

СОДЕРЖАНИЕ

Корячко В.П., Перепелкин Д.А. Построение дерева оптимальных маршрутов корпоративной сети в условии динамического изменения нагрузки на ее линиях связи	7
Арбузов В.А., Головлев П.А. Задачи и проблематика оснастки трехмерной модели четвероногого персонажа	18
Артюшин А.М., Юдаев Ю.А. Моделирование распределения электрических полей методом фиксированных узлов	20
Асташин В.А., Шкарин С.С. Алгоритмы декодирования итеративного кода.....	24
Бакулев А.В., Бакулева М.А. Алгоритм выделения параллельных регионов на основе управляющей структуры программы	27
Баранчикова Е.А. Особенности проектирования систем защиты малых информационных систем	30
Бердников В.М. Многокритериальный синтез ансамбля кодовых последовательностей с нулевой зоной автокорреляционной функции для наземной радионавигационной системы	37
Бобиков А.И., Лимонова А.С. Нечеткое управление с прямой и обратной связями нелинейными объектами первого порядка.....	41
Богонатов В.А., Скворцов С.В. Разработка хэш-функций для современных компиляторов.....	50
Бодров О.А. Формирование систем фазоманипулированных сигналов для асинхронных адресных систем передачи информации	53
Бузыкканов С.Н. Технология передачи телеметрической информации с пониженным уровнем шумов квантования	55
Варнавский А.Н. Применение нейросетевой обработки биомедицинских сигналов для выявления различных патологий.....	59

Виноградова М.Е. Процедура параметризации акустических помех для решения задач классификации	62
Горюнов И.В. Совершенствование системы управления вузом на основе сбалансированной системы показателей.....	65
Грицай Е.В., Дондик Е.М., Скоробогатова Н.Е. Система распознавания знаков дактильной речи.....	68
Дмитриев В.Т., Картавенко Я.О. Объективная оценка качества декодированного речевого сигнала на основе изменения спектральной динамики критических полос спектра.....	74
Дондик Е.М., Скоробогатова Н.Е., Хватова А.С. Система распознавания знаков дактильной речи на основе веера векторов	77
Козлов М.А., Скворцов С.В. Гибридный алгоритм сортировки массивов для многоядерных процессоров.....	81
Кондрашкин А.В., Трофимова И.П. Система электронного документооборота стационарного отделения больницы	84
Корячко Д.А., Тишкин Р.В. Графовые модели сегментации изображений.....	87
Кузьмина Е.М., Сосулин Ю.А. Оценивание параметров регрессионных моделей объектов иерархической структуры	93
Линьков В.А., Линьков П.В., Линьков Ю.В., Мусолин А.К., Пушкин В.А., Рожков О.В., Самохин А.Н., Сулов А.И. Оценка электроэффективности архитектуры трехфазных электрических сетей с нелинейными нагрузками компьютерного типа	100
Лоцманов Е.В., Елистратов Ю.П. Механизмы, формулирующие реалистичное перемещение летающих существ	104
Лоцманов Е.В., Елистратов Ю.П. Сущность и проблемы реализации технологии мускульно-скелетной анимации	106
Медведев Р.Е. Ключевые критерии выбора платформы для системы дистанционного обучения	108

Миронова К.В. Устойчивое управление линейными системами переменной структуры	110
Мирошин В.М., Перепелкин Д.А. 3D моделирование алгоритмов адаптивной маршрутизации корпоративной сети	114
Нестеров А.В. Системы компьютерного зрения в промышленных технологических процессах	117
Покровский П.С. Устойчивая процедура синтеза сигнального созвездия для создания универсальной системы передачи информации	122
Ревуцкий В.А. Алгоритм классификации помехоустойчивых кодов	126
Рудаков В.Е., Скворцов С.В. Автоматизированное модульное тестирование программ	131
Семин Д.С. Универсальный алгоритм помехозащищенного кодирования на базе нелинейного преобразования в поле Галуа	133
Скоз Е.Ю. Формализация проектных процедур архитектурного проектирования информационных систем	136
Скоз Е.Ю., Цыцаркин Ю.М. Представление разбиений производящими функциями	145
Солдатов В.В., Солдатова О.В. Использование анимационных моделей и прогнозирование природных и техногенно-опасных ситуаций на основе космических снимков	151
Телков И.А. Организация универсальных параллельных вычислений с использованием возможностей NVIDIA CUDA	156
Фокин Р.В. Особенности организации практических занятий по информатике и информационным технологиям при обучении слушателей образовательных учреждений ФСИН России	162

Хруничев Р.В. Реферирование текста с применением статистических методов обработки текстовой информации на основе частоты встречаемости слов в документе и определения их веса	165
Цветков И.А. Обращающий самомодифицируемый алгоритм в алфавите с одной дополнительной буквой.....	169
Шлыков Е.С., Миловзоров О.В. Об одном из подходов к созданию систем планирования механосборочного производства	179

В.П. КОРЯЧКО, Д.А. ПЕРЕПЕЛКИН

Рязанский государственный радиотехнический университет

ПОСТРОЕНИЕ ДЕРЕВА ОПТИМАЛЬНЫХ МАРШРУТОВ КОРПОРАТИВНОЙ СЕТИ В УСЛОВИИ ДИНАМИЧЕСКОГО ИЗМЕНЕНИЯ НАГРУЗКИ НА ЕЕ ЛИНИЯХ СВЯЗИ

Рассматривается задача построения дерева оптимальных маршрутов корпоративной сети в условиях динамического изменения нагрузки. Предложенный в работе алгоритм позволяет производить построение оптимальных маршрутов корпоративной сети с меньшей трудоемкостью по сравнению с известными алгоритмами в условиях динамического изменения нагрузки на ее линиях связи.

Необходимость обеспечения качественного обслуживания современного трафика, передаваемого через IP-сети, обуславливает высокие требования к эффективности передачи пакетов данных от отправителя к получателю. Загрузка и пропускная способность линий связи корпоративной сети динамически меняются, что в свою очередь, может приводить к частой рассылке служебной информации об изменении маршрутов [1]. Задача маршрутизации в сетях решается при условии, что кратчайший маршрут, обеспечивающий передачу пакета за минимальное время, зависит от топологии сети, пропускной способности и нагрузки на линии связи. Топология сети изменяется в результате отказов узлов и линий связи и отчасти при развитии телекоммуникационной системы (подключение новых узлов и линий связи). Пропускная способность каналов связи определяется типом передающей среды и зависит от уровня шумов и параметров аппаратуры. Наиболее динамическим фактором является нагрузка на линии связи, изменяющаяся довольно быстро и в трудно прогнозируемом направлении.

Для выбора оптимального маршрута каждый узел связи должен располагать информацией о состоянии телекоммуникационной системы в целом – всех остальных узлов и линий связи. Данные о текущей топологии сети и пропускной способности линий связи предоставляются узлам без затруднений. Однако нет способа для точного предсказания состояния нагрузки в сети. Поэтому при решении задачи маршрутизации могут использоваться данные о состоянии нагрузки, запаздывающие (из-за конечной скорости передачи информации) по отношению к моменту принятия решения о направлении передачи пакетов. Следовательно, во всех случаях алгоритмы маршрутизации выполняются в условиях неопределенности текущего и будущего состояния телекоммуникационной системы.

В современных корпоративных сетях для продвижения пакетов по сети используют протоколы, выполняющие статическую и адаптивную (динамическую) маршрутизацию.

При статической маршрутизации все записи в таблице имеют неизменяемый, статический статус, что подразумевает бесконечный срок их жизни. Записи о маршрутах составляются и вводятся в память каждого маршрутизатора вручную администратором сети. При изменении состояния сети администратору необходимо срочно отразить эти изменения в соответствующих таблицах маршрутизации, иначе может произойти их рассогласование, и сеть будет работать некорректно.

При адаптивной маршрутизации все изменения конфигурации сети автоматически отражаются в таблицах маршрутизации благодаря протоколам маршрутизации. Эти протоколы собирают информацию о топологии связей в сети, что позволяет им оперативно обрабатывать все текущие изменения.

Анализ современных алгоритмов адаптивной маршрутизации в корпоративных сетях показывает, что дистанционно-векторные алгоритмы используют в своей работе алгоритм Беллмана – Форда, трудоемкость которого составляет $O(N^3)$, где N – число маршрутизаторов в корпоративной сети, а алгоритмы состояния связей базируются на алгоритме Дейкстры вычислительной трудоемкостью $O(N^2)$.

При изменении пропускной способности линий связи в представленных алгоритмах происходит полный перерасчет таблиц маршрутизации. С увеличением размера корпоративной сети трудоемкость этой операции растет полиномиально, что не может не сказаться на производительности всей сети в целом. Точное определение множества оптимальных маршрутов позволяет уменьшить трудоемкость построения таблиц маршрутизации.

Представим корпоративную сеть в виде неориентированного взвешенного связного графа $G = (V, E, W)$ показанного на рис. 1, где V – множество вершин, $|V| = N$, E – множество ребер, $|E| = M$, W – множество весов ребер.

Пусть на графе G в некоторый момент времени уже решена задача поиска кратчайших путей до всех вершин множества $V_s = V \setminus v_s$ из начальной вершины v_s , т. е. построено дерево кратчайших путей с корнем в вершине v_s . Обозначим это дерево как T_g . На рис. 1 жирными линиями обозначено построенное дерево кратчайших путей. Рассмотрим множество ребер E графа G . По признаку вхождения ребер в дерево T_g можно разделить исходное множество E на два подмножества: $E_l \in T_g$ и $E_r \notin T_g$, $E_l \cup E_r = E$.

Множество ребер дерева E_T – множество ребер дерева T_g для графа G . Для заданного графа G , согласно свойству дерева, мощность множества E_T будет равняться мощности множества V минус единица $|E_T| = |V| - 1$.

Множество ребер замены для дерева E_R – множество ребер графа G , не вошедших в дерево T_g . При соответствующих условиях некоторое ребро $e_{i,j} \in E_R$, инцидентное вершинам v_i и v_j может перейти во множество ребер дерева E_T , заменив собой некоторое ребро $e_{k,p} \in E_T$. При этом инцидентность ребра $e_{k,p}$ вершине v_i или v_j является обязательным условием. В свою очередь ребро $e_{i,j}$ перейдет во множество E_R .

Будем называть такие переходы **парными переходами** и обозначать $e_{i,j} - e_{k,p}$.

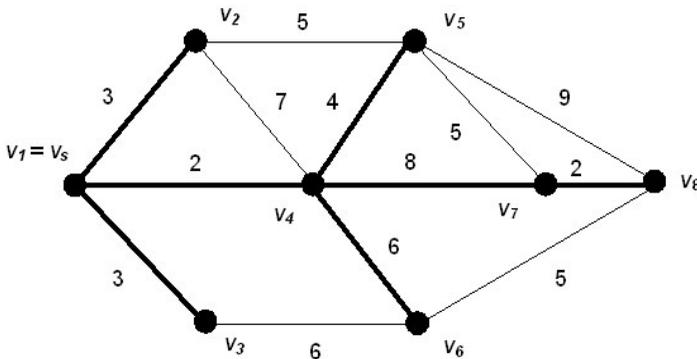


Рис. 1. Граф G корпоративной сети

Парные переходы возможны в двух случаях: при уменьшении веса ребра $e_{k,p}$ до некоторого порогового значения и при увеличении веса ребра $e_{i,j} \in E_T$. При этом возможен такой случай, при котором изменение веса ребра $e_{i,j} \in E_T$ повлечет за собой исключение некоторого ребра $e_{o,u} \in E_T$ и включение в это множество ребра $e_{k,p} \in E_R$. Изменение веса ребра $e_{k,p} \in E$ может привести к одному или нескольким парным переходам, но может и не привести к перестановкам, вследствие чего конфигурация дерева кратчайших путей не изменится. Количество парных переходов зависит от величины изменения и положения ребра в дереве кратчайших путей (для ребер $e_{i,j} \in E_T$). Любое ребро $e_{i,j} \in E$ может попасть в дерево T_g только посредством срабатывания парного перехода.

Маршрутная степень вершины $ms(v_i)$ – число неповторяющихся ребер $e_{ij} \in E$, инцидентных вершине v_i , через каждое из которых можно построить простой путь между вершинами v_i и v_s .

Теорема 1. Для любого ребра $e_{ij} \in E_T$, инцидентного некоторым вершинам v_i и v_j , маршрутные степени которых больше единицы, при заданной конфигурации графа, неизменных весах других ребер существует такое значение веса w_{ij}^t , что при $w_{ij} > w_{ij}^t$ ребро e_{ij} становится ребром замены и переходит во множество E_R .

Доказательство. Пусть $\pi_{i,u}$ – текущий кратчайший путь и оценка кратчайшего пути $d_i > d_j$. Для вершины v_i , степень которой больше единицы, следовательно, $|P_i| > 1$, среди $\pi_{i,k} \in P_i$ найдется такой кратчайший путь $\pi_{i,p}$, для которого $e_{ij} \notin \pi_{i,p}$. Длина $\pi_{i,p}$ -пути составляет $d_{i,p}$. Тогда при увеличении веса ребра w_{ij} на величину, большую $d_{i,p} - d_{i,u}$ кратчайший путь до вершины v_i изменится на путь $\pi_{i,p}$. Т.е. при $w_{ij} > w_{ij} + (d_{i,p} - d_{i,u})$ ребро e_{ij} становится ребром замены и переходит во множество E_r . В данном случае $w_{ij}^t = w_{ij} + (d_{i,p} - d_{i,u})$. Теорема доказана.

Величину w_{ij}^t будем называть **точкой вхождения в дерево** для ребра e_{ij} .

Отношение парного перехода r_i – отношение соответствия элемента e_{ij} множества E_T элементу $e_{k,p}$ множества E_R , такое, что при увеличении веса ребра e_{ij} так, что $w_{ij} > w_{ij}^t$ имеет место парный переход $e_{ij} - e_{k,p}$.

Теорема 2. Для любого ребра $e_{k,p} \in E_R$, находящегося в отношении парного перехода с некоторым ребром $e_{ij} \in E_T$ с заданной точкой вхождения в дерево w_{ij}^t существует такое значение веса $w_{k,p}^t$, что при $w_{k,p} < w_{k,p}^t$ ребро $e_{k,p}$ становится веткой дерева T_g и переходит во множество E_T .

Доказательство. Пусть $\pi_{i,u}$ – текущий кратчайший путь, содержащий ребро e_{ij} . Пусть $\pi_{i,p}$ кратчайший путь, содержащий $e_{k,p}$. Из доказательства, приведенного выше, следует, что $d_{i,u} + (w_{ij}^t - w_{ij}) = d_{i,p}$. Т.е. $d_{i,u} = d_{i,p} - (w_{ij}^t - w_{ij})$. При $d_{i,p} < d_{i,u}$ ребро $e_{k,p}$ станет веткой дерева T_g и перейдет во множество E_T . В данном случае $w_{k,p}^t = w_{k,p} - (w_{ij}^t - w_{ij})$. Теорема доказана.

Теорема 3. Для элементов парного отношения r_i $e_{ij} \in E_T$ и $e_{k,p} \in E_r$ при известной точке вхождения в дерево w_{ij}^t и $w_{k,p}^t$ справедливо, что $w_{ij}^t - w_{ij} = w_{k,p} - w_{k,p}^t$.

Доказательство. Пусть ребра e_{ij} и $e_{k,p}$ инцидентны некоторой вершине v_i , т.е. $k = i$. Пусть кратчайший путь между v_s и v_i , содержащий e_{ij} имеет длину $d_{i,k}$. Известно, что при увеличении веса ребра e_{ij} до значения, превышающего w_{ij}^t , кратчайший путь до v_i изменится таким образом, что будет включать в себя ребро $e_{k,p}$ и иметь длину,

равную $d_{i,l}$. Т.е. $d_{i,l} - d_{i,k} = w_{i,j}^t - w_{i,j}$. С другой стороны, при уменьшении ребра $e_{k,p}$ до значения, меньшего $w_{k,p}^t$, кратчайший путь до v_i изменится таким образом, что будет включать в себя ребро $e_{k,p}$ и будет иметь длину, равную $d_{i,l}$. Т.е. $d_{i,l} - d_{i,k} = w_{k,p} - w_{k,p}^t$. Таким образом, $w_{i,j}^t - w_{i,j} = w_{k,p} - w_{k,p}^t$. Теорема доказана.

Следствие 1. Если для элементов парного отношения r_i , $e_{i,j} \in E_T$ и $e_{k,p} \in E_R$, соответствующих весов $w_{i,j}$ и $w_{k,p}$ при известной точке вхождения в дерево $w_{k,p}^t$ изменился вес ребра $e_{i,j}$ до значения w_1 , то $w_{k,p}^t = w_{k,p}^t + (w_1 - w_{i,j})$.

Следствие 2. Если для элементов парного отношения r_i , $e_{i,j} \in E_T$ и $e_{k,p} \in E_R$, соответствующих весов $w_{i,j}$ и $w_{k,p}$ при известной точке вхождения в дерево $w_{k,p}^t$ изменился вес ребра $e_{k,p}$ до значения w_2 , то $w_{i,j}^t = w_{i,j}^t + (w_2 - w_{k,p})$.

Теорема 4. Ребра $e_{i,j} \in E_T$ и $e_{k,p} \in E_R$, находящиеся в одном отношении $r_i \in R$, инцидентны одной и той же вершине v_i при условии, что v_i является листом дерева кратчайших путей.

Доказательство. После превышения веса ребра $e_{i,j}$ точки вхождения в дерево, оно будет исключено из дерева и вершина v_i окажется не связанной со всеми остальными вершинами в дереве. Чтобы в результате получилось дерево T_g с исходным множеством вершин, необходимо в дерево добавить ребро, связывающее вершину v_i с некоторой вершиной v_p в дереве. Таким ребром по определению парного отношения является $e_{k,p} \in E_R$. Это ребро будет инцидентно вершине v_i . Теорема доказана.

Отношение парного перехода, в общем случае, может быть не задано. Действительно, кратчайшие пути до вершины маршрутной степени 1 на заключительном шаге будут содержать одно и то же ребро вне зависимости от веса этого ребра. Т.е. для некоторой вершины v_i с маршрутной степенью, равной 1 и инцидентного ей ребра $e_{i,j}$, через которое проходит путь между вершинами v_i и v_s , точка вхождения в дерево для ребра $e_{i,j}$ будет равна бесконечности. Иными словами, изменение веса ребра $e_{i,j}$ в данных условиях приведет к изменению оценки длины пути до вершин множества $V_T^{(i)}$. Конфигурация дерева T_g при этом останется неизменной.

Множество кратчайших путей до вершин графа G из исходной вершины v_s π_i ($i=1, 2, \dots, N$) определяет множество оценок кратчайших путей до этих вершин d_i ($i=1, 2, \dots, N$).

Рассмотрим следующий случай: пусть изменению подвергся вес некоторого ребра $e_{i,j}$, инцидентного вершинам v_i и v_j . Допустим, что v_j находится ниже по иерархии в дереве T_g относительно вершины v_i .

Тогда изменению подвергнутся оценки длин кратчайших путей π_k всех вершин v_k , входящих в V_j -дерево $V_T^{(j)}$.

Рассмотрим некоторую вершину v_i , которая является листом дерева кратчайших путей T_g . Для данной вершины определен кратчайший путь $\pi_{i,k}$ и оценка длины кратчайшего пути $d_{i,k}$. Пусть ребро $e_{i,j} \in E_T$, инцидентное вершине v_i , является заключительным ребром в пути $\pi_{i,k}$ и входит в отношении парного перехода r_i с ребром $e_{k,p} \in E_R$. Ребро $e_{k,p}$, по теореме 4, инцидентно вершине v_i . Пусть $e_{k,p}$ инцидентно также вершине v_p , $\pi_{j,p}$ – кратчайший путь до v_p . Тогда при условиях наступления парного перехода $e_{i,j} - e_{k,p}$ оценка длины кратчайшего пути до вершины v_i станет равным $d_{j,p} + w_{k,p}$. Кратчайший путь до v_i $\pi_{i,r}$ будет состоять из пути $\pi_{j,p}$ и ребра $e_{k,p}$. Таким образом, множество парных переходов R определяет множество кратчайших путей и оценок длин для вершин графа G в условиях невозможности использовать текущие кратчайшие пути. Т.е. для вершины v_i существует путь $\pi_{i,r}$ такой, что $d_{i,r} = d_{i,k} + (w_{i,j}^t - w_{i,j})$. Т.е. для заданного графа при увеличении веса ребра $e_{i,j}$, входящего в дерево кратчайших путей T_g оценка длины кратчайшего пути до вершины v_i не может быть больше, чем $d_{i,r}$.

Среди элементов множества E_R можно выделить такие элементы-ребра, которые для некоторого заданного состояния графа G не могут попасть во множество E_T при изменениях весов ребер $e_{i,j} \in E_T$. Для таких ребер $e_{k,p} \in E_R$ вхождение в дерево T_g возможно только при уменьшении их собственного веса $w_{k,p}$ при условии единичного изменения.

Таким образом, во множестве E_R можно выделить 2 подмножества.

Множество ребер замены E_S для дерева – это такое подмножество множества E_R , элементы-ребра которого участвуют, по крайней мере, в одном отношении парного перехода.

Множество непарных ребер E_P – это такое подмножество множества E_R , элементы-ребра которого не участвуют ни в одном отношении из множества R .

В общем случае множество E_P может быть пустым $|E_P| = 0$. Множество E_S будет пустым только при условии, что исходный связный граф G является деревом, и задача поиска кратчайших путей в этом случае лишена смысла.

Теорема 5. Для любого ребра $e_{i,j} \in E_S$, инцидентного некоторым вершинам v_i и v_j , маршрутные степени которых больше двух, при заданной конфигурации графа, неизменных весах других ребер существует такое значение веса $w_{i,j}^s$, что при $w_{i,j} > w_{i,j}^s$ ребро $e_{i,j}$ становится непарным ребром и переходит во множество E_P .

Доказательство. Пусть $\pi_{i,u}$ – текущий кратчайший путь, $d_i > d_j$ и $e_{i,j}$ – последнее ребро пути $\pi_{i,u}$. Для вершины v_i , маршрутная степень которой больше двух, следовательно, $|P_i| > 2$, среди $\pi_{i,k} \in P_i$ найдется такой кратчайший путь $\pi_{i,p}$, для которого $(e_{k,p} \notin \pi_{i,p}) \cap (e_{i,j} \notin \pi_{i,p})$. Пусть ребро $e_{o,u}$ является последним ребром пути $\pi_{i,p}$. Длина $\pi_{i,p}$ -пути составляет $d_{i,p}$. Тогда при увеличении веса ребра $w_{k,p}$ на величину, большую $d_{i,p} - d_{i,u}$ ребро $e_{o,u}$ заменит ребро $e_{k,p}$ в парном переходе $e_{i,j} - e_{k,p}$. Новый переход будет $e_{i,j} - e_{o,u}$. Т.е. при $w_{k,p} > w_{k,p} + (d_{i,p} - d_{i,u})$ ребро $e_{k,p}$ становится непарным ребром и переходит во множество E_p . В данном случае $w_{k,p}^s = w_{k,p} + (d_{i,p} - d_{i,u})$. Теорема доказана.

Величину $w_{k,p}^s$ будем называть **точкой вхождения во множество замены** для ребра $e_{k,p}$.

Если для ребра $e_{i,j} \in E$ на шкале значений весов определена точка вхождения в дерево $w_{i,j}^t$ и точка вхождения во множество замены $w_{i,j}^s$, то $w_{i,j}^t \leq w_{i,j}^s$. На рис. 2 представлено положение точек на шкале значений весов ребра.

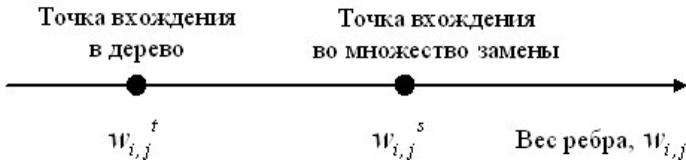


Рис. 2. Расположение значений $w_{i,j}^t$ и $w_{i,j}^s$ на шкале значений весов ребра $e_{i,j}$

Теорема 6. Для любого ребра $e_{i,j} \in E_S$, инцидентного некоторым вершинам v_i и v_j , маршрутные степени которых больше двух, при заданной конфигурации графа, неизменных весах других ребер существует такое ребро $e_{k,p} \in E_p$ и такое значение его веса $w_{k,p}^s$, что при $w_{k,p} < w_{k,p}^s$ ребро $e_{k,p}$ становится ребром замены и переходит во множество E_S .

Доказательство. Пусть $\pi_{i,u}$ – текущий кратчайший путь, $d_i > d_j$ и $e_{o,u}$ – последнее ребро пути $\pi_{i,u}$. В силу доказанного в теоремах 1 и 2, имеем, что существует некое отношение парного перехода $r_i \in R$, для ребер $e_{o,u}$, $e_{i,j}$. Пусть $\pi_{i,s}$ – кратчайший путь, содержащий ребро $e_{i,j}$, и $\pi_{i,p}$ – кратчайший путь, содержащий ребро $e_{k,p}$. Из доказательства теоремы 1 следует, что $d_{i,s} + (w_{i,j}^s - w_{i,j}) = d_{i,p}$. Т.е. $d_{i,s} = d_{i,p} - (w_{i,j}^s - w_{i,j})$. При $d_{i,p} < d_{i,s}$ ребро $e_{k,p}$ станет ребром замены и перейдет во множество E_S . Теорема доказана.

Отношение парного перехода множества замены rs_i – отношение соответствия элемента e_{ij} множества E_S элементу $e_{k,p}$ множества E_P , такое, что при увеличении веса ребра e_{ij} так, что $w_{ij} > w_{ij}^s$ имеет место парный переход $(e_{ij} - e_{k,p})^s$, при котором ребро e_{ij} исключается из множества E_S и включается в E_P , а ребро $e_{k,p}$ исключается из множества E_P и включается во множество E_S . При этом парный переход $r_i e_{o,u} - e_{ij}$ заменяется переходом $e_{o,u} - e_{k,p}$.

Теорема 7. Для элементов отношения парного перехода для множества замены $r_i e_{ij} \in E_S$ и $e_{k,p} \in E_P$ при известной точке вхождения во множество замены w_{ij}^s и $w_{k,p}^s$ справедливо, что $w_{ij}^s - w_{ij} = w_{k,p} - w_{k,p}^s$.

Доказательство. Пусть ребра e_{ij} и $e_{k,p}$ инцидентны некоторой вершине v_i . Пусть кратчайший путь между v_s и v_i , содержащий e_{ij} имеет длину $d_{i,k}$. Известно, что при увеличении веса ребра e_{ij} до значения w_{ij}^s , ребро e_{ij} станет непарным ребром и длина пути $\pi_{i,k}$ будет равна $d1$. Т.е. $d1 - d_{i,k} = w_{ij}^s - w_{ij}$. С другой стороны, при уменьшении ребра $e_{k,p}$ до значения $w_{k,p}^s$, ребро e_{ij} станет непарным ребром и длина пути $\pi_{i,k}$ будет равна $d2$. Т.е. $d2 - d_{i,k} = w_{k,p} - w_{k,p}^s$. В силу того, что в момент, когда ребра находятся в точках вхождения во множество замены, оба пути до рассматриваемой вершины, которые проходят соответственно через e_{ij} и $e_{k,p}$ будут равны, следовательно, $d1 = d2$. Таким образом, $w_{ij}^s - w_{ij} = w_{k,p} - w_{k,p}^s$. Теорема доказана.

Следствие 1. Если для элементов парного отношения для множества замены rs_i , $e_{ij} \in E_S$ и $e_{k,p} \in E_P$, соответствующих весов w_{ij} и $w_{k,p}$ при известной точке вхождения во множество замены $w_{k,p}^s$ изменился вес ребра e_{ij} до значения w_1 , то $w_{k,p}^s = w_{k,p}^s + (w_1 - w_{ij})$.

Следствие 2. Если для элементов парного отношения для множества замены rs_i , $e_{ij} \in E_S$ и $e_{k,p} \in E_P$, соответствующих весов w_{ij} и $w_{k,p}$ при известной точке вхождения во множество замены $w_{k,p}^s$ изменился вес ребра $e_{k,p}$ до значения w_2 , то $w_{ij}^s = w_{ij}^s + (w_2 - w_{k,p})$.

Обозначим w_{ij} – вес ребра, соединяющего вершины v_i и v_j ; nw_{ij} – новое значение веса, полученное в результате динамического увеличения нагрузки на линии связи корпоративной сети. Вершина v_i располагается ниже по иерархии дерева кратчайших путей относительно v_j .

Обозначим V_T – множество вершин, до которых найден кратчайший путь из начальной вершины, V_R – множество остальных вершин. $V_R \cup V_T = V$, $V_R \cap V_T = \emptyset$.

Для графа G корпоративной сети, представленного на рис. 1 множество ребер дерева $E_T = \{e_{1,2}, e_{1,3}, e_{1,4}, e_{4,5}, e_{4,6}, e_{4,7}, e_{7,8}\}$; множество ребер замены $E_S = \{e_{2,5}, e_{2,4}, e_{3,6}, e_{5,7}, e_{6,8}\}$; множество непарных ребер будет $E_P = \{e_{3,8}\}$.

При увеличении веса ребра $e_{ij} \in E$ возможны два варианта:

1. $e_{ij} \in E_T$, $nw_{ij} > w_{ij}$;
2. $e_{ij} \in E_R$, $nw_{ij} > w_{ij}$.

Рассмотрим первый вариант, то есть увеличение веса ребра $e_{ij} \in E_T$. Например, если рассмотреть ребро $e_{4,5}$, то для него точка вхождения в дерево будет составлять 6, а точка вхождения во множество замены – 13. При этом данное ребро находится в отношении парной перестановки с ребром $e_{2,5}$, а после попадания $e_{4,5}$ во множество непарных ребер, эта парная перестановка примет вид $e_{2,5} - e_{5,7}$. Пусть увеличился вес ребра $e_{4,5}$, например, $nw_{4,5} = 13$. Тогда граф G корпоративной сети примет вид, показанный на рис. 3.

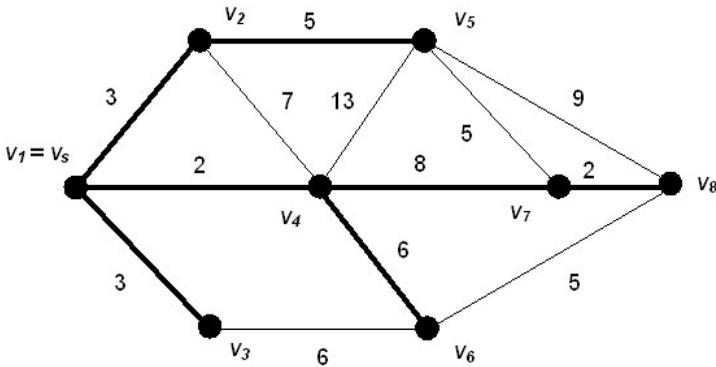


Рис. 3. Увеличение веса ребра $e_{4,5}$

Рассмотрим второй вариант, то есть увеличения веса ребра $e_{ij} \in E_R$. Пусть увеличился вес ребра $e_{2,4}$, например $nw_{2,4} = 12$. После увеличения веса ребра $e_{2,4}$ дерево кратчайших путей не изменится, граф G корпоративной сети будет иметь такой же вид, как и на рис. 1.

При уменьшении веса ребра $e_{ij} \in E$ также возможны два варианта:

1. $e_{ij} \in E_T$, $nw_{ij} < w_{ij}$;
2. $e_{ij} \in E_R$, $nw_{ij} < w_{ij}$.

Рассмотрим первый вариант, то есть уменьшение веса ребра $e_{ij} \in E_T$. Например, если рассмотреть ребро $e_{4,5}$, то для него точка вхождения в дерево будет составлять 6, а точка вхождения во множество замены – 13. Пусть уменьшился вес ребра $e_{4,5}$, например, $nw_{4,5} = 3$. Уменьшение веса ребра, входящего в кратчайший путь приведет только к уменьшению длины кратчайшего пути, поэтому дерево кратчайших путей не изменится.

Рассмотрим второй вариант, то есть уменьшение веса ребра $e_{ij} \in E_R$. Пусть уменьшился вес ребра $e_{2,5}$, например $nw_{2,5} = 2$. Так как

ребро $e_{2,5}$ находится в отношении парной перестановки с ребром $e_{4,5}$ и потенциал вершины V_5 изменился, то произойдет парный переход $e_{4,5} - e_{2,5}$. Тогда граф G корпоративной сети примет вид, показанный на рис. 4.

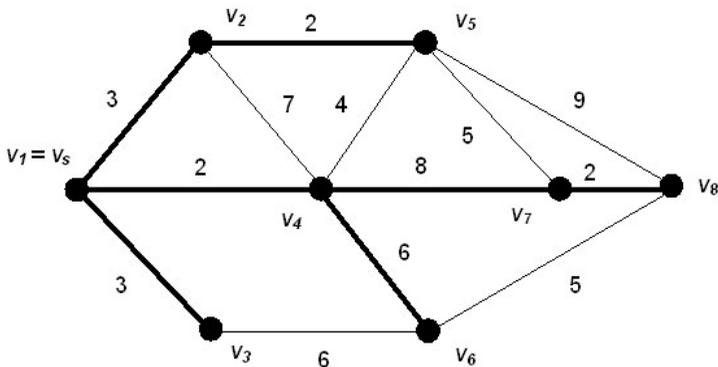


Рис. 4. Уменьшение веса ребра $e_{2,5}$

Доказанные выше теоремы и сформулированные следствия позволяют разработать алгоритм, вычисляющий оптимальные маршруты до вершин графа без полного повторного построения дерева оптимальных маршрутов при динамическом изменении нагрузки. Определение точек вхождения в дерево и точек вхождения во множество замены для всех ребер графа G позволяет определить те оптимальные маршруты, которые изменятся или изменят оценки своих длин. Это позволило значительно уменьшить трудоемкость построения дерева оптимальных маршрутов корпоративной сети.

Рассмотрим работу алгоритма адаптивной маршрутизации при динамическом изменении нагрузки на линиях связи корпоративной сети. Укрупненная схема алгоритма имеет следующий вид:

Шаг 1. Первоначальная инициализация. Определить веса линий связи $w_{i,j}$ между вершинами и сформировать множество ребер графа сети.

Шаг 2. Используя алгоритм Дейкстры, определить маршруты графа сети и построить дерево кратчайших путей.

Шаг 3. Исходное множество ребер графа разбить на подмножество ребер, которые входят в дерево кратчайших путей и подмножество ребер замены, которые не входят в дерево кратчайших путей.

Шаг 4. Для каждого ребра графа определить точки вхождения в дерево кратчайших путей и во множество ребер замены.

Шаг 5. Для вершины, являющейся листом дерева, произвести поиск всех парных переходов без ограничений. Эти списки для удобства дальнейшей работы привязываются к вершине, инцидентной рассматриваемому ребру и расположенной ниже по иерархии.

Шаг 6. Если вершина не является листом дерева, то вычислить парные переходы для этой вершины и выбрать лучшие значения потенциалов парных переходов для потомков вершины и собственных парных переходов. Подобная процедура выполняется для формирования списков парных переходов в случае динамического изменения веса ребра.

Шаг 7. Для каждой вершины формируется полный список парных переходов. Число элементов в каждом из этих списков не превышает количества вершин графа. Такое решение позволяет отказаться от предварительной сортировки потенциалов или приращений для парных переходов без значительного усложнения алгоритма обработки изменения.

Шаг 8. Анализируя полученную используемым протоколом маршрутизации информацию, определить, произошло ли изменение веса какого-либо ребра:

- а) если да, то перейти к шагу 9;
- б) иначе - к шагу 8.

Шаг 9. Используя список парных переходов, определить, требуется ли сделать парный переход:

- а) если да, то перейти к шагу 10;
- б) иначе – к шагу 14.

Шаг 10. Для вершины, у которой потенциал уменьшился и у которой в список парных переходов входит ребро с изменившейся метрикой, определить путь минимальной длины и поместить ребро, которое привело к уменьшению потенциала вершины в дерево кратчайших путей, а сменившееся ребро из дерева кратчайших путей – во множество ребер замены.

Шаг 11. Определить, уменьшился ли потенциал других вершин графа корпоративной сети, расположенных выше по иерархии, после выполнения парного перехода:

- а) если да, то перейти к шагу 12;
- б) иначе – к шагу 13.

Шаг 12. Для каждой вершины, у которой потенциал уменьшился определить путь минимальной длины и просмотреть список парных переходов. Если новый минимальный путь для каждой вершины содержит ребро из списка парных переходов, то поместить данное ребро

в дерево кратчайших путей, а ребро из дерева кратчайших путей – во множество ребер замены.

Шаг 13. Построить новое дерево кратчайших путей графа сети.

Шаг 14. а) передать пакеты по доступным эквивалентным маршрутам;

б) установить флаг передачи.

Шаг 15. Пересчитать точки вхождения в дерево и во множество замены, переформировать список парных переходов для каждой измененной вершины.

Шаг 16. Перейти к шагу 8.

Разработанный алгоритм позволяет за счет сбора дополнительной информации учесть возможность динамического изменения нагрузки на линиях связи корпоративной сети и не производить полный пересчет маршрутных таблиц. Применение алгоритма позволило снизить трудоемкость расчета таблиц маршрутизации до величины порядка $O(k \cdot N)$, где k – число фактически выполненных парных переходов. Таким образом, разработанный алгоритм является эффективным при построении дерева оптимальных маршрутов в корпоративной сети в условиях динамического изменения нагрузки на ее линиях связи.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Куракин Д.В. Маршрутизаторы для глобальных телекоммуникационных сетей и реализуемые в них алгоритмы // Информационные технологии. 1996. №2.

А.В. АРБУЗОВ, П.А. ГОЛОВЛЕВ

Рязанский государственный радиотехнический университет

ЗАДАЧИ И ПРОБЛЕМАТИКА ОСНАСТКИ ТРЕХМЕРНОЙ МОДЕЛИ ЧЕТВЕРОНОГОГО ПЕРСОНАЖА

Рассматриваются техническая сторона подготовки к анимации и проблематика скелетной трехмерной анимации четвероногих персонажей. Область применения создание анимационных роликов, мультфильмов, а так же создание визуального ряда для кинофильмов.

Постановка задачи. Определить аспекты, влияющие на реалистичность трехмерной анимации четвероногих персонажей, на этапах создания оснастки модели, в пакете трехмерного моделирования Autodesk Maya.

Анимация трехмерных персонажей это высшая ступень анимации, с тех пор как она прошла эволюцию от кукольной до компьютерной [1]. Задача ее заключается в создании и управлении трехмерным персонажем.

Для того чтобы решить поставленную задачу необходимо подойти к ней с нескольких сторон.

Создать корректную с технической, художественной точки зрения трехмерную модель персонажа.

Используемый программный пакет заставляет двигаться трехмерную модель персонажа при помощи скелета. А дальнейшая реалистичность движений зависит не только от навыков аниматора, но и от того, грамотно ли построен скелет с точки зрения анатомии и учета требований программного пакета. Важно понимать, что анатомически верно выстроенные суставы способны качественно повлиять на дальнейшую анимацию, в то время как некорректно выставленные суставы, способны её испортить. Имеется в виду – расположение суставов относительно геометрической сетки, ориентация локальных поворотных осей суставов, требования к сетке модели в местах постановки суставов.

Привязка трехмерной модели к созданному скелету – важный этап работы над настройкой персонажа. Здесь определяется качество присоединения модели, и количество времени, которое уйдет на исправление некорректных деформаций трехмерной сетки, особенно заметных при анимации примыкающих объемов - это места присоединения ног к телу, шеи к телу и т.д.

Создание структур, упрощающих анимацию. Эти добавочные структуры позволяют выполнять манипуляции с несколькими суставами, входящими в ее цепь, осуществляя управление одним маркером. Правильная постановка таких структур упростит анимацию. На этом этапе так же важно осуществить контроль над каждой такой цепочкой, а именно за поведением геометрической сетки модели, в соответствии с анатомией.

Создание элементов управления, назначение каждого из которых будет интуитивно понятно аниматору и обеспечит легкость манипулирования управляющими структурами, а соответственно и трехмерной моделью при анимации. Для четвероногих персонажей этот этап очень важен, поскольку анимировать четвероногий персонаж сложнее, чем двуногий [2]. На этом этапе следует заметить возможность автоматизации некоторых движений, посредством написания выражений (expression) для отдельных элементов оснастки (и не только), например, для элементов управления хвостом и ушами.

Корректная настройка персонажа способна обеспечить реалистичную анимацию. Однако реалистичность анимации определяется не только грамотным скелетным каркасом, его настройкой и опытом аниматора, но и мышцами. Поскольку мышцы осуществляют деформации малозаметные глазу, но придающие движению реализм.

Имитация мышц осуществляется дополнительной структурой скелета [3], создаваемой специально для этой цели. Так же симуляция мышц осуществляется трехмерными объектами, помещаемыми в области персонажа, где необходимо показать воздействие мышц. Однако в результате этих методов усложняется настройка персонажа. В последних версиях пакета Autodesk Maya мышцы имитируются специальным модулем, где являются реально существующими объектами, а не имитацией.

Заключение. Решение поставленных задач из проведенного обзора, способно обеспечить оснастку трехмерной модели четвероногого персонажа, которая привнесет реализм и упростит создаваемую анимацию.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. М. Адамс, Э. Миллер, М. Симс. Maya 5 для профессионалов. Спб.: изд-во «Питер», 2004.
2. С. Цыпчин. Понимая Maya. М.: издательство ООО "Арт Хаус медиа", 2007.
3. К. Мараффи. Создание персонажей в Maya. Моделирование и анимация. Издательство: Вильямс (2004 г.)

А.М. АРТЮШИН, Ю.А. ЮДАЕВ

Рязанский государственный радиотехнический университет

МОДЕЛИРОВАНИЕ РАСПРЕДЕЛЕНИЯ ЭЛЕКТРИЧЕСКИХ ПОЛЕЙ МЕТОДОМ ФИКСИРОВАННЫХ УЗЛОВ

В данной статье рассматривается моделирование электрических полей методом фиксированных узлов. Приводятся графики зависимостей погрешности расчета от числа итераций при различных условиях.

Моделирование распределения электрических полей основано на решении уравнения Пуассона при заданных граничных условиях:

$$\Delta U = f. \quad (1)$$

Найти аналитическое решение уравнения (1) удастся лишь в частных случаях. Более общим методом расчета электрического поля

является метод конечных разностей, который основан на дискретизации уравнения Пуассона. В рамках рассматриваемых задач предполагалось, что диэлектрическая проницаемость ε различных материалов не является тензорной величиной. В этом случае уравнение (1) в цилиндрической системе координат с учетом объемного заряда $\rho(r, \varphi, z)$ принимает вид:

$$\frac{1}{r} \frac{\partial U}{\partial r} + \frac{\partial^2 U}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 U}{\partial \varphi^2} + \frac{\partial^2 U}{\partial z^2} = - \frac{\rho(r, \varphi, z)}{\varepsilon \varepsilon_0}. \quad (2)$$

Граничные условия, в зависимости от типа прикладных задач, могут определяться задачами Дирихле, Неймана или их комбинацией:

$$U|_{(r,z,\varphi) \in \Gamma} = \psi(r, z, \varphi), \quad \left. \frac{\partial U}{\partial \mathbf{n}} \right|_{(r,z,\varphi) \in \Gamma} = \psi(r, z, \varphi), \\ \left(\alpha_1 U + \alpha_2 \frac{\partial U}{\partial \mathbf{n}} \right) \Big|_{(r,z,\varphi) \in \Gamma} = \psi(r, z, \varphi). \quad (3)$$

Для аппроксимации частных производных во внутренних узлах неравномерной сетки (r_i, z_j) , $i = 1, 2, \dots, n-1$, $j = 1, 2, \dots, m-1$ с шагами hr_1, hr_2, hz_1, hz_2 использовались разностные выражения [1]:

$$\frac{\partial U}{\partial r} = \frac{1}{hr_2 + hr_1} \left(hr_1 \frac{U_{i+1,j} - U_{i,j}}{hr_2} + hr_2 \frac{U_{i,j} - U_{i-1,j}}{hr_1} \right), \\ \frac{\partial^2 U}{\partial r^2} = \frac{2}{hr_2 + hr_1} \left(\frac{U_{i+1,j} - U_{i,j}}{hr_2} + \frac{U_{i-1,j} - U_{i,j}}{hr_1} \right), \\ \frac{\partial^2 U}{\partial z^2} = \frac{2}{hz_2 + hz_1} \left(\frac{U_{i,j+1} - U_{i,j}}{hz_2} + \frac{U_{i,j-1} - U_{i,j}}{hz_1} \right). \quad (4)$$

Расчет распределения электрического поля методом конечных разностей является достаточно эффективным методом, но для увеличения точности результатов моделирования необходимо, например, уменьшать пространственные шаги. Это приводит к увеличению узлов в расчетной области Ω и количеству итераций при решении системы линейных уравнений. Ускорить вычислительный процесс и одновременно увеличить точность расчета удалось при использовании разработанного метода фиксированных узлов, суть которого состоит в следующем.

При расчете электрического поля с некоторой заданной погрешностью ε_{ps} на первом этапе моделирования в расчетной области Ω строится сетка с большим шагом разбиения, чем шаг, необходимый для расчета с заданной погрешностью. При этом на полученной сетке погрешность вычислений ε_{ps_i} больше заданной и итерационный про-

цесс прекращается при достижении этой промежуточной погрешности eps_1 .

На втором этапе моделирования расчетная область Ω покрывается более мелкой сеткой, которая позволяет осуществлять расчет с погрешностью eps . Мелкая сетка строится таким образом, чтобы в часть ее узлов попали все узлы предыдущей сетки. Значение потенциала в этих узлах принимаются равными значению потенциала рассчитанного ранее. Каждый такой узел отмечается, а значение потенциала в нем фиксируется.

Затем проводится расчет системы линейных уравнений на более мелкой сетке во всех ее узлах за исключением ранее отмеченных, которые выступают в качестве условий Дирихле. Расчет прекращается при достижении заданной погрешности eps , после чего снимаются ограничения на значения потенциала в отмеченных фиксированных узлах. На последнем этапе расчет проводится во всех узлах сетки, включая ранее отмеченные, до достижения заданной точности.

К достоинству созданного метода расчета можно отнести возможность получения результатов с заданной точностью при сокращении количества итераций и без увеличения размерности вычислительного шаблона. При моделировании распределения электрического возможен ручной ввод фиксированных узлов в характерных областях (обычно области с максимальной напряженностью поля E), который позволяет многократно ускорить вычислительный процесс [2].

Для предложенного метода фиксированных узлов было проведено тестирование алгоритма расчета и программного обеспечения. Расчетная область представляла собой цилиндр с высотой и радиусом равными 1 см и граничными условиями: $U(r, 0) = 0$; $U(r, 1) = 100$; $U(1, z) = 0$.

На рис.1 показаны зависимости относительной погрешности расчета от числа итераций. Относительная погрешность расчета определяется по максимальной разности значений потенциала в узле на двух итерациях. Расчет проводился методом фиксированных узлов. Начальная погрешность задавалась равной 0,01 для первой кривой и 0,0025 для второй. Для третьей кривой расчет проводился обычным методом последовательной верхней релаксации. Использование метода фиксированных узлов приводит к повышению быстродействия расчетов. Графики имеют два максимума. Первый максимум связан с переходом на новую сетку, а второй - со снятием фиксации в узлах этой сетки.

На рис.2 показаны зависимости относительной погрешности от числа итераций для расчетной области, которая представляла беско-

нечный цилиндрический конденсатор с двумя диэлектрическими обкладками.

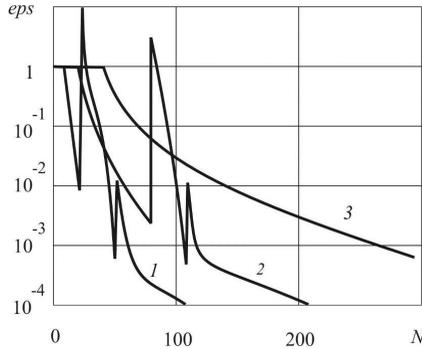


Рис. 1. Зависимости погрешности расчета от числа итераций. 1 - фиксированные узлы $\epsilon_{ps1} = 0,01$; 2 - фиксированные узлы, $\epsilon_{ps1} = 2,5 \cdot 10^{-3}$; 3 - обычный расчет, $\epsilon_{ps1} = 6,25 \cdot 10^{-4}$

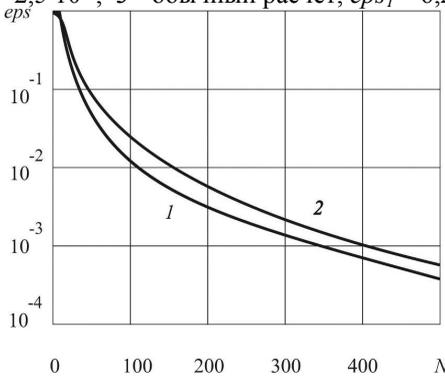


Рис. 2. Графики зависимости погрешности расчета от числа итераций

1 - равномерный шаг разбиения; 2 - соотношение для шагов 1:4
 Данная задача имеет аналитическое решение [3], которое сравнивалось с результатами численного моделирования. Радиус внутренней обкладки 1 см, радиус внешней обкладки 3 см, высота цилиндра 1 см. Граничные условия учитывали бесконечный цилиндрический конденсатор. Граница между двумя диэлектриками располагалась на расстоянии 2 см для первого графика и 1 см для второго. Относительная диэлектрическая проницаемость первого диэлектрика $\epsilon_1 = 1$, второго $\epsilon_2 = 2$. Граничные условия: $U(1, z) = 100$; $U(3, z) = 0$. Для первого графика шаг разбиения был равномерный по всей расчетной области. Для вто-

рого графика на границе двух диэлектриков выполнялось условие:
 $hr_2 = 4hr_1$.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Курбатов П. А., Аринчин С. А. Численный расчет электромагнитных полей. М.: Энергоатомиздат. 1984. 168 с.
2. Арефьев А.С., Антошкин В.А, Юдаев Ю. А. Расчет электрического поля на иррегулярных сетках // Информатика и прикладная математика. РГПУ, Рязань,1999. С.31-37
3. Говорков В. А. Электрические и магнитные поля. М.: Энергия. 1968. 214 с.

В.А. АСТАШИН, С.С. ШКАРИН

Рязанский государственный радиотехнический университет

АЛГОРИТМЫ ДЕКОДИРОВАНИЯ ИТЕРАТИВНОГО КОДА

Рассматриваются алгоритмы декодирования, ориентированные на исправление зависимых ошибок в непрерывном коде.

В соответствии с алгоритмом кодирования итеративного кода [1] контрольные разряды вычисляются путем суммирования по модулю 2 информационных разрядов по строкам и столбцам массива.

Ошибка при передаче блока вызовет нарушение четности числа единиц в столбце и строке массива. Процесс исправления ошибки заключается в поиске координат пересечения строки и столбца, в которых нарушено правило четности.

С некоторыми изменениями алгоритм кодирования блочного кода можно применить для непрерывного кода. На рис.1 показан буфер, предназначенный для формирования одного из вариантов непрерывного кода.

Ячейки первой, второй и третьей строк буфера заполняются информационными разрядами, а ячейки нулевой и четвертой строк предназначены для контрольных разрядов кода. В процессе сдвига информации вправо происходит вычисление контрольных разрядов. Элементы массива, участвующие в вычислении контрольных разрядов, выделены на рисунке пунктирными линиями.

Уравнения для вычисления контрольных разрядов:

$$X[0,5] = (A[3,8] + A[2,7] + A[1,6]) \bmod 2, \quad (1)$$

$$Y[4,1] = (X[0,5] + A[1,4] + A[2,3] + A[3,2]) \bmod 2. \quad (2)$$

Для декодирования информации необходим такой же буфер. Информационные и контрольные разряды в параллельном виде поступает на вход буфера и сдвигается вправо. Одновременно с помощью уравнений (3) и (4), полученных из (1) и (2), вычисляются разряды синдромов **В** и **С**, предназначенных для вычисления координат ошибок:

$$b_i = (X[0,5] + A[1,6] + A[2,7] + A[3,8]) \bmod 2, \quad (3)$$

$$c = (X[0,5] + A[1,4] + A[2,3] + A[3,2] + Y[4,1]) \bmod 2. \quad (4)$$

X[0,8]	X[0,7]	X[0,6]	X[0,5]	X[0,4]	X[0,3]	X[0,2]	X[0,1]
A[1,8]	A[1,7]	A[1,6]	A[1,5]	A[1,4]	A[1,3]	A[1,2]	A[1,1]
A[2,8]	A[2,7]	A[2,6]	A[2,5]	A[2,4]	A[2,3]	A[2,2]	A[2,1]
A[3,8]	A[3,7]	A[3,6]	A[3,5]	A[3,4]	A[3,3]	A[3,2]	A[3,1]
Y[4,8]	Y[4,7]	Y[4,6]	Y[4,5]	Y[4,4]	Y[4,3]	Y[4,2]	Y[4,1]

Рис. 1. Буфер кодирующего устройства

Число разрядов в синдроме **В** определяется размером строки в буфере, а для синдрома **С** достаточно выделить один разряд:

$$B = b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8; \quad C = c.$$

Разряды синдромов поступают на входы двух регистров, в которых происходит сдвиг синхронно со сдвигом информации в буфере приемника. В момент, когда $C=1$, содержимое синдрома **В** позволяет определить координаты места ошибки.

Рассмотрим алгоритм декодирования, ориентированный на исправление ошибок в **столбце** массива.

Таблица 1

Столбец с ошибками	Синдром В	Синдром С	Ошибка
8	1000000	0	-
7	1100000	0	-
6	1110000	0	-
5	1111000	1	X[0,5]
4	0011100	1	A[1,4]
3	0000110	1	A[2,3]
2	0000001	1	A[3,2]

Поступающие на вход буфера ошибки сдвигаются вправо и последовательно участвует в вычислении синдромов **В** и **С**. Состояния синдромов в процессе сдвига и исправления ошибок показаны в табл. 1. После исправления очередной ошибки соответствующий разряд в синдроме **В** обнуляется, и синдромы сдвигаются вправо синхронно с информацией в буфере.

Номер строки, в которой исправляется ошибка, вычисляется по формуле: $N = (m-1)/2$, где m – номер **крайней слева** позиции единицы в синдроме **V** (при условии $C=1$).

Чтобы код исправил ошибки в нескольких соседних столбцах достаточно включить в систему проверок более удаленные информационные разряды. Например, уравнения (5) и (6) представляют систему проверок, которая позволяет вычислить синдромы для исправления ошибок в двух соседних столбцах массива.

$$b_i = (X[0,9] + A[1,11] + A[2,13] + A[3,15]) \bmod 2, \quad (5)$$

$$c_i = (Y[4,1] + A[3,3] + A[2,5] + A[1,7] + X[0,9]) \bmod 2. \quad (6)$$

Номер строки с ошибкой в момент ее исправления вычисляется по формуле: $N = (m-1)/4$.

Рассмотрим алгоритм декодирования, ориентированный на исправление ошибок в **строке**. В табл. 2 показан процесс исправления ошибок, во второй строке массива.

Таблица 2

Ошибки в строке	Синдром V	Синдром C	Исправлена
11000000	000000	0	-
11100000	100000	0	-
11110000	110000	0	-
11111000	111000	0	-
11111100	111100	1	A[2,3]
11111110	111110	1	A[2,3]
11111110	111110	1	A[2,3]

В общем случае номер строки, где в данный момент исправляется ошибка, вычисляется по формуле: $N = (r-1)/2$,

где r – номер позиции **крайней справа** единицы в синдроме **V**.

После исправления очередной ошибки обнуляется соответствующий разряд в синдроме **V**, и синдромы сдвигаются в регистрах право.

Рассмотрим другой вариант формирования контрольных разрядов кода:

$$X[0,5] = (A[3,8] + A[2,7] + A[1,6]) \bmod 2, \quad (7)$$

$$Y[4,5] = (X[0,5] + A[1,5] + A[2,5] + A[3,5]) \bmod 2. \quad (8)$$

Слагаемые, входящие в уравнения, выделены на рисунке пунктирными линиями. Используя уравнения (7) и (8), можно получить систему проверок для вычисления синдромов. Система позволяет исправлять групповые ошибки в строке, номер которой определяется из уравнения: $N=r-1$, где r – номер **крайней справа** позиции единицы в синдроме **V**.

Все, рассмотренные выше алгоритмы декодирования для зависимых ошибок, способны исправлять независимые одиночные ошибки в непрерывном коде.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Элайес П. Безошибочное кодирование. Сб. «Коды с обнаружением и исправлением ошибок». ИЛ. 1956.
2. У. Питерсон, Э. Уэлдон. Коды, исправляющие ошибки. Изд. «Мир». 1976.
3. Бородин Л.Ф. Введение в теорию помехоустойчивого кодирования. М.: Советское радио. 1968.

А.В. БАКУЛЕВ, М.А. БАКУЛЕВА

Рязанский государственный радиотехнический университет

АЛГОРИТМ ВЫДЕЛЕНИЯ ПАРАЛЛЕЛЬНЫХ РЕГИОНОВ НА ОСНОВЕ УПРАВЛЯЮЩЕЙ СТРУКТУРЫ ПРОГРАММЫ

Рассматривается алгоритм предварительного крупнозернистого распараллеливания последовательной программы, заданной в форме промежуточного алгебраического представления, на основе выделения регионов из управляющей структуры программы.

В работе [1] предлагается подход к обеспечению мобильности программ для последующего их высокопроизводительного исполнения в среде вычислительных систем с многоядерными процессорами. В рамках данного подхода была представлена модель мобильного параллельного промежуточного кода программ, основанная на семействе алгебраических систем типов данных и операций языков программирования [2]. Выявление потенциального параллелизма последовательной программы, заданной в форме подобного промежуточного алгебраического представления, основывается на анализе зависимостей составляющих её частей друг от друга. В качестве подобных частей, в зависимости от выбранного масштаба рассмотрения — мелкозернистого или крупнозернистого, могут рассматриваться отдельные инструкции, группы операторов, итерации цикла и т.д. Методы и технологии мелкозернистого распараллеливания в настоящее время хорошо проработаны [3], однако они не способны обеспечить эффективность выполнения последовательных программ при многоядерных вычислениях. Поэтому на стадии предварительного распараллеливания формируется крупнозернистое разбиение промежуточного представления

программы. В качестве основы для подобного разбиения была выбрана управляющая структура программы, традиционно представляемая в виде управляющего графа программы (уграфа) [4, 5]. Синтез уграфа на основе предложенного промежуточного представления является простым и однозначным, так как структура управления явно выражается в нем с помощью концепции охраняемых операторов и специальных операций — стражей и объединителей [2].

Следующим шагом на пути к предварительному крупноблочному распараллеливанию является формирование графа программных зависимостей [4], представляющего из себя разбиение программы на связанные области — регионы. Регионы являются подграфами уграфа, порожденными теми вершинами-операторами, условия выполнения которых, одни и те же, что является очень важным для организации эффективного распараллеливания. Задача определения регионов сводится к однократному просмотру промежуточного представления программы и сопоставлению с каждым охраняемым оператором отдельного региона. Алгоритм, реализующий заданные действия, описывается следующей последовательностью шагов.

Вход. Композиция операций промежуточного представления программы $c_{np} = c_1 \otimes c_2 \otimes \dots \otimes c_{n-1} \otimes c_n$; начальные значения счетчика операций промежуточного представления $i = 1$ и счетчика регионов $j = 0$; список регионов программы $L_G = \emptyset$; список связей между регионами программы $L_C = \emptyset$.

Шаг 1. Без нарушения общности, все операции промежуточного представления программы заключаются в новый охраняемый оператор \tilde{n}_0^{gd} , предикатом которого является логическая константа ИСТИНА. Создаётся глобальный регион g_0 , соответствующий вновь введенному оператору \tilde{n}_0^{gd} , g_0 добавляется к списку регионов программы $L_G = L_G \cup \{g_0\}$.

Шаг 2. Последовательно просматриваются операции композиции $c_{i\delta}$ слева направо (в прямом направлении), начиная с i -ой, пока не отыщется операция объединитель $c_q(c_{k_1}^{gd}, c_{k_2}^{gd}, \dots, c_{k_m}^{gd}) \in \{c^{mg}, c^{mc}\}$, или не будет достигнута операция c_n — конец композиции.

Если поиск завершен неудачей, то переход на шаг 7.

Шаг 3. $i = n + 1$. Фиксируется текущий регион g_p , в который вложена операция c_q .

Шаг 4. В список регионов L_G добавляется по региону для каждого стража $c_{k_1}^{gd}, c_{k_2}^{gd}, \dots, c_{k_m}^{gd}$, указанного в объединителе c_q :

$$L_G = L_G \cup \{g_{k_1}, g_{k_2}, \dots, g_{k_m}\}.$$

Шаг 5. В список связей регионов L_C добавляются связи между регионом g_p и $\forall g_{k_j} \mid j = \overline{1, m} : L_C = L_C \cup \{(p, k_1), (p, k_2), \dots, (p, k_m)\}$.

Шаг 6. Переход на шаг 2.

Шаг 7. Дерево регионов построено и представлено списком вершин L_G и списком дуг L_C .

Рассчитаем сложность предложенного алгоритма. Просмотр всех операций композиции потребует выполнения n -действий. Для каждого из m найденных объединителей, потребуется выполнить k операций по добавлению нового элемента со списками регионов и связей регионов. Таким образом, общее количество выполняемых действий не превысит величины $n + m \cdot k$, причём m и k значительно меньше n . Следовательно, оценка общей сложности алгоритма составит $O(n)$ от общего количества операторов программы.

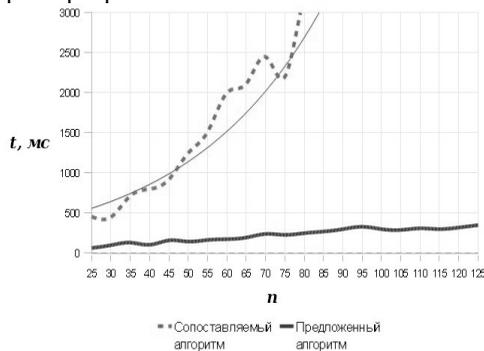


Рис. 1

Для проведения проверки алгоритма были написаны две программы. Первая программа выполняла последовательность шагов предложенного алгоритма. Входные данные задавались в виде списка управляющих операций, выходом являлся список регионов и список связей между регионами. Вторая программа производила построение графа регионов известными методами [4]. Результаты выполнения программ сравнивались, производились измерения времени их выполнения, показанные на рисунке. Вид данной зависимости подтверждает теоретически обоснованную сложность предложенного алгоритма.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Бакулев А.В. Алгоритм синтеза параллельной реализации последовательной программы для вычислительных систем, построенных на базе многоядерных процессоров//Вестник РГРТУ. Выпуск 30. Рязань: РГРТУ, 2009. С. 43-49.
2. Бакулев А.В. Представление управляющих конструкций универсальных языков в алгебраической системе управляющих операций// Новые информационные технологии: Межвуз. сб. науч. трудов. Рязань: РГРТА, 2002. с.14-18.
3. Корячко В.П., Скворцов С.В., Телков И.А. Архитектуры многопроцессорных систем и параллельные вычисления: учеб. пособие.- М.: Высш.шк., 1999.
4. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение. - СПб.: БХВ-Петербург, 2003.
5. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. - СПб.: БХВ-Петербург, 2004.

Е.А. БАРАНЧИКОВА

Рязанский государственный радиотехнический университет

ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ СИСТЕМ ЗАЩИТЫ МАЛЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Рассматриваются особенности малых информационных систем, связанные с ограниченными ресурсами на защиту информации. Проведен анализ защищенности системы до и после реализации предложенной политики безопасности на базе методики анализа угроз и уязвимостей и предложен подход, позволяющий снизить риски, связанные с вероятностью потери, искажения и компрометации данных.

Современные информационные системы (ИС) являются основой бизнес-процессов компаний и предприятий различных форм и назначений. Однако присущая всем ИС проблема – уязвимость – заставляет уделять особое внимание их защите от несанкционированных воздействий, способных привести к нарушению конфиденциальности, целостности или доступности информации.

Достаточно остро эта проблема стоит для предприятий малого бизнеса, на которых внедряются в основном так называемые малые ИС. Одной из главных особенностей проектирования таких систем является ограниченное количество ресурсов, выделяемых на разработку и внедрение системы, а также на ее обслуживание.

Поэтому в большинстве случаев невозможны реализация в должной мере физической защиты информации, приобретение

«тяжелых» СУБД и другого программного обеспечения (ПО), необходимого для организации комплексной защиты ИС, а также обучение работающего персонала и наличие постоянного квалифицированного администратора ИС.

Традиционно система защиты информации рассматривается как совокупность организационных, программных, программно-аппаратных и технических средств защиты информации (антивирусов, межсетевых экранов, систем обнаружения вторжений, систем контроля доступа и т.п.). При построении такой системы защиты информации в большинстве случаев учитываются лишь технологические угрозы информационной безопасности. При проектировании и внедрении систем защиты информации на угрозы организационного характера часто не обращают должного внимания, хотя они, как правило, сказываются на информационной системе не менее критично, чем угрозы технологического характера.

Особую трудность при создании малых ИС вызывает отсутствие на предприятии специальных подразделений или сотрудников, отвечающих за информационную безопасность. Это приводит к тому, что уделяется недостаточное внимание не только технологическим, но и организационным мерам по защите ИС, т.е. нет постоянного администрирования системы, контроля доступа к ИС и других средств физической и программной защиты информации.

Следствием этого являются проблемы, связанные со своевременным обнаружением «дыр» в защите ИС, возможностью несанкционированного входа в ИС, доступом посторонних лиц в рабочее помещение.

Также следует отметить и другие проблемы, связанные с отсутствием на предприятии целенаправленной кадровой политики безопасности:

а) персонал, работающий с ИС, в большинстве случаев не имеет необходимых знаний и навыков в области защиты информации (создание «простых» паролей, отсутствие контроля за своим рабочим местом и т.д.);

б) отсутствует возможность постоянного контроля за персоналом;

в) отсутствует проверка персонала на лояльность.

Это, в свою очередь, приводит к тому, что вся информация находится в свободном доступе для сотрудников предприятия. Чтобы этого избежать, необходимо принять ряд организационных мер для разграничения доступа сотрудников различного ранга, а также клиентов предприятия к ИС.

При этом надо добиваться, чтобы на стадии проектирования ИС в нее были заложены возможность разграничения доступа к различным классам информации, а также возможность предоставления различных прав доступа для различных групп пользователей.

Необходимо отметить ставшую актуальной в последнее время дополнительную угрозу для безопасности ИС, связанную с развитием телекоммуникаций. Электронная почта, доступ к различным справочным службам и прочим информационным ресурсам являются неотъемлемой частью бизнес-процессов. Угрозы безопасности связанные со «спамом», вирусами, перегрузками каналов, увеличением оплачиваемого трафика, сбоев в работе операционной системы. Все это вместе приводит к затруднению работы персонала, а также ставит под угрозу функционирование всего предприятия в целом.

Разработка методики оценки риска – достаточно трудоемкая задача. Во-первых, такая методика должна всесторонне описывать информационную систему, ее ресурсы, угрозы и уязвимости. Сложность заключается в том, чтобы построить максимально гибкую модель информационной системы, которую можно было бы настраивать в соответствии с реальной системой. Во-вторых, методика оценки рисков должна быть предельно прозрачна, чтобы владелец информации, использующий ее, мог адекватно оценить ее эффективность и применимость к своей конкретной системе.

Для определения соответствующих мер защиты ИС следует использовать системный подход. Решение о том, как обеспечить защиту, где реализовать защиту в ИС, какими должны быть типы и качество мер и средств защиты, требует проведения соответствующего анализа уязвимости и рисков.

Процесс анализа рисков включает:

1. оценку возможных потерь неработоспособности автоматизированной ИС;
2. анализ потенциальных угроз и уязвимых мест системы;
3. выбор оптимальных по цене мер и средств защиты, которые сокращают риск до приемлемого уровня.

Можно выделить следующие риски для малых ИС:

1. хищение информации;
2. потеря информации в результате сбоев в операционной системе, отказов аппаратной части, стихийных бедствий и других факторов;
3. «спам» – любая информация, полученная предприятием без явного запроса и приводящая к увеличению трафика и уменьшению пропускной способности каналов;

4. вирусы.

На сегодняшний день существует две основные методики оценки рисков информационной безопасности: метод оценки рисков, основанный на построении модели угроз и уязвимостей, и метод оценки рисков, основанный на построении модели информационных потоков.

Произведем расчет рисков на основе модели угроз и уязвимостей.

Чтобы оценить риск информации, необходимо проанализировать все угрозы, действующие на ИС, и уязвимости, через которые возможна реализация угроз. С точки зрения базовых угроз информационной безопасности существуют два режима работы алгоритма: одна базовая (суммарная) угроза и несколько угроз.

Произведем расчет рисков исходя из того, что существует одна базовая угроза - потеря информации.

1. На первом этапе рассчитываем уровень угрозы Th на основе критичности и вероятности реализации угрозы через данную уязвимость. Уровень угрозы показывает, насколько критичным является воздействие данной угрозы на ресурс с учетом вероятности ее реализации:

$$Th_i = \frac{ER_i}{100} \times \frac{P(V)_i}{100} \quad (1)$$

где ER_i – критичность реализации i -ой угрозы (указывается в процентах);

$P(V)_i$ – вероятность реализации i -ой угрозы через данную уязвимость (указывается в процентах).

2. Чтобы рассчитать уровень угрозы по всем уязвимостям CTh , через которую возможна реализация данной угрозы на ресурсе, просуммируем полученные уровни угроз через конкретные уязвимости по следующей формуле:

$$CTh_i = 1 - \prod_{i=1}^n (1 - Th_i) \quad (2)$$

3. Аналогично рассчитываем уровень угроз по ресурсу $CThR$, учитывая все угрозы, действующие на ресурс:

$$CThR_i = 1 - \prod_{i=1}^n (1 - CTh_i) \quad (3)$$

4. Риск по ресурсу R_i рассчитываем следующим образом:

$$R_i = CThT_i \times D_i \quad (4)$$

где D_i – критичность ресурса в год (задается в уровнях).

При задании риска в уровнях в качестве значения критичности берется оценка уровня.

5. Риск по информационной системе CR рассчитываем по формуле (5):

$$CR = \left(1 - \prod_{i=1}^n \left(1 - \frac{R_i}{100} \right) \right) \times 100 \quad (5)$$

В результате работы алгоритма получаем, что риск по информационной системе на основе ОС MS Windows по одной суммарной угрозе без организации защиты стремится к 90-95 %.

Рассмотрим наиболее вероятные угрозы и методы снижения их рисков.

1. При хищении информации требуется реализация защиты на уровне ОС, СУБД, прикладных программ. Необходимо производить резервное копирование данных, разбиение всех пользователей на основные группы и подгруппы с предоставлением соответствующих прав доступа к различным видам информации.

2. Потеря информации может происходить в результате сбоев в операционной системе, отказов аппаратуры ЭВМ, стихийных бедствий и других факторов. Основным способом борьбы с данной угрозой является резервное копирование, которое должно осуществляться на нескольких уровнях:

- оперативное резервирование – в течение дня на сервере;
- запись резервных копий на оптические носители;
- хранение дублей информации в других помещениях, с использованием шифрования данных.

3. «Спам» может быть получен любым предприятием, работающим в сети Internet. Существуют определенные методы борьбы со «спамом», однако им присуща одна черта: чем больше «спама» отсеивает система, тем больше важной электронной почты отсеивается вместе со «спамом». Существуют методы борьбы, основанные на стандартах передачи почты в сетях, а также методы, основанные на системах искусственного интеллекта, но у последних велика вероятность «ложной тревоги», в результате которой необходимая переписка может быть автоматически удалена, будучи принята за «спам».

4. Для борьбы с вирусами, следует использовать различные методы. У ЭВМ должен быть закрыт доступ из Internet к рабочим станциям (брандмауэр). В случае использования ОС семейства Windows такая защита имеет большую эффективность. Пользователи, работающие в сети, не должны иметь администраторских полномочий. Тогда полученные ими вирусы не смогут атаковать ОС, а будут инфицировать только пользовательские данные. Необходимо периодически производить антивирусное сканирование рабочих станций. Также важным является организационный метод борьбы с вирусами: персонал фирмы должен быть проинструктирован о путях потенциального попадания вируса из Internet.

Для защиты информационной системы в рамках малого предприятия от угроз следует принять следующие организационные меры.

1. Разработать, утвердить на уровне руководства и довести до сведения пользователей нормативные документы по информационной безопасности: политику безопасности, инструкции и регламенты по использованию средств обработки информации, правила предоставления доступа к информационным ресурсам. Должна быть предусмотрена ответственность за невыполнение требований данных нормативных документов.

2. Обязать каждого сотрудника подписать соглашение о конфиденциальности (о неразглашении конфиденциальной информации, обрабатываемой в компании), которое разрабатывается в соответствии с действующим законодательством и политикой безопасности компании. Предупредить сотрудников о последствиях нарушения соглашения о конфиденциальности.

3. Периодически проводить обучение пользователей по вопросам информационной безопасности, в частности детализируя методы, используемые злоумышленниками. Обучение пользователей рекомендуется завершать практическими тренировками и зачетом.

4. Использовать различные программные средства защиты информации, минимизирующие возможное влияние угроз организационного характера. Как один из вариантов можно рассматривать использование ОС класса Unix, система безопасности которых значительно превышает по качеству аналогичные предложения ОС семейства Windows. В рамках малого предприятия следует обратить внимание на бесплатно распространяемые ОС класса Unix, такие как Linux, Open Solaris, FreeBSD, OpenBSD...

В результате применения описанных мер риск по информационной системе на основе рабочих станций ОС MS Windows

и серверов под управлением ОС класса Unix по одной суммарной угрозе стремится к 25-30 % (см. рисунок).

Таким образом, предложенный подход к повышению уровня защиты малых ИС позволил при жестких финансовых ограничениях добиться значительного результата.

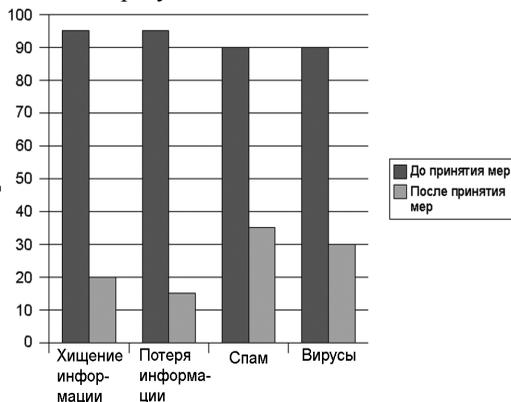


Рис. 1. Риски информационных систем

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Домарев В. В.* Безопасность информационных технологий. Методология создания систем защиты.- ДиаСофт, 2004. – 688 с.
2. Герасименко В. А., Малюк А. А. Основы защиты информации. – М.: МИФИ, 1997.
3. Куканова Н. Современные методы и средства анализа и управление рисками информационных систем компаний. <http://BugTraQ.ru>;
4. Диденко П. Основы безопасности для веб-разработчиков;
5. Грушо А. А. Теоретические основы защиты информации. - М.: Изд. Агентства "Яхтсмен", 1996. - 188 с.

В.М. БЕРДНИКОВ

Рязанский государственный радиотехнический университет

**МНОГОКРИТЕРИАЛЬНЫЙ СИНТЕЗ АНСАМБЛЯ
КОДОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ С НУЛЕВОЙ
ЗОНОЙ АВТОКОРРЕЛЯЦИОННОЙ ФУНКЦИИ
ДЛЯ НАЗЕМНОЙ РАДИОНАВИГАЦИОННОЙ СИСТЕМЫ**

Предложен алгоритм синтеза ансамбля кодовых последовательностей с нулевой зоной автокорреляционной функции для систем с кодовым разделением абонентов. Показана эффективность и работоспособность данного алгоритма на примере синтеза ансамбля из 8 кодовых последовательностей длиной 256 элементов.

Введение. Одним из преимуществ наземной радионавигационной системы (НРНС) является возможность синтеза ансамбля навигационных сигналов (НС) исходя из характеристик местности на которой должна быть обеспечена навигация объектов. Так при наличии априорных данных о структуре местности или расположении станций ретранслированных помех могут быть оценены наиболее характерные интервалы задержек отраженных сигналов относительно НС на линиях прямого видимости от опорных станций (ОС). Таким образом, синтез ансамбля кодовых последовательностей (КП) с нулевой зоной (НЗ) автокорреляционной функции (АКФ) [1, 2] позволяет уменьшить негативное влияние ретранслированных помех и многолучевости в дальней зоне (вне главного лепестка АКФ).

Обоснование требований к синтезу ансамбля КП. Из представленных алгоритмов синтеза (например, в [3, 4]) помимо требований к корреляционным свойствам ансамбля КП, как правило, не используется дополнительное условие о наличии НЗ АКФ, кроме того не обосновывается метод выбора «худшей» КП на каждой итерации из всего ансамбля при наличии нескольких показателей качества (ПК). Также не поясняется, какое количество символов выбранной КП инвертируется и сколько попыток производится до замены КП на новую. Для устранения перечисленных выше недостатков зададим следующие требования к синтезу ансамбля КП.

Пусть при синтезе ансамбля КП будут использоваться следующие ПК:

1) Среднее максимальных значений уровня боковых лепестков (УБЛ) АКФ в области задержек шириной $\Delta\tau_{zcz}$ среди всех КП ансамбля K_{zcz} . Данную область не обязательно выбирать около главного лепестка АКФ, но также можно в любом другом заданном интерва-

ле и таким образом адаптивно подстраиваться к текущей помеховой обстановки.

2) Среднее максимальных значений взаимокорреляционной функции (ВКФ) среди всех пар КП ансамбля $\gamma_{ВКФ}$.

3) Среднее максимальных значений УБЛ АКФ среди всех КП ансамбля $\gamma_{АКФ}$.

Следовательно, можно записать следующий функционал минимизации:

$$F = \alpha_1 K_{ZCZ} + \alpha_2 \gamma_{ВКФ} + \alpha_3 \gamma_{АКФ} \rightarrow \min \quad \text{при } \alpha_1 + \alpha_2 + \alpha_3 = 1. \quad (1)$$

Выбор конкретной «худшей» КП из данных трех (или меньше, в случае совпадений) осуществляется исходя из того насколько каждый ПК приближен к заданным весовым коэффициентам α_i .

Для определения данного приближения необходимо установить нижнюю границу для каждого ПК и определить их соотношение между собой в долях.

Зададим нижнюю границу для $\gamma_{ВКФ}$ и $\gamma_{АКФ}$ равной границе Велча (нормированной к длине КП):

$$W = \sqrt{(N-1)/(NL-1)}.$$

Исходя из того, что в идеальном случае K_{ZCZ} должно быть равно нулю, тогда перепишем функционал (1) в следующем виде:

$$F_1 = \alpha_1 (W - K_{ZCZ}) + \alpha_2 \gamma_{ВКФ} + \alpha_3 \gamma_{АКФ} \quad \text{при } \alpha_1 + \alpha_2 + \alpha_3 = 1. \quad (2)$$

Далее вычисляется соотношение ПК между собой в долях следующим образом:

$$pQ_i = (1 / (Q_i - W)) / \sum_{i=1}^3 (1 / (Q_i - W)), \quad (3)$$

где Q_i – i -й показатель качества.

После этого, по нахождению максимума среди разностей

$$\Delta_i = \alpha_i - pQ_i \quad (4)$$

определяется i -й ПК и соответствующая ему «худшая» КП, которой и требуется оптимизация.

Предложим следующую процедуру оптимизации структуры КП (5):

1. Один проход покоординатного спуска. Если улучшения функционала F_1 нет, то далее.

2. Один проход модифицированного покоординатного спуска [5]. Если улучшения функционала F_1 нет, то далее.

3. Случайный выбор двух битов и их инверсия (n_2 попыток). Если улучшения функционала F_1 нет, то далее.

4. Случайный выбор трех битов и их инверсия (n_3 попыток). Если улучшения функционала F_1 нет, то далее.

5. Случайный выбор N_{bits} битов и их инверсия (n_k попыток). Если улучшения функционала F_1 нет, то далее.

6. Замена «худшей» КП на новую и повторение пп. 1-5.

Выполнение пункта 6, как правило, приводит к откату функционала (2) к худшей величине и таким образом, необходимо обеспечить наиболее оптимальное количество попыток n_k для каждого шага процедуры с целью выхода из локального минимума за минимальное время.

Таким образом, входными параметрами данного алгоритма синтеза являются: весовые коэффициенты α_i , определяющие важность каждого ПК; объем ансамбля N ; длина КП L ; ширина НЗ АКФ $\Delta\tau_{ZCZ}$ и ее расположение для каждой КП ансамбля; максимальное количество битов для инверсии N_{bits} ; время, отведенное на синтез ансамбля T_{syn} ; количество попыток (инверсии битов) $n_{max i}$ выделенное на оптимизацию КП для данного ПК до замены текущей КП на новую; доля $\mathbf{p}_i = [p_{i2}, p_{i3}, \dots, p_{ik}]$ от общего количества попыток инверсии $n_{max i}$ и соответственно количество попыток $\mathbf{n}_i = [n_{i2}, n_{i3}, \dots, n_{ik}]$ для каждого шага процедуры оптимизации структуры КП i -го ПК ($n_{max i} = p_{i2}n_{max i} + p_{i3}n_{max i} + \dots + p_{ik}n_{max i} = n_{i2} + n_{i3} + \dots + n_{ik}$); коэффициент k_{max} , определяющий увеличение количество попыток $n_{max i}$ после замены КП ($n_{max new i} = k_{max} n_{max i}$), если при этом не произошло улучшения функционала F_1 .

Экспериментальная часть. Для проверки эффективности и работоспособности предложенного алгоритма синтеза ансамбля КП с НЗ АКФ было проведено исследование при следующих исходных данных: набор $\alpha_i = [0.8 \ 0.1 \ 0.1]$; длина КП $L = 256$; объем ансамбля $N = 8$; ширина НЗ АКФ $\Delta\tau_{ZCZ} = 4\tau_y$; время, отведенное на синтез $T_{syn} = 3$ часа; максимальное количество битов для инверсии $N_{bits} = 5$, как наиболее целесообразное для данной длины КП L ; коэффициент увеличения числа попыток после замены худшего кода $k_{max} = 1.2$; количест-

во попыток инверсии элементов КП $n_{\max i} = [218 \ 7725 \ 3573]$ (данные значения получены в ходе предварительных исследований для каждого ПК); доля от общего количества попыток инверсии элементов КП до замены $\mathbf{p}_1 = [0.22 \ 0.22 \ 0 \ 0.56]$, $\mathbf{p}_2 = [0.66 \ 0.17 \ 0.1 \ 0.07]$, $\mathbf{p}_3 = [0.13 \ 0.15 \ 0.22 \ 0.5]$ (данные значения получены в ходе предварительных исследований для каждого ПК).

Результаты синтеза ансамбля КП с НЗ АКФ представлены ниже:

$K_{ZCZ} = 0$; $K_{\max ZCZ} = 0$; $\gamma_{BK\Phi} = 0.1554$; $\gamma_{\max BK\Phi} = 0.1641$; $\gamma_{AK\Phi} = 0.125$; $\gamma_{\max AK\Phi} = 0.1406$.

Выводы. Из анализа полученных результатов следует, что АКФ у каждой КП ансамбля имеет нулевое значение в области задержки шириной 4 элемента, при этом максимальное значение ВКФ среди всех пар КП хуже на 0.0196 (12 %), чем у ансамбля Вейла и на 0.0349 (21 %), чем у ансамбля Голда. Данный ансамбль КП получен за 3 ч синтеза на двухъядерном процессоре частотой 3 ГГц.

Таким образом, наличие НЗ АКФ позволяет полностью нейтрализовать негативное влияние ретранслированных помех и многолучевости в дальней зоне в заданных областях задержек. Данный алгоритм обеспечивает возможность оперативно адаптировать ансамбль используемых КП к конкретной помеховой обстановки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Tang, X.H., and P. Z. Fang, and S. Matsufuji, "Lower bounds on correlation of spreading sequence set with low or zero correlation zone," *Electronic Letters*, Vol. 36, No. 6, pp. 210–218, March 2000.

2. P. Z. Fan, N. Suehiro, N. Kuroyanagi, and X. M. Deng, "A class of binary sequences with zero correlation zone," *IEEE Electron. Lett.*, vol. 35, pp. 777-779, May 1999.

3. J. J. Rushanan, "Weil Sequences: A Family of Binary Sequences with Good Correlation Properties," *IEEE International Symposium on Information Theory*, Seattle, Washington, July 9-14, 2006.

4. Gold R. Optimal binary sequences for spread spectrum multiplexing (Corresp.), *IEEE Transactions on Information Theory*, 13 (4), pp. 619–621, 1967.

5. Кириллов С.Н., Бакке А.В., Бодров О.А. Численный алгоритм оптимизации системы фазоманипулированных сигналов // *Радиотехника*. 1997. № 10 С.49-52.

А.И. БОБИКОВ, А.С. ЛИМОНОВА

Рязанский государственный радиотехнический университет

НЕЧЕТКОЕ УПРАВЛЕНИЕ С ПРЯМОЙ И ОБРАТНОЙ СВЯЗЯМИ НЕЛИНЕЙНЫМИ ОБЪЕКТАМИ ПЕРВОГО ПОРЯДКА

Предложен закон управления с прямой связью, который позволяет придать системе с нечеткой обратной связью возможность безошибочного воспроизведения постоянного задающего воздействия. В качестве примера рассмотрено применение этого закона управления к реальному объекту, описываемого нелинейной нечеткой моделью первого порядка.

Задачи стабилизации и слежения - две типичных задачи управления. В общем, задача слежения [1] является более трудной, чем задача стабилизации применительно к нелинейным системам.

Недавно, нечеткие регуляторы типа Такаги-Сугено (ТС) были успешно применены для расчета нелинейных систем [2], [3], построенных как ТС- системы. ТС модели являются универсальными аппроксиматорами для нелинейных систем. Нечеткая модель - это система с нечеткой логикой, перед которой ставится задача: по значениям входного сигнала сформировать значения выходного сигнала. Идея построения ТС модели состоит в том, чтобы каждое правило в базе правил определяло диапазон входа для модели, в котором приемлема линейная аппроксимация. С помощью ТС моделей нелинейный объект управления описывается рядом линейных моделей, так называемых локальных моделей. Механизм инференции осуществляет плавную (гладкую) интерполяцию между каждой локальной линейной моделью, чтобы обеспечить получение глобальной нелинейной модели. Затем, используя концепцию параллельно распределенной компенсации (ПРК) [4], можно для каждой локальной модели применить технологию проектирования линейных контроллеров и найти локальный контроллер, другими словами построить нечеткую обратную связь по состоянию (НОС) [5]. Результирующий глобальный контроллер является нелинейным нечетким плавным сопряжением локальных контроллеров и обеспечивает стабилизацию нечеткой модели в целом.

Расчет управления в задаче слежения - важная проблема для практических приложений. Например, слежение при автоматизированном управлении (робототехника, система технического зрения), слежение при управлении двигателями самолета. Однако, задача слежения для систем на основе модели ТС еще слабо изучена, особенно для непрерывных систем.

Хотя в [6], предложен метод линеаризации обратной связью, чтобы спроектировать нечеткий регулятор в системе слежения для дискретных (цифровых) систем, но как указано в [7], регулятор (контроллер) не обеспечивает устойчивость для неминимально-фазового объекта управления. Также этот способ управления гарантирует устойчивость системы только по начальным условиям, т.е. в отсутствии задающего воздействия, что безусловно ограничивает возможности его применения. Другим возможным подходом является введение супервизорного управления [8]. Однако возникают относительные трудности при его реализации.

В данной работе рассматривается подход, который путем введения прямой связи, позволяет придать системе с нечеткой обратной связью способность обрабатывать безошибочно в установившемся режиме постоянное задающее воздействие.

Пусть известна глобальная непрерывная нечеткая модель Такаги-Сугено непрерывного нелинейного объекта первого порядка

$$\text{Если } y \text{ есть } A_l, \text{ То } \dot{y} = a_l y(t) + b_l u(t), \quad l = \overline{1, N}, \quad (1)$$

где N – число правил, A_l ($j=1, 2, \dots, n$) нечеткие множества, $u(t)$ управляющий сигнал, $y(t)$ – управляемая величина, и (a_l, b_l) коэффициенты l -ой локальной динамической модели.

Цель управления заключается в том, что требуется построить нечеткий регулятор, обеспечивающий безошибочное воспроизведение постоянного задающего воздействия $v = const$ после окончания переходного процесса.

Используя стандартный метод нечеткой инференции [9], т.е. используя синглтонную фаззификацию, алгебраическое произведение для вычисления степени истинности правил и центр тяжести для дефаззификации, выход ТС нечеткой модели (1) можно представить как

$$\dot{y}(t) = \sum_{l=1}^N \lambda_l(y) [a_l y(t) + b_l u(t)], \quad (2)$$

где

$$a(y) = \sum_{l=1}^N \lambda_l(y) a_l, \quad b(y) = \sum_{l=1}^N \lambda_l(y) b_l. \quad (3)$$

$\lambda_l(y)$ есть нормированная степень истинности l -го правила, определяемая выражением

$$\lambda_l(y) = \frac{\mu_l(y)}{\sum_{i=1}^N \mu_i(y)}. \quad (4)$$

$\mu_i(y)$ есть степень принадлежности управляемой величины к нечеткому множеству A_i , другими словами, степень истинности i -го правила, причем

$$\lambda_i(x) \geq 0, \sum_{i=1}^N \lambda_i(x) = 1. \quad (5)$$

Заметим, что модель (2) является нелинейной моделью, т.к. функции принадлежности являются нелинейными функциями управляемой величины. Модель (2) фактически является нечеткой моделью в переменных состояния.

Нечеткая обратная связь для этого объекта определяется, как

$$u(t) = - \sum_{j=1}^N \lambda_j(y) k_j y(t), \quad (6)$$

где

$$k_l = \frac{-a_{ll} + a_l}{b_l} \quad (7)$$

скалярный коэффициент локальной обратной связи по состоянию, a_{ll} есть желаемые значения полюсов замкнутых локальных динамических моделей. Отсюда выражение (2) можно переписать в виде

$$\dot{y}(t) = \sum_{l=1}^N \sum_{j=1}^N \lambda_l(y) \lambda_j(y) (a_l - b_l k_j) y(t) = f(y) y(t),$$

где

$$f(y) = \sum_{l=1}^N \sum_{j=1}^N \lambda_l(y) \lambda_j(y) (a_l - b_l k_j) = \sum_{l=1}^N \sum_{j=1}^N \lambda_l(y) \lambda_j(y) a_{lj}$$

или

$$f(y) = a(y) - b(y)k(y). \quad (7)$$

Здесь

$$a_{lj} = a_l - b_l k_j. \quad (8)$$

Нетрудно убедиться, что полученная система не в состоянии обрабатывать не равное нулю задающее воздействие $v \neq 0$, другими словами, такая система не удовлетворяет требованию нулевой установившейся ошибки $e_{уст} = v - y_{уст}$, $y_{уст}$ - установившееся значение выхода системы.

Заметим, что для устойчивой l -ой локальной динамической модели с коэффициентом усиления $k_{ойl} = -b_l / a_l$, можно обеспечить

безошибочное воспроизведение в установившемся режиме постоянно-го воздействия за счет последовательного включения элемента прямой связи с коэффициентом усиления

$$k_{npl} = -a_l / b_l, \quad (9)$$

обратным коэффициенту усиления l -ой локальной динамической модели. Тогда нечеткая прямая связь для всего объекта определяется как

$$u_{np}(t) = \sum_{j=1}^N \lambda_j(y) k_{npj} v(t) = k_{np}(y) v(t),$$

где

$$k_{np}(y) = \sum_{j=1}^N \lambda_j(y) k_{npj} = -a(y) / b(y) \quad (10)$$

коэффициент усиления глобальной прямой связи.

Поэтому, предполагая, что $b(y) \neq 0$ (это условие выполняется для многих объектов), резонно с целью достижения поставленной задачи использовать другой закон управления, а именно, закон управления с прямой и обратной связью

$$u(t) = k(y)[v - y(t)] + k_{np}(y)v(t). \quad (11)$$

Выражению (11) соответствует структурная схема нечеткой системы управления, показанная на рис. 1.

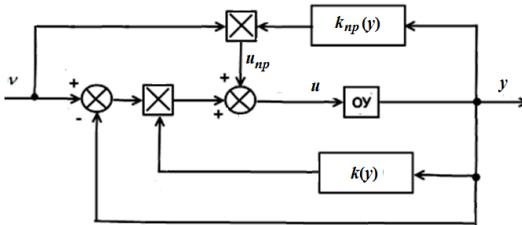


Рис. 1

Легко показать, что в случае устойчивости такая система безошибочно воспроизводит в установившемся режиме постоянное задающее воздействие. Подставляя (11) в (2), с учетом (3) и (10) получаем уравнение нечеткой системы управления с прямой и обратной связью

$$\dot{y}(t) = a(y) + b(y)\{k(y)[v(t) - y(t)] - a(y)/b(y)v(t)\},$$

так что

$$\dot{y}(t) = f(y)[y(t) - v(t)]. \quad (12)$$

В случае устойчивости эта нечеткая замкнутая система в установившемся режиме, т.е. при $y(t) = y_{уст} = const$ обеспечивает безошибочное воспроизведение постоянного задающего воздействия $v(t) = v = const$, так как из последнего уравнения следует, что $\dot{y}(t) = 0$, только, если $y_{уст} = v$.

Найдем условия, при выполнении которых ошибка слежения с течением времени стремится к нулю. Используем квадратичную функцию Ляпунова [10, 11]

$$V(t) = \frac{1}{2} e^2(t).$$

При этом ее производная

$$\dot{V} = e(t)\dot{e}(t),$$

где ошибка слежения

$$e(t) = v - y(t).$$

Следовательно,

$$\dot{e}(t) = -\dot{y}(t).$$

С учетом выражения (12) получаем

$$\dot{e}(t) = f(y)[v - y(t)] = f(y)e(t).$$

Отсюда

$$\dot{V}(t) = f(y)e^2(t).$$

Чтобы с течением времени ошибка стремилась к нулю, нужно обеспечить отрицательность первой производной функции Ляпунова

$$\dot{V}(t) < 0$$

или

$$f(y) < 0. \quad (13)$$

Полученное условие устойчивости (10) содержит ограничения на выбор коэффициентов локальных обратных связей по состоянию в неявном виде. Чтобы записать это условие в явном виде, преобразуем выражение (7) для функции $f(y)$:

$$f(y[i]) = \sum_{l=1}^N \lambda_l^2(y) a_{ll} + 2 \sum_{j=1}^{N-1} \sum_{l=j+1}^N \lambda_j(y) \lambda_l(y) \left(\frac{a_{jl} + a_{lj}}{2} \right), \quad (14)$$

где коэффициенты $a_{jl} = a_j - b_j k_l$, $a_{lj} = a_l - b_l k_j$, $l = \overline{1, N}$. Выражение (14) получено из

$$\begin{aligned} \left(\sum_{j=1}^N \lambda_j^2(y) \right)^2 &= (\lambda_1(y) + \lambda_2(y) + \dots + \lambda_N(y))^2 = \lambda_1^2(y) + \lambda_2^2(y) + \dots + \lambda_N^2(y) + \\ &+ 2\lambda_1(y)\lambda_2(y) + \dots + 2\lambda_1(y)\lambda_N(y) + 2\lambda_2(y)\lambda_3(y) + 2\lambda_2(y)\lambda_4(y) + \dots + \\ &+ 2\lambda_2(y)\lambda_N(y) + \dots + 2\lambda_{N-1}(y)\lambda_N(y) = \sum_{j=1}^N \lambda_j^2(y) + 2 \sum_{j=1}^{N-1} \sum_{l=j+1}^N \lambda_j(y)\lambda_l(y) = 1, \end{aligned}$$

и дальнейшее преобразования приводят к следующему выражению

$$\begin{aligned} f(y[i]) &= \sum_{l=1}^N \lambda_l^2(y)(a_{ll} - 1) + \sum_{j=1}^N \lambda_j^2(y) + \sum_{j=1}^{N-1} \sum_{l=j+1}^N \lambda_j(y)\lambda_l(y)(a_{jl} + a_{lj} - 2) + \\ &+ 2 \sum_{j=1}^{N-1} \sum_{l=j+1}^N \lambda_j(y)\lambda_l(y) = \sum_{l=1}^N \lambda_l^2(y)(a_{ll} - 1) + 2 \sum_{j=1}^{N-1} \sum_{l=j+1}^N \lambda_j(y)\lambda_l(y) \left(\frac{a_{jl} + a_{lj}}{2} - 1 \right) + \\ &+ \sum_{j=1}^N \lambda_j^2(y) + \sum_{j=1}^{N-1} \sum_{l=j+1}^N \lambda_j(y)\lambda_l(y) = \sum_{l=1}^N \lambda_l^2(y)(a_{ll} - 1) + 2 \sum_{j=1}^{N-1} \sum_{l=j+1}^N \lambda_j(y)\lambda_l(y) \left(\frac{a_{jl} + a_{lj}}{2} - 1 \right) + \\ &+ 1, \end{aligned}$$

Т. к. $\lambda_i^2 > 1, 0 < \lambda_j < 1$, то нетрудно придти к следующему достаточному условию глобальной асимптотической устойчивости нечеткой системы с прямой и обратной связью

$$a_{ll} < 0, \quad l = \overline{1, N},$$

$$\frac{a_{jl} + a_{lj}}{2} < 0 \Rightarrow a_{jl} + a_{lj} < 0 \quad l < j \leq N.$$

Пример. В качестве объекта управления рассмотрим уравнительный бак с жидкостью бочкообразной формы [9], внешний вид которого представлен на рис. 2. Цель управления заключается в том, чтобы за счет регулирования притока жидкости через верхнюю трубу, обеспечить стабилизацию уровня жидкости в баке на заданном уровне, который может принимать значения от 0.001м до 10м.

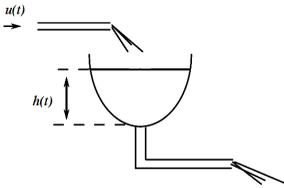


Рис. 2

Уравнение этого объекта в непрерывной форме

$$\frac{dh(t)}{dt} = \frac{-c\sqrt{2gh(t)}}{ah(t) + b} + \frac{1}{ah(t) + b} u(t), \quad (13)$$

где $a = 1\text{м}$, $b = 2\text{м}^2$ - параметры, характеризующие форму бака, $c = 1\text{м}^2$ - определяет параметр выходной трубы, g - ускорение свободного падения ($g = 9.8\text{м}/\text{с}^2$).

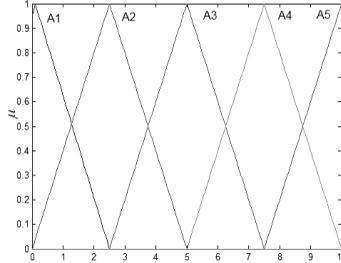


Рис. 3

В результате идентификации были получены следующие 5 правил, описывающих данный объект управления:

Если $h(t)$ есть A_1 , то $\dot{h}(t) = -3.3333h(t) + 0.4762u(t)$,

Если $h(t)$ есть A_2 , то $\dot{h}(t) = -0.3111h(t) + 0.2222u(t)$,

Если $h(t)$ есть A_3 , то $\dot{h}(t) = -0.1414h(t) + 0.1429u(t)$,

Если $h(t)$ есть A_4 , то $\dot{h}(t) = -0.0851h(t) + 0.1053u(t)$,

Если $h(t)$ есть A_5 , то $\dot{h}(t) = -0.0583h(t) + 0.08333u(t)$.

Найдем, используя (9), коэффициенты усиления прямых связей для локальных динамических моделей:

$$k_{np1} = 6.9998, k_{np2} = 1.4001, k_{np3} = 0.9895, k_{np4} = 0.8072, k_{np5} = 0.6996.$$

Выберем желаемые полюсы локальных замкнутых систем:

$$p_1 = -a_{11} = -0.4, \quad p_2 = -a_{22} = -0.4, \quad p_3 = -a_{33} = -0.4, \quad p_4 = -a_{44} = -0.4,$$

$p_5 = -a_{55} = -0.4$. Применяя формулы (7) и (8), находим коэффициенты

локальных обратных связей по состоянию

$$k_1 = -6.1598, \quad k_2 = 0.4001, \quad k_3 = 1.8097, \quad k_4 = 2.9905, \quad k_5 = 4.102$$

и значения

$$a_{12} + a_{21} = -2.4662, \quad a_{13} + a_{31} = -3.4562,$$

$$a_{14} + a_{41} = -4.1939, \quad a_{15} + a_{51} = -4.8319,$$

$$a_{23} + a_{32} = -0.9118, \quad a_{24} + a_{42} = -1.1028,$$

$$a_{25} + a_{52} = -1.3142, \quad a_{34} + a_{43} = -0.8444,$$

$$a_{35} + a_{53} = -0.9366, \quad a_{45} + a_{54} = -0.8245.$$

При этом

$$f(y) = \lambda_1^2 a_{11} + \lambda_2^2 a_{22} + \lambda_3^2 a_{33} + \lambda_4^2 a_{44} + \lambda_5^2 a_{55} + \lambda_1 \lambda_2 (a_{12} + a_{21}) + \lambda_1 \lambda_3 (a_{13} + a_{31}) + \\ + \lambda_1 \lambda_4 (a_{14} + a_{41}) + \lambda_1 \lambda_5 (a_{15} + a_{51}) + \lambda_2 \lambda_3 (a_{23} + a_{32}) + \lambda_2 \lambda_4 (a_{24} + a_{42}) + \\ + \lambda_2 \lambda_5 (a_{25} + a_{52}) + \lambda_3 \lambda_4 (a_{34} + a_{43}) + \lambda_3 \lambda_5 (a_{35} + a_{53}) + \lambda_4 \lambda_5 (a_{45} + a_{54}).$$

Результаты моделирования системы управления баком в MATLAB/Simulink показаны на рис. 4. При этом в качестве задающего воздействия используется постоянный сигнал, который изменяет свое значение каждые 15 секунд. Форма управляющего сигнала приведена на рис. 5.

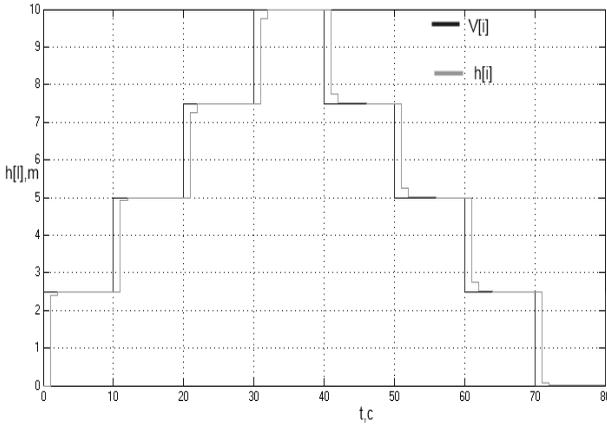


Рис. 4

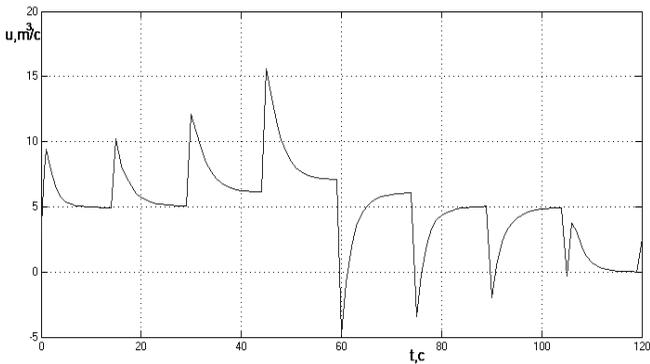


Рис. 5

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Chung-Shi Tseng, “Fuzzy Tracking Control Design for Nonlinear Dynamic Systems via T-S Fuzzy model”, *IEEE Trans on Fuzzy Systems*, vol. 9, No. 3, June 2001.

2. T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Trans. Syst., Man, Cybern.*, vol. 15, pp. 116–132, Jan. 1985.

3. A. Jadbabaie, M. Jamshidi, and A. Titli, “Guaranteed-cost design of continuous-time Takagi-Sugeno fuzzy controllers via linear matrix inequalities,” in *IEEE World Congress Computational Intelligence*, Anchorage, AK, May 1998, pp. 268–273.

4. Tanaka K., Cugeno M. Stability analysis and design of fuzzy control systems. *Fuzzy Sets Syst.* Vol. 45. No. 2, 1992.

5. Бобиков А.И. Интеллектуальные системы управления (Проектирование нечетких ПИД-контроллеров и нечетких обратных связей, нейронные сети): учеб. пособие. - Рязань: РГРТА, 2008.

6. C. C. Kung and H. H. Li, “Tracking control of nonlinear systems by fuzzy model-based controller,” in *Proc. IEEE Int. Conf.*, vol. 2, July 1997, pp. 623–628.

7. H. Ying, “Analytical analysis and feedback linearization tracking control of the general Takagi-Sugeno fuzzy dynamic systems,” *IEEE Trans. Syst., Man., Cybern.*, vol. 29, pp. 290–298, May 1999.

8. Бобиков А.И., Сивов С.А. Нечеткое супервизорное управление объектами первого порядка. – Вестник РГРТУ. № 3 (выпуск 33). Рязань, 2010.

9. Passino K., Yurkovich S., “Fuzzy control”, Addison Wesley Longman, 1998.

10. Мирошник И.В. Теория автоматического управления. Нелинейные и оптимальные системы. - СПб.: Питер, 2006.

11. H. K. Khalil, *Nonlinear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

В.А. БОГОНАТОВ, С.В. СКВОРЦОВ

Рязанский государственный радиотехнический университет

РАЗРАБОТКА ХЕШ-ФУНКЦИЙ ДЛЯ СОВРЕМЕННЫХ КОМПИЛЯТОРОВ

Рассмотрена задача построения «хорошей» хеш-функции для формирования хеш-таблицы идентификаторов, применяемой при разработке трансляторов. Представлены результаты исследования эффективности предложенных алгоритмов вычисления хеш-функций.

В реальных исходных текстах программ количество идентификаторов столь велико, что даже логарифмическую зависимость времени поиска от их числа нельзя признать удовлетворительной. Эффективным решением данной проблемы является использование хеш-адресации на основе хеш-функций [1].

Хеш-функцией F называется некоторое отображение множества входных элементов R на множество целых неотрицательных чисел Z :

$$F(r) = n; r \in R; n \in Z.$$

Множество допустимых входных элементов R называется областью определения хеш-функции. Множеством значений хеш-функции F называется подмножество M из множества целых неотрицательных чисел Z , содержащее все значения, возвращаемые функцией F :

$$\forall r \in R : F(r) \in M ; \\ \forall m \in M : \exists r \in R : F(r) = m .$$

При работе с таблицей идентификаторов хеш-функция должна выполнять отображение имен идентификаторов на множество целых неотрицательных чисел. Областью определения хеш-функции будет множество всех возможных имен идентификаторов.

Проблема выбора хеш-функции не имеет универсального решения. Хеширование обычно происходит за счет выполнения над цепочкой символов некоторых простых арифметических и логических операций. Самой простой хеш-функцией для символа является код внутреннего представления в компьютере литеры символа. Эту хеш-функцию можно использовать и для цепочки символов, выбирая первый символ в цепочке [1].

Очевидно, что такая примитивная хеш-функция будет неудовлетворительной: при ее использовании возникнет проблема – двум различным идентификаторам, начинающимся с одной и той же буквы, будет соответствовать одинаковое значение хеш-функции. Такая ситуация, когда двум или более идентификаторам соответствует одно и то же значение хеш-функции, называется коллизией.

Существует ограничение, делающее невозможным создание взаимно однозначной хеш-функции для идентификаторов. Дело в том, что в реальности область значений любой хеш-функции ограничена размером доступного адресного пространства компьютера. Поэтому организовать взаимно однозначное отображение бесконечного множества идентификаторов на конечное множество адресов невозможно [1, 3].

Таким образом, создать взаимно однозначную хеш-функцию на практике невозможно. Следовательно, неизбежно и возникновение коллизий.

Создание эффективной хеш-функции – это отдельная задача разработчиков компиляторов, и полученные результаты, как правило, держатся в секрете. Хорошая хеш-функция распределяет поступающие на ее вход идентификаторы равномерно на все имеющиеся в распоряжении адреса, чтобы свести к минимуму количество коллизий. В настоящее время существует множество хеш-функций, но, как было показано выше, идеального хеширования достичь невозможно.

Рассмотрим два разработанных алгоритма вычисления хеш-функции, которые отличаются от их классических аналогов, приведенных в работах [1, 2], более высокой скоростью работы и меньшим числом коллизий.

Алгоритм 1 (на основе логических операций).

1. Вычисление маски хеш-таблицы на основе ее размера:

$$\begin{aligned} \text{usefulBits} &= (\text{int}) \text{floor} (\log (\text{tableSize}) / \log (2)); \\ \text{MASK} &= (0\text{xFFFFFFFF F}) \gg \gg (32 - \text{usefulBits}). \end{aligned}$$

2. Для каждого символа ch_i выполняется следующая операция:

$$\begin{aligned} T &= ch_i \ll i / (\text{usefulBits}); \\ \text{hashresult} &= \text{hashresult} \oplus T. \end{aligned}$$

3. Применение маски к результирующему коду:

$$\text{res} = \text{hashresult} \& \text{MASK}.$$

Алгоритм 2 (на основе оптимального постоянного множителя).

1. Вычисление маски хеш-таблицы на основе ее размера.
2. Для каждого символа ch_i выполняется следующая операция:

$$\begin{aligned} \text{hashresult} &= \text{hashresult} * 37; \\ \text{hashresult} &= \text{hashresult} + ch_i. \end{aligned}$$

3. Применение маски к результирующему коду.

Произведем сравнение эффективности разработанных хеш-функций с широко известной реализацией Питера Вэйнбергера (из AT&T Bell Laboratories) под названием PJW-32 (hashpww) [2].

Результаты сравнения названных выше алгоритмов представлены для трех размеров хеш-таблицы (256, 512 и 1024 ячеек). Вычисле-

ние значений хеш-функции выполнялось для всех идентификаторов, содержащихся в трех программах на языке PASCAL. Заметим, что в любой из них содержалось не более 256 уникальных имен.

Алгоритм	Количество коллизий (среднее)	Вероятность коллизии (средняя)	Число уникальных идентификаторов (среднее)	Размер таблицы
1	81	32.92 %	246	256
2	86	34.95 %	246	256
PJW-32	121	49.18 %	246	256
1	56	22.76 %	246	512
2	54	21.95 %	246	512
PJW-32	85	34.55 %	246	512
1	27	10.97 %	246	1024
2	37	15.04 %	246	1024
PJW-32	57	23.17 %	246	1024

Представленная выше сравнительная таблица показывает, что разработанные алгоритмы создают коллизию хеш-кодов с меньшей вероятностью, чем стандартный алгоритм PJW-32. Уменьшение количества коллизий позволяет сократить вычислительные затраты на их разрешение [1], что в свою очередь ускорит процесс компиляции программы.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Грис Д. Конструирование компиляторов для цифровых вычислительных машин. - М.: Мир, 1975.
2. Ахо А., Сети Р., Ульман Д. Компиляторы. Принципы, технологии, инструменты. - М.: Издательский дом «Вильямс», 2003.
3. Гордеев А.В., Молчанов А.Ю. Системное программное обеспечение. - СПб.: Питер, 2001.

О.А. БОДРОВ

Рязанский государственный радиотехнический университет

ФОРМИРОВАНИЕ СИСТЕМ ФАЗОМАНИПУЛИРОВАННЫХ СИГНАЛОВ ДЛЯ АСИНХРОННЫХ АДРЕСНЫХ СИСТЕМ ПЕРЕДАЧИ ИНФОРМАЦИИ

Рассматриваются вопросы формирования систем фазоманипулированных сигналов с учетом различных показателей качества для асинхронных адресных систем передачи информации.

В асинхронных адресных системах передачи информации (ААСПИ) широко применяются фазоманипулированные (ФМн) сигналы. В ААСПИ все абоненты работают в общей полосе частот и поэтому, при передаче информации сигналы различных абонентов создают взаимные помехи. Для уменьшения взаимных и структурных помех необходимо формировать системы ФМн сигналов с минимальными лепестками взаимнокорреляционной функции (ВКФ) [1]. Как отмечено в [2], для обеспечения заданной помехоустойчивости необходимо выбирать системы кодовых последовательностей по критерию вероятности ошибки $P_{ош}$.

В общем случае плотность вероятности взаимной помехи – асимметричная функция, поэтому для определения $P_{ош}$ необходимо учитывать и нечетные моменты, что приводит к выражению [3]

$$P_{ош} \approx [1 - F(q)] \left\{ 1 + (\alpha_s / 6\sqrt{n})(q - q^2) + (\gamma / 24n)(q^4 - 3q^2) + (\alpha_s^2 / 72n)(q^5 - 10q^4 + 15q^2) \right\}$$

где $F(q)$ - интеграл вероятности, $q = 1/\sigma\sqrt{n}$ - отношение сигнал/помеха, $n = 2V_M$, $\sigma^2 = \frac{1}{2N-1} \sum_{i=1}^{2N-1} (R_V[i] - m_1)^2$ - дисперсия системы сигналов, N - длина кодовой последовательности, $R_V[i]$ - отсчеты

ВКФ j -го и k -го абонентов, $m_1 = \frac{1}{2N-1} \sum_{i=1}^{2N-1} R_V[i]$ - математическое

ожидание ВКФ, $\alpha_s = [V(V-1)]^{-1} \sum_{k=1}^V \sum_{\substack{j=1 \\ k \neq j}}^V \alpha_{jk}$ - коэффициент асимметрии

системы, $\alpha_{jk} = M_3 / \sigma^3$, $M_3 = \frac{1}{2N-1} \sum_{i=1}^{2N-1} (R_V[i] - m_1)^3$ - третий цен-

тральный момент, $\gamma = [V(V-1)]^{-1} \sum_{k=1}^V \sum_{\substack{j=1 \\ k \neq j}}^V \gamma_{jk}$ - коэффициент эксцесса системы, $\gamma_{jk} = (M_4/\sigma^4) - 3$, $M_4 = \frac{1}{2N-1} \sum_{i=1}^{2N-1} (R_V[i] - m_1)^4$ - четвертый центральный момент.

Кроме вероятности ошибки $P_{ош}$ при синтезе системы сигналов следует учитывать и вероятность правильного обнаружения $P_{пр}$, которая тем выше, чем ниже уровень боковых лепестков (УБЛ) автокорреляционной функции (АКФ) [1], т.к. любые частичные корреляции, определяющиеся боковыми лепестками АКФ, могут привести к ложному срабатыванию и нарушению работы приемника.

Таким образом, главными показателями качества для системы ФМн сигналов являются как вероятность ошибки $P_{ош}$, так и УБЛ АКФ. Наибольшее влияние на величину вероятности ошибки $P_{ош}$ оказывает коэффициент эксцесса ВКФ γ [2]. Использование данного показателя качества позволяет резко уменьшить вычислительные затраты при синтезе систем ФМн сигналов. Таким образом, в качестве критерия приближения рекомендуется использовать следующий функционал:

$$M[R_A, \gamma] = \alpha R_{Amax} + (1 - \alpha)\gamma, \quad (1)$$

В ходе экспериментальных исследований было определено оптимальное значение весового коэффициента функционала (1) $\alpha = 0.6$. Проводилось сравнение характеристик трех систем ФМн сигналов с числом символов $N = 64$ (таблица 1): I - системы, основанной на кодовых последовательностях Уолша; II - производной системы сигналов, кодовые последовательности которой получаются при помощи посимвольного умножения последовательностей Уолша на производящую последовательность [1]; III - системы, кодовые последовательности которой синтезированы по критерию (1). Из анализа таблицы 1 видно, что при практически одинаковой дисперсии σ^2 всех трех систем сигналов коэффициент эксцесса γ для синтезированной системы ФМн сигналов III минимален. А следовательно, зависимость вероятности ошибки от отношения сигнал-помеха $P_{ош}(q)$ для системы III лучше приближается к потенциально предельному значению $1 - F(q)$, чем системы I и II.

Таблица 1

Тип системы сигналов	σ^2	γ	$R_{V \max} / \sqrt{N}$
I	0.0078	20	7.875
II	0.0077	0.64	3.125
III	0.0078	0.56	2.166

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Варакин Л.Е. Системы связи с шумоподобными сигналами. - М.: Радио и связь, 1985. -384с.
2. Варакин Л.Е. Выбор систем сигналов для ААСС при когерентном приеме // Электросвязь. 1971. Т.25. № 12. С.51-59.
3. Кириллов С.Н., Бакке А.В., Бодров О.А. Многокритериальный синтез систем фазоманипулированных сигналов для асинхронных систем передачи информации // МНТС "Проблемы передачи и обработки информации в информационно-вычислительных сетях": Тез докл. М. 1995. С.16-17.

С.Н. БУЗЫКАНОВ

Рязанский государственный радиотехнический университет

ТЕХНОЛОГИЯ ПЕРЕДАЧИ ТЕЛЕМЕТРИЧЕСКОЙ ИНФОРМАЦИИ С ПОНИЖЕННЫМ УРОВНЕМ ШУМОВ КВАНТОВАНИЯ

Предложен алгоритм передачи телеметрических сигналов, позволяющий существенно снизить шумы при использовании малой разрядности квантования. Показано, что для экспериментального телеметрического сигнала снижение шумов квантования может составлять около 3 раз при неизменной частоте дискретизации.

Цифровая передача телеметрических сигналов широко применяется в системах автоматизации производственных процессов на различных предприятиях. При этом важным вопросом является снижение требуемой пропускной способности канала передачи, что обеспечивает повышение информативности канала и снижение стоимости системы. Требуемая пропускная способность канала определяется частотой дискретизации сигнала и разрядностью квантования каждого отсчета. Минимум частоты дискретизации определяется теоремой В.А. Котельникова [1] и является

фундаментальной величиной, что препятствует ее использованию для снижения требований к каналу. Другой важной величиной является разрядность квантования, выбор которой осуществляется исходя из требований к точности представления сигналов.

Для снижения ошибок, возникающих при квантовании телеметрического сигнала с малой разрядностью представления, предлагается использовать обработку в весовом пространстве Соболева W_2^1 [2...5], которое задается выражением

$$\|f(t)\|_W = (1-\alpha) \int_{-\infty}^{\infty} |f(t)|^2 dt + \alpha \int_{-\infty}^{\infty} |f'(t)|^2 dt,$$

где $f(t)$ - обрабатываемый сигнал, $f'(t)$ - его производная, α - весовой параметр пространства W_2^1 .

В соответствии с приведенным определением, для построения системы передачи предлагается использовать следующую схему

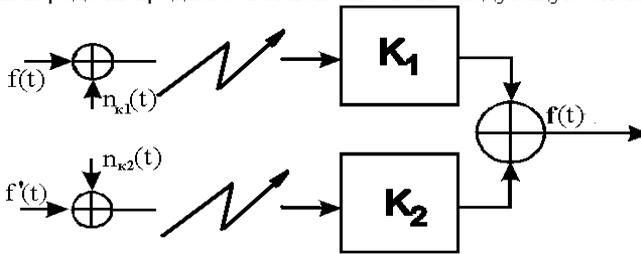


Рис. 1. Схема передачи цифрового сигнала в весовом пространстве Соболева W_2^1

где K_1 и K_2 фильтры обработки информации на приемной стороне. Как показано в [2...5], выражение для фильтров K_1 и K_2 в пространстве Соболева имеют вид:

$$K_1(\omega) = (1-\alpha) / (1-\alpha + \alpha\omega^2),$$

$$K_2(\omega) = j\omega\alpha / (1-\alpha + \alpha\omega^2).$$

Усредняя по реализациям шума квантования, получаем выражение для среднеквадратической ошибки представления сигнала на выходе системы:

$$\overline{\varepsilon^2(\omega)} = \frac{(1-\alpha)^2 P_{k1}(\omega) + \alpha^2 \omega^2 P_{k2}(\omega)}{(1-\alpha + \alpha\omega^2)^2}, \quad (1)$$

где $P_{k1}(\omega)$, $P_{k2}(\omega)$ - спектральная плотность шума квантования в каналах сигнала и производной соответственно.

Для определения оптимального значения α_{opt} , минимизирующего ошибку на выходе системы, возьмем производную выражения (1) по α и приравняем ее нулю. В результате получаем:

$$\alpha_{opt} = \frac{P_{k1}(\omega)}{P_{k1}(\omega) + P_{k2}(\omega)}.$$

Существуют определенные допущения, вводимые относительно шума квантования [6]:

- Шум не коррелирован с квантуемым сигналом;
- Любые два отсчёта последовательности не коррелированы, то есть шум квантования является процессом типа «белый шум»;
- Распределение вероятности ошибок квантования является равномерным по диапазону ошибок квантования.

Учитывая данные допущения, можно записать $MP_{k1}(\omega) = P_{k2}(\omega)$ что существенно упрощает выражение для оптимального значения α :

$$\alpha_{opt} = \frac{1}{1 + M}.$$

Рассмотрим применение предложенной системы передачи телеметрических сигналов. В качестве тестового сигнала использовалась сумма синусоид различной частоты и амплитуды.

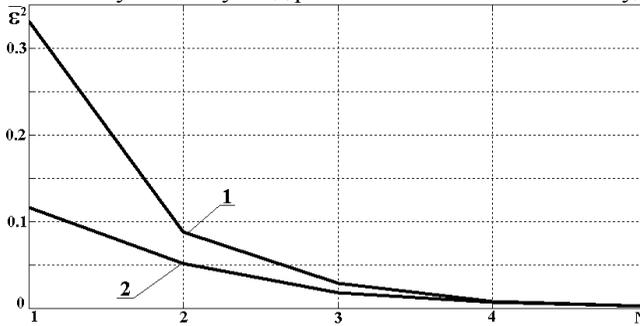


Рис. 2. Зависимость среднеквадратической ошибки ε^2 восстановления тестового сигнала от разрядности квантования 2^N

На рисунке 2 приведена зависимость среднеквадратической ошибки ε^2 восстановления тестового сигнала от разрядности квантования 2^N . Кривая 1 соответствует квантованному сигналу в пространстве L_2 , пропущенному через низкочастотный фильтр,

кривая 2 – обработке квантованного сигнала по предложенной схеме в весовом пространстве Соболева W_2^1 .

Из анализа рисунка 2 следует, что разработанный алгоритм позволяет снизить ошибку квантования сигнала примерно в три раза. При анализе рисунка 2 следует учитывать, что обработка сигнала в весовом пространстве Соболева позволяет в два раза снизить частоту дискретизации сигнала в каждом канале и, таким образом, общее число отсчетов при передаче сигналов остается неизменным.

Работа выполнена в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009-2013 годы (ГК 02.740.11.0470, ГК 16.740.11.0269, ГК .14.740.11.0326), а также при поддержке гранта Президента Российской Федерации для государственной поддержки молодых российских ученых МК-4899.2011.9 (договор №16.120.11.4899-МК).

СПИСОК ИСПОЛЪЗУЕМЫХ ИСТОЧНИКОВ

1. Котельников В.А. О пропускной способности «эфира» и проволоки в электросвязи. // Материалы к 1 Всесоюзному съезду по вопросам реконструкции дела связи и развития слаботочной промышленности. М.: Управление связи РККА, 1933. (Репринт Успехи физических наук, 176:7 (2006), С.762–770)
2. Кириллов С.Н., Бузыкканов С.Н. Алгоритм дискретного спектрального анализа сигналов в модифицированном пространстве Соболева // Автометрия. 2003. №1. С.88-94.
3. Бузыкканов С.Н., Кириллов С.Н. Вычисление спектра сигналов в модифицированном пространстве Соболева на основе быстрого преобразования Фурье // Автометрия. 2006. 42, №4. С.48-56.
4. Кириллов С.Н., Бузыкканов С.Н. Оценка спектральной плотности мощности сигналов в модифицированном пространстве Соболева // Радиотехника. 2002. 45, №12. С.46-51. (Известия высших учебных заведений)
5. Кириллов С.Н., Бузыкканов С.Н. Алгоритм восстановления аналогового сигнала в модифицированном пространстве Соболева // Автометрия. 2005. 41, №2. С.75-80.
6. Гольденберг Л.М., Матюшкин Б.Д., Поляк М.Н. Цифровая обработка сигналов: Учеб. пособие для вузов. М.: Радио и связь, 1990. 256 с.

А.Н. ВАРНАВСКИЙ

Рязанский государственный радиотехнический университет

ПРИМЕНЕНИЕ НЕЙРОСЕТЕВОЙ ОБРАБОТКИ БИМЕДИЦИНСКИХ СИГНАЛОВ ДЛЯ ВЫЯВЛЕНИЯ РАЗЛИЧНЫХ ПАТОЛОГИЙ

Для определения состояния человека и выявления различных патологий необходим анализ биомедицинских сигналов, который может осуществляться с помощью нейронных сетей. Поэтому актуальной является задача развития способов нейросетевой обработки биомедицинских сигналов.

Одним из наиболее важных биомедицинских сигналов является электрокардиосигнал (ЭКС). Нейронные сети можно использовать для обнаружения различных видов нарушений сердечного ритма. Рассмотрим способ определения водителя ритма сердца с помощью нейронной сети.

Для определения водителя ритма необходимо проанализировать взаимосвязь возбудений предсердий и желудочков в каждом кардиоцикле ЭКС, то есть определить, какой отдел сердца возбуждается первым [1].

Для повышения достоверности автоматического определения вида водителя ритма сердца в режиме реального времени предлагается использовать двухслойную нейронную сеть (рисунок 1), входными сигналами для которой являются сигнал опорной точки на TP-сегменте s_{OT} и результаты сравнений с пороговыми уровнями сигналов преобразований ЭКС при выделении положительных $y^{(-P)}$ и отрицательных $y^{(P)}$ зубцов P, обычных неизмененных $y^{(QRS)}$ и расширенных деформированных $y^{(QRS*)}$ QRS-комплексов:

$g=1..6$ – номера нейронов 0-ого слоя;

$l=1..5$ – номера нейронов 1-ого слоя;

$n=1..5$ – номера нейронов 2-ого слоя;

номера n можно поставить в соответствие с видом ритма: $n=1$ соответствует синусовому ритму, $n=2$ – предсердному ритму, $n=3$ – ритму из АВ-соединения, $n=4$ – желудочковому ритму;

ω_g – входной сигнал g нейрона 0-ого слоя, причем

$$\omega_{j+1} = y^{(j)}, \omega_6 = s_{OT};$$

z_l – выходное значение l нейрона 1-ого слоя;

обратная связь между нейронами $HU_1^{(1)}$ и $HU_1^{(0)}$ позволяет учесть сравнение моментов появления сигналов $y^{(-P)}$ и $y^{(P)}$ относительно $y^{(QRS)}$, $\omega_1 = z_1$;

Ψ_n - выходное значение n нейрона 2-ого слоя (логический сигнал n -ого вида водителя ритма);

Σ_l, Σ_n - суммы взвешенных входных сигналов, подаваемых на входы l нейрона 1-ого слоя и n нейрона 2-ого слоя соответственно;

$F_l(\Sigma_l), F_n(\Sigma_n)$ - пороговые функции активации l нейрона 1-ого слоя и n нейрона 2-ого слоя соответственно, $\theta_l^{(1)}, \theta_n^{(2)}$ - значения порогов l нейрона 1-ого слоя и n нейрона 2-ого слоя соответственно.

Σ_l и z_l определяются следующим образом:

$$\Sigma_l = \sum_{r=1}^6 w_{r,l}^{(1)} \cdot \omega_r = \sum_{r=2}^6 w_{r,l}^{(1)} \cdot \omega_r + w_{1,1}^{(1)} \cdot z_1 \text{ и}$$

$$z_l = F_l(\Sigma_l) = \begin{cases} 1, & \text{если } \left(\sum_{r=2}^6 w_{r,l}^{(1)} \cdot \omega_r + w_{1,1}^{(1)} \cdot z_1 \right) \geq \theta_l^{(1)}, \\ 0 & \text{иначе.} \end{cases}$$

Σ_n и Ψ_n определяются следующим образом:

$$\Sigma_n = \sum_{r=1}^5 w_{r,n}^{(2)} \cdot z_n \text{ и } \Psi_n = F_n(\Sigma_n) = \begin{cases} 1, & \text{если } \left(\sum_{r=1}^5 w_{r,n}^{(2)} \cdot z_n \right) \geq \theta_n^{(2)}, \\ 0 & \text{иначе.} \end{cases}$$

Полученные в результате обучения сети значения порогов равны

$$\theta^{(1)} = \begin{pmatrix} 2 \\ 2 \\ 1 \\ 2 \\ 2 \\ 1 \end{pmatrix} \text{ и } \theta^{(2)} = \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \end{pmatrix},$$

а значения весовых коэффициентов $w^{(1)}$ и $w^{(2)}$ представлены в виде матриц [2]:

$$w^{(1)} = \begin{pmatrix} w_{1,1}^{(1)} & w_{1,2}^{(1)} & w_{1,3}^{(1)} & w_{1,4}^{(1)} \\ w_{2,1}^{(1)} & w_{2,2}^{(1)} & w_{2,3}^{(1)} & w_{2,4}^{(1)} \\ w_{3,1}^{(1)} & w_{3,2}^{(1)} & w_{3,3}^{(1)} & w_{3,4}^{(1)} \\ w_{4,1}^{(1)} & w_{4,2}^{(1)} & w_{4,3}^{(1)} & w_{4,4}^{(1)} \\ w_{5,1}^{(1)} & w_{5,2}^{(1)} & w_{5,3}^{(1)} & w_{5,4}^{(1)} \\ w_{6,1}^{(1)} & w_{6,2}^{(1)} & w_{6,3}^{(1)} & w_{6,4}^{(1)} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & -1 & 0 & 0 \\ -1 & 1 & -1 & 0 \\ -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix},$$

$$w^{(2)} = \begin{pmatrix} w_{1,1}^{(2)} & w_{1,2}^{(2)} & w_{1,3}^{(2)} & w_{1,4}^{(2)} \\ w_{2,1}^{(2)} & w_{2,2}^{(2)} & w_{2,3}^{(2)} & w_{2,4}^{(2)} \\ w_{3,1}^{(2)} & w_{3,2}^{(2)} & w_{3,3}^{(2)} & w_{3,4}^{(2)} \\ w_{4,1}^{(2)} & w_{4,2}^{(2)} & w_{4,3}^{(2)} & w_{4,4}^{(2)} \\ w_{5,1}^{(2)} & w_{5,2}^{(2)} & w_{5,3}^{(2)} & w_{5,4}^{(2)} \end{pmatrix} = \begin{pmatrix} 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Таким образом, двухслойный персептрон можно использовать для анализа ЭКС с целью определения вида водителя ритма. Предполагается использовать такие нейросетевые структуры для анализа других биомедицинских сигналов в режиме реального времени.

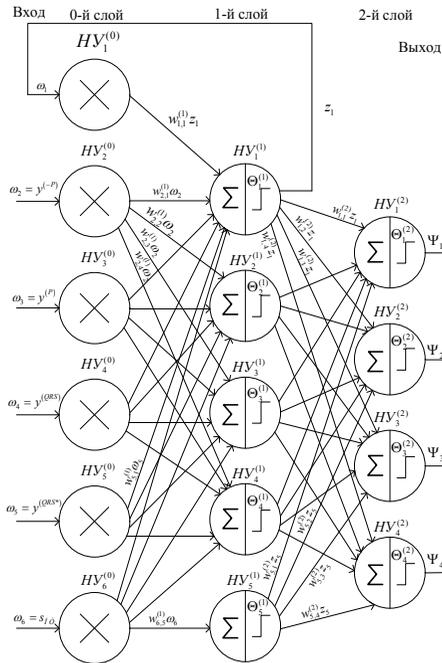


Рис. 1. Двухслойный персептрон для определения вида водителя ритма сердца

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Мурашко В.В., Стругинский А.В. Электрокардиография: Учеб. пособие. М.: ООО «МЕДпресс». 1998. 313с.
2. Варнавский А.Н. Применение двухслойной нейронной сети для классификации вида несинусового ритма //Физика полупроводников. Микроэлектроника. Радиоэлектронные устройства. Межвуз. сб. научных трудов. Рязань. 2010. С. 43-47.

М.Е. ВИНОГРАДОВА

Рязанский государственный радиотехнический университет

ПРОЦЕДУРА ПАРАМЕТРИЗАЦИИ АКУСТИЧЕСКИХ ПОМЕХ ДЛЯ РЕШЕНИЯ ЗАДАЧ КЛАССИФИКАЦИИ

Представлена процедура параметризации акустических помех, позволяющая осуществить классификацию акустических помех от двигателей автомобилей различных типов и обеспечивающая вероятность правильной классификации типа помехи около 90%.

Введение. Системы распознавания образов находят все более широкое применение в различных отраслях науки и техники как при решении задач криминалистического характера, так и в системах передачи информации, робототехнике, медицине [0...0]. В области криминалистики на данный момент большинство работ посвящено решению задач автоматизированной идентификации личности говорящего, однако, не меньший интерес представляет описание акустической среды звукозаписи в целях определения условий получения фонограммы [0]. В системах передачи информации может найти применение система управления параметрами кодека для адаптации к определенному типу акустических помех (АП) при изменении качества речевого сигнала на выходе канала связи [0]. Для создания такой системы необходим алгоритм классификации АП.

Цель работы. Цель работы - обоснование алгоритма, позволяющего по акустической реализации шума автомобильного двигателя идентифицировать возможные тип и марку автомобиля.

Обоснование выбора способа преобработки. В современных системах распознавания речи и музыки при решении задач параметризации чаще всего используются спектральные распределения сигналов [0,0,0]. Один из подходов [0,0] состоит в разбиении частотного диапазона сигнала на полосы с учетом особенностей слухового восприятия человека [0], и получении энергии сигнала в каждом поддиапазоне. Более целесообразным видится применение энтропийного подхода к разбиению диапазона частот сигнала при решении задачи параметризации. Процедура параметризации включает следующие этапы:

1. Периодограммным методом, описанным в [0], производится оценка спектральной плотности мощности (СПМ) для каждого из сигналов АП тестовой выборки при помощи окна Хемминга для 50% наложения.

2. Определяется энтропия распределения СПМ сигналов тестовых

вой выборки [0] $h_i(x) = -\int p_i(x) \ln(p_i(x)) dx$, где $p_i(x)$ - плотность распределения вероятности мощности спектральной составляющей f_i .

Шум автомобильного двигателя присутствует в диапазоне частот 0 ... 22 кГц. Из анализа экспериментальных данных следует, что наиболее информативными являются составляющие в диапазоне частот 5 ... 22 кГц. Это обусловлено спецификой определяемых АП. Высоко-частотные составляющие соответствуют шумам вибрации кузова и деталей автомобиля и определяются общей формой автомобиля, типом кузова, качеством сборки и подгона деталей, тогда как низкочастотные составляющие соответствуют шумам, обусловленным рабочим процессом, а так же структурными шумами двигателя.

3. Производится оценка информативных полос Δf_i и определяется энергия сигнала в каждом из указанных поддиапазонов.

Обоснование структуры алгоритма классификации и экспериментальные исследования. В рамках работы была собрана база данных для 32 автомобилей (22 легковых и 10 грузовых), как отечественного, так и зарубежного производства. Для каждого автомобиля получено по девять акустических реализаций шума двигателя при трех положениях устройства звукозаписи внутри салона автомобиля и трех снаружи в формате моно, windows PCM, при частоте дискретизации 44100 Гц, разрядности квантования – 32 бит. Запись производилась при помощи диктофона Olympus LS-10.

Структурная схема предложенного алгоритма классификации приведена на рисунке 2.

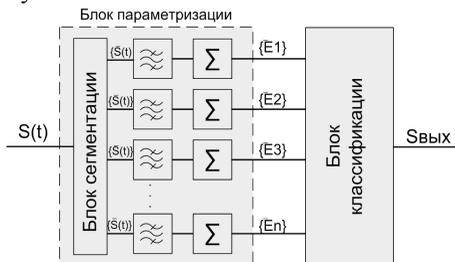


Рис. 1. Структурная схема алгоритма классификации АС

На вход системы классификации поступает АП $S(t)$, в блоке параметризации производится сегментация на кадры, фильтрация в полосах Δf_i и оценка энергии спектральной плотности мощности, содержащейся в различных полосах частот сегментов.

Блок классификации реализован на искусственной нейронной сети (ИНС). Экспериментально определено, что наименьшее значение ошибки на обучающей выборке дает использование трехслойной нейронной сети с числом нейронов первого и второго слоев $L_1 = L_2 = 2N + 1$, где N – число рассматриваемых классов автомобилей (легковой отечественный, легковой иномарка, грузовой, внедорожник, микроавтобус), с тангенциальными активационными функциями. В качестве алгоритма обучения был выбран OSS-метод (OSS - Ones Step Secant Algorithm), основанный на квазиньютоновской оптимизации целевой функции. Топология ИНС представлена на рисунке 2.

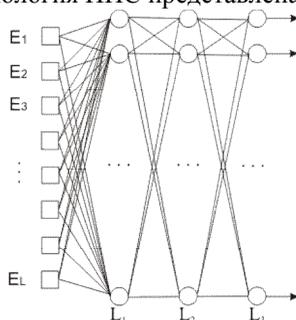


Рис. 2. Топология ИНС

Выводы. В результате экспериментальных исследований установлено, что вероятность правильной классификации при разделении автомобилей на пять классов при указанном подходе составляет около 90%.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Аграновский А.В., Леднов Д.А. Теоретические аспекты алгоритмов обработки и классификации акустических сигналов, М.: Радио и связь, 2004. — 164 с.
2. X. Huang, A. Acero, H. Hon. Spoken language processing: a guide to theory, algorithm, and system development. — Prentice Hall PTR, 2001. — P. 936.
3. Стукалов Д.Н. Алгоритмы цифровой обработки речевых сигналов при воздействии акустических помех, дис. Кандидата технических наук, Рязань 1996
4. А.П. Ефимов и др. Акустика: справочник/ под ред. М.А. Сапожкова. М.: Радио и связь, 1989. — 336 с.
5. С.Л. Марпл-мл., Цифровой спектральный анализ и его приложения – М.: Мир, 1990., - 584 с.

6. К.Шеннон, Работы по теории информации и кибернетике, Связь при наличии шума, пер. под редакцией Харкевича, 1963 г. М.: издательство иностранной литературы, 824 с.

И.В. ГОРЮНОВ

Рязанский государственный радиотехнический университет

СОВЕРШЕНСТВОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ ВУЗОМ НА ОСНОВЕ СБАЛАНСИРОВАННОЙ СИСТЕМЫ ПОКАЗАТЕЛЕЙ

Рассматриваются вопросы, связанные с применением сбалансированной системы показателей в вузе. Отражены проблемы, с которыми сталкиваются образовательные учреждения при разработке вузовской системы показателей.

В настоящее время особую актуальность приобретает задача повышения качества подготовки выпускников образовательных учреждений (ОУ) высшего профессионального образования (ВПО).

Одной из важнейших проблем многих государственных вузов на современном этапе является несоответствие применяемых структур и методов управления, необходимость постоянного развития и улучшения количественных и качественных показателей работы структурных подразделений ОУ, что в конечном итоге способствует усилению позиций конкретного вуза в условиях нестабильной и агрессивной внешней среды.

Разрабатываемые в вузах системы менеджмента качества являются интегрированными и ориентированными на современные информационные технологии. Их эффективность оценивается с применением качественных и количественных показателей.

Используемые в вузах показатели можно разделить на группы, представленные в таблице 1.

Из таблицы 1 видно, что деятельность современного вуза характеризуется большим количеством различных показателей. Это вызывает значительные трудности их сбора, обработки, хранения и анализа.

Таблица 1. Основные группы показателей

Группа	Характеристика
Статистические показатели	Характеризуют состояние и цели в сфере развития образовательного процесса (число обучающихся различных направлений и специальностей; количество студентов, окончивших курс с определенным баллом и т.п.)

Результаты образовательной деятельности	Количество выпускников, продолживших образование в магистратурах и аспирантурах; процент устроившихся на работу в течение полугода после выпуска, и т.п.
Показатели удовлетворенности различных групп потребителей уровнем качества образования	Характеризуют уровень удовлетворенности потребителей, оцениваемый на основе опросов и анкетирования как внутри вуза, так и за его пределами
Результаты научно-исследовательской деятельности	Характеризуют уровень качества научно-исследовательской деятельности (показатель объемов внешнего финансирования, получаемого в результате победы в грантах и конкурсах, а также показатели количества монографий и других публикаций)
Связь с местным сообществом	Характеризуются количеством мероприятий, проведенных для местного сообщества и ролью университета в жизни общества
Показатели международной деятельности	Количество и доля иностранных студентов; число международных исследовательских проектов и международного финансирования, субъективная оценка вуза международным академическим сообществом
Рейтинг вузов	Характеризует позицию вузов на национальном и международном уровне
Показатели обеспеченности ресурсами	Библиотечное и информационное обслуживание, оборудование аудиторий и лабораторий, развитие инфраструктуры и т.п.
Показатели экономической эффективности	Снижение отдельных видов затрат, показатели финансовой устойчивости, диверсификация внебюджетных источников доходов и т.п.
Социальные показатели	Характеризуют социальную поддержку потребителей (количество и доля студентов из определенных социальных или национальных групп, доступ к образованию студентов с ограниченными возможностями и т.п.)
Показатели выполнения нормативных требований	Характеризуют уровень выполнения требований внешних и внутренних нормативных документов (система оценки качества управления, внутренних правил и т.п.)

В результате могут возникнуть следующие проблемы:

1. Привлечение большого количества преподавателей, сотрудников и представителей администрации к работе с показателями.
2. Несогласованность показателей между собой по уровням управления вузом.

3. Чрезмерное количество модификаций показателей.
4. Дублирование показателей вместо их агрегации, трансформации и интеграции на более высоких уровнях.
5. Применение большинства показателей только для контроля и формальной отчетности.
6. Частое возникновение ситуации, когда одни и те же показатели определяются несколько раз разными подразделениями университета.
7. Высокий уровень нагрузки при формировании и сборе показателей на кафедры и отделы.
8. Отсутствие информационной поддержки при сборе и анализе показателей.

Анализ показателей должен быть направлен на принятие управленческих решений, способствующих разработке и реализации стратегии развития вуза.

Из множества различных методов управления, позволяющих вузам реализовать свою стратегию, т.е. трансформировать стратегические цели и задачи в конкретные действия, наиболее эффективным является сбалансированная система показателей (ССП – Balanced ScoreCard, BSC).

Сбалансированная система показателей является инструментом стратегического управления, который позволяет связать стратегические цели с оперативными действиями, позволяющими реализовывать стратегию; учесть как финансовые, так и нефинансовые показатели, что необходимо для оценки деятельности университета, связанной с нематериальными активами и информацией; своевременно реагировать на изменения в бизнес-процессах посредством различия показателей, которые измеряют достигнутые результаты, и показателей, которые отображают процессы по достижению этих результатов.

Суть методологии сбалансированной системы показателей заключается в том, что она переводит миссию и общую стратегию вуза в систему четко поставленных целей и задач, а также показателей, определяющих степень их достижения с точки зрения четырех перспектив «Финансы», «Клиенты», «Внутренние процессы», «Развитие и обучение» (рисунок 1).

Разработка СПП включает в себя пять укрупненных этапов:

1. Определение стратегических целей
2. Построение «стратегической карты»
3. Выбор показателей
4. Определение целевых значений
5. Разработка стратегических мероприятий

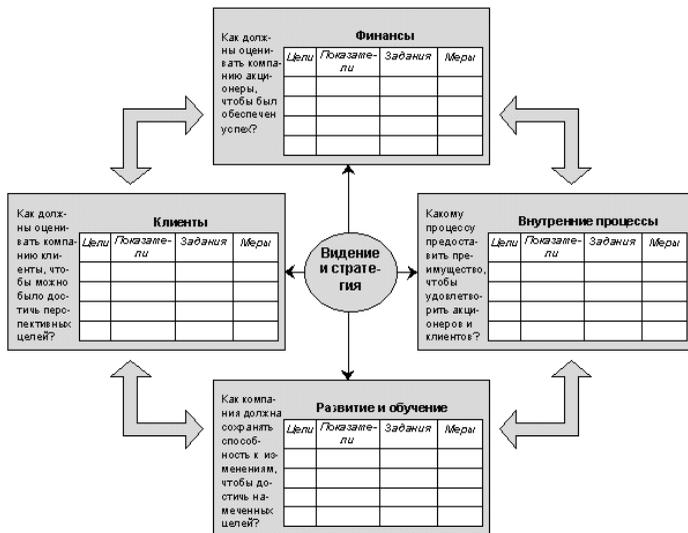


Рис. 1. Сбалансированная система показателей

Каждый этап основан на результатах предыдущего этапа, что позволяет преобразовать стратегию в форму конкретных показателей оперативного управления; трансформировать и развернуть стратегию по всем уровням организационной структуры управления вузом; наладить обратную связь с целью корректировки или пересмотра стратегии с учетом требований внешней среды. При этом необходимо осуществлять информационную поддержку сбора, обработки, хранения и анализа показателей.

Е.В. ГРИЦАЙ, Е.М. ДОНДИК, Н.Е. СКОРОБОГАТОВА
Рязанский государственный радиотехнический университет

СИСТЕМА РАСПОЗНАВАНИЯ ЗНАКОВ ДАКТИЛЬНОЙ РЕЧИ

Рассматривается система распознавания знаков дактильной речи, использующая контурное представление знаков.

Распространение информационных технологий в окружающей повседневной жизни ставит вопрос оказания помощи людям с дефектами слуха и речи. Они испытывают значительные трудности в общении с окружающими, в том числе при посещении поликлиник, учреждений социальной сферы и т.д., а кроме того существуют и проблемы

обучения таких детей дактильной и жестовой речи в детских садах и школах.

Пока из-за сложностей в настоящее время остается не решенной проблема автоматического распознавания жестовой речи. Поэтому задача распознавания дактильной речи является в наши дни весьма актуальной.

Общий алгоритм функционирования системы распознавания дактильных знаков включает ряд последовательных процедур таких как селекция знаков, выделения и кодирование контуров, формирование базы вектор-контуров обучающей выборки, формирование вектор-контура исследуемого знака, реализация процедуры распознавания знака, отображение результата распознавания дактилямы.

Каждый контур анализируемого знака Γ кодируется набором элементарных векторов $\gamma(i)$ $i=0,1,\dots,k-1$, где k – составляет число элементарных контуров $\Gamma = \{\gamma(i)\}_{k-1}$.

Таким образом, комплекснозначный код контура $\Gamma = \{\gamma(i)\}_{k-1}$ будет представлен на растровой плоскости изображения последовательностью элементарных векторов как функции от дискретного комплексного переменного. Первый элементарный вектор $\gamma(0)$ выходит из начальной точки 0 и направлен в ближайший из 8 пикселей с комплексной координатой $(\pm 1 \pm j)$. Следующий элементарный вектор $\gamma(1)$ принимает за начало отсчета конец вектора $\gamma(0)$ и имеет комплексное направление $(1+0j)$. Замкнутый вектор-контур Γ имеет вид:

$$\Gamma = \{ \gamma(0), \dots, \gamma(i), \gamma(i+1) \dots, \gamma(k-1) \}, \quad (1)$$

Так что последний элементарный вектор $\gamma(k-1)$ заканчивается, в точке 0. Принимается по умолчанию, что кодирование контура осуществляется по часовой стрелке.

Так как дискретность комплексного переменного совпадает с шагом растрового представления изображений это существенно упрощает описание и переводит анализ контуров в формат векторного анализа. Для решения задач распознавания в системе должно соблюдаться условие замкнутости контуров. Это означает, что исследуемый контур Γ и эталонный контур V должны соблюдать условие равенства нулю суммы всех составляющих их элементарных векторов

$$\sum_{i=0}^{k-1} \gamma(i) = 0 \quad \text{и} \quad \sum_{i=0}^{k-1} v(i) = 0.$$

Такое представление контуров последовательностью комплексных чисел, задающих элементарные вектора, предоставляет возможность аналитической операций с контурами. Полное равенство исследуемого Γ и эталонного V векторов обеспечивается при их комплексном представлении

$$\Gamma = \{\gamma_1(i) + j\gamma_2(i)\} \text{ и } V = \{v_1(i) + jv_2(i)\}, \quad (3)$$

в виде равенств

$$\gamma_1(i) = v_1(i); \quad \gamma_2(i) = v_2(i), \quad i=0,1,\dots,k-1. \quad (4)$$

Однако в процессе распознавания обеспечить полную идентичность всей последовательности элементарных векторов затруднительно. Поэтому система распознавания знаков предусматривает возможность смещения начальной точки отсчета (начального пикселя) на d элементов контура. Новые смещенные элементарные вектора $\gamma^{(d)}$ получаются по формуле

$$\gamma^{(d)} = \gamma(i+d),$$

т.е. номера векторов сдвигаются.

При этом текущая векторная $\beta(i)$ сумма элементов кода $\gamma(i)$, на i -ом элементе кода контура определяется как

$$\beta(i) = \sum_{s=0}^i \gamma(s), \quad (5)$$

формирует другой суммарный код контура $B = \{\beta(i)\}_{k-1}$, при $i=0,1,\dots,k-1$.

Для замкнутого контура $\beta(0) = \gamma(0)$ и $\beta(k-1) = 0$.

Код контура Γ и текущий суммарный код B инвариантны к переносу знака дактилемы по полю изображения, что значительно упрощает решение задачи распознавания.

Кроме того использование в качестве информативных признаков замкнутых вектор-контуров позволяет выполнить распознавание при изменении масштаба знака дактилемы и его наклона. В качестве решающей процедуры распознавания в системе используется вычисление нормированного скалярного произведения вектор-контура анализируемой дактилемы Γ и вектор-контура V каждого из числа хранящихся в базе данных знаков.

Скалярное произведение двух векторов Γ и V , представляемые в виде (3), будет определяться соотношением

$$(\Gamma, V) = \sum_{i=0}^{k-1} [\gamma_1(i)v_1(i) + \gamma_2(i)v_2(i)] + j \sum_{i=0}^{k-1} [\gamma_2(i)v_1(i) - \gamma_1(i)v_2(i)] \quad (6)$$

и является для выбранного способа кодирования контуров комплексным числом.

Нормированное скалярное произведение также будет комплексной величиной

$$p = \frac{(\Gamma, V)}{\|\Gamma\| \|V\|}, \quad (7)$$

где $\|\Gamma\|$ и $\|V\|$ - нормы соответствующих вектор-контуров.

Модуль нормированного скалярного произведения и является мерой близости вектор-контуров распознаваемой и эталонной дактилемы. При необходимости сравнения векторов различающейся размерности возможно сравнение спектрального разложения вектор-контуров с введением дополнительных нулевых элементарных контуров для обеспечения корректного распознавания.

Предлагаемая система распознавания дактильной речи работает со статическими монохромными изображениями знаков дактилем. В основу ее заложены следующие алгоритмы:

- алгоритм «жука». «Жук» начинает движение с белой области по направлению к черной. Как только он попадает на черный элемент, он поворачивает налево и переходит к следующему элементу. Если этот элемент белый, то жук поворачивается направо, иначе - налево. Процедура повторяется до тех пор, пока жук не вернется в исходную точку. Координаты точек перехода с черного на белое и с белого на черное и описывают границу объекта;

- кодирование контура изображения элементарными векторами;

- сравнение изображений на основе нормированного скалярного произведения [1].

Пользователь загружает изображение при помощи кнопок меню «Файл»-«Открыть» (рис. 1).

Далее изображение загружается в программу для дальнейшей обработки. При нажатии на кнопку «Определить контур» происходит распознавание контура изображения (рис. 2).

Для определения буквы, соответствующей этому изображению, следует нажать на кнопку «Сравнить», которая сравнивает изображение с файлом библиотеки. Результатом данного действия будет показ на экране соответствующей буквы (рис. 3).

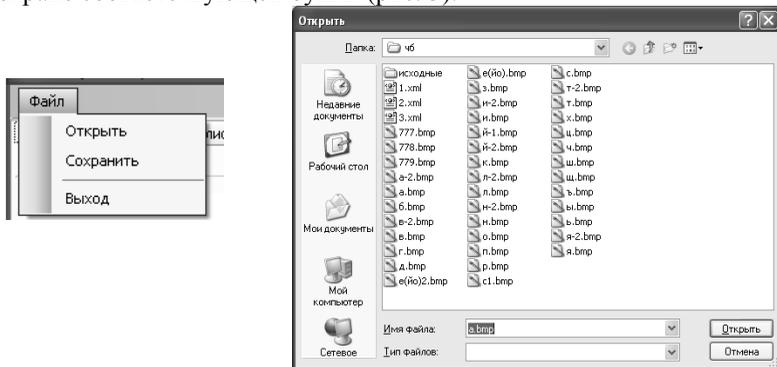


Рис. 1. Меню загрузки изображения

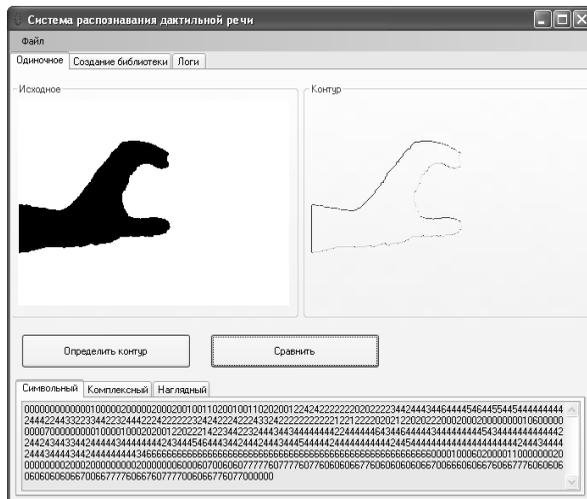


Рис. 2. Распознавание контура изображения

Программа имеет возможность создания библиотеки изображений и записи ее соответствующий файл (рис. 4).

Система разработана в Visual Studio 2008 на языке в C#.

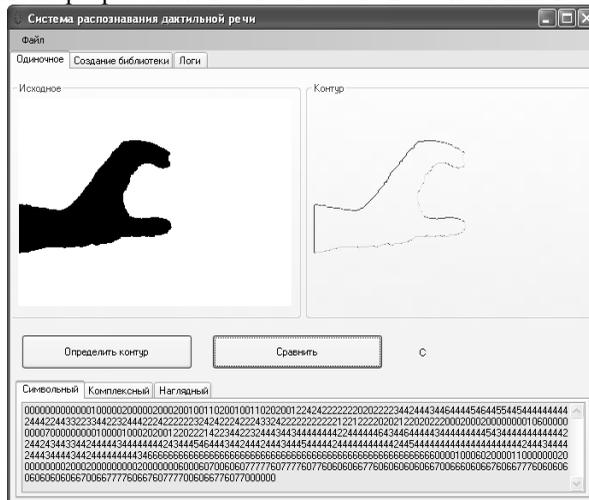


Рис. 3. Результат определения буквы

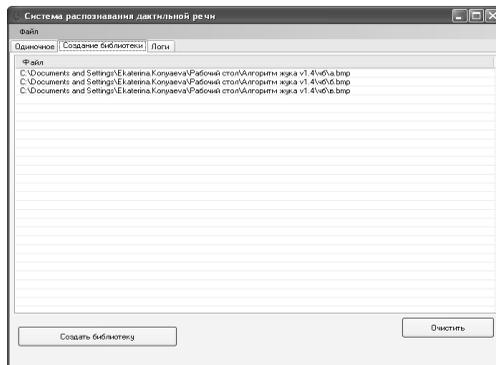


Рис. 4. Создание файла библиотеки

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Я.А.Фурман, А.В Кревецкий, А.К. Передреев, А.А. Роженов, Р.Г. Хафизов, И.Л Егошина, А.Н. Леухин; Под ред. Я.А. Фурмана Введение в контурный анализ; приложения к обработке изображений и сигналов/. — 2-е изд., испр. — М.: ФИЗМАТЛИТ, 2003. — 592 с. — ISBN 5-9221-0374-1.

2. Фурман А.Я. Спектральный анализ замкнутых полигональных контуров плоских изображений //Радиотехника.-1994.- №12.-С.41-44.

3. Furman Ya.A., Yashin V.V. Extraction and Linear Filtering of Closed Polygonal Contours of Images // Pattern Recognition and Image analysis.1994.-V.4, №2.-P.146-166.

В.Т. ДМИТРИЕВ, Я.О. КАРТАВЕНКО

Рязанский государственный радиотехнический университет

**ОБЪЕКТИВНАЯ ОЦЕНКА КАЧЕСТВА
ДЕКОДИРОВАННОГО РЕЧЕВОГО СИГНАЛА
НА ОСНОВЕ ИЗМЕНЕНИЯ СПЕКТРАЛЬНОЙ
ДИНАМИКИ КРИТИЧЕСКИХ ПОЛОС СПЕКТРА**

Предложен алгоритм объективной оценки качества речевого сигнала (РС) на выходе декодера на основе изменения спектральной динамики в пределах критических полос. Обоснованы параметры алгоритма и доказана высокая степень корреляции субъективной и объективной оценок качества речи.

Введение. Физический смысл фонетической функции, являющейся основой концепции восприятия речи, предложенной А.А. Пироговым [1..3] для оценки объективного качества речи, заключается в изменении спектральной динамики речи. Согласно этой концепции, фонемы отличаются друг от друга прежде всего изменениями спектрального распределения [2, 3]. Функция ощущения спектральной динамики (ФОСД) для дискретного сигнала имеет следующий вид:

$$P_H(f_n, N_{seg}) = P_2(f_n, N_{seg}) - P_1(f_n, N_{seg}) = \sum_{m=2}^M \sum_{n=0}^{N_{seg}} \lg \left(\frac{|Y(f_n, m \cdot N_{seg})|}{|Y(f_n, mN_{seg} - n)|} \right) - \sum_{m=2}^M \sum_{n=0}^{N_{seg}} \lg \left(\frac{|X(f_n, m \cdot N_{seg})|}{|X(f_n, mN_{seg} - n)|} \right), \quad (4)$$

где $X(f_n, mN_{seg})$ и $Y(f_n, mN_{seg})$ – спектры m -го сегмента с длиной N_{seg} исходного и декодированного РС, $X(f_n, mN_{seg} - n)$ и $Y(f_n, mN_{seg} - n)$ – спектры m -го сегмента с длиной N_{seg} , сдвинутого на величину n , исходного и декодированного РС, N_{seg} – количество отсчетов в сегменте РС, M – количество сегментов РС [2, 3].

Постановка задачи. Предложено оценивать изменение спектральной динамики РС в пределах критических частот:

$$MFOSD = \sum_{m=1}^M \sum_{g=1}^G \beta_g \sum_{n=1}^{N_{b,g}} P_H(f_n, N_{seg} m) = \sum_{m=1}^M \sum_{g=1}^G \beta_g \times \sum_{n=1}^{N_{b,g}} \left(\lg \left(\frac{|Y(f_n, N_{seg} m)|}{|Y(f_n, N_{seg} (m-1))|} \right) - \lg \left(\frac{|X(f_n, N_{seg} m)|}{|X(f_n, N_{seg} (m-1))|} \right) \right) \quad (5)$$

где β_g – коэффициент «значимости» спектральных составляющих, определяемый исходя из свойств человеческого слуха [4], $X(f_n, g, N_{seg} m)$ и $Y(f_n, g, N_{seg} m)$ – g -я критическая полоса спектра m -го сегмента с длиной N_{seg} исходного и декодированного РС, $N_{b,g}$ – количество отсчетов в g -ой критической полосе спектра РС, G – общее количество критических полос.

Экспериментальная часть. Для проверки точности оценки предложенной модификации ФОСД по сравнению с известными методами оценки качества РС для набора кодеров проводился эксперимент, структурная схема которого показана на рисунке 1.

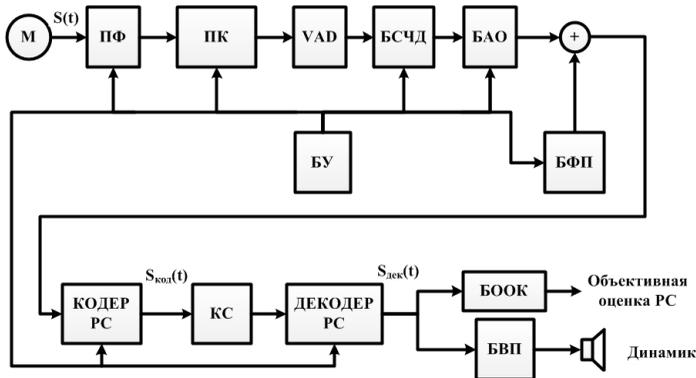


Рис. 1. Схема получения объективной оценки качества речи

На рисунке 1 используются следующие обозначения: М – микрофон, ПФ – полосовой фильтр, ПК – программный конвертор, VAD (Voice Activity Detector) – детектор речевой активности, БСЧД – блок сокращения частотного диапазона, БАО – блок амплитудного ограничения, БФП – блок формирования помехи, БУ – блок управления, КОДЕР РС – первичный кодек РС, КС – канал связи, БООК – блок объективной оценки качества РС, БВП – блок восстановления пауз.

Для сравнения точности исследуемых методов объективной оценки качества речи использовалась их корреляция с показателями субъективной оценки по разборчивости и узнаваемости, полученными согласно ГОСТ Р 50840-95 [5].

Исследования проводились для основных кодеров систем передачи: AMR-WB при скорости передачи 23,85 кбит/с, ICELP – 8 кбит/с, G729 – 8 кбит/с, G723.1 – 6,3 кбит/с, MMBE – 2,4 кбит/с.

Полученные в результате эксперимента коэффициенты корреляции объективной и субъективной оценок качества представлены в таблице 1. Темным цветом в таблице выделены максимальные коэффициенты корреляции.

Таблица 1. Результаты эксперимента

Кодеки Методы	AMR-WB		ICELP		G729		G723.1		MMBE	
	Разб	Узн	Разб	Узн	Разб	Узн	Разб	Узн	Разб	Узн
BSD	0,77	0,69	0,89	0,80	0,85	0,73	0,97	0,66	0,52	0,98
CD	0,01	0,84	0,02	0,86	0,03	0,76	0,02	0,81	0,03	0,86
COSH	0,36	0,43	0,87	0,87	0,60	0,58	0,37	0,88	0,57	0,18
ESC	0,08	0,16	0,92	0,90	0,87	0,78	0,92	0,83	0,70	0,92
ISD	0,63	0,60	0,49	0,52	0,62	0,59	0,73	0,14	0,20	0,86
MBSD	0,73	0,63	0,81	0,73	0,88	0,77	0,97	0,61	0,47	0,98
PESQ	0,12	0,14	0,98	0,92	0,87	0,76	0,93	0,83	0,46	0,92
SD	0,76	0,84	0,95	0,86	0,87	0,76	0,97	0,81	0,84	0,86
CKO	0,76	0,68	0,04	0,10	0,72	0,60	0,94	0,50	0,50	0,98
SNR	0,81	0,77	0,10	0,16	0,75	0,63	0,92	0,49	0,50	0,98
SNR _{segm}	0,40	0,23	0,50	0,37	0,37	0,26	0,26	0,33	0,49	0,39
SP	0,28	0,13	0,95	0,88	0,74	0,72	0,74	0,16	0,58	0,98
MFOSD	0,84	0,82	0,95	0,96	0,87	0,81	0,94	0,89	0,87	0,89

Как следует из таблицы, построенной на основе экспериментальных исследований, предложенный алгоритм в большинстве случаев обеспечивает наилучшие коэффициенты корреляции для всех рассмотренных кодеков РС.

Заключение. Показано, что предложенный алгоритм оценки качества на основе изменения спектральной динамики критических полос спектра обеспечивает коэффициенты корреляции по разборчивости и узнаваемости не ниже 0,8 для рассмотренных алгоритмов кодирования РС.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Соболев В.Н. Информационные технологии в синтетической акустике. Монография – М.: ИРИАС, 2007 – 360 с.
2. Пирогов А. А. Синтетическая телефония.— М.: Связьиздат, 1963. – 120 с.
3. Вокодерная телефония. Под ред. А.А. Пирогова - М.: Связь, 1974. – 536 с.
4. Цвикер Э., Фельдкеллер Р. Ухо как приемник информации. 2-е изд., перераб. и доп. - М.: Связь, 1971. - 255с.
5. ГОСТ Р 50840-95. Передача речи по трактам связи. М.: Госстандарт России, 1995 – 180с.

Е.М. ДОНДИК, Н.Е. СКОРОБОГАТОВА, А.С. ХВАТОВА
Рязанский государственный радиотехнический университет

СИСТЕМА РАСПОЗНАВАНИЯ ЗНАКОВ ДАКТИЛЬНОЙ РЕЧИ НА ОСНОВЕ ВЕЕРА ВЕКТОРОВ

Рассматривается система с представлением дактильных знаков веером векторов, которые для распознавания сравниваются с эталонными в формате согласованной пространственной фильтрации

Количество людей, использующих в качестве средства коммуникации жестовую и дактильную речь достаточно велико. На учете во Всероссийском обществе глухих состоят на учете более 230 тысяч человек. Такой речью пользуются постоянно до трех процентов населения и если учесть тех, кто вынужден с ними общаться в семье и других местах, то число людей, использующих этот язык, увеличится.

Большинство нормально слышащих людей не знают языка жестов, а число переводчиков катастрофически мало, поэтому глухие и слабослышащие испытывают большие трудности в общении, особенно при посещении любых учреждений социальной сферы, поликлиник, магазинов и т.д.

Одним из путей помощи таким людям в общении с окружающим миром является создание систем перевода жестового языка.

И если недавно разработанная система перевода обычного текста и речи в жестовую или дактильную позволяет донести информацию для аудитории слабо слышащих, то обратное направление перевода жестовой и дактильной речи в общепринятую, доступную окружающим, пока не решено.

И если перевод жестовой речи затруднителен из-за необходимости использовать систему многомерного отслеживания и распознавания динамических образов, что требует мощных вычислительных и, возможно, стационарных систем, то перевод дактильной речи выглядит доступнее. Однако пока остается неизвестным её практическое решение и существует проблема перевода дактильной речи в обычную, хотя бы в побуквенном произношении, в реальном или замедленном варианте. А потребность в этом возникает у каждого глухого и слабослышащего, причем, как сугубо индивидуальная проблема - при необходимости сказать что-либо врачу, продавцу в магазине или служащему в учреждении.

Поэтому возникает потребность в разработке таких систем перевода, на первых порах хотя бы только дактильной речи. Тем более, что рассматриваемая система может быть представлена в качестве прило-

жения к сотовым телефонам, которые функционально готовы к использованию подобной распознающей системы, т.к. обладают функцией видеозаписи изображений демонстрируемых дактильных знаков и имеют возможность воспроизводить речь[1].

Методы и алгоритмы, положенные в основу и опробованные в системе распознавания дактильной речи, возможно, смогут быть использованы в дальнейшем для разработки более сложной системы распознавания жестовой речи.

Сложность построения системы перевода дактильных знаков заключается в особенности демонстрации (дактилирования) самой речи.

Во-первых, дактильная речь в соответствии с правилами дактилирования, должна быть непрерывной, плавной с непрерывными переходами от одного демонстрируемого знака к другому, что требует от системы отслеживания всех переходных промежуточных образов и выбора единственного информационного знака.

Во-вторых, дактильная речь может демонстрироваться на различном расстоянии от видеокамеры и людьми с различными размерами руки. Распознающая система должна быть инвариантна к масштабу изображения дактилемы.

В-третьих, положение руки при демонстрации знака не фиксировано в поле кадра и может смещаться и наклоняться в зависимости от привычки демонстрирующего дактилемы, поэтому распознающая система должна быть инвариантна к смещению и повороту распознаваемого знака.

Эти особенности приводят к необходимости разработки системы распознавания со специфическими функциональными свойствами, отличающими их от обычных распознающих систем. Прежде всего, система должна обеспечить селекцию самого распознаваемого знака из непрерывной последовательности промежуточных образов и только после этого соотнести его с одним из счетного множества эталонных дактильных знаков.

Распознающая часть системы использует контурное выделение знаков [2] с последующим представлением их веером векторов. Под веером векторов подразумевается упорядоченная совокупность радиус-векторов, проведенных с постоянным шагом по углу к контуру изображения, как приведено на рисунке 1.

Видеопоследовательность цветных кадров дактильной речи преобразуется в подсистеме селекции в бинарные изображения с последующим выделением контуров в форме комплекснозначных элементарных контурных векторов[1,2].

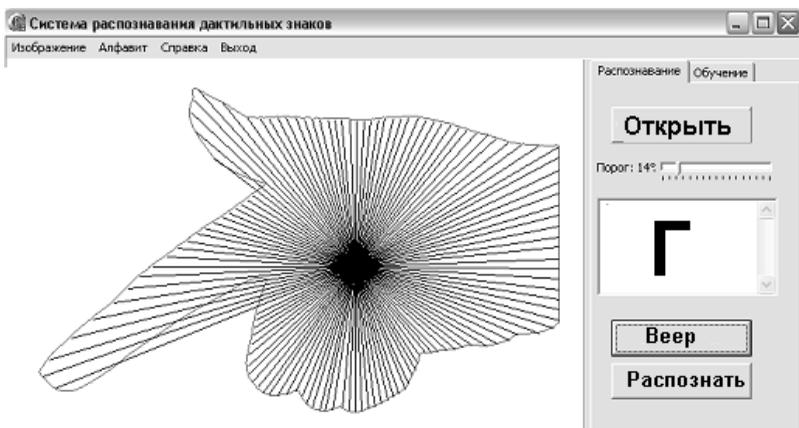


Рис. 1. Представление дактильных знаков веером векторов в подсистеме распознавания

Для обеспечения возможности распознавания дактильных знаков необходимо выделить их информативную часть, сосредоточенную в ладони руки. Отделение неинформативной части кадра изображения предполагает две последовательные процедуры: определение направления руки в кадре и отсечение ладони на уровне минимальной ширины. Таким образом из матрицы данных удаляются все неинформативные элементы и т.к. контур знака должен быть замкнутым, достраивается прямая линия на уровне среза.

При построении веера следует иметь в виду, что радиус-вектора могут пересекать контур в нескольких местах, поэтому в качестве считываемой выбирается наиболее удаленная точка пересечения, как наиболее чувствительная к вариациям знаков.

Выбор числа векторов веера k определяется спектральным составом линии контура и, в частности, изрезанностью и скоростями перепада между его крайними значениями. Выполнив разложение по спектру контура наиболее сложного знака, можно определить минимальный шаг по контуру для его адекватного представления векторами веера. Для спектральных разложений и других операций преобразования с векторами необходимо обеспечить совмещение полюса векторов с центром тяжести образа знака.

Распознавание основано на сравнении представляемого веером векторов дактильного знака, полученного из видеопоследовательности после соответствующих преобразований, с полным набором эталонных знаков. При этом веера эталонных знаков представляются в формате согласованных фильтров [3], обеспечивающих попарно ком-

плексную сопряженность с распознаваемыми векторами. В результате действительная составляющая их векторной суммы позволяет оценить меру схожести эталона с распознаваемой дактилемой.

Импульсная характеристика фильтра для каждого вектора представляется в виде $\lambda(n) = l^*(k - n - 1)$ и дает при подаче на вход неизвестного произвольного веера $L^r \{l^*(n)\}_{0,(k-1)}$ отклик в виде

$$\eta(m) = \sum_{n=0}^{k-1} l^r(n) l^*(n - m + k - 1).$$

При окончании цикла обработки веера, когда текущий момент принимает значение $m=k-1$, на выходе фильтра образуется величина, равная скалярному произведению эталонного веера векторов в комплексно-сопряженном виде $L^* \{l^*(n)\}_{0,(k-1)}$ и распознаваемого веера $L^r \{l^r(n)\}$, что дает

$$\eta(k-1) = \sum_{n=0}^{k-1} l^r(n) l^*(n)$$

В случае совпадения обрабатываемого веера векторов с эталонным $L^r=L$ выходная величина соответствующего пространственного согласованного фильтра равна энергии веера

$$\eta(k-1) = \sum_{n=0}^{k-1} l(n) l^*(n) = \|L\|^2$$

Это возможно при полной компенсации значений аргументов всех векторов и вычислению суммы квадратов их модулей. Модуль выходного отсчета позволяет оценить независимо от размеров векторов меру схожести $\xi(m)$ фильтруемого веера с эталонным веером, если сравнивать нормированные значения

$$\xi(m) = \frac{|\eta(m)|}{\|L^r\| \|L\|}$$

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Дондик Е.М., Скоробогатова Н.Е. Система распознавания и воспроизведения знаков дактильной речи // Сб. науч. трудов «Инновационные подходы к применению информационных технологий в профессиональной деятельности», НАЧОУ ВПО СГА.- Белгород: ГиК, 2010.-С.287-291.

2. Фурман Я.А. Согласованная фильтрация контуров изображений // Радиотехника.-1995.-№6.-С.30-33

3. Furman Ya.A., Yanshin V.V. Extraction and Linear Filtering of Closed Polygonal Contours of Images // Pattern Recognition and Image Analysis.-1994.-V.4, №2.

М.А. КОЗЛОВ, С.В. СКВОРЦОВ

Рязанский государственный радиотехнический университет

ГИБРИДНЫЙ АЛГОРИТМ СОРТИРОВКИ МАССИВОВ ДЛЯ МНОГОЯДЕРНЫХ ПРОЦЕССОРОВ

Рассматривается распараллеливание задачи сортировки данных на многоядерных процессорах. Приводятся результаты многопоточной реализации и исследования разработанной программы сортировки массивов.

В связи с появлением многоядерных процессоров и систем на их базе становится актуальной задача определения классов задач, которые возможно эффективно обрабатывать на таком оборудовании. Фундаментальные алгоритмы уже сейчас нуждаются в доработке под современные вычислительные мощности, причем требуется создание новых методик работы с данными, которые будут оптимизированы под многопоточность. Одним из подходов к решению этой задачи является использование гибридных алгоритмов, учитывающих особенности как однопоточной, так и параллельной реализации [1].

Разработку гибридного алгоритма покажем на примере решения задачи сортировки, которая относится к фундаментальным алгоритмам обработки и манипулирования с данными. Известно много последовательных алгоритмов сортировки [2], однако их комбинация может давать интересные результаты с точки зрения их параллельной реализации. Учитывая широкое распространение многоядерных процессоров, целесообразно рассмотреть многопоточную реализацию алгоритмов сортировки и сравнить ее с однопоточным вариантом.

Чтобы эффективно использовать ресурсы многоядерных систем воспользуемся методом "разделяй и властвуй" [1]. Для этого следует разбить выполняемый алгоритм на независимые части, которые можно назначить разным системным потокам.

Для решения задачи сортировки массивов воспользуемся известными алгоритмами сортировки простыми вставками и слиянием, выбор которых обусловлен следующими причинами:

- они являются устойчивыми;
- сортировка простыми вставками имеет оценку сложности $O(n^2)$, но является самой быстрой среди прямых методов для малых размеров сортируемых массивов;
- сортировка слиянием имеет линейную вычислительную сложность $O(n \log n)$.

Предлагаемый комбинированный (гибридный) алгоритм включает следующие шаги.

Шаг 1. Разбиение входной последовательности на части фиксированного размера m .

Шаг 2. Организация нескольких параллельных потоков T_1, T_2, \dots, T_k , количество которых равно числу ядер процессора k , и сортировка отдельных частей независимо друг от друга с использованием алгоритма простых вставок.

Шаг 3. Слияние отсортированных частей массива длиной m элементов в последовательном (однопоточном) режиме.

На рис. 1 условно показана схема выполнения двух первых шагов алгоритма, порядок слияния отсортированных частей (третий шаг) показан на рис. 2.

Алгоритм реализован на языке Java, в котором есть поддержка многопоточности. При создании потоков учитывалось, что Java создает тяжеловесные (системные) потоки [3]. Исследование предложенного алгоритма проводилось на двухядерной системе.

В процессе эксперимента оценивалось время исполнения разработанного гибридного алгоритма и двух последовательных алгоритмов сортировки - слиянием и простыми вставками. Рассматривались массивы достаточно большой длины ($n = 1000, 10000, 100000, 1000000$), заполненные случайным образом. Сравнение соответствующих программ выполнялось по средним значениям времени работы, полученным по 10 опытам для каждого n .

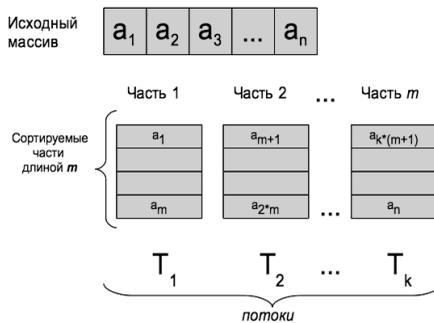


Рис. 1. Разбиение исходного массива на части и их независимая сортировка

Результаты эксперимента представлены на рис. 3 и позволяют сделать следующие выводы. В среднем прирост скорости для числа сортируемых элементов от 100000 до 1000000 составил 18-20%, на меньших размерах массивов - только 7-12% по сравнению с сортировкой слиянием. Когда число элементов массива было больше 100000, время сортировки вставками не фиксировалось, так как оно было

слишком большое. Заметим, что во всех случаях $m = 512$, причем эта величина была подобрана экспериментально с учетом наибольшего выигрыша в скорости сортировки.

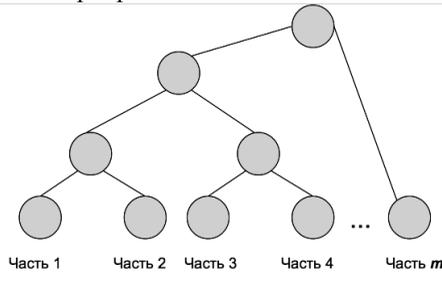


Рис. 2. Порядок слияния отсортированных частей

Таким образом на основе полученных данных можно полагать, что многопоточная реализация алгоритма гибридной сортировки, основанного на известных алгоритмах сортировки слиянием и простыми вставками, позволяет добиться увеличения быстродействия на больших объемах данных, а на малых объемах дает не худшие результаты. При этом гибридный алгоритм также обладает свойством устойчивости, что важно при решении практических задач.

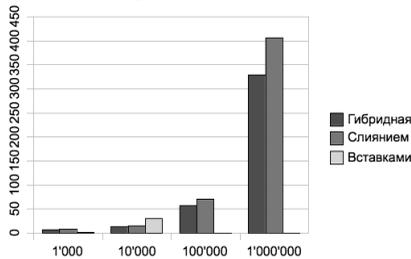


Рис. 3. Зависимость времени сортировки (мс) от размера массива

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Миллер Р., Боксер Л. Последовательные и параллельные алгоритмы: Общий подход. – М.: БИНОМ. Лаборатория знаний, 2006. – 406 с.
2. Кормен Т., Лейзерсон Ч., Ривест Р., Шайтан К. Алгоритмы, построение и анализ. – М.: Издательский дом «Вильямс», 2010. – 1296 с.

З. Эккель Б. Философия Java. Библиотека программиста. - Спб.: Питер, 2010. - 640 с.

А.В. КОНДРАШКИН, И.П. ТРОФИМОВА

Рязанский областной фонд обязательного медицинского страхования
Рязанский государственный радиотехнический университет

**СИСТЕМА ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА
СТАЦИОНАРНОГО ОТДЕЛЕНИЯ БОЛЬНИЦЫ**

Рассматривается автоматизированная система, предназначенная для накопления и обработки информации о лечебной деятельности круглосуточного стационара больницы.

Анализ уровня компьютеризации медицинских учреждений Рязанской области показывает, что в практической работе различных служб больниц области в настоящее время применяется около 35 основных программ. В их числе такие программы, как «Российский государственный медико-дозиметрический регистр», «Льготный рецепт», «Учет оказания врачебной помощи населению в системе обязательного медицинского страхования», «Вакцинация», «Зарплата». Эти программные комплексы, в основном, направлены на решение вопросов, связанных с предоставлением реестров для финансирования лечебно-профилактического учреждения и не касаются вопросов накопления и обработки информации о лечебной деятельности больницы. Рассматриваемая система обеспечивает автоматизацию процессов документирования лечебного процесса в учреждениях здравоохранения.

Изучение документооборота, сопровождающего оформление больного в круглосуточный стационар, его лечение и выписку из стационара, позволило выявить наиболее слабые места, подлежащие автоматизации в первую очередь.

Так, например, в Муниципальных учреждениях здравоохранения отсутствует база данных (БД), содержащая полную достоверную информации о населении, прикрепленном к лечебно-профилактическому учреждению. В связи с этим в настоящее время отсутствует возможность оперативной проверки достоверности предоставляемых пациентом документов (полис, паспорт, СНИЛС) и оперативного обновления данных о застрахованном населении. В случаях ошибочных данных страховые компании и Рязанский областной фонд обязательного медицинского страхования не производят оплату лечения больного. Лечебное учреждение несет экономические потери.

Довольно часто возникает ситуация, когда необходимо оперативно получить информацию о больном из архива. В старых историях болезни содержатся сведения об аллергических реакциях пациента на те или иные лекарственные препараты, о перенесенных заболеваниях, травмах, оперативных вмешательствах, о группе крови и резус-факторе и т.п. В настоящее время работа с архивными данными занимает много времени и в экстренных случаях жизненно важна для пациента информация оказывается недоступной. Создание электронной версии истории болезни позволило бы в сжатые сроки получать необходимые сведения.

Работа любого медицинского учреждения сопровождается формированием определенных отчетов, составляемых врачами и статистическим кабинетом учреждения. Традиционно такие документы готовятся вручную, отнимая у сотрудников много времени и требуя повышенного внимания. Появление ошибок в документах практически неизбежно. Необходимость автоматизации этой рутинной работы очевидна.

На текущем этапе разработки системы созданы локальная БД и пользовательское приложение, автоматизирующее документооборот хирургического отделения больницы. БД содержит всю справочную информацию, необходимую для документирования лечебного процесса, в том числе и информацию поликлинического отделения. Это сведения о пациентах и лечащих врачах стационара, данные страховых компаний, классификатор травм, справочник возможных исходов лечения, классификатор диагнозов, относящихся к травмам, справочник диагнозов согласно международной классификации болезней. Последний справочник содержит только те диагнозы, которые утверждены фондом медицинского страхования для данного медицинского учреждения.

Система обеспечивает автоматизированный учет больных, создание электронной истории болезни, формирование статистической карты выбывшего из стационара и всех необходимых отчетов о работе отделения.

Интерфейс системы построен так, чтобы максимально освободить пользователя от необходимости ручного ввода информации в процессе ведения БД, а также при формировании различного рода документов.

Клиентское приложение системы основано на экранных формах и обеспечивает документальное сопровождение всех этапов работы хирургического отделения больницы:

- поступление больного в стационар;

- внесение сведений о лечении больного в отделении;
- выписка больного из отделения;
- составление отчетов.

При поступлении нового больного в стационар вводятся сведения о больном, и создается электронная история болезни пациента, которая хранится в формате Microsoft Word. Для предотвращения неполного ввода данных возможность сохранения информации предоставляется лишь в том случае, когда окажутся заполненными все необходимые поля формы. Предусмотрена возможность использования уже имеющихся в БД сведений о пациенте.

На этапе прохождения лечения для любого больного в историю болезни вводятся все необходимые сведения, а именно: данные анализов, дневники осмотров, температурный лист. Здесь же заносятся сведения о диагнозе больного. При вводе и редактировании данных истории болезни предусмотрена возможность использования информации, хранящейся в справочниках БД отделения.

На этапе выписки больного из отделения окончательно формируется электронная версия истории болезни согласно форме № 003, утвержденной Минздравом РФ. Все недостающие данные, необходимые для оформления соответствующей документации, добавляются автоматически. На этом же этапе генерируется статистическая карта выбывшего из стационара.

Электронные версии истории болезни сохраняются в архиве и в экстренных случаях вся необходимая информация о больном предоставляется в режиме on-line.

На всех этапах прохождения лечения предусмотрена возможность просмотра и поиска необходимых данных, а также выполняется контроль дублирования данных о больном, дубликаты записей удаляются.

В системе предусмотрена процедура импорта в БД сведений, предоставляемых страховыми компаниями и Рязанским областным фондом обязательного медицинского страхования. Эти данные предоставляются в виде файла LPU.dbf. В системе данные этого файла не просто копируются в БД лечебно-профилактического учреждения, заменяя предыдущую версию, а с помощью специальной процедуры обновляют изменившиеся данные или добавляют записи о новых клиентах. В тех случаях, когда в предоставляемом файле содержатся сведения о клиенте, имеющем несколько страховых полисов, генерируется соответствующий список. Данные об устаревших страховых полисах удаляются вручную.

В системе предусмотрены процедуры автоматического формирования следующих отчетов, предоставляемых лечебным учреждением в Управление здравоохранения области:

- "Структура выписанных из стационара больных". Отчет отражает информацию о структуре контингента, лечившегося в стационаре;

- "Годовой отчет по травме" (форма №57, утверждена постановлением Госкомстата РФ от 29.06.99 №49), отражающий информацию о структуре травм пациентов отделения;

- "Статистическая карта выбывшего из стационара" (форма №066), отражающая информацию о случае лечения пациента отделения.

Созданная система защиты приложения обеспечивает конфиденциальность информации о пациентах, ограничивает доступ к приложению работникам других отделений больницы, предотвращает распространение приложения без разрешения разработчика системы и руководства больницы.

Для разработанной системы средствами пакета Help & Manual 4.5 создано полнофункциональное руководство пользователя в виде стандартного файла классической справки WinHelp, который располагает функциями удобного перехода к выбранному разделу справки с помощью гиперссылок. Руководство пользователя встроено в АИС и открывается при нажатии клавиши помощи «F1», причем для каждой экранной формы автоматически открывается свой раздел справки. Также существует возможность использования файла справки в автономном режиме.

Разработанная система успешно прошла испытание и используется в работе Муниципального учреждения здравоохранения "Захаровская центральная районная больница".

Д.А. КОРЯЧКО, Р.В. ТИШКИН

ФГУП «ГНПРКЦ «ЦСКБ-Прогресс» - филиал «ОКБ «Спектр»

ГРАФОВЫЕ МОДЕЛИ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ

Анализируется ряд подходов решения задачи сегментации изображений, основанных на использовании методов теории графов.

Интеллектуальная обработка изображений (таких как аэрокосмические, изображения быстротекущих процессов и т.д.) невозможна без качественного решения задачи сегментации изображений.

Существует множество подходов к решению задачи сегментации, основанных на структурных методах (морфологические методы, граничные методы, модели активного контура, динамические и вероятностные деформируемые модели и др.) и стохастических методах (пороговая классификация, метод классификаторов, кластерный анализ и др.). В то же время существует множество задач сегментации, успешное решение которых возможно с использованием аппарата теории графов.

Проанализируем ряд подходов к решению задачи сегментации изображений, основанных на использовании методов теории графов.

1. Метод нормализованного разреза. В работе [1] предложен метод нормализованного разреза (Normalized Cut), основанный на нахождении разрезов в графе по специальным критериям. Задача сегментации сводится к задаче разбиения графа на непересекающиеся подграфы. В случае разбиения изображения на 2 сегмента графовая интерпретация сводится к следующей модели.

В графе $G=(V,E)$ множество вершин V можно разбить на два непересекающихся подмножества A и B , $A \cup B=V$, $A \cap B=\emptyset$, простым удалением ребер, соединяющих две части. В результате получается два подграфа $G_1=(A,E_1)$ и $G_2=(B,E_2)$, каждый из которых может претендовать на организацию самостоятельного сегмента. Если каждому ребру $[x,y] \in E$ графа G приписать вес (или стоимость) $w(x,y)$, то степень сходства между подграфами $G_1=(A,E_1)$ и $G_2=(B,E_2)$ может быть вычислена как общий вес (стоимость) ребер, входящих в разрез

$$c(A,B) = \sum_{x \in A, y \in B} \omega(x,y). \quad (1)$$

Величину $\omega(x,y)$ можно интегрировать как степень близости (или коэффициент корреляции) между вершинами x и y .

Оптимальным делением графа на два подграфа является то, которое имеет минимальную стоимость разреза. Задача поиска минимального разреза графа хорошо изучена [2], существует множество эффективных алгоритмов ее решения. Хотя есть экспоненциальное количество таких разрезов, нахождение минимального разреза графа, хорошо изученная проблема и существуют эффективные алгоритмы ее решения.

Метод нормализованного разреза предусматривает помимо вычисления стоимости общего веса ребер, соединяющих два подграфа,

вычисление стоимости как долю всех ребер, соединяющих узлы графа $G=(V,E)$.

$$Nc(A,B) = \frac{C(A,B)}{S(A,V)} + \frac{C(A,B)}{S(B,V)}, \quad (2)$$

где $S(A,V) = \sum_{u \in A, t \in V} w(u,t)$ - стоимость всех связей от узлов подграфа $G_1 = (A, E_1)$ к остальным узлам графа $G=(V,E)$. $S(B,V)$ - стоимость всех связей от узлов подграфа $G_2 = (B, E_2)$ к остальным узлам графа $G=(V,E)$.

При таком определении диссоциации между подграфами, разрез, который разделяет маленькие изолированные узлы, не будет иметь малую стоимость нормализованного разреза. В случае, показанном на рис. 1, видно, что стоимость разрезов P_1 и P_2 в узлах n_1 и n_2 будет сто процентной от общей связи вершин графа G с этим узлом.

Так же можно определить значение всех нормированных связей внутри групп (подграфов):

$$Na(A,B) = \frac{S(A,A)}{S(A,V)} + \frac{S(B,B)}{S(B,V)}, \quad (3)$$

где $S(A,A)$ и $S(B,B)$ - соответственно общая стоимость ребер, соединяющих узлы в подграфах $G_1 = (A, E_1)$ и $G_2 = (B, E_2)$.

Использование двух частных критериев при разбиении графа на два подграфа, один из которых минимизирует диссоциацию между подграфами, а второй максимизирует ассоциацию внутри подграфов практически идентична и при решении практических задач могут быть использованы независимо друг от друга.

2. Метод вложенных разрезов [5]. Основная идея метода вложенных разрезов (Nested Cuts) заключается в разделении каждого пикселя P от других соседних пикселей по критерию минимальной стоимости.

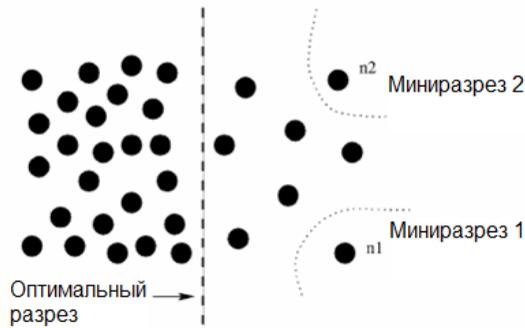


Рис. 1. Иллюстрация определения минимального разреза

Для решения задачи строится неориентированный взвешенный граф $G=(V,E)$, где V – множество пикселей и E – множество ребер, соединяющих соседние пиксели, в соответствии с заданными свойствами соседства. Вес $w(p,q)$ ребра $[p,q]$, соединяющего вершины p и q графа G , определяет сходство (корреляцию) между пикселями p и q . Задача разрезания графа $G=(V,E)$ заключается в разбиении множества вершин V на непересекающиеся подмножества V_1, \dots, V_k таким образом, чтобы суммарный вес между пикселями, входящими в сегменты V_i и V_j был бы минимальным. Результат реализации метода проиллюстрирован на рис.2.

На рис. 2а показана исходная ситуация, на которой видно, что пиксель q может принадлежать как сегменту A , так и сегменту B . В результате работы алгоритма оказалось, что пиксель q по критерию минимального веса принадлежит сегменту B .

3. Последовательный метод сегментации изображений с нечеткими границами двумерных объектов. Метод основан на вычислении разрезов графа с целью уточнения границ сегментов, полученных в результате сегментации графа поиском водоразделов [3].

Исходными данными для алгоритма являются изображение и результат его сегментации с неточно обнаруженными границами. Начальная сегментация представляется в виде отдельного изображения, пиксели которого имеют значения от 1 до N , где N - число полученных сегментов, а каждому сегменту соответствует свое число.

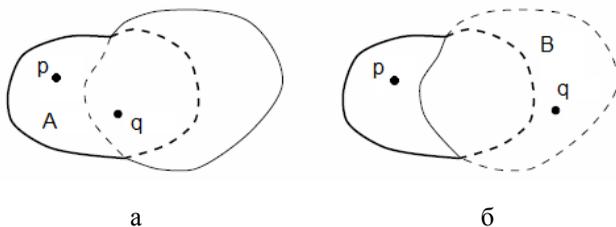


Рис. 2. Иллюстрация метода вложенных разрезов

После сегментации величина отклонения границ сегментов от искомым границ объектов зависит от извилистости этих границ. Для уточнения границ обычно используется алгоритм Зонга-Суня [4]. Этот алгоритм позволяет определить коридоры сразу между всеми сегментами изображения.

Для уточнения границ сегментов можно использовать алгоритм для построения планарного сеточного графа.

Планарный сеточный граф строится следующим образом. Определяется исток S и сток T графа. Кроме истока S и стока T в граф добавляются вершины, соответствующие пикселям, участвующим в построении графа. Каждая вершина соединяется ребрами с соседними, как показано на рис. 3.

Вес ребра W_{ij} между соседними вершинами i и j , соответствующих этим вершинам и расстоянию между ними. После определения всех W_{ij} определяется минимальный разрез, который можно определить методами поиска максимального потока в сети [2]. В результате выполнения этого алгоритма большая часть пикселей, отнесется к какому-нибудь сегменту. В случае необходимости можно провести постобработку неразмеченных пикселей с целью корректировки результатов сегментации.

Следует отметить еще одно направление работ по сегментации изображений, отраженное в работах [6,7]. В этих работах предлагается изображение объекта представлять в виде графа, вершинами которого являются углы пикселей, а ребрами – стороны пикселей. С помощью функции градиента или любого другого оператора обнаружения границ каждому ребру присваивается вес. Те ребра, которые разделяют схожие по интенсивности пиксели, имеют большее весовое значение, тогда как “граничные” ребра – маленькое весовое значение. Таким образом, проблема нахождения оптимальной границы сводится к по-

иску пути наименьшего веса. Вес ребра, разделяющего два пикселя p и q , определяется как

$$C(p, q) = I_{\max} - |I(p) - I(q)|, \quad (4)$$

где $I(p)$, $I(q)$ – значение интенсивностей в точках p и q соответственно, I_{\max} – максимальное значение интенсивности на данном изображении.

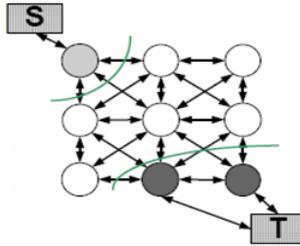


Рис. 3. Планарный сеточный граф

Основным преимуществом данного метода является то, что он хорошо работает даже при нечетком разделении между сегментируемыми областями. Но при этом метод требует представления изображений в виде графа.

В заключение следует отметить следующее. Структурные методы сегментации используют информацию о структуре сегментируемого объекта. Стохастические методы основаны на статистическом анализе данных. Графовые методы позволяют осуществлять процедуру сегментации на базе информации, которая может быть представлена функциями веса или стоимости между соседними вершинами графа, которые идентифицируются пикселями исследуемых изображений.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. J. Shi and J. Malik. Normalized cuts and image segmentation. IEEE Transaction on pattern analysis and machine intelligence, v. 22, №8, 2000, p.p. 888-905. Е.В. Заугольнова, Д.В. Юрин. ВМиК, Московский государственный университет, Москва, Россия. ФГУП НИИ ОПТЭКС, Зеленоград, Россия
2. Форд Фалкерсон. Поток в сетях. Изд. Мир.
3. Е.В. Заугольнова, Д.В. Юрин. Алгоритм уточнения предварительной сегментации изображений с нечеткими слабоконтрастными границами двумерных объектов. International Conference Graphicon 2006, Novosibirsk, Akademgorodok, 2006.

4. T.Y. Zhang, C.Y. Suen. A fast parallel Algorithm for Thinning Digital Patterns // Commun. acm, v. 27 №3, pp339-344.
5. Olga Veksler. Image Segmentation by Nested Cuts // IEEE Computer Vision and Pattern Recognition, June 2000 (CVPR'00), p.339-344.
6. R.M. Haralick, L.G. Sapiro. Image segmentation Techniques. Computer Vision, Graphics and Image Processing, 29, p. 100-132, 1985.
7. N.R. Pal, S.K. Pal. A Review on Image Segmentation Techniques. Pattern Recognition, 26(9), p. 1277-1294, 1993.

Е.М. КУЗЬМИНА, Ю.А. СОСУЛИН

Рязанский государственный радиотехнический университет

ОЦЕНИВАНИЕ ПАРАМЕТРОВ РЕГРЕССИОННЫХ МОДЕЛЕЙ ОБЪЕКТОВ ИЕРАРХИЧЕСКОЙ СТРУКТУРЫ

Рассматриваются вопросы построения регрессионных моделей технологических процессов и многоагрегатных объектов в случае возможности их описания в виде структурных моделей

Современные технологические процессы представляют собой сложные объекты, выходные переменные которых зависят от множества факторов. Большинство объектов исследования не могут быть описаны как детерминированные вследствие воздействия на них большого количества неконтролируемых, а часто вообще неизвестных факторов, влияние которых на выходные величины невозможно учесть. Для исследования подобных объектов применяются статистические методы, состоящие в описании объектов исследования линейными по параметрам регрессионными моделями с использованием множественной регрессии.

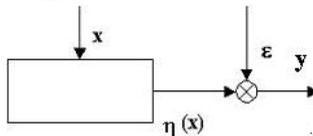


Рис. 1. Общая структура объекта

Если вид модели объекта исследования известен, процедура построения модели сводится к определению оценок неизвестных параметров одним из методов оценивания. Например, эта задача может быть решена обработкой экспериментальных данных по методу наименьших квадратов (МНК) для получения оценок коэффициентов регрессионной модели заданного вида.

При решении задач идентификации сложных статистических объектов или технологических процессов обычно используется приведенная схема объекта, где $\mathbf{x} = (x_1, x_2, \dots, x_p)^T$ – вектор входных факторов объекта; y – случайная выходная величина; $\boldsymbol{\varepsilon}$ – случайная ошибка. Линейная регрессионная модель подобного статистического объекта, получаемая на основании экспериментальных данных, традиционно представляется в виде стохастической зависимости выходной величины объекта от его входных факторов:

$$\mathbf{y} = \eta(\mathbf{x}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}, \quad (1)$$

где $\mathbf{y} = (y_1, y_2, \dots, y_N)^T$ – вектор значений зависимой переменной, N – количество наблюдений, $\mathbf{x} = (x_1, x_2, \dots, x_m)^T$ – вектор входных факторов; η – известная функция взаимосвязи; $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N)$ – вектор случайных ошибок с нулевым математическим ожиданием и дисперсией; $\boldsymbol{\theta} = (\theta_0, \theta_1, \dots, \theta_k)^T$ – вектор неизвестных коэффициентов (параметров) модели, оценки которых необходимо получить. Выбор метода построения математической модели определяется наличием у экспериментатора априорной информации о модели.

Модель в матричной форме имеет вид:

$$\mathbf{y} = \mathbf{F}\boldsymbol{\theta} + \boldsymbol{\varepsilon}, \quad (2)$$

где $\mathbf{y} = (y_1, y_2, \dots, y_N)^T$ – вектор наблюдений зависимой величины; \mathbf{F} – матрица значений функций – регрессоров от входных факторов \mathbf{x} , состоящая из N строк вида $\mathbf{F} = (\mathbf{f}_0(\mathbf{x}), \mathbf{f}_1(\mathbf{x}), \dots, \mathbf{f}_k(\mathbf{x}))$; $\boldsymbol{\varepsilon}$ – вектор случайных ошибок.

МНК – оценки вектора $\hat{\boldsymbol{\theta}}$ могут быть получены в соответствии со следующим выражением [1]:

$$\hat{\boldsymbol{\theta}} = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \mathbf{y}. \quad (3)$$

Во многих случаях исследования реальных объектов решение задачи построения регрессионных моделей затруднено необходимостью определения значительного числа неизвестных параметров, существенно превышающего размерность вектора входных факторов. Количество наблюдений, необходимое для решения задач параметрической идентификации объекта, непосредственно связано с числом оцениваемых параметров k , входных факторов m и видом аппроксимирующего полинома. Увеличение степени полинома и количества входных факторов приводят к существенному росту требуемого для оценивания количества наблюдений, увеличению размерностей матриц, ис-

пользуемых для получения оценок, их плохой обусловленности вследствие возможной мультиколлинеарности входных факторов x . Отрицательные последствия мультиколлинеарности проявляются в падении точности оценивания неизвестных коэффициентов модели (1), чувствительности модели к выбору точек наблюдения [1]. В этих случаях возникает необходимость в использовании специальных алгоритмов оценивания для устранения указанных недостатков.

Иногда экспериментатор может располагать информацией о структурных особенностях реального объекта. Структурированный объект может быть представлен в виде совокупности блоков, различным образом связанных между собой промежуточными выходными величинами. Тогда оценивание общей модели может быть выполнено с использованием структурных регрессий, связывающих выходные величины блоков с соответствующими входными переменными.

Общим элементом структурированного объекта может быть соединение блоков в соответствии со схемой (рис. 2). В этом случае входом блока второго уровня будут являться выходы предыдущих блоков, а также множество факторов x_3 этого блока.

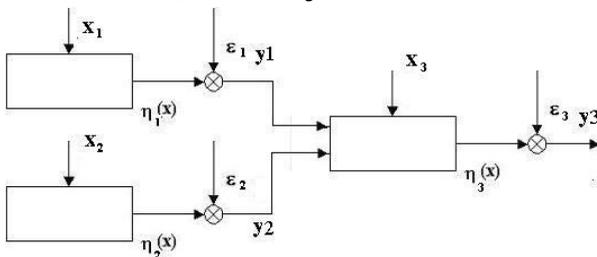


Рис. 2. Схема структурированного объекта

Например, можно представить объект как совокупность двух параллельных блоков первого уровня, которые описывают зависимость выходной величины от части независимых переменных $x_1 = (x_1, x_2, \dots, x_{m1})^T$, $x_2 = (x_{m1+1}, x_{m1+2}, \dots, x_{m2})^T$, $m1 + m2 \leq m$, найти их уравнения регрессии с использованием МНК, а затем полученные модели использовать в качестве входных факторов блока второго уровня. Значения таких входов могут быть вычислены по полученным уравнениям регрессии. Кроме того, в модель добавляются неучтенные на первом уровне входные переменные $x_3 = (x_{m1+m2+1}, x_{m1+m2+2}, \dots, x_m)^T$ и их взаимодействия в зависимости

от выбранного вида модели. Если принять, что выходы первого уровня входят в модель линейно, модель второго уровня примет вид

$$\begin{array}{ll} \text{В виде уравнений:} & \text{В матричной форме:} \\ y_1 = \eta_1(x_1, a_1) + \varepsilon_1, & y_1 = F_1 a_1 + \varepsilon_1 \\ y_2 = \eta_2(x_2, a_2) + \varepsilon_2, & y_2 = F_2 a_2 + \varepsilon_2 \\ y_3 = \eta_3(x_3, y_1, y_2, d) + \varepsilon_3 & y_3 = F_3 d + \varepsilon_3 \end{array} \quad (4)$$

где y_1, y_2 - значения выходных величин блоков первого уровня, y_3 - вектор выходной величины блока второго уровня, a_1 и a_2 - векторы коэффициентов уравнений регрессии 1 - го уровня, d - коэффициенты модели второго уровня, x_1, x_2, x_3 - векторы входных факторов первого и второго уровня, матрица F_3 содержит столбцы $f(y_1)$ и $f(y_2)$, соответствующие вхождению y_1 и y_2 в уравнение модели 2 уровня.

Коэффициенты a_1 , a_2 и d моделей системы (4) связаны с коэффициентами θ модели (1) линейным преобразованием:

$$\theta = A d, \quad (5)$$

где матрица A - матрица $((k+1)*t)$ линейного преобразования коэффициентов, сформированная на основании анализа взаимосвязи системы (2) и модели (1); t - размерность вектора d . Матрица A состоит из столбцов - коэффициентов моделей блоков первого уровня. Расположение коэффициентов внутри столбца соответствует правилу размещения входных факторов и их взаимодействий в уравнении регрессии для y_3 (4).

Значения величин y_1 и y_2 , входящие в матрицу F_3 уравнения модели блока второго уровня (4), обеспечиваются значениями факторов x_1 и x_2 первого уровня, которые определяются по уравнениям регрессии:

$$\begin{array}{l} \hat{y}_1 = F_1 \hat{a}_1, \\ \hat{y}_2 = F_2 \hat{a}_2 \end{array} \quad (6)$$

В этом случае найденные по уравнениям (6) значения всех факторов x_1 и x_2 будут обеспечивать уровни варьирования величин \hat{y}_1 и \hat{y}_2 , т.е. на втором уровне для оценивания d реализуется матрица \hat{F}_3 , содержащая столбцы оценок \hat{y}_1 и \hat{y}_2 . Если обозначить остатки структурных моделей первого уровня через e_1 и e_2

$$\begin{array}{l} e_1 = y_1 - \hat{y}_1, \\ e_2 = y_2 - \hat{y}_2, \end{array} \quad (7)$$

то обобщенная ошибка, приведенная к выходу блока второго уровня, определяется в соответствии с формулой:

$$\boldsymbol{\varepsilon}_{3об} = \boldsymbol{\varepsilon}_3 + \mathbf{e}_1 d_{y1} + \mathbf{e}_2 d_{y2}, \quad (8)$$

где d_{y1} и d_{y2} - коэффициенты при входных факторах y_1 и y_2 во второй структурной модели. Из формулы (8) следует, что ошибки моделей блоков первого уровня, умноженные на соответствующий коэффициент, проходят на выход блока второго уровня, действуя как добавочный шум.

Оценивание моделей первого уровня производится с использованием МНК, получаемые оценки являются несмещенными, состоятельными и обладают свойством эффективности. Можно показать, что математическое ожидание (МО) ошибки (8) равно нулю, а ковариационная матрица возмущений $\boldsymbol{\varepsilon}_{3об}$ выражается как

$$Cov\{\boldsymbol{\varepsilon}_{3об} \boldsymbol{\varepsilon}_{3об}^T\} = (\sigma_3^2 + \sigma_1^2 d_{y1} + \sigma_2^2 d_{y2}) I_N = \sigma_{3об}^2 I_N. \quad (9)$$

Докажем несмещенность оценок $\hat{\mathbf{d}}$ блока второго уровня. Для этого запишем выражения для оценок $\hat{\mathbf{d}}$:

$$\hat{\mathbf{d}} = (\hat{\mathbf{F}}_3^T \hat{\mathbf{F}}_3)^{-1} \hat{\mathbf{F}}_3^T \mathbf{y}_3, \quad (10)$$

$$\mathbf{y}_3 = \hat{\mathbf{F}}_3 \mathbf{d} + \boldsymbol{\varepsilon}_3 + \mathbf{e}_1 d_{y1} + \mathbf{e}_2 d_{y2}$$

$$\hat{\mathbf{d}} = \mathbf{d} + \Delta \mathbf{d}, \text{ где } \Delta \mathbf{d} = (\hat{\mathbf{F}}_3^T \hat{\mathbf{F}}_3)^{-1} \hat{\mathbf{F}}_3^T \boldsymbol{\varepsilon}_{3об} \quad (11)$$

Математическое ожидание оценок $\hat{\mathbf{d}}$

$$\begin{aligned} E\{\hat{\mathbf{d}}\} &= \mathbf{d} + E\{\Delta \mathbf{d}\} = \mathbf{d} + E\left\{(\hat{\mathbf{F}}_3^T \hat{\mathbf{F}}_3)^{-1} \hat{\mathbf{F}}_3^T \boldsymbol{\varepsilon}_{3об}\right\} = \mathbf{d} + \\ &+ E\left\{(\hat{\mathbf{F}}_3^T \hat{\mathbf{F}}_3)^{-1} \hat{\mathbf{F}}_3^T \mathbf{e}_1 d_{y1} + (\hat{\mathbf{F}}_3^T \hat{\mathbf{F}}_3)^{-1} \hat{\mathbf{F}}_3^T \mathbf{e}_2 d_{y2} + (\hat{\mathbf{F}}_3^T \hat{\mathbf{F}}_3)^{-1} \hat{\mathbf{F}}_3^T \boldsymbol{\varepsilon}_3\right\} = \\ &= \mathbf{d} + (\hat{\mathbf{F}}_3^T \hat{\mathbf{F}}_3)^{-1} \hat{\mathbf{F}}_3^T d_{y1} E\{\mathbf{e}_1\} + (\hat{\mathbf{F}}_3^T \hat{\mathbf{F}}_3)^{-1} \hat{\mathbf{F}}_3^T d_{y2} E\{\mathbf{e}_2\} + \\ &+ (\hat{\mathbf{F}}_3^T \hat{\mathbf{F}}_3)^{-1} \hat{\mathbf{F}}_3^T E\{\boldsymbol{\varepsilon}_3\} = \mathbf{d} \end{aligned} \quad (12)$$

При выводе последнего выражения использованы предположки МНК о равенстве нулю математических ожиданий ошибок.

Для доказательства несмещенности оценок, получаемых на выходе блока второго уровня пересчетом с использованием матрицы преобразования \mathbf{A} , воспользуемся тем, что МО случайной матрицы

есть неслучайная матрица с элементами, равными МО соответствующих им элементов случайной матрицы, т.е.

$$E\{\hat{\mathbf{A}}\} = \mathbf{A}; E\{\Delta\hat{\mathbf{A}}\} = \bar{\mathbf{0}}. \quad (13)$$

$$\hat{\boldsymbol{\theta}} = \hat{\mathbf{A}} \hat{\mathbf{d}} = (\mathbf{A} + \Delta\mathbf{A})(\mathbf{d} + \Delta\mathbf{d}) = \mathbf{A}\mathbf{d} + \Delta\mathbf{A}\mathbf{d} + \mathbf{A}\Delta\mathbf{d} + \Delta\mathbf{A}\Delta\mathbf{d} \quad (14)$$

$$E\{\hat{\boldsymbol{\theta}}\} = E\{\mathbf{A}\mathbf{d} + \Delta\mathbf{A}\mathbf{d} + \mathbf{A}\Delta\mathbf{d} + \Delta\mathbf{A}\Delta\mathbf{d}\} = E\{\mathbf{A}\mathbf{d}\} + E\{\Delta\mathbf{A}\mathbf{d}\} + E\{\mathbf{A}\Delta\mathbf{d}\} + E\{\Delta\mathbf{A}\Delta\mathbf{d}\} = \mathbf{A}\mathbf{d} + 0 + 0 + E\{\Delta\mathbf{A}\Delta\mathbf{d}\} \quad (15)$$

Оценки будут не смещенными при равенстве нулю последнего слагаемого в (15). Рассмотрим вид матрицы $\Delta\mathbf{A}$, представляющей собой блочную матрицу, в которой первый столбец состоит из нулей, второй содержит значения вектора $\Delta\mathbf{a}1$, третий – значения $\Delta\mathbf{a}2$, где $\Delta\mathbf{a}1$ и $\Delta\mathbf{a}2$ определяются выражениями

$$\Delta\mathbf{a}1 = \hat{\mathbf{a}}1 - \mathbf{a}1 = (\mathbf{F}_1^T \mathbf{F}_1)^{-1} \mathbf{F}_1^T \boldsymbol{\varepsilon}_1, \quad (16)$$

$$\Delta\mathbf{a}2 = \hat{\mathbf{a}}2 - \mathbf{a}2 = (\mathbf{F}_2^T \mathbf{F}_2)^{-1} \mathbf{F}_2^T \boldsymbol{\varepsilon}_2$$

Аналогично (16) определяется $\Delta\mathbf{d} = (\hat{\mathbf{F}}_3^T \hat{\mathbf{F}}_3)^{-1} \hat{\mathbf{F}}_3^T \boldsymbol{\varepsilon}_{3об}$. Тогда произведение $\Delta\mathbf{A}\Delta\mathbf{d}$ представляет собой вектор, содержащий в первых двух строках произведения $\Delta\mathbf{a}1\Delta d_{y1}$ и $\Delta\mathbf{a}2\Delta d_{y2}$, а в остальных строках – нули.

$$\Delta\mathbf{A}\Delta\mathbf{d} = \begin{bmatrix} \Delta\mathbf{a}1\Delta d_{y1} \\ \Delta\mathbf{a}2\Delta d_{y2} \\ \bar{\mathbf{0}} \end{bmatrix} \quad (17)$$

Здесь Δd_{y1} и Δd_{y2} - отклонения оценок коэффициентов структурной модели второго уровня системы при входных переменных $y1$ и $y2$ от их математических ожиданий.

Вычисление МО вектора (17) сводится к определению МО произведения векторов $\Delta\mathbf{a}1$ и $\Delta\mathbf{a}2$ на вектор $\Delta\mathbf{d}^T$, соответствующие компоненты которого представляют собой элементы МО вектора (17). Докажем равенство нулю МО первой строки (17). Используя (16), получим:

$$\begin{aligned}
E\{\Delta \mathbf{a} \Delta \mathbf{d}^T\} &= E\left\{\left(\mathbf{F}_1^T \mathbf{F}_1\right)^{-1} \mathbf{F}_1^T \boldsymbol{\varepsilon}_1 \varepsilon_{3ia} \dot{\hat{\mathbf{F}}}_3 \left(\hat{\mathbf{F}}_3^T \hat{\mathbf{F}}_3\right)^{-1}\right\} = \\
&= \left(\mathbf{F}_1^T \mathbf{F}_1\right)^{-1} \mathbf{F}_1^T E\left\{\boldsymbol{\varepsilon}_1 \varepsilon_{3ia} \dot{\hat{\mathbf{F}}}_3 \left(\hat{\mathbf{F}}_3^T \hat{\mathbf{F}}_3\right)^{-1}\right\} = \\
&= \mathbf{M}_1 E\left\{\boldsymbol{\varepsilon}_1 \varepsilon_3 \dot{\hat{\mathbf{F}}}_3 + \boldsymbol{\varepsilon}_1 d_{y1} \mathbf{e}_1^T + \boldsymbol{\varepsilon}_1 d_{y2} \mathbf{e}_2^T\right\} \mathbf{M}_3,
\end{aligned} \tag{18}$$

где $\mathbf{M}_1 = \left(\mathbf{F}_1^T \mathbf{F}_1\right)^{-1} \mathbf{F}_1^T$, $\mathbf{M}_3 = \hat{\mathbf{F}}_3 \left(\hat{\mathbf{F}}_3^T \hat{\mathbf{F}}_3\right)^{-1}$.

Выражение (18) раскладывается на сумму МО слагаемых:

$$\begin{aligned}
E\{\Delta \mathbf{a} \Delta \mathbf{d}^T\} &= \mathbf{M}_1 E\left\{\boldsymbol{\varepsilon}_1 \varepsilon_3 \dot{\hat{\mathbf{F}}}_3\right\} \mathbf{M}_3 + \mathbf{M}_1 d_{y1} E\left\{\boldsymbol{\varepsilon}_1 \mathbf{e}_1^T\right\} \mathbf{M}_3 + \\
&+ \mathbf{M}_1 d_{y2} E\left\{\boldsymbol{\varepsilon}_1 \mathbf{e}_2^T\right\} \mathbf{M}_3 = \mathbf{M}_1 d_{y1} E\left\{\boldsymbol{\varepsilon}_1 \mathbf{e}_1^T\right\} \mathbf{M}_3 + \mathbf{M}_1 d_{y2} E\left\{\boldsymbol{\varepsilon}_1 \mathbf{e}_2^T\right\} \mathbf{M}_3.
\end{aligned} \tag{19}$$

Здесь учтено, что выполняются основные предпосылки регрессионного анализа, в частности, что случайные ошибки в разных наблюдениях некоррелированы, поэтому МО их произведений равны нулю. Следовательно, равно нулю и второе слагаемое в (19).

Найдем МО произведения $\boldsymbol{\varepsilon}_1 \mathbf{e}_1^T$:

$$\begin{aligned}
E\left\{\boldsymbol{\varepsilon}_1 \mathbf{e}_1^T\right\} &= E\left\{\boldsymbol{\varepsilon}_1 (\mathbf{y}_1 - \hat{\mathbf{y}}_1)^T\right\} = E\left\{\boldsymbol{\varepsilon}_1 (\mathbf{F}_1 \mathbf{a}_1 + \boldsymbol{\varepsilon}_1 - \mathbf{F}_1 \hat{\mathbf{a}}_1)^T\right\} = \\
&= E\left\{\boldsymbol{\varepsilon}_1 [\mathbf{F}_1 (\mathbf{a}_1 - \hat{\mathbf{a}}_1) + \boldsymbol{\varepsilon}_1]^T\right\} = E\left\{\boldsymbol{\varepsilon}_1 [\mathbf{F}_1 (-\Delta \mathbf{a}_1) + \boldsymbol{\varepsilon}_1]^T\right\} = \\
&= E\left\{\boldsymbol{\varepsilon}_1 (-\Delta \mathbf{a}_1^T \mathbf{F}_1^T + \boldsymbol{\varepsilon}_1^T)\right\} = E\left\{\boldsymbol{\varepsilon}_1 \boldsymbol{\varepsilon}_1^T - \boldsymbol{\varepsilon}_1 \Delta \mathbf{a}_1^T \mathbf{F}_1^T\right\} = \\
&= E\left\{\boldsymbol{\varepsilon}_1 \boldsymbol{\varepsilon}_1^T\right\} - E\left\{\boldsymbol{\varepsilon}_1 \left(\left(\mathbf{F}_1^T \mathbf{F}_1\right)^{-1} \mathbf{F}_1^T \boldsymbol{\varepsilon}_1\right)^T \mathbf{F}_1^T\right\} = \\
&= \sigma_1^2 \mathbf{I}_N - E\left\{\boldsymbol{\varepsilon}_1 \boldsymbol{\varepsilon}_1^T\right\} \mathbf{F}_1 \left(\mathbf{F}_1^T \mathbf{F}_1\right)^{-1} \mathbf{F}_1^T = \sigma_1^2 \mathbf{I}_N - \sigma_1^2 \mathbf{C}_1,
\end{aligned} \tag{20}$$

где $\mathbf{C}_1 = \mathbf{F}_1 \left(\mathbf{F}_1^T \mathbf{F}_1\right)^{-1} \mathbf{F}_1^T$.

$$\begin{aligned}
E\{\Delta \mathbf{a} \Delta \mathbf{d}^T\} &= \mathbf{M}_1 d_{y1} E\left\{\boldsymbol{\varepsilon}_1 \boldsymbol{\varepsilon}_1^T\right\} \mathbf{M}_3 = \mathbf{M}_1 \left(\sigma_1^2 \mathbf{I}_N - \sigma_1^2 \mathbf{C}_1\right) d_{y1} \mathbf{M}_3 = \\
&= \left(\mathbf{M}_1 \mathbf{I}_N - \mathbf{M}_1 \mathbf{C}_1\right) \sigma_1^2 d_{y1} \mathbf{M}_3 = \\
&= \left(\mathbf{M}_1 \mathbf{I}_N - \left(\mathbf{F}_1^T \mathbf{F}_1\right)^{-1} \mathbf{F}_1^T \mathbf{F}_1 \left(\mathbf{F}_1^T \mathbf{F}_1\right)^{-1} \mathbf{F}_1^T\right) \sigma_1^2 d_{y1} \mathbf{M}_3 = \\
&= \left(\mathbf{M}_1 \mathbf{I}_N - \mathbf{M}_1 \mathbf{I}_N\right) \sigma_1^2 d_{y1} \mathbf{M}_3 = 0.
\end{aligned}$$

Аналогично можно доказать, что вторая строка вектора (17) тоже равна нулю.

Окончательно:

$$E\{\hat{\theta}\} = Ad = \theta,$$

т.е. получаемые в соответствии с последовательной процедурой оценки параметров $\hat{\theta}$ приведенной формы модели (1) не смещены.

Предлагаемый подход к оцениванию параметров моделей может быть обобщен на случай объектов, допускающих разбиение на большее количество блоков первого уровня. Основным преимуществом предлагаемого подхода является сокращение размерности пространства входных переменных при раздельном исследовании блоков объекта.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Айвазян С.А. Прикладная эконометрика. Основы эконометрики. Том 2. – М.: ЮНИТИ-ДАНА, 2001. – 432 с.

**В.А. ЛИНЬКОВ, П.В. ЛИНЬКОВ, Ю.В. ЛИНЬКОВ,
А.К. МУСОЛИН, В.А. ПУШКИН,
О.В. РОЖКОВ, А.Н. САМОХИН, А.И. СУСЛОВ**

Рязанский государственный радиотехнический университет

ОЦЕНКА ЭЛЕКТРОЭФФЕКТИВНОСТИ АРХИТЕКТУРЫ ТРЕХФАЗНЫХ ЭЛЕКТРИЧЕСКИХ СЕТЕЙ С НЕЛИНЕЙНЫМИ НАГРУЗКАМИ КОМПЬЮТЕРНОГО ТИПА

Рассматриваются условия для построения оптимальной архитектуры трехфазных электрических сетей с нелинейными нагрузками компьютерного типа для минимизации электропотребления в ВУЗах.

При возрастающих с каждым годом тарифах на электроэнергию становится актуальной проблема минимизации потребления ВУЗами электроэнергии. Для выявления скрытых резервов минимизации электропотребления и отработки новых методик снижения потерь, авторами был произведен контроль качества электроэнергии на территории ВУЗа. Испытания проведены в соответствии с ГОСТ Р 53333-2008 аттестованным приборам для измерения качества электроэнергии «Прорыв КЭ» в течение 3 месяцев, круглосуточно в рабочие и выходные дни. Анализ результатов показал несоответствие по некоторым пунктам ГОСТ-Р 53333-2008 [1], а именно по коэффициенту n -ой гармонической составляющей напряжения (превышение уровня тех или иных гармоник) и по коэффициенту несимметрии напряжений по нулевой последовательности (разбалансировка величин мощностей по фазам А, В, С от хаотичного времени включения и выключения нагрузок).

Коэффициент несимметрии напряжения по нулевой последовательности K_{0Ui} в процентах определялся, как результат i -го наблюдения вычисляется по формуле (1) рекомендуемой ГОСТ 13109-97[2]

$$K_{0Ui} = \frac{\sqrt{3}U_{0(1)i}}{U_{1(1)i}} \cdot 100 \quad (1)$$

$U_{1(1)i}$ — действующее значение междуфазного напряжения прямой последовательности основной частоты (выражение 2)

где $U_{0(1)i}$ — действующее значение напряжения нулевой последовательности основной частоты трехфазной системы напряжений в i -ом наблюдении (выражение 3),

$$U_{1(1)i} = \sqrt{\frac{1}{12} \left[\left(\sqrt{3}U_{AB(1)i} + \sqrt{4U_{BC(1)i}^2 - \left(\frac{U_{BC(1)i}^2 - U_{CA(1)i}^2}{U_{AB(1)i}} + U_{AB(1)i} \right)^2} \right)^2 + \left(\frac{U_{BC(1)i}^2 - U_{CA(1)i}^2}{U_{AB(1)i}} \right)^2 \right]} \quad (2)$$

где $U_{AB(1)i}$, $U_{BC(1)i}$, $U_{CA(1)i}$ — действующие значения междуфазных напряжений основной частоты в i -ом наблюдении

$$U_{0(1)i} = \frac{1}{6} \sqrt{\left[\frac{U_{BC(1)i}^2 - U_{CA(1)i}^2}{U_{AB(1)i}} - 3 \cdot \frac{U_{B(1)i}^2 - U_{A(1)i}^2}{U_{AB(1)i}} \right]^2 + \left[\sqrt{4U_{BC(1)i}^2 - \left(U_{AB(1)i} - \frac{U_{BC(1)i}^2 - U_{CA(1)i}^2}{U_{AB(1)i}} \right)^2} - 3 \sqrt{4U_{B(1)i}^2 - \left(U_{AB(1)i} - \frac{U_{B(1)i}^2 - U_{A(1)i}^2}{U_{AB(1)i}} \right)^2} \right]^2} \quad (3)$$

Для анализа динамики трансформации искрений и корреляции их появления в зависимости от всплесков мгновенной мощности использовались следующие вспомогательные приборы: «Анализатор предвестников аварий и пожаров в электрических сетях», структура которого приведена в Л.[3] и «Анализатор качества сети», структура которого приведена в Л [4].

Появление гармоник в электрических сетях ВУЗов обусловлено наличием нелинейных нагрузок (компьютерного типа), т.е. форма тока потребляемая нагрузкой отличается от формы питающего напряжения. Более 80% потребляемой мощности в ВУЗе, приходится на так называемые «компьютерные нагрузки». Это системные блоки, мониторы, копировальные аппараты, кондиционеры, энергосберегающие осветительные приборы, холодильники, системы охраны и пожарной безопасности, серверы и т.д. Электрические сети многих ВУЗов были спроектированы в 50-60 годах прошлого века без учета «компьютерных нагрузок», которых в то время не было. При этом общее количество компьютеров, во многих ВУЗах превышает 2000 шт. которые, в

свою очередь разделяются на три класса по типу используемых блоков питания. 1)- без цепей коррекции коэффициента мощности (FP), 2)-с пассивной коррекцией FP; 3) - с активной коррекцией FP, и подклассы по типу загрузки материнской платы и мощности видеокарты а) - работа с текстами; б)- работа с 3D графикой (мощность, потребляемая компьютером из сети в разных режимах работы, может отличаться в 2 раза).

В результате работ по анализу качества электроэнергии и диагностики качества электрической сети ВУЗа было получено следующее (4) условие (расшифровка компонент которого, приводится в выражениях (5-9)) - для минимизации электропотребления и осуществления балансировки трехфазной сети содержащей нелинейные (компьютерные) нагрузки:

$$\left. \begin{aligned} FS_A(t) &= FS_B(t) = FS_C(t) \\ PF_A &= PF_B = PF_C = 1 \\ D_A + D_B + D_C &= 0 \end{aligned} \right\}, \quad (4)$$

где

$$D = \sqrt{S^2 - P^2 - Q^2} - \text{мощность искажения при наличии гармоник}; \quad (5)$$

$$PF = \frac{P}{S} - \text{коэффициент мощности}; \quad (6)$$

$$\text{где } S^2 = \left(\sum_{n=1}^{\infty} U_h^2 \right) \cdot \left(\sum_{n=1}^{\infty} I_h^2 \right) - \text{полная мощность}; \quad (7)$$

$$P = \sum_{h=1}^{\infty} U_h I_h \cos \varphi_h - \text{активная мощность сигнала содержащего}$$

гармоники; (8)

$$Q = U_1 I_1 \sin \varphi_1 - \text{реактивная мощность для основной частоты}; \quad (9)$$

Приведенное выше условие идеальной балансировки трехфазной сети проводимой с целью минимизации потребляемой электрической энергии, предъявляет следующие требования к потребителям энергии:

1. Величины мгновенной полной потребляемой мощности по каждой из фаз должны быть равны и синхронизированы по времени.

2. Емкостные и индукционные составляющие нагрузки должны, подобраны так, чтобы их величины компенсировали друг друга.

3. Соблюдались условия отсутствия гармоник, резонансов, искрения контактов, разогрева проводников и изоляторов, отсутствие неконтролируемых утечек электроэнергии и электрокапиллярных эффектов.

4. Коммутация нагрузок осуществляется в момент перехода напряжения через ноль.

5. Осуществление мониторинга и нейтрализации постановщиков активных и пассивных помех в электрическую сеть ВУЗа.

Заключение

Для оптимизации потребления электроэнергии необходимо проинформировать следующие мероприятия:

1. провести аттестацию и электронную маркировку всех компьютеров подключенных к электрической сети с использованием «штрих-кода» где указано: 1) точка присоединения, 2) фаза (А или В или С), 3) фактическая полная мощность, 4) реактивная мощность и мощность искажения, 5) режимы работы с текстами или с 3D графики, 6).метки времени включения/выключения компьютера.

2. с целью оперативного получения достоверной информации о фактических мощностных характеристиках каждой нелинейной нагрузки необходимо разработать методику и оборудование для бесконтактного съема информации и маркировки компьютеров.

3. по окончании сбора исходной информации производится моделирование с целью получения оптимальной архитектуры сети с учетом временных рамок начала и окончания учебных занятий или других работ в каждой аудитории и реальной нагрузочной способности электрической сети.

Эффективность предложенных мероприятий подтверждено физическим моделированием архитектуры компонентов компьютерных классов в реальных условиях ВУЗа. Для оценки экономии электроэнергии использовался измеритель показателей качества электрической энергии Ресурс-UF-2М-3Т52-5-100-1000 с погрешностью измерения $\pm 0.5\%$. Минимальная экономия активной мощности составила от 4 до 12%, в зависимости от первоначальной случайно организованной «удачной» или «неудачной» компоновки компьютеров по фазам А-В-С и других компонентов электрической сети влияющих на величину коэффициента мощности.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. ГОСТ Р 53333-2008 Контроль качества электрической энергии в системах электроснабжения общего назначения.

2. ГОСТ 13109-97. Межгосударственный стандарт. Электрическая энергия. Совместимость технических средств электромагнитная. Нормы качества электрической энергии в системах электроснабжения общего назначения.

3. Патент RU № 2309461 «Способ определения времени возникновения предпожарной ситуации от искрения в электрической сети и устройство для его осуществления». Линьков В.А, Линьков П.В. Линьков Ю.В.

4. А.С. SU № 1755203 «Анализатор качества сети» Линьков В.А.

Ю.П. ЕЛИСТРАТОВ, Е.В. ЛОЦМАНОВ

Рязанский государственный радиотехнический университет

МЕХАНИЗМЫ, ФОРМУЛИРУЮЩИЕ РЕАЛИСТИЧНОЕ ПЕРЕМЕЩЕНИЕ ЛЕТАЮЩИХ СУЩЕСТВ

Введение. Анимация полёта любого летающего существа сложна как таковая, ведь при её создании необходимо учитывать множество факторов способствующих созданию анимации, сколько не будь приближенной к полету, осуществляемому в реальном мире.

Принцип создания реалистичной анимации необходимо начинать изучать, находясь ещё "на земле", в первую очередь следует коснуться основных механизмов движения и скелетно-мышечного строения прототипа.

Крайне сложно достоверно передать тончайшие механизмы работы мышц, которые способны в мгновение сократиться и придать определённой области тела совершенно другое положение, тем не менее, ключом к успеху является именно "подвижность".

Постановка задачи. В настоящей работе предлагается способ автоматизации движений летающих существ (рассмотренный на примере птиц), механизмы перемещения в пространстве.

Крыло это сложная система взаимодействия различных факторов, таких как размах и удлинение, углы установки и углы атаки, сопротивление нагрузкам и т.д. направленных на организацию полёта.

При анимации прототипа крыла использовался программный код управления суставами и, соответственно, прикреплённым к ним перьевым оперением. Так, например, группа больших маховых перьев управлялась посредством манипуляции запястья.

Таким способом реализованы маховые движения, и сгибание-разгибание перьев во время взмаха крыла в соответствующем направлении.

Плечевые кости птицы расположены благоприятным образом для прочной работы крыла. Здесь наблюдается тот случай, известный

из практики самолётостроения, когда центр тяжести совпадает с центром жесткости, т.е. с той точкой, около которой крыло поворачивается под действием крутящего момента.

Аэродинамическая сила приложена в центре давления (рис. 1).

Если же все три центра совмещаются в одной точке (центр жесткости), то кручения не будет и крыло только изгибается. Эти условия являются необходимыми для устранения вибрации крыла, и птицы в совершенстве этим пользуются, избегая возникновения больших силовых напряжений в структуре крыла.

На (рис. 2) показано положение линий всех трёх центров в крыле при его аэродинамических испытаниях под нагрузкой, пунктиром обозначено смещение профиля.



Рис. 1. Положение трех центров в крыле птицы

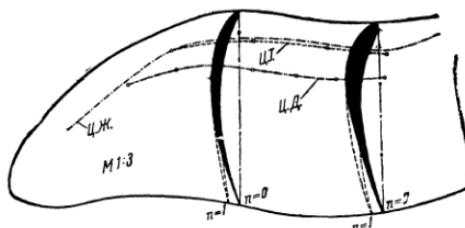


Рис. 2. положение Ц.Т. и Ц.Ж. в крыле, упругие изменения крыла при разных нагрузках

Впоследствии прототипу крыла были добавлены дополнительные возможности, такие как изгиб в разных направлениях, расправление и складывание перьев, перенастроенные ввиду изложенных выше свойств.

В результате крыло получилось более гибким к различным вариантам управления.

Заключение. Данный механизм настройки прототипа крыла подходит для организации перемещения в пространстве многих летающих существ.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. И.Н. Виноградов. Аэродинамика птиц – парителей. М.: Досарм, 1951
2. А.Ф. Климов, А.И.Акаевский. Анатомия животных. М., 2003.

3. О. Лилянталь Полет птиц как основа искусства летать. Материалы для систематики техники полёта. М.: Институт компьютерных исследований. 2002.

4. Н.Е. Жуковский. О парении птиц. Берлин 1922.

Е.В. ЛОЦМАНОВ, Ю.П. ЕЛИСТРАТОВ

Рязанский государственный радиотехнический университет

СУЩНОСТЬ И ПРОБЛЕМЫ РЕАЛИЗАЦИИ ТЕХНОЛОГИИ МУСКУЛЬНО-СКЕЛЕТНОЙ АНИМАЦИИ

Введение. В современном мире анимационное искусство приобрело большую значимость и актуальность в связи с постоянным развитием новых компьютерных технологий. Одним из методов анимации является использование систем с мышцами.

Область применения мускульной анимации весьма широка. В некоторых фильмах и не только, наравне с живыми персонажами, играют компьютерные, поэтому для придания реалистичности изобретаются все более совершенные технологии анимации [1].

Постановка задачи. Сегодня существует множество методик увеличения реалистичности анимации персонажей. Сравним способ простой анимации при помощи костей и анимации при помощи костно-мускульной системы.

Такой объект как кожа – достаточно сложен для имитации программой, поскольку он способен производить различные, едва уловимые эффекты. При простом способе имитации кожи с помощью костей зачастую анимация получается менее убедительной, потому что кожа, как было сказано ранее, воспроизводит едва уловимые колебания, которые делают ее похожей на настоящую, что и затрудняет анимацию. Однако существуют методы для решения этих проблем, но они более затратные, как по времени, так и по качеству, чем мускульные системы [2].

Новый способ анимации при помощи мышц и создания скелетного каркаса позволяют приблизиться к реально существующим объектам, а соответственно и более взвешенно подходить к процессу создания анимации персонажа. Тем самым удастся добиться более эффективного восприятия динамики движений объекта в соответствии с его анатомическими особенностями.

При создании в трехмерной программе анимации, необходимо учитывать, что человеческое тело и тело животного имеет огромное

число различных типов мышц. Ввиду сложности решаемой проблемы, нужно упрощать эту систему. Поэтому для имитации, например, используются мышцы в форме эллипсоидов. Кроме того, само количество мышц в теле весьма высоко. Все мышцы нужно классифицировать и выделять их в более простые формы, для того чтобы быстро и эффективно их использовать. Те же решения могут применяться и к коже.

Связь между кожей и тканями является довольно сложным механизмом, в котором прослеживаются такие особенности как растяжение и сжатие мышц, эффекты тряски, импульс которых передается коже, «скольжение» кожи по мышце и многие другие свойства.

Заключение. Таким образом, можно сделать следующие выводы, рассмотрев преимущества и недостатки двух методов.

1. Анимация при помощи костей.

Абстрактное представление мышц. Воздействие мышц только имитируется на коже, как таковые объекты мышц не видимы. Соответственно могут возникать неточности в имитации, связанные с пониманием аниматором их работы.

Сложность использования. Чем больше мышц, тем сложнее их контролировать, задачи имитации сложных мускульных систем таким способом является достаточно сложным, а порой и не выполнимым с необходимой точностью.

2. Использование костно-мускульной системы.

Мышцы являются реально существующими объектами. Мышцы крепятся к скелету и имеют прямое назначение, которое имеют мышцы на реальном объекте, хотя и зачастую упрощенное.

Имитируемые мышцы имеют свойства, подобные реальным мышцам. Объекты мышц имеют настраиваемые свойства, также характерные для мышц реальных, такие как сжатие, растяжение, динамика и другие свойства.

Исходя из вышеописанного сравнения можно сделать вывод о том, что костно-мускульные системы являются более удобными в использовании.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. К. Мараффи Создание персонажей в Maya. Моделирование и анимация.: издательство ООО «Вильямс», 2008.

2. Scott Eaton. Anatomy for 3D artists.
www.3dworldmag.com/page/3dworld?entry=anatomy_for_3d_artists

Р.Е. МЕДВЕДЕВ

Рязанский государственный радиотехнический университет

КЛЮЧЕВЫЕ КРИТЕРИИ ВЫБОРА ПЛАТФОРМЫ ДЛЯ СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ

Рассматривается набор критериев выбора платформы для создания системы дистанционного обучения.

Внедрение информационных технологий в деятельность человека создало потребность в качественно новом обучении, наступило время дистанционного обучения, основанного на широком применении компьютерных и сетевых технологий.

Большое значение организации дистанционного обучения играет выбор обучающей среды, обеспечивающей организацию учебного процесса. Ошибочный выбор среды обучения может привести к неправильному пониманию учебного материала, отсутствию интереса к обучению у учащихся или взаимосвязи между учителем и студентом, тем самым снижая эффективность обучения. К основным критериями выбора можно отнести следующие:

- функциональность. Следует помнить, что система дистанционного обучения должна не только предоставить студенту учебную программу, но и обеспечить возможность сетевого контакта как с преподавателем (возможность оперативного получения ответов на задаваемые вопросы), так и с другими студентами, изучающими данный курс (обмен мнениями). В связи с этим, необходимо наличие максимального набора вариантов коммуникации между студентом и преподавателем (чаты, форумы, система мгновенного обмена сообщениями, аудио и видеосвязь). Система должна предоставлять удобные и эффективные средства администрирования (управление курсами и обучаемыми) а так же контроля деятельности обучаемых (анализ активности, расчет ключевых показателей);

- надежность. Данный критерий определяет степень защиты системы от несанкционированного доступа (хакерских атак), степень устойчивости системы в различных режимах работы, а так же характеризует удобство администрирования и простоту использования системы. Показатель влияет непосредственно на эффективность использования системы и отношение пользователей к ней;

- стоимость. Данный критерий связан как с местом обучения (специализированный класс, работа, дом), так и с ценой программного продукта, которая складывается из стоимости самой системы (она может быть и бесплатной, в случае использования систем с открытым

исходным кодом), и затрат на ее внедрение, разработку курсов и сопровождение;

- наличие средств разработки контента. Наличие собственного редактора учебного контента позволит облегчить разработку учебных курсов и реализовать единое представление образовательных материалов различного назначения;

- поддержка Sharable Content Object Reference Model (SCORM). SCORM — стандарт, разработанный для систем дистанционного обучения. Данный стандарт содержит требования к организации учебного материала и всей системы дистанционного обучения. SCORM позволяет обеспечить совместимость (возможность взять один и тот же учебный материал и, не внося в него изменений, использовать его в различных системах дистанционного обучения) компонентов и возможность их многократного использования. Учебный материал представлен отдельными небольшими блоками, которые могут включаться в разные учебные курсы и использоваться системой дистанционного обучения независимо от того, кем, где и с помощью каких средств были созданы;

- система проверки знаний. При традиционном обучении проверка знаний проводится для самоконтроля обучаемого, с тем, чтобы помочь ему выявить “пробелы” в пройденном материале, при дистанционном обучении возникает вопрос о качестве полученных знаний. Признание за пределами учебного заведения выдаваемых им дипломов и сертификатов зависит от того на сколько реальные знания их обладателей соответствуют номинальным знаниям. Поэтому система должна не только предоставлять возможности контроля успеваемости студента для заинтересованной стороны, но и, эффективно оценивать степень усвоения обучаемым учебного материала;

- удобство использования. Это важный критерий, поскольку потенциальные пользователи не будут использовать технологию, которая кажется громоздкой или создает трудности при навигации. Поэтому система должна иметь простой и понятный интерфейс пользователя для студентов и преподавателей, которые, вполне возможно, не являются профессионалами в области информационных технологий;

- модульность. Система должна быть организована как набор микромодулей или блоков учебного материала. Это позволит не только повторно использовать одни и те же модули в учебных целях, но и быстро и эффективно адаптировать систему к текущим потребностям, как с помощью изменения настроек модуля, так и использованием различных комбинаций модулей;

- обеспечение доступа. Система должна предоставлять доступ к учебной программе вне зависимости с пространственными, временными и прочими факторами, ограничивающими возможности обучаемых. Система должна максимально использовать различные способы представления информации (текст, графика, видео, анимация, звуковое сопровождение).

Необходимо отметить, что при дистанционном обучении предъявляются более строгие требования к учебному материалу, он должен быть разработан более тщательно, чем это обычно делается при очном обучении. Опыт проведения дистанционного обучения показал наибольшую эффективность бригадного подхода при разработке и организации учебного процесса, необходимо совместное участие преподавателей, специалистов по разработке учебного материала и использованию разных технологий, экспертов, редакторов, администраторов, технического персонала.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Ибрагимов И.М. Информационные технологии и средства дистанционного обучения. – М.: Академия, 2007. – 336с.
2. Трайнев В.А., Гуркин В.Ф., Трайнев О.В. Дистанционное обучение и его развитие. – М.: Дашков и Ко, 2006. – 296с.
3. Бакалов В.П., Крук Б.И., Журавлева О.Б. Дистанционное обучение. Концепция, содержание, управление. – М.: Горячая Линия – Телеком, 2008. – 108с.

К.В. МИРОНОВА

Рязанский государственный радиотехнический университет

УСТОЙЧИВОЕ УПРАВЛЕНИЕ ЛИНЕЙНЫМИ СИСТЕМАМИ ПЕРЕМЕННОЙ СТРУКТУРЫ

Рассматриваются вопросы построения регуляризирующих алгоритмов для устойчивого управления линейными системами переменной структуры, используемых в технологиях автоматического управления и в информационных технологиях.

Введение. Колоссальные возможности вычислительных машин позволяют встраивать их в технологически линии обработки информации, использующих математические модели протекающих процессов. Однако зачастую подобные модели носят неустойчивый характер, решения задач сильно зависят от колебаний исходных данных. Для борьбы с подобными негативными особенностями разработаны специ-

альные методы регуляризации алгоритмов (см., например, [1,2]). В свою очередь, самих методов регуляризации уже настолько много (а зачастую они и сложны), что встает вопрос о выборе из их числа наиболее простых (с вычислительной точки зрения) алгоритмов, «заточенных» под конкретные задачи.

Именно такая конкретизация регуляризации и предложена в данной статье, развивающей работу [3] и рассматривающей вопросы устойчивого управления линейными системами, которые, в свою очередь, входят составными частями в автоматизированные системы управления или в информационные системы.

1. Линейная модель с управлением

На рис.1 представлена рассматриваемая линейная модель с управляемыми параметрами.

Вся система состоит из шести блоков $B.1 - B.6$. На блок $B.1$ подается начальное уравнение $U=U_0$.

Под воздействием управления и формируется квадратная $n \times n$ матрица A и n – вектор b . Матрица A и вектор b передаются на блок $B.2$, который вычисляет решение x уравнения

$$Ax = b. \quad (1)$$

Зависимость элементов A и b от уравнения u говорит о переменной структуре уравнения (1) и тем самым всей линейной системы управления. Блок $B.3$ вычисляет норму отклонения решения x от заданного номинала x_0 . В блоке $B.4$ норма отклонения Δ сравнивается с заданным порогом ε . Если отклонение Δ не превышает порог, то система сработала «успешно», в блоке $B.6$ формируется управление $U=U_1(x)$, которое продолжает управление процессом. Если же отклонение Δ превысило порог ε , то в блоке $B.5$ формируется новое управление $U=U_2(x)$, которое возвращается на блок $B.1$. Снова формируется матрица A и вектор b до тех пор, пока не будет получено приемлемое отклонение Δ . Такова работа линейной системы.

2. Постановка задачи

Ставится задача: сконструировать такую линейную модель, которая бы устойчиво решала уравнение (1) в том смысле, чтобы минимизировать отклонение Δ и на его основе построить «оптимальное» управление $U=U_1(x)$, не прибегая, тем самым, к «возвратному» управлению $U=U_2(x)$.

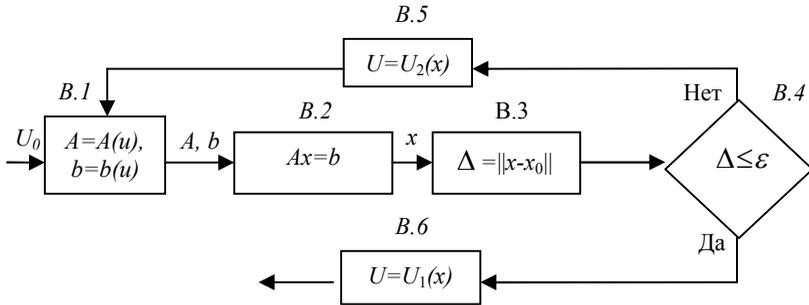


Рис. 1

3. Решение задачи

Первый способ. Представим уравнение (1), в котором матрица A состоит из строк $A = (a_1, \dots, a_n)$, в развернутом виде

$$\begin{cases} (a_1, x^T) = b_1 \\ \dots \\ (a_n, x^T) = b_n \end{cases}, \quad (2)$$

где обозначения очевидны. Систему (2) заменим на равносильную ортогональную систему:

$$Cx = d, \quad (3)$$

которая будет играть роль регуляризующего оператора и в которой матрица C состоит из строк $C = (c_1, \dots, c_n)$, ортогональных друг другу:

$$(\forall i, j; i \neq j) \quad (c_i, c_j) = 0.$$

Систему (3) можно построить по системе (2), используя стандартную процедуру ортогонализации [2]:

$$c_1 = a_1, \quad (\forall 2 \leq k \leq n) \quad c_k = a_k - \sum_{i=1}^{k-1} (a_i, c_i) c_i \quad (4)$$

$$d_1 = b_1, \quad (\forall 2 \leq k \leq n) \quad d_k = b_k - \sum_{i=1}^{k-1} (a_i, c_i) b_i \quad (5)$$

Для решения системы (3) применим устойчивую «процедуру проектирования», описанную в работе [3].

Очевидно, что равенства

$$\prod_i: (c_i, x^T) = d_i, \quad i = 1, \dots, n, \quad (6)$$

описывают в аффинном пространстве $A^n = \langle 0, R^n \rangle$ ортогональные друг другу гиперплоскости \prod_i . Тогда, как нетрудно видеть, имеет место следующая теорема.

Теорема. *Малые изменения элементов матрицы C или вектора d могут привести только к малым изменениям решения x системы (3).*

Следовательно, устойчивость рассматриваемой системы переменной структуры достигается путем замены в блоке $B.1$ матрицы A на матрицу C , вектора b на вектор d , и замены в блоке $B.2$ системы (2) на систему (3).

Второй способ. Если система (см. рис.1) не допускает процедуры замены, описанной в первом способе решения, предлагается идея регуляризации на основе идей работы [1].

Расширим линейную систему рис.1, внося в блок $B.2$ в качестве регуляризирующего оператора дополнительное слагаемое

$$KEx = Kx_0, \quad (7)$$

где E – единичная матрица надлежащей размерности, K – достаточно большая константа, x_0 – прежнее номинальное значение вектора x .

Построим новую систему, которая охватывает новый блок $B.2$:

$$(A + KE)x = b + Kx \quad (8)$$

В силу того, что матрица KE – ортогональная, то матрица $A + KE$ «весьма близка» к ортогональной и, следовательно, имеет место следующая теорема.

Теорема 2. Малое изменение элементов матрицы A и вектора b могут привести только к небольшим изменениям вектора x системы (7).

Как следствие, будет мало меняться и решение системы (2).

4. Моделирование

Проиллюстрируем второй способ решения конкретным примером. В блоке $B.1$ матрица A состоит из строк $a_1 = (1, 1)$, $a_2 = (1, 1 + u)$, вектор $b = (2, 3 + u)^T$. Номинал блока $B.3$ – вектор $x_0 = (0, 2)^T$. Любым методом легко можно получить решение уравнения (1): $x_2 = \frac{1+u}{u} \rightarrow +\infty$, $x_1 = \frac{u-1}{u} \rightarrow -\infty$ при $u \rightarrow +0$. Следовательно, блок $B.2$ работает не устойчиво и система управления будет задействовать «возвратное» управление $u = u_2(x)$.

«Утяжелим» блок $B.2$ регуляризирующим оператором вида (7): константа $K = 100$,

$$\begin{cases} x_1 + 101x_2 = 202, \\ 100x_1 + (1+u)x_2 = 3+u \end{cases} \quad (9)$$

Система (9) уже устойчива, её решение, полученное «методом проектирования», мало отличается от номинала и имеет вид:

$$x_2 = \frac{20399 - u}{10200 - u} \rightarrow 1,99990\dots, \quad x_1 = 202 - 101 x_1 \rightarrow 0,00991\dots \text{ при } u \rightarrow +0.$$

Таким образом, в рассматриваемом случае «утяжелением» блока В.2 достигнуто устойчивое управление системой переменной структуры.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Тихонов А.Н., Арсенин В.Я. Методы решения некорректных задач. М.: Наука, 1979.
2. Бакушинский А.Б., Гончарский А.В. Некорректные задачи. Численные методы и приложения. М.: Изд. МГУ, 1989.
3. Миронова К.В. Метод проектирования решения СЛАУ. Сложность метода проектирования решения СЛАУ. / Межвуз. сборник научн. трудов «Математическое и программное обеспечение вычислительных систем». 2011. С.17-20, 72-74.

В.М. МИРОШИН, Д.А. ПЕРЕПЕЛКИН

Рязанский государственный радиотехнический университет

3D МОДЕЛИРОВАНИЕ АЛГОРИТМОВ АДАПТИВНОЙ МАРШРУТИЗАЦИИ КОРПОРАТИВНОЙ СЕТИ

Рассмотрены цели создания и основные особенности разработанной программной системы 3D моделирования алгоритмов адаптивной маршрутизации корпоративной сети.

Алгоритмы маршрутизации применяются для определения наилучшего пути пакетов от источника к приёмнику и являются основой любого протокола маршрутизации. Для формулирования алгоритмов маршрутизации корпоративная сеть рассматривается как неориентированный граф. При этом маршрутизаторы являются узлами, а физические линии между ними – рёбрами соответствующего графа. Каждому ребру графа присваивается определённое число – стоимость, зависящая от физической длины линии связи, скорости передачи данных или задержки передачи. Разработанная программная система анализирует 4 алгоритма: алгоритм Дейкстры, алгоритм Беллмана-Форда, алгоритм Йена и алгоритм парных переходов. Алгоритм Дейкстры применяется в протоколе OSPF и имеет трудоёмкость построения маршрутов $O(N^2)$. Алгоритм Йена на данный момент не используется в протоколах маршрутизации, он рассмотрен с целью анализа подхода, используемого

при расчёте оптимальных маршрутов и имеет трудоёмкость $O(N^3)$. Алгоритм Беллмана-Форда применяется в протоколах IGRP, EIGRP и имеет трудоёмкость $O(N^3)$. Разработанный в последнее время алгоритм парных переходов позволяет производить расчет оптимальных маршрутов без полного пересчета дерева кратчайших путей. В этом алгоритме ребра графа делятся по признаку вхождения в дерево оптимального маршрута на два подмножества: E_T – множество ребер дерева оптимальных маршрутов и множество ребер замены E_R , не вошедших в дерево оптимальных маршрутов. При соответствующих условиях некоторое ребро $e_{i,j} \in E_R$, инцидентное вершинам v_i и v_j может перейти во множество ребер дерева E_T , заменив собой некоторое ребро $e_{k,p} \in E_T$. Такие переходы называют парными переходами. Данный алгоритм имеет трудоёмкость расчета кратчайших путей равную $O(kN)$.

Изменение пропускной способности линий связи приводит к полному перерасчету маршрутной таблицы. Точное определение множества кратчайших путей, которые изменятся после изменения пропускной способности линии связи, позволяет уменьшить трудоемкость рассматриваемой задачи. Из всех алгоритмов наиболее оптимальным является алгоритм парных переходов. Он имеет наименьшую трудоёмкость и способен реагировать на динамические изменения в корпоративной сети.

Разработанная программная система 3D моделирования алгоритмов адаптивной маршрутизации корпоративной сети разработана в среде Embarcadero Delphi XE. Основными программными единицами являются модули RouteAlgorithms и RouteGraphic. Модуль RouteAlgorithms содержит процедуры реализации алгоритмов маршрутизации. Модуль RouteGraphic отвечает за визуальное представление программной системы. Предлагаемая система обладает современным пользовательским интерфейсом, который реализован на использовании сторонних компонентов подключаемых графических библиотек. При большом количестве различных программных средств, специализирующихся в области адаптивной маршрутизации, 3D визуализация корпоративной сети и наглядность отображения ее структуры играют огромную роль как с точки зрения эффективности функционирования сети, так и с точки зрения конкурентоспособности самого программного продукта.

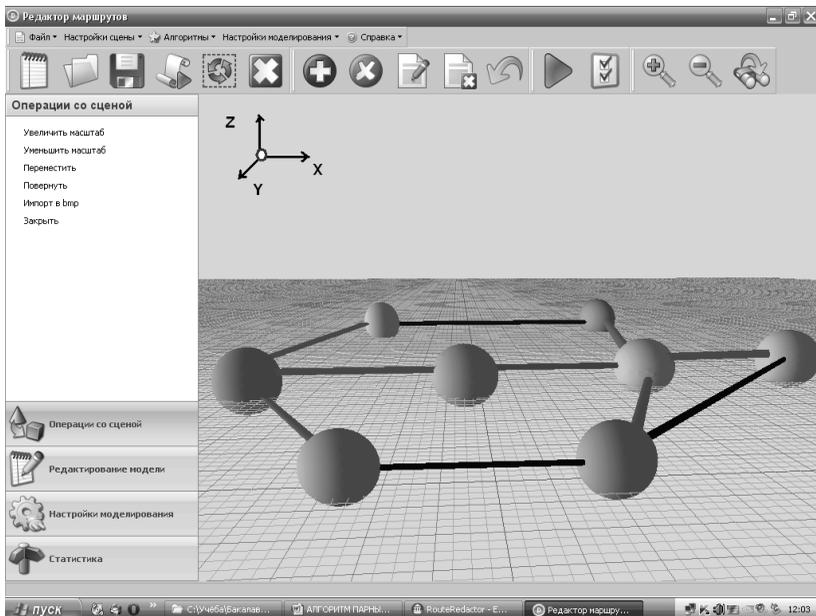


Рис. 1. Интерфейс системы 3D моделирования сети

Поэтому в разрабатываемой программной системе применяется трехмерное отображение объектов сетевой структуры (рис. 1). 3D визуализация используется для более наглядного представления структуры сети при проектировании, в процессе размещения основных элементов: маршрутизаторов и линий связи, а также позволяет оперативно, т.е. в режиме реального времени реагировать на все динамические изменения. 3D отображение реализуется с помощью стандартной графической библиотеки OpenGL. Недостатком данного подхода является увеличение загрузки ЦП на 10-15%.

В дальнейшем предполагается усовершенствование функциональных возможностей программной системы за счет включения процедур мониторинга и адаптивного управления корпоративной сети.

А.В. НЕСТЕРОВ

Рязанский государственный радиотехнический университет

СИСТЕМЫ КОМПЬЮТЕРНОГО ЗРЕНИЯ В ПРОМЫШЛЕННЫХ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССАХ

В статье описана типовая структура систем компьютерного зрения. Приведены примеры применения и осуществлен обзор текущих направлений применения таких систем в промышленности. Определяются проблемы разработки систем компьютерного зрения и пути их решения.

На промышленных линиях, с которых ежесекундно сходят сотни товаров, трудится множество людей, следящих за внешним видом выпускаемых предметов и вручную отбраковывающих негодные изделия. В результате получается, что производительность таких линий ограничивается не мощностью оборудования, а скоростью работы сотрудников ОТК[1].

Напрашивается решение - автоматизировать данные сферы деятельности, освободить работников от однообразного труда, а задачу визуального контроля отдать системам компьютерного зрения [1].

Составные части систем компьютерного зрения: оптическая система, преобразователь свет-сигнал, электронный тракт ввода элементов изображения в память ЭВМ или специального вычислителя, математическое обеспечение сбора и обработки необходимых зрительных данных [2].

В результате применения таких систем качество и производительность должны значительно повыситься: когда системы машинного зрения внедрялись на текстильных и химических заводах, скорость автоматизированного конвейера возрастала до 100 км/ч. Немаловажно и то, что компьютеру неведома усталость и не нужен отдых. Вдобавок появляется возможность сбора подробной информации по каждому рабочему месту, оперативного анализа и выявления ошибок в технологическом процессе с дальнейшим совершенствованием последнего.

Системы компьютерного зрения (СКЗ) призваны и во многих случаях уже решают задачи по дополнению или даже замене человека в областях деятельности, связанных со сбором и анализом зрительной информации. Уровень их использования в прикладных областях является одним из наиболее ярких и наглядных интегральных показателей уровня развития высоких технологий в самых различных отраслях промышленности[2].

50% всех разрабатываемых в мире СКЗ эксплуатируются в задачах контроля качества. Это, прежде всего, визуальный контроль за

процессом сборки, цветом и качеством поверхности продукции, внешним видом и чистотой упаковки, правильностью и разборчивостью этикеток, уровнем жидкости во всевозможной таре и т. д. Примерно 10% этих задач выполняются системами трехмерного зрения. Отдельная область использования СКЗ на производстве - проведение всевозможных визуальных измерений параметров технологических процессов и, в частности, определение размеров предметов [1].

20% - СКЗ для проектов автоматизации производства и внедрения промышленных роботов. Такие СКЗ упрощают самые разные виды высокоточной деятельности (сборка и разборка, фасовка, покраска, сварка, утилизация), облегчают транспортировку грузов, применяются в системах учета, маркировки, регистрации и сортировки продукции [1].

Оставшиеся 30% это системы компьютерного зрения, не имеющие прямого отношения к производству. Таким образом, именно промышленность – один из главных потребителей таких систем.

Ниже приведены примеры успешного внедрения систем компьютерного зрения в различных отраслях европейской промышленности [1].

1. На лесопилке удалось достичь ежегодной экономии 1 млн. евро за счет повышения точности распиловки древесины.

2. В пивоварне за счет ускорения сортировки бутылок ежемесячно экономится 50 тыс. евро.

3. СКЗ на заводе по производству магнитов окупилась за полгода.

4. Оценка качества бисквитов на кондитерской линии осуществляется со скоростью 60 пирожных в секунду.

5. Сортировка керамической плитки выполняется со скоростью 300 шт. в секунду.

6. Точность распознавания оттенков изделий из кожи повышена с 85 до 97%.

7. Качество производимого листового стекла увеличилось на 2-4%

Несмотря на значительные успехи, достигнутые в области компьютерного зрения, эффективное его использование в качестве средства автоматизации приходится, прежде всего, на наиболее развитые производства с общей высокой культурой и технологией. В плохо организованных средах и исследовательских задачах на компьютерное зрение возлагается роль дополнения, расширяющего возможности и повышающего эффективность человеческого зрительного анализа, при ведущей и определяющей роли человека-оператора или исследователя.

Основные областями промышленности, в которых применяются системы компьютерного зрения:

1. Машиностроение и автомобилестроение – контроль геометрических размеров, контроль отверстий и их взаимного расположения, контроль сборки и пр;

2. Полиграфическая промышленность – контроль дефекты, разрывы;

3. Фармацевтическая, пищевая промышленность – контроль геометрических размеров, включения, вкрапления, цветовая окраска, отсутствие таблетки в блистере, наличие маркировки;

4. Горнодобывающая промышленность – определение гранулометрического состава руды;

5. Радиоэлектронная промышленность – контроль маркировки компонентов (микросхемы, дискретные элементы), контроль геометрических размеров, контроль наличия (отсутствия) выводов компонентов

6. Общее для всех отраслей промышленности это поточные линии – контроль включений, вкраплений, проверка уровня налива, наличие колпачка и других элементов, дефекты тары, перекос этикетки, наличие маркировки;

Главным недостатком СКЗ считается отсутствие единых стандартов на оборудование и программные интерфейсы. Пользователи СКЗ постоянно жалуются на невозможность использовать системы разных производителей, а интеграционные продукты в этой области стоят недешево. В плане аппаратной стандартизации большинство поставщиков ориентируются сегодня на типовые коммерческие решения - прежде всего цифровое видео (Digital Video), подразумевающее формирование видеоизображения с разрешением 720x480 пикселей и частотой 30 кадр./с. Но разработчики систем автоматической микросборки, заинтересованные в точном позиционировании манипуляторов промышленных роботов, требуют повышения разрешающей способности камер, а это ведет к их сильному удорожанию. Кроме того, по мере улучшения качества сканируемого изображения будет расти нужда в эффективном системном ПО для быстрого захвата видеоданных, их очистки, сжатия и хранения. По всей видимости, со временем такие функции будут реализованы аппаратно, в связи с чем можно предсказать рост спроса на встраиваемые СКЗ.

Заманчивой и многообещающей выглядит концепция мультиспектральных СКЗ, обрабатывающих изображения не только в области видимого человеком спектра, но и получаемых с помощью радарных или лазерных установок, а также инфракрасных камер (есть примеры

применения СКЗ при распознавании тепловых полей при анализе качества горячих булочек в кондитерской промышленности). Основная помеха на этом пути развития машинного зрения - стоимость и сложность эксплуатации соответствующих датчиков[1].

Продолжительна и сложна процедура калибровки СКЗ при настройке на определенную предметную область. Нередко она требует значительного времени и вычислительных ресурсов для обучения нейронной сети. Прикладное ПО также нередко отличается неудобным интерфейсом, а перенастройка системы на новую номенклатуру, да еще в реальном масштабе времени, на чем нередко настаивает пользователь, чаще всего невозможна - необходимо перепрограммирование СКЗ. А покупателям уже нужны системы, способные анализировать продукцию на конвейере не поштучно, а разом - осматривая все, что находится в данный момент времени на ленте, и мгновенно оценивая состояние всех изделий[1].

Ощущается нехватка хороших математических алгоритмов, ориентированных на компьютеры с высокой производительностью и поддерживающих параллельную обработку. Остается пока нерешенной проблема распознавания схожих, но по-разному освещенных объектов, и вряд ли с ней удастся справиться в обозримом будущем[1].

Негативно сказываются на рынке завышенные ожидания потребителей, не очень высокая прибыльность уже существующих СКЗ, небольшое число хороших продуктов, а также возможность решать производственные задачи другим путем - без СКЗ, с помощью дешевого персонала.

Но все эти проблемы временные. Ведущие поставщики уже договариваются о единых стандартах, для чего планируется создать общедоступный репозиторий знаний и типовых шаблонов продуктов, определить программные интерфейсы, а также выработать соглашение по языкам описания и представления данных СКЗ [1].

Разрабатываются системы компьютерного зрения, поддерживающие взаимодействие с различными типами устройств ввода и захвата изображения. Применяется принцип модульного построения программного обеспечения СКЗ, что позволяет выпускать универсальные продукты, пользователь которых самостоятельно выбирает какие программные модули из доступных в ПО распознавания и обработки изображения требуются в каждом конкретном случае на различных производствах. Развиваются библиотеки компонент компьютерного зрения, позволяющего облегчить создание программного обеспечения для СКЗ. Примером такой библиотеки может служить Open Source Computer Vision Library (OpenCV).

В данную библиотеку, написанную на языке Си под ОС Windows и Linux, входит свыше 500 функций, обеспечивающих компьютеру восприятие глубины пространства. OpenCV обеспечивает интерфейс для импорта функций библиотеки в пакет Matlab, наиболее широко используемый в промышленных и академических исследовательских разработках, связанных с компьютерным зрением [3].

На основе данной библиотеки успешно реализованы различные типы систем компьютерного зрения в различных областях промышленности. Открытость данной библиотеки способствует увеличению охвата широкого спектра задач и распространению систем компьютерного зрения.

Этому способствует и удешевление материальной базы систем компьютерного зрения. Полнофункциональная система машинного зрения стоит теперь в 10-20 раз дешевле нежели например образцы 1990-2000-х годов, а простую программно-аппаратную систему распознавания промышленных образцов можно приобрести за 2-3 тыс. долл., при этом львиная часть суммы уйдет на ПО. Столь резкое снижение себестоимости произошло благодаря массовому распространению мощных процессоров и удешевлению компьютеров и видеокамер[1]. Например, цветную цифровую видеокамеру, подключаемую к обычному USB-порту, можно в любом компьютерном магазине приобрести в розницу за 30-50 долл. Соответственно, не нужна специальная плата ввода. Поэтому основной акцент в совершенствовании СКЗ приходится на ПО [1].

На основании вышеизложенного можно сделать вывод, что системы компьютерного зрения являются современным средством повышения качества и скорости производства, имеют четко выраженную тенденцию к усовершенствованию и удешевлению программно-аппаратной базы, что в свою очередь ведет к широкому распространению таких систем в различных областях промышленности.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Бобровский С. Когда машины прозреют. / С. Бобровский // PCWEEK/RE (Компьютерная неделя). – 2004. - №5 (419)
2. Шапиро Л., Стокман Дж. Компьютерное зрение.: Пер. с англ. – М.: Бином. Лаборатория знаний, 2009. – 752 с.
3. Берд К. В объеме и красках / К. Берд // Домашний компьютер. – 2002. - №2

П.С. ПОКРОВСКИЙ

Рязанский государственный радиотехнический университет

**УСТОЙЧИВАЯ ПРОЦЕДУРА СИНТЕЗА СИГНАЛЬНОГО
СОЗВЕЗДИЯ ДЛЯ СОЗДАНИЯ УНИВЕРСАЛЬНОЙ СИСТЕМЫ
ПЕРЕДАЧИ ИНФОРМАЦИИ**

Разработаны устойчивые процедуры синтеза сигнальных созвездий, позволяющие изменять соотношение между энергетической эффективностью и устойчивостью к воздействию нелинейных искажений, возникающих в приемо-передающем тракте.

Введение. Значительная часть современных подходов к синтезу сигнальных созвездий (СЗ) спектрально эффективных видов модуляции учитывает лишь требование минимума средней мощности радиосигнала при заданном значении пропускной способности радиоканала системы передачи информации (СПИ) (ТQAM и I-TQAM) [1]. Для спутниковых СПИ необходимо дополнительно учитывать требования к нелинейным искажениям. В настоящее время разработаны методы организации устойчивых к искажениям спутниковых каналов СПИ на основе APSK-сигналов [2]. Недостаток данного подхода заключается в жесткой привязке к характеристикам тракта передачи. Альтернативным методом борьбы с нелинейными искажениями является минимизация величины пик-фактора радиосигналов [3]. Целесообразность данного подхода видится в возможности разработки единой процедуры синтеза СЗ для различных сфер применения СПИ.

Цель работы – разработка процедуры синтеза СЗ для создания универсальной СПИ.

Постановка задачи. В случае уверенного приема пропускная способность СПИ определяется расстоянием $2d$ между ближайшими соседними точками СЗ ($\mathbf{S} = \{s_k\} \quad k = \overline{1, M}$) и процедура синтеза сводится к задаче размещения на комплексной плоскости непересекающихся окружностей радиуса d [1].

Помимо учета требований к нелинейным искажениям для повышения энергетической эффективности СЗ необходимо более плотно использовать динамический диапазон передающего тракта [3]. Оценку эффективности использования динамического диапазона передающего тракта целесообразно производить с учетом плотности упаковки описанных выше окружностей, т.е.

$$G\{s(t)\} = Md^2 / \max_{k=1, M} \{(|s_k| + d)^2\}. \quad (1)$$

По аналогии с [3], пик-фактор можно представить в виде:

$$\Pi\{s(t)\} = M \max_{k=1, M} \{|s_k|\} / \sum_{k=1}^M |s_k|^2. \quad (2)$$

Таким образом, задача синтеза СЗ имеет вид:

$$\min_{S \in C^M} \Pi\{s(t)\} \text{ и } \min_{S \in C^M} 1/G\{s(t)\} \text{ при } |s_k - s_r| \geq 2d \quad \forall k, r = \overline{1, M}. \quad (3)$$

Применение характеристик Π и G в задаче (3) малоэффективно, так как использование метрики Чебышевского пространства приводит к большому числу локальных оптимумов [4]. Поэтому встает задача обоснования критериев качества, косвенно влияющих на уровень Π и G .

Обоснование целевой функции. Среднюю мощность СЗ можно представить в виде суммы критериев $K_1 = D\{|s_k|\}$ и $K_2 = E\{|s_k|\}^2$ ($D\{\bullet\}$, $E\{\bullet\}$ – операции вычисления дисперсии и математического ожидания соответственно). Здесь критерий K_1 позволяет косвенно влиять на пик-фактор, так как является количественной характеристикой разброса точек СЗ относительно среднего значения. Управление слагаемым K_2 в совокупности с ограничением на d обеспечивает группирование сигнальных точек вблизи начала координат, т.е. косвенно воздействует на параметр G .

Таким образом, задачу (3) можно представить в виде:

$$\min_{S \in C^M} J\{S\} \text{ при } |s_k - s_{k+r}| \geq 2d, \quad \forall k = \overline{1, (M-1)} \text{ и } \forall r = \overline{k, M}, \quad (4)$$

где $J\{S\} = (1 - \lambda)K_1 + \lambda K_2$, коэффициент $\lambda \in [0; 1]$ позволяет при синтезе изменять вес критериев K_1 , K_2 для адаптации СПИ к решению широкого круга задач.

Устойчивая процедура синтеза СЗ. Постановка (4) представляет задачу M -мерной оптимизации нелинейной целевой функции $J\{S\}$ на невыпуклом множестве допустимых решений, заданном нелинейными ограничениями. Наиболее эффективным подходом в данном случае является метод с модифицированной функцией Лагранжа (ММФЛ) [5], однако применение его при решении задачи (4) выявило неустойчивость получаемых решений – при заданном значении коэффициента λ и различных начальных условиях S_0 результатом оптимизации являются различные СЗ $S(\lambda)$.

Неустойчивость ММФЛ объясняется тем, что в плотной структуре СЗ перемещение одного элемента приводит к движению множества соседних точек. Т.е. целесообразно одновременное ужесточение

требований на выполнение всех ограничений задачи (4). Для чего предлагается использовать функцию штрафа вида:

$$L_1\{\mathbf{S}, \alpha_j, w\} = \sum_{k=1}^{M-1} \sum_{r=k}^M \left(\max\{d(1+\alpha_j) - |s_k - s_{k+r}|/2, 0\} / \alpha_j \right)^{2w}, \quad (5)$$

где $\alpha_j > 0$ - коэффициент, задающий жесткость накладываемых ограничений на j -й итерации ($\alpha_{j-1} > \alpha_j$); $w \in \mathbb{N}$.

Результаты исследований, представленные на рисунке 1, показали, что наибольшая устойчивость процедуры синтеза наблюдается при $w=3$ (снижение разброса Π , G на 50 % относительно ММФЛ). Величины разброса пик-фактора $\delta\Pi$ и плотности упаковки δG оценивались выражениями вида:

$$\delta\Pi = \int_0^1 \sqrt{D\{\Pi\{\mathbf{S}(\lambda)\}\}} d\lambda, \quad \delta G = \int_0^1 \sqrt{D\{G\{\mathbf{S}(\lambda)\}\}} d\lambda.$$

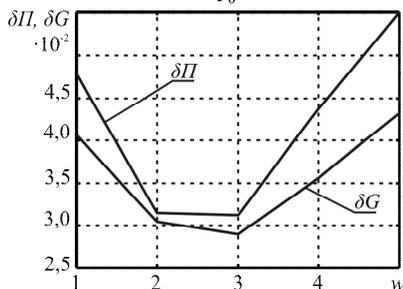


Рис. 1. Зависимости величин $\delta\Pi$ и δG от числа w

Для анализа свойств СЗ, синтезируемых при $w=3$, проведены эксперименты по оптимизации 20 случайных наборов \mathbf{S}_0 при $M=16$, $d=1$. Полученные зависимости величин Π и G от коэффициента λ представлены на рисунке 2, где сплошными линиями показаны средние значения, а точками – границы разброса.

Анализ полученных зависимостей выявил точки, для которых процедура синтеза выдает устойчивые решения. Процедура позволила получить СЗ I-TQAM ($\lambda \approx 0,5$) и PSK-16 ($\lambda \approx 0,0$), что говорит об адекватности предложенного подхода.

СЗ, соответствующее $\lambda = 0,16$ – глобальному максимуму $G(\lambda)$ (см. рисунке 3), объединяет достоинства известных схем I-TQAM и APSK (улучшение по пик-фактору +15 % и -11 %, по средней энергии -6 % и +10 % соответственно).

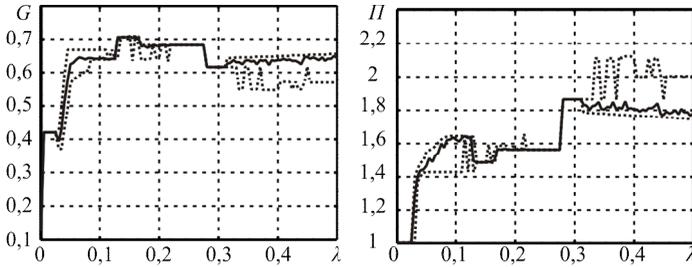


Рис. 2. Зависимости величин Π и G от коэффициента λ

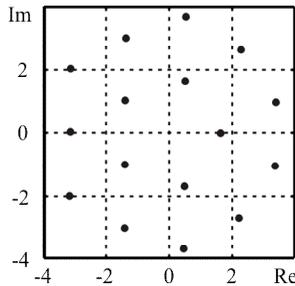


Рис. 3. Синтезированное сигнальное созвездие ($\lambda = 0,16$)

Выводы. Разработанная процедура синтеза позволяет изменением величины λ варьировать энергетическую эффективность и устойчивость к воздействию нелинейных искажений получаемых СЗ. Предложенный подход дает возможность формирования СЗ объемом 32 и более элементов с заданными характеристиками, обеспечивая адаптацию СПИ к решению широкого круга задач.

Работа выполнена в рамках ФЦП ННПКИР (ГК №П501 от 13.05.2010 г. и №16.740.11.0269 от 24.09.2010 г.) при поддержке гранта Президента РФ МК-5000.2010.9 (договор №02.120.11.5000-МК).

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. S.-J. Park and M.-K. Byeon. Irregularly Distributed Triangular Quadrature Amplitude Modulation // PIMRC, 2008. – Режим доступа: <http://202.194.20.8/proc/PIMRC2008/content/papers/1569106953.pdf>.

2. K. P. Liolis and etc. Amplitude Phase Shift Keying Constellation Design and its Applications to Satellite Digital Video Broadcasting/ Digital Video, edited by F. De Rango, Croatia: INTECH, 2010. – 500 p.

3. Феер К. Беспроводная связь. Методы модуляции и расширения спектра: Пер. с англ.– М.: Радио и связь, 2000. – 520с.

4. Гуткин Л.С. Теория оптимальных методов радиоприёма при флуктуационных помехах. – М.: Сов. радио, 1972. – 448 с.

5. Моисеев Н.Н., Иванилов Ю.П., Столярова Е.М. Методы оптимизации. – М.: Наука, 1978. – 352 с.

В.А. РЕВУЦКИЙ

Рязанский государственный радиотехнический университет

АЛГОРИТМ КЛАССИФИКАЦИИ ПОМЕХОУСТОЙЧИВЫХ КОДОВ

Предложен алгоритм классификации помехоустойчивых кодов (ПК), позволяющий определять тип ПК, содержащийся в анализируемой последовательности (АП). Показано, что предложенный алгоритм устойчив к появлению ошибок в канале связи.

Введение. На настоящий момент в современных спутниковых системах передачи информации используется множество стандартов, в состав которых входят различные компоненты [1, 2, 3].

Возможны ситуации, когда необходимо организовать сеанс радиосвязи, не имея полной информации о компонентах стандарта связи. Подобная неопределенность может также иметь место при решении задач радиомониторинга системы передачи информации [3]. При этом одним из неизвестных составляющих могут быть тип и параметры ПК. Данным положением обусловлена актуальность разработки алгоритма классификации ПК.

Важно, что одной из составляющих задачи классификации ПК является задача определения его типа.

Цель работы. Разработка алгоритма классификации по типу ПК в интересах решения задачи определения структуры и параметров помехоустойчивого кодера на основе анализа выходной последовательности.

Постановка задачи. Известно [3,4,5], что кодовые слова блочного кода не пересекаются во времени и практически статистически независимы. Сверточный кодер характеризуется преемственностью собственных состояний, что приводит к сильной статистической зависимости кодовых слов на его выходе.

Так как множество наиболее распространенных ПК можно разделить на блочные и сверточные, то задача определения типа кода сводится к проверке 2-х гипотез: H_1 - код сверточный, H_2 - код блочный. При этом для проверки всех гипотез достаточно подтвердить или опровергнуть только одну [6]. Исходя из структуры последовательности на выходе сверточного кодера - I , можно задать статистику $g(I)$

[6], свойства которой указывают на подтверждение или опровержение гипотезы H_1 .

Ввиду статистической зависимости состояний сверточного кодера a, b, c, d, e, \dots , для получения адекватной статистики $g(I)$ необходимо выделить в соответствующие моменты времени, сегментов длины $n_{\text{сун}} = n_{\text{ист}}$ из I . Следовательно, имеет смысл оценивать статистические свойства последовательности сегментов длины $X \cdot n_{\text{сун}}$, где X – целое положительное число. Данные сегменты соответствуют сериям состояний кодера, которые в свою очередь статистически зависимы. В качестве функции $g(I)$ выбирается функция $g(I, n_{\text{сун}}, X)$ и считается, что моменты смены состояний кодера известны. Данная функция является 2-мерным законом распределения вероятности последовательного появления различных сегментов длины $L = X \cdot n_{\text{сун}}$. Т.е. $g(I, n_{\text{сун}}, X) = W(N_1, N_2)$, где N_1 и N_2 последовательные непересекающиеся во времени сегменты длиной $X \cdot n_{\text{сун}}$ бита из I . Для правильного выделения сегментов из АП необходимо проверить различные гипотезы о значении параметра n сверточного ПК: $H_{1.1} - n = 2; H_{1.2} - n = 3; H_{1.3} - n = 4$.

Обоснование структуры алгоритма. Далее на рисунках 1,а и 1,б приведены функции $W(N_1, N_2)$ при $n_{\text{сун}} = 2, X = 4$ для последовательности сверточного ПК с параметрами $m = 5, n = 2, k = 1$ и последовательности двоичного БЧХ кода с параметрами $n = 63, k = 36$ соответственно.

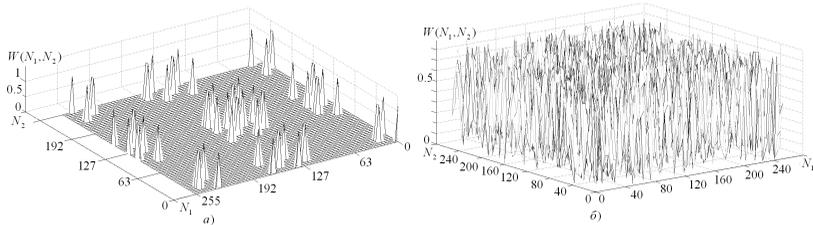


Рис. 1. ФПВ $W(N_1, N_2)$ для а) сверточного ПК с параметрами (5.2.1) б) блочного ПК с параметрами (63. 36)

Из анализа рисунка 1 следует, что закон распределения $W(N_1, N_2)$ для последовательностей сверточного кода имеет множество ярко выраженных пиков в отличие от последовательностей блочного ПК и случайной последовательности. Данный эффект имеет место при $X \geq 4$.

Для оценки статистической взаимосвязи между сегментами АП предлагаются следующие числовые критерии:

1) Разность средних значений по множеству локальных максимумов и минимумов массива значений $W_{\text{норм}}(N_1, N_2)$: $K_1 = \overline{\{R_{\text{max}}\}} - \overline{\{R_{\text{min}}\}}$, где $W_{\text{норм}}(N_1, N_2) = W(N_1, N_2) / \overline{W}(N_1, N_2)$ - зависимость $W(N_1, N_2)$, нормированная к среднему значению. Множества $\{R_{\text{max}}\}$ и $\{R_{\text{min}}\}$ состоят из элементов $W_{\text{норм}}(N_1, N_2)$, превышающих и не превышающих значение «1» соответственно.

2) Отношение общего числа элементов в $W_{\text{норм}}(N_1, N_2)$ к числу элементов в $\{R_{\text{max}}\}$: $K_2 = 2^{2 \cdot X} / V(\{R_{\text{max}}\})$.

Далее необходимо рассмотреть зависимость данных критериев от типа и параметров ПК, содержащегося в АП.

Экспериментальная часть. В эксперименте по получению массивов значений $W_{\text{норм}}(N_1, N_2)$ использовались ПК различного типа и параметров. На выходе канала передачи информации, как правило, происходят ошибки детектирования переданных символов [3...6], поэтому рассматриваемые ПК приведены в следующей таблице совместно с критериями K_1 и K_2 в случае наличия и отсутствия ошибок.

Параметры (m.n.k) сверточного ПК	Ошибки отсутствуют $P_{\text{ош}} = 0$.				Ошибки присутствуют $P_{\text{ош}} = 10^{-2}$			
	$X = 3$; $n_{\text{сум}} = 3$.		$X = 4$; $n_{\text{сум}} = 2$.		$X = 3$; $n_{\text{сум}} = 3$.		$X = 4$; $n_{\text{сум}} = 2$.	
	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2
5.3.1	23	22.3	25.4	25.6	19.9	20.1	21.4	21.6
8.3.1	15.1	15.2	18.3	18.1	11.1	10.8	15.3	14.9
3.2.1	16	16	18	17.9	13	12.7	14	13.9
5.2.1	10.1	10.9	13.1	12.9	6	6.5	9.5	9.9
7.2.1	5.9	6.1	3.9	4	4.7	4.9	3	3.2
3.3.2	15.6	16	8	8.1	11.2	11.8	6.1	5.7
Параметры (n.k) блочного БЧХ кода	$X = 3$; $n_{\text{сум}} = 3$.		$X = 4$; $n_{\text{сум}} = 2$.		$X = 3$; $n_{\text{сум}} = 3$.		$X = 4$; $n_{\text{сум}} = 2$.	
	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2
63.36	0.5	2.1	0.11	0.35	2.1	0.65	1.3	0.12
127.64	0.5	2.1	0.11	0.5	1.1	0.78	0.7	1.05
255.131	0.48	2	0.115	0.48	0.33	0.71	1.7	0.1
Параметры (n.k) блочного кода Риды-Соломона								
15.7	0.25	1.52	0.39	0.73	1.22	0.11	0.3	0.19
31.15	0.18	1.77	0.38	0.12	1.07	0.08	0.44	0.02

63.33	0.12	0.88	0.52	0.43	0.5	0.51	0.33	0.74
128.64	0.45	0.7	0.02	1.15	0.49	0.9	0.03	0.02
255.127	0.45	0.77	0.02	0.43	0.73	0.61	0.68	0.13
255.223	0.45	0.8	0.02	0.6	0.15	0.32	0.15	0.72
Случайная последовательность	0.65	0.41	0.44	0.5	0.32	0.38	0.61	0.51

Далее графически (рисунок 2а,б) на плоскости критериев (K_1, K_2) представлены последовательности, содержащие рассматриваемые ПК для случая совпадения истинного и гипотетического параметров $n_{\text{гип}} = n_{\text{ист}}$.

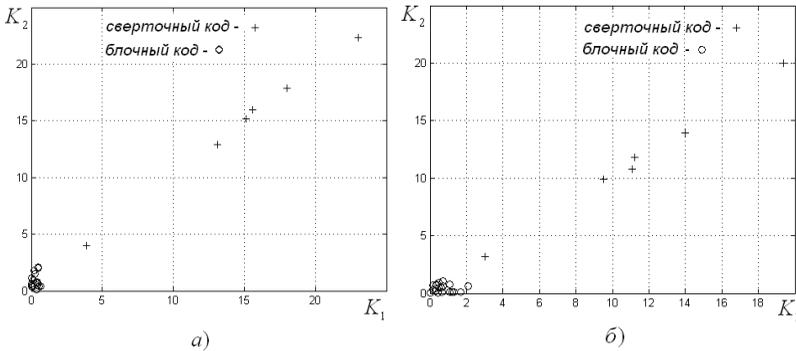


Рис. 2. Расположение на плоскости критериев (K_1, K_2) последовательностей блочных и сверточных ПК с различными параметрами а) в отсутствие ошибок, б) при наличии ошибок вероятности $P_{\text{ош}} = 10^{-2}$

Вывод 1. Из анализа рисунка 2 следует, что последовательности сверточного ПК при полном совпадении гипотетического и истинного параметра n на плоскости критериев (K_1, K_2) располагаются в обособленной области: $K_1 \geq 2.8; K_2 \geq 3$. Данная область не содержит случайных последовательностей и последовательностей блочного ПК, следовательно, для повышения надежности алгоритма необходимо использовать оба предложенных критерия K_1 и K_2 . При несовпадении гипотетического и истинного параметра n и значении $X \geq 3$ последовательности сверточного ПК не попадают в область расположения последовательностей блочного кода и случайной последовательности.

Вывод 2. Наличие ошибок в канале передачи информации приводит к смещению последовательностей, содержащих сверточные ПК, по направлению к «0» и сохранению их в обособленной области $K_1 \geq 2.6; K_2 \geq 2.1$. При этом свойства, указанные в выводе 1 сохраняются.

С учетом выводов, сделанных по результатам экспериментальных исследований, логическая схема предложенного алгоритма классификации ПК имеет следующий вид (рисунок 3).

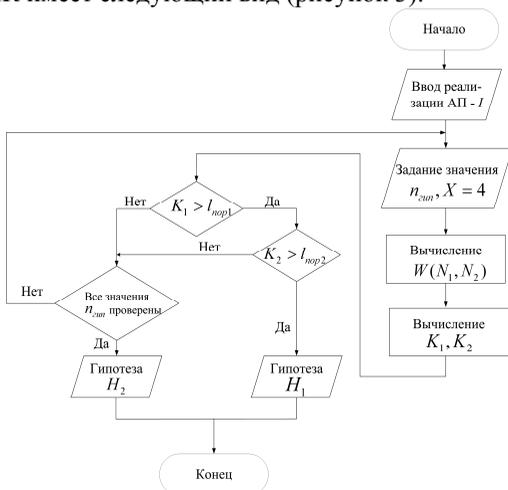


Рис. 3. Логическая схема алгоритма

В данной схеме с учетом $P_{ош} \leq 10^{-2}$ пороговые значения составляют $l_{пор1} = 2.6$ и $l_{пор2} = 2.1$.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Proximity-1 Space Link Protocol—Coding and Synchronization Sublayer. Blue Book. Issue April 2003. Электронный ресурс. Режим доступа: <http://public.ccsds.org/publications/SLS.aspx>
2. TC Synchronization and Channel Coding. Blue Book. Issue 1. September 2003. Электронный ресурс. Режим доступа: <http://public.ccsds.org/publications/SLS.aspx>
3. Бернард С. Цифровая связь, Пер. с англ. – М. : Издательский дом “Вильямс”, 2003. – 1104с
4. Блейхуд Р. Теория и практика кодов, контролирующих ошибки. М.: Мир, 1986 – 576с.
5. Никитин Г.И., Сверточные коды. – СПб.:СПбГУАП, 2001. – 80с.
6. Г. Ван Трис, Теория обнаружения, оценок и модуляции, Том 1, под ред. проф. В.И. Тихонова, Масачусетский Технологический Университет, 1972 г.

В.Е. РУДАКОВ, С.В. СКВОРЦОВ

Рязанский государственный радиотехнический университет

АВТОМАТИЗИРОВАННОЕ МОДУЛЬНОЕ ТЕСТИРОВАНИЕ ПРОГРАММ

Рассмотрены основные идеи модульного тестирования. Показана необходимость использования модульных тестов при разработке сложных программных систем.

Идея модульного тестирования состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже оттестированных местах программы, а также позволяет быстро обнаружить и устранить ошибки.

Модульное тестирование заключается в изолированной проверке каждого отдельного элемента путем запуска тестов в искусственной среде, где элемент - наименьший компонент, который можно скомпилировать. Каждый элемент тестируется изолированно, поэтому необходимо использовать драйверы и заглушки.

Драйверы - это модули тестов, которые запускают тестируемый элемент, а заглушки заменяют недостающие компоненты. Они вызываются соответствующими элементами и выполняют следующие действия:

- возвращаются к элементу, не выполняя никаких других действий;
- отображают трассировочное сообщение и в некоторых случаях предлагают тестировщику продолжить тестирование;
- возвращают постоянное значение или предлагают тестировщику самому ввести возвращаемое значение;
- осуществляют упрощенную реализацию недостающей компоненты;
- имитируют исключительные или аварийные условия.

Целью модульного тестирования является получение работоспособного кода с наименьшими затратами. Его применение оправдано тогда и только тогда, когда оно дает больший эффект, нежели другие методы.

Отсюда следует, что нет смысла писать тесты на весь код. Некоторые ошибки проще обнаружить на более поздних стадиях. Написание тестов для кода, потенциально подверженного изменениям, более выгодно, чем для кода, изменение которого не предполагается. Слож-

ная логика меняется чаще, чем простая. Следовательно, в первую очередь имеет смысл писать модульные тесты на сложную логику. А на простую логику писать позднее или вообще тестировать другими методами.

«Хороший» модульный тест должен удовлетворять следующим условиям:

- должен выполняться автоматически и работать быстро;
- его можно повторить несколько раз;
- его легко можно запустить в любой момент времени жизненного цикла проекта;
- любой член проектной группы может запустить его.

Укажем преимущества, которые дает использование автоматизированных модульных тестов.

Во-первых, *тесты предотвращают появление ошибок в новом коде*. Ошибки выявляются и исправляются в самом начале разработки, а количество пропущенных ошибок в новом коде сокращается. Это обеспечивается следующими факторами:

- тесты являются ранними клиентами кода и позволяют выявлять многие ошибки сразу;
- тесты заставляют разработчика писать код мелкими шагами и больше внимания уделять программе, которую он пишет. Если разработчик чрезмерно ускоряется - то шанс допустить ошибку значительно повышается;
- тесты позволяют убедиться в работоспособности кода на самых ранних этапах разработки, когда другие части системы еще не готовы.

Во-вторых, *тесты позволяют рефакторить код без риска его нарушить*.

При внесении изменений в хорошо протестированный код риск появления новых ошибок значительно ниже. Если новая функциональность приводит к ошибкам, модульные тесты, если они конечно есть, сразу же это покажут. Для кода, на который нет тестов, ошибку можно обнаружить спустя значительное время, когда с ним работать будет намного сложнее.

Хорошо протестированный код легко переносит рефакторинг. Уверенность в том, что изменения не нарушат существующую функциональность, придает уверенность разработчикам и увеличивает эффективность их работы. Если существующий код хорошо покрыт тестами, разработчики будут чувствовать себя намного свободнее при внесении архитектурных решений, которые призваны улучшить дизайн кода.

В заключении статьи отметим, что написание автоматизированных модульных тестов, позволяют выявлять ошибки на ранних стадиях разработки, тем самым сокращая время на устранение ошибок. Всегда можно внести изменения в код, написанный в начале проекта, не боясь нарушить уже существующий функционал. Также написание модульных тестов сокращает время разработки программного продукта, так как существенно снижается время на его отладку.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Roy Osherove The art of unit testing with exemples in .NET. Manning, 2009.

Д.С. СЕМИН

Рязанский государственный радиотехнический университет

УНИВЕРСАЛЬНЫЙ АЛГОРИТМ ПОМЕХОЗАЩИЩЕННОГО КОДИРОВАНИЯ НА БАЗЕ НЕЛИНЕЙНОГО ПРЕОБРАЗОВАНИЯ В ПОЛЕ ГАЛУА

Представлен универсальный алгоритм помехозащищенного кодирования, позволяющий осуществлять процедуру внесения избыточности широким набором современных кодов и обеспечивающий информационную и структурную скрытность битового потока с криптостойкостью до 10^{53} ключей. Показано, что применение режима защиты информации приводит к увеличению вероятности ошибки на 0,5...1,5 дБ.

Введение. Задача увеличения помехоустойчивости и скрытности передачи информации в настоящее время является весьма актуальной. Разработано большое число алгоритмов криптографической защиты, не удовлетворяющих дополнительным требованиям помехоустойчивости. Вместе с тем предложены алгоритмы помехозащищенного кодирования [1, 2] (стохастические коды, алгоритм Мак-Элис и его модификации и д.р.), однако недостатками данных алгоритмов является возрастание вычислительной сложности при увеличении криптографической стойкости. В [3] предложен универсальный алгоритм помехозащищенного кодирования, позволяющий уменьшить вычислительные затраты при реализации.

Цель работы. Целью данной работы является исследование свойств помехоустойчивости универсального алгоритма помехозащищенного кодирования, обеспечивающего заданную степень криптографической скрытности.

Описание помехозащищенного алгоритма. Алгоритм основан на внесении нелинейности в фильтр с бесконечной импульсной характеристикой [3]. При этом структурную схему кодирующего устройства можно разделить на рекурсивную и нерекурсивную части, уравнения для которых записываются в виде:

$$c_i = \sum_{j=1}^{p_b} b_{i-j}(v_{i-j}), \text{ и } v_i = a_0(u_i + \sum_{j=1}^{p_a} a_j(v_{i-j})), \quad (1)$$

где u_i - отсчеты входного вектора информационного сообщения, c_i - отсчеты выходного вектора закодированного сообщения, $a_j(x)$ и $b_j(x)$ - функции кодера, p_a и p_b - число функций рекурсивной и нерекурсивной частей кодирующего устройства соответственно. Все описанные выше операции выполняются согласно арифметике полей Галуа [4].

Обратное к данному, декодирующее устройство будет иметь аналогичную структуру, с функциями $a_j^{обп}(x) = b_j(x)$, $b_j^{обп}(x) = a_j(x)$, $j = 1..p$ $a_0^{обп}(x) = b_0^{-1}(x)$, $b_0^{обп}(x) = a_0^{-1}(x)$.

Функции $b_j(x)$ необходимы для внесения избыточности в информационный поток и определяются, как $b_j(x) = g_j x$, где g_j - коэффициенты помехоустойчивого кода. При этом могут формироваться, все известные помехоустойчивые коды, образующая матрица которых, может быть записана в циклической форме [4] (коды Боуза-Чоудхури-Хоквенгема (БЧХ), Рида-Соломона, Рида-Майлера, квадратично-вычетные, Голея, сверточные, некоторые типы Галлагера и др.).

Функции $a_j(x)$ рекурсивной части устройства необходимы для выполнения процедуры защиты информации, и могут задаваться в полиномиальной форме (в частности может использоваться функция произвольного вида):

$$a_j(x) = \sum_{i=1}^d a_{j,i} x^i, \quad (2)$$

где $a_{j,i}$ элементы поля Галуа. При этом число коэффициентов полинома d , определяет число степеней свободы кодирующего устройства, а следовательно криптографическую стойкость всей системы. Также для увеличения информационной скрытности к закодированному слову прибавляется псевдослучайная последовательность (ПСП). Подобный способ защиты информации может быть отнесен к модифика-

ция алгоритма гаммирования [5], криптостойкость которого зависит от периода ПСП.

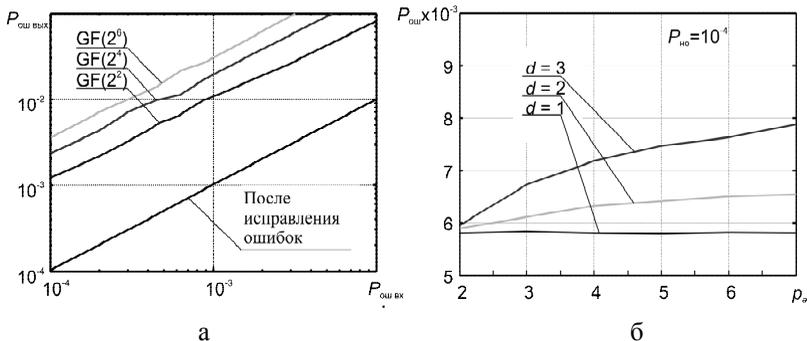


Рис. 1. Результаты имитационного моделирования:

- вероятность ошибки на выходе от вероятности ошибки на входе при работе кодирующего устройства в различных полях Галуа,
- вероятность ошибки на выходе схемы от числа рекурсивных функций, при использовании полиномов степени d

Имитационное моделирование. Имитационное моделирование универсального кодирующего устройства, с целью изучения помехоустойчивых свойств, проводилось для блочных бинарных кодов БЧХ (длиной 63 бит), при использовании ПСП на базе кодового слова длиной 128 бит. На рисунке 1 а изображен график зависимости усредненной вероятности ошибки на выходе системы от вероятности ошибки на входе (после проведения процедуры исправления ошибок) для случаев работы кодирующего устройства в полях Галуа $GF(2^2)$, $GF(2^4)$, $GF(2^6)$. Увеличение вероятности ошибки связано с тем, что появление искаженных символов в закодированном слове \bar{c} , число которых лежит за границей исправляющей способности кода, приводит к эффекту размножения ошибок при обратном криптографическом преобразовании. Вместе с тем увеличение степени с $d=1$ до $d=3$ приводит к значительному увеличению числа ключей с 10^{38} до 10^{53} (рисунок 1 б, при $GF(2^5)$, $\rho_a=5$ и вероятности необнаруживаемой ошибки $P_{\text{но}}=2 \cdot 10^{-4}$) и незначительному увеличению вероятности ошибки на 25%.

Выводы. Таким образом, показано, что применение универсального алгоритма помехозащищенного кодирования позволяет увеличить скрытность информационного потока до 10^{53} ключей при уменьшении помехоустойчивости на $0,5 \dots 1,5$ дБ (для сигналов BPSK в канале с белым Гауссовым шумом при вероятности ошибки 10^{-4}).

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. С.Н. Кириллов, Д.Е. Крысяев, Суан Нгиа Фам. Исследование устойчивости алгоритма Мак-Элис к действию комплекса помех // Электросвязь. - 2008. - N 2. - С. 44-46.
2. Осмоловский С.А. Стохастические коды, исправляющие ошибки с гарантированной точностью. // Системы и средства связи, телевидения и радиовещания. М.: АО "ЭКОС". 2001. - № 2,
3. Кириллов С.Н. Семин Д.С. Помехозащищенный кодер канала связи на основе БИХ-фильтра. Вестник РГРТУ №2 2009. 27 - 30 С.
4. Блейхут Р. Теория и практика кодов, контролирующих ошибки: Пер. с англ. – М.: Мир, 1986. – 576 с.
5. Бабаш А.В., Шанкин Г.П. Криптография. Под ред. В.П. Шерстюка, Э.А. Применко. – М.: СОЛОН-Р, 2002. – 512 С.

Е.Ю. СКОЗ

Рязанский государственный радиотехнический университет

ФОРМАЛИЗАЦИЯ ПРОЕКТНЫХ ПРОЦЕДУР АРХИТЕКТУРНОГО ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

Сформулированы и методически обоснованы подходы групповой интерпретации операционного поля третьего эпистемологического уровня систем с поведением. Реализована модель абстрактной группы в форме таблиц Кэли, аксиоматические критерии принадлежности исходного множества к определенному виду групп и построения их на основе образующих и определяющих соотношений.

Основные проектные процедуры системы в рамках процесса проектирования это функциональный анализ, синтез и системный анализ создают последовательно три архитектурных решения системы. В целом это многостадийный процесс, в котором чередуется синтез и верификация (анализ). Синтез обеспечивает преобразование функциональной архитектуры в проектную, которая расставляет системные элементы, декомпозирует их с последующей идентификацией интерфейсов (внутренних и внешних) и ограничений. Связанные с синтезом работы включают выбор предпочтительного решения или предпочтительной расстановки из множества альтернативных решений и достижения понимания воздействий на затраты, графики работ, характеристики качества и риски. По мере необходимости используется и системный анализ для оценивания альтернатив: для идентификации, оценивания и количественного определения рисков и выбора оптимально-

го подхода к смягчению рисков. Проектная архитектура включает матрицы прослеживаемости и распределения требований, фиксирующее распределение между системными элементами.

Полная системная архитектура включает все проектные архитектуры процессов жизненного цикла и проектные архитектуры проекта.

Стремление упорядочить и унифицировать знания в задачах синтеза архитектурных решений и построить обобщенные архитектуры систем прослеживается в работах ведущих специалистов по информационным технологиям и в стандартах ISO и IEEE. Причем в последнем стандарте формализована эталонная архитектура обобщенного предприятия и методологии GERAM (Generalized Enterprise – Reference Architecture and Methodology) версии 1.6.2. Эта модель обобщенной архитектуры аккумулирует в себе предшествующие модели архитектур типа:

- Simosa (Автоматизированная система управления производством – Архитектура открытых систем);
- GRAI/GIM (Взаимосвязанные графики, результаты и действия – Интегрированная методология);
- PERA (Эталонная архитектура предприятия Пердью).

Эталонная архитектура обобщенного предприятия (GERAM) определяет лишь общие концепции, рекомендуемые для использования в проектах построения и интеграции.

Из вышесказанного вытекает важный вывод – основу синтеза архитектуры составляет идентификация элементов, их взаимное расположение и взаимосвязь. Анализ альтернативных вариантов и их оценка требует разработки соответствующих критериев оценки качества вариантов построения. В связи с чем, наиболее подходящим методом формализации и системного анализа, по нашему мнению, является иерархическая системная модель Дж. Клира, позволяющая синтезировать варианты архитектурных решений и отслеживать динамику изменений в критических точках этапов процесса проектирования информационных продуктов.

Наиболее содержательными уровнями (как показано в предыдущих работах), позволяющими синтезировать архитектурные решения и проводить их системный анализ это:

- порождающие системы с поведением;
- структурированные системы;
- метасистемы.

Продолжая исследование и разработку моделей анализа-синтеза методологического понятия «архитектура» информационных систем масштаба проекта и предприятия в целом, определим ее как описание

(модель) базового расположения и взаимосвязей частей системы (физического либо концептуального объекта либо сущности).

Постановка задачи. Определим исходную систему, как пятерку

$$S = (O, I, I, Q, \varepsilon),$$

где O – проектируемая система объекта, I – конкретная представляющая система, I – обобщенная представляющая система, Q – полный канал наблюдения, ε – канал конкретизации (абстрагирования).

Два компонента (I и ε) представляют интерфейс между конкретной предметной областью и системой.

Поведение – это форма задания ограничений.

Тогда форма представления порождающей системы с поведением в общем виде следующая:

$$F_{GB} = (I, M_G, f_{GB}),$$

где M – маски, определенные на I , f_B – функции поведения, определенная через I и M .

Функция поведения $f_B(C)$ для детерминированной системы определяется относительными частотами по формуле:

$$f_B(c) = \frac{N(c)}{\sum_{\alpha \in c} N(\alpha)},$$

где $N(\alpha)$ – общее число имеющихся выборок из данных по используемой маске. $N(c)$ – число появлений в данных сочетаний по используемой маске. Обобщенный критерий качества систем с поведением – обобщенная нечетность (Шенноновская энтропия) определяется однозначно:

$$H(f(x) | x \in X) = - \sum_{x \in X} f(x) \log_2 f(x),$$

где $f(x)$ – функция поведения

Чтобы иметь возможность производить оценку в рамках последних двух, необходимо ввести структурирование операционного поля, для чего введем групповую формализацию исходного множества как модель с определенными в ней отношениями.

Введем определение конечной группы и группы перестановок, которые наиболее адекватно отвечают методологическим особенностям исходной системы.

Определение 1. Группа называется конечной, если конечно множество ее элементов. Число элементов в множестве конечной группы называется ее порядком.

Определение 2. Подстановной системой n называют взаимно однозначное отображение любого конечного множества, содержащего n элементов на самого себя.

Исходная система, заданная множеством M на уровне порождающих систем, для которых значения выборочных переменных, заданных в форме «ключей качества» $(0,1)$, в общем виде может быть формализована как группа подстановок. Причем, в соответствии с теоремой Кэли, любая конечная группа порядка n изоморфна некоторой группе подстановок системы n .

Учитывая важность введения групповой интерпретации, изложим соображения, послужившие основой для принятия базисной концепции, что исходное операционное поле порождающих систем есть специальная группа – группа подстановок конечного множества $\{1, 2 \dots n\}$.

Во-первых, предварительный анализ состава и механизма формирования операционного поля инвариантного ограничения позволяет сделать вывод, что оно формируется искусственно.

Во-вторых, имея ввиду первое утверждение, включение в рассмотрение всех возможных реализаций выборочных переменных позволит получить в исходное рассмотрение системы все элементы группы подстановок.

В-третьих, первые два допущения открывают возможность сформировать модель группы подстановок, на базе которой исследуется внутренняя семантическая структура со снижением сложности исходной задачи, для чего можно использовать теорему о сопряженности элементов группы.

Имея ввиду вышесказанное, зададим исходное множество значений выборочных переменных инвариантного ограничения определяемого маской для подсистем «Знания» и «Инфраструктура».

S_7	S_8	S_{13}	S_{18}	${}^5\tilde{F}_B$
1	0	1	1	0,12
0	1	1	1	0,29
1	1	0	1	0,18
1	1	1	1	0,18
1	0	1	0	0,06
1	0	0	1	0,12
1	1	1	0	0,06

S_7	S_8	S_{16}	S_{17}	S_{18}	${}^3\tilde{F}_B$
1	0	1	0	1	0,12
0	1	0	1	1	0,18
1	1	1	1	1	0,23
1	1	1	0	1	0,12
1	0	0	1	0	0,06
0	1	1	1	1	0,12
1	1	1	1	0	0,06
1	0	1	1	1	0,06
1	0	0	0	1	0,06

Пронумеруем значения выборочных переменных и представим исходные матрицы в символьном виде без учета последнего столбца.

S_7	S_8	S_{13}	S_{18}
a_1	a_2	a_3	a_4
a_2	a_3	a_4	a_1
a_3	a_4	a_1	a_2
a_4	a_1	a_2	a_3

S_7	S_8	S_{16}	S_{17}	S_{18}
a_1	a_2	a_3	a_4	a_5
a_2	a_3	a_4	a_5	a_1
a_3	a_4	a_5	a_1	a_2
a_4	a_5	a_1	a_2	a_3
a_5	a_1	a_2	a_3	a_4

Приняв за единичные первые множества состояний при отображении их самих в себя φ_i , на группе подстановок, вводится алгебраическая операция умножения.

Задать на M алгебраическую операцию равносильно заданию таблицы, называемой таблицей Кэли.

Переходя на представление элементов группы унитарными матрицами, получим таблицу Кэли для группы S_6 (Табл. 1).

Таблица Кэли обладает свойствами, позволяющими выявить все структурные особенности исходного множества на предмет возможности описания ее групповыми методами.

Пусть σ – цикл и $M_\sigma = \{a_1, a_2, \dots, a_i\}$ где все элементы расположены так, что $a_{i+1} = a_i \sigma$. Каждый цикл σ можно отождествлять с кортежем $(a_1 a_2 \dots a_i)$. Зная этот кортеж, можно однозначно восстановить σ . Зная этот кортеж, можно однозначно восстановить σ . Если σ – произвольная подстановка и $\sigma = \sigma_1 \sigma_2 \dots \sigma_k$ – ее разложение на независимые циклы, то σ_i отвечает кортеж $(a_1^i a_2^i \dots a_{i_i}^i)$ и сопоставив σ последовательности независимых кортежей – циклов можно получить:

$$(a_1^1 a_2^1 \dots a_{i_1}^1) \dots (a_1^k a_2^k \dots a_{i_k}^k).$$

Каждая такая последовательность полностью определяет подстановку σ , и умножение двух подстановок можно реализовать на языке их разложения на циклы.

Важными характеристиками циклического и транспозиционно-го представления элементов группы подставок являются графы преобразований заданного множества (а) $\varphi = b$, где a, b – элементы множества M (точки на плоскости) соединенные стрелкой. Все это и есть граф преобразований φ . Пусть преобразование φ множества $M = \{1, 2, 3, 4, 5, 6\}$ задано $\varphi =$

$\left. \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 2 & 5 & 6 & 2 \end{array} \right\}$. Тогда соответствующий граф преобразований φ (рис.1).

Поскольку $(1) \varphi=3$, точки 1 и 3 соединяются стрелкой в направлении от 1 к точке 3. Аналогично и остальные точки 2, 3, 4, 5, 6. Граф произвольного преобразования φ может состоять в общем случае из одной или нескольких не связанных между собой частей, каждая из которых составляет одно целое (рис. 2). При этом отдельная связная часть графа преобразований φ может состоять лишь из одной точки с «петлей», то есть со стрелкой, которая выходит из этой точки и заканчивается в ней.

Пусть a – такая точка. тогда для $a \in M$ справедливо равенство $a(\varphi)=a$.

Такие элементы называются неподвижными точками преобразования φ . Если же для $a \in M$ условие не выполняется $a(\varphi) \neq a$, то a называется подвижной точкой преобразования φ . Количество подвижных точек преобразования называется степенью этого преобразования. Если φ – некоторое преобразование M и a – произвольный элемент из M , последовательность $(a_0)=a, (a_1) \varphi=a_1, (a_2) \varphi=a_2 \dots (a_n) \varphi=a_{n+1} \dots$ элементов из M называется орбитой элемента a для преобразования φ . В общем случае граф каждой перестановки состоит из определенного числа не связанных между собой циклов (рис. 2). Назовем две перестановки на множестве M взаимно простыми, если их множества подвижных точек не имеют общих элементов.

Определение. Каждая перестановка на конечном множестве M раскладывается в произведение взаимно простых циклов, причем это разложение однозначно с точностью до порядка множителей.

Длиной орбиты называется число ее элементов. Каждую перестановку разлагают на независимые циклы. Пусть v_1 – число циклов из одного символа; v_2 – число циклов из двух символов и т. д. Тогда, поскольку всего затрачивается ровно n символов в любой данной перестановке из S_n , то очевидно

$$1v_1+2v_2+\dots+nv_n=n. \quad (1)$$

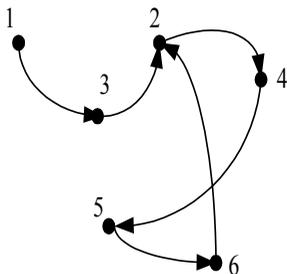


Рис. 1

Таблица 1

	1 0 0 0 1 0 0 0 1	0 1 0 1 0 0 0 0 1	0 1 0 0 0 1 1 0 0	0 0 1 0 1 0 1 0 0	0 0 1 1 0 0 0 1 0
1 0 0 0 1 0 0 0 1	1 0 0 0 1 0 0 0 1	0 1 0 1 0 0 0 0 1	0 1 0 1 0 0 0 0 1	0 0 1 0 1 0 1 0 0	0 0 1 1 0 0 0 1 0
0 1 0 1 0 0 0 0 1	0 1 0 1 0 0 0 0 1	1 0 0 0 1 0 0 0 1	1 0 0 0 0 1 0 1 0	0 0 1 1 0 0 0 1 0	0 0 1 0 1 0 1 0 0
0 1 0 1 0 0 0 0 1	0 1 0 1 0 0 0 0 1	0 0 1 0 1 0 1 0 0	0 0 1 1 0 0 0 1 0	1 0 0 0 0 1 0 1 0	1 0 0 0 1 0 0 0 1
0 0 1 0 1 0 1 0 0	0 0 1 0 1 0 1 0 0	0 1 0 1 0 0 0 0 1	0 1 0 1 0 0 0 0 1	1 0 0 0 1 0 0 0 1	1 0 0 0 0 1 0 1 0
0 0 1 1 0 0 0 1 0	0 0 1 1 0 0 0 1 0	1 0 0 0 0 1 0 1 0	1 0 0 0 1 0 0 0 1	0 1 0 1 0 0 0 0 1	0 1 0 1 0 0 0 0 1
1 0 0 0 0 1 0 1 0	1 0 0 0 0 1 0 1 0	0 0 1 1 0 0 0 1 0	0 0 1 0 1 0 1 0 0	0 1 0 1 0 0 0 0 1	0 1 0 1 0 0 0 0 1

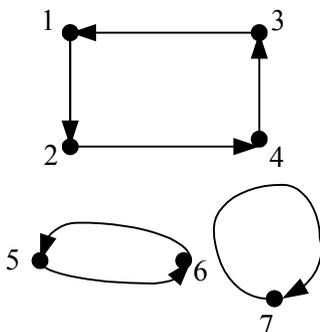


Рис. 2

Все перестановки группы S_n , обладающие одинаковой циклической структурой, образуют некоторый класс сопряженных элементов в S_n . Каждое решение уравнения (1) в положительных целых числах v_1, v_2, \dots, v_n характеризует некоторый класс в S_n . Следовательно, полное число классов группы S_n есть полное число решений уравнения (1). Пусть

$$\begin{aligned} v_1 + v_2 + \dots + v_n &= \lambda_1 \\ v_2 + \dots + v_n &= \lambda_2 \\ &\dots \\ v_n &= \lambda_n. \end{aligned} \quad (2)$$

Очевидно

$$\begin{aligned} \lambda_1 + \lambda_2 + \dots + \lambda_n &= n, \\ \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n &\geq 0. \end{aligned} \quad (3)$$

Тогда справедливо определение. Число классов в группе S_n равно разбиению n на положительные целые числа, удовлетворяющие условию (3). Весом разбиения $[\lambda_1, \lambda_2, \dots, \lambda_n]$ называется число $\rho = \sum_{i=1}^n \lambda_i$. По известному разбиению числа n можно восстановить циклическую структуру перестановки, если воспользоваться соотношениями

$$\begin{aligned} v_1 &= \lambda_1 - \lambda_2 \\ v_2 &= \lambda_2 - \lambda_3 \\ &\dots \\ v_n &= \lambda_n. \end{aligned} \quad (4)$$

Число перестановок в классе, имеющих одну и ту же циклическую структуру (v) равно

$$n(v) = \frac{n!}{v_1!^{v_1} v_2!^{v_2} \dots v_n!^{v_n}} \cdot n^v \quad (5)$$

Пример разбиения группы S_n на сопряженные классы. Прежде разбиваем число n на целые числа $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ при выполнении условия (3). Их всего пять $[4], [3\ 1], [2^2], [2\ 1^2], [1^4]$, поэтому S_4 имеет пять классов. Найдем соответствующие циклические структуры перестановок каждого класса из соотношения (4); например, для разбиения $[3\ 1]$ $\lambda_1=3, \lambda_2=1$ и, согласно (4), $v_1=2, v_2=1$; следовательно, получается циклическая структура $(1^2\ 2^1)$. Число элементов в каждом рассматриваемом классе группы S_4 можно определить по формуле (5), с использованием которой составлена таблица классов группы S_4 .

Разбиение	Циклическая структура	Число элементов в классе
$[4]$	(1^4)	1
$[3\ 1]$	$(1^2\ 2^1)$	6
$[2^2]$	(2^2)	3
$[2\ 1^2]$	$(1^1\ 3^1)$	8
$[1^4]$	(4^1)	6
Полное число элементов:		24

Теорема. Полное число неприводимых представлений данной конечной группы в точности равно числу ее классов.

Итак, каждый класс группы S_4 можно однозначно характеризовать заданием соответствующего разбиения, а индексы этих разбиений удобно использовать для однозначной характеристики неприводимых представлений группы S_4 .

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. IEEE Standard for Application and Management of the Systems Engineering Process. IEEE – Std – 1220 – 1998 (Revision of IEEE Std 1220 – 1992)/
2. *Клир Дж.* Системология. Автоматизация решения системных задач/ Пер. с англ. М.А. Зуева. М.: Радио и связь, 1990. 540 с.
3. *Гроссман И., Магнус В.* Группы и их графы. Пер с англ. Цукерман Г.М. М.: Мир, 1971. 247 с.

Е.Ю. СКОЗ, Ю.М. ЦЫЦАРКИН

Рязанский государственный радиотехнический университет

ПРЕДСТАВЛЕНИЕ РАЗБИЕНИЙ ПРОИЗВОДЯЩИМИ ФУНКЦИЯМИ

Разработан критерий оценки семантических аспектов системы в форме производящих функций циклического разбиения групп перестановок. Разработаны критерии оценки качества на базе абсолютных чисел Стирлинга.

В комбинаторных задачах, при подсчете числа объектов, при наличии ограничений, искомым решением является последовательность $a_0, \dot{a}_1, \dot{a}_2, \dots$, где \dot{a}_k - число объектов «размерности» k . В этом случае удобно последовательности $\dot{a}_0, \dot{a}_1, \dot{a}_2, \dots$, поставить в соответствие формальный ряд

$$A(k) = \sum_{k=0}^{\infty} a_k x^k, \quad (1)$$

называемый производящими функциями для последовательностей. Название «формальный ряд» означает, что (1) трактуется как удобная запись последовательности. В данном случае несущественно для каких значений переменной x он сходится. Поэтому не будем вычислять значения такого ряда для переменной x , а будем выполнять некоторые операции на рядах и определять коэффициенты при отдельных степенях переменной x .

Представление групповых циклических структур абсолютными числами Стирлинга первого типа. Функция поведения $f_B(c)$ вероятностной системы классического типа – это класс вероятностных мер, определяемых относительными частотами состояний c , однако возможности анализа системы позволяют затронуть лишь синтаксические аспекты выборки. Представление в форме циклической групповой выборки с разложением в производящие рекуррентные соотношения позволяют анализировать и семантику выборки.

$$\begin{aligned} \sum_{\gamma \in \tilde{A}_x} (x(\gamma))^{k+1} &= \left| \{(x_1, x_2, \dots, x_{k+1}, \gamma) \in \binom{x+1}{j=1} P; \gamma \in \bigcap_{j=1}^{k+1} \tilde{A}_{x_j}\} \right| = (x_x(\gamma) + 1)^k = \sum_{x \in P} \sum_{\gamma \in \tilde{A}_x} (x(\gamma))^k = \\ &= \sum_{x \in P} \sum_{\gamma \in \tilde{A}_x} \sum_{\alpha \in P, \gamma \in \tilde{A}_\alpha} \sum_{i=0}^k \binom{k}{i} (x_\alpha(\gamma))^{k-1} = \sum_{x \in P} \sum_{i=0}^k \binom{k}{i} \sum_{\gamma \in \tilde{A}_x} (x_x(\gamma))^{k-1} =: A \end{aligned}$$

Так как Γ_x для $1 \leq m \leq k$ транзитивно оперирует в качестве базисной (ус-

тойчивой) подгруппы от Γ_m многократно, то далее основываясь на предположениях математической индукции:

$$\begin{aligned} A &= \sum_{x \in P} \left(\sum_{i=0}^{k-1} \binom{k}{i} \right) B(k-i)[\tilde{A}_x] + [\tilde{A}_x] = \sum_{x \in P} \left(\sum_{i=0}^{k-1} \binom{k}{i} \right) B(k-i)[\tilde{A}_x] + B(0)(\tilde{A}_x) = \\ &= \sum_{x \in P} \left(\sum_{i=0}^k \binom{k}{i} \right) B(i)[\tilde{A}_x] = \sum_{x \in P} B(k+1)[\tilde{A}_x] = B(k+1)[\tilde{A}_x] = B(k+1)[\tilde{A}] \end{aligned}$$

Так как для Γ при условии $i \leq k$, I – многократно транзитивно оперирует, следует утверждение: Высказывание теоремы 1 не переносится на k – многократные транзитивные множества перестановок.

Циклы перестановок.

1. Числа Стирлинга первого типа (вида). Пусть $P - V$ – множество и $k \in N_0$. Буквами $S(v, k)$ обозначим число перестановок из S_p , которые распадаются на k циклов, или другими словами траектории которых образуют k – разбиение от P . Числа называются беззначными или абсолютными числами Стирлинга первого вида (типа).

Теорема. Числа Стирлинга 1-го вида $S(v, k)$ без знака удовлетворяют для $k \in N$ рекуррентной формуле

$$S(v+1, k) = s(v, k-1) + vS(v, k) \quad (2)$$

при начальных условиях $s(v, 0) = (0, k) = 0$ для $v, k \in N$ и $s(0, 0) = 1$.

Доказательство. Очевидно числа Стирлинга 1-го вида выполняют начальные условия. Отсюда предположим $v \geq 0$ и $k \geq 1$, тогда согласно определению получаем $S(v, k-1)$ перестановки $\varphi \in S_{v+1}$ с $\varphi(v+1) = v+1$, которые распадаются на циклы k .

Если теперь в $\varphi \in S_v$ есть перестановка с траекториями k , тогда для $i \in Z_v$ перестановка $\psi \in S_{v+1}$ определенная через

$$\psi_i(j) := \varphi(j) \text{ для } j \in Z_v \setminus \{i\}$$

$$\psi_i(i) := v+1$$

$$\psi_i(v+1) := \varphi_i(i)$$

имеет точно такие же k -траектории и получается $\psi_i \neq \psi_m$ для $i \neq m$. Отсюда, следовательно, получим из каждой $S(v, k)$ перестановки из S_v , которая распадается на k -циклы, такие v -перестановки из S_{v+1} , которые также распадаются на k -циклы, и для которых $v+1$ не является неподвижной точкой. Итак, если такие же перестановки получаются из S_{v+1} , для которых $v+1$ не является неподвижной точкой, следует вышеуказанная рекуррентная формула.

В качестве чисел Стирлинга 1-го вида обозначим

$$S(v, k) := (-1)^{v+k} s(v, k) \quad (2.1)$$

Следует теорема: Числа Стирлинга 1-го вида $S(v, k)$ удовлетворяют для $k \in N$ рекуррентной формуле

$$s(v+1, k) = S(v, k-1) - vS(v, k) \quad (*)$$

при начальных значениях $s(v, 0) = s(0, k) = 0$ для $v, k \in N$ и $S(0, 0) = 1$.

Теорема. Для $v \in N_0$ по теореме определится полином $(Z)_v$. То есть

$$\sum_{k=0}^{\infty} s(v, k) \cdot Z^k = \sum_{k=0}^v s(v, k) \cdot Z^k = (Z)_v \quad (**)$$

обычная производящая функция.

Доказательство. Для $v=0$ вышеприведенное утверждение очевидно правильно. Теперь пусть $v \in N$. Обозначим постоянный коэффициент полинома $(Z)_v$ через $s'(v, k)$, тогда вследствие $s'(v+1, 0) = s'(v, 0) = 0$ получается

$$\begin{aligned} \sum_{k=0}^{\infty} s'(v+1, k)r^k &= (Z)_{v+1} = (Z-v)(r)_v = (Z-v) \sum_{k=1}^{\infty} s'(v, k)r^k = \\ &= \sum_{k=1}^{\infty} s'(v, k)r^{k+1} - \sum_{k=1}^{\infty} vs'(v, k)r^k = \sum_{k=2}^{\infty} s'(v, k-1)r^k - \sum_{k=1}^{\infty} vs'(v, k)r^k = \\ &= \sum_{k=1}^{\infty} (s(v, k-1) - vs'(v, k))r^k \end{aligned}$$

Вид чисел Стирлинга 1-го типа. При сравнении коэффициентов оказывается, что числа $s(v, k)$ и $s'(v, k)$ удовлетворяют одинаковой рекуррентной формуле. При соблюдении исходных условий из (*) для чисел $s'(v, k)$ получается $s'(v, k) = s(v, k)$ и следует утверждение.

Теорема. Для $v \in N_0$ следует:

$$\sum_{k=0}^{\infty} s(v, k)r^k = \sum_{k=0}^v s(v, k) = v!$$

Доказательство. Подставим в (**) $r := v$, вследствие чего получим:

$$(Z)_v = v! = |S_v| = \sum_{k=0}^v s(v, k)$$

В качестве примеров приводим следующие тождества

$$s(v, v) = 1 \text{ для всех } v \in N_0$$

$$s(v, v-1) = -1 \binom{v}{\varepsilon} \text{ для всех } v \in N$$

$$s(v, 1) = (v-1)! \text{ для всех } v \in N$$

$$\sum_{k=0}^{\infty} s(v, k) r^k = (-1)^v (-r)^v \text{ для всех } v \in N_0.$$

Таблица 1. Числа Стирлинга 1-го вида $s(v, k)$

$v \backslash k$	0	1	2	3	4	5	6	7	8	9	10
0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	-1	2	-6	24	-120	720	-5040	40320	-382880
2	0	0	1	-3	11	-50	274	-1764	13068	-109584	1026576
3	0	0	0	1	-6	35	-225	1624	-13132	118124	1172700
4	0	0	0	0	1	-10	85	-735	6769	-67284	723680
5	0	0	0	0	0	51	-15	175	-1960	22449	-269325
6	0	0	0	0	0	0	1	-21	322	-4536	63273
7	0	0	0	0	0	0	0	1	-28	546	-9450
8	0	0	0	0	0	0	0	0	1	-36	870
9	0	0	0	0	0	0	0	0	0	1	-45

Циклы перестановок.

Из вышеприведенных формул следует – прямые и обратные числа Стирлинга 1-го и 2-го вида, образующие инверсионную пару $s(v, k)$ и $S(v, k)$. В частности, получается формула инверсий Стирлинга

$$\sum_{i=k}^V s(v, k) \cdot S(v, k) = \sum_{i=0}^V s(v, i) \cdot S(i, k) = \sigma_{v, k}.$$

Указатель циклических групп подстановок. Пусть $P - V$ – множество $\Gamma < S_P$ группа подстановок. Зададим $\gamma \in \Gamma$ подстановку у которой $\lambda_i(\gamma)$ – количество траекторий длины $i \in N$. Итак γ – есть подстановка, состоящая из $[1 \lambda_1(\gamma) + 2 \lambda_2(\gamma) + \dots + v \lambda_v(\gamma)]$.

Под указателем цикличности Γ будем понимать полином

$$Z(\gamma) = Z(\Gamma : r_1, r_2 \dots r_V) := \left| \Gamma^{-1} \right| \sum_{\gamma \in \Gamma} \prod_i Z_i^{\lambda_i(\gamma)} \text{ из } Z(z_1, z_2 \dots z_V).$$

Обозначим через $h(\Gamma, \lambda_1, \lambda_2 \dots \lambda_V)$ количество подстановок $\gamma \in \Gamma$ типа цикличности $(1 \lambda_1 + 2 \lambda_2 + \dots + v \lambda_V)$, то получим:

Теорема. Справедливо следующее выражение

$$\begin{aligned}
 Z(\Gamma) &= Z(\Gamma : z_1, z_2 \dots z_v) = \\
 &= |\Gamma|^{-1} \sum_{1\lambda_1 + 2\lambda_2 + \dots + v\lambda_v = V} h(\Gamma : \lambda_1, \lambda_2 \dots \lambda_v) \cdot \prod_{i=1}^V Z_i^{\lambda_i} = \\
 &= |\Gamma|^{-1} \cdot \sum_{\lambda_1, \lambda_2 \dots \lambda_v = 0}^{\infty} h(\Gamma : \lambda_1, \lambda_2 \dots \lambda_v) \cdot \prod_{i=1}^V Z_i^{\lambda_i}
 \end{aligned}$$

Доказательство. Утверждение очевидно, если отметить что $h(\Gamma, \lambda_1, \lambda_2 \dots \lambda_v)$ в любом случае (каждый раз) равно нулю всегда, когда $1\lambda_1 + 2\lambda_2 + \dots + v\lambda_v \neq N_0$.

Показатели цикличности групп подстановок.

Теорема. Для $v \in N$ в S_v имеется $s(v,1) = (v-1)!$.

Если подстановка состоит из одного цикла длины v , то это значит

$$h(s_v : 0 \ 0 \ 0 \ \dots \ 0, 1) = (v-1)!$$

Согласно вышеприведенным формулам можно получить:

$$v! \cdot \left(\prod_{i=1}^V (i!)^{\lambda_i} \lambda_i! \right)^{-1}.$$

Разбиение V -множества P типа $(1\lambda_1 + 2\lambda_2 + \dots + v\lambda_v)$. Симметричная группа каждого λ_i эквивалентного класса с I элементами содержит подстановки, одинаковые по структуре и их число равно $(i-1)!$. Вследствие чего

$$\frac{(i-1)!^{\lambda_i}}{i!^{\lambda_i}} = \frac{1}{i^{\lambda_i}} \text{ для } i \in Z_V.$$

В итоге получается **теорема**. Если $1\lambda_1 + 2\lambda_2 + \dots + v\lambda_v = V_c$, $\lambda_1, \dots, \lambda_v \in N_0$, то

$$h(S_V; \lambda_1, \lambda_2 \dots \lambda_v) = v! \cdot \left(\prod_{i=1}^V \lambda_i! \cdot i^{\lambda_i - 1} \right). \quad (***)$$

Теорема. Для $\Gamma < S_p$ и $c(p) = v$ действительно

$h(\Gamma : V, 0, 0, \dots, 0) = 1$ где $Z\{id_p\} = Z_1^V$ - показатель цикличности тривиальной группы постановок $\{id_p\}$.

С помощью (***) следует.

Теорема. (5.15). Если $p - V$ - множество, то

$$Z(S_p) = Z(S_p; z_1, z_2, \dots, z_v) =$$

$$\sum_{1\lambda_1 + 2\lambda_2 + \dots + v\lambda_v = V} \prod_{i=1}^V (\lambda_i! \cdot i^{\lambda_i})^{-1} Z_i^{\lambda_i} \quad (****)$$

показатель цикличности симметрической группы.

Циклы подстановок. Так как каждая v_i подстановка из S_p определенного типа, следует

$$\sum_{\lambda_1, \lambda_2, \dots, \lambda_V=0} h(S_p; \lambda_1, \lambda_2, \dots, \lambda_V) = V!$$

Положим $Z_i=1$ для всех $i \in Z_V$ и из теоремы (***) получим:
 $Z(S_p; 1, 1, \dots, 1) = 1$.

Теперь рассчитаем показатель цикличности для разных групп подстановок.

Теорема. Если $P - V$ – множество с $v=m+c$ и $m \in 2Z$ как $e \in \{0, 1\}$ то

$$\begin{aligned} Z(A_p) &= Z(A_p; z_1, z_2, \dots, z_V) = \\ &= \sum_{\lambda_1+2\lambda_2+\dots+v\lambda_v=0} (1 + (-1)^{\lambda_2+\lambda_4+\dots+\lambda_M}) \prod_{i=1}^V (\lambda_i! i^{\lambda_i})^{-1} \cdot Z_i^{\lambda_i} : \end{aligned}$$

цикличности знакопеременной группы a_p .

Доказательство. Группа a_p состоит из (четной) прямой подстановки $\frac{V!}{2}$ от S_p . Подстановка $\gamma \in S_p$ циклического типа имеет вид

$(1\lambda_1+2\lambda_2+\dots+v\lambda_V)$ тогда нечетная, если количество $\lambda_2+\lambda_4+\dots+\lambda_m$ ее циклов четной длины – нечетное, если это выполняется, то:

$$1 + (-1)^{\lambda_2+\lambda_4+\dots+\lambda_M} = 0.$$

Из теоремы (5.15) следует это утверждение.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Einführung in die Kombinatorik mit einem Anhang ü formale Potenzen. Heinz-Richard Halder und Werner Heise. Institut für Mathematik der Technischen Universität München. V. XII. 1977. с. 7, 61-71s.

В.В. СОЛДАТОВ, О.В. СОЛДАТОВА

Рязанский государственный радиотехнический университет

**ИСПОЛЬЗОВАНИЕ АНИМАЦИОННЫХ МОДЕЛЕЙ И
ПРОГНОЗИРОВАНИЕ ПРИРОДНЫХ
И ТЕХНОГЕННО - ОПАСНЫХ СИТУАЦИЙ
НА ОСНОВЕ КОСМИЧЕСКИХ СНИМКОВ**

В статье рассмотрены общие принципы построения 3D объектов, а также точности координатной привязки элементов земной на космических изображениях.

Космические услуги были доступны давно, но оставались невос требованными. Пожары 2010 года в ЦФО послужили основой для создания аналитического и картографического 3D моделирования на основе космических снимков (рис. 1).

Космические услуги были доступны давно, но оставались невос требованными. Пожары 2010 года в ЦФО послужили основой для создания аналитического и картографического 3D моделирования на основе космических снимков. Благодаря полученным моделям появляется возможность прогнозировать и контролировать процессы подтопления, переувлажнения почв, накопления тяжелых металлов, определять оптимальные сроки для сева сельхозкультур, делать прогнозы по урожайности, проводить экологический мониторинг окружающей среды и особо опасных объектов. Использование данных моделей позволяет обеспечивать пожаробезопасность региона, круглосуточный мониторинг ЛЭП, трубопроводов. Аппаратура низкого разрешения спутников дистанционного зондирования земли не способна обеспечить детальное воспроизведение потенциально опасных объектов. Обработанная информация при помощи Геоинформационных технологий (ГИС – технологий) может с успехом использоваться для наведения на исследуемые объекты искусственные спутники земли (ИСЗ), имеющих аппаратуру более высокого разрешения и передавать в оперативном режиме космические снимки с географической привязкой координат [1]. Реализовать данный процесс возможно при помощи отечественного ИСЗ «Ресурс-ДК», но и этих данных бывает недостаточно для предсказания дальнейшего развития процесса. В связи с этим возникла необходимость создавать 3D – модели, привязанные к электронной карте.

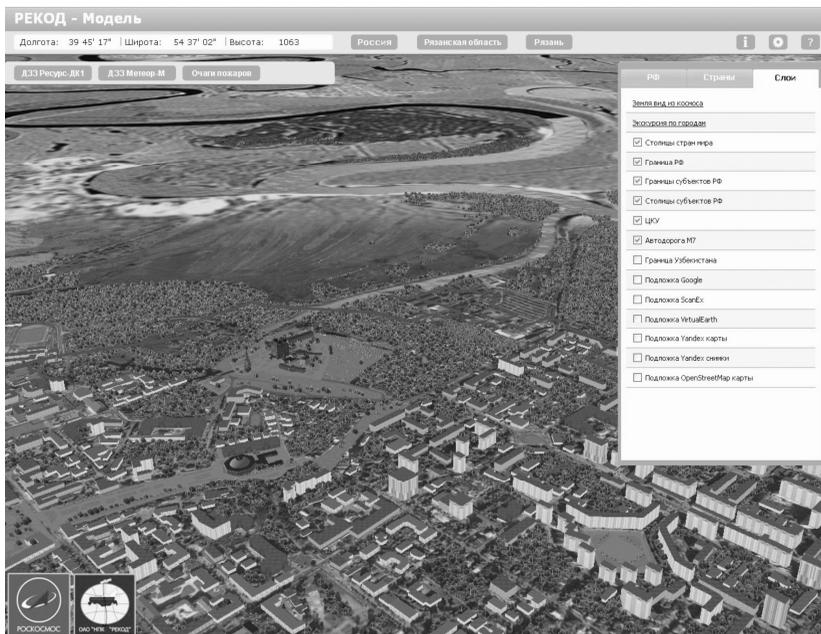


Рис. 1. Фрагмент 3D модели г. Рязани

Актуальность и потенциальные заказчики проекта. Мониторинг окружающей среды актуален для МЧС, министерства экологии и природных ресурсов, лесного хозяйства, министерства культуры, министерство транспорта и дорожного хозяйства, министерства сельского хозяйства, владельцев трубопроводов и линий электропередач и т.д.

Новизна проекта. Для достижения этой цели было выбрано одно из перспективных направлений – использование результатов Дистанционного Зондирования Земли (ДЗЗ), компьютерная обработка космических снимков высокого разрешения и 3D моделирование наземных объектов и происходящих процессов. Существующие в настоящее время способы экологического мониторинга только фиксируют происходящие наземные процессы, а также существуют попытки вероятностного предсказания событий. За день до землетрясения в Новой Зеландии были зафиксированы признаки подготовки сдвигов тектонических плит (рис. 2) [2]. Существование 3D модели помогло бы смоделировать аварии на важных объектах и предупредить развитие аварийных ситуаций на дорогах, мостах, трубопроводах, разрушение промышленных и административных зданий.

С помощью 3D анимации, графических средств анимации появилась возможность визуально воспроизводить и моделировать аварийные ситуации, пожары, наводнения, землетрясения и другие стихийные бедствия. Объемное моделирование затопляемых населенных пунктов и муниципальных образований позволяет предсказывать с определенной степенью вероятности затопление жилых, промышленных и опасных объектов. Особый контроль может быть обеспечен при помощи 3D моделей за стратегически важными объектами. Построенные модели позволят детально воспроизводить последствия пожаров, наводнений, а также способны фиксировать и прогнозировать реальный нанесенный ущерб.

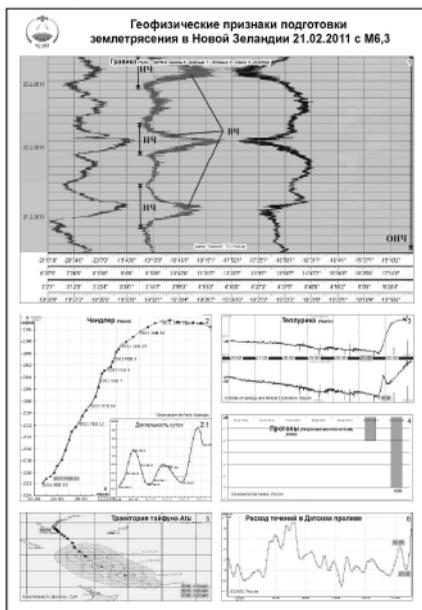


Рис. 2. Признаки подготовки землетрясения в Новой Зеландии 21.02.2011

В процессе реализации проекта необходимо решить следующие задачи:

1. Получение оперативных и заказ архивных космических снимков в различном разрешении и спектральном диапазоне.

Космические снимки высокого, среднего и низкого разрешения, в том числе информации научной и экспериментальной аппаратуры для муниципальных образований предоставляет Научный центр оперативного мониторинга Земли (НЦ ОМЗ). Пользователи космической

информации ДЗЗ имеют возможность наряду с данными высокого разрешения с ИСЗ «Ресурс-ДК» приобретать данные среднего разрешения с ИСЗ «Метеор-М». На борту спутника «Метеор-М» установлен комплекс многофункциональной спектральной съемки (КМСС), который имеет следующие технические характеристики:

- количество спектральных каналов – 3;
- номинальные спектральные интервалы: **1** - 0,535-0,575нм;
2 - 0,630-0,680нм; **3** - 0,760-0,900нм;
- полоса захвата при двух одновременно работающих камерах – 900 км;
- разрешение — около 60 м, 100 м.

Для мониторинга пожарной обстановки наиболее важен [3] инфракрасный интервал. Данные инфракрасной съемки накладываются на цифровую карту видимого диапазона и выявляются нарушение теплового режима критически опасных объектов, находятся очаги лесных пожаров, нагрев изоляторов ЛЭП.

Периодичность съемки – 4-5 раз в неделю. Для экологического мониторинга это является вполне приемлемо. Кроме того, имеется возможность воспользоваться данными из архива снимками с разрешением 1м. Используя Интернет телефонию, радиосвязь, рекламу, возможно, получать информацию (возникновение несанкционированных свалок, нарушения экологии и т.д.) от населения и других заинтересованных лиц.

2. Визуальная обработка космических снимков и нахождение требуемой информации.

Полученные снимки с помощью компьютера детально обрабатываются и находятся точные координаты районов пожаров, наводнений, неисправности изоляторов ЛЭП, промышленные свалки отходов, аварии трубопроводов и т.д.

3. Используя полученные координаты и навигационную аппаратуру на местности фиксируем аварийную обстановку с помощью фото и видео аппаратуры.

4. Обработанные материалы наземной съемки привязываются к найденным координатам с помощью свободно распространяемой программной оболочки ОАО "НПК "РЕКОД" [3].

5. На основе аналитических материалов, предоставляемых НЦ ОМЗ, известных алгоритмов цифровой обработки изображений [4,5], проектируем анимационную 3D модель развития дальнейших событий.

Полученную 3D модель и анимационный прогноз представляем потенциальным заказчикам.

Исходя из опыта прошлых летних пожаров, 3D моделированием объектов заинтересовалось министерство экологии и промышленных ресурсов Рязанской области, МЧС, Министерство промышленности и наукоемких технологий. В августе 2011 года использовались данные МЧС о пожарной обстановке Рязанской области. Координаты возникающих пожаров наносились на оперативную космическую карту ЦФО и были представлены на доклад председателю правительства Путину В.В. в администрации Рязанской области.

Основными преимуществами проекта, являются:

- востребованность в различных областях;
- наглядное графическое представление требуемых данных об экологической и другой техногенной обстановке;
- возможность прогнозирования дальнейшего развития критических событий с элементами анимации и 3D графики.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Еремеев В.В., Козлов Е.П. Повышение точности и надежности координатной привязки снимков от геостационарных космических систем наблюдения земли по электронным картам // Вестник РГРТУ. – Вып.28. 2009. – С. 6 – 12.
2. <http://www.ntsomz.ru/zakaz/data>
3. www.npkrekod.ru
4. Р.Гонсалес, Р. Вудс. Цифровая обработка изображений. Москва: техносфера, 2006г.
5. Цифровая обработка сигналов и ее применения: Материалы докладов 12 Международной конференции.– Москва, 2009.– Т. VI.– С. 256-259.

И.А. ТЕЛКОВ

Рязанский государственный радиотехнический университет

ОРГАНИЗАЦИЯ УНИВЕРСАЛЬНЫХ ПАРАЛЛЕЛЬНЫХ
ВЫЧИСЛЕНИЙ С ИСПОЛЬЗОВАНИЕМ
ВОЗМОЖНОСТЕЙ NVIDIA CUDA

Приводится описание организации параллельных вычислений универсального типа (не только геометрических) на базе геометрических процессоров (*GPU – Graphic Processor Unit*) фирмы *NVIDIA* с использованием технологии *CUDA (Compute Unified Device Architecture)*. Анализ предназначен для построения математической модели параллельных вычислений по технологии *CUDA* и является продолжением работы [1].

По состоянию на 2011 год, программная модель *CUDA* преподается почти в 300 университетах по всему миру.

В отличие от *CPU (Central Processor Unit)*, ориентированных на решение последовательных задач, *GPU* изначально предназначен для параллельных вычислений.

Производители *CPU* стремятся повысить мощи своих вычислений за счет использования архитектуры *SIMD* (одна команда обрабатывает вектор данных) [1,2], а в *GPU* применяется ее разновидность - *SIMT (Single Instruction Multiple Threads* - одна инструкция обрабатывает несколько потоков). *SIMT*, в отличие от классической *SIMD*, не требует преобразования данных в векторы и допускает произвольные ветвления в потоках.

GPU предназначены для параллельных вычислений с большим количеством математических операций. В *CPU* значительная доля ресурсов направлена на управления данными. На рис.1 показано, сколько места в *CPU* и *GPU* занимает разнообразная логика.

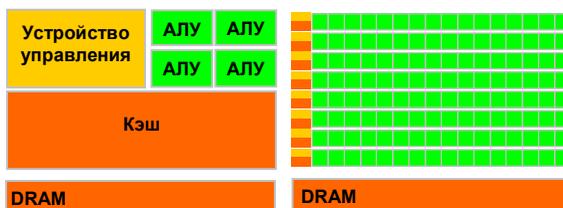


Рис. 1. Разница в архитектуре CPU и GPU [3]

1. Состав NVIDIA CUDA

Состав *NVIDIA CUDA* приведен на рис.2. *CUDA* включает два *API (Application Programm Interface)*: высокого уровня (*CUDA Runtime API – функции времени исполнения*) и низкого (*CUDA Driver API*).

Одновременное использование обоих уровней в рамках одной программы невозможно. В любом случае предполагается наличие знаний об устройстве графического ускорителя *NVIDIA*.

Существует более высокий уровень работы с функциями *CUDA API* - библиотеки математических функций. На данный момент предоставляется две библиотеки – *CUBLAS* и *CUFFT*. Первая из них – *CUBLAS (CUDA Basic Linear Subprograms)* предназначена для решения задач линейной алгебры, а вторая – *CUFFT (CUDA Fast Fourier)* – для обработки сигналов на базе быстрого преобразования Фурье.

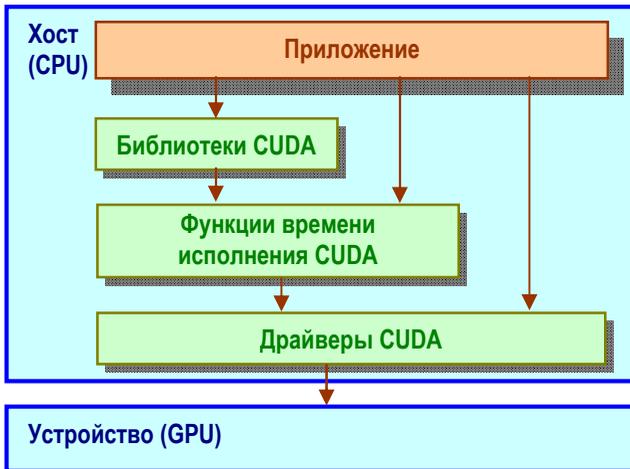


Рис. 2. Состав CUDA API

2. Терминология CUDA

Вычислительная архитектура CUDA основана на концепции *SIMT* и понятии мультипроцессора. Под мультипроцессором понимается многоядерный *SIMT*-процессор, позволяющий на каждом шаге выполнять на всех ядрах только одну команду. Каждое ядро (*kernel*) мультипроцессора при этом скалярное.

Под **устройством (device)** в *CUDA* понимается графический ускоритель *NVIDIA* (рис.2,3), имеющий поддержку драйвера *CUDA*, а под **хостом (host)** – программные средства, расположенные в основной оперативной памяти компьютера и использующие *CPU* для управления устройством (*device*).

Последовательность данных, направляемых на обработку, называется **потокком (thread)**. Потокки объединяются в группы по 32 потока – **варпы (warp)**, которые и являются минимальными единицами обра-

ботки данных в CUDA. Под **блоком** (*block*) понимается множество потоков (от 64 до 512) или варпов (от 2 до 16).

Для повышения производительности работы (за счет распараллеливания вычислений) при большом количестве ядер *GPU* (рис.4) блоки объединяются в **сетки** (*grid*). В противном случае все вычисления будут выполняться последовательно.

На рис. 4, для обозначения потоков используются двухмерные индексы. *CUDA* предоставляет возможность работать также с трехмерными, двухмерными или одномерными индексами. В общем случае индексы представляют собой трехмерные векторы. Для каждого потока будут известны: индекс потока внутри блока и индекс блока внутри сетки `blockIdx`. При запуске все потоки будут отличаться только этими индексами.

3. Модель памяти CUDA

Иерархическая организация графических систем *NVIDIA* подразумевает аналогичную иерархию при построении системы памяти. Можно выделить пять видов (уровней) памяти.

1. *Глобальная память* – память, доступная для всех мультипроцессоров системы. Ее размер составляет от 256 Мбайт до 1.5 Гбайт (и до 4 Гбайт на *Tesla*). Она обладает высокой пропускной способностью, но не кэшируется.

2. *Локальная память* – это небольшой объём памяти, к которому имеет доступ только один потоковый процессор. Она относительно медленная — такая же, как и глобальная.

3. *Разделяемая память* – это 16-Кбайтный блок памяти с общим доступом для всех потоковых процессоров в мультипроцессоре. Эта память имеет быстродействие близкое к регистровой. Используется в виде кэша первого уровня.

4. *Память констант* – область памяти объемом 64 Кбайта, доступная только для чтения всеми мультипроцессорами. Она кэшируется по 8 килобайт на каждый мультипроцессор, но достаточно медленная. Константная память очень удобна в использовании. Можно размещать в ней данные любого типа и читать их при помощи простого присваивания.

5. *Текстурная память* – блок памяти, доступный для чтения всеми мультипроцессорами. Текстурная память оптимизирована под выборку двумерных данных (графических текстур) и изначально предназначена для поддержки процесса текстурирования трехмерных объектов при прямом использовании *GPU* (для решения задач ускорения графических и геометрических построений). Выборка данных осуществляется при помощи текстурных блоков *GPU*, поэтому предоставляются воз-

возможности линейной интерполяции данных без дополнительных затрат. Кэшируется по 8 килобайт на каждый мультипроцессор. Медленная, как и глобальная память.

Физически все перечисленные виды памяти представляют собой единую память, известную как видеопамять графического ускорителя. Они отличаются друг от друга алгоритмами кэширования и моделями доступа. CPU может обновлять и запрашивать только внешнюю память: глобальную, константную и текстурную.



Рис. 3. Организация устройства

4. Детальная структура GPU

При более детальном рассмотрении (рис.5) архитектура NVIDIA GPU состоит из шейдерных ядер. Шейдерное ядро состоит из восьми *TPC* (*Texture Processor Cluster*) – кластеров текстурных процессоров. Кластеры, в свою очередь, состоят из двух потоковых мультипроцессоров (*SM – Streaming Multiprocessor*). Потоковый мультипроцессор включает в себя два конвейера: *front end* (решает задачи чтения и декодирова-

ния инструкций) и *back end* (конечный вывод инструкций). Кроме того, в него входят восемь шейдерных процессоров *SP* (*shader processor*) и два суперфункциональных блока *SFU*. За каждый шаг работы (такт) *front end* выбирает варп и обрабатывает его. Мультипроцессоры *SM* могут обращаться к *GDDR*.

Для работы с *CUDA* требуется *GPU GeForce* не ниже серии 8. Технология *CUDA* поддерживает операционные системы *Windows (XP и Vista)*, *Linux* и *Mac OS X*. В качестве инструментальных средств можно использовать программные средства *Microsoft Visual Studio* и *Borland C++ Builder* или любая среда, работающая с языком *C*.

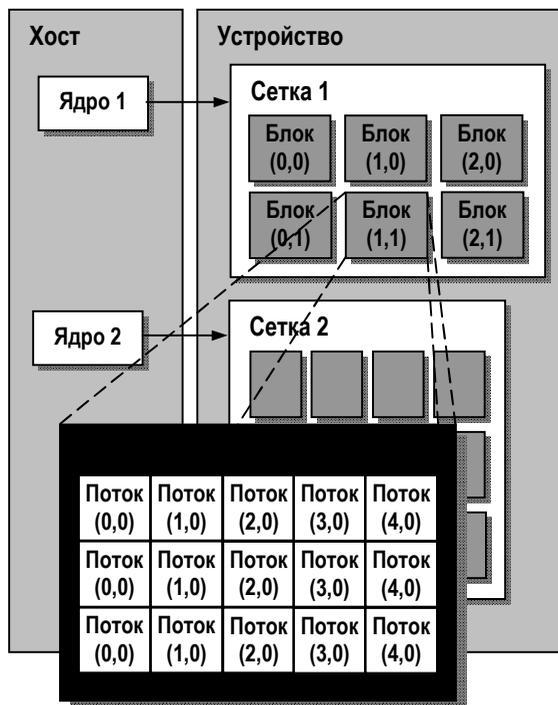


Рис. 4. Организация потоков в *CUDA*

Последние модели *NVIDIA GPU* содержат до 480 (*GeForce GTX 480* и *GeForce GTX 570*) и 512 (*GeForce GTX 580*) шейдерных процессоров (ядер), выполненных по 40 нм технологии. Отдельный вариант *CUDA* реализован для графических станций семейства *NVIDIA Tesla*, содержащих до 960-1792 шейдерных процессоров.

В настоящее время в качестве языков программирования NVIDIA CUDA поддерживается программирование на C/C++, *Apple OpenCL*, *DirectX Compute*, *FORTRAN*, *Java* и *Python* [3].

Приведенный анализ *NVIDIA CUDA* предназначен для построения математической (тензорной модели) архитектуры *CUDA* и формального описания организации ее функционирования в задачах оптимизации параллельных вычислений.

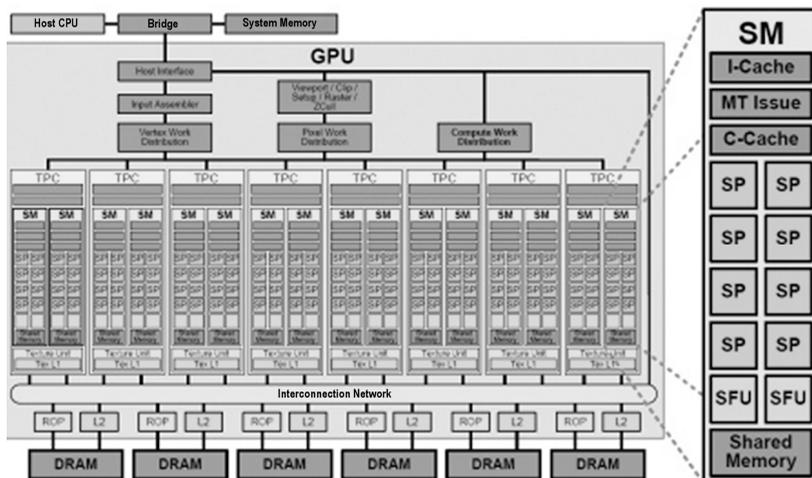


Рис. 5. Детальная структура CUDA GPU [4,5]

СПИСОК ИСПОЛЪЗУЕМЫХ ИСТОЧНИКОВ

1. И.А.Телков. Математическая модель многоядерных процессоров в пространстве состояний. // Информационные технологии в научных исследованиях: межвуз. Сб. науч. Трудов. Рязань, 2010. С.135-140.
2. Майерс Г. Архитектура современных ЭВМ: В 2-х т. М.: Мир, 1985. Кн.1. 364 с; Кн.2. 312 с.
3. NVIDIA CUDA™ C Programming Guide. Version 3.2. – NVIDIA Corporation, 2010. 183 p.
4. NVIDIA CUDA™ (официальный сайт фирмы) http://www.nvidia.com/object/cuda_home.html
5. CUDA Zone (официальный сайт CUDA русифицированный вариант) http://www.nvidia.ru/object/cuda_home_new_ru.html

Р.В. ФОКИН
Академия ФСИН России

ОСОБЕННОСТИ ОРГАНИЗАЦИИ ПРАКТИЧЕСКИХ ЗАНЯТИЙ ПО ИНФОРМАТИКЕ И ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ ПРИ ОБУЧЕНИИ СЛУШАТЕЛЕЙ ОБРАЗОВАТЕЛЬНЫХ УЧРЕЖДЕНИЙ ФСИН РОССИИ

Рассмотрены перспективы использования технологии виртуальных машин при преподавании информатики и информационных технологий в высших учебных заведениях ФСИН России. Показано современное состояние уровня развития технологии виртуальных машин.

Эффективное функционирование уголовно-исполнительной системы невозможно без высококвалифицированных кадров. В условиях реализации Концепции развития уголовно-исполнительной системы Российской Федерации до 2020 года существенно повышаются требования к подготовке сотрудников ФСИН, в том числе и в области информационных технологий.

Обучение сотрудников УИС информационным технологиям немислимо без качественного компьютерного практикума. Однако его трудно организовать при использовании ограниченных пользовательских профилей (они необходимы для обеспечения безопасности компьютерных систем и сетей), т.к. в данном случае слушатели не имеют возможности доступа ко многим ресурсам компьютерной системы. В таких условиях невозможно на должном уровне осуществлять практическую подготовку или, другими словами, формировать умения и навыки по таким разделам дисциплин как операционные системы, компьютерные сети и многим другим.

Достаточно эффективным решением данной проблемы является использование в образовательном процессе технологий виртуальных машин. Достоинства такого подхода неоспоримы – каждый слушатель получает полный доступ к виртуальной компьютерной системе без риска нанести ущерб реальному оборудованию.

Виртуальные машины эмулируют работу реального компьютера. На виртуальном компьютере можно выполнять настройку базовой системы ввода/вывода (BIOS), эмулировать подключение периферийных устройств, производить установку и настройку операционных систем, устанавливать программное обеспечение и осуществлять работу с ним.

Инструмент виртуальной машины представляет собой комплекс программного обеспечения, устанавливаемый на операционную систему реального компьютера. Операционная система, под управлением

которой, работает система виртуальных машин, называется хостовой. Управление виртуальными машинами в системе виртуальных машин осуществляет специальный монитор, играющий роль посредника между виртуальными машинами и реальным оборудованием компьютерной системы.

Пользователь не имеет доступа к монитору виртуальных машин, а взаимодействие с виртуальными машинами осуществляет с помощью консоли, имеющей графический интерфейс.

При работе с виртуальной машиной пользователь конфигурирует ее аппаратное обеспечение и устанавливает на нее операционную систему, которая называется гостевой. Современные системы виртуальных машин поддерживают достаточно большое количество гостевых операционных систем (обычно семейства Windows, Linux, Unix, MacOS).

Алгоритм создания виртуальной машины на компьютере:

- запуск системы виртуальных машин;
- выбор типа операционной системы;
- выделение объема оперативной памяти;
- резервирование размера жесткого диска;
- выбор периферийных устройств;
- запуск созданной виртуальной машины;
- установка операционной системы.

При создании виртуальной машины следует учитывать, что она будет «потреблять» аппаратные ресурсы реальной компьютерной системы. Поэтому оборудование компьютерных классов должно обеспечивать возможность одновременного запуска двух операционных систем: гостевой и хостовой. Другими словами, требования, предъявляемые к аппаратным ресурсам компьютера (микропроцессору, оперативной памяти), операционной системой увеличиваются минимум в два раза.

Особые требования предъявляются к размерам жесткого диска, особенно в условиях многопользовательского режима работы компьютера. Каждый пользователь будет создавать свою виртуальную машину, для которой необходимо место.

С другой стороны использование технологий виртуальных машин в образовательном процессе открывает широкие возможности повышения качества и наглядности практических занятий при значительной экономии аппаратных ресурсов и затрат на обслуживание компьютерных классов (экономия достигается за счет того, что на одном компьютере реализуется несколько виртуальных компьютерных

систем). При этом становится возможным достижение ряда учебно-методических и дидактических задач, например:

- слушатель имеет возможность конфигурировать BIOS, устанавливать и настраивать операционные системы и программное обеспечение, выполнять дисковые операции без возможности нарушения целостности компьютерной системы;

- имеется возможность администрирования виртуальной компьютерной сети без риска нанесения ущерба реальной компьютерной сети образовательного учреждения;

- имеется возможность переноса созданной виртуальной машины на другие компьютеры, что позволяет организовать самостоятельную работу слушателя.

Технология виртуальных машин может быть использована при изучении следующих разделов:

- операционные системы;
- программное обеспечение компьютера;
- компьютерные сети и др.

Таким образом, технология виртуальных машин является перспективным средством подготовки сотрудников уголовно-исполнительной системы в области информатики и информационных технологий.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Концепция развития уголовно-исполнительной системы Российской Федерации до 2020 года. Утверждена распоряжением правительства Российской Федерации от 14 октября 2010 г. №1772-р

2. Рыжова Н.И., Королева Н.Ю., Ляш О.И. Изучение основ работы с виртуальными машинами будущими учителями информатики. // Информатика и образование, №8 2008 с.

3. Виртуальная машина. Материал из Википедии – свободной энциклопедии. [Электронный ресурс]. URL: <http://ru.wikipedia.org>

4. Гуляев А. Виртуальные машины – несколько компьютеров в одном. – СПб.: Питер, 2006. – 224 с.

Р.В. ХРУНИЧЕВ

Рязанский государственный радиотехнический университет

**РЕФЕРИРОВАНИЕ ТЕКСТА С ПРИМЕНЕНИЕМ
СТАТИСТИЧЕСКИХ МЕТОДОВ ОБРАБОТКИ ТЕКСТОВОЙ
ИНФОРМАЦИИ НА ОСНОВЕ ЧАСТОТЫ ВСТРЕЧАЕМОСТИ
СЛОВ В ДОКУМЕНТЕ И ОПРЕДЕЛЕНИЯ ИХ ВЕСА**

Рассматривается применение статистических методов анализа информации для задачи автоматизированного реферирования текстов на основе законов Ципфа определения частоты и веса значимого слова/словосочетания.

Сегодня машинная обработка текстов на естественном языке приобретает большое значение и становится всё более распространенной. Это связано с тем, что объем данных в сети Интернет, корпоративных хранилищах данных, базах данных и др. местах их локализации растет быстрыми темпами, и ручная их обработка сегодня уже не представляется возможной. Становится необходимым адаптировать уже созданные или разрабатывать новые методы обработки информации. На данный момент разработано уже достаточно много методов анализа неструктурированных данных. Эти методы основаны на разных подходах и используют различные способы реализации. Но применяя те, или иные методы необходимо исходить из того, насколько полно и точно они отражают релевантность запросу, и, конечно, учитывать сложность реализации данных методов, а также затраты программно-технических ресурсов.

Исходя из перечисленных выше требований, для поставленной задачи автоматизированного реферирования текста предлагается использовать статистический метод на основе частоты встречаемости слов/словосочетаний и определения их веса. Данный метод основан на законах Ципфа. Он показал, что независимо от того, кто писал текст, слова с большим количеством букв встречаются в тексте реже коротких слов. На основании этого он вывел закон «количество – частота»: частота и количество слов, входящих в текст с этой частотой связаны

между собой. Частота слова f определяется как: $f = \frac{n}{N}$, где n – количество одинаковых слов в документе, а N – общее количество слов в документе. Тогда вероятность P обнаружения слова будет определяться как: $P = \frac{f}{N}$ [1].

Некоторые слова могут иметь одинаковую частоту, то есть входить в текст равное количество раз. Если сгруппировать их, взяв только одно значение из каждой группы и расположить частоты по мере убывания, пронумеровав их, то порядковый номер частоты будет называться рангом частоты. Так, наиболее часто встречающиеся слова будут иметь ранг 1, следующие за ними – 2 и т.д. Ципф показал также, что произведение C вероятности обнаружения слова в тексте на ранг частоты R_f есть величина приблизительно постоянная: $C = \frac{P * R_f}{N}$.

Если построить график, отложив по одной оси (оси X) частоту вхождения слова, а по другой (оси Y) -- количество слов в данной частоте, то получившаяся кривая будет сохранять свои параметры для всех без исключения созданных текстов (рис. 1).

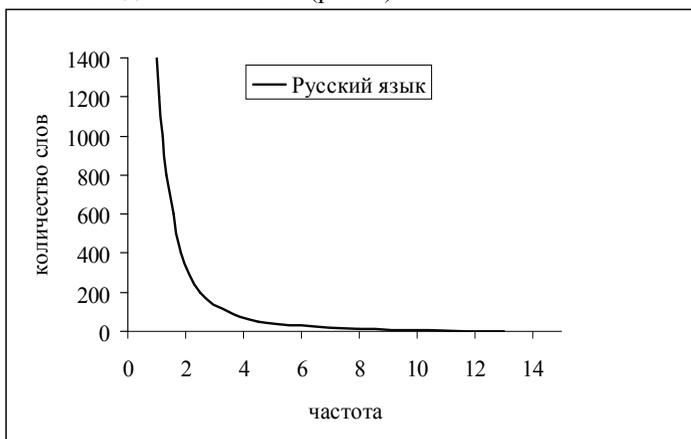


Рис. 1. Зависимость частоты вхождения слова от количества слов в данной частоте

Теперь необходимо адаптировать законы Ципфа для задачи автоматизированного реферирования текстов, т.е. с их помощью извлечь слова, отражающие смысл текста. Построим график зависимости ранга от частоты (рис. 2).

Применение на практике законов Ципфа в программах статистического анализа текста показывают, что наиболее значимые слова лежат в средней области графика. Слова, которые попадают слишком часто, в основном оказываются предлогами, местоимениями и т.п. Редко встречающиеся слова тоже не имеют решающего смыслового значения, за исключением тех случаев, когда текст является специализированным и содержит редкие термины и определения.

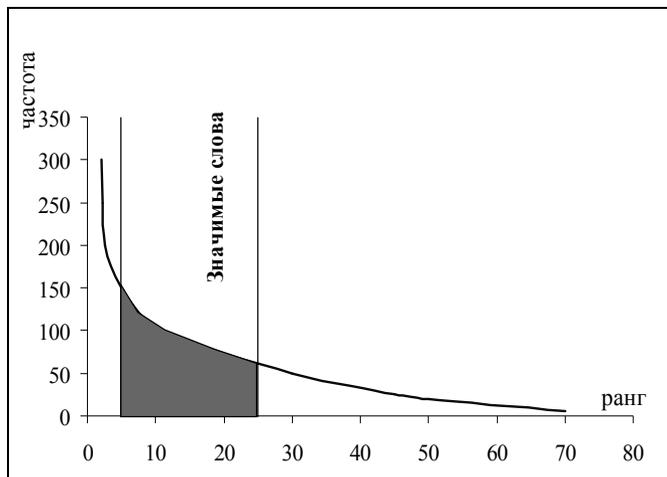


Рис. 2. Зависимость ранга частоты от частоты встречаемости слова в тексте

Применительно к задаче автоматизированного реферирования необходимо чётко определиться с границами области диапазона значимых слов. Если поставить границы широко, то нужные термины потеряются в большом количестве вспомогательных слов; если же диапазон узок – потеряются смысловые термины. Подбирая данный параметр можно, например, руководствоваться общим объемом текста или сформировать так называемый «стоп-лист», куда будут включены слова, которые не имеют смысловой нагрузки [3].

Определимся теперь с весовыми коэффициентами. Выше мы рассматривали только отдельно взятый документ, не обращая внимания на то, что он входит в хранилище данных (корпоративную базу данных) наряду с множеством других документов. Если представить всё хранилище данных как единый документ, что имеет смысл в связи с тем, что в нём хранятся документы единой тематической направленности, то к нему можно будет применить те же законы, что и к единичному документу. Проанализировав документы хранилища данных на предмет вхождения в них одних и тех же терминов, мы, естественно, обнаружим, что многие из них встречаются практически во всех документах, при этом все попутные термины не несущие смысловой нагрузки можно, например, занести в «стоп-лист». Чтобы избавиться от лишних слов и в тоже время поднять рейтинг значимых слов, вводят инверсную частоту термина f_u . Значение этого параметра тем

меньше, чем чаще слово встречается в документах базы данных. Данный параметр вычисляют по формуле:

$$f_u(k) = \log\left(\frac{N_\partial}{n_k}\right),$$

где $f_u(k)$ – инверсная частота термина k , N_∂ – количество документов в базе данных, n_k – количество документов с термином k . Теперь каждому термину можно присвоить весовой коэффициент, отражающий его значимость:

$$W_{kj} = f_{kj} * f_u(k),$$

где W_{kj} – вес термина k в документе j и f_{kj} – частота термина k в документе j [4].

В качестве терминов могут выступать не только отдельные слова, но и словосочетания.

Как видно данный метод статистического анализа позволяет выделить из всего множества документов хранилища данных только документы релевантные запросу и расположить их в порядке убывания частоты или веса значимого слова/словосочетания. После такого анализа становится довольно просто вывести на экран именно те документы, которые содержат нужную информацию.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. <http://webpavilion.ru/статьи/закон-ципфа>
2. А.Б. Кутузов. Корпусная лингвистика: курс лекций. Лекция 2 «Основные задачи и направления корпусной лингвистики». - Тюмень: ТюмГУ, 2008 г. - 5 с.
3. Ю.А. Семенов. Современные поисковые системы. - ГНЦ ИТЭФ - 2010 г. - 17 с.
4. Трегубов А.А., Пескова О.Ю. Статистические характеристики текстовых документов Технологии информационного общества — Интернет и современное общество: VII Всероссийская объединенная конференция.. — СПб.: СПбГУ, 2004. С. 73–76.

И.А. ЦВЕТКОВ

Рязанский государственный радиотехнический университет

ОБРАЩАЮЩИЙ САМОМОДИФИЦИРУЕМЫЙ АЛГОРИФМ
В АЛФАВИТЕ С ОДНОЙ ДОПОЛНИТЕЛЬНОЙ БУКВОЙ

Предлагается обращающий алгоритм с одноразовыми пополнениями слева, для которого доказывается, что его результат — это обращение любого исходного слова во входном алфавите. Сравнивается сложность предлагаемого алгоритма и двух известных обращающих алгоритмов.

Введение. Предложенный в статье [3] самомодифицируемый алгоритм отличается от нормального алгоритма Маркова [1] тем, что при переработке слова схема нормального алгоритма Маркова не изменяется, а схема самомодифицируемого алгоритма *может изменяться* посредством добавления формул перед заданными формулами схемы или в конце ее и удаления формул, выполнившихся заданное число раз.

Частный случай самомодифицируемого алгоритма — это алгоритм с начальной схемой, содержащей формулы: 1) простые вида $A \rightarrow B$; 2) заключительные вида $A \rightarrow \bullet B$; 3) пополняющие вида $A \rightarrow B \uparrow C \rightarrow \otimes D$ с одноразовым пополнением $C \rightarrow \otimes D$. Здесь A, B, C, D — слова в заданном алфавите X_1 , не содержащем символов из множества $\{\rightarrow, \uparrow, \bullet, \otimes\}$.

В обозначениях из [3] формулы из пп. 1–3 записываются так: 1) $A \infty B$; 2) $A \infty \bullet B$; 3) $A \infty B 10 \infty C 10 D$. Полагаем, что алфавит X_1 не содержит символов из множества $\{0, 1, \infty, \bullet\}$.

Простые и заключительные формулы применяются так же, как для нормального алгоритма Маркова [1]. Применение пополняющей формулы вида $A \rightarrow B \uparrow C \rightarrow \otimes D$ с одноразовым пополнением $C \rightarrow \otimes D$ состоит в выполнении указанной слева от символа \uparrow формулы $A \rightarrow B$ как простой формулы с последующим *добавлением* в начало схемы алгоритма (слева при записи схемы как слова) одноразовой формулы $C \rightarrow \otimes D$. Ее выполнение возможно со следующего шага алгоритма. Одноразовая формула выполняется как простая формула и затем сразу *удаляется* из схемы алгоритма.

Самомодифицируемый алгоритм с начальной схемой, содержащей простые, заключительные, пополняющие формулы с одноразовыми пополнениями, назовем *алгоритмом с одноразовыми пополнениями слева*.

Один из основных нормальных алгорифмов Маркова — это *обращающий* алгорифм [1, § 32]. Для любого исходного слова во входном алфавите результатом такого алгорифма является *обращение* исходного слова. Например, обращение слова *aabca* — это слово *acbaa*.

В статье [2] предложен *обращающий самопополняемый слева алгорифм с простым и заключительным пополнениями*, в схеме которого *нет одноразовых формул*. В схеме алгорифма из [2] используются пополняющие формулы вида $A \rightarrow B \uparrow C \rightarrow D$ с простым пополнением $C \rightarrow D$ и вида $A \rightarrow B \uparrow C \rightarrow \bullet D$ с заключительным пополнением $C \rightarrow \bullet D$, где A, B, C, D — слова в заданном алфавите, не содержащем символов из множества $\{\rightarrow, \uparrow, \bullet\}$.

В настоящей статье предлагается *обращающий самомодифицируемый алгорифм*, являющийся алгорифмом с *одноразовыми пополнениями слева*. Доказывается теорема (и необходимые для нее леммы) о том, что результат такого алгорифма — это *обращение любого* исходного слова во входном алфавите. Сравняется сложность предлагаемого алгорифма, известного *обращающего нормального алгорифма Маркова* [1, § 32] и *обращающего самопополняемого слева алгорифма*, предложенного в [2].

Обозначения и термины. Пусть $X, V = \{\rightarrow, \uparrow, \bullet, \otimes, \#\}$, $E = \{0, 1, \infty\}$ такие алфавиты, что $X \cap V = \emptyset$; $X \cap E = \emptyset$; $V \cap E = \emptyset$. Применяем букву $\alpha \notin X \cup V \cup E$. Для алфавита H обозначаем $|H|$ его мощность, H^* совокупность всех слов в H , включая пустое слово Λ . Конкатенацию слов P и Q обозначаем PQ .

Пусть $I(P, Q) = 1$, если P является подсловом слова Q ; в противном случае $I(P, Q) = 0$. Обозначим $R(P, Q, T)$ слово, полученное заменой в слове P первого слева вхождения в него подслова $\mathfrak{B} \neq \mathfrak{V}$ на подслово T . Условимся, что $R(P, Q, T) = P$ при $I(Q, P) = 0$; $R(P, \Lambda, T) = TP$.

Если не указано иное, то далее x — это *любая* буква алфавита X ; A, B, C — *любые* слова из X^* .

Определим формулы $F_1 = \Lambda \rightarrow \alpha$, $F_2 = \alpha \rightarrow \bullet \Lambda$, $F_3^{(x)} = \alpha x \rightarrow \alpha \uparrow \Lambda \rightarrow \otimes x$, $F_4^{(x)} = \Lambda \rightarrow \otimes x$ (где $x \in X$) и множество

$G = \{F_3^{(x)} : x \in X\}$, в котором $|X|$ формул. Здесь и далее фигурные скобки, двоеточие и \in — метасимволы. Множество G неупорядоченное. В схеме алгоритма формулы из этого множества могут записываться в произвольном порядке с символом $\#$ после каждой формулы.

Обозначим $S = G\#F_2\#F_1\#$, $S_x = F_4^{(x)}\#S$.

Пусть P — это формула схемы алгоритма. Определим, что $U(P) = 4I(\otimes, P) + 2I(\uparrow, P) + I(\bullet, P)$; $T(P) = \circ$ при $U(P) \neq 1$, $T(P) = \bullet$ при $U(P) = 1$; $W(P) = \Lambda$ при $U(P) \in \{0; 1; 4; 5\}$, $W(P) = R_1(P, \uparrow)\#$ при $U(P) \notin \{0; 1; 4; 5\}$; $K(P) = R_2(P, \rightarrow)$; $L(P) = R_1(R(R_2(P, \uparrow), \bullet, \Lambda), \otimes, \Lambda), \rightarrow)$. Здесь $R_1(P, t)$ — слово, полученное заменой в слове P первого слева вхождения буквы t и всех букв слева от нее на Λ ; $R_1(P, t) = P$ при $I(t, P) = 0$; $R_2(P, u)$ — слово, полученное заменой в слове P первого слева вхождения буквы u и всех букв справа от нее на Λ ; $R_2(P, u) = P$ при $I(u, P) = 0$.

Для множества формул G и слова J условимся записывать $I(K(G), J) = 0$, если $I(K(P), J) = 0$ для каждой формулы P из G ; в противном случае $I(K(G), J) = 1$. Если во множестве G для слова J есть такая единственная формула P , что $I(K(P), J) = 1$, то полагаем $U(G) = U(P)$, $T(G) = T(P)$, $W(G) = W(P)$, $K(G) = K(P)$, $L(G) = L(P)$.

Пусть $F(Z) = n$, если $Z = P_n\#P_{n-1}\#\dots\#P_1\#$ — это схема алгоритма; $n \geq 1$; P_i , $i = \overline{1, n}$, — формула или множество формул. Длиной схемы называем число букв в ней, включая буквы алфавита V . Если схема — это слово в алфавите H , то мощностью алфавита схемы называем число $|H \setminus V|$, где $H \setminus V$ — разность множеств H и V .

Обозначим \vec{A} обращение слова A ; \vec{A} — это слово из X^* . Имеем $\vec{\Lambda} = \Lambda$, $\vec{Bx} = x\vec{B}$.

Пусть P — слово, $Y = P_m\#P_{m-1}\#\dots\#P_1\#$ — схема алгоритма, где P_j , $j = \overline{1, m}$, — формула или множество формул; $m \geq 1$. Определим, что кортежи $\langle Y, P \rangle$ и $\langle Z, Q \rangle$ связаны бинарным отношением $[P_i]$, $i \in \{1; 2; \dots; m\}$, для элементов которого задано значение $T(P_i)$ (обозначаем этот факт $\langle Y, P \rangle [P_i] T(P_i) \langle Z, Q \rangle$; переход от $\langle Y, P \rangle$ к $\langle Z, Q \rangle$

называем *шагом алгоритма*), если и только если $I(K(P_j), P) = 0$ для $j = \overline{i+1, m}$, $I(K(P_i), P) = 1$; причем $Z = W(P_i)Y$ при $U(P_i) \notin \{4, 5\}$, $Z = P_m \# P_{m-1} \# \dots \# P_{i+1} \# P_{i-1} \# \dots \# P_1 \#$ при $U(P_i) \in \{4, 5\}$; $Q = R(P, K(P_i), L(P_i))$.

Пусть Q_1 — слово, $Z_1 = P_m \# P_{m-1} \# \dots \# P_1 \#$ — схема алгоритма, где P_i , $i = \overline{1, m}$, — формула или множество формул; $m \geq 1$. Определим, что кортежи $\langle Z_1, Q_1 \rangle$ и $\langle Z, P \rangle$ связаны *рефлексивным, транзитивным* бинарным отношением \mapsto , для элементов которого задано значение $T(P_{i_n})$ (обозначаем этот факт $\langle Z_1, Q_1 \rangle \mapsto T(P_{i_n}) \langle Z, P \rangle$), если и только если выполнено одно из двух условий:

1) $Z = Z_1$ и $P = Q_1$ (при этом $T(P_{i_n}) = \circ$);

2) существует последовательность $\langle Z_1, Q_1 \rangle [P_{i_1}] T(P_{i_1}) \langle Z_2, Q_2 \rangle$, $\langle Z_2, Q_2 \rangle [P_{i_2}] T(P_{i_2}) \langle Z_3, Q_3 \rangle$, ..., $\langle Z_n, Q_n \rangle [P_{i_n}] T(P_{i_n}) \langle Z, P \rangle$ (для нее применяем обозначение $\langle Z_1, Q_1 \rangle [P_{i_1}] T(P_{i_1}) \langle Z_2, Q_2 \rangle [P_{i_2}] \dots [P_{i_n}] T(P_{i_n}) \langle Z, P \rangle$), где $n \geq 1$; $i_j \in \{1; 2; \dots; F(Z_j)\}$, $j = \overline{1, n}$; $T(P_{i_k}) = \circ$, $k = \overline{1, n-1}$; $T(P_{i_n}) \in \{\circ, \bullet\}$.

Для любых $Y = P_m \# P_{m-1} \# \dots \# P_1 \#$, P , Z , Q , $i \in \{1; 2; \dots; m\}$ из $\langle Y, P \rangle [P_i] T(P_i) \langle Z, Q \rangle$ следует $\langle Y, P \rangle \mapsto T(P_i) \langle Z, Q \rangle$, где $m \geq 1$.

Если $\langle Z_1, Q_1 \rangle \mapsto T(P_{i_n}) \langle Z, P \rangle$ (где $T(P_{i_n}) = \bullet$) и Q_1 — это *исходное* слово, то схему $Z_1 = P_m \# P_{m-1} \# \dots \# P_1 \#$ называем *начальной* схемой, число n — *числом шагов переработки слова* Q_1 начальной схемой Z_1 , схему Z — *конечной* схемой, слово P — *результатом* алгоритма с начальной схемой Z_1 . Здесь P_i , $i = \overline{1, m}$, — это формула или множество формул; $m \geq 1$. Алфавит исходных слов называем *входным алфавитом* и далее обозначаем X .

При переходе на следующую строку сочетание символов $[P] \circ$ (где P — формула), указанное в конце предыдущей строки, не повторяется в начале следующей строки.

Символ \blacklozenge показывает окончание доказательства.

Предлагаемый алгоритм. Предлагаемый обращающий само-модифицируемый алгоритм имеет начальную схему S . Она же является его конечной схемой. При переработке исходного слова на очередном шаге схемой может быть S или S_x для каждой буквы x этого исходного слова.

Назначение формул схем S и S_x следующее.

Простая формула F_1 выполняется только на шаге 1. Она добавляет слева к исходному слову подслово α .

Если исходное слово *пустое*, то на шаге 2 заключительная формула F_2 заменяет в текущем слове α подслово α на пустое подслово Λ . На этом выполнение алгоритма заканчивается с результатом $\Lambda = \bar{\Lambda}$.

Если исходное слово $A = xB$ *непустое* длины λ , то на шаге 2 пополюющая формула $F_3^{(x)}$ из множества G заменяет в слове αxB подслово αx на подслово α и добавляет в начало схемы S одноразовую формулу $F_4^{(x)}$. Текущей схемой станет S_x , текущим словом — αB .

На шаге 3 одноразовая формула $F_4^{(x)}$ добавляет слева к слову αB подслово x и удаляется из схемы S_x . Текущей схемой станет S , текущим словом — $x\alpha B$.

На шагах 4, 5, ..., $2\lambda + 1$ пополюющие формулы $F_3^{(x)}$ из множества G и одноразовые формулы $F_4^{(x)}$ для букв x из подслова B завершают получение подслова \bar{A} . После шага $2\lambda + 1$ текущей схемой будет S , текущим словом — $\bar{A}\alpha$.

На шаге $2\lambda + 2$ заключительная формула F_2 заменяет в текущем слове $\bar{A}\alpha$ подслово α на пустое подслово Λ . На этом выполнение алгоритма заканчивается с результатом \bar{A} , являющимся обращением исходного непустого слова A .

В обозначениях из [3] начальная схема предлагаемого обращающего само-модифицируемого алгоритма записывается как слово $H_1 \# \alpha \infty \bullet \# \infty \alpha \#$ в алфавите $X \cup \{\alpha\} \cup E \cup \{\bullet, \#\}$. Здесь $H_1 = \{\alpha x \infty \alpha 10 \infty 10 x : x \in X\}$.

Вспомогательные утверждения. Для доказательства теоремы о предлагаемом алгоритме потребуются леммы 1–10.

Лемма 1. $\langle S, A \rangle [F_1] \circ \langle S, \alpha A \rangle$.

Доказательство. Рассмотрим схему S . Так как $\alpha \notin X^*$, получим $I(\alpha x, A) = 0$ (при любом $x \in X$) и $I(\alpha, A) = 0$, поэтому $I(K(G), A) = 0$ и $I(K(F_2), A) = 0$.

Из $I(\Lambda, A) = 1$ следует $I(K(F_1), A) = 1$. Так как $U(F_1) = 0$, $T(F_1) = \circ$, $W(F_1) = \Lambda$, $K(F_1) = \Lambda$, $L(F_1) = \alpha$, $R(A, \Lambda, \alpha) = \alpha A$, получим $\langle S, A \rangle [F_1] \circ \langle S, \alpha A \rangle$. ♦

Лемма 2. $\langle S, \Lambda \rangle [F_1] \circ \langle S, \alpha \rangle$.

Доказательство. Согласно лемме 1 имеем $\langle S, A \rangle [F_1] \circ \langle S, \alpha A \rangle$ для любого $A \in X^*$. Тогда при $A = \Lambda$ получим $\langle S, \Lambda \rangle [F_1] \circ \langle S, \alpha \rangle$. ♦

Лемма 3. $\langle S, A\alpha \rangle [F_2] \bullet \langle S, A \rangle$.

Доказательство. Рассмотрим схему S . Так как $\alpha \notin X^*$, получим $I(\alpha x, A\alpha) = 0$ при любом $x \in X$, поэтому $I(K(G), A\alpha) = 0$.

Из $I(\alpha, A\alpha) = 1$ следует $I(K(F_2), A\alpha) = 1$. Так как $U(F_2) = 1$, $T(F_2) = \bullet$, $W(F_2) = \Lambda$, $K(F_2) = \alpha$, $L(F_2) = \Lambda$, $R(A\alpha, \alpha, \Lambda) = A$ (с учетом того, что $\alpha \notin X^*$), получим $\langle S, A\alpha \rangle [F_2] \bullet \langle S, A \rangle$. ♦

Лемма 4. $\langle S, \alpha \rangle [F_2] \bullet \langle S, \Lambda \rangle$.

Доказательство. Согласно лемме 3 имеем $\langle S, A\alpha \rangle [F_2] \bullet \langle S, A \rangle$ для любого $A \in X^*$. Тогда при $A = \Lambda$ получим $\langle S, \alpha \rangle [F_2] \bullet \langle S, \Lambda \rangle$. ♦

Лемма 5. $\langle S, \Lambda \rangle \mapsto \bullet \langle S, \bar{\Lambda} \rangle$.

Доказательство. Согласно леммам 2 и 4 имеем соответственно $\langle S, \Lambda \rangle [F_1] \circ \langle S, \alpha \rangle$ и $\langle S, \alpha \rangle [F_2] \bullet \langle S, \Lambda \rangle$. Следовательно, $\langle S, \Lambda \rangle \mapsto \bullet \langle S, \bar{\Lambda} \rangle$. Так как $\Lambda = \bar{\Lambda}$, получим $\langle S, \Lambda \rangle \mapsto \bullet \langle S, \bar{\Lambda} \rangle$. ♦

Лемма 6. $\langle S, A\alpha x B \rangle [G] \circ \langle S_x, A\alpha B \rangle$.

Доказательство. Рассмотрим схему S . Из $I(\alpha x, A\alpha x B) = 1$ следует $I(K(G), A\alpha x B) = 1$ при любом $x \in X$. Так как $U(G) = 6$, $T(G) = \circ$, $W(G) = \Lambda \rightarrow \otimes x \#$, $K(G) = \alpha x$, $L(G) = \alpha$,

$S_x = W(\bar{G})S = \Lambda \rightarrow \otimes x \# S = F_4^{(x)} \# S$, $R(A\alpha x B, \alpha x, \alpha) = A\alpha B$ (с учетом того, что $\alpha \notin X^*$), получим $\langle S, A\alpha x B \rangle [G] \circ \langle S_x, A\alpha B \rangle$. \blacklozenge

Лемма 7. $\langle S_x, A\alpha B \rangle [F_4^{(x)}] \circ \langle S, x A\alpha B \rangle$.

Доказательство. Рассмотрим схему S_x . Из $I(\Lambda, A\alpha B) = 1$ следует $I(K(F_4^{(x)}), A\alpha B) = 1$ при любом $x \in X$. Так как $U(F_4^{(x)}) = 4$, $T(F_4^{(x)}) = \circ$, $W(F_4^{(x)}) = \Lambda$, $K(F_4^{(x)}) = \Lambda$, $L(F_4^{(x)}) = x$, $R(A\alpha B, \Lambda, x) = x A\alpha B$, получим $\langle S_x, A\alpha B \rangle [F_4^{(x)}] \circ \langle S, x A\alpha B \rangle$. \blacklozenge

Лемма 8. $\langle S, A\alpha x B \rangle \mapsto \circ \langle S, x A\alpha B \rangle$.

Доказательство. Согласно леммам 6 и 7 имеем $\langle S, A\alpha x B \rangle [G] \circ \langle S_x, A\alpha B \rangle$ и $\langle S_x, A\alpha B \rangle [F_4^{(x)}] \circ \langle S, x A\alpha B \rangle$ соответственно. Тогда $\langle S, A\alpha x B \rangle \mapsto \circ \langle S, x A\alpha B \rangle$. \blacklozenge

Лемма 9

$$\langle S, \alpha AC \rangle \mapsto \circ \langle S, \bar{A}\alpha C \rangle. \quad (1)$$

Доказательство. Используем метод правой индукции по построению слова A [1, с. 16].

1. При $A = \Lambda$ (1) имеет место по определению отношения \mapsto с учетом того, что $\bar{\Lambda} = \Lambda$.

2. Покажем, что если (1) имеет место для некоторого $A \in X^*$, то (1) имеет место и при замене A на Ax . Пусть для некоторого $A \in X^*$ при любом $C \in X^*$ имеет место (1). Обозначим $C = xB$. Тогда $\langle S, \alpha Ax B \rangle \mapsto \circ \langle S, \bar{A}\alpha x B \rangle$. По лемме 8 имеем $\langle S, A\alpha x B \rangle \mapsto \circ \langle S, x A\alpha B \rangle$ для любых слов $A \in X^*$ и $B \in X^*$. Тогда $\langle S, \bar{A}\alpha x B \rangle \mapsto \circ \langle S, x \bar{A}\alpha B \rangle$ с учетом того, что $\bar{A} \in X^*$. Поэтому $\langle S, \alpha Ax B \rangle \mapsto \circ \langle S, x \bar{A}\alpha B \rangle$. Имеем $x \bar{A} = \overleftarrow{A}x$. Таким образом, $\langle S, \alpha Ax B \rangle \mapsto \circ \langle S, \overleftarrow{A}x\alpha B \rangle$. Следовательно, если (1) имеет место для

некоторого $A \in X^*$ при любом $C \in X^*$, то (1) имеет место и при замене A на Ax для $C = B$.

На основании пп. 1 и 2 заключаем, что (1) имеет место при любом $A \in X^*$. ♦

Лемма 10. $\langle S, \alpha A \rangle \mapsto \circ \langle S, \vec{A} \alpha \rangle$.

Доказательство. По лемме 9 имеем (1) для любых слов $A \in X^*$ и $C \in X^*$. Тогда при $C = \Lambda$ получим $\langle S, \alpha A \rangle \mapsto \circ \langle S, \vec{A} \alpha \rangle$. ♦

Основное утверждение. Докажем основное для предлагаемого алгоритма утверждение — теорему о том, что результат этого алгоритма является обращением *любого* исходного слова во входном алфавите X .

Теорема. $\langle S, A \rangle \mapsto \bullet \langle S, \vec{A} \rangle$.

Доказательство. Пусть $A = \Lambda$. Тогда $\vec{A} = \vec{\Lambda}$ и $\langle S, \Lambda \rangle \mapsto \bullet \langle S, \vec{\Lambda} \rangle$ в силу леммы 5.

Пусть $A \neq \Lambda$. Тогда $\langle S, A \rangle [F_1] \circ \langle S, \alpha A \rangle$ в силу леммы 1, $\langle S, \alpha A \rangle \mapsto \circ \langle S, \vec{A} \alpha \rangle$ в силу леммы 10, $\langle S, \vec{A} \alpha \rangle [F_2] \bullet \langle S, \vec{A} \rangle$ в силу леммы 3 с учетом того, что $\vec{A} \in X^*$. Следовательно, $\langle S, A \rangle \mapsto \bullet \langle S, \vec{A} \rangle$. ♦

Характеристики предлагаемого алгоритма. Сравним предлагаемый обращающий *самомодифицируемый* алгоритм, являющийся алгоритмом с *одноразовыми пополнениями* слева, с имеющими ту же мощность алфавита схемы обращающим *нормальным* алгоритмом Маркова [1, с. 142] и обращающим *самополняемым* слева алгоритмом с *простым* и *заключительным* пополнениями, в схеме которого *нет одноразовых формул* [2].

В принятых в настоящей статье обозначениях этот обращающий нормальный алгоритм Маркова имеет начальную схему $M = \alpha \alpha \alpha \rightarrow \alpha \alpha \# G_1 \# \alpha \alpha \rightarrow \bullet \Lambda \# G_2 \# \Lambda \rightarrow \alpha \#$ (она же является конечной схемой). Здесь $G_1 = \{ \alpha \alpha x \rightarrow x \alpha \alpha : x \in X \}$ и $G_2 = \{ \alpha x y \rightarrow y \alpha x : x \in X, y \in X \}$ — множества формул (где запятая — это метасимвол), имеющие мощности $|X|$ и $|X|^2$ соответственно. Формулы из каждого из этих множеств могут записываться в схеме M в произвольном порядке с символом $\#$ после каждой формулы.

В принятых в настоящей статье обозначениях этот обращающий самопополняемый слева алгоритм с простым и заключительным пополнениями, в схеме которого нет одноразовых формул, имеет начальную схему $D = \alpha\alpha\alpha \rightarrow aa \uparrow \alpha \rightarrow \Lambda\#\alpha\alpha \rightarrow \alpha\alpha\alpha \uparrow aa \rightarrow \bullet\Lambda\#G_2\#\Lambda \rightarrow \alpha\#$ и конечную схему $D_1 = \alpha \rightarrow \Lambda\#aa \rightarrow \bullet\Lambda\#D$. Здесь $a \in X$. Формулы из множества G_2 могут записываться в схеме D в произвольном порядке с символом $\#$ после каждой формулы.

Сравнение выполним по двум характеристикам, определяющим сложность алгоритма: 1) число шагов переработки слова; 2) длина схемы (начальной и конечной). Для любого нормального алгоритма Маркова начальная и конечная схемы равны, поэтому равны их длины.

Для исходного слова длины λ число шагов $l_1(\lambda)$ переработки начальной схемой M обращающего нормального алгоритма Маркова и число шагов $l_2(\lambda)$ переработки начальной схемой D обращающего самопополняемого слева алгоритма с простым и заключительным пополнениями, в схеме которого нет одноразовых формул, определяются так (см. [2]): $l_1(\lambda) = 1 + \lambda(\lambda + 5)/2$ при $\lambda > 0$; $l_1(0) = 3$; $l_2(\lambda) = 3 + \lambda(\lambda + 3)/2$ при $\lambda > 0$; $l_2(0) = 5$.

Для исходного слова длины λ число шагов $l_3(\lambda)$ переработки начальной схемой S предлагаемого обращающего самомодифицируемого алгоритма определяется так: $l_3(\lambda) = 2 + 2\lambda$ при $\lambda \geq 0$.

Для входного алфавита X длины начальной и конечной схем M обращающего нормального алгоритма Маркова равны $8|X|^2 + 8|X| + 15$; длины начальной схемы D и конечной схемы D_1 обращающего самопополняемого слева алгоритма с простым и заключительным пополнениями, в схеме которого нет одноразовых формул, равны $8|X|^2 + 25$ и $8|X|^2 + 33$ (максимальная длина схемы) соответственно [2].

Для входного алфавита X длина начальной схемы S (она же является конечной схемой) предлагаемого обращающего самомодифицируемого алгоритма равна $9|X| + 7$; длина схемы S_x (максимальная длина схемы этого алгоритма в процессе переработки слова) равна $9|X| + 11$ при любом $x \in X$.

Мощности алфавитов схем M , D и S одинаковы и равны $|X| + 1$.

Пример. Для входного алфавита $X = \{a, b\}$ предлагаемый обращающий самомодифицируемый алгоритм имеет начальную схему $S = F_3^{(a)} \# F_3^{(b)} \# F_2 \# F_1 \#$. Здесь $F_3^{(a)} = \alpha a \rightarrow \alpha \uparrow \Lambda \rightarrow \otimes a$; $F_3^{(b)} = \alpha b \rightarrow \alpha \uparrow \Lambda \rightarrow \otimes b$.

Мощность алфавита схемы S равна 3. Ее длина равна 25. Длина каждой из схем S_a и S_b равна 29. Здесь $S_a = F_4^{(a)} \# S$; $S_b = F_4^{(b)} \# S$; $F_4^{(a)} = \Lambda \rightarrow \otimes a$; $F_4^{(b)} = \Lambda \rightarrow \otimes b$.

Для этого алфавита X схемы M , D и D_1 имеют длины 63, 57 и 65 соответственно; мощности алфавитов схем M и D равны 3.

Для исходного слова Λ имеем $\langle S, \Lambda \rangle [F_1] \circ \langle S, \alpha \rangle [F_2] \bullet \langle S, \Lambda \rangle$, т.е. $\langle S, \Lambda \rangle \mapsto \bullet \langle S, \overleftarrow{\Lambda} \rangle$, так как $\overleftarrow{\Lambda} = \Lambda$. Число шагов переработки слова Λ начальной схемой S равно 2, начальной схемой M — 3, начальной схемой D — 5.

Для исходного слова aab имеем

$$\begin{aligned} & \langle S, aab \rangle [F_1] \circ \langle S, \alpha aab \rangle [F_3^{(a)}] \circ \langle S, \alpha ab \rangle [F_4^{(a)}] \circ \\ & \langle S, a\alpha ab \rangle [F_3^{(a)}] \circ \langle S, a\alpha b \rangle [F_4^{(a)}] \circ \langle S, a\alpha\alpha b \rangle [F_3^{(b)}] \circ \\ & \langle S_b, a\alpha\alpha \rangle [F_4^{(b)}] \circ \langle S, ba\alpha\alpha \rangle [F_2] \bullet \langle S, ba\alpha \rangle. \end{aligned}$$

Тогда $\langle S, aab \rangle \mapsto \bullet \langle S, ba\alpha \rangle$, поэтому $\langle S, aab \rangle \mapsto \bullet \langle S, \overleftarrow{aab} \rangle$.

Число шагов переработки слова aab начальной схемой S равно 8, начальной схемой M — 13, начальной схемой D — 12.

Заключение. В статье предложен обращающий самомодифицируемый алгоритм в форме алгоритма с однократными пополнениями слева. Доказана теорема (и необходимые для нее леммы) о том, что результатом предлагаемого алгоритма является обращение *любого* исходного слова во входном алфавите.

В сравнении с известным обращающим нормальным алгоритмом Маркова [1, § 32] и предложенным в статье [2] обращающим самопополняемым слева алгоритмом с простым и заключительным пополнениями, в схеме которого нет однократных формул, предлагаемый

в настоящей статье алгоритм в общем случае имеет при той же мощности алфавита схемы *меньшую сложность*:

1) меньшее число шагов переработки слова (кроме слов длины 1 в сравнении с алгоритмом из [1] — для таких слов число шагов переработки у предлагаемого алгоритма и алгоритма из [1] одинаковое и равно 4);

2) меньшую длину схемы (начальной и конечной).

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Марков А.А., Нагорный Н.М. Теория алгоритмов. — 2-е изд. — М.: ФАЗИС, 1996. — 496 с.

2. Цветков И.А. Обращающий самопополняемый слева алгоритм в алфавите с одной дополнительной буквой // Математическое и программное обеспечение вычислительных систем: межвуз. сб. науч. тр. — М.: Горячая линия — Телеком, 2008. — С. 4–9.

3. Цветков И.А. Самомодифицируемый алгоритм как обобщение нормального и самопополняемых алгоритмов Маркова // Вестник РГРТА. — 2005. — Вып. 16. — С. 13–19.

Е.С. ШЛЫКОВ, О.В. МИЛОВЗОРОВ

Рязанский государственный радиотехнический университет

ОБ ОДНОМ ИЗ ПОДХОДОВ К СОЗДАНИЮ СИСТЕМ ПЛАНИРОВАНИЯ МЕХАНОСБОРОЧНОГО ПРОИЗВОДСТВА

Рассматриваются принципы создания систем планирования машиностроительного производства для малых высокотехнологичных предприятий.

Системы планирования и управления работой производственных предприятий известны уже достаточно давно. Этой проблеме было посвящено большое количество работ еще в советский период. Основным подходом к созданию таких систем на российских машиностроительных предприятиях был подход, основанный на двух главных моментах:

- базирование на традиционной для российских предприятий маршрутно-операционной технологии;
- учет конкретных особенностей машиностроительного предприятия.

Этот подход привел к тому, что, системы планирования в основном создавались службами АСУ конкретных предприятий, были уни-

кальными, глубоко специализированными и не могли быть использована на других предприятиях. Системы, основы которых были заложены еще в советский период, в 90-е годы прошлого века совершенствовались вместе с развитием средств вычислительной техники, возможностей сетевых операционных систем и языковых средств программирования. Однако они по-прежнему оставались уникальными, т.е. разработанными с учетом конкретных особенностей производства. Базой для алгоритмизации этих систем были различные эвристические алгоритмы, характерные для принятия решения о планировании производства на данном конкретном предприятии.

В начале 90-х годов в связи с общесистемным кризисом российского промышленного производства, резким сокращением объемов производства и переходом на рыночные отношения системы планирования потеряли свою актуальность, поскольку в это время предприятия были вынуждены решать более насущные проблемы, связанные, прежде всего, с выживанием в новых условиях. Но уже к концу 90-х годов интерес к системам планирования вновь оживляется и к настоящему времени продолжает возрастать.

Структурные изменения, которым подверглись практически все машиностроительные предприятия России, привели к необходимости в новых подходах к созданию систем планирования производства. Эти изменения в той или иной степени коснулись всех структурных подразделений. На смену отделов и служб АСУ пришли ИТ отделы, причем в резко сокращенном виде. Если раньше предприятие располагало целым штатом квалифицированных программистов, которые были в состоянии создавать комплексные программные решения, то теперь функции ИТ отдела кардинальным образом изменились. Основными задачами новых информационных служб стали выбор имеющихся на рынке САПР готовых унифицированных систем, обеспечивающих автоматизацию тех или иных работ на предприятии и их адаптация к конкретным условиям. При этом зачастую основным аргументом в пользу покупки программного продукта является его дешевизна.

Результатом такого подхода стала «лоскутная» автоматизация, в результате которой информационное пространство предприятия формировалось из набора слабо связанных между собой САПР.

Осознание того, что «лоскутная» автоматизация – тупиковый путь, привело к возрастанию интереса у предприятий к комплексным решениям, формированию единой базы данных на основе PDM-систем. При этом зарубежные системы в силу своей дороговизны и недостаточной адаптированности к российскому производству все больше уступают место российским системам. В первую очередь, это

комплекс программных продуктов 1С, а также системы, специализирующиеся на автоматизации машиностроительных производств, таких как Лощман (Аскон), T-Flex DOCs (Топ Системы). Современные тенденции по созданию информационного пространства на российских машиностроительных предприятиях таковы, что все больше производителей машиностроительной продукции, причем, не только крупных, но и мелких, стремятся сформировать свое информационное пространство на базе современных систем PDM.

Среди различных систем, обеспечивающих автоматизированное управление жизненным циклом изделия российской разработки есть и системы, предназначенные для решения задач планирования. К ним относятся и такие системы, как Парус, Галактика. Однако опыт работы промышленных предприятий показывает, что и эти системы не обеспечивают в полной мере потребности предприятия (долгий и не всегда корректный процесс адаптации, некомплексное решение вопросов).

В последнее время в России стали появляться машиностроительные предприятия, реализующие нетрадиционные для постсоветского пространства подходы к организации производства. Для них характерно широкое применение станков с ЧПУ, минимизация складских помещений, резкое уменьшение сроков освоения изделия и его изготовления. Эти предприятия имеют сравнительно небольшую численность персонала. Небольшое, но динамичное и высокотехнологичное предприятие имеет явные преимущества перед традиционными масштабными машиностроительными предприятиями. Оно способно быстро переориентироваться на выпуск перспективной продукции с минимальными материальными и временными затратами. При этом за основу производственного планирования здесь берутся не традиционные маршрутно-операционные технологические процессы, а некие обобщенные технологические карты, для которых нет устоявшихся стандартов – как утвержденных на уровне ГОСТ, так и на уровне СТП.

Производственный процесс на таком предприятии может быть выстроен следующим образом. Инженерная группа получает комплект электронной конструкторской документации на изделия и данные по планируемым объемам производства. На их основе осуществляется разработка управляющих программ для станков с ЧПУ, формируются заявки на необходимые материалы и другие виды ресурсов, планируется загрузка оборудования, и, в конечном итоге, формируется производственный план, который в сжатые сроки запускается в производство. Основными критериями для планирования в этом случае могут быть либо минимальные временные затраты на освоение и изготовление партии и минимальные затраты на складские площади, либо точ-

ная привязка по срокам изготовления партий деталей с тем, чтобы обеспечить непрерывный ход производственного процесса. В последнем случае возможна минимизация расходов на транспортировку и временное хранение материалов, деталей и изделий за счет выверенной логистической модели.

В традиционном понимании такое планирование можно отнести к оперативно-календарному виду планирования. Совершенно очевидно, что традиционный подход к планированию такого типа производства, основанный на маршрутно-операционных технологических процессах, не годится, а, следовательно, и все имеющиеся на сегодняшний день унифицированные системы планирования использоваться не могут. Отсюда возникает задача к формированию новых подходов к созданию систем планирования динамичных машиностроительных производств малой мощности.

В качестве базовых понятий для формализации задачи планирования возьмем понятия «ресурс» и «работа». Под ресурсом будем понимать любой физический объект, который непосредственно используется, или на основе которого осуществляется производственный процесс (таблица 1).

Таблица 1. Виды ресурсов и их характеристика

Вид ресурса	Краткая характеристика ресурса
Производственные ресурсы	Все ресурсы, стоимость использования которых включена в стоимость конечного продукта
Ресурсы обеспечения производства	Представляют собой все ресурсы, которые регламентируют производственные процессы
Материальные ресурсы	Сырьё, материалы, запатентованные технологии
Финансовые ресурсы	Финансовая обеспеченность производства
Людские ресурсы	Работники

Под понятием «работа» будем понимать любой процесс, осуществляемый на базе ресурсов и с использованием ресурсов, направленный на производство машиностроительного изделия.

Ресурсы делятся на две большие группы: воспроизводимые и невоспроизводимые. Невоспроизводимые (складируемые, накапливаемые) ресурсы в процессе выполнения работ расходуются полностью, не допуская повторного использования. Не использованные в данный отрезок времени, они могут использоваться в дальнейшем. Иными словами, такие ресурсы можно накапливать с последующим расходом запасов. Примерами таких ресурсов являются топливо, пред-

меты, средства труда однократного применения, а также – финансовые средства.

Воспроизводимые (не складированные, не накапливаемые) ресурсы сохраняют свою натурально-вещественную форму и по мере высвобождения могут использоваться на других работах. Примерами ресурсов типа «мощности» являются люди и средства труда многократного использования (машины, станки, механизмы и т.д.).

Как ресурс, так и работа имеют иерархическую структуру, которая позволяет описывать эти базовые понятия с разной степенью детализации в зависимости от уровня рассмотрения.

Так, например, структура ресурса «завод» может быть представлена совокупностью цехов и отделов, которые в свою очередь подразделяются на участки и группы, состоящие из конкретных людей и оборудования, на котором они работают (рис. 1).

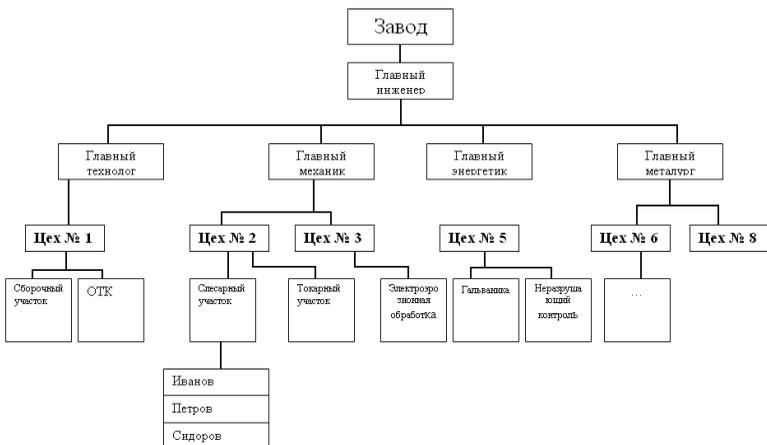


Рис. 1. Пример иерархической структуры ресурса

Иерархичность понятия «работа» легко пояснить на примере процесса изготовления любого машиностроительного изделия. Изделие состоит из узлов, которые, в свою очередь, состоят из деталей. Последние могут быть изготовлены производством, либо приобретены службой материально-технического снабжения (например, стандартные детали крепежа). Т.е. работа по изготовлению изделия будет включать в себя работы по приобретению комплектующих, изготовлению деталей, сборке узлов. На самом верхнем уровне будет осуществляться собственно сборка изделия.

Исходя из базовых понятий «работа» и «ресурс» задачу планирования производственного процесса можно рассматривать, как процесс взаимного увязывания работ и ресурсов (рис. 2). При этом все связи проецируются на ось времени, что позволяет закреплять те или иные виды ресурсов для выполнения планируемых работ на определенный календарный срок.

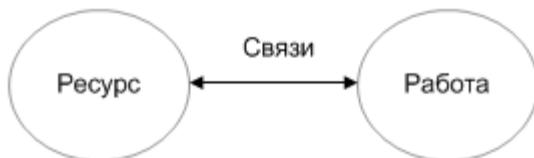


Рис. 2. К пояснению задачи планирования

Различные виды ресурсов, в свою очередь, также увязываются между собой.

Рассмотрим упрощенный пример взаимосвязи ресурсов в следующей задаче:

Задача – изготовить деталь N. Покажем взаимосвязи ресурса 1 – «Рабочие» и ресурса 2 – «Оборудование».

Ресурс 1 – «Рабочие»

Основные свойства ресурса:

Профессия;

Квалификация;

Зарплата;

Календарь рабочего времени...

Ресурс 2 – «Оборудование»

Основные свойства (признаки) ресурса

Выполняемые виды обработки;

Число одновременно управляемых координат;

Стоимость (остаточная);

Амортизация;

Фонд рабочего времени;

Средняя стоимость минимального периода работы;

Квалификация оператора...

Свойства ресурса могут описываться, как отдельными элементами, так и целыми наборами и даже многомерными массивами. Например, свойство «Выполняемые виды обработки» можно представить в виде двухмерной таблицы с установлением взаимосвязи между обрабатываемыми поверхностями деталей и возможностями станка. При

этом можно учесть и геометрические характеристики обрабатываемой поверхности, и точностные, и физико-механические.

Для описания доступности ресурсов в разные моменты времени можно использовать календарь. Для удобства календари в системе можно разделить на:

1. Организационно-производственный календарь (ОПК);
2. Календарь ресурсов (КР).

ОПК календарь представляет собой общее описание производственного календаря организации (фонд рабочего времени).

КР описывает доступность ресурсов во времени (соответственно, каждый ресурс имеет свой календарь, в котором содержатся данные о доступности). Календарь ресурсов дает возможность наглядного связывания конкретных людей, станков (оборудованию) и ресурсов (про Stanовка отпусков, отгулов, индивидуальное рабочее время, профилактический ремонт и пр.) и обеспечит возможность дополнительной коррекции графиков и отражения появляющихся внештатных ситуаций как в ОПК.

Для выполнения процедуры связывания ресурсов между собой и с работами используется набор условий. В нашем случае для обеспечения возможности использования станка (ресурс «оборудование») и рабочего в процессе изготовления детали необходимо одновременное выполнение следующих условий:

Стоимость детали должна быть достаточно высокой, чтобы оправдать использование станка;

- Сложность детали должна максимально соответствовать возможностям станка;

- Виды обработки, выполняемые на станке (токарная, фрезерная, сверлильно-расточная, электроэрозионная, термическая и т.п.), должны соответствовать тем технологическим преобразованиям, которые необходимы для изготовления детали;

- Специальность задействованного рабочего должна соответствовать используемому оборудованию, в нашем случае – «Оператор станков с ЧПУ»;

- Квалификация рабочего должна быть достаточной для работы на выбранном станке;

- И рабочий и станок должны быть не загружены другими заказами.

Среди вышеперечисленных условий есть жесткие, которые должны соблюдаться обязательно, и нежесткие. К последним относятся первые два условия. Их можно рассматривать как дополнительные критерии при решении задачи минимизации затраченных средств (ма-

териальных и временных), хотя, в случае необходимости (например, когда на первый план выдвигаются жесткие сроки изготовления), ими можно пренебречь.

Из приведенного примера видно, что при планировании производства малого машиностроительного предприятия необходимо исходить из следующих основных аспектов:

- с точки зрения технологических возможностей (основной аспект);
- с точки зрения срочности работы (дополнительный аспект при минимизации стоимости изделия);
- с точки зрения стоимости работы (дополнительный аспект при минимизации сроков изготовления изделия).

Действительно, некоторые виды работ могут быть выполнены только на определенном станке. Некоторые работы можно выполнить с потерей времени на других станках (более простых станках с ЧПУ, либо универсальных станках без ЧПУ), либо, если работа очень срочная, она может быть выполнена на станках более высокого уровня, с существенным увеличением стоимости, но с сокращением времени обработки.

Задача планирования заключается в установлении соответствия между работами и ресурсами, закреплении за работами в течение необходимого времени определенных видов ресурсов. При этом процесс связывания и закрепления во временном контексте работ и ресурсов должен обеспечивать минимизацию времени выполнения проекта, стоимости изготовления изделия, либо иную (в т.ч. многокритериальную) оптимизацию в зависимости от текущих потребностей заказчика.

Для формализации задачи планирования базовые понятия «ресурс» и «работа» можно описать с использованием математического аппарата теории множеств. Например, работников можно описать в виде множества

$$R = \bigcup R_i,$$

в котором каждый элемент описывается кортежем

$$R_i = (Pr, Qw, Zp, Cal), \text{ где}$$

Pr – элемент «профессия»;

Qw – элемент «квалификация», т.е. разряд рабочего;

Zp – элемент «зарплата» (тариф или оклад);

Cal – элемент «календарь рабочего времени».

Аналогично можно представить и все прочие ресурсы. Например, оборудование представляется множеством

$$O = \bigcup O_i,$$

где $O_i = (Mk, NNC, Prc, Am, Fwt, Cmin, Qw)$;

Mk – вид обработки;

NNC – число одновременно управляемых координат ЧПУ;

Prc – стоимость остаточная;

Am – амортизация;

Fwt – фонд рабочего времени;

$Cmin$ – средняя стоимость минимального периода работы;

Qw – квалификация оператора.

Отдельные элементы кортежей R_i и O_i обеспечивают реляционную связь. В нашем примере это элемент Qw .

Вывод: Предложенный подход по формальному описанию базовых элементов производственного процесса позволяет достаточно просто перейти к формированию Технического задания на разработку базы знаний, на основе которой будет осуществляться процесс планирования производственной деятельности малого высокотехнологичного машиностроительного предприятия.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Миловзоров О.В., Шлыков Е.С. К вопросу о разработке системы планирования, как составляющей программного комплекса T-Flex // «Новые технологии в учебном процессе и производстве», Рязань, РИ МГОУ, 2009г. Материалы межвузовского семинара и научно-технической конференции студентов, молодых ученых и специалистов.

2. Корячко В.П., Шлыков Е.С., Миловзоров О.В. Основные подходы к организации и представлению данных при проектировании системы планирования // Материалы 14-й всероссийской научно-технической конференции студентов, молодых ученых и специалистов «Новые информационные технологии в научных исследованиях и в образовании НИТ-2009». Рязань, РГРТА, 2009.

3. Шлыков Е.С., Миловзоров О.В. К вопросу о принципах организации системы планирования машиностроительным предприятием // «Новые технологии в учебном процессе и производстве», Рязань, РИ МГОУ, 2010г. Материалы межвузовского семинара и научно-технической конференции студентов, молодых ученых и специалистов.