга	Бесконечное универсальное множество $U \subseteq R^1$	$\rho(s_i, s_j) = \int_U  \mu_{s_i}(u) - \mu_{s_j}(u)  du$
Относительное расстояние Хемминга	Бесконечное универсальное множество $U \subseteq R^1$	$\tilde{\rho}(s_i, s_j) = \frac{1}{ U } \int_U  \mu_{s_i}(u) - \mu_{s_j}(u)  du$
Расстояние Евклида	Конечное универсальное множество $U$ ( $ U =n$ )	$\rho_e(s_i, s_j) = \sqrt{\sum_{k=1}^n (\mu_{s_i}(u_k) - \mu_{s_j}(u_k))^2}$
Относительное расстояние Евклида	Конечное универсальное множество $U$ ( $ U =n$ )	$\tilde{\rho}_e(s_i, s_j) = \frac{1}{\sqrt{n}} \sqrt{\sum_{k=1}^n (\mu_{s_i}(u_k) - \mu_{s_j}(u_k))^2}$
Расстояние Евклида	Бесконечное универсальное множество $U \subseteq R^1$	$\rho_e(s_i, s_j) = \sqrt{\int_U (\mu_{s_i}(u) - \mu_{s_j}(u))^2 du}$
Относительное расстояние Евклида	Бесконечное универсальное множество $U \subseteq R^1$	$\tilde{\rho}_e(s_i, s_j) = \frac{1}{\sqrt{ U }} \sqrt{\int_U (\mu_{s_i}(u) - \mu_{s_j}(u))^2 du}$

Выбор того или иного расстояния зависит от природы рассматриваемой проблемы. Для нечетких множеств можно использовать и другие функции расстояния.

**Определение 11.** Пусть  $U$  - универсальное множество. Будем называть лингвистическим пространством совокупность лингвистических шкал

$$S = (S^1, \dots, S^k),$$

где  $S^j = \{s_l^j\}_{l=\overline{1, n_j}}$  ( $j = \overline{1, k}$ ) - лингвистическая шкала, с мощностью  $|S^j| = n_j$ , при этом каждый терм лингвистической шкалы есть нечеткая переменная  $\langle s_l^j, \mu_{s_l^j}, U \rangle$ .

Таким образом, лингвистическое пространство содержит набор лингвистических шкал, которые можно построить на универсальном множестве  $U$ . В рамках данного пространства можно ввести понятие расстояния между лингвистическими шкалами.

**Лемма 1.** Пусть  $S^i, S^j \in \mathcal{S}$ . Тогда

$$d(S^i, S^j) = \frac{\int_{u \in U} \left| \max_{p=\overline{1, n_i}} \mu_p^i(u) - \max_{l=\overline{1, n_j}} \mu_l^j(u) \right| du}{\int_{u \in U} \max_{k=\overline{1, m}; g=\overline{1, n_i}} \{\mu_g^k(u)\} du} \quad (1)$$

является метрикой в пространстве  $\mathcal{S}$ ,  $d: \mathcal{S} \times \mathcal{S} \rightarrow [0, 1]$ .

*Доказательство.*

Для доказательства леммы необходимо проверить выполнение аксиом расстояния.

*Аксиома 1.*

$$d(S^i, S^i) = \frac{\int_{u \in U} \left| \max_{p=\overline{1, n_i}} \mu_p^i(u) - \max_{l=\overline{1, n_i}} \mu_l^i(u) \right| du}{\int_{u \in U} \max_{k=\overline{1, m}; g=\overline{1, n_i}} \{\mu_g^k(u)\} du} = \frac{\int_{u \in U} |0| du}{\int_{u \in U} \max_{k=\overline{1, m}; g=\overline{1, n_i}} \{\mu_g^k(u)\} du} = 0.$$

*Аксиома 2.* Свойство симметричности

$$\begin{aligned} d(S^i, S^j) &= \frac{\int_{u \in U} \left| \max_{p=\overline{1, n_i}} \mu_p^i(u) - \max_{l=\overline{1, n_j}} \mu_l^j(u) \right| du}{\int_{u \in U} \max_{k=\overline{1, m}; g=\overline{1, n_i}} \{\mu_g^k(u)\} du} = \frac{\int_{u \in U} \left| \max_{l=\overline{1, n_j}} \mu_l^j(u) - \max_{p=\overline{1, n_i}} \mu_p^i(u) \right| du}{\int_{u \in U} \max_{k=\overline{1, m}; g=\overline{1, n_j}} \{\mu_g^k(u)\} du} = \\ &= d(S^j, S^i) \end{aligned}$$

*Аксиома 3.* Свойство транзитивности



$$\begin{aligned}
d(S^i, S^j) + d(S^j, S^h) &= \frac{\int_U \left| \max_{p=1, n_j} \mu_p^i(u) - \max_{l=1, n_j} \mu_l^j(u) \right| du}{\int_U \max_{k=1, m; g=1, n_k} \{ \mu_g^k(u) \} du} + \\
&+ \frac{\int_U \left| \max_{l=1, n_j} \mu_l^j(u) - \max_{o=1, n_k} \mu_o^h(u) \right| du}{\int_U \max_{k=1, m; g=1, n_k} \{ \mu_g^k(u) \} du} \geq \\
&\geq \frac{\int_U \left( \left| \max_{p=1, n_j} \mu_p^i(u) - \max_{l=1, n_j} \mu_l^j(u) \right| + \left| \max_{l=1, n_j} \mu_l^j(u) - \max_{o=1, n_k} \mu_o^h(u) \right| \right) du}{\int_U \max_{k=1, m; g=1, n_k} \{ \mu_g^k(u) \} du} \geq \\
&\geq \frac{\int_U \left| \max_{p=1, n_j} \mu_p^i(u) - \max_{o=1, n_k} \mu_o^h(u) \right| du}{\int_U \max_{k=1, m; g=1, n_k} \{ \mu_g^k(u) \} du} = d(S^i, S^h).
\end{aligned}$$

Таким образом, предложенная формула удовлетворяет аксиомам расстояния и может быть использована в качестве метрики в пространстве  $\mathbb{S}$ .

Лемма доказана.

**Лемма 2.** Пусть  $S^i, S^j \in S, \rho$  - расстояние между нечеткими множествами, тогда

$$d(S^i, S^j) = \frac{\sum_{k=1}^{n_A} \min_{l=1, n_{\beta_2}} \rho(S_k^{\beta_1}; S_l^{\beta_2})}{n_A}, \quad (2)$$

где номера шкал определяются в виде

$$\beta_1 = \begin{cases} i, & \text{если } i < j \\ j, & \text{иначе} \end{cases}, \quad \beta_2 = \begin{cases} i, & \text{если } i > j \\ j, & \text{иначе} \end{cases},$$

является метрикой в пространстве  $S, d: S \times S \rightarrow [0, 1]$ .

Доказательство.

Доказательство леммы сводится к проверке выполнения аксиом, определяющих функцию расстояния.

*Аксиома 1.*

$$d(S^i, S^i) = \frac{\sum_{k=1}^n \min_{l=1, n} \rho(S_k^i; S_l^i)}{n_A} = \frac{\sum_{k=1}^n 0}{n_A} = 0$$

*Аксиома 2.* Свойство симметричности

$$d(S^i, S^j) = \frac{\sum_{k=1}^{n_A} \min_{l=1, n_{\rho_2}} \rho(s_k^{\beta_1}, s_l^{\beta_2})}{n_A} = \frac{\sum_{k=1}^{n_A} \min_{l=1, n_{\rho_2}} \rho(s_l^{\beta_2}, s_k^{\beta_1})}{n_A} = d(S^j, S^i),$$

$$\text{где } \beta_1 = \begin{cases} i, & \text{если } i < j \\ j, & \text{иначе} \end{cases}, \quad \beta_2 = \begin{cases} i, & \text{если } i > j \\ j, & \text{иначе} \end{cases},$$

**Аксиома 3. Свойство транзитивности**

Т.к.  $\rho$  - расстояние между нечеткими множествами, то справедливо неравенство:

$$\rho(s_k^i, s_l^j) + \rho(s_l^j, s_\delta^h) \geq \rho(s_k^i, s_\delta^h) \quad \forall k = \overline{1, n_i}, l = \overline{1, n_j}, \delta = \overline{1, n_h}$$

$$\forall q_1 = \overline{1, m_1}, q_2 = \overline{1, m_2}, q_3 = \overline{1, m_3}:$$

$$\min_{d_1=1, n_1} \rho(s_{d_1}^{\eta_1}, s_{q_1}^{m_1}) + \min_{d_2=1, n_2} \rho(s_{d_2}^{\eta_2}, s_{q_2}^{m_2}) \geq \min_{d_3=1, n_3} \rho(s_{d_3}^{\eta_3}, s_{q_3}^{m_3}),$$

где

$$\eta_1 = \begin{cases} i, & \text{если } i < j \\ j, & \text{иначе} \end{cases} \quad m_1 = \begin{cases} i, & \text{если } i > j \\ j, & \text{иначе} \end{cases}$$

$$\eta_2 = \begin{cases} j, & \text{если } j < h \\ h, & \text{иначе} \end{cases}, \quad m_2 = \begin{cases} j, & \text{если } j > h \\ h, & \text{иначе} \end{cases},$$

$$\eta_3 = \begin{cases} i, & \text{если } i < h \\ h, & \text{иначе} \end{cases} \quad m_3 = \begin{cases} i, & \text{если } i > h \\ h, & \text{иначе} \end{cases}$$

$$\begin{aligned} &\Rightarrow \sum_{q_1=1}^{n_1} \sum_{q_2=1}^{n_2} \sum_{q_3=1}^{n_3} \min_{d_1=1, n_1} \rho(s_{d_1}^{\eta_1}, s_{q_1}^{m_1}) + \sum_{q_1=1}^{n_1} \sum_{q_2=1}^{n_2} \sum_{q_3=1}^{n_3} \min_{d_2=1, n_2} \rho(s_{d_2}^{\eta_2}, s_{q_2}^{m_2}) \geq \\ &\geq \sum_{q_1=1}^{n_1} \sum_{q_2=1}^{n_2} \sum_{q_3=1}^{n_3} \min_{d_3=1, n_3} \rho(s_{d_3}^{\eta_3}, s_{q_3}^{m_3}) \end{aligned}$$

$$\begin{aligned} &\Rightarrow n_{m_1} n_{m_2} \sum_{q_1=1}^{n_1} \min_{d_1=1, n_1} \rho(s_{d_1}^{\eta_1}, s_{q_1}^{m_1}) + n_{m_1} n_{m_2} \sum_{q_2=1}^{n_2} \min_{d_2=1, n_2} \rho(s_{d_2}^{\eta_2}, s_{q_2}^{m_2}) \geq \\ &\geq n_{m_1} n_{m_2} \sum_{q_3=1}^{n_3} \min_{d_3=1, n_3} \rho(s_{d_3}^{\eta_3}, s_{q_3}^{m_3}) \quad | : (n_{m_1} \cdot n_{m_2} \cdot n_{m_3}) \end{aligned}$$

$$\begin{aligned} &\Rightarrow \frac{\sum_{q_1=1}^{n_1} \min_{d_1=1, n_1} \rho(s_{d_1}^{\eta_1}, s_{q_1}^{m_1})}{n_{m_1}} + \frac{\sum_{q_2=1}^{n_2} \min_{d_2=1, n_2} \rho(s_{d_2}^{\eta_2}, s_{q_2}^{m_2})}{n_{m_2}} \geq \frac{\sum_{q_3=1}^{n_3} \min_{d_3=1, n_3} \rho(s_{d_3}^{\eta_3}, s_{q_3}^{m_3})}{n_{m_3}} \\ &\Rightarrow d(S^i, S^j) + d(S^j, S^h) \geq d(S^i, S^h). \end{aligned}$$

Таким образом, предложенная формула удовлетворяет аксиомам

расстояния и может быть использована в качестве метрики в пространстве  $\mathbb{S}$ .

Лемма доказана.

Очевидно, что можно определять и другие метрики, удовлетворяющие аксиомам расстояния.

**Пример.** Вычислить расстояние между индивидуальными лингвистическими шкалами  $S^1$ ,  $S^2$  и  $S^3$  экспертов  $E = \{e_1, e_2, e_3\}$ . Лингвистические шкалы экспертов представлены в табл. 2.

**Табл. 2.** Индивидуальные лингвистические шкалы экспертов

Эксперт	Индивидуальная шкала	Определение термов
$e_1$	$S^1 = \{s_i^1\}_{i=\overline{1,5}}$	$s_1^1 \rightarrow (l_1^1, a_1^1, r_1^1) = (0,0,0.3)$ , $s_2^1 = (0.3,0.3,0.2)$ , $s_3^1 = (0.2,0.5,0.3)$ , $s_4^1 = (0.3,0.8,0.2)$ , $s_5^1 = (0.2,1,0.2)$
$e_2$	$S^2 = \{s_i^2\}_{i=\overline{1,4}}$	$s_1^2 \rightarrow (0,0,0.4)$ , $s_2^2 \rightarrow (0.4,0.4,0.2)$ , $s_3^2 \rightarrow (0.2,0.6,0.2)$ , $s_4^2 \rightarrow (0.2,0.8,0.2)$
$e_3$	$S^3 = \{s_i^3\}_{i=\overline{1,3}}$	$s_1^3 = (0,0,0.4)$ , $s_2^3 = (0.4,0.4,0.4)$ , $s_3^3 = (0.4,0.8,0.2)$

Результат вычислений расстояний между лингвистическими шкалами по формуле (2) с использованием расстояния Хемминга для термов шкалы представлен в табл.3.

**Табл. 3.** Расстояние между шкалами

$d(S^i, S^j)$	$S^1$	$S^2$	$S^3$
$S^1$	-	0,113	0,171
$S^2$	0,113	-	0,108
$S^3$	0,171	0,108	-

Получили, что  $d(S^3, S^2) \cup d(S^1, S^2) \cup d(S^1, S^3)$ . Данные значения показывают, что, лингвистическая шкала  $S^1$  ближе к шкале  $S^2$ , чем к шкале  $S^3$ , т.е. степень рассогласованности шкалы  $S^1$  со шкалой  $S^2$  минимальна.

### Заключение

В данной работе предложены формулы для вычисления расстояния между лингвистическими шкалами, удовлетворяющие аксиомам расстояния. Одним из применений расстояния между лингвистическими шкалами является его использование в алгоритме расчета коэффициента рассогласованности лингвистической шкалы. Данный коэффициент, в свою очередь, используется в оптимизационной задаче построения универсальной шкалы для группы экспертов, которая позволяет эффективно проводить экспертизы.

### Литература

1. Рыжов А.П. Элементы теории нечетких множеств и ее приложений. М.: Диалог – МГУ, 2003. 81с.
2. Леденева Т.М. Обработка нечеткой информации. Воронеж: Изд-во Воронежский государственный университет, 2006. 233с.

3. Леденева Т.М., Погосян К.С. Согласование лингвистических экспертных оценок в процедуре группового выбора // Вестник ВГУ, Серия: Системный анализ и информационные технологии. 2010. № 2. С. 125-130.
4. Погосян К.С. Алгоритм построения оптимальной лингвистической шкалы в рамках экспертного оценивания // Системы управления и информационные технологии. 2011. № 3.1(45). С. 180-185.
5. Кофман А. Введение в теорию нечетких множеств: Пер. с франц. М.: Радио и связь, 1982. 432с.
6. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976. 168 с.
7. Пегат А. Нечеткое моделирование и управление. М.: БИНОМ. Лаборатория знаний, 2009. 798 с.

**Савченко В.В.,**

Нижегородский государственный лингвистический университет им.  
Н. А. Добролюбова,  
профессор, зав. каф. математики и информатики  
[svv@lunn.ru](mailto:svv@lunn.ru)

**Акатьев Д.Ю.**

профессор каф. математики и информатики  
[akatjev@lunn.ru](mailto:akatjev@lunn.ru)  
Нижегородский государственный лингвистический университет им.  
Н. А. Добролюбова

***Адаптивный алгоритм формирования фонетической  
базы данных для систем автоматического  
распознавания речи методом фонетического  
декодирования слов<sup>28</sup>***

**Аннотация**

*Рассмотрена проблема вариативности разговорной речи в задаче формирования фонетической базы данных. Предложена для её решения когнитивная акустическая модель типа фонетического кластера, определённого на множестве минимальных звуковых единиц по принципу минимума информационного рассогласования в метрике Кульбака-Лейблера. Разработан адаптивный алгоритм наполнения каждого кластера однотипными (одноименными) звуковыми единицами из непрерывного потока речи диктора. Рассмотрен пример практической реализации такого алгоритма. Представлены результаты его математического моделирования.*

**Введение**

При анализе разговорной (устной) речи на русском языке мы опираемся на наши точные знания в отношении его фонетического строя, количественного и качественного состава используемой фонетической системы, а также закономерностей ее функционирования в разговорной речи. Этими знаниями мы пользуемся, например, при транскрибировании потока речи. Однако при анализе разговорной речи на неизвестном нам языке нам недоступна, в общем случае, информация, относящаяся к его фонетической структуре. В этом случае мы можем, либо, опираясь на наш лингвистический опыт, давать участкам речевого потока приблизительную интерпретацию в рамках Международного фонетического алфавита, либо,

---

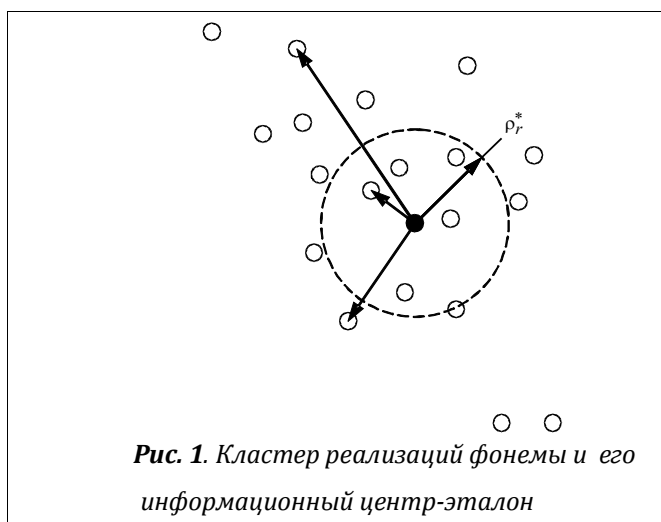
28 Работа выполнена при финансовой поддержке Минобрнауки РФ по государственному контракту № 07.514.11.4137 ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы»

обратившись к акустическим понятиям, членить речь на некие минимальные звуковые единицы (МЗЕ) с соответствующими метками. Очевидно, что второй подход, положенный в основу информационной теории восприятия речи (ИТВР) [1], со всех точек зрения наиболее информативен и универсален. Множество меток всех МЗЕ и составит, в таком случае, звуковой строй данного диалекта или языка, или его фонетическую базу данных (ФБД).

### **Постановка задачи**

Проблема состоит в том, что разговорная речь по своим акустическим характеристикам широко варьируется, причем, не регулярным образом, не только от одного языка к другому, но и от одного носителя к другому носителю одного и того же языка. В указанных условиях становится проблематичной сама идея выделения *повторяющегося* набора МЗЕ из разговорного потока речи. Кроме того, длительность отдельных МЗЕ не превышает нескольких миллисекунд, и это главное препятствие для применения традиционных методов теоретической лингвистики к разговорной (устной) речи. С другой стороны, до настоящего времени проблема не была преодолена и методами экспериментальной фонетики. И главная причина здесь – отсутствие *адекватной системы описания отдельных фонем*.

В поисках путей решения указанной проблемы в упомянутой выше и недавно созданной ИТВР само понятие «фонема» впервые было строго определено в теоретико-информационном смысле как *«множество однородных МЗЕ, объединенных в кластер по критерию минимального информационного рассогласования (МИР) в метрике Кульбака-Лейблера»*. Условно говоря, человеческий мозг объединяет и запоминает в себе как нечто целое (в виде абстрактного образа) разные образцы (произношения) каждой отдельной фонемы в соответствующей «сфере» своей памяти вокруг абстрактного «центра» с заданным «радиусом» (рис. 1).



Нетрудно понять, что данным определением одновременно решается множество актуальнейших проблем в области фонетического анализа речи: и ее вариативности, и априорной неопределенности, и адекватного описания звукового строя языка в терминах ФБД и, наконец, проблема обновления ФБД без разрушения их структуры.

### **Принцип МИР**

Несмотря на существующие различия в реализациях некоторой  $r$ -ой

фонемы все они воспринимаются человеком как нечто общее, иначе речь утратила бы свою информативность. Можно поэтому утверждать, что одноименные реализации  $\mathbf{x}_{r,j}$ ,  $j = \overline{1, J_r}$ ,  $J_r \gg 1$ , в сознании человека группируются в соответствующие классы или речевые образы фонем  $X_r = \{\mathbf{x}_{r,j}\}$ ,  $r = \overline{1, R}$ , вокруг некоторого центра – эталонной метки данного образа. В информационной теории восприятия речи указанные эталоны определяются в строгом, теоретико-информационном смысле: речевая метка  $\mathbf{x}_r^* \subset X_r$  образует *информационный центр-эталон*  $r$ -го речевого образа, если в пределах множества  $X_r$  она характеризуется минимальной суммой информационных рассогласований по Кульбаку-Лейблеру относительно всех других его меток-реализаций  $\mathbf{x}_{r,j}$ ,  $j = \overline{1, J_r}$ .

Нетрудно увидеть, что именно в понятии информационного центра (ИЦ)  $r$ -го множества реализаций одноименных МЗЕ  $X_r$  дается наиболее информативное описание свойств соответствующей фонемы. А множество всех ИЦ  $\{\mathbf{x}_r^*\}$  определяет понятие ФБД для данного диктора. Одновременно становится очевидным и механизм формирования самого этого множества. Сначала анализируемый (входной) речевой сигнал  $X(t)$  в дискретном времени  $t = 0, 1, \dots$  разбивается на ряд последовательных сегментов данных  $\mathbf{x}(t)$  длиной в одну МЗЕ: примерно 10 – 15 мс. После этого каждый такой парциальный сигнал рассматривается в пределах конечного списка фонем  $\{X_r\}$  и отождествляется с той  $X_\bullet$  из них, которая отвечает критерию МИР относительно вектора  $\mathbf{x}(t)$ . Это известная формулировка критерия МИР в задачах автоматического распознавания речи. Задача существенно упрощается, если воспользоваться гауссовой (нормальной) аппроксимацией закона распределения каждой фонемы вида  $\mathbf{P}_r = N(\mathbf{K}_r)$ , где  $\mathbf{K}_r$  - автокорреляционная матрица (АКМ) размера  $n \times n$ ,  $n \geq 1$ .

### **Синтез адаптивного алгоритма**

Предположим, что речевой образ каждой фонемы  $X_r = \{\mathbf{x}_{r,j}\}$ ,  $r = \overline{1, R}$  представлен по-прежнему конечным (объема  $J_r > 1$ ) множеством своих различных векторов-реализаций  $\mathbf{x}_{r,j}$ ,  $j = \overline{1, J_r}$ , составленных из  $L$  последовательных во времени отсчетов одноименных МЗЕ  $\{x_{r,j}(t)\}$  с периодом  $T = 1/(2F) = \text{const}$ . Здесь  $F$  - верхняя граница частотного диапазона речевого тракта. Рассматривая каждую такую реализацию в режиме «скользящего окна» длиной  $n$  отсчетов ( $n \ll L$ ), будем иметь  $(L - n)$  векторов (столбцов) данных  $\{\mathbf{x}_{r,j,i}\}$  размерностью

$n = const$  каждый. Используя после этого формулу среднего арифметического, определим по ним выборочную оценку для АКМ гипотетического гауссова распределения  $\mathbf{P}_r = N(\mathbf{K}_r)$ .

Проблемы возникают, однако, в случае отсутствия априори классифицированных выборок  $\{X_r\}$ , т.е. при распознавании образов «без учителя». Автоматический анализ фонетического состава речи чаще всего относится именно к такому кругу задач. И статистические характеристики фонем, и их используемое каждым диктором число  $R$  из общего списка зависят от особенностей его речевого аппарата. Здесь требуется алгоритм с самообучением, или адаптивный алгоритм фонетического анализа речи (ФАР). Для решения данной задачи в информационной теории разработан специальный инструмент: информационный  $(R+1)$ -элемент. Информационный  $(R+1)$ -элемент – это условный термин, обозначающий устройство или алгоритм для автоматической классификации или распознавания сигнала  $\mathbf{x}$  в пределах некоторого множества классов-альтернатив  $\mathbf{P}_r, r = \overline{1, R}$ . В основе его функционирования применяется статистический подход и критерий МИР. В отличие от других аналогичных алгоритмов с  $R$  выходами  $(R+1)$ -элемент имеет дополнительный,  $(R+1)$ -й выход, который сигнализирует об отказе при распознавании образов одновременно от всех  $R$  заданных альтернатив. Указанная особенность и служит основой для построения эффективного алгоритма распознавания образов в условиях априорной неопределенности. Задача сводится к последовательности задач статистической классификации «с учителем» при переменном (нарастающем) числе альтернатив  $R=1,2,\dots$

Выделим в анализируемом речевом сигнале  $X(t)$  от некоторого диктора первые  $L$  отсчетов из соображений сохранения в них свойства приближенной стационарности или однородности распределения  $\mathbf{P}_r$ . Например, при стандартной частоте дискретизации телефонного канала связи в 8 кГц обычно полагают  $L = 100\dots 200$  (это те же 10 – 15 мс). Используем полученный минимальный сегмент данных  $\mathbf{x}_1 = \{x_1, \dots, x_L\}$  в качестве обучающей выборки  $X_1$  для оценивания АКМ первой МЗЕ из сигнала. Соответствующий закон распределения  $\mathbf{P}_1 = N(\hat{\mathbf{K}}_1)$  – это первый из элементов нашего будущего списка. После этого приравниваем  $R = 1$  и берем второй сегмент выборки для анализа:  $\mathbf{x}_2 = \{x_{L+1}, \dots, x_{2L}\}$ . Следуя выражению для решающей статистики МИР, определим для него удельную величину информационного рассогласования (ВИР) [2]

$$\rho(X_2, X_r) = \rho_r(\mathbf{x}) \Big|_{\mathbf{x} = \mathbf{x}_2} \quad (1)$$

относительно первой МЗЕ (при равенстве  $r = 1$ ). Полученный результат сопоставляется с порогом по ВИР в роли допустимой величины рассогласований между разными реализациями одних и тех же фонем



устной речи:

$$\rho(X_2, X_r) \leq \rho_0. \quad (2)$$

При нарушении данного неравенства в нашем начальном списке фонем появится второй элемент, и вслед за этим приравниваем число выявленных фонем  $R = 2$ . В противном случае принимается решение об объединении выборок  $X_1$  и  $X_2$  в один речевой образ  $\mathbf{P}_1$ : в качестве или одной МЗЕ удвоенной длительности  $L_r = 2L$ , если выборки смежные, или двух разных реализаций первой фонемы, если выборки не стыкуются. Равенство  $R=1$  в обоих случаях сохраняется.

Нетрудно понять, что в форме условия (2) реализуется проверка гипотез об однородности выборок, а понятие фонемы определяется здесь как кластер однородных МЗЕ по критерию МИР. Это типичная формулировка информационного  $(R+1)$ -элемента.

### **Фонетический анализ речи**

Вычисления по схеме (1), (2) повторяются циклически для всех последующих сегментов данных из речевого сигнала  $X(t)$ , причем повторяются «нарастающим итогом» для переменного значения  $R=2,3,\dots$ . Каждый очередной сегмент данных сопоставляется по правилу (2) одновременно со всеми  $R$  множествами  $\{X_r\}$  из текущего списка фонем. При этом не исключается возможность объединения одного и того же сегмента данных с элементами одновременно нескольких разных множеств. В результате будем иметь список фонем с некоторым фиксированным числом элементов  $R^*$ . Это важная характеристика как анализируемого речевого сигнала, так и самого диктора. Чем больше значение  $R^*$  для конкретного диктора, тем богаче с фундаментальной, фонетической точки зрения его речь. В данном выводе и состоит, по-видимому, главный смысл и назначение фонетического анализа речи. Однако здесь же возникает и очевидная проблема: чрезмерно большое число фонем в речи диктора – это признак ее нечеткости, или не информативности. С точки зрения качества устной речи первостепенный интерес, безусловно, представляет собой множество четких МЗЕ. Его, в таком случае, и следует считать основным итогом ФАР. Поэтому логика подсказывает: после выполнения всех перечисленных выше вычислений некоторые «фонемы» из окончательного списка можно исключить как маргинальные.

Добавим к сказанному, что предложенный алгоритм имеет множество разнообразных модификаций за счет, главным образом, применения рекуррентных вычислительных процедур корреляционно-спектрального анализа. Среди них наибольший интерес представляет метод обеляющего фильтра (МОФ), основанный на авторегрессионной модели МЗЕ.

В ранних работах [2, 3] было показано, что в асимптотике, когда  $n \rightarrow \infty$ , и при гауссовом распределении речевого сигнала  $\mathbf{P}_r = N(\mathbf{K}_r)$  с

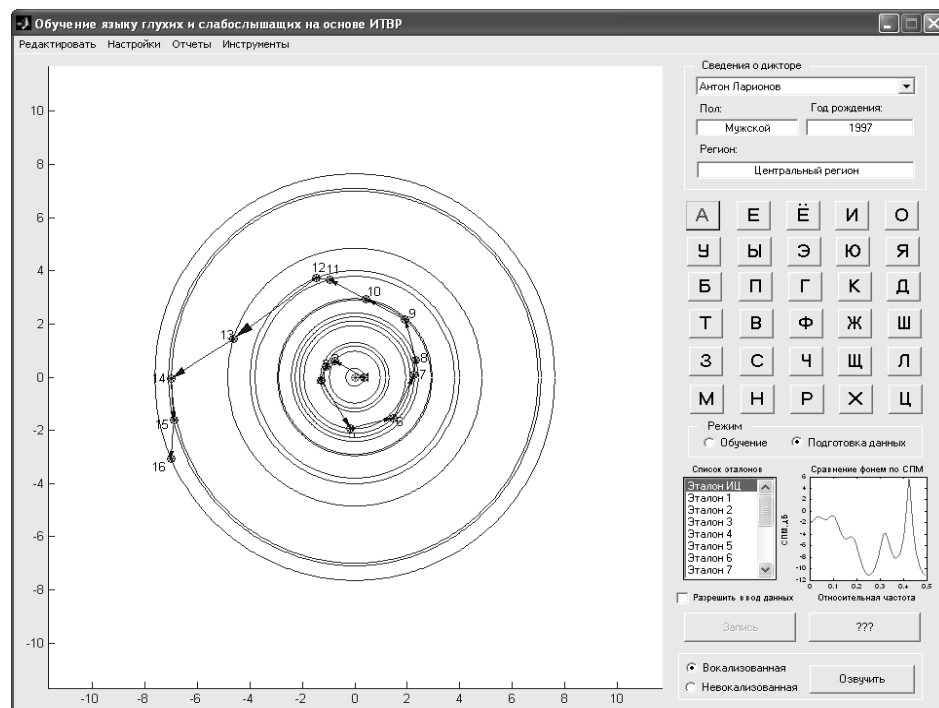
обратной АКМ ленточной структуры выражение для оптимальной решающей статистики из выражения (1) сводится к виду

$$\rho_{x,r} = \frac{1}{F+1} \sum_{f=0}^F \frac{\left| 1 + \sum_{m=1}^P a_r(m) e^{-j\pi m f / F} \right|^2}{\left| 1 + \sum_{m=1}^P a_x(m) e^{-j\pi m f / F} \right|^2} - 1 \geq 0. \quad (3)$$

Здесь  $\{a_x(m)\}$ ,  $\{a_r(m)\}$  - два вектора авторегрессионных (АР) коэффициентов: входного сигнала и  $r$ -го эталона, оба одного порядка  $p > 1$ . Это стандартная формулировка МОФ в частотной области. Преимуществом данной интерпретации критерия МИР является, прежде всего, возможность его эффективной реализации в адаптивном варианте на основе быстрых вычислительных процедур авторегрессионного анализа, таких как метод Берга и др. Именно такой вариант МОФ был реализован в дальнейшем для проведения его экспериментальных исследований в типовой задаче ФАР.

### **Результаты математического моделирования**

Для экспериментальных исследований предложенного алгоритма (1)...(3) была разработана информационная система фонетического анализа разговорной русской речи, основной интерфейс которой показан на рис. 2.



**Рис. 2. Главное окно программы**

Программа экспериментальных исследований была разбита на два этапа [4]. На первом этапе осуществлялось формирование базы эталонов МЗЕ по группе тестируемых дикторов, а на втором – исследование

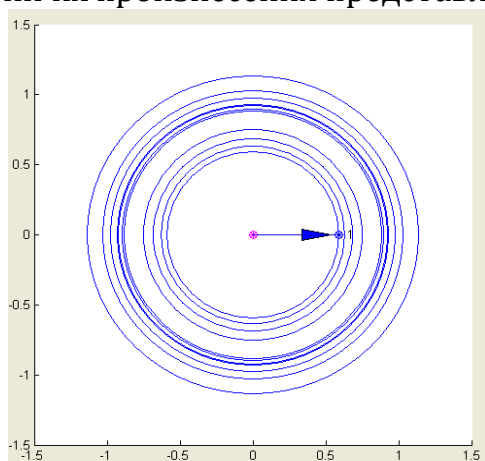
особенностей звукового строя речи тех же дикторов в комфортных и некомфортных условиях. На обоих этапах для работы применялись специальные программные и аппаратные средства: динамический микрофон AKG D77 S и ламповый микрофонный предусилитель ART TUBE MP Project Series USB. Частота дискретизации встроенного АЦП была установлена равной 8 кГц – общепринятая частота при обработке устной речи. Испытания проводились на ноутбуке следующей конфигурации: Asus X50V, 1024 Мбайт ОЗУ, Windows XP, Matlab 6.5. Формирование фонетической базы эталонов происходило следующим образом.

Вначале для каждой из основных (продолжительных) фонем русского языка было записано в комфортных условиях по одному образцу МЗЕ от выбранного диктора-мужчины. Затем к этим образцам были добавлены эталоны того же диктора в тех же условиях, но произнесённые в разное время суток. При этом диктор произносил каждую фонему по 15-20 раз. Звуковой сигнал вводился в информационную систему в реальном времени в режиме «Подготовка данных». Всего, таким образом, было сформировано шесть персональных баз эталонов от шести дикторов-мужчин, а также две базы эталонов от дикторов-женщин.

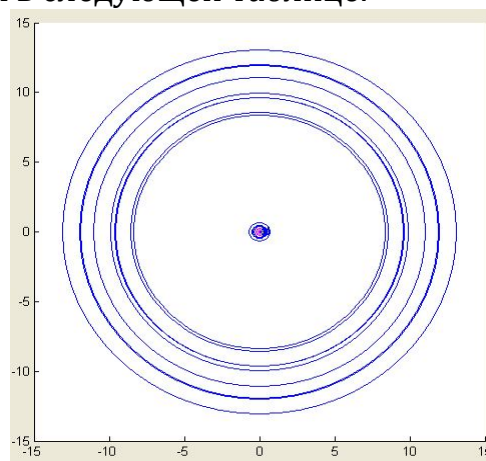
На втором этапе каждый диктор в заведомо менее комфортных условиях: в нашем случае – после значительной физической нагрузки (пульс 140-160 ударов в мин.) произносил каждую из 21 фонем по 10 - 15 раз. И каждый раз информационной системой фиксировался соответствующий результат: текущее значение ВИР по отношению к заранее сформированной базе эталонов. Цель данного эксперимента – выбрать из общего списка фонем национального языка те фонемы, которые наиболее остро реагируют в своих реализациях на условия произнесения их диктором. Смысл этой цели очевиден: настраивая информационную систему на наиболее чувствительные фонемы, мы гарантируем максимальную чувствительность нашего восприятия по отношению к эмоциональному и физическому состоянию диктора. Важнейший момент – это количественная характеристика степени возбуждения диктора, а именно: ВИР между фонемами в текущем сигнале и их эталонами. Для иллюстрации сказанного на рисунках ниже представлены две диаграммы ВИР при произнесении фонемы «Х» некоторым диктором-мужчиной в комфортных (рис. 3) и некомфортных (рис. 4). условиях. Здесь центр окружностей характеризует положение первого эталона в пределах Х-кластера одноименных МЗЕ.

А каждая окружность – это результат очередного произнесения фонемы. Ее радиус определяется значением ВИР по отношению к эталону. Чем больше радиус, тем хуже качество произнесения. Видно, что при изменении условий на некомфортные в среднем на порядок (!) увеличилась вариативность произнесений данного диктора (см. шкалу делений по оси абсцисс). Аналогичные результаты были получены и для других дикторов из контрольной группы. Средние значения ВИР для типичных диктора-

мужчины и диктора-женщины по всему списку фонем в зависимости от условий их произнесения представлены в следующей таблице.



**Рис. 3.** Величина информационного рассогласования при произнесении фонемы «Х» диктором-мужчиной в комфортных условиях



**Рис. 4.** Величина информационного рассогласования при произнесении фонемы «Х» диктором-мужчиной в некомфортных условиях

**Табл. ВИР дикторов в зависимости от условий их монолога**

Диктор ы Фонемы	Диктор-муж.: в комфортных условиях		Диктор-жен.: в комфортных условиях	
	некомфортных условиях		некомфортных условиях	
А	0.77	0.53	1.02	3.84
О	7,6	20.03	6.4	16.4
У	3.12	4.06	12.7	8.3
Э	5.73	9.21	7.17	8.35
Ш	1.47	1.22	2.36	1.73
Щ	0.94	1.73	1.59	2.64
Р	0.72	2.49	2.03	2.71
С	0.58	0.51	0.82	0.93
В	1.38	1.14	1.82	1.92
З	3.87	4.69	4.14	4.73
Ж	0.69	0.71	0.94	1.19
И	2.51	3.04	3.92	4.08
М	1.94	6.14	1.01	3.32
Л	4.7	0.69	2.04	1.86
Ль	2.19	1.54	1.91	1.79
Ф	1.78	1.91	1.83	1.89
Х	0.86	6.9	0.91	4.7
Ч	1.96	1.94	2.13	2.11
Е	3.81	4.57	5.17	5.89
Ы	2.49	3.18	3.67	4.29
Н	1.48	0.41	3.5	2.84

Здесь серым фоном отмечены наиболее чувствительные к условиям

своего произнесения фонемы. Видно, что, по крайней мере, три из них: «Х», «М» и «О» одинаково высокочувствительны как в исполнении мужчин, так и женщин. Все остальные фонемы, напротив, мало чувствительны к условиям произнесения. Именно такого рода фонемы могут быть положены в основу распознавания речи диктора или семантического анализа произнесенных им слов и фраз.

### **Заключение**

К числу приоритетных направлений применения ИТВР и ее когнитивной кластерной модели МЗЕ (рис. 1) наряду с задачами прикладной лингвистики: распознаванием и семантическим анализом разговорной русской речи относятся также проблемы современной диалектологии. *Как сопоставить разные диалекты между собой по степени их объективной близости или различий на базовом, фонетическом, уровне? И какова количественная мера таких различий? Какие тенденции: сближения или удаления по фонетическому строю доминируют в настоящий момент в процессе исторического развития тех или иных диалектов? И, наконец, как можно лучше обучиться данному диалекту или, напротив, максимально ослабить его?* Благодаря методологии ИТВР и ФБД впервые в мировой науке открываются возможности дать четкие ответы на все перечисленные выше и подобные им вопросы. А это стимулирует, в свою очередь, научные исследования в области не только современной лингвистики, но и информатики и, прежде всего, прикладной информатики. Таким образом, полученные в статье результаты открывают качественно новые возможности для решения целого ряда актуальных задач, которые до настоящего времени остаются не решенными или решены неудовлетворительно.

### **Литература**

1. Савченко В.В. Информационная теория восприятия речи // Изв. вузов России. Радиоэлектроника. 2007. Вып. 6. С. 3–9.
2. Савченко В.В., Губочкин И.В. Фонетический анализ речи методом переменного дерева. // Изв. вузов России. Радиоэлектроника. 2008. Вып. 3. С. 14–20.
3. Савченко В.В. Фонема как элемент информационной теории восприятия речи. // Изв. вузов России. Радиоэлектроника. 2008. Вып. 4 С. 3–11.
4. Савченко В.В., Акатьев Д.Ю., Губочкин И.В. Исследование звукового строя национального языка на основе информационной теории восприятия речи // Вестник ННГУ им. Н.И.Лобачевского. Сер. Информационные технологии. 2010. №3 (1). С.215 – 222.

## **Анализ показателей качества функционирования SIP-сервера с гистерезисным управлением нагрузкой**

### **Аннотация**

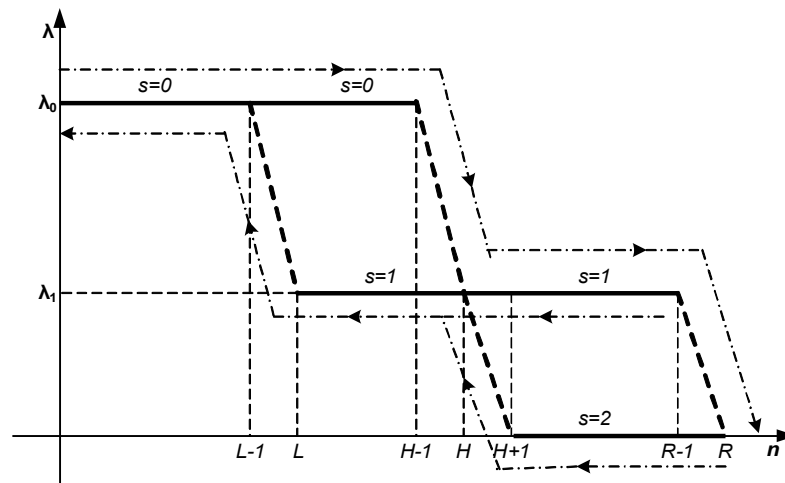
*Целью работы является анализ параметров функционирования SIP-сервера, работающего в условиях гистерезисного управления нагрузкой, что является основным механизмом для предотвращения различного рода перегрузок в современных телекоммуникационных сетях. Для анализа этих параметров в данной статье предлагается модель процесса управления поступающей на сервер нагрузкой в виде системы массового обслуживания типа M/G/1 с гистерезисным управлением входящим потоком и групповым поступлением заявок для учета особенностей обслуживания некоторых SIP-сообщений, как, например, сообщений NOTIFY. Построенная модель позволяет получить соотношения для расчета стационарного распределения вероятностей длины очереди, а также формулы для вычисления основных показателей качества функционирования SIP-сервера, таких как вероятность нахождения в состоянии перегрузки, среднее время возврата и средняя длительность цикла управления.*

### **Введение**

В современных сетях связи для предотвращения перегрузок крайне необходимым является использование инструментов управления нагрузкой. Одним из самых простых и интуитивно понятных таких инструментов является механизм гистерезисного управления нагрузкой, основанный на порогах длины очереди, который был предложен рабочими группами IETF (Internet Engineering Task Force) для предотвращения перегрузок в сетях сигнализации на базе протокола SIP (Session Initiation Protocol) [1-4].

В работе рассматривается один из вариантов гистерезисного управления нагрузкой с тремя порогами: порог обнаружения перегрузки  $H$ , порог снижения перегрузки  $L$  и порог сброса нагрузки  $R$  (рис. 1). Механизм работает следующим образом: при достижении длиной очереди порога  $H$  система переходит в режим перегрузки, интенсивность поступающего потока сообщений снижается. Чтобы режим функционирования не менялся слишком часто, интенсивность поступающего потока не восстанавливается до прежних значений сразу же, как только длина очереди падает ниже порога  $H$ , а только тогда, когда уровень занятости буферной памяти снизится до порога снижения перегрузки  $L$ . Аналогично, при достижении длиной очереди порога  $R$  поступающая нагрузка полностью сбрасывается,

и восстанавливается только тогда, когда уровень занятости буфера снижается до порога  $H$ .



**Рис.1.** Гистерезисное управление нагрузкой

Для построения более адекватной модели функционирования SIP-сервера необходимо также учитывать групповой характер поступления некоторых SIP-сообщений, таких как сообщения NOTIFY при предоставлении услуг присутствия [5]. В связи с этим, а также с целью получения более общих результатов, мы описываем функционирование системы в терминах системы массового обслуживания типа  $M^{[X]}|G|1|\langle L, H \rangle|\langle H, R \rangle$ . Анализ подобной модели с бесконечной очередью и без учета группового поступления заявок проведен в работе [6] на основе аппарата мартингалов. В данной статье получена система уравнений для расчета стационарного распределения построенной модели при помощи аппарата марковских процессов восстановления [7]. Кроме того, получены формулы для некоторых характеристик системы, представляющих интерес для анализа показателей качества функционирования SIP-сервера, таких как вероятность нахождения в режиме перегрузки, среднее время нахождения в режимах перегрузки и сброса нагрузки, а также средняя длительность цикла управления.

#### **Описание модели**

Обозначим  $X(t)$  - двумерный случайный процесс с пространством состояний  $S$

$$S = \left\{ (j, s) \left| \begin{array}{l} j = \overline{0, R}, \quad s = 0 \\ j = \overline{L, R}, \quad s = 1 \\ j = \overline{H+1, R}, \quad s = 2 \end{array} \right. \right\}$$

и его подмножествами  $S_i = \{(j, s) \in S \mid s = i\}$ ,  $i = \{0, 1, 2\}$ , где  $j$  обозначает количество заявок в системе, а  $s$  - режим функционирования системы в

момент времени  $t$ . Заявки поступают на прибор группами, поток групп заявок является пуассоновским с интенсивностью  $\lambda_s, s = \{0, 1, 2\}, \lambda_0 > \lambda_1, \lambda_2 = 0$ . В каждой группе поступает случайное число заявок с вероятностью  $l_i$  того, что поступит ровно  $i$  заявок, а в  $k$  группах поступит  $i$  заявок с вероятностью  $l_i^k$ . Длительность обслуживания является случайной величиной с функцией распределения  $B(x)$  и средним  $b^{(1)} < \Gamma$ .

Пусть  $t_1 < t_2 < \dots$ , где  $t_n$  - момент окончания обслуживания  $n$ -й заявки. Для упрощения анализа будем считать, что режим функционирования  $s$  может меняться только в моменты  $t_n, n > 0$ . Тогда состояния дискретного случайного процесса  $X(t_n + 0)$  образуют вложенную цепь Маркова. Пространство состояний  $X(t_n + 0)$  и его подмножества описываются следующим образом:

$$\tilde{S} = \left\{ (j, s) \left| \begin{array}{ll} j = \overline{0, H-2}, & s = 0 \\ j = \overline{L, R-2}, & s = 1 \\ j = \overline{H+1, R-1}, & s = 2 \end{array} \right. \right\},$$

$$\tilde{S}_i = \{ (j, s) \in \tilde{S} \mid s = i \}, \quad i = \{0, 1, 2\}$$

Обозначим стационарные распределения процессов  $X(t)$  и  $X(t_n + 0)$  соответственно  $\{p_{j,s}\}$  и  $\{q_{j,s}\}$ .

$$p_{j,s} = \lim_{t \rightarrow \Gamma} P\{X(t) = (j, s)\};$$

$$q_{j,s} = \lim_{n \rightarrow \Gamma} P\{X(t_n + 0) = (j, s)\}.$$

Для выражения переходных вероятностей цепи Маркова обозначим  $\beta_k^s$  вероятность того, что за время обслуживания одной заявки в режиме  $s$  в систему поступит еще  $k$  заявок.

$$\beta_k^s = \int_0^{\infty} e^{-\lambda_s x} \frac{(\lambda_s x)^k}{k!} dx.$$

Тогда система уравнений равновесия для вероятностного распределения  $\{q_{j,s}\}$  принимает вид



$$\begin{aligned}
q_{j,0} &= q_{0,0} \sum_{i=1}^{j+1} l_i \sum_{k=0}^{j-i+1} l_{j-i+1}^k \beta_k^0 + \sum_{i=1}^{\min(j+1, H-2)} q_{i,0} \sum_{k=0}^{j-i+1} l_{j-i+1}^k \beta_k^0 + \delta_{j, L-1} \beta_0^0 q_{L,1}, \\
&\quad j = \overline{0, H-2}; \\
q_{j,1} &= q_{0,0} \sum_{i=1}^{j+1} l_i \sum_{k=0}^{j-i+1} l_{j-i+1}^k \beta_k^0 + \sum_{i=1}^{H-2} q_{i,0} \sum_{k=0}^{j-i+1} l_{j-i+1}^k \beta_k^0 + \\
&\quad + \sum_{i=L}^{\min(j+1, R-2)} q_{i,1} \sum_{k=0}^{j-i+1} l_{j-i+1}^k \beta_k^1 + \delta_{j, H} q_{H+1,2}, \quad j = \overline{H-1, R-2}; \\
q_{j,1} &= \sum_{i=L}^{j+1} q_{i,1} \sum_{k=0}^{j-i+1} l_{j-i+1}^k \beta_k^1, \quad j = \overline{L, H-2}; \\
q_{R-1,2} &= q_{0,0} \sum_{i=1}^{\infty} l_i \sum_{j=\max(0, R-i)}^{\infty} \sum_{k=0}^j l_j^k \beta_k^0 + \sum_{i=1}^{H-2} q_{i,0} \sum_{j=R-i}^{\infty} \sum_{k=0}^j l_j^k \beta_k^0 + \\
&\quad + \sum_{i=L}^{R-2} q_{i,1} \sum_{j=R-i}^{\infty} \sum_{k=0}^j l_j^k \beta_k^1; \\
q_{j,2} &= q_{R-1,2}, \quad j = \overline{H+1, R-2};
\end{aligned} \tag{1}$$

где  $\delta_{ij}$  - символ Кронекера.

Теперь, используя вероятностное распределение  $\{q_{j,s}\}$ , можно найти вероятностное распределение  $\{p_{j,s}\}$  для процесса  $X(t)$ .

$$\begin{aligned}
p_{0,0} &= C^{-1} \frac{1}{\lambda_0} q_{0,0}; \\
p_{j,0} &= \frac{C^{-1}}{\lambda_0} \left\{ q_{0,0} \sum_{i=0}^j l_i \sum_{k=0}^{j-i} l_{j-i}^k \left( 1 - \sum_{n=0}^k \beta_n^0 \right) + \right. \\
&\quad \left. + \sum_{i=1}^{\min(j, H-2)} q_{i,0} \sum_{k=0}^{j-i} l_{j-i}^k \left( 1 - \sum_{n=0}^k \beta_n^0 \right) \right\}, \quad j = \overline{1, R-1}; \\
p_{R,0} &= \frac{C^{-1}}{\lambda_0} \left\{ q_{0,0} \sum_{i=0}^{\infty} l_i \sum_{j=\max(0, R-i)}^{\infty} \sum_{k=0}^j l_j^k \left( 1 - \sum_{n=0}^k \beta_n^0 \right) + \right. \\
&\quad \left. + \sum_{i=1}^{H-2} q_{i,0} \sum_{j=R-i}^{\infty} \sum_{k=0}^j l_j^k \left( 1 - \sum_{n=0}^k \beta_n^0 \right) \right\}, \\
p_{j,1} &= \frac{C^{-1}}{\lambda_1} \sum_{i=L}^{\min(j, R-2)} q_{i,1} \sum_{k=0}^{j-i} l_{j-i}^k \left( 1 - \sum_{n=0}^k \beta_n^1 \right), \quad j = \overline{L, R-1}; \\
p_{R,1} &= \frac{C^{-1}}{\lambda_1} \sum_{i=L}^{R-2} q_{i,1} \sum_{j=R-i}^{\infty} \sum_{k=0}^j l_j^k \left( 1 - \sum_{n=0}^k \beta_n^1 \right); \\
p_{j,2} &= C^{-1} b^{(1)} q_{R-1,2}, \quad j = \overline{H+1, R},
\end{aligned} \tag{2}$$

где

$$C = \left( \frac{1}{\lambda_0} + b^{(1)} \right) q_{0,0} + b^{(1)} (1 - q_{0,0}) = b^{(1)} + \frac{1}{\lambda_0} q_{0,0}.$$

### Анализ вероятностно-временных характеристик

Теперь мы можем найти вероятность нахождения системы в режиме перегрузки суммированием соответствующих вероятностей распределения  $\{p_{j,s}\}$

$$P(S_1) = \sum_{(j,s) \in S_1} p_{j,s} = \sum_{j=L}^R p_{j,1}. \quad (3)$$

Аналогично вычисляется вероятность нахождения системы в режиме сброса нагрузки.

$$P(S_2) = \sum_{(j,s) \in S_2} p_{j,s} = \sum_{j=H+1}^R p_{j,2}. \quad (4)$$

Пусть  $P_0$  - матрица переходных вероятностей на подпространстве  $\tilde{S}_0$  размерности  $H-1 \times H-1$ .

$$(P_0)_{ij} = \begin{cases} \sum_{i=1}^{j+1} l_i \sum_{k=0}^{j-i+1} l_{j-i+1}^k \rho_k^0, & i=0, j=\overline{0, H-2}; \\ \sum_{k=0}^{j-i+1} l_{j-i+1}^k \rho_k^0, & i=\overline{1, H-2}, j=\overline{i-1, H-2}; \\ 0, & j < i-1. \end{cases}$$

Рассмотрим поведение процесса  $X(t_n+0)$  в подмножестве состояний  $\tilde{S}_0$ . Выходя из подпространства  $\tilde{S}_1$  в  $\tilde{S}_0$ , система всегда попадает в состояние  $(L-1,0)$ , поэтому начальное распределение в подпространстве  $\tilde{S}_0$  можно выразить вектором

$$\mathbf{e}_L^T = \left( \underbrace{0, \dots, 0}_{L-1}, \underbrace{1, 0, \dots, 0}_{H-L} \right).$$

Пусть также  $\mathbf{a}^T = (a_0, a_1, \dots, a_{H-2})$ , где  $a_i = 1 - \sum_{j=0}^{H-2} (P_0)_{ij}$  - вероятность того, что система, находясь в состоянии  $(i,0)$ , выйдет из подпространства  $\tilde{S}_0$  на следующем шаге цепи Маркова. Тогда  $\mathbf{e}_L^T P_0^n \mathbf{a}$  - вероятность того, что система выйдет из подмножества  $\tilde{S}_0$  ровно через  $n+1$  шаг процесса  $X(t_n+0)$ , а среднее число шагов  $\tilde{\tau}_0$  равно

$$\tilde{\tau}_0 = \sum_{n=0}^{\infty} (n+1) \mathbf{e}_L^T P_0^n \mathbf{a} = \mathbf{e}_L^T (\mathbf{I} - P_0)^{-1} \mathbf{1}. \quad (5)$$

Чтобы получить среднее время  $\tau_0$  нахождения процесса  $X(t)$  в подпространстве  $S_0$  необходимо домножить выражение (5) на среднюю длительность одного шага цепи Маркова  $X(t_n+0)|_{\tilde{S}_0}$ .

$$\tau_0 = \left( b^{(1)} + \frac{1}{\lambda} \frac{q_{0,0}}{P(S_0)} \right) \mathbf{e}_L^T (\mathbf{I} - \mathbf{P}_0)^{-1} \mathbf{1}. \quad (6)$$

В стационарном режиме система входит и выходит из подмножества состояний  $S_0$  одинаковое число раз, отсюда находим  $\tau_{12}$  - среднее время нахождения в подпространстве  $S_1$  и  $S_2$ .

$$\tau_{12} = \tau_0 \frac{P(S_1 \cup S_2)}{P(S_0)}. \quad (7)$$

Из формул (6) и (7) легко найти среднюю длительность цикла управления  $\tau$ :

$$\tau = \tau_0 + \tau_{12}. \quad (8)$$

### **Заключение**

В статье для анализа показателей качества функционирования SIP-сервера построена модель типа M/G/1 с гистерезисным управлением и групповым поступлением заявок. На основании модели были выписаны системы уравнений, позволяющие вычислять стационарное распределение вероятностей системы, получены формулы основных вероятностно-временных характеристик: вероятности нахождения в режиме перегрузки и режиме сброса нагрузки, среднее время возврата из режима перегрузки и среднюю длительность цикла управления.

### **Литература**

1. Rosenberg J. Requirements for Management of Overload in the Session Initiation Protocol. // RFC 5390, – декабрь 2008.
2. Hilt V., Noel E., Shen C., Abdelal A. Design Considerations for Session Initiation Protocol (SIP) Overload Control. // RFC 6357, август 2011.
3. Montagna S., Pignolo M. Load control techniques in SIP signaling networks using multiple thresholds. // 13th International Telecommunications Network Strategy and Planning Symposium, NETWORKS, 2008. – pp. 1–17.
4. Абаев П.А., Гайдамака Ю.В., Самуйлов К.Е. Гистерезисное управление нагрузкой в сетях сигнализации. // Вестник РУДН. Серия Математика. Информатика. Физика, 2011, №4. – С. 55-73.
5. Chi C., Hao R., Wang D., Cao Z. IMS presence server: traffic analysis & performance modeling. // International Conference on Network Protocols, 2008. – pp. 63-72.
6. Roughan M., Pearce C.E.M. A martingale analysis of hysteretic overload control. // Advances in Performance Analysis, 2000, №1. – pp. 1-30.
7. Самуйлов К.Е., Сопин Э.С. К анализу системы M<sup>[X]</sup>|G|1|г с прогулками прибора. // Вестник РУДН. Серия Математика. Информатика. Физика, 2011, №1. – С. 91-97.

**СЕКЦИЯ 6. НАУЧНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В  
ОБРАЗОВАНИИ И НАУКЕ**

**Гайфуллин Б.Н.,**

Негосударственное образовательное учреждение УКЦ Интерфейс,  
директор,  
bulat@interface.ru

**Туманов В.Е.**

Институт проблем химической физики РАН,  
зав. лабораторией  
tve@icp.ac.ru

## ***Предметно-ориентированные системы научной осведомленности в науке и образовании***

### **Аннотация**

*В статье рассмотрены теоретические основы концепции предметно-ориентированных систем научной осведомленности, как узкоспециализированных информационно-аналитических систем для сбора, верификации, хранения, извлечения, интеллектуального анализа данных и производства новых знаний. Такие системы являются выделенным классом систем поддержки и принятия решений на основе технологии хранилищ данных и знаний в узкоспециализированных научных областях.*

### **Введение**

В результате научных экспериментов и компьютерного моделирования постоянно накапливаются большие объемы данных, которые организуются в электронные информационные ресурсы: базы и хранилища данных, электронные информационные и вычислительные системы, научные центры данных. Такие информационные ресурсы становятся местом накопления, хранения, верификации, извлечения, использования и распространения профессиональных и корпоративных знаний.

Эффективное развитие науки и высоких технологий требует интенсивной обработки и анализа фундаментальных знаний, накопленных в различных исследовательских организациях, что приводит к потребности в развитии информационных технологий накопления, извлечения и анализа предметно-ориентированных профессиональных знаний на основе разработки универсальных и специализированных моделей организации и представления научных данных и знаний в электронных ресурсах.

Таким образом, создание информационных ресурсов в Интернет, предназначенных для сбора, хранения, верификации, извлечения, распространения и производства новых профессиональных знаний в различных предметных областях является актуальной научной и научно-практической задачей.

### ***Системы научной осведомленности***

В последнее время бурное развитие получили системы деловой осведомленности или бизнес-аналитики (Business Intelligence System, BI System), в основе которых лежат технологии складирования данных (Data Warehousing), анализа и извлечения знаний (Data Mining) [1]. Успех в использовании систем деловой осведомленности в бизнесе привел к идее разработки и создания систем научной осведомленности (Science Intelligence System, SI System).

Одна из первых попыток дать определение систем научной осведомленности была предпринята в статье R. Hackathorn [2].

*Определение 1.* Система научной осведомленности есть «информационная инфраструктура, которая обеспечивает принятие решений и совместную работу научного сообщества в рамках выделенной предметной области знаний» [2].

Там же рассмотрена общая архитектура таких систем в разрезе категорий основных пользователей и использования современных информационных технологий.

В системах научной осведомленности научные решения, методология и методы исследований интегрируются в общую библиотеку решений, а данные из разнородных источников интегрируются в общее хранилище данных, которое через предметно-ориентированные информационные ресурсы поставляет информацию пользователям: ученым, технологам, студентам, преподавателям, представителям промышленности и государства. Особенностью систем научной осведомленности является предоставление пользователям, помимо собственно профессиональной информации, инструментария для анализа данных.

К основным требованиям, которые предъявляются к таким системам, относятся следующие.

Поскольку интенсивность потоков получаемых научных данных непрерывно растет, особенно в крупномасштабных научных проектах, используемые информационные технологии должны быть способны обработать эти потоки данных, что приводит к требованию масштабируемости данных в таких системах.

Источники данных зачастую имеют территориальную распределенность, что приводит к требованию наличия **стандартов** представления данных и открытости систем научной осведомленности для приема и обработки научных данных.

Как правило, полученные электронные массивы научных данных требуют применения сложных методов анализа, многократно и под различными углами зрения, что требует представления данных в форме, пригодной для применения различных методов интеллектуального анализа данных (Data Mining).

От систем научной осведомленности требуется также наличие механизмов своевременной доставки данных конечному потребителю по первому требованию, также включая поиск опубликованных данных в

системе научных библиотек.

Поскольку системы научной осведомленности предназначены для коллективного использования и совместной работы, они должны предоставлять быстрый доступ к данным и простые способы обмена информацией между работающими, возможно, в разных странах специалистами.

Перечисленные общие требования к системам научной осведомленности приводят к необходимости использования таких информационных технологий, как порталные технологии, технологии складирования данных, XML технологии, вычислительных кластеров, серверов высокой готовности и ряда других. Реализация указанных требований в полном масштабе возможна лишь в крупномасштабных, хорошо финансируемых научных проектах.

Как видно из выше сказанного, системы научной осведомленности можно рассматривать как класс научных информационных систем, ориентированных на интеллектуальный анализ данных с целью поддержки принятия решений в сфере исследований и получения новых профессиональных знаний.

- Элемент перечисления (маркированный список)
- Элемент перечисления

Предметно-ориентированные системы научной осведомленности

В национальном стандарте США [3] термин «производство знаний» определяется как:

- разработка и обеспечение новых знаний (JECD 1966:2);
- обстоятельства, при которых люди, группы людей и организации успешно генерируют новые знания и практики (OECD 2000:39).

Далее в тексте работы под производством знаний понимается извлечение новых знаний из эмпирических данных в рамках компьютерной системы с участием человека и использованием методов прикладного искусственного интеллекта.

В нашей работе [4] было дано определение **предметно-ориентированных систем научной осведомленности**, как узкоспециализированных систем научной осведомленности, которые кроме возможности решения задач интеллектуального анализа данных наделены способностью производства новых профессиональных знаний.

*Определение 2.* Предметно-ориентированная система научной осведомленности есть информационная инфраструктура, которая обеспечивает интеллектуальный анализ данных, принятие решений, производство новых знаний и совместную работу научного сообщества в рамках выделенной узкоспециализированной профессиональной области.

Общая программная архитектура такой системы показана на Рис. 1.

Как показано на рисунке 1, основными компонентами предметно-ориентированной системы научной осведомленности являются, помимо

инструментария интеллектуального анализа данных, хранилище знаний и подсистема производства профессиональных знаний, включающая набор встроенные экспертных систем и искусственных нейронных сетей.



**Рис. 1.** Общая программная архитектура предметно-ориентированных систем научной осведомленности

С учетом требования анализа и производства знаний в программно-технологическую архитектуру предметно-ориентированных систем научной осведомленности предъявляются следующие дополнительные программные компоненты:

- *Компонент фактографических научных баз данных*, содержащих экспериментальные или модельные данные, в частности, фундаментальные константы, числовые и лингвистические характеристики химических или физических процессов.
- *Компонент интеллектуального анализа данных*. Поскольку новые инструменты научных исследований обладают исключительной точностью, увеличивается точность и качество фактографических данных. Для анализа таких данных с целью нахождения тонких эффектов, упущенных в предыдущих исследованиях, требуется набор алгоритмов, позволяющий проводить сложный анализ данных.
- *Компонент производства новых знаний*. Найденные в результате интеллектуального анализа данных эмпирические закономерности позволяют строить прогнозы значений физических или химических процессов и оценивать значение фундаментальных характеристик материалов. Это создает предпосылки для встраивания в предметно-ориентированные системы научной осведомленности элементов прикладного искусственного интеллекта, например, экспертных систем для производства новых знаний и их сохранения в системе.
- *Компонент распространения профессиональных знаний (дистанционного обучения)*. Наличие такого компонента в системе делает ее более привлекательной для использования и распространения предметно-ориентированных знаний, а также служит привлечению заинтересованного круга профессиональных пользователей к производству новых знаний.

Предметно-ориентированные системы научной осведомленности могут быть созданы в научных проектах меньшего масштаба. При хорошо



организованной и спроектированной системе метаданных, они легко могут быть интегрированы в более крупные системы научной осведомленности с учетом территориальной распределенности последних.

Для разработки и создания предметно-ориентированных систем научной осведомленности целесообразно использовать технологии мультиагентных систем. Компонентом такой системы становится интеллектуальный агент, который можно представить в виде веб - приложения, наделенного искусственным интеллектом, и расположенного за некоторым внешним порталом. При этом сам агент ориентирован на обработку научных данных в узкоспециализированном разделе предметной области. При наличии протокола взаимодействия между такими агентами, система научной осведомленности в целом строится поэтапно.

### ***Хранилище знаний предметно-ориентированных систем научной осведомленности***

Центральным компонентом предметно-ориентированных систем научной осведомленности является хранилище знаний.

Перед тем как дать определение хранилища знаний, напомним определение хранилища данных, приведенное в [5]. Под ***хранилищем данных*** понимается «предметно-ориентированная, интегрированная, неизменяемая и поддерживающая хронологию электронная коллекция данных для поддержки принятия решений».

Подобно хранилищу данных, хранилище знаний может рассматриваться как предметно-ориентированная, интегрированная, содержащая временные ряды и поддерживающая процесс принятия решений электронная коллекция. Однако, в отличие от хранилища данных, хранилище знаний является комбинацией неизменяемых и изменяемых объектов и компонент, и должна хранить не только данные, но и знания предметной области.

В процессе исследования данных методами интеллектуального анализа полученные знания могут изменяться. Так могут измениться параметры центра кластера в зависимости от поступления новых данных или используемого алгоритма кластеризации. Или может быть уточнен вид зависимости в данных. Например, параболический тренд может оказаться справедливым только в определенном интервале, а вне этого интервала будет иметь место линейный тренд. Компонент производства знаний может генерировать данные, которые являются прогнозом значений некоторого параметра, но экспериментального подтверждения не имеют, т.е. знания, хранимые в хранилище знаний, могут изменяться и дополняться в результате его эксплуатации.

Дадим следующее определение ***хранилища знаний***.

***Определение 3.*** Хранилище знаний есть предметно-ориентированная, интегрированная, поддерживающая временные ряды данных электронная коллекция, которая содержит данные, знания, процедуры генерирования

знаний и используется для анализа и исследования данных, производства новых знаний и поддержки принятия решений.

Определение, приведенное выше, является расширением определения, данного в [6]. Существенное отличие настоящего определения состоит в добавлении в хранилище знаний компонентов интеллектуального анализа данных и средств генерирования знаний.

Общая программно-аппаратная архитектура хранилища знаний приведена на Рис. 2.



**Рис. 2.** Общая программно-аппаратная архитектура хранилища знаний

Как видно из рисунка 2, хранилище знаний включает хранилище данных для исследования (Exploration Data Warehouse), которое содержит неизменяемые во времени данные (экстенциональный фрагмент хранилища данных), базу знаний выделенной предметной области, встроенные механизмы производства профессиональных знаний, хранилище производных данных (интенциональный фрагмент хранилища данных), интерфейс пользователя и интерфейс эксперта. Интерфейс пользователя предназначен для выборки и распространения знаний, а интерфейс эксперта - для производства новых знаний.

Таким образом, предметно-ориентированную систему научной осведомленности можно рассматривать как интеллектуальную систему в сети Интернет, назначением которой является сбор, хранение, верификация, извлечение, распространение и производство новых предметно-ориентированных знаний в конкретной предметной области человеческого знания.

Использование при проектировании и создании такой системы порталных и мультиагентных технологий, т.е. создание их как веб - ориентированных программных систем с использованием элементов прикладного искусственного интеллекта, делает предметно-ориентированные системы научной осведомленности более удобными для поэтапной разработки.

В основу модели данных системы можно положить совокупность

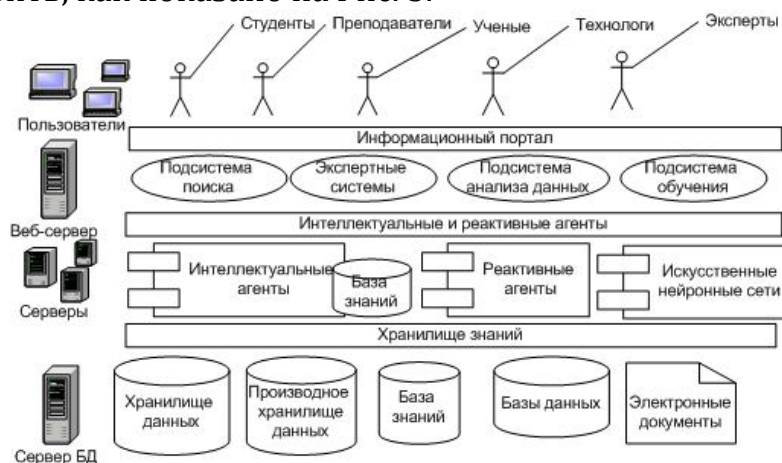
связанных киосков данных, активно взаимодействующих как с базой знаний, так и с внешней средой. В качестве модели представления знаний можно использовать таблицы решений, правила продукций, фреймы, семантические сети.

### **Программно-технологическая архитектура предметно-ориентированной системы научной осведомленности в Интернет**

Предметно-ориентированные системы научной осведомленности предназначены для решения следующих задач:

- сбор, верификация и хранение фактографических и текстовых данных в рамках конкретно выбранной предметной области;
- поиск данных фактографических и текстовых данных в рамках конкретно выбранной предметной области;
- извлечение данных для выполнения интеллектуального анализа данных и сохранение результатов анализа в хранилище данных системы;
- производство и распространение новых данных и профессиональных знаний в рамках конкретно выбранной предметной области.

Программно-технологическую архитектуру системы научной осведомленности безотносительно к конкретной программной области можно представить, как показано на Рис. 3.



**Рис. 3. Программно-технологическая архитектура системы научной осведомленности**

Как показано на рисунке, система состоит из нескольких программных слоев. Первый слой реализован как предметно-ориентированное веб - приложение, которое предоставляет пользователю интерфейс и принимает управляющие решения. Данное веб – приложение предоставляет доступ к следующим программным компонентам системы: информационной подсистеме, аналитической подсистеме, подсистеме дистанционного обучения, встроенной подсистеме объяснений и подсистеме производства новых профессиональных знаний.

Второй слой состоит и интеллектуальных агентов, реактивных агентов и обученных искусственных нейронных сетей, которые реализуют

работу встроенных в портал экспертных систем и выполняют функции поиска информации. Агенты распределены в различных узлах локальной сети.

Третий слой представляет собой хранилище знаний, которое состоит из:

- хранилищ экспериментальных данных, реализованных как связанные киоски данных (интенциональные киоски данных);
- производных хранилищ данных, в которые пользователи имеют возможность заносить полученные ими в результате работы с системой новые значения (экстенциональные киоски данных);
- базы знаний, которая состоит из правил продукций, таблиц решений, процедур выполнения расчетов, алгоритмов кластерного анализа данных и общих фактов, используемых экспертными системами;
- электронных документов, которые в частности представляют собой материалы учебных лекций по предметной области, а также тезаурусы терминов и файлы подсистемы объяснений.

Активными элементами (агентами) производства новых знаний в системе являются встроенные в портал или размещенные за порталом экспертные системы и искусственные нейронные сети, представленные как набор взаимодействующих веб-сервисов.

Полученные в результате использования пользователем активных элементов могут быть сохранены в хранилище знаний. Такая возможность делает систему активной и позволяет заинтересованному научному сообществу накапливать в ней новые знания.

Возможность пополнения системы предметными знаниями накладывает на ее функционал определенные ограничения. Такие ограничения связаны с необходимостью обеспечить достоверность вносимых в нее данных. Поэтому хранилище знаний системы состоит из двух разделов: базового раздела, составленного экспертами по данным научных публикаций, и раздела, произведенного пользователями системы.

При попытке занесения новых данных в блок контекстного управления на основе логических рассуждений делает ряд проверок и выводов о достоверности этих данных, а затем принимает решение либо о занесении данных с определенным показателем их надежности, либо об отказе в запоминании данных.

Одной из особенностей архитектуры системы, что делает ее привлекательной для использования в профессиональном образовании, является наличие подсистемы дистанционного обучения. Основу подсистемы дистанционного обучения составляют веб-ориентированные курсы лекций, разработанные для обучения студентов и аспирантов. Курсы включают в себя тексты лекций, электронные задачки, разнесенные по лекциям, и тесты.

Подсистема объяснений блока контекстного управления обращается

в случае необходимости к фрагментам лекций и управляющего тезауруса основных терминов при выполнении вычислений.

Многократное преобразование исходных данных и информации в предметно-ориентированных системах научной осведомленности, формирование предметных знаний и производство новых знаний делает решение задачи управления знаниями актуальным, поскольку система функционирует в среде Интернет.

Управление знаниями является набором формальных процедур управлением знаниями с целью обеспечения доступа и многократного использования заинтересованными группами с использованием новых технологий.

В предметно-ориентированной системе научной осведомленности знания рассредоточены по взаимосвязанным киоскам данных (факты) и базе знаний (правила, семантические сети). Для скоординированного их использования и производства новых знаний необходимо разработать формальную структурированную схему знаний. Сформулируем определение схемы знаний предметно-ориентированной системы научной осведомленности, исходя из представления о процессе управления знаниями.

Общий процесс управления знаниями включает:

- Анализ предметной области. Идентификацию предметно-ориентированной области в ее терминах.
- Концептуальное моделирование предметной области, основанное на создании терминологического глоссария (управляемого словаря) и определения взаимосвязи между терминами.
- Сбор знаний из полнотекстовых источников, Интернет и информационных научных систем.
- Конструирование базы знаний и тестирования ее компетенции на соответствие требованиям.

Под схемой знаний будем понимать структурированную область знаний, которая может быть определена как отношение

$$\langle A; \cup_i \Gamma_i; M; P; \Sigma; \Omega \rangle, \quad (9.1)$$

где  $A$  - управляющий словарь (Глоссарий терминов и принципов предметной области);

$\cup_i \Gamma_i$  - физическая структура данных;

$M$  - метаданные;

$P$  - модели предсказания и их интерпретация в терминах предметной области;

$\Sigma$  - база знаний;

$\Omega$  - механизмы взаимодействия компонент системы.

Схему знаний можно смоделировать на основе раскрашенных сетей Петри, фреймов, семантической сети, продукционных правил или таблиц

решений.

### **Заключение**

Проведен анализ требований к системе и определены базовые функциональные возможности системы, предложена программно-технологическая архитектура системы, сформулирована концепция предметно-ориентированной системы научной осведомленности.

Полученные результаты позволяют предположить, что разработка и публикация в сети Интернет предметно-ориентированных систем научной осведомленности на основе использования хранилищ знаний позволит научному сообществу создавать распределенные сети для сбора, хранения, извлечения, интеллектуального анализа, распространения и производства знаний в узкоспециализированных областях исследований и технологий. Создание таких систем возможно небольшими коллективами ученых.

Включение в такие системы подсистемы дистанционного обучения предметно-ориентированным знаниям значительно расширяет круг ее потенциальных пользователей (студентов и аспирантов), что способствует самостоятельному формированию у них профессиональных знаний, а преподавательскому составу высших учебных заведений предоставляет дополнительный учебный материал и электронный ресурс-справочник.

Таким образом, предложена теоретическая концепция проблемно-ориентированных систем управления, принятия решений и оптимизации объектов интеллектуального научного исследования. Такие системы могут стать основой для разработки методов получения, анализа и обработки экспертной информации в научной и технологической сферах деятельности.

### **Литература**

1. Thierauf R.J. Effective Business Intelligence Systems. - Westport. Quorum Books. 2001. 392 p.
2. Hackathorn R. Science Intelligence. Can a Business Intelligence Approach Enable «Smart» Science? DM Review. 2005. [Электронный ресурс] / Режим доступа: <http://www.DMReview.com>
3. Proposed Draft American Standard. Knowledge Management Vocabulary. April, 2003.
4. Туманов В.Е. Предметно-ориентированные системы научной осведомленности в физической химии радикальных реакций // Сборник трудов пятой международной научно-практической конференции «Исследование, разработка и применение высоких технологий в промышленности». Том. 12. «Высокие технологии, фундаментальные и прикладные исследования, образование». Спб.: Издательство Политехнического университета, 2008. С. 41-42.
5. Inmon W.H. Building the Data Warehouse. - John Wiley. 3rd Ed. 2004. 412 p.
6. Firestone J. M. Enterprise Information Portals and Knowledge Management. - Oxford: Butterworth-Heinemann. 2003. 422 p.

**Кальдин Д.А.,**

ФГБОУ ВПО «Саратовский государственный технический университет  
имени Гагарина Ю.А.», аспирант  
[diman\\_ne@mail.ru](mailto:diman_ne@mail.ru)

**Мурашев Д.А.,**

Энгельсский технологический институт (филиал) ФГБОУ ВПО  
«Саратовский государственный технический университет имени Гагарина  
Ю.А.», к.ф.-м.н, доцент  
[murashev@mail.ru](mailto:murashev@mail.ru)

**Клинаев Ю.В.,**

Энгельсский технологический институт (филиал) ФГБОУ ВПО  
«Саратовский государственный технический университет имени Гагарина  
Ю.А.», д.ф.-м.н, профессор  
[klinaev@techn.sstu.ru](mailto:klinaev@techn.sstu.ru)

**Терин Д.В.**

Энгельсский технологический институт (филиал) ФГБОУ ВПО  
«Саратовский государственный технический университет имени Гагарина  
Ю.А.», к.ф.-м.н, доцент  
[terinden@mail.ru](mailto:terinden@mail.ru)

### ***Оптимизация алгоритмов компьютерного моделирования трехмерного физического пространства***

В природе существует множество взаимосвязанных объектов, визуальное моделирование которых вместе с окружающим физическим пространством является одним из эффективных методов их изучения. Такие технологии моделирования основаны на реализации математического описания трёхмерных объектов в динамике их взаимодействия с помощью специальных программ, - так называемых «движков». Наибольшее развитие такие программы получили в игровых приложениях. Игровой движок — это центральный программный компонент интерактивных приложений с графикой, обрабатываемой в реальном времени [1]. Основной задачей движка является обеспечение полноценного функционирования моделируемого трёхмерного окружения.

Не затрагивая индустрию компьютерных игр, можно выделить следующие направления развития и применения алгоритмов виртуализации:

- моделирование физических процессов в сложных системах, близких к окружающему пространству нашей или другой планетной системы;
- проведение экспериментов с людьми, но без их участия, например, в

медицинских целях;

- тренировка навыков специализированного персонала, например, пожарных, врачей, военных и т.п., – разработка компьютерных тренажёров;

В настоящее время развитию данного направления компьютерного моделирования в нашей стране не уделяется должного внимания. Не за горами то время, когда виртуальные миры, виртуальная реальность окажутся достаточно правдоподобными и похожими на окружающий нас мир в силу технических возможностей.

Основной частью движка давно принято считать визуализатор, поскольку именно то, что мы видим и формирует у нас самое глубокое впечатление. Для отображения виртуальной трёхмерной среды на экране компьютера необходимо программно реализовать определённый алгоритм и формат представления данных о трёхмерной пространстве.

### ***О проблемах моделирования пространства***

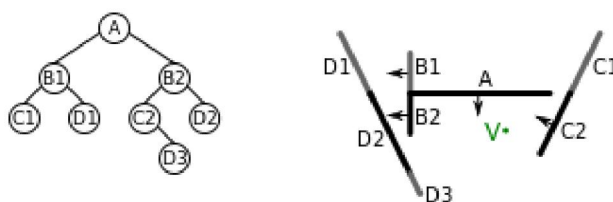
Геометрическое описание окружающей среды не является сложной задачей и в наипростейших случаях математическое моделирование пространственного расположения физических объектов сводится к описанию объектов трёхмерного пространства координатами их точек с той или иной точностью. Но настолько подробное описание, при всей его полезности и информативности, не может достаточно удобно обрабатываться. Одной из причин этого являются потребные значительные вычислительные мощности для обработки, хранения и обеспечения передачи координатной и иной информации (цвет, чёткость, яркость, контрастность и т.п.), в зависимости от качества компьютерной визуализации реального объекта и технических требований к приложению моделирования.

Чтобы обеспечить визуализацию моделей трёхмерного пространства в режиме реального времени необходимы различные ухищрения и допущения при разработке моделей в плане их упрощения, но без потери адекватности.

Рассмотрим наиболее распространённые из них.

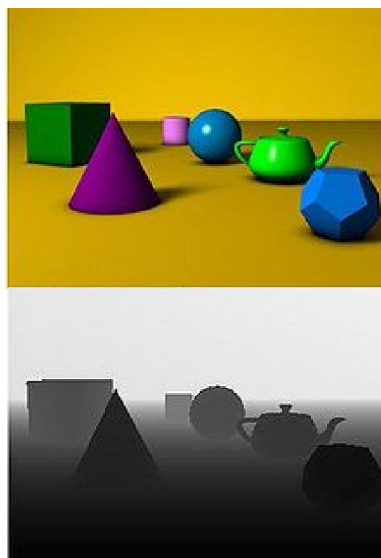
Двоичное разбиение пространства (Binary Space Partition, BSP). BSP-дерево — это структура данных, используемая в трёхмерной графике. В BSP-дереве каждый узел связан с разбивающей прямой или плоскостью в 2-мерном или 3-мерном пространстве соответственно. При этом все объекты, лежащие с фронтальной стороны плоскости относятся к фронтальному поддереву, а все объекты, лежащие с оборотной стороны плоскости относятся к оборотному поддереву [2]. Таким образом, мы не обрабатываем всю модель мира в каждом проходе алгоритма, а лишь только видимую её часть. Недостатком данного алгоритма принято считать необходимость замкнутости моделируемого пространства. Ниже проиллюстрировано дерево для двухмерного случая пространства.





**Рис. 1.** Данные в BSP-дереве

Z-буферизация — в компьютерной трёхмерной графике способ учёта удалённости элемента изображения. Представляет собой один из вариантов решения «проблемы видимости». Очень эффективен и практически не имеет недостатков, если реализуется аппаратно. Основным недостаток Z-буферизации состоит в потреблении большого объёма памяти: в работе используется так называемый буфер глубины или Z-буфер [2]. Наглядное представление хранимых данных отражено ниже.



**Рис. 2.** Данные в Z-буфере

Ray casting, Метод «бросания лучей» — один из методов рендеринга в компьютерной графике, при котором сцена строится на основе замеров пересечения лучей с визуализируемой поверхностью [3]. Обладая отличной скоростью, этот метод отличается крайне низкой точностью вычислений, что ограничивает его применение лишь для узкого круга локальных задач.

#### **Перспективы развития.**

Для создания оптимальной и адекватной модели окружающего пространства необходимо чётко представлять аппаратные возможности современных компьютеров и тенденции их развития. Анализируя этапы развития вычислительной техники за последние два десятка лет [4], можно заключить, что приоритетными являются следующие преимущества:

- многопоточность вычислительных операций,
- высокая производительность вычислений на графических адаптерах,
- доступность больших объёмов, как оперативной, так и

энергонезависимой видов памяти.

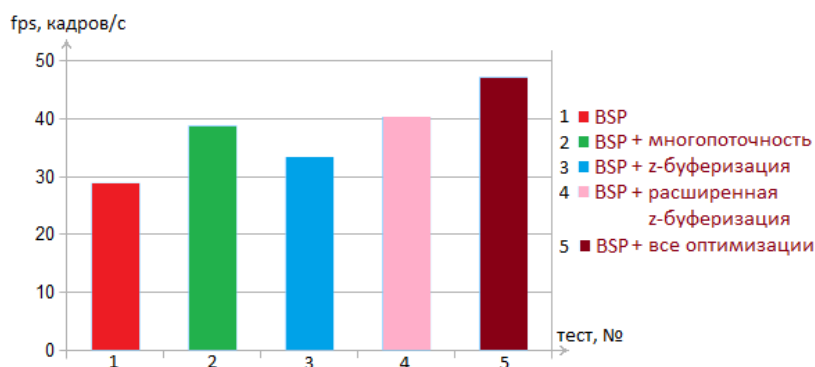
Ориентируясь на эти особенности, можно с большей уверенностью заниматься разработкой новых и усовершенствованием уже существующих алгоритмов.

За базовую структуру хранения окружающего пространства примем двоичные деревья. Но чтобы обеспечить наибольшую производительность при большей насыщенности и точности объектов виртуального мира необходимо переопределить логическую модель ветвей и листьев «дерева».

Ориентируясь на многопоточность вычислений, следует хранить в них не просто координаты объектов, а более широкие структуры, группы объектов. Поскольку время является лимитирующим фактором, правильнее будет параллельно обработать несколько объектов, необязательно только нужных.

Имея в своём распоряжении достаточный объём оперативной памяти, можно интегрировать в двоичное дерево алгоритм z-буферизации, что уже используется в современных продуктах [5]. Но помимо параметра удалённости объектов, следует хранить информацию о важности и размере объекта. Например, на расстоянии 100 м от наблюдателя, десятиэтажное здание будет хорошо видно и его нужно видеть, а вот спичечный коробок на таком же расстоянии будет не заметен (или вовсе не нужен, если модель строится для изучения архитектуры). Не обрабатываемые бесполезные объекты освобождают ресурсы вычислительных систем для более детального моделирования нужных объектов или для увеличения числа моделей.

### **Оценка производительности.**



**Рис. 3. Диаграмма производительности**

Описанные оптимизации алгоритма двоичного разбиения пространства были реализованы и протестированы. Использовалось виртуальное пространство со среднестатистическим набором объектов больших и малых размеров. Тестирование проводилось на современной платформе на базе двухъядерного процессора производства фирмы Intel 2012 года. Анализ проводился по количеству кадров изображения просчитанных вычислительной системой за секунду. Взяты средние

значения по результатам 10 тестов. Результаты отражены в диаграмме.

### **Заключение**

Проведённые эксперименты говорят о том, что некоторые улучшения скорости обработки удастся получить, прибегнув к комбинации алгоритмов и проведя их оптимизацию и расширение.

С другой стороны, одновременное использование всех оптимизаций не даёт результат равный сумме приростов производительности в каждом отдельном случае. Это объясняется перекрёстными издержками каждого метода (затраты времени на служебные операции), что и приводит к этой нелинейности.

Важно заметить и то, что не все алгоритмы успешно поддаются распараллеливанию.

Усиление влияния тех или иных оптимизаций, в зависимости от области применения и объектов моделирования, должно дать больший прирост производительности.

Развитие и совершенствование методов компьютерного моделирования, по мере совершенствования аппаратной базы вычислительных средств, является весьма перспективным направлением исследований в плане достижения в программных технологиях моделирования адекватности виртуальной реальности и окружающего нас мира.

### **Литература**

1. Игровой движок [Электронный ресурс] // Википедия, свободная энциклопедия [сайт]. URL: [http://ru.wikipedia.org/wiki/Игровой\\_движок](http://ru.wikipedia.org/wiki/Игровой_движок) (дата обращения: 10.09.2012).
2. Методы Z-сортировки [Электронный ресурс] // Википедия, свободная энциклопедия [сайт]. URL: [http://ru.wikipedia.org/wiki/Категория:Методы\\_Z-сортировки](http://ru.wikipedia.org/wiki/Категория:Методы_Z-сортировки) (дата обращения: 10.09.2012).
3. Ray casting [Электронный ресурс] // Википедия, свободная энциклопедия [сайт]. URL: [http://ru.wikipedia.org/wiki/Ray\\_casting](http://ru.wikipedia.org/wiki/Ray_casting) (дата обращения: 10.09.2012).
4. Леонид Черняк Многоядерные процессоры и грядущая параллельная революция. – «Открытые системы», № 04, 2007
5. Quake engine [Электронный ресурс] // Wikipedia, the free encyclopedia [сайт]. URL: [http://en.wikipedia.org/wiki/Quake\\_engine](http://en.wikipedia.org/wiki/Quake_engine) (дата обращения: 05.10.2012).

## ***Применение компьютерной графики для визуализации процессов***

### **Аннотация**

*В статье затрагиваются вопросы применения разнообразных технологий для визуализации исследуемых процессов, а также графического представления данных.*

### **Введение**

Известно, что для наглядности исследуемых процессов используются средства представления данных в графическом виде. Кроме того, с помощью компьютерной графики можно создавать и полноценные приложения, такие как всевозможные автомобильные, авиа и другие тренажеры.

В настоящее время существует целое множество различных программных пакетов, осуществляющих поддержку научных исследований и ведения учебного процесса, в том числе, это касается визуализации информации.

Данная статья раскрывает возможности различных технологий, выполняющих упомянутые функции, а также содержит анализ достоинств и недостатков таких технологий.

### **Обзор существующих технологий**

Рассмотрим несколько технологических средств, которые успешно могут быть использованы в качестве вспомогательных при проведении учебного процесса и научных исследований.

В качестве графической поддержки исследований может пониматься не только построение графиков, а также формирование анимации исследуемых процессов и представления данных. Поэтому множество используемого программного обеспечения можно разделить на несколько групп.

Первая группа представляет собой методы и специализированное программное обеспечение, позволяющее строить графики в двумерном и трехмерном виде. Вторая группа представляет собой набор инструментов для формирования двумерной и трехмерной анимации, а третья группа содержит как первые методы, так и вторые.

Остановимся подробнее на рассмотрении представителей указанных групп графических инструментов.

К первой группе следует отнести программные пакеты MathCad, Maple, Mathematica. Основная задача этих программных средств состоит в

обеспечении исследователя необходимыми графическими компонентами и решениями систем.

Вторая группа содержит в себе в основном различные среды программирования и всевозможные графические пакеты, такие как 3D MAX, Blender, CAD системы, с помощью которых возможно создавать не только 2D и 3D модели, но и анимацию.

В состав третьей группы входят некоторые среды программирования, например Microsoft Visual Studio и пакет прикладных программ MATLAB от компании Mathworks.

Наибольший интерес представляет именно третья группа, поэтому перейдем к более подробному рассмотрению особенностей и методов упомянутого программного обеспечения.

### ***Графическая поддержка исследования с помощью пакета MATLAB-Simulink***

Пакет прикладных программ MATLAB [1] содержит в себе две среды разработки – собственно сам MATLAB и подсистема имитационного моделирования Simulink.

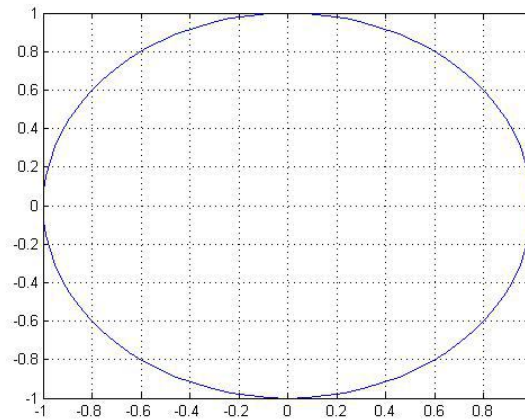
Для решения задач компьютерной графики подходят обе части пакета, однако каждая обладает своими специфическими подзадачами. В среде MATLAB удобно осуществлять построение вспомогательных графиков, и в принципе возможно формирование анимации, хотя для этого необходимо обладать навыками программирования, в частности, условных и циклических конструкций.

Рассмотрим работу этой части пакета на примере. Предположим, исследователь сформировал функцию и теперь необходимо построить ее график:

***Табл. 1. Пример графической функции пакета MATLAB***

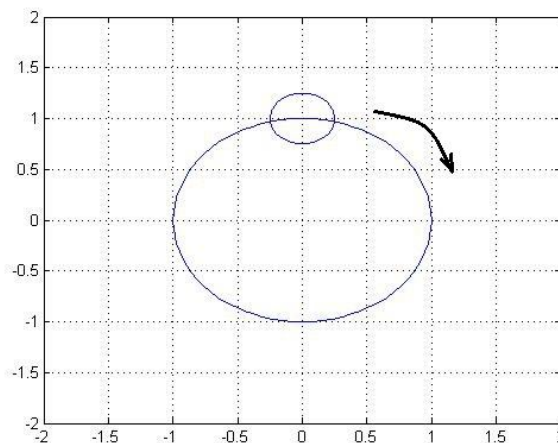
Функция	Команды MATLAB
$x^2 + y^2 = r^2$ Окружность	t=0:pi/40:2*pi; r=1; x=r*sin(t); y=r*cos(t); plot(x,y); grid on;
Две окружности: $x^2 + y^2 = r^2$ $x^2 + y^2 = \frac{r^2}{16}$ вращение по орбите	t=0:pi/40:2*pi; r=1; n=10; x=r*sin(t); y=r*cos(t); for k=0:pi/n:2*pi plot(x,y); grid on; hold on; axis([-2 2 -2 2]) x1=r/4*sin(t); y1=r/4*cos(t);

	<pre> xx=x1+r*sin(k); yy=y1+r*cos(k); plot(xx, yy); pause(.1) hold off; End </pre>
--	--



**Рис. 1.** Статический рисунок в MATLAB

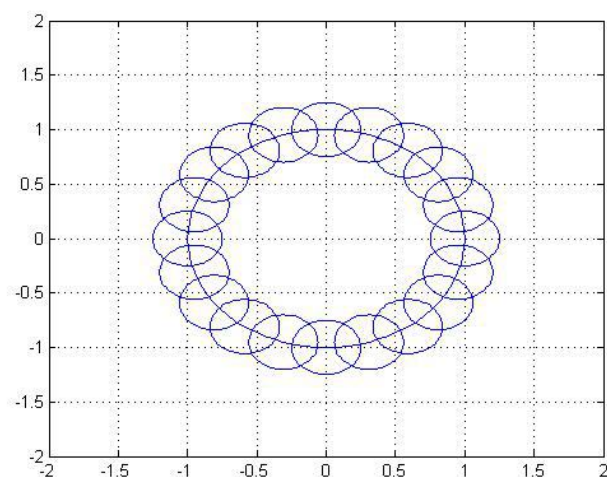
В первом примере строится график окружности. Во втором примере формируются графики двух окружностей, причем здесь присутствует динамика, т.е. одна окружность вращается относительно другой по заданному в программе закону. В примере (на рис. 2) малая окружность движется относительно большей по ее контуру как по орбите.



**Рис. 2.** Движение малой окружности по орбите

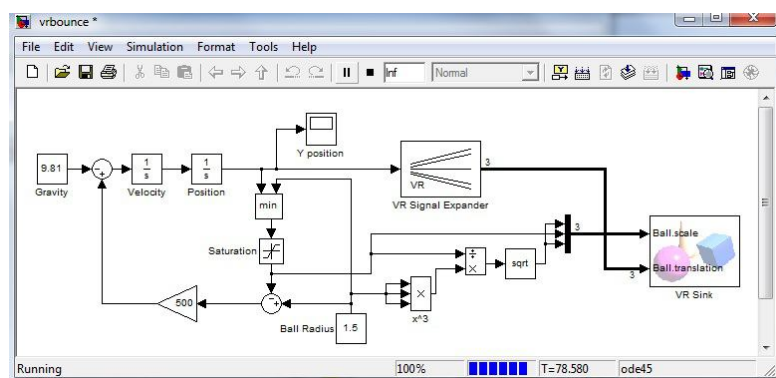
Другая часть пакета, а именно, подсистема Simulink предназначена для имитационного моделирования исследуемых динамических систем и процессов. В состав этой подсистемы входит множество инструментов, один из которых непосредственно предназначен для формирования управляемой анимации в трехмерном виде. Причем основной фактор

удобства использования указанного средства состоит в том, что анимация исследуемого процесса нисколько не помешает исследованиям, а лишь увеличит количество блоков в схеме модели. Этим инструментом является «3D Animation toolbox», подробно рассмотренным в статье [2] (в версиях MATLAB R2008 и ниже инструмент назывался «Virtual Reality toolbox»). В нем с помощью блоков, похожих на схемотехнику можно сформировать динамику любой системы или процесса и создать анимированную модель.



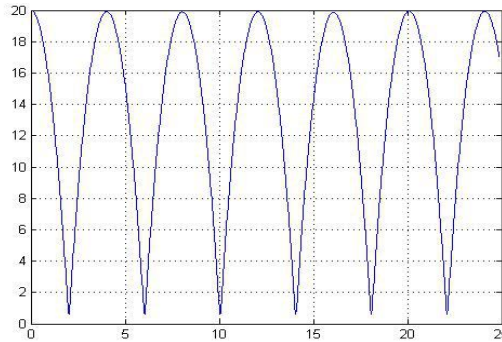
**Рис. 3.** Результат вращения окружности

Пусть задана математическая модель движения прыгающего мяча. Сформируем компьютерную модель движения мяча с помощью подсистемы Simulink с учетом последующего создания анимации, т.е. подключив инструмент «3D Animation toolbox». Компьютерная модель приведена на рис. 4.



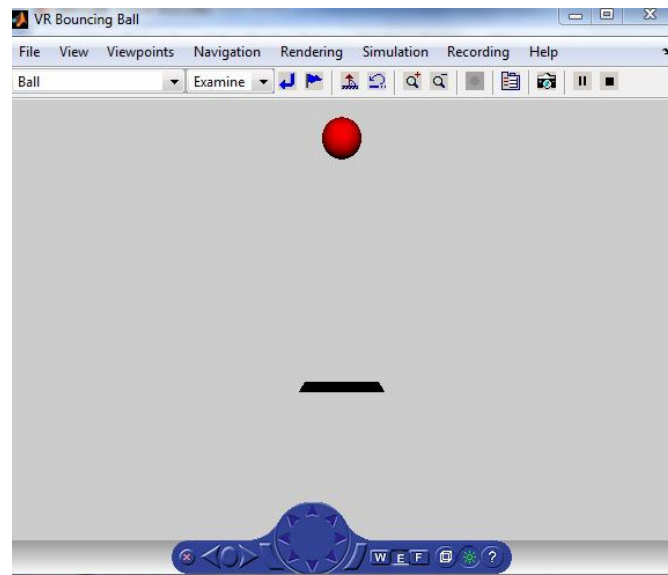
**Рис. 4.** Модель движения прыгающего мяча

График движения мяча со временем представляет собой следующий процесс, который приведен на рис. 5.



**Рис. 5. Форма движения прыгающего мяча**

Одновременно с графиками движения мяча создается окно с анимацией такого движения. Пример окна с анимацией представлен на рис. 6.



**Рис. 6. Модель движения прыгающего мяча**

### **Графическая поддержка среды Microsoft Visual Studio**

Другой подход к формированию графической поддержки исследований предлагает использование среды программирования Microsoft Visual Studio. Здесь доступны различные языки программирования такие, как Visual Basic, C++, C#, Java, F# и другие.

Для формирования графических приложений в подобных средах разработки требуется подключение графических библиотек, таких как Direct3D (DirectX) или OpenGL/AL.

В статье уделим внимание технологии XNA Game Studio Framework [3] на языке C#, которая основана на использовании графической библиотеки DirectX. Указанная технология предназначена в основном для создания игровых приложений для персональных компьютеров, мобильных устройств и игровых приставок. Однако в нашем случае интересны



возможности данной среды формирования анимации и прочей графики.

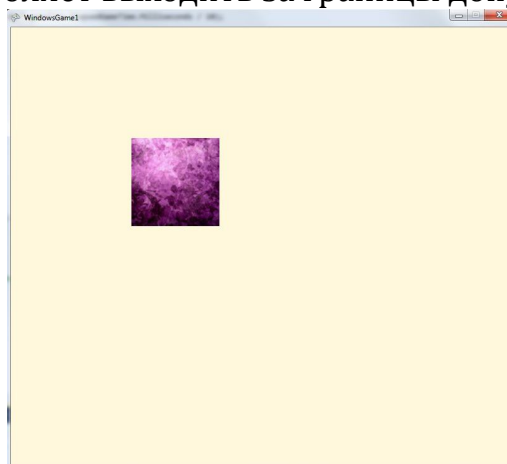
Рассматриваемое средство достаточно удобно, поскольку изначально имеет графическое предназначение. Структура приложения (табл. 2) состоит из пяти групп, каждая из которых имеет свой набор правил и функциональность.

**Табл. 2.** Структура XNA приложения

№	Структура приложения	Предназначение блоков
1	<pre>Public Game1() {graphics = new GraphicsDeviceManager(this); Content.RootDirectory = "Content"; W = graphics.PreferredBackBufferWidth = 800; H = graphics.PreferredBackBufferHeight = 700; rect = new Rectangle(0, 0, 140, 140); }</pre>	Начальная инициализация используемых объектов
2	<pre>protected override void LoadContent() { spriteBatch = new SpriteBatch(GraphicsDevice); spaceTexture = Content.Load &lt;Texture2D&gt; ("quad"); }</pre>	Метод LoadContent загрузки текстур, моделей
3	<pre>protected override void Update(GameTime gameTime) { if (k.IsKeyDown(Keys.Escape)) this.Exit(); rect.Offset(dx*gameTime.ElapsedGameTime. Milliseconds / 10, dy*gameTime.Elapsed GameTime .Milliseconds / 10); if (rect.Left &lt; 0) dx = Math.Abs(dx); if (rect.Top &lt; 0) dy = Math.Abs(dy); if (rect.Right &gt; Width) dx = -Math.Abs(dx); if (rect.Bottom &gt; Height) dy = -Math.Abs(dy); base.Update(gameTime); }</pre>	Метод Update обработки нажатия клавиш и ограничения на движение
4	<pre>protected override void Draw(GameTime gameTime) { GraphicsDevice.Clear(Color.Cornsilk); spriteBatch.Begin(); spriteBatch.Draw(spaceTexture, rect, Color.White); spriteBatch.End(); base.Draw(gameTime); }</pre>	Метод Draw отрисовки объектов

Приведенная в таблице 2 структура кода программы осуществляет отрисовку квадратного объекта (рис. 7) и осуществляет его перемещение в

пространстве, но не позволяет выходить за границы допустимой области.



**Рис. 7.** Движение квадрата ударяющегося о границы области

Однако кроме создания анимации в XNA Framework можно создавать массивы данных или импортировать информацию из баз данных. Поэтому можно сказать, что указанная технология позволяет визуализировать данные. Например, можно из какой-либо базы данных сохранить данные, записать их в виде структуры, например, дерева данных и затем визуализировать с помощью предлагаемой среды. Пример такого приложения приведен на рис. 8.



**Рис. 8.** Графическое представление данных

#### Преимущества и недостатки средств графической поддержки

В качестве основного достоинства системы MATLAB-Simulink следует отметить возможность визуализации процесса и сопутствующих данных, непосредственно исследуемых в рамках рассматриваемой работы. Однако в системе имеются и недостатки. Наиболее заметный недостаток состоит в отсутствии обратной связи с графическим приложением в системе MATLAB и некоторые трудности при реализации обратной связи в подсистеме Simulink. В этих системах практически невозможно управлять с клавиатуры какой-либо анимацией.

Тем временем в среде разработки Visual Studio и ее графической подсистеме XNA Game Studio Framework возможность управления анимацией присутствует изначально, что является неоспоримым достоинством этой системы. Кроме того, получаемая анимация является отдельным самостоятельным приложением. Тем не менее, есть в системе и недостатки, в частности, это касается необходимости описания самих

объектов и физики их движения, т.е. здесь нет возможности воспользоваться готовой математической моделью, а придется все законы движения описывать сначала.

### **Заключение**

Все рассмотренные средства графической поддержки могут быть успешно применены в различных ситуациях. Если нужно построить статические или динамические графики, то следует воспользоваться средствами системы MATLAB. Если необходимо посмотреть как будет вести себя динамическая система, описанная конкретной математической моделью, то можно прибегнуть к помощи подсистемы имитационного моделирования Simulink с ее графическим инструментом «3D Animation toolbox». В случае, если требуется создать самостоятельное графическое приложение с возможностью управления с клавиатуры, мышью или другим устройством ввода, тогда следует обратить внимание на среду разработки Microsoft Visual Studio вместе с XNA Game Studio Framework.

### **Литература**

1. Веремей Е.И. Система MATLAB в учебном процессе для специалистов по теории управления и информационным технологиям. //Тр. Первой международной научно-практической конференции «Современные информационные технологии и ИТ-образование». – М.:МАКС Пресс, 2005. – С. 516–523.
2. Лепихин Т.А. 3D-анимация на базе пакета Virtual Reality Toolbox. Труды V Международной научной конференции «Проектирование инженерных и научных приложений в среде MATLAB», Украина, Харьков, 2011. С. 397–403.
3. Kurt Jaegers XNA 4.0 Game Development by Example Beginner's Guide. Packt Publishing Ltd. ISBN: 978-1-84969-066-9, 2010, p. 428.

## ***Задача уточнения результатов измерений путем согласования с априорными связями***

### ***Введение***

Измеряемая информация в различных исследовательских задачах представляется в виде числовых рядов в массивах данных, записанных с некоторым шагом дискретизации по времени. Между измеряемыми переменными (числовыми рядами) часто существует априорная избыточность в виде имеющихся связей между ними (уравнений). Эта избыточность может быть использована для уточнения и корректировки измеренных данных, как указывается в работе [3].

Измеренные массивы данных геометрически должны находиться в некотором подмножестве, определяемом связями между имеющимися переменными. Например, множество значений трех переменных при отсутствии связей между ними составляет трехмерное пространство, наличие одной связи определяет уже такое двумерное подмножество как поверхность, а при двух связях одномерную линию пересечения двух поверхностей. Если значения некоторых измерений находятся за пределами подмножества, то перемещение таких точек в сторону этого подмножества, несомненно приблизит их значения к истинным.

Чаще всего связи между рядами переменных бывают представлены в виде алгебраических и дифференциальных уравнений. Последний вид связей требует более подробного рассмотрения. Геометрически общим решением обыкновенного дифференциального уравнения (ОДУ) первого порядка является семейство функций одной переменной. Из этого семейства можно выделить одну из функций  $y(t)$  (частное решение), которое определяется как исходящее из некоторой начальной точки  $\{t_0, y_0\}$  (задача Коши), или как удовлетворяющее значениям  $y_i$  при различных значениях  $t_i$ , чаще концевых (краевая задача).

Дальнейшее изложение посвящено выявлению дифференциальных связей, присутствующих в математической модели полета некоторого объекта и возможному подходу к разработке алгоритмов согласования измеренных данных с априорными дифференциальными связями.

### ***Моделирование полета***

При традиционном математическом моделировании движения некоторого объекта обычно предполагается, что составляющая его система материальных точек представляет собой твердое тело (недеформируемая конструкция). Поскольку расстояния между любыми точками твердого

тела сохраняются неизменными, то движение такой системы материальных точек, сводится к сумме двух движений:

- перемещение центра масс твердого тела (на вектор  $\mathbf{r}$ )
- вращение точек тела вокруг центра масс (вокруг вектора угловой ориентации  $\Phi$ )

Из теоретической механики известно [1], что в *инерциальной* (неускоренной) системе отсчета, которая находится в состоянии покоя, или движется равномерно и прямолинейно, уравнения движения твердого тела в векторной форме принимают наиболее простой вид. На практике чаще используются уравнения движения, записанные в проекционной системе координат (СК), которая уже не является инерциальной. Например, в СК, фиксированной относительно твердого тела, уравнение движения центра масс под действием суммарного вектора внешних сил  $\mathbf{F}$  и уравнение вращения вокруг центра масс под воздействием суммарного момента  $\mathbf{M}$ , для системы из  $k$  материальных точек с массами  $m_k$ , скоростями  $\mathbf{V}_k$ , на расстояниях  $\mathbf{r}_k$  от центра масс примут следующий вид:

$$\frac{d\mathbf{p}}{dt} + \boldsymbol{\omega} \times \mathbf{p} = \mathbf{F} \qquad \frac{d\mathbf{L}}{dt} + \boldsymbol{\omega} \times \mathbf{L} = \mathbf{M} \quad (1)$$

где  $\mathbf{p} = \sum m_k \cdot \mathbf{V}_k$  – вектор количества движения системы точек

$\mathbf{L} = \sum \Delta \mathbf{r}_k \times m_k \mathbf{V}_k$  – вектор момента количества движения

$\boldsymbol{\omega}$  – вектор скорости вращения системы точек (или СК)

Для однозначного определения положения твердого тела, систему динамических уравнений (1) необходимо замкнуть кинематическими уравнениями, связывающими векторы движения  $\mathbf{V}$  и  $\boldsymbol{\omega}$  этого тела (или подвижной СК) с векторами  $\mathbf{r}$  и  $\Phi$  его положения относительно инерциальной СК:

$$\frac{d\mathbf{r}}{dt} = \mathbf{V} \qquad \frac{d\Phi}{dt} = \boldsymbol{\omega} \quad (2)$$

Если воспользоваться сведениями о регламентированных в ГОСТ 20058-80 [2] системах координат, на которые преимущественно проецируются векторные компоненты уравнений движения летательных аппаратов, то в качестве инерциальной системы отсчета следует выбрать систему координат связанную с поверхностью Земли, оси которой фиксированы относительно этой поверхности. Такая система отсчета участвует в суточном вращении Земли вокруг своей оси и в годовом вращении вокруг Солнца. Однако порядок перегрузок, обусловленных влиянием этих вращений, весьма мал, даже для гиперзвукового ВС. В качестве проекционной (неинерциальной), традиционно выбирается *связанная* СК, неподвижно фиксированная относительно движущегося тела.

Проекциями векторных компонентов уравнений (1) и (2) на оси указанных СК будут являться следующие 12 переменных (придерживаясь в дальнейшем обозначений стандарта ГОСТ 20058-80):

$V_{kx}, V_{ky}, V_{kz}$  – проекции вектора  $\mathbf{V}_k$  земной скорости ВС

на оси связанной СК;  
 $\omega_x, \omega_y, \omega_z$  – проекции вектора  $\omega$  скорости вращения ВС  
на оси связанной СК;  
 $L, H, Z$  – проекции вектора  $r$  положения ВС  
на оси нормальной земной СК;  
 $\psi, \vartheta, \gamma$  – проекции вектора  $\phi$  угловой ориентации ВС  
на промежуточные оси

Векторы  $p$  и  $L$  желательно определить как усредненные для всей совокупности точек объекта. Можно показать, что исходя из понятия центра масс, определяемого как  $r = \sum m_k \cdot r_k / \sum m_k$  и используя фиксированное положение связанной СК, эти векторы приводятся к интегральному для всей недеформируемой системы точек виду [1]:

$$p = m \cdot V \quad L = I \cdot \omega \quad (3)$$

где  $m$  – общая масса системы материальных точек

$V$  – вектор абсолютной скорости центра масс

$I$  – тензор инерции

Для симметричного по оси  $Z$  объекта:

$$I = \begin{pmatrix} I_x & -I_{xy} & 0 \\ -I_{yx} & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} = \begin{pmatrix} \int_m (y^2 + z^2) dm & -\int_m xy \cdot dm & 0 \\ -\int_m yx \cdot dm & \int_m (x^2 + z^2) dm & 0 \\ 0 & 0 & \int_m (x^2 + y^2) dm \end{pmatrix}$$

Здесь  $x, y, z$  – координаты точки  $dm$  в связанной СК

Внешние силы  $F$  и моменты  $M$ , составляющие правые части уравнений (5), должны включать в себя как минимум силу тяжести  $G$ , силы и моменты от двигателей –  $P$  и  $M_p$ , а также от аэродинамического воздействия –  $R_A$  и  $M$ . Чтобы не загромождать выкладки, желательно отдельно выделить силу тяжести, заданную в нормальной СК –  $G = \{0, -mg, 0\}$ , а остальные внешние силы и моменты представить в общем виде, обозначив их в соответствии с ГОСТ 20058-80 как:

$$R = P + R_A \quad \text{и} \quad M_R = M_p + M.$$

При переходе к другому базису – новые значения координат вектора выражаются через старые путем их линейного преобразования (отображения). Наиболее приспособленным средством для линейных операций, является аппарат матричного исчисления, в котором наиболее важной является операция умножения матриц.

При выводе рабочих формул преобразований координат [3] необходимо отметить следующие особенности матричных операций:

1. В трехмерном пространстве три координаты (компоненты)  $x, y, z$  вектора могут быть представлены в виде матрицы-строки или матрицы-столбца. Линейное преобразование матрицы-столбца

соответствует умножению его справа на матрицу преобразования. Таким образом, матрица каждого последующего преобразования становится левым сомножителем. В случае матрицы-строки производится ее умножение слева уже на транспонированную матрицу преобразования, являющуюся уже правым сомножителем.

2. Неважно, в какой последовательности выполняется произведение матриц, однако важен их порядок в этом произведении. В общем случае произведение  $\mathbf{A} \times \mathbf{B}$  не равно  $\mathbf{B} \times \mathbf{A}$ , вследствие чего важно четко соблюдать очередность выполнения поворотов.
3. Перемещение или поворот системы координат относительно неподвижного объекта эквивалентен переносу или повороту объекта относительно неподвижной СК в обратную сторону.
4. Желательно также выразить векторное произведение двух векторов, например  $\mathbf{a}$  и  $\mathbf{b}$ , через произведение матриц. Такую матрицу несложно подобрать, исходя из координатного определения векторного произведения:

$$\mathbf{a} \times \mathbf{b} = \begin{pmatrix} a_y \cdot b_z - a_z \cdot b_y \\ a_z \cdot b_x - a_x \cdot b_z \\ a_x \cdot b_y - a_y \cdot b_x \end{pmatrix} = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix} \cdot \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}$$

В результате преобразований и выкладок получена система ОДУ в компактной матричной форме, приведенная к виду, в котором в левых частях находятся только матрицы-столбцы производных:

Уравнения поступательного движения центра масс

$$\begin{pmatrix} dV_{kx}/dt \\ dV_{ky}/dt \\ dV_{kz}/dt \end{pmatrix} = \begin{pmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{pmatrix} \cdot \begin{pmatrix} V_{kx} \\ V_{ky} \\ V_{kz} \end{pmatrix} + g \cdot \begin{pmatrix} -\sin \vartheta \\ -\cos \vartheta \cos \gamma \\ \cos \vartheta \sin \gamma \end{pmatrix} + \frac{1}{m} \cdot \begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix} \quad (4)$$

$$\begin{pmatrix} dL/dt \\ dH/dt \\ dZ/dt \end{pmatrix} = \begin{pmatrix} \cos \psi & 0 & \sin \psi \\ 0 & 1 & 0 \\ -\sin \psi & 0 & \cos \psi \end{pmatrix} \cdot \begin{pmatrix} \cos \vartheta & -\sin \vartheta & 0 \\ \sin \vartheta & \cos \vartheta & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{pmatrix} \cdot \begin{pmatrix} V_{kx} \\ V_{ky} \\ V_{kz} \end{pmatrix} \quad (5)$$

Уравнения вращения вокруг центра масс

$$\begin{pmatrix} d\omega_x/dt \\ d\omega_y/dt \\ d\omega_z/dt \end{pmatrix} = \frac{1}{(I_x I_y - I_{xy}^2)} \cdot \begin{pmatrix} I_y & I_{xy} & 0 \\ I_{xy} & I_x & 0 \\ 0 & 0 & \frac{(I_x I_y - I_{xy}^2)}{I_z} \end{pmatrix} \cdot \left\{ \begin{pmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{pmatrix} \cdot \begin{pmatrix} I_x & -I_{xy} & 0 \\ -I_{xy} & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} \cdot \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} + \begin{pmatrix} M_{Rx} \\ M_{Ry} \\ M_{Rz} \end{pmatrix} \right\} \quad (6)$$

$$\begin{pmatrix} d\psi/dt \\ d\vartheta/dt \\ d\gamma/dt \end{pmatrix} = \begin{pmatrix} 0 & \frac{\cos \gamma}{\cos \vartheta} & -\frac{\sin \gamma}{\cos \vartheta} \\ 0 & \sin \gamma & \cos \gamma \\ 1 & -\frac{\sin \vartheta \cos \gamma}{\cos \vartheta} & \frac{\sin \vartheta \sin \gamma}{\cos \vartheta} \end{pmatrix} \cdot \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (7)$$

Окончательно полученная система ОДУ имеет инвариантный вид, пригодный для описания баллистики движения любого твердого тела (транспортного средства) в проекциях на связанную систему координат. Различия между ними будут определяться только их массами, моментами инерции и действующими на них силами и моментами.

### **Согласование измерений**

Искомым решением матричной системы ОДУ (4) – (7), является одно из его частных решений – вектор-функция  $\mathbf{y}(t)$  с компонентами  $\{V_x, V_y, V_z, \omega_x, \omega_y, \omega_z, L, H, Z, \psi, \vartheta, \gamma\}$ , определяемая их начальными значениями при  $t = t_0$  –  $\{V_{x0}, V_{y0}, V_{z0}, \omega_{x0}, \omega_{y0}, \omega_{z0}, L_0, H_0, Z_0, \psi_0, \vartheta_0, \gamma_0\}$ . Можно сказать, что 12 дифференциальных связей, наложенных на 13-мерное пространство (плюс переменная  $t$ ), определяют в нем одномерное подмножество в виде некоторой линии.

При наличии измерений всех 12 переменных (12 числовых рядов), алгоритм их связывания с 12 уравнениями определялся бы следующим естественным образом. Систему ОДУ 1-го порядка (4) – (7) можно записать в следующем векторном виде:

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}) \quad \text{или} \quad \mathbf{y}(t_k) = \mathbf{y}_0 + \int_{t_0}^{t_k} \mathbf{f}(t, \mathbf{y}) dt \quad (8)$$

где  $\mathbf{y}(t_k)$  – значения вектор-функции  $\mathbf{y}(t)$  в точках  $t_k$

Измеренные значения вектора  $\mathbf{y}_k$  должны совпадать с полученными путем численного интегрирования –  $\mathbf{y}(t_k)$  по формуле (8). В общем случае следует проводить интегрирование последовательно, порциями точек  $t_k$ . (например, воспользовавшись алгоритмом [5]).

Задача подбора (корректирования) значений  $\mathbf{y}_k$ , является прежде всего задачей оптимизации.. В качестве критериев близости уточняемых значений вектора  $\mathbf{y}_k$  к точному решению системы ОДУ –  $\mathbf{y}(t_k)$ , могут служить следующие два, применяемых в вычислительной практике:

- метод коллокаций, состоящий в точном выполнении равенства нулю длины вектора невязки  $\Delta \mathbf{y}_k = \mathbf{y}_k - \mathbf{y}(t_k)$  в точках  $t_k$  ( $k = 0, 1, .. m$ ).
- метод наименьших квадратов, вытекающий из условия минимума суммы квадратов всех невязок на заданной системе точек  $t_k$  (суммы скалярных произведений векторов невязки):

$$\min_{\mathbf{y}_k} \sum_k (\Delta \mathbf{y}_k \cdot \Delta \mathbf{y}_k)$$

На практике все обстоит гораздо сложнее. Представления сил и моментов динамических уравнений (их сложность и полнота), включающих в себя как минимум силы и моменты от двигателя и аэродинамического воздействия, зависят от характера решаемой задачи. Построение достаточно полной и адекватной модели сил и моментов – чрезвычайно трудная задача.

Если использовать понятие перегрузки  $\mathbf{n}$ , которая представляет собой отношение вектора внешних сил к весу объекта



$$\mathbf{n} = \frac{\mathbf{R}}{mg} \quad \text{отсюда} \quad \begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix} = mg \cdot \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix},$$

то при наличии ее проекций в перечне измеряемых параметров, уравнения (4) следует привести к виду:

$$\begin{pmatrix} dV_{kx}/dt \\ dV_{ky}/dt \\ dV_{kz}/dt \end{pmatrix} = \begin{pmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{pmatrix} \cdot \begin{pmatrix} V_{kx} \\ V_{ky} \\ V_{kz} \end{pmatrix} + g \cdot \begin{pmatrix} n_x - \sin \vartheta \\ n_y - \cos \vartheta \cos \gamma \\ n_z + \cos \vartheta \sin \gamma \end{pmatrix} \quad (9)$$

Из всех полученных матричных уравнений наиболее точными являются кинематические (5) и (7). В уравнениях (4) и (6) изначально заложена неопределенность из-за сложности и громоздкости описания действующих на объект сил и моментов. Поэтому в качестве дифференциальных связей в первую очередь следует задействовать кинематические уравнения (5) и (7), а также (9). От использования матричного уравнения (6), включающего в себя проекции вектора моментов, лучше отказаться.

Кроме того, некоторые из кинематических переменных вообще не измеряются, например,  $L$ ,  $H$ ,  $Z$  или выражаются опосредствовано через другие параметры (как  $\psi$ ,  $\vartheta$ ,  $\gamma$ ).

После решения этих вопросов, рассматриваемый подход к повышению точности оценивания измеренных координат и параметров движения ВС (или другого объекта) может найти применение в следующих задачах:

В задачах инерциальной навигации для уточнения вычисляемых параметров движения объекта в режиме реального времени.

При расследованиях летных происшествий для восстановления и уточнения записанных параметров движения ВС и предполагаемой траектории.

## Литература

1. Халфман Р. Динамика: Пер.с англ. –М.: Наука. Гл. ред. физ.-мат. литер., 1972. – 568с.
2. ГОСТ 20058-80. Динамика летательных аппаратов в атмосфере. Термины, определения и обозначения. – М.: Изд-во стандартов, 1981. – 53 с.
3. Касьянов В.А. Моделирование полета: Монография. –К.: НАУ, 2004.– 400 с.
4. Митюков В.В., Извольский И.В. Методика преобразования координат при моделировании движения твердого тела. // Научно-технический журнал «Автоматизация процессов управления». – Ульяновск : НПО «Марс», 2010. – № 4 (22). – с. 16-21.
5. Митюков В.В. Унифицированный алгоритм дифференцирования и интегрирования дискретных числовых массивов. «Современные информационные технологии и ИТ-образование». Сборник докладов V междунар. научно-практич. конференции: учебно-методическое пособие. Под ред. проф. В.А. Сухомлина. – М.: ИНТУИТ.РУ, 2010. – с. 509-513.

**Периг А.В.,**

Донбасская государственная машиностроительная академия (ДГМА),  
г. Краматорск, ст. преп.  
[olexander.perig@gmail.com](mailto:olexander.perig@gmail.com)

**Голоденко Н.Н.,**

Донбасская национальная академия строительства и архитектуры,  
доц.  
[nik\\_nik\\_Gold@mail.ru](mailto:nik_nik_Gold@mail.ru)

**Бойко И. И.**

ДГМА, студент  
[bojkoiva@yandex.ru](mailto:bojkoiva@yandex.ru)

***О расчетной гидродинамической оценке влияния входной подвижной стенки штампа на кинематические особенности вязкого течения полимерного материала при равноканальном угловом прессовании***

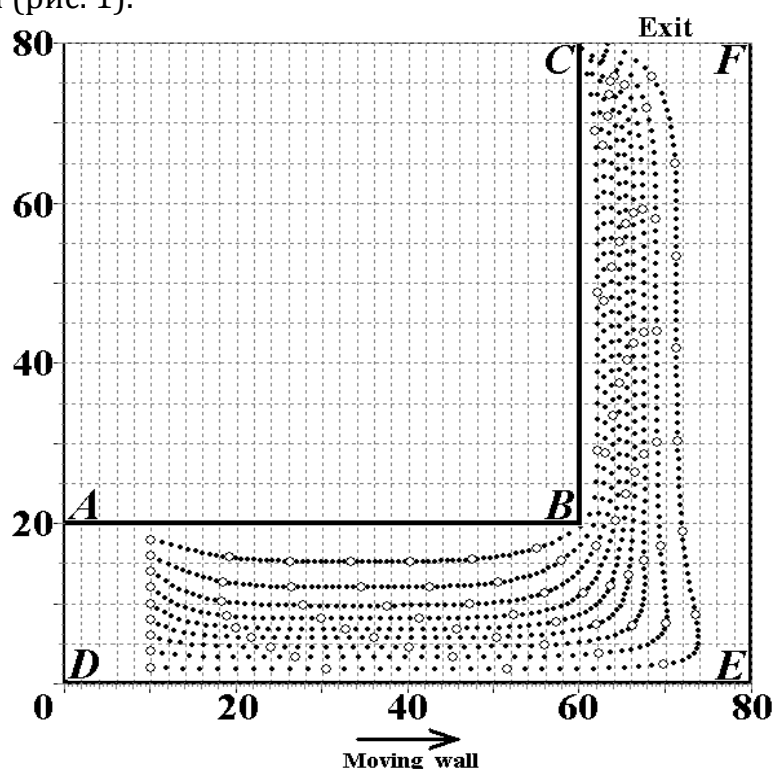
**Аннотация**

*В рамках реализации учебного программного обеспечения для интенсификации научно-исследовательской студенческой работы будущих инженеров-металлургов и инженеров-химиков поставлена и численно решена краевая задача для уравнений Навье-Стокса в форме уравнений переноса вихря, моделирующая процесс равноканального углового прессования вязкой сплошной среды в угловом штампе с подвижной входной стенкой. Полученные численные результаты представлены в виде плоских и пространственных эпюр для линий тока, полей скоростей и напряжений. Разработанное программное обеспечение демонстрирует возможности гидродинамического моделирования и расширяет представления целевой аудитории о динамике процессов интенсивного деформирования материалов в металлургии и химической технологии.*

**Введение**

К особенностям преподавания ВТУЗовского курса информатики для будущих инженеров-химиков и инженеров-металлургов можно отнести значительную интегрированность данной дисциплины с курсами прикладной математики, математического моделирования, реологии и механики сплошных сред на фоне непрерывного сокращения аудиторных часов, что обуславливает актуальность разработки учебного программного обеспечения, обеспечивающего связное изложение основных концепций всех упомянутых дисциплин. В качестве расчетного примера, иллюстрирующего особенности предлагаемого нами подхода, рассмотрим

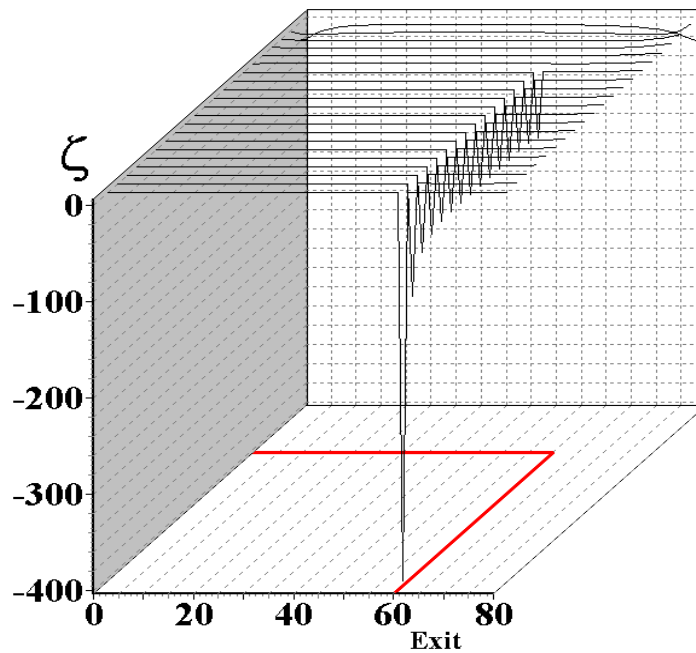
задачу о равноканальном угловом прессовании (РКУП) материалов в постановке которой требуется рассчитать энергосиловые параметры данного технологического процесса при непрерывном перетекании материала из входного в выходной каналы углового штампа с одинаковой шириной канала [1]-[2]. При этом отметим недостаточную изученность в литературе ([1]-[2]) эффектов влияния подвижных стенок углового штампа на динамику процессов РКУП при деформировании полимерных материалов. В связи с этим представляется актуальной постановка и решение краевой задачи для уравнений Навье-Стокса в форме уравнений переноса вихря (УПВ), позволяющей выполнить гидродинамическую оценку эффекта влияния входной подвижной стенки углового штампа на кинематические особенности вязкого течения несжимаемой ньютоновской сплошной среды (рис. 1).



**Рис. 1. Расчетное поле линий тока**

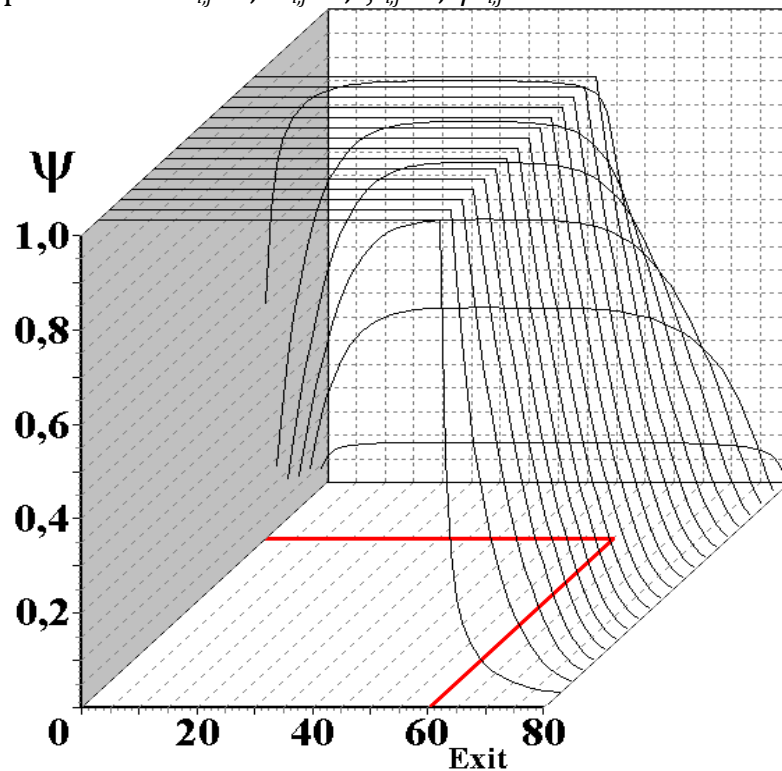
### **Математическая модель технологического процесса**

Построение гидродинамической модели процесса РКУП в равноканальном угловом штампе с подвижной входной стенкой *DE* свяжем с численным интегрированием краевой задачи для УПВ в безразмерной форме (рис. 1). Безразмерное УПВ может быть записано в виде ДУЧП II порядка  $\partial \zeta / \partial t = -\mathbf{Re} \cdot (\partial(u\zeta) / \partial x + (\partial(v\zeta) / \partial y) + \partial^2 \zeta / \partial x^2 + \partial^2 \zeta / \partial y^2; \zeta = \partial u / \partial y - \partial v / \partial x$ , где  $\zeta$  (рис. 2) и  $\psi$  (рис. 3) – безразмерные функции вихря и тока;  $u, v$  – безразмерные  $x$ -,  $y$ - проекции скорости  $w$  вязкого потока модели деформируемого материала (рис. 4);  $x, y$  – безразмерные координаты;  $\mathbf{Re}$  – число Рейнольдса;  $t$  – безразмерное время.



**Рис. 2.** Расчетное поле функции вихря

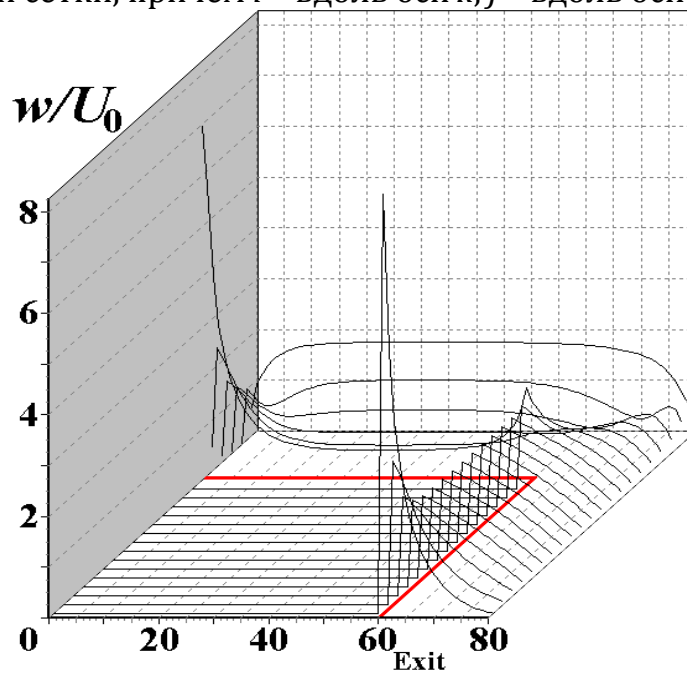
Поскольку рассматривается установившийся режим вязкого течения физической модели полимерного материала через прямоугольный угловой штамп  $ABC-DEF$  с входной подвижной стенкой  $DE$  (рис. 1), то начальные условия для УПВ могут приниматься в виде грубого приближения к стационарному решению:  $u^0_{ij}=0$ ;  $v^0_{ij}=0$ ;  $\zeta^0_{ij}=0$ ;  $\psi^0_{ij}=0$ .



**Рис. 3.** Расчетное поле функции тока

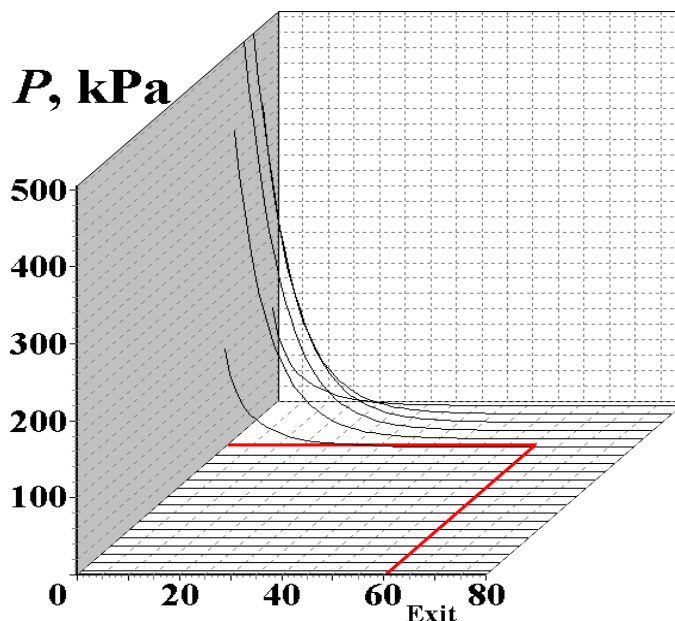
Граничные условия для УПВ запишем с учетом физического явления прилипания вязкого материала к стенкам  $ABC-DEF$  углового штампа при

РКУП (рис. 1): для левой границы потока  $ABC$  имеем узловые значения функции тока  $\psi_{ij}=0$ ; для правой границы потока  $DEF$   $\psi_{ij}=1$ ; для участков стенок  $AB$  и  $EF$  узловые значения функции вихря  $\zeta_{ij}=2\bullet(\psi_{ij-1} - \psi_{ij})/\eta^2$ ; для участка  $BC$  имеем  $\zeta_{ij}=2\bullet(\psi_{i+1,j}-\psi_{ij})/\xi^2$ ; для участка подвижной стенки  $DE$ , движущегося со скоростью  $U_b$ ,  $\zeta_{ij}=2\bullet(\psi_{ij+1} - \psi_{ij} - \eta(U_b/U_0))/\eta^2$ ; для угловой точки  $E$ , лежащей в вершине вогнутого угла,  $\zeta_{ij}=0$ ; для угловой точки  $B$  в разностном уравнении, записанном для узла  $(i, j - 1)$ ,  $\zeta_{ij}= 2\bullet\psi_{ij-1}/\eta^2$ , а в разностном уравнении, записанном для узла  $(i + 1, j)$ ,  $\zeta_{ij}= 2\bullet\psi_{i+1,j}/\xi^2$ ; на входе  $AD$ :  $\psi_{0,j}=\psi_{1,j}$ ,  $u_{0,j}=u_{1,j}$ ,  $v_{0,j}=0$ ; на выходе  $CF$  имеем:  $\psi_{i,n}=\psi_{i,n-4} - 2\bullet\psi_{i,n-3}+\psi_{i,n-1}$ ,  $\zeta_{i,n}=\zeta_{i,n-4} - 2\bullet\zeta_{i,n-3}+\zeta_{i,n-1}$ ,  $u_{i,n}=0$ ; где  $\xi$  – безразмерный шаг координаты вдоль оси  $x$ ;  $\eta$  – безразмерный шаг координаты вдоль оси  $y$ ;  $i, j$  – номера ячеек конечноразностной сетки, причем  $i$  – вдоль оси  $x$ ,  $j$  – вдоль оси  $y$ .



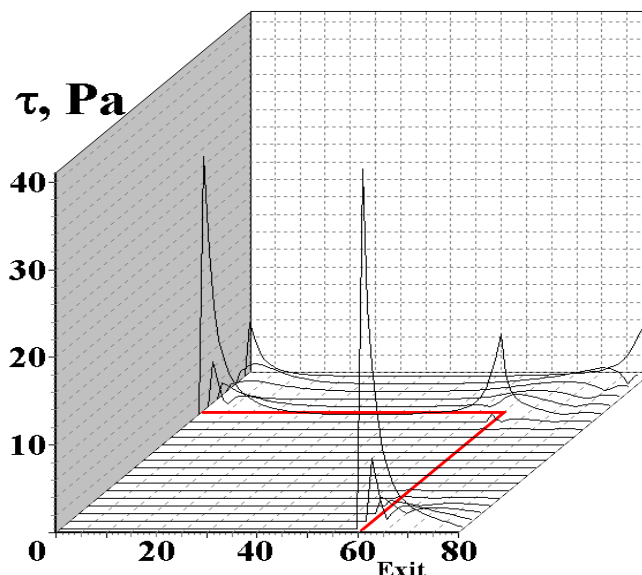
**Рис. 4.** Модуль относительной скорости вязкого потока

На рис.1 – рис.6 приведены результаты численного интегрирования УПВ с вышеприведенными начальными условиями для стационарного вязкого течения и граничными условиями для полного прилипания вязкого материала к стенкам углового штампа. При этом размерная характерная скорость прессования материала составляет  $\underline{U}_0=270 \text{ } \mu\text{m/s}$ . Подвижная входная стенка  $DE$  штампа движется в направлении экструдированного вязкого потока со скоростью  $\underline{U}_b=2\bullet\underline{U}_0$ , вдвое большей характерной скорости потока  $\underline{U}_0$ .



**Рис. 5.** Расчетное поле давления прессования

Расчетные эпюры на рис.1 – рис.6 определены для следующих числовых значений параметров: ширина каждого канала углового штампа  $a=40 \text{ mm}$ ; плотность вязкой пластилиновой модели полимера  $\rho=1850 \text{ kg/m}^3$ ; предел текучести пластилина  $\sigma_s=217 \text{ kPa}$ ; удельные теплоемкость  $c=1.004 \text{ kJ/(kg}\cdot\text{K)}$  и теплопроводность  $\lambda=0.7 \text{ J/(m}\cdot\text{s}\cdot\text{K)}$ ; кинематическая вязкость  $\nu_{vis}=0.073 \text{ m}^2/\text{s}$ ; число Рейнольдса для вязкой модели  $Re = \frac{U_0 a}{\nu_{vis}} = 1.48 \cdot 10^{-4}$ . Число шагов координатной сетки по горизонтали 80, по вертикали – 100, шаг координаты 2 mm. Относительная погрешность итераций составляет 0,001.



**Рис. 6.** Расчетное поле касательных напряжений

#### **Обсуждение результатов численного моделирования**

Неустойчивости решения, зарождающиеся на входной границе  $AD$  (рис. 1), распространяются вниз по потоку, а неустойчивости, зарождающиеся на выходной границе  $CF$  – вверх по потоку. Численное

решение УПВ даёт результаты, согласующиеся с физическим экспериментом, лишь в том случае, когда вход  $AD$  и выход  $CF$  потока находятся достаточно далеко от интересующей нас области  $ABDE$ , в идеале на бесконечности. Начальные точки для построения линий тока (рис. 1) берутся не у границы координатной сетки, соответствующей входу  $AD$  потока ( $i = 0$ ), а на некотором расстоянии от неё с одинаковым шагом  $a/10$  по вертикали. Точно так же следует отбросить ячейки сетки у выхода  $CF$ . Расчетные всплески на пространственных эпюрах на рис. 4 и рис. 6 показывают формирование физических зон интенсивной макроскопической ротации в объеме вязкого деформируемого континуума.

### **Выводы**

В рамках разработки учебного программного обеспечения для организации научно-исследовательской студенческой работы будущих инженеров-металлургов и инженеров-химиков был предложен и реализован численный гидродинамический подход к анализу вязкого течения аморфного вязкого материала при РКУП через угловой штамп с подвижной входной стенкой  $DE$ . Предложенный алгоритм основан на численном конечноразностном решении краевых задач для уравнений Навье-Стокса в форме уравнений переноса вихря. Разработанный алгоритм описывает установившиеся плоские течения вязких несжимаемых ньютоновских жидкостей в угловых штампах с подвижной стенкой, а также корректно учитывает влияние входного и выходного каналов штампа. В рамках предложенного численного подхода учет наличия подвижной входной стенки  $DE$  углового штампа, движущейся параллельно направлению угловой экструзии со скоростью  $U_b$ , реализован в виде соответствующего граничного условия для данной стенки  $DE$ . Предложенное граничное условие содержит скорость  $U_b$  и записано для безразмерной функции вихря  $\zeta$ , относящейся к узлам разностной сетки, принадлежащих подвижной стенке  $DE$ .

Разработанное программное обеспечение находит применение в рамках изложения общих и специализированных курсов классической механики, гидродинамики и математического моделирования для студентов металлургических и машиностроительных специальностей Донбасской государственной машиностроительной академии (ДГМА, г. Краматорск, Украина).

### **Литература**

1. Периг А.В. О динамике вязкого течения аморфных материалов при равноканальном угловом прессовании // VIII Российская ежегодная конференция молодых научных сотрудников и аспирантов «Физико-химия и технология неорганических материалов»: сборник материалов. М.: ИМЕТ РАН, 2011. С. 660–662.
2. Периг А.В., Голоденко Н.Н., Жбанков Я.Г. и др. Кинематические особенности вязкого течения аморфного материала при равноканальном многоугловом прессовании через двухповоротный прямоугольный штамп // Письма о материалах. 2011. Т. 1. № 4. С. 217–221.

## О специальных многомерных сплайнах лагранжевого типа

1. Точки  $x_0, x_1, \dots, x_n \in E^n$  такие, что векторы  $\Delta x_1, \dots, \Delta x_n$  линейно независимы (где  $\Delta x_j \doteq x_j - x_0$ ), порождают симплекс, который будем обозначать в виде  $\langle x_0, x_1, \dots, x_n \rangle$ .

Через  $\{e_1, \dots, e_n\}$  обозначим стандартный базис в  $E^n$ . (Заметим, что векторы  $\Delta x_1, \dots, \Delta x_n$  также образуют базис этого пространства.) Очевидно, квадратная матрица  $X = (X_{kj})$  порядка  $n$ , состоящая из скалярных произведений  $X_{kj} \doteq (e_k, \Delta x_j)$ , обратима, то есть существует матрица  $Y = (Y_{ik})$  такая, что  $YX = E_n = XY$  (где  $E_n$  – это единичная матрица порядка  $n$ ). Следовательно, для всех  $i, j \in K$  справедливы равенства

$$\sum_{k \in K} Y_{ik} X_{kj} = \sum_{k \in K} Y_{ik} (e_k, \Delta x_j) = \delta_{ij}, \quad \sum_{i \in K} X_{ij} Y_{ik} = \sum_{i \in K} (e_i, \Delta x_j) Y_{ik} = \delta_{ij},$$

где  $K \doteq \{1, \dots, n\}$ ,  $\delta_{ij}$  – символ Кронекера. Для любого  $i \in K$  определим функцию  $\varphi_i(\xi) = \sum_{k \in K} Y_{ik} (e_k, \xi - x_0) = \delta_{ij}$ ,  $\xi \in E^n$ , и пусть  $\varphi(\cdot) = 1 - \sum_{i \in K} \varphi_i(\cdot)$ .

2. Через  $\mu_0, \mu_1, \dots, \mu_n$  обозначим барицентрические координаты точки  $\xi \in E^n$  относительно вершин симплекса  $S = \langle x_0, x_1, \dots, x_n \rangle$ . Другими словами, если  $N = \{0, 1, \dots, n\} = \{0\} \cup K$ , то имеют место равенства  $\sum_{i \in N} \mu_i = 1$  и  $\sum_{i \in N} \mu_i x_i = \xi$ . Легко показать, что  $\varphi_i(\xi) = \mu_i$  для всех  $i \in N$ .

Как следствие, справедливы равенства

$$\xi = x_0 + \sum_{i \in K} \varphi_i(\xi) \Delta x_i, \quad \xi = x_0 + \sum_{i \in K} \varphi_i(\xi) x_i - \sum_{i \in K} \varphi_i(\xi) x_0 = \sum_{i \in N} \varphi_i(\xi) x_i.$$

Таким образом, барицентрические координаты точки  $\xi \in E^n$  представляют собой совокупность чисел  $(\varphi_0(\xi), \varphi_1(\xi), \dots, \varphi_n(\xi))$ . В частности,  $\varphi_i(x_j) = \delta_{ij}$  для всех  $i, j \in N$  и, следовательно,  $\varphi_i(\xi) = 0$  – это уравнение грани  $\text{conv}\{x_k : k \in N \setminus \{i\}\}$  симплекса  $S$ .

3. Зафиксируем целое неотрицательное число  $m$  и введем в рассмотрение частично упорядоченное множество мультииндексов

$$N_m = N(m) = \left\{ \alpha = (\alpha_0, \alpha_1, \dots, \alpha_n) : \alpha_i \in E, \alpha_i \geq 0, \sum_{i \in N} \alpha_i = m \right\},$$

где по определению полагаем  $\alpha < \beta$ , если  $\alpha_i \leq \beta_i$  для всех  $i \in K$  (очевидно,  $\alpha_0 \geq \beta_0$ ). Справедливо равенство  $\text{card } N_m = \binom{n+m}{m}$ .



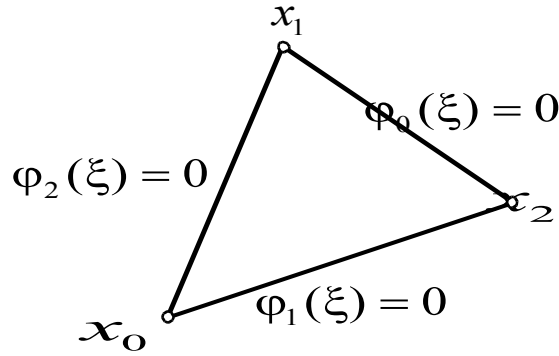


Рис. 1. Грани симплекса,  $n=2$

Для любого  $\alpha \in N_m$  полагаем по определению  $x_\alpha = \frac{1}{m} \sum_{i \in N} \alpha_i x_i$ . Очевидно,  $x_\alpha \in \text{conv} \{x_i : i \in N\}$ , а так как числа  $\frac{1}{m} \alpha_i, i \in N$  – это барицентрические координаты точки  $x_\alpha$ , то  $\varphi_i(x_\alpha) = \frac{1}{m} \alpha_i$ .

Полагаем, далее,  $\varphi_k^{[0]}(\xi) \doteq 1$  и  $\varphi_k^{[l]}(\xi) = \prod_{i=1}^l (m \varphi_k(\xi) + 1 - i)$  для всех  $l \in E$ . Наконец, для любого  $\alpha \in N_m$  полагаем  $\varphi^\alpha(\xi) = \prod_{k \in N} \frac{1}{\alpha_k!} \varphi_k^{[\alpha_k]}(\xi)$ . Другими словами, при  $m \in E$  для любого  $\alpha \in N_m$  справедливо представление

$$\varphi^\alpha(\xi) \doteq \prod_{k \in N: \alpha_k > 0} \frac{1}{\alpha_k!} \prod_{i=1}^{\alpha_k} (m \varphi_k(\xi) + 1 - i). \quad (1)$$

Зафиксируем мультииндексы  $\alpha, \beta \in N_m$ . Если  $\alpha \neq \beta$ , то легко показать, что  $\varphi_k^{[\alpha_k]}(x_\beta) = 0$  для некоторого  $k \in N$ , поэтому  $\varphi^\alpha(x_\beta) = 0$ . Если же  $\alpha = \beta$ , то для всех  $k \in N$  имеет место цепочка равенств

$$\varphi_k^{[\alpha_k]}(x_\beta) = \prod_{i=1}^{\alpha_k} (m \varphi_k(x_\beta) + 1 - i) = \prod_{i=1}^{\alpha_k} (\beta_k + 1 - i) = \prod_{i=1}^{\alpha_k} (\alpha_k + 1 - i) = \alpha_k!,$$

поэтому  $\varphi^\alpha(x_\beta) = 1$ . Таким образом,  $\varphi^\alpha(x_\beta) = \delta_{\alpha\beta}$  для любых  $\alpha, \beta \in N_m$ .

Последнее обстоятельство позволяет легко доказать, что совокупность  $\{\varphi^\alpha(\xi)\}_{\alpha \in N(m)}$  состоит из линейно независимых многочленов.

**4.** Через  $P_m[\xi]$  обозначим конечномерное пространство полиномов переменной  $\xi \in E^n$  степени не выше  $m$ . (Степенью монома  $\xi^\alpha = \prod_{i \in K} \xi_i^{\alpha_i}, \alpha \in N_m$ , называется сумма  $|\alpha| = \sum_{i \in K} \alpha_i$ , а степенью полинома – максимум из степеней его мономов. Совокупность таких мономов образует базис пространства  $P_m[\xi]$ , поэтому  $\dim P_m[\xi] = \binom{n+m}{m}$ .)

Очевидно,  $\varphi^\alpha(\xi) \in P_m[\xi]$  для любого  $\alpha \in N_m$ , следовательно, совокупности  $\{\xi^\alpha\}_{\alpha \in N(m)}$  и  $\{\varphi^\alpha(\xi)\}_{\alpha \in N(m)}$ , каждая из которых состоит ровно из  $\binom{n+m}{m}$  линейно независимых функций, являются базисами в пространстве  $P_m[\xi]$ . Первый базис называем далее *стандартным*, а второй – *S-базисом*, подчеркивая его происхождение от симплекса  $S$ . В работе [1] приведены прямая и обратная матрицы перехода от стандартного базиса к

S-базису.

**Теорема 1.** Пусть заданы симплекс  $S = \langle x_0, x_1, \dots, x_n \rangle$  и функция  $F: E^n \rightarrow E$ . В пространстве  $P_m[\xi]$  существует ровно один полином  $P$  такой, что  $P(x_\alpha) = F(x_\alpha)$  для всех  $\alpha \in N_m$  (где  $x_\alpha = \frac{1}{m} \sum_{i \in N} \alpha_i x_i$ ). Этот многочлен представим в виде

$$P(\xi) = \sum_{\alpha \in N(m)} F(x_\alpha) \varphi^\alpha(\xi). \quad (2)$$

Действительно,

$$P(x_\beta) = \sum_{\alpha \in N(m)} F(x_\alpha) \varphi^\alpha(x_\beta) = \sum_{\alpha \in N(m)} F(x_\alpha) \delta_{\alpha\beta} = F(x_\beta)$$

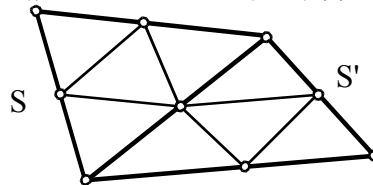
для всех  $\beta \in N_m$ , а единственность представления имеет место постольку, поскольку  $\{\varphi^\alpha(\xi)\}_{\alpha \in N(m)}$  – это S-базис в  $P_m[\xi]$ .

Теорема в несколько иных терминах доказана в статье [2].

**Следствие.** Пусть  $F: E^n \rightarrow E$  – это полином степени не выше  $m$ , то есть  $F \in P_m[\xi]$ , а  $\{\varphi^\alpha(\xi)\}_{\alpha \in N(m)}$  – это S-базис, порожденный симплексом  $S = \langle x_0, x_1, \dots, x_n \rangle$ . Справедливо тождество

$$F(\xi) \equiv \sum_{\alpha \in N(m)} F(x_\alpha) \varphi^\alpha(\xi). \quad (3)$$

**5.** Зафиксируем функцию  $F: E^n \rightarrow E$ , индекс  $r \in N$  и два симплекса  $S = \langle x_0, x_1, \dots, x_n \rangle$ ,  $S' = \langle x'_0, x'_1, \dots, x'_n \rangle$  такие, что  $x_i = x'_i$  для всех  $i \in N \setminus \{r\}$ .



**Рис. 2.** Два смежных симплекса триангуляции,  $n=2$ ,  $m=2$

Симплексы порождают в пространстве  $P_m[\xi]$  два базиса: S-базис  $\{\varphi^\alpha(\xi)\}_{\alpha \in N(m)}$  и S'-базис  $\{\varphi'^\alpha(\xi)\}_{\alpha \in N(m)}$ . Как следствие, определены два полинома вида (2):

$$P(\xi) = \sum_{\alpha \in N(m)} F(x_\alpha) \varphi^\alpha(\xi) \text{ и } P'(\xi) = \sum_{\alpha \in N(m)} F(x'_\alpha) \varphi'^\alpha(\xi),$$

где использованы обозначения  $x_\alpha = \frac{1}{m} \sum_{i \in N} \alpha_i x_i$  и  $x'_\alpha = \frac{1}{m} \sum_{i \in N} \alpha_i x'_i$ .

Значения полиномов  $P(\cdot)$  и  $P'(\cdot)$  совпадают на общей грани

$$\text{conv}\{x_i: i \in N \setminus \{r\}\} = \text{conv}\{x'_i: i \in N \setminus \{r\}\}.$$

Последнее утверждение предоставляет нам возможность осуществлять непрерывную стыковку полиномов вида (2), заданных на смежных симплексах. Данное обстоятельство позволяет аппроксимировать функции нескольких переменных сплайнами, построенными в соответствии с формулой (2) на произвольной триангулированной области.

**6.** В работах [3–5] получены точные формулы для коэффициентов и невязок оптимальных аппроксимирующих сплайнов простейших задач математической физики: для уравнения теплопроводности, для волнового уравнения и для уравнения переноса. Проиллюстрируем эти результаты на

примере уравнения переноса. При фиксированных  $\gamma \neq 0$  и  $\tau > 0$  в качестве приближенного решения задачи

$$\frac{\partial u}{\partial t} = \gamma \frac{\partial u}{\partial \xi}, \quad u(0, \xi) = \phi(\xi), \quad \xi \in [0, 1], \quad u(t, 0) = \rho(t), \quad t \in [0, \tau],$$

предлагается использовать оптимальный сплайн задачи

$$J \doteq \left\| \frac{\partial u}{\partial t} - \gamma \frac{\partial u}{\partial \xi} \right\|_{L_2(\Pi)}^2 \rightarrow \min, \quad u \in \sigma_\omega(\Pi). \quad (4)$$

Через  $\sigma_\omega(\Pi)$  (при фиксированном  $\omega \in [-1, 1]$ ) обозначено конечномерное пространство, состоящее из предложенных лагранжевых сплайнов, зависящих от коэффициентов  $u_j$ ,  $j=1, \dots, N$  (через  $N$  обозначено число узлов), и определенных в прямоугольнике  $\Pi = [0, \tau] \times [0, 1]$ .

Пусть  $n \doteq N-1$ ,  $h \doteq 1/N$ ,  $\theta \doteq \gamma \tau / h$ ,  $\rho_0 \doteq \rho(0)$ ,  $\rho_1 \doteq \rho(\tau)$ . Полагаем далее, что  $|\theta| > 1$ , тогда  $R = -[(1+\theta^2)\omega^2 - 4\theta\omega + 3\theta^2 - 1]/4 < 0$ . Используем также обозначения  $x \doteq \frac{1-R}{R}$ ,  $\alpha \doteq \frac{1-R-\theta}{R}$ ,  $\beta \doteq \frac{1-R+\theta}{R}$ ,

$$\phi_j \doteq \phi(jh), \quad u_j \doteq u(\tau, jh), \quad y_j \doteq u_j - \phi_j, \quad j = 0, 1, \dots, N, \quad (5)$$

$$w_j \doteq \frac{\theta}{R} (\phi_j - \phi_{j-1}), \quad j = 1, \dots, N, \quad W_k \doteq (1-\theta)w_k + (1+\theta)w_{k+1}, \quad k = 1, \dots, n.$$

Единственное решение  $u_j$ ,  $j=1, \dots, N$ , задачи (4) порождается совокупностью чисел

$$\begin{cases} y_N = -\frac{1}{U_N(x) - \beta U_n(x)} \left( -B_{n1}(x) y_0 + \sum_{j=1}^n B_{nj}(x) W_j - (1-\theta) U_n(x) w_N \right), \\ y_k = \frac{1}{U_n(x)} \left( -B_{k1}(x) y_0 - B_{kn}(x) y_N + \sum_{j=1}^n B_{kj}(x) W_j \right), \quad k = 1, \dots, n \end{cases} \quad (6)$$

(согласно (5) коэффициенты  $u_j$  линейно вычислимы через значения  $y_j$ ). Заметим еще, что оба знаменателя в (6) не равны нулю. В представлении использованы многочлены Чебышева, определяемые рекурсией:  $U_{-1}(x) \doteq 0$ ,  $U_0(x) \doteq 1$ ,  $U_{k+1}(x) \doteq 2x U_k(x) - U_{k-1}(x)$ , и полиномы

$$B_{ij}(x) \doteq (-1)^{i+j} \begin{cases} U_{i-1}(x) U_{n-j}(x), & \text{if } i \leq j \\ U_{j-1}(x) U_{n-i}(x), & \text{if } i \geq j \end{cases}, \quad i, j = 1, \dots, n.$$

Пусть вектор  $\xi \doteq (\xi_0, \xi_1, \dots, \xi_n)'$  состоит из граничных элементов

$$\xi_0 \doteq \tau \left( \frac{\rho_1 - \rho_0}{\tau} - \gamma \frac{\phi_1 - \phi_0}{h} \right), \quad \xi_i \doteq \gamma \tau \left( \frac{\phi_i - \phi_{i-1}}{h} - \frac{\phi_{i+1} - \phi_i}{h} \right), \quad i = 1, \dots, n,$$

а матрица  $A(\omega) = (A_{ij}(\omega))$ ,  $i, j = 0, 1, \dots, n$ , состоит из элементов

$$A_{ij}(\omega) \doteq \frac{1}{R(\alpha\beta - 1)} \begin{cases} \delta_{i,j+1} + 2x\delta_{ij} + \delta_{i,j-1}, & \text{if } (i, j) \neq (0, 0), \\ \alpha, & \text{if } (i, j) = (0, 0). \end{cases}$$

(При  $|\theta| > 1$  справедлива оценка  $\alpha\beta - 1 > 0$ .)

**Теорема 2.** Если  $J_*(\omega)$  – это минимум функционала  $J$  в пространстве  $\sigma_\omega(\Pi)$ , то

$$J_*(\omega) = \frac{h}{2\tau} (A^{-1}(\omega) \xi, \xi) \quad (7)$$

при любых  $\omega \in [-1, 1]$  и  $\theta$  таких, что  $|\theta| > 1$  (при любых  $N > (\tau|\gamma|)^{-1}$ ). Имеет место точная формула для элементов матрицы  $A^{-1}(\omega)$ :

$$A_y^{-1}(\omega) = \frac{R(\alpha\beta-1)}{U_N(x)-\beta U_n(x)} (-1)^{i+j} \begin{cases} (U_i(x)-\beta U_{i-1}(x)) U_{n-j}(x), & \text{if } i \leq j, \\ (U_j(x)-\beta U_{j-1}(x)) U_{n-i}(x), & \text{if } i \geq j. \end{cases}$$

Матрица  $A(\omega)$  имеет простой спектр:  $\lambda_0(\omega) < \lambda_1(\omega) < \dots < \lambda_n(\omega)$ , причем  $0 < \lambda_*(\omega) < \lambda_0(\omega)$  и  $\lambda_n(\omega) < 2(1-2R)/(1-2R-\theta^2)$ ; мы полагаем  $\lambda_*(\omega) = 2/(1-2R-\theta^2)$  при  $\theta < -1$  и  $\lambda_*(\omega) = 1/(1-R+\theta)$  при  $\theta > 1$ . (Оба знаменателя положительны.)

**Теорема 3.** Если  $\bar{\omega} \doteq 2\theta/(1+\theta^2)$ , то

$$\max_{\omega \in Q[-1,1]} \lambda_0(\omega) = \lambda_0(\bar{\omega}) \quad \text{и} \quad \max_{\omega \in Q[-1,1]} \lambda_*(\omega) = \lambda_*(\bar{\omega}).$$

Таким образом, для решения задачи (4) имеет место точная формула (6) (следует лишь пересчитать коэффициенты  $u_j$  через значения  $y_j$ ). Мы имеем также точную формулу (7) для минимума функционала (4). Следовательно, для любого  $\varepsilon > 0$  имеется возможность решить неравенство  $J_*(\omega) < \varepsilon$  и получить априори достаточное число узлов  $N$ . Более того, для этой цели имеется возможность решить неравенства

$$\frac{h}{2\tau} \frac{1}{\lambda_0(\omega)} \|\xi\|^2 < \varepsilon \quad \text{или} \quad \frac{h}{2\tau} \frac{1}{\lambda_*(\omega)} \|\xi\|^2 < \varepsilon.$$

В вычислениях используется параметр  $\bar{\omega} \doteq 2\theta/(1+\theta^2)$ .

## Литература

1. Родионов В.И. О применении специальных многомерных сплайнов произвольной степени в численном анализе // Вестн. Удмуртск. ун-та. Матем. Мех. Компьют. науки. 2010. № 4. С. 146–153.
2. Nicolaidides R.A. On a class of finite elements generated by Lagrange interpolation // SIAM J. Numer. Anal. 1972. № 9. P. 435–445.
3. Родионов В.И., Родионова Н.В. Точные формулы для коэффициентов и невязки оптимального аппроксимирующего сплайна простейшего уравнения теплопроводности // Вестн. Удмуртск. ун-та. Матем. Мех. Компьют. науки. 2010. № 4. С. 154–171.
4. Родионова Н.В. Точные формулы для коэффициентов и невязки оптимального аппроксимирующего сплайна простейшего волнового уравнения // Вестн. Удмуртск. ун-та. Матем. Мех. Компьют. науки. 2012. № 1. С. 144–154.
5. Родионов В.И. Точные формулы для коэффициентов и невязки оптимального аппроксимирующего сплайна простейшего уравнения переноса // Материалы IV международной научной конференции «Современные проблемы прикладной математики, теории управления и математического моделирования (ПМТУММ-2011)». Воронеж: Издательско-полиграфический центр Воронежского государственного университета, 2011. С. 252–253.

**Романчук С. П.,**

ЭТИ (филиал) СГТУ, ассистент  
[romanchuk\\_sergey@bk.ru](mailto:romanchuk_sergey@bk.ru)

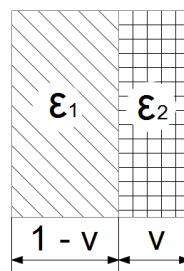
**Терин Д.В.**

ЭТИ (филиал) СГТУ, к.ф.-м.н., доцент  
[terinden@mail.ru](mailto:terinden@mail.ru)

## ***Математическое моделирование композитных сред***

В текущее время интерес к изучению нанокомпозитных сред с каждым годом растет. Это обусловлено тем, что постоянно нарастает потребность в новых материалах с заданными электрофизическими свойствами. Нанокомпозитные среды являются базой для разработки новых материалов, т.к. свойства композита могут кардинально отличаться от свойств его компонент. Экспериментальные данные демонстрируют ранее не известные зависимости физических свойств наноматериалов от геометрической формы и размеров частиц. Одной из проблем проектирования материалов с необходимыми свойствами является отсутствие необходимых программных инструментов. В этой статье рассматривается структура узкоспециализированного программного комплекса, при помощи которого осуществляется возможность проектирования композитных сред и прогнозировать их свойства с наперед заданными свойствами компонент. Для разработки программного комплекса были проанализированы некоторые модели комплексной диэлектрической проницаемости гетерогенных дисперсных систем.

Бинарная система, имеющая слоистую структуру (теория Максвелла)



***Рис. 1. Бинарная система***

В набор математических моделей программного комплекса входит бинарная система, имеющая слоистую структуру (рис. 1).

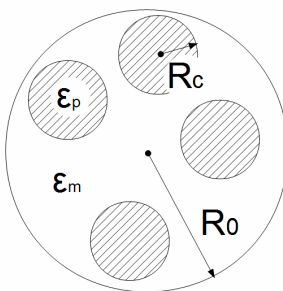
Общая формула:

$$\varepsilon = \frac{\varepsilon_1 \cdot \varepsilon_2}{\varepsilon_1 + v \cdot (\varepsilon_1 - \varepsilon_2)} \quad (1)$$

где  $\varepsilon$  – комплексная диэлектрическая проницаемость композита,  
 $\varepsilon_1$  – комплексная диэлектрическая проницаемость первого элемента,  
 $\varepsilon_2$  – комплексная диэлектрическая проницаемость второго элемента,  
 $v$  – объемная доля второго компонента.

### ***Разбавленные дисперсные системы сферических частиц (теория Вагнера)***

Вагнер предложил теорию поляризации поверхности раздела для дисперсной системы, в которой сферические частицы равномерно распределены в дисперсной среде (система разбавлена). В данной теории рассматривается в среде с комплексной диэлектрической проницаемостью  $\varepsilon_m$  некоторая сферическая область  $R_0$ , содержащая  $N$  маленьких сфер с радиусом  $R_c$  и комплексной диэлектрической проницаемостью  $\varepsilon_p$  (рис. 2).



**Рис. 2.** Разбавленные дисперсные системы сферических частиц  
 Общая формула:

$$\varepsilon = \frac{\varepsilon_m \cdot 2\varepsilon_m + \varepsilon_p - 2V(\varepsilon_m - \varepsilon_p)}{2\varepsilon_m + \varepsilon_p + V(\varepsilon_m - \varepsilon_p)} \quad (2)$$

где  $\varepsilon$  – комплексная диэлектрическая проницаемость композита,  
 $\varepsilon_m$  – комплексная диэлектрическая проницаемость среды,  
 $\varepsilon_p$  – комплексная диэлектрическая проницаемость фазы,  
 $V$  – объемная доля второго компонента.

$$V = \frac{R_c^3}{R_0^3} N \quad (3)$$

где  $R_c$  – радиус частиц,  
 $R_0$  – радиус рассматриваемой области,  
 $N$  – количество частиц на рассматриваемую область.

### ***Теория эффективной среды.***

Важную роль в физике нанокомпозитных сред играет так называемая модель эффективной среды. Суть этой модели состоит в том, что ансамбль нанокластеров можно рассматривать как некую новую среду с эффективной диэлектрической проницаемостью. Очевидным преимуществом данного подхода является то, что в его рамках для анализа распространения излучения в нанокомпозитной среде нет необходимости решать уравнения Максвелла в каждой точке пространства. Как правило, в

модели эффективной среды для оптических задач пользуются электростатическим приближением, условием которого является малость как размера наночастиц, так и расстояния между ними по сравнению с длиной оптической волны в среде. В противном случае неизбежно встает задача рассеяния на составляющих нанокомпозитную среду частицах и интерференции рассеянных волн. В рамках модели эффективной среды мы можем, зная оптические параметры каждого из компонентов композитной среды, а также их концентрацию и геометрическую форму, определить эффективные параметры всей среды как целого [1].

Общая формула:

$$p\alpha = \frac{\varepsilon_s(3\varepsilon_c + (\alpha - 1)(\varepsilon_c + 2\varepsilon_s)) - \varepsilon_{eff}(3\varepsilon_s + (\alpha - 1)(\varepsilon_c + 2\varepsilon_s))}{2\varepsilon_{eff}((\alpha - 1)\varepsilon_c + 2(\alpha + 1)\varepsilon_s) + \varepsilon_s((\alpha + 2)\varepsilon_c + 2(\alpha - 1)\varepsilon_s)} + \frac{(1 - p\alpha)\varepsilon_g - \varepsilon_{eff}}{\varepsilon_g + 2\varepsilon_{eff}} \quad (4)$$

где  $\varepsilon_{eff}$  – комплексная диэлектрическая проницаемость композита,

$\varepsilon_c$  – комплексная диэлектрическая проницаемость ядра,

$\varepsilon_p$  – комплексная диэлектрическая проницаемость оболочки,

$\varepsilon_g$  – комплексная диэлектрическая проницаемость газа,

$p$  – объемная доля второй фазы,

$\alpha = D^3d^{-3}$ ,  $d$  и  $D$  – диаметры ядра и наночастицы.

$$\varepsilon_c = \varepsilon'_c + i\varepsilon''_c, \varepsilon_p = \varepsilon'_p + i\varepsilon''_p \quad (5)$$

После подстановки (5) в (4), мы получаем алгебраическое каноническое уравнение с комплексными коэффициентами  $A_i = a_i + jb_i$ :

$$a_0 = 2\varepsilon_g\varepsilon_s'\varepsilon_c' - 2\varepsilon_g\varepsilon_s''\varepsilon_c'' + \varepsilon_g\alpha\varepsilon_c'\varepsilon_s' - \varepsilon_g\alpha\varepsilon_c''\varepsilon_s'' + 2\varepsilon_g\alpha(\varepsilon_s')^2 - 2\varepsilon_g\alpha(\varepsilon_s'')^2 - 2\varepsilon_g(\varepsilon_s'')^2 + 2\varepsilon_g(\varepsilon_s'')^2$$

$$a_1 = -2\alpha(\varepsilon_s')^2 - 2(\varepsilon_s'')^2 - 6p\alpha(\varepsilon_s')^2 + 6p\alpha(\varepsilon_s'')^2 + 6p\alpha^2(\varepsilon_s')^2 - 6p\alpha^2(\varepsilon_s'')^2 + 2(\varepsilon_s')^2 - 2\varepsilon_s'\varepsilon_c' + 2\varepsilon_s''\varepsilon_c'' + 3p\alpha^2\varepsilon_s'\varepsilon_c' - 3p\alpha^2\varepsilon_s''\varepsilon_c'' + 6p\alpha\varepsilon_s'\varepsilon_c' - 6p\alpha\varepsilon_s''\varepsilon_c'' - \alpha\varepsilon_c'\varepsilon_s' + \alpha\varepsilon_c''\varepsilon_s'' + 2\varepsilon_g\alpha\varepsilon_c' + 4\varepsilon_g\alpha\varepsilon_s' + 2\alpha(\varepsilon_s'')^2 - p\alpha^2\varepsilon_g\varepsilon_s'\varepsilon_c' + p\alpha^2\varepsilon_g\varepsilon_s''\varepsilon_c'' - 2p\alpha^2\varepsilon_g(\varepsilon_s')^2 + 2p\alpha^2\varepsilon_g(\varepsilon_s'')^2 - p\alpha\varepsilon_g(\varepsilon_s')^2 + p\alpha\varepsilon_g(\varepsilon_s'')^2 + p\alpha\varepsilon_g\varepsilon_s'\varepsilon_c' - p\alpha\varepsilon_g\varepsilon_s''\varepsilon_c'' - 2\varepsilon_g\varepsilon_c' + 2\varepsilon_g\varepsilon_s' - 2p\alpha^2\varepsilon_g\varepsilon_c' - 4p\alpha^2\varepsilon_g\varepsilon_s' + 2p\alpha\varepsilon_g\varepsilon_c' - 2p\alpha\varepsilon_g\varepsilon_s';$$

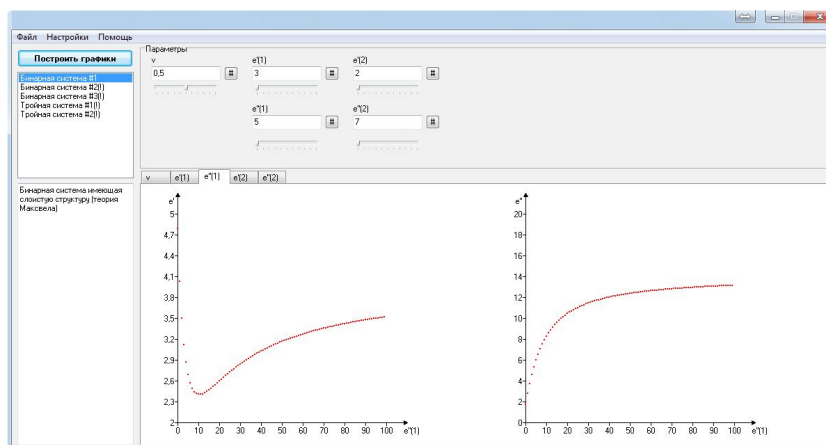
$$a_2 = 2\varepsilon_c' - 2\varepsilon_s' - 2p\alpha(\varepsilon_s')^2 + 2p\alpha(\varepsilon_s'')^2 - 4p\alpha^2(\varepsilon_s')^2 + 4p\alpha^2(\varepsilon_s'')^2 + 2p\alpha^2\varepsilon_c' + 4p\alpha^2\varepsilon_s' - 2p\alpha\varepsilon_c' + 2p\alpha\varepsilon_s' - 2\alpha\varepsilon_c' - 4\alpha\varepsilon_s' - 2p\alpha^2\varepsilon_s'\varepsilon_c' + 2p\alpha^2\varepsilon_s''\varepsilon_c'' + 2p\alpha\varepsilon_s'\varepsilon_c' - 2p\alpha\varepsilon_s''\varepsilon_c'';$$

$$b_0 = \varepsilon_g\varepsilon_c'\varepsilon_s'' - 4\varepsilon_g\varepsilon_s''\varepsilon_s' + 2\varepsilon_g\varepsilon_c''\varepsilon_s' + 2\varepsilon_g\varepsilon_s''\varepsilon_c' + 4\varepsilon_g\varepsilon_s''\varepsilon_s' + \varepsilon_g\varepsilon_c''\varepsilon_s';$$

$$b_1 = 4\varepsilon_s''\varepsilon_s' - 2\varepsilon_c''\varepsilon_s' - 2\varepsilon_s''\varepsilon_c' - 2\varepsilon_g\varepsilon_c'' + 2\varepsilon_{ge2}'' + p\alpha\varepsilon_g\varepsilon_s'\varepsilon_c'' + p\alpha\varepsilon_g\varepsilon_s''\varepsilon_c' - p\alpha^2\varepsilon_g\varepsilon_s'\varepsilon_c'' - p\alpha^2\varepsilon_g\varepsilon_s''\varepsilon_c' - \alpha\varepsilon_c''\varepsilon_s' - \alpha\varepsilon_c'\varepsilon_s'' - 4\alpha\varepsilon_s''\varepsilon_s' - 4p\alpha^2\varepsilon_g\varepsilon_s'\varepsilon_s'' - 2p\alpha\varepsilon_g\varepsilon_s'\varepsilon_s'' + 2\varepsilon_g\alpha\varepsilon_c'' + 4\varepsilon_g\alpha\varepsilon_s'' - 2p\alpha^2\varepsilon_g\varepsilon_c'' + 2p\alpha\varepsilon_g\varepsilon_c'' - 4p\alpha^2\varepsilon_g\varepsilon_s'' - 2p\alpha\varepsilon_g\varepsilon_s'' + 3p\alpha^2\varepsilon_s'\varepsilon_c'' + 3p\alpha^2\varepsilon_s''\varepsilon_c' - 12p\alpha\varepsilon_s'\varepsilon_s'' + 6p\alpha\varepsilon_s'\varepsilon_c'' + 12p\alpha^2\varepsilon_s''\varepsilon_s' + 6p\alpha\varepsilon_s''\varepsilon_c';$$

$$b_2 = -2\alpha\varepsilon_c'' - 4\alpha\varepsilon_s'' + 2\varepsilon_c'' - 2\varepsilon_s'' - 4p\alpha\varepsilon_s'\varepsilon_s'' - 2p\alpha^2\varepsilon_s'\varepsilon_c'' - 2p\alpha^2\varepsilon_s''\varepsilon_c' - 8p\alpha^2\varepsilon_s'\varepsilon_s'' + 2p\alpha^2\varepsilon_c'' - 2p\alpha\varepsilon_c'' + 4p\alpha^2\varepsilon_s'' + 2p\alpha\varepsilon_s'' + 2p\alpha\varepsilon_s'\varepsilon_c'' + 2p\alpha\varepsilon_s''\varepsilon_c'.$$

В программном комплексе использованы методы решения полиномиальных уравнений с комплексными переменными: модифицированный метод «обруча», Дженкинса – Трауба, Дюран – Кернера, Аберта – Эрлиха.



**Рис. 3.** Основное окно программного комплекса

Программный комплекс позволяет исследовать диэлектрическую проницаемость композитной среды на основе введенных значений свойств компонент материала. Таких, как диэлектрическая проницаемость каждого компонента, объемная доля каждого компонента, форма и размер частиц. Имеется возможность указывать интервал значений свойств среды. Таким образом, при изучении диэлектрической проницаемости бинарной композитной среды имеющую слоистую структуру, система вычисляет в среднем около 1000 полиномиальных уравнений с комплексными переменными.

Детально исследован модифицированный метод «обруча», который позволяет за небольшое количество итераций вычислять корни полиномиальных уравнений  $n$ -степени с комплексными переменными. Принцип работы метода опирается на тот факт, что  $f(z)$  – аналитическая во всей комплексной плоскости функция, и на принцип максимума модуля аналитической функции.

$$f(z) = a_0 z^n + a_1 z^{n-1} + \dots + a_n, (a_0 \neq 0) \quad (6)$$

В общих чертах алгоритм выглядит так: для начала фиксируются несколько величин: начальный угол  $\alpha$  ( $0 \leq \alpha < 2\pi$ ), начальный шаг  $h$  ( $0 < h < 2\pi/3$ ), «точность» вычислений  $E$  ( $E > 0$ ). Далее в комплексной плоскости строится окружность с центром в  $O(E_0, R)$ , центр окружности на первом шаге работы алгоритма находится в центре координат, а радиус задается как параметр.

На следующем этапе на обруче фиксируется некоторое количество точек  $N = 360 / h$ . В каждой зафиксированной точке вычисляется значение модуля многочлена  $f(z)$ , таким образом получаем набор значений  $F_i$  ( $i = 0, N$ ). Далее находим наименьший индекс  $s$  такой, что  $F_s \leq F_i$  ( $i = 0, N$ ), и если  $s = 0$ , то радиус обруча уменьшаем вдвое, иначе центр обруча перемещаем в точку с индексом  $s$ , а радиус увеличиваем вдвое. Далее если  $R < E$ , то объявляем центр обруча корнем, иначе продолжаем спуск.



h	$\alpha$											
	0	30	60	90	120	150	180	210	240	270	300	330
1	78	76	74	82	74	70	72	70	78	88	66	66
5	60	60	64	70	72	62	72	64	62	66	66	60
35	78	66	60	64	78	84	74	76	70	70	60	78
65	66	54	68	56	62	104	64	70	54	70	62	80
95	60	60	82	68	76	62	90	64	64	70	68	54
125	34	34	64	66	74	66	38	38	80	70	70	62
155	34	64	62	72	66	76	42	58	74	62	62	52
185	34	34	34	34	34	34	54	38	36	36	36	58
215	34	34	34	34	34	34	54	38	36	36	36	60
245	34	34	34	34	34	34	54	38	36	36	36	56
275	34	34	34	34	34	34	54	60	36	36	36	56
305	34	34	34	34	34	34	54	60	36	36	36	56
335	34	34	34	34	34	34	54	62	36	36	36	70
360	34	36	36	38	36	36	36	36	36	36	36	34

**Рис. 4.** Статистика итераций при различных параметрах алгоритма

Таким образом, можно увидеть, что основными параметрами алгоритма является начальный угол  $\alpha$ , шаг  $h$  и начальный радиус, а также точность вычислений.

Были проведены подсчеты производительности алгоритма при различных начальных значениях  $0 < h < 360$ ,  $0 < \alpha < 360$  (рис. 4). Значениями таблицы является среднее количество итераций алгоритма. Из таблицы видно, что минимальное количество итераций выполняется при значениях:  $185 < h < 360$  и  $0 < \alpha < 150$

### Литература

1. Buchelnikov V.D. Heating of metallic powders by microwaves: experiment and theory./ V.D.Buchelnikov, D.V.Louzguine-Luzgin, G. Xie, S. Li, N. Yoshikawa, M. Sato, A.P. Anzulevich, I.V. Bychkov, A. Inoue // J. Applied Physics, 2008, 104, P. 113505-1-113505-10.
2. Биленко Д.И. Электродинамические свойства неупорядоченных сред/Д.И.Биленко, Ю.Н. Галишникова, Е.И. Хасина и др. // Физика полупроводников и полупроводниковая электроника. Саратов: Изд-во Саратов. ун-та, 1986. С.32–51.
3. Биленко Д.И. Комплексная диэлектрическая проницаемость. Плазменный резонанс свободных носителей заряда в полупроводниках. Саратов: Изд-во Саратов. ун-та, 1999. 44 с.
4. O. Aberth, Iteration methods for finding all zeros of a polynomial simultaneously. MATHEMATICS OF COMPUTATION, 1973, pp. 339 – 344.

**Синицын И.Н.,**

ИПИ РАН, начальник отдела  
[sinitsin@dol.ru](mailto:sinitsin@dol.ru)

**Шаламов А.С.,**

ИПИ РАН, консультант  
[a-shal5@yandex.ru](mailto:a-shal5@yandex.ru)

**Корепанов Э.Р.,**

ИПИ РАН, зав. сектором  
[ekorepanov@ipiran.ru](mailto:ekorepanov@ipiran.ru)

**Агафонов Е.С.**

ИПИ РАН, н.с.  
[eagafonov@ipiran.ru](mailto:eagafonov@ipiran.ru)

### ***Опыт создания и обучения средствам интегрированной логистической поддержки изделий наукоемкой продукции в среде MATLAB***

В настоящее время наметилась тенденция увеличения темпа развития систем компьютерной математики (СКМ) в целом [1,2]. Некоторые СКМ, например MathCad, Map1e и MATLAB, развивались настолько быстро, что их новые версии выходили ежегодно, а иногда и более часто. Наряду с положительными моментами (быстрое обновление систем и рост их функциональности) это имеет и отрицательные моменты - некоторые версии систем выходят довольно сырыми, замеченные в предшествующих версиях недостатки и недоделки вовремя не устраняются, в новые версии вносится мало действительно новых возможностей, затрудняется выпуск литературы по новым версиям. Наиболее популярны СКМ, изначально ориентированные на решение задач в образовательной сфере и мощные СКМ для выполнения серьезных расчетов и на реализацию математического моделирования сложных систем и устройств.

Несколько необычной кажется большая и постоянно растущая популярность мощной матричной системы MATLAB, явно ориентированной на численные вычисления и впитавшей в себя все возможности матричных вычислений, созданные за полвека. Как и в MathCad, в MATLAB есть ограниченные возможности аналитических вычислений с помощью пакета расширения SymbolicMath Toolbox с встроенным ядром символьных вычислений системы Map1e (а в последних реализациях еще и MuPAD). Несомненно, большой популярности MATLAB способствовали тщательно

отработанные со времен создания больших ЭВМ методы матричных вычислений для матриц больших размеров и наличие одного из лучших пакетов блочного математического имитационного моделирования Simulink.

В технологии проведения научных расчетов СКМ стали своего рода суперкалькуляторами, практически мгновенно решающими сложные уравнения, вычисляющие интегралы и производные, строящими графики любых функций и т.д. В новейших реализациях MATLAB появились средства моделирования электрических и электронных цепей и схем с применением физических моделей компонентов и моделей, применяемых в программах схемотехнического моделирования PSPICE.

MATLAB стала признанным во всем мире средством и языком решения научно-технических вычислений и осуществления визуально-ориентированного блочного математического моделирования различных явлений, систем и устройств.

В [3] был дан обзор работ ИПИ РАН в области стохастических информационных технологий научных исследований в среде MATLAB.

Рассматриваемые СКМ находят успешное применение не только в научной и образовательной сфере, но и в финансово-промышленном секторе. Практически все крупные, территориально распределенные организации рано или поздно сталкиваются с необходимостью построения учетно-аналитических информационных систем для грамотного управления и оптимизации распределения и расходования своих ресурсов. При реализации таких систем приходится осуществлять сбор большого объема разнородной информации в некоторой центральной подсистеме, выполнять статистическую обработку данных, вычислять различные аналитические показатели текущего состояния, строить прогнозы дальнейшего развития, осуществлять визуализацию полученных результатов.

Если MATLAB ориентирован на объемные численные вычисления с матрицами, то система управления базами данных (СУБД) Oracle занимает лидирующие позиции в тех случаях, когда необходимо осуществлять хранение и быструю обработку очень больших объемов информации. Совместное использование MATLAB и СУБД Oracle для построения описываемых информационных систем позволяет получать удачные и надежно работающие решения в данной области.

В начале 2012 года вышла книга [3]. В ней дается систематическое изложение теоретических основ одного из новейших направлений в области экономики послепродажного обслуживания изделий наукоемкой продукции (ИНП) длительного использования – интегрированной логистической поддержки. Приводятся также результаты работ, выполненных в институте проблем информатики Российской академии наук в рамках Программы «Информационные технологии и анализ сложных систем» Отделения нанотехнологий и информационных

технологий РАН.

Интегрированная логистическая поддержка (ИЛП) – это система научно-исследовательских, проектно-конструкторских, организационно-технических, производственных и информационно-управленческих технологий, средств и практических мероприятий, используемых (применяемых) в течение жизненного цикла ИНП, направленных на достижение минимальных затрат по обслуживанию и ремонту ИНП при обеспечении требуемых характеристик и показателей функционального качества и технической готовности продукции при ее эксплуатации.

Излагаемые новые научные подходы позволяют кардинально реформировать традиционные системы создания и эксплуатации ИНП путем внедрения методов рационального и, по возможности, оптимального управления процессами расходования временных, материальных, трудовых и др. ресурсов на всех стадиях жизненного цикла (ЖЦ) изделий по критериям экономической целесообразности и эффективности.

Первыми на этот путь вступили западные страны, когда в 90-ых годах XX века были резко сокращены бюджеты военных ведомств, что вынудило их искать возможности экономии финансовых ресурсов, сохраняя при этом основные показатели обороноспособности. Были выработаны новые стандарты по созданию вооружений и военной техники (ВВТ), обеспечивающие снижение стоимости ее ЖЦ. Контракты, заключаемые с поставщиками ВВТ, стали содержать требования по неукоснительному соблюдению правил и процедур, обеспечивающих заданные характеристики и показатели качества и экономичности продукции. В настоящее время на этот путь переводятся и отрасли экономики гражданского назначения. В основу серии этих международных стандартов легла CALS – методология как первооснова современных глобальных интегрированных информационно - коммуникационных систем (CALS – Continuous Acquisition and Life cycle Support – интегрированная поддержка изделий) — непрерывная информационная поддержка поставок и жизненного цикла). Эта методология позволяет создавать виртуальные предприятия (ВП) из фирм - участников ЖЦ ИНП, интегрируя их на информационной основе.

Основные принципы ИЛП и интеграции ВП изложены в стандартах MIL STD 1388 (США) и DEF STAN 00-60 (общеевропейский стандарт).

Стандарт DEF STAN 00-60, тесно связан со спецификациями ASD (AeroSpace and Defense Industries Association of Europe – Европейской ассоциации аэрокосмической и оборонной промышленности):

- S1000D «International specification for technical publications utilizing a Common Source Data Base» («Международная спецификация на технические публикации, использующие общую исходную базу данных»);
- S2000M «International specification for Material management»

(«Международная спецификация на организацию управления материально-техническим обеспечением») и др.

Важно отметить, что в этих стандартах отсутствуют указания на методы и средства, позволяющие достигать реального минимума затрат.

Участники ВП, по сути, самостоятельно пытаются решать свои проблемы или прибегают к услугам фирм – разработчиков программных средств для решения частных (сервисно – ориентированных) задач по управлению процессами. При этом, как правило, информация об использовании каких-либо методов решения этих задач остается закрытой, что приносит неопределенность при оценке покупателем гарантий достоверности таких программ. Таким образом, напрашивается вывод об определенной неполноте современного содержания CALS, выражаемой отсутствием теоретических обоснований путей достижения провозглашаемых ею целей. Для полноценного использования преимуществ данной методологии необходима разработка теории систем ИЛП, включающей в себя единый комплекс современных методов и средств информационного и математического моделирования, оценивания и оптимального управления процессами на всех стадиях ЖЦ комплексной системы «ИЛП – системы послепродажного обслуживания (СППО)», вносящими основной вклад в ее стоимость и качество.

В книге, помимо общих понятий, вводящих читателя в курс дела, даются постановки и решения практически по всему комплексу задач ИЛП, обеспечивающих управление стоимостью ЖЦ ИЛП и ее минимизацию.

В части 1 «Интегрированная логистическая поддержка» (главы 1-3), излагаются элементы современной системы послепродажного обслуживания изделий, рассматриваются основы современной ИЛП. Особое внимание уделяется управлению ЖЦ, системам послепродажного обслуживания и их информационному моделированию.

Часть 2 (главы 4-6) посвящена теории гибридных стохастических систем и компьютерной поддержке исследований и разработок. Приводятся необходимые сведения из теории вероятностей, стохастического анализа и стохастических уравнений. Даются вероятностные распределения процессов в гибридных стохастических системах. Рассматриваются задачи, методы и средства компьютерной поддержки исследований и разработок.

В части 3 «Основы математического моделирования, анализа и синтеза систем послепродажного обслуживания» (главы 7-9) рассматриваются вопросы математического моделирования процессов в системах послепродажного обслуживания, моделирование систем послепродажного обслуживания со смешанными потоками расходования, восстановления и пополнения ресурсов, а также статистический анализ и синтез систем послепродажного обслуживания.

Часть 4 (главы 10-12) посвящена определению и анализу показателей экспортного потенциала ИЛП при проектировании. Проводится анализ

технико-экономических факторов, определяющих экспортный потенциал ИМП. Рассматриваются вопросы разработки математической модели процессов эксплуатации ИМП для оценки показателей экспортного потенциала, а также оценки влияния эксплуатационно-технических характеристик на экспортный потенциал ИМП.

В заключительной части 5 (главы 13 и 14) рассматриваются задачи управления поддержкой послепродажного обслуживания на стороне заказчика и на стороне поставщика, а также моделирование инвестиционных процессов ИЛП в условиях современных финансовых рынков.

Особенностью отечественного подхода к решению основной задачи ИЛП является широкое использование методов математического программирования и теории гибридных стохастических систем. При этом упомянутые технико-экономические модели приобретают динамический характер.

Результаты решения задачи моделирования и оптимизации интегрального качества и стоимости послепродажного обслуживания ИМП на примере одной из типовых распределенных систем прикладной информатики показывают, что поставленные цели минимизации затрат могут действительно достигаться в рамках использования стандартизированной CALS концепции. Но это возможно лишь за счет включения дополнительно информационно-вычислительного комплекса прогнозирования процессов и формирования долговременной Программы поставок потребных ресурсов на произвольном периоде эксплуатации. Кроме того, решение этой задачи оптимального планирования позволяет также существенно повысить ритмичность производства ИМП, что является весьма критичным для изготовителей наукоемкой продукции, а также эффективность планирования ее использования по назначению.

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 10-07-00021) и программы ОНИТ РАН "Интеллектуальные информационные технологии, системный анализ и автоматизация" (проект 1.7).*

### **Литература**

1. Дьяконов В.П. Системы компьютерной математики на выходе из кризиса // Системы компьютерной математики и их приложения: материалы XII международной научной конференции. Смоленск: изд.-во СмолГУ, 2011. Вып. 12. С. 16-24.
2. Сеницын И.Н., Шаламов А.С., Сергеев И.В., Белоусов В.В., Агафонов Е.С. Развитие средств интегрированной логистической поддержки изделий наукоемкой продукции на основе систем компьютерной математики // Системы компьютерной математики и их приложения: материалы XIII международной научной конференции, посвященной 75-летию профессора Э.И. Зверовича. Смоленск: Изд.-во СмолГУ, 2012. Вып. 13. С. 119-124.
3. Сеницын И.Н., Сеницын В.И., Корепанов Э.Р., Белоусов В.В., Агафонов Е.С. Стохастические информационные технологии в среде MATLAB в задачах компьютерной

поддержки научных исследований // Системы компьютерной математики и их приложения: материалы XII международной научной конференции. Смоленск: Изд.-во СмолГУ, 2011. Вып. 12. С. 57-59.

4. Синицын И.Н., Шаламов А.С. Лекции по теории систем интегрированной логистической поддержки. М.: ТОРУС ПРЕСС, 2012. 624 с.

## ***Вопросы использования Interactive Data Language при создании научных приложений***

### ***Введение***

Бурное развитие методов дистанционного зондирования Земли (ДЗЗ) в последние десятилетия связано с новыми возможностями наблюдений из космоса с помощью бортовой научной аппаратуры космических аппаратов (КА). Измерения с борта КА характеристик уходящего электромагнитного излучения системы "атмосфера - подстилающая поверхность" позволяют получать разнообразную информацию о параметрах состояния атмосферы и подстилающей поверхности, необходимую при решении таких задач, как исследование общей циркуляции атмосферы, численный прогноз погоды, слежение за опасными метеорологическими явлениями, изучение климатических изменений и определяющих их факторов [1].

Приборами, с помощью которых можно решать указанные выше задачи являются СВЧ-радиометры. Такие приборы предназначены для применения в качестве инструментария при решении фундаментальных проблем системы океан-атмосфера, а также для глобального мониторинга в интересах Росгидромета.

СВЧ-радиометр обеспечивает зондирование системы океан-атмосфера как в окнах прозрачности атмосферы, так и в линиях поглощения кислорода (52-57 ГГц) и водяного пара (183 ГГц), что позволяет:

- восстанавливать вертикальные профили температуры и влажности атмосферы;
- определять интегральную влажность атмосферы, водозапас облаков и интенсивность осадков;
- регистрировать такие параметры океана, как состояние (бальность волнения) поверхности, скорость приводного ветра и др.

СВЧ-радиометры ранее устанавливались на борт метеорологических КА типа «Метеор». Учитывая высокую потребность в данных СВЧ-радиометрии использование подобного рода бортовой аппаратуры планируется и далее как на низких и геостационарных орбитах. Поэтому актуальным является применение современных информационных технологий для обработки получаемых данных. Целью настоящей работы является демонстрация возможностей языка программирования Interactive Data Language (IDL) на примере создания приложения для автоматизации



обработки исходных данных СВЧ-радиометрии с целью выявления метеорологических параметров: приводного ветра, водозапаса облачности и интенсивности осадков.

### ***Особенности языка IDL***

Рассмотрим особенности языка программирования IDL. В 1970-х гг. сотрудник Лаборатории атмосферной и космической физики (США) Дэвид Стерн (David Stern) создал язык программирования Rufus (прообраз IDL). В 1977г. он организовал собственную фирму Research Systems, Inc. (RSI), которая стала заниматься разработкой языка программирования для решения научных задач IDL. В 1977 г. вышла первая версия языка IDL. В 1992 г. было введено понятие «виджет» (widget — элемент графического интерфейса), благодаря чему создание IDL приложений получило новый виток развития.

Язык IDL является кроссплатформенным языком программирования, что позволяет создавать приложения на различных платформах в различных операционных системах, таких как Microsoft Windows, Linux и другие.

При проведении исследований, как правило, используется большой объем данных, которые необходимо быстро обработать. Это требование стало ключевым для языка IDL. Оно воплотилось в IDL в виде простого синтаксиса, т. е. нет необходимости в составлении циклов и оптимизации времени выполнения функций для работы с массивами.

Язык IDL предлагает пользователям большое количество разнообразных математических преобразований и функций, способных помочь решить достаточно сложные задачи. Для более детального анализа полученных результатов IDL обладает мощными средствами визуализации данных. Визуализация данных в IDL поддерживает два типа рендеринга: программный рендеринг и рендеринг с использованием аппаратных 3D ускорителей, которые поддерживают библиотеку OpenGL.

В большинстве случаев визуализация данных через OpenGL обладает лучшим качеством изображения, чем визуализация через программный рендеринг. Но необходимо учитывать, то что программный рендеринг может быть единственно доступным решением для систем, в которых нет аппаратных 3D ускорителей, например, при использовании несовместимых OpenGL X серверов этот тип рендеринга строит одноразовую визуализацию быстрее, чем OpenGL и позволяет избежать возможных ошибок, возникающих с драйверами устройств.

IDL позволяет работать с файлами различных форматов, в том числе с графическими, например, BMP, JPEG, PNG, форматами, разработанными специально для научных задач, например, HDF, CDF, NCDF и несколькими распространенными форматами хранения данных, например, XML, ASCII. Существует возможность работы с данными в формате DICOM (Digital Imaging and Communications in Medicine).

Программы, разработанные на языке IDL, можно вызывать из

внешних программ, созданных на других языках программирования или в IDL программах использовать функции, реализованные на C/C++, FORTRAN, Java. Такое взаимодействие IDL с языками программирования позволяет гибко подходить к решению сложных задач.

Благодаря этим преимуществам IDL нашел широкое применение при обработке космических снимков и геофизических данных. Он активно применяется в структурах национальных космических агентств: NASA (США), JAXA (Япония), CNES (Франция), Роскосмос.

Рассмотрим особенности применения объектно-ориентированного программирования, визуализации данных и построения графического интерфейса пользователя.

### ***Особенности описания классов и объектов в IDL***

В IDL существует возможность применения как процедурной, так и объектно-ориентированной парадигм. Рассмотрим более подробно применение объектно-ориентированной парадигмы.

В IDL данные члены-класса хранятся в виде именованных структурах. В терминологии IDL применяется выражение «class structure».

```
struct={Class1, field1:0L, field2:1B}
```

Любой объект, созданный из этой структуры (класса), будет иметь 2 свойства – field1 и field2.

Методы членов-класса имеют следующие отличия от обычных процедур:

- в названии имеется имя класса;
- все методы автоматически получают аргумент self (аналог this в Java);
- для вызова методов используется оператор «стрелка» -> .

При наследовании структур классов в IDL имена полей должны быть уникальны, а имена методов могут повторяться. Жизненный цикл объекта в IDL делится на три этапа:

- создание и инициализация объекта;
- использование объекта;
- удаление объекта.

Для создания объекта необходимо вызвать функцию-контейнер OBJ\_NEW, имеющую следующий синтаксис:

```
A=OBJ_NEW( [Имя_класса[, аргументы]] )
```

При этом будет вызван соответствующий метод конструктор (если есть), который имеет строгое имя – INIT и является членом-класса.

Для удаления объектов используется процедура OBJ\_DESTROY. Если в классе (или одном из суперклассов) есть метод CLEANUP, то будет вызван этот метод с переданными аргументами. Если такого метода нет, то IDL сам удаляет объект. Метод CLEANUP является аналогом метода-деструктора. При этом, как метод INIT, так и метод CLEANUP нельзя вызвать не используя OBJ\_NEW и OBJ\_DESTROY соответственно.

### ***Визуализация научных данных средствами IDL***

Для работы с изображениями IDL снабжен существенным количеством подпрограмм, позволяющим проводить обработку изображений. Подробно вопросы обработки изображений изложены в работах [2, 3]

IDL может работать с 4 типами изображений [4, 5]:

- бинарные (0 – черный, 1 – белый);
- в тонах серого (0 – черный, 255 – белый);
- индексированные (с применением Lookup tables (LUT));
- RGB (каждый компонент 0-255).

Для отображения в IDL могут использоваться две системы графики: Object Graphics и Direct Graphics. Эти системы подробно рассмотрены в работах [6, 7] соответственно.

Результаты обработки данных СВЧ-радиометрии необходимо привязать к картографическим координатам в соответствующих картографических проекциях.

Картографические проекции преобразуют координаты в координаты на плоскости. Существует много картографических проекций, каждая из которых обладает собственной техникой преобразования сферических координат в координаты на плоскости.

IDL имеет 19 встроенных картографических проекций. IDL обладает возможностью создания картографических проекций, конвертированием данных из одной проекции в другую и отображением данных в этих проекциях.

Картографическая база данных, которая поставляется с IDL от Всемирного банка данных (WDB), является достаточно полной для отображения политических границ, береговых линий и других земельных/водных границ. Пример визуализации картографической информации приведен на Рис. 1



**Рис. 1.** Центрированное изображение в масштабе 1:20.000.000(о-в Гренландия)

### **Проектирование графического интерфейса пользователя средствами IDL**

IDL позволяет создавать графические интерфейсы пользователя с помощью виджетов. Виджеты (графические элементы управления) – кнопки, скроллы, таблицы, метки и т.д. Графический интерфейс,

реализованный на IDL, представляет собой древовидную иерархию – каждый виджет имеет одного «родителя» и ни одного или несколько «потомков», с одним исключением, что верхний виджет иерархии не имеет «родителя». Этот виджет называется базовым.

Любое виджет приложение должно начинаться с базового виджета. При создании виджетов только создаются соответствующие структуры данных, которые на экране не отображаются. Для того, чтобы они визуализировались, необходимо вызвать соответствующую процедуру, после выполнения которой виджеты, входящие в состав иерархии, отобразятся на экране.

Жизненный цикл приложений с виджетами:

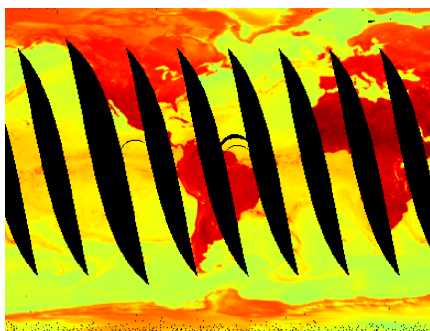
1. Создание иерархии виджетов.
2. Создание подпрограммы обработки событий.
3. Реализация виджетов.
4. Регистрация программы с помощью XMANAGER, которая активизирует функции обработчики событий.
5. Взаимодействие с приложением.
6. Уничтожение виджетов.

Виджеты могут обладать пользовательским значением. То есть каждый виджет содержит переменную, в которой хранится необходимая информация. Это значение игнорируется виджетом и требуется для удобства программиста при разработке программ.

Для обмена данными в приложениях с графическим интерфейсом пользователя необходимо создать структуру, которая будет содержать ID виджета и дополнительные данные. Затем создать ссылку на нее и передать ее как пользовательское значение в базовый виджет. В обработчике событий необходимо получить эту информацию из полученной структуры события.

### ***Приложение обработки данных СВЧ-радиометров***

Входными данными для приложения метеорологической обработки данных СВЧ-радиометров являются данные в формате HDF.



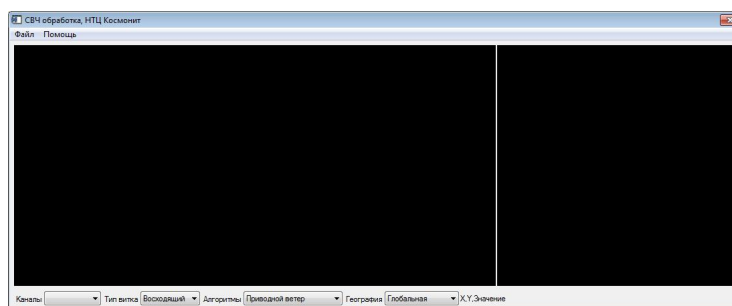
***Рис. 2. Фрагмент обзорного снимка СВЧ-радиометра***

Формат данных HDF получил большое распространение в обработке и хранении данных ДЗЗ. Он стал своего рода стандартом для хранения и передачи информации в зарубежных системах ДЗЗ. Полное наименование -

формат иерархических данных (Hierarchical Data Format). Организация-разработчик: Национальный центр суперкомпьютерных приложений - National Center for Supercomputing Applications (NCSA).

Исходные файлы представляют собой обзорные снимки земной поверхности (т.е. компиляция всех полувитков за сутки). Размер такого изображения составляет 720x360 пиксел.

Приложение имеет графический интерфейс пользователя, пример которого приведен на Рис. 3. Графический интерфейс пользователя позволяет проводить интерактивную обработку данных СВЧ-радиометрии. Приложение построено по модульному принципу, что позволит расширить список алгоритмов обработки в будущем.



*Рис. 3. Графический интерфейс программы обработки*

Алгоритмы определения метеорологических параметров базируются на эмпирических зависимостях, которые были получены в работах [8,9,10]. С учетом этих зависимостей, количественные значения метеорологических параметров определяются с помощью линейной комбинации каналов СВЧ-радиометра различной поляризации.

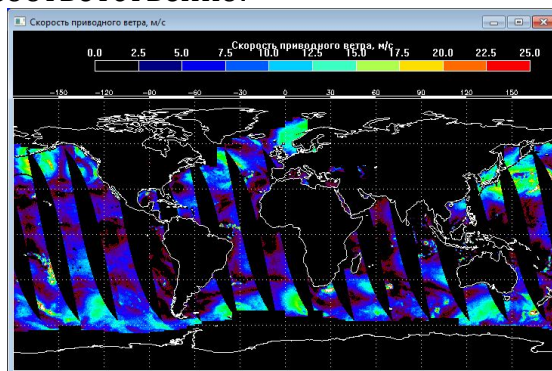
1. Алгоритм определения скорости ветра над океаном позволяет получить скорость ветра поверхности океана от 3 до 25 м/с с точностью не хуже 2 м/с. Скорость ветра у поверхности приводится к высоте 19,5 м над поверхностью. Небольшой дождь и пары воды значительно ослабляют микроволновое излучение на выбранных частотах равных 19, 22 и 37 ГГц. В частности СВЧ-излучение, испускаемое с поверхности океана, содержит информацию о скорости ветра [8].

2. Алгоритм определения водозапаса над океаном был разработан и опубликован в работе [9]. Этот алгоритм определяет интегральное содержание воды в облаках над океаном в  $100 \text{ кг/м}^2$  с точностью до  $0,1 \text{ кг/м}^2$  с интервалом квантования  $0,5 \text{ кг/м}^2$ .

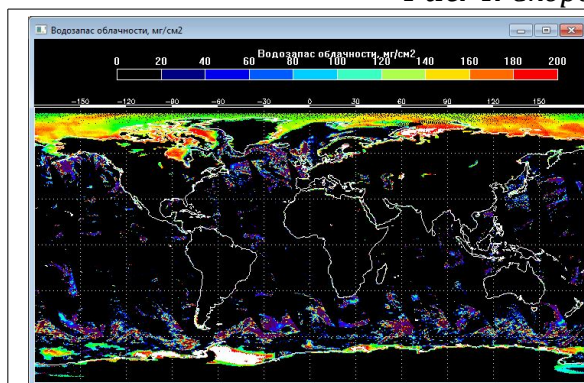
3. Алгоритм определения интенсивности осадков над океаном были разработаны и опубликованы в работе [10]. Алгоритм использует каналы с частотами 19V, 19H, 22V, 37V, 37H, 91V, и 91H и приведен к высоте 12,5 км. Данный алгоритм имеет точность 5 мм/ч, интервал квантования 1 мм/ч.

В результате работы программы обработки вычисляются значения

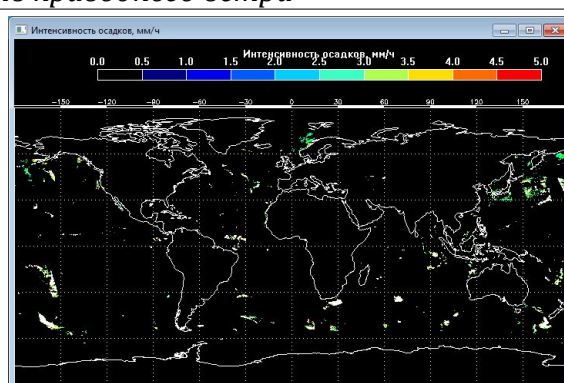
соответствующих метеорологических параметров, которые визуализируются и наносятся на картографическую основу. Результат сохраняется в формате JPEG. Примеры полученных результатов обработки: скорость приводного ветра, водозапас облачности, интенсивность осадков приведены на Рис. 3-5 соответственно.



**Рис. 4. Скорость приводного ветра**



**Рис. 5. Водозапас облаков**



**Рис. 6. Интенсивность осадков**

Таким образом, рассмотрены возможности языка IDL, отражающие особенности при описании классов и объектов, визуализации научных данных, в том числе с применением картографических возможностей. Эти особенности были учтены при создании приложения, обладающего графическим интерфейсом пользователя, обработки данных СВЧ-радиометров для определения метеорологических параметров.

### Литература

1. John R. Jensen. Remote sensing of the environment: An Earth resource perspective (2nd Edition). Prentice Hall, 2006. 608 p.
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2005. 1072с.
3. Jonh R. Jensen, Introductory Digital Image Processing (3rd Edition). Prentice Hall, 2004. 544p.
4. Liam E. Gumley. Practical IDL Programming. Morgan Kaufmann, 2001. 508 p.
5. Kenneth P. Bowman. An Introduction to Programming with IDL: Interactive Data Language. - Academic Press, 2005. 304p.
6. Ronn Kling. Power Graphics with IDL: A Beginners Guide to IDL Object Graphics. - KRS, inc, 2002. 74p.

7. Fanning, D. Coyote's Guide to Traditional IDL Graphics. – Coyote Book Publishing, 2011. 480 p.
8. M.A. Goodberlet, C.T. Swift, J.C. Wilkerson. Ocean Surface Wind Speed Measurements of the Special Sensor Microwave/Imager (SSM/I) // IEEE Transactions on Geoscience and remote sensing. Vol. 28, №5, 1990.
9. Weng, F, N. Grody, R. Ferraro, A. Basist, and D. Forsyth, 1997: Cloud liquid water climatology from the Special Sensor Microwave Imager, J. Climate, 10, 1086-1098.
10. Ferraro, R.R., 1997: SSM/I derived global rainfall estimates for climatological applications. J. of Geophys. Res, 102, 16,715-16,735.



## ***Системы символьной математики как технологические и математические инструменты инновационного учебного процесса***

### **Аннотация**

*Предлагается концепция современного инновационного учебного процесса на базе использования систем символьной математики как технологических и математических инструментов. Обсуждаются роль этих систем в физико-математическом образовательном процессе. Описываются особенности основных систем символьной математики применительно к проблемам учебного процесса в курсах физики.*

### ***1. Системы символьной математики и их роль в учебном процессе.***

Современные науку и образование уже невозможно представить без повсеместного использования компьютерных технологий, вычислительной техники и различного математического обеспечения. Однако важным остается вопрос, как и в каком качестве их использовать при решении основной задачи высшей школы – *подготовке высококвалифицированных научных и инженерных кадров, способных решать современные научные и практические проблемы.*

Компьютерные ресурсы необходимые для успешного решения такой образовательной задачи должны удовлетворять определенным требованиям: обеспечивать преемственность и фундаментальность физико-математического и инженерного научного образования, не замыкаться на трудоемкое общения с компьютером и обладать самыми мощными и адекватными математическими возможностями.

Таковыми инструментами являются системы символьной математики – интерактивные многофункциональные компьютерные системы высокого интеллектуального уровня, сочетающие простоту использования с мощностью самых современных математических инструментов. Их справочные материалы [3-8], содержат не только сведения по работе, но и развернутые современные учебники математики, справочники по физике и другим дисциплинам. Поэтому системы символьной математики являются интерактивной учебно-исследовательской виртуальной средой.

Работа в такой среде предполагает невольное «подтягивание» пользователя до необходимого уровня, что является своеобразной дополнительной образовательной технологией. Такие системы получили всемирное признание в научных и академических кругах как инструменты



научных исследований и прикладных инженерных расчетов; но они только начинают использоваться в образовании. При этом системы символьной математики обладают большой образовательной эффективностью, и опыт их развития указывает на перспективы их применения в учебном процессе.

Несмотря на «математическое происхождение» систем, они адекватны потребностям физико-математического и инженерного образования, как средства, содержащие самые разнообразные математические инструменты, необходимые для физических и прикладных исследований: от обработки массивов данных до квантовых систем и космологии.

Сами по себе системы символьной математики не являются средствами, «автоматически» поднимающими математический уровень пользователей: они могут служить полезными инструментами как при компьютерном решении задач, так и в самом процессе изучения физико-математических дисциплин (причем, эти стороны использования систем символьной математики, конечно же, взаимно обусловлены).

Не заменяя преподавателя и не подменяя стандартный учебный процесс, но требуя от пользователя необходимого уровня математической подготовки для общения на входных языках и составления аналитико-численных алгоритмов решения задач, системы символьной математики (имеющие встроенные инструменты верификации, в том числе и пошаговой, и обладающие обратными связями в виде интерактивных комментариев и подсказок), могут служить, с одной стороны, своеобразными контролерами качества математических выкладок, а с другой – обеспечивать учебный процесс информационного типа.

Соответствие задач физико-математического и инженерного образования и возможностей систем символьной математики делает последние естественной компьютерной средой обучения и учебно-исследовательской деятельности. Действительно, учебный процесс требует таких компьютерных средств, которые обеспечивали бы адекватный уровень математических вычислений и не требовали бы специальной программистской подготовки и трудоемкого освоения математического обеспечения.

В справочных материалах систем символьной математики имеются, помимо прочего, имеются развернутые современные учебники по математике, справочники по математике, физике и другим дисциплинам: их можно рассматривать как учебно-исследовательские среды.

Использование системы символьной математики в образовании:

- создает особую информационную среду обучения, основанную на современных компьютерных технологиях;
- позволяет органично сочетать фундаментальность образования с интерактивными и дистанционными формами обучения;
- обеспечивает профессиональную преемственность обучения, основанную на академической и научной практике работы с ССМ.

Системы символьной математики дают возможность проводить обучение деятельностного типа, изменить общение студентов с преподавателем и студентов между собой, использовать новые формы выполнения практических и домашних заданий. Они дают возможность преемственного подхода к традиционным практическим занятиям: с одной стороны на компьютере можно решать традиционные задачи, а с другой решение таких задач может быть существенно расширено за счет возможностей компьютера (подробный анализ полученного решения: наглядное представление результатов в аналитической и графической формах).

Рабочие листы систем представляют собой так называемые «живые электронные книги»: вычисления, преобразования, построения графиков и т.п. происходит интерактивно, а пользователь имеет возможность вносить свои поправки в значения параметров, изменять функции и делать дополнения (все это сразу или по команде отражается на рабочем листе). Все это делает системы символьной математики настоящим интерактивным учебным пособием.

Использование систем символьной математики в учебном процессе обеспечивает также дополнительные возможности в:

- применении индивидуальных и групповых формах выполнения заданий;
- организации выполнения заданий в очной и дистанционной формах;
- формулировке систем персональных заданий для каждого студента различных степеней сложности, учитывающих его образовательный уровень и скорость усвоения материала;
- расширении круг решаемых задач: от коротких типовых примеров до комплексных исследовательских заданий;
- развитии индивидуальных возможностей студентов.

Интегрирование систем символьной математики с системами администрирования и контроля может обеспечить все потребности современного учебного процесса.

## **2. Особенности основных систем символьной математики**

Для обеспечения учебного процесса можно рекомендовать системы MATHCAD, MAPLE, MATHEMATICA, что связано с простотой их использования и нацеленностью не только на научные исследования, но и учебный процесс. Поскольку принципы работы с системами символьной математики методически проработаны, возможно их постепенное освоение в ходе самого учебного процесса. Рекомендации по выбору MATHCAD, MAPLE, MATHEMATICA для обеспечения учебного процесса связан с простотой их использования и их нацеленностью не только на научные исследования, но и на учебный процесс.

MATHCAD наиболее прост в освоении и использовании; обладает наиболее дружелюбным интерфейсом и требует самых минимальных

инструкций, необходимых для того, чтобы уже первом занятии начать решать конкретные задачи: MATHCAD можно рекомендовать для использования на начальных этапах учебного процесса – для студентов младших курсов (в том числе и не обладающих навыками работы в сложных компьютерных системах).

MARLE и MATHEMATICA более сложные системы, обладающие самыми мощными вычислительными ресурсами, способными обеспечить любой математический уровень современного учебного процесса. Освоение и использование MARLE и MATHEMATICA требует более основательной математической и программистской подготовки: их можно рекомендовать для использования на последующих этапах обучения – студентам средних и старших курсов и аспирантов (при этом навыки работы в MATHCAD оказываются полезными для освоения MARLE и MATHEMATICA в виду общего подхода к работе в системах символьной математики – использования входных языков и программирование хода решения задач с использованием символьных процессоров).

Системы символьной математики способны обеспечить весь спектр математических операций, необходимых при изучении физико-математических дисциплин: алгебраические и символьные преобразования, дифференцирование и интегрирование, решение уравнений и систем уравнений, разложение функции в ряд, работа с векторами и матрицами, операции с комплексными функциями, использование булевых операций, построение двумерных и трехмерных графиков и др. По сравнению с MATHCAD системы MARLE и MATHEMATICA обладают расширенными возможностями и, помимо указанных операций, способны символьно решать обыкновенные дифференциальные уравнения, системы дифференциальных уравнений, уравнения в частных производных; в них реализованы возможности тензорного анализа, работы со специальными функциями и многие другие средства. Это означает, что в тех случаях, когда для решения задач теоретической физики возможностей MATHCAD не хватает, можно использовать системы MARLE и MATHEMATICA.

При этом профессиональный подход к вычислительным задачам, особенно при получении новых результатов, предполагает (когда это возможно) использование разных систем символьной математики и проверку результатов с помощью разных систем и подходов. Современный учебный процесс для широкого круга будущих научных работников и инженеров должен обеспечивать овладение практическими навыками проведения как натурного, так и вычислительного эксперимента с высокотехнологичными методами получения, обработки и анализа экспериментальных и теоретических данных. Это, в свою очередь, предполагает использование современного программного обеспечения, умения осуществлять как численные, так и аналитические вычисления, анализ и визуализацию результатов.

Во многих случаях это способны обеспечить системы символьной

математики, с помощью которых возможно осуществить в процессе обучения:

- 1) цифровой метод получения и компьютерная обработка данных разного типа;
- 2) аналитические вычисления и преобразования;
- 3) математическое компьютерное моделирование (от использования шаблонов моделей до модельных построений разного уровня);
- 4) вычислительный эксперимент (изучение процессов для разных условий и параметров);
- 5) интерактивная визуализация явлений и процессов (в том числе тех, которые недоступны для традиционного форм обучения).

Таким образом, системы символьной математики являются естественными современными программными ресурсами для обеспечения учебного процесса информационно-инновационного типа. Они способны обеспечить преемственность и фундаментальность образования на базе современного программного обеспечения.

Практическое применение систем символьной математики в учебном процессе можно осуществить на базе комплексов программ, использующих, с одной стороны широкие возможности самих систем, а с другой – программно реализующих общие и специальные методы физико-математических дисциплин, их содержательную часть и аналитико-теоретические подходы.

Заметим, что возможности систем символьной математики позволяют (что, собственно, отражает методологические основы теоретической и математической физики, и что возможно реализовать в системах) построить ход решения (или их отдельные этапы) физических и математических задач в виде программных блоков и комплексов, которые можно трактовать как универсальные реализации решений общих теоретических и учебных проблем, так и рассматривать как составные части для создания комплексов для решения крупных прикладных проблем.

В связи с этим разрабатываемые программные комплексы можно использовать на лекциях, практических и лабораторных занятиях (в виде компьютерного практикума моделирующего типа с использованием систем символьной математики) в курсах элементарной, общей, теоретической и математической физики; они могут стать универсальными учебно-исследовательскими ресурсами, применяемыми студентами и аспирантами при выполнении учебно-исследовательских курсовых и дипломных работ и диссертаций.

В настоящее время большое число вузов России используют в учебном процессе ССМ (включая, MATHCAD, MAPLE и MATHEMATICA MATHLAB). Среди университетов, строящих подготовку кадров на основе ССМ – самые престижные вузы мира, например, Princeton University; California Institute of Technology; University of Oxford и т.п.)

Технологиями, основанными на использовании в образовании систем символьной математики, охватывается практически весь спектр академических дисциплин в области точных наук, а также ряд гуманитарных наук.

Системы символьной математики используются:

- при проведении практических занятий и лабораторных работ;
- в виде «живых» электронных справочников и книг
- как рабочие материалы для банка задач и выполнения домашних и дистанционных заданий;
- как универсальные средства для создания учебных пособий и наглядных визуализаций.

Тем не менее, существует большой ресурс для использования ССМ в образовании, связанный:

- с универсализацией учебных материалов (и доведения их до уровня, соответствующего фундаментальным педагогическим принципам);
- с разработкой технологических основ многоцелевого и многовариантного учебного процесса, сочетающего очные и дистанционные формы обучения и обеспечивающие индивидуальность «траектории обучения» для каждого студента;
- с расширением области применения ССМ для дисциплин, использующих сложный математический аппарат;
- с развитием подходов, обеспечивающих естественную связь учебно-научной деятельности и исследовательской работы;
- с обеспечением преемственного поступательного процесса подготовки кадров, включающего профориентацию школьников, реализацию мотивационных потребностей студентов и развитие профессиональных компетенций молодых специалистов и ученых.

### ***3. Компьютерный практикум с использованием систем символьной математики***

Требования, предъявляемые к современным специалистам, и развитие компьютерные технологии обосновывают необходимость внедрения в учебный процесс новых типов занятий. Одним из них являются практические занятия в форме компьютерного практикума по физике. Современный ученый или инженер должен обладать не только навыками экспериментальной деятельности, но и методиками моделирования и теоретического анализа.

В науке и промышленности реализуются подходы, когда строительство дорогостоящего, например энергетического, объекта, не начинается без построения многочисленных моделей и проведения виртуальных экспериментов по изучению свойств объекта и его поведения в экстремальных ситуациях, а все современные научные данные получают на установках, которые интегрированы в компьютерную среду.

Следовательно, современный физический практикум уже сейчас должен сочетать овладение практическими навыками проведения

эксперимента с высокотехнологичными методами получения, обработки и анализа экспериментальных данных.

Поэтому компьютерный практикум по общей физике создан не как альтернатива традиционному физическому практикуму, а как новая технология обучения. Он является инновационной аналитической частью общего физического практикума, интегрированной с практическими занятиями и другими видами учебной деятельности.

Основные идеи, положенные в основу проведения компьютерного практикума:

1. Компьютерный метод получения и обработки экспериментальных данных.

2. Компьютерное моделирование физических явлений разного типа.

3. Численный эксперимент для физических процессов.

4. Визуализация физических явлений и процессов.

Компьютерный практикум выполняется студентами первого, второго и третьего курсов, изучающих общую физику.

В зависимости от специальности и количества времени, отводимого на изучение физики, могут отличаться количество тем и объем заданий практикума.

Методическим обеспечением компьютерного практикума по общей физике является:

– учебные пособия по практикуму (изданные в традиционном виде) [1, 2];

– электронные материалы: программы и комплексы программ в виде рабочих листов систем.

Учебные пособия и электронные материалы содержат:

– теоретическое введение по изучаемой теме, содержащее основные законы и формулы;

– задания компьютерного практикума с указанием последовательности и методов выполнения;

– формулировки методов исследования изучаемых явлений;

– примеры выполнения заданий и отчетов;

– вычислительные примеры и шаблоны.

Для сопровождения компьютерного практикума необходимо учебное пособие [1] по использованию специализированных систем символьной математики для решения физических задач. Оно содержит необходимые для физики приемы работы с системами символьной математики, математические примеры вычислительных операций, основные сведения и примеры обработки экспериментальных данных.

Необходимым техническим обеспечением компьютерного практикума является:

– персональные компьютеры, компьютерные классы, дополнительное цифровое оборудование;

– системы символьной математики и вспомогательные

компьютерные программы.

Компьютерный практикум по общей физике можно реализовать в системах MATHCAD, MAPLE, MATHEMATICA. Основной математической системой практикума является MATHCAD: он наиболее прост в освоении и обладает всеми необходимыми для задач общей физики инструментами. MATHCAD можно рекомендовать для студентов с двухгодичным курсом общей физики и небольшим количеством учебных часов. Для студентов физических и физико-технических специальностей можно рекомендовать использование более сложных систем MAPLE и MATHEMATICA. Практически использование ССМ можно начать с MATHCAD для всех студентов первых курсов с постепенным подключением систем MAPLE и MATHEMATICA на втором и третьем курсах.

Структура компьютерного практикума должна соответствовать учебным планам курсов общей физики и охватывать все основные разделы программы: классическая механика; механические колебания и волны; термодинамика и молекулярная физика; электричество и магнетизм; оптика; квантовая физика.

В рамках компьютерного практикума может естественным образом проводиться обработка экспериментальных данных, включающая: вычисления доверительного интервала; вычисление косвенных погрешностей; построение графиков (с использованием полиномиальной регрессии и других методов).

Использование компьютерных методик обработки экспериментальных данных означает не только упрощение работы при сложных и громоздких вычислениях, но и возможность дополнительного анализа данных, следствием которого могут быть дополнительные измерения или необходимость более тщательного выполнения работы. Использование ССМ дает возможность представлять отчеты не только с полными аналитическими выкладками и численными расчетами, но и проконтролировать все шаги выполняемой работы.

#### ***4. Концепция инновационного учебного процесса на базе систем символьной математики***

В физико-математических и инженерных дисциплинах основные законы, теоремы, зависимости параметров и т. д. представляются в виде функциональных аналитических соотношений – символьных формул. И именно символьные вычисления и преобразования составляют наиболее существенную содержательную часть всей вычислительной работы в этих дисциплинах [3-7].

До сих пор учебный процесс, обеспечивающий изучение таких дисциплин, в основном, обходится без использования компьютерных технологий в части использования символьных вычислений: существующие учебные материалы либо являются оцифрованными учебниками традиционного типа (различного программного воплощения), либо представляют собой отдельные программные ресурсы для численных

вычислений, построения графиков и т.п. При этом системы компьютерного тестирования в области точных наук основаны на контроле «текстовых выражений» или численных значений и на выборке правильного результата из нескольких формул-картинок.

Это означает существенное ограничение образовательного потенциала компьютерных технологий в области точных наук и приводит, в свою очередь, к подходу типа «делаем как допускает система, а не как того требует содержание дисциплины», что особенно наглядно в попытках компьютерной проверки знаний, представленной естественными символьными соотношениями, в виде не всегда корректных их текстовых описаний или представлений и зачастую проверяющих не сами знания, а лишь демонстрирующих несовершенство компьютерной системы или программы, неспособной адекватно представить и проверить материал точных наук.

Другая проблема, которая имеет место в учебном процессе в области преподавания точных наук, – обеспечение интерактивности учебных материалов (что совершенно естественно для таких предметов) и организация дистанционного учебного процесса. Знание основных законов в виде формул и соотношений требует выполнения многочисленных выкладок и преобразований, умения выводить формулы и применять их для разных ситуаций, «традиционные компьютерные технологии» способны обеспечить лишь «традиционные формы» представления знаний.

С другой стороны «несовершенство» таких подходов ограничивает дистанционный учебный процесс в части обмена информацией (огромные по размеру файлы, невозможность проверки заданий в автоматическом режиме и т.п.) между учащимся, учебным заведением и преподавателем: существенная часть учебных материалов (задания, контрольные и т.п.) существует в бумажном виде и пересылается обычной почтой или по электронной почте в виде отсканированных страниц и/или материалов, выполненных в текстовых редакторах.

Эти проблемы естественным образом могут быть решены использованием в учебном процессе систем символьной математики, которые в состоянии полностью обеспечить сопровождение преподавания точных наук: интерактивный тип материалов, малый объем пересылаемых электронных файлов-рабочих листов, электронный банк заданий и типовых задач, процесс обучения с представлением материала в аналитическом виде (с учетом других обширных возможностей систем символьной математики), реальное дистанционное взаимодействие студент – преподаватель, а также контроль знаний в естественном математическом символьно-формульном виде.

Именно такой подход, основанный на использовании современных систем символьной математики и который, с одной стороны, реализует все базовые фундаментальные принципы образования физико-математических и инженерных курсов, а с другой – обеспечивает учащихся



и преподавателей самыми современными математическим и технологическими компьютерными средствами, сочетающими аналитические вычисления, численный анализ и качественную визуализацию данных и результатов, составляет содержание инновационного учебного процесса.

Отметим, что компьютерные технологии в образовании могут быть эффективными только при соответствии программных возможностей математического обеспечения, качества электронных учебных материалов и профессионального уровня преподавателей (как в предметной области, так и в области использования компьютерных технологий).

Реализация такого инновационного учебного процесса может быть осуществлена посредством разработки и внедрения программных комплексов на основе систем символьной математики, их интеграцией в систему преподавания, администрирования и контроля, а также с адаптацией этих комплексов с другими компьютерными технологиями, сопровождающими современный учебный процесс [1-7].

### **Литература**

1. Тихоненко А.В. Компьютерные математические пакеты в курсе общей физики. Обнинск: ИАТЭ, 2003. 84 с.
2. Тихоненко А.В. Компьютерный практикум по общей физике. Части 1 - 5. Обнинск: ИАТЭ, 2003-2004.
3. Тихоненко А.В. Компьютерные математические пакеты в курсе «Линейные и нелинейные уравнения физики». Части 1 и 2. Обнинск: ИАТЭ, 2005.
4. Тихоненко А.В. Решение краевых задач для двумерного уравнения Лапласа методом разделения переменных в MAPLE. Обнинск: ИАТЭ, 2005. 80 с.
5. Тихоненко А.В. Векторный анализ в прикладных математических пакетах. Обнинск: ИАТЭ, 2006. – 80 с.
6. Тихоненко А.В. Решение уравнения Шредингера для одномерного рассеяния в MAPLE и MATHEMATICA. Обнинск: ИАТЭ, 2005. 80 с.
7. Тихоненко А.В. Компьютерные аналитические методы решения задач электростатики и магнитостатики. Обнинск: ИАТЭ, 2008. 48 с.

**Толстолицкий В.Ю.,**

ННГУ им. Н.И. Лобачевского, профессор кафедры уголовного процесса  
и криминалистики юридического факультета  
[tolvlad@yandex.ru](mailto:tolvlad@yandex.ru)

**Кузенкова Г.В.**

ННГУ им. Н.И. Лобачевского, доцент Центра прикладной  
информатики факультета ВМК  
[kuzenkovagv@mail.ru](mailto:kuzenkovagv@mail.ru)

### ***Программное обеспечение в обучении следователей раскрытию убийств***

Повышение эффективности работы следователей в современных условиях практически невозможно без внедрения в их деятельность новых информационных технологий и современных средств компьютерной техники. В качестве шага к информатизации деятельности следователя по решению задачи раскрытия убийств в ННГУ им. Н.И. Лобачевского была разработана компьютерная программа «ФОРВЕР» (авторы-разработчики: В.Ю. Толстолицкий, С.Н. Карпенко, А. Рыбочкин). В основе работы программы лежит обработка базы раскрытых уголовных дел и выдвижение версий о признаках преступника, которые составляют в совокупности его поисковый портрет.

Данная программа позволяет следователю сформировать вероятный портрет преступника, который затем становится ориентиром для поиска реального лица, совершившего преступление [1, 2]. Самый простой способ использования программы «ФОРВЕР» сводится к методу аналогии, то есть поиску аналогичного убийства в компьютерной базе данных. Этот метод полностью повторяет мыслительный процесс следователя. Когда число раскрытых конкретным следователем убийств переваливает за определенную величину, то его «опытность» выражается в обнаружении повторяемости во вновь расследуемых преступлениях некоторых сторон, ранее уже наблюдавшихся им в преступлениях прошлых лет. При таком подходе обращение к программе позволяет просмотреть всю базу уголовных дел и найти подобные преступления, а при необходимости поднять их из архива.

Поиск осуществляется простой сортировкой всех занесенных в электронную базу дел по признакам, указанным следователем, например, по полу и возрасту жертвы, способу убийства и сокрытия следов. Набор признаков позволяет отсортировать группу уголовных дел, аналогичных между собой и раскрываемому делу. В компьютерной литературе данный прием называют методом «поиска ближайшего соседа» - case based

reasoning – CBR [3]. Метод аналогии позволяет принимать решение независимо от того, сколько архивных уголовных дел найдено. Достаточно всего одного аналога. Данный метод не основан на статистических данных, поэтому для него не является ограничением проблема малого числа наблюдений. Кроме указанного метода поиска по аналогии, программа «ФОРВЕР» может быть использована более сложными способами. В том числе, для расчета условных вероятностей и принятия решений на основе основной теоремы теории вероятностей – теоремы гипотез (Байеса).

Таким образом, применение программы позволяет выделить информационные процессы в качестве самостоятельного предмета анализа и отделить работу с информацией от принятия процессуальных решений, организации расследования и других сторон раскрытия и расследования убийств. Это позволяет, во-первых, добиться качественного улучшения информационного обеспечения раскрытия и расследования преступлений и, во-вторых, освободить следователей, постоянно работающих в условиях высокой нагрузки, от выполнения значительного объема рутинной технической работы, требующей больших временных затрат. Программа прошла апробацию Следственным комитете РФ и принята для изучения приказом Министерства образования и науки Российской Федерации от 29 ноября 2011 г. N 2763 «О федеральных государственных требованиях к минимуму содержания дополнительной профессиональной образовательной программы профессиональной переподготовки федеральных государственных гражданских служащих Следственного комитета Российской Федерации, назначенных на должности следователей следственных отделов Следственного комитета Российской Федерации по городам, районам и приравненным к ним специализированных, в том числе военных следственных отделов».

При внедрении в практику компьютерной программы «ФОРВЕР» был обнаружен ряд проблем:

1. Следователи, не выделяют раскрытие в самостоятельное действие. Поэтому требуется создание ситуационных задач, в которых следователь-практик осознает самостоятельность выполняемого им действия. Необходимо разъяснить, что объективно существующее действие следователя, это не только те следственные действия, которые перечислены в уголовно-процессуальном законе РФ, а так же объясняются причины отсутствия в учебниках криминалистики такого действия и причины отсутствия указания в УПК РФ такой обязанности следователя, как раскрытие преступления. Ссылаясь на закон об оперативно розыскной деятельности (п. 2.), следователь полагает, что раскрытие входит в обязанности только оперативных сотрудников. Сложность внедрения программы заключается в том, что требуется еще убедить следователя, что новому действию надо обучать его, а не оперативника.

2. Ряд следователей считает, что осмотр трупа является только фиксацией непосредственно воспринимаемого при визуальном осмотре.

Выдвижение версий (с помощью программы или без нее) лишь мешает объективной и непредвзятой оценке обстановки. Поэтому необходимо разъяснять и демонстрировать на примерах повышение информативности осмотра с помощью программы, позволяющей выдвигать следственные версии.

3. Многие следователи при выдвижении версий не пользуются понятийным мышлением. В результате, восприятие подчиняется ситуации, то есть непосредственному впечатлению. Применяется метод наглядно-образной реконструкции механизма преступления, который опирается преимущественно на чувственное познание. Попытка использовать одновременно систему версий в основе которой лежат вербально зафиксированные понятия (существенные признаки), сбивает с привычного способа выдвижения версий на основе образного мышления.

4. В четвертых, следователи не пользуются счетом, а воспринимают обстановку непосредственно. За счетом стоит понятие количественной стороны явлений, понятия вероятности и условной вероятности. Программа «ФОРВЕР» позволяет рассчитать условную вероятность признаков лица, совершившего преступление. Однако, для усвоения и использования понятия условной вероятности необходимо осуществлять подсчет встречаемости признаков и понимать смысл корреляционных взаимосвязей. Используя образное мышление и метод аналогии (собственный прошлый опыт), следователь подменяет частотную вероятность версии субъективной оценкой искомой вероятности и утверждает, что может достичь результата в поставленной педагогом задаче быстрее и проще. Субъективная вероятность первых двух наиболее вероятных версий, как правило, действительно очень близка к частотной вероятности, вычисленной по программе. Однако, оказывается не ясным то, на чем основан достигнутый результат, кроме того, следователь не может пересмотреть выдвинутые версии при получении новой информации по уголовному делу.

5. Сказывается отсутствие опыта работы с базами данных, что не только не позволяет понять частотную суть условных вероятностей, но имеет более негативные последствия. В силу того, что единственное криминалистически осмысленное действие следователь СКР осуществляет в виде заполнения карточек ИВЦ МВД по единому учету преступлений, то базы данных у него прочно ассоциируются лишь с уголовной статистикой. Поскольку никакой обратной связи он не получает от заполнения статистических карточек, то для него базы данных ассоциируются с бесполезной работой.

Кроме того, существенную роль при обучении играют следующие факторы: состав слушателей курса, изучаемый материал и время, отведенное на обучение. «Скоротечность» обучения и небольшой срок отрыва следователей от работы, когда, зачастую, они еще «погружены» в свои нераскрытые дела создавало установку, которая выражалась в двух

вариантах: во-первых, желании найти немедленного решения конкретной (оставленной дома) задачи, а во-вторых, в требовании высокой практической отдачи от занятия, на фоне неприятия теоретических знаний, особенно таких, которые требуют пересмотра привычного стереотипа раскрытия и расследования.

Таким образом, возникает проблема связи знаний и действий. Согласно работам Гальперина П.Я., Салминой Н.Г. [4, 5]: основу процесса усвоения составляет не восприятие, а действие (внешнее, практическое, или внутреннее, умственное), знания же – это всего лишь средства обучения действиям. С этой точки зрения, ожидания слушателей представляли собой дидактически обоснованные требования создать у каждого из них за время обучения новое действие, содержанием которого являлось раскрытие убийства на основе нового криминалистического средства – компьютерной программы «ФОРВЕР».

До обучения следователь использует в раскрытии преступления здравый смысл и бытовой характер восприятия обстановки. Эмпирически сформированное действие по раскрытию убийства характеризуется тем, что опирается на непосредственные впечатления следователя, получаемые им при осмотре места преступления. При этом за единицу анализа он воспринимает отдельное преступление. Действительно, в криминалистической литературе [6] подчеркивается уникальность конкретного преступления, обеспечивающая возможность идентификации субъекта, примененного орудия или оружия, следов преступления. Однако при раскрытии преступления, на первом плане стоит не идентификационная задача, а диагностическая (классификационная), то есть отнесения конкретного преступления к группе аналогичных, ранее раскрытых преступлений (весьма условно, ее можно отнести к установлению групповой принадлежности).

Для решения этой задачи следователю требуется выделить существенные признаки преступления. Перечень **криминалистически значимых признаков** составляет основу программы «ФОРВЕР». Но следователь не обучен выделению признаков, характеризующих следственную ситуацию и обстановку, в которой совершено преступление. Переход к мышлению с помощью признаков является основой обучения работе с программой и ее применения на практике.

Большие возможности для повышения эффективности обучения, по нашему мнению, дает теория поэтапного формирования умственных действий [7]. Эта теория рассматривает учение как систему определенных видов деятельности, осуществление которой приводит обучаемого к новым знаниям и умениям. Вначале знания формируются как результат внешней деятельности и постепенно становятся внутренними, проходя через определенный ряд этапов. Обучающийся овладевает новым умственным действием лишь тогда, когда он сам выполняет это действие, а не наблюдает со стороны. Однако для того у обучаемого должна иметься

ориентировочная основа действия, позволяющая выполнить его правильно. В любом действии можно выделить три функциональные части: ориентировочную, исполнительную и контрольную. Ориентировочная часть (основа) действия (ООД) выполняет следующие функции: выбор рационального приема исполнения действия, последовательность операций и качественное осуществление этой последовательности. Исполнительная часть связана с непосредственным преобразованием объекта действия. Контрольная часть обеспечивает сопоставление полученных результатов с заданными образцами, коррекцию ООД и исполнение действия.

На рисунке 1 представлена схема этапов действия при раскрытии преступлений (убийств, совершенных без свидетелей). Отличие данной схемы от стандартной процедуры – этап 3 – включение компьютерной программы как нового криминалистического средства. Самой трудной задачей в нашей работе оказалась задача по созданию обучающего сценария, в котором это новое средство стало неотъемлемой частью процедуры раскрытия преступления.

Рассмотрим этапы формирования умственного действия применительно к организации раскрытия преступления с применением компьютерной программы. На первом этапе задается мотивация к выполнению действия. Для этого может служить обсуждение путей выполнения задания группой обучаемых, так называемое погружение в привычную среду. Обучаемым выдаются фото-таблицы, полученные при осмотрах трупов. Далее наступает следующий этап – этап материального действия<sup>29</sup>, где обучаемые и выполняют действия с развертыванием все новых для них операций. После изучения фотографий мест происшествия, от следователя требуется преобразовать информацию из визуальной формы в систему признаков, которые несут криминалистически значимую информацию, содержащуюся в фото-таблицах. Только после этого, следователь начинает быстро ориентироваться в интерфейсе программы «ФОРВЕР» и может в ней составить запрос с помощью набора существенных признаков. Фактически, программа организует восприятие следователя, позволяет выделить информативные признаки, повысить информативность осмотра, рассчитать вероятности версий и условные вероятности признаков, составляющих портрет предполагаемого преступника. Вот здесь то и происходит переворот в мышлении следователя: метод аналогии работает; на первый план выходят количественные, а не качественные характеристики преступления.

Материальное действие сменяется третьим этапом его формирования – действием в виде громкой речи. При этом происходит

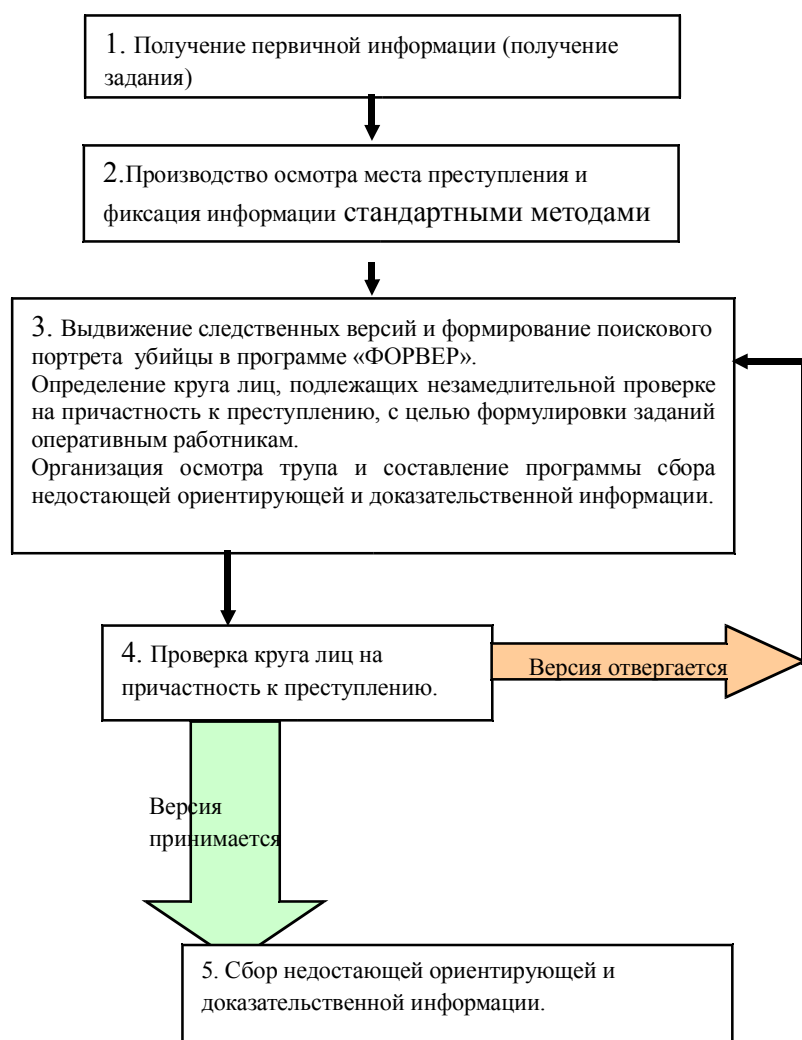
---

29 Отработке материальной формы действия в теории поэтапного формирования умственных действий придается чрезвычайно большое значение. Именно здесь происходит накопление фактологического материала, на основе которого выявляются необходимые связи и закономерности, различного рода противоречия и несоответствия между старыми и только что полученными знаниями.

дальнейшее обобщение компонентов материального действия. Для этого применяется обсуждение в группе обучаемых полученных результатов по второму этапу. Осмысление результатов второго этапа может потребовать возврата к нему, т.е. повторного применения программы уже с новыми уточненными криминалистическими признаками. Этап перевода действия в умственный план осуществляется во время самостоятельной работы обучаемых по выполнению аналогичных заданий и составления плана расследования.

**Пример задания:**

На руки слушателям выдается цветная фото-таблица из реального уголовного дела. Из фотографии № 1 видно, что потерпевшей является женщина около 40 лет, обнаруженная мертвой в своей квартире, на шее туго намотанное полотенце, лицо синюшно-багрового цвета. На фотографии № 2 виды в пепельнице окурки сигарет той марки, которую не курила потерпевшая.



**Рис. 1.** Схема этапов действия при раскрытии преступлений (убийств, совершенных без свидетелей)

Дополнительная информация: со слов дочери из квартиры ничего не похищено, дверь закрыта обычным образом на ключ, ключ хозяйки не обнаружен, потерпевшая проживала одна, не замужем, имела дочь.

Далее требуется выделить криминалистически значимые признаки из данного объема информации.

Полный перечень признаков должен быть следующим:

- Место совершения убийства – частная территория потерпевшей.
- Пол жертвы – женский.
- Возраст жертвы – 35-45 лет.
- Способ убийства – удушение (странгуляционная асфиксия).
- Мотив – предположительно бытовой.

При решении задачи слушатели, как правило (от 70% до 80% слушателей на курсе), ограничиваются указанием двух, а именно четвертым и пятым признаками. Остальные признаки не воспринимаются следователями как носители криминалистически значимой информации. Причиной является ориентированность следователя на поиск материальных следов преступления, тогда как пол и возраст жертвы к ним не относятся.

На следующем этапе слушателям предлагается составить поисковый портрет преступника. В этом случае слушатели отказываются выполнять задание, мотивируя это тем, что им не достаточно информации. Для решения поставленной задачи преподаватель показывает, как организуется работа по оценке фото-таблиц с помощью программы «ФОРВЕР». Слушатели видят весь перечень признаков, которые являются криминалистически значимыми в данной ситуации: место, пол, возраст, способ, мотив. Производится демонстрация работы программы с этими признаками. Результатом становится вероятностный портрет предполагаемого преступника: 37% - сожитель, 25% - знакомый, по 12,5% - муж, пасынок, сосед. Учитывая исходные данные для разработки остаются три версии: сожитель, знакомый, сосед. По тем же известным признакам программа позволяет установить, что преступник мужчина, не женат, ранее судим, в возрасте 29-45 лет.

После получения вероятностного портрета предполагаемого преступника слушатели должны запросить сведения, собранные оперативными работниками о характеристиках круга лиц, которые попадают под указанный портрет. Наибольшее сходство с вероятностным портретом обнаруживается у соседа по лестничной площадке, мужчины в возрасте 39 лет, неженатым, ранее судимым за аналогичное преступление по 105 ч.2, п.к. (убийство, сопряженное с изнасилованием или насильственными действиями сексуального характера).

Анализируя собственный опыт работы по обучению следователей применения программы «ФОРВЕР» в раскрытии убийств, можно сказать, что обучение следователей в качестве пользователя не решает задачи внедрения новых информационных технологий в деятельности раскрытия



и расследования преступлений. Нужно менять методику обучения по производству отдельных следственных действий. Изложенный выше подход создает основу решения дидактической задачи не только для программы «ФОРВЕР», но других компьютерных криминалистических средств.

### **Литература**

1. Толстолицкий В.Ю. Методология криминалистики в информационную эпоху. Н.Новгород: Изд-во ФГОУ ВПО «ВГАВТ», 2008. С. 110-120.
2. Толстолицкий В.Ю., Фесик П.Ю. Криминалистически значимые признаки, позволяющие определить пол преступника, при раскрытии убийств с помощью программы «Форвер» // «Черные дыры» в Российском законодательстве, 2009, № 4. С. 126-129.
3. Романов В.П. Интеллектуальные информационные системы в экономике: уч. Пособие. М.: Экзамен, 2007. С. 360.
4. Салмина Н.Г. Виды и функции материализации в обучении. М.: Изд-во МГУ. 1981. С.3.
5. Гальперин П.Я. Психология как объективная наука. М.: Институт практической психологии, 1998.
6. Ларин А.М. Криминалистика и паракриминалистика. М.: Изд-во БЕК, 1996. С. 117.
7. Талызина Н.Ф. Управление процессом усвоения знаний. М.: Из-во МГУ, 1975, 343с.

**Черняев С.И.,**

филиал МГТУ им. Н.Э.Баумана в г. Калуге, профессор  
[ambler@list.ru](mailto:ambler@list.ru)

**Семененко М.Г.,**

Финансовый университет при Правительстве Российской Федерации,  
филиал в г. Калуге, доцент  
[msemenenko@mail.ru](mailto:msemenenko@mail.ru)

**Кондратьева С.Д.**

филиал МГТУ им. Н.Э.Баумана в г.Калуге, доцент  
[ksd@pi8plus.ru](mailto:ksd@pi8plus.ru)

## ***Моделирование синергетических объектов с помощью облачных сервисов системы Mathematica***

### **Аннотация**

*В статье рассмотрены примеры моделирования самоорганизующихся систем с использованием облачных сервисов системы Mathematica*

### **Введение**

Появление компьютеров изменило все области научного познания. В «докомпьютерную» эру наука была «линейной», т. е., как правило, полученные уравнения рассматривались в линейном приближении. Поэтому получаемые результаты не давали полного представления об изучаемом явлении. Исследование нелинейных явлений привело к множеству интересных явлений, невозможных в рамках линейных представлений. Только нелинейные теории смогли описать солитоны и турбулентность в жидкости, цикличность экономического развития, взаимодействие народа и правительства и множество других естественных и социальных явлений. Подобные исследования привели к появлению новых терминов (например, введенных Б. Мандельбротом фракталов) и новой науки, названной Г. Хакеном синергетикой.

В настоящее время выходит множество книг, посвященных синергетике. Как правило, в этих книгах представлены непосредственные итоги самых разнообразных исследований, но не показано, как их можно получить на экране вашего персонального компьютера. В то же время современные системы вычислительной математики дают уникальную возможность специалисту в своей узкой области, отличной от вычислительной математики, повторить исследование, увидеть его результаты и, возможно, сделать собственные выводы и развить собственные идеи.

Ниже мы рассмотрим два классических примера

самоорганизующихся систем, моделирование которых можно провести, используя возможности мощной вычислительной системы Mathematica. На сайте wolfram.com размещены демо-версии некоторых программ, дающих представление о работе ресурса, а также интерактивные примеры, коллекция которых постоянно пополняется (demonstrations.wolfram.com). Эти примеры можно просчитывать, задавая различные параметры, загрузить файл на свой компьютер и работать с ним. Причем не обязательно на ПК должна быть установлена сама программа, поскольку в 2011 г. разработчик выпустил бесплатное приложение Wolfram CDF плеер, позволяющее запускать файлы Mathematica без установки программы. Плеер можно скачать с сайта по адресу <http://www.wolfram.com/cdf-player/>.

### **Брюсселятор**

Брюсселятором называется одномерная модель протекания химических реакций [1, 2]. Пусть вещества-реагенты А и В превращаются в продукты реакции D и E в цепочке промежуточных превращений:



Предположим, что протеканием обратных реакций в системе можно пренебречь, концентрации веществ А и В поддерживаются постоянными, а продукты реакции D и E либо химически инертны, либо немедленно удаляются из системы. Если движение частиц происходит только вдоль некоторой оси  $x$ , а изменением концентраций в плоскости, перпендикулярной этой оси, можно пренебречь, то мы имеем так называемую одномерную систему. Тогда, согласно законам химической кинетики [3], концентрации промежуточных продуктов X и Y описываются системой дифференциальных уравнений в частных производных:

$$\begin{aligned} \frac{\partial X}{\partial t} &= D_X \frac{\partial^2 X}{\partial x^2} + k_1 A - k_2 B X + k_3 X^2 Y - k_4 X, \\ \frac{\partial Y}{\partial t} &= D_Y \frac{\partial^2 Y}{\partial x^2} + k_2 B X - k_3 X^2 Y. \end{aligned}$$

Здесь  $k_{1-4}$  – константы скоростей прямых реакций, а  $D_X$  и  $D_Y$  – коэффициенты диффузии веществ X и Y соответственно. Большими буквами обозначаются концентрации соответствующих веществ.

В безразмерном виде написанную выше систему уравнений можно записать как

$$\begin{aligned} \frac{\partial X}{\partial \tau} &= D_X \frac{\partial^2 X}{\partial z^2} + A - (B + 1)X + X^2 Y, \\ \frac{\partial Y}{\partial \tau} &= D_Y \frac{\partial^2 Y}{\partial z^2} + B X - X^2 Y. \end{aligned}$$

Здесь  $\tau = t/k_4$ , координата  $z = x/L$ , где  $L$  – характерный размер системы (например, длина сосуда, в котором протекает реакция), концентрации веществ X и Y измеряются в единицах  $(k_4/k_3)^{1/2}$ , а концентрация вещества А – в единицах  $(k_4^3/k_1^2 k_3)^{1/2}$ .

Рассмотрим стационарное состояние системы, в котором

концентрации  $X$  и  $Y$  однородны в пространстве и не меняются с течением времени. Приравнявая нулю частные производные по времени в левой части рассматриваемой системы уравнений, получаем значения равновесных концентраций:  $X = X_0 = A, Y = Y_0 = B/A$ .

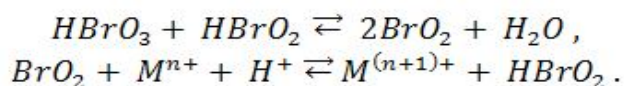
Анализ системы уравнений [4] показывает, что начиная с некоторого критического значения  $B$ , пространственно-однородные решения становятся неустойчивыми. В этом случае по истечении определенного времени (времени релаксации) система может приближаться к стационарному состоянию в виде немонотонных пространственно-периодических структур (так называемые диссипативные структуры). При увеличении параметра  $B$  в рассматриваемой системе можно также наблюдать возникновение периодических колебаний концентраций. Такие автоколебания в химических системах называют реакцией Белоусова-Жаботинского, по именам ученых, впервые наблюдавших их экспериментально.

Соответствующий численный эксперимент реализован на сайте [5]. Пример реализации показан на рис. 1. В коде программы использована новая опция Manipulate в системе Mathematica8, с помощью которой можно создавать интерактивные движки. Причем при открытии программы в CDF-плеере можно изменять «в живую» значения параметров, даже если сама система на компьютере не установлена. Изменяя параметр  $B$ , можно также наблюдать установление в системе равновесных концентраций.

### **Реакция Белоусова-Жаботинского**

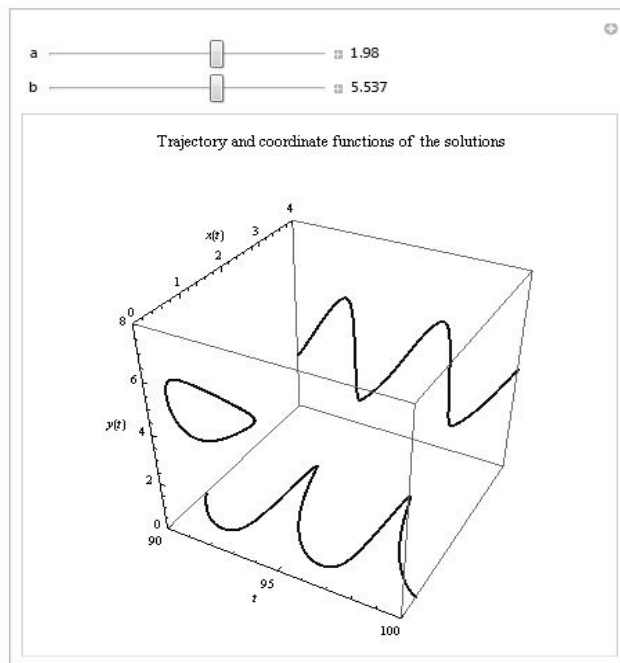
Драматическая история открытия химической колебательной реакции Белоусова-Жаботинского (известной в зарубежной печати как BZ-реакция) подробно описана С.Э. Шнолем в [6]. Если взять раствор, содержащий в определенных количествах и разведении лимонную кислоту, сульфат церия, бромат калия и серную кислоту, можно наблюдать чередующиеся изменения окраски раствора от бесцветной к желтой. Если вместо сульфата церия использовать железо-фенантролиновый комплекс, окраска раствора будет меняться с красного цвета на синий.

Колебательный цикл можно качественно описать следующим образом [7-8]. Первая стадия реакции представляет собой цепную реакцию окисления типа



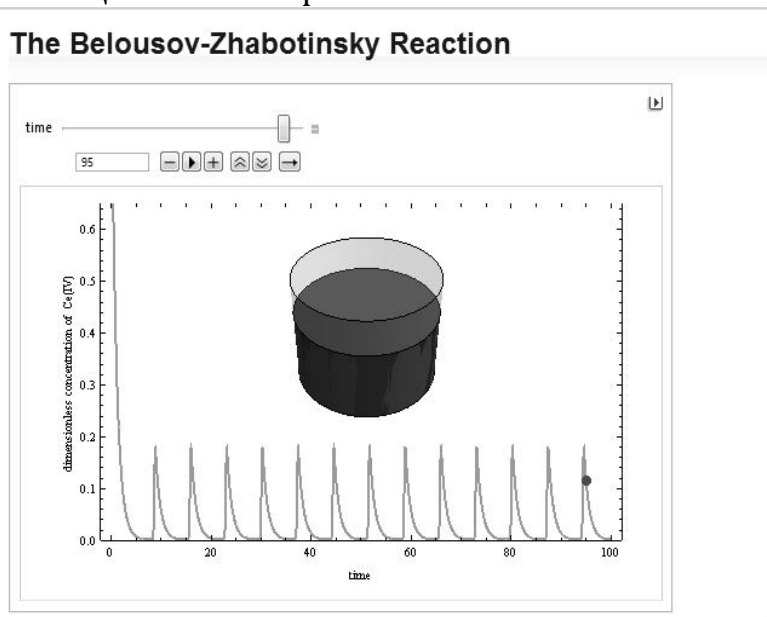
Кислота  $HBrO_2$  играет роль автокатализатора. Ингибирование цепной реакции происходит при участии иона  $Br^-$ . Когда концентрация  $M^{(n+1)+}$  в системе достаточно велика, скорость образования  $Br^-$  и его концентрация в растворе также высоки. В результате происходит торможение цепной реакции окисления  $Mn^{n+}$  и концентрация  $M^{(n+1)+}$  начинает падать, достигая некоторого порогового значения. После этого начинает резко расти концентрация  $HBrO_2$ , что приводит к ускорению цепной реакции и

концентрации  $M^{(n+1)+}$ . Затем процесс циклически повторяется. Колебания концентраций окисленной и восстановленной форм катализатора сопровождаются колебаниями окраски раствора от бесцветной к желтой, если катализатор — ионы церия, или от голубой к красной в случае фенантролинового комплекса железа.



**Рис. 1.** Периодические колебания концентрации в брюсселяторе

Пример математического моделирования процесса можно найти на сайте [9]. Там же можно найти описание модели и код программы. Если система Mathematica не установлена на компьютере, можно проводить моделирование с помощью CDF плеера.



**Рис. 2.** Моделирование реакции Белоусова-Жаботинского

## Литература

1. Николс Г., Пригожин И. Самоорганизация в неравновесных системах: От диссипативных структур к упорядоченности через флуктуации. Пер. с англ. М.: Мир, 1979. – 512 с.
2. Пригожин И., Пленсдорф Г. Термодинамическая теория структуры, устойчивости и флуктуаций. М.: УРСС, 2003. 280 с.
3. Физическая химия. Теоретическое и практическое руководство / Под ред. акад. Б. П. Никольского. Л.: Химия, 1987. 880 с.
4. Кунин С. Вычислительная физика. Пер. с англ. М.: Мир, 1992. 518 с.
5. Сайт корпорации Wolfram, Inc [Электронный ресурс]. – Режим доступа :<http://demonstrations.wolfram.com/HopfBifurcationInTheBrusselator/>, свободный. – Загл. с экрана.
6. Шноль С. Э. Герои, злодеи, конформисты отечественной науки. М.: Книжный дом «ЛИБРОКОМ», 2010. – 720 с. (Наука в СССР: Через тернии к звездам).
7. Жаботинский А. М. Концентрационные автоколебания. М.: Наука, 1974. 178 с.
8. Сайт о химии [Электронный ресурс]. – Режим доступа: <http://www.xumuk.ru/encyklopedia/488.html>, свободный. – Загл. с экрана.
9. Сайт корпорации Wolfram, Inc [Электронный ресурс]. – Режим доступа :<http://demonstrations.wolfram.com/TheBelousovZhabotinskyReaction/>, свободный. – Загл. с экрана.

## ***Компьютерное моделирование двупараметрических бифуркаций в задачах нелинейной динамики***

### ***1. Введение***

Важное место при изучении бифуркационных явлений занимает компьютерное моделирование. Как правило, чем сложнее бифуркация, тем большее значение принимает необходимость компьютерного моделирования. Более того, при изучении сложных бифуркационных явлений компьютерные вычисления часто выходят на первый план. Аналитические методы исследования задач о бифуркациях, как правило, сталкиваются с трудностями вычислительного характера при анализе конкретных моделей. Поэтому здесь актуальным направлением является разработка методов компьютерного моделирования для изучения сложных бифуркационных явлений.

Программы для исследования бифуркационных явлений создавались в разных местах и с разными целями. Одним из основоположников здесь является А.И.Хибник, который еще в 70-е годы прошлого столетия в Научно-исследовательском вычислительном центре АН СССР разработал комплекс программ на ФОРТРАНе для однопараметрического исследования бифуркаций периодических решений систем обыкновенных дифференциальных уравнений. Эти программы стимулировались, в первую очередь, интересами биологов [1]. Существенный вклад в разработку программ компьютерного моделирования бифуркаций внесли Хэссард Б., Казаринов Н., Вэн И. [2], и, в особенности, Ю.А.Кузнецов[3], принимавший непосредственное участие в разработке таких программ **CONTENT**, **MATCONT** и **AUTO**, широко используемых в теоретических и прикладных исследованиях. Эти программы позволяют проводить исследование различных нелинейных динамических систем, моделировать и визуализировать динамику, строить бассейны притяжения, вычислять ляпуновские показатели, проводить бифуркационный анализ, вычислять размерностные характеристики и т.д. В Интернете есть несколько сайтов, на которых поддерживается обширный список программного обеспечения для исследования нелинейных динамических систем (см., например, [www.dynamicalsystems.org](http://www.dynamicalsystems.org)).

Указанные программы носят в основном пользовательский характер. Они малоэффективны в учебном процессе подготовки специалистов в области математики и информационных технологий, когда требуется не только провести исследование какого-либо явления, но и получить соответствующие теоретические сведения, необходимые при разработке

соответствующих программ. В настоящей работе предлагаются алгоритмы исследования двухпараметрических бифуркационных задач. Такие задачи часто возникают в теории и приложениях [3, 4]. Предлагаемые алгоритмы реализованы в пакете MatLab. Выбор MatLab как основного инструмента определяется использованием большого числа векторов, матриц и их преобразований. В программный комплекс вошли как теоретические построения, необходимые в учебном процессе, так сами алгоритмы и программы, позволяющие строить решения бифуркационных задач в терминах исходной постановки. Для всех основных сценариев двухпараметрических бифуркаций реализованы программы численного расчета, позволяющие получить бифурцирующие решения и соответствующие значения параметров.

Ниже приводятся основные положения предлагаемого операторного метода исследования двухпараметрических бифуркационных задач, его приложения к задаче локализации языков Арнольда [3-5] нелинейных динамических систем и некоторые численные результаты.

## 2. Операторный метод.

Приведем в краткой форме основные положения операторного метода [6] исследования многопараметрических бифуркационных задач. Ограничимся рассмотрением двухпараметрических задач для операторных уравнений на плоскости.

Рассмотрим зависящее от двумерного параметра  $\mu = (\alpha, \beta) \in R^2$  операторное уравнение

$$x = B(\mu)x + b(x, \mu), \quad x \in R^2, \quad (1)$$

в котором квадратная матрица  $B(\mu)$  второго порядка непрерывно дифференцируемо зависит от  $\mu$ , а нелинейность  $b(x, \mu)$  также гладко зависит от  $\mu$  и представима в виде

$$b(x, \mu) = b_2(x, \mu) + b_3(x, \mu) + \tilde{b}_4(x, \mu),$$

где  $b_2(x, \mu)$  и  $b_3(x, \mu)$  содержат, соответственно, квадратичные и кубические по  $x$  слагаемые, а  $b_4(x, \mu)$  является гладкой по  $x$ , при этом  $\tilde{b}_4(x, \mu) = O(\|x\|^4), x \rightarrow 0$ , равномерно по  $\mu$ .

Уравнение (1) при всех значениях  $\mu$ , имеет нулевое решение  $x=0$ . Говорят [6], что значение  $\mu_0$  является точкой бифуркации ненулевых решений уравнения (1), если существует последовательность  $\mu_k \rightarrow \mu_0$  такая, что при уравнение (1) имеет ненулевое решение  $x = x_k$ , причем  $\|x_k\| \rightarrow 0$  при  $k \rightarrow \infty$ .

Как правило, бифуркации ненулевых решений уравнения (1) имеют направленный характер; приведем соответствующее определение. Пусть  $e \in R^2$  - некоторый ненулевой вектор. Значение  $\mu_0$  параметра  $\mu$  назовем правильной точкой бифуркации уравнения (1) по направлению вектора  $e$ , если существуют  $\varepsilon_0 > 0$  и определенные при  $\varepsilon \in [0, \varepsilon_0]$  непрерывные функции такие, что:

- $\mu(0) = \mu_0, x(0) = 0$ ;



- $\|x(\varepsilon) - \varepsilon e\| = o(\varepsilon)$  при  $\varepsilon \rightarrow 0$ ;
- для каждого  $\varepsilon \geq 0$  вектор  $x(\varepsilon)$  является решением уравнения (1) при  $\mu = \mu(\varepsilon)$ . Векторы  $x(\varepsilon)$  и значения  $\mu(\varepsilon)$  назовем бифурцирующими решениями уравнения (1).

**Лемма 1.** Пусть значение  $\mu_0$  параметра  $\mu$  является правильной точкой бифуркации уравнения (1) по направлению вектора  $e$ . Тогда вектор  $e$  будет собственным для матрицы  $B(\mu_0)$ , отвечающим собственному значению 1.

Ниже будем предполагать, что матрица  $B(\mu_0)$  имеет полупростое собственное значение 1 кратности 2; другими словами, пусть  $B(\mu_0) = I$ , где  $I$  - единичная матрица второго порядка. Обозначим  $\mu_0 = (\alpha_0, \beta_0)$  и  $B_0 = B(\mu_0)$ .

Пусть  $e, g$  и  $e^*, g^*$  - две пары линейно независимых векторов, выбранные исходя из соотношений:

$$(e, e^*) = (g, g^*) = 1, (e, g^*) = (g, e^*) = 0. \quad (2)$$

Положим

$$S = \begin{bmatrix} B'_\alpha(\alpha_0, \beta_0)e, e^* & B'_\beta(\alpha_0, \beta_0)e, e^* \\ B'_\alpha(\alpha_0, \beta_0)e, g^* & B'_\beta(\alpha_0, \beta_0)e, g^* \end{bmatrix} \quad (3)$$

Здесь  $B'_\alpha$  и  $B'_\beta$  - матрицы, полученные дифференцированием матрицы по  $\alpha$  и  $\beta$  соответственно.

**Теорема 1.** Пусть

$$\det S \neq 0. \quad (4)$$

Тогда  $\mu_0$  является правильной точкой бифуркации уравнения (1) по направлению вектора  $e$ .

Ниже используются обозначения

$$b_2 = b_2(e, \alpha_0, \beta_0), b_3 = b_3(e, \alpha_0, \beta_0), \quad (5)$$

$$b'_{2x} = b'_{2x}(e, \alpha_0, \beta_0), b'_{2\alpha} = b'_{2\alpha}(e, \alpha_0, \beta_0), b'_{2\beta} = b'_{2\beta}(e, \alpha_0, \beta_0). \quad (6)$$

Положим

$$Fh = [(h, e^*)B'_\alpha e + (h, g^*)B'_\beta e], \quad h \in R^2, \quad (7)$$

где обозначено  $B'_\alpha = B'_\alpha(\alpha_0, \beta_0)$  и  $B'_\beta = B'_\beta(\alpha_0, \beta_0)$ . В силу условия (4) линейный оператор  $F: R^2 \rightarrow R^2$  обратим. Положим

$$\Gamma_0 = F^{-1}: R^2 \rightarrow R^2. \quad (8)$$

**Лемма 2.** Оператор  $\Gamma_0 = F^{-1}$  вычисляется по формуле  $\Gamma_0 y = J_\alpha(y)e + J_\beta(y)g$ . Здесь функционалы  $J_\alpha(y)$  и  $J_\beta(y)$  - это компоненты вектора

$$J(y) = \begin{bmatrix} J_\alpha(y) \\ J_\beta(y) \end{bmatrix}$$

который вычисляется по формуле  $J(y) = -S^{-1}y(y)$ , где  $S$  - матрица

(3) и

$$\gamma(y) = \begin{bmatrix} (y, e^*) \\ (y, g^*) \end{bmatrix}.$$

Положим далее

$$e_1 = \Gamma_0 b_2, \quad \alpha_1 = J_\alpha(b_2), \quad \beta_1 = J_\beta(b_2), \quad (9)$$

$$e_2 = \Gamma_0(\varphi + b_3), \quad \alpha_2 = J_\alpha(\varphi + b_3), \quad \beta_2 = J_\beta(\varphi + b_3), \quad (10)$$

здесь

$$\begin{aligned} \varphi = & \alpha_1 B'_\alpha \Gamma_0 b_2 + \beta_1 B'_\beta \Gamma_0 b_2 + \frac{\alpha_1^2}{2} B''_{\alpha\alpha} e + \alpha_1 \beta_1 B''_{\alpha\beta} e + \frac{\beta_1^2}{2} B''_{\beta\beta} e + \\ & + b'_{2x} \Gamma_0 b_2 + \alpha_1 b'_{2\alpha} + \beta_1 b'_{2\beta} \end{aligned} \quad (11)$$

Здесь  $\Gamma_0$  - оператор (8),  $B'_\alpha, B'_\beta, B''_{\alpha\alpha}, B''_{\alpha\beta}, B''_{\beta\beta}$  - матрицы, полученные дифференцированием матрицы  $B(\alpha, \beta)$  по  $\alpha$  и (или)  $\beta$  нужное число раз в точке  $(\alpha_0, \beta_0)$ ; используются также обозначения (5) и (6).

**Теорема 2.** Существующие в условиях теоремы 1 бифурцирующие решения  $x(\varepsilon), \alpha(\varepsilon)$  и  $\beta(\varepsilon)$  уравнения (1) представимы в виде

$$x(\varepsilon) = \varepsilon e + \varepsilon^2 e_1 + \varepsilon^3 e_2 + o(\varepsilon^3), \quad (12)$$

$$\alpha(\varepsilon) = \alpha_0 + \varepsilon \alpha_1 + \varepsilon^2 \alpha_2 + o(\varepsilon^2), \quad \beta(\varepsilon) = \beta_0 + \varepsilon \beta_1 + \varepsilon^2 \beta_2 + o(\varepsilon^2). \quad (13)$$

### 3. Локализация языков Арнольда.

В качестве приложения рассматривается зависящая от двумерного параметра  $\mu(\alpha, \beta)$  динамическая система с дискретным временем

$$x_{n+1} = A(\alpha, \beta) x_n + a(x_n, \alpha, \beta), \quad n = 0, 1, 2, \dots, \quad x_n \in R^2, \quad (14)$$

в которой  $A(\alpha, \beta)$  - это матрица

$$A(\alpha, \beta) = (1 + \alpha) Q(\beta), \quad (15)$$

где

$$Q(\beta) = \begin{bmatrix} \cos 2\pi(\theta_0 + \beta) & -\sin 2\pi(\theta_0 + \beta) \\ \sin 2\pi(\theta_0 + \beta) & \cos 2\pi(\theta_0 + \beta) \end{bmatrix}. \quad (16)$$

Предполагается, что нелинейность  $\alpha(x, \alpha, \beta)$ , представима в виде:

$$\alpha(x, \alpha, \beta) = \alpha_2(x, \alpha, \beta) + \alpha_3(x, \alpha, \beta) + \tilde{\alpha}_4(x, \alpha, \beta), \quad (17)$$

где  $\alpha_2(x, \alpha, \beta)$  и  $\alpha_3(x, \alpha, \beta)$  содержат, соответственно, квадратичные и кубические по  $x$  слагаемые, а  $\tilde{\alpha}_4(x, \alpha, \beta)$  является гладкой по  $x$ , при этом  $\tilde{\alpha}_4(x, \alpha, \beta) = O(\|x\|^4), x \rightarrow 0$ , равномерно по  $\alpha$  и  $\beta$ .

Матрица имеет пару простых собственных значений

$$\lambda(\alpha, \beta) = (1 + \alpha) e^{\pm 2\pi(\theta_0 + \beta)i}. \quad (18)$$

При этом матрица  $A(0, 0)$  имеет пару простых собственных значений  $e^{\pm 2\pi(\theta_0 + \beta)i}$ , где  $0 < \theta_0 \leq \frac{1}{2}$  и рационально:  $\theta_0 = \frac{p}{q}$  - несократимая дробь.

В указанных предположениях положение равновесия  $x^* = 0$  системы (14) при  $\mu_0 = (0, 0)$  является негиперболическим, а значение  $\mu = \mu_0$  является

бифуркационным.

Пусть  $P$  - это плоскость параметров  $\mu=(\alpha, \beta)$  системы (14). Сценарии бифуркаций в окрестности точки  $x^*=0$  равновесия системы (14) определяются характером перехода параметра  $\mu \in P$  через точку  $\mu_0$ . Здесь могут возникать или исчезать периодические решения различных периодов.

Одним из основных сценариев (но не единственным) здесь является бифуркация  $q$ -циклов системы (14), когда при значениях параметров  $\mu$  близких к  $\mu_0$  у системы (14) возникают циклы периода  $q$ , при этом амплитуды циклов стремятся к нулю при стремлении точки  $\mu$  к  $\mu_0$ .

На плоскости  $P$  образуется характерная структура областей режимов нелинейной системы (14), которая представляет собой области синхронизации с разным соотношением параметров  $\alpha$  и  $\beta$ . Эти области имеют клювообразную форму или языка  $\Psi(\alpha^*, \beta^*)$  вершины которых лежат в тех точках кривой  $(\alpha^*, \beta^*)$ , в которых матрица  $A(\alpha^*, \beta^*)$  имеет собственные значения  $e^{\pm 2\pi(\theta_0 + \beta)i}$  с рациональным  $\theta^* : \theta^* = \frac{1}{m}$ .

Такие языки соответствуют областям значений параметров  $(\alpha, \beta)$ , при которых система (14) имеет периодические режимы периода  $m$ , амплитуды которых стремятся к нулю при стремлении точки  $(\alpha, \beta)$ , к  $(\alpha^*, \beta^*)$ . Другими словами, множество  $\Psi(\alpha^*, \beta^*)$  содержит те последовательности  $(\alpha_k, \beta_k) \rightarrow (\alpha^*, \beta^*)$ , при которых реализуется сценарий бифуркации  $m$ -циклов системы (14).

Таким образом, указанные языки  $\Psi(\alpha^*, \beta^*)$  соответствуют рационально синхронизированным (в естественном смысле) соотношениям параметров и . Между указанными языками существуют области квазипериодических режимов с иррациональным соотношением параметров. Основные черты этой картины были выявлены российским математиком В.И.Арнольдом [5], так что система языков синхронизации, соответствующих рационально синхронизированным соотношениям параметров, получила название языков Арнольда.

Указанная структура областей режимов имеет локальный характер. При удалении параметров  $\alpha$  и  $\beta$  от точки  $(\alpha^*, \beta^*)$  области периодических режимов вытесняют квазипериодические, и языки начинают перекрываться. Становится возможным хаос. Систему языков Арнольда можно наблюдать в возбуждаемых периодическим сигналом автоколебательных системах, в задачах о взаимной синхронизации двух автоколебательных систем и др. (см., например, [7]).

При  $m \geq 5$  языки Арнольда соприкасаются в точке  $e^{2\pi\theta i}$ ; в этом случае язык Арнольда  $\Psi(l, m)$  в малой окрестности точки  $e^{2\pi\theta i}$  фактически вырождается в кривую. При  $m \leq 4$  язык Арнольда  $\Psi(l, m)$  представляет собой существенно более широкое множество. Такое устройство языков Арнольда обуславливается структурой так называемых резонансных членов в тейлоровском разложении отображения  $F(x, \mu)$  в нуле. За

существование циклов малых периодов  $m \leq 4$  отвечают главные резонансные члены. Соответственно, циклы малых периодов у системы (14) наблюдаются достаточно часто, а длиннопериодические циклы (при  $m \geq 5$ ) являются нетипичными и наблюдаются редко.

Положим

$$e(t) = \begin{bmatrix} \cos t \\ \sin t \end{bmatrix}, \quad g(t) = \begin{bmatrix} -\sin t \\ \cos t \end{bmatrix}, \quad (19)$$

и

$$b_2(t) = Q^{q-1}a_2(e(t), \mu_0) + Q^{q-2}a_2(Qe(t), \mu_0) + \dots \\ + Qa_2(Q^{q-2}e(t), \mu_0) + a_2(Q^{q-1}e(t), \mu_0), \quad (20)$$

где  $Q=Q(0)$ , а  $Q(\beta)$  - матрица (16). Аналогично определяется вектор-функция  $b_3(t)$ . Далее, определим функции

$$\alpha_1(t) = -\frac{1}{q}(b_2(t), e(t)), \quad \beta_1(t) = -\frac{1}{2\pi q}(b_2(t), g(t)) \quad (21)$$

$$\chi(t) = q \left[ \frac{1}{2} \alpha_1^2(t)(1+q) - 2\pi \beta_1^2(t)(1+\pi q) \right] e(t) + \alpha_1(t)\beta_1(t)q(1+2\pi+2\pi q)g(t) + \\ + b'_{2x}(t)(\alpha_1(t)e(t) + \beta_1(t)g(t)) + \alpha_1(t)b'_{2\alpha}(t) + \beta_1(t)b'_{2\beta}(t), \quad (22)$$

$$\alpha_2(t) = -\frac{1}{q}(\chi(t) + b_3(t), e(t)), \quad \beta_2(t) = -\frac{1}{2\pi q}(\chi(t) + b_3(t), g(t)). \quad (23)$$

Для каждого  $t \in [0, 2\pi]$  определим кривую  $\Gamma(p, q, e(t))$ , определяемую равенством

$$z = (1 + \alpha(\varepsilon, t))e^{2\pi(\theta_0 + \beta(\varepsilon, t))i}, \quad 0 \leq \varepsilon \leq 1. \quad (24)$$

(точка каждой из кривых  $\gamma(p, q, e(t))$  при совпадает с точкой  $e^{2\pi\theta_0 i}$ ).

Языком Арнольда  $\Psi(p, q)$  системы (14) будем называть множество

$$\Psi(p, q) = \bigcup_{t \in [0, 2\pi]} \gamma(p, q, e(t)). \quad (25)$$

Приведем теперь основные утверждения работы, позволяющие локализовать определенные равенством (25) языки Арнольда  $\Psi(p, q)$  системы (14). Здесь принципиально различными являются случаи  $q \geq 5$  и  $q \leq 4$ . Первый из этих случаев называют слабрезонансным, а второй - сильнорезонансным. Рассмотрим сначала слабрезонансный случай.

**Теорема 3.** Пусть  $q \geq 5$ . Тогда язык Арнольда  $\Psi(p, q)$  системы (14) определяется равенством (25), в котором  $\gamma(p, q, e(t))$  - это (при фиксированном  $t$ ) кривая, описываемая уравнением (24); здесь

$$\alpha(\varepsilon, t) = \alpha_2\varepsilon^2 + \varepsilon^3\alpha_3(\varepsilon, t), \quad \beta(\varepsilon, t) = \beta_2\varepsilon^2 + \varepsilon^3\beta_3(\varepsilon, t), \quad (26)$$

$\alpha_2$  и  $\beta_2$  - числа (23) (т.е. указанные функции принимают постоянные значения), а функции  $\alpha_3(\varepsilon, t)$  и  $\beta_3(\varepsilon, t)$ , непрерывны и являются  $2\pi$ -периодическими по  $t$ .

Из равенств (24) и (26) следует, что для  $q \geq 5$  при малых  $\varepsilon \geq 0$  языки Арнольда  $\Psi(p, q)$  системы (14) чрезвычайно узкие. А именно, если числа

$\alpha_2$  и  $\beta_2$  являются ненулевыми, то множество  $\Psi(p, q)$  локально можно отождествить с кривой  $\Psi(p, q)$ , описываемой уравнением

$$z = (1 + \alpha_2 \xi) e^{2\pi(\theta_0 + \beta_2 \xi)i}, 0 \leq \xi \leq 1, \quad (27)$$

начинающейся (при  $\xi=0$ ) из точки  $e^{\varphi_0 i}$  на единичной окружности  $S \in C$ ; здесь  $\varphi_0 = 2\pi p/q$ .

Рассмотрим теперь сильнорезонансный случай, т.е. пусть  $2 \leq q \leq 4$ . В этом случае языки Арнольда  $\Psi(p, q)$  системы (14) в естественном смысле существенно шире, чем при  $q \geq 5$ . Например, верна

**Теорема 4.** Пусть  $q=4$ . Тогда язык Арнольда  $\Psi(p, q)$  системы (14) определяется равенством (23) (где  $p=1$  и  $q=4$ ), в котором  $\gamma(p, q, e(t))$  - это (при фиксированном  $t$ ) кривая, описываемая уравнением (24) при

$$\alpha(\varepsilon, t) = \alpha_2(t)\varepsilon^2 + \varepsilon^3 \alpha_3(\varepsilon, t), \quad \beta(\varepsilon, t) = \beta_2(t)\varepsilon^2 + \varepsilon^3 \beta_3(\varepsilon, t), \quad (28)$$

здесь  $\alpha_2(t)$  и  $\beta_2(t)$  - функции (23) (при  $q=4$ ), а функции  $\alpha_3(\varepsilon, t)$  и  $\beta_3(\varepsilon, t)$ , непрерывны и являются  $2\pi$ -периодическими по  $t$ .

Таким образом, для  $q=4$  языки Арнольда  $\Psi(p, q)$  системы (14) локально можно отождествить с совокупностью (по  $t \in [0, 2\pi]$ ) кривых, описываемых уравнениями

$$z = (1 + \alpha_2(t)\xi) e^{2\pi(\theta_0 + \beta_2(t)\xi)i}, 0 \leq \xi \leq 1. \quad (29)$$

#### 4. Пример.

В качестве примера рассмотрим дискретную систему

$$x_{n+1} = A(\alpha, \beta)x_n + \alpha_3(x_n), n = 0, 1, 2, \dots, \quad (30)$$

в которой  $A(\alpha, \beta) = (1 + \alpha)Q(\beta)$ , где

$$Q(\beta) = \begin{bmatrix} \cos 2\pi(0,25 + \beta) & -\sin 2\pi(0,25 + \beta) \\ \sin 2\pi(0,25 + \beta) & \cos 2\pi(0,25 + \beta) \end{bmatrix},$$

а нелинейность  $\alpha_3(x)$  имеет вид  $\alpha_3(x) = \begin{bmatrix} x_1^3 + 2x_2^3 \\ 2x_1x_2^2 \end{bmatrix}$ . Так как  $Q(0) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ , то в

этом примере имеем  $\theta_0 = 1/4$ . Для локализации языка Арнольда  $\Psi(1, 4)$  системы (30) воспользуемся теоремой 4. Из этой теоремы следует, что множество  $\Psi(1, 4)$  локально можно отождествить с совокупностью кривых (29). Вычислим функции  $\alpha_2(t)$  и  $\beta_2(t)$ . Имеем:  $b_2(t) \equiv 0$  и

$$b_3(t) = Q^3 \alpha_3(e(t)) + Q^2 \alpha_3(Qe(t)) + Q \alpha_3(Q^2 e(t)) + \alpha_3(Q^3 e(t)),$$

где  $Q = Q(0)$ . Несложные вычисления приводят к равенству

$$b_3(t) = 2 \begin{bmatrix} 2 \sin^2 t \cos t + \sin^3 t - 2 \cos^3 t \\ 2 \sin t \cos^2 t - 2 \sin^3 t - \cos^3 t \end{bmatrix}$$

Тогда

$$\alpha_2(t) = -\frac{1}{4}(b_3(t), e(t)) = \frac{1}{4} \cos 2t (4 \cos 2t + \sin 2t),$$

$$\beta_2(t) = -\frac{1}{8\pi}(b_3(t), g(t)) = \frac{1}{8\pi} (1 + \cos^2 2t - 4 \sin 4t).$$

Подставляя эти формулы в (29) и проведя анализ полученного



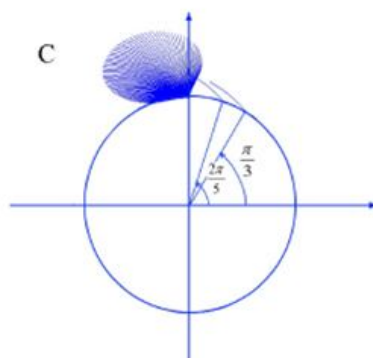
равенства получим, что локально язык Арнольда  $\Psi(1,4)$  системы (30) заключен между двумя кривыми  $\gamma_1$  и  $\gamma_2$ , которые описываются, соответственно, уравнениями

$$z = (1 + \alpha_1 \xi) e^{2\pi(0,25 + \beta_1 \xi)i}, \quad z = (1 + \alpha_2 \xi) e^{2\pi(0,25 + \beta_2 \xi)i} \quad (0 \leq \xi \leq 1);$$

здесь

$$\alpha_1 = \frac{1}{2}, \beta_1 = \frac{3 - \sqrt{17}}{16\pi}, \quad \alpha_2 = \frac{4 - \sqrt{17}}{8}, \quad \beta_2 = \frac{3}{16\pi}$$

Полученный результат подтверждается и прямым численным вычислением языка Арнольда  $\Psi(1,4)$  системы (30) в соответствии с формулами теоремы 4 (см. Рис. 5).



**Рис. 5.** Языки Арнольда системы (30)

На этом рисунке изображены языки Арнольда системы (30), отвечающие одному сильному ( $q=4$ ) резонансу (на рисунке этому языку отвечает широкое множество) и двум слабым ( $q=5$  и  $q=6$ ) резонансам (на рисунке этим языкам отвечают фактически две кривые).

### Литература

1. Базыкин А.Д., Кузнецов Ю.А., Хибник А.И. Портреты: бифуркаций (Бифуркационные диаграммы динамических систем на плоскости) // Серия "Новое в жизни, науке, технике. Математика, кибернетика". М.: Знание, 1989.
2. Хэссард Б., Казаринов Н., Вэн И. Теория и приложения бифуркации рождения цикла. М.: Мир, 1985.
3. Kuznetsov Yu. A. Elements of Applied Bifurcation Theory. // Applied Mathematical Sciences (V. 112), Springer-Verlag, New-York etc., 1995.
4. Гукенхеймер Дж., Холмс Ф. Нелинейные колебания, динамические системы и бифуркации векторных полей. - Москва-Ижевск: Ин-т компьютер. исслед., 2002.
5. Арнольд В.И. Геометрические методы в теории обыкновенных дифференциальных уравнений. - Ижевск: Регулярная и хаотическая динамика, 2000.
6. Юмагулов М.Г. Операторный метод исследования правильной бифуркации в много-параметрических системах // Доклады АН. 2009. Т. 424, № 2. С. 177-180.
7. Магницкий Н.А., Сидоров С.В. Новые методы хаотической динамики. - М.: Едиториал УРСС, 2004.

## **СЕКЦИЯ 7. ШКОЛЬНОЕ ОБРАЗОВАНИЕ ПО ИНФОРМАТИКЕ**

**Волкова Е.П.,**

Липецкий государственный педагогический университет, старший  
преподаватель  
[volkova.lenochka@mail.ru](mailto:volkova.lenochka@mail.ru)

**Павлов Е.Н.,**

Департамент образования администрации города Липецка  
[umz@inbox.ru](mailto:umz@inbox.ru)

**Шуйкова И.А.**

Липецкий государственный педагогический университет, к.т.н.,  
доцент  
[shujkova\\_i\\_a@inbox.ru](mailto:shujkova_i_a@inbox.ru)

### ***Дополнительное образование школьников, проявляющих способности в изучении информатики, в системе образования города Липецка***

#### **Аннотация**

*В работе изложены подходы к непрерывному образованию школьников, проявляющих способности к изучению отдельных дисциплин, в рамках деятельности созданного в городе Липецке Центра «Стратегия» как Центра дополнительного образования детей.*

В последние годы усилилось внимание общества, Правительства РФ, Президента лично к проблеме работы с талантливыми детьми и молодежью. Это одно из значимых направлений инициативы президента «Наша новая школа», утвержденной в 2010 году, это и целая Концепция общенациональной системы выявления и развития молодых талантов, утвержденная в апреле 2012 года, это Указ Президента о мерах по реализации государственной политики в области образования и науки от 7 мая 2012 года. Суть этих документов в отношении работы с одаренными обучающимися в том, что обозначается миссия государства – обеспечение по возможности равных условий для выявления и поддержки одаренных детей и молодежи, независимо от сферы одаренности, социального положения и имущественных возможностей семей.

В России к настоящему времени сложились отдельные образовательные территории, где успешно и с прицелом на будущее работают с одаренными детьми и подростками, готовя их к включению в дальнейшую творческую профессиональную жизнь [1]. В разных субъектах Российской Федерации эта работа ведется по-разному: создаются интернаты для детей, проявляющих способности; существуют специализированные школы; разрабатываются и внедряются в



образовательную практику дистанционные курсы для одаренных детей. В Липецке с целью создания муниципальной системы выявления и сопровождения одаренных детей в 2010 году был создан Центр дополнительного образования детей «Стратегия», образовательная деятельность которого началась в январе 2011 года. Идея создания Центра базировалась на многолетнем опыте департамента образования администрации города Липецка по работе с одаренными детьми и явилась консолидацией усилий общеобразовательных учреждений города по данному направлению работы[2]. Образовательная деятельность Центра осуществляется за счет субсидий городского бюджета, что обеспечивает бесплатное дополнительное образование детей, проявляющих способности к изучению отдельных дисциплин. Конкурсный прием школьников в Центр является гарантом высокого уровня подготовки и сложившийся мотивации к интеллектуальному труду обучающихся «Стратегии»

Центр оказывает следующие муниципальные услуги: предоставление дополнительного образования по дополнительным образовательным программами предоставление дополнительного образования по подготовке обучающихся к заключительному этапу Всероссийской олимпиады школьников. Всего в Центре реализуется 19 предметных направлений подготовки для школьников 7-11 классов: математика, физика, информатика, русский язык, литература и т.д. Организована работа и научно-исследовательских групп по математике и физике. В Центре проходят обучение ежегодно более 580 человек в малокомплектных группах не более десяти человек по каждому предмету и четырем возрастным направлениям. Таким образом, можно говорить о непрерывном образовании школьников, посещающих занятия Центра «Стратегия», сочетающем в себе общеобразовательный курс предметной подготовки (школьное обучение) и курс обучения по дополнительной образовательной программе по данному предмету (обучение в Центре). Обучение ведется по авторским образовательным программам высококвалифицированными вузовскими преподавателями, имеющими опыт подготовки школьников к олимпиадам всероссийской олимпиады школьников. Одна из стратегий психолого-педагогического сопровождения одаренных подростков в Центре - коучинг как это движение от цели к решению проблем. Именно такая стратегия должна быть ведущей в работе с одаренными детьми, для которых проблемы выстраивания жизни, проблемы смысла и целей самоактуализации являются самыми значимыми и самыми трудными [3].

Остановимся подробнее на дополнительном образовании школьников, проявляющих способности к изучению информатике. Подготовка к олимпиаде по информатике, как, впрочем, и подготовка к любой другой олимпиаде, требует отбора детей с определенным уровнем мотивации. В январе каждого года проводятся вступительные испытания в Центр «Стратегия» для школьников 7-10 классов, по результатам которых происходит зачисление ребят. Заметим, что в 7 классе многие школьники

не владеют навыками программирования им предлагается теоретическая работа по составлению алгоритма для решения предложенной задачи. В этом случае важно развитое логическое мышление и умение предлагать идеи решения задачи в виде алгоритмов на естественном языке. В старших классах в качестве вступительных испытаний предлагается набор олимпиадных задач уровня муниципального этапа всероссийской олимпиады школьников.

Для успешного выступления в олимпиадах необходима целенаправленная подготовка, которая должна обеспечить отличное владение практическими и теоретическими знаниями. Каждый годовой курс обучения в Центре «Стратегия» (с января по декабрь текущего календарного года) рассчитан на подготовку в объеме 144 академических часов (4 академических часа в неделю). Курс обучения по информатике в 7 классе имеет своей целью овладение навыками программирования, изучение основных алгоритмов, развитие логического мышления и привитие интереса к решению олимпиадных задач по программированию. В качестве развития у школьников навыков решения олимпиадных задач, обучающимся предлагается решать задачи с сайтов, содержащих встроенные проверяющие системы. Работа в системе позволяет адаптировать школьников к олимпиадам по программированию, так как практика показывает, что при тестировании предлагаемых задач оказываются не учтенными их тонкости, из-за которых программа может работать не на всех тестах, в результате чего решение не засчитывается. Привитие интереса к работе с такого рода системами, развитие соревновательного и конкурентного духа между школьниками в одной группе можно рассматривать как важнейшую мотивационную составляющую в подготовке к решению задач по программированию. Преподавателями для обучающихся Центра «Стратегия» в начале года разрабатываются образовательные траектории их участия в олимпиадах школьников. Среди которых можно выделить турнир Архимеда по программированию (г.Москва), Дистанционная олимпиада школьников по информатике (Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики), Московская олимпиада по информатике (Департамент образования г.Москвы, МГУ им. М.В.Ломоносова, МИОО, ВШЭ, МЦНМО) и др. Залогом дальнейшего успеха школьников является формирование у них навыков самостоятельной работы, участие в индивидуальных соревнованиях по программированию. Только при условии высокой мотивации к участию в соревнованиях у семиклассников (восьмиклассников) можно добиться в дальнейшем устойчивых результатов обучения методам программирования в 9-11 классах.

Особую роль в работе с одаренными детьми играют летние и зимние профильные загородные смены, давно апробированные в нашей стране. В Липецке второй год подряд проводятся летние информационно-

математические смены, участники которых – обучающиеся и преподаватели Центра «Стратегия». В рамках летних смен ребята проходят интенсивный курс обучения по программированию, участвуют в практикумах-тренировках решения олимпиадных задач, занимаются с психологами с целью развития навыков саморегуляции, развития соревновательного духа, индивидуальной «настройке» на результат. В качестве преподавателей в летние школы приглашаются не только преподаватели вузов, работающие с ребятами в течение года, но и студенты-старшекурсники Липецкого государственного педагогического и Липецкого государственного технического университетов – участники командных и индивидуальных соревнований по программированию. Школьникам очень важно общение с молодыми, близкими себе по духу студентами, которые примером своих собственных результатов могут служить подросткам образцами для подражания. Такие смены позволяют сплотить коллектив, сформировать команды для участия в соревнованиях, выработать устойчивое желание двигаться вперед. Отметим, что финансирование участия школьников в такого рода сменах осуществляется за счет средств городского бюджета. То есть, участие в смене для ребят, показавших наилучшие результаты обучения на занятия в учебном году, является полностью бесплатным. Наряду с летними профильными сменами с 2013 года будут организовываться зимние профильные смены для способных ребят.

По инициативе управления образования и науки липецкой области в течение последних пяти лет в области проводится очно-заочная школа «Одаренный ребенок» осенью-зимой каждого года [4]. Целью школы является дистанционная подготовка обучающихся 7-10 классов, не имеющих возможности заниматься на очных занятиях (например, живущих в удаленной сельской местности), к муниципальному и региональному этапу всероссийской олимпиады школьников. Занятия проводятся при помощи системы ин-класс, в которой по заранее определенному расписанию размещаются теоретические конспекты занятий и задания для домашнего выполнения. Сданные по графику выполненные домашние задания проверяются преподавателями, на основе чего формируется динамический рейтинг каждого участника ОЗШ «Одаренный ребенок». Предусмотрены и видеоконференции, в ходе проведения которых преподаватели разбирают наиболее сложные задания и отвечают на вопросы. По окончании работы очно-заочной школы проходит, так называемый, он-лайн день, на котором ребята и преподаватели встречаются уже лично на завершающих очных занятиях. В 2012 году контингент школьников липецкой области и города Липецка, участвующих в очно-заочной школе составил более 300 человек. По материалам школы выпускаются печатные пособия, которые впоследствии смогут получить не только новые участники очно-заочной школы, но и школьные учителя.

Для наиболее заинтересованных подростков, интересующихся

современной наукой, Центр «Стратегия» организовал работу научно-популярного лектория Центра, работа которого проходит ежемесячно в виде публичных лекций известных преподавателей вузов города. На лекциях по тематике физико-математических и компьютерных дисциплин выступили доценты факультета физико-математических и компьютерных дисциплин с темами: «Компьютер и человек – полвека вместе. IT-эволюция и направления дальнейшего развития», «Криптографические алгоритмы – взгляд изнутри. Математические основы криптографии», «Системы искусственного интеллекта. Математические основы и реализация», «Инварианты – одинаковые и разные». Приглашенными лекторами стали представители НИЯУ МИФИ, которые представили участникам лектория доклады на темы: «Невидимая вселенная», «Эксперимент «Памела», «Создание системы распределенных компьютерных вычислений GRID».

Все вышеперечисленные события формируют необходимую для развития способностей социальную среду.

Социальная среда в ее широком и узком смысле влияет на развитие ребенка так же, как и одаренные дети оказывают свое влияние на среду, составляя ее духовно-нравственный потенциал. И от того, насколько грамотно будут выстроены стратегии работы с одаренными детьми, зависит не только будущий творческий продукт - суть ожиданий от одаренных детей, но и устойчивость нашего общества в целом [3].

Системность работы Центра может быть обеспечена только всеми вышеописанными направлениями дальнейшего развития. И первые шаги на этом пути уже есть.

### **Литература**

1. Рубцов В. В. Работа с одаренными детьми требует ювелирной точности. //Учительская газета». 1 марта 2011.
2. Шуйкова И.А., Павлов Е.Н. Выявление и сопровождение одаренных детей в системе дополнительного образования города Липецка //Материалы Всероссийской научно-практической конференция «Одаренные дети и молодежь: диагностика и мониторинг развития одаренности», МГППУ, март 2012.
3. Юркевич В.С. Одаренные дети: сегодняшние тенденции и завтрашние вызовы. //Психологическая наука и образование №4/2011.
4. Сайт очно-заочной школы «Одаренный ребенок». <http://school.strategy48.ru/>

**Гладких И.Ю.**

Тульский государственный педагогический университет  
им. Л.Н. Толстого, студент  
[g-iu@yandex.ru](mailto:g-iu@yandex.ru)

**Якушин А.В.**

Тульский государственный педагогический университет  
им. Л.Н. Толстого, заведующий кафедрой информатики и МОИ  
[yakushin@tspu.tula.ru](mailto:yakushin@tspu.tula.ru)

### ***Изучение робототехники в средней школе***

Прогресс общества в XXI веке во многом определяется развитием космических технологий, нанотехнологий и робототехники. [1]

В эпоху построения постиндустриального общества всеобщая информационная грамотность населения и, в первую очередь, подрастающего поколения, играют определяющую роль для общественного прогресса.

Современные технологии развиваются быстрыми темпами, предоставляя все новые и новые возможности для удовлетворения повседневных информационных потребностей людей. К сожалению, наблюдается все больший и больший разрыв между уровнем информатизации социума и уровнем грамотности людей, необходимым для использования предоставляемых современными технологиями возможностями.

Современные информационные и компьютерные технологии, системы автоматизированного управления требуют специалистов соответствующей квалификации. Для подготовки таких кадров необходимо начинать изучение робототехники уже в средней школе. При изучении архитектуры компьютера или программирования важно осознать взаимосвязь между программным обеспечением и функционированием устройств. Работа с роботом является одним из лучших способов демонстрации и изучения принципов взаимодействия аппаратной и программных частей компьютерной техники. В отличие от математических моделей (в том числе обычной программы), все изменения в настройках робота ощутимы. Каждое изменение как настроек, так и результатов можно точно определить, что не всегда возможно в физических моделях.

Робототехнические технологии позволяют реализовывать принципы дифференцированного и индивидуального подхода к обучению. На занятии преподаватель дает возможность каждому обучаемому самостоятельно работать с роботизированной платформой, что позволяет ему детально разобрать новый материал по своей схеме.

Начинать занятия робототехникой в школах можно с любого класса,

так к этому моменту ребенок уже владеет базовыми знаниями об окружающем его мире, обладает живым воображением и возможностями для его описания.

Робототехника включает в себя как электрические, так и механические технологии. В промышленности роботы помогают автоматизировать технологические процессы и увеличить производительности труда. Изучение конструирования роботов в средней школе дает широкий простор для реализации межпредметных связей на стыке математики, физики, технологии, биологии и других областей знания.

Одной из многофункциональных систем является набор Lego Mindstorms. Это платформа пользуется популярностью во всем мире. Достоинствами данной платформы являются простота сборки и легкость программирования и управления.

Использование Lego-конструкторов во внеурочной и урочной деятельности повышает мотивацию учащихся к обучению, т.к. при этом требуются знания практически из всех учебных дисциплин от искусств и истории до математики и естественных наук. Межпредметные занятия опираются на естественный интерес к разработке и постройке различных деталей. Разнообразие конструкторов Lego позволяет заниматься с учащимися по разным направлениям. Lego-конструирование – это современное средство обучения детей. Занятия помогают решить проблему занятости детей, а также способствуют многостороннему развитию личности ребенка и побуждают получать знания дальше.

Сейчас существует множество различных пособий и книг по робототехнике, но большинство из них требует специальных знаний, так же слабо представлены доступные учебные курсы ориентированные на школьный уровень.

На кафедре информатики и МОИ Тульского государственного педагогического университета им. Л.Н. Толстого, в рамках педагогического сотрудничества с лицеем №4 г. Тулы, проводится разработка методических материалов для проведения занятий со школьниками по основам робототехники.

Цель курса «Робототехника для школьников» состоит в создании комплексного методического обеспечения для проведения занятий в робототехническом кружке.

Для реализации цели необходимо разработать теоретические материалы по основам робототехники; материалы для практических занятий; создать набор проектных заданий для организации самостоятельной работы школьников.

Планируемые результаты обучения.

У обучающихся должно сложиться представление: об эволюции и современном состоянии робототехники; о существующих роботизированных платформах и способах их применения; о процессе

технического конструирования и его особенностях; о способах работы с платформой Lego Mindstorms; о путях повышения своей компетентности через овладение навыками создания и управления роботами.

Участие в занятиях должно помочь учащимся: понять роль и место роботизированных и информационных технологий в формировании общей культуры современного человека; приобрести начальную подготовку по использованию различных технических узлов, датчиков в роботизированных устройствах; повысить свою информационную и коммуникативную компетентность.

Учащиеся будут знать: основные понятия робототехники, особенности конструирования роботов для решения прикладных задач; характеристики и основные параметры устройств, используемых в робототехнике; основные принципы программного управления роботом; принципы и способы автоматического управления; принципы работы программы Lego MindStorms Education NXT.

Будут уметь: использовать знания в области робототехники и информационных технологий для создания простых устройств; мотивированно выбирать различные технические узлы для реализации взаимодействия робота с внешним миром; использовать программу Lego MindStorms Education NXT для реализации программного управления; модифицировать и корректировать созданные роботизированные устройства

Приобретут навыки: в работе с роботизированными платформами; в использовании меню, панели инструментов программы Lego MindStorms Education NXT; в самостоятельной работе с электронным учебным пособием и средой электронного обучения; в работе в группе над общим проектом.

Преподавание курса включает инновационные формы работы с учащимися: работа в среде электронного обучения, использование реальных аппаратных и программных средств в учебной деятельности, применения современных коммуникационных средств для общения учащихся как между собой так и с преподавателем, также используются традиционные формы занятий: лекционные, практические (лабораторные) занятия и самостоятельная работа. Самостоятельная работа предназначена для выполнения индивидуального задания, например, в рамках группового проекта. Упор в освоении курса сделан на практические занятия.

Курс «Робототехника для школьников» разбит на два модуля: начальный и базовый. Начальный модуль включает в себя как теоретические аспекты робототехники, так и практические вопросы. Основной упор здесь делается на конструкторские и аппаратно-технические решения. Рассматриваются вопросы, связанные с историей робототехники, устройством роботов, приводится математическое описание роботов и узлов робота, изучаются основы программного управления. Практическая часть модуля посвящена платформе Lego MindStorms и созданию простых моделей роботов. Содержание

практической части начального модуля:

- Платформа Lego MindStorms: общие принципы работы с конструктором Lego MindStorms; конструкция, органы управления и дисплей; создание простейшей модели; элементы управления и датчики; приводы.

- Базовые принципы программного управления: интерфейс программы Lego MindStorms Education NXT; основы программирования; программные блоки; воспроизведение звуков; различные варианты движения; сложные виды движения ( поворот, движение по кривой, движение по ломаной).

Базовый курс ориентирован на программирование: учитывая знания, полученные ранее, ученик создает программу решающую поставленную задачу и максимально эффективную для собранной конфигурации робота. Кроме того в сферу решаемых задач по робототехнике теперь входят алгоритмические аспекты, проблемы эффективности решений, функциональности конструкций, а так же ряд вопросов более высокого уровня, такие как: взаимодействие робота и «станции» управления, взаимодействие роботов, управление без контроллеров, ИИ (искусственный интеллект) и ВИ (виртуальный интеллект), и т.п. Полученные теоретические знания о функционировании роботов, из начального курса, существенно дополняются и расширяются сведениями из смежных дисциплин: физики, химии, математики и многих других. Рассматриваются теоретическое описание аппаратной составляющей робота от макро- до микро- уровней.

Содержание практической части базового модуля:

- создание простых монофункциональных роботов: робот, движущийся по заданной траектории; робот, находящий выход из простого лабиринта; робот, обходящий препятствия; робот-футболист;

- робот, следующий по линии: базовый алгоритм следования по линии; следование по линии, не имеющей самопересечений; адаптивный алгоритм следования по линии на основе конечного автомата.

Практическим результатом изучения курса «Робототехника для школьников» является выполнение индивидуального или группового проектного задания, реализация которого осуществляется в несколько этапов.

Проблемно-целевой этап.

Прежде чем, начнется работа над проектом, разработчики должны ответить на ряд вопросов:

- Для чего создается данный проект? чем вызвана необходимость его создания? существует ли на самом деле потребность в этом проекте? как в дальнейшем будет использоваться данный проект? кто выступит в роли той целевой группы, для которой создается данный проект? найдет ли он своих потребителей?

- Каким должен быть проект для того, чтобы отвечать полностью



поставленным задачам?

- Кто будет создавать проект? в какой мере сможет он (смогут они) воплотить творческий замысел руководителя, реализовать задуманное? какие из необходимых им для реализации проекта знаний, умений и навыков учащиеся имеют сейчас, будут иметь к моменту исполнения требуемого вида работы?

- Как лучше распределить обязанности среди членов бригады, если исполнителей несколько?

То есть, на первом этапе осуществляются выбор проблемной области, постановка задач, определяется конечный вид создаваемого продукта, его назначение и круг пользователей, происходит формирование состава проектной бригады и распределение обязанностей. Этот этап завершается формулировкой темы проекта и определением вида его завершенной формы, написанием краткой аннотации проекта.

Этап разработки сценария и технического задания.

На данном этапе предполагается отбор содержания и определяется примерный объем проекта, производится его предельная детализация, прописываются роли всех участников проекта, сроки исполнения ими каждого вида работы.

Этап практической работы.

На этом этапе ведется работа по воплощению в жизнь поставленных задач, которая требует от всех участников предельной исполнительности, слаженности в действиях, а также значительных усилий от руководителя проекта по координации деятельности участников проекта и постоянного контроля за ходом и сроками производимых работ.

Этап предварительной защиты

На данном этапе осуществляется предварительный просмотр проекта, выявляются недоработки, намечаются пути устранения выявленных недостатков, производится корректировка.

Этап презентации – публичной защиты проекта

На этом этапе производится представление проделанной работы, дается оценка проекту членами аттестационной комиссии.

Таким образом, метод проектов позволяет затрагивать различные сферы будущей профессиональной деятельности, установить уровень сформированности компетенций, что доказывает целесообразность его применения в процессе обучения.

Применение метода проектов предполагает высокую степень свободы обучаемых, им дается направление для исследования и проявления компетенций.

Разработка курса ориентирована на создание методического пособия для учителя в виде курса для среды электронного обучения MOODLE.

Курс «Робототехника для школьников» позволит учащимся переходить от манипулирования с заранее подготовленными наборами деталей к созданию собственных механизмов и приспособлений, используя

различные устройства и базовую микросхему-контроллер.

### **Литература**

1. Юревич Е.И. Основы роботехники. СПб.: БХВ-Петербург, 2007. 416 с.
2. Брага Н. Создание роботов в домашних условиях. М.: НТ Пресс, 2007. 368с.

### ***Использование технологии проблемно-диалогического обучения как один из путей формирования интеллектуальной и творческой личности***

В докладе изложена эффективная технология – технология проблемно-диалогического обучения, при которой учащиеся часть времени на уроке работают самостоятельно. Эта технология позволяет заменить урок объяснения нового материала, уроком открытия знаний, так же она применима для любого этапа урока и для любого предмета. Приведены примеры проблемных ситуаций на уроках математики и информатики.

Мы живем в “информационную” эпоху. Это означает то, что теперь самым важным продуктом считается информация, с которой мы должны научиться работать: искать, обрабатывать, хранить. Поэтому школьное образование ставит задачу – подготовить учащихся к жизни и профессиональной деятельности в информационной высокоразвитой среде, получить образование с использованием современных технологий обучения.

В настоящее время информатика один из немногих инновационных и востребованных предметов, делающих школу современной, приближенной к жизни и запросам общества. Содержание обучения информатики не ограничивается на самом деле только информационными технологиями, а несет в себе большой потенциал, присущий данному предмету. Ни компьютер, ни информационные технологии сами по себе не способны сформировать у выпускников присущие им качества (этические, интеллектуальные и другие), они являются лишь вспомогательными средствами решения мировоззренческих задач, а найти эти решения учащийся может лишь с помощью грамотного, творчески работающего учителя.

Задачей учителя на уроках математики, информатики и других предметах является формирование у учащихся информационной компетентности – одного из основных приоритетов в современном общем образовании, который носит общеучебный и общеинтеллектуальный характер.

Что нужно современному молодому человеку в век современных информационных технологий для того, чтобы чувствовать себя комфортно в новых социально-экономических условиях жизни? Какую роль должна играть школа? В меняющемся мире меняются требования к выпускнику школы. Он должен быть творческим, самостоятельным, ответственным,

коммуникабельным человеком, способным решать проблемы личные и коллектива. Ему должны быть присущи потребность к познанию нового, умение находить и отбирать нужную информацию.

Для того чтобы развить личность ученика, его необходимо подготовить к самостоятельной деятельности в условия нового информационно развитого общества. Сформировать информационную культуру, коммуникативные навыки, привить умение исследовательской деятельности и принятия сложных решений. Во время урока, кроме обучения, необходимо подготовить конкурентоспособного, готового существовать в сложившейся социальной среде человека.

Кроме того, использование ИКТ на уроках позволит повысить качество образовательного процесса, углубит межпредметные связи. Именно поэтому постепенная интеграция информатики с другими общеобразовательными предметами становится с каждым годом все активнее. Использование ИКТ – это требование времени, но главное место в учебном процессе принадлежит учителю. ИКТ позволяют по-новому «зазвучать» уроку, используя самые различные источники информации: текстовую, звуковую, графическую и видеоинформацию.

На современном этапе меняется и позиция учителя. Он перестает быть вместе с учебником носителем “знания”, которое он пытается передать ученику. Его главной задачей становится мотивировать учащихся на проявление инициативы и самостоятельности.

Могу утверждать, что новые технологии самостоятельного обучения имеют в виду, прежде всего повышение активности учащихся: истина, добытая путем собственного напряжения усилий, имеет огромную познавательную ценность. Успех обучения в конечном итоге определяется отношением учащихся к учению, их стремлению к познанию, осознанным и самостоятельным приобретением знаний, их активностью.

Чем разнообразнее и эмоциональнее представлен подлежащий усвоению материал, тем успешнее процесс обучения. Безусловно, это требует изменения формы общения учителя и ученика, превращая обучение в деловое сотрудничество, а это усиливает мотивацию обучения, приводит к необходимости поиска новых моделей занятий, повышает индивидуальность и интенсивность обучения.

В процессе обучения учителю необходимо выбирать такие технологии обучения, при которых учащийся большую часть времени работает самостоятельно и учится планированию, организации, самоконтролю и оценке своих действий и деятельности в целом. Одной из таких технологий является технология проблемно-диалогического обучения. Эта технология позволяет заменить урок объяснения нового материала уроком «открытия» знаний и она применима для любого этапа урока и для любого предмета.

Например: на уроке изучения нового материала должны быть проработаны два звена:

1. Постановка учебной проблемы (это этап формулирования темы урока или вопроса для исследования);
2. Поиск решения (это этап формулирования нового знания).

Постановку учебной проблемы и поиск ее решения осуществляют ученики в ходе специально организованного учителем диалога.

Различают два вида диалога:

1. Побуждающий
2. Подводящий.

**Побуждающий диалог** состоит из отдельных стимулирующих реплик, которые помогают ученику работать творчески, и поэтому развивает творческие способности учащихся. На этапе постановки проблемы этот метод выглядит следующим образом. Сначала учителем создается проблемная ситуация, а затем произносятся специальные реплики для осознания противоречия и формулирования проблемы учениками. На этапе поиска решения учитель побуждает учеников выдвинуть и проверить гипотезы, т.е. обеспечивает «открытие» знаний путем проб и ошибок.

**Подводящий диалог** представляет собой совокупность (посильных учащимся) вопросов и заданий, которая активно развивает логическое мышление учеников. На этапе постановки проблемы учитель постепенно подводит учеников к формулированию темы. На этапе поиска решения он выстраивает логическую цепочку к новому знанию. При этом подведение к знанию может осуществляться как от поставленной проблемы, так и без нее.

Таким образом, на проблемно-диалогических уроках учитель сначала с помощью диалога помогает поставить учебную задачу, т.е. сформулировать тему урока или вопрос для исследования. Тем самым у школьников вызывается интерес к новому материалу, познавательная мотивация. Затем учитель с помощью диалога организует поиск решения, т.е. «открытие» знания школьниками. При этом достигается понимание материала учениками, так как «нельзя не понимать того, до чего додумался лично».

Технология проблемно-диалогического обучения является **результативной** (обеспечивает высокое качество усвоения знаний, эффективное развитие интеллекта и творческих способностей школьников, воспитывает активную личность), **здоровьесберегающей** (позволяет снижать нервно-психические нагрузки учащихся за счет стимуляции познавательной мотивации и «открытия» знаний) и **общепедагогической** (реализуется на любой ступени и в любом предмете).

Желание учиться у ученика не появляется само собой, его надо целенаправленно и последовательно формировать, и для учителя принципиально важно – что и как формировать. И поэтому проблема создания адаптивной среды как условия, обеспечивающего самореализацию школьников, должна быть одной из главных. И помогает в решении этой проблемы, на мой взгляд, ТПДО (технология проблемно-

диалогического обучения). ТПДО – это такая организация учебных занятий в виде диалога, которая предполагает создание под руководством учителя проблемных ситуаций и активную самостоятельную деятельность учащихся по их разрешению, такими методами, решения которых возможно ему еще неизвестны, в результате чего происходит творческое овладение профессиональными знаниями, навыками, умениями и развитие мыслительных способностей.

Цель использования данного метода — творческое, интеллектуально-познавательное усвоение учеником заданного предметного материала.

Задачи использования: формировать у детей умения переносить полученные знания на новые, нестандартные ситуации, стимулировать учеников к дальнейшей работе, актуализировать (сделать значимой для самого ребенка) учебную цель, поставленную на уроке.

Главная задача каждого учителя сегодня - не только обеспечить прочное и осознанное усвоение знаний, умений и навыков, но и развитие способностей учащихся, приобщение их к творческой деятельности.

Проблемное обучение – это «начальная школа» творческой деятельности. Поэтому, для меня, в процессе обучения, главным является постановка перед учащимися на уроках проблем и стремление решить их вместе с ними.

Главный фактор занимательности – это приобщение учащихся к творческому поиску, активизация их самостоятельной исследовательской деятельности, так как уникальность занимательной задачи служит мотивом к учебной деятельности, развивая и тренируя мышление вообще и творческое, в частности.

Несколько примеров проблемных ситуаций на уроках математики и информатики:

1) Тема “Начальные сведения об операционной системе”

В ходе урока учитель знакомит учащихся с понятием операционной системы, какие программы, служебные, прикладные, входят в операционную систему.

Далее учащиеся знакомятся с этапами загрузки операционной системы: сначала изучается теория вопроса, затем с помощью мультимедиапроектора они визуально наблюдают за этапами загрузки операционной системы с комментарием учителя. После чего проделывают то же самое, но парами, сидя на рабочих местах за персональными компьютерами.

Следующий этап урока заключается во фронтальном опросе учащихся по основным устройствам компьютера (к этому моменту изучена тема “Функциональная схема компьютера”): для чего служат устройства компьютера?

Проблемная ситуация: Как можно назвать одним словом все то, что было перечислено? И имеет ли операционная система “это”? Учащиеся должны прийти к понятию “функции”. Учитель подводит

учащихся к этому понятию. Как и любой другой объект, предмет или явление операционная система тоже имеет ряд конкретных функций.

Далее учащиеся с наглядным показом учителя (учитель не называет эти функции) пытаются сами сформулировать эти функции операционной системы (ряд проблемных ситуаций: учитель создает – учащиеся разрешают).

## 2) Тема: «Формулы сокращённого умножения»

Создание проблемных ситуаций через противоречие нового материала старому, уже известному.

Вычисляем  $(3 + 5)^2 = 8^2 = 64$

Попробуем сосчитать по-другому:  $(3 + 5)^2 = 3^2 + 5^2 = 9 + 25 = 34$

Проблемная ситуация создана. Почему разные результаты?

$(3 + 5)^2 \neq 3^2 + 5^2$

Вывод: формулы сокращённого умножения необходимо знать и использовать для решения.

## 3) $28k + 30n + 31m = 365$

Задание для учащихся:

Найти хотя бы одно решение уравнения.

Комментарий учителя к уравнению:

Говорят уравнение, вызывает сомнение, но итогом сомнения может быть озарение!

Кто увидел? Кто догадался? Кто решил?

Не забывайте: “Смотреть – не значит видеть!”

(Ответ: 365 – это количество дней в году, 28 – количество дней в феврале, 30 – количество дней имеют 4 месяца в году, 31 – количество дней имеют 7 месяцев в году. Тогда:  $28 \cdot 1 + 30 \cdot 4 + 31 \cdot 7 = 365$ ).

## 4) Тема “Алгоритмическая структура ветвление”

Ученики знают понятие языка программирования, умеют составлять и запускать программы на основе линейных алгоритмов. В начале урока дается задание простого содержания практического характера: нужно написать программу для вычисления значения функции  $y = x^2$  для  $x$ , вводимого с клавиатуры. Никаких проблем то и не возникнет!

Затем предлагается задача вычислить значение функции  $y = \sqrt{x}$ . Дается задание вычислить значение функции при  $x = 4$ ,  $x = 9$ ,  $x = -4$ .

Для последнего варианта программа выдаст ошибку. Возникла проблемная ситуация: программа не может вычислить пример. Что для этого нужно сделать, ведь структуру, которую использовали в алгоритме, не подходит? (об алгоритмической структуре ветвление они еще не знают).

Поступит идея, что  $x$  должно быть непременно больше 0, но этот довод не принимается. В условии было сказано, что  $x$  вводится с клавиатуры, то есть  $x$  – любое. Как же быть?

Нужно подойти к мысли, что  $x$  можно ввести любое, но не все  $x$  годятся для вычисления. Следовательно, между вводом  $x$  и вычислением  $y$

необходим этап проверки  $x$  на доступность.

И при этом незаметно начинается новая тема – алгоритмическая структура ветвление, ее реализация средствами Turbo Pascal. В данном случае использовался педагогический прием практического затруднения, а это способствует активной мыслительной деятельности учащихся.

#### 5) Тема «Сумма $n$ -первых членов арифметической прогрессии»

Изучение вопроса о сумме  $n$ -первых членах арифметической прогрессии в 9-ом классе начинаю с рассказа: “Примерно 200 лет тому назад в одной из школ Германии на уроке математики учитель предложил ученикам найти сумму первых 100 натуральных чисел. Все принялись подряд складывать числа, а один ученик почти сразу же дал правильный ответ. Имя этого ученика Карл Фридрих Гаусс. В последствии он стал великим математиком. Как удалось Гауссу так быстро подсчитать эту сумму?”

Проблемная ситуация: как найти быстро сумму первых 100 натуральных чисел?

Решение проблемы  $(1 + 100) \times 50 = 5050$

Последовательность чисел 1, 2, 3,...,100 является арифметической прогрессией. Теперь выводим формулу суммы  $n$ -первых членов арифметической прогрессии.

#### 4) Тема «Линейные уравнения с одной переменной».

Создание проблемных ситуаций через умышленно допущенные учителем ошибки.

Решаю быстро уравнение:

$$(3X + 7) \cdot 2 - 3 = 17$$

$$6X + 14 - 3 = 17$$

$$6X = 17 - 14 - 3$$

$$6X = 0$$

$X = 0$  Естественно при проверке ответ не сходится Проблемная ситуация. Поиск ошибки. Дети решают проблему. После этого учащиеся очень внимательно следят за мыслью и решением учителя. Результат - внимательность и заинтересованность на уроке.

Проблемное обучение – система методов и средств, обеспечивающих возможности творческого участия учащихся в процессе усвоения новых знаний, формирование творческого мышления и познавательных интересов личности. Проблемное обучение возникло в истории педагогики как реакция на схоластические методы обучения, как ответ на систему методов, использующих в качестве основных тренировку и запоминание без понимания, усваиваемого учебного материала. В истории педагогики Проблемное обучение имело различные названия – сократический метод обучения (по имени древнегреческого философа Сократа), эвристический метод обучения, исследовательское обучение и другие.

Сегодня под проблемным обучением понимается такая организация



учебных занятий, которая предполагает создание под руководством учителя проблемных ситуаций и активную самостоятельную деятельность учащихся по их разрешению, в результате чего и происходит творческое овладение профессиональными знаниями, навыками, умениями и развитие мыслительных способностей.

Не так важно приобретенное знание, как развитие способности мышления.

### **Литература**

1. Гринько О.Е. Роль ИКТ в формировании информационных компетенций учителя и учащихся. Тезисы докладов VI Региональной научно-практической конференции 28 марта 2012 г. Под ред. проф. А.С. Потапова. Воронеж: Изд-во ВГПУ, 2012 52 с.
2. Мельникова Е.Л. Проблемно-диалогическое обучение, <http://pdo-mel.ru>, 2011.

## **Научно-исследовательская деятельность в научном обществе учащихся (секция «Информатика»)**

### **Аннотация**

*В докладе рассматриваются вопросы научно-исследовательской деятельности и развития познавательной самостоятельности учащихся в процессе подготовки и участия в работе секции «Информатика» конференции Научного Общества Учащихся (НОУ), а также работа секции в период с 2007 по 2012 год.*

### **Научно-исследовательская деятельность и развитие познавательной самостоятельности**

На современном этапе сложилось твердое убеждение в том, что недостаточно сообщить учащимся определенные знания, необходимо еще научить их творчески мыслить и самостоятельно работать. Ведь они получают только основы знаний для дальнейшего обучения и, в дальнейшем, профессиональной деятельности, и если при этом не будут постоянно повышать свой творческий уровень, то это может привести к сложностям при выполнении индивидуальных заданий, курсовых, выпускных квалификационных работ.

Развитие техники и технологий идет очень быстрыми темпами, и востребованный современный специалист должен быть готов к самостоятельному, быстрому и экономному поиску и использованию новых научных данных, получаемых из всевозможных источников информации. Необходимым сегодня становятся также умение организовать новые исследования, теоретически обосновать поставленные задачи, увидеть и осуществить их практическое решение, а также донести свое видение проблемы и решений до определенной целевой аудитории. Это означает, что на первый план в обучении выходит вопрос формирования познавательной самостоятельности и навыков научно-исследовательской деятельности [1]. Актуальной является и задача формирования у учащихся устойчивой потребности в получении новых знаний, в изменении и преобразовании самого себя. Объясняется это необходимостью воспитания личности, способной решать множество проблем, разнообразных теоретических и практических задач.

В формировании познавательного интереса можно выделить несколько этапов: любопытство, любознательность, устойчивый интерес. Первое представляет собой естественную реакцию человека на все неожиданное, интригующее и может быть вызвано неожиданным

результатом опыта, интересным фактом, наблюдением. Более высокой стадией интереса является любознательность, когда учащийся проявляет желание глубже разобраться и понять изучаемое явление, когда он задает вопросы, участвует в обсуждении результатов демонстраций, приводит свои примеры, читает дополнительную литературу и т. д. Любознательность менее ситуативна, чем любопытство, но она может и не распространиться на изучение всего предмета. Следующая стадия – формирование устойчивого интереса к предмету, когда учащегося захватывает процесс постижения новых знаний, самостоятельного решения проблем и нестандартных задач. Школьники, проявившие любознательность и познавательный интерес, могут привлекаться к работе в школьном научном обществе. Самостоятельная исследовательская деятельность во многом способствует развитию познавательного интереса. Она может проводиться как в рамках программ дополнительного образования, так и при выполнении обязательной школьной программы [4].

Стремительно меняющаяся жизнь заставляет пересматривать роль и значение исследовательского поведения в жизни и исследовательских методов обучения в практике образования. Становится все более очевидно, что умения и навыки исследовательского поиска требуются не только тем, чья жизнь уже связана или будет связана с научной работой, они необходимы каждому человеку. В первую очередь, такие навыки могут пригодиться при обучении в высших учебных заведениях. Научно – исследовательская деятельность является такой формой обучения, которая позволяет наиболее полно выявлять и развивать интеллектуальные и потенциальные творческие способности, а также максимально приблизить учебную деятельность к познавательной и развить познавательный интерес.

Научно-исследовательская работа – достаточно сложная деятельность и для ее участников, и для ее организаторов, которая характеризуется педагогической целесообразностью приобщения старшеклассников к самостоятельной работе над выбранной тематикой, что способствует формированию теоретического мышления. Кроме того, при работе над собственными задачами участники Научного Общества Учащихся (НОУ) учатся логически последовательно излагать материал [5].

#### ***Секция «Информатика» конференции НОУ***

Как уже упоминалось, развитию познавательного интереса и познавательной самостоятельности способствует выполнение самостоятельной исследовательской работы. Одной из форм организации такой работы является научное общество учащихся, работа которого давно стала сложившейся традицией в подготовке учащихся к дальнейшему обучению в ВУЗах. С 1986 года научное общество существует и при Воронежском государственном университете [2].

На выполнение индивидуальных и коллективных творческих

заданий в рамках НОУ отводится часть учебного года с сентября по март. Учащиеся могут самостоятельно избирать тему для работы и научного руководителя. По итогам работы над индивидуальными исследовательскими темами учащиеся представляют творческие отчеты.

Завершенные работы представляются на школьном заседании научного общества, а заключительным мероприятием, подводящим итоги работы за год, является конференция НОУ, проводимая в Воронежском государственном университете (ВГУ) в начале апреля. На этой конференции представляются доклады, ставшие лучшими на школьном этапе. Оценивание уровня отобранных на школьном этапе работ производится преподавателями, аспирантами, выпускниками, магистрантами и студентами факультета Прикладной математики, информатики и механики.

При подготовке к конференции НОУ выполняются практические работы, учащиеся получают навыки обработки и оформления результатов исследований, работы с научной литературой, выполнению реферативной части работы. Участие в подготовке к конференции является важным фактором формирования познавательной самостоятельности, что в дальнейшем может существенно облегчить работу будущих студентов при выполнении курсовых работ, индивидуальных заданий, выпускных квалификационных работ и магистерских диссертаций.

Следует отметить, что существуют различные программы дополнительного образования, ориентированные на изучение основ научно-исследовательской деятельности, однако, большая их часть ориентирована на узкоспециализированные области. Работа в рамках НОУ в секции «Информатика» в этом аспекте выгодно отличается от многих других: зачастую доклады затрагивают не только область информатики как таковой, но другие области знаний, например, при создании обучающих web-ресурсов по различным предметам. Межпредметность предполагает наличие большей эрудиции в разных областях, чем при выполнении работ в одной области, а также формирование навыков исследовательского труда. Можно сказать, что в последние годы интерес к предмету «Информатика и ИКТ» значительно возрос. Более того, изменились и мотивы изучения этого предмета. Изначально компьютер являлся новым и незнакомым устройством, вызывающим желание познакомиться с ним. Но в дальнейшем для многих учащихся компьютер стал фактически бытовым прибором, знакомым с малых лет. Следовательно, стали необходимы новые подходы к формированию познавательных интересов учащихся на уроках информатики. В частности, учителю нужно выявить интересы школьника и предложить такие задания, которые опирались бы на эти интересы. При этом задача должна содержать не только материал, представляющий интерес для учащихся, но и новую информацию, связанную с содержанием предмета [7].

Полученные в процессе работы над своей темой навыки могут пригодиться учащимся при выполнении самостоятельных творческих

заданий в любой области. Кроме того, учащиеся привыкают к определенной форме общения во время конференции: обучаются слушать, задавать вопросы, четко отвечать, дискутировать, также формируется навык защиты творческих работ.

Работы могут выполняться индивидуально или в составе коллектива соавторов, что позволяет учащимся получить навыки как самостоятельной, так и коллективной познавательной деятельности.

Важным моментом в работе НОУ является выполнение работ на базе различных школ с последующим представлением лучших докладов на общей сессии в ВГУ, что позволяет обмениваться опытом и вносить коррективы в дальнейшую деятельность. Зачастую, учащиеся, выполнившие свою первую самостоятельную научно-исследовательскую работу, в следующем году представляют заметно более сильные и серьезные доклады. Значительными с точки зрения качества представляемых работ и характера их восприятия является изложение алгоритма представления докладов, представляемые на проводимых в течение года консультациях и пленарном заседании секции «Информатика» НОУ. Этот алгоритм может использоваться при организации научно-исследовательской работы учащихся в школьных научных сообществах, а также может быть адаптирован для проведения научно-исследовательской работы учащихся по другим предметам школьного курса.

Конференция НОУ ориентирована на учащихся старших классов. Возраст участников НОУ составляет обычно 15-18 лет. Однако следует отметить, что за последние годы появилось значительное число учащихся средних классов, в том числе, учащихся 5 классов, что оказало заметное влияние на характер представляемых работ и привело, к сожалению, к увеличению числа работ, носящих реферативный характер. Это объясняется господствующей в школе тенденцией по выработке навыков подбора материала по заданной теме и приобретению учащимися реферативных навыков. Однако следует отметить, что учащиеся, занимающиеся в научном обществе в течение нескольких лет, к выпускному классу приобретают хорошие навыки организации научно-исследовательской деятельности и публичного представления полученных результатов, что, безусловно, облегчает дальнейший процесс обучения и помогает в воспитании активной жизненной позиции.

Деятельность учащихся организуется по подсекциям по нескольким основным направлениям, состав которых может меняться в зависимости от интересов учащихся, выполняющих творческие работы. Традиционно в формате секции предусмотрена подсекция «Системное и прикладное программное обеспечение». Кроме того, в течение нескольких лет доклады были посвящены тематике «Звук, изображение, моделирование», «Обучение и тестирование», «Web-приложения», «Моделирование и обработка видео и изображений», «Web-дизайн», «Компьютерная графика и программирование», «Web-дизайн и мультимедиа».

Рассмотрим результаты работы секции «Информатика» конференции НОУ за 2007-2012 годы [2].

Существенную часть докладов в 2007 году составили доклады подсекции «Web-дизайн», в работе которой приняли участие 32 учащихся из общего числа в 57 участников.

Участие в работе XXIII конференции в 2008 году принимали 69 учащихся, из которых только один был представителем 9 класса. Все три проводившиеся подсекции характеризовались хорошим явкой участников и высоким уровнем представления докладов.

Год 2009 отличался от предыдущих тем, что впервые в рамках работы секция «Информатика» конференции НОУ проводилась секция для учителей «Принципы преподавания информатики в общеобразовательной школе», в которой приняли участие учителя, осуществлявшие руководство научно-исследовательской деятельностью учащихся в НОУ.

Участие в работе секции в юбилейном 2010 году приняли 52 учащихся, которые представили 44 доклада. Особенностью этого года стала высокая активность учащихся 11 классов, причем большая их часть заинтересовалась работой подсекции «Web-дизайн», а единственная представительница 8 класса, которая принимала участие и в работе последующих конференций, представила доклад в подсекции «Системное и прикладное программное обеспечение».

В 2011 году к участию в работе в секции «Информатика» было заявлено 57 докладов. В некоторых случаях произошло изменение темы доклада, часть докладов была перенесена в другие подсекции по причине более точного соответствия тематике. Причиной этого стало то, что аннотации многих докладов не дали полного представления о сути работы, что повлияло на распределение докладов по подсекциям. Следует отметить более высокую по сравнению с предыдущими конференциями активность учащихся 9 классов, число выступлений которых составило почти треть от общего числа представленных докладов.

В текущем, 2012 году, работа велась в рамках подсекций «Системное и прикладное программное обеспечение», «Обучающие web-ресурсы», «Общие вопросы информатики». К обсуждению было представлено 39 докладов 42 участников, что означает, что в этом году были представлены в основном индивидуальные разработки. При этом следует отметить участие ученицы 5 класса и 2 учеников 8 класса, что говорит о том, что интерес к научному обществу и выполнению индивидуальных творческих работ стал проявляться и в среднем звене школы. Для оценивания представленных работ были выработаны требования, учитывающие как собственные достижения автора, так и его эрудированность в рассматриваемой области, а также качество представления работы и оформления презентации. В числе предъявляемых требований можно назвать:

- использование известных результатов и фактов;
- наличие сведений о разработках в выбранной области;
- формулировка темы и полнота ее раскрытия;
- использование знаний, получаемых вне школьной программы;
- новизна и практическая значимость работы;
- является ли представленный результат самостоятельно разработанным программным обеспечением или рефератом по выбранной теме;
- качество ПО, адекватность выбора среды разработки ПО;
- сложность и оригинальность проекта;
- логика изложения и убедительность рассуждений;
- умение докладчика представлять проект и отвечать на вопросы.

При определении лучших работ учитывается также интерес участников к представленным докладам. Ежегодно лучшие доклады награждаются дипломами I, II и III степени и почетными грамотами, часть докладов-победителей была опубликована. Начиная с этого года, в научном обществе учащихся при ВГУ авторам лучших работ присуждаются именные дипломы. Для секции «Информатика» - это диплом имени к.ф.-м.н. Виталия Евгеньевича Калечица, основателя кафедры математического обеспечения ЭВМ факультета Прикладной математики, информатики и механики.

Для повышения эффективности работы секции «Информатика» и оперативности предоставления информации был реализован сайт секции [3].

### **Выводы**

Итак, суммируя, можно сказать, что в результате работы над проектами в рамках НОУ учащиеся приобретают следующие навыки:

- формулирования темы научного исследования и определения ее ценности с позиций различных категорий лиц;
- разработки плана работы;
- организации научно-исследовательской работы,
- работы с научной и научно-популярной литературой и фактическими материалами;
- обработки полученных данных;
- ассоциативного, логического и образного мышления;
- индивидуальной или коллективной работы;
- публичного представления результатов своей деятельности;
- творческих и аудитивных навыков;
- участия в дискуссиях;
- уважительного отношения к результатам не только своего интеллектуального труда, но и интеллектуального труда других участников.

Очевидно, что среди перечисленных навыков присутствуют и

обучающие, и воспитательные, и развивающие элементы. Участие в работе НОУ способствует развитию и совершенствованию познавательных интересов, творческой активности, расширению кругозора, развитию познавательной самостоятельности и интереса, как к информатике, так и к другим предметам, имевшим отношение к проведенной научно-исследовательской работе.

Кроме того, нельзя не отметить, что проведение самостоятельной исследовательской работы способствует также развитию и воспитанию различных личностных качеств, таких, как усидчивость, аккуратность, настойчивость, самостоятельность мышления и действия, целеустремленность, индивидуальность и других.

Задачи, решаемые учащимися при подготовке к конференции НОУ, можно охарактеризовать как задачи повышенной трудности, не являющиеся стандартными как по условию, так и по методам решения. Все это требует от учащихся и наличия творческого начала, и предельного внимания, и воли в преодолении трудностей, и способностей к освоению нового материала, выходящего за рамки обычных школьных курсов, и хороших навыков в решении обычных школьных задач. Все в совокупности приводит к тому, что в ряды участников НОУ попадают неординарные, способные ученики. В связи с этим проведение самостоятельной научно-исследовательской работы имеет еще одно немаловажное значение – социально-психологическая адаптация личности учащегося, преодоление дискомфорта одаренного ребенка, формирование успешности, подготовка к самостоятельной жизнедеятельности в условиях рыночных отношений [6]. Многие выпускники НОУ связывают свое дальнейшее обучение с теми направлениями наук, по которым проводили научные исследования и достигли определенных результатов, причем при дальнейшем обучении показывают хорошие результаты при выполнении самостоятельных заданий, курсовых и выпускных квалификационных работ. Замечено, что высокая активность, интерес к исследовательской работе сохраняются и после поступления в высшие учебные заведения. Школьники из НОУ, как правило, становятся активными членами студенческого научного общества, ведут серьезную исследовательскую работу, принимают участие в работе секции «Информатика» конференции НОУ уже в качестве членов комиссии. Некоторые участники НОУ, получив образование, помимо основной профессиональной деятельности, ведут занятия в детских секциях и кружках. Многие участники секции Информатика в дальнейшем добились значительных успехов в научной и профессиональной деятельности.

### **Литература**

1. Односум Л.А. Формирование готовности к самостоятельной творческой деятельности у будущего инженера в вузе : диссертация ... кандидата педагогических наук : 13.00.08. Калининград, 2006. 187 с.: ил. РГБ ОД, 61 07-13/669 // [Электр. рес.]



<http://www.lib.ua-ru.net/diss/cont/307694.html>

2. Информация для абитуриентов ВГУ // [Электр. рес.] [www.abitur.vsu.ru](http://www.abitur.vsu.ru)
3. Крыжановская Ю.А. Секция "Информатика" конференции Научного Общества Учащихся. Разработка сайта секции / Ю.А. Крыжановская, Д.В. Игнатенко // Современные информационные технологии и ИТ-образование: сб. науч. тр. VI междунар. науч.-практ. конф. М., 2011. С. 617-621
4. Тувина Л.П., Кадолова Н.В. Проект школьного научного общества учащихся "Поиск". // [Электр. рес.] <http://festival.1september.ru/articles/418816/>
5. Васильева Н.М. Научно-исследовательская деятельность учащихся в Городском научном обществе (ЦЭОКДЮТиО) // [Электр. рес.] <http://kids-vn.ru/op/op-00029.html>
6. Шаракова-Рангина Н.П. Программа развития школьного НОУ "Гранит" // [Электронный ресурс] <http://festival.1september.ru/articles/571995/>
7. Левченко И.В., Карташова Л.И. Компьютер как средство формирования познавательного интереса школьника на уроках информатики // МГПУ в московском и российском образовательном пространстве: Материалы международной научной конференции. Вып. I. М.: МГПУ, 2007. С. 260-265.

## ***Визуальные средства программирования роботов и их использование в школах***

### **Аннотация**

*В российских школах и вузах активно внедряются робототехнические конструкторы как учебные пособия для преподавания информатики и кибернетики. В статье предложены критерии сравнения визуальных сред программирования роботов, существенные для школьной информатики, приведено описание популярных существующих сред, представлена разрабатываемая на кафедре системного программирования СПбГУ система Qreal:Robots.*

### **Введение**

В школьной информатике активно используется понятие «исполнитель», как некая сущность, которая выполняет команды, описанные в программе. В качестве исполнителя в российских школах активно внедряются робототехнические конструкторы, самым популярным из которых на данный момент является конструктор Lego Mindstorms NXT [1]. Он позволяет из блока управления, моторов, сенсоров и соединительных деталей собирать роботов, способных под управлением программы сложным образом взаимодействовать с окружающим миром.

Существует довольно много систем, позволяющих программировать такие роботы, как текстовых, так и визуальных. В текстовых средах, как правило, применяются языки программирования, похожие на язык С, поэтому они слишком сложны для первоначального обучения информатике в младших и средних классах школы. С педагогической точки зрения более интересны визуальные средства программирования роботов, пользоваться которыми могут иногда даже дошкольники, не умеющие ещё читать. Существует несколько популярных средств, шире всего распространена на данный момент среда Robolab [2], иногда в школах используются среды NXT-G и Microsoft Robotics Developer Studio [3]. В этой статье проводится анализ перечисленных сред с точки зрения пригодности их для преподавания информатики и кибернетики в школах, делаются выводы об имеющихся недостатках, определяются направления развития подобных систем, и представлена разрабатываемая на кафедре системного программирования Санкт-Петербургского Государственного Университета среда программирования QReal:Robots.

### **Мотивация использования программирования роботов в школах**

Идея использовать роботов при начальном обучении информатике

родилась неслучайно. Проблема, на которую указывал ещё Ф. Брукс в своей известной книге «Мифический человеко-месяц» [4], заключается в том, что программы нематериальны, их невозможно увидеть. Кроме того, даже представить себе программу не так просто – каждый человек “видит” программу по-разному. Людям, которые программируют впервые, приходится сразу же работать с абстрактными понятиями, и судить о правильности своих программ они могут только по внешним проявлениям их работы – какой ответ программа выведет на экран. При этом может быть совсем не очевидно, как программа работает, и что делать, если выводимый ею ответ неправильный, что нужно делать, чтобы получить правильный ответ. К тому же часто случается так, что программа работает неправильно, но правильный ответ всё-таки выводит. Всё это делает изучение информатики весьма сложным.

И отечественные, и зарубежные методисты давно осознают эту проблему, поэтому традиционно начальное обучение информатике проводится с использованием концепции исполнителя – некоторого, зачастую воображаемого, устройства, способного выполнять простые команды в некотором простом окружении. Один из самых известных исполнителей, применяемых в школах – «черепашка» LOGO [5], разработанная американским программистом, психологом и педагогом Сеймуром Пейпертом в 1967 году. Исполнитель «черепашка» может перемещаться по экрану, оставляя за собой след, которым вычерчиваются различные фигуры. Черепашка подчиняется командам простого интерпретируемого языка, позволяющего описывать её перемещения и повороты. Таким образом, процесс исполнения программы визуализируется движением исполнителя по экрану, и если программа работает неправильно, это будет сразу видно.

В Советском Союзе преподавание информатики как школьного предмета началось во многом благодаря усилиям академика А.П. Ершова и его коллектива, в который входили Г.А. Звенигородский и Н.А. Юнерман. Ими была разработана отечественная учебная система «Робик» [6], основанная в основном на тех же принципах, что и LOGO. Ими же были разработаны методики и программы преподавания информатики в школах, где понятие «исполнитель» занимало ключевую позицию.

Однако исполнитель, перемещающийся по экрану, всё же недостаточно нагляден. Сеймур Пейперт в своих экспериментах использовал механическую черепашку [7] – реальный, осязаемый объект, исполняющий программу, что оказалось гораздо понятнее для школьников, чем черепашка, движущаяся по экрану. Современные технологии позволяют создавать недорогие механические устройства, управляемые загружаемой в них программой, либо непосредственно с компьютера, поэтому идея использования материальных исполнителей в школьной информатике получила второе рождение, из-за чего получил распространение конструктор Lego Mindstorms NXT. Для преподавания

информатики с использованием этого конструктора существуют методические пособия (например, [8]).

Робототехнический конструктор довольно сложно программировать: из набора деталей могут быть собраны самые разные конструкции, поэтому программировать приходится в терминах оборотов моторов, подключённых к определённым портам управляющего блока, а не в терминах движений и поворотов. Это, безусловно, делает процесс обучения более творческим, поскольку школьники могут собрать своего собственного исполнителя, но и более сложным с точки зрения написания для этого исполнителя программ. Проблема сложности программирования преодолевается использованием наглядных визуальных языков и удобных графических редакторов для составления программ из блоков, представляющих элементарные команды, такие как «включить мотор», «гудок» и т.д. Таким образом, начинающие работают с графическими языками программирования, а более опытные школьники постепенно переходят на текстовые С-образные языки. В комплекте с конструктором поставляется графическая среда программирования NXT-G, поэтому визуальные языки среди использующих Mindstorms NXT весьма популярны.

#### ***Критерии сравнения средств программирования роботов***

Определим критерии, по которым будем сравнивать существующие средства программирования роботов. Поскольку в контексте данной работы нас интересует использование роботов в школьной информатике, основной акцент в обзоре будет сделан на свойства продуктов, которые важны именно в этой сфере.

1. Возможность создавать довольно сложные программы, включающие в себя нетривиальные математические выражения, циклы, ветвления, переменные, параллельные задачи – применение таких средств должно дать возможность иллюстрировать содержательный материал из информатики и кибернетики, например, понятие регуляторов.
2. Простота и удобство в работе. Неудобный пользовательский интерфейс создаёт дополнительную когнитивную нагрузку на школьников и усложняет восприятие и без того сложного материала.
3. Наличие встроенных средств отладки, чтобы школьники могли следить за ходом выполнения своей программы и её состояниями, имели бы инструмент для эффективного поиска ошибок.
4. Возможность перехода от графической формы программы к текстовой, чтобы школьники старших классов, серьёзно занимающиеся программированием, имели возможность смотреть на то, как их программа выглядит на более приближенном к индустриальному программированию текстовом языке, и могли бы, возможно, вносить в программу правки в той же среде, в которой они привыкли работать.
5. Необходима русскоязычная среда разработки, поскольку школьники

зачастую ещё не владеют иностранными языками, а необходимость работать со словарём существенно усложняет восприятие материала.

6. Цена – каким бы хорошим ни был продукт, если он стоит дорого, не все школы могут себе его позволить.
7. Желательно, чтобы продукт продолжал развиваться и адаптироваться к новым операционным системам и аппаратному обеспечению.

### ***Среда NXT-G***

Среда NXT-G – единственное средство программирования, которое поставляется в комплекте с конструктором Lego Mindstorms NXT. Среда базируется на системе LabView, среде визуального программирования от компании National Instruments. В LabView в качестве языка программирования используется визуальный язык G (из-за чего NXT-G и получила своё название). Язык G моделирует процесс вычислений, ориентированный на данные, в котором явно задаётся не последовательность выполнения операторов, а связи между блоками по данным. Блок программы может предоставлять некоторые выходные данные, которые могут служить входными данными для другого блока. Блоки начинают исполняться, когда имеют данные на всех входах. Если сразу несколько блоков имеют данные на всех входах, они исполняются параллельно. Такой подход довольно сильно отличается от подхода, принятого в императивном программировании, но тем не менее он широко распространён среди инженеров и учёных. Например, на тех же принципах основана другая известная визуальная среда программирования научных вычислений и моделирования Matlab/Simulink.

Основная проблема этой среды состоит в довольно слабой поддержке математических выражений. Математические формулы здесь, как и вся программа, строятся из блоков. Есть блоки арифметических операций, блоки чтения и записи значения в переменную, блок, считывающий значение константы, блоки, считывающие показания с сенсоров. Таким образом, даже чтобы запрограммировать несложную формулу, требуется изображать блоками дерево разбора выражения, которое эту формулу задаёт. Для иллюстрации серьёзности этой проблемы достаточно сказать, что программа, представляющая пропорционально-дифференциальный регулятор для движения робота вдоль линии или вокруг препятствия, на языке C занимает порядка десятка строк, тогда как на NXT-G не помещается на одном экране и весьма сложна для понимания. Таким образом, первому из предложенных критериев – пригодности для иллюстрации содержательного материала информатики и кибернетики – NXT-G не соответствует. В основном поэтому NXT-G и не получил широкого распространения в школах.

Что касается простоты и удобства в работе, среда NXT-G специально создавалась для начинающих и поэтому довольно эргономична. По мнению некоторых пользователей она даже слишком эргономична, поскольку не

даёт произвольно размещать блоки на диаграмме, автоматически (и не всегда удачно) прокладывает соединительные линии между блоками и т.д. Для применения NXT-G в школьных классах оказалась важна ещё такая его особенность: большая часть свойств элементов не отображается на диаграмме, а доступна только через редактор свойств, что делает невозможным показ всей программы на проекторе. Никаких средств отладки NXT-G не имеет, текстовая форма программы не порождается, русификация существует, но неофициальная. К плюсам продукта следует отнести то, что он распространяется вместе с конструктором и доступен для скачивания с сайта производителя бесплатно. Кроме того, продукт до сих пор развивается и обновляется. Средствами LabView возможно добавление сторонних блоков, кроме того, сам NXT-G позволяет выделить набор блоков в подпрограмму и использовать её как новый блок.

### ***Среда Robolab***

Среда Robolab [2] так же, как и NXT-G, базируется на среде LabView. Robolab специально создавалась для школьного образования и с самого начала своего развития учитывала пожелания школьных учителей и специфику преподавания в школах. Пример специфичного для школ решения, реализованного в Robolab – наличие нескольких уровней возможностей среды. На самом простом уровне доступны только некоторые возможности визуального языка, и программа строится заполнением пустых мест в шаблоне посредством выбора блоков из всплывающего меню. Это позволяет создавать только самые простые программы, имеющие стандартную структуру: команды управления моторами, за которыми следует блок, ожидающий наступления какого-либо события. Причём, этот уровень имеет четыре подуровня, и на первых трёх подуровнях программа может состоять только из одного такого "шага". Сделано всё это для того, чтобы дать возможность детям в начальной школе или даже детском саду пользоваться программой – в столь раннем возрасте они вполне могут не уметь читать. На втором уровне (который тоже состоит из нескольких подуровней) пользователи могут рисовать уже настоящие диаграммы, размещая произвольным образом блоки из палитры и соединяя их линиями, определяющими поток управления. Разница между подуровнями заключается в количестве доступных в палитре блоков, первые подуровни имеют меньше блоков с меньшим количеством параметров. Разбиение на уровни и подуровни организовано так, чтобы дети могли осваивать среду программирования практически без помощи учителя, руководствуясь лишь интуицией. Отзывы учителей, приведённые в [2], показывают, что этой цели удалось достигнуть.

Математические выражения Robolab поддерживает гораздо лучше, чем NXT-G, позволяя писать произвольные выражения в текстовом виде. В Robolab имеется возможность использовать в формулах тригонометрические функции, обращаться напрямую к значениям показаний сенсоров. Циклы в Robolab реализованы довольно необычно –

есть блок "метка" и блок "переход к метке", передача управления никак больше не визуализируется. Имеются условные операторы, возможность порождать параллельные процессы, блоки для управления этими процессами, а также средства работы с подпрограммами. На Robolab можно просто и довольно удобно реализовать даже довольно сложные программы, и Robolab вполне подходит для иллюстрации материала из кибернетики до младших курсов вузов.

По другим критериям Robolab показывает несколько худшие характеристики. Приложение было создано в конце 1990-х годов, с тех пор его интерфейс практически не менялся, поэтому сейчас он выглядит несовременно. Кроме того, он довольно неудобен. Специализированных средств отладки в Robolab нет, хотя есть возможность снимать показания с робота и отображать на экране компьютера данные. Текстовое представление программы Robolab порождать не может. Русификация присутствует, но лишь частично, некоторые элементы управления не переведены. Robolab бесплатен, одна лицензия по стоимости сравнима с робототехническим набором, что для школ довольно дорого. Развитие Robolab идёт в основном путём добавления новых блоков, сама среда давно не изменялась.

Несмотря на указанные недостатки, Robolab на данный момент является основной средой, используемой в школах для разработки программ для роботов. По отзывам учителей, у многих имеется желание от него отказаться и заменить на что-нибудь более современное, однако пока на рынке не существует продуктов, которые могли бы составить ему серьёзную конкуренцию.

### ***Среда Microsoft Robotics Developer Studio***

Среда Microsoft Robotics Developer Studio [3] – разработка компании Microsoft, предназначенная для программирования сложных многопоточных приложений с реактивной моделью поведения, используемых для управления робототехническими системами. Необходимость создания таких приложений есть не только в робототехнике, поэтому Robotics Developer Studio используется и для создания приложений, к робототехнике не относящихся (например, социальная сеть MySpace использует MRDS как составную часть серверного ПО [9]). Программы в Robotics Developer Studio рисуются в виде диаграмм на визуальном языке VPL (Visual Programming Language), представляющем собой по сути визуализатор связей между отдельными параллельно исполняемыми компонентами (или веб-сервисами), из которых состоит программа. Система состоит из следующих крупных частей.

- Concurrency and Coordination Runtime (CCR) – библиотека для работы с параллельными и асинхронными потоками данных. Библиотека позволяет прозрачно организовывать распределённые и параллельные вычисления, исполняя задачи на разных

вычислительных устройствах. Это весьма полезно при программировании роботов, поскольку программы для роботов по природе реактивны и требуют обработки потоков данных одновременно с нескольких сенсоров, причём часть вычислений может быть сделана прямо на роботе, часть – на компьютере вне него.

- Decentralized Software Services (DSS) – среда времени выполнения, обеспечивающая представление компонентов программы в виде веб-сервисов и упрощающая организацию взаимодействия между ними. Взаимодействие между веб-сервисами ведётся по специальному протоколу Decentralized Software Services Protocol (DSSP). DSS позволяет создавать распределённые приложения, которым не важно, на каком вычислительном устройстве выполняется тот или иной компонент, на одном из компьютеров робота или на компьютере вовне, лишь бы они были связаны единой сетью.
- Visual Programming Language (VPL) – визуальный язык и редактор для него, используемый для конфигурирования сервисов. Сервисы можно перетащить на диаграмму, связать их входы и выходы, настроить их атрибуты. Получающаяся на таком языке диаграмма сильно напоминает диаграммы LabView, отображая зависимость между компонентами по данным.
- Visual Simulation Environment (VSE) – трёхмерная среда симуляции поведения робота в виртуальном мире. Обладает довольно богатыми возможностями по симуляции физики и богатыми средствами отображения трёхмерной графики, что позволяет строить сложные и красиво выглядящие модели мира, с которым робот может взаимодействовать. В поставку среды включено несколько моделей окружения, в том числе модель квартиры, в которой работает так называемая "стандартная модель" робота, трёхколёсная платформа с установленным на ней ноутбуком, сенсором Microsoft Kinect, инфракрасными датчиками расстояния и сонаром.

Необходимо отметить, что в сфере школьного образования Microsoft Robotics Developer Studio используется очень редко. Главная причина этого заключается в том, что среда рассчитана в основном на симуляцию и не может эффективно взаимодействовать с реальным роботом. Для LEGO Mindstorms NXT есть возможность управления по каналу Bluetooth, но залить программу на робот возможности нет – на роботе нет возможности запустить .NET-машину. Реальные роботы, управляемые MRDS, обычно гораздо сложнее и дороже того, что можно использовать в школах (стандартная платформа, например, имеет в своём составе ноутбук, который один, скорее всего, дороже всего набора Mindstorms). Управления по Bluetooth недостаточно для решения задач, требующих малого времени реакции робота, из-за больших задержек послышки-приёма Bluetooth-пакетов, что делает MRDS неприменимой для большой области решаемых в



школе задач. Симуляции же тоже оказывается недостаточно, потому что даже с хорошим физическим движком MRDS создаёт модель некоторого идеального мира, в котором большого количества проблем, решаемых алгоритмами кибернетики, просто не возникает. Даже простая задача, решаемая на реальном роботе, может оказаться нагляднее и полезнее школьникам, чем сложная программа, исполняемая на модели в симуляторе.

Вторая важная причина очень узкого распространения MRDS в школах – модель вычислений, в ней используемая. Представление программы в виде набора взаимосвязанных распределённых веб-сервисов может быть удобным для опытных программистов, но начинающим тяжело понять принципы, лежащие в основе такой модели. Сложные механизмы взаимодействия веб-сервисов во многом спрятаны с помощью визуального языка VPL, но всё же требуется некоторое понимание происходящих в системе процессов для того, чтобы рисовать содержательные диаграммы. В целом можно сказать, что MRDS больше подходит для студентов или профессиональных программистов, чем для школьников. Среда хоть и позволяет писать сколь угодно сложные программы, но делать это можно довольно нетривиальным и специфическим образом, что сильно снижает её ценность как иллюстративного материала.

Что касается других критериев, система довольно удобна в работе, имеет средства отладки и кодогенерации, однако, в силу своей специфики, эти средства довольно сложны для использования школьниками. Русификация системы отсутствует, однако же, система распространяется свободно и активно развивается.

### **Выводы**

Сводная таблица с результатами сравнения существующих визуальных средств программирования роботов представлена ниже.

*Табл. 1. Сравнение средств программирования роботов*

	<i>NXT-G</i>	<i>Robolab</i>	<i>MRDS</i>
Возможность создания сложных программ	Слабая поддержка сложных математических выражений	Присутствует всё, что требуется школьникам	Сколь угодно сложные программы
Простота и удобство в работе	Довольно эргономично, но неудобно для уроков	Удобна, но устаревший интерфейс	Довольно удобна, но сложна
Средства отладки	Нет	Нет	Есть отладчик
Текстовое представление программы	Нет	Нет	Генерирует код на C#
Русификация	Есть, неофициальная	Есть, частичная	Нет
Цена	Бесплатна	Порядка 5000 рублей за	Бесплатна

		лицензию	
Развитие	Развивается	Сама среда не развивается, только добавляют новые блоки	Развивается

Можно сделать вывод, что существующие среды, кроме Robolab, слабо подходят для преподавания информатики и кибернетики в школах. Кроме того, как видно из таблицы, среда Robolab также имеет ряд существенных недостатков, наиболее важный из которых для школ – высокая стоимость. Таким образом, существует потребность в создании бесплатной среды программирования, схожей по функциональности с Robolab, но имеющей средства отладки, генерации кода, русификацию и при этом удобную в работе.

### ***Среда QReal:Robots***

Среда программирования QReal:Robots [10] разрабатывается на кафедре системного программирования Санкт-Петербургского Государственного Университета с 2011 года на базе DSM-платформы QReal [11]. Разработка началась после того, как платформой QReal заинтересовались на кафедре прикладной кибернетики университета, и совместно со школьными учителями были определены требования к среде. На данный момент система находится в работоспособном состоянии, хотя некоторая нестабильность работы мешает её активному внедрению в школьный образовательный процесс.

В QReal:Robots модель вычислений строится вокруг понятия потока исполнения, в отличие от подхода, принятого в рассмотренных системах. Связи между блоками на диаграмме указывают, какой блок будет исполняться следующим, зависимости по данным между блоками не визуализируются. Такой подход оказывается более удобным для восприятия, поскольку ближе к императивным языкам программирования и интуитивному представлению о программе как последовательности команд исполнителю.

Рассмотрим QReal:Robots с точки зрения предложенных критериев. Математические выражения в QReal:Robots могут задаваться в текстовой форме, в любом месте, где требуется численное значение, кроме того, есть отдельный блок "функция", предназначенный для записи математического выражения. Имеется возможность использовать переменные и обращаться к текущим показаниям сенсоров прямо из выражения. В языке имеется поддержка алгоритмических конструкций – ветвлений, циклов, параллельных потоков исполнения. Удобство пользовательского интерфейса QReal:Robots обеспечивается, во-первых, возможностями базовой технологии QReal, во-вторых, исследованиями удобства пользовательского интерфейса конкретно QReal:Robots. Базовая технология обеспечивает, например, распознавание жестов мышью, которое используется в QReal:Robots для быстрого рисования связей между

блоками – достаточно с зажатой правой кнопкой мыши провести линию между двумя блоками, чтобы соединить их. Проводилось анкетирование школьников по вопросам удобства использования QReal:Robots, был выявлен ряд замечаний, которые были исправлены.

Наличие развитых средств отладки является основным преимуществом QReal:Robots перед рассмотренными системами. Первое – это возможность интерпретации программы на компьютере с дистанционным управлением роботом по Bluetooth или USB. При этом среда подсвечивает текущий исполняемый блок, визуализируя ход выполнения программы и давая возможность понять, какой блок вызвал ошибку. Второе – наличие двухмерной модели робота, которая может быть использована как исполнитель программы вместо реального робота. Двухмерная модель имитирует трёхколёсную тележку, применяемую, например, в футболе роботов. Модель позволяет задавать расположение и тип сенсоров, и объекты реального мира, такие как стены и линии на полу, с которыми взаимодействуют сенсоры. Двухмерная модель позволяет отлаживать программы без использования настоящего робота, что делает цикл отладки гораздо быстрее, и позволяет использовать QReal:Robots без доступа к роботу вообще, что может быть полезно для школ, пока не закупивших робототехнические конструкторы.

Среда QReal:Robots имеет генератор кода на языке C по диаграммам. Сгенерированный код можно скомпилировать и загрузить на робот для автономного исполнения, при этом его можно просмотреть прямо в среде QReal. Среда изначально разрабатывалась для русскоязычной аудитории, поэтому не требует русификации, бесплатна (и разрабатывается как продукт с открытым исходным кодом), продолжает активно развиваться силами студентов.

### ***Заключение***

Рассмотренные в статье среды программирования роботов обладают рядом недостатков, затрудняющих их применение в российских школах. Наиболее соответствует потребностям школьных учителей на данный момент среда Robolab, но она несколько устарела и небесплатна, поэтому существует потребность в новой среде программирования роботов. Такой средой может стать QReal:Robots.

Среда QReal:Robots была представлена на "Открытых состязаниях Санкт-Петербурга по робототехнике" и на робототехническом фестивале "Робофест 2012" в Москве. В качестве доказательства применимости среды QReal:Robots к реальным задачам, решаемым школьниками, команда студентов приняла участие в соревнованиях в движении робота по линии с программой, реализованной целиком на QReal:Robots. Несмотря на то, что для студентов участие в соревнованиях стало практически первым опытом решения задач робототехники, им удалось показать довольно неплохие результаты, заняв места в середине таблицы, несмотря на то, что довольно многие участники использовали специально созданные для этой задачи

роботы. QReal:Robots представлялся также в виде стендовых докладов, где вызвал большую заинтересованность у потенциальных пользователей. Было проведено анкетирование удобства пользовательского интерфейса, которое показало, что продукт достаточно хорош, чтобы вызывать у пользователей симпатию и желание им пользоваться.

### **Литература**

1. LEGO Mindstorms homepage, URL: <http://mindstorms.lego.com/en-us/Default.aspx>
2. Portsmore, Merredith, ROBOLAB: Intuitive Robotic Programming Software to Support Life Long Learning, APPLE Learning Technology Review, Spring/Summer 1999
3. Microsoft Robotics Developer Studio homepage, URL: <http://www.microsoft.com/robotics/>
4. Брукс Ф., Мифический человеко-месяц, или Как создаются программные системы, Символ-Плюс, 2010. 304 с.
5. MyRobot, Язык программирования Лого, URL: <http://myrobot.ru/logo/aboutlogo.php>
6. Звенигородский Г.А., Описание языка Робик, URL: <http://ershov.iis.nsk.su/archive/eaindex.asp?lang=1\&did=7639>
7. cyberneticzoo.com, 1969 – The Logo Turtle – Seymour Papert et al, URL: <http://cyberneticzoo.com/?p=1711>
8. Филиппов С.А., Робототехника для детей и родителей. М.: Наука, 2011. 264 с.
9. Michael S. Scherotter, CCR at MySpace, URL: <http://channel9.msdn.com/Shows/Communicating/CCR-at-MySpace>
10. Брыксин Т.А., Литвинов Ю.В., Среда визуального программирования роботов QReal:Robots // Материалы международной конференции «Информационные технологии в образовании и науке». Самара, 2011. С. 332-334.
11. Терехов А.Н., Брыксин Т.А., Литвинов Ю.В., Смирнов К.К., Никандров Г.А., Иванов В.Ю., Такун Е.И., Архитектура среды визуального моделирования QReal. // Системное программирование. Вып. 4. СПб.: Изд-во СПбГУ. 2009. С. 171-196

***Возможности применения информационно-коммуникационных технологий в качестве средства формирования Смыслжизненных ориентаций старшеклассников***

Активные процессы информатизации общества, начавшиеся в середине XX века, повлекли за собой изменения в общественной, экономической и политической жизни страны. Существенные преобразования произошли также на персональном уровне. С появлением персональных компьютеров, сотовой связи, глобальной сети Интернет трансформировались сферы общения, познания, развлечения. Учёные предлагают выделять новую ступень эволюции, когда человек из Homo Sapiens превращается в Homo Informaticus (Г. И. Аркадьев) или Homo Cyberus (В. А. Плешаков).

Наибольшее влияние новые технологии оказывают на молодое поколение, поскольку именно в юношеском возрасте человек наиболее восприимчив к изменению условий жизни. Современные подростки родились в мире, где активно развиваются информационные технологии. Они естественным образом восприняли бурное развитие Интернета и сотовой связи. Виртуальная реальность для них является неотъемлемой частью повседневной жизни.

Специфическая виртуальная культура, возникающая в информационном поле Интернета, наполнена своими нормами и ценностями. Её влияние на личность подрастающего поколения стало предметом изучения Г. И. Аркадьева, Т. А. Бондаренко, Е. П. Белинской, А. Е. Войскунского, А. Е. Жичкиной, И. С. Кона, В. Х. Манерова, В. А. Плешакова, В.С. Собкина и др. Учёные пришли к выводу, что информационное поле сети Интернет оказывает существенное влияние на ценностно-смысловую сферу личности. На наш взгляд, особенно значимым это воздействие оказывается в юношеском возрасте.

Ранний юношеский возраст, приходящийся на старшие классы школы, является сензитивным периодом для формирования личностных новообразований. Это время характеризуется активным развитием интеллектуальных и эмоциональных процессов. Формируются мировоззрение, система ценностных ориентаций и убеждения личности. У старшеклассников появляется интерес к своему внутреннему миру, происходит расширение временной перспективы, возникают вопросы о цели и смысле собственного существования.

Важным интегративным психическим новообразованием, отражающим внутреннюю работу личности по поиску цели и смысла своего существования, являются смысложизненные ориентации. Впервые в научный оборот этот термин был введен Д. А. Леонтьевым. Он рассматривал смысложизненные ориентации в качестве составной части смысла жизни, представляющей собой единство трёх компонентов жизнедеятельности и временных ориентаций жизни: цели (будущего), процесса (настоящего) и результата (прошлого). Цель определяется принятой личностью системой ценностей, процесс – эмоциональной насыщенностью жизни, а результатом является удовлетворённость самореализацией [5].

Смысложизненные ориентации связывают воедино личность и социальную среду. Они являются продуктом рефлексии событий внешнего мира. Необходимой основой формирования смысложизненных ориентаций являются знания. Только имея о вещах и явлениях предварительную информацию, человек может сформировать о них собственные суждения. В наши дни у подростков имеется неограниченный по объёму и скорости доступ к информации. Современные средства связи обеспечивают выход во всемирную сеть, позволяя быть всё время «он-лайн».

Как справедливо отмечает В. А. Плешаков, сегодня киберпространство становится новой средой жизнедеятельности человека. Сама интернетизация жизни становится насущной потребностью. Активное использование средств информационно-коммуникационных технологий (далее ИКТ) влечёт за собой процесс качественных изменений в мотивационно-потребностной сфере и сознании личности. Этот процесс обозначается учёным как киберсоциализация. В. А. Плешаков также говорит о необходимости сопровождения и адекватной помощи молодёжи в формировании киберкультуры личности [7.].

Для того, чтобы не уступить свои позиции в качестве важнейшего института социализации личности, школе необходимо учитывать современные социальные реалии. Отчасти эта проблема решается в рамках информатизации образования, одним из направлений которой является формирование информационной культуры обучающихся. Однако, практика образовательных учреждений показывает, что основной акцент делается на обучении умению работать с прикладным программным обеспечением. В то же время, такой важный компонент информационной культуры как информационная ценностно-смысловая сфера личности, в которой мировоззрение, потребности, мотивы, цели и стремления ориентированы на гуманистические ценности, выделенные С. Д. Каракозовым, упускается из вида [4]. В этой связи становится особенно актуальным изучение возможностей формирования смысложизненных ориентаций обучающихся средствами ИКТ.

Процесс становления смысложизненных ориентаций является двуединым – одновременно внутренним и внешним. Во внутреннем плане

человеком проводится самостоятельная сложная работа по осознанию самого себя и познанию окружающего мира. На основании соотнесения этих двух действий происходит рефлексивная оценка и изменение своего поведения, целей и задач, которые человек ставит перед собой. Во *внешнем плане* на процесс формирования смысложизненных ориентаций старшеклассников влияет специально организованная деятельность взрослого, в частности, педагога.

В исследовании Т. В. Шрейбер было выявлено, что для того, чтобы начался процесс осмысления личностью целей и смысла собственной жизни, необходимо повышение уровня притязаний и высокий уровень целей, стимулирование творческого и интеллектуального потенциала, рефлексивной деятельности, а также минимизация ухода от социальных контактов и расширение коммуникативных программ [9]. Кроме того, мы принимаем во внимание положение В. Франкла о том, что «Смысл должен быть найден, но не может быть создан» [8, с. 36]. Но найден в процессе вживания в ценности, прочувствования их на собственном опыте.

Современные информационно-коммуникационные технологии обладают амбивалентными свойствами. Нам важно использовать их сильные стороны и нейтрализовывать негативное воздействие. Основные направления работы по формированию смысложизненных ориентаций обучающихся средствами ИКТ состоят из следующего.

Во-первых, необходимо противопоставить альтруистические ценности прагматическим. Культурная среда Интернета и весь объем ее информации является для молодежи значимой и социально одобряемой [1]. Однако, поскольку средства массовой коммуникации функционируют по экономическим законам, информация, транслируемая через них, призвана формировать установку на потребление и поддержание материального благополучия. Мощное воздействие на систему ценностей личности происходит при восприятии человеком ролевых моделей, значений, смыслов и образов, содержащихся в информации, транслируемой через Интернет. При этом, чем интенсивнее происходит взаимодействие человека с информационной средой, тем глубже трансформируются представления, смыслы и нормы. Однако, если это взаимодействие разнопланово, то ценностный репертуар становится более богатым и насыщенным [3].

В связи с этим, важной задачей является формирование альтруистических мотивов. Для её реализации необходимо организовывать в рамках сетевых проектов акции милосердия, помощи детям-сиротам, ветеранам Великой Отечественной Войны. Взаимодействие с учениками школ других городов позволит увидеть масштабы и значимость проводимой работы. Знакомство с духовным наследием русской культуры, с произведениями искусства, философской мысли возможно организовать через посещение виртуальных библиотек и музеев, наполнение силами учеников разнообразных сервисов Web 2.0.

Школьникам важно дать стимул для размышления над «главными

вопросами жизни» (Н. И. Пирогов) через просмотр и обсуждение видеофильмов, в которых конструктивно решаются моральные дилеммы. Создание социальных видеороликов будет способствовать выработке активной жизненной позиции обучающихся.

Во-вторых, индивидуализм целесообразно заменить коллективизмом. Социальные сети, созданные для того, чтобы объединить людей во всём мире, получили обратный эффект. Пытаясь уйти от тревожащей социальной действительности и получить эмоциональную поддержку, люди стали всё больше замыкаться в пространстве своего внутреннего мира. Социальные контакты стали поддерживаться всё реже, в отличие от виртуальных. Эту негативную сторону интернет-коммуникации возможно нивелировать за счёт объединения старшеклассников в совместные сетевые проекты с учащимися других школ. Телеконференции, общение через Skype позволяют также объединить пользователей со схожими интересами.

В-третьих, следует развить эмоциональную сферу. Виртуальная реальность расширяет границы творческой свободы, предоставляет человеку возможность создавать желаемые в эмоциональном отношении атмосферу и ситуации. При этом неосознанное стремление к душевному комфорту провоцирует индивида аккумулировать в этой реальности все то, что вызывает позитивные эмоции, и исключать негативные. Такая ориентация превращается в установку. Это делает личность эмоционально неотзывчивой и черствой за пределами виртуального мира [6]. Формирование эмоциональной сферы, на наш взгляд, необходимо проводить по трём направлениям: развивать понимание чувств другого, восприятие прекрасного, рефлекссию собственных чувств. Первое может быть реализовано в расширении социальных контактов, второе – через знакомство с произведениями высокого искусства в противовес массовой культуре и эстетике безобразного. Третье направление эффективно решается за счёт ведения личных блогов и сетевых дневников.

В-четвёртых, формировать видение перспективы собственного будущего. Исследование И. Л. Васильевой интернет-зависимых студентов показало, что у них отсутствуют четкие цели на будущее, придающие жизни осмысленность, вектор развития и временную перспективу. Они не верят в то, что способны контролировать события своей жизни, и убеждены в иллюзорности свободы и поэтому не стремятся найти возможности для самореализации [2]. Следовательно, важной целью на старшей ступени школы является помощь школьникам в ценностном наполнении основ профессионального самоопределения, формирование ответственного отношения к своей судьбе и построение перспективы собственного будущего. В этом нам могут помочь различные он-лайн-тесты по профориентации, виртуальные экскурсии по вузам, беседы с успешными людьми через Skype, проведение вебинаров, видеоконференций и др.

Таким образом, жизнь в информационном обществе требует от



молодёжи умения уверенно обращаться с информационными потоками, творчески преобразовывать их и не быть объектом манипуляций. Этого невозможно достичь без твёрдой уверенности в собственных целях и ценностях. На наш взгляд, информационно-коммуникационные технологии, несмотря на их амбивалентность, обладают большим потенциалом в деле формирования смысложизненных ориентаций молодёжи. Это обусловлено социальным заказом, отвечает потребностям и интересам учащихся и соответствует основному каналу развития смысложизненных ориентаций – приращению знаний. Однако подобная работа осложняется отсутствием методической литературы. Разработка выделенных нами направлений может оказать существенную помощь в таком сложном и важном деле.

### **Литература**

1. Аркадьев Г. И. Влияние поля современных информационных технологий на личность в юношеском возрасте : автореф. дис. ... канд. психол. наук : 19.00.13 / М., 2003 - 24 с.
2. Васильева И. Л. Особенности структурно-содержательных характеристик смысложизненных ориентаций интернет-зависимых студентов вуза // В мире научных открытий. – 2010. – Январь. – С. 121–125.
3. Ерохин Д. О. Информационная среда как фактор формирования ценностей молодежи : автореф. дис. ... канд. психол. наук : 19.00.05 / М., 2011 - 28 с.
4. Каракозов С.Д. Информационная культура в контексте общей теории культуры личности //Педагогическая информатика. - 2000. - № 2. - С. 41-55.
5. Леонтьев Д.А. Психология смысла: природа, строение и динамика смысловой реальности. - М.: Смысл, 2003. — 487 с. - ISBN 5-89357-082-0
6. Писчиков В.С., Холин М.М. Духовный мир личности и виртуальная реальность Психологические проблемы смысла жизни и акме: Электронный сборник материалов XV симпозиума / Под ред. Г.А. Вайзер, Н.В.Кисельниковой. – М.: УРАО «Психологический институт», 2010. - С. 45-46.
7. Плешаков В.А. Теория киберсоциализации человека: монография / В. А. Плешаков, М.: Номо Cyberus, 2011 – 397 с.
8. Франкл В. Э. Человек в поисках смысла / Под общ. ред. Л. Я. Гозмана, Д.А. Леонтьева. – М.: Прогресс, 1990. – 366 с.
9. Шрейбер Т. В. Особенности саморегуляции старшеклассников с различным уровнем осмысленности жизни // Вестник Удмуртского университета. - 2005. - № 9. - С. 63-78.

**Семилуцкая Л.В.**

Муниципальное бюджетное общеобразовательное учреждение  
вечерняя (сменная) общеобразовательная школа №1, г. Воронеж, учитель  
информатики и ИКТ  
[clv08@inbox.ru](mailto:clv08@inbox.ru)

## ***Использование проектной деятельности на уроках информатики в вечерней школе***

Если человек в школе не научился  
творить, то и в жизни он всегда  
будет подражать, копировать, т.к.  
мало таких, которые бы  
научившись копировать, умели  
сделать самостоятельное  
приложение этих сведений.

Л.Н. Толстой

В докладе рассматривается метод введения творческих заданий, который представляет собой продукт проектной деятельности, метод обучения на основе реальных ситуаций и его использование в преподавании информатики и ИКТ в школе.

Успех в современном мире во многом определяется способностью человека организовать свою жизнь как проект: определить дальнюю и ближайшую перспективу, наметить план действий и, осуществив его, оценить, удалось ли достичь поставленных целей.

Одним из приоритетных направлений школы является информатизация. Темпы информатизации современного общества, все более широкое использование им в компьютерных и сетевых технологий.

Каковы особенности вечерней школы? Перед каждым учителем возникает ряд проблем, над решением которых ему приходится работать в сложных условиях (особенности контингента учащихся). Вот некоторые из них: Как обеспечить успешность обучения большинства учащихся? Как направить творческую энергию школьников на уроках в нужное русло, если учесть, что в 9-12-й класс приходят учащиеся из разных школ с разным уровнем подготовки (а порой и без знаний компьютера)? В этом есть особенность обучения информатики и ИКТ.

Зачастую в рамках урока не удастся осуществить работу, связанную с формированием ИКТ — компетенций в полной мере с теми учащимися, которые не имеют навыков работы с компьютером или теми, кто хочет углубить свои знания и навыки. Для развития познавательных интересов учеников при обучении я применила определенные методики. И в связи с этим я решила составить программу спецкурса по информатике и ИКТ. Так,

например, элективные курсы в 12 классе «Информационные технологии в бизнесе».

Введение творческих заданий в курс информатики помогают решить многие педагогические проблемы, которые не решаются привычными методами. Я воспитываю творческую личность, которая развивается в постоянно изменяющихся условиях, а творческий человек успешен в профессиональной деятельности.

Именно создание учащимися творческих учебных работ помогает достигнуть высокого результата на пути решения этой задачи. Чаще всего творческие работы представляют собой продукт проектной деятельности.

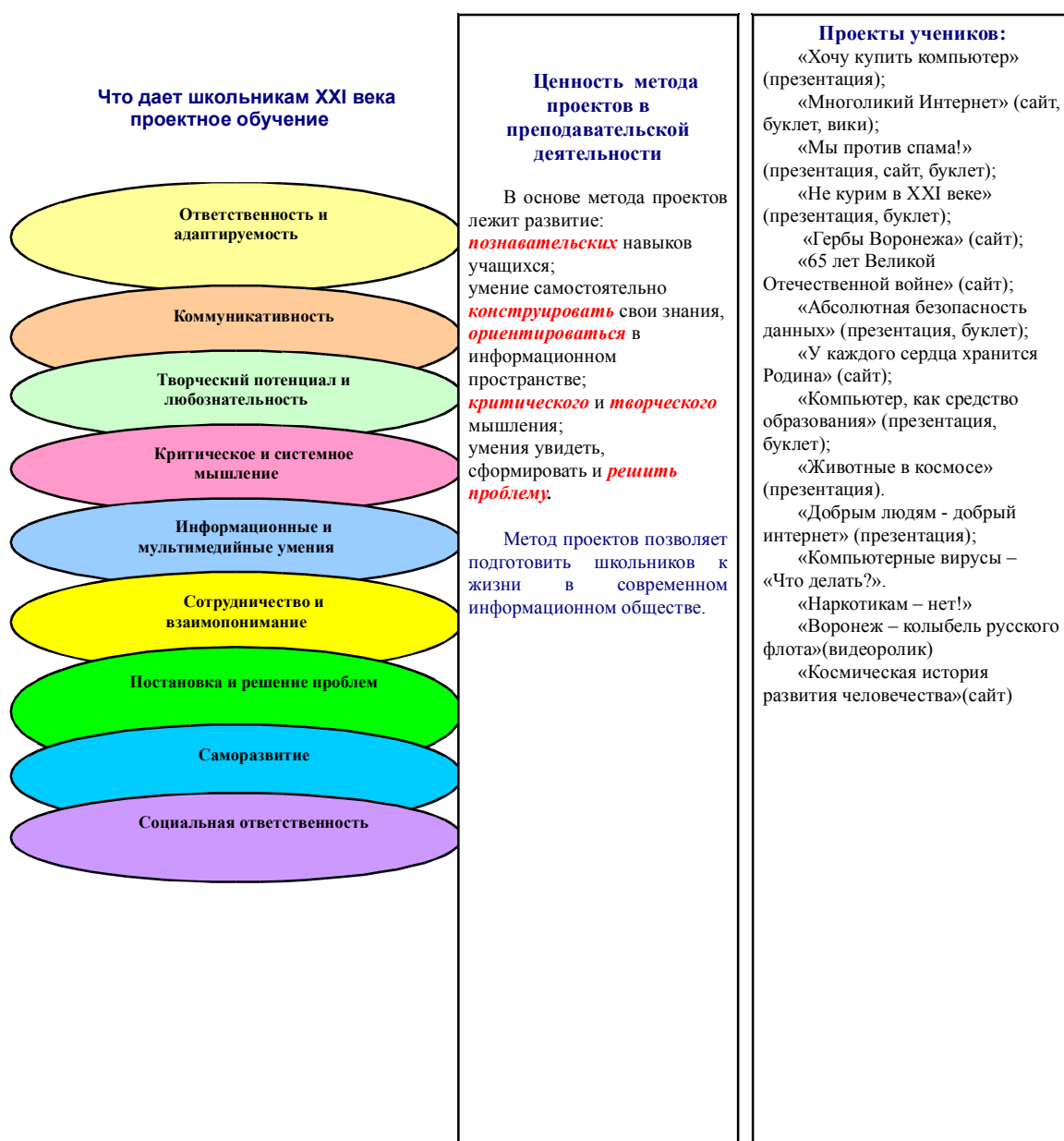
С понятием творчества неразрывно связано понятие интеллекта. Под ним понимается совокупность самых общих умственных способностей, обеспечивающих человеку успех в решении разнообразных задач.

Информационно культурный, грамотный человек должен уметь осознать, когда необходима информация, должен быть способен найти, оценить и эффективно использовать полученную информацию, уметь взаимодействовать с традиционными и автоматизированными средствами ее хранения.

На уроках информатики отдаю приоритет именно практической деятельности учащихся, через которую и происходит овладение ИКТ. Идеальным средством для решения этой задачи служит метод проектов. Проектная деятельность в зависимости от реализуемых образовательных задач может принимать всяческие формы: это может быть грандиозный проект на целый учебный год и более (он может быть и сквозным, то есть не прерывающим классические занятия), а может – проектом на один или несколько уроков (мини-проект). Так, например, проект «Многоликий Интернет», который включает в себя создание различных документов, использующих разные программы, рассчитан на несколько занятий. Участвуя с учениками 10-12 классов в самом интересном проекте «Партнерство и образование» мы получили грамоту от лаборатории Касперского, что дало толчок к дальнейшему участию в сетевых конкурсах: «Мой университет», «Мой класс – моей школе»: в дистанционном мероприятии - конкурс рисунков «Мир космоса», за который получили диплом; в областном конкурсе антинаркотической социальной рекламы «Сделай свой выбор» (награждены дипломом), в областном конкурсе «Вселенная. Земля. Космос», с презентацией «Тюльпаны Байконура» и сайтом «Космическая история развития человечества», а также в международном конкурсе видеороликов «Мой любимый край» - «Воронеж – колыбель русского флота», за который получили сертификат и свидетельство о публикации.

Участие в ученических проектах дает возможность каждому ученику раскрыть свой творческий потенциал, приобрести навыки научно-исследовательской работы, расширить свои знания по предмету, реализовать оригинальные идеи, научиться различным формам презентации проектно-

исследовательских работ. Любой проект связан с прогнозированием, а потому может служить эффективным инструментом развития интеллекта и креативности ученика в обучении и воспитании.



**Рис. 1.** Что дает школьникам XXI века проектное обучение.

В результате многолетней практики, я для себя решила, что необходимо активно использовать проектную деятельность учащихся на уроках. Я не утверждаю, что проектная работа поможет решению всех проблем в обучении, но это очень эффективное средство при изучении многих тем по информатике и ИКТ.

Человек от природы одарен способностью открытия нового.

Современному ученику важно привить умение самостоятельно приобретать знания. Умение работать с информацией, анализировать ее, видеть и решать возникающие в разных областях проблемы. Задачи интеллектуального и нравственного развития приобрели особую актуальность. Не просто усвоение знаний, а умение их творчески применять для получения нового знания, развитие самостоятельного критического мышления, аналитических навыков — вот новый ориентир образования, который воплотился в компетентностном подходе. Аналитическая деятельность становится важнейшей характеристикой современного общества.

Почему же метод проектов является эффективным средством развития аналитических умений учащихся? Потому что в основе метода лежит развитие познавательных навыков, умений самостоятельно приобретать свои знания, ориентироваться в информационном пространстве, развитие критического и творческого мышления, умение увидеть, сформулировать и решить проблему. Метод проектов позволяет организовать деятельность учеников таким образом, чтобы условия для развития аналитических умений были благоприятными.

Внедрение метода проектов в практику образования позволяет развивать аналитические умения обучающегося, к которым относятся следующие умения:

- производить диалектический анализ и синтез изучаемого материала;
- выделять главное и существенное,
- классифицировать материал;
- систематизировать, делать выводы и обобщения, резюме;
- аргументировать свои утверждения;
- проводить самоанализ, накапливать, отбирать и обрабатывать материал;
- оформлять свои наблюдения и выводы определенным образом
- выявлять основную задачу (проблему) и определять способы ее оптимального решения;
- пользоваться исследовательскими умениями.

Поэтому формами организации занятий по некоторым темам, на мой взгляд, и следует выбирать исследовательские проектные работы.

Для решения обозначенной проблемы применяю в работе метод обучения на основе реальных ситуаций. Преимуществом этого метода является возможность оптимально сочетать теорию и практику, что представляется достаточно важным при подготовке учеников.

В целях активизации использую такие приёмы:

- создать необычную атмосферу во время деятельности, сделать урок нестандартным;
- добавить в практическую деятельность соревновательный момент, тогда она обретёт хотя бы временную значимость;
- особым образом сформулировать цели деятельности

обучающегося, обозначив его личную заинтересованность.

Проект подразумевает исследовательскую деятельность. Так в 9 классе чтобы составить генеалогическое древо своей семьи необходимо общение с представителями старших поколений (бабушки, прабабушки), а итогом становится проект «Иерархическая модель. Генеалогическое древо семьи», выполненный средствами векторной графики в Word.

Важным шагом при работе над проектом является выбор темы. В старших классах (10 -12 классы) проектная деятельность продолжается по следующим направлениям: «Семейный бюджет»; «Бизнес-план компьютерной фирмы», «Домашняя видеотека» и др.

При работе над проектом «Хочу купить компьютер» учащиеся должны обосновать выбор устройств, входивших в системный блок, критерием будет являться быстродействие и стоимость системного блока. Используя прайс-листы различных компьютерных фирм, должны обосновать стоимость системного блока данной компьютерной фирмы. При защите учащиеся рассказывают о причинах актуальности выбранной темы, доказывают правильность выбор системного блока, при этом высказывают рекомендации своих одноклассников.

Свои творческие работы учащиеся представляют в разной форме, в зависимости от целей и содержания. Выполненные работы рецензируются учителями - предметниками, потом оцениваются на различных уровнях. Учащиеся выступают со своими исследованиями на уроках, классных часах.

Для активизации познавательного интереса обучающихся к другим областям знаний на уроках информатики помогают и межпредметные связи: решение проблем других областей средствами информационных технологий (например, построение графиков и диаграмм в MS Excel, используя математические формулы; создание рисунков в графическом редакторе, посвященных 50-летию полета Гагарина в космос «Вселенная, земля, космос», создание плакатов «Через тернии - в космос», «Безопасный Интернет», стенда для кабинета «Галерея информатиков»; создание буклетов для проектов: «Сделай свой выбор», «Многоликий Интернет», «Здоровый образ жизни»; создание плакатов для проекта «Мы за чистый компьютер», «Мы против спама», создание видеоролика «Воронеж – колыбель русского флота».

Хочется лишь отметить, что связь между предметами всегда можно найти, даже если она не явно видна. Понимая и осознавая необходимость интеграции, учителя нашей школы: информатики, математики, биологии, химии, ЗОЖ, краеведения в рамках проектной деятельности разрабатывают и проводят интегрированные уроки. Тесное сотрудничество учителей-предметников делает процесс обучения более качественным и интересным.

Опыт работы заключается в создании собственной системы уроков, разработке системы их проведения, разработке дидактических, контролирующих материалов (в том числе и электронных) и методики их

применения в учебном процессе. Все полученные навыки учащиеся применяют для создания проектов на различные темы по информатике, а также и по другим предметам с использованием исследовательской деятельности.

Собственные наблюдения показали, что:

1. В целом проектная методика является эффективной инновационной технологией, которая значительно повышает уровень компьютерной грамотности, внутреннюю мотивацию учащихся, уровень самостоятельности учеников, их толерантность и общее интеллектуальное развитие;
2. В ходе проектной деятельности развиваются основные компетенции:
  - информационные;
  - коммуникативные;
  - учебно-познавательные.
3. Развиваются творческие способности учащихся и возможность для самовыражения.
4. Развивается самостоятельность добывать знания и пользоваться ими для решения новых познавательных и практических задач.

Подводя итог, можно сказать, что применение идей метода проектов вполне оправдано. Собственные наблюдения показали, что в целом проектная методика является эффективной инновационной технологией, которая значительно повышает уровень компьютерной грамотности, уровень самостоятельности учеников, их толерантность и общее интеллектуальное развитие.

Только творчески работающий учитель может быть образцом и стимулом для творческого развития своих учеников. Поэтому надо быть учителем, которого ждут наши ученики.

### **Литература**

1. Леонтович А. Учебно-исследовательская деятельность школьников как модель педагогической технологии // Народное образование. 2002.
2. Сергеев И.С. Как организовать проектную деятельность учащихся. М., 2005.
3. Полат Е.С. Новые педагогические и информационные технологии в системе образования. М.: Академия, 2007.
4. Всероссийский съезд учителей информатики 24 марта 2011 года <http://it.teacher.msu.ru/plenary/sadovnichiy>

**Фаттахова Г.А.**

МБОУ «Мари-Турекская средняя общеобразовательная школа»,  
учитель; аспирантка Поволжского государственного  
технологического университета  
[galina-himik@mail.ru](mailto:galina-himik@mail.ru)

### ***Электронный образовательный ресурс «Школа юного лесовода» как основной инструмент обучения членов школьного лесничества***

Для реализации важнейших целей современного дополнительного образования актуальным является поиск и разработка эффективных образовательных технологий и технологических средств, способствующих развитию познавательных навыков, умения самостоятельно конструировать свои знания.

Использование электронных образовательных ресурсов в системе дополнительного образования позволило бы:

- организовать разнообразные формы деятельности обучаемых по самостоятельному извлечению и представлению знаний;
- применять весь спектр возможностей современных информационных и телекоммуникационных технологий в процессе выполнения разнообразных видов учебной деятельности;
- диагностировать и оценивать интеллектуальные возможности обучаемых, уровень подготовки к занятию, соизмерять результаты усвоения материала в соответствии с требованиями государственного образовательного стандарта по внеурочной деятельности;
- создавать условия для осуществления индивидуальной самостоятельной учебной деятельности обучаемых, формировать навыки самообучения, саморазвития, самосовершенствования, самообразования, самореализации;
- обеспечить педагогов, обучаемых и родителей актуальной информацией, соответствующей целям и содержанию образования;
- создать основу для общения педагогов, обучаемых и родителей, нацеленного на повышение эффективности обучения [1].

Цель создания электронного пособия «Школа юного лесовода» – помочь педагогам, включившимся в процесс информатизации образования для повышения эффективности образовательного процесса, а также сократить время на подготовку к занятиям, используя предложенный цифровой контент.

Обучение с использованием данного электронного образовательного ресурса позволит обеспечить формирование базовых компетентностей в



условиях перехода к новым образовательным стандартам через систему школьных лесничеств.

Ресурсы представлены по темам, изучаемым по программе экологической школы по направлению «Юные лесоводы» для обучающихся 8-10 классов общеобразовательной школы в рамках внеурочной деятельности или в системе дополнительного образования. Примерное тематическое планирование имеет рекомендательный характер. При его составлении исходили из принципа избыточности изучаемых тем, что позволит использовать данные материалы при работе с другими программами экологического направления.

### ***Современные средства обучения в системе дополнительного образования***

Для достижения образовательных результатов, отвечающих новым запросам общества, в школе и в системе дополнительного образования необходимо использовать современные средства обучения, которые должны обеспечивать выполнение требований федерального государственного образовательного стандарта (ФГОС).

Достижение личностных, метапредметных и предметных результатов освоения образовательной программы дополнительного образования невозможно без комплексного использования в образовательном процессе всей совокупности существующих средств обучения – как традиционных, так и средств обучения, функционирующих на базе компьютерных технологий [2].

Ключевой характеристикой образования становится формирование общего деятельностного базиса как системы универсальных учебных действий, определяющих способность личности учиться, познавать, сотрудничать в познании и преобразовании окружающего мира [3].

Возможности использования достижений новых технологий (свободный доступ к разнообразным информационным ресурсам, дистанционность, мобильность, интерактивность, возможность формирования социальных образовательных сетей и образовательных сообществ, возможность моделирования и анимирования различных процессов и явлений и пр.) определяют новый подход к оснащению образовательного процесса средствами обучения.

Использование в образовательном процессе интерактивных средств обучения позволяет обучающимся проявить самостоятельность при изучении нового материала, в работе с текстом, раскрывающим основное содержание предмета, оценить свой уровень подготовки по конкретной теме на данный момент времени.

Использование средств мультимедиапроекции позволяет проиллюстрировать изучаемый материал, провести автоматизированный контроль знаний по определенной теме или по курсу в целом, применить особые формы подачи информации, доступной данному ученику, группе обучающихся, выстроить индивидуальную траекторию обучения.

Использование конкретной технологии проведения занятия по электронному пособию учитель может осуществлять с учетом индивидуальных возможностей обучающихся, в том числе и в плане владения информационно-коммуникационными технологиями, а также наличия различного рода интерактивного оборудования.

Программное обеспечение «Школы юного лесовода» позволяет педагогу экономить время при подготовке учебного материала, записывать его в память компьютера и в классе использовать материал, сопровождая показ только устными объяснениями и комментариями, что поднимает эффективность обучения на новый уровень.

Наглядная форма подачи материала, высокое качество изображения на интерактивной доске, удобная система навигации, постоянный доступ ко всем разделам способствуют лучшему усвоению материала и делают занятие интересным и неустойчивым. Информативность и красочность представленного визуального материала облегчают восприятие и создают дополнительный стимул к изучению основ лесоведения и охраны окружающей среды.

Для контроля за качеством обучения и самоконтроля применяется компьютерное тестирование, которое представлено «классической» схемой тестирования с вопросами и несколькими вариантами ответов, что позволяет значительно сократить временные рамки, необходимые на обработку результатов проверки знаний и свести к минимуму субъектное влияние педагога на результат измерения.

Изучение каждой темы по программе направления «Юные лесоводы» можно заканчивать контрольным тестированием, которое позволяет ученику выяснить, насколько глубоко он усвоил учебный материал. Таким образом, промежуточное тестирование фиксирует переход от одной темы к другой. В результате осуществляется постоянная обратная связь обучаемого с педагогом, позволяющая повысить эффективность процесса усвоения знаний.

Таким образом, организацию учебного процесса с использованием электронных образовательных ресурсов можно производить в классе и индивидуально в виде самообразования школьников.

При изучении программы дополнительного образования электронные образовательные ресурсы можно применять как при изложении нового материала – визуализация знаний, так и осуществляя контроль и проверку.

Исследования, проведенные психологом школы, показали, что в ходе мультимедийного занятия объем поданного, и соответственно, усвоенного материала можно увеличивать без риска перегрузить учеников. Информация, продублированная через различные сенсорные пути, через текст, видео, графику и звук, усваивается лучше и сохраняется гораздо дольше.

***Структура электронного пособия «Школа юного лесовода»***

Электронное пособие состоит из 8 разделов, названия которых расположены с левой стороны слайда.

Названия разделов:

- главная страница,
- программа,
- теория,
- практикум,
- контроль,
- глоссарий,
- литература,
- об авторе.

Переход с одного раздела на другой осуществляется с помощью гиперссылок, настроенных на каждой странице для быстрого доступа к необходимой информации.

На главной странице пособия обозначены ближние и дальние цели, способствующие мотивации обучающихся к изучению материала, показывающие значимость экологических знаний и сформированности основ экологической культуры.

В разделе «Программа» прописаны основные структурные дефиниции, такие как пояснительная записка, цели и задачи программы, условия ее реализации, ожидаемые результаты, тематический план, краткое содержание, список литературы вынесен для удобства пользователей в отдельный раздел.

Раздел «Теория» состоит из 8 слайдов в соответствии с блоками учебной программы.



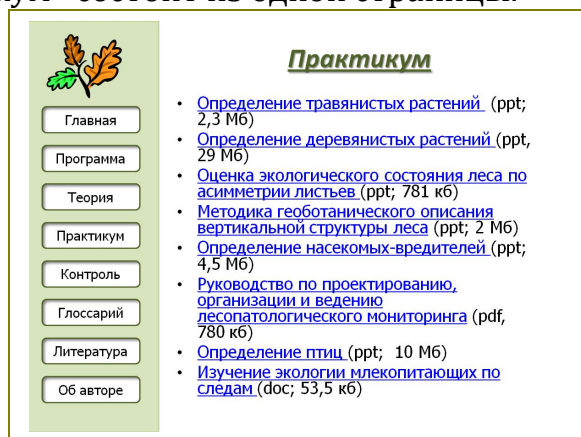
**Рис.1.** Первая страница раздела «Теория»

На каждой странице имеется заголовок темы раздела, например «Лесовосстановление и лесоразведение», далее идет маркированный список тем, рассматриваемых на странице, например «Заготовка и хранение семян». После названия теоретического вопроса в скобках указаны расширение документа .doc, .ppt или .pdf, а так же размер. На каждом слайде раздела расположена управляющая кнопка – возврат на первый слайд

«Теория».

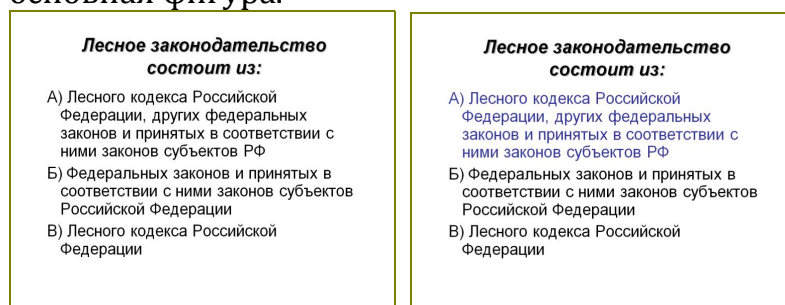


**Рис.2.**Пятая страница раздела «Теория»  
Раздел «Практикум» состоит из одной страницы.



**Рис.3.** Страница раздела «Практикум»

Раздел «Контроль» составлен по всем теоретическим разделам курса и состоит из двух типов заданий: тестовые и контрольные вопросы. Тестовые задания выполнены в виде презентаций в программе МО PowerPoint 2003. На каждом слайде по одному вопросу и три ответа, правильный из которых только один, при наведении курсора на правильный ответ он выделяется другим цветом, меняется фон фигуры или появляется основная фигура.



**Рис. 4.** Вопрос-ответ

Контрольные вопросы выполнены в программе MO Word 2003. В вопросниках не содержатся ответы, но их можно найти в теоретическом материале пособия, глоссарии или воспользовавшись методическими пособиями из списка литературы, предложенном в ЭОР «Школа юного лесовода».



**Рис.5.** Страница раздела «Контроль»

Раздел «Глоссарий» состоит из нескольких страниц. На первой странице размещена таблица, в которой обозначены литеры, при нажатии на которые открывается страница с терминами на эту группу. На каждой странице терминологического словаря внизу расположены управляющие кнопки, которые позволяют вернуться на первую страницу глоссария либо перейти на следующую.

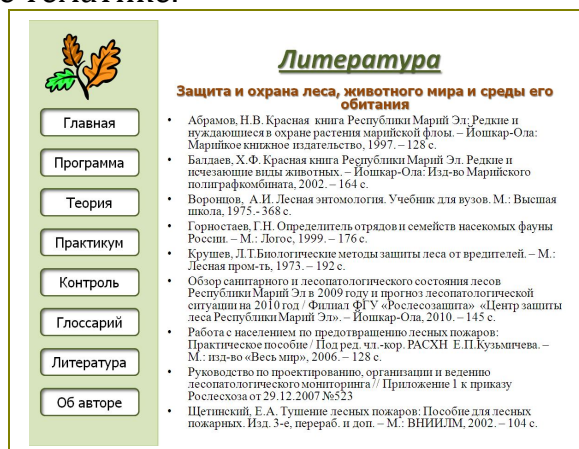
Для того, чтобы найти интересующее определение необходим свободный доступ к сети Интернет, так как глоссарий настроен на формулировки терминов, размещенных на сайте Федерального агентства лесного хозяйства по адресу <http://www.rosleshoz.gov.ru/terminology/>.



**Рис.6.** Первая страница раздела «Глоссарий»

Раздел «Литература» включает подборку литературных источников по каждой теме раздела «Теория». К странице списка литературы по теме можно перейти из раздела «Теория», нажав на кнопку «Литература» на

соответствующем слайде, например страница теоретической темы «Защита и охрана леса, животного мира и среды его обитания» и страница списка литературы по этой же тематике.



**Рис.8.** Страница рекомендуемого списка литературы по теме «Защита и охрана леса, животного мира и среды его обитания»

На последней странице содержатся основные сведения об авторе образовательного ресурса: образование, место работы, значимые достижения, публикации.

### **Краткое содержание электронного пособия «Школа юного лесовода»**

В разделе «Программа» представлена рабочая программа, по которой осуществляется теоретическая и практическая подготовка юных лесоводов в школьном лесничестве.

Раздел «Теория» состоит из семи тем, каждая из которых разбита на изучаемые вопросы с ответами.

Содержание темы «Охрана природы»:

- Научные основы охраны окружающей среды;
- Охрана атмосферного воздуха;
- Охрана водных ресурсов;
- Охрана почв и недр;
- Охрана растительного и животного мира;
- Экологические основы ресурсосберегающих технологий;
- Правовые основы охраны природы;
- Охрана окружающей среды на объектах работ».

При изучении темы «Лесоводство» в электронном пособии можно найти материал на следующие вопросы:

- Основы ботаники;
- Лесные почвы;
- Дендрология;
- Понятие о лесе;
- Лесное законодательство Российской Федерации;
- Рубки лесных насаждений;

- Содействие лесному возобновлению;
- Технология заготовки древесины;
- Транспорт древесины.

Содержание темы «Основы лесной таксации»:

- Таксация ствола растущего и срубленного дерева;
- Таксация насаждения;
- Краткие сведения о лесном фонде;
- Подготовка лесосечного фонда;
- Основные понятия о лесоустройстве;
- Пороки древесины;
- Классификация и стандартизация лесных материалов;
- Обмер и учет лесоматериалов.

В теме «Защита и охрана леса, животного мира и среды его обитания» рассматриваются следующие вопросы:

- Основы лесной энтомологии;
- Основы лесной фитопатологии;
- Основы лесозащиты;
- Лесные звери и птицы;
- Охрана среды обитания;
- Основы лесной пирологии;
- Охрана лесов от лесонарушений.

Тема «Лесовосстановление и лесоразведение» состоит из трех блоков, которые в свою очередь представлены теоретическими вопросами.

Первый блок «Основы лесосеменного дела»:

- Заготовка и хранение семян;
- Показатели и методы оценки качества семян;
- Способы подготовки семян к посеву.

Второй блок «Лесные питомники»:

- Системы обработки почвы;
- Применение удобрений и гербицидов;
- Технология выращивания посадочного материала.

Третий блок «Производство лесных культур»:

- Посадка лесных культур;
- Оценка качества лесокультурных работ и лесных культур.

Тема «Механизация лесохозяйственных работ» представлена презентацией.

В теме «Использование лесов» рассматриваются вопросы о видах пользования, таких как заготовка древесины, заготовка живицы, заготовка и сбор недревесных лесных ресурсов, заготовка пищевых лесных ресурсов и сбор лекарственных растений, рекреационная деятельность.

Тема «Охрана труда» содержит необходимую информацию о Федеральном законе «Об основах охраны труда в РФ», трудовом кодексе, производственной санитарии и гигиене труда, производственном

травматизме и его профилактике и иллюстративный материал по оказанию первой медицинской помощи.

По материалам раздела «Практикум» можно познакомиться с методиками проведения исследований в лесу по темам:

- Определение травянистых растений;
- Определение деревянистых растений;
- Оценка экологического состояния леса по асимметрии листьев;
- Методика геоботанического описания вертикальной структуры леса;
- Определение насекомых-вредителей;
- Руководство по проектированию, организации и ведению лесопатологического мониторинга;
- Определение птиц;
- Изучение экологии млекопитающих по следам.

Предложенные методические рекомендации могут применяться как на практических занятиях после изучения темы, так и во время летней практики. Например, после изучения темы «Таксация леса» можно провести практикум по одному из блоков исследования (определение диаметра ствола, полноты и т.д.) по методике геоботанического описания вертикальной структуры леса.

Раздел «Контроль» представлен тестовыми заданиями. По каждой теме предлагается 10 вопросов с ответами.

Количество контрольных вопросов по темам разное и не включает готовые ответы, их необходимо найти самостоятельно пользуясь материалом ЭОР «Школа юного лесовода». Вопросы составлены по темам лесное законодательство, лесной фонд, рубки леса, лесная таксация, лесовосстановление, селекция и лесное семеноводство, охрана леса, болезни и вредители деревьев и животный мир.

В разделе «Глоссарий» рассмотрены основные понятия и термины, знание которых пригодится не только юным лесоводам, но и профессионалам, вся информация структурирована по буквам алфавита.

В разделе «Литература» приведены тематические списки литературных источников, доступных как в библиотеках, так и в электронных библиотеках сети Интернет.

Используя в работе электронное пособие, каждый педагог легко сможет выстроить логическую схему проведения занятия. Например, тема «Основы лесной энтомологии» → тест «Вредители леса» → практикум «Определение насекомых-вредителей» → практикум «Лесопатологический мониторинг» → контрольные вопросы «Болезни и вредители леса» → литература из раздела «Защита и охрана леса, животного мира и среды его обитания».

### **Выводы**

5.Электронный образовательный ресурс «Школа юного лесовода» облегчает понимание изучаемого материала за счет различных способов



подачи материала и допускает адаптацию в соответствии с потребностями, уровнем подготовки, интеллектуальными возможностями обучающегося;

6.Использование ЭОР на занятиях позволяет педагогу проводить занятие в форме самостоятельной работы за компьютерами, индивидуализировать работу с обучающимися, с помощью компьютера быстро и эффективно контролировать знания;

7.Информационные технологии значительно расширяют возможности предъявления учебной информации и способствуют повышению мотивации обучающихся к обучению.

Занимаясь проблемой организации и успешного функционирования школьного лесничества с 2006 года в системе дополнительного образования, считаю, что данное направление перспективно и заслуживает особое внимание среди педагогов дополнительного образования и учителей занятых внеурочной деятельностью в рамках реализации федерального государственного образовательного стандарта второго поколения.

Изучение теоретического материала с использованием ЭОР, с последующим закреплением на практических занятиях дает высокий уровень усвоения знаний, возможности их применения в повседневной жизни, а значит, способствует формированию умения учиться.

Использование электронного пособия «Школа юного лесовода» позволяет успешно реализовать подготовку обучающихся не одного кружка или объединения, а охватить желающих со всего муниципального образования путем дистанционного обучения и тем самым повысить их экологическую культуру и результаты участия в конкурсах различного уровня.

Еще одним преимуществом использования электронного образовательного ресурса в подготовке юных лесоводов является то, что оно динамично и всегда может быть изменено с учетом всех пожеланий обучающихся по данной программе и их кураторов, а также сориентировано на уровень общей подготовки ученика.

### **Литература**

1. Обзор цифровых образовательных ресурсов [Электронный ресурс]. Режим доступа: <http://metod.magint.edu.ru/LEVEL1/theory/cor.aspx>.
2. Федеральный государственный образовательный стандарт [Электронный ресурс]. Режим доступа: [www.edu.ru](http://www.edu.ru).
3. Формирование универсальных учебных действий в основной школе: от действия к мысли. Система заданий : пособие для учителя / Под ред. А.Г. Асмолова. - М.: Просвещение, 2010. - 159 с.

**Явич М.П.**

старший преподаватель, Грузинский технический университет,  
департамент информатики и систем управления  
[webmax2006@yahoo.com](mailto:webmax2006@yahoo.com)

## ***Теоретические основы информатики. Концепции создания электронных обучающих средств и интеллектуальных компьютерных игр для детей***

### **Аннотация**

*Предлагаются концепции создания электронных обучающих средств и интеллектуальных компьютерных игр для детей. Концепции разработаны для детей, у которых родной язык грузинский, но могут быть использованы и в более широком масштабе. При создании электронных обучающих средств необходимо учитывать специфические национальные особенности, традиции, лингвистические и языковые системы. В основе эргономики должно быть наличие дружественного интерфейса, отсутствие элементов отрицательного воздействия на психику, спокойные цветовые гаммы, отсутствие агрессивности в используемом материале, непонятности в алфавитных и цифровых обозначениях, четкая постановка вопросов в контролирующих программах.*

Разрабатываемые электронные обучающие средства должны быть ориентированы на тот контингент, на ту аудиторию, которой они предназначаются. На сегодняшний день существующие концепции разработки электронных обучающих средств ориентированы в основном на взрослого человека, на помощь ему в повышении квалификации, получении второго образования. Но следует учитывать, что психология мышления взрослого человека, его способность к восприятию резко отличается от психологических особенностей ребенка. Для создания дружеского интерфейса в пособиях для детей следует учитывать, как возрастные психологические особенности детского восприятия, так и национальные, их традиции, лингвистическую и языковую специфичность окружающей ребенка среды.

Рассмотрим психологические особенности ребенка в зависимости от возраста. К трем годам ребенок достигает определенного уровня социальной компетентности: он проявляет интерес к другому человеку, испытывает доверие к нему, стремится к общению и взаимодействию со взрослыми и сверстниками. Интеллектуальная компетентность выражается прежде всего в том, что ребенок активно интересуется окружающим его миром, бытовыми предметами, игрушками, уже распознает цвет, форму, величину, яркость и т.п., владеет в определенной мере родным языком, способен выражать радость, огорчение, страх,

удивление, удовольствие. Он уже достаточно самостоятелен и инициативен, проявляются начала креативности. Именно в этом возрасте наблюдается первичный, еще не осознанный интерес к компьютеру. Следующий этап в развитии сознания наступает перед началом дошкольного возраста. В младшем дошкольном возрасте (к пяти годам) достаточно четко проявляются черты компетентности, в особенности, интеллектуальной, когнитивности и креативности. Дети более сознательно играют в компьютерные игры, учатся самостоятельно включать и выключать компьютер. Интеллектуальная компетентность связана с достаточно высокой мыслительной активностью. Ребенок уже практически полностью владеет родным языком, ему понятны определенные нюансы, рифмы, словосочетания. Эмоциональность отличается большим многообразием способов выражения своих чувств: радости, грусти, огорчения, удовольствия. Свобода поведения пятилетнего ребенка выражается в стремлении совершать независимые поступки - подбирать для себя определенные компьютерные игры, пытаться проводить их поиск в Интернете. К семи годам существенно повышается уровень произвольности и свободы поведения, что связано с возросшими умственными и интеллектуальными возможностями ребенка, уверенностью в своих силах. Появляется в большей мере заинтересованность, он уже способен понимать разный характер компьютерных игр и сайтов, более четко подбирает игры. В случае, если уже умеет читать, пытается проводить поиск определенных сайтов в Интернете.

Мышление младшего школьника в достаточной мере отличается от мышления дошкольника. Это проявляется в более высоком темпе развития, а также в качественных и структурных изменениях, происходящих в интеллектуальных процессах мышления во время обучения. Следует учитывать тот факт, что у всех детей постепенно развиваются три вида мышления: наглядно-действенное, наглядно-образное, словесно-логическое [1], причем в различном темпе с учетом индивидуальных особенностей ребенка. Последний вид мышления в дошкольный период жизни ребенка относительно слабо развит, но уже в младшем школьном возрасте ситуация меняется, и при нормальном развитии детей третий вид мышления приближается к аналогичному виду мышления взрослого человека. В младшем школьном возрасте у ребенка ярко проявляется уверенность в себе, самостоятельность, чувство собственного достоинства, инициативность, интеллектуальная компетентность, достаточно высокая степень когнитивности и креативности. Ребенок свободно владеет родным языком, понимает его лингвистические особенности, национальные традиции. Ребенок этого возраста уже отличается большим богатством и глубиной переживаний, разнообразием их проявлений и в то же время большей сдержанностью эмоций. Желание к компьютерной деятельности вполне осознано. Ребенок уже способен к работе с компьютером, к

освоению определенных программ и основ программирования компьютерной графики. Его решения могут быть оригинальными, вариантными. Однако появляется желание длительного нахождения за компьютером, что нецелесообразно, необходимо ограничение этого времени. Приводимые данные свидетельствуют о необходимости разработок концепций создания электронных обучающих средств (ЭОС) для детей, ориентируясь на возраст и национальность ребенка. Приводимые ниже концепции использованы при разработке ряда электронных пособий для детей, родным языком которых является грузинский. Целью разработки концепции создания электронных обучающих средства (ЭОС) для школьников младших классов является повышение уровня знаний и интеллекта школьника благодаря появлению условий для самостоятельного изучения учебного материала в удобном для него месте и времени; более глубокой индивидуализации обучения и обеспечения условий для его вариантности; развитие как образного, так и логического мышления; возможности использования мультимедийных информационных материалов; возможности автоматической генерации большого количества различных заданий и контроля и более объективной оценки знаний, благодаря автоматизированному контролю; возможности обучения поиску информации в различных поисковых системах; возможности точного определения индивидуальных особенностей каждого учащегося, благодаря вариативности материала и обратной связи; возможности использования в учебном процессе технологий мультимедиа и гипертекстовых систем. ЭОС должны отвечать следующим дидактическим требованиям: научность, доступность, наглядность, системность и последовательность в обучении, возможность заинтересовать ребенка, высокое качество в освоении знаний, единство образовательных, воспитательных и развивающих функций обучения, возможность индивидуального обучения и высокая адаптивность в обучении в зависимости от развития ребенка, обеспечение целостности и непрерывности дидактического цикла обучения, максимальное использование компьютерного учебного материала при его сравнительно небольшом объеме, учитывая особенности возраста.

ЭОС должны быть ориентированы на психологические особенности детей младшего школьного возраста [1], с учетом специфических национальных особенностей, традиций, лингвистических и языковых систем и соответствовать эстетическим, эргономическим и технико-экономическим требованиям. ЭОС – средство дополнительного образования в виде электронного издания для локального использования, выпускаемое в ограниченном тираже в виде идентичных экземпляров и не является электронным аналогом соответствующих традиционных учебников. ЭОС разрабатывается на основе средств программирования, гипертекстовых технологий, мультимедийных технологий и может сочетать в себе все вышеперечисленные средства в самых различных

соотношениях. ЭОС, разработанная на основе интеллектуальных технологий, способна к определению ошибочных, либо недостаточных знаний школьника, позволяет обучать его на высоком уровне, контролировать его знания и в случае необходимости давать ему дополнительные объяснения соответствующего раздела (которые школьник недостаточно усвоил), адаптировать процесс обучения, исходя из индивидуальных особенностей с учетом умственного развития и интеллектуальных особенностей каждого ребенка. В этом случае ребенок не только получает знания и контролируется их усвоение, но и имеется возможность дать ему (в случае недостаточного уровня усвоения) дополнительные пояснения, определить пробелы в обучении. Материал, заложенный в ЭОС, должен быть систематизирован и структурирован, учитывать межпредметные особенности изучаемого в школе материала. Последовательность подачи должно быть аргументировано и логично, способствовать прочности усвоения как школьных знаний, так и расширения интеллектуального кругозора ребенка. При разработке ЭОС наиболее целесообразно использование программных средств Microsoft Word и Microsoft PowerPoint основного пакета MS Office.

Учитывая, что обучаемый — это ребенок 7-11 лет с еще формирующейся психикой, из семей с различным уровнем образованности, с разными генетически заложенными умственными способностями и интеллектом, ЭОС должны способствовать постепенному, но постоянному развитию детей. Материал, заложенный в ЭОС является средством дополнительного образования и не должен противоречить основным целям обучения ребенка, использование ЭОС должно быть сугубо индивидуальным и рассчитано на детей с различным уровнем развития. Его использование не должно вызвать у ребенка чувства противоречия. Внедрение ЭОС должно быть ступенчатым, деликатным при наличии высококвалифицированного педагога, который способен правильно преподнести необходимость использования ЭОС, заинтересовав большинство учеников. Для детей этого возраста педагог фактически должен быть не только учителем, но и психологом, который может быстро и адекватно отреагировать на любые проблемы ребенка в процессе обучения [2].

Соблюдение дидактических принципов применимости материала [3], его доступности для детей разного уровня, новизны, пригодности для ребенка способствует более успешному развитию. При создании ЭОС необходимо учитывать ряд специфических требований: адаптивность курса, интерактивность самого обучения – интерактивный диалог на родном языке, четкий контроль знаний, реализация возможностей компьютерной визуализации учебной информации, как средства повышения интеллектуального уровня ребенка в процессе обучения. Разрабатываемые ЭОС должны обладать когнитивностью [4]. Использование в процессе обучения контролирующих систем — это

наиболее эффективный путь к повышению уровня восприятия и переработки информации и, как следствие, эффективности обучения. В случае использования компьютерных технологий повышается достоверность оценки полученных знаний, определяется действительный уровень школьника, независимо от случайных факторов и субъективных позиции учителя. Необходимо создание креативности учебного процесса.

При использовании ЭОС основной упор делается на творческую работу ребенка, возможно с использованием методик эвристического образования. Трудности с проблемой интерфейса креативной среды не должны стать барьером, возникающим перед ребенком, наоборот, в случае наличия у педагога высокого уровня знаний, креативная среда может явиться одним из решающих факторов в развитии умственных способностей и интеллекта ученика. Для этого необходима талантливая система тестирования – подбор вопросов, их формулировок и интерпретации ответов, что позволяет получить наиболее полную и достоверную оценку знаний ребенка. Использование в процессе обучения младшего школьника ЭОС является основным средством для облегчения усвоения и понимания наиболее существенных понятий, утверждений и примеров, более глубокого понимания, как программного учебного материала, так и освоенного в результате применения компьютерных технологий, используя генетически заложенные умственные и интеллектуальные способности ребенка и его развитие.

ЭОС должны соответствовать существующим методологическим требованиям [5,6]. В основе эргономики должно быть наличие дружелюбного интерфейса, отсутствие элементов отрицательного воздействия на психику (цветовые гаммы, агрессивность текста, непонятность алфавитных и цифровых обозначений). При создании ЭОС необходим учет психологических и санитарно-гигиенических требований при работе с компьютерами для детей данного возраста. При соблюдении всех принципов, заложенных в вышеизложенную концепцию, ЭОС становится мощным средством интеллектуального уровня ребенка.

На личность сегодняшнего дошкольника оказывает значительное влияние бурное развитие ИКТ. Уже в возрасте 3 лет ребенок начинает интересоваться компьютерными устройствами, которые он видит дома. Процесс развития ребенка можно определить как две стадии социализации: первичную, характерную для раннего детства, и вторичную – стадию формирования процессов саморегуляции [7]. Уже в возрасте 4-5 лет дошкольник понимает, что, используя компьютер, он может стать участником игрового процесса. Это приводит к изменению его желаний и, как следствие, метода воспитания. Учитывая, что воспитание - это целенаправленный процесс воздействия на личность ребенка, который является в данном случае объектом, необходим достаточно продуманный подход со стороны родителей и педагогов в выборе компьютерных игр, оценивая их положительные и отрицательные стороны [8,9]. Большинство

из находящихся в коммерческой продаже игр не обладают дружественным интерфейсом, имеют ярко выраженную агрессивность в содержании. С одной стороны, это приводит к появлению у ребенка агрессивности, враждебности, что может проявляться в его поведении. С другой стороны, в компьютерной игре он может выражать реакцию на уже имеющиеся у него агрессивные импульсы, чувство гнева, злость – все, что не одобряется его окружением. Игра может стать формой эскапизма с появлением компьютерной зависимости, способом ухода от действительности, беспомощностью ребенка. К сожалению, подобного типа игры наиболее рейтинговые [10].

Все вышесказанное говорит о необходимости создания для дошкольника другого типа игр с высоким интеллектуальным уровнем (ИИ) [11], которые позволят расширить возможности ИКТ в воспитании ребенка; повысить его интеллектуальный уровень на основе ИКТ и компьютерных технологий; развить у него логическое мышление; создать условия для самостоятельной творческой работы (желательно под надзором взрослых); получить интеллектуальную мультимедийную информацию о различных странах, обычаях народов, явлениях природы и т.д.; дать ребенку информацию и возможность услышать в игровой форме различные языки народов мира; проводить с ребенком занятия по родному языку с указанием конкретных лингвистических ошибок; проводить занятия по иностранным языкам основываясь на выборе родителей в зависимости от его национальности и места проживания; в более старшем дошкольном возрасте начинать обучение его алфавиту, чтению, цифрам, простейшим арифметическим операциям; дать ему возможность более глубокого изучения компьютерных технологий, научить использовать простейшие компьютерные программы [12]. При отборе и подготовке подобных ИИ следует учитывать дидактические критерии сложности и трудоемкости. Компьютерные дидактические интеллектуальные игры являются одним из видов компьютерных обучающих программ, в которых должны сочетаться учебные и игровые цели. Необходимо соблюдать и другие дидактические критерии, такие, как доступность и годность выбранной формы материала, научность, наглядность, последовательность обучения. Ребенка нужно заинтересовать, необходимо единство воспитательных и развивающих функций обучения, структурно-функциональной связываемости, целостности и непрерывности дидактического цикла обучения, небольшой объем предлагаемого материала. Основной принцип при разработке - глубокая проработка материала, представление его в таком виде и с таким интерфейсом, которые позволили бы ребенку наиболее эффективно использовать ИИ. Особенное внимание следует уделять креативности игрового материала. Необходимо не нарушать желание ребенка к творчеству, к новаторству, к поиску своего «Я». Креативность среды - один из основных факторов, способствующих умственному развитию ребенка, наряду с когнитивностью. Необходимо наличие программ, которые

позволят ребенку контролировать свои знания и умения. Эти системы должны обладать особым уровнем интеллекта, чтобы не оттолкнуть ребенка от желания их использовать.

ИИ должны быть ориентированы на психологические особенности ребенка дошкольного возраста с учетом специфических национальных особенностей, традиций, лингвистических и языковых систем.

В определенной мере в ИИ должно присутствовать тестирование, причем на интеллектуальном уровне. Это позволит приучить ребенка как к использованию данного метода, что будет необходимо ему в более старшем возрасте, так и станет средством повышения его умственных способностей, образованности. Тесты должны быть простыми, хорошо составленными, понятными ребенку, учитывать национальные традиции и окружение. Наряду с традиционными вопросами, должны предлагаться более сложные, учитывающие национальную специфику.

Интеллектуальные игры должны соответствовать разработанной психологами системе эстетических, эргономических, психолого-педагогических и технико-экономических требований. Особое внимание следует обратить на цветовую гамму, цветовую палитру, яркость и контрастность изображения на экране монитора. Цвета необходимо подбирать насыщенные, контрастные по отношению друг к другу, встречающиеся в природе. Необходима тщательная проработка дизайна. Ничего не должно вызывать у ребенка отрицательных эмоций. Необходимо соблюдение рекомендуемых санитарно-гигиенических требований при работе с компьютером в этом возрасте.

Разработанная концепция определяет возможность создания ИИ, необходимых для умственного и интеллектуально развития детей, способных оградить их психику от пагубного влияния большинства существующих на рынке компьютерных игр.

### **Литература**

1. Смирнова Е.О. Детская психология. М.: Владос, 2003. 368 с.
2. Вачков И.В. Самосознание учителей и учащихся: полисубъектный подход: монография. М.: Издательство СГИ, 2000. 114 с.
3. Хуторской А.В., Андрианова Г.А. Информатика и ИКТ в начальной школе. М: Бином, 2009. 152 с.
4. Ратанова Т.А. Когнитивная дифференцированность как основа личностного развития // Материалы III Всероссийского съезда психологов, 25-28 июня 2003 г. СПб, т.6, 2003. С.503-507.
5. Абраменко А.И. Педагогико-эргономические условия безопасного и эффективного использования вычислительной техники, информатизации и коммуникации в сфере общего среднего образования // Информатика и образование. 2000. №10. С.26-37.
6. Учение и зрение: методическое пособие по охране, гигиене и профилактике зрения. Сост. З.П. Малева. Челябинск: Изд-во «Околица», 2002. 127 с.
7. Безруких М.М., Логинова Е.С. Возрастные особенности структуры саморегуляции деятельности у детей 4- 5 лет // Мир психологии. 2002. № 1. С. 121-126.



8. Зворыгина Е. Педагогические подходы к компьютерным играм для дошкольников // Информатика и образование. 1990. №6. С.94-102.
9. Корецкий В. Компьютерные игры и нравственный выбор // Информатика и образование. 1989. №2. С. 119-121.
10. Войскунский А.Е. Актуальные проблемы зависимости от Интернета // Психологический журнал. № 1, 2004. С. 90-100.
11. Subrahmanyam K., Kraut R. E., Greenfield P.M., Gross E.F. The impact of home computer use on children's activities and development // Available as [WWW document] URL [www.futureofchildren.org/urs\\_doc/](http://www.futureofchildren.org/urs_doc/)
12. Макалатия А.Г. Мотивация в компьютерных играх // 3-я Российская конференция по экологической психологии (Москва, 15-17 сентября 2003 г.). Тезисы. М.: Психологический институт РАО, 2003. С. 358-361.

**СЕКЦИЯ 8. ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ  
ПРОГРАММИРОВАНИЕ, ГРИД-ТЕХНОЛОГИИ,  
ПРОГРАММИРОВАНИЕ НА ГРАФИЧЕСКИХ ПРОЦЕССОРАХ**

**Астахова И.Ф.,**

Воронежский государственный университет, профессор  
[astachova@list.ru](mailto:astachova@list.ru)

**Коробкин Е.А.**

Воронежский государственный университет, аспирант  
[Evgeniy-voronez@mail.ru](mailto:Evgeniy-voronez@mail.ru)

## ***Технология CUDA для метода дискретных элементов в параллельной среде***

Гранулированные материалы, такие как песок, грунт, порошки или зерно, являются неотъемлемой частью нашего мира. Существует потребность в создании симуляции динамики таких материалов. В горнодобывающей промышленности – это изучение самообрушения пород; в сельскохозяйственной отрасли изучение поведения зерна в каналах имеет большое значение.

Метод Дискретного Элемента или МДЭ описывает способ симуляции сыпучих сред. В нем каждая частица материала представляется как тело из механики Ньютона, которое имеет массу и координаты центра, и на которое действуют силы. Таким образом, частицы взаимодействуют друг с другом по определенным законам и возможно наблюдать движение системы в целом. В настоящее время существует несколько комплексов приложений для моделирования частиц с помощью МДЭ. Примером могут служить Open Source проекты LIGGGHTS и Yade, которые решают целый спектр научных задач, связанных с Методом Дискретных Элементов. Устройства находятся в широком доступе, а их мощностей хватает для небольших научных исследований с участием сотен тысяч частиц, тогда как использовать ЦП (центральный процессор) можно для расчета очень небольших систем за приемлемое время.

Технология CUDA, представленная компанией NVIDIA в 2006 г., предоставляет интерфейс низкого уровня для решения задач программирования в параллельной среде с помощью видеокарт NVIDIA и идеально подходит для целей данной работы. В данной работе, используя архитектуру параллельных вычислений CUDA, необходимо реализовать программное и аппаратное обеспечение, позволяющее производить симуляцию гранулированных частиц. Изучить производительность такого программного комплекса относительно классических способов реализации, использующих центральный процессор в качестве основной вычислительной единицы.

### ***Технология CUDA***

Технология CUDA использует мощности графических процессоров.

Данные устройства оптимизированы для работы с числами с плавающей точкой одинарной точности; их архитектура сильно отличается от архитектуры ЦП.

Графические процессоры (ГП) сконцентрированы на вычислениях больше, чем на контроле выполнения (условные переходы) и кэшировании данных. Также ГП ориентируются на параллельные вычисления. Поэтому данные устройства являются периферийными для решения конкретных вычислительных задач, которые возможно выполнять в параллельной среде, при этом основная логика приложения выполняется на ЦП.

Для того чтобы технология CUDA была масштабируема, вводится три ключевых абстракции: иерархия групп потоков, разделенная память и барьерная синхронизация. Программист должен разделить задачу на несколько подзадач, которые выполняются независимо разными блоками потоков, которые в свою очередь выполняют маленькую часть этой подзадачи. Разделенная память и синхронизация позволяют этим потокам работать вместе. Масштабируемость достигается за счет того, что блок может быть выполнен на любом ядре ГП, в любой последовательности, параллельно или последовательно, так что программа может быть выполнена на любом количестве доступных ядер.

CUDA расширяет язык C вводя понятия *ядер* (kernel) – функций на языке C, которые, когда вызываются, выполняются N раз на N различных потоках CUDA.

Ядро определяется с помощью приставки **\_\_global\_\_**, а количество потоков определяется с помощью нового синтаксиса конфигурации запуска **<<<...>>>**. Каждый поток, который выполняет ядро, получает уникальный индекс, доступ к которому можно получить через встроенную переменную **threadIdx**.

Для удобства, переменная **threadIdx** – трехкомпонентный вектор, так что потоки могут быть идентифицированы по одномерному, двумерному и трехмерному индексу, образуя при этом одномерную, двумерную или трехмерную сетку потоков. Таким образом обеспечивается прозрачность в работе с такими объектами как массив, матрица или объемная сетка.

Количество потоков в блоке ограничено из-за того, что все потоки должны быть расположены в пределах одного процессорного ядра. В настоящее время, максимальное количество потоков на блок равняется 1024. Однако ядро может быть выполнено на нескольких блоках, так что общее количество потоков равняется количеству потоков в блоке умноженному на количество блоков.

Блоки также организованы в одномерную, двумерную или трехмерную сетку. Каждый блок в сетке может быть идентифицирован одномерным, двумерным или трехмерным индексом, доступ к которому в ядре можно получить через переменную **blockIdx**. Размерность блока можно узнать из переменной **blockDim**.

Блоки должны быть независимы: они могут запускаться в любом

порядке, параллельно или последовательно. Это необходимо для того, чтобы каждый блок мог быть запланирован на любом процессорном ядре, не зависимо от их количества; это позволяет писать код, который масштабируется по мере увеличения количества ядер.

### ***Иерархия памяти***

Потоки CUDA могут иметь доступ к различным пространствам памяти во время выполнения. Каждый поток имеет свое локальное пространство памяти. Каждый блок потоков имеет разделенную память, которая видима всем потокам этого блока. Все потоки имеют доступ к единой *глобальной* памяти.

Существует еще два источника данных только для чтения – *текстурная память* и *константная память*. Глобальная, текстурная и константная памяти оптимизированы для различных действий. Текстурная память также позволяет производить фильтрацию данных для некоторых форматов.

Центральный процессор называется *host* (хост), а графический процессор называется *device* (девайс) в терминах CUDA. Без хоста невозможно функционирование параллельного приложения, так как сама программа по-прежнему выполняется, используя ЦП, и только лишь отдельные фрагменты выполняются на девайсе. Хост и девайс поддерживают свое собственное пространство в памяти – память хоста (*host memory*) и память девайса (*device memory*). Так что программа управляет глобальной, константной и текстурной памятью с помощью обращения к API.

### ***Использование CUDA как средство повышения производительности МДЭ***

Рассмотрим способ ускорения вычислений, связанных с Методом Дискретных Элементов, с помощью CUDA. Дана система частиц с определенными физическими параметрами, и помещена в некоторое начальное состояние. Используя вычислительные мощности ГП (графического процессора), необходимо переводить систему в новое состояние. Переход системы из одного состояния в новое и далее будем называть *циклом системы*.

Большинство частиц находятся на относительно большом расстоянии друг от друга. Для данной частицы лишь небольшое количество других частиц находится рядом с ней, а производить расчет сил с другими частицами нет необходимости. Для того чтобы отсекал эти частицы, пространство делится на ячейки одинакового размера, образуя при этом *сетку*. Каждая такая ячейка содержит объекты, которые в ней находятся. Так как для каждой частицы можно узнать, в какой ячейке она находится, то также легко можно узнать, какие частицы находятся вместе с ней.

Для простоты, количество ячеек в сетке одинаково по всем трем измерениям. Ячейка задается координатами и представляет собой куб со стороной  $l$ , которая определяется диаметром. Когда частица находится в

данной ячейке, то необходимо проверять только те частицы, которые находятся в 26 окружающих ее ячейках. Предлагаются следующие способы оптимизации доступа к глобальной памяти:

- Кэширование данных в разделенной памяти;
- Линейный доступ к данным.

Помимо глобальной памяти – основного хранилища данных во время вычислений – существует еще разделенная память, доступная всем потокам в пределах блока. Это память небольшого размера, но доступ к ней осуществляется гораздо быстрее. Встроенные средства синхронизации позволяют использовать разделенную память как кэш данных: каждый поток загружает немного данных из глобальной памяти в разделенную, затем все потоки используют только разделенную память. Это сокращает количество обращений к глобальной памяти и ведет к ускорению работы приложения в целом.

Глобальная память оптимизирована таким образом, чтобы максимизировать производительность линейного доступа к данным. Несколько запросов к данным объединяются в одну транзакцию, если они вмещаются в один сегмент, размер которого зависит от запрашиваемых данных.

Таким образом, если несколько потоков выбирают данные, которые расположены друг за другом, то эти данные выбираются за один раз. Если все данные расположены в случайном порядке, то количество транзакций увеличивается и это ведет к снижению производительности.

В свете вышесказанного, предлагается модифицированный метод хранения частиц в ячейках.

Для работы с технологией CUDA необходим графический процессор, который ее поддерживает. Все ГП компании NVIDIA, начиная с серии 8800, поддерживают данную технологию. Однако более новые модели поддерживают более новую версию CUDA 2.1, которая имеет расширенные возможности. В качестве графического процессора была выбрана видеокарта NVIDIA GeForce GTS 450. Данный ГП имеет относительно невысокую цену, приемлемую производительность и поддерживает последнюю версию технологии.

Также необходимо установить на компьютер CUDA Toolkit – средства разработчиков. Это пакет из программных продуктов, позволяющий создавать параллельные приложения. Он включает в себя компилятор, библиотеки времени выполнения, профайлер, а также исчерпывающую справочную информацию. CUDA Toolkit интегрируется в Microsoft Visual Studio, поэтому это приложение также необходимо. Использовалась версия 2010.

Само приложение является графическим. Трехмерная графика выводится на экран с помощью технологии DirectX 10. Для этого был использован DirectX SDK. Данный выбор был сделан в виду того, что автор

данной работы уже ознакомлен с технологией. В качестве альтернативы можно использовать OpenGL.

Приложение состоит из двух окон: окно консоли и главное графическое окно. В окне консоли выводится информация об ошибках и событиях.

В результате выполненной работы было создано приложение, позволяющее выполнять симуляцию частиц с помощью метода дискретных элементов, используя архитектуру параллельных вычислений CUDA. Было получено, что с помощью данной технологии производительность МДЭ можно увеличить на порядок, по сравнению с классическими решениями, использующими центральный процессор.

В архитектуру CUDA изначально включена возможность масштабирования, а благодаря доступности, данная технология претендует на широкое использование в самых различных сферах научной и повседневной деятельности.

### **Литература**

1. Wikipedia.[Электронный ресурс], [http://ru.wikipedia.org/wiki/Метод\\_Дискретного\\_Элемента](http://ru.wikipedia.org/wiki/Метод_Дискретного_Элемента)
2. CUDA. [Электронный ресурс], [http://www.nvidia.ru/object/cuda\\_home\\_new\\_ru.html](http://www.nvidia.ru/object/cuda_home_new_ru.html).

***К вопросу о реализуемости процесса анализа  
устойчивости систем управления на параллельных  
платформах***

Транспорт, работающий на бензине, является ответственным за значительную часть загрязнений атмосферы в мире. По данным Saber и Venayagamoorthy [1] вклад транспорта, работающего на ископаемом топливе, в суммарное количество мировых выбросов составляет 24%. Потребность в уменьшении загрязнений окружающей среды и озабоченность глобальными изменениями климата на планете приводит к необходимости использования нового поколения транспортных средств: электромобилей.

Однако на пути такого решения стоят несколько очень важных проблем, одной из которых является дополнительная нагрузка на сеть, создаваемая электромобилями (Electrical Vehicle - EV), в случае их широкого распространения на рынке. Очевидно, что для зарядки такого мощного устройства как электромобиль из сети, требуется большое количество электроэнергии, которое сравнимо с общим количеством электроэнергии, потребляемым всеми электроприборами, используемыми одной семьей, если речь идет о зарядке EV дома [2].

Естественно, что для стимуляции использования экологически чистого транспорта автовладельцами возможности заряжать электромобили дома недостаточно. Важным аспектом, связанным с распространением EV является задача построения территориально распределенных станций подзарядки электромобилей. Необходимо построение интеллектуальной инфраструктуры заправочных станций (возле парковочных мест, в придорожной зоне, в особенности возле оживленных магистралей), способных справиться с возможными перегрузками и распределить потоки транспортных средств оптимальным образом. Самым главным критерием надежности системы управления (СУ) такой инфраструктурой должно являться обеспечение функционирования ее объектов во всем диапазоне возможных входных параметров. В общем случае анализ алгоритмов, лежащих в основе работы СУ, является сложной задачей, для решения которой сегодня могут быть использованы современные параллельные вычислители.

При неконтролируемой стратегии зарядки EV для того, чтобы обеспечить необходимую мощность сети, потребуются дополнительные генераторы, если все электромобили будут заряжаться одновременно, или



же усиление существующих. Специалисты из Pacific Northwest National Laboratory оценивают долю рынка США, которую может обслужить существующая энергосистема в 84% [2]. Однако для России этот процент будет гораздо ниже и поскольку существующая ситуация заставляет серьезно задуматься об альтернативных источниках энергии для транспорта, то необходимо будет серьезно изменить существующую электросеть. Причем эти изменения потребуют не только увеличения мощности генераторов, но также создания интеллектуальных систем контроля. Увеличение нагрузки и сложность предсказания ее распределения во времени (выбор времени зарядки целиком зависит от владельца транспортного средства, от его привычек и графика дня, и повлиять на это весьма трудно) приведут к пиковым нагрузкам на сеть, повысят вероятность превышения максимально возможной мощности для распределяющего трансформатора.

Суммарное потребление электроэнергии подсетью существенно возрастает при зарядке электрических батарей. Это может приводить к значительным пикам нагрузки потребления и тока, что в свою очередь очень негативно отражается на электросети. Коэффициент нагрузки (Load Factor) рассчитывается следующим образом

$$Load\ Factor = \frac{P_{max}}{P_{aver}}$$

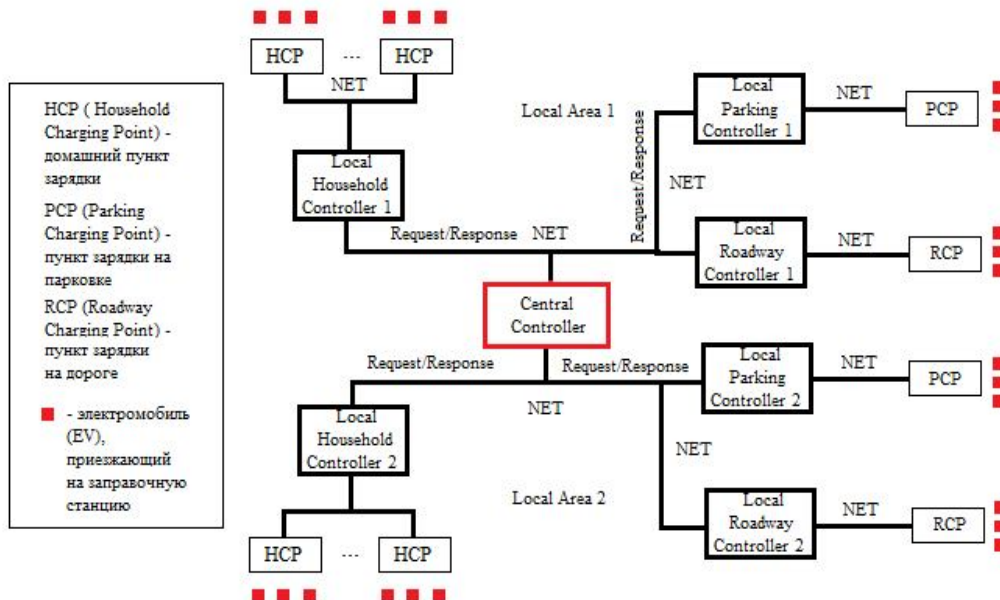
где  $P_{max}$  - максимальная нагрузка на подсеть, а  $P_{aver}$  - средняя нагрузка за исследуемый период.

Будем считать целью алгоритма управления максимальное сглаживание пиков в сети заправок, а критерием устойчивости входение LF, и соответственно пиков нагрузки, в некоторый определенный диапазон.

Одним из критериев регулирования нагрузки на систему может быть стоимость зарядки, а именно низкая цена электроэнергии в часы небольшой нагрузки, и высокая в часы большой. Это позволяет перераспределять нагрузку и поощрять пользователя к зарядке в выгодное для него и для состояния системы время.

Рассмотрим задачу анализа устойчивости алгоритма управления сетью заправок. Будем исходить из предположения, что датчики, работающие на каждой заправочной станции и являющиеся источниками входной информации для центрального контроллера, вырабатывающего управляющее решение для сети заправочных станций, могут выходить из строя и давать ложные показания. Для оценки устойчивости алгоритма управления, т.е. его способности предотвращать перегрузку питающей электросети, необходимо проверить его корректную работу в случае поступления на вход искаженных данных. Это могут быть и показания, которые не возникают в реальной жизни, однако могут выдаваться датчиками в случае их неисправности. Учет неисправности датчиков при оценке устойчивости ведет к факториальному увеличению сложности задачи.

Для управления такой сложной иерархической системой, необходимо обеспечить взаимодействие между ее узлами, чтобы повысить интеллектуальность СУ и позволить перераспределять ресурсы в зависимости от нагрузки сети. Мы предполагаем, что управляющие узлы должны обмениваться информацией - запросами и ответами с помощью некоторого протокола.



**Рис. 1.** Схема запросов электроэнергии от различных видов заправочных станций и обмена сообщениями для сетевой интеллектуальной инфраструктуры

На рис. 1 изображена схема запросов, поступающих от электромобилей, приезжающих на различные виды заправок (дома, на парковках, возле дорог). На схеме представлена сеть локальных контроллеров и один центральный, которые обмениваются между собой информацией и на основе ответа (Response) принимают решение о доступе к источнику энергии или же отказе. Для того чтобы система была устойчивой, нужно чтобы и в случае неверного ответа от узла, к которому был отправлен запрос на необходимую информацию (это может быть СС (Central Controller), или же соседнее управляющее устройство) было выработано корректное решение каждым локальным контроллером.

Взаимодействие контроллеров через передачу сообщений может обеспечить управление сетью, однако стратегии управления, которые реализуют как центральный, так и локальные контроллеры, могут быть весьма разнообразны. Выбор оптимальной стратегии и оценка ее устойчивости к широкому диапазону нагрузок является сложной задачей многокритериальной оптимизации, для решения которой необходимо использовать методы имитационного моделирования, поскольку эту задачу нельзя решить аналитическими методами.

Необходимо проверить весь диапазон состояний каждого объекта,

что ведет к большой вычислительной сложности задачи. Поскольку характер потребности автомобилей в зарядке носит случайный характер, то для определения устойчивости имитационной модели требуется генерация некоторого количества симуляций для каждого состояния системы S. Если количество состояний одного датчика K и в системе имеется N таких датчиков, то - число возможных состояний. Таким образом, общее количество необходимых симуляций составляет .

### **Анализ алгоритма работы центрального контроллера сетевой интеллектуальной структуры заправочных станций для электромобилей**

Для иллюстрации применимости предлагаемого подхода проведен анализ упрощенной модели, состоящей из центрального контроллера и двух заправок, каждая из которых имеет свой собственный локальный контроллер. Предполагается, что на каждой из заправок в определенный момент времени может заряжаться только один электромобиль. Таким образом, каждая из заправок создает одну очередь из автомобилей, желающих зарядиться.

Для подстанций электрической сети, имеющих достаточно высокую мощность, нагрузку можно рассматривать как случайную величину с нормальным законом распределения. Эта гипотеза базируется на центральной предельной теореме теории вероятностей и математически ее результат можно представить в следующем виде:

$$P_t = \sum_{i=1}^n P_i, \quad (1)$$

где  $P_i$  - мощность отдельного потребителя;  $P_t$  - значение электрической нагрузки узла электрической сети, к которому присоединено n потребителей. Данное представление, справедливое для статической нагрузки, которую мы и будем рассматривать, позволяет оценивать общую нагрузку на узел сети и может использоваться контроллером для оценки состояния системы в текущий момент времени.

Пусть нагрузка на 1 заправке в некоторый момент времени  $t$  обозначается  $P_{PCP1}(t)$ , на второй  $P_{PCP2}(t)$ , а суммарная нагрузка на подстанцию, обслуживающую эти 2 заправочные станции  $P_{total}(t)$ , тогда принимая во внимание (1)

$$P_{total}(t) = P_{PCP1}(t) + P_{PCP2}(t).$$

Коэффициент нагрузки (LF - Load Factor) для данной системы рассчитывается как

$$LF = \frac{P_{maxtotal}}{P_{avertotal}}. \quad (2)$$

Чем ближе значение коэффициента нагрузки к 1, тем выше оптимизирована сеть и тем лучше для системы энергоснабжения, резкие выбросы же крайне нежелательны.

Для того чтобы симулировать очереди из машин создается поток заявок, каждая из которых характеризуется скоростью зарядки  $v_{charge}$  кВт/ч,

энергией полностью заряженного аккумулятора  $W_{cap}$  и уровнем заряда в процентах  $q$ . Для каждой станции существуют несколько возможных режимов зарядки, характеризующихся соответствующей скоростью заряда - 3, 5, 10, 20, 30, 40 и 50 кВт/ч. Было принято допущение, что каждый из обслуживаемых автомобилей ставит своей целью полную зарядку аккумулятора (например, если станция сильно удалена от места, где можно опять пополнить уровень заряда). Следовательно, общее количество энергии, которое необходимо для зарядки рассчитывается как:

$$P_{req} = W_{cap} * q,$$

а время, необходимое для этого

$$t_{charge} = \frac{P_{req}}{v_{charge}} = \frac{W_{cap} * q}{v_{charge}}.$$

Объектом анализа на устойчивость был выбран следующий простой алгоритм центрального контроллера:

при превышении некоторой суммарной нагрузки  $P_{refuse}$  на подсистему происходит отказ в обслуживании тем пользователям, оцениваемое время зарядки для которых превышает некоторое значение  $t_{refuse}$ .

Очевидно, что такой алгоритм слишком прост для реальной системы, однако и его можно использовать для оценки сложности задачи. За критерий устойчивости моделируемой системы примем вхождение коэффициента нагрузки в заданный диапазон.

В данном случае функции локального контроллера сведены к минимуму - он просто пересылает полученный запрос на выдачу электроэнергии центральному контроллеру, управление этой простой сетью, таким образом, получается полностью централизованным. Для проверки устойчивости алгоритма управления центральным контроллером на всем множестве входных значений, будем варьировать параметры  $P_{refuse}$  и  $t_{refuse}$ , собирая среднее статистическое значение LF и совершая  $N$  симуляций обработки двух очередей заявок с заправок на протяжении суток для каждой пары этих коэффициентов, задающих алгоритм управления. Варьирование параметра  $P_{refuse}$  достигалось изменением значения коэффициента  $\alpha$ , такого что

$$P_{refuse} = \alpha \cdot P_{max},$$

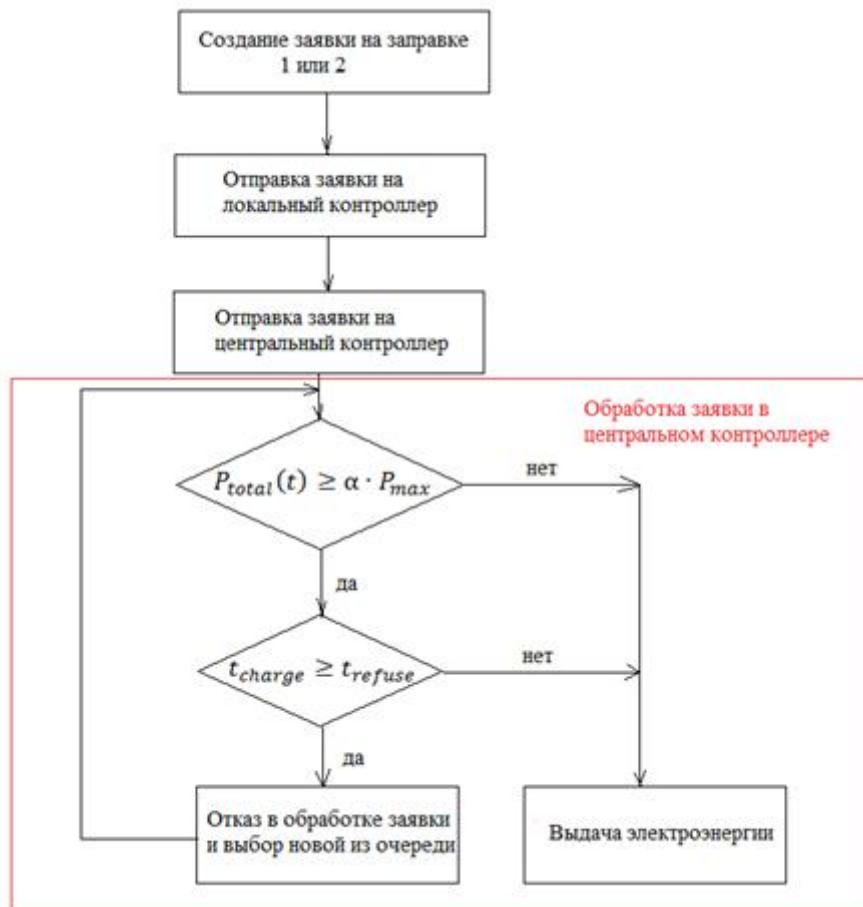
где  $P_{max}$  - максимально допустимая для общей подстанции нагрузка. В проведенном эксперименте  $P_{max}$  являлось фиксированной величиной.

Структурная схема алгоритма работы центрального контроллера приведена на рис. 2. Разработанная программа позволяет оценить устойчивость его алгоритма управления. Для этого значения коэффициента  $\alpha$  варьируются на отрезке  $[0,1]$  с шагом 0.05, а параметра  $t_{refuse}$  (в минутах) на отрезке  $[30;600]$  с шагом 15 минут.

Результаты были получены для значения  $N=1000$ , при котором значение среднее значение LF (за 24 часа) на  $N$  симуляциях имеет достаточно небольшое отклонение от математического ожидания (было

определено путем экспериментов).

На рис. 3 представлен график зависимости коэффициента нагрузки от варьируемых параметров, на котором можно видеть при каких значениях происходит резкий скачок LF, при том, что оптимальным является приближение к 1.



**Рис. 2.** Структурная схема алгоритма работы центрального контроллера.

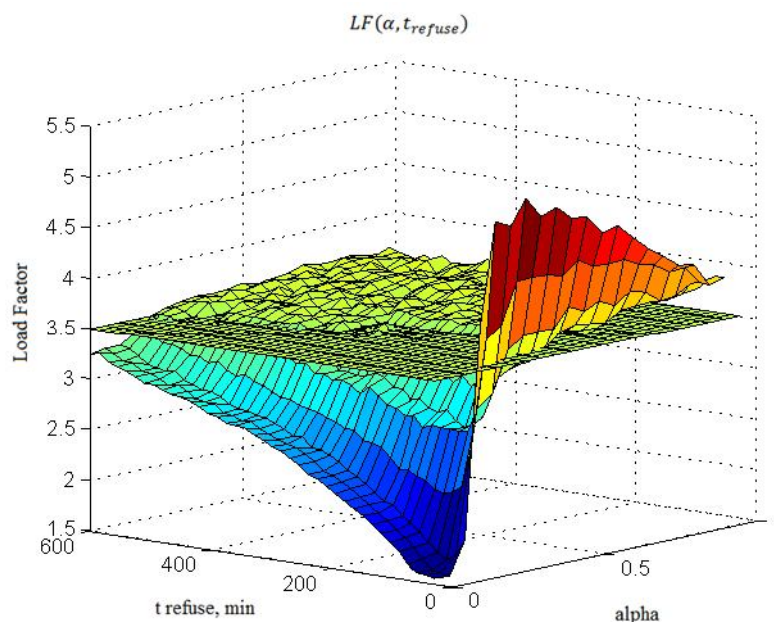
Приняв за критерий устойчивости исследуемого алгоритма  $LF \leq 3.5$ , можно определить значения параметров  $\alpha$  и  $t_{refuse}$ , при которых поведение системы является некорректным.

Таким образом, при анализе даже простого алгоритма с изменением параметров  $\alpha$  на отрезке  $[0;1]$  с шагом 0.05,  $t_{refuse}$  на отрезке  $[30;600]$  с шагом 15 минут и количеством симуляций  $N=1000$ , суммарное время, потребовавшееся разработанной однопоточной программе для вычислений на мощном современном компьютере составило 54 минуты. Мы получили среднее время вычисления одного элемента из результирующей матрицы  $t_{LF_1} \approx 4,15$  с.

Проведенные исследования подтвердили возможность оценки устойчивости СУ с помощью средств имитационного моделирования и позволили определить временные характеристики процесса анализа. Полученные результаты показывают высокую вычислительную сложность



данной задачи даже для упрощенной модели, состоящей из двух локальных контроллеров пунктов зарядки электромобилей, запрашивающих разрешение на подключение абонентов у центрального контроллера. Это обуславливает необходимость переноса процесса моделирования в параллельную среду.



**Рис. 3.** Визуализация значений  $LF(\alpha, t_{refuse})$ , при которых математическое ожидание коэффициента нагрузки не удовлетворяет критерию устойчивости системы управления сетью заправок

### **Обоснование возможности использования распределенных вычислений**

Задача анализа устойчивости прямо связана с рассмотрением всех возможных состояний системы, что в свою очередь требует больших временных затрат и вычислительных мощностей. Впрочем, проблема длительности симуляций всегда существует, когда речь идет о моделировании сложных систем. Вследствие этого предполагается, что необходимо использование технологий параллельного программирования для распределения вычислений между ядрами и процессорами суперкомпьютера.

В общем случае, если анализ алгоритма управления распределением ресурсов сети заправочных станций проводится по некоторому набору варьируемых коэффициентов вектора  $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ , однозначным образом определяющих свойства алгоритма, то каждый из параметров может принимать некоторый набор значений  $s_{\alpha_1}, s_{\alpha_2}, \dots, s_{\alpha_n, m_i}$ . Вычисления, необходимые для оценки устойчивости алгоритма, для каждого состояния вектора являются совершенно независимыми друг от друга. Это дает возможность распределить их между ядрами многопроцессорной системы, где каждый узел будет работать со своими параметрами

$\{\alpha_1(s_{\alpha_1 m_j}), \alpha_2(s_{\alpha_2 m_j}), \dots, \alpha_n(s_{\alpha_n m_j})\}$ . Такой подход позволит сократить время вычислений пропорционально числу задействованных узлов.

### **Литература**

1. Saber A.Y. and Venayagamoorthy G.K. (2009). One million plug-in electric vehicles on the road by 2015. In: Proc. of 12th International IEEE Conference on Intelligent Transportation Systems, St. Louis, MO, October 3-7, pp. 141-147.
2. Davydova A., Lupin. S., Chakirov R., Vagapov Y. Optimization of In-home Charging Schedule of Plug\_in Electric Vehicle // Conference on Electrical and Computer Systems, Ottawa, Ontario, Canada 22-24 August 2012, Paper No. 204.

**Лупин С.А.,**

Национальный исследовательский университет «МИЭТ», Москва  
профессор, e-mail: [lupin@miee.ru](mailto:lupin@miee.ru)

**Тан Шейн,**

Национальный исследовательский университет «МИЭТ», Москва  
аспирант, e-mail: [mieegenius@gmail.com](mailto:mieegenius@gmail.com)

**Чжо Мью Хтун**

Национальный исследовательский университет «МИЭТ», Москва  
аспирант, e-mail: [kyawmyohtun.2013@gmail.com](mailto:kyawmyohtun.2013@gmail.com)

## ***Параллельная реализация алгоритмов дискретной оптимизации***

### ***Введение***

Алгоритмы дискретной оптимизации используются при решении различных прикладных задач. В общем случае они направлены на нахождение некоторых координат в  $N$ -мерном пространстве, при которых значение функционала, характеризующего состояние системы, достигает оптимума. Практических задач, которые сводятся к дискретной оптимизации, как и методов их решения, существует довольно много, однако большинство из них позволяют получить не точное решение задачи, а лишь некоторое приближение к нему. Для каждого класса задач, точность такого приближения оценивается с помощью сравнения результатов работы алгоритмов с заранее найденным точным или известным нам лучшим решением. Во многих приложениях для его нахождения используется ресурсоемкий метод полного или усеченного (ветвей и границ) перебора. Например, для задачи о ранце, которая относится к  $NP$ -сложным и используется в системах криптозащиты, опыт использования метода ветвей и границ описан в [1]. Для  $NP$ -полных задач уже при размерности порядка  $N = 20$  такой подход не может быть реализован даже на современных вычислительных системах, поскольку число вариантов решения при этом будет составлять  $20! \approx 2.4 \cdot 10^{18}$ . С учетом числа операций, необходимых для вычисления оценки каждого варианта, число машинных команд составит  $10^{20} - 10^{21}$ . Это значит, что время выполнения программы даже на суперкомпьютерах с петафлопной производительностью, займет более 1 года.

В качестве алгоритма, который позволяет найти координаты оптимального решения в  $N$ -мерном пространстве, авторами настоящей статьи предлагается использовать параллельные реализации алгоритмов случайного поиска. В качестве задач дискретной оптимизации рассматриваются задачи назначения на узкие места и квадратичного



назначения.

### **Задачи дискретной оптимизации**

#### **Задача квадратичного назначения**

К задаче квадратичного назначения сводится, в частности, задача размещения элементов на коммутационной плате, которая формулируется следующим образом [2]. Пусть задано множество элементов  $e_1, e_2, \dots, e_N$ . Вся схема соединений задается квадратной матрицей соединений  $\mathbf{R}$  с числом элементов  $N \times N$ , номера строк и столбцов которой соответствуют элементам схемы, а элементы матрицы  $r_{ij}$  определяют связи элементов друг с другом. Матрица  $\mathbf{D}$  определяет расстояния  $d_{ij}$  между посадочными местами элементов на коммутационной плате. Алгоритмы размещения минимизируют квадратичный функционал, косвенно оценивающий суммарную длину печатных соединений:

$$F = \sum_{i=1}^N \sum_{j=1}^N r_{ij} d_{p(i)p(j)}, \quad (1)$$

где  $p(i)$  задает номер позиции, присвоенной  $i$ -му элементу.

Для нахождения точного решения необходимо произвести перебор  $N!$  различных вариантов размещения элементов. В алгоритмах случайного поиска решение получается путем выбора лучшего варианта из некоторого множества случайно сгенерированных размещений.

#### **Задача назначения на узкие места**

К задаче назначения на узкие места сводится задача диспетчеризации в распределенных системах обслуживания, которую можно представить в следующем виде [3]: пусть  $N$  – число заявок, поступивших от обслуживаемых объектов;  $M$  – число обслуживающих объектов;  $y_{ij}$  – целевая переменная, равная 1, если  $j$ -ый объект используется для обслуживания  $i$ -ого вызова и равная 0 в противном случае;  $r_{ij}$  – время следования  $j$ -ого объекта к источнику  $i$ -ой заявки.

Тогда целевая функция системы (2), минимизирующая время реакции, имеет вид:

$$T_{react} = \max(r_{ij} y_{ij}, \forall i = \overline{1, N}; \forall j = \overline{1, M}) \rightarrow \min. \quad (2)$$

При этом:

$$\sum_{i=1}^N y_{ij} \leq 1, j = \overline{1, M}. \quad (3)$$

В такой постановке, число обработчиков заявок должно быть больше или равно числу самих заявок  $M \geq N$ . Исходной информацией для алгоритмов распределения заявок служит матрица  $\mathbf{R}$ , элементы которой  $r_{ij}$ , кроме времени движения обработчика к объекту, могут характеризовать и возможность выполнения объектом своей функции по отношению к источнику заявки. В этом случае минимизируя  $T_{react}$ , мы оптимизируем не только время движения, но и комплексный показатель эффективности системы обслуживания.

### ***Программная реализация алгоритмов***

На каждой итерации алгоритма случайного поиска осуществляется генерация случайного решения и находится значение целевой функции для него. Если это решение лучше известного нам, то оно запоминается и процесс продолжается заданное число итераций. Сложность алгоритма определяется числом итераций. Точность решения, получаемого случайными алгоритмами, в общем случае зависит от количества итераций.

При параллельной реализации алгоритма исходят из того, что его итерации независимы по данным [4]. Это позволяет распределить число итераций между отдельными процессами или потоками, реализующими алгоритм, которые не будут взаимодействовать между собой в процессе вычислений. Синхронизация необходима только после завершения каждым из них расчетов. В качестве общего решения берется лучший вариант, найденный всеми параллельными ветвями программы.

#### ***Использование библиотеки MPI***

Параллельная программа решения задачи квадратичного назначения реализуется для кластерной многопроцессорной вычислительной системы и использует библиотеку MPI [4]. Общее число итераций в алгоритме равномерно распределяется между процессорами кластера, пересылка промежуточных результатов вычислений в процессе работы алгоритма не производится.

#### ***Использование библиотеки OpenMP***

Параллельная программа решения задачи назначения на узкие места реализована для многоядерной вычислительной системы и использует библиотеку OpenMP [4]. Общее число итераций в алгоритме равно произведению количества ядер в системе на количество итераций, выполняемое каждым из них.

Одной из основных проблем при реализации алгоритмов случайного поиска является быстрое получение случайного вектора с неповторяющимися элементами, описывающего решение.

В реализованных авторами алгоритмах используется метод ускорения генерации случайного вектора [5]. Это достигается за счет того, что случайное число получается как комбинация двух случайных чисел меньшей размерности, что снижает вероятность повторов.

#### ***Результаты работы алгоритмов***

Поскольку случайные алгоритмы ведут неупорядоченный поиск, то вероятность нахождения точного решения будет зависеть от соотношения числа рассмотренных вариантов к общему числу возможных вариантов решений. При  $N=18$  число вариантов решений будет равно  $18! \approx 6.4 \cdot 10^{15}$  и современные вычислительные системы еще позволяют провести полный перебор вариантов, однако даже при незначительном увеличении размерности это потребует больших затрат времени, уже при  $N=24$ , число вариантов составит  $24! \approx 0.6 \cdot 10^{24}$ . С практической точки зрения представляют интерес решения, которые дают случайные алгоритмы за  $10^6$

–  $10^{10}$  итераций. При соотношении числа итераций к числу вариантов меньшим  $10^6$ , вероятность нахождения точного решения будет пренебрежимо мало отличаться от нуля. Задачей проводимых исследований является определение точности параллельных алгоритмов случайного поиска при решении *NP*-полных задач дискретной оптимизации.

### **Задача квадратичного назначения**

В качестве тестовой для задачи квадратичного назначения используется *тест-задача Штейнберга* [1]. Размерность задачи составляет  $N=36$ , точного решения для нее в литературе не приводится. Она используется в качестве оценочной для алгоритмов решения задачи размещения элементов. Особенностью задачи является разреженная матрица **R** и слабо выраженная кластеризация элементов.

Результаты работы параллельного алгоритма [6] на многопроцессорной кластерной вычислительной системе представлены в табл. 1. В каждом случае эксперимент повторялся 5 раз, а полученные данные усреднялись.

Таблица 1. Решение задачи квадратичного назначения ( $N = 36$ )

Число итераций	Число процессоров			
	1	8	16	32
$10^3$	6820	6120	6156	5817
$10^4$	5901	5639	5780	5400
$10^5$	5566	5200	5532	5418
$10^6$	5150	4954	5084	4935
$10^7$	4751	4850	4830	4699

### **Задача назначения на узкие места**

В идеальном случае, оценку точности решений алгоритмов, нужно давать, сравнивая их результаты с точным решением, но далеко не для всех задач такое решение может быть получено. Для задачи назначения на узкие места нижняя граница точного решения может быть получена с помощью последовательного решения задачи линейного назначения для преобразованной матрицы **R**. Этот фактор повлиял на выбор ее в качестве объекта исследований. Для рассмотренных в работе примеров такое решение составляет:  $T_{react} = 141$  (для  $N = 18$ ) и  $T_{react} = 151$  (для  $N = 24$ ).

Исследования проводились на рабочей станции, имеющей два четырехядерных процессора Intel. Число задействованных потоков совпадало с числом доступных приложению ядер. Результаты работы алгоритма представлены в табл. 2 и 3.

Таблица 2. Решение задачи назначения на узкие места ( $N = 18$ )

Число итераций	$T_{react}$					
	Эксперименты					Среднее значение
	1	2	3	4	5	
$10^4$	189	177	178	180	176	180
$5 \cdot 10^4$	167	155	170	167	162	164

$10^5$	146	162	169	153	132	152
$5*10^5$	141	146	143	146	141	143
$10^6$	141	143	143	143	143	143
$5*10^6$	143	143	141	141	141	142
$10^7$	141	141	143	141	143	142

Таблица 3. Решение задачи назначения на узкие места (  $N = 24$  )

Число итераций	$T_{react}$					
	Эксперименты					Среднее значение
	1	2	3	4	5	
$10^4$	342	353	342	307	348	338
$5*10^4$	330	333	245	345	321	314
$10^5$	316	342	316	339	312	325
$5*10^5$	314	293	285	299	196	277
$10^6$	245	204	225	239	225	227
$5*10^6$	221	187	221	230	225	217
$10^7$	204	204	199	194	199	200
$5*10^7$	196	187	181	188	188	188
$10^8$	185	193	186	185	185	187
$5*10^8$	186	174	178	185	185	182
$10^9$	171	185	178	151	171	171
$5*10^9$	151	170	173	169	151	163

Как и в предыдущем случае, было проведено пять серий вычислительных экспериментов. Поскольку решение является целочисленным, полученные данные были усреднены с округлением. Ячейки, соответствующие точному решению задачи, выделены серым цветом. Результаты исследований быстродействия алгоритма для  $N = 24$  отражены в табл. 4.

Таблица 4. Время решения задачи назначения на узкие места (  $N = 24$  )

Число итераций	Время решения (сек)					Среднее значение
	1	2	3	4	5	
$10^4$	0.1	0.08	0.1	0.09	0.1	0.094
$5*10^4$	0.12	0.1	0.11	0.1	0.12	0.11
$10^5$	0.14	0.15	0.14	0.13	0.14	0.14
$5*10^5$	0.39	0.39	0.39	0.38	0.39	0.39
$10^6$	0.78	0.72	0.67	0.74	0.71	0.72
$5*10^6$	3.3	3.59	3.28	3.28	3.34	3.36
$10^7$	6.42	6.82	6.24	6.49	6.21	6.47
$5*10^7$	32.1	31.1	32.1	32.9	32.1	32.1
$10^8$	66.8	61.4	66.7	63	61.5	64
$5*10^8$	319	322	319	320	320	320
$10^9$	662	639	626	622	642	638
$5*10^9$	3264	3322	3292	3434	3431	3349

### Обсуждение результатов

Проанализируем полученные данные с точки зрения точности

алгоритмов и эффективности их распараллеливания.

### ***Задача квадратичного назначения***

Результаты эксперимента (табл.1) подтверждают, что и при параллельной реализации алгоритмов с увеличением числа итераций точность решения повышается для любого количества процессоров, участвовавших в эксперименте. Итерации равномерно распределялись между узлами системы и зависимость точности решения от числа процессоров менее очевидна. Хотя лучшее решение и получается всегда при максимальном числе узлов системы, худшее не всегда соответствует системе с одним узлом. Наблюдаются и некоторые другие отклонения. Объяснить это можно тем, что инициализация генератора случайных чисел в программе происходит от системного таймера. Кластер собран на двухпроцессорных модулях, для которых характерно то, что оба процессора используют общий системный таймер. Это приводит к тому, что в процессах, работающих на таких узлах, генераторы случайных чисел достаточно сильно коррелированы. Этот факт и снижает эффективность вычислений. Для устранения этого явления следует использовать внешний генератор случайных последовательностей.

### ***Задача назначения на узкие места***

Во всех экспериментах был реализован алгоритм, использующий все доступные для вычислений ядра процессоров. В отличие от многопроцессорных систем, все потоки многоядерного процессора используют единственный таймер, поэтому инициализация генераторов случайных чисел проводилась с помощью случайного массива, полученного за пределами потока.

Результаты исследования точности решений параллельной реализации (табл. 2 и 3) не противоречат общепринятой теории – увеличение числа итераций повышает точность. Отметим, что точные решения задачи алгоритм стабильно находит даже при соотношении числа итераций к числу возможных вариантов решений  $10^{-10}$  для размерности  $N=18$  и  $10^{-15}$  для размерности  $N=24$ . Казалось бы, при таком низком соотношении точные решения получить невозможно. Однако это справедливо лишь для тех задач, в которых точное решение является единственным. В работе не ставилась задача определить количество различных вариантов решений, обладающих минимальным значением  $T_{react}$ , но полученные данные косвенно подтверждают, что существует некоторое множество таких решений. Это и определяет высокую точность алгоритмов.

Рассмотрим еще один аспект, отражающий временные характеристики работы случайных алгоритмов. В табл. 4 представлены результаты исследования быстродействия алгоритма случайного поиска при решении задачи с  $N=24$ . Точное решение получается алгоритмом при числе итераций  $10^9$  и более. При этом многопоточное приложение затрачивает на расчеты около 1 часа при использовании рабочей станции,

имеющей два четырехядерных процессора Intel. Для алгоритмов, которые позволяют получить точное или близкое к нему решение, используемое для оценки точности других алгоритмов, это является приемлемым.

Но как определить необходимое число итераций, в том случае, если нам неизвестна нижняя граница оптимизируемого функционала? Одним из косвенных признаков получения алгоритмом локального оптимума, который может быть использован на практике, является тот факт, что в течение определенного числа итераций найденное решение не изменяется. В силу причин, отмеченных выше, необходимое число итераций будет зависеть не только от типа оптимизационной задачи, но также и от ее условий.

На наш взгляд, в качестве критерия окончания работы для алгоритмов случайного поиска целесообразно только временное ограничение. При необходимости реализации длительных вычислений, работу алгоритма можно разделить на несколько этапов. Например,  $10^{10}$  итераций можно получить путем 10 запусков алгоритма с числом итераций  $10^9$ . Кроме того, можно использовать вычислительные системы с большим числом ядер.

### **Заключение**

Таким образом, проведенные исследования подтвердили высказанное предположение о высокой эффективности параллельных реализаций алгоритмов случайного поиска при решении *NP*-полных задач дискретной оптимизации, встречающихся при цифровой обработке информации. Получаемые с их помощью решения могут быть использованы как в качестве эталонных для оценки работы быстрых алгоритмов, так и в самостоятельном виде в информационных системах, не критичных ко времени обработки.

### **Литература**

1. Колпаков Р. М., Посыпкин М. А. Об оценках вычислительной сложности варианта параллельной реализации метода ветвей и границ для задачи о ранце // Известия РАН: Теория и системы управления. № 5, 2011, С. 74-83.
2. Селютин В.А. Машинное конструирование электронных устройств. М.: Сов. радио, 1977.
3. Лупин С.А., Мью Мьинт Ту. Оценка вычислительной сложности основных этапов моделирования распределенных систем // Естественные и технические науки. 2004. №4.
4. Лупин С.А., Посыпкин М.А. Технологии параллельного программирования. Учебное пособие // М.: ИД «ФОРУМ»: ИНФРА-М, 2007.
5. Лупин С.А., Бажанов Е.И., Дорошенко Е.С., Подкопаев И.В. Практическая работа на кластере под управлением Microsoft HPC Server 2008: учебное пособие. М.: Москва, МИЭТ, 2011. 60 с.
6. Зей Яр Вин. Распараллеливание итерационных алгоритмов для многопроцессорных систем // Системный анализ и информационно-управляющие системы: Сборник научных тр. под ред. д.т.н., проф. В.А. Бархоткина. М.: МИЭТ, 2008. С. 164-168.

## ***Грид-систем из персональных компьютеров в России: текущее состояние и перспективы***

### ***Введение***

Современная наука и производство немыслимы без масштабных расчетов, требующих колоссальной вычислительной мощности, которую можно обеспечить только в рамках распределенных и параллельных систем. В настоящее время существует несколько подходов к организации распределенных вычислений. Наибольшую производительность обеспечивают специализированные суперкомпьютеры большой мощности, которых насчитывается несколько сотен по всему миру. Они могут решать широкий спектр вычислительных задач, но требуют больших расходов на создание и эксплуатацию.

Несколько меньшей производительностью обладают так называемые сервисные гриды, которые соединяют кластеры, установленные в различных организациях. Это решение дешевле по отношению к суперкомпьютерам, но также требует выделенных ресурсов и значительных усилий, связанных с эксплуатацией. Инфраструктура сервисных гридов состоит из набора сервисов, обеспечивающих доступ к брокерам ресурсов, информационным службам, хранилищам данных, вычислительным ресурсам. Пользователи сервисных гридов имеют соответствующие права доступа к предоставляемым сервисам. Контроль доступа к ресурсам осуществляется посредством сертификатов безопасности. Хорошо известны следующие технологии создания сервисных гридов, как Globus, LCG-2/gLite (EGEE), ARC [и Unicore].

На нижнем уровне рассматриваемой иерархии находятся грид-системы персональных компьютеров (ГСПК). ГСПК основаны на наблюдении, что большую часть времени персональные компьютеры либо простаивают, либо загружены лишь на некоторую небольшую долю своей мощности. ГСПК обеспечивают возможность объединения свободных вычислительных мощностей домашних компьютеров и персональных компьютеров учреждений в единую распределенную систему для решения сложных вычислительных задач.

В отличие от сервисных гридов, грид-системы из персональных компьютеров легко установить и поддерживать. Как правило, требуется одна рабочая станция, на которой запускается серверная часть инфраструктуры. Пользователи со всего мира имеют возможность подключать свои персональные компьютеры к этому ресурсу, предоставляя

тем самым свободные ресурсы своих компьютеров для работы приложений, размещенных на центральном сервере. Грид-системы из персональных компьютеров являются наиболее дешевым решением, обеспечивающим большую вычислительную мощность. ГСПК обладают огромным потенциалом роста – в настоящее время в мире насчитывается более одного миллиарда персональных компьютеров и их число стремительно увеличивается.

К сожалению, далеко не все распределенные приложения могут эффективно выполняться на подобных системах из-за серьезных ограничений, накладываемых возможностями по передаче данных и высокой вероятностью отказа узлов, участвующих в вычислениях. Вместе с тем, достаточно широкий класс практических задач укладывается в модель управляющий-рабочие, которая является основной моделью приложения в ГСПК. К этому классу относятся многие переборные и комбинаторные задачи, моделирование методом Монте-Карло, задачи идентификации и многие другие. Для таких задач использование ГСПК оправдано и позволяет разгрузить суперкомпьютеры и сервисные гриды. Резюмируя можно сказать, что грид-системы персональных компьютеров являются дешевой альтернативой суперкомпьютерам и сервисным гридам и для ряда задач могут их успешно заменять.

### ***BOINC – Основная платформа для разработки систем ГСПК***

BOINC (Berkeley Open Infrastructure for Network Computing) [1] представляет собой платформу с открытым кодом для организации проектов добровольных вычислений. Разработка системы ведется в U.C. Berkeley Spaces Sciences Laboratory (США) исследовательской группой, которая также разрабатывала проект SETI@home. Работа над BOINC была начата в 2002 году с целью создания универсальной программной платформы для проектов подобного рода, которая бы упростила процесс развертывания необходимой инфраструктуры и разработки приложений. Первый проект добровольных вычислений на основе BOINC был запущен в 2004 году. В настоящее время насчитывается более 80 публичных проектов на основе BOINC, делая платформу стандартом де-факто в данной области. Система также часто используется для организации внутренних гридов из персональных компьютеров.

Программное обеспечение BOINC состоит из двух основных частей: серверной, которая обеспечивает функционирование проекта, и клиентской, размещаемой на машинах добровольцев. Каждый проект на основе BOINC создается и функционирует независимо от других проектов, поддерживая собственный центральный сервер и веб-сайт. Для участия в проекте добровольцы устанавливают на своих компьютерах универсальный клиент BOINC, распространяемый с сайта платформы и доступный для всех основных операционных систем.

Проект на основе BOINC идентифицируется с помощью уникального адреса (URL), являющегося одновременно главной страницей веб-сайта



проекта и точкой входа для клиентов. В рамках проекта могут выполняться несколько приложений, состав которых может со временем изменяться. В рамках BOINC предусмотрена поддержка широкого класса вычислительных приложений, которые могут быть сформулированы в виде совокупности из большого количества независимых заданий. Платформа обеспечивает надежное и эффективное выполнение приложений в динамичной распределенной среде, реализуя механизмы планирования заданий, передачи данных и обработки отказов. Существующие приложения на таких языках, как C, C++ и FORTRAN, могут быть использованы в рамках BOINC без или с минимальной модификацией их кода.

После установки клиента BOINC, добровольцы могут подключить клиент к одному или нескольким проектам. При этом пользователь может указать то, каким образом в процентном отношении ресурсы его компьютера будут распределены между данными проектами. При подключении к проекту пользователь фактически дает разрешение на выполнение на своей машине любых исполняемых файлов, загруженных с сервера проекта. Несмотря на то, что BOINC обеспечивает определенную изоляцию выполняемого на клиентской стороне кода (sandboxing), в общем случае пользователь самостоятельно должен убедиться в подлинности, защищенности и научной значимости проекта.

Для учета индивидуального вклада каждого из добровольцев в проект в BOINC реализован механизм учета кредитов, которые вычисляются пропорционально процессорному времени, использованному для выполнения заданий проекта. В BOINC также предусмотрена возможность экспорта информации о кредитах пользователя на уровне отдельного проекта. Это, а также поддержка глобальной идентификации пользователя по его адресу электронной почты, позволяет агрегировать и делать доступной сводную статистику кредитов пользователя по всем проектам.

### ***Состояние исследований по тематике ГСПК в России***

#### ***Проект DEGISCO***

Целью проекта седьмой рамочной программы DEGISCO[2], стартовавшего в 2010 и завершившегося в 2012 году, было развитие и популяризация технологий ГСПК по всему миру. В проекте принимали участие 12 партнеров из 11 различных стран. В рамках проекта было проведено много мероприятий (выставок, выступлений, семинаров и тренингов), основной целью которых было распространение информации о технологиях ГСПК и обмен опытом использования ГСПК.

Основным результатом работы проекта явилось создание международной федерации ГСПК[3]. Международная федерация грид-систем из персональных компьютеров является организацией, объединяющей людей из различных компаний, университетов и исследовательских институтов, заинтересованных в использовании вычислительной мощности такого типа и желающих обменяться опытом

друг с другом. Своим участникам федерация предоставляет несколько услуг: личные встречи на семинарах, доступ к документации, форум, веб-портал и консультации экспертов.

В настоящее время в состав федерации входят 46 организаций. Российское отделение представлено четырьмя организациями:

- Институт прикладных математических исследований Карельского научного центра РАН,
- Институт вычислительных технологий сибирского отделения РАН,
- Институт системного анализа РАН,
- Институт динамики систем и теории управления сибирского отделения РАН.

Как минимум в трех из перечисленных организаций существуют научные коллективы, ведущие исследования по тематике ГСПК. Содержание этих исследований будет рассмотрено подробнее далее.

#### ***Исследования, проводимые ИПМИ Кар НЦ РАН***

В Институте прикладных математических исследований проводятся исследования по нескольким направлениям, связанным с ГСПК. Во-первых, это – разработанный интерактивный веб-портал для работы с BOINC. Этот портал облегчает развертывание BOINC-приложения, инициацию вычислений, получение результатов расчетов. С использованием разработанного портала было создано несколько распределенных приложений. Это, например, приложение, направленное на проведение кванто-механических расчетов с использованием пакета Firefly [4]. Также был реализован распределенный вариант обработки больших объемов данных[5].

#### ***Исследования, проводимые в ИДСТУ СО РАН***

Основным достижением Института динамики систем и теории управления является разработка и поддержание проекта SAT@home[6,7], который на данный момент является наиболее успешным российским проектом в области добровольных вычислений. На момент написания статьи в проекте принимают участие 2755 добровольцев, общее число компьютеров, подключенных к проекту составляет 7087, средняя реальная производительность составляет 1.169 TFlops.

SAT@home – исследовательский проект на платформе BOINC, использующий мощности распределенной вычислительной системы для решения трудных и практически важных задач (обращения дискретных функций, дискретной оптимизации, биоинформатики и т.д.), которые могут быть эффективно сведены к задаче о выполнимости булевых формул. Сначала в рамках данного проекта успешно решалась задача криптоанализа шрифта A5/1. Были улучшены существующие результаты в этой области [8]. На данный момент в проекте решаются задачи поиска ортогональных пар диагональных латинских квадратов порядка 9 и 10. В ближайшем будущем авторы проекта рассчитывают найти тройку взаимно

ортогональных латинских квадратов порядка 10 либо доказать отсутствие такой тройки.

### ***Исследования, проводимые в ИСА РАН***

Институт системного анализа РАН является одним из партнеров в проекте DEGISCO и членом международной федерации грид-систем из персональных компьютеров с момента ее основания. На базе Центра грид-технологий и распределенных вычислений ИСА РАН функционируют тестовая, учебная и рабочая вычислительные инфраструктуры BOINC.

Учебная инфраструктура используется для обучения студентов в рамках практических занятий и при выполнении курсовых и дипломных работ. Тестовая инфраструктура используется для апробации BOINC-проектов перед их размещением в рамках рабочей инфраструктуры.

В рамках рабочей инфраструктуры в настоящий момент функционирует проект [OPTIMA@home](http://OPTIMA@home) [9]. Данный проект направлен на решение задач глобальной оптимизации в распределенной среде. В настоящее время в рамках проекта решается задача нахождения минимума потенциальной энергии взаимодействия атомов однородного молекулярного кластера.

Коллектив центра грид-технологий и распределенных вычислений проводит различные мероприятия, направленные на популяризацию технологии ГСПК в России. Регулярно проводятся тренинги и семинары по технологиям ГСПК. На конференции Грид'2012 в Дубне была проведена секция, посвященная тематике ГСПК. Также регулярно тематика ГСПК представляется на ежегодной выставке Softool в рамках экспозиции Российской академии наук. Поддерживается функционирование сайта российского отделения международной федерации грид-систем из персональных компьютеров [10], на котором размещается информация о проводимых в России исследованиях в области ГСПК.

### ***Российские добровольцы, предоставляющие компьютерные ресурсы***

В России существует сообщество пользователей, которые предоставляют ресурсы своих компьютеров для добровольных вычислений. Сайт сообщества [11] регулярно обновляется. Недавно вышла статья участников [12], посвященная деятельности, успехам и задачам сообщества. В настоящее время зарегистрировано более 100 команд российских участников из которых не менее 12 являются активными и принимают участие в различных проектах.

Роль добровольцев, предоставляющих ресурсы персональных компьютеров для различных проектов добровольных вычислений очень важна. В будущем планируется расширение взаимодействия и сотрудничества между исследователями, которые создают проекты BOINC и сообществом добровольцев. В настоящее время такое сотрудничество уже проводится в рамках проекта SAT@home – российские добровольцы не только предоставили большие вычислительные ресурсы, но и помогли

существенно улучшить технологические стороны проекта. В работе секции, посвященной ГСПК, которая прошла в рамках конференции GRID'2012 летом 2012 года, приняли участие представители российского сообщества добровольцев, что также может быть расценено как важный шаг на пути к сотрудничеству.

### ***Перспективы развития грид-систем из персональных компьютеров в России***

У технологий грид-систем из персональных компьютеров в России существует большой потенциал развития. Можно предположить, что с ростом пропускной способности домашнего интернета, увеличением мощности персональных компьютеров, число компьютеров, присоединяемых к проектам, будет расти, что приведет к росту суммарной вычислительной мощности российских сетей ГСПК. Определенные надежды в последнее время связываются с развитием мобильного интернета. В настоящее время мобильные телефоны и планшеты обладают достаточно мощными процессорами и могут быть использованы для вычислений.

Участники российского отделения Международной федерации грид-систем из персональных компьютеров предполагают продолжать активную деятельность по популяризации технологий ГСПК. Целью этой работы является распространение и внедрение этих технологий в практику работы научных и производственных коллективов, что позволит существенно расширить мощность используемых ими вычислительных ресурсов. Как следствие ожидается повышение точности и объема проводимых расчетов, что открывает возможности для получения новых научных результатов.

Предполагается создание на базе проекта OPTIMA@home зонтичного Российского проекта добровольных вычислений, в рамках которого различные исследовательские и производственные коллективы смогут размещать свои приложения. Предполагается, что такой подход позволит консолидировать больше вычислительных ресурсов, чем, если бы каждое из приложений размещалось бы в отдельном проекте.

Другим важным направлением развития ГСПК является создание гибридных вычислительных систем, в состав которых входят как ресурсы добровольцев, так и суперкомпьютерные и грид-ресурсы. Такие системы будут обладать рекордной вычислительной мощностью и их использование открывает перспективы для решения сложных задач, которые не были решены до этого.

### **Литература**

1. Anderson D. P. Boinc: A system for public-resource computing and storage. In R. Buyya, editor, Fifth IEEE/ACM International Workshop on Grid Computing, pages 4-10, 2004
2. Проект DEGISCO <http://degisco.eu/>
3. Международная федерация ГСПК <http://desktopgridfederation.org/>
4. Ивашко Е.Е., Никитина Н.Н. Организация квантовохимических расчетов с

использованием пакета Firefly в гетерогенной Грид на базе BOINC // Научный сервис в сети Интернет: экзафлопсное будущее. Труды Международной суперкомпьютерной конференции (Абрау-Дюрсо, 19-24 сентября 2011 г), 2011. С. 178–181.

5. Ивашко Е.Е., Головин А.С. Методы Data Mining для анализа больших массивов данных в гетерогенной грид на базе BOINC. Труды Международной суперкомпьютерной конференции (Абрау-Дюрсо, 17-22 сентября 2012 г), 2012.

6. Проект добровольных вычислений SAT@home <http://sat.isa.ru/pdsat/>

7. Заикин О.С., Посыпкин М.А., Семёнов А.А., Храпов Н.П. Организация добровольных вычислений на платформе BOINC на примере проектов OPTIMA@home и SAT@home // CAD/CAM/CAE Observer # 3 (71) / 2012. С. 87-92.

8. Posypkin, Mikhail and Semenov, Alexander and Zaikin, Oleg (2012) Using BOINC desktop grid to solve large scale SAT problems. Computer Science, 13 (1). pp. 25-34. ISSN 1508-2806.

9. Проект добровольных вычислений SAT@home <http://boinc.isa.ru/dcsdg/>

10. Сайт русского отделения международной федерации ГСПК <http://desktopgrid.ru/>

11. Сайт сообщества российских участников проектов BOINC <http://www.boinc.ru/>

12. Андреев А., Манзюк М., Ватутин Э. Весь мир как суперкомпьютер. Троицкий вариант, 16(110), С. 7-9

## ***Сравнительный анализ инфраструктур для облачных вычислений***

### ***Введение***

Благодаря парадигме IaaS становится возможным виртуализировать практически любую аппаратную составляющую, что является безусловно полезным при желании предоставить пользователю значительные вычислительные ресурсы удаленно. Такие задачи возникают и в процессе обучения информационным технологиям: удобно становится один раз установить и настроить, скажем, образ операционной системы с нужными учебными примерами, а затем благодаря интерфейсу, предоставляемому облаком, запускать этот образ для каждого студента. Подобное «учебное» облако становится решением проблем, связанных с бесконечной настройкой и переносимостью учебных сред. Для того, чтобы внедрить подобное облако, необходимо прежде всего ознакомиться с существующими облачными решениями.

В классическом варианте облачной вычислительной системы можно выделить шесть основных компонент.

Во-первых, имеются физические машины и их операционные системы. Мы выделяем их, потому что: а) если процессоры, используемые в машинах, не поддерживают технологию аппаратной виртуализации, то мы можем говорить лишь о паравиртуализации, и это существенно снижает как вычислительные способности системы, так и выбор программного обеспечения в дальнейшем; б) системы с открытым кодом в большей степени, чем другие, должны быть достаточно гибки для работы со многими гостевыми ОС (а коммерческие могут быть узко специализированы на работу с определенным набором программных компонент).

Во-вторых, сетевая компонента. Использование сети влечет за собой DNS, DHCP и организацию подсетей физических машин. Это также включает виртуализацию внутри сети для того, чтобы дать каждой виртуальной машине уникальный MAC адрес. Все компоненты должны быть соответственно сконфигурированы для работы с виртуальными узлами в той же степени, что и с физическими.

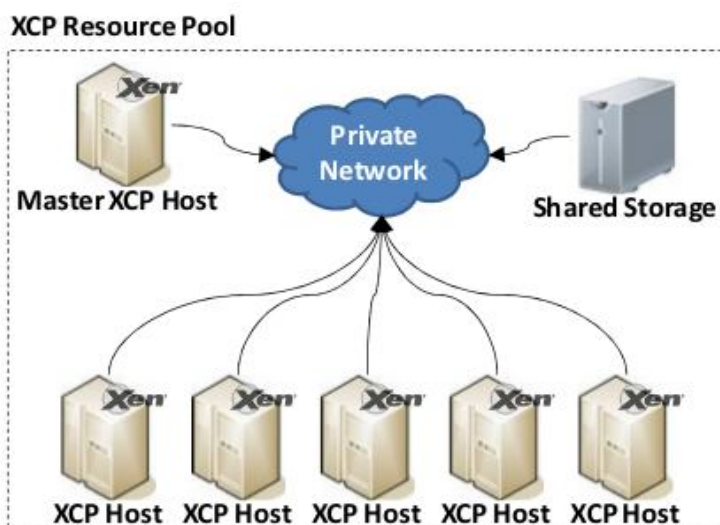
В-третьих, использование гипервизора. Гипервизор позволяет работать виртуальным машинам. В добавок к мониторингу виртуальных машин, гипервизоры используют средство libvirt для того, чтобы контролировать запуск и останов машин. Кроме того, каждый гипервизор должен быть запущен и настроен индивидуально под каждую машину.

Четвертая компонента облачной инфраструктуры – образы виртуальных машин, хранящиеся на виртуальном «жестком диске» – репозитории, охватывающем все образы всех машин, запущенных когда-либо в облаке. Естественно, необходимо предусмотреть и архивацию, и надлежащее обслуживание репозитория.

В-пятых, необходимый интерфейс для пользователей.

И, наконец, облачная среда как таковая – система, связывающая воедино все шесть компонент, распределяющая ресурсы пользователям, обслуживающая сеть и так далее.

### ***Сравнение облачных инфраструктур***



*Рисунок 1. Архитектура Xen Cloud Platform*

### **Xen Cloud Platform**

Гипервизор Xen – готовое решение для виртуализации оборудования, который предоставляет промежуточный слой между аппаратным обеспечением сервера и его операционной системой. Гипервизор Xen поддерживает несколько «виртуальных серверов», работающих под разными операционными системами на единственном физическом сервере. Путь Xen проделали многие современные облачные сервисы, такие как Amazon EC2, Nimbus и Eucalyptus.

В настоящее время Xen представил Xen Cloud Platform (XCP) как способ облачной виртуализации инфраструктуры. Однако, в отличие от других облачных проектов с открытым кодом, XCP не предоставляет полной архитектуры для облачных вычислений. Их цель – предоставить средство с автоматической конфигурацией для поддержки облачных платформ.

Архитектура XCP основана на XCP хостах, которые ответственны за обслуживание гостевых операционных систем. В соответствии с Xen.org, эти хосты объединены в общее хранилище ресурсов, и, используя это хранилище, виртуальные машины могут быть запущены или перезапущены на любом хосте. Мастер-хост XCP обеспечивает интерфейс

администратора и пересылает управляющие сообщения другим хостам.

К несомненным достоинствам Xen можно отнести гибкость настройки, надежный гипервизор с огромными возможностями, такими как динамическое изменение ресурсов для каждой виртуальной машины; поддержку миграции образов в процессе работы (live migration). XCP может быть комбинирован с другими инфраструктурами, прежде всего с OpenStack, который использует XenServer в качестве координирующего узла.

К недостаткам XCP относится, прежде всего, сложность внедрения, необходимость ручной настройки практически всех компонентов. С другой стороны, буквально за два года развития XCP документация стала существенно подробнее, многие дистрибутивы Linux стали поддерживать Xen и XCP.

### **OpenNebula**

Идея проекта OpenNebula с открытым кодом – частное или корпоративное облако с большими вычислительными возможностями. Позиционируется как замена стандартным дата-центрам для компаний.

Интерфейс, предоставляемый OpenNebula – обертка вокруг XML-RPC, который тоже может быть использован. Настраиваемость OpenNebula касается как администраторов, так и пользователей облака. С точки зрения администратора, богатые возможности для переноса виртуальных машин имеются у OpenNebula благодаря файловой системе NFS, хранящей все образы дисков облака. Централизация, полученная в OpenNebula, хороша для ее администрирования.

Пользователь в OpenNebula имеет возможность задавать конфигурационный файл при порождении виртуальной машины, за счет чего можно получить, вообще говоря, любые ресурсы, которые может предоставить облако. Обратной стороной является высокая вероятность ошибочной конфигурации виртуальной машины.

Благодаря NFS вычислительные узлы могут не иметь много места на их хранилищах. Однако, сама NFS может стать узким горлышком в системе, требуя много ресурсов – стандартная установка на 64 узла предполагает использование одного терабайта памяти.

К тому же, NFS не предполагает шифрования трафика внутри файловой системы, что может сильно сказаться на безопасности.

### **Ubuntu Enterprise Cloud**

Ubuntu Enterprise Cloud (UEC) – программная среда для создания облачной инфраструктуры, поддерживаемая компанией Canonical. Главная цель, которую ставили перед собой разработчики UEC – простота развертывания облачной инфраструктуры.

UEC определяет три уровня безопасности: идентификация и аутентификация, изоляция сетей и изоляция вычислительной машины как таковой. Уровень идентификации и аутентификации обеспечиваются локально генерирующимися сертификатами X.509, изоляция сети важна

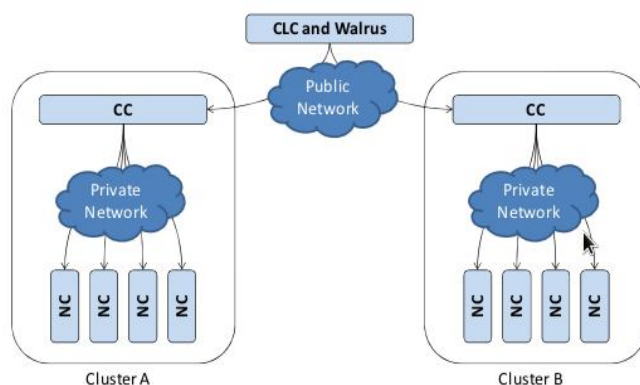


для предотвращения резких скачков трафика, и, наконец, уровень изоляции вычислительной машины состоит из сетевой изоляции, изоляции операционной системы машины и изоляции машины-гипервизора.

Инфраструктура UEC была основана на архитектуре Eucalyptus вплоть до версии Ubuntu 11.10 Server, сейчас используется архитектура OpenStack.

### **Eucalyptus**

Eucalyptus был разработан как open-source ответ на коммерческое облако EC2. Eucalyptus использует в качестве интерфейса для пользователя программу, которая называется euca2ools, которая очень похожа на программу из облака Amazon, использующуюся в качестве front-end для пользователей.



**Рисунок 2:** Компоненты системы Eucalyptus

Полная архитектура и спецификация Eucalyptus была опубликована, и много документации по Eucalyptus в настоящее время доступно в сети Интернет. Основные компоненты среды Eucalyptus:

1. Node Controller (NC). Эта компонента запускается на всех машинах, которые будут в дальнейшем использоваться для запуска гостевых ОС. NC отвечает за очередность и контроль системного программного обеспечения (собственно, ОС и гипервизора) и за выполнение запросов «отвечающего за него» Cluster Controller (CC). Роль NC во всем этом – сбор системной информации, такой как физические ресурсы узла и состояния виртуальных машин на узле. Всю эту информацию он отправляет на CC. Также NC помогает CC в авторизации, исполнении запросов гипервизора.
2. Cluster Controller (CC). Эта компонента обычно выполняется на машине, являющейся кластером, или на любой машине, имеющей две связи – со множеством NC и Cloud Controller (CLC). CC отвечает за сбор и отправку информации о виртуальных машинах, запущенных на различных узлах и составляет для них расписание.
3. Storage Controller (Walrus). Эта компонента представляет собой сервис хранилища данных, который предоставляет механизм для доступа и содержания образов виртуальных машин и пользовательской информации. Walrus основан на технологиях web-сервисов и

полностью совместим с интерфейсом Amazon's Simple Storage Service.

4. Cloud Controller (CLC). Эта компонента является отправной точкой для «входа» в облако для пользователей. Ее основная цель – предоставление и управление виртуализированными ресурсами. CLC ответственен за предоставление менеджерам ресурсов узлов, за составление расписания, и внедрение его в СС.

Вся инфраструктура Eucalyptus подчеркивает его коммерческого вдохновителя. Во-первых, сильное разграничение пользовательского пространства и пространства администратора. Доступ как root позволяет администратору системы делать на узле все, что возможно вплоть до физического уровня. Пользователи системы могут пользоваться только теми инструментами, которые предоставляет web-интерфейс. За некоторым исключением, Eucalyptus предотвращает практически все попытки пользователей как-то вмешаться в работу вычислительного узла.

Администратор Eucalyptus определяет пять доступных конфигураций ресурсов, которые пользователь вправе назначить для своей виртуальной машины.

Конфигурация программного обеспечения предполагает использование децентрализованного подхода, и как кластеры могут быть представлены в такой системе в качестве одного узла, так и отдельные узлы могут быть объединены в кластер.

Eucalyptus поддерживает распределенное хранилище данных, Walrus, которое на самом деле повторяет аналогичное хранилище в технологии от Amazon. Пользователям выделена некоторая часть общего хранилища, которое они могут использовать. Хранилище разделено на структуры, которые «раскиданы» по облаку. Однако, образ виртуальной машины хранится на вычислительном узле. Кроме того, виртуальная машина может быть поднята и на узле хранилища, однако она может работать только обособленно от других машин.

Децентрализованный подход Eucalyptus со множеством кластеров, распределенным хранилищем и локально хранимыми образами дисков способствует включению в облако множества машин.

Основными достоинствами Eucalyptus являются простота внедрения, совместимость с коммерческим облаком EC2, обширная документация и большое сообщество пользователей. К недостаткам можно отнести невысокую гибкость и настраиваемость.

### **OpenStack**

Разработчики OpenStack заявляют те же цели, которые заявляли разработчики OpenNebula – готовые решения для частных и корпоративных облаков, с хорошей масштабируемостью и быстрым доступом к ресурсам. Рассмотрим подробнее компоненты этой архитектуры.

1. Хранилище файлов «Swift» позволяет обмениваться файлами. Здесь лежат рабочие файлы пользователей, хранилище реализует обмен

ими в ходе работы.

2. Репозиторий образов виртуальных машин «Glance». По сути, является интерфейсом для «Swift». Пользователь оперирует образами репозитория, но информация на самом деле хранится в «Swift».
3. Вычислительная компонента «Nova». Дает доступ непосредственно к ресурсам системы по запросу.
4. Пользовательский интерфейс предоставляет компонента «Horizon». Это веб-интерфейс, позволяющий выполнять все функции по администрированию образов виртуальных машин, в том числе и контроль доступа.
5. Модуль авторизации и аутентификации «Keystone» решает эти проблемы для всех компонент OpenStack, а также предоставляет каталог всех сервисов конкретного облака.
6. Сетевая компонента «Quantum» позволяет пользователям создавать свои сети и затем подключать к ним различные компоненты.

OpenStack в современном виде поддерживается компанией Canonical, на базе этой технологии работает Ubuntu Enterprise Cloud, который входит во все серверные версии Ubuntu Server, начиная с 11.10. Стоит заметить, что минимально рекомендуемое число машин – 6, в то время как Eucalyptus можно было установить даже на одну машину.

К преимуществам OpenStack можно отнести современную поддержку крупнейших компаний, легкое внедрение и дальнейшую эксплуатацию, а также хорошую возможность масштабирования.

К недостаткам этой технологии относятся требовательность к ресурсам и сложная архитектура, сочетающая в себе большое число компонент.

### **Заключение**

Рассмотренные выше технологии позволяют реализовывать как большие, сложные системы с привлечением больших вычислительных мощностей, востребованные крупными компаниями, так и небольшие частные облака. Большие возможности настройки позволяют эффективно администрировать имеющиеся ресурсы. Хорошая поддержка позволяет реализовывать практически любые задачи, в том числе и образовательные.

### **Литература**

1. Peter Sempolinski, Douglas Thain. A Comparison and Critique of Eucalyptus, OpenNebula and Nimbus. Париж, 2009.
2. Damien Cerbelaud, Shishir Garg, Jeremy Huylebroeck. Opening The Clouds: Qualitative Overview of the State-of-the-art Open Source VM-based Cloud Management Platforms. Сан-Франциско, 2010.
3. Patrícia Takako Endo, Glauco Estácio Gonçalves, Judith Kelner, Djamel Sadok. A Survey on Open-source Cloud Computing Solutions. Universidade Federal de Pernambuco, 2010.

## **Параллельные и последовательные варианты метода динамического программирования**

### **Постановка задачи о ранце**

Задача о ранце — одна из задач комбинаторной оптимизации. Это название она получила от максимизационной задачи укладки как можно большего числа нужных вещей в рюкзак при условии, что общий объём (или вес) всех предметов ограничен. Подобные задачи часто возникают в экономике, прикладной математике, криптографии. В общем виде задачу можно сформулировать так: из неограниченного множества предметов со свойствами «стоимость» и «вес» требуется отобрать некое число предметов таким образом, чтобы получить максимальную суммарную стоимость при одновременном соблюдении ограничения на суммарный вес.

Пусть имеется  $n$  предметов, каждый из которых имеет ценность  $p_j > 0$  и вес  $w_j > 0$ ,  $j = 1, 2, 3, \dots, n$ . Имеется ранец, грузоподъемность которого есть  $C$ , при этом  $\sum_{j=1}^n w_j > C$ , т.е. все предметы в ранец положить невозможно. Необходимо положить в ранец набор предметов с максимальной суммарной ценностью. Введем переменных:

$$x_j = \{0, 1\},$$

0 = если предмет с номером не кладется в ранец,

1 = если предмет с номером кладется в ранец,

$j = 1, 2, 3, \dots, n$ .

После введения этих переменных суммарная ценность и грузоподъемность соответственно имеют вид:

$$f(x_1, x_2, x_3, \dots, x_n) = \sum_{j=1}^n p_j x_j,$$

Поэтому задача о ранце имеет вид:

$$\begin{aligned} \sum_{j=1}^n p_j x_j &\rightarrow \max, \\ \sum_{j=1}^n w_j x_j &\leq C, \\ x_j &\in \{0, 1\}, j = 1, 2, 3, \dots, n. \end{aligned}$$

Естественно предположить, что  $p_j > 0, 0 < w_j < C$ ,  $j = 1, 2, 3, \dots, n$ . Множество допустимых решений этой задачи — это множество  $n$ -мерных

булевых векторов  $x = (x_1, x_2, x_3, \dots, x_n)$ , удовлетворяющих условию  $\sum_{j=1}^n w_j x_j \leq C$ .

### **Алгоритм динамического программирования решения задачи о ранце**

Алгоритм динамического программирования решает задачу с помощью рекуррентного соотношения, называемого уравнением Беллмана.

$$f_j(i) = \max(p_j + f_{j-1}(i - w_j), f_{j-1}(i))$$

$$j = 1, 2, 3, \dots, n; i = 1, 2, 3, \dots, C;$$

$$f_j(k) =$$

Максимальное значение для грузоподъемности  $i$  и  $j$  переменных,

$p_j$  = цена предмета с номером  $j$ ,

$w_j$  = вес предмета  $j$ ,

$i$  = текущая грузоподъемность.

Решением задачи является вектор из нулей и единиц, задающий значения переменных.

Методы динамического программирования основаны на принципе оптимальности Беллмана. Табличный алгоритм динамического программирования реализует принцип оптимальности с помощью двумерной матрицы размерности . Необходимым условием применимости этого алгоритма является целочисленность коэффициентов задачи.

**Табл. 1. Таблица алгоритма**

j - предметы	1				$f_1(3)$			
	2				$f_2(3) = \max(p_2 + f_1(3 - w_2), f_1(3))$			
	3							
	.							
	.							
	.							
	n							
		0	1	2	3	.	.	C
					i - грузоподъемность			

В таблице столбцы обозначают вес от нуля до максимального значения (т.е. грузоподъемность), а строки обозначают количество предметов. Прямой ход алгоритма динамического программирования решения задачи о ранце использует уравнение Беллмана для вычисления очередного значения в ячейке таблицы. Строки заполняются сверху вниз, каждая строка – слева направо. Рассмотрим пример задачи о ранце

$$3x_1 + 2x_2 + 2x_3 + 7x_4 \rightarrow \text{Max}$$

$$2x_1 + 3x_2 + 2x_3 + 3x_4 \leq 6$$

Таблица, заполненная в результате работы метода динамического программирования приведена ниже (Табл. 3).

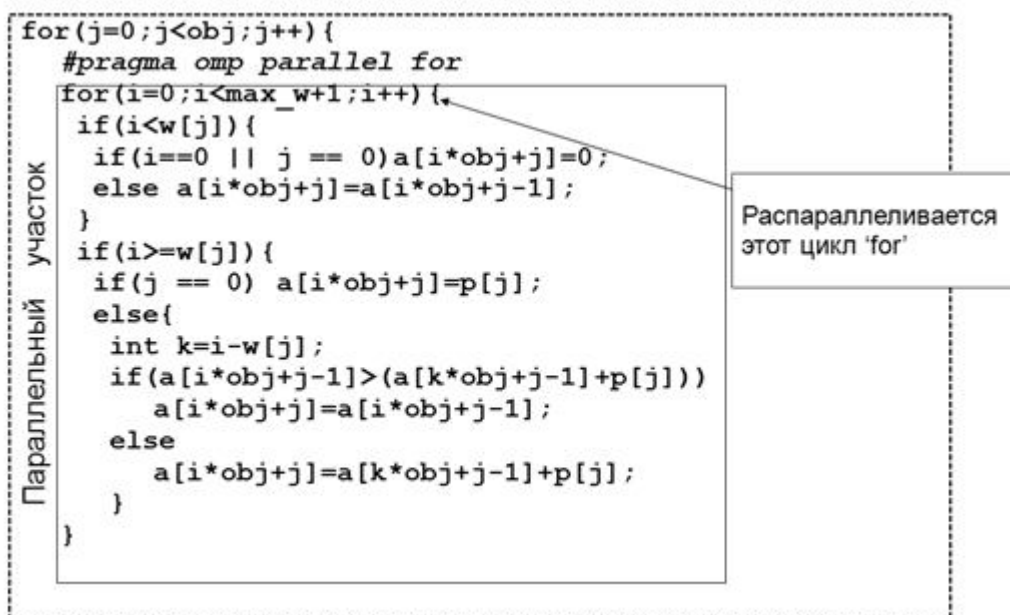
**Табл. 2.** Таблица метода динамического программирования для примера

j - предметы	1	0	0	3	3	3	3	3
	2	0	0	3	3	3	5	5
	3	0	0	3	3	5	5	5
	4	0	0	3	7	7	10	10 (Результат)
	0	1	2	3	4	5	6	
i - грузоподъемность								

Максимальное значение целевой функции оказывается в правом верхнем углу.

### **Задача о ранце на OpenMP**

Для параллельной реализации применялась технология OpenMP. OpenMP является реализацией многопоточности. На параллельных участках главный поток разделяется на определенное количество подчинённых тредов и задача распределяется между ними.



**Рис. 1.** Параллельный участок в коде

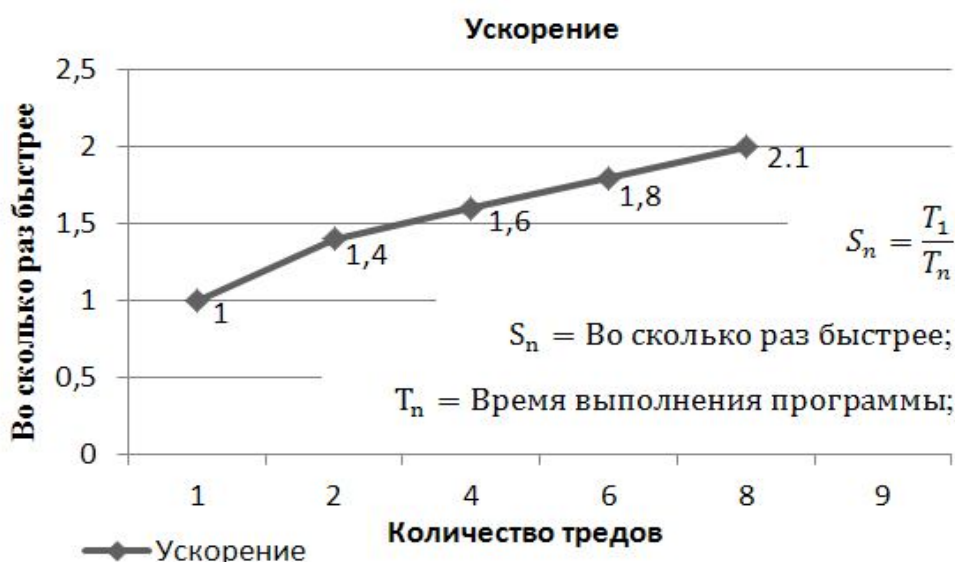
Для распараллеливания основного цикла прямого хода применяется директива parallel for. В результате итерации цикла распределяются между параллельными потоками.

### **Эксперименты**

Эксперименты выполнялись с  $n=5000$  переменными на компьютере с двумя процессорами Intel Core 2 Quad E5335 2,00 ГГц и 8 Гб памяти. Эксперименты проводились для различного числа потоков. Графики зависимости времени выполнения от числа потоков приведены на Рис. 2 и



**Рис. 2.** График времени выполнения задачи



**Рис. 3.** График ускорения

Здесь показан график ускорения. Видно, что чем больше количество потоков, тем выше ускорение. Графики для ускорения и времени решения демонстрируют эффективность предложенного подхода к распараллеливанию алгоритма динамического программирования.

#### **Алгоритм динамического программирования со списком**

Алгоритм динамического программирования со списком использует список пар, первый элемент которых содержит суммарную стоимость положенных в ранец предметов, а второй – оставшийся лимит по весу, т.е. разность между грузоподъемностью ранца и суммарным весом положенных в него предметов. Концепция доминирования позволяет



сокращать перебор в методе динамического программирования за счет удаления неперспективных состояний.

Пара  $(a, b)$  доминирует пару  $(c, d)$ , если  $a \geq c$  и  $b \geq d$ . Очевидно, что при условии, что мы можем докладывать в ранец одинаковые предметы, пара  $(c, d)$  никогда не приведет к лучшим результатам по сравнению с парой  $(a, b)$ . Поэтому пару  $(c, d)$  можно удалить из списка.

Приведем примеры доминирования. Например, пусть имеется две пары  $(4, 7)$  и  $(3, 5)$ , тогда  $(4, 7)$  доминирует пару  $(3, 5)$ , потому что  $(7 \geq 5)$  и  $(4 \geq 3)$ . Поэтому пару  $(3, 5)$  можно удалить. Другой пример, если у нас есть две пары  $(13, 11)$  и  $(15, 10)$ . Тогда  $(13, 11)$  не доминирует  $(15, 10)$  и  $(15, 10)$  не доминирует  $(13, 11)$ . В этом случае обе пары сохраняются в списке.

При  $j=0$  имеем список  $L_0 = (0, C)$ , состоящий из одной пары с суммарной стоимостью положенных предметов, равной нулю, и остаточной грузоподъемностью, равной грузоподъемности ранца. Последующий  $L_j$  список получается из списка  $L_{j-1}$  в два этапа. На первом этапе  $w_j$  вес и ценность  $p_j$  пункта  $j$  добавляется к весу и ценности всех пар, содержащихся в списке  $L_{j-1}$ . Получаем новый список  $L'_{j-1}$ . Это покомпонентное дополнение будем обозначать через

$$L'_{j-1} = L_{j-1} \oplus (p_j - w_j)$$

Все пары вида  $(a, b)$ , где  $b < 0$ , удаляются из списка, т.к. они не удовлетворяют ограничению по весу. На втором этапе два списка  $L_{j-1}$  и  $L'_{j-1}$  объединяются для получения  $L_j$ . При этом происходит исключение доминируемых пар.

Алгоритм динамического программирования со списком:

$$L_0 = (0, C)$$

Для  $j=1$  до  $n$ ;

$$L'_{j-1} = L_{j-1} \oplus (p_j - w_j) \text{ //добавить для всех элементов в } L_{j-1}$$

Удалить все пары  $(\bar{w}, \bar{p}) \in L'_{j-1}$  с условием  $\bar{w} > C$ .

$$L_j = \text{MergeLists}(L'_{j-1}, L_{j-1}) \text{ //объединение двух списков.}$$

Возвращать возможные пары в  $L_n$ .

Рассмотрим пример

### **Эксперименты для динамического программирования со списком**

В экспериментах выполнялись десять задач с 500 переменными на компьютере с двумя процессорами Intel(R)Core(TM) 2 Duo T5670 1,80 ГГц и 4 Гб памяти.

$$\begin{aligned} 3x_1 + 2x_2 + 2x_3 + 7x_4 &\rightarrow \text{Max} \\ 2x_1 + 3x_2 + 2x_3 + 3x_4 &\leq 6 \end{aligned}$$

**В данном случае коэффициенты  $p_j$  и  $w_j$  генерируются случайным образом в диапазоне  $[1, R]$ . Во всех случаях  $p_j$  и  $w_j$  распределяются равномерно в заданном интервале с диапазоном  $R=10000$ .**



**Табл. 3. Таблица метода динамического программирования со списком для примера**

Условие начало					
Шаг - 1	Список - 1	(0, 6)			
Добавление ( $p_1, -w_1$ )					
Шаг - 2	Список - 1	(0, 6)			
	Список - 2	(3, 4)			
	Список - 3	(0, 6)	(3, 4)		
Добавление ( $p_2, -w_2$ )					
Шаг - 3	Список - 1	(0, 6)	(3, 4)		
	Список - 2	(2, 3)	(5, 1)		
	Список - 3	(0, 6)	(3, 4)	(5, 1)	
Добавление ( $p_3, -w_3$ )					
Шаг - 4	Список - 1	(0, 6)	(3, 4)	(5, 1)	
	Список - 2	(2, 4)	Область диаграммы		
	Список - 3	(0, 6)	(3, 4)	(5, 2)	
Добавление ( $p_4, -w_4$ )					
Шаг - 5	Список - 1	(0, 6)	(3, 4)	(5, 2)	
	Список - 2	(7, 3)	(10, 1)		
	Список - 3	(0, 6)	(3, 4)	(7, 3)	(10, 1)

**Табл. 4. Таблица результатов со списками**

Кол. задач	Время выполнения ДП (сек)	Время выполнения ДП со списком (сек)	Шаги ДП	Шаги ДП со списком
1	34	15	50100501	1503546
2	33	15	50100501	1419281
3	32	12	50100501	1203467
4	33	12	50100501	1255444
5	34	10	50100501	1016818
6	33	11	50100501	1234950
7	34	15	50100501	1569151
8	32	18	50100501	1673445
9	33	11	50100501	1265404
10	34	11	50100501	1214087
Средние значения	33.2 сек	13 сек	50100501	1335559.3

### **Заключение**

Алгоритм динамического программирования со списком, во-первых, сокращает перебор, а во-вторых, не требует целочисленности

коэффициентов задачи. Мы видим, что время выполнения задачи алгоритма динамического программирования зависит от  $(N \times C)$ , а время выполнения задачи алгоритма динамического программирования со списками зависит от сложности задачи. В работе реализованы оба варианта алгоритма динамического программирования и проведено экспериментальное сравнение алгоритмов. Результаты сравнения подтверждают более высокую эффективность метода динамического программирования со списком по сравнению с табличным.

### **Литература**

1. Лупин С.А., Посыпкин М.А.. Технологии параллельного программирования.
2. Сигал И.Х., Иванова А.П. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы.
3. Kellerer H., Pferschy U., Pisinger D. Knapsack Problems.–Springer Verlag, 2004–546 p.

## ***Архитектура информационной среды для доступа и использования данных, полученных при зондировании Земли из Космоса***

### **Аннотация**

*В докладе описывается модель архитектуры информационной среды для доступа и использования данных дистанционного зондирования Земли на основе архитектуры предметных посредников с использованием специализированного интерфейса пользователя российской grid-сети.*

Исследование Земли всегда было важной составляющей человеческого существования. В последнее десятилетие спутниковые системы зондирования Земли достигли принципиально нового уровня развития. Они отличаются частотой наблюдений, высокой стабильностью и глобальностью.

Одновременно разрабатывается современная техника съемки Земли из Космоса, и создаются совершенно новые методы обработки спутниковых данных для выявления отдельных характеристик окружающей среды.

Уже накоплены колоссальные объемы данных, грамотное использование которых позволяет, с одной стороны, создавать различные прикладные системы для решения повседневных потребностей общества, с другой — на новом уровне решать многочисленные научные и прикладные задачи, связанные с исследованием состояния и динамики природных объектов.

Однако данные формируются в весьма сложных условиях наблюдения и содержат в себе специфичные искажения. В связи с этим выдвигаются новые требования к вычислительным алгоритмам, задаются специфичные критерии качества обработки данных. Таким образом, космические снимки, их хранение и обработка представляет собой гетерогенную систему. Здесь неизбежна разработка специальных быстрых алгоритмов обработки, целесообразно применение суперкомпьютеров, распределенных систем и grid-систем.

Актуальной представляется задача разработки архитектуры распределенного хранения информации обеспечения устойчивого доступа к информационным ресурсам для возможности их дальнейшего использования.

### ***Данные в ГИС***

Данные ДЗЗ – данные дистанционного зондирования Земли,

полученные с помощью космических аппаратов и применяемые в геоинформационных системах (ГИС) (к примеру, картография). ГИС позволяет нам видеть, понимать, интерпретировать и визуализировать данные в различных форматах, которые показывают отношения, закономерности и тенденции в виде карт, глобусов, отчетов и диаграмм.

На сегодняшний день в России не существует единой инфраструктуры, позволяющей использовать имеющуюся информацию из-за специфики аэрокосмических изображений. Основными особенностями ГИС являются:

- Большое количество данных, получаемых при зондировании Земли и их объем. Для обработки такой информации необходимы новые высокопроизводительные системы обработки информации на основе параллельных вычислений с применением хранилищ данных, высокоскоростной среды передачи данных и использования суперкомпьютеров.
- Нет единой инфотелекоммуникационной и вычислительной среды. Существующие станции приема практически не взаимодействуют между собой.
- Отсутствует обмен данными, что затрудняет развитие научных исследований, требующих интеграции разнородных данных по большим территориям.
- Существующие интернет-порталы доступа к данным ДЗЗ (типа [nakarte.ru](http://nakarte.ru) и [Googlemaps](http://Googlemaps)) непригодны для использования в задачах, требующих точного и полноценного геоинформационного пространства, так как не соответствуют повышенным требованиям к точности географической привязки, своевременности и оперативности обновления информации, индивидуальной настройке.
- Сложность настройки инструментария для приема данных на станциях приема при смене космических аппаратов ДДЗ.
- ГИС технологии мало применяются в хозяйственной деятельности (в сельском хозяйстве, природопользовании, градостроительстве, мониторинге чрезвычайных ситуаций).
- Множество форматов данных ДЗЗ, что создает дополнительную сложность использования таких данных в разнообразных приложениях.

При организации пространственных данных необходимо учитывать следующие принципы[1]:

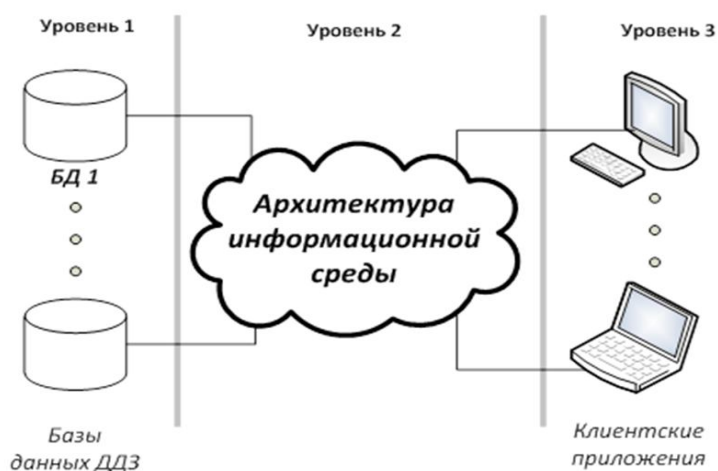
- данные должны создаваться единожды и поддерживаться там, где это можно сделать наиболее эффективно;
- должна быть создана возможность объединения пространственных данных из разных источников;
- к данным должны иметь доступ многие пользователи и приложения;
- пространственные данные, полученные на одном уровне управления,

- должны легко передаваться на все другие уровни;
- пространственные данные, необходимые для эффективной хозяйственной деятельности, должны быть доступны на условиях, не препятствующих их активному использованию;
- организация пространственных данных должна обеспечивать их легкий поиск, оценку пригодности и актуальности для определенных целей и условий их получения.

Для успешного решения проблемы использования данных ДДЗ необходимо преобразовать гетерогенную среду в гомогенную, то есть в систему с едиными методами доступа к различной по своей физической сущности структуре данных.

### ***Уровни информационной системы***

Проблема обмена данными достаточно актуальна и сводится к решению технической стороны вопроса. Для успешного обмена требуется только наличие достаточно полного описания формата, поддержка которого входит в необходимое условие функционирования системы. Большинство распространённых форматов, используемых для обмена данными, являются открытыми. Кроме описаний графических форматов (векторных и растровых) необходимо обращать внимание на спецификации, полезные для использования в геоинформационных системах (базы данных, стандартные и специализированные библиотеки и т.п.)[2].



***Рис.1. Основные уровни информационной среды***

Характерные особенности ГИС и принципы организации пространственных данных вносят дополнительные сложности в реализацию распределенной архитектуры. При создании информационной среды должно быть согласие уровней между собой и к выбираемым решениям или технологиям. Следует выделить три основных уровня (Рис. 1):

1. оборудование вычислительной сети, каналов и линий передачи данных, рабочих мест пользователей, системы хранения данных;
2. операционные системы, сетевые службы и сервисы по управлению

доступом к ресурсам, программное обеспечение среднего слоя;

3. прикладное программное обеспечение, информационные сервисы и среды, ориентированные на пользователей.

Так на первом уровне возникают проблемы связанные с территориально распределенной структурой информационной среды и отсутствием надежных каналов связи. Дополнительно возникают проблемы из-за несогласованности оборудования.

На втором уровне возникает проблемы, связанные с плохой системной интеграцией ИС. Это связано с наличием нескольких несогласованных центров ответственности за развитие технологий и отсутствием утвержденной архитектуры ИС. Архитектура второго уровня ИС на данном этапе представляет собой разрозненные и слабо связанные подсистемы с разными операционными средами, согласованные друг с другом только на уровне закрепления IP-адресов или обмена сообщениями.

Состояние третьего уровня архитектуры ИС можно охарактеризовать следующим образом: в основном завершен переход от локальных программных приложений, опирающихся на локальный набор данных, к корпоративным клиент-серверным информационным системам, обеспечивающим доступ пользователей к оперативным базам данных.

### ***Технологии распределенного хранения***

Распределенные информационные системы невозможно рассматривать вне контекста темы распределенных баз данных. Поэтому были рассмотрены имеющиеся технологии распределенного хранения данных для подбора наиболее оптимальной файловой системы для использования в разрабатываемой архитектуре информационной среды.

Распределенная база данных (Distributed DataBase - DDB) – это такая база данных, которая включает фрагменты из нескольких баз данных, при этом эти фрагменты располагаются на различных узлах сети компьютеров, и, возможно управляются различными системами управления базами данных (СУБД) [3]. Можно выделить такие свойства, которые характеризуют идеальную DDB:

- Локальная автономия (localautonomy) означает, что управление данными на каждом из узлов распределенной системы выполняется локально.
- Независимость узлов (norelianceoncentralsite) означает, что в распределенной системе узлы равноправны и независимы друг от друга, т.к. база данных на каждом узле имеет полный словарь данных и защиту от несанкционированного доступа.
- Прозрачность расположения (locationindependence) говорит о том, что обращение к распределенной базе данных должно выполняться без учета местонахождения необходимых данных.
- Прозрачная фрагментация (fragmentationindependence) определяется возможностью распределенного размещения данных, логически

представляющих собой единое целое. Можно выделить фрагментацию горизонтальную (распределение строк одной логической таблицы в таблицах на других узлах) и вертикальную (столбцы распределены по различным узлам).

- Прозрачное тиражирование (replicationindependence) означает возможность переноса изменений объектов исходной базы данных в базы других узлов средствами, невидимыми пользователю распределенной системы.

- Непрерывные операции (continuousoperation) - это свойство означает и непрерывный доступ к данным.

- Обработка распределенных запросов (distributedqueryprocessing) – возможность производить операции с распределенными базами данных с помощью таких же языковых средств, которые используются для работы над локальной базой данных.

- Обработка распределенных транзакций (distributedtransactionprocessing) – возможность выполнения операций обновления распределенной базы данных (INSERT, UPDATE, DELETE) с применением двухфазового или двухфазного протокола фиксации транзакций, и при этом не повреждая целостность и согласованность данных.

- Независимость от оборудования (hardware independence) - узлами распределенной системы могут служить любые компьютеры.

- Независимость от операционных систем (operationssystem independence) – возможность выбора операционной системы для каждого из узлов.

- Прозрачность сети (network independence) – разнообразие поддерживаемых сетевых протоколов.

- Независимость от баз данных (database independence). Это качество означает, что в распределенной системе могут мирно сосуществовать СУБД различных производителей, и возможны операции поиска и обновления в базах данных различных моделей и форматов.

- Существует множество файловых систем, которые можно классифицировать различным способом [4]. Однако для решения задачи в рамках данного проекта были выбраны наиболее важные свойства DDB, такие, как месторасположение, независимость от операционных систем и оборудования и непрерывность операций. Исходя из приведенных выше критериев, наиболее распространенными системами для распределенного хранения данных являются AFS, Lustre, GPFS, dCache, DPM, Xrootd. Выбор файловой системы также зависит и от программного обеспечения анализа данных.

### ***Функциональная структура информационной среды для доступа и использования данных ДЗЗ***

Одним из главных направлений практической работы было

определение компонентов самой системы и используемого в ней программного обеспечения, включающих применение стандартов и протоколов, их международное согласование, создание «соединяющих» компонентов (точки входа, интерфейса, реестра, потока данных), распознавание пользователей, виртуального хранилища, предоставление ресурсов данных, создание механизмов для научной обработки данных, установка и сопровождение реестров ресурсов и систем поддержки пользователей.

Для возможности использования данных ДЗЗ и данных геоинформационных систем, которые имеют такие особенности, как огромный размер и разнообразие применяемых к данным программных средств, предполагает наличие в разрабатываемой архитектуре следующих средств:

- информационные ресурсы;
- реестры, содержащие описания представляемых ресурсов (реестры метаданных);
- средства программирования;
- интерфейсы для доступа к ресурсам и их использования.

Исходя из перечисленных средств, инфраструктура должна содержать уровни:

1. уровень данных;
2. уровень взаимодействия приложений с информационными ресурсами;
3. уровень способов решения задач (то есть возможность отображения разнообразных ресурсов в соответствии с необходимым результатом).

Таким образом, основными компонентами информационной системы являются (Рис. 2):

1. Виртуальное хранилище данных (ВХД);
2. Реестр метаданных – РМ (будет содержать описания имеющихся ресурсов);
3. Система доступа к данным – СДД (система, позволяющая подключиться к базе данных);
4. Программные средства для планирования и написания программ, необходимых для реализации уровня взаимодействия приложений с информационными ресурсами;
5. Программы доступа к разрозненной информации (преобразование клиентского запроса в запрос на языке данных, получение результата от ресурса на «клиентском» языке);
6. Портал, обеспечивающий возможность взаимодействия задания программ, способов решения задач и отображения результатов.

Так как основная цель создания инфраструктуры является совместный доступ к данным и их активное использование, то система



предоставляет собой совокупность ресурсов (а именно готовых программ решения задач, библиотек методов) и информационных ресурсов (данные ДЗЗ), включает реестры метаданных предоставленных ресурсов, средства программирования в конкретной grid-среде, стандартные интерфейсы для доступа к ресурсам.



**Рис.2. Основные компоненты информационной системы**

Данная архитектура подобна архитектуре среды предметных посредников. Такой подход предполагает использование специальных средств, которые взаимодействуют между приложением и ресурсом на основе определения прикладной области определенной задачи. Основными компонентами промежуточного слоя являются предметные посредники, существующие независимо от информационных ресурсов [5]. Уровень предметных посредников вводится как часть информационных систем, создаваемых для решения научных задач. Для каждой предметной области при решении некоторого класса задач каждый предметный посредник задает свою спецификацию, используя каноническую информационную модель для представления предметной области и унифицированного отображения разнообразных видов моделей информационных ресурсов [6]. В данной работе данный подход реализуется через исполняющую среду и специальные средства – адаптеры.

Исполнительная среда обеспечивает обработку запросов пользователя, поступающих через портал, преобразовывает запросы пользователя в соответствующий программный запрос с помощью адаптера, получает результат и обратно передает пользователю обработанный запрос.

Хранилище метаданных используется для хранения метаданных исполняющей среды, используемой при преобразовании запросов и для сохранения промежуточных результатов запросов, возвращаемых ресурсами, и для выполнения остаточных запросов над этими данными.

Интерфейс представления исполняющей среды выполняет функцию удобного представления выводимой информации, включая сегментацию информационных объектов, их агрегирование и слияние.

Создание Web-интерфейса для работы с геопространственными данными позволит конечному пользователю значительно облегчить поиск и использование данных ДЗЗ в условиях распределенных информационных сред.

Тогда упрощенная схема grid-вычислений с использованием высокоуровневого интерфейса (среды gLite) выглядит так (Рис. 3):



**Рис.3.**Схема взаимодействия пользователя через Web-интерфейс

### **Описание компонентов архитектуры информационной среды для доступа и использования данных ДЗЗ**

Система представляет собой адаптированное программное обеспечение для использования в grid-среде средств доступа к каталогам и реестрам метаданных. В настоящее время для проектирования и поддержки grid-систем используются различные пакеты промежуточного слоя – исполняющей среды, в частности GlobusToolkit 4.0, gLite, AliEn, NorduGrid и ряд других. Применяется также специализированное программное обеспечение, необходимое для разных виртуальных организаций.

Пакет gLite является наиболее полным решением для GRID, включая как базовые низкоуровневые программы, так и ряд служб высокого уровня. gLite распространяется на условиях лицензии открытого кода. В нем интегрированы компоненты из лучших на настоящий момент проектов промежуточного программного обеспечения (ППО), к примеру Condor и GlobusToolkit, компоненты проекта LCG. gLite является одним из лучших базово-инструментальных средств, совместимых с такими планировщиками, как PBS, Condor и LSF. gLite разработан с учетом свойств интероперабельности и содержит базовые службы, облегчающие построение приложений GRID для любых прикладных областей.

Службы gLite соответствуют требованиям SOA (Service Oriented Architecture). Из этого следует, что при необходимости данный продукт можно легко связать с другими GRID-службами, а также, что будет существенно облегчен переход на новые стандарты GRID.

Для реализации архитектуры информационной среды для доступа и использования данных потребуется установка программного обеспечения gLite 3.2 в составе следующих компонентов:

- glite-BDII (Berkeley Database Information Index)
- glite-LFC\_mysql (LCG File Catalogue)
- glite-SE\_dpm\_mysql (Storage Element)
- glite-TORQUE\_client
- glite-UI (User Interface)
- glite-VOBOX (Virtual Organization)
- glite-WN (WorkerNode)

Рассматриваемые компоненты обладают возможностью организации целостной системы предоставления пользователям территориально распределенных информационных, аналитических и вычислительных ресурсов в режиме виртуальных организаций.

### ***Реализация Web-интерфейса пользователя***

Web-интерфейс позволит пользователю работать через браузер и осуществлять следующие действия:

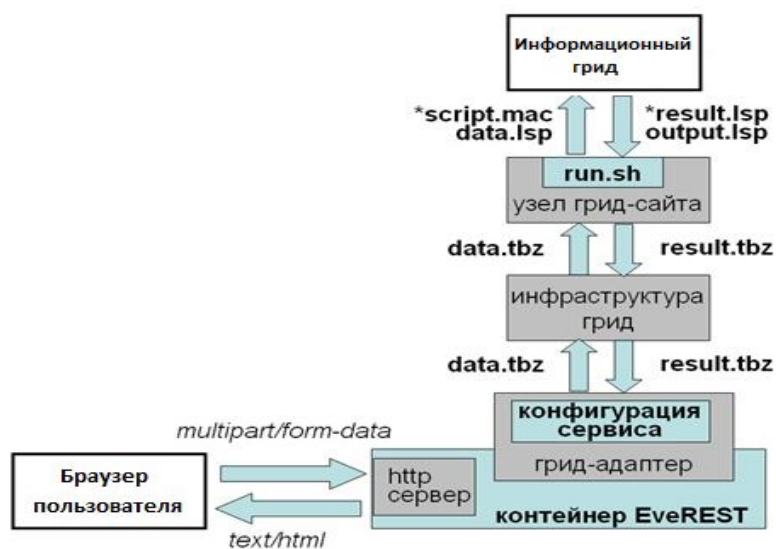
- авторизовать пользователя для запуска комплекса и проводить его сертификацию в виртуальной организации;
- подготавливать задание (включая создание и редакцию начальных данных и конфигурационных файлов) в соответствии с требованиями пакета;
- запускать прикладной пакет в инфраструктуре grid-полигона (при необходимости – на произвольном или избранном grid-ресурсе);
- вести мониторинг выполнения задания (включая останов и перезапуск);
- по завершении – получить результаты запроса.

Доступ к вычислительным и информационным ресурсам grid-сети предоставляется в соответствии с правами, полученными пользователями

при регистрации в качестве членов ВО, а также происходит фиксация использования grid-ресурсов каждым зарегистрированным пользователем. Средства разграничения прав доступа пользователей к ресурсам и сервисам поддерживается с помощью технологии API библиотеки OpenSSL, прокси-сертификатов и атрибутивных сертификатов VOMS.

В среде действует специальная система распределения прав доступа пользователей к данным, программным и вычислительным ресурсам: пользователи, не прошедшие регистрацию, имеют доступ к очень ограниченному подмножеству ресурсов. После прохождения регистрации пользователю назначается открытая политика доступа.

Запрос пользователя соответствует формату, интересующих его данных, только должны быть загружены пользователем через Web-интерфейс в архиве типа gzip, т.е. преобразованы в запрос исполняющей среды.



**Рис. 4.** Архитектура исполняющей среды

Исполняющая среда разрабатываемой инфраструктуры для доступа и использования данных ДЗЗ состоит из конфигурационного файла сервиса для контейнера REST-сервисов EveREST, входящего в состав платформы EveREST, шаблона дескриптора задания и нескольких файлов-сценариев для подготовки исходных данных и обработки результатов работы grid-заданий (Рис.4).

Конфигурационный файл написан на языке JSON в соответствии с правилами, принятыми в EveREST и содержит описание интерфейса сервиса, вместе с параметрами реализации сервиса. Шаблон дескриптора задания представляет собой дескриптор grid-задания, в котором перед загрузкой в grid-сеть производится подстановка обозначенных специальным образом параметров вызова сервиса.

**Назначение Web-интерфейса в информационной среде для доступа и использования данных ДЗЗ**

Разработанный полигон инфраструктуры применим для доступа и использования геопространственных данных, получаемых с помощью зондирования Земли из Космоса.

Формирование и запуск счётных заданий в grid-сеть, контроль хода их выполнения и получение результатов вычислений осуществляется через Web-интерфейс пользователя, скрывающим от пользователя детали и сложности работы самой grid-сети, а непосредственный счёт на вычислительных ресурсах выполняется с использованием конкретного программного обеспечения, на которое направлен запрос. Данное ПО должно быть установлено на рабочих узлах grid-сайтов. Все операции в Web-интерфейсе выполняются в рамках модели безопасности GRID, построенной на цифровых прокси-сертификатах формата X.509, и технологии виртуальных организаций.

Взаимодействие между Web-интерфейсом и ресурсами происходит посредством следующих компонент:

- виртуального интерфейса пользователя (web-браузера);
- программного адаптера интерфейса grid;
- сервисов инфраструктуры grid-сети.

### **Заключение**

Представленная архитектура формулировалась с целью соответствия требованиям универсальности, расширяемости, доступности в использовании. Были проанализированы существующие технологии распределенного хранения разрозненных данных, выявлены наиболее оптимальные файловые системы для использования в разрабатываемой информационной среде.

Изложенные положения были использованы при разработке прототипа архитектуры распределенной гетерогенной информационной среды для доступа и использования данных ДЗЗ с реализацией программного комплекса, который является специализированным интерфейсом пользователя российской grid-сети. Такое решение позволит пользователю в терминах прикладной области составить задание для решения научных задач над множеством неоднородных распределенных информационных ресурсов.

### **Литература**

1. Серебряков В.Б. Региональный центр космического мониторинга // Пространственные данные. 2008. №1. С. 52-55
2. Форматы геоданных – [Электронный ресурс]. URL: <http://www.geocad.ru/soft/formats>
3. Классификация файловых систем. Открытые системы. 2006. – №06 – [Электронный ресурс]. URL: <http://www.osp.ru/os/2006/06/2700700/>
4. Лобанов А.К. Методы построения систем хранения данных. JetInfoOnline. 2003. №7
5. Вовченко А.Е., Калиниченко Л.А., Ступников С.А. Семантический Грид, основанный на концепции предметных посредников. М.: Институт проблем информатики

РАН, 2011. С. 170-191.

6. Архитектура промежуточного слоя предметных посредников для решения задач над множеством интегрируемых неоднородных распределенных информационных ресурсов в гибридной грид-инфраструктуре виртуальных обсерваторий. Информатика и ее применения / Брюхов Д.О., Вовченко А. Е., Желенкова О.П. [и др.] 2008. Т. 2.Вып.1. С. 2-34.

## ***Применение технологии виртуализации для изучения комбинированных вычислительных инфраструктур***

### **Аннотация**

*В настоящее время для изучения и практического освоения новых программных продуктов широко используются технологии виртуализации. На основе технологий виртуализации в рамках одной вычислительной машины полностью моделируется (виртуализуется) работа нескольких других вычислительных машин. Данная статья посвящена применению технологий виртуализации для обучения технологиям интеграции разнородных вычислительных грид-ресурсов. Рассматриваются проблемы, являющиеся специфическими для реализации различных типов грид-инфраструктур в виртуальной среде. Также в рамках статьи описываются современные методы обучения грид-системам и технологиям их интеграции посредством виртуальной среды.*

### **Технологии виртуализации**

Технологии виртуализации позволяют в рамках одного компьютера смоделировать работу одного или нескольких других компьютеров. На каждый из виртуальных компьютеров можно будет установить отдельную ОС. Посредством технологий виртуализации решается широкий спектр учебных и практических задач. Технологии виртуализации позволяют в рамках одного современного компьютера развернуть учебную инфраструктуру из нескольких виртуальных машин.

В настоящее время существует множество как коммерческих, так и свободных реализаций программного обеспечения виртуализации. Программа, осуществляющая виртуализацию на компьютере, будем называть *гипервизором*. Операционную систему, под управлением которой функционируют виртуальные машины, в рамках данной работы будем называть *базовой*. Операционную систему, работающую в виртуальной машине – *гостевой*. По принципу функционирования технологии виртуализации можно разделить на три типа:

- *Полная виртуализация* – такой тип виртуализации, при котором центральный процессор базовой машины моделирует полный набор аппаратного обеспечения виртуальной машины. Недостатком данного типа виртуализации является низкая эффективность работы. Преимуществом данного подхода к виртуализации являются минимальные требования к аппаратно-программной платформе как базовой, так и виртуальной

машины.

- *Паравиртуализация* – тип виртуализации предполагающий модификацию виртуальной операционной системы. При данном подходе в рамках виртуальной машины может быть установлена операционная система, ядро которой специальным образом модифицировано для запуска в виртуальной среде. Необходимость адаптации ядра операционной системы накладывает серьёзные ограничения на область применения данной технологии. Преимуществами использования данной технологии является высокая производительность, гибкое управление работающими виртуальными машинами, возможность запуска виртуальных машин при отсутствии аппаратной поддержки технологии виртуализации.

- *Аппаратная виртуализация* – виртуализация на аппаратном уровне. Реализация аппаратной виртуализации возможна только если процессор и материнская плата базовой машины поддерживают технологию аппаратной виртуализации. Большинство современных процессоров (таких как intel и AMD) имеют встроенную поддержку виртуализации. Недостатком применения данной технологии являются ограничения на аппаратное обеспечение базовой машины. Преимуществом использования данной технологии является максимально возможная эффективность и минимальные ограничения на применение гостевой ОС.

### ***Разновидности вычислительных грид-систем***

В настоящее время вычислительные грид-системы разделяют на два основных типа: грид-системы из персональных компьютеров (ГСПК) и сервисные грид-системы.

*Грид-системы из персональных компьютеров* в первую очередь предполагают, что в качестве вычислительного ресурса будут использоваться домашние или офисные компьютеры. Если взять современный домашний, офисный компьютер или ноутбук, и проследить среднюю по времени загруженность его процессора, то можно увидеть, что процессор в целом использует всего несколько процентов своего вычислительного ресурса, а более чем 90% (а это колоссальные вычислительные мощности) простаивают. Современные персональные компьютеры оснащаются достаточно мощными процессорами, это является необходимым для обеспечения возможности комфортной работы для пользователя в моменты пиковой нагрузки на процессор. Моменты пиковой нагрузки на процессор при стандартном использовании персонального компьютера являются кратковременными, а большую часть времени процессор не является загруженным. Технологии ГСПК позволяют использовать незадействованное при стандартном использовании процессорное время для научных вычислений.

Принцип работы ГСПК состоит в том, что на вычислительных узлах устанавливается и настраивается клиентское программное обеспечение, которое выполняет периодические запросы удаленному серверу на наличие заданий для своей платформы. Если на центральном сервере таковые



задания имеются, то клиентская машина скачивает задание в виде исполняемого файла с необходимыми данными, и запускает его, результат работы приложения возвращается обратно на сервер.

Существует несколько современных реализаций ГСПК (BOINC, XwNep, OurGrid, Condor, X-com, SARD). Наибольшее распространение к настоящему моменту получила платформа BOINC (Berkeley Open Infrastructure Network Computing).

Программное обеспечение BOINC можно разделить на две компоненты: клиентскую и серверную части. Клиентская часть устанавливается на вычислительном узле. В её задачи входит:

1. Подключиться к одному из проектов, к какому именно указывает владелец машины.
2. Запрашивать задания у центрального сервера.
3. Скачивать задания с сервера, если они там есть.
4. Запускать у себя скачанные задания.
5. Результаты работы заданий отсылать обратно на сервер.

Серверная часть программного обеспечения BOINC выполняет следующие действия:

1. Создает задания для пересылки на вычислительные узлы.
2. Отвечает на клиентские запросы, отправляет задания на вычислительные узлы.
3. Получает результаты работы задания и передает их для дальнейшей обработки.
4. Содержит в себе web-сервер для получения информации о проекте через web-интерфейс.

Аналогично распределённое приложение для инфраструктуры BOINC можно разделить на две основные компоненты: клиентскую и серверную части распределённого приложения.

Клиентская часть распределённого приложения и есть исполняемый файл, запускаемый на вычислительном узле. Она выполняет основную вычислительную нагрузку. Серверная часть распределённого приложения создает задания (в терминологии BOINC расчетные блоки — workunits) для клиентских узлов. Как правило, расчётный блок состоит из исполняемого файла клиенткой части, объединённый со специфическим для конкретного задания входного файла с данными. После отправки задания в вычислительную инфраструктуру серверная часть распределённого приложения ждет результатов задания. Получив из инфраструктуры все результаты заданий, серверная часть производит их обработку, и создает единый результат работы распределённого приложения.

В *грид-системах сервисного типа* взаимодействие между поставщиком и потребителем ресурсов осуществляется посредством набора сервисов. Функционирование сервисом обеспечивается специальным программным обеспечением промежуточного уровня (ППО). Основными реализациями ППО сервисного грида являются gLite [1], Globus,

Unicore. Общий для них набор сервисов: безопасность, авторизация, поиск ресурсов, доступ к удалённым данным.

Для обеспечения информационной безопасности в инфраструктуре сервисных грид-систем используется механизм цифровых сертификатов. Таким образом необходимым условием функционирования узлов инфраструктуры сервисного грида является взаимодействие с сертификационными центрами.

### ***Комбинированные грид-инфраструктуры***

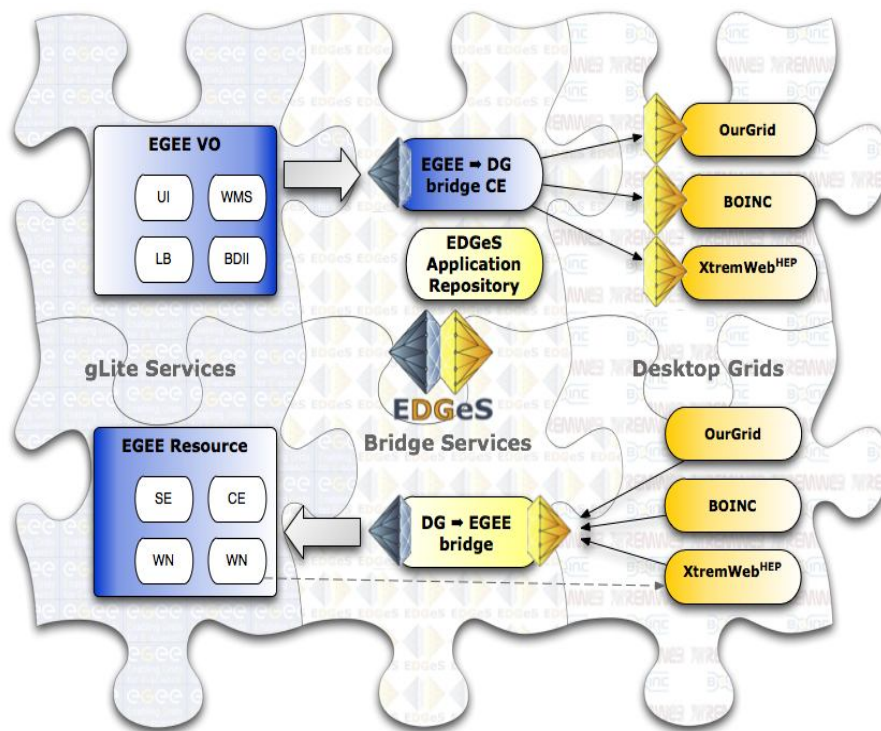
Для интеграции разнородных вычислительных грид-ресурсов в Институте Автоматизации Исследований Академии Наук Венгрии (MTA SZTAKI) было разработано программное обеспечение Generic Grid to Grid Bridge (3G-Bridge) [2]. Программное обеспечение 3G-Bridge позволяет запускать задания из инфраструктуры сервисного грида на выполнение в инфраструктуре грид-систем из персональных компьютеров и наоборот. Механизм мостов позволяет осуществлять интеграцию сервисных гридов и ГСПК на системном уровне, т.е. прозрачным для пользователя образом. На данный момент этот подход реализован для связи грид-инфраструктуры EGEE/EGI с несколькими ГСПК (Рис. 1). Суть подхода состоит в специальном программном компоненте, который, опираясь на абстрактное понятие задания, может быть использован для интеграции двух грид-систем. По выполняемым функциям интегрирующее программное обеспечение можно подразделить на два типа:

- *Мост EGEE  $\Rightarrow$  DG*, обеспечивающий запуск заданий сервисного грида в инфраструктуре ГСПК. Данное соединение функционирует как Computing Element (CE) сервисного грида, где задания вместо вычислительных узлов направляются в инфраструктуру грида из персональных компьютеров (BOINC, XWHEP, OurGrid). Взаимодействие различных типов грид-систем обеспечивается тремя основными программными компонентами:

1. Функционирующий на стороне gLite модифицированный Computing Element, который отправляет принятые из инфраструктуры сервисного грида задания на удалённый мост. Данный CE поставляется в качестве модуля YAIM, и может быть установлен и настроен вместе с другими компонентами gLite.
2. На стороне сервера ГСПК функционирует специальный адаптер, отвечающий за получение заданий, их преобразование для новой инфраструктуры, и выполнение.
3. Репозиторий приложений (Application Repository — AP), содержащий информацию о всех приложениях, проходящих через данный мост.

- *Мост DG  $\Rightarrow$  EGEE*, позволяющий, наоборот, запускать задания ГСПК в инфраструктуре EGEE. Поскольку принцип работы и основное программное обеспечение зависит от типа подключаемой инфраструктуры ГСПК, для каждой из них создана отдельная реализация моста:

1. **Мост BOINC  $\Rightarrow$  EGEE**, который функционирует как клиент BOINC, отправляя скачанные задания в виртуальную организацию EGEE. В инфраструктуре сервисного грида задание запускается специальной программой (jobwrapper), которая запускает приложение BOINC, и эмулирует для него окружение клиента BOINC.
2. **Мост XWHEP  $\Rightarrow$  EGEE**, подключающий рабочие узлы EGEE к гриду XWHEP путём запуска рабочих компонентов инфраструктуры в виде заданий EGEE.
3. **Мост OurGrid  $\Rightarrow$  EGEE**, который непосредственно запускает задания на вычислительных узлах виртуальной организации EGEE.



**Рис. 1.** Основные элементы инфраструктуры, обеспечивающие взаимодействие разнородных грид-систем

### **Реализация учебного грид-полигона на базе виртуальных машин**

Основным препятствием для запуска сервисных грид-систем на локальных виртуальных инфраструктурах является необходимость взаимодействия элементов грид-сети с внешними сертификационными центрами. Проблемы, связанные с локальным использованием сервисных грид-систем также возникнут для сегмента сервисных грид-систем комбинированной вычислительной инфраструктуры. Для использования ГСПК в рамках локальной грид-инфраструктуры технических сложностей нет.

В рамках МТА SZTAKI для демонстрации своих разработок был предложен макет комбинированной вычислительной грид-инфраструктуры [3]. Макет представляет собой набор из пяти виртуальных машин.

Для обеспечения совместимости с различными платформами виртуализации виртуальные машины имеют формат qemu qcow2. Данный формат полностью совместим с технологией виртуализации XEN. Кроме того программной утилитой qemu-img образы виртуальных машин могут быть сконвертированы в образы виртуальных машин Vmware и VirtualBox.

Сегмент сервисного грида комбинированной инфраструктуры основан на технологии gLite. Сегмент ГСПК – на технологии BOINC. Список компонентов сервисного грида и ГСПК, содержащихся в виртуальных машинах приведён в таблице 1.

*Таблица 1. Список виртуальных машин и основных компонентов программного обеспечения*

<i>№</i>	<i>Название ВМ</i>	<i>Операционная система</i>	<i>Список компонентов</i>
1	edgi-testui	SL* 5, 64 бит	Пользовательский Интерфейс (User Interface - UI).
2	edgi-testvoms	SL 5, 64 бит	Сервис Управления Виртуальными Организациями (Virtual Organization Management Service).
3	edgi-testwms	SL 4, 32 бит	Система Управления Нагрузкой (Workload Management System – WMS).
4	edgi-testce	SL 4, 32 бит	Вычислительный Элемент (Computing Element – CE) инфраструктуры glite, кэш репозитория приложений элемента 3g-bridge.
5	edgi-testboinc	Debian 5.0	BOINC-сервер, BOINC-клиент, 3g-bridge.

\*SL - Scientific Linux.

Для решения проблемы взаимодействия элементов gLite с сертификационными центрами в рамках макета используется локальный сертификационный центр, и все элементы инфраструктуры сервисного грида настроены на использование данного локального сертификационного центра. Кроме того в рамках учебного макета используется упрощенная система сертификации SimpleCA.

### ***Заключение***

В настоящее время приобретают всё большую популярность как ГСПК, так сервисные гриды. Вместе с грид-системами также приобретают популярность технологии их интеграции. Задача подготовки специалистов в области грид-систем и технологий их интеграции приобретает всё большую актуальность.

Рассмотренный в данной статье макет виртуальной вычислительной инфраструктуры даёт учащемуся возможность локально работать только с отдельной реализацией системы 3g-Bridge (gLite to BOINC). Построение

аналогичных макетов для интеграции других типов грид-систем заметно улучшит качество подготовки специалистов по грид-системам.

На основе виртуальных макетов инфраструктур могут быть организованы практические занятия для студентов [4]. Кроме того мощности современных вычислительных машин позволяют в рамках персонального компьютера студента развернуть виртуальный макет для испытания технологий распределённых вычислений в домашних условиях.

### **Литература**

1. gLite. <http://glite.cern.ch/>
2. E. Urbach, P. Kacsuk, Z. Farkas, G. Fedak, G. Kecskeme'ti, O.Lodygensky, A. Cs. Marosi, Z. Balaton, Zolta'n; G. Caillat, G. Gomba's, A.Kornafeld, J. Kova'cs, H. He, R. Lovas: EDGeS: Bridging EGEE to BOINC and Xtrem Web, Journal of Grid Computing, 2009, Vol 7, No. 3, pages 335 -354
3. Страница загрузки макета: <http://www.edgi-grid.eu/downloads/vmimages/v2.0/>
4. V.A. Sukhomlin, A.P. Afanasiev, A.L. Kalinichenko, M.A. Posypkin, S.A. Stupnikov, O.V. Sukhoroslov, On Professional Training and Education in the Field of Grid Technologies and Distributed Computing // тезисы конференции GRID'2010, <http://grid2010.jinr.ru/files/pdf/sukhomlin.pdf>

**Чжо Мью Хтун,**

Национальный исследовательский университет «МИЭТ», Москва  
аспирант, e-mail: [kyawmyohtun2013@gmail.com](mailto:kyawmyohtun2013@gmail.com)

**Чжо Чжо Лин**

Национальный исследовательский университет «МИЭТ», Москва  
аспирант, e-mail: [kklin1500@gmail.com](mailto:kklin1500@gmail.com)

## ***Точность итерационного алгоритма решения задачи распределения нагрузки в системах обслуживания***

### ***Введение***

Центральное место в управлении распределенными системами обслуживания (РСО) занимает задача распределения поступающих заявок между обработчиками или задача диспетчеризации. В работе под РСО мы будем понимать совокупность подвижных обслуживающих объектов, закрепленных за определенной территорией, и единую систему управления ими. В задачу системы управления входит обеспечение реакции на заявки, поступающие от расположенных на этой территории неподвижных источников.

Задачу диспетчеризации можно отнести к задачам дискретной оптимизации. Кроме очевидного требования по точности решения, алгоритмы, лежащие в основе систем поддержки принятия решений, должны обеспечивать и высокое быстродействие.

Широкий класс оптимизационных алгоритмов составляют методы итерационного улучшения определенного известного решения, которые основаны на пошаговом изменении некоторого решения задачи, называемого начальным. На каждой итерации алгоритма осуществляется подсчет целевой функции и, если ее значение изменяется в лучшую сторону, измененное решение становится начальным и процесс продолжается заданное число шагов. Сложность алгоритма определяется числом итераций. Точность решения, получаемого итерационными алгоритмами, в общем случае зависит от начального решения и количества итераций. Существует множество различных стратегий изменения решения на очередной итерации. В настоящей работе мы исследовали алгоритм, основанный на стратегии случайного поиска.

Модель распределенной системы обслуживания.

Математическую модель РСО, ориентированную на решение задачи диспетчеризации представим в следующем виде [1]:

$N$  – число заявок, поступивших от обслуживаемых объектов;

$M$  – число обслуживающих объектов;

$u_{ij}$  – целевая переменная, равная 1, если  $j$ -ый объект используется для обслуживания  $i$ -ого вызова и равная 0 в противном случае;

$r_{ij}$  – время следования  $j$ -ого объекта к источнику  $i$ -ой заявки.

Тогда, целевая функция системы, минимизирующая время реакции, имеет вид:

$$T_{react} = \max(r_{ij}y_{ij}, \forall i = \overline{1, N}; \forall j = \overline{1, M}) \rightarrow \min$$

При этом: 
$$\sum_{i=1}^N y_{ij} \leq 1, j = \overline{1, M} \quad (1).$$

В такой постановке, число обработчиков заявок должно быть больше или равно числу самих заявок  $M \geq N$ . Исходной информацией для алгоритмов распределения заявок служит матрица  $R = \{r_{ij}\}$ , элементы которой, кроме времени движения обработчика к объекту, могут характеризовать и возможность выполнения объектом своей функции по отношению к источнику заявки.

### **Итерационный алгоритм**

Рассмотрим последовательность шагов исследуемого итерационного алгоритма при решении задачи диспетчеризации.

1. В качестве начального решения возьмем некую матрицу  $Y = \{y_{ij}\}_{NM}$ , удовлетворяющую условию (1). Найдем  $T_{react} = \max(r_{ij}y_{ij}, \forall i = \overline{1, N}; \forall j = \overline{1, M})$ .
2. Произведем изменение в известном решении, с помощью случайных парных обменов. Механизм формирования пары элементов, между которыми производится обмен обработчиками, заключается в получении двух случайных величин в диапазоне от 0 до М, номеров строк в матрице  $Y = \{y_{ij}\}_{NM}$ , с последующей их перестановкой и получением новой матрицы  $Y'' = \{y_{ij}\}_{MN}$ . Найдем  $T''_{react}$ . Оценим изменение времени реакции системы. Если  $T''_{react} < T_{react}$ , запоним этот вариант решения и положим  $T''_{react} = T_{react}$ .
3. Повторим шаг 2 заданное число раз\*.
4. Решением задачи будет последний вариант, запоненный на шаге 2.
5. \*В качестве критериев завершения работы в итерационных алгоритмах, использующих рандомизацию, применяются различные подходы:
  - достижение заданной точности;
  - ограничение на время решения задачи;
  - ограничение на число итераций.

Для системы управления РСО критичным является именно время решения задачи, поскольку оно непосредственно влияет на ее характеристики. Это и определяет тот факт, что на практике используется критерий 2, который не зависит от производительности вычислительной системы. Задание числа итераций алгоритма в шаге 3 определяется исходя из значения ограничения по времени. Однако, для исследования точностных свойств алгоритмов, используют критерии 1 или 3. В работе мы используем 3-ий критерий, т.к. первый критерий не обеспечивает

гарантированного завершения работы алгоритма.

Важной характеристикой итерационных алгоритмов является сходимость. Поскольку на каждой итерации алгоритма мы обеспечиваем не ухудшение решения, процесс является сходящимся.

### **Результаты вычислительного эксперимента**

Точность получаемых решений и быстродействие итерационного алгоритма исследовались путем решения тестовых примеров размерностью  $N = M = 18$  и  $N = M = 24$  для числа итераций от  $10^4$  до  $10^8$ . Ниже представлены результаты исследований (таблицы 1-4). В качестве начального, использовалось решение задачи, полученное с помощью последовательного (минимаксного) алгоритма [2]. В каждом случае проводился пятикратный запуск программы и усреднение полученных результатов.

*Таблица 1 - Точность решения ( $N = 18$ , начальное время реакции - 318)*

Число итераций	Время реакции системы					Среднее значение
$10^4$	216	303	308	318	253	279
$5 \cdot 10^4$	275	257	242	265	210	249
$10^5$	251	239	251	209	241	238
$5 \cdot 10^5$	212	216	224	216	229	219
$10^6$	209	217	215	188	229	211
$5 \cdot 10^6$	201	208	204	202	202	203
$10^7$	201	208	201	188	180	195
$5 \cdot 10^7$	188	171	186	178	185	181
$10^8$	168	178	187	168	163	172

*Таблица 2 - Быстродействие алгоритма ( $N = 18$ )*

Число итераций	Время решения задачи (сек)					Среднее значение
$10^4$	0.03	0.038	0.035	0.044	0.04	0.037
$5 \cdot 10^4$	0.147	0.153	0.145	0.155	0.154	0.151
$10^5$	0.285	0.291	0.304	0.31	0.308	0.299
$5 \cdot 10^5$	1.43	1.447	1.511	1.535	1.567	1.498
$10^6$	2.895	2.991	3.106	3.066	3.254	3.062
$5 \cdot 10^6$	14.51	15.39	14.96	15.95	15.59	15.28
$10^7$	29.70	31.83	32.75	31.76	32.64	31.74
$5 \cdot 10^7$	137.4	143.1	157.4	149.8	163.6	150.3
$10^8$	285.5	297.6	304.2	316.0	323.0	305.3

*Таблица 3 - Точность решения ( $N = 24$ , начальное время реакции - 377)*

Число итераций	Время реакции системы					Среднее значение
$10^4$	353	367	377	362	362	364



$5 \cdot 10^4$	342	365	354	357	351	353
$10^5$	353	361	344	350	338	349
$5 \cdot 10^5$	325	348	327	348	345	338
$10^6$	294	321	259	327	245	289
$5 \cdot 10^6$	239	243	237	307	237	252
$10^7$	215	245	293	230	239	244
$5 \cdot 10^7$	216	230	215	215	216	218
$10^8$	215	215	216	215	209	214

Таблица 4 - Быстродействие алгоритма ( $N = 24$ )

Число итераций	Время решения задачи (сек)					Среднее значение
$10^4$	0.032	0.035	0.033	0.0373	0.035	0.034
$5 \cdot 10^4$	0.166	0.155	0.154	0.169	0.157	0.160
$10^5$	0.348	0.357	0.364	0.352	0.352	0.354
$5 \cdot 10^5$	1.648	1.651	1.635	1.646	1.654	1.647
$10^6$	3.244	3.276	3.26	3.261	3.276	3.263
$5 \cdot 10^6$	16.36	16.8	16.52	16.36	17.39	16.69
$10^7$	34.02	34.1	34.32	34.25	34.52	34.24
$5 \cdot 10^7$	168.6	168.0	167.0	168.2	169.9	168.3
$10^8$	336.9	333.8	358.8	357.7	358.2	349.1

Отметим, что при размерности задачи  $N = 18$  и числе итераций  $10^6$  время решения задачи не превышает 3 секунд, при этом время реакции РСО по отношению к начальному решению удается уменьшить на 25-30%. Тесты подтверждают и хорошую сходимость алгоритма. При увеличении числа итераций точность решения повышается практически для всех серий экспериментов.

При увеличении размерности решаемой задачи до  $N = 24$  (таблицы 3 и 4) и тенденция к увеличению точности найденных решений при росте числа итераций сохраняется.

Как и в случае меньшей размерности, для числа итераций  $10^6$  время решения задачи незначительно превышает 3 секунды, при этом время реакции также удается уменьшить на 20-25%.

Вычислительные эксперименты проводились на компьютере с процессором Intel Core Duo T5270 @1.40GHz и памятью 2.0 GB. Различие по времени решения задачи объясняется выбранной стратегией улучшения начального распределения, которая основана на случайных парных обменах. В случае совпадения выбранных на этапе 2 алгоритма строк требуется повторная генерация одного из случайных чисел.

В качестве критерия завершения работы итерационных алгоритмов, основанных на рандомизации, как правило, используют ограничение по числу итераций. Альтернативным критерием является предельное время работы алгоритма. Одним из косвенных признаков достижения

алгоритмом локального оптимума, который может быть также использован на практике, является тот факт, что в течение определенного числа итераций найденное решение не изменяется. В системах поддержки принятия решений, с учетом их специфики, критерием завершения работы алгоритма диспетчеризации должно быть предельное время работы алгоритма.

### **Заключение**

Проведенные исследования показывают, что итерационный алгоритм, основанный на случайных парных перестановках, может быть использован в системах поддержки принятия решений диспетчерских служб РСО. Он обеспечивает снижение времени реакции системы за счет повышения точности решения задачи распределения поступающих заявок.

Дальнейшие исследования направлены на снижение времени решения задачи при большом числе итераций. Это может быть обеспечено путем распараллеливания программного кода и запуска приложения на многоядерном процессоре.

### **Литература**

1. Абаев А.В. Моделирование временных характеристик оперативной деятельности подразделений ГПС. // Информационные технологии и проблемы математического моделирования сложных систем: Сборник научных трудов, выпуск 6. Иркутск: ИрГУПС, 2008.
2. Чжо Мью Хтун. Исследование эффективности алгоритмов решения задачи диспетчеризации. // Современная техника и технологии: Исследования и разработки – 2011 г. Международная научная заочная конференция: сборник докладов. Липецк, 2011. С 40-42.

**СЕКЦИЯ 10. ЦИФРОВЫЕ ТЕХНОЛОГИИ ОБРАБОТКИ  
СИГНАЛОВ И УПРАВЛЕНИЯ**

**Бондаренко М.А.**

Московский государственный университет им. М.В. Ломоносова,  
факультет Вычислительной математики и кибернетики, аспирант  
[max.bond@bk.ru](mailto:max.bond@bk.ru)

## ***Программное и алгоритмическое обеспечение системы технического зрения по автоматизированной обработке изображений ближнего космоса***

### **Аннотация**

*В статье представлена алгоритмическая реализация автоматического распознавания объектов околоземного пространства на примере распознавания метеоров в ночном небе по одиночным кадрам и определения их физических характеристик, таких как абсолютная светимость и звёздная величина. В ходе выполнения работы был разработан новый метод медианной фильтрации, отличающийся от классического метода более высоким качеством фильтрации и скоростью работы от 1,2 до 1,5 раз быстрее. Все изложенные методы реализованы в программном продукте *Meteors Magnitude v1.2*, представленном ниже. Полученные результаты представляют ценность, как для фундаментальных астрофизических исследований, так и для решения прикладных задач навигации летательных аппаратов и ИСЗ.*

**Ключевые слова:** фотометрия околоземного пространства, система технического зрения, обработка изображений, система улучшенного и синтезированного видения.

### **Введение**

Активное развитие цифровой фотометрии околоземного пространства обусловлено, прежде всего, требованиями к безопасности полётов ИСЗ, летательных аппаратов в верхних слоях атмосферы, космических аппаратов и станций от столкновения с малыми телами Солнечной системы и космическим мусором. Следующее ключевое приложение - фундаментальные исследования природы малых тел Солнечной системы. Особо следует отметить задачу заблаговременного предупреждения бомбардировки Земли малыми космическими телами.

Фотометрия околоземного пространства характеризуется большими объёмами получаемой информации, из которой очень малая доля представляет полезные данные. Отсюда следует высокая трудоёмкость или практическая невозможность получения точного научного результата при ручном анализе фотометрии оператором. Таким образом, автоматический анализ фотометрии представляет большую практическую ценность.

### **Способ обработки кадров фотометрии**

Ниже предлагается общая схема обработки кадра для

автоматического распознавания объектов в ночном небе:

- Вычитание шумового фона
- Фильтрация
- Повышение контраста
- Сегментация объектов по заданному набору признаков
- Выделение характеристик сегментированных объектов
- Анализ характеристик объектов
- Принятие решения

#### **Вычитание шумового фона**

Начальное вычитание шумового фона обусловлено спецификой задачи: наличие большого количества неинформативных областей с относительно невысокой яркостной вариацией: звёзды и прочие объекты на фоне ночного неба. Исходя из особенностей обрабатываемого изображения, предполагается, что фоновые неинформативные пиксели располагаются по его краям.

Пусть исходное изображение

$$F(x, y) \in \mathbb{R}^{M \times N}, x = 0 \dots M - 1, y = 0 \dots N - 1,$$

тогда обработанное изображение  $G(x, y) \in \mathbb{R}^{M \times N}$  имеет вид:

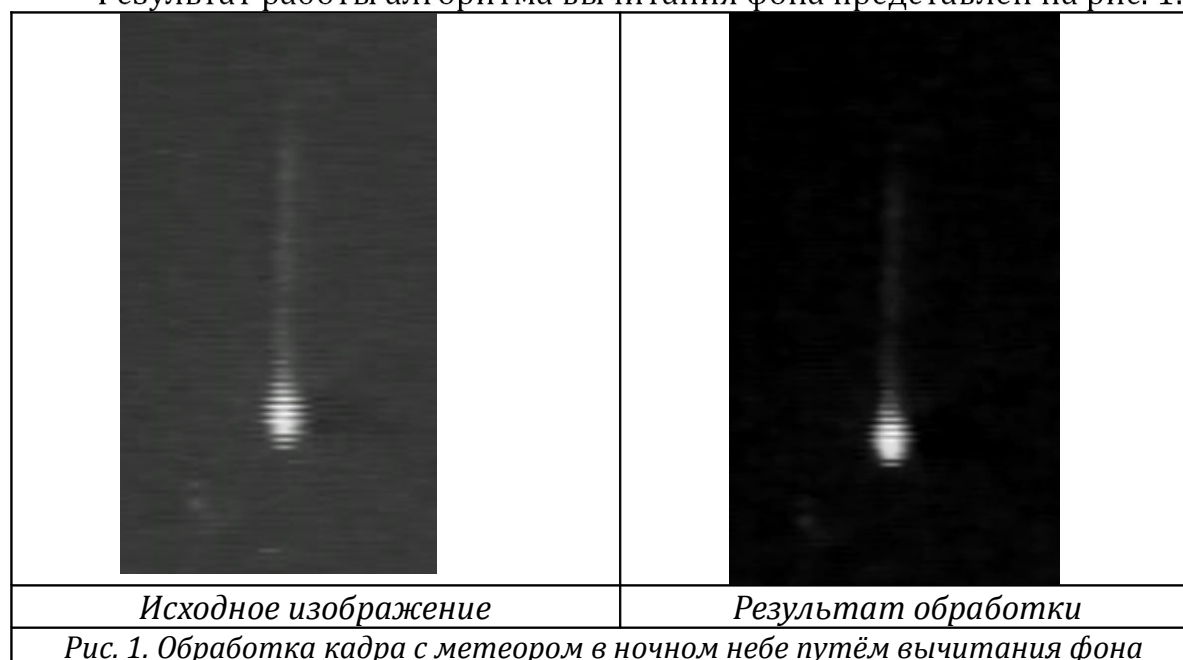
$$G(x, y) = \begin{cases} F(x, y) - k, & F(x, y) \geq k \\ 0, & F(x, y) < k' \end{cases}$$

где

$$k = \frac{1}{2 \cdot (N + M)} \cdot \left( \sum_{\substack{y=0 \\ x=0, M-1}}^{y=N-1} F(x, y) + \sum_{\substack{x=M-1 \\ y=0, N-1}}^{x=M-1} F(x, y) \right),$$

$$\forall x = 0 \dots M - 1, \forall y = 0 \dots N - 1.$$

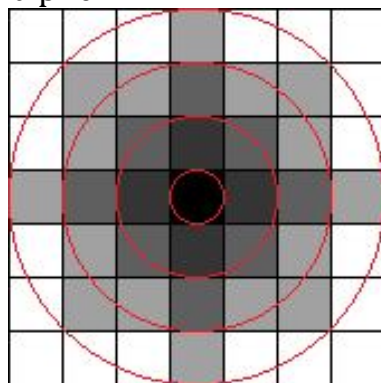
Результат работы алгоритма вычитания фона представлен на рис. 1:



Как видно из рис. 1, шумовая составляющая кадра с метеором существенно снизилась, сохранив при этом полезный сигнал практически без потерь.

### **Фильтрация кадра**

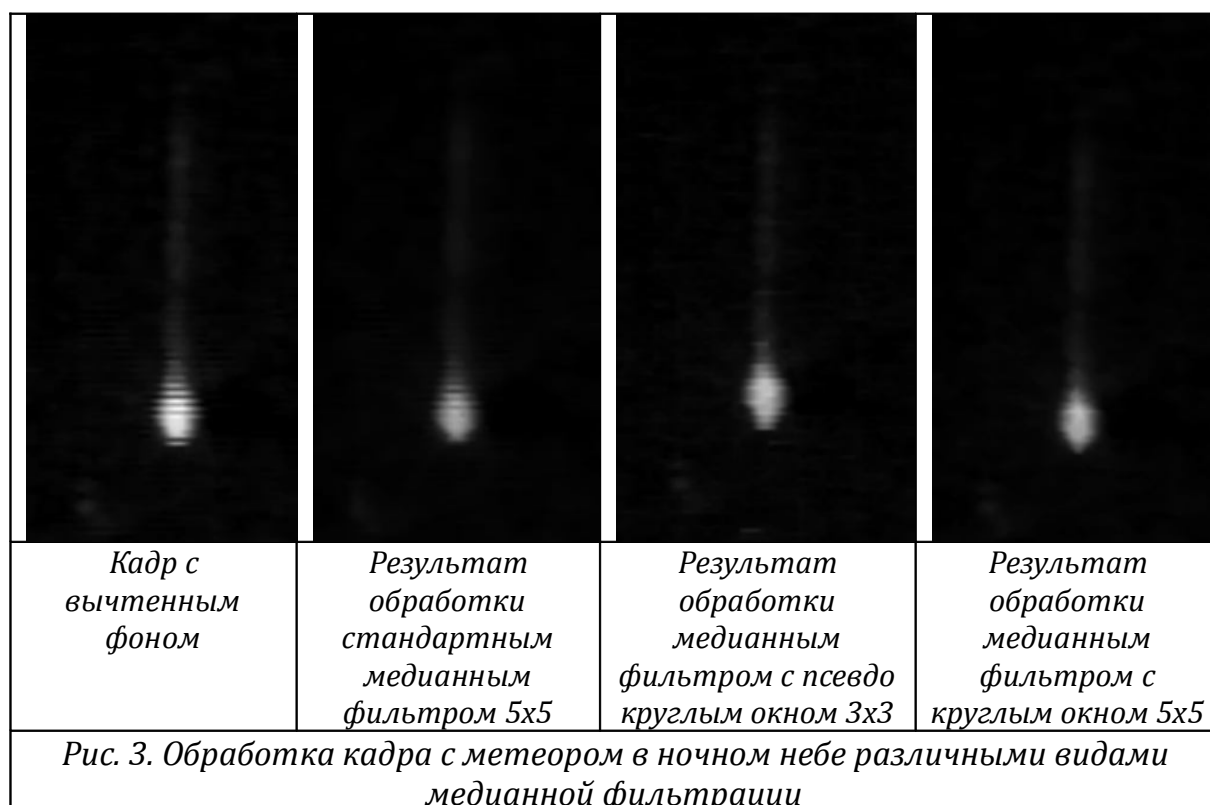
Помимо присутствия на обрабатываемых кадрах фонового шума, имеет место импульсный шум, связанный с чувствительностью матрицы цифровой камеры и погрешностью оцифровки полученного видеосигнала. А также шум обусловленный структурой матрицы цифровой камеры: в нашем случае это характерные полосы на изображении. Для подавления импульсного шума используется модификация стандартного медианного фильтра, как наиболее подходящая для его устранения [1]. Модификация заключается в использовании эллиптической окрестности пикселя внутри маски. Использование эллиптической маски обусловлено знанием об истинной форме звёзд и сгорающих метеоров. Пусть размер квадратной маски задаётся как  $2 \cdot r + 1$  на  $2 \cdot r + 1$ , где  $r \geq 0$  – целое число, радиус маски. (При  $r = 0$  имеем тождественный фильтр). Тогда условие принадлежности пикселя эллиптической маске задаётся функцией:  $a \cdot x^2 + b \cdot y^2 \leq r^2$ , где  $x, y$  – координаты пикселя на изображении,  $a > 0, b > 0$  – главные диаметры эллипса. В данной реализации  $a = b = 1$ , то есть используются те пиксели квадратной маски, которые ограничены функцией, задающей круг: так получается псевдо круглая маска. Вид псевдо круглых масок представлен на рис. 2:



*Рис. 2. Псевдо круглые маски размера 1x1, 3x3, 5x5, 7x7 – разным цветом показаны пиксели, принадлежащие разным маскам*

Результаты подавления импульсного и матричного шумов кадра с вычтенным фоном представлены на рис. 3.

Как видно из рис. 3, медианный фильтр с псевдо круглым окном хорошо подавляет матричный шум в отличие от стандартного медианного фильтра, который не смог существенно устранить матричный шум. И в тоже время предлагаемый фильтр не так сильно искажает полезный сигнал. В реализации алгоритма используется маска 5x5.



### **Коррекция кадра**

Для усиления полезного сигнала используется алгоритм линейного растяжения гистограммы яркостей изображения:

$$G(x, y) = (2^b - 1) \cdot \frac{G(x, y) - G_{\min}}{G_{\max} - G_{\min}}, \quad x = 0 \dots M - 1, \quad y = 0 \dots N - 1,$$

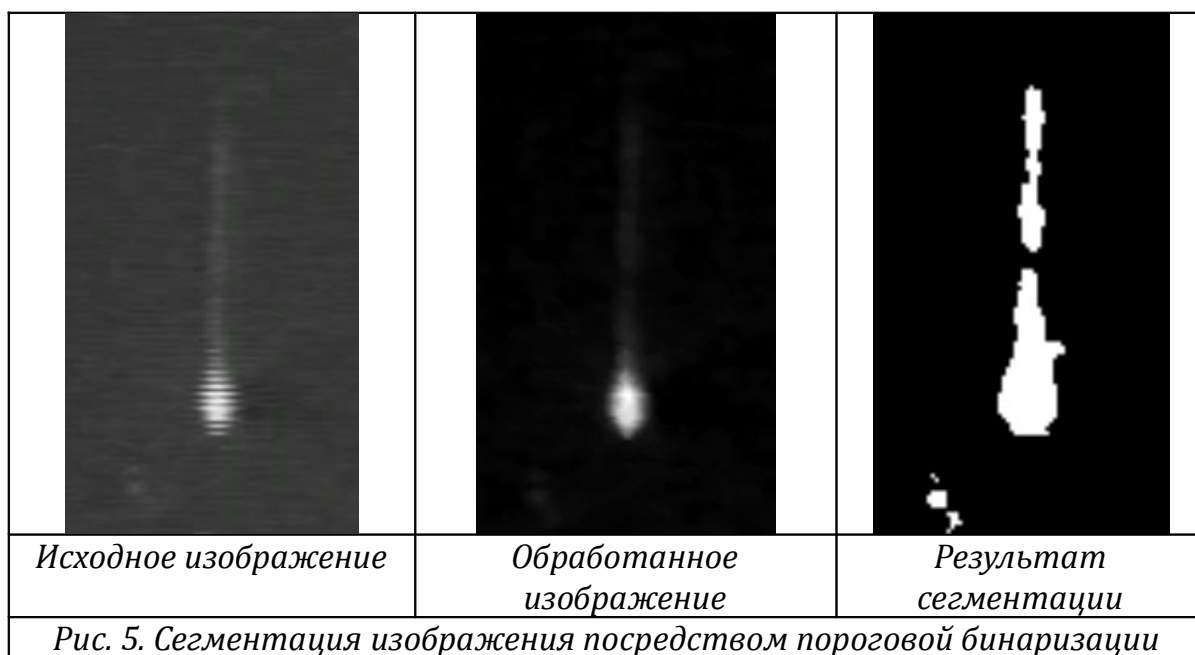
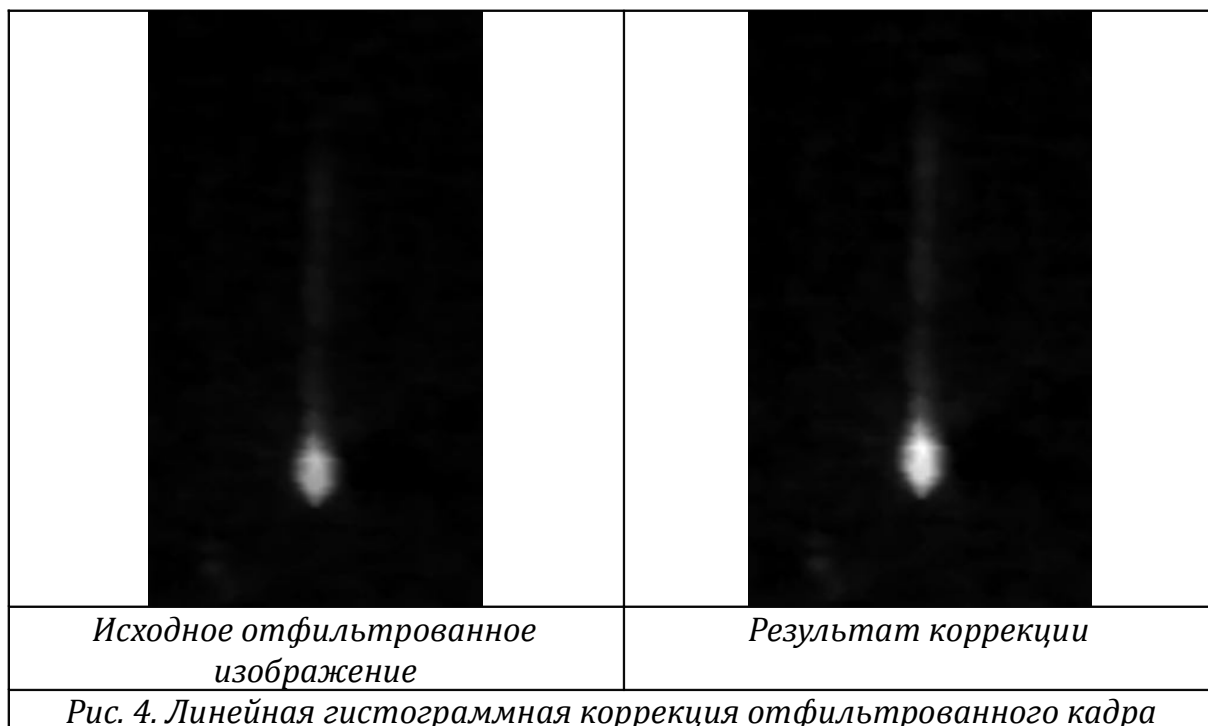
$$G_{\max} = \max_{x, y} G(x, y), \quad G_{\min} = \min_{x, y} G(x, y),$$

где  $G$  – обрабатываемое изображение размера  $M \times N$ ,  $b$  – битность пикселя изображения  $G$ . Результат работы алгоритма представлен на рис. 4.

Использование линейного гистограммного преобразования объясняется важностью сохранения яркостного распределения в кадре для более точного установления абсолютной светимости и звёздной величины объектов.

### **Реализация сегментации областей интереса**

Сегментация областей интереса (метеоров, звёзд и прочих объектов в небе) производится посредством пороговой бинаризации кадра как наиболее быстрого и одновременно эффективного метода. Порог бинаризации  $b$  выбирался опытным путём из диапазона яркостей  $[0, 32]$  для 8-битных изображений так, чтобы не отсегментировать оставшиеся шумовые составляющие изображения, и в тоже время выделить полезные объекты. Чем меньше шумовых составляющих на изображении, тем этот порог ближе к 0. В данной реализации  $b = 16$ . Результат сегментации кадра представлен на рис. 5.



#### **Анализ сегментированных объектов**

Для анализа формы объектов используется алгоритм закрашки связной области, который выделяет следующую полезную информацию (структуру из пяти полей) об объектах:

- Главные диаметры объектов
- Число пикселей
- Пиксельная плотность объекта - отношение числа пикселей объекта к числу пикселей ограничивающей его прямоугольной области
- Координаты центра прямоугольной области, ограничивающей



объект

- Суммарная пиксельная яркость

При автоматическом распознавании объекта уже работаем не с изображением, а с указанной структурой.

### **Автоматическое распознавание метеоров на одиночных кадрах**

Алгоритм распознавания метеоров строится на основании их характерных признаков.

Во-первых, суммарная яркость метеора, как правило, превосходит яркость большинства звёзд, что означает условие на  $C_{min}$  — минимальное число пикселей объекта, которое определяется экспериментально. В данной реализации  $C_{min} = 16$ .

Во-вторых, метеор представляет собой вытянутый объект, что означает условие на отношение главных диаметров в зависимости от пиксельной плотности объекта. Пусть  $D_x, D_y$  — главные диаметры объекта, выраженные в пикселях изображения по горизонтали и вертикали соответственно. Тогда объект вписан в прямоугольник с длинами сторон  $D_x$  и  $D_y$ . Пусть далее  $S$  — число пикселей объекта, его пиксельная площадь. Тогда пиксельная плотность данного объекта

$$\rho_{pixel} = \frac{S}{D_x \cdot D_y}$$

Очевидно, справедливо неравенство:  $0 < \rho_{pixel} \leq 1$ . Пусть далее  $N = \frac{D_y}{D_x}$  — отношение главных диаметров объекта. Экспериментально установлено (на выборке из 100 кадров с метеорами), что для вертикально падающих метеоров  $N > N_{min} > 1,5$ . В данной реализации  $N_{min} = 1,66$ . Метеоры могут лететь под углом, и для реального метеора  $N$  может быть меньше  $N_{min}$ , однако. Поэтому необходимо ввести зависимость между  $N$  и  $\rho_{pixel}$ :

11. Если у объекта  $N \leq N_{min}$ , то необходимым условием вытянутости будет  $\rho_{pixel} < \rho_{min}$ .

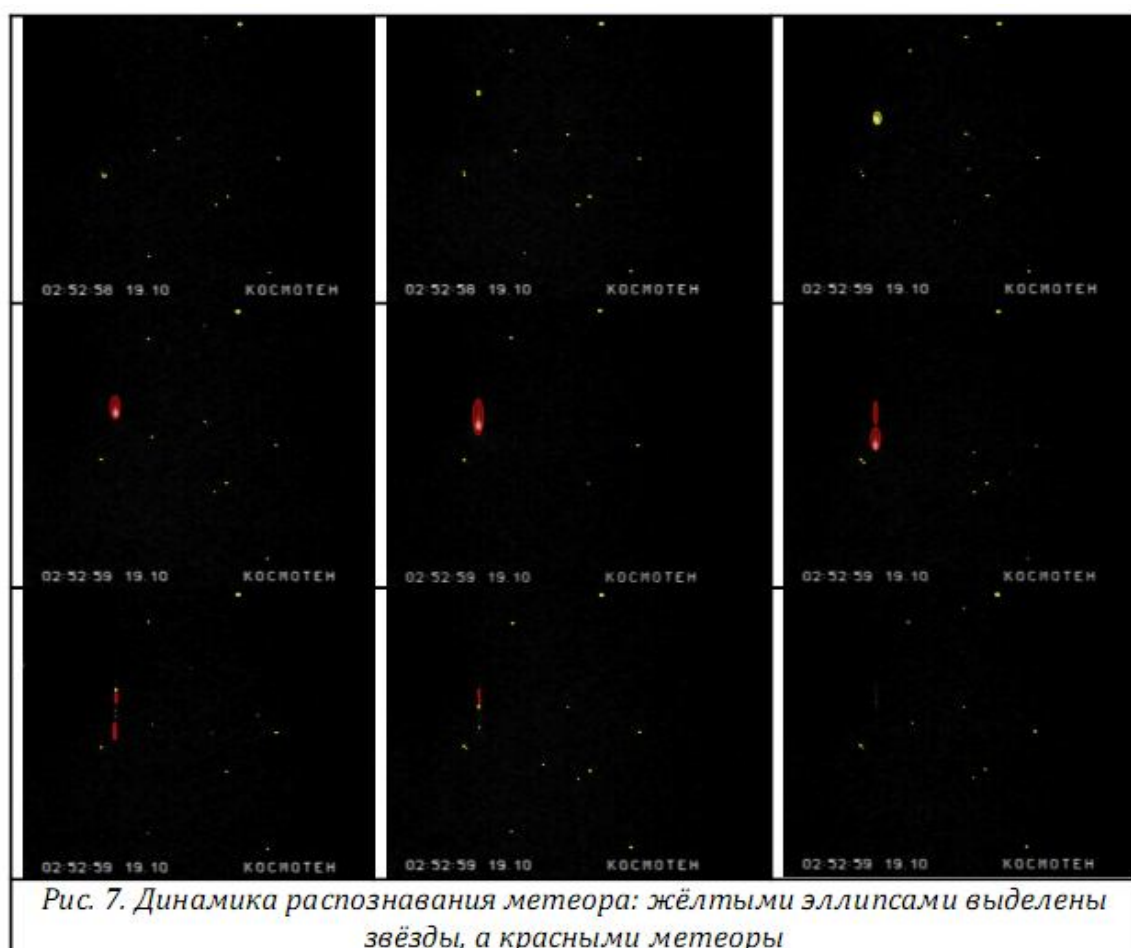
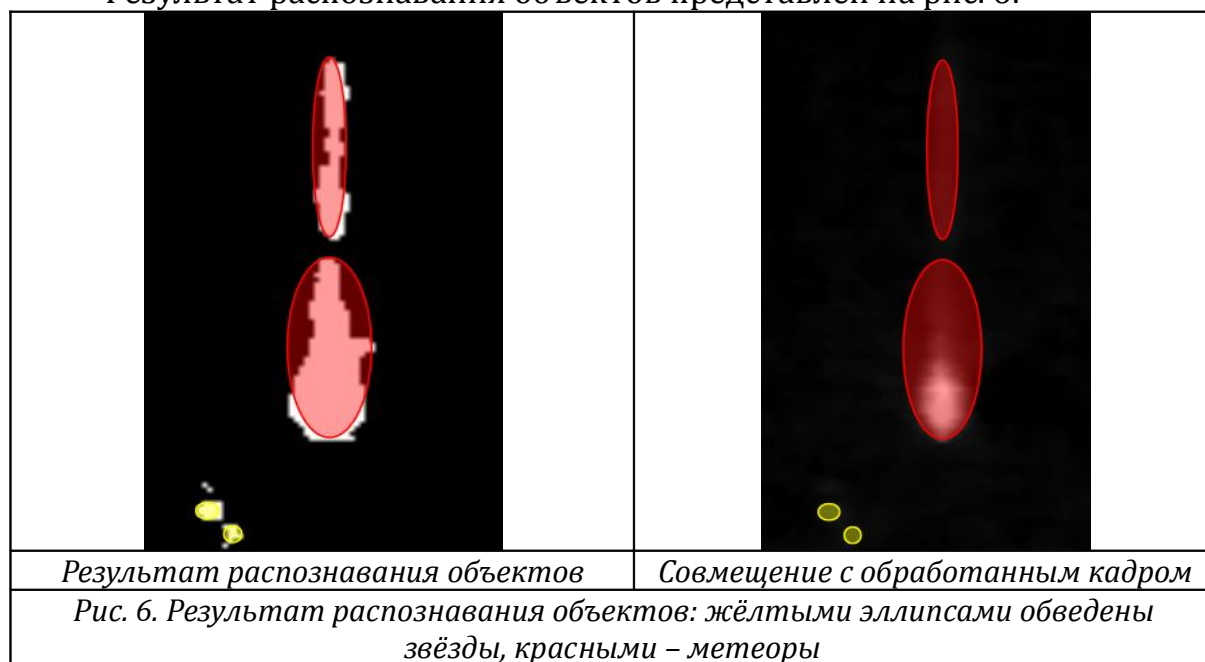
12. Если, наоборот,  $N > N_{min}$ , то  $\rho_{pixel} \geq \rho_{min}$

Таким образом, условие вытянутости объекта можно записать в виде логического выражения:

$$N > N_{min} \text{ xor } \rho_{pixel} < \rho_{min}$$

Данное условие вытянутости является необходимым, но не является достаточным: можно привести пример, когда по данному условию объект с  $N = 1$  будет считаться вытянутым: это объект в форме бублика. Но вероятность попадания «бублика» в кадр в данной задаче ничтожно мала. В данной реализации  $\rho_{min} = 0,33$ .

Результат распознавания объектов представлен на рис. 6:



На рис. 7 показана динамика распознавания метеора в ночном небе Земли с помощью предложенного метода, что позволяет судить о его

качестве.

### ***Вычисление звёздных величин объектов***

Вычисление производится на основании данных о суммарной пиксельной яркости объектов, что соответствует светимостям объектов по формуле из [2]:

$$N_2 = N_1 + 2,5 * \log_{10} \left( L_1 / L_2 \right),$$

где  $N_{1,2}$  - абсолютные звёздные величины объектов,

$L_{1,2}$  - их суммарные пиксельные яркости (светимости).

По каждому «эталонному» объекту, для которого пользователь задал звёздную величину, считаются звёздные величины прочих объектов. Для каждой непомеченной звезды имеем набор вычисленных звёздных величин, среднее арифметическое значение которого и будет результирующим значением. Тем самым каждый дополнительный ввод позволяет повысить точность вычисления звёздных величин для неизвестных объектов.

### ***Точность вычисления звёздных величин***

При тестировании на реальных снимках (с учётом шума и слабого контраста) звёздные величины (з.в.) известных звёзд по одной заданной были вычислены с погрешностями в пределах 0,5 от з.в. Дополнительный ввод снизил погрешность вычислений примерно до 0,25 от з.в.

### ***Особенности реализации***

Все описанные алгоритмы были программно реализованы в программе определения характеристик звёзд и метеоров по фотографиям под названием *Meteors Magnitude v1.2* написанной на языке C# с использованием технологии .NET и библиотек для создания оконных интерфейсов WPF 4.0. Программа имеет следующие возможности:

- Чтение и обработка файлов форматов bmp, png, jpg, jpeg - работа с одиночными кадрами.
- Автоматическое распознавание звёзд и метеоров на кадре.
- Фильтрация и коррекция входных кадров.
- Просмотр увеличенной области интереса изображения – инструмент «экранный телескоп».
- Работа с объектами различных типов: «звезда», «метеор» и «летательный аппарат». Возможность изменения типа объекта или исключения его из рассмотрения.
- Включение/отключение режима сегментации объектов с запоминанием всех изменений, вносимых пользователем.
- Работа с объектами внутри области интереса.
- Ввод звёздной величины объекта позволяет вычислить эту характеристику для остальных объектов. Звёздная величина уточняется при дополнительном вводе.

- Возможность очистки всех данных введенных пользователем.
  - Полная интерактивная инструкция пользователя
- На рис. 8 представлен скриншот работы программы.

### ***Скорость работы алгоритмов***

Скорость выполнения программы определялась на компьютере с процессором Intel Core i7 620M с тактовой частотой 2,67 ГГц.

Обработка и анализ кадра размером в 1392x1032 пикселя заняла 2,2 секунды процессорного времени без учёта времени вывода результата распознавания на экран.

Алгоритм модифицированной медианной фильтрации с псевдо круглой маской 5x5 показал себя быстрее: на данном кадре он отработал за 1,7 секунды, в то время как стандартный медианный фильтр 5x5 отработал за 2,5 секунды.

Повышение скорости работы модифицированного алгоритма медианной фильтрации объясняется сортировкой меньшего по объёму массива для каждого пикселя изображения. В случае стандартной медианы 5x5 приходится сортировать 25 элементов массива, в то время как для модифицированной медианы с псевдо круглой маской это число равно 13 (рис. 2), что почти в 2 раза меньше.

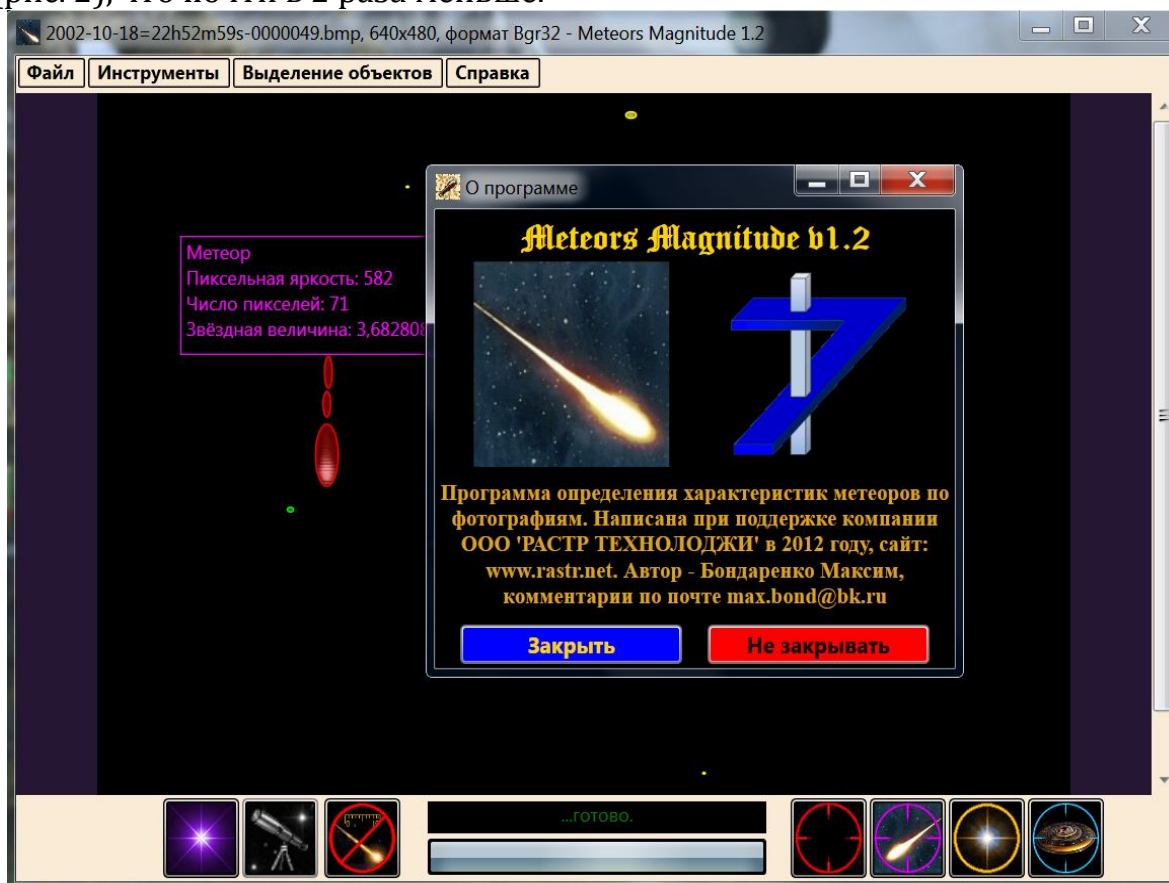


Рис. 8. Программа по автоматическому распознаванию звёзд и метеоров и определению их характеристик

### **Перспективы развития проекта Meteors Magnitude**

1. В ближайшее время планируется доработка программы:

- добавление возможности построения графиков динамики блеска метеоров,
- усовершенствование алгоритмов путём добавления анализа пары смежных кадров.

2. В перспективе планируется совмещение написанной программы с измерительными цифровыми мегапиксельными камерами научно-производственной фирмы ООО «РАСТР ТЕХНОЛОДЖИ» [3].

Реализация пункта 2 позволит обрабатывать последовательность калиброванных кадров в реальном времени. Это означает не только вычисление звёздных величин объектов с точностями, сопоставимыми со звёздными каталогами, но и определение их массы (путём анализа скорости сгорания в атмосфере), расстояния от поверхности Земли, линейные размеры (путём съёмки несколькими синхронно работающими камерами) и среднюю плотность объекта. Это, несомненно, позволит отличать метеоры от летательных аппаратов и от «неподвижных» звёзд.

### **Благодарности**

Работа одобрена Бондаренко Андреем Викторовичем, ген. директором научно-производственной фирмы ООО «РАСТР ТЕХНОЛОДЖИ».

Отдельное спасибо д.ф.м.н. Багрову Александру Викторовичу, ведущему научному сотруднику ИА РАН и к.ф.м.н. Леонову Владиславу Александровичу, старшему научному сотруднику ИА РАН за постановку интересной задачи, хороший чай и вкусные пряники.

### **Литература**

1. Р. Гонсалес, Р. Вудс, Цифровая обработка изображений. М.: Техносфера, 2005. С. 196.
2. Куликовский П. Г. Справочник любителя астрономии под ред. В.Г. Сурдина, М.:УРСС, V издание, 2002. С. 149.
3. Техническая документация на цифровые камеры и программное обеспечение. ООО «РАСТР ТЕХНОЛОДЖИ». 2012 г. Москва, [www.rastr.net](http://www.rastr.net).
4. Бондаренко А.В. Докучаев И.В., Князев М.Г. Телевизионная видеокамера с цифровой обработкой в реальном времени // Современная электроника. № 3, 2006. С. 50.

## ***Вопросы оптимизации цифровых систем управления и обработки сигналов***

### **Аннотация**

*В докладе обсуждаются базовые аспекты общей методологии аналитического проектирования цифровых систем с применением оптимизационного подхода. Основное внимание уделяется частной ситуации с конкретной реализацией этого подхода для синтеза линейных стационарных систем с обратной связью, функционирующих в дискретном времени. Предлагается спектральный метод минимизации функционалов, представленных нормами передаточных функций цифровых систем как элементов пространств Харди  $H_2$  и  $H_\infty$ .*

### **1. Введение**

Применение цифровых динамических систем управления и обработки информации, базирующихся на современных компьютерных устройствах и технологиях, в настоящее время приобрело всеобъемлющий характер практически во всех отраслях науки, техники, экономики, образования и т.д. Естественно, что в основе построения и использования таких систем лежат различные общие идеи и конкретные способы формализации соответствующих содержательных задач, прежде всего – на математическом уровне.

Это обстоятельство определяет особую ответственность прикладных математиков и ИТ-специалистов за выбор, модификацию и разработку математических методов моделирования, анализа, синтеза и практической реализации цифровых систем. Эти методы должны в максимальной степени соответствовать возможностям и функциональной ориентации тех компьютерных средств, с помощью которых они реализуются, что в конечном итоге определяет суммарную эффективность процессов принятия решений, управления и обработки информации.

Современное состояние компьютерных технологий дает возможность выделить две стратегические линии, находящиеся в настоящее время в стадии бурного развития и постоянного совершенствования:

- параллельные вычисления, суперкомпьютеры (Parallel Computing and Supercomputers) и способы их эффективного применения;
- встраиваемые вычисления и системы (Embedded Computing and Systems) во всем многообразии их использования.

Оба указанных направления служат базой для реализации современных цифровых систем, хотя они имеют принципиальное различие.



В рамках первого из них используются мощные вычислительные ресурсы, предоставляемые пользователю, а второе, напротив, связано с существенными их ограничениями. Тем не менее, специальная ориентация математических моделей и методов на реализацию цифровых систем в соответствующей вычислительной среде позволяет существенно повысить эффективность их практического применения.

В частности, здесь представляется специализированный спектральный подход к синтезу цифровых систем с обратной связью, базирующийся на идеологии оптимизации линейных стационарных систем в пространствах Харди  $H_2$  и  $H_\infty$  [1,2].

Принятый подход позволяет уменьшить время вычислений и объем оперативной памяти, что имеет особую значимость в тех ситуациях, когда по тем или иным причинам расчетный алгоритм синтеза используется для перенастройки в ходе функционирования систем, работающих в режиме реального времени. Это имеет особое значение для адаптивной реализации в составе различных встраиваемых систем автоматического управления и обработки сигналов.

## 2. Постановка задач $H$ -оптимизации

Пусть задана математическая модель неизменяемой линейной стационарной части системы, функционирующей в дискретном времени  $n$ , в виде следующего разностного уравнения:

$$A(q)y[n] = B(q)u[n] + d[n], \quad n \in N^1, \quad (1)$$

где  $y, u$  и  $d$  – скалярные дискретные процессы:  $y$  – контролируемая переменная,  $u$  – сигнал обратной связи (управление),  $d$  – внешнее воздействие;  $A(q)$  и  $B(q)$  – взаимно простые полиномы степеней  $v$  и  $m \leq v-1$  от оператора  $q$  сдвига на такт вперед.

Будем замыкать объект (1) обратной связью

$$u = W(q)y, \quad (2)$$

с передаточной функцией  $W(q) = W_1(q)/W_2(q)$ ,  $W_1, W_2$  – полиномы. В результате замыкания получим SISO-систему (1), (2) с уравнением  $[A(q) - B(q)W(q)]y = d$ , где  $d = \{d[t]\}$  – случайный стационарный эргодический процесс со спектральной плотностью мощности

$$S_d(e^{j\omega})|_{e^{j\omega}=z} \equiv S_1(z)S_1(z^{-1}), \quad S_1(z) \equiv N(z)/T(z), \quad (3)$$

$N(z)$  и  $T(z)$  – шуровские полиномы (все их корни находятся в открытом единичном круге  $D$  на комплексной плоскости).

Качество работы замкнутой цифровой системы (1), (2) в данном случае имеет смысл оценивать среднеквадратичным функционалом

$$I = I(W) = \langle y^2 \rangle + k^2 \langle u^2 \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N [y^2[n] + k^2 u^2[n]], \quad (4)$$

который задан на ее движениях,  $k = \text{const}$ .

Если функция  $S_d(e^{j\omega})$  задана, то решается задача о синтезе оптимальной обратной связи, обеспечивающей минимум функционала (4) на множестве стабилизирующих регуляторов (2).

Если же функция  $S_d$  априори не задана, но известна её принадлежность допустимому множеству

$$\mathfrak{R} = \left\{ S_d(e^{j\omega}) : \frac{1}{\pi} \int_0^\pi S_d(e^{j\omega}) d\omega \leq 1 \right\}, \quad (5)$$

то рассматривается задача о синтезе гарантирующей обратной связи по отношению к функционалу (4) с учетом (5).

Введем в рассмотрение множество  $RL$  рациональных дробей  $\rho(z)$  без полюсов на единичной окружности, а также гильбертово пространство

$RL_2 \subset RL$  правильных функций с нормой  $\|\rho\|_2^2 = \frac{1}{\pi} \int_0^\pi |\rho(e^{j\omega})|^2 d\omega$ . Справедливо

представление  $RL_2 = RH_2 \oplus RH_2^\perp$ , где пространство  $RH_2$  содержит все функции аналитические в круге  $D$ , а его дополнение  $RH_2^\perp$  – вне этого круга. Введем также пространство  $RL_\infty \subset RL$  правильных дробей, имеющих норму  $\|\rho\|_\infty = \sup_{\operatorname{Re} z \geq 1} |\rho(z)| = \max_{\omega \in [0, \pi]} |\rho(e^{j\omega})|$ , выделяя из него подпространство  $RH_\infty$ , включающее все элементы аналитические вне единичного круга  $D$ .

Нетрудно убедиться в том, что указанные выше задачи оптимизации могут быть строго формализованы в следующем виде:

$$J_2(W) = \|H(W)S_1\|_2^2 \rightarrow \min_{W \in \Omega_2}, \quad \Omega_2 = \{W : H(W) \in RH_2\}; \quad (6)$$

$$J_\infty(W) = \|H(W)\|_\infty^2 \rightarrow \min_{W \in \Omega_\infty}, \quad \Omega_\infty = \{W : H(W) \in RH_\infty\}, \quad (7)$$

где  $H(W)$  – обобщенная передаточная функция замкнутой системы

$$H(z, W)H(z^{-1}, W) \equiv H_y(z, W)H_y(z^{-1}, W) + k^2 H_u(z, W)H_u(z^{-1}, W). \quad (8)$$

Здесь  $H_y = 1/(A - BW)$  и  $H_u = W/(A - BW)$  – передаточные функции этой системы по выходу и по управлению соответственно.

Заметим, что функционалы  $J_2$  и  $J_\infty$  в задачах (6) и (7) нелинейно зависят от функций  $W$ , что существенно затрудняет их поиск. Эту трудность можно снять параметризацией множества стабилизирующих обратных связей, как показано в работе [3], вводя варьируемые функции-параметры  $\Phi(z)$  формулами

$$\Phi(z) = \alpha(z)H_y(z) + \beta(z)H_u(z), \quad Q(z) = A(z)\beta(z) + B(z)\alpha(z), \quad (9)$$

где  $\alpha$  и  $\beta$  – любые полиномы, для которых  $Q$  – шуровский полином.

Формула (9) и тождество  $AH_y - BH_u \equiv 1$  дают однозначную связь

$$W = L_\Phi(\Phi) = \frac{A\Phi - \alpha}{B\Phi + \beta}, \quad \Phi = L_\Phi^{-1}(W) = \frac{\alpha + \beta W}{A - BW}, \quad (10)$$

а также выражения  $H_y = (B\Phi + \beta)/Q$  и  $H_u = (A\Phi - \alpha)/Q$

Введем в рассмотрение два множества функций  $\Phi(z)$  со всеми полюсами внутри единичного круга:  $\Omega_2^\Phi = L_\Phi^{-1}(\Omega_2)$ ,  $\Omega_\infty^\Phi = L_\Phi^{-1}(\Omega_\infty)$ .

Тогда оптимизационные задачи (6) и (7) эквивалентны задачам

$$I_2 = I_2(\Phi) = J_2(L_\Phi(\Phi)) = \|H(\Phi)S_1\|_2^2 \rightarrow \min_{\Phi \in \Omega_2^\Phi}, \quad (11)$$

$$I_\infty = I_\infty(\Phi) = J_\infty(L_\Phi(\Phi)) = \|H(\Phi)\|_\infty^2 \rightarrow \min_{\Phi \in \Omega_\infty^\Phi}, \quad (12)$$



где  $H(\Phi)\bar{H}(\Phi) \equiv H_y(\Phi)\bar{H}_y(\Phi) + k^2 H_y(\Phi)\bar{H}_y(\Phi)$ .

Здесь и далее черта над дробью означает обращение переменной  $z$ .

Нетрудно показать, что функционалы  $I_2(\Phi)$  и  $I_\infty(\Phi)$ , представленные формулами (11) и (12), однозначно преобразуются к виду

$$I_2(\Phi) = \frac{1}{\pi} \int_0^\pi \left[ |T_1(e^{j\omega}) - T_2(e^{j\omega})\Phi(e^{j\omega})|^2 + k^2 |T_3(e^{j\omega})|^2 \right] S_d(\omega) d\omega, \quad (13)$$

$$I_\infty(\Phi) = \max_{\omega \in [0, \pi]} \left[ |T_1(e^{j\omega}) - T_2(e^{j\omega})\Phi(e^{j\omega})|^2 + k^2 |T_3(e^{j\omega})|^2 \right], \quad (14)$$

где  $T_1$ ,  $T_2$  и  $T_3$  – рациональные дроби с шуровскими знаменателями

$$T_1 = (k^2 \alpha \bar{A} - \beta \bar{B}) / QG, \quad T_2 = \bar{G} / Q, \quad T_3 = 1/G, \quad (15)$$

причем полином  $G(z)$  – шуровский результат факторизации

$$k^2 A\bar{A} + B\bar{B} \equiv G\bar{G}. \quad (16)$$

### 3. Синтез оптимальных обратных связей

Представление минимизируемого функционала  $I_2(\Phi)$  в виде (13) позволяет легко найти оптимальную функцию-параметр  $\Phi_{02}(z)$ , а затем и передаточную функцию  $W_0 = L_\Phi(\Phi_{02})$  оптимальной обратной связи.

**Теорема 1.** На множестве  $\Omega_2^\Phi$  существует единственная функция  $\Phi_{02}(z)$ , обеспечивающая минимум функционала  $I_2(\Phi)$ , причем

$$\Phi_{02} = \frac{(k^2 \alpha \bar{A} - z^{v-m} \beta \bar{B}) S_1 - RQ}{G\bar{G}S_1}, \quad R(z) = \sum_{i=1}^v \frac{\tilde{G}(z)}{z - g_i} \frac{k^2 \tilde{A}(g_i) S_1(g_i)}{B(g_i) \tilde{G}'(g_i)}, \quad (17)$$

где  $g_i$  ( $i = \overline{1, v}$ ) – простые корни полинома  $\tilde{G}(z)$ .

**Доказательство.** После подстановки (15) в (13) имеем:

$$I_2(\Phi) = \left\| \left( \frac{k^2 \alpha \bar{A} - \beta \bar{B}}{QG} - \frac{\bar{G}}{Q} \Phi \right) S_1 \right\|_2^2 + k^2 \left\| \frac{S_1}{G} \right\|_2^2. \quad (18)$$

С учетом тождества  $|G(e^{-j\omega}) / G(e^{j\omega})| \equiv 1$  получим

$$\left\| \left( \frac{k^2 \alpha \bar{A} - \beta \bar{B}}{QG} - \frac{\bar{G}}{Q} \Phi \right) S_1 \right\|_2^2 = \|MS_1 - L\Phi S_1\|_2^2, \quad M = \frac{k^2 \alpha \bar{A} - \beta \bar{B}}{QG}, \quad L = \frac{G}{Q}.$$

Разложим  $MS_1 \in RL_2$  в сумму ортогональных элементов с помощью теплицева  $\Theta_M$  и ганкелева  $\Gamma_M$  операторов с индексами  $M$ :

$$\|\Theta_M(S_1) + \Gamma_M(S_1) - L\Phi S_1\|_2^2 = \|\Theta_M(S_1) - L\Phi S_1\|_2^2 + \|\Gamma_M(S_1)\|_2^2,$$

поскольку  $(\Theta_M(S_1) - L\Phi S_1) \perp \Gamma_M(S_1)$ . При этом из сепарации рациональной дроби  $M(z)S_1(z)$  имеем

$$M(z)S_1(z) \equiv \frac{k^2 \alpha(z) \tilde{A}(z) - z^{v-m} \beta(z) \tilde{B}(z)}{Q(z) \tilde{G}(z)} \cdot \frac{N(z)}{T(z)} \equiv \frac{M_1(z)}{Q(z)T(z)} + \frac{R(z)}{\tilde{G}(z)}, \quad (19)$$

откуда следует, что  $\Theta_M(S_1) = M_1 / QT$ ,  $\Gamma_M(S_1) = R / \tilde{G}$ .

Из факторизации (16) находим

$$g_i^{v-m} \tilde{B}(g_i) = -k^2 A(g_i) \tilde{A}(g_i) / B(g_i), \quad i = \overline{1, v},$$

что в соответствии с (19) при учете (9) позволяет получить

$$R(g_i) = k^2 \tilde{A}(g_i) S_1(g_i) / B(g_i), \quad i = \overline{1, v}. \quad (20)$$

Зная числа  $R(g_i)$ , с помощью интерполяционной формулы Лагранжа находим полином  $R(z)$  (17), причем из (19) следует

$$\Theta_M(S_1) = (k^2 \alpha \tilde{A} - z^{v-m} \beta \tilde{B}) S_1 / (Q \tilde{G}) - R / \tilde{G}, \quad \Gamma_M(S_1) = R / \tilde{G}. \quad (21)$$

Обращаясь к (18)  $I_2(\Phi) = \|\Theta_M(S_1) - L\Phi S_1\|_2^2 + \|\Gamma_M(S_1)\|_2^2 + k^2 \|S_1 / G\|_2^2$ , замечаем, что второе и третье слагаемые не зависят от параметра  $\Phi$ . Тогда минимум функционала достигается при условии  $\Theta_M(S_1) - L\Phi S_1 = 0$ , отсюда, с учётом (21), следует (17).

В соответствии с формулой (21) нетрудно показать, что знаменатель  $\Phi_{02}(z)$  – шуровский полином. Кроме того, при условии  $\Phi = \Phi_{02}$  функционал  $I_2$  принимает конечное значение  $I_2(\Phi_{02}) = \|R / G\|_2^2 + k^2 \|S_1 / G\|_2^2$ , следовательно  $\Phi_{02} \in \Omega_2^\Phi$ , что и доказывает теорему. ■

*Замечание: знак «~» обозначает обратный полином.*

На базе теоремы 1 [5] легко найти решение задачи (8) в виде передаточной функции  $W_0$  оптимального регулятора, а также исследовать ее существенные спектральные особенности. Действительно, подставляя формулу (17) в (10) после выполнения достаточно простых преобразований с учетом (9) и (16) получаем единственным образом

$$W_0(z) = \frac{[A(z)T(z)R(z) + z^{v-m}N(z)\tilde{B}(z)]/\tilde{G}(z)}{[B(z)T(z)R(z) - k^2N(z)\tilde{A}(z)]/\tilde{G}(z)}. \quad (22)$$

С учетом (20) легко видеть, что деление на полином  $\tilde{G}(z)$  здесь осуществляется нацело, а выражение  $\Delta_0(z) = -N(z)G(z)$  дает характеристический полином замкнутой системы (1), (2) причем минимум функционала определяется формулой  $J_{20} = \min_{W \in \Omega_2} J_2(W) = \|R / G\|_2^2 + k^2 \|S_1 / G\|_2^2$ .

#### 4. Синтез гарантирующих обратных связей

Теперь рассмотрим задачу  $H_\infty$ -оптимизации (7), для решения которой будем искать такую функцию  $\Phi(z) \in \Omega_\infty^\Phi$ , что

$$I_\infty(\Phi) = \|H(z, \Phi)\|_\infty^2 \leq \gamma, \quad (23)$$

где  $\gamma > 0$  – заданное вещественное число. Очевидно, что минимально достижимое  $\gamma$  в (23) определяет решение задачи (7).

Заметим, что в соответствии с (14) задача (23) принимает вид

$$I_\infty(\Phi) = \|H(z, \Phi)\|_\infty^2 = \max_{\omega \in [0, \pi]} \left[ |T_1(e^{j\omega}) - T_2(e^{j\omega})\Phi(e^{j\omega})|^2 + k^2 |T_3(e^{j\omega})|^2 \right] \leq \gamma. \quad (24)$$

Для решения задачи (24) введем вспомогательное число  $I_g$ :

$$I_g = \max_{\omega \in [0, \pi]} k^2 |T_3(e^{j\omega})|^2 = k^2 \max_{\omega \in [0, \pi]} |1 / G(e^{j\omega})|^2. \quad (25)$$

Будем говорить, что в задаче (24) имеет место регулярная ситуация, если для любых  $\Phi(z) \in \Omega_\infty^\Phi$  выполняется условие  $I_\infty(\Phi) > I_g$ . Если же существуют такие функции  $\Phi(z) \in \Omega_\infty^\Phi$ , что  $I_\infty(\Phi) = I_g$ , то ситуацию будем называть вырожденной.

**Теорема 2.** В регулярной ситуации задача (31) об ограничении нормы обобщённой передаточной функции  $H(z, \Phi)$  сводится к задаче о поиске такой функции  $\Phi(z) \in \Omega_\infty^\Phi$ , чтобы выполнялось условие  $\|(T_1 + T_2\Phi)P\|_\infty^2 \leq 1$ , где  $P(z) \in RH_\infty$  – весовой множитель, однозначно определяемый начальными данными.

**Доказательство.** Введём обозначение  $\gamma = \varepsilon + I_g$ , где  $\varepsilon > 0$  – такое число,

чтобы нашлись функции  $\Phi(z) \in \Omega_\infty^\Phi$ , обеспечивающие (31), т.е.

$$|T_1(e^{j\omega}) - T_2(e^{j\omega})\Phi(e^{j\omega})|^2 \leq \gamma - k^2|T_3(e^{j\omega})|^2 \quad \forall \omega \in [0, \pi]. \quad (26)$$

Согласно (25)  $\gamma - k^2|T_3(e^{j\omega})|^2 > 0 \quad \forall \omega \in [0, \pi]$ , поскольку  $\gamma > I_g$ . Тогда существует такая дробь  $L_\gamma(z)$ , что  $|L_\gamma(e^{j\omega})|^2 = \gamma - k^2|T_3(e^{j\omega})|^2 \quad \forall \omega \in [0, \pi]$ ,

$$L_\gamma(z)L_\gamma(z^{-1}) \equiv \gamma - \frac{k^2}{G(z)\overline{G(z)}}, \quad L_\gamma(z) \equiv \frac{R_\gamma(z)}{G(z)}, \quad (27)$$

где шуровский полином  $R_\gamma$  определяется факторизацией

$$R_\gamma(z)\overline{R_\gamma(z)} \equiv \gamma G(z)\overline{G(z)} - k^2. \quad (28)$$

Тогда имеем  $|T_1(e^{j\omega}) - T_2(e^{j\omega})\Phi(e^{j\omega})|^2 \leq |R_\gamma(e^{j\omega})/G(e^{j\omega})|^2 \quad \forall \omega \in [0, \pi]$  после подстановки (27) в (26), или  $\|T_1 - T_2\Phi\|_\infty^2 \leq 1$ , где  $P = G(z)/R_\gamma(z)$ , что и доказывает теорему. ■

**Теорема 3.** Задача о поиске функции  $\Phi(z) \in \Omega_\infty^\Phi$ , указанной в теореме 3, имеет решение для таких и только для таких величин  $\gamma > I_g$ , для которых знакоположительна эрмитова матрица  $L(\gamma) = \{l_{ij}(\gamma)\}$ , построенная для корней  $g_i$  полинома  $\tilde{G}(z)$ :

$$l_{ij} = (1 - \beta_i \overline{\beta_j}) / (1 - 1/g_i \overline{g_j}), \quad \beta_i = -\overline{B(g_i)} / (A(g_i)R_\gamma(g_i)), \quad i, j = \overline{1, v}. \quad (29)$$

**Доказательство.** Подставляя выражения  $P = G(z)/R_\gamma(z)$  и (15) в формулу  $Z(z, \gamma, \Phi) = P(z, \gamma)[T_1(z) - T_2(z)\Phi(z)]$ , получим

$$Z(z, \gamma, \Phi) = \frac{G(z)}{R_\gamma(z)} \left[ \frac{k^2 \alpha(z) \overline{A(z)} - \beta(z) \overline{B(z)}}{Q(z)G(z)} - \frac{\overline{G(z)}}{Q(z)} \Phi(z) \right], \quad (30)$$

откуда с учетом  $\overline{G(g_i)} = 0$  следует  $Z(g_i, \gamma, \Phi) = -\overline{B(g_i)} / [A(g_i)R_\gamma(g_i)]$ , поскольку из (16) имеем  $k^2 \overline{A(g_i)} = -B(g_i) \overline{B(g_i)} / A(g_i)$ . Тогда из (29)  $\beta_i = Z(g_i)$ ,  $i = \overline{1, v}$ , причем  $|g_i| > 1$ , и для функций  $\Phi$  должно быть

$$\|Z(z)\|_\infty \leq 1, \quad Z(g_i) = \beta_i, \quad i = \overline{1, v}, \quad (31)$$

что определяет задачу Неванлинны-Пика [2], причем необходимым и достаточным условием существования её решения является неотрицательная определённость эрмитовой матрицы Пика  $L(\gamma)$ . ■

**Теорема 4.** Пусть  $Z(z, \gamma)$  – любое решение задачи (31) Неванлинны-Пика при заданной величине  $\gamma > I_g$ . Тогда для него существует единственная функция  $\Phi = \Phi(z, \gamma) \in \Omega_\infty^\Phi$ , определяемая формулой

$$\Phi(z, \gamma) = \frac{-Z(z, \gamma)Q(z)R_\gamma(z) + k^2 \alpha(z) \overline{A(z)} - \beta(z) \overline{B(z)}}{G(z)\overline{G(z)}}, \quad (32)$$

для которой  $P(z, \gamma)[T_1(z) + T_2(z)\Phi_0(z, \gamma)] \equiv Z(z, \gamma)$ .

**Доказательство.** Пусть задано число  $\gamma > I_g$ , для которого  $L(\gamma) \geq 0$ , и найдено решение задачи (31). Тогда, в соответствии с (30),

$$\frac{G}{R_\gamma} \left( \frac{k^2 \alpha \overline{A} - \beta \overline{B}}{Q G} - \frac{\overline{G}}{Q} \Phi(z) \right) = Z, \quad (33)$$

откуда следует формула (32). Нетрудно показать на основании (41), (19) и (10), что числа  $g_i$  не входят в состав полюсов  $\Phi$ , поскольку одновременно являются и нулями этой функции. Согласно (24) имеем

$I_\infty(\Phi(z, \gamma)) = \|H(z, \Phi(z, \gamma))\|_\infty^2 \leq \gamma < \infty$ , т.е.  $H \in RH_\infty$ , и  $\Phi(z, \gamma) \in \Omega_\infty^\Phi$ . ■

Заметим, что минимальное число  $\gamma = \gamma_0 \geq I_g$  для условия (23), легко найти как минимальное решение уравнения  $\delta_{\min}(L(\gamma)) = 0$  на интервале  $[I_g, \infty)$ , где  $\delta_{\min}$  – минимальное собственное значение матрицы  $L(\gamma)$ . В силу особенностей задачи (31), при  $\gamma = \gamma_0$  она имеет одно решение ([2])

$$Z(z, \gamma_0) = m_1(z)/m_2(z), \quad (34)$$

причем  $m_1(z) = \tilde{m}_2(z)$ , где  $m_2(z)$  – шуровский полином.

**Теорема 5.** Решением задачи (7) гарантирующего синтеза в регулярной ситуации по отношению к функционалу (4) с учётом (5) является обратная связь (2) с передаточной функцией

$$W_{0\infty}(z) = \frac{W_{\infty 1}(z)}{W_{\infty 2}(z)} = \frac{[A(z)\tilde{m}_2(z)R_\gamma(z)z^\nu + z^{\nu-m}\tilde{B}(z)m_2(z)]/\tilde{G}(z)}{[B(z)\tilde{m}_2(z)R_\gamma(z)z^\nu - k^2\tilde{A}(z)m_2(z)]/\tilde{G}(z)}, \quad (35)$$

где полином  $m_2$  определяется решением (34) задачи (31) при условии  $\gamma = \gamma_0$ ,  $R_\gamma(z)$  – шуровский результат факторизации (28), где  $\gamma = \gamma_0$ .

**Доказательство.** Пусть найдено число  $\gamma = \gamma_0 > I_g$ , для которого построим передаточную функцию  $W_{0\infty} = L_\Phi(\Phi_0)$ :

$$W_{0\infty} = \frac{A\Phi_0 - \alpha}{B\Phi_0 + \beta} = \frac{-Q(AZR_\gamma + \bar{B})/\bar{G}}{-Q(BZR_\gamma - k^2\bar{A})/\bar{G}} \quad (36)$$

с учетом (9) и (16). Вводя обратные полиномы и учитывая (34), приходим к формуле (35). Аналогично доказательству теоремы 1, легко показать, что в ее числителе и знаменателе имеет место делимость на полином  $\tilde{G}(z)$ , а характеристический полином замкнутой системы представляется выражением  $\Delta_{0\infty} \equiv AW_{\infty 2} - BW_{\infty 1} \equiv -m_2G$ .

И, наконец, в соответствии с формулой (34), решение задачи (31) является *равномерно пропускающей (all pass)* функцией, поскольку  $|Z(e^{j\omega})| = |m_2(e^{-j\omega})|/|m_2(e^{j\omega})| \equiv 1$ . Используя формулы (8) и (35) легко показать, что таким же свойством обладает и обобщённая передаточная функция  $H_{0\infty}(z)$ , для которой  $|H_{0\infty}(e^{j\omega})|^2 \equiv \gamma_0$ . ■

### 5. Пример синтеза цифровых систем

Пусть задан дискретный линейный объект с моделью

$$(q^2 + 0.215q - 1.18)y[n] = (q + 0.990)u[n] + d[n], \quad n \in N^1, \quad (37)$$

для которого  $A(z) = z^2 + 0.215z - 1.18$ ,  $B(z) = z + 0.990$ ,  $\nu = 2$ ,  $m = 1$ . Введем спектральную плотность  $S_d$  с единичной  $H_2$ -нормой:

$$S_d(e^{j\omega})|_{e^{j\omega}=z} \equiv S_1(z)S_1(z^{-1}), \quad S_1(z) \equiv N(z)/T(z), \quad (38)$$

где  $N(z) = 0.301z + 0.255$  и  $T(z) = z^2 + 1.82z + 0.828$  – шуровские полиномы, и будем рассматривать функционал с весами  $k \in (0, 2]$ :

$$I = I(W) = \langle y^2 \rangle + k^2 \langle u^2 \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N (y^2[n] + k^2 u^2[n]). \quad (39)$$

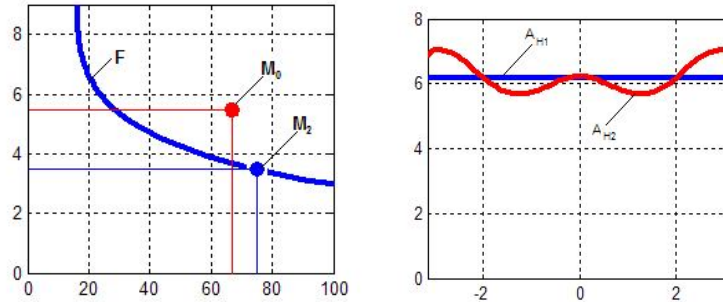
Решая задачу (6) для значения  $k = 0.153$ , последовательно получим:  
 $G = -1.05z^2 - 0.964z + 0.0263$ ,  $\tilde{G} = 0.0263z^2 - 0.964z - 1.05$ ,  $\bar{G} = \tilde{G}/z^2$ ,  $g_1 = 37.7$ ,  $g_2 = -1.06$ ,  
 $R(z) = 0.0133z - 0.509$ ,  $W_{01}(z) = 0.506z^3 + 0.194z^2 - 0.751z - 0.474$ ,  $W_{02}(z) = 0.506z^2 + 0.902z + 0.403$ .

Для объекта (37), замкнутого регулятором  $u = (W_{01}/W_{02})y$ , имеем следующие значения  $H_2$ -норм и минимизируемого функционала:

$$J_y = \|H_y S_1\|_2^2 = 3.41, \quad J_u = \|H_u S_1\|_2^2 = 77.9, \quad J_{20} = \|HS_1\|_2^2 = 5.23. \quad (40)$$

На рис. 1 показана кривая  $J_y = F(J_u)$  для оптимальных замкнутых систем при  $k \in (0, 2]$ , причем точка  $M_2$  соответствует (40).

Для решения задачи (7) при тех же исходных данных с весом  $k = 0.153$  используем полученные ранее полиномы  $G$  и  $\tilde{G}$  с корнями  $g_1, g_2$ . В соответствии с (25) находим  $I_g = 6.18$ , а решение уравнения  $\delta_{\min}(L(\gamma)) = 0$  приводит к гарантии  $\gamma_0 = 6.19$ .



**Рис. 1.** Кривая  $J_y = F(J_u)$ . **Рис. 2.** Функции  $A_{H1}(\omega)$  и  $A_{H2}(\omega)$

После выполнения факторизации (28) при условии  $\gamma = \gamma_0$  находим полиномы  $R_\gamma(z) = -2.54z^2 - 2.47z + 0.0673$  и  $m_2(z) = 11.3z - 0.300$  для решения (34) задачи Неванлинны-Пика (31). Далее по формуле (35) формируем функцию  $W_{0\infty}(z) = W_{\infty 1}(z)/W_{\infty 2}(z)$ :

$$W_{\infty 1}(z) = 29.1z^5 + 3.64z^4 - 33.3z^3 - 9.91z^2 + 0.286z,$$

$$W_{\infty 2}(z) = 29.1z^4 + 26.2z^3 - 0.892z^2 + 0.257z - 0.00671.$$

Соответствующая частотная характеристика  $A_{H1}(\omega) = |H_{0\infty}(e^{j\omega})|^2$  представлена на рис. 2, а величины  $J_y, J_u$  для найденного регулятора дают точку  $M_0$  на рис. 1.

Заметим, что полученные передаточные функции не являются правильными дробями, что делает невозможной их непосредственную реализацию. Однако положение может быть исправлено методами, указанными в работе [4]. В частности, в качестве приближения к оптимальной гарантии построена реализуемая передаточная функция  $W_g = W_{g1}/W_{g2}$ , где  $W_{g1} = 2.21z^2 - 0.627z - 2.84$ ,  $W_{g2} = z^2 + 3.39z + 2.39$ , с частотной характеристикой  $A_{H2}(\omega) = |H(e^{j\omega}, W_g)|^2$  (рис. 2).

## 6. Заключение

Вопросы оптимизации по нормам пространства Харди  $H_2$  и  $H_\infty$  в последние десятилетия находятся в центре особого внимания специалистов, занимающихся цифровыми системами управления и обработки сигналов. Тем не менее, в рамках  $H$ -теории продолжают интенсивные исследования по повышению эффективности алгоритмов синтеза и получаемых с их помощью решений.

В данной работе предложен один из вариантов экономичных

алгоритмов расчета оптимальных цифровых систем. Обратим особое внимание на тот факт, что синтез передаточной матрицы  $W(z)$  обратной связи, однозначно определяет соответствующий цифровой фильтр, обладающий линейным и стационарным свойством. Его реализация в виде программного кода не вызывает никакого труда и легко автоматизируется. В этом смысле привлечение оптимизационного подхода в рассматриваемом случае можно трактовать как автоматизированное формирование программного обеспечения, реализующего обратные связи в замкнутых цифровых системах.

### **Литература**

1. Поляк Б.Т., Щербаков П.С. Робастная устойчивость и управление. – М.: Наука, 2002. – 303 с.
2. Doyle J., Francis B., Tannenbaum A. Feedback control theory. New York: Macmillan Publ. Co., 1992. – 227 p.
3. Алиев Ф. А., Ларин В. Б., Науменко К. И., Сунцев В. Н. Оптимизация линейных инвариантных во времени систем управления. – К.: Наукова думка, 1978. – 327 с.
4. Веремей Е. И., Петров Ю. П. Метод синтеза оптимальных регуляторов, допускающий техническую реализацию // Математические методы исследования управляемых механических систем. – Л.: Изд-во Ленингр. ун-та, 1982. – С. 24 – 31.
5. Веремей Е.И. Алгоритмы решения одного класса задач  $H_\infty$ -оптимизации систем управления // Известия РАН. Теория и системы управления. — 2011. – № 3. – С. 52 – 61.

**Вэй Ян Лвин, Шагин А.В.**

д-р техн. наук  
Национальный исследовательский университет “МИЭТ”,  
Москва, Россия  
[waiyanlwin49@gmail.com](mailto:waiyanlwin49@gmail.com)

## ***Адаптивная система автоматического управления движением***

Во всем мире по мере развития автомобильной промышленности растут плотность транспортных потоков на дорогах и средняя скорость движения автомобилей, что ведет к увеличению числа автомобильных катастроф. Как правило, причина несчастных случаев – недостаточная информированность водителя о ситуации на дороге, чему способствует ограниченная зона оптической видимости, особенно в ночное время суток, в дождь, туман, снег, при дыме, пыли и т.п. Поэтому сегодня ученые должны снизить вероятность возникновения аварийных ситуаций, а в будущем полностью заменить водителя.

В данной работе рассматривается адаптивная система автоматического управления движением автомобиля, обосновывается выбор её составных частей. В работе предложена структура адаптивной системы автоматического управления движением, проведена разработка функциональной схемы системы управления антенной, проведено моделирование преобразования частоты для решения задач управления шаговым двигателем.

Основу таких информационных систем составляют радиоэлектронные и оптические средства, обеспечивающие обзор и впереди и, сзади и, сбоку автомобиля. В зависимости от сложности и вариантов решения задач разрабатываемые информационные системы для транспортных средств делятся на типы.

Все недостатки, присущие системам такого типа, исключены в системе, которая использует многолучевые датчики и локаторы, работающие с высокой скоростью обзора пространства. Обладая высокой информативностью, она обеспечивает автоматическое управление автомобилем практически в любых дорожных ситуациях и погодных условиях. Система представляет собой качественно новый этап развития радиолокационных систем, предназначенных для применения на наземном транспорте. Здесь объединены достижения в области радиолокации, антенной техники мм-диапазона длин волн, а также программно-аппаратных средств цифровой обработки радиолокационных сигналов. Система радиовидения открывает новый этап в управлении движением транспортных средств в условиях ограниченной оптической видимости и создает предпосылки для создания системы автоматического управления

транспортными средствами. Разработанная система радиовидения предназначена для управления автомобилем в условиях ограниченной оптической видимости, когда движение автомобиля при визуальном контроле либо по данным оптических датчиков затруднено, а порой невозможно. Таким образом, не только повышается безопасность движения автомобиля в условиях ограниченной оптической видимости, но и открываются совершенно новые возможности: управление движением в отсутствии оптической видимости.

Общий принцип построения и работы антенной и приемопередающей частей основан на традиционной схеме импульсной РЛС кругового обзора. Антенна имеет веерообразную диаграмму направленности, т. е. узкую в азимутальной и достаточно широкую в угломестной плоскостях, рис(1).



Рис.1. Компоненты в автомобиле

При вращении такая антенна обеспечивает не только требуемый обзор в вертикальной и горизонтальной плоскостях, но и измерение азимутальных координат объектов локации. Этот образец конструктивно выполнен в виде трех функциональных модулей (рис.2):

- антенного и приемопередающего модуля (АПМ),
- специализированного вычислительного модуля (СВМ),
- индикаторного устройства (ИУ).

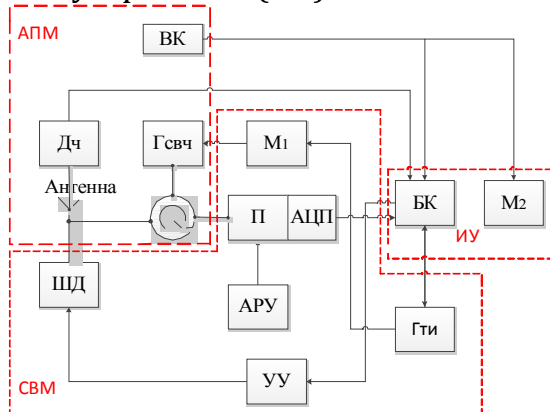


Рис.2 Структура адаптивной системы автоматического управления движением

ДЧ - датчик  
 ШД - шаговый электродвигатель  
 ВК - видеокамера  
 Гсвч- генератор (сверхвысокая частота)  
 М1 - модулятор



АРУ - автоматическая регулировка усиления  
УУ - устройство управления (ШД)  
П(ацп)- процессор(аналого-цифровой преобразователь)  
БК - бортовой компьютер  
Гти - генератор тактовых импульсов  
М2 - монитор

Приёмопередающий модуль (АПМ) включает в себя датчик, антенну, генератор (сверхвысокой частоты) и видеокамеру.

Специализированный вычислительный модуль (СВМ) состоит из АРУ, процессора, включающего АЦП, модулятора, генератора тактовых импульсов, устройства управления (ШД).

Индикаторное устройство (ИУ) включает в себя монитор и бортовой компьютер. Антенный и приёмопередающий модуль и видеокамера устанавливаются на крыше. Вычислительный и индикаторный модули устанавливаются в автомобиле. Сигналы с видеокамеры поступают на бортовой компьютер и монитор. Бортовой компьютер управляет процессами сбора и преобразования информации и контролирует устройство управления (ШД) вращением антенны. Датчик измеряет угловой шаг антенны и посылает эту информацию бортовому компьютеру. Генератор тактовых импульсов формирует периодическую последовательность импульсов, которые воздействуют на модулятор. Видеоимпульсы последнего модулируют СВЧ-колебания генератора, в результате чего образуются радиоимпульсы, которые поступают в антенну. Отраженные от объектов радиоимпульсы поступают на вход приемника с некоторой задержкой, после чего с помощью АЦП и бортового компьютера преобразуются в двухмерное радиолокационное изображение дорожной ситуации.

Системы подобного класса адекватно решают задачу радиовидения автодороги и могут быть использованы для управления автомобилями в условиях ограничения или полного отсутствия оптической видимости, формируемые радиолокационные изображения дают полное представление об обстановке на трассе. Легковые автомобили устойчиво наблюдаются на дальностях до 100–150 м, а грузовые – до 200–250 м, также хорошо различаются дорожные знаки и ограждения.

Для повышения характеристик проводится его техническое усовершенствование, заключающееся в разработке новой антенны с электронным сканированием, в переходе на 4-мм диапазон длин волн и в применении частотной модуляции. Это повысит разрешающую способность радара до 2 м, предельную дальность - до 500 м. При этом масса СВЧ-модуля снизится до 1 кг, а общая масса – до 2 кг.

Совокупность всех этих технических новшеств позволит создать малогабаритный промышленный образец радиолокационной системы с высокими тактико-техническими характеристиками, обеспечивающими решение задачи управления автомобилем в условиях отсутствия

оптической видимости дорожной обстановки, а также приступить к созданию системы автоматического управления автомобилем.

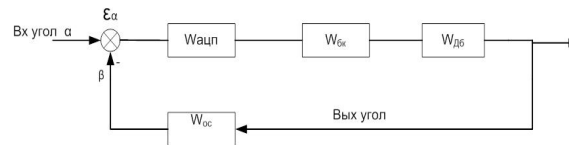


Рис.3 Функциональная схема системы управления антенной

Где  $W_{\text{АЦП}}$  - передаточная функция аналого-цифрового преобразования;  $\varepsilon_{\alpha}$  - передаточная функция цифрового сравнивающего устройства и рассогласования (отклонения);  $W_{\text{БК}}$  - передаточная функция блока бортового компьютера;  $W_{\text{ДБ}}$  - передаточная функция двигателя;  $W_{\text{ОС}}$  - передаточная функция обратной связи на рисунке.

Задающий угол  $\alpha$  сигнала и работающий угол  $\beta$  сигнала сравниваются,  $W_{\text{АЦП}}$  (аналого-цифровое преобразование) преобразует сигнал, бортовой компьютер получает сигнал и формирует сигнал управления двигателем. По цепи обратной связи передаётся сигнал угла на сравнивающее устройство.

Передаточная функция системы автоматического управления (САУ) с обратной связью(-) может быть представлена выражением:

$$W_{(p)} = W_{\text{АЦП}} * W_{\text{БК}} * W_{\text{ДБ}} / (1 + W_{\text{АЦП}} W_{\text{БК}} W_{\text{ДБ}} W_{\text{ОС}})$$

Рассматриваемая адаптивная система автоматического управления движением, обеспечивает возможность изменять параметры в зависимости от воздействий действующих на объект управления. Проведён анализ и моделирование особенностей построения и функционирования адаптивной системы автоматического управления движением от режимов управления антенной. Создан и разработан комплекс математических моделей для оценки скорости вращения антенной. Для определения характеристик системы управления проведено математическое моделирование в пакете MATLAB. Модель имеет два варианта контроллера: один для управления положением, другой для управления скоростью, рис.4 а,б.

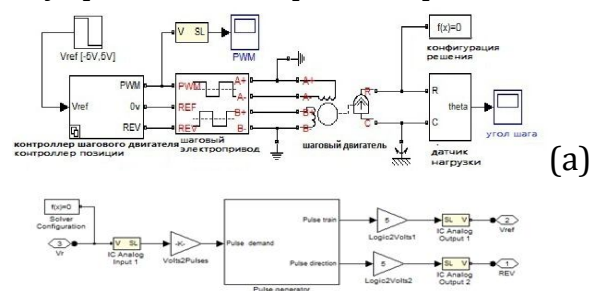
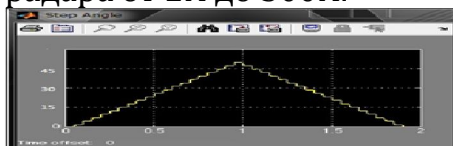


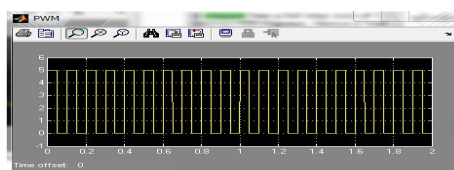
Рис.4. Моделирование управлением ШД на MATLAB(b).

Шаговый двигатель - имеет полный размер шага, равный 1.8 степени. При контроле за положением  $V_{\text{ref}} = +5V$  соответствует 100 шагам, то есть 180 степеням.. При регулировке скорости  $V_{\text{ref}} = +5V$  соответствует 100

шагам(180 степеням) в одну секунду. Следовательно, если  $V_{ref} = +0.5V$ , то за одну секунду вал двигателя пройдет 18 степеней. Если автомобиль постоянно движется со скоростью 30 миль(1609 метров) в час, то за одну секунду он проедет почти 13метров. Что соответствует 180 степеням в секунду. Рабочий диапазон радара от 2м до 500м.



*Рис.6 .Результат моделирования углового шага*



*Рис.7. Выходной сигнал широтно-импульсного модулятора*

На этом рисунке показаны результаты моделирования углового шага и широтно-импульсного модулятора. Модель на уровне системы удобна для изучения адаптивной системы с шаговым двигателем и определения уменьшится ли угол шага, при данной нагрузке. Она может также использоваться для настройки контроллера ШД с целью улучшения работы.

Таким образом в работе достигнуты следующие результаты. Проведено моделирование работы системы управления ШД в программном пакете MATLAB, позволяющее оценить параметры вращения антенны в зависимости от скорости движения автомобиля.

### **Литература**

1. Савин М.М., Елсуков В.С., Пятин О.Н. Теория автоматического управления. Высшее образование, 2007.
2. Сенигов П.Н. Теория автоматического управления. Конспект лекций. М., 2000.
3. Расторгуев В, Нуждин В., Сидоров Н., Сулимов Ю. Система радиовидения « Авто радар» управление движением автомобиля. Наука, технология, Бизнес 5/2000.
4. Дискретный электропривод с шаговыми двигателями, под общ. ред . М. Г. Чиликина. М.: Энергия, 1971. Авторы: Б. А. Ивоботенко, В. П. Рубцов, Л. А. Садовский, В. К. Цаценкин, М. Т. Чиликин.
4. Вэй Ян Лвин. Выбор и обоснование составных частей адаптивной системы автоматического управления автомобилем // Микроэлектроника и информатика-тезисы докладов секция-7 (2011г).

## ***Некоторые аспекты структуризации сеансов спутниковой связи***

### **Аннотация**

*В докладе рассмотрены некоторые аспекты организации систем спутниковой связи, связанные со структуризацией сеансов, как в режиме немедленной ретрансляции сигналов, так и с использованием каналов межспутниковой коммуникации. Предложены алгоритмы составления временной программы сброса информации, а также расписаний сеансов связи наземных станций.*

### **1. Введение**

К настоящему времени известно несколько видов систем спутниковой связи, отличающихся, в первую очередь, построением космического сегмента. К ним относятся системы с космическими аппаратами на геостационарной, эллиптических и низких круговых орбитах, каждая из которых имеет много разновидностей.

В настоящее время в России и за рубежом развернуты работы по созданию телекоммуникационных систем, использующих ретрансляторы на низкоорбитальных космических аппаратах с высотой орбит 800-1500 км. Создание спутниковых систем связи на низких орбитах обусловлено:

- перегруженностью геостационарной орбиты, приводящей к значительным ограничениям при создании новых систем по точкам стояния космического аппарата (КА) и параметров каналов ретрансляции;
- возможностью использования эффекта Доплера для определения местоположения объектов;
- практическое использование на линии КА-Земля более низких частот, что обеспечивает возможность работы абонента на обычных всенаправленных антеннах, а также существенное повышение энергетических характеристик радиолинии за счет уменьшения расстояния от приемо-передающей станции до КА.

Отдельные спутники в конкретный момент времени могут обеспечить охват только ограниченных областей. Расчет указанных зон радиовидимости (ЗРВ) районов может быть осуществлен с помощью формул, описанных в работе [1]. Для расширения зоны покрытия, космический сегмент должен содержать множество спутников. Систему спутников называют орбитальной группировкой. В большинстве коммерческих спутниковых систем связи (ССС) космический сегмент

состоит из спутников с одним типом орбит.

Будем считать, что все корреспонденты, обслуживаемые данной ССС, условно разбиты на группы, обладающие определенными сбросовыми адресами. Кроме того, полагаем, что каждый корреспондент имеет заданный приоритет. Возникает задача составления временной программы сброса сообщений корреспондентам с учетом имеющихся ограничений, обусловленных различными техническими причинами.

## 2. Задача составления программы сброса сообщений

Пусть определено множество

$$\Omega_0 = \{ [t_{nj}^z, t_{kj}^z] \mid j=1,2,\dots,s, t_{nj}^z \leq t_{kj}^z \},$$

где  $j$  — условный номер ЗРВ,  $t_{nj}^z$  — время начала  $j$ -ой ЗРВ, а  $t_{kj}^z$  — время ее конца. Для любого номера  $j = 1, 2, \dots, s$ , считаются заданными точки (время)  $t_{0j}^{(m)} \in [t_{nj}^z, t_{kj}^z]$ ,  $m=1,2$ , которые являются наиболее удобными (оптимальными) точками сброса (ТС) сообщения для соответствующего корреспондента. Указанные точки будем называть опорными.

Назовем сеансом сброса отрезок времени

$$[t_{nj}^c, t_{kj}^c], t_{nj}^c > t_{nj}^z, t_{kj}^c < t_{kj}^z,$$

на котором разрешено назначать точку сброса.

Приоритетность ТС зависит от приоритетности сбросового адреса, от приоритетности корреспондентов и от количества корреспондентов, имеющих на своих сеансах связи единственную опорную точку. Корреспонденты, принадлежащие одному и тому же сбросовому адресу, могут объединяться и обслуживаться одной общей точкой сброса.

Пусть

$$\Omega = \{ [t_{nj}^c, t_{kj}^c] \mid j=1,2,\dots,r, t_{nj}^c \leq t_{kj}^c \},$$

— множество всех сеансов, причем  $r \geq s$ . Каждой точке сброса с номером  $i$  можно поставить в соответствие некоторое положительное число  $p_i$ , характеризующее приоритет точки сброса, приоритет соответствующего корреспондента и его сбросового адреса.

Точка  $t_q$ ,  $q \in Q$  называется ориентиром, если существует, по крайней мере, один номер  $i_q$  сеанса сброса, при котором  $t_q \in [t_{ni_q}^c, t_{ki_q}^c]$ .

Ясно, что в общем случае существует не единственный подобный номер  $i_q$ .

Пусть  $I_q$  — совокупность всех указанных номеров. С каждым ориентиром естественно будут связаны следующие параметры:

- суммарный вес  $p_q = \sum_{i \in I_q} p_i$ ;
- суммарное отклонение данного ориентира от соответствующих опорных точек  $u_q = \sum_{i \in I_q} |t_q - t_{0i}|$ .

Для каждого  $q$  можно выделить некоторый интервал постоянства, в котором указанные параметры не изменяются. В самом деле, таким

интервалом будет служить отрезок  $[t_{nq}, t_{kq}]$ , где

$$t_{nq} = \max_{i \in I_q} t_{ni}^c, \quad t_{kq} = \min_{i \in I_q} t_{ki}^c.$$

Таким образом, задача составления временной программы сброса сообщений сводится к выбору такого подмножества  $Q'$  из множества  $Q$ , при котором одновременно будут выполнены следующие условия:

$$\bigcap_{k=1}^L I_{q_k} = \emptyset; \quad (1)$$

$$\sum_{k=1}^L p_{q_k} \rightarrow \max; \quad (2)$$

$$\sum_{k=1}^L u_{q_k} \rightarrow \min; \quad (3)$$

$$t_{q_m} + \Delta t \leq t_{q_{m+1}} \leq t_{q_{m+2}} - \Delta t, \quad m = 1, 2, \dots, L-2, \quad (4)$$

причем число точек сброса  $L$  заранее не фиксируется,  $\Delta t$  — минимально допустимое расстояние по времени между точками сброса,

### **3. Алгоритм составления временной программы**

Укажем один из возможных алгоритмов решения поставленной задачи (1) – (4). Сначала заметим, что если  $\bigcap_{j=1}^s [t_{nj}^c, t_{kj}^c] = \emptyset$ , то ее решение почти тривиально. Предположим, что множество  $\bigcup_{j=1}^s [t_{nj}^c, t_{kj}^c]$  является связным, следовательно, не существует такого  $i_1 \in \{1, 2, \dots, s\}$ , что

$$[t_{ni_1}^c, t_{ki_1}^c] \cap \left\{ \bigcup_{\substack{j=1, s \\ j \neq i_1}}^s [t_{nj}^c, t_{kj}^c] \right\} = \emptyset.$$

Пусть  $t_n = \min_i t_{ni}^c$ ,  $t_k = \max_i t_{ki}^c$ . Выберем на интервале  $[t_n, t_k]$  максимально возможное число ориентиров, удовлетворяющих условию (4). Каждому ориентиру  $t_q$  поставим в соответствие только тот интервал, для которого  $u_q$  принимает наименьшее значение. Тем самым, получим указанное выше множество  $I_q$ . Для каждого ориентира подсчитываем суммарный вес и определяем интервал постоянства. Заметим, что выбор точки контакта  $t_{c_1}$  на одном из них, разбивает  $[t_n, t_k]$  на два интервала  $[t_n, t_{c_1} - \Delta t]$  и  $[t_{c_1} + \Delta t, t_k]$ . На каждом из них опять ставится задача выбора точек  $t_{c_1}$  и  $t_{c_2}$ , что приводит к очередному разбиению каждого из них на два интервала и т. д.

В результате получаем рекурсивную процедуру, с помощью которой находится окончательное решение задачи.

### **4. Структуризация сеансов связи с межспутниковой маршрутизацией**

Новые спутниковые системы способны взаимодействовать, минуя наземную станцию, что позволяет повысить надежность и сократить задержку в передаче данных. В этом случае говорят о ретрансляции с использованием межспутниковых линий связи (МЛС). При этом для каждого космического аппарата обеспечена возможность связи со

спутниками на одной с ним орбите (спереди и сзади), а также с двумя спутниками в соседних орбитальных плоскостях.

Информация о каждом корреспонденте, обеспечиваемом спутниковой связью, содержится в памяти шлюзовой станции (ШС) наземного пункта, к которой приписан данный корреспондент.

При построении плана работы системы связи полагаем, что любой спутник в каждый момент времени находится в зоне действия не более чем одного наземного пункта, а каждая шлюзовая станция «видит» на сеансе только один КА.

Пусть

$$\bar{\Omega}_p^r = \{[\bar{\tau}_{ni}^r, \bar{\tau}_{ki}^r] \mid i = 1, 2, \dots, s, \bar{\tau}_{ni}^r \leq \bar{\tau}_{ki}^r\}$$

— множество сеансов корреспондентов, оставшихся в маршруте на сеансе  $[t_n^{pr}, t_k^{pr}]$  шлюзовой станции  $p$ -ого наземного пункта ( $p = 1, 2, \dots, N$ ) при взаимодействии с  $r$ -ым спутником ( $r = 1, 2, \dots, M$ ). Тогда для любого  $i = 1, 2, \dots, s$  выполняется соотношение  $[\bar{\tau}_{ni}^r, \bar{\tau}_{ki}^r] \cap [t_n^{pr}, t_k^{pr}] \neq \emptyset$ .

Введем в рассмотрение множество

$$\tilde{\Omega}_p^w = \{[\tilde{\tau}_{ni}^w, \tilde{\tau}_{ki}^w] \mid i = 1, 2, \dots, l, \tilde{\tau}_{ni}^w < \tilde{\tau}_{ki}^w\}$$

— множество сеансов корреспондентов, обслуживаемых  $w$ -ым спутником ( $w \neq r$ ), причем для любого  $i = 1, 2, \dots, l$  и любых допустимых  $\alpha \neq p, \beta \neq r$  выполнены условия

$$[\tilde{\tau}_{ni}^w, \tilde{\tau}_{ki}^w] \cap [t_n^{\alpha\beta}, t_k^{\alpha\beta}] = \emptyset$$

и

$$[\tilde{\tau}_{ni}^w, \tilde{\tau}_{ki}^w] \cap [t_n^{pr}, t_k^{pr}] \neq \emptyset.$$

В силу сделанных предположений корреспонденты, соответствующие множеству  $\tilde{\Omega}_p^w$ , в режиме немедленной ретрансляции (НР) обслужены быть не могут. На сеансе  $[t_n^{pr}, t_k^{pr}]$  ШС наземного пункта указанные корреспонденты обеспечиваются связью лишь с использованием МЛС.

НР-группой под номером  $j$  с началом в точке  $t_n^{(j)}$  и концом в точке  $t_k^{(j)}$  называется совокупность из  $s_j$  корреспондентов множества  $\bar{\Omega}_p^r$ , для которых справедливо соотношение

$$[t_n^{(j)}, t_k^{(j)}] \subset [\bar{\tau}_{ni_\alpha}^r, \bar{\tau}_{ki_\alpha}^r], \alpha = 1, 2, \dots, s_j,$$

и, кроме того,  $t_k^{(j)} - t_n^{(j)} \geq \Delta T_{\text{НР}}$ , где  $\Delta T_{\text{НР}}$  — минимально допустимая длительность группы немедленной ретрансляции,  $\Delta T_{\text{НР}} = \Delta T_{\text{П}} + \Delta \tau$ . Здесь  $\Delta \tau$  — минимально допустимое время обслуживания одного корреспондента в режиме НР, а  $\Delta T_{\text{П}}$  — время, необходимое для подготовки к проведению сеанса немедленной ретрансляции.

Необходимо сформировать не более  $q$  НР-групп, содержащих максимально возможное количество корреспондентов, с учетом того, что один корреспондент может входить только в одну группу. Следовательно, необходимо, чтобы

$$\sum_{j \leq q} s_j \rightarrow \max$$

при условии

$$t_k^{(j)} - t_n^{(j)} \geq s_j \Delta \tau + \Delta T_{\text{п}}.$$

Кроме НР-групп на сеансе ШС формируется группа корреспондентов, обеспечивающихся с использованием межспутниковых линий связи (МЛС-группа). Длительность этой группы составляет не менее  $\Delta T_{\text{МЛС}}$ , где  $\Delta T_{\text{МЛС}}$  имеет фиксированное значение. Эта группа формируется из корреспондентов множества  $\tilde{\Omega}_p^w$ .

Если  $\Delta T_{\text{МЛС}} \leq t_k^{pr} - t_n^{pr} \leq \Delta T_{\text{МЛС}} + \Delta T_{\text{НР}}$ , то НР-группы не формируются, а длительность МЛС-группы полагается равной длительности сеанса ШС берегового пункта, т.е.

$$t_k^{(\text{МЛС})} - t_n^{(\text{МЛС})} = t_k^{pr} - t_n^{pr}.$$

Рассмотрим один из возможных способов построения НР-групп. Разобьем множество  $\bar{\Omega}_p^r$  на четыре множества, исходя из взаимного расположения сеанса корреспондента относительно сеанса ШС наземного пункта. Если для некоторого допустимого  $i$  справедливы неравенства  $\bar{\tau}_{ni}^r < t_n^{pr}$  и  $\bar{\tau}_{ki}^r < t_k^{pr}$ , то полагаем  $\bar{\tau}_{ni}^r = t_n^{pr}$  и включаем его в множество

$$\Omega_1 = \{[\bar{\tau}_{ni}^r, \bar{\tau}_{ki}^r] \mid \bar{\tau}_{ni}^r = t_n^{pr}, \bar{\tau}_{ki}^r \in (t_n^{pr}, t_k^{pr})\}.$$

Если выполнены соотношения  $\bar{\tau}_{ni}^r > t_n^{pr}$  и  $\bar{\tau}_{ki}^r > t_k^{pr}$ , то будем считать, что  $\bar{\tau}_{ki}^r = t_k^{pr}$  и включаем такого корреспондента в множество

$$\Omega_2 = \{[\bar{\tau}_{ni}^r, \bar{\tau}_{ki}^r] \mid \bar{\tau}_{ki}^r = t_k^{pr}, \bar{\tau}_{ni}^r \in (t_n^{pr}, t_k^{pr})\}.$$

Если некоторого корреспондента  $\bar{\tau}_{ni}^r \leq t_n^{pr}$  и  $\bar{\tau}_{ki}^r \geq t_k^{pr}$ , то полагаем  $\bar{\tau}_{ni}^r = t_n^{pr}$ ,  $\bar{\tau}_{ki}^r = t_k^{pr}$  и, тем самым, включаем его в множество

$$\Omega_3 = \{[\bar{\tau}_{ni}^r, \bar{\tau}_{ki}^r] \mid \bar{\tau}_{ni}^r = t_n^{pr}, \bar{\tau}_{ki}^r = t_k^{pr}\}.$$

Наконец, все оставшиеся корреспонденты из множества  $\bar{\Omega}_p^r$  попадут в множество  $\Omega_4 = \{[\bar{\tau}_{ni}^r, \bar{\tau}_{ki}^r] \mid \bar{\tau}_{ni}^r \in (t_n^{pr}, t_k^{pr}), \bar{\tau}_{ki}^r \in (t_n^{pr}, t_k^{pr})\}$ .

Понятно, что если в  $\Omega_i$  содержится  $n_i$  элементов,  $i = \overline{1,4}$ , то  $n_1 + n_2 + n_3 + n_4 = s$ . Кроме того, интервал  $[t_n^{pr}, t_k^{pr}]$ , соответствующий сеансу ШС  $p$ -го берегового пункта, разбивается точками  $\bar{\tau}_{ni}^r$  и  $\bar{\tau}_{ki}^r$  на совокупность подинтервалов  $[\tau_q, \tau_{q+1}]$ .

Построение любой НР-группы основывается на том, что группа может располагаться лишь на объединении соседних интервалов

$$\bigcup_{i=i_1}^{i_1+v} [\tau_i, \tau_{i+1}].$$

Поэтому при формировании группы сначала проверяется условие: нельзя ли создать группу на одиночных интервалах, затем на объединениях по два, по три и т.д.

Опишем алгоритм формирования группы на некотором промежуточном этапе, когда исследуется возможность построения группы на объединениях  $(v+1)$  интервалов.

Для каждого допустимого  $i$  рассматривается интервал  $[\tau_i, \tau_{i+v+1}]$  и вычисляется наибольшее целое число  $k_g^{(i)}$ , для которого выполняется



неравенство  $\tau_{i+v+1} - \tau_i \geq k_g^{(i)} \Delta \tau + \Delta T_{\Pi}$ . Если величина  $k_g^{(i)} < 1$ , то на отрезке  $[\tau_i, \tau_{i+v+1}]$  НР-группу создать нельзя, а поэтому переходим к следующему интервалу  $[\tau_{i+1}, \tau_{i+v+2}]$ . Понятно, что  $k_g^{(i)}$  общее число корреспондентов, которое может «обслужить» интервал  $[\tau_i, \tau_{i+v+1}]$ .

Первыми просматриваются элементы множества  $\Omega_1$ , так как в нем содержатся наиболее быстро заканчивающиеся сеансы корреспондентов, и для них проверяется условие включения корреспондента в группу. Если окажется, что в результате оказались отобранными  $k_g^{(i)}$  корреспондентов, то переходим к построению варианта следующей группы.

Если перебор всех отрезков из указанного множества не позволил отобрать  $k_g^{(i)}$  корреспондентов, то последовательно друг за другом рассматриваются элементы множества  $\Omega_4$ , затем  $\Omega_3$ , и последними — отрезки из множества  $\Omega_2$ .

Допустим, что в итоге в первую группу было отобрано  $l_g^{(i)}$  корреспондентов. Если  $l_g^{(i)} < s$ , то в множестве  $\bar{\Omega}_p^r$  еще остались «необслуженные» корреспонденты, поэтому переходим к выбору варианта второй группы. Зафиксируем интервал

$$[\tau_i, \tau_i + l_g^{(i)} \Delta \tau + \Delta T_{\Pi}],$$

$l_g^{(i)} \leq k_g^{(i)}$ , т.е. устанавливаем длительность первой группы в точности соответствующей числу корреспондентов, приписанных к ней.

Построение второй группы начинается с необходимого отступа от конца первой группы на величину  $\Delta \tau_g$ . Таким образом, полагаем, что вторая группа может быть создана на интервале

$$[\tau_i + l_g^{(i)} \Delta \tau + \Delta T_{\Pi} + \Delta \tau_g, t_k^{pr}],$$

если его длительность не меньше  $\Delta T_{НР}$ .

Все корреспонденты, не вошедшие в первую группу, опять разбиваются на четыре множества аналогично тому, как были построены множества  $\Omega_i$ . Данный интервал разбивается граничными точками временных отрезков, соответствующих указанным корреспондентам, на интервалы  $[\theta_i, \theta_{i+1}]$ . По-прежнему считаем, что группа может быть сформирована на объединении  $(\mu+1)$  соседних интервалов. Если  $\mu=0$ , то рассматриваются одиночные интервалы, если  $\mu=1$ , то — объединение по два,  $\mu=2$  — по три и т.д. Рассмотрим процесс построения группы более подробно на примере фиксированного значения  $\mu$ .

Для любого допустимого  $i$  определим число  $m_g^{(i)}$  корреспондентов, для каждого из которых справедливо соотношение  $[\theta_i, \theta_{i+\mu+1}] \subset [\bar{\tau}_{m_a}, \bar{\tau}_{k_{i_a}}]$ ,  $\alpha=1,2,\dots, m_g^{(i)}$ . Пусть  $m_g = \max_i m_g^{(i)}$ , а  $n_g^{(i)}$  — наибольшее целое число, для которого справедливо неравенство

$$\theta_{i+\mu+1} - \theta_i \geq n_g^{(i, \mu)} \Delta \tau + \Delta T_{\Pi}.$$

Очевидно, что  $n_g^{(i, \mu)}$  — максимальное число корреспондентов, которых можно «обслужить» на этом интервале. Если  $n_g^{(i, \mu)} \geq 1$ , то длительность

указанного интервала достаточна для создания на нем группы. Пусть  $n_g = \max_{i, \mu} n_g^{(i, \mu)}$ . Тогда если выполнено  $n_g \geq m_g$ , то вариантом второй группы становится интервал, на котором достигнут максимум.

В указанном процессе с увеличением числа объединяемых интервалов величина  $m_g$  уменьшается. Величина  $n_g$  чаще всего равная нулю на одиночных интервалах, так как их длительность недостаточна для обслуживания всех корреспондентов, «приписанных» к этому отрезку, становится отличной от нуля при объединении соседних интервалов. В таком случае почти всегда существует  $\mu$ , для которого  $m_g = n_g$ . Если не найдено такое  $\mu$ , то в качестве варианта второй группы берется наиболее длинный интервал, которому соответствует число корреспондентов большее, чем он может «обслужить». В этом случае во вторую группу включаем максимально возможное на данном интервале число корреспондентов. На этом заканчивается построение варианта второй группы. Далее вычисляется общее число корреспондентов, включенных в первые две группы, и сравнивается с аналогичным числом, полученным на предыдущих шагах алгоритма, и из них выбирается наибольшее. Таким образом, осуществляем перебор всевозможных вариантов создания двух групп обслуживания корреспондентов, затем трех и т.д.

Рассмотрим теперь процесс формирования МЛС-группы, куда входят корреспонденты из множества  $\tilde{\Omega}_p^w$ , т.е. корреспонденты которых невозможно обслужить с помощью немедленной ретрансляции. В данном случае используется алгоритм маршрутизации, основанный на периодическом характере изменений в орбитальной группировке. Вся шкала времени делится на интервалы стационарности, определяемые таким образом, что топология меняется только в начале и конце интервалов, оставаясь постоянной до начала следующего интервала. На каждом таком интервале можно решать задачу статической маршрутизации уже известными методами. Результатом ее решения для каждого интервала является соответствующая таблица маршрутизации, которая отправляется спутнику в заголовке данных в соответствующем интервале.

## Литература

1. Ермолин В.С. Определение временных параметров зон видимости района. /Тр. Второй Международной конференции «Дифференциальные уравнения и их приложения». Саранск, 1996. С. 71–77.
2. Еремеев В.В., Ермолин В.С. Алгоритмы назначения точек контакта. /Тр. Третьей Международной конференции «Дифференциальные уравнения и их приложения». Саранск, 1998. С. 130–131.
3. Еремеев В.В., Реброва Я.В. Структуризация сеансов космической связи с межспутниковой маршрутизацией. /Тр. XXXVIII научной конференции «Процессы управления и устойчивость». СПб, 2007. С. 440–445.

**Жабко А.П.,**

Санкт-Петербургский госуниверситет, профессор,  
[zhabko@apmath.spbu.ru](mailto:zhabko@apmath.spbu.ru)

**Медведева И.В.**

Санкт-Петербургский госуниверситет, аспирантка  
[medv.irina@gmail.com](mailto:medv.irina@gmail.com)

## ***Оценка области асимптотической устойчивости решений дифференциально-разностных систем запаздывающего типа***

### **Аннотация**

*В данной работе предлагается метод оценки области асимптотической устойчивости решения стационарной системы, имеющей экспоненциально устойчивое линейное приближение. В основу положен конструктивный метод построения квадратичного функционала Ляпунова-Красовского для системы линейного приближения.*

### **Введение**

Рассмотрим систему стационарных дифференциально-разностных уравнений запаздывающего типа

$$\dot{y}(t) = f(y(t), y(t-h_1), \dots, y(t-h_m)),$$

где  $y \in R^n$ ,  $0 = h_0 < h_1 < \dots < h_m = h$  – постоянные запаздывания, векторная функция  $f(z_0, \dots, z_m)$  определена, непрерывна и непрерывно дифференцируема в  $R^{n \times (m+1)}$ . Предположим, что эта система имеет нулевое решение, и линейное приближение в окрестности нулевого решения есть экспоненциально устойчивая система

$$\dot{x}(t) = \sum_{j=0}^m A_j x(t-h_j).$$

Далее в этом сообщении мы будем, опираясь на квадратичные функционалы Ляпунова-Красовского, построенные для системы линейного приближения, оценивать область асимптотической устойчивости нулевого решения исходной системы.

Во втором пункте приводятся предварительные сведения, необходимые при построении квадратичных функционалов для линейных систем. В третьем пункте излагается теоретическое обоснование метода и конструктивный алгоритм проверки положительной определенности построенных функционалов. Четвертый пункт посвящен основному результату доклада, а именно, алгоритму приближения области асимптотической устойчивости. В пятом пункте иллюстрируются возможности предложенного алгоритма на примере оценки критических значений запаздывания для случая системы с одним сосредоточенным запаздыванием.

### Обзор результатов для линейных систем

Рассмотрим проблему анализа устойчивости линейных стационарных систем с запаздывающим аргументом вида

$$\dot{x}(t) = \sum_{j=0}^m A_j x(t-h_j), \quad (1)$$

где  $x \in R^n$ ,  $A_j \in R^{n \times n}$ ,  $j=0, \dots, m$  – постоянные матрицы;  $0=h_0 < h_1 < \dots < h_m=h$  – постоянные запаздывания. Применение второго метода Ляпунова к исследованию этой проблемы предполагает, как известно, построение квадратичного функционала Ляпунова – Красовского, структура которого впервые предложена Н.Н. Красовским и Ю.М. Репиным в работах [1, 2]. Однако вопрос существования и единственности таких функционалов оставался открытым. В статье [3] было введено понятие матрицы Ляпунова и доказано, что, если система (1) экспоненциально устойчива, то существует единственный квадратичный функционал, который имеет вид

$$v(x_t) = x^*(t)U(0)x(t) + 2x^*(t) \sum_{j=1}^m \int_{-h_j}^0 U(-\theta - h_j) A_j x(t+\theta) d\theta + \\ + \sum_{k=1}^m \sum_{j=1}^m \int_{-h_k}^0 x^*(t+\theta_1) A_k^* \left( \int_{-h_j}^0 U(\theta_1 + h_k - \theta_2 - h_j) A_j x(t + \theta_2) d\theta_2 \right) d\theta_1, \quad (2)$$

и производная которого в силу системы (1) совпадает с заданной квадратичной формой

$$w(x(t)) = -x^*(t)Wx(t).$$

Здесь через  $x_t$  обозначен сегмент  $\{x(\xi) | \xi \in [t-h, t]\}$ , а  $U(\tau)$  – матрица Ляпунова, ассоциированная с симметрической матрицей  $W$  определение матрицы и способ ее вычисления приведены в статье [4]. Отметим, что необходимым и достаточным условием существования матрицы Ляпунова, а значит, и функционала (2), является так называемое *условие Ляпунова* — условие отсутствия у системы (1) двух собственных чисел, сумма которых равна нулю [5]. Это означает, в частности, существование функционала структуры (2) в случае экспоненциальной устойчивости системы (1).

Достаточные условия экспоненциальной устойчивости линейных стационарных систем, предполагающие анализ положительной определенности функционалов Ляпунова – Красовского, предложены в работе [6], однако именно анализ положительной определенности функционала (2) является одной из главных проблем при таком подходе. Более того, в работе [5] показано, что функционал (2) в случае экспоненциальной устойчивости системы (1) допускает лишь кубическую оценку снизу на множестве непрерывных функций. В статье [4] вводятся так называемые функционалы полного типа, производная которых в силу системы зависит, в том числе и от состояния системы в момент с запаздыванием, и такие функционалы допускают квадратичную оценку снизу, однако важной остается проблема конструктивного исследования их положительной определенности.

Предложенный нами в работе [7] конструктивный подход к исследованию экспоненциальной устойчивости системы (1) восходит к

исследованиям Б.С. Разумихина [8]. Идея этого подхода заключается в том, чтобы исследовать положительную определенность функционала (2) не на произвольных непрерывных функциях, а на более узком классе функций, задаваемом множеством

$$S = \{x_t : \|x(t + \sigma)\| \leq \|x(t)\|, \quad \sigma \in [-h, 0]\},$$

что позволяет предложить конструктивный конечный алгоритм исследования экспоненциальной устойчивости исходной системы. Как показано в настоящей работе, этот подход может быть адаптирован и к исследованию неустойчивости, а также применен для нахождения критических значений запаздывания систем рассматриваемого класса.

### **Теоремы об устойчивости и описание метода**

Предлагаемый метод исследования экспоненциальной устойчивости системы (1) основан на следующих необходимом и достаточном условиях экспоненциальной устойчивости, доказанных в работе [7].

**Теорема 1.** Пусть система (1) экспоненциально устойчива по Ляпунову. Тогда для любой положительно-определенной матрицы существует функционал такой, что

$$1. \left. \frac{dv(x_t)}{dt} \right|_{(1)} = -x^*(t)Wx(t);$$

$$2. \text{Существует } c_1 > 0 \text{ такое, что } v(x_t) \geq c_1 \|x(t)\|^2 \text{ при } x_t \in S.$$

**Теорема 2.** Пусть существует функционал такой, что

$$1. \left. \frac{dv(x_t)}{dt} \right|_{(1)} = -x^*(t)Wx(t), \text{ где матрица } W \text{ положительно определена};$$

$$2. v(x_t) \geq c_1 \|x(t)\|^2 \text{ при } x_t \in S \text{ при некотором } c_1 > 0.$$

Тогда система (1) экспоненциально устойчива по Ляпунову.

Опуская подробности, приведем здесь идею предлагаемого метода применительно к системе с одним запаздыванием вида

$$\dot{x}(t) = Ax(t) + Bx(t-h), \quad t \geq t_0, \quad (3)$$

где  $A, B \in R^{n \times n}$  – постоянные матрицы,  $h \geq 0$ . Построить конструктивный метод исследования устойчивости системы (3) позволяет замена множества  $S$  в теоремах 1, 2 множеством

$$S_0 = \{x_t : \|x^{(k)}(t + \sigma)\| \leq (\|A\| + \|B\|)^k \|x(t)\|, \\ \sigma \in [-h, 0], \quad k = 0, 1, \dots\},$$

при которой сформулированные теоремы остаются справедливыми.

Разобьем отрезок  $[t-h, t]$  на  $N$  частей одинаковой длины  $\Delta = h/N$  точками  $t_j = t - j\Delta, j = \overline{0, N}$  и приблизим вектор-функцию  $x_t \in S_0$  на каждом из отрезков  $[t_{j+1}, t_j]$  вектором  $q_j(t, \theta)$  состоящим из полиномов третьей степени переменной  $\theta \in [-\Delta, 0]$ , коэффициенты которых определяются из условий

$$\begin{aligned} q_j(t, 0) &= x(t_j), & q_j(t, -\Delta) &= x(t_{j+1}), & q'_j(t, 0) &= \dot{x}(t_j), \\ q'_j(t, -\Delta) &= \dot{x}(t_{j+1}). \end{aligned} \quad (4)$$

Условия (4) гарантируют гладкость сплайна, состоящего из функций  $q_j(t, \theta)$  а их количество определяет кубический порядок функций, используемых для приближения  $x_i \in S_0$ . Подставляя такое приближение как функцию величин, стоящих в правых частях соотношений (4), в функционал (2) вместо  $x(t+\theta)$  и оценивая его погрешность с помощью формулы Тейлора и неравенств, определяющих множество  $S_0$ , получим оценку снизу для функционала (2) в виде

$$v(x_t) \geq \bar{v}_3(\bar{x}, h, N, p) = \bar{v}_3^0(\bar{x}, h, N, p) - \bar{\delta}(h, N) \|p\|^2, \quad \bar{\delta}(h, N) > 0. \quad (5)$$

Здесь вектор  $\bar{x}$  размерности  $n(2N+1)$  представляет собой конкатенацию векторов  $\bar{x}^{(i)}$  при увеличении  $i$  от 1 до  $N$ , а каждый из векторов  $\bar{x}^{(i)}$  размерности  $2N+1$  может быть записан в виде

$$\bar{x}^{(i)} = (\bar{x}_1^{(i)}, \dots, \bar{x}_{2N+1}^{(i)})^*, \\ \bar{x}_j^{(i)} = x_i(t - j\Delta), j = \overline{1, N}, \bar{x}_{N+j+1}^{(i)} = \dot{x}_i(t - j\Delta), j = \overline{0, N}, i = \overline{1, n}.$$

В оценке (5) также принято обозначение  $p = x(t)$ , а  $\bar{v}_3^0(\bar{x}, h, N, p)$  представляет собой квадратичную форму относительно компонент векторов  $\bar{x}$  и  $p$ . Формулы, определяющие квадратичную форму  $\bar{v}_3(\bar{x}, h, N, p)$  в скалярном случае, приведены в работе [7]. Введем также обозначение

$$\bar{x}_j = (\bar{x}_j^{(1)}, \dots, \bar{x}_j^{(n)})^*, \quad j = \overline{1, 2N+1}.$$

Исследование положительной определенности функционала (2), согласно теореме 2, может быть сведено к поиску такого значения  $N$ , при котором положителен минимум квадратичной формы  $\bar{v}_3(\bar{x}, h, N, p)$ , найденный на множестве

$$\bar{S}_N = \left\{ \bar{x} \left| \begin{array}{l} \dim(\bar{x}) = n(2N+1), \|\bar{x}_j\| \leq \|p\|, j = \overline{1, N}, \\ \|\bar{x}_{N+j+1}\| \leq (\|A\| + \|B\|)\|p\|, j = \overline{0, N} \end{array} \right. \right\}.$$

Отметим, что компоненты вектора  $x(t)$  были выделены в оценке (5) в отдельный вектор  $p$  для того, чтобы добиться линейности ограничений множества  $\bar{S}_N$ . Не умаляя общности, в указанной задаче минимизации можно положить  $\|p\|=1$  поэтому, окончательно, достаточным условием экспоненциальной устойчивости системы (3) является существование такого значения при котором функция

$$z(h, N) \stackrel{\text{def}}{=} \min_{\substack{\bar{x} \in \bar{S}_N \\ \|p\|=1}} \bar{v}_3(\bar{x}, h, N, p) = \min_{\substack{\bar{x} \in \bar{S}_N \\ \|p\|=1}} \bar{v}_3^0(\bar{x}, h, N, p) - \bar{\delta}(h, N)$$

положительна при фиксированных параметрах  $A, B$  и  $h$  системы.

В работе [9] показана сходимость описанного метода в следующем смысле. Фиксируем матрицы  $A$  и  $B$  системы (3) и предположим, что  $(\bar{h}_1, \bar{h}_2)$  — некоторый интервал ее экспоненциальной устойчивости, причем  $\bar{h}_1, \bar{h}_2$  — критические значения запаздывания. Пусть  $0 < \bar{h}_1 < \bar{h}_2 < +\infty$ . Выберем некоторое значение  $\tilde{h} \in (\bar{h}_1, \bar{h}_2)$  и фиксируем  $\tilde{N}$  такое, что  $z(\tilde{h}, N) > 0$  для любого  $N \geq \tilde{N}$ . При  $N \geq \tilde{N}$  определим числа



$$h_N^{(1)} = \sup_{\substack{h < \bar{h} \\ z(h, N) \leq 0}} h, \quad h_N^{(2)} = \inf_{\substack{h > \bar{h} \\ z(h, N) \leq 0}} h.$$

В силу непрерывности функции  $z(h, N)$  в случае экспоненциальной устойчивости системы (3) при заданном  $N$  мы получаем интервал  $(h_N^{(1)}, h_N^{(2)})$  на котором описанный метод гарантирует экспоненциальную устойчивость системы (3):  $z(h, N)$  при  $h \in (h_N^{(1)}, h_N^{(2)})$ . Ясно, что  $(h_N^{(1)}, h_N^{(2)}) \subset (\bar{h}_1, \bar{h}_2)$ . Показано [8], что последовательности  $\{h_N^{(1)}\}_{N=\bar{N}}^{+\infty}$  и  $\{h_N^{(2)}\}_{N=\bar{N}}^{+\infty}$  сходятся к критическим значениям запаздывания:

$$\lim_{N \rightarrow +\infty} h_N^{(1)} = \bar{h}_1, \quad \lim_{N \rightarrow +\infty} h_N^{(2)} = \bar{h}_2.$$

Доказательство сходимости метода опирается на тот факт, что

$$\bar{v}_3(\bar{x}, h, N, p) \xrightarrow{N \rightarrow +\infty} v(x_t)$$

на функциях  $x_t \in S_0 \cap \{x_t | x(t) = p, \|p\| = 1\}$  откуда следует, что функция  $\bar{\delta}(h, N)$ , содержащая члены, отвечающие за оценку погрешности рассмотренного приближения, стремится к нулю при  $N \rightarrow +\infty$ . Этот факт объясняет, в частности, существование значения  $\bar{N}$ . Сходимость метода играет важную роль в его применении, поскольку она означает, что с увеличением  $N$  гарантируемый методом интервал экспоненциальной устойчивости системы (3) «изнутри» стремится к точному интервалу ее экспоненциальной устойчивости.

Отметим, что аналогичным образом может быть обоснован конструктивный метод исследования неустойчивости системы (1), опирающийся на теоремы о неустойчивости, сформулированные в работе [7]. Как следует из этих теорем, достаточным условием неустойчивости системы (1) является существование такого значения  $N$ , что

$$\bar{z}(h, N) \stackrel{\text{def}}{=} \min_{\substack{\bar{x} \in \bar{S}_N \\ \|p\|=1}} \bar{v}_3^0(\bar{x}, h, N, p) + \bar{\delta}(h, N) < 0.$$

Аналогами чисел  $h_N^{(1)}$  и  $h_N^{(2)}$  при анализе неустойчивости системы (3) будут, таким образом, числа

$$\tilde{h}_N^{(1)} = \sup_{\substack{h < \tilde{h} \\ \bar{z}(h, N) \geq 0}} h, \quad \tilde{h}_N^{(2)} = \inf_{\substack{h > \tilde{h} \\ \bar{z}(h, N) \geq 0}} h,$$

где  $\tilde{h}$  – произвольная точка промежутка неустойчивости. Значения  $\tilde{h}_N^{(1)}$  и  $\tilde{h}_N^{(2)}$  определены при  $N \geq \tilde{N}$ , причем число  $\tilde{N}$  таково, что  $\bar{z}(\tilde{h}, N) < 0$  для любого  $N \geq \tilde{N}$ .

### **Оценка области асимптотической устойчивости**

Снова рассмотрим систему

$$\dot{y}(t) = f(y(t), y(t-h_1), \dots, y(t-h_m)), \quad (6)$$

линейное приближение которой в окрестности нулевого решения описывается экспоненциально устойчивой системой (1). Производную функционала  $v(y_t)$ , определенного равенством (2), в силу системы (6) представим в виде

$$\begin{aligned}\hat{w}(y_t) = & -y^*(t)W y(t) \\ & + 2 \left[ \sum_{j=1}^m \int_{-h_j}^0 U(-\theta - h_j) A_j y(t + \theta) d\theta + U(0) y(t) \right]^* \\ & \cdot [f(y(t), y(t - h_1), \dots, y(t - h_m)) - \sum_{j=0}^m A_j y(t - h_j)].\end{aligned}$$

Введем функцию  $d(c) = \max_{S_c} \hat{w}(y_t)$ , где

$$S_c = \{y_t : \|y(t + \sigma)\| \leq \|y(t)\| = c, \quad \sigma \in [-h, 0]\}.$$

Функция  $d(c)$  определена и непрерывна при  $c \geq 0$ ,  $d(0) = 0$  и отрицательна в некоторой окрестности нуля. Поэтому существует положительное число  $\hat{c} \leq \infty$  такое, что при  $0 < c < \hat{c}$  справедливо неравенство  $d(c) < 0$ .

**Определение.** Областью асимптотической устойчивости нулевого решения системы (6) будем называть множество  $A \subset \{\varphi \in C([-h, 0])\}$  таких векторных функций, для которых выполнено предельное соотношение  $y(t, \varphi) \rightarrow 0$  при  $t \rightarrow +\infty$ .

**Теорема 3.** Множество  $S_a \cap C([-h, 0])$  содержится в области асимптотической устойчивости  $A$ , если для некоторого положительного числа  $c < \hat{c}$  выполняется неравенство

$$\max_{\varphi \in S_a} v(\varphi) \leq \min_{\varphi \in S_c} v(\varphi).$$

**Замечание.** Теорема 3 остается справедливой, если множество заменить множеством

$$D_c = \{y_t : \|y(t + \sigma)\| \leq \|y(t)\| = c, \|\dot{y}(t + \sigma)\| \leq k, \quad \sigma \in [-h, 0]\},$$

в котором

$$k = \max_{S_c} \|f(y(t), y(t - h_1), \dots, y(t - h_m), \|\cdot\|.$$

### 5. Поиск критических значений запаздывания.

Рассмотрим одно важное применение изложенного метода, которое во многом опирается на его сходимости. Фиксируем матрицы и системы (3), и предположим, что  $(\bar{h}_1, \bar{h}_2)$  – интервал ее экспоненциальной устойчивости, а  $(\bar{h}_2, \bar{h}_3)$  – интервал ее неустойчивости (или наоборот), критические значения неизвестны. Описанный выше метод исследования экспоненциальной устойчивости и неустойчивости системы (3) позволяет предложить способ численного поиска критического значения  $\bar{h}_2$ .

Пусть известны значения  $\tilde{h} \in (\bar{h}_1, \bar{h}_2)$ ,  $\tilde{\tilde{h}} \in (\bar{h}_2, \bar{h}_3)$  и найдены соответствующие им значения  $\tilde{N}$  и  $\tilde{\tilde{N}}$  такие, что

$$z(\tilde{h}, \tilde{N}) > 0, \quad \tilde{z}(\tilde{\tilde{h}}, \tilde{\tilde{N}}) < 0. \quad (7)$$

Величины  $\tilde{h}, \tilde{\tilde{h}}$  могут быть найдены перебором значений  $N$ , удовлетворяющих соответственно одному из условий (7), или каким-то другим способом. Зададим некоторую точность  $\varepsilon > 0$  и, последовательно увеличивая  $N$ , будем строить члены последовательностей  $\{h_N^{(2)}\}_{N \geq \tilde{N}}$  и  $\{\tilde{h}_N^{(1)}\}_{N \geq \tilde{\tilde{N}}}$  с точностью  $\varepsilon$  (до сотых, тысячных и т.д.) так, чтобы выполнялись условия



$$z(\tilde{h}, \tilde{N}) > 0, \quad \tilde{z}(\tilde{h}, \tilde{N}) < 0. \quad (7)$$

На  $n$ -м шаге будем проверять условие

$$\min_{j=\overline{0,k}} \tilde{h}_{\tilde{N}+j}^{(1)} - \max_{l=\overline{0,k}} h_{\tilde{N}+l}^{(2)} = \varepsilon. \quad (8)$$

Если при некотором  $k$  это условие выполняется, то критическое значение  $\bar{h}_2$  найдено с точностью  $\varepsilon$  между максимумом и минимумом в нем. Поскольку метод сходится, такое  $k$  обязательно существует, если только рассматриваемые интервалы экспоненциальной устойчивости и неустойчивости действительно являются соседними. Значит, если при достаточно большом количестве шагов найти значение  $k$  при котором выполняется (8), не удалось, можно попробовать применить описанную выше процедуру для других начальных значений  $\tilde{h}$  и  $\tilde{N}$ . Численные эксперименты подтверждают работоспособность такого метода поиска критических значений запаздывания.

### Литература

1. Красовский Н.Н. Об аналитическом конструировании оптимального регулятора в системе с запаздываниями времени // Прикл. математика и механика, 1962. Т. 26, вып. 1. С. 39–51.
2. Репин Ю.М. Квадратичные функционалы Ляпунова для систем с запаздыванием // Прикл. математика и механика, 1965. Т. 29. С. 564–566.
3. Kharitonov V.L., Zhabko A.P. Lyapunov-Krasovskii approach to the robust stability analysis of time-delay system. // IEEE Trans., Automatica, vol 39, 2003, 15–20.
4. Харитонов В.Л. Функционалы Ляпунова с заданной производной // Вестн. С.-Петербург. ун-та. Сер. 10. 2005. Вып. 1-2. С. 110–117, 199–207.
5. Huang W. Generalization of Liapunov's theorem in a linear delay systems // J. of Mathematical Analysis and Applications. 1989. Vol. 142. P. 83–94.
6. Красовский Н.Н. О применении второго метода Ляпунова для уравнений с запаздываниями времени // Прикл. математика и механика, 1956. Т. 20. С. 315–327.
7. Жабко А.П., Медведева И.В. Алгебраический подход к анализу устойчивости дифференциально-разностных систем // Вестн. С.-Петербург. ун-та. Сер. 10. 2011. Вып. 1. С. 9–20.
8. Разумихин Б.С. Об устойчивости систем с запаздыванием // Прикл. математика и механика, 1956. Т. 20, вып. 4. С. 500–512.
9. Медведева И.В. О сходимости одного метода анализа устойчивости систем с запаздыванием // Процессы управления и устойчивость: Труды 43-й международной научной конференции аспирантов и студентов / Под ред. А. С. Ерёмина, Н. В. Смирнова. СПб.: Издат. Дом С.-Петербург. ун-та, 2012. С. 26–31.

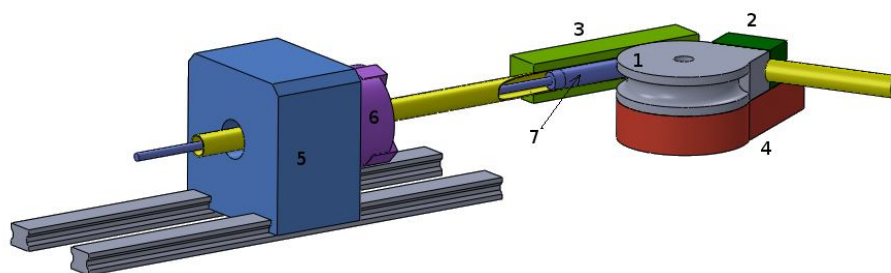
## ***Реализация цифровой системы управления трубогибочным станком***

В настоящее время большая часть сложного станочного промышленного оборудования снабжается цифровыми системами управления. Использование таких систем позволяет обеспечить не только высокое качество продукции, но и повысить эффективность и безопасность производства. Среди существующих типов трубогибочных станков особое положение занимают станки, работающие по принципу намотки. С их помощью могут быть получены трубы сложной конфигурации, с гибом в нескольких плоскостях. Совместно со специальными контрольно-измерительными комплексами такие станки позволяют достичь наилучшего на текущем уровне развития технологий качества гибки.

Принципиальная схема такого станка приведена на рис. 1. В качестве основных подвижных частей станка выделяются:

- ролик (вал), обозначенный на рис. 1. цифрой 1, является шаблоном, определяющим радиус гiba трубы.
- зажим (2), обеспечивающий фиксацию подвижного участка трубы при гibe в канавке вала
- прижим (3), необходимый для предотвращения смещения трубы в горизонтальной плоскости при гibe
- гибщик (4), обеспечивающий силовое воздействие на трубу при гibe
- каретка (5), реализующая продольное перемещение трубы
- захват(6), фиксирующий неподвижный конец трубы при гibe в заданном положении относительно продольной оси трубы и осуществляющий ее поворот
- дорн (7), используемый для снижения искажений поперечного профиля трубы при гibe.

В зависимости от требований к производительности процесса гибки осуществляется автоматизация как отдельных узлов, так и станка целиком. Несмотря на существующие конструктивные отличия металлообрабатывающих станков, часто они реализуются таким образом, чтобы обеспечить возможность их использования совместно с готовым универсальным решением в виде стойки ЧПУ. Однако в силу специфики принципа работы и широкого диапазона конструкторских решений для основных узлов трубогибочных станков такой подход для них обычно не применяется.



**Рис. 1.** Конструктивная схема станка

При разработке системы ЧПУ можно выделить следующие основные задачи:

- управление исполнительными механизмами;
- построение человеко-машинного интерфейса;
- интеграция создаваемого оборудования в имеющиеся производственные информационные системы.

Для обеспечения точности перемещений узлов станка используется управление по принципу обратной связи. Контролируемой величиной при этом является положение подвижной части станка, измеряемое с помощью датчика относительно рамы. В качестве управляющего сигнала в большинстве случаев используется уровень напряжения. Для электроприводов, подключенных через преобразователи частоты, этот сигнал позволяет задавать скорость их вращения. Для гидравлических приводов, которые часто находят применение на трубогибочных станках, с помощью уровня напряжения может контролироваться пропускная способность клапанов.

При моделировании процессов управления в мехатронных системах широко применяется абстракция «ось» как совокупность привода, датчика положения и закона управления, обеспечивающих позиционирование отдельного компонента. В качестве стандартного функционала для «оси» можно выделить относительное и абсолютное позиционирование, движение с заданной скоростью, калибровка (поиск нулевого положения), детектирование ошибочных ситуаций. В зависимости от рассматриваемых задач этот набор может быть расширен.

Базовый закон управления «осью» при перемещениях имеет следующую структуру. В соответствии с классическим подходом теории управления рассматривается модель, формирующая некоторое программное движение объекта, относительно которого стабилизируется поведение системы в отклонениях. Управляющий сигнал для «оси» представляет собой сумму программной и стабилизирующей частей. В простейшем случае программное движение и управление формируются как решение задачи оптимального быстрогодействия для перевода из точки в

точку линейной системы второго порядка вида  $\ddot{x}=u$  с учетом ограничений на величину управляющего сигнала  $|u| \leq u_{\max}$  и максимально допустимое значение скорости перемещения  $|\dot{x}| \leq \dot{x}_{\max}$ .

Для случаев, когда управляющий сигнал задает скорость изменения регулируемой величины, стабилизация может осуществляться за счет простейшего пропорционального регулятора. В силу интегрирующих свойств такого привода достигается важное свойство астатизма по регулируемой координате.

В отличие от механических способов передачи усилий, для гидравлических приводов в большей степени характерно нелинейное поведение под нагрузкой, что требует применения нетривиальных подходов к построению регуляторов. Одним из возможных способов борьбы с этой особенностью может быть введение в контур управления специальных компенсирующих нелинейных звеньев.

Большая часть перемещений узлов станка при гибке осуществляется последовательно, однако для некоторых специальных режимов может потребоваться их совместная работа. Необходимость координированного управления несколькими «осями» обеспечивается за счет задания программного движения извне зависимой «оси». В простейшем случае реализуются линейные соотношения между заданными положениями ведущей и ведомой «осей».

В соответствии с техническим заданием спецификой данного проекта является возможность задания профиля скорости движения подвижной секции гибщика в виде кусочно-постоянной зависимости от угла поворота. Это позволяет получить больший контроль над процессом гибки. Данный результат также может быть достигнут за счет внешнего способа задания программного движения для «оси» гибщика.

Реализация законов управления таким сложным техническим устройством, как трубогибочный станок, предполагает использование системы реального времени. Необходимость в ней обусловлена требованиями не только к точности позиционирования, но также к надежности и безопасности. Наряду с другими решениями в сфере автоматизации для этих целей широко применяются программируемые логические контроллеры (ПЛК). В качестве их отличительных особенностей можно назвать:

- гарантированный временной интервал выполнения задач, в том числе с возможностью его настройки;
- надежность – предполагается, что контроллер при работе не требует обслуживания;
- широкие возможности программирования – при поддержке средой разработки программного обеспечения для контроллера стандарта МЭК 61131-3 доступно в качестве альтернативы пять языков программирования;

- удобство работы со средствами ввода-вывода – производители предлагают множество блоков расширения, от простейших терминалов цифровых входов и выходов до сложных специализированных модулей, позволяющих подключать несколько «осей», при этом регистры ввода-вывода непосредственно отображаются в память задач и обеспечивается адресация вплоть до отдельных бит.

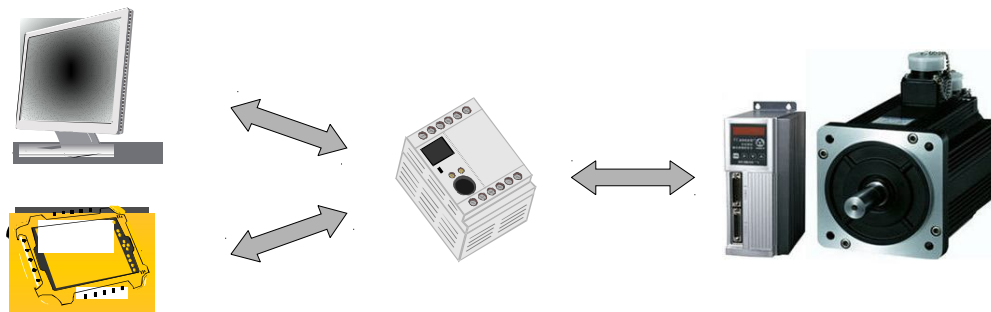
На сегодняшний день существует множество различных вариантов ПЛК, различающихся не только аппаратной базой, но и функциональными возможностями. В частности, в условиях постоянного взаимодействия оператора и машины привлекательным выглядит совмещение функций ПЛК и РС-совместимого компьютера. Это позволяет получить мощные вычислительные возможности для ПЛК, выполняющего одновременное управление несколькими «осями» и использовать стандартные средства разработки для создания графического интерфейса оператора.

Рабочее место оператора станка с ЧПУ предполагает следующую функциональность:

- предоставление информации о текущем состоянии станка;
- ручное управление основными узлами станка;
- программирование заданий обработки деталей и их выполнение в полуавтоматическом и автоматическом режимах;
- отображение информации об ошибках.

Отсутствие возможности непосредственного подключения средств отображения информации напрямую к ПЛК приводит к необходимости организации информационного обмена между ПЛК и рабочим местом оператора (рис.2). Важным при этом является поддержка этими двумя компонентами стандартизованных интерфейсов и открытых протоколов обмена.

Для трубогибочного станка в качестве исходных данных при создании программы гибки выступает требуемая конфигурация трубы, а также технологическая информация об особенностях применяемого материала заготовки.



**Рис. 2. Применение ПЛК**

Существуют несколько возможных способов задания конфигурации

трубы. Наиболее часто используется ее задание в виде ломаной, для которой задается массив точек перегиба. Также встречается более близкий к используемой в станках системе координат вариант - когда задаются перемещение вдоль продольной оси неподвижного участка трубы, угол поворота вокруг этой оси и уголгиба. Первый вариант, как более универсальный, задействуется и при обмене данными с контрольно-измерительными комплексами, позволяющими получить трехмерную модель готовой трубы.

Каждой конфигурации трубы соответствует определенная последовательность действий исполнительных органов станка. Необходимость выполнения этих действий как в ручном, так и в автоматическом режиме приводит к выделению совокупности команд управления отдельными узлами станка. Для «осей» эти команды должны обеспечивать доступ к их стандартной функциональности. Каждый вспомогательный узел станка также характеризуется совокупностью допустимых команд. Как программа гибки, так и действия оператора, выражаемые в выборе тех или иных элементов графического интерфейса пользователя, будут приводить к выполнению определенных последовательностей элементарных команд. Таким образом, одним из основных компонентов программного обеспечения оператора является интерпретатор этих команд. Дополнительную гибкость при настройке ПО оператора обеспечивает возможность описания интерфейсных элементов и их функциональности в виде редактируемых текстовых файлов.

Конфигурация основных параметров «осей» осуществляется при наладке либо ремонте станка и не предполагает участия оператора. Задание параметров режимов гибки, напротив, требует непосредственного участия и знания оператором средств программирования, предоставляемых ПО станка. Кроме этого, возможность использования готовых программ гибки, хранимых в базе данных компьютера оператора, позволяет использовать станок относительно малоквалифицированному персоналу.

Рассмотрение примера практического использования ПЛК в решении конкретных задач управления на взгляд автора рекомендуется в рамках курсов «Цифровые системы реального времени» и «Автоматизированные системы сбора и обработки информации».

### **Литература**

1. Информационная система Beckhoff. Режим доступа: [http://infosys.beckhoff.com/index\\_en.htm](http://infosys.beckhoff.com/index_en.htm) (дата обращения 10.09.2012)
2. В.Е. Зюбин Программирование ПЛК: языки МЭК 61131-3 и возможные альтернативы. «Промышленные АСУ и контроллеры». - М.: Научтехлитиздат, №11.-2005.- С.31-35.

**Кристаллинский Р.Е.,**

Смоленский государственный университет, г. Смоленск  
[krist1940@rambler.ru](mailto:krist1940@rambler.ru)

**Белоусов В.В.**

Институт проблем информатики РАН, г. Москва  
[vbelousov@ipiran.ru](mailto:vbelousov@ipiran.ru)

## **Опыт решения задач об отыскании оптимального управления для линейных систем с постоянными коэффициентами и приводимым к ним**

### **Аннотация**

*Рассмотрена задача отыскания оптимального управления средствами системы Mathematica. Решение подобного класса задач часто необходимо в учебных курсах «Дифференциальные уравнения», «Математическое моделирование». Подобные курсы составляют основу учебных планов как математических, так и технических специальностей в различных учебных заведениях. Как правило, примеры, необходимые для иллюстрации и закрепления знаний, требуемых при изучении рассматриваемых методов, сложны. Поэтому авторы курсов используют очень простые примеры. К сожалению, такие примеры не позволяют глубоко проникнуть в суть проблемы. Использование средств компьютерной математики позволяет насытить курс достаточным количеством нетривиальных примеров, что положительно сказывается на результатах обучения.*

Пусть система, описывающая поведение объекта управления, имеет вид

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (1)$$

где  $A, B$  – матрицы размера  $(n \times n), (n \times q)$  соответственно, на управление ограничений не наложено,  $u \in U = R^q; t \in T = [t_0, t_1]$  – интервал времени функционирования системы, моменты начала процесса  $t_0$  и окончания процесса  $t_1$  заданы, правый конец траектории  $x(t_1)$  – свободен.

Начальное условие  $x(t_0) = x_0 \in R^n$  задано и определяет начальное состояние.

Пусть функционал управления имеет следующий вид

$$F = \frac{1}{2} \int_{t_0}^{t_1} [x^T(t) S x(t) + u(t)^T Q u(t)] dt + \frac{1}{2} [x^T(t_1) H x(t_1)]. \quad (2)$$

Здесь  $S$  и  $H$  – неотрицательно определённые симметрические матрицы размера  $(n \times n)$ , а  $Q$  – положительно определённая

симметрическая матрица размера  $(n \times q)$ .

Требуется найти пару  $d^* = (x^*(.), u^*(.))$ , на которой достигается минимум функционала (2).

В работе [1] предлагается следующий алгоритм решения рассматриваемой задачи.

Рассматривается система дифференциальных уравнений следующего вида

$$\begin{aligned}\dot{x}(t) &= Ax(t) + BQ^{-1}B^T y(t), \\ \dot{y}(t) &= -Ay(t) + Sx(t).\end{aligned}\tag{3}$$

Находится решение приведенной системы дифференциальных уравнений, удовлетворяющее краевым условиям

$$x(t_0) = x_0, y(t_1) = -Hx(t_1).\tag{4}$$

Тогда оптимальное управление находится следующим образом

$$u^*(t) = Q^{-1}B^T y(t).\tag{5}$$

В имеющейся литературе решение рассматриваемой задачи приводятся только для очень простых случаев. Например, в [1] рассматривается только случай, когда  $x(t)$  и  $u(t)$  – векторы.

Более общее решение рассматриваемой задачи можно получить, используя систему компьютерной математики, способную производить достаточно сложные символьные вычисления. На наш взгляд для этой цели более всего подходит система Mathematica.

Приведём соответствующую программу вычислений.

Вводим размерность матрицы  $A$ .

$n = 4$ ;

Вводим матрицу  $A$ .

```
A = ( {  
  {1.3, 0.27, 0.35, 0.11},  
  {0.38, 1.7, 0.24, 0.23},  
  {0.8, 0.26, 1.4, 0.32},  
  {0.41, 0.29, 0.15, 2.1}  
});
```

Вводим размерность матрицы  $B$ .

$n = 4$ ;  $m = 3$ ;

Вводим матрицу  $B$ .

```
B = ( {  
  {2.4, 3.8, 1.5},  
  {1.9, 2.6, 5.3},  
  {4.8, 3.7, 6.1},  
  {1.9, 2.5, 3.6}  
});
```



Вводим размерность матрицы  $S$ .

$n = 4$ ;

Вводим матрицу  $S$ .

```
S = ( {  
    {1.2, 0.18, 0.23, 0.6},  
    {0.18, 2.1, 0.27, 0.31},  
    {0.23, 0.27, 1.5, 0.24},  
    {0.6, 0.31, 0.24, 1.1}  
});
```

Находим собственные числа матрицы  $S$ .

Eigenvalues[S]

Все собственные числа матрицы  $S$  положительны. Матрица  $S$  является положительно определённой.

Вводим размерность матрицы  $Q$ .

$n = 3$ ;

Вводим матрицу  $Q$ .

```
Q = ( {  
    {2.8, 1.2, 0.89},  
    {1.2, 3.4, 2.1},  
    {0.89, 2.1, 4.7}  
});
```

Находим собственные числа матрицы  $Q$ .

Eigenvalues[Q]

{6.76768, 2.53236, 1.59996}

Матрица  $Q$  является положительно определённой.

Вводим размерность матрицы  $H$ .

$n = 4$ ;

Вводим матрицу  $H$ .

```
H = ( {  
    {4.3, 2.1, 1.4, 1.7},  
    {2.1, 3.6, 1.8, 0.95},  
    {1.4, 1.8, 2.8, 1.5},  
    {1.7, 0.95, 1.5, 2.9}  
});
```

Находим собственные числа матрицы  $H$ .

Eigenvalues[H]

Матрица  $H$  является положительно определённой.

Находим коэффициенты системы уравнений (3).

$X = \text{Table}[x_i, \{i, 1, 4\}]$ ;

$Y = \text{Table}[y_i, \{i, 1, 4\}]$ ;

$Z = \text{Join}[X, Y]$ ;

$U1 = A.X + B.\text{Inverse}[Q].\text{Transpose}[B].Y$ ;

$U2 = A.Y + S.X$ ;

```

U = Join[U1,U2];
Находим матрицу коэффициентов системы дифференциальных
уравнений (3)
W = Table[Coefficient[U[[i]],Z[[j]]],{i,1,8},{j,1,8}];
Находим общее решение системы дифференциальных уравнений (3).
C1 = Table[c,{i,1,8}];
Z1[t_] = MatrixExp[W t].C1;
Z1[t_] = ComplexExpand[Z1[t]]/.I*0;
X1[t_] = Table[Z1[t][[i]],{i,1,4}];
Y1[t_] = Table[Z1[t][[i]],{i,5,8}];
Находим оптимальное управление.
XX0 = {0.1,0.3,0.2,0.7};
t0 = 0.1;
XX1 = {0.5,0.8,0.7,0.9};YY1 = -H.XX1;
t1 = 0.8;
T1 = Table[X1[t0][[i]] == XX0[[i]],{i,1,4}];
T2 = Table[Y1[t1][[i]] == YY1[[i]],{i,1,4}];
T = Join[T1,T2];
R = Solve[T,C1]
Y1[t_] = Y1[t]/.R[[1]];
u[t_] = Inverse[Q].Transpose[B].Y1[t];
Находим траекторию системы, соответствующую этому управлению.
u1[t_] = Expand[B.u[t]];
Clear[X]
A1 = MatrixExp[A t].Inverse[MatrixExp[A t0]];
A2 = Chop[MatrixExp[At0 ∫0t MatrixExp[- A τ] u1[τ] dτ];
A3 = Chop[MatrixExp[At0 ∫0t0 MatrixExp[- A τ] u1[τ] dτ].;
X[t_] = Expand[A1.(XX0+A2-A3)];
X[t_] = Chop[X[t]];

```

Предложенная программа позволяет решать задачу поиска оптимального управления в более сложных, чем рассматривалось ранее случаях. Предложенный подход позволяет повысить эффективность обучения решению сложных задач математического моделирования.

В работе [2] рассмотрены примеры решения задач анализа и фильтрации процессов в линейных эредитарных стохастических системах, приводимых к стохастическим дифференциальным уравнениям.

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 10-07-00021) и программы ОНИТ РАН "Интеллектуальные информационные технологии, системный*

*анализ и автоматизация" (проект 1.7).*

### **Литература**

1. Иванько В.И., Ермошина О.В., Кувыркин Г.Н. Вариационное исчисление и оптимальное управление. М.: Издательство МГТУ им. Баумана, 2001.
2. Сеницын И.Н., Сеницын В.И., Корепанов Э.Р., Белоусов В.В., Сергеев И.В., Басилашвили Д.А. Опыт моделирования эредитарных стохастических систем // Сборник докладов XIII Международной научно-технической конференции «Кибернетика и высокие технологии XXI ве-ка» (С&Т 2012). Воронеж: НПФ «Саквоее» ООО, 2012. Т.2. С. 346-357.

## **Алгоритм фонемной сегментации в модуле обучения системы распознавания речи**

### **Аннотация**

*Рассматривается применение алгоритма фонемной сегментации речевого сигнала на основе вейвлет-преобразования с целью автоматизации процесса обучения системы распознавания речи. Проводится анализ результатов работы реализованного модуля в составе системы преобразования речи в текст.*

Одним из важных этапов в работе системы распознавания речи является ее предварительное обучение. Данный процесс предполагает формирование базы данных признаков для определенных сегментов речевого сигнала. В качестве единиц распознавания могут быть использованы фонологические единицы: аллофоны, фонемы, дифоны, слоги, слова или некоторые их сочетания. Задача ручного выделения диктором таких сегментов из потока слитной речи требует значительных затрат времени и подвержена ошибкам. Кроме того, практически невозможно воспроизвести результаты ручной сегментации вследствие изменчивости человеческого зрительного и слухового восприятия. Автоматическая сегментация не безошибочна, однако она непротиворечива по своей сути, и её результаты воспроизводимы. В случае возникновения ошибок результат автоматической сегментации может корректироваться диктором. Точность сегментации в значительной степени определяет надежность автоматического распознавания речи

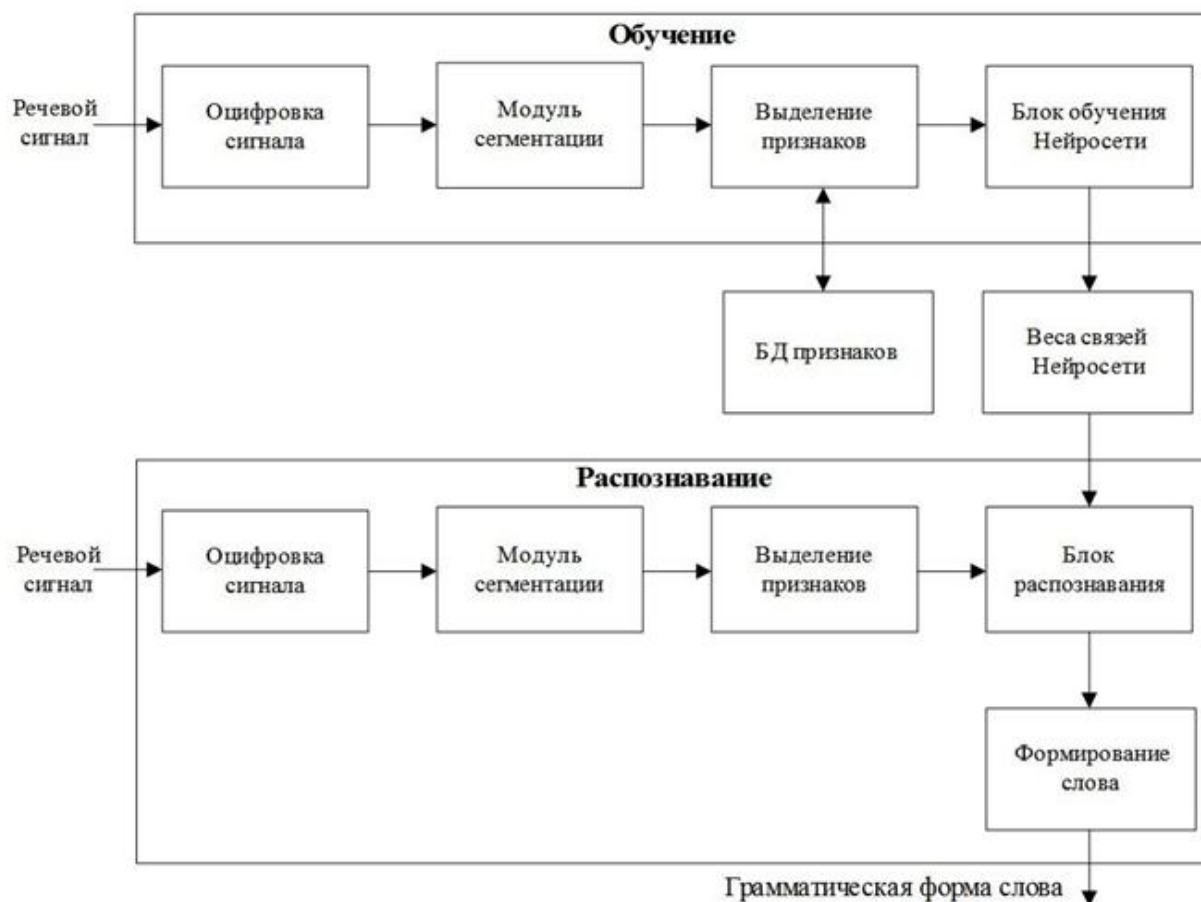
Целью внедрения алгоритма автоматической сегментации в модуль обучения является повышение эффективности работы системы распознавания речи, за счет сокращения времени обучения и уменьшения количества ошибок.

При построении системы «Речь-текст» в качестве распознаваемых единиц использовались фонемы. Преимущество фонемного подхода связано с использованием наименьшего числа отличительных фонологических классов, которые должны быть распознаны.

Рассмотрим алгоритм обучения на примере работы системы «Речь-текст» [1]. Структурная схема приложения приведена на рисунке 1.

На вход модуля обучения поступает речевой сигнал, который оцифровывается с определенной частотой дискретизации и разрядностью преобразования, эти параметры выбираются опционально перед началом

записи речи. С помощью алгоритма выделения слов, система разбивает речевой поток на отдельные изолированные участки.



**Рис. 1.** Структурная схема системы «Речь-текст»

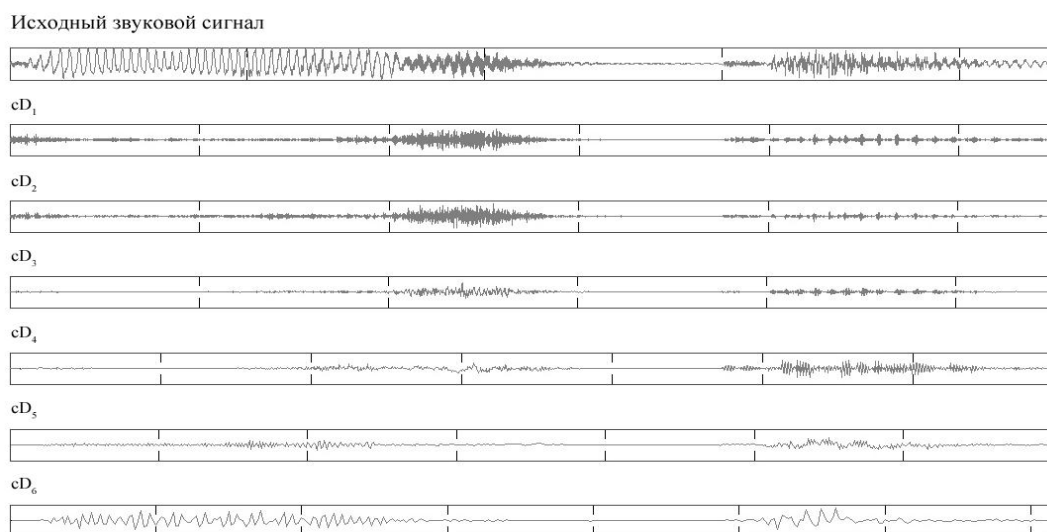
В процессе обучения диктор вручную определяет границы фонемы в звуковом сигнале посредством графического режима выделения. После чего для каждого выделенного сигнала фонемы на основе метода кратномасштабного вейвлет-преобразования вычисляются признаки, которые заносятся в базу данных и в дальнейшем будут использоваться в качестве обучающей выборки для нейронной сети. Применение алгоритма автоматической сегментации на этапе формирования базы признаков фонем существенно ускорит и упростит процесс обучения системы, практически исключая ручную работу, а также предоставит дополнительную информацию о количестве фонем в изолированном слове во время работы системы в режиме распознавания.

Важнейшим этапом анализа речевого сигнала является выделение его информативных признаков, которые могли бы однозначно его характеризовать. Существует набор математических методов, анализирующих речевой спектр. Вейвлет-преобразование одномерного сигнала – это его представление, сконструированное из материнского вейвлета, обладающего определенными свойствами за счет операций

сдвига во времени и изменения временного масштаба [2]. Идея состоит в представлении сигнала последовательностью образов с разной степенью детализации, что позволяет выявлять его локальные особенности и классифицировать их по интенсивности.

При исследовании сигналов полезно их представление в виде совокупности последовательных приближений грубой (аппроксимирующей) и уточненной (детализирующей) составляющих с последующим их уточнением итерационным методом. Каждый шаг уточнения соответствует определенному масштабу анализа (уровню декомпозиции). Такое представление каждой составляющей сигнала вейвлетами можно рассматривать как во временной, так и в частотной областях.

Вейвлет-коэффициенты аппроксимации соответствуют передаточной характеристике фильтра низких частот, а детализации — высокочастотному фильтру, можно рассматривать поведение речевого сигнала в различных частотных диапазонах. Частоты ниже 125 Гц не исследуются, т.к. не содержат информации, важной для задачи сегментации. Это обусловлено природой человеческой речи, охватывающей интервал 150 – 4000 Гц. Таким образом, достаточно шести уровней разложения. Вейвлет-разложение речевого сигнала на шесть уровней детализации представлено на рисунке 2.



**Рис. 2.** Детализирующие коэффициенты шести уровней декомпозиции

Сигнал разбивается на окна или фреймы. Количество окон определяется из отношения количества отсчетов исходного сигнала и выбранного размера окна достаточного для обнаружения границ межфонемного перехода. Размер окна для исходного сигнала – 512 отсчетов или 20 мс. Однако с изменением частоты дискретизации сигнала, размер окна для исходного сигнала может увеличиваться.

С увеличением уровня декомпозиции размер окна сокращается в 2 раза, это связано с уменьшением количества детализирующих

коэффициентов на последующих уровнях.

Для дальнейшего определения предполагаемых границ между окнами производится вычисление средней энергии вейвлет-коэффициентов в пределах определенного окна необходимо. Для каждого уровня строится числовая последовательность:

$$E_{j,k} = 10 \lg \sum_{k=0}^{L_j-1} d_{j,k}^2,$$

где  $d$  – детализирующие коэффициенты;  $k$  – номер вейвлет-коэффициента.

Для определения предполагаемых границ между соседними окнами текущего уровня детализации используется следующее неравенство:

$$|E_{j,i+1} - E_{j,i}| > \eta,$$

где  $\eta$  – порог энергии вейвлет коэффициентов, который определяется экспериментально.

Количество найденных границ между окнами суммируется и усредняется по количеству уровней декомпозиции. Если сумма превышает заданный пороговый коэффициент  $g$ , изменяющийся в пределах  $(0; 1)$ , то отмечается межфонемный переход.

В зависимости от порогового коэффициента изменяется чувствительность обнаружения границ между фонемами.

Для программной реализации модуля сегментации, использовалась система Matlab. Были разработаны следующие функции:

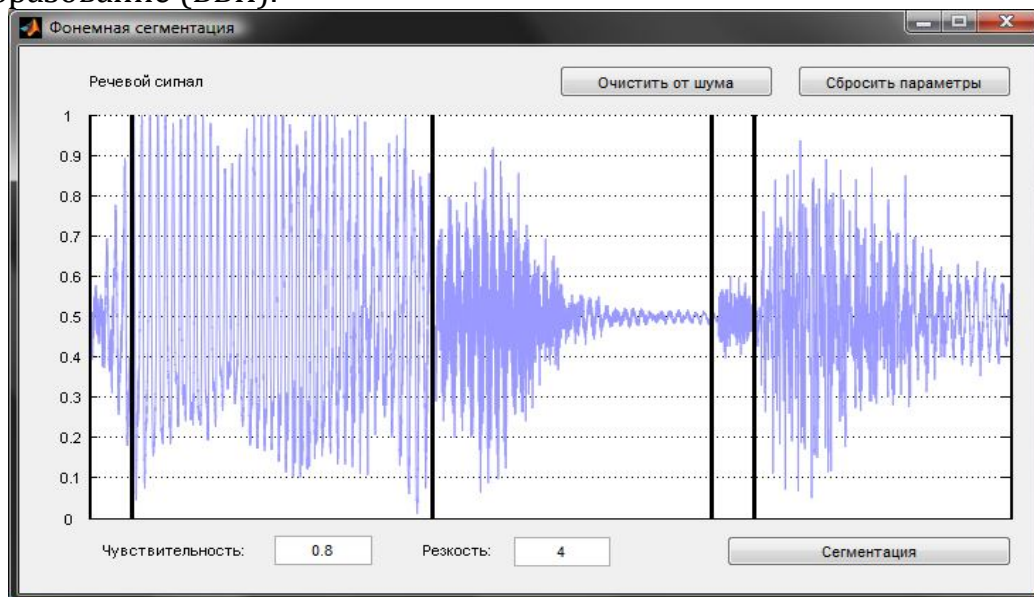
- Автоматическая сегментация речевых потоков;
- Предварительная обработка звукового сигнала;
- Изменение параметров сегментации;
- Редактирование результатов автоматической сегментации (удаление, перемещение границ) в ручном режиме;
- Прослушивание сегмента непосредственно при редактировании;
- Формирование списка фонем.

Алгоритм данного модуля сегментации включает в себя два коэффициента, которые определяют варианты расстановки границ сегментации: коэффициент чувствительности распознавания и коэффициент, реагирующий на быстроту изменения энергии вейвлет-показателей. Оба коэффициента находятся опытным путем.

До декомпозиции сигнала или вейвлет-анализа, необходимо выполнить предварительную обработку речевого сигнала, которая предполагает нормализацию сигнала и очистку от шума.

Нормализация сигнала позволяет уменьшить погрешности распознавания, связанные с тем, что диктор может произносить слова с разным уровнем громкости. В данном модуле использована пиковая нормализация. Если входной звуковой сигнал имеет слишком малый уровень громкости, то после нормализации может появиться шум, поэтому предусмотрена процедура шумоподавления, применяющаяся также в

случае, если исходный речевой сигнал содержит различного рода шумы. Для очистки сигналов от шума используется быстрое вейвлет-преобразование (БВП).



**Рис. 3.** Интерфейс модуля сегментации

На рисунке 3 представлен модуль сегментации в рабочем состоянии. Вертикальные линии, разделяющие исходный звуковой сигнал, отображают результат автоматической сегментации. В случае возникновения ошибок, пользователь имеет возможность переопределить предполагаемые границы и проверить результат. Доступны функции удаления границ, а также изменения таких параметров сегментации, как чувствительность и резкость. При нажатии кнопки «Сегментация» список фонем будет отображаться в главном окне системы «Речь-текст». После этого диктор должен сформировать выборку и обучить систему.

Для проведения экспериментальных исследований были записаны случайные слова, которые прошли предварительную обработку и автоматическую сегментацию.

Анализ результатов сегментации позволил сделать следующие выводы относительно алгоритма в целом:

- Взрывные согласные, стоящие перед гласными, делятся на два сегмента, соответствующие паузе и взрыву
- Переход от гласных к шипящим выделяется в отдельный сегмент
- Хорошо выделяются согласные твердые
- Плохо выделяются согласные мягкие
- Не выделяются согласные между двух гласных
- Появляется достаточно большое количество лишних границ при разделении согласных, реже – гласных
- Разделение рядом стоящих гласных происходит при минимальном пороговом коэффициенте, другими словами при большей



чувствительности, однако это создает лишние границы

- Границы слова, как правило, определяются правильно
- Границы согласных, как правило, определяются верно, однако между границами согласных появляются дополнительные границы;
- Автоматизированная сегментация занимает значительно меньшее время диктора, чем время затрачиваемое только на ручную сегментацию (приблизительно на 70%)

Анализ результатов показал, что требуются дополнительные исследования, связанные с определением границ фонем, в частности добавление к разработанному алгоритму, новых информативных признаков, предварительно изучив их характеристики на участках перехода между фонемами.

Основными проблемами данного алгоритма сегментации являются:

- появление лишних границ-переходов
- низкая чувствительность к переходам между гласными

Возможный способ решения данных проблем заключается в усреднении энергии рядом стоящих фреймов, а также возможностью опционально выбирать размер окна исходного сигнала. В случаях когда рядом находятся два гласных звука возможно применение формантного анализа.

Внедрение алгоритма фонемной сегментации в модуль обучения системы «Речь-текст» позволило сократить время формирования обучающей выборки на 70% по сравнению с ручным режимом работы.

### **Литература**

1. Кирякова, Г. С. Вероятностно-сетевая модель преобразования речи в текст. / Г. С. Кирякова, М. С. Медведев; Красн. гос. техн. ун-т. Красноярск, 2005. 9 с. Деп. в ВИНТИ 11.10.05, № 1300- В2005.

2. Вейвлет-анализ. Основы теории: пер. с нем. : учеб. пособие по спец. 01.02 "Прикл. математика": рекомендовано УМО в обл. электроники и прикл. математики. Блаттер, Кристиан. Б.м., 2006. 271с.: ил.

3. Медведев М. С. Интеллектуальный комплекс для исследования речевого сигнала. / М. С. Медведев, Г. С. Кирякова, А. С. Ли; Информационные технологии в науке, образовании, телекоммуникации и бизнесе '09: материалы XXXVI Международн. конференц. молодых ученых. Ялта-Гурзуф, 2009.

## **Оптимизация управления подвижными объектами с ограниченными внешними возмущениями**

### **Введение**

В большинстве задач, рассматриваемых в литературе (в частности, [1 – 4, 6]), возмущения либо отсутствуют, либо задаются в конкретном виде, к примеру, считаются неограниченно убывающими с течением времени. В данной работе рассматривается задача о динамической компенсации (подавлении) внешнего возмущения, о котором отсутствует какая-либо информация, за исключением факта его ограниченности, с заданием меры ограничения. В такой ситуации требуется выбрать закон управления, который давал бы наилучший возможный результат по качеству динамики для наихудшего варианта ограниченного возмущения.

Важным показателем качества управления является длительность переходного процесса, которая напрямую связана со степенью устойчивости рассматриваемой системы: чем ближе степень устойчивости к нулю, тем медленнее завершается переходный процесс.

Цель данной работы заключается в построении закона автоматического управления движением морского судна, обеспечивающего наименьшее отклонение судна по курсу при воздействии на него ограниченных внешних возмущений и желаемую степень устойчивости линейной замкнутой системы.

### **Математическая формулировка задачи**

Рассмотрим линейную стационарную систему

$$\begin{aligned}\dot{x} &= Ax + Bu + Hd(t), \\ y &= Cx,\end{aligned}\tag{1}$$

где  $x(t) \in R^n$  – вектор состояния системы,  $y(t) \in R^m$  – выход системы,  $u \in R^r$  – управляющее воздействие,  $d(t) \in R^l$  – ограниченное внешнее возмущение, удовлетворяющее условию:

$$\|d(t)\|_{\infty} \leq 1, \quad 0 \leq t < \infty.\tag{2}$$

Будем искать управление в форме линейной обратной связи по состоянию

$$u = Kx\tag{3}$$

Матрица усиления  $K$  подлежит определению.

Обозначим через  $\alpha_p$  – требуемую степень устойчивости характеристического полинома матрицы линейной замкнутой системы, а через  $\alpha = \alpha(K)$  – фактическую.

Требуется найти такой закон автоматического управления движением вида (3), который будет компенсировать ограниченные

внешние воздействия, обеспечивая заданное ограничение выхода и желаемую степень устойчивости характеристического полинома замкнутой системы.

С учетом введенных обозначений сформулированная задача примет следующий вид: необходимо найти матрицу коэффициентов  $K$  регулятора  $u = Kx$  такую, что

$$\|y(t, K)\|_{\infty} \leq \mu \text{ и } \alpha(K) \geq \alpha_p. \quad (4)$$

### **Методы и алгоритмы решения**

В качестве базового подхода предлагается использовать метод компенсации ограниченных внешних возмущений, предложенный в [5], который основан на применении инвариантных эллипсоидов.

**Определение.** Эллипсоид с центром в начале координат

$$\varepsilon_x = \{x \in R^n : x^T P^{-1} x \leq 1\}, \quad P > 0$$

называется **инвариантным** по переменной  $x$  (по состоянию) для динамической системы (1), (2), если из условия  $x(0) \in \varepsilon_x$  следует  $x(t) \in \varepsilon_x$  для всех моментов времени  $t \geq 0$ . Матрицу  $P$  будем называть матрицей эллипсоида  $\varepsilon_x$ .

**Замечание.** В определении и далее по тексту факт положительной определенности матрицы будем обозначать как  $P > 0$ .

Другими словами, любая траектория системы  $x(t)$ , исходящая из эллипсоида  $\varepsilon_x$ , в каждый последующий момент времени принадлежит  $\varepsilon_x$ .

Аналогичным образом определяется инвариантный эллипсоид по выходу системы  $y$ :

$$\varepsilon_y = \{y \in R^m : y^T (CPC^T)^{-1} y \leq 1\},$$

где  $P > 0$  – матрица эллипсоида  $\varepsilon_x$ .

Множество инвариантных эллипсоидов по выходу  $y$  позволяет оценить степень воздействия внешних возмущений на выход системы. Таким образом, выбрав из множества инвариантных эллипсоидов  $\varepsilon_y$  минимальный по некоторому критерию, мы ограничим влияние внешнего воздействия на выход системы  $y(t)$ .

В качестве критерия для выбора минимального инвариантного эллипсоида используем целевую функцию  $f(P) = \text{tr}(CPC^T)$ , которая определяет сумму квадратов полуосей инвариантного эллипсоида по выходу системы (1).

Для обеспечения требуемой степени устойчивости предлагается использовать метод, основанный на построении вспомогательного полинома с наперед заданной степенью устойчивости.

В работе [4] доказана теорема, описывающая алгоритм построения полинома с заранее заданной степенью устойчивости:

**Теорема.** Для любого вектора  $\gamma \in E^{n_d}$  степень устойчивости полинома

$$\Delta^*(s, \gamma) = \begin{cases} \tilde{\Delta}^*(s, \gamma), & \text{если } n_d - \text{четное;} \\ (s + a_{d+1}(\gamma, \alpha)) \tilde{\Delta}^*(s, \gamma), & \text{если } n_d - \text{нечетное,} \end{cases} \quad (5)$$

не меньше наперед заданной величины  $\alpha > 0$ , и наоборот, если степень устойчивости некоторого полинома  $\Delta(s)$  не меньше величины  $\alpha > 0$ , то можно указать такой вектор  $\gamma \in E^{n_d}$ , что справедливо тождество  $\Delta(s) \equiv \Delta^*(s, \gamma)$ , где

$$\tilde{\Delta}^*(s, \gamma) = \prod_{i=1}^d (s^2 + a_i^1(\gamma, \alpha)s + a_i^0(\gamma, \alpha)), \quad (6)$$

$$d = [n_d / 2], \quad a_i^1(\gamma, \alpha) = 2\alpha + \gamma_{i1}^2, \quad a_i^0(\gamma, \alpha) = \alpha^2 + \gamma_{i1}^2\alpha + \gamma_{i2}^2, \quad i = \overline{1, d}, \quad (7)$$

$$a_{d+1}(\gamma, \alpha) = \gamma_{d0}^2 + \alpha, \quad \gamma = \{\gamma_{11}, \gamma_{12}, \gamma_{21}, \gamma_{22}, \dots, \gamma_{d1}, \gamma_{d2}, \gamma_{d0}\} \quad (8)$$

Таким образом, задав произвольный вектор  $\gamma$ , по формулам (5) – (8) можно построить полином, который будет обладать требуемой степенью устойчивости.

Для построения системы автоматического управления движением на основе решения задачи (4), предлагается модифицировать метод компенсации ограниченных внешних воздействий (описанный в [5]), объединив его с методом обеспечения желаемой степени устойчивости (представленным в [4]) путем построения вспомогательного полинома.

Разработанный алгоритм состоит в следующем:

1. Следуя методу компенсации внешних воздействий:

1.1. Разрешаем уравнение

$$AP + PA^T - \gamma BB^T + \alpha P + \alpha^{-1} HH^T = 0, \quad P > 0.$$

относительно матричной переменной  $P$ , выражая ее компоненты через параметры  $\alpha$  и  $\gamma$ .

1.2. используем  $P(\alpha, \gamma)$  для определения коэффициентов регулятора  $k_1, k_2, k_3, k_4$  как функций параметров  $\alpha$  и  $\gamma$  с помощью соотношений:

$$u = -\gamma B^T P^{-1}(\alpha, \gamma)x = Kx \Rightarrow K(\alpha, \gamma) = -\gamma B^T P^{-1}(\alpha, \gamma)$$

2. Замыкаем систему (1) управлением  $u = K(\alpha, \gamma)x$ .

3. Строим характеристический полином замкнутой системы, зависящий от  $\alpha$  и  $\gamma$

$$d(s) = \det(sE - A - BK(\alpha, \gamma)C).$$

4. Задаем требуемую степень устойчивости  $\eta$  и строим вспомогательный полином  $\Delta^*(s)$  по методу, описанному в [4].

5. Приравнивая коэффициенты при соответствующих степенях в характеристическом и вспомогательном полиномах, получаем систему уравнений, решением которой является некоторое множество параметров  $\alpha$  и  $\gamma$ .

6. Решаем задачу минимизации  $\min_{\alpha, \gamma > 0} \text{tr}(CP(\alpha, \gamma)C^T)$ , учитывая дополнительные ограничения на параметры  $\alpha$  и  $\gamma$  из предыдущего шага

алгоритма.

7. Используя найденные  $\alpha$  и  $\gamma$ , по формуле  $u = YP^{-1}x$  получаем вектор  $K$  коэффициентов регулятора.

### **Пример реализации предложенного подхода**

Описанный алгоритм был применен для построения системы автоматического управления морским транспортным судном водоизмещением 6000 т.

В качестве математической модели объекта управления была принята система обыкновенных линейных дифференциальных уравнений, описывающая боковое движение морского судна

$$\begin{aligned}\dot{\beta} &= a_{11}\beta + a_{12}\omega + b_1\delta + h_1d(t), \\ \dot{\omega} &= a_{21}\beta + a_{22}\omega + b_2\delta + h_2d(t), \\ \dot{\varphi} &= \omega, \\ \dot{\delta} &= u,\end{aligned}\tag{9}$$

Здесь используются следующие обозначения (рис. 1):

$\omega$  – угловая скорость относительно вертикальной оси,

$\varphi$  – курс (положительным считается поворот на левый борт),

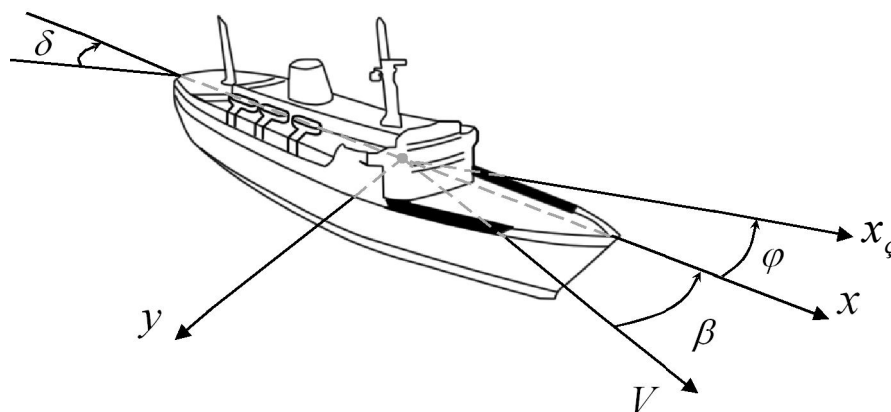
$\delta$  – угол отклонения вертикальных рулей (положительным считаем отклонение на левый борт),

$\beta$  – угол дрейфа (угол между вектором скорости и продольной осью судна),

$u$  – управляющее воздействие,

$d(t)$  – внешнее возмущение, определяемое порывами ветра и морским волнением.

Предполагается, что величина  $d(t)$  удовлетворяет ограничению (2).



**Рис. 1. Динамические переменные**

В качестве компенсирующего управления будем использовать регулятор в форме статической обратной связи по состоянию,

представленный уравнением

$$u = k_1\beta + k_2\omega + k_3\varphi + k_4\delta, \quad (10)$$

где  $k_1, k_2, k_3, k_4$  – коэффициенты, которые подлежат выбору в процессе решения задачи синтеза.

На рули и скорость их поворота (т.е. на управление) накладываются следующие ограничения:  $|\delta| \leq 30^\circ$ ,  $|u| \leq 3^\circ/\text{с}$ .

Преобразуем систему (9), обозначив вектор состояния через  $x = (\beta, \omega, \varphi, \delta)^T$ .

В соответствии с введенными обозначениями систему (9) можно записать в виде (1):

$$\begin{aligned} \dot{x} &= Ax + Bu + Hd(t), \\ y &= Cx, \end{aligned}$$

с матрицами  $A, B, C, H$ , имеющими постоянные компоненты:

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & b_1 \\ a_{21} & a_{22} & 0 & b_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad H = \begin{pmatrix} h_1 \\ h_2 \\ 0 \\ 0 \end{pmatrix}.$$

Тогда система (1), замкнутая управлением (3), примет вид

$$\begin{aligned} \dot{x} &= A_c x + H_c d(t), \\ y &= Cx, \end{aligned} \quad (11)$$

где  $A_c = A + BKC = A + BK$ ,  $H_c = H$ ,  $K = (k_1 \ k_2 \ k_3 \ k_4)$ .

При фиксированной скорости хода коэффициенты в модели (9) принимают следующие значения:

$$\begin{aligned} a_{11} &= -0,03408, \quad a_{12} = 0,56, \quad a_{21} = 0,015, \quad a_{22} = -0,306, \\ b_1 &= -0,0099, \quad b_2 = -0,00417, \quad h_1 = -0,0648, \quad h_2 = -0,0046. \end{aligned}$$

Зададим требуемую степень устойчивости  $\eta = 0,06$  и применим для нахождения коэффициентов регулятора разработанный алгоритм. В результате получен закон управления со следующими коэффициентами:

$$k_1 = 3,0981, \quad k_2 = 57,7472, \quad k_3 = 23,9121, \quad k_4 = -1,2793.$$

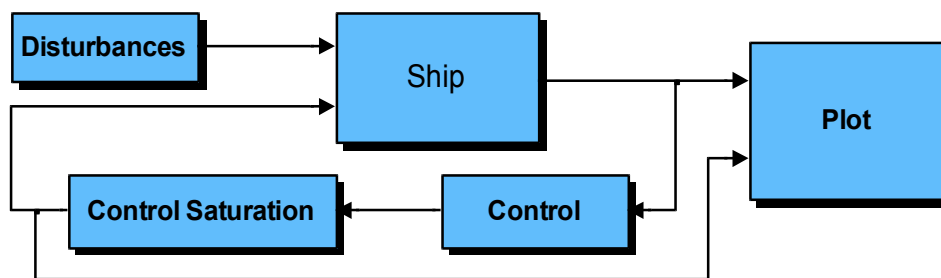
При использовании полученного регулятора собственные числа матрицы замкнутой системы принимают значения

$$\begin{aligned} \lambda_1 &= -0,1027, \quad \lambda_2 = -0,5465 + 0,0147i, \\ \lambda_3 &= -0,5465 - 0,0147i, \quad \lambda_4 = -0,5368, \end{aligned}$$

т.е. степень устойчивости можно оценить числом 0,1.

Для проверки качества найденного закона управления проведем имитационное моделирование в среде MATLAB-Simulink при воздействии на судно случайного ограниченного внешнего возмущения.

В качестве прикладного программного обеспечения была построена Simulink-модель системы управления, схема которой изображена на рис. 2.



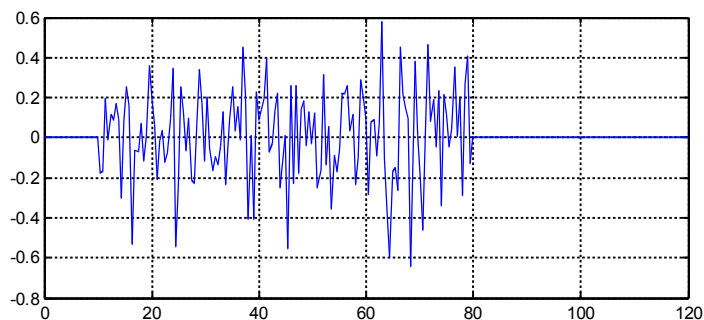
**Рис. 2.** Общая схема Simulink-модели системы

Построенная Simulink-модель объединяет в себе следующие блоки:

- блок **Ship** – объект управления;
- блок **Control** – центральное устройство формирования управляющего сигнала;
- блок **Control Saturation** – корректировка управляющего сигнала с учетом технических ограничений объекта управления;
- блок **Disturbances** – внешнее воздействие;
- блок **Plot** – визуализация динамических процессов.

Для проверки качества найденного закона управления проведем имитационное моделирование в среде MATLAB-Simulink при воздействии на судно ограниченного внешнего возмущения.

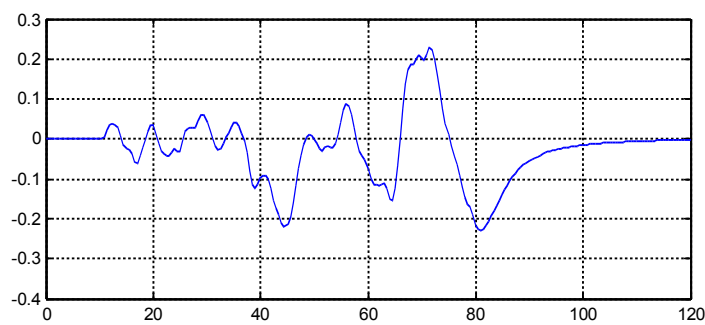
Пусть ограниченное внешнее возмущение представляет собой последовательность случайных ограниченных «всплесков» продолжительностью 70 секунд (рис. 3).



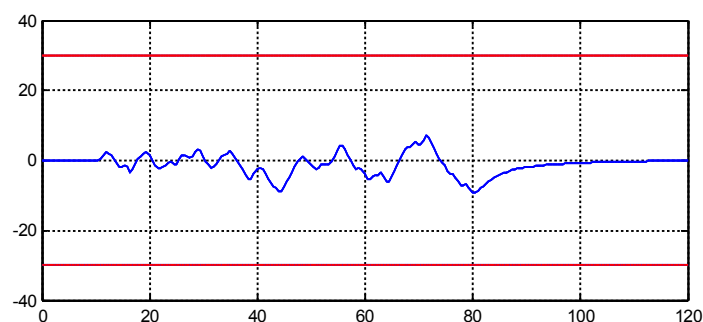
**Рис. 3.** Ограниченное внешнее возмущение

Графики изменения динамических переменных при воздействии такого возмущения изображены на рис. 4 и 5.

Как видно из рис. 4, отклонение от курса составляет менее  $0,3^\circ$ , что является хорошим показателем. Время стабилизации курса судна после окончания действия ограниченного возмущения составляет приблизительно 20 секунд. Данный пример иллюстрирует адекватную реакцию управления на случайное ограниченное воздействие.



**Рис. 4.** *Изменение курса*



**Рис. 5.** *Отклонение вертикальных рулей*

### **Заключение**

В результате проведенных исследований разработан алгоритм оптимальной компенсации ограниченных внешних воздействий с учетом требований к степени устойчивости. Разработанный алгоритм реализован в интегрированной среде MATLAB и его работоспособность проиллюстрирована на примере реального транспортного судна водоизмещением 6000 т.

### **Литература**

1. Веремей Е.И., Корчанов В.М., Коровкин М.В., Погожев С.В. Компьютерное моделирование систем управления движением морских подвижных объектов. СПб.: СПбГУ, 2002. 370 с.
2. Веремей Е.И. Синтез законов многоцелевого управления движением морских объектов // Гироскопия и навигация, 2009. Вып. 4. С. 3–14.
3. Лукомский Ю.А., Корчанов В.М. Управление морскими подвижными объектами. СПб.: Элмор, 1996. 320 с.
4. Веремей Е.И. Обеспечение заданной степени устойчивости регуляторами с неполной информацией // Изв. АН СССР. Техническая кибернетика, 1986. Вып. 4. С. 123–130.
5. Поляк Б.Т., Щербаков П.С. Робастная устойчивость и управление. М.: Наука, 2002. 303 с.
6. Fossen T.I. Guidance and control of ocean vehicles. John Wiley and Sons. New York, 1999. 494 p.



## ***Многоцелевое управление подвижными объектами в режиме реального времени***

### ***Введение***

В данной работе рассматривается задача о построении системы автоматического управления движением морского судна, позволяющей удерживать его на заданном курсе. В настоящее время накоплен существенный практический опыт проведения исследовательских и проектных работ по развитию таких систем. На его основе можно сформировать широкий круг содержательных задач, как в традиционных, так и в новых вариантах постановки, состоящих в обоснованном выборе структуры и параметров законов управления для различных режимов функционирования.

Математическая формализация этих постановок приводит к соответствующим задачам синтеза, решение которых имеет конечной целью практическую реализацию законов управления на борту.

Следует отметить, что задачи управления движением МПО отличаются многофункциональностью, многорежимностью, многокритериальностью и многомерностью. Синтез систем управления движением при этом осложняется нелинейностью привода и влиянием внешних факторов – морского волнения, ветра, импульсных воздействий.

В работах [1 – 5] представлена теория многоцелевого подхода к синтезу, учитывающая указанные особенности систем управления движением морских судов. Эта теория построена на единой методологии разработки алгоритмического обеспечения, основанной на применении современных оптимизационных методов теории управления и идеологии автоматизированного компьютерного проектирования. В отличие от известных методов синтеза законов управления, улучшающих отдельные динамические характеристики, многоцелевой подход поддерживает комплексное проектирование систем.

Подобные системы в линейном приближении должны быть обязательно устойчивыми, однако устойчивость является далеко не единственным требованием, предъявляемым к ним. В частности важно, чтобы система управления обладала свойством астатизма по регулируемой координате, т.е. способностью приводить ошибку регулирования к нулю при наличии постоянного внешнего воздействия.

Проектирование таких систем является нетривиальной задачей, постановка и решение которой с привлечением современных математических и компьютерных методов существенно зависит от типа

судна, его параметров и назначения. В связи с этим, цель данной работы состоит в нахождении закона автоматического управления движением морского судна, обеспечивающего астатизм замкнутой системы по курсу.

### **Постановка задачи**

Рассмотрим LTI-систему с одним входом и одним выходом

$$\begin{aligned}\dot{x} &= Ax + bv, \\ w &= Cx + dv,\end{aligned}\quad (1)$$

где  $v$  и  $w$  – входной и выходной сигнал соответственно.

**Определение.** Линейная стационарная система (1) со входом  $v$  и выходом  $w$  называется *астатической по выходной переменной*, если для ступенчатого входного сигнала  $v(t) \equiv v_0 \cdot 1(t)$  при любом вещественном  $v_0$  выполняется условие:  $w_0 = 0$ , где  $w_0$  – значение выхода, соответствующее положению равновесия системы с постоянным входным сигналом  $v_0$ .

Существо задачи состоит в том, чтобы для МПО со входом  $f$  и выходом  $\varphi$  построить закон автоматического управления, обеспечивающий астатизм замкнутой системы по курсу, т.е. найти такой закон управления  $u$ , который обеспечивал бы выполнение равенства  $\lim_{t \rightarrow \infty} \varphi(t) = 0$  для любого числа  $f_0$  при условии, что входное воздействие формируется по закону  $f(t) \equiv f_0 \cdot 1(t)$ .

### **Синтез астатических законов управления**

Традиционный подход к обеспечению астатизма по выходу, связанный с использованием структуры пропорционально-интегрально-дифференциального (ПИД) регулятора, описан в [6].

Другой подход к обеспечению астатизма состоит в использовании специализированного скоростного регулятора по состоянию [1,3], уравнение которого имеет следующую структуру:

$$u = \mu \dot{x} + v u, \quad (2)$$

Скоростной регулятор однозначно строится в силу уравнений динамики объекта на базе исходного управления  $u = kx + k_0 \delta$ . При этом коэффициенты исходного и скоростного управления определяются с обязательным учетом следующих требований:

- замкнутая линейная система должна быть устойчивой;
- перерегулирование  $P$  и длительность переходного процесса  $T_p$  не должны превосходить заданных величин, т.е.

$$P \leq P_0, \quad T_p \leq T_{p0};$$

- при переходе к скоростному закону коэффициенты регулятора должны обеспечивать астатизм замкнутой системы по выходу:

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

Покажем, что регулятор (2) обеспечивает астатизм системы по выходу, считая, что его коэффициенты гарантируют устойчивость. С этой целью рассмотрим уравнения замкнутой системы управления движением МПО

$$\begin{aligned}
\dot{x} &= Ax + B\delta + Hf, \\
\dot{\delta} &= u, \\
y &= Cx, \\
u &= \mu\dot{x} + \nu y.
\end{aligned}
\tag{3}$$

Из этих уравнений следует, что в статике, если возможности рулей ее допускают, имеем:

$$\begin{aligned}
Ax + B\delta + Hf &= 0, \\
y &= 0.
\end{aligned}
\tag{4}$$

Таким образом, структура регулятора (2) гарантирует обеспечение астатизма по выходу при условии устойчивости замкнутой системы.

С учетом отмеченного обстоятельства, идея предлагаемого метода обеспечения астатизма по курсу состоит в последовательном поиске коэффициентов исходного (базового) закона управления, обеспечивающего выполнение первых двух требований с переходом к скоростному закону управления (2) в силу уравнений объекта.

Отметим, что функционалы  $P$  и  $T_p$  определяют противоречивые требования к регулятору. Для достижения определенного компромисса между ними при поиске коэффициентов базового закона будем использовать интегральный квадратичный функционал

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt,$$

заданный на движениях замкнутой системы

$$\begin{aligned}
\dot{x} &= Ax + B\delta + Hf, \\
\dot{\delta} &= u, \\
y &= Cx, \\
u &= kx + k_0\delta.
\end{aligned}
\tag{*}$$

Минимизация этого функционала позволяет найти коэффициенты  $k, k_0$  базового стабилизирующего регулятора.

Схема поиска коэффициентов базового закона выглядит следующим образом:

1. Выделяем каким-либо способом вектор  $\gamma \in E^p$  параметров, от выбора которых однозначно зависят знакоположительная матрица  $Q = Q(\gamma)$  и положительно-определенная матрица  $R = R(\gamma)$  и задаем начальные приближения для его компонентов.

2. Решаем задачу LQR-оптимального синтеза для системы (\*) с интегральным квадратичным функционалом

$$J = J(\gamma) = \int_0^{\infty} (x^T Q(\gamma) x + u^T R(\gamma) u) dt,$$

определяя при этом коэффициенты  $k = k(\gamma), k_0 = k_0(\gamma)$  базового стабилизирующего регулятора.

3. Пересчитываем базовый регулятор в силу линейных уравнений объекта к скоростной форме (2), полагая при этом внешнее воздействие нулевым и определяя ее коэффициенты  $\mu = \mu(\gamma), \nu = \nu(\gamma)$ .

4. На движениях получившейся замкнутой системы (3) определяем величины функционалов  $P = P(\gamma)$  и  $T_p = T_p(\gamma)$ , а также вычисляем значение вспомогательного функционала

$$I = I(\gamma) = P(\gamma) - P_0 + |P(\gamma) - P_0| + T_p(\gamma) - T_{p0} + |T_p(\gamma) - T_{p0}|.$$

5. Если для данного вектора  $\gamma$  имеем  $I = I(\gamma) > 0$ , с помощью любого численного метода спуска задаем новое приближение вектора  $\gamma$  и повторяем вычисления по пунктам 2 – 5, минимизируя функционал  $I = I(\gamma)$  до достижения им нулевого глобального экстремума, соответствующего выполнению желаемых ограничений.

### **Пример синтеза для транспортного судна**

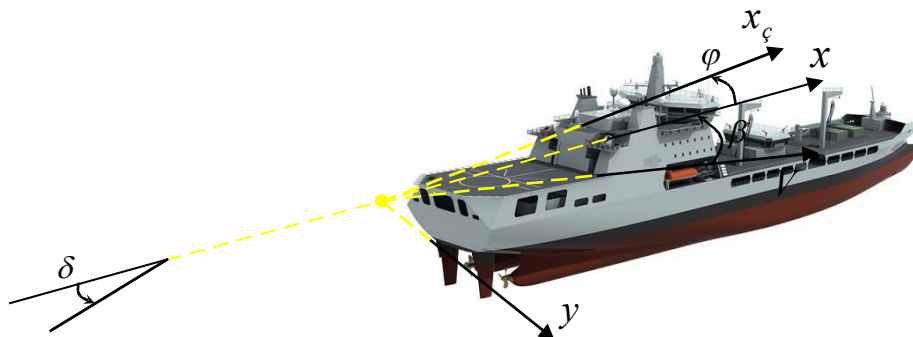
В качестве объекта управления для построения закона управления был выбран транспортный корабль с водоизмещением 6000 т.

Примем в качестве математической модели объекта управления следующие линейные стационарные (LTI) дифференциальные уравнения, приближенно описывающие его движение по курсу

$$\begin{aligned}\dot{\beta} &= a_{11}\beta + a_{12}\omega + b_1\delta + h_1f(t), \\ \dot{\omega} &= a_{21}\beta + a_{22}\omega + b_2\delta + h_2f(t), \\ \dot{\phi} &= \omega, \\ \dot{\delta} &= u.\end{aligned}\tag{5}$$

На отклонения рулей и на скорость их поворота (т.е. на управление) накладываются следующие технические ограничения:  $|\delta| \leq 30^\circ, |u| \leq 3^\circ/\text{с}$ .

Реализуем предложенную схему для рассматриваемой конкретной ситуации с математической моделью МПО в виде (5).



**Рис. 1. Основные параметры объекта управления**

Здесь используются следующие обозначения (рис. 1):

- $\omega$  – угловая скорость относительно вертикальной оси,
- $\phi$  – курс (положительным считается поворот на левый борт),
- $\delta$  – угол отклонения вертикальных рулей (положительным считаем отклонение на левый борт),
- $\beta$  – угол дрейфа (угол между вектором скорости и продольной осью судна),
- $u$  – управляющее воздействие,
- $F(t) = h_1f(t)$  – боковая возмущающая сила,
- $M(t) = h_2f(t)$  – возмущающий момент по курсу,

$f(t)$  – ступенчатое внешнее воздействие, определяемое влиянием на судно порывов ветра.

В данном случае имеем  $x = (\beta, \omega, \varphi, \delta)^T$  – вектор состояния системы,  $y = x$  – ее выход. В соответствии с введенными обозначениями систему (5) можно переписать в форме

$$\begin{aligned} \dot{x} &= Ax + Bu + Hf, \\ y &= Cx, \end{aligned} \quad (6)$$

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & b_1 \\ a_{21} & a_{22} & 0 & b_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad H = \begin{pmatrix} h_1 \\ h_2 \\ 0 \\ 0 \end{pmatrix}.$$

Тогда система (5), замкнутая управлением

$$u = Kx = (k_1 \quad k_2 \quad k_3 \quad k_4)x, \quad (7)$$

примет вид:

$$\begin{aligned} \dot{x} &= A_c x + H_c f(t), \\ y &= Cx, \end{aligned} \quad (8)$$

где  $A_c = A + BKC = A + BK$ ,  $H_c = H$ .

Введём квадратичный функционал

$$\begin{aligned} J = J(u) &= \int_0^{\infty} (\eta_1^2 \omega^2 + \eta_2^2 \varphi^2 + \rho_3^2 u^2) dt = \\ &= \int_0^{\infty} ((\rho_1^2 + 0.0001 \gamma_1)^2 \omega^2 + (\rho_2^2 + 0.0001 \gamma_2)^2 \varphi^2 + \rho_3^2 u^2) dt, \end{aligned} \quad (9)$$

где  $\rho_1^2, \rho_2^2, \rho_3^2$  – вещественные параметры. При этом будем считать, что величины  $\rho_1^2$  и  $\rho_2^2$  – заданы, а  $\gamma_1, \gamma_2, \rho_3^2$  – параметры, которые являются компонентами варьируемого вектора  $\gamma$  и подлежат поиску.

Конкретизация приведенной выше схемы синтеза приводит к следующему вычислительному алгоритму:

1. Задаем начальные значения параметров функционала.
2. Решаем задачу LQR-синтеза с критерием оптимальности (9) и находим вектор коэффициентов регулятора  $K$ .
3. Исключим из закона управления (7) переменные  $\beta$  и  $\delta$ , разрешив относительно них первые два уравнения системы (5), и подставим их в закон управления, используя соответствующие величины первых производных. Это дает управление в виде

$$u = \mu_1 \dot{\beta} + \mu_2 \dot{\omega} + \mu_3 \dot{\omega} + v \varphi, \quad (10)$$

$$\text{где } \mu_1 = \frac{k_1}{(a_{11}/b_1 - a_{21}/b_2)b_1} + k_4 \left( \frac{1}{b_1} - \frac{a_{11}}{b_1^2 (a_{11}/b_1 - a_{21}/b_2)} \right),$$

$$\mu_2 = \frac{-k_1}{(a_{11}/b_1 - a_{21}/b_2)b_2} + k_4 \frac{a_{11}}{b_1 b_2},$$

$$\mu_3 = \frac{-k_1 (a_{12}/b_1 - a_{22}/b_2)}{(a_{11}/b_1 - a_{21}/b_2)} + k_2 + k_4 \left( -\frac{a_{12}}{b_1} + \frac{a_{11} (a_{12}/b_1 - a_{22}/b_2)}{b_1 (a_{11}/b_1 - a_{21}/b_2)} \right), \quad v = k_3.$$

4. На движениях замкнутой системы вычисляем функционалы

перерегулирования и быстродействия и определяем значение вспомогательного функционала  $I$ .

5. Минимизируем вспомогательный функционал за счет варьирования используемых параметров до достижения нулевого глобального экстремума.

Таким образом, сначала фиксируем параметры функционала, затем находим вектор коэффициентов регулятора  $K$ , который доставляет минимум функционалу (9), по нему вычисляем новые коэффициенты закона управления (10) и изучаем динамический процесс. Если качество процесса нас не устраивает, то возвращаемся к шагу 1 и изменяем начальные значения параметров функционала.

Коэффициенты рассматриваемого судна в линейной модели (5) при фиксированной скорости хода имеют следующие значения:

$$a_{11} = -0,03408, \quad a_{12} = 0,56, \quad a_{21} = 0,015, \quad a_{22} = -0,306, \quad b_1 = -0,0099, \quad b_2 = -0,00417.$$

Для проведения контрольного моделирования зададим возмущения  $F$  и  $M$  так, чтобы положение равновесия в системе (5) определялось равенствами  $\beta_0 = 1,5^\circ$ ,  $\delta_0 = 12^\circ$ .

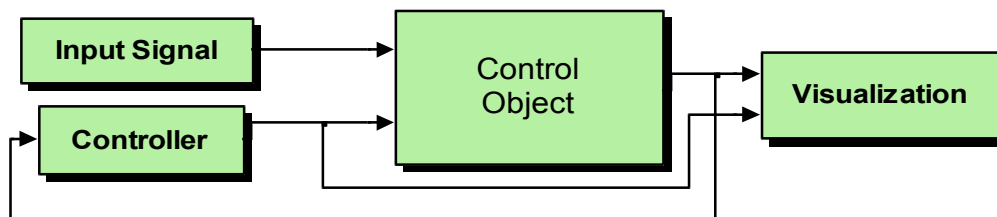
Из системы (5) получаем:

$$\begin{aligned} h_1 &= -a_{11}\beta_0 - b_1\delta_0, \\ h_2 &= -a_{21}\beta_0 - b_2\delta_0. \end{aligned} \quad (11)$$

Из (11) находим

$$h_1 = 0,00296, \quad h_2 = 0,000481.$$

Для компьютерного и имитационного моделирования процессов, происходящих в динамической системе, анализа ее свойств, проверки качества найденного закона управления, была использована подсистема Simulink среды MATLAB. Базовой частью прикладного программного обеспечения в данном случае служит Simulink-модель объекта управления, блок-схема которой изображена на рис. 2.



**Рис. 2.** Блок-схема компьютерной модели

Представленная модель содержит в себе следующие блоки:

блок **Control Object** – объект управления,

блок **Controller** – центральное устройство управления,

блок **Input Signal** – внешнее воздействие на систему,

блок **Visualization** – визуализация динамических процессов.

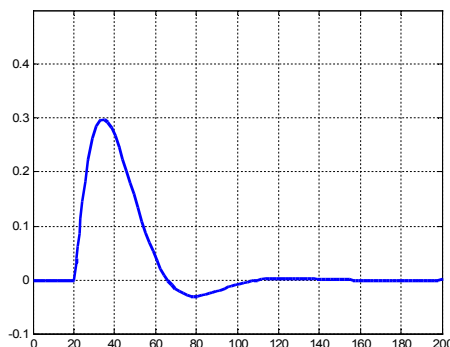
В результате реализации алгоритма построения скоростного регулятора получен закон управления с коэффициентами

$$k_1 = 0,955, \quad k_2 = 6,574, \quad k_3 = 0,941, \quad k_4 = -0,3504.$$

При этом параметры функционала (9) установились на следующих значениях:

$$\eta_1^2 = 4,98631, \eta_2^2 = 0,01502, \rho_3^2 = 0,01694.$$

Действительно, указанный закон управления обеспечивает астатизм замкнутой системы по курсу, что проиллюстрировано на рис. 3.



**Рис. 3.** Изменение курса при ступенчатом воздействии

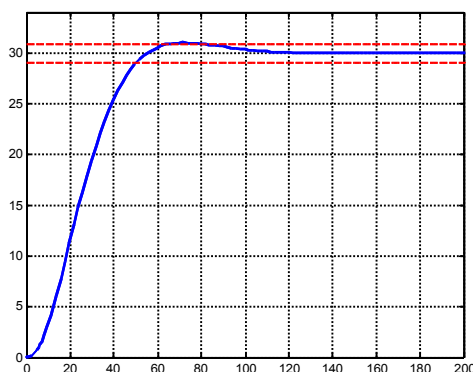
При использовании скоростного регулятора собственные числа матрицы замкнутой системы принимают следующие значения:

$$\lambda_1 = -0,323 + 0,111i, \lambda_2 = -0,323 - 0,111i, \lambda_3 = -0,101, \lambda_4 = -0,0543.$$

Степень устойчивости линейной замкнутой системы оценивается константой 0,05.

Для проверки качества найденного управления оно было использовано для автоматического выполнения маневра, состоящего в повороте морского судна по курсу на  $30^\circ$ .

Графики изменения динамических переменных представлены на рис. 4 и 5.

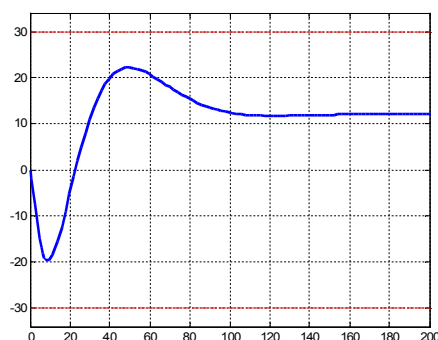


**Рис. 4.** Изменение курса при маневре

**Рис. 5.** Отклонение рулей при маневре

Как видно из рис. 4, при использовании скоростного регулятора по состоянию полностью отсутствует перерегулирование по курсу, а переходный процесс завершается за 47 секунд, что почти в 2 раза быстрее, чем при использовании классического ПИД-регулятора [6]. При этом рис. 5 демонстрирует плавную работу рулей, без перехода в крайние допустимые

положения.



### **Заключение**

В данной работе предложен и реализован способ построения управления, обеспечивающего астатизм замкнутой системы по курсу. Указанный алгоритм реализован в интегрированной среде MATLAB. В пакете Simulink проведено компьютерное моделирование построенной системы управления движением транспортного судна.

### **Литература**

1. Веремей Е.И., Корчанов В.М., Коровкин М.В., Погожев С.В. Компьютерное моделирование систем управления движением морских подвижных объектов. СПб.: СПбГУ, 2002. 370 с.
2. Лукомский Ю.А., Корчанов В.М. Управление морскими подвижными объектами. СПб.: Элмор, 1996. 320 с.
3. Веремей Е.И. Синтез законов многоцелевого управления движением морских объектов // Гироскопия и навигация, 2009. Вып. 4. С. 3–14.
4. Fossen T.I. Guidance and control of ocean vehicles. John Wiley and Sons. New York, 1999. – 494 p.
5. Вагущенко Л.Л., Цымбал Н.Н. Системы автоматического управления движением судна. Одесса: Латстар, 2002. 310 с.
6. Смирнов М.Н., Федорова М.А. Компьютерное моделирование системы астатической стабилизации курса морского судна // Процессы управления и устойчивость: Труды 41-й международной научной конференции аспирантов и студентов. СПб.: Издат. Дом С.-Петербур. ун-та, 2010. С. 495-500.



## ***Синтез робастного цифрового регулятора для системы магнитной левитации***

### ***1. Введение***

Вопрос обеспечения робастных свойств систем управления является одним из ключевых при решении практических задач. Значимость этого вопроса определяется наличием различных неопределенностей в задании математических моделей, используемых при синтезе алгоритмов управления, включая дополнительную неучтенную динамику и внешние возмущения.

В работе рассматривается задача синтеза цифрового регулятора, обеспечивающего робастную устойчивость замкнутой системы с учетом желаемых модальных свойств. Важной особенностью предлагаемой процедуры синтеза является построение такого регулятора, для которого собственные числа замкнутой номинальной системы располагаются внутри заданной области в единичном круге, а любая замкнутая возмущенная система устойчива. При этом неопределенность математической модели объекта управления представляется в частотной области.

В качестве практического приложения приводится задача синтеза цифрового регулятора для системы магнитной левитации. Неопределенность математической модели этой системы вызвана сложностью точного описания силы магнитного поля, действующей на объект управления. Работоспособность и эффективность предложенного подхода к синтезу иллюстрируется примером компьютерного имитационного моделирования в среде MATLAB/Simulink.

### ***2. Математическая модель магнитной левитации***

На рис. 1 представлена общая схема системы магнитной левитации. На стальной шарик, находящийся в воздухе, действует две силы: сила тяжести  $F_g$ , направленная вертикально вниз, и сила притяжения  $F_m$ , создаваемая электромагнитом, когда по нему течет ток  $I$ . На рисунке приняты следующие обозначения:  $V$  – напряжение, подаваемое на катушку,  $R$  – сопротивление катушки,  $L$  – ее индуктивность,  $x_b$  – расстояние от электромагнита до шарика. Начало системы координат  $O_{xy}$  расположено на поверхности электромагнита, ось  $Ox$  направлена вертикально вниз.

Запишем систему нелинейных дифференциальных уравнений, представляющих математическую модель магнитной левитации. Для этого введем переменные:  $x_1 = x = x_b$ ,  $x_2 = \dot{x}_b$ ,  $x_3 = I$ ,  $u = V$ . На основе законов

электрических цепей и второго закона Ньютона, а также с учетом выражения для силы притяжения электромагнита [2], составим уравнения математической модели:

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = g - \frac{1}{2} \cdot \frac{K_m x_3^2}{M x_1^2}, \quad \dot{x}_3 = -\frac{R}{L} x_3 + \frac{1}{L} u. \quad (1)$$

Здесь  $M$  – масса шарика,  $g$  – гравитационная постоянная,  $K_m$  – магнитная постоянная. Управляющим воздействием является напряжение  $u$ , а контролируемой переменной – смещение шарика  $x_b$ .

В рассматриваемой системе измеряются две величины: сила тока  $I$  в контуре и положение шарика  $x_b$ . Последняя величина измеряется с помощью оптического датчика, встроенного в пьедестал. Дополним систему (1) уравнениями измерений:

$$y_1 = x_1, \quad y_2 = x_3. \quad (2)$$

Важно отметить, что уравнения (1), (2) приближенно представляют динамику системы ввиду сложности описания электромагнитного поля и наличия дополнительных неучтенных воздействий с его стороны. При этом наибольшие трудности возникают при описании поведения шарика вблизи поверхности электромагнита.

Составим уравнения линейного приближения для системы (1) в окрестности положения равновесия  $(x_{10}, 0, x_{30}) = (x_{b0}, 0, I_0)$ , соответствующего фиксированному напряжению  $u = u_0$ . Заметим, что значения  $x_{b0}$  и  $I_0$  связаны соотношением:  $x_{b0} = \sqrt{K_m / 2gM} I_0$ . Пусть  $\bar{x}_1 = x_1 - x_{10}$ ,  $\bar{x}_2 = x_2$ ,  $\bar{x}_3 = x_3 - x_{30}$  и  $\bar{u} = u - u_0$  – переменные, определяющие отклонения динамических параметров от положения равновесия. Тогда система в отклонениях примет вид

$$\dot{\bar{x}}_1 = \bar{x}_2, \quad \dot{\bar{x}}_2 = \frac{2g}{x_{b0}} \bar{x}_1 - \frac{2g}{I_0} \bar{x}_3, \quad \dot{\bar{x}}_3 = -\frac{R}{L} \bar{x}_3 + \frac{1}{L} \bar{u}. \quad (3)$$

Аналогично, из (2) получаем уравнения измерений:

$$\bar{y}_1 = \bar{x}_1, \quad \bar{y}_2 = \bar{x}_3, \quad (4)$$

где  $\bar{y}_1 = y_1 - y_{10}$ ,  $\bar{y}_2 = y_2 - y_{20}$ . Запишем линейную модель (3), (4) в матричной форме:

$$\begin{pmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_2 \\ \dot{\bar{x}}_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ a_{21}^0 & 0 & a_{23}^0 \\ 0 & 0 & a_{33}^0 \end{pmatrix} \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ b_0 \end{pmatrix} \bar{u}, \quad \begin{pmatrix} \bar{y}_1 \\ \bar{y}_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{pmatrix}. \quad (5)$$

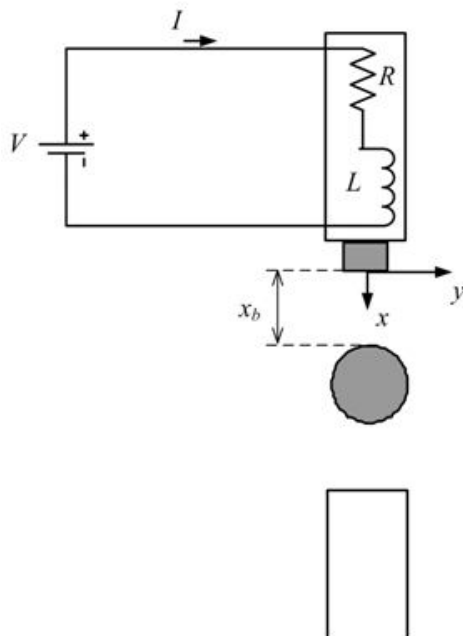


Рис.1. Схема системы магнитной левитации

Здесь  $a_{21}^0 = \frac{2g}{x_{b0}}, a_{23}^0 = -\frac{2g}{I_0}, a_{33}^0 = -\frac{R}{L}, b_0 = \frac{1}{L}$ . Теперь, если подставить в (5)

значения физических параметров устройства магнитной левитации, то получим номинальную линейную модель, представляющую динамику системы в окрестности положения равновесия.

Отметим, что нулевое положение равновесия системы (5) является неустойчивым, что соответствует неустойчивости равновесия исходной нелинейной системы (1) по вертикальному положению шарика.

В силу неточности описания магнитного поля, значения коэффициентов в модели (5) являются приближенными. При этом линейные модели, соответствующие различным положениям шарика  $x_b$ , отличаются главным образом значением коэффициента  $a_{21}$  [2]. В связи с этим, будем считать, что модель (5) является номинальной, а область неопределенности для модели объекта ограничим, варьируя значение коэффициента  $a_{21}$  в заданных пределах.

### **3. Синтез робастного цифрового регулятора**

Рассмотрим процедуру синтеза робастного цифрового регулятора на примере SISO-системы, представленной в дискретном времени математической моделью:

$$y = P_n(z)u, \quad (6)$$

где  $P_n(z)$  – номинальная передаточная функция объекта. Будем считать, что объект (6) стабилизируется регулятором

$$u = W(z, \mathbf{h})y, \quad (7)$$

где  $W(z, \mathbf{h})$  – передаточная функция регулятора с фиксированной структурой (заданы степени полиномов в числителе и знаменателе),  $\mathbf{h} \in E^r$  – вектор настраиваемых параметров, подлежащих выбору при синтезе закона управления.

В дальнейшем будем полагать, что объект управления с номинальной математической моделью (6) подвержен воздействию неструктурированных возмущений и его действительная модель имеет вид:

$$y = P(z)u, \quad (8)$$

где  $P(z)$  – передаточная функция возмущенного объекта, отличная от номинальной. Введем допустимую границу возмущения номинальной математической модели. Для этого, согласно [1], зададим ограничение сверху амплитудно-частотной характеристики (АЧХ) относительного возмущения модели:

$$\left| \Delta_0(e^{j\omega}) \right| = \left| \frac{P(e^{j\omega}) - P_n(e^{j\omega})}{P_n(e^{j\omega})} \right| \leq \beta, \quad \omega \in [0, \pi], \quad (9)$$

где  $\beta > 0$  – заданное фиксированное число. Условие (9) определяет допустимый «коридор» для вариаций АЧХ фактического (возмущенного) объекта. Тогда множество всевозможных передаточных функций (8) возмущенных моделей имеет вид:

$$\Phi = \left\{ P : P = P_n(1 + \Delta_0), \quad |\Delta_0(e^{j\omega})| \leq \beta, \quad \omega \in [0, \pi] \right\}.$$

Потребуем, чтобы регулятор (7) с вектором настраиваемых параметров  $\mathbf{h}$  стабилизировал любой объект с возмущенной моделью (8), при условии  $P(z) \in \Phi$ , то есть обеспечивал робастную устойчивость замкнутой системы. Известно, что достаточным условием сохранения устойчивости замкнутой системы (8), (7) для любых относительных возмущений модели объекта, удовлетворяющих (9), является выполнение следующего неравенства [1]:

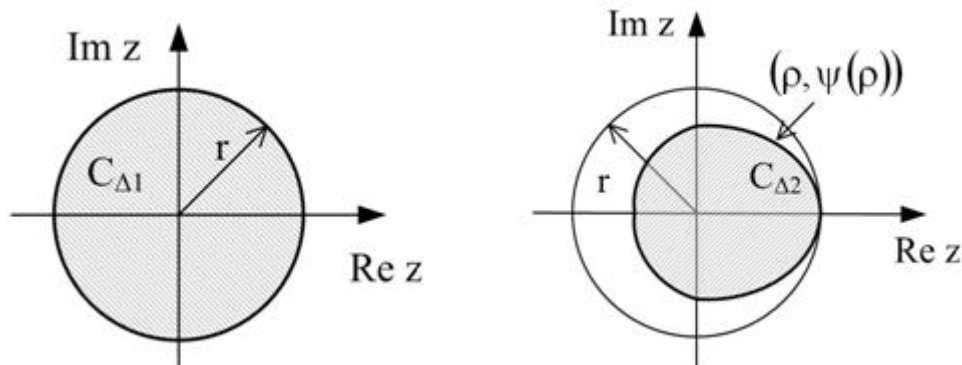
$$\|T(z, \mathbf{h})\|_{\infty} = \max_{\omega \in [0, \pi]} |T(e^{j\omega}, \mathbf{h})| < 1/\beta, \quad (10)$$

где  $T(z, \mathbf{h}) = W(z, \mathbf{h})(1 + P_n(z)W(z, \mathbf{h}))^{-1}P_n(z)$ .

Помимо требования робастной устойчивости, будем также учитывать желаемые модальные свойства замкнутой системы. Обозначим через  $\Delta_3(z, \mathbf{h})$  – характеристический полином замкнутой номинальной системы (6), (7). Тогда выбор вектора параметров  $\mathbf{h}$  регулятора должен обеспечивать выполнение условия

$$\delta_i(\mathbf{h}) \in C_{\Delta}, \quad i = \overline{1, n_d}, \quad n_d = \deg \Delta_3(z, \mathbf{h}), \quad (11)$$

где  $\delta_i$  – корни характеристического полинома  $\Delta_3(z, \mathbf{h})$ ,  $n_d$  – степень полинома  $\Delta_3(z, \mathbf{h})$ ,  $C_{\Delta}$  – область желаемого расположения корней внутри единичного круга. Рассмотрим два варианта задания области  $C_{\Delta}$  внутри единичного круга, показанные на рис. 2.



**Рис. 2.** Области  $C_{\Delta 1}$  и  $C_{\Delta 2}$  желаемого расположения корней

Формальное определение указанных областей имеет вид:

$C_{\Delta} = C_{\Delta 1} = \{z \in \mathbb{C}^1 : |z| \leq r\}$ , где  $r \in (0, 1)$  – заданное вещественное число;

$C_{\Delta} = C_{\Delta 2} = \{z \in \mathbb{C}^1 : z = \rho \cdot e^{\pm i\varphi}, 0 \leq \rho \leq r, 0 \leq \varphi \leq \psi(\rho)\}$ , где  $r \in (0, 1)$  – заданное вещественное число,  $\psi(\xi)$  – вещественная функция переменной  $\xi \in (0, r]$ , принимающая значения из отрезка  $[0, \pi]$ , причем  $\psi(r) = 0$ .

Смысл введения этих областей состоит в следующем. Первая из них определяет ограничение снизу на степень устойчивости, т.е. на длительность переходных процессов в замкнутой системе, а вторая, в дополнение к этому, определяет ограничение на меру колебательности.

На основе требования робастной устойчивости (10) и с учетом

желаемых модальных свойств (11), введем следующее определение допустимого множества настраиваемых параметров регулятора (7):

$$\Omega_H = \left\{ \mathbf{h} \in E^r : \delta_i(\mathbf{h}) \in C_\Delta, i = \overline{1, n_d}, \|T(z, \mathbf{h})\|_\infty < 1/\beta \right\}. \quad (12)$$

Поставим задачу об оптимальном выборе вектора параметров  $\mathbf{h}$  регулятора (7) на допустимом множестве (12). Для этого на движениях замкнутой номинальной системы (6), (7) зададим некоторый функционал, характеризующий качество процессов управления, вида

$$J = J(\{y[i]\}, \{u[i]\}) = J(\mathbf{W}(q, \mathbf{h})) = J(\mathbf{h}) \geq 0.$$

Здесь  $\{u[i]\}$  и  $\{y[i]\}$  – последовательности, представляющие цифровые сигналы на входе и выходе номинальной модели (6).

Рассмотрим следующую задачу параметрического синтеза:

$$J = J(\mathbf{h}) \rightarrow \inf_{\mathbf{h} \in \Omega_H}, \quad (13)$$

где  $\Omega_H$  – допустимое множество (12) настраиваемых параметров. Отметим, что задача оптимизации (13) является специфическим вариантом задачи нелинейного программирования, для решения которой удобно выполнить параметризацию рассматриваемых областей  $C_\Delta$  с использованием  $n$ -мерных вещественных векторов, как показано в [3]. Обозначим через  $\mathbf{h}^*$  решение задачи (13). Тогда результатом синтеза является оптимальный регулятор вида (7) с передаточной функцией  $W(z, \mathbf{h}^*)$ .

#### 4. Реализация для системы магнитной левитации

Рассмотрим пример синтеза цифрового робастного регулятора для реального устройства MAGLEV магнитной левитации [4]. В силу конструктивных особенностей устройства, положение шарика  $x_b$  может варьироваться в диапазоне от 0 до 0.014 м. При этом  $x_b = 0$ , когда шарик «прилипает» к магниту и  $x_b = 0.014$  м, если шарик находится на пьедестале. Ниже приведены значения физических параметров системы:

$$L = 0.4125 \text{ Н}, R = 11 \text{ Ом}, K_m = 6.5308\text{E-}005 \text{ Н} \cdot \text{м}^2 / \text{А}^2, M = 0.068 \text{ кг}.$$

Рассмотрим положение равновесия системы для смещения шарика  $x_{b0} = 0.006$  м. Учитывая уравнения (1), получаем, что этому смещению соответствует равновесное значение силы тока  $I_0 = 0.86$  А. Подставив данные величины в (5), построим номинальную грубую модель, описывающую поведение системы в окрестности положения равновесия:

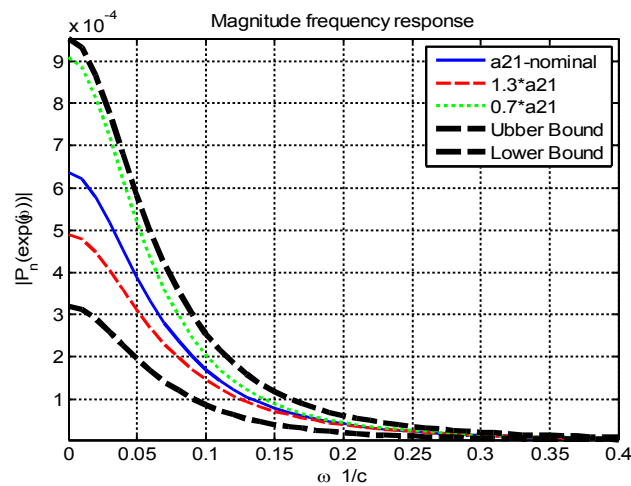
$$\begin{pmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_2 \\ \dot{\bar{x}}_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 3270 & 0 & -22.88 \\ 0 & 0 & -26.67 \end{pmatrix} \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 2.42 \end{pmatrix} \bar{u}, \quad \begin{pmatrix} \bar{y}_1 \\ \bar{y}_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{pmatrix}. \quad (14)$$

Собственные числа матрицы разомкнутой системы (14) равны  $\lambda_1 = 57.18$ ,  $\lambda_2 = -57.18$ ,  $\lambda_3 = -26.67$ , что свидетельствует о её неустойчивости. Сформируем уравнения дискретной модели на основе (14) с шагом дискретности  $T_s = 0.002$  секунды. В результате получим:

$$\begin{pmatrix} \bar{x}_1[k+1] \\ \bar{x}_2[k+1] \\ \bar{x}_3[k+1] \end{pmatrix} = \begin{pmatrix} 1.0065 & 0.002 & 0 \\ 6.5543 & 1.0065 & -0.0447 \\ 0 & 0 & 0.9481 \end{pmatrix} \begin{pmatrix} \bar{x}_1[k] \\ \bar{x}_2[k] \\ \bar{x}_3[k] \end{pmatrix} + \begin{pmatrix} 0 \\ -0.0001 \\ 0.0047 \end{pmatrix} \bar{u}[k]. \quad (15)$$

Собственные числа матрицы разомкнутой системы (15) равны  $\mu_1 = 1.1212$ ,  $\mu_2 = 0.8919$ ,  $\mu_3 = 0.9481$ . Найдем номинальную передаточную функцию  $P_n(z)$  объекта (15) от входа  $\bar{u}$  к выходу  $\bar{x}_1 = x_b - x_{b0}$ .

Ограничим область неопределенности модели (15), варьируя значение коэффициента  $a_{21}$  непрерывной модели (14) в пределах  $\pm 30\%$  от номинального значения. На рис. 3 показаны амплитудно-частотные характеристики номинальной модели и возмущенных моделей, со значением коэффициента  $\tilde{a}_{21}$  равным  $0.7a_{21}$  и  $1.3a_{21}$ . Выберем значение  $\beta$  в условии (9) таким образом, чтобы допустимый «коридор» вариации АЧХ возмущенного объекта включал в себя все построенные кривые. На рис. 3 показан допустимый «коридор», соответствующий значению  $\beta = 0.5$ . То есть, будем в дальнейшем считать, что АЧХ относительного возмущения удовлетворяет условию:  $|\Delta_0(e^{j\omega})| \leq 0.5$ ,  $\omega \in [0, \pi]$ .



**Рис. 3.** Допустимый «коридор» вариации частотной характеристики

Примем следующую структуру регулятора для модели (15):

$$\bar{u}[k] = K\bar{x}[k], \quad (16)$$

где  $K \in E^3$  – вектор настраиваемых параметров, подлежащих выбору в процессе синтеза закона управления. Зададим область  $C_\Delta$  желаемого расположения корней характеристического полинома замкнутой номинальной системы (15), (16) как показано на рис. 4. Данная область определяется параметрическими уравнениями:

$$\psi(\rho) = \begin{cases} \text{tg}(\theta) \ln(r/\rho), & r \exp(-\pi/\text{tg}(\theta)) \leq \rho \leq r, \\ \pi, & 0 < \rho \leq r \exp(-\pi/\text{tg}(\theta)), \end{cases}$$

где  $r = 0.95$ ,  $\theta = \pi/6$ .

В итоге, рассмотрим следующую задачу параметрического синтеза:

$$J = J(K) \rightarrow \inf_{K \in \Omega \subseteq E^3}, \quad (17)$$

где допустимое множество  $\Omega$ , с учетом заданных ограничений на допустимые вариации АЧХ возмущенного объекта и желаемой области  $C_\Delta$  расположения корней, имеет вид:

$$\Omega = \{ \mathbf{K} \in \mathbb{E}^3: \delta_i(\mathbf{K}) \in C_\Delta, i = \overline{1,3}, \|T(z, \mathbf{K})\|_\infty < 2 \}.$$

Здесь  $\delta_i$  – собственные числа замкнутой номинальной системы (15), (16). В результате решения задачи синтеза (17) получаем следующий вектор параметров оптимального регулятора:

$$\mathbf{K} = (0.18 \quad -27.36 \quad -0.26).$$

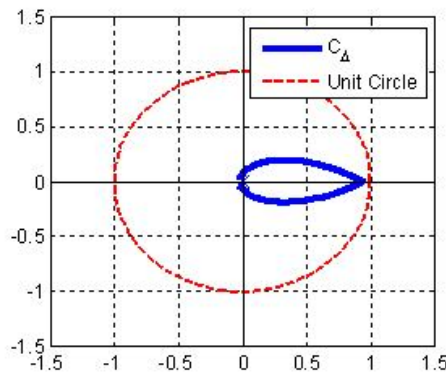
Важно отметить, что управляющий сигнал  $u$ , который подается на катушку с током, вычисляется как сумма двух компонент:

$$u[k] = u_c[k] + \bar{u}[k] = u_c[k] + \mathbf{K}(\mathbf{x}[k] - \mathbf{x}^*),$$

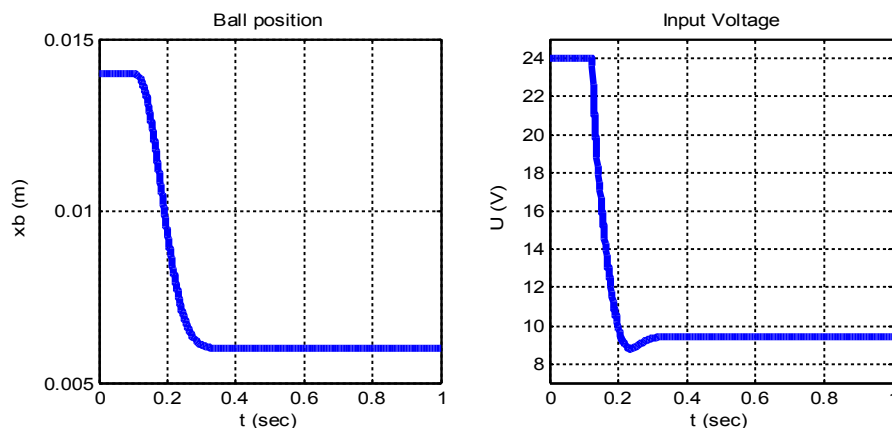
где  $u_c = u_c(x_{b0})$  – постоянное слагаемое, компенсирующее силу тяжести в положении равновесия, а  $\mathbf{x}^*$  – заданный командный сигнал. Величина  $u_c$  вычисляется на основе уравнений (1) по следующей формуле:

$$u_c(x_{b0}) = R\sqrt{2gM/K_m}x_{b0}. \quad \text{Так как значение}$$

**Рис.4. Область желаемого расположения корней**



коэффициента  $K_m$  зависит от положения шарика  $x_b$  и известно только его приближенное значение, то для обеспечения астатизма замкнутой системы по выходу  $x_b$  необходимо вместо приведенного выше закона управления перейти к его аналогу, в котором участвуют не сами компоненты вектора состояния, а их производные. Естественно, что реализация такого закона управления возможна только с использованием асимптотического наблюдателя. На рис. 5 представлен пример использования синтезированного регулятора для стабилизации шарика в заданной точке  $x_b^* = 0.006$  м.



**Рис. 5. Стабилизация заданного положения шарика  $x_b^* = 0.006$  м**

## 5. Заключение

В работе рассмотрена задача синтеза робастного цифрового регулятора для системы магнитной левитации. Выполнена формализованная постановка задачи параметрического синтеза регулятора с учетом требований робастной устойчивости и обеспечения желаемых

модальных свойств замкнутой системы. Результат синтеза продемонстрирован для системы магнитной левитации на примере стабилизации положения шарика в заданной точке.

### **Литература**

1. Веремей Е.И. Анализ в среде MATLAB робастных свойств систем стабилизации плазмы // Exponenta Pro: Математика в приложениях: Науч.-практ. журн. 2003. №3. С. 20-27.
2. Сотникова М.В. Идентификация линейной модели магнитной левитации в среде MATLAB // Тр. IV международной научной конференции «Проектирование инженерных и научных приложений в среде MATLAB». 2009. С. 507-522.
3. Веремей Е.И., Сотникова М.В. Стабилизация плазмы на базе прогноза с устойчивым линейным приближением // Вестн. С.-Петерб. ун-та. Сер. 10. 2011. Вып. 1. С. 116-133.
4. MAGLEV: Magnetic Levitation Plant. User Manual. Quanser.



## Оглавление

<b>ПЛЕНАРНОЕ ЗАСЕДАНИЕ .....</b>	<b>3</b>
<i>Тихомиров В.П.</i> От электронного обучения к Smart – Университету.....	4
<i>Кореньков В.В.</i> Распределенная система для обработки, хранения и анализа экспериментальных данных Большого адронного коллайдера.....	8
<i>Ким А.К.,</i> <i>Бычков И.Н.,</i> <i>Волконский В.Ю.,</i> <i>Воробушков В.В.,</i> <i>Груздов Ф.А.,</i> <i>Михайлов М.С.,</i> <i>Парахин Ю.Н.,</i> <i>Сахин Ю.Х.,</i> <i>Семенухин С.В.,</i> <i>Слесарев М.В.,</i> <i>Фельдман В.М.</i> Архитектурная линия «Эльбрус» сегодня: микропроцессоры, вычислительные комплексы, программное обеспечение.....	21
<i>Сухомлин В.А.</i> Новые методологические решения и международные стандарты в области маги- стерского ИТ-образования.....	30
<i>Ромасевич П.В.,</i> <i>Смирнова Е.В.</i> Направления сотрудничества компании D-Link с высшими учебными заведения- ми.....	42
<b>СЕКЦИЯ 1. ИТ-ОБРАЗОВАНИЕ: МЕТОДОЛОГИЯ, МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ.....</b>	<b>50</b>
<i>Моисеев Е.И.,</i> <i>Ложкин С.А.,</i> <i>Тихомиров В.В.</i> Наука и образование на факультете ВМК МГУ.....	51
<i>Монахов В.М.,</i> <i>Монахов Д.Н.</i> Технологический учебник полного цикла и управление качеством формируемых профессиональных компетенций.....	60
<i>Бахусова Е.В.</i> Нечёткая математика для ИТ- специалистов.....	76
<i>Бедный А.Б.,</i> <i>Гергель В.П.,</i> <i>Ерушкина Л.В.,</i> <i>Кузенков О.А.</i> Основная образовательная программа ННГУ подготовки бакалавров по направ- лению ФИИТ на английском языке.....	83
<i>Белоус В.В.,</i> <i>Бобровский А.В.,</i>	

Добряков А.А.,  
Карпенко А.П.,  
Смирнова Е.В.

Ментально-структурированное интегральное оценивание качества обучения. .92

Бронов С.А.,  
Степанова Е.А.,  
Кудрявцева Ю.М.,  
Афонасенко Н.С.,  
Камилов И.К.

Методологические проблемы автоматизированного формирования образовательных программ в рамках ФГОС ВПО.....103

Варнавский А.Н.

Дистанционные курсы, учитывающие текущее психофизиологическое состояние обучающегося.....112

Давлеткиреева Л.З.,  
Махмутова М.В.

Инновационная модель подготовки ИТ-специалиста в образовательной среде вуза.....118

Зиндер Е. З.,

Юнатов И. Г.

От метамоделей к практике: основы устройства адаптивной системы описания и измерения профессиональных компетенций.....129

Зубарева Е.В.,  
Сухомлин В.А.

Таксономия направлений диверсификации программ подготовки бакалавров и магистров в области информационных технологий.....146

Иртегов Д.В.,  
Чурина Т.Г.

Мониторинг подготовки одаренных студентов и выпускников вузов, прошедших обучение в рамках специальных учебных групп.....159

Ишакова Е.Н.,  
Медведева М.В.

Управление рисками подготовки будущих программных инженеров на основе методологии IDEF0.....167

Каминский С.Е.,  
Лукьянец О.Ф.

Самый действенный способ научиться чему-либо самому – попробовать научить этому ... компьютер.....173

Козел О.Н.,  
Каракозов С.Д.,  
Рыжова Н.И.

Информационно-вычислительная компетентность выпускника вуза (на примере учителя информатики).....182

Коржик И.А.,  
Протасова И.В.,  
Толстобров А.П.

Тестовая система Moodle и качество тестовых заданий.....187

Крюков С.В.

Разработка образовательного стандарта ЮФУ для бакалавриата по направлению подготовки «Бизнес-информатика» и ООП по профилю «Бизнес-аналитика»....	197
<i>Курзаева Л.В.</i>	
Компетентностный подход в подготовке ИТ-специалистов: обзор в поисках точки опоры.....	208
<i>Нечаев В.В.,</i>	
<i>Панченко В.М.,</i>	
<i>Комаров А.И.</i>	
Методическое обеспечение ИТ-образования в вузе: от технологии подготовки учебных материалов к организации индивидуальной учебной деятельности обучающегося.....	215
<i>Петухова Т.П.</i>	
Опыт модульного проектирования профессиональных образовательных программ подготовки техников-программистов в соответствии с ФГОС СПО.....	224
<i>Подбельский В.В.,</i>	
<i>Максименкова О.В.</i>	
Особенности формулировок тестовых заданий по программированию.....	233
<i>Пузанкова А.Б.</i>	
Оптимизация инженерно-графической подготовки студентов.....	245
<i>Сутягин М.В.,</i>	
<i>Сторожик И.В.</i>	
Международные стандарты в области технологий совместного обучения.....	256
<i>Титова С.В.</i>	
Условия успешной интеграции мобильных устройств в учебный процесс.....	262
<i>Усольцев В.Л.</i>	
Современные проблемы информатики в подготовке магистров образования по программе «Информационные технологии в физико-математическом образовании».....	270
<i>Фёдоров А.А.</i>	
Стратегии работы с классами источников финансирования в интернет-пространстве.....	277
<b>СЕКЦИЯ 2. E-LEARNING, ИТ В ОБРАЗОВАНИИ .....</b>	<b>289</b>
<i>Абдалова О. И.,</i>	
<i>Гураков А.В.,</i>	
<i>Сметанин С. В.,</i>	
<i>Шульц Д. С.</i>	
Технология организации и проведения вебинаров.....	290
<i>Брылева В.А.,</i>	
<i>Сафонова О.П.</i>	
Технология организации и методического сопровождения электронной веб-ориентированной среды для учебного процесса гуманитарного профиля.....	296
<i>Воронов М.П.,</i>	
<i>Часовских В.П.</i>	
Проектирование виртуального тренажера управления предприятием лесной промышленности средствами нечетких моделей и его реализация с применением средств ASP.NET .....	305
<i>Горохова Л.А.</i>	
Невозможное возможно: дистанционная «Школа юного переводчика» на базе LMS MOODLE.....	312

<i>Грищенко Г.А.</i>	
Разработка информационной системы «Электронное портфолио преподавателя» на платформе 1С:ПРЕДПРИЯТИЕ 8.2.....	320
<i>Зеленко Л.С.,</i>	
<i>Загуменнов Д.А.</i>	
Принципы разработки виртуальной обучающей системы «3Ducation».....	326
<i>Исаченко А.Н.</i>	
Об использовании информационных ресурсов и программных средств в курсе «Модели данных и СУБД».....	334
<i>Матчин В.Т.,</i>	
<i>Свечников С.В.</i>	
Использование веб-системы тестирования при проведении занятий в ВУЗе.....	340
<i>Сиговцев Г.С.,</i>	
<i>Чарута М.А.</i>	
Сравнительное оценивание электронных учебных ресурсов.....	346
<i>Симонова Е.В.</i>	
Обучение распределенному решению сложных задач на основе мультиагентного подхода.....	352
<i>Тихонов А.В.,</i>	
<i>Мерзляков А.А.,</i>	
<i>Богданов В.С.</i>	
Дистанционные практики интеграции образовательного и научно-исследовательского процессов.....	363
<i>Фролов Н.Н.,</i>	
<i>Груничев А.В.,</i>	
<i>Малиованов М.В.,</i>	
<i>Хмелев Р.Н.</i>	
Опыт разработки и использования контрольно-обучающих комплексов по техническим дисциплинам (на примере дисциплины «Теплотехника»).....	371
<i>Штанюк А.А.</i>	
Системы управления проектами для поддержки учебного процесса.....	380
<b>СЕКЦИЯ 3. ОБРАЗОВАТЕЛЬНЫЕ РЕСУРСЫ И ЛУЧШАЯ ПРАКТИКА ИТ-ОБРАЗОВАНИЯ.....</b>	<b>386</b>
<i>Астахова И.Ф.,</i>	
<i>Фалалеева Ю.Л.</i>	
Построение нечеткой гибридной системы и ее применение к образовательным структурам.....	387
<i>Демиденко А.П.,</i>	
<i>Кан М.В.,</i>	
<i>Сундеев Д.Г.</i>	
Интерактивная система оценки качества образования.....	392
<i>Колчков В.И.</i>	
Практика создания и применения образовательного интернет-ресурса.....	403
<i>Милов В.Р.,</i>	
<i>Егоров Ю.С.,</i>	
<i>Алипова Н.А.</i>	
Формирование учебного контента в интеллектуальной системе поддержки информационных процессов подразделения ВУЗа.....	412
<i>Мусаелян А.Г.</i>	

Из опыта работы по технологическому учебнику полного цикла «Математика» .....	423
<i>Нарышкин Д. Г.</i>	
Компьютерная математика в курсе «Физическая химия» .....	430
<i>Преснецова В.Ю.,</i>	
<i>Демина Ю.А.</i>	
Использование информационных технологий при управлении вузом.....	441
<i>Симонова И.В.,</i>	
<i>Козлов О.А.,</i>	
<i>Бочаров М.И.</i>	
Дидактические единицы по основам информационной безопасности в стандартах подготовки специалистов для системы образования.....	447
<i>Симуни М.Л.</i>	
Использование инструментальной системы Sparse Assist при обучении программированию задач обработки разреженных матриц.....	457
<i>Терехов А.Н.</i>	
Роль инноваций в обучении аспирантов в области программной инженерии...	464
<i>Тихоненко А.В.</i>	
Современные образовательные ресурсы на базе компьютерных технологий....	474
<b>СЕКЦИЯ 4. ИССЛЕДОВАНИЯ И РАЗРАБОТКИ В ОБЛАСТИ НОВЫХ ИТ И ИХ ПРИЛОЖЕНИЙ.....</b>	<b>481</b>
<i>Абашев А.В.</i>	
<i>Терин Д.В.</i>	
<i>Мурашев Д.А.</i>	
Разработка компонента для формирования библиографии и перекрестных ссылок в MSWord.....	482
<i>Абдрахманова А.М.</i>	
Использование двумерных штрихкодов для создания системы позиционирования в помещении.....	488
<i>Ваграменко Я.А.,</i>	
<i>Назаренко А.П.,</i>	
<i>Сарьян В.К.,</i>	
<i>Сущенко Н.А.,</i>	
<i>Шелупанов А.А.,</i>	
<i>Беляков К.О.,</i>	
<i>Мещеряков Р.В.</i>	
Опыт создания доверенной среды для образовательных инфокоммуникационных услуг.....	495
<i>Васильев С.Н.,</i>	
<i>Смирнова Н.В.,</i>	
<i>Суконнова А.А.,</i>	
<i>Душкин Д.Н.,</i>	
<i>Абраменков А.Н.,</i>	
Методы интеллектуализации обучающих систем.....	503
<i>Горлушко Д.С.</i>	
Особенности организации регрессионного тестирования компиляторов на вычислительных комплексах серий «Эльбрус-3m» и «МЦСТ-R».....	520
<i>Давлеткиреева Л.З.,</i>	
<i>Чусавитина Г.Н.</i>	

Анализ и установление уровня зрелости информационной инфраструктуры организации для управления непрерывностью бизнеса.....	529
<i>Ермаков Е.Ю.</i>	
Анализ и сравнение процессов обработки запроса к таблице в параллельных колоночных и строчных хранилищах данных.....	545
<i>Кадан А.М.,</i>	
<i>Кадан М.А.</i>	
Технологии бизнес-моделирования и разработка дополнительной функциональности языка BPEL.....	556
<i>Казенников А.О.,</i>	
<i>Трифонов Н.И.</i>	
Гибридный алгоритм синтаксического разбора для анализа новостных потоков .....	564
<i>Липунцов Ю.П.</i>	
Технологии организации хранения открытых связанных данных и использование их в образовании.....	575
<i>Любецкий В.А.</i>	
Компьютерное моделирование в задачах регуляции работы генов и эволюции организмов.....	586
<i>Namiot Dmitry</i>	
A peer-to-peer model for location sharing services.....	596
<i>Романов В.Ю.</i>	
Реализация метамодели языка UML на основе хранилища данных фирмы Google .....	605
<i>Ромасевич Е.П.,</i>	
<i>Ромасевич П.В.</i>	
Исследование влияния трафика пиринговых сетей на работу сети MetroEthernet смешанной архитектуры на основе имитационной модели.....	611
<i>Рукавичников А.В.,</i>	
<i>Базарон С.А.</i>	
Способ распознавания и оценивания ответов на тестовые задания «Типовая задача».....	619
<i>Тхан Зо У</i>	
Параллельные вычисления в системах визуализации данных.....	631
<i>Чусавитин М.О.</i>	
Применение ИТ-решений при управлении непрерывностью бизнеса.....	635
<i>Якимович Б.А.,</i>	
<i>Щенятский А.В.,</i>	
<i>Вологдин С.В.</i>	
Разработка пакета прикладных программ по повышению энергоэффективности системы централизованного теплоснабжения.....	643
<b>СЕКЦИЯ 5. ТЕОРЕТИЧЕСКАЯ ИНФОРМАТИКА.....</b>	<b>655</b>
<i>Агеносов А.В.,</i>	
<i>Андреева Д. Д.,</i>	
<i>Хмелькова Н.В.</i>	
Математические методы комбинированной оценки прогноза спроса в логистике .....	656
<i>Verkhovsky Boris S.</i>	

Parametric Presentation of Generators: Application in Cryptographic Systems Design .....	664
<i>Гайдамака Ю.В.,</i>	
<i>Самуйлов А.К.</i>	
Модель процесса обмена данными в потоковой P2P сети с применением стратегии Layered Coding Aware Rarest-First.....	667
<i>Емельченков Е.П.,</i>	
<i>Мунерман В.И.,</i>	
<i>Синицын И.Н.</i>	
Опыт использования систем компьютерной математики в управлении процессом обучения математическим дисциплинам.....	673
<i>Кристалинский В.Р.,</i>	
<i>Белоусов В.В.</i>	
Прогнозирование результатов обучения на основе нечеткого и хаотического моделирования.....	680
<i>Мокров Е.В.,</i>	
<i>Самуйлов К.Е.</i>	
Модель облачных вычислений в виде системы массового обслуживания.....	685
<i>Мунерман В.И.,</i>	
<i>Сенчилов В.В.,</i>	
<i>Синицын И.Н.</i>	
Системы компьютерной математики как средство формирования детерминированного и стохастического воображения у учащихся.....	690
<i>Пархоменко С.С.</i>	
Об оценке времени выполнения приложения в распределённых вычислительных сетях.....	699
<i>Панюкова Т.А.</i>	
Алгоритм построения оптимального эйлерова покрытия для многосвязного графа.....	706
<i>Погосян К.С.</i>	
Расстояние между лингвистическими шкалами.....	714
<i>Савченко В.В.,</i>	
<i>Акатьев Д.Ю.</i>	
Адаптивный алгоритм формирования фонетической базы данных для систем автоматического распознавания речи методом фонетического декодирования слов.....	725
<i>Сопин Э.С.</i>	
Анализ показателей качества функционирования SIP-сервера с гистерезисным управлением нагрузкой.....	734
<b>СЕКЦИЯ 6. НАУЧНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В ОБРАЗОВАНИИ И НАУКЕ.....</b>	<b>740</b>
<i>Гайфуллин Б.Н.,</i>	
<i>Туманов В.Е.</i>	
Предметно-ориентированные системы научной осведомленности в науке и образовании.....	741
<i>Кальдин Д.А.,</i>	
<i>Мурашев Д.А.,</i>	
<i>Клинаев Ю.В.,</i>	
<i>Терин Д.В.</i>	

Оптимизация алгоритмов компьютерного моделирования трехмерного физического пространства.....	751
<i>Лепихин Т.А.</i>	
Применение компьютерной графики для визуализации процессов.....	756
<i>Митюков В.В.</i>	
Задача уточнения результатов измерений путем согласования с априорными связями.....	764
<i>Периг А.В.,</i>	
<i>Голоденко Н.Н.,</i>	
<i>Бойко И. И.</i>	
О расчетной гидродинамической оценке влияния входной подвижной стенки штампа на кинематические особенности вязкого течения полимерного материала при равноканальном угловом прессовании.....	770
<i>Родионов В.И.</i>	
О специальных многомерных сплайнах лагранжевого типа.....	776
<i>Романчук С. П.,</i>	
<i>Терин Д.В.</i>	
Математическое моделирование композитных сред.....	781
<i>Синицын И.Н.,</i>	
<i>Шаламов А.С.,</i>	
<i>Корепанов Э.Р.,</i>	
<i>Агафонов Е.С.</i>	
Опыт создания и обучения средствам интегрированной логистической поддержки изделий наукоемкой продукции в среде MATLAB.....	786
<i>Скрипачев В.О.</i>	
Вопросы использования Interactive Data Language при создании научных приложений.....	792
<i>Тихоненко А.В.</i>	
Системы символьной математики как технологические и математические инструменты инновационного учебного процесса.....	800
<i>Толстоуцкий В.Ю.,</i>	
<i>Кузенкова Г.В.</i>	
Программное обеспечение в обучении следователей раскрытию убийств.....	810
<i>Черняев С.И.,</i>	
<i>Семененко М.Г.,</i>	
<i>Кондратьева С.Д.</i>	
Моделирование синергетических объектов с помощью облачных сервисов системы Mathematica.....	818
<i>Юмагулов М.Г.</i>	
Компьютерное моделирование двухпараметрических бифуркаций в задачах нелинейной динамики.....	823
<b>СЕКЦИЯ 7. ШКОЛЬНОЕ ОБРАЗОВАНИЕ ПО ИНФОРМАТИКЕ .....</b>	<b>831</b>
<i>Волкова Е.П.,</i>	
<i>Павлов Е.Н.,</i>	
<i>Шуйкова И.А.</i>	
Дополнительное образование школьников, проявляющих способности в изучении информатики, в системе образования города Липецка.....	832
<i>Гладких И.Ю.</i>	
<i>Якушин А.В.</i>	



Изучение робототехники в средней школе.....	837
<i>Гринько О.Е.</i>	
Использование технологии проблемно-диалогического обучения как один из путей формирования интеллектуальной и творческой личности .....	843
<i>Крыжановская Ю.А.</i>	
Научно-исследовательская деятельность в научном обществе учащихся (секция «Информатика»).....	850
<i>Литвинов Ю.В.</i>	
Визуальные средства программирования роботов и их использование в школах .....	858
<i>Ожогина Е.М.</i>	
Возможности применения информационно-коммуникационных технологий в качестве средства формирования Смыслоразнозначных ориентаций старшеклассников.....	869
<i>Семилуцкая Л.В.</i>	
Использование проектной деятельности на уроках информатики в вечерней школе .....	874
<i>Фаттахова Г.А.</i>	
Электронный образовательный ресурс «Школа юного лесовода» как основной инструмент обучения членов школьного лесничества.....	880
<i>Явич М.П.</i>	
Теоретические основы информатики. Концепции создания электронных обучающих средств и интеллектуальных компьютерных игр для детей.....	890
<b>СЕКЦИЯ 8. ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ ПРОГРАММИРОВАНИЕ, ГРИД-ТЕХНОЛОГИИ, ПРОГРАММИРОВАНИЕ НА ГРАФИЧЕСКИХ ПРОЦЕССОРАХ .....</b>	<b>898</b>
<i>Астахова И.Ф.,</i>	
<i>Коробкин Е.А.</i>	
Технология CUDA для метода дискретных элементов в параллельной среде.....	899
<i>Давыдова А.П.</i>	
К вопросу о реализуемости процесса анализа устойчивости систем управления на параллельных платформах.....	904
<i>Лупин С.А.,</i>	
<i>Тан Шейн,</i>	
<i>Чжо Мью Хтун</i>	
Параллельная реализация алгоритмов дискретной оптимизации.....	912
<i>Посыпкин М.А.</i>	
Грид-систем из персональных компьютеров в России: текущее состояние и перспективы.....	919
<i>Сорокин А.И.</i>	
Сравнительный анализ инфраструктур для облачных вычислений.....	926
<i>Си Ту Тант Син</i>	
Параллельные и последовательные варианты метода динамического программирования .....	932
<i>Устименко О.В.</i>	
Архитектура информационной среды для доступа и использования данных, полученных при зондировании Земли из Космоса.....	939
<i>Храпов Н.П.</i>	

Применение технологии виртуализации для изучения комбинированных вычислительных инфраструктур.....	951
<i>Чжо Мью Хтун,</i>	
<i>Чжо Чжо Лин</i>	
Точность итерационного алгоритма решения задачи распределения нагрузки в системах обслуживания.....	958
<b>СЕКЦИЯ 10. ЦИФРОВЫЕ ТЕХНОЛОГИИ ОБРАБОТКИ СИГНАЛОВ И УПРАВЛЕНИЯ.....</b>	<b>963</b>
<i>Бондаренко М.А.</i>	
Программное и алгоритмическое обеспечение системы технического зрения по автоматизированной обработке изображений ближнего космоса.....	964
<i>Веремей Е.И.</i>	
Вопросы оптимизации цифровых систем управления и обработки сигналов....	974
<i>Вэй Ян Лвин, Щагин А.В.</i>	
Адаптивная система автоматического управления движением.....	983
<i>Еремеев В.В.</i>	
Некоторые аспекты структуризации сеансов спутниковой связи.....	988
<i>Жабко А.П.,</i>	
<i>Медведева И.В.</i>	
Оценка области асимптотической устойчивости решений дифференциально-разностных систем запаздывающего типа.....	995
<i>Коровкин М.В.</i>	
Реализация цифровой системы управления трубогибочным станком.....	1002
<i>Кристалинский Р.Е.,</i>	
<i>Белоусов В.В.</i>	
Опыт решения задач об отыскании оптимального управления для линейных систем с постоянными коэффициентами и приводимым к ним .....	1007
<i>Медведев М.С.</i>	
Алгоритм фонемной сегментации в модуле обучения системы распознавания речи.....	1012
<i>Смирнов М.Н.</i>	
Оптимизация управления подвижными объектами с ограниченными внешними возмущениями .....	1018
<i>Смирнова М.А.</i>	
Многоцелевое управление подвижными объектами в режиме реального времени.....	1025
<i>Сотникова М.В.</i>	
Синтез робастного цифрового регулятора для системы магнитной левитации .....	1033