

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

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



ПОБЕДИТЕЛЬ КОНКУРСА ИННОВАЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ ВУЗОВ

**Сборник трудов
конференции молодых ученых**

Выпуск 4

**МАТЕМАТИЧЕСКОЕ
МОДЕЛИРОВАНИЕ
И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**



**САНКТ-ПЕТЕРБУРГ
2009**

В издании «Сборник трудов конференции молодых ученых, Выпуск 4. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ публикуются работы, представленные в рамках VI Всероссийской межвузовской конференции молодых ученых, которая будет проходить 14–17 апреля 2009 года в Санкт-Петербургском государственном университете информационных технологий, механики и оптики.



СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007-2008 годы и успешно реализовал инновационную образовательную программу «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий», что позволило выйти на качественно новый уровень подготовки выпускников и удовлетворять возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях науки. Реализация этой программы создала основу формирования программы дальнейшего развития вуза до 2015 года, включая внедрение современной модели образования.

СИСТЕМНЫЙ АНАЛИЗ, МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И УПРАВЛЕНИЕ В ТЕХНИЧЕСКИХ СИСТЕМАХ

УДК 519.71

СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ ОЦЕНКЕ ТИПОВЫХ ТРАЕКТОРИЙ ПОЛЕТА

Л.М. Неугодникова

(Уфимский государственный авиационный технический университет)

Научный руководитель – д.т.н., профессор В.Н. Ефанов

(Уфимский государственный авиационный технический университет)

Целью настоящего исследования является разработка алгоритмического и программного обеспечения бортовой системы поддержки принятия решений, предлагаемой для оценки топливно-временных характеристик типовых траекторий полета, содержащихся в памяти автоматической бортовой системы самолетовождения. Для решения поставленной задачи разработан программный комплекс «Компара», позволяющий рассчитывать все необходимые параметры опорных траекторий с учетом аэродинамических характеристик конкретного летательного аппарата.

Ключевые слова: типовая траектория, топливно-временная оценка, аэродинамика

Введение

Глобальная модернизация национальных систем управления воздушным движением, вызванная необходимостью более эффективной организации воздушного пространства, становится все более актуальной и для российской Единой системы организации воздушного движения (ЕС ОВД). Планируется существенное повышение степени технического обеспечения действующих и вновь открываемых международных и внутренних воздушных трасс средствами воздушной и наземной связи, наблюдения и управления, отвечающими требованиям ИКАО.

Внедрение новейших технологий устранил необходимость в негибких и экономически неэффективных наземных системах проводки и обеспечения воздушных трасс, а также в связанных с ними навигационных инфраструктурах. В этих условиях наземные диспетчеры должны будут руководствоваться концепцией тактического (локального) эшелонирования, основанной на параметрах местоположения и вектора скорости самолетов, а не концепцией стратегического эшелонирования, основанной на параметрах траектории полета, что подразумевает выполнение полетов по назначенным маршрутам с заданными высотами и скоростями, как это делается сегодня. Таким образом, реализуется совершенно иной подход к управлению воздушным движением, который вкладывает новый смысл в понятие «полет по приборам».

При этом следует учесть, что бортовые системы пилотажно-навигационного оборудования современных самолетов предоставляют экипажу огромный поток информации. В то же время координация работы всех систем возложена на экипаж. В таких условиях перед экипажем современного самолета встают серьезнейшие проблемы, связанные с обеспечением ситуационной уверенности при пилотировании:

- сложность учета аэродинамических характеристик самолета применительно к конкретным условиям эксплуатации (загрузка самолета, погодные условия) при выборе оптимального маршрута;
- необходимость комплексной оценки текущего эксплуатационного состояния бортовых систем на основе разнородной, быстро меняющейся информации;

- требование оперативного принятия решений в условиях изменяющейся воздушной обстановки на основе прогноза влияния таких изменений на дальнейший полет.

Важным средством решения указанных проблем является использование интеллектуальных бортовых систем поддержки принятия решений (СППР). С их помощью экипаж оказывается в состоянии обеспечить уверенное, надежное и высокоэффективное пилотирование, используя обобщенную интегрированную информацию от бортовых систем, переработанную СППР в режиме реального времени.

Постановка задачи

Предлагается концепция создания СППР, базирующаяся на результатах моделирования полета самолета по заданной совокупности типовых траекторий. С этой целью в работе решаются следующие задачи:

- разработка программного комплекса для оценки топливно-временных характеристик траекторий на основе математической модели пространственного движения летательного аппарата (ЛА);
- разработка алгоритма принятия решения при выборе оптимальной опорной траектории полета;
- апробация разработанных инструментальных средств на примере самолета ТУ-204.

Современные аэронавигационные базы данных располагают большим количеством запрограммированных маршрутов полета, снабженных обширным списком аэропортов отправления и прибытия, а также характерными поворотными пунктами маршрута (ППМ). Приводятся расчетные значения времени полета и соответствующие расходы топлива. Однако при использовании таких типовых траекторий возникают следующие трудности:

- в конкретных сложившихся обстоятельствах оптимальная траектория может не принадлежать к множеству типовых;
- в силу различных причин, в первую очередь связанных с неблагоприятными погодными условиями, конечный пункт маршрута может быть изменен;
- может возникнуть ситуация, когда экипаж сам должен принимать решение об изменении конечного пункта;
- при полете по выбранной траектории необходима серьезная коррекция без изменения пункта назначения.

В описанных условиях перед экипажем встает задача оценки типовых траекторий с учетом индивидуальных характеристик самолета, а также формирования новой опорной траектории. Для решения этой задачи предлагается программный комплекс оценки типовых траекторий «Компара». Этот программный комплекс разработан в системе MATLAB и представляет собой совокупность программ и файл функций, а также пакет данных в виде текстовых файлов.

Опишем основные этапы работы с программой «Компара». Перед запуском программы необходимо выбрать условия полета – высоту (по умолчанию – 11 км) и начальную скорость (по умолчанию 94% от крейсерской). Затем в программу загружаются следующие данные: физические константы, геометрические характеристики самолета (по умолчанию – ТУ-204 [1]), характеристики силовой установки (по умолчанию – ТРДД ПС-90А [2]) и тестируемая траектория. Перечисленные значения хранятся в рабочей папке MATLAB в виде текстовых файлов.

Алгоритм работы программного комплекса «Компара».

Первоначально траектория, состоящая из m ППМов, предварительно обрабатывается следующим образом: ППМы, в которых изменение направления не происходит, редуцируются. Расчетная траектория из j ППМов ($j \leq m$) будет

представлять собой чередование прямолинейных участков, соединенных дугами (виражами) с началом в точке стандартной схемы вылета по приборам (ССВП) – ППМ₍₁₎ и окончанием в точке стандартного маршрута входа в зону аэродрома прибытия (СВАП) – ППМ_(j).

Дальнейший алгоритм работы программного комплекса «Компара» выглядит следующим образом:

- определяются параметры атмосферы, коэффициент лобового сопротивления самолета и воздушная скорость в м/с по текущему числу M ;
- вычисляется изменение скорости ($\Delta V = 0$ в установившемся прямолинейном полете);
- корректируются коэффициенты лобового сопротивления и подъемной силы, вычисляется потребная для полета тяга, по характеристике двигателя вычисляется мгновенный расход топлива;
- корректируются текущие координаты местоположения самолета, проверяется достижение точки начала маневра;
- при достижении точки начала маневра производится расчет виража.

Этот цикл повторяется $j-2$ раз, после чего рассчитывается прямолинейный участок ППМ_(j-2) – СВАП.

В основе программы лежит следующая математическая модель движения центра масс самолета. Рассматривается установившееся движение – полет на постоянной высоте со скоростью, не превышающей заданное значение на $\pm 0,05 M$. В таком случае динамика движения ЛА может быть описана системой (1).

$$\left. \begin{aligned} P &= Q + G \\ G &= Y \end{aligned} \right\} \quad (1)$$

где P – потребная тяга, Q – лобовое сопротивление, G – сила тяжести, Y – подъемная сила.

Поворот траектории в ППМе может быть реализован на вираже по дуге рассчитанного радиуса. Уравнения динамики ЛА в этом случае описываются системой (2).

$$\left. \begin{aligned} m \frac{dV}{dt} &= P_p \cdot \cos \alpha - Q \\ G &= Y \cdot \cos \gamma_c \\ Y \cdot \sin \gamma_c &= + \frac{mV^2}{r} \end{aligned} \right\} \quad (2)$$

где V – воздушная скорость, m – масса, r – расчетный радиус виража, α – угол атаки, γ_c – угол крена, P_p – располагаемая тяга.

Подробно расчет виража приведен в [3].

После выполнения соответствующих расчетов в графических окнах выводятся следующие зависимости:

- исходная и расчетная траектории;
- поляр от времени;
- скорость в м/с от времени;
- углы крена и атаки, мгновенный расход, потребная тяга, перегрузка и коэффициент подъемной силы – по запросу пользователя.

Обсуждение результатов

Для оценки эффективности разработанного программного комплекса рассмотрим следующий пример. Проведем моделирование:

- одинаковых маршрутов полета на различных высотах;

- различных траекторий с одинаковыми начальными и конечными условиями.

В качестве тестовой принята траектория протяженностью 1800 км, в составе которой имеются 2 ППМа, в которых необходимо изменение направления полета. В начальный момент времени скорость полета $M1=0,72$ и должна быть увеличена до крейсерской $M=0,77$. Проведем сравнительный анализ траекторий одинаковой формы и протяженности, пролегающих на различных высотах: 11, 8 и 4 км, начальная масса самолета равна 95 т. В табл. 1 приведены координаты, образующие траекторию, а также время и количество топлива, необходимое для достижения каждого ППМа.

Таблица 1. Результаты моделирования полета на различных высотах

	X, км	Y, км	H = 11 км		H = 8 км		H = 4 км	
			Время, с	Расход, кг	Время, с	Расход, кг	Время, с	Расход, кг
ССВП	200	0	0	0	0	0	0	0
ППМ1	600	450	2749	5004	2619	7181	2492	11507
ППМ2	1000	450	4496	8146	4301	11746	4079	18816
СВАП	1600	1000	8024	14429	7666	20805	7274	33468

Из анализа данных, приведенных в таблице, следует, что с увеличением высоты полета его длительность (За время прохождения ППМа принято время завершения маневра) несущественно увеличивается, в то время как расход топлива значительно уменьшается. Также из таблицы следует, что полет по заданной траектории невозможно выполнить на высоте 4 км, т.к. для этого требуется недопустимо большой запас топлива – более 33 т.

Теперь рассмотрим возможности оперативного изменения маршрута движения при возникновении непредвиденных факторов. С этой целью проведем сравнительный анализ возможных траекторий полета на одинаковой высоте, с одинаковыми исходными и конечными точками в условиях, когда между точками ССВП и СВАП находятся обширные опасные метеообразования, не позволяющие проложить прямолинейный маршрут. Бортовой оперативно-советующей системой предложены 3 варианта траекторий преодоления угроз, координаты ППМов сведены в табл. 2 (нулевой обозначена прямолинейная траектория, на рисунке показана пунктиром).

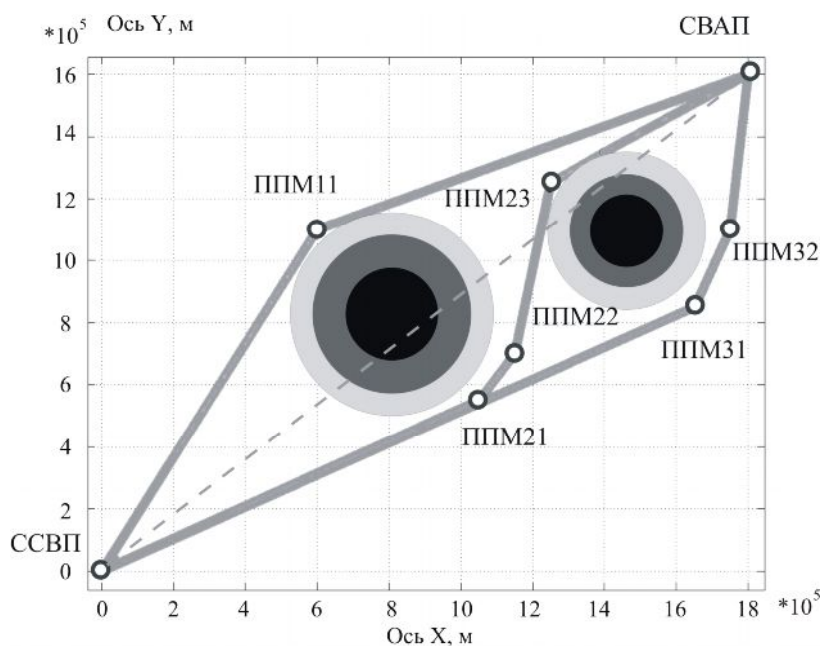


Рисунок. Варианты безопасных траекторий

Таблица 2 Варианты безопасных траекторий

H=11 км M=0,77	Траектория № 0		Траектория № 1		Траектория № 2		Траектория № 3	
	X, км	Y, км	X, км	Y, км	X, км	Y, км	X, км	Y, км
ССВП	0	0	0	0	0	0	0	0
ППМ1	–	–	600	1100	1050	550	1650	850
ППМ2	–	–	–	–	1150	700	1750	1100
ППМ3	–	–	–	–	1250	1250	–	–
СВАП	1800	1600	1800	1600	1800	1600	1800	1600

Проведем сравнительный анализ этих траекторий.

Пусть полет происходит в тех же условиях: высота – 11 км, крейсерская скорость $M=0,77$, начальная масса самолета равна 95 т. В табл. 3 приведены топливно-временные характеристики траекторий.

Таблица 3. Топливо-временные оценки траекторий

	Траектория № 0		Траектория № 1		Траектория № 2		Траектория № 3	
	Время, с	Расход, кг	Время, с	Расход, кг	Время, с	Расход, кг	Время, с	Расход, кг
ССВП	0	0	0	0	0	0	0	0
ППМ1	–	–	5605	10114	5300	95686	8271	14851
ППМ2	–	–	–	–	6085	10974	9420	16889
ППМ3	–	–	–	–	8570	15395	–	–
СВАП	10648	19039	11282	20164	11388	20368	11620	20777

За время прохождения ППМа принято время завершения маневра.

В табл. 4 приведена сравнительная характеристика тестируемых траекторий. Для наглядности значения для прямолинейной траектории приняты равными 100%.

Таблица 4. Сравнительные характеристики траекторий

№	Время		Пройденный путь		Расход топлива	
	сек	%	км	%	кг	%
0	10648	100	2408	100	19039	100
1	11282	105,95	2552	105,98	20164	105,9
2	11388	106,94	2576	106,97	20368	106,98
3	11620	109,13	2629	109,18	20777	109,13

Проанализировав таблицу, можно сделать вывод, что наименьшая длина безопасной траектории ССВП–СВАП составляет 2552 км, 105,89% от прямолинейной, следовательно, самой экономически выгодной оказалась траектория № 1. Также заметим, что выполнение маневров (виражей) требует большего расхода топлива, чем прямолинейный полет, поэтому при сравнении траекторий 1 и 2 разница во времени составляет 0,99%, а по расходу – 1,01%.

В итоге при возникновении ситуации, подобной показанной на рисунке, СППР, основанная на разработанном программном комплексе, может выдать следующие рекомендации:

- сначала обогнуть большую по площади опасность и затем проложить прямолинейный курс к пункту назначения (Траектория 1);
- попытка «проскочить» между опасными участками не имеет смысла – выигрыш по времени и расходу отсутствует (Траектория 2);
- не следует двигаться по траектории, по ходу все дальше отклоняющейся от первоначального курса (Траектория 3).

Заключение

Разработанный программный комплекс позволяет проводить оценку топливно-временных характеристик семейства опорных траекторий и решать следующие задачи:

- проводить анализ траекторий, проложенных на различных высотах;
- проводить анализ траекторий, проложенных на одной высоте, содержащих различное количество несовпадающих ППМов.

При этом программный комплекс «Компара» предоставляет пользователю следующие возможности:

- проводить анализ траекторий для широкого парка современных самолетов в диапазоне скоростей $M < 1$;
- осуществлять оценку любых маршрутов полета – как загруженных из специальных баз данных, так и заданных пользователем;
- обеспечивает наглядность и удобство представления результатов – в виде графиков или текстовых файлов.

Таким образом, программный комплекс «Компара» обеспечивает информационную поддержку СППР, предоставляя для каждой из анализируемых опорных траекторий количественные оценки их топливно-временных параметров.

В дальнейшем планируется расширение функциональных возможностей разработанного программного комплекса с целью обеспечения анализа пространственных трехмерных траекторий, а также последующая его интеграция с программным модулем оптимизации траектории полета по критерию минимума риска.

Литература

1. Ригмант В.В. ТУ-204 – Прошлое, настоящее, будущее // Аэрокосмическое обозрение. – 2006. – № 2. – С. 88–93.
2. ОАО «Авиадвигатель» – ОАО «Пермский Моторный Завод», <http://www.avid.ru/products> – Яз. рус., англ.
3. Аэромеханика полета: Динамика самолета: Учебник для авиационных вузов / Бочкарев А.С., Андреевский В.В., Белоконов В.М., и др.; Под. ред. Бочкарева А.С. и Андреевского В.В. – 2-е изд., перераб. и доп. – М.: Машиностроение. – 1985. – 360 с.
4. Ефанов В.Н. Бортовые системы управления полетом: путь к свободному воздушному пространству // «Мир Авионики». – 2000. – № 1. – С. 11–21.
5. Джандгава Г.И., Герасимов Г.И., Рогалев А.П. и др. Интеллектуальные интегрированные комплексы бортового оборудования маневренных летательных аппаратов // Приборы и системы. Управление, контроль, диагностика. – 2000. – № 8. – С. 8–14.

РАСЧЕТ КОЭФФИЦИЕНТОВ ПРОХОЖДЕНИЯ И ОТРАЖЕНИЯ В СИСТЕМЕ СВЯЗАННЫХ КВАНТОВЫХ ВОЛНОВОДОВ

А.В. Беляев, М.И. Гаврилов, А.Н. Ситников

Научный руководитель – д.ф.-м.н., профессор И.Ю. Попов

В работе произведено построение математической модели системы связанных волноводов. С ее помощью появилась возможность численно получать зависимость коэффициентов прохождения от геометрических и энергетических параметров системы.

Ключевые слова: квантовая физика, связанные волноводы, численное моделирование

Введение

Принципиально новые возможности в построении вычислительных систем открывает перед нами квантовая механика, так как сложность квантовой системы возрастает экспоненциально относительно ее размера [1].

Проблема квантовых вычислений тесно связана со сложностью физической реализации. В настоящее время существует несколько возможных элементных баз для квантового компьютера: связанные ионы, ядерный магнитный резонанс в жидкости, квантовые точки и др. [2]. Каждая из них имеет свои преимущества, но и соответствующие недостатки. Идеальной для реализации базы пока не существует, поэтому актуальна проблема исследования и разработки наноустройств для квантовых вычислений.

Целью данной работы является исследование поведения плоских слабосвязанных наноструктур, как возможной основы для реализации квантовых элементов.

В работе произведено построение математической модели системы связанных волноводов. С ее помощью появилась возможность численно получать зависимость коэффициентов прохождения от геометрических и энергетических параметров системы.

Это позволяет подобрать необходимые характеристики для требуемого в конкретных задачах поведения системы. Кроме того, данную модель можно использовать для поиска стационарных состояний волновой функции для заданной системы.

Математическая модель

Для исследования возможности построения квантовых операций при «волноводной» [3, 4] интерпретации очень важно знание коэффициентов прохождения и отражения в системе при различных значениях ее параметров. Рассмотрим систему двух волноводов, связанных через отверстия, с условиями Дирихле на границе (рис. 1).

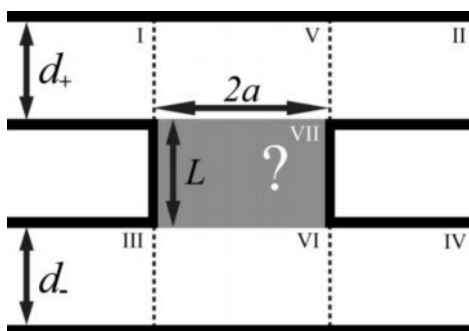


Рис. 1. Схема математической модели системы связанных волноводов

При этом допускается, что в области связи могут быть введены некоторые дополнительные условия, зависящие только от вертикальной координаты, например, потенциал, электрическое поле, и др. Пусть ширины верхнего и нижнего волноводов равны d_+ и d_- соответственно, $b = 2a$ – ширина окна, L – высота области связи (если область связи отсутствует, то $L = 0$). Положим также полную энергию электрона равной k^2 .

В силу квантовых свойств системы поперечная составляющая волновой функции в волноводе ширины d может выступать только как суперпозиция квантованного набора состояний $\sin\left(\frac{n\pi}{d}y\right)$. Воспользовавшись этим свойством, зададим в первой граничной области волновую функцию в виде [5]:

$$\Psi_I = \sum_{n=1}^{\infty} \sin\left(\frac{n\pi}{d_+}(y-L-d_-)\right) \left(a_n e^{ik_n^+ x} + b_n e^{-ik_n^+ x} \right),$$

где $k_n^+ = \sqrt{k^2 - (n\pi/d_+)^2}$. Соответствующие коэффициенты для волновых функций остальных трех граничных областей Ψ_{II} , Ψ_{III} и Ψ_{IV} положим равными c_n и d_n , g_n и h_n , p_n и q_n соответственно.

В области взаимодействия положим волновую функцию в виде:

$$\Psi_{VII} = \sum_{n=1}^{\infty} \sin\left(\frac{n\pi}{2a}x\right) \Phi(y, k'_n).$$

где $k'_n = \sqrt{k^2 - (n\pi/2a)^2}$. В областях, прилегающих к отверстию, представим ее в виде суммы продольной и поперечной составляющих

$$\Psi_V = \sum_{n=1}^{\infty} \sin\left(\frac{n\pi}{d_+}(y-L-d_-)\right) \left(e_n e^{ik_n^+ x} + r_n e^{-ik_n^+ x} \right) + \sum_{i=1}^{\infty} t_n \sin\left(\frac{n\pi}{2a}x\right) \Phi^+(y, k'_n).$$

Аналогично вводим коэффициенты v_n , u_n и w_n для волновой функции нижнего волновода в области VI . Совокупность функций $\Phi(y, k'_n)$ и $\Phi^{\pm}(y, k'_n)$ образуют базисный набор функций по которым раскладывается y -составляющая в этих областях. Так, например, в отсутствие внешних полей они будут иметь вид:

$$\Phi(y, k'_n) = f_n e^{ik'_n y} + e_n e^{-ik'_n y},$$

$$\Phi^+(y, k'_n) = \sin(k'_n(y - d_+ - L - d_-)),$$

$$\Phi^-(y, k'_n) = \sin(k'_n y).$$

Будем так же обозначать символами $\phi(y, k'_n)$ и $\phi^{\pm}(y, k'_n)$ производные по y соответствующих функций.

Проведем согласование разложения на границах областей по значениям и первым производным вдоль нормали:

$$\Psi_i = \Psi_j, \quad \frac{\partial \Psi_i}{\partial n} = \frac{\partial \Psi_j}{\partial j}.$$

Уравнения, соответствующие «вертикальным» граничным условиям, умножаем на $\sin\left(\frac{\pi n}{d_{\pm}} y\right)$ и интегрируем по длине границы. В силу ортогональности базиса синусов имеем:

$$\int_0^{d_{\pm}} \sin\left(\frac{n\pi}{d_{\pm}}\right) \sin\left(\frac{m\pi}{d_{\pm}}\right) dy = \frac{\pi}{2} \delta_{mn}.$$

Введем так же обозначение A_{nm}^{\pm} :

$$A_{nm}^+ = \frac{m\pi}{bd_+k_n^+} \int_{L+d_-}^{L+d_-+d_+} \Phi^+(y, k_m) \sin\left(\frac{n\pi}{d_+}(y-L-d_-)\right) dy,$$

$$A_{nm}^- = \frac{m\pi}{bd_-k_n^-} \int_0^{d_-} \Phi^-(y, k_m) \sin\left(\frac{n\pi}{d_-} y\right) dy.$$

Таким образом, во введенной системе обозначений, получаем систему линейных уравнений. Решение дает значение выходных параметров b_m , c_m , h_m и p_m относительно входных a_m , d_m , g_m и q_m :

$$\begin{aligned} b_n &= d_n + i \sum_{m=1}^{\infty} A_{nm}^+ \left(1 - (-1)^m e^{ik_n^+ b}\right) t_n, & c_n &= a_n + i \sum_{m=1}^{\infty} A_{nm}^+ \left(1 - (-1)^m e^{-ik_n^+ b}\right) t_n, \\ h_n &= q_n + i \sum_{m=1}^{\infty} A_{nm}^- \left(1 - (-1)^m e^{ik_n^- b}\right) w_n, & p_n &= g_n + i \sum_{m=1}^{\infty} A_{nm}^- \left(1 - (-1)^m e^{-ik_n^- b}\right) w_n. \end{aligned}$$

Сами значения t_n и w_n определяются из «горизонтальной» системы связи в районе окна:

$$\begin{cases} t_n \Phi^+(L+d_-, k'_n) &= \Phi(L+d_-, k'_n), \\ w_n \Phi^-(d_-, k'_n) &= \Phi(d_-, k'_n), \\ \phi(L+d_-, k'_n) &= i \sum_{l=1}^{\infty} C_{nl}^+ t_l + X_n + t_n \phi^+(L+d_-, k'_n), \\ \phi(d_-, k'_n) &= i \sum_{l=1}^{\infty} C_{nl}^- w_l + Y_n + w_n \phi^-(d_-, k'_n), \end{cases} \quad (1)$$

где

$$C_{nl}^{\pm} = 2(1+(-1)^{n+l}) \sum_{m=1}^{\infty} B_{nm}^{\pm} A_{ml}^{\pm} (1-(-1)^n e^{ik_m^{\pm} b}), \quad B_{nm}^{\pm} = \frac{\pi^2 m n}{d_{\pm} ((n\pi)^2 - (bk_n)^2)},$$

$$X_n = \sum_{m=1}^{\infty} 2B_{nm}^+ (a_m (1-(-1)^n e^{ik_m^+ b}) + d_m (1-(-1)^n e^{-ik_m^+ b})),$$

$$Y_n = \sum_{m=1}^{\infty} 2(-1)^m B_{nm}^- (g_m (1-(-1)^n e^{ik_m^- b}) + d_m (1-(-1)^n e^{-ik_m^- b})).$$

Заметим также, что в случае $L=0$ данная система преобразуется к более простому виду:

$$\begin{cases} t_n \Phi^+(d_-, k'_n) & = w_n \Phi^-(d_-, k'_n), \\ i \sum_{l=1}^{\infty} C_{nl}^+ t_l + X_n + t_n \phi^+(d_-, k'_n) & = i \sum_{l=1}^{\infty} C_{nl}^- w_l + Y_n + w_n \phi^-(d_-, k'_n). \end{cases}$$

Для численных вычислений бесконечные суммы ограничиваются некоторой константой N , в рамках которой и производятся вычисления. Матрица, соответствующая исходной системе уравнений в силу большего размера имеет гораздо более плохую обусловленность, что негативно сказывается при численном решении. Плохая обусловленность вызвана волновыми функциями, соответствующим комплексным значениям энергии k_n^{\pm} и k'_n . Скорее всего, при аналитическом решении бесконечной системы, соответствующие коэффициенты системы должны обращаться в ноль.

Заметим, что после построения системы появляется возможность решения ряда задач.

Во-первых, появляется возможность решить задачу рассеяния. Для этого положим $a_n = \delta_{n1}$, а d_n , g_n и q_n тождественно равными нулю. Решив систему и найдя выходные параметры b_n , c_n , h_n и p_n , коэффициенты прохождения определяются исходя из формул:

$$\begin{aligned} T_{12} &= Re \sum_{n=1}^{\infty} |c_n|^2 k_n^+ / k_1^+, & T_{11} &= Re \sum_{n=1}^{\infty} |b_n|^2 k_n^+ / k_1^+, \\ T_{14} &= Re \sum_{n=1}^{\infty} |p_n|^2 k_n^- / k_1^-, & T_{13} &= Re \sum_{n=1}^{\infty} |h_n|^2 k_n^- / k_1^-. \end{aligned}$$

Во-вторых, облегчается поиск связанных состояний. Для этого некоторый выбранный параметр (например, L или k) варьируется в некоторых пределах для достижения нулевого значения определителя матрицы (1). В этом случае система при нулевых векторах начальных условий имеет ненулевое решение, которое и определяет соответствующую собственную функцию.

Анализ системы без внешнего электрического поля

В случае отсутствия внешних полей в области окна, разумно рассматривать набор функций $\Phi^{\pm}(y, k_m)$ как граничные составляющие собственных функций потенциальной ямы шириной $(d_+ + L + d_-)$:

$$\begin{aligned} \Phi^+(y, k_n) &= \sin(k_n(y - d_+ - L - d_-)), \\ \Phi^-(y, k_n) &= \sin(k_n y). \end{aligned}$$

Соответствующие константы A_{nm}^{\pm} :

$$\begin{aligned} A_{nm}^+ &= -\frac{mn\pi^2 \sin(k'_m d_+)}{bk_n^+ ((n\pi)^2 - (d_+ k'_m)^2)}, \\ A_{nm}^- &= -(-1)^n \frac{mn\pi^2 \sin(k'_m d_-)}{bk_n^- ((n\pi)^2 - (d_- k'_m)^2)}. \end{aligned}$$

Пусть, в секторе VII будет потенциальный барьер высотой V , тогда функция Φ будет иметь вид:

$$\Phi(y, k'_n) = f_n e^{ik''_n y} + e_n e^{-ik''_n y},$$

$$\phi(y, k'_n) = ik''_n (e^{ik''_n y} - e_n e^{-ik''_n y}),$$

где $k''_n = \sqrt{(k'_n)^2 - V} = \sqrt{k_n^2 - (n\pi/b)^2 - V}$.

Перепишем систему (1), используя выбранную систему функций:

$$\begin{cases} i \sum_{l=1}^{\infty} C_{nl}^+ t_l + X_n + k'_n t_n \cos(k'_n d_+) + k''_n \frac{\sin(k''_n d_-)}{\sin(k''_n L)} (t_n \cos(k''_n L) + w_n) = 0, \\ i \sum_{l=1}^{\infty} C_{nl}^- w_l + Y_n + k'_n w_n \cos(k'_n d_-) + k''_n \frac{\sin(k''_n d_-)}{\sin(k''_n L)} (t_n + w_n \cos(k''_n L)) = 0. \end{cases}$$

Пусть $R_n = k'_n \cos(k'_n d_-) + k''_n (k''_n L) \sin(k''_n d_-)$.

В случае $d_+ = d_- = d$ (соответственно, $C_{nl}^+ = C_{nl}^- = C_{nl}$) система допускает разделение переменных. Обозначим $U_n = t_n + w_n$ и $V_n = t_n - w_n$, тогда:

$$\begin{cases} i \sum_{l=1}^{\infty} C_{nl} U_n + (X_n + Y_n) + R_n U_n + k''_n \frac{\sin(k''_n d)}{\sin(k''_n L)} U_n = 0, \\ i \sum_{l=1}^{\infty} C_{nl} V_l + (X_n - Y_n) + R_n V_n - k''_n \frac{\sin(k''_n d)}{\sin(k''_n L)} V_n = 0. \end{cases} \quad (2)$$

Численное моделирование

Было проведено численное моделирование для случая отсутствия внешних полей. Даже в таком упрощенном случае в зоне взаимодействия, функции коэффициентов прохождения в зависимости от параметров системы являются сложными и мало предсказуемыми. Само моделирование происходило на основе решения системы (2).

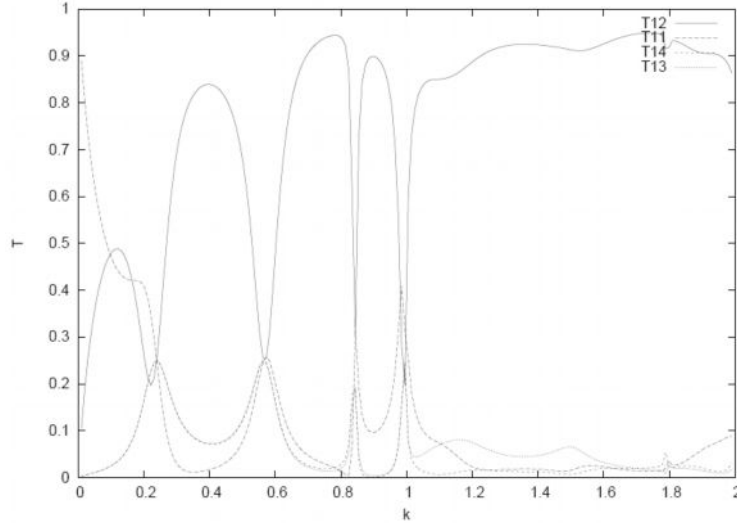


Рис. 2. Зависимость коэффициентов прохождения от энергии для системы с $V = 0$, $L = d = b$

Варьируя параметры системы можно получить ярко выраженный резонанс (рис. 3). Данный результат хорошо согласуется с аналитическими результатами, полученными в [6, 7].

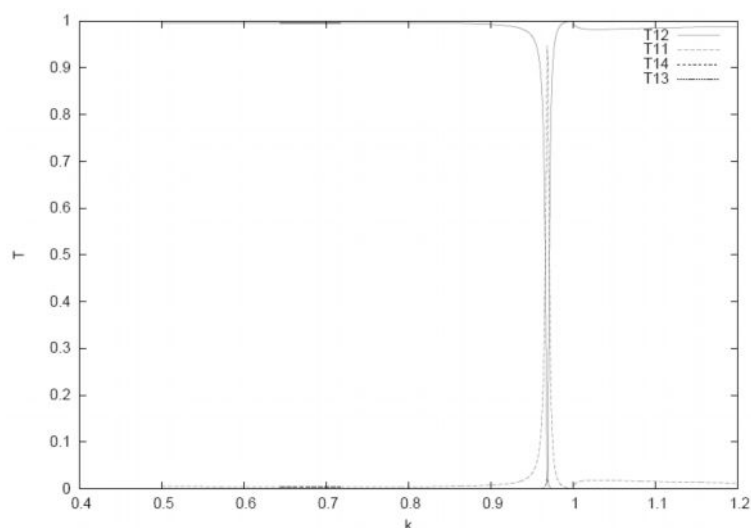


Рис. 3. Эффект полного отражения в системе с $V = 2.8$, $L = d$, $b = 0.4d$

Заключение

В работе произведено построение математической модели системы связанных волноводов. С ее помощью появилась возможность численно получать зависимость коэффициентов прохождения от геометрических и энергетических параметров системы. Это позволяет подобрать необходимые характеристики для требуемого в конкретных задачах поведения системы. Кроме того, данную модель можно использовать для поиска стационарных состояний волновой функции для заданной системы.

Литература

1. Фейнман Р. Моделирование физики на компьютерах. // Квантовый компьютер и квантовые вычисления. – Т. II. – Ижевск. – 1999. – С. 96.
2. Валиев К.А., Кокин А.А. Квантовые компьютеры: надежды и реальность. М.: R&C Dynamics. – 2001. – 358 с.
3. Gavrilov M.I., Gortinskaya L.V., Pestov A.A., Popov I.Yu., Tesovskaya E.S. Quantum Algorithms Implementation Using Quantum Wires System. // Proceedings of the ICO Topical Meeting on Optoinformatics Information Photonics 2006, September 4–7, 2006, SPb, Russia. – PP. 327–329.
4. Popov I.Yu., Gortinskaya L.V., Gavrilov M.I., Pestov A.A., Tesovskaya E.S. Weakly coupled quantum wires and layers as an element of quantum computer. // Письма в ЭЧАЯ. – 2007. – Т.4. – №2(138). – С. 237–243.
5. Abramowitz M., Stegun I.A. Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. 10th printing, Dover Publications, Inc, New York. – 1972.
6. Exner P. and Seba P. Bound states in curved quantum waveguides // J. Math. Phys. 1989. – V. 30 (11). – PP. 2574–2580 14.
7. Gortinskaya L.V., Popov I.Yu., Tesovskaya E.S. // Proc. of Intern. Seminar @Day on Diffraction' 2003. St. Petersburg. – 2003. – 52 p.

РЕШЕНИЕ МАТРИЧНЫХ ИГР В НЕЧЕТКИХ СМЕШАННЫХ СТРАТЕГИЯХ

П.П. Петгай

Научный руководитель – д.ф.-м.н., профессор И.Ю. Попов

В статье приводится описание алгоритма нахождения оптимальных смешанных стратегий в антагонистических играх. Обосновывается неэффективность данного метода для одиночных игр и игр, в которых мы обладаем какой-либо дополнительной информацией о психологии поведения противника, которая по своей природе имеет нечеткий характер. Предложены новые правила выбора нечетких смешанных стратегий для антагонистических игр с нечеткой информацией. Рассмотрен расчетный пример.

Ключевые слова: теория игр, теория нечетких множеств, математические методы в экономике

Постановка задачи и классическое решение

Не многие исследования в области математики удостоены присуждения Нобелевской премии. Дело в том, что за абстрактные математические результаты Нобелевская премия не присуждается, но можно получить премию, если прикладные исследования будут признаны специалистами из других областей: физики, экономики, химии или медицины. Именно так произошло с теорией игр. Ее методы нашли многочисленные применения в военных науках, биологии, психологии, социологии, в области информационных технологий и, конечно же, в области экономики. Достаточно лишь сказать, что за 40 лет присуждения Нобелевских премий по экономике 5 раз премии присуждались за достижения в области теории игр.

В 1994 году Нобелевская премия по экономике была присуждена Джону Нэшу «за анализ равновесия в теории некоалиционных игр». До выхода в свет диссертации Джона Нэша исследования проводились основном в области игр с нулевой суммой, где суммарный выигрыш одних участников равен суммарному проигрышу других. Нэш исследовал игры с ненулевой суммой, в которых возможны ситуации, выгодные сразу для всех участников. Так, например, в случае спора руководства завода и профсоюза о повышении заработной платы он может закончиться забастовкой, что не выгодно сразу обоим участникам спора, тогда как от определенного соглашения все участники конфликта могут в общей сложности остаться в выигрыше. Нэш выделил специальный класс ситуаций в игре (т.е. фиксированных стратегий всех участников игры) – такие ситуации, от которых не выгодно отклоняться *в одиночку* никому из игроков. Позже такие ситуации были названы ситуациями *равновесия по Нэшу*.

Ситуации равновесия могут существовать и в *антагонистических* играх – играх двух игроков, в которых величина выигрыша одного численно равна величине проигрыша другого. Такие игры удобно представлять в форме матриц, строкам которых соответствуют стратегии 1-ого игрока, столбцам – стратегии 2-ого, а элементы матрицы $a_{i,j}$ показывают величину выигрыша 1-ого игрока (и, соответственно, проигрыша 2-ого), если 1-ый игрок выберет стратегию i , а 2-ой игрок – стратегию j . При выборе стратегии i гарантированный доход 1-ого игрока составит $\min_j a_{i,j}$, независимо от стратегии 2-ого игрока. Естественно стремиться максимизировать свой минимально возможный доход и выбрать стратегию i : $\max_i (\min_j a_{i,j})$ – это принцип выбора *максиминной* стратегии. Похожие рассуждения можно применить и к действиям 2-ого игрока. При выборе стратегии j его максимально возможный проигрыш составит

$\max_i a_{i,j}$. Естественно выбрать такую стратегию, которая минимизировала бы максимально возможный проигрыш, т.е. выбрать стратегию j : $\min_j (\max_i a_{i,j})$ – выбрать *минимаксную* стратегию поведения. Можно легко доказать, что верно следующее неравенство: $\max_i (\min_j a_{i,j}) \leq \min_j (\max_i a_{i,j})$ [1]. При этом, если достигается равенство, то соответствующая ему ситуация (i^*, j^*) будет *равновесной*, т.е. никому из игроков будет не выгодно отклоняться от нее в одиночку. Действительно, чтобы ситуация (i^*, j^*) была равновесной необходимо и достаточно, чтобы элемент a_{i^*,j^*} был максимальным в столбце и минимальным в строке. Как не сложно сообразить, в этом случае этот же элемент будет соответствовать гарантированному выигрышу 1-ого игрока при максиминной стратегии и гарантированному проигрышу 2-ого игрока при минимаксной стратегии, т.е. они совпадут. При этом совпасть они могут только для равновесных стратегий. В этом случае ситуация называется *решением игры*.

Тем не менее, на практике ситуаций равновесия может оказаться несколько, что не дает возможности однозначно предсказать исход игры для случая, когда игроки выбирают максиминную и минимаксную стратегии соответственно. Что еще хуже, если равенства нет, то ситуации равновесия в игре не существует! Для решения указанной проблемы был предложен метод решения матричных игр в *смешанных стратегиях*. Данный метод удобен для *многократно повторяющихся* игр. В основе метода лежит идея *рандомизации* выбора стратегий игроками [2].

По определению, *смешанной стратегией* 1-ого игрока называется набор неотрицательных чисел $\{x_1, x_2, \dots, x_m\}$, удовлетворяющий условию $\sum_{i=1}^m x_i = 1$. В этом случае x_i интерпретируется как вероятность применения 1-ым игроком i -ой стратегии. Аналогично, набор неотрицательных чисел $\{y_1, y_2, \dots, y_n\}$, удовлетворяющий условию

$\sum_{j=1}^n y_j = 1$ является смешанной стратегией 2-ого игрока, y_j интерпретируется как

вероятность выбора 2-ым игроком j -ой стратегии. Если задать смешанные стратегии игроков, то математическое ожидание выигрыша 1-ого игрока составит

$F(x, y) = \sum_{i=1}^m \sum_{j=1}^n a_{i,j} x_i y_j$. Если 1-ый игрок по-прежнему будет руководствоваться

максиминными соображениями, то ему необходимо выбрать такое распределение вероятностей своих стратегий (т.е. такой набор чисел $\{x_1, x_2, \dots, x_m\}$), при котором он

сможет максимизировать наименьший ожидаемый выигрыш $\max_{x \in X} \left[\min_{y \in Y} \sum_{i=1}^m \sum_{j=1}^n a_{i,j} x_i y_j \right]$,

где $X = (x_1, x_2, \dots, x_m)$, $Y = (y_1, y_2, \dots, y_n)$. Аналогично, руководствуясь минимаксной стратегией, 2-ой игрок изберет такое распределение вероятностей, чтобы

минимизировать наибольший ожидаемый проигрыш: $\min_{y \in Y} \left[\max_{x \in X} \sum_{i=1}^m \sum_{j=1}^n a_{i,j} x_i y_j \right]$. В этом

случае *оптимальными смешанными стратегиями* игроков (x^*, y^*) будут называться такие стратегии, при которых достигается ситуация *равновесия* в игре, т.е. равенство

$\max_{x \in X} \left[\min_{y \in Y} \sum_{i=1}^m \sum_{j=1}^n a_{i,j} x_i y_j \right] = \min_{y \in Y} \left[\max_{x \in X} \sum_{i=1}^m \sum_{j=1}^n a_{i,j} x_i y_j \right]$. Джон фон Нейман доказал

фундаментальную теорему теории матричных игр, согласно которой любая игра имеет решение (т.е. ситуацию равновесия) в смешанных стратегиях.

Существуют различные методы нахождения решений матричных игр в смешанных стратегиях, одним из наиболее простых из которых является переход к решению эквивалентной задачи линейного программирования $f_1(x) = x_{m+1} \rightarrow \max$ при

ограничениях $D_1 = \{x \in R^{m+1} \mid \sum_{i=1}^m a_{i,j} x_i \geq x_{m+1} \forall j \in \overline{1, n}; \sum_{i=1}^m x_i = 1; x_i \geq 0, i \in \overline{1, m}\}$ для 1-ого игрока.

Для 2-ого игрока получим, соответственно, задачу $f_2(y) = y_{n+1} \rightarrow \min$ при ограничениях $D_2 = \{y \in R^{n+1} \mid \sum_{j=1}^n a_{i,j} y_j \leq y_{n+1} \forall i \in \overline{1, m}; \sum_{j=1}^n y_j = 1; y_j \geq 0, j \in \overline{1, n}\}$. При этом,

не сложно заметить, что задачи 1-ого и 2-ого игроков являются двойственными друг другу, а значит, оптимальные значения целевых функций этих задач совпадают. Отсюда по сути уже следует существование решения в смешанных стратегиях.

Критика метода смешанных стратегий и переход к нечеткости

Как было показано ранее, в чистых стратегиях в игре может не существовать ситуации равновесия.

Ориентация на оптимальные смешанные стратегии позволяет первому игроку максимизировать наименьший ожидаемый выигрыш 1-ого игрока и минимизировать наибольший ожидаемый проигрыш 2-ого игрока. Однако такой выбор может быть оправдан лишь для многократно повторяющихся игр. В отдельной игре каждый из игроков даже при реализации «оптимальной» смешанной стратегии может очень сильно проиграть! В отдельных играх смешанные стратегии не только не обещают максимального выигрыша, но даже не защищают от максимально возможного проигрыша, т.е. каждый игрок может принять худшее из всех возможных решений!

Выбор максиминных и минимаксных стратегий хорошо подходит для осторожных игроков. Он позволяет не проиграть слишком много, не выиграть слишком мало. Но некоторые игроки могут быть более склонны к риску, и ориентироваться на те стратегии, которые могут обеспечить им максимально возможный выигрыш, некоторые стараются максимизировать средний ожидаемый выигрыш, т.е. выбор минимакса или максимина не является единственно возможным хорошим в некотором смысле решением.

Наконец, ориентация на смешанные стратегии подразумевает отсутствие какой-либо информации о психологии соперника, предполагая лишь, что он действует рационально. Однако, как заметил Герберт Саймон в своей теории ограниченной рациональности, большинство людей рациональны только отчасти и эмоциональны либо иррациональны в остальных ситуациях. Выбор рационального решения нередко подразумевает какие-либо сложные вычисления, которыми игроки могут не владеть или же они могут не иметь возможности их осуществить (из-за трудоемкости вычислений, отсутствия всей необходимой информации или каких-либо еще причин). Возможно также, что затраты на принятие оптимального решения могут быть недопустимо высоки и не окупят возможных преимуществ, поэтому рационально в такой ситуации будет принять какое-либо другое «достаточно хорошее» решение.

На практике мы можем иметь о противнике какую-либо дополнительную информацию и на ее основе делать какие-либо предположения относительно его поведения. При этом наши предположения практически всегда носят нечеткий характер. К примеру, наши предположения могут иметь такой вид: «если противник

самоуверен, то, скорей всего, он выберет стратегию максимизации максимального дохода».

Оперировать с подобными высказываниями и принимать на их основе количественное рациональное решение помогает аппарат созданной Лотфи Заде теории нечетких множеств. В отличие от классической теории множеств, где каждый элемент может либо принадлежать, либо не принадлежать какому-либо множеству, в теории нечетких множеств предполагается, что любой элемент x принадлежит множеству A с некоторой степенью принадлежности $\mu_A(x) \in [0,1]$. Пара $(A, \mu_A(x))$ образует нечеткое множество. Несмотря на кажущееся сходство с вероятностной мерой, функция принадлежности по своей природе имеет совершенно другой характер. Так, если бы мы с вероятностью 0,7 предполагали, что наш противник склонен к риску, то это означало бы лишь, что из 100 встречавшихся нам похожих в каком-либо смысле противников примерно 70 были склонны к риску. Такие данные хороши в условиях полной неопределенности относительно нашего противника и при наличии необходимой статистики. Степень принадлежности равная 0,7 к множеству противников склонных к риску можно интерпретировать примерно, как что противник скорей склонен к риску, чем не склонен, но уверенность в этом не полная. Такая уверенность хоть и субъективна, но зато не предполагает наличия каких-либо статистических данных, учитывает конкретного противника и, так или иначе, формализует всю совокупность наших знаний о нем и нашего опыта. Если мы играем с ребенком или домохозяйкой, или же мы играем с каким-либо ученым, опытным игроком, специалистом в данной области, то такие знания глупо не учитывать, хотя они, конечно же, не гарантируют правильность наших предположений. Часто мы, так или иначе, знаем нашего противника и можем, хотя бы на интуитивном уровне, предполагать, чем он будет руководствоваться при принятии решения.

В общем случае наши предположения могут носить достаточно сложный характер. К примеру, мы можем предполагать, что «если противник *опытный* и *ведет себя самоуверенно*, то он будет руководствоваться принципом оптимальных смешанных стратегий, если выпавшая стратегия будет *не слишком плохой*». Если противник *осторожный*, то он будет руководствоваться максиминной стратегией. Если противник *склонен к риску* и *не опытен*, то он выберет стратегию, соответствующую максимально возможному доходу. Таким образом, мы можем составить целую базу нечетких правил. При этом нам необходимо эвристически или аналитически задать степени принадлежности противника к каждому из нечетких множеств. Также необходимо научиться осуществлять логические операции над нечеткими множествами. Так, в первом примере нам необходимо обработать целых три нечетких множества: $A = \{\text{опытный противник}\}$, $B = \{\text{самоуверенный противник}\}$ и $C = \{\text{не слишком плохая стратегия}\}$ и на основе их обработки осуществить процедуру нечеткого вывода. Методам обработки подобных высказываний посвящен следующий раздел данной статьи.

Процедура обработки нечетких правил в антагонистических играх

Определим классическим образом основные логические операции над нечеткими множествами [3–5].

- Объединение (соответствует логической операции «или»):

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)].$$

- Пересечение (соответствует логической операции «и»):

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)].$$

- Дополнение (соответствует логической операции «не»): $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$.

Тогда обработку нечетких правил вида: $\{ \Pi_i: \text{если } (x \text{ есть } A_i) \text{ и } (y \text{ есть } B_i), \text{ то } z_i \}$, где A_i, B_i – некоторые нечеткие множества, а z_i – конкретное решение можно произвести следующим образом.

Сначала оценим степень нашей уверенности в результате. Для конкретного противника определяем степень истинности для предпосылок каждого правила $\mu_{A_i}(x_0), \mu_{B_i}(y_0)$. Для нашего примера, уверенность в том, что противник выберет стратегию z_i составит $\alpha_i = \min[\mu_{A_i}(x_0), \mu_{B_i}(y_0)]$, т.к. использовалась операция дизъюнкции, т.е. логического пересечения двух высказываний.

Затем определяем уверенность α_i для каждой из возможных стратегий. Если мы считаем, что противник выберет стратегию z_i , то оптимальным решением по этой стратегии для него будет некоторое решение j^* . Нашим оптимальным ответом будет выбор решения $i^* : \max_i a_{i,j^*}$.

Теперь мы можем воспользоваться *модифицированным вариантом смешанных стратегий*, для выбора оптимальной стратегии, основанной на обработке базы нечетких правил. Определим вероятности выбора каждого из допустимых решений как

$$p_i = \frac{\alpha_i}{\sum_i \alpha_i}.$$

Теперь мы можем считать, что с вероятностью p_i противник будет руководствоваться стратегией z_i , согласно которой он выберет решение j^* , на которое мы с этой же вероятностью ответим решением $i^* : \max_i a_{i,j^*}$.

Расчетный пример

Продемонстрируем предложенную идею на конкретном примере.

Предположим, что как нам, так и нашему сопернику известна следующая матрица

$$\text{игры: } M = \begin{bmatrix} 2 & -3 & 4 & -5 \\ -3 & 4 & -5 & 6 \\ 4 & -5 & 6 & -7 \\ -5 & 6 & -7 & 8 \end{bmatrix}.$$

Из матрицы видно, что как мы, так и наш противник должны выбрать одно из 4 решений. В каждой строке матрицы есть как положительные, так и отрицательные выигрыши, соответственно, данная игра может быть как выигрышной, так и проигрышной для каждого из игроков. Если мы попытаемся руководствоваться максиминной стратегией, то выберем решение 1 или решение 2 (т.к. в этом случае мы проиграем не больше 5 у.е., а при выборе решения 3 или 4 был риск проиграть 7 у.е.). Руководствуясь минимаксной стратегией, наш противник выберет решение 1 (т.к. в этом случае он проиграет не более 4 у.е., тогда как при выборе другого решения для него был риск проиграть больше). $5 \neq 4$, а значит ситуации равновесия по Нэшу в данной игре нет. Действительно, при *таком* выборе возможна реализация одной из двух ситуаций: или мы выиграем 2 у.е., или проиграем 3. Однако, если бы мы точно знали, что противник будет руководствоваться минимаксной стратегией, то выбрали бы решение 3, что гарантировала бы нам выигрыш 4 у.е., т.е. указанные ситуации не являются равновесными, т.к. отклоняться от них выгодно. Итак, в чистых стратегиях указанная игра ситуаций равновесия не имеет.

Предположим для примера, что мы сформулировали следующую базу из 3-х нечетких правил:

П₁: {Если противник *неопытный* или *склонен к риску*, то он выберет решение по принципу максимизации максимума возможного дохода}.

П₂: {Если противник *осторожный*, то он выберет решение по минимаксному принципу}.

П₃: {Если противник *опытный*, то он выберет решение по минимаксному принципу, если средний выигрыш при таком решении будет *не слишком низким* и решение, максимизирующее средний выигрыш *в противном случае*}.

Заметим, что данная база, конечно же, не претендует на полноту. Однако, что есть, то есть. Если мы считаем необходимым, то можем дополнить правила.

Зададим соответствующие нечеткие множества:

A={опытный противник}. Мы можем считать, например, что, если противник играет в эту игру не более чем в 3-ий раз, то он точно неопытный, если более чем в 20-ый – точно опытный. Промежуточные результаты мы можем, например, аппроксимировать с помощью прямой, задав функцию принадлежности следующим образом:

$$\mu_A(x) = \begin{cases} 0, & x \leq 3 \\ \frac{x-3}{21-3}, & 3 < x \leq 20. \\ 1, & x > 20 \end{cases}$$

B={осторожный противник}. Формализовать конкретно выбор функции принадлежности здесь достаточно сложно. Но можно исходить при оценке из наших интуитивных (экспертных) соображений. К примеру, если мы точно уверены, что противник осторожный, то задать $\mu_B(y_0) = 1$, если к осторожным мы его причислить никак не можем, то задать $\mu_B(y_0) = 0$, если сложно сказать, т.е. данных, на основе которых мы можем делать хоть какие-то предположения у нас совершенно недостаточно, то задать $\mu_B(y_0) = 0.5$ и т.д.

C={противник склонный к риску}. Заметим, что противники склонные к риску неосторожны и наоборот, т.е. можно считать, что $C = \bar{B}$ (дополнение к множеству B).

D={средний выигрыш слишком низкий}. К примеру, мы можем считать, что средний выигрыш при решении точно слишком низкий, если он отрицательный, и точно не слишком низкий, если составляет более 2/3 максимально возможного выигрыша. Для простоты снова аппроксимируем промежуточные значения с помощью прямой, и зададим соответствующую функцию принадлежности следующим образом:

$$\mu_D(z) = \begin{cases} 1, & z \leq 0 \\ \frac{\frac{2}{3}z_{\max} - z}{\frac{2}{3}z_{\max} - 0}, & 0 < z < \frac{2}{3}z_{\max} \\ 0, & z \geq \frac{2}{3}z_{\max} \end{cases}$$

Рассчитаем значения степени принадлежности нашего противника к каждому из указанных нечетких множеств. Пусть мы знаем, что противник играет в 9-ый раз. Тогда

$\mu_A(x_0) = \frac{9-3}{21-3} = \frac{1}{3}$. Пусть мы предполагаем, что противник скорее осторожный, чем

нет, но точно не уверены. Тогда предположим, что $\mu_B(y_0) = 0.7$. Наконец, максимально возможный выигрыш противника в нашей игре равен $z_{\max} = 7$. Если он выберет минимаксную стратегию, то, согласно этой стратегии, как было замечено ранее, оптимальным для него будет решение 1. В этом случае его средний выигрыш составит

$$z = -\frac{2 + (-3) + 4 + (-5)}{4} = \frac{1}{2}, \text{ следовательно, } \mu_D(z) = \frac{\frac{2}{3} \cdot 7 - \frac{1}{2}}{\frac{2}{3} \cdot 7 - 0} = \frac{25}{28}$$

Теперь рассчитаем степень уверенности в каждом из решений противника.

Для *максимизации максимума дохода* будем иметь: $\alpha_1 = \max[1 - \frac{1}{3}, 1 - 0.7] = \frac{2}{3} \approx 0.67$. Мы взяли функцию максимума, т.к. «или», $1 - \frac{1}{3}$, т.к. «неопытный» - дополнение к опытному, $1 - 0.7$, т.к. «склонный к риску» – дополнение к «осторожному».

Аналогично, для *минимаксного* принципа по правилу P_2 : $\alpha_2 = 0.7$ (т.к. принадлежность к множеству осторожных противников мы уже определили).

Для *минимаксного* принципа по правилу P_3 : $\alpha_3 = \min[\frac{1}{3}, 1 - \frac{25}{28}] = \frac{3}{28}$. Взяли минимум, т.к. противник должен быть опытным «и» выигрыш должен быть не слишком низкий, $1 - \frac{25}{28}$, т.к. «не слишком низкий» - дополнение к «слишком низкий».

Заметим, что в случае *не очень низкого* среднего выигрыша по минимаксному принципу *опытный* противник выберет, как и во втором правиле, минимаксное решение. Таким образом, минимаксное решение принимается в одном из двух случаев, устраивает любой, а значит, здесь логика «или», следовательно, степень уверенности в минимаксном решении должна быть рассчитана как

$$\beta_2 = \max[\alpha_2, \alpha_3] = \max[0.7, \frac{3}{28}] = 0.7$$

Наконец, для принципа *максимизации среднего выигрыша* мы будем иметь: $\alpha_4 = \min[\frac{1}{3}, \frac{25}{28}] = \frac{1}{3} \approx 0.33$.

Не следует удивляться тому, что сумма полученных уверенностей для всех правил превышает 1, ведь указанные множества *не* являются взаимоисключающими. К примеру, противник может быть одновременно и опытным, и осторожным, или, напротив, быть при этом склонным к риску (скорей всего, разумному). Однако для подсчета соответствующих вероятностей нам необходимо провести нормировку.

$p_1 = \frac{\alpha_1}{\alpha_1 + \beta_2 + \alpha_4} = \frac{0.67}{0.67 + 0.7 + 0.33} \approx \frac{2}{5}$ – оценка вероятности выбора противником стратегии максимизации максимума дохода. Аналогично, $p_2 = \frac{\beta_2}{\alpha_1 + \beta_2 + \alpha_4} \approx \frac{2}{5}$ – вероятность выбора минимаксной стратегии, $p_3 = \frac{\alpha_4}{\alpha_1 + \beta_2 + \alpha_4} \approx \frac{1}{5}$ – вероятность выбора стратегии максимизации среднего выигрыша.

По принципу *максимизации максимума* доходов противник выберет решение 3 или 4, т.к. в этом случае у него есть шанс получить 7 у.е., а при любом другом решении он выиграет не более 5 у.е. Тогда, каждый из этих выборов будем считать

равновероятным с $p_4 = \frac{1}{5}$. Если он выберет решение 3, то нашим оптимальным ответом будет *решение 3*. Если он выберет стратегию 4, то нашим оптимальным ответом будет *решение 4* (совпадение случайно!). По *минимаксному* принципу противник выберет решение 1, нашим оптимальным ответом будет *решение 3*. Наконец, по принципу *максимизации среднего выигрыша* (сумма значений в столбце, взятая с обратным знаком) противник выберет решение 1 или решение 3. Их также можно считать равновероятными с вероятностями $p_5 = \frac{1}{10}$, но нашим оптимальным ответом в любом случае здесь будет *решение 3*.

Просуммируем соответствующие вероятности для каждого из наших оптимальных ответов. Решение 3 мы должны принять с вероятностью $p_6 = \frac{1}{5} + \frac{2}{5} + \frac{1}{5} = \frac{4}{5}$ и решение 4 с вероятностью $p_7 = \frac{1}{5}$. Это и есть оптимальные смешанные стратегии. На практике их можно реализовать, к примеру, по методу *пропорциональной рулетки*: разбить рулетку на 5 одинаковых секторов, 1 закрасить черным цветом, 4 – белым. Если стрелка укажет на белый сектор, то выбрать решение 3, если на черный – решение 4.

Заключение

В статье обоснована неэффективность классического метода смешанных стратегий для одиночных игр и игр, в которых мы обладаем какой-либо дополнительной информацией о психологии поведения противника. Подобная информация практически всегда носит нечеткий характер. Для антагонистических игр с нечеткой информацией предложены правила выбора оптимальных смешанных нечетких стратегий. За рамками рассмотрения статьи остались методы принятия решений для неантагонистических игр с нечеткой информацией, игр с нечеткими выигрышами, игр с несчетным нечетким множеством возможных решений. В дальнейшем планируется провести исследование указанных задач.

Литература

1. Петросян Л.А. и др. Теория игр. – М.: Высшая школа, Книжный дом «Университете». – 1998. – 304 с.
2. Конюховский П.В. Математические методы исследования операций в экономике. – СПб: Изд-во Санкт-Петербургского университета. – 2008. – 395 с.
3. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб: БХВ-Петербург. – 2005. – 736 с.
4. Кричевский М.Л. Интеллектуальные методы в менеджменте. – СПб: Питер. – 2005. – 304 с.
5. Штовба С.Д. Проектирование нечетких систем средствами MATLAB. – М.: Горячая линия – Телеком. – 2007. – 288 с.

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ В ЗАДАЧАХ БИОЛОГИЧЕСКОЙ ЭВОЛЮЦИИ

А.Р. Ямалетдинова

Научный руководитель – д.ф.-м.н., профессор Г.П. Мирошниченко

В статье рассматривается модифицированная модель Лотки-Вольтера (применяемая биологами для изучения взаимодействия популяций) с применением генетического алгоритма. В классической модели Лотки-Вольтера с помощью дифференциальных уравнений учитывается естественная смертность и смертность за счет внутривидовой конкуренции. В модифицированной модели процессы рождения моделируются с помощью генетических операторов – 1-точечный кроссовер и битовая мутация. Сравниваются результаты работы классической и модифицированной модели Лотки-Вольтера.

Ключевые слова: генетический алгоритм, ген, хромосома, локус, функция приспособленности, популяция, генотип, трофическая функция

Введение

Генетические алгоритмы (ГА) применяются для решения многих конкретных научных и технических проблем. Эффективность ГА заключена в его способности манипулировать одновременно многими параметрами [1]. Основная особенность системы Лотка-Вольтера состоит в том, что они описывают колебания плотности популяций [3].

При описании ГА используются определения, заимствованные из генетики.

ГА – это адаптивные методы функциональной оптимизации, основанные на компьютерном имитационном моделировании биологической эволюции.

Популяция – это конечное множество особей.

Особи, входящие в популяцию, в ГА представляются хромосомами с закодированными в них множествами параметров задачи.

Хромосома – последовательность генов.

Ген – это последовательность кодов (генотипов), которая содержит всю информацию, необходимую для того, чтобы создать функциональный организм с определенными характеристиками (генотипом).

Локус или позиция указывает место размещения данного гена в хромосоме

Жизненный цикл популяции – это несколько случайных скрещиваний (посредством кроссовера) и мутаций, в результате которых к популяции добавляется какое-то количество новых индивидуумов. Отбор в генетическом алгоритме – это процесс формирования новой популяции из старой [4].

Постановка задачи

Для решения поставленной задачи рассмотрим популяцию из N особей жертв и M особей хищников, разбитых на 4-е группы. Для отбора особей на скрещивание будем использовать рулеточную селекцию. В качестве генетических операторов будем использовать 1-точечный кроссовер и битовую мутацию. Генерируются случайные последовательности интервалов ожидания пуассоновского процесса между рождениями и мутациями хищников и жертв.

Инициализация, т.е. формирование исходных популяций хищников и жертв заключается в случайном выборе заданного количества хромосом (особей), представляемых двоичными последовательностями фиксированной длины (в нашем случае – 8 бит).

Оценивание приспособленности хромосом в популяциях состоит в расчете функции приспособленности для каждой хромосомы в этой популяции. Чем больше

значение этой функции, тем выше «качество» хромосомы. На каждой итерации ГА приспособленность каждой особи оценивается при помощи функции приспособленности, и на этой основе создается следующая популяция особей, составляющих множество потенциальных решений проблемы, например, задачи оптимизации. Очередная популяция в ГА называется поколением, а к вновь создаваемой популяции применяется термин «новое поколение» или «поколение потомков». В роли функции приспособленности жертв будет выступать функция $Fq0(x) = 2x^2 + 1$, хищников — $Fq1(x) = 5x^2 + 1$.

Оператор селекции осуществляет отбор хромосом в соответствии со значениями их функции приспособленности. При использовании метода рулетки оценка каждой хромосомы пропорциональна размеру соответствующей ей участка на колесе рулетки. Поэтому чем больше значения функции приспособленности, тем больше сектор участка на колесе рулетки. Все колесо рулетки соответствует сумме значений функции приспособленности всех хромосом рассматриваемых популяций. Каждой хромосоме соответствует сектор колеса рулетки v_k ($k = 0, 1, 2, 3$), выраженный в процентах

согласно формуле $v_k = ps_k \cdot 100\%$, где $ps_k = \frac{FQ_k}{\sum_{k=0}^3 FQ_k}$, причем FQ_k – значение функции

приспособленности хромосомы, а ps_k – вероятность селекции хромосомы. Выбор родителей производится в результате последовательных запусков колеса рулетки. В результате процесса селекции создается родительская популяция, также называемая родительским пулом.

Количество потомства зависит от количества информации $I_k = \log_2 \frac{FQ_k}{FQ_k}$,

содержащегося в признаках особи о том, что она выживет и даст потомство. Поскольку количество потомства особи пропорционально ее приспособленности, то естественно считать, что если это количество информации:

- *положительно*, то данная особь выживает и дает потомство, численность которого пропорциональна этому количеству информации (в нашем случае максимальное число потомков у хищников составляет 4, а у жертв – 8);
- *меньше или равно нулю*, то особь доживает до половозрелого возраста, но потомства не дает (его численность равна нулю)/

Применение генетических операторов

Оператор скрещивания. На первом этапе скрещивания выбираются пары хромосом из родительской популяции. На данном этапе хромосомы из родительской популяции объединяются в пары. Это производится случайным образом. Далее для каждой пары отобранных таким образом родителей разыгрывается позиция гена (локус) в хромосоме, определяющая так называемую точку скрещивания (точка разрыва). Точка разрыва – участок между соседними битами в строке. Случайным образом выбирается одна из 7 точек разрыва. Обе родительские структуры разрываются на два сегмента по этой точке. Затем, соответствующие сегменты различных родителей склеиваются и получаются два генотипа потомков.

Оператор мутации. Мутация – стохастическое изменение части хромосом. Каждый ген строки, которая подвергается мутации, с вероятностью мутации (обычно очень маленькой) меняется на другой ген. Хромосомы, полученные в результате применения генетических операторов к хромосомам временной родительской популяции, включаются в состав новой популяции. Она становится так называемой текущей популяцией для данной итерации ГА [4].

Классическая модель Лотки-Вольтера

Рассмотрим систему двух видов: волки и зайцы. Рост численности волков ведет к сокращению поголовья зайцев. Вызванный этим дефицит пищи приводит к сокращению численности волков, что в свою очередь способствует развитию популяции зайцев.

$$F(t, y) = \begin{cases} c_1 \cdot y_0 + a_{12} \cdot y_0 \cdot y_1 \\ c_2 \cdot y_1 + a_{21} \cdot y_0 \cdot y_1 \end{cases}$$

система двух дифференциальных уравнений, описывающих взаимодействие двух популяций [5].

$c_1 = 100$ – численность вида растет, $c_2 = -20$ – численность вида уменьшается;
 $a_{12} = -1$ – взаимодействие хищников и жертв, $a_{21} = 2$ – взаимодействие хищников и жертв. Результаты расчета изображены на рис. 1.

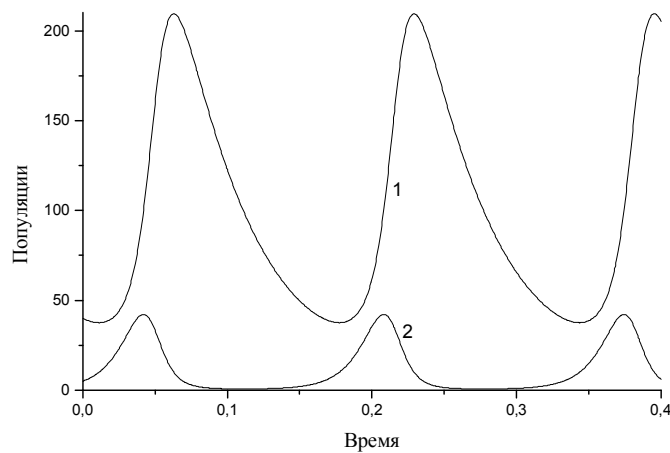


Рис. 1. Колебания численности хищников (1) и жертв (2) при наличии конкуренции

Модифицированная модель Лотки-Вольтера

В модифицированной модели Лотки-Вольтера процесс рождения учитывается только в блоке ГА. Рассмотрим систему 8 дифференциальных уравнений, описывающих взаимодействие 8 популяций. N_i ($i = 0 \dots 3$) – жертвы, N_i ($i = 4 \dots 7$) – хищники. Введем коэффициенты, отражающие наличие внутривидовой конкуренции хищников и жертв (ν), коэффициенты смертности жертв (α), трофические функции хищников и жертв, уровень насыщения (А) и КПД (К) хищников (коэффициент полезного действия «переработки» биомассы жертв в собственную биомассу).

$V(x)$ – трофическая функция хищника, т.е. количество жертв, потребляемых одним хищником за единицу времени. $V(x) = A \cdot x^2 / (K + x^2)$ [5]. $W(x)$ – функция жертв, т.е. чем больше жертв, тем медленнее умирают хищники. $W(x) = m \cdot e^{-x \cdot \gamma}$. Группа жертв

N_2 обладает следующими свойствами: быстрые, «глупые» (не чувствуют хищников), малые коэффициенты внутривидовой конкуренции (плохо конкурируют), большие коэффициенты смертности, большие КПД и уровень насыщения хищников (хорошо адаптированы к среде), малая скорость рождения. Группа жертв N_3 обладает противоположными свойствами. Группа хищников N_6 обладает такими свойствами: быстрые, «глупые» (не чувствуют жертв), малые коэффициенты внутривидовой конкуренции (плохо конкурируют), малые значения m и большие γ (медленно умирают), малая скорость рождения. Группа хищников N_7 обладает противоположными свойствами.

Процессы умирания описываются усеченными уравнениями Лотка-Вольтера:

$$\begin{aligned} \frac{dN_0}{dt} &= - \left(\sum_{i=0}^3 \alpha_{0,i} N_i + V0(N_0) \cdot \left(\sum_{p=4}^7 N_p \right) + \sum_{p=0}^7 N_0 \cdot N_p \cdot v_{0,p} \right) & \frac{dN_4}{dt} &= - \left(W4 \left(\sum_{\varphi=0}^3 N_{\varphi} \right) \cdot N_4 + \sum_{p=4}^7 N_4 \cdot N_p \cdot v_{4,p} \right) \\ \frac{dN_1}{dt} &= - \left(\sum_{i=0}^3 \alpha_{1,i} N_i + V1(N_1) \cdot \left(\sum_{p=4}^7 N_p \right) + \sum_{p=0}^7 N_1 \cdot N_p \cdot v_{1,p} \right) & \frac{dN_5}{dt} &= - \left(W5 \left(\sum_{\varphi=0}^3 N_{\varphi} \right) \cdot N_5 + \sum_{p=4}^7 N_5 \cdot N_p \cdot v_{5,p} \right) \\ \frac{dN_2}{dt} &= - \left(\sum_{i=0}^3 \alpha_{2,i} N_i + V2(N_2) \cdot \left(\sum_{p=4}^7 N_p \right) + \sum_{p=0}^7 N_2 \cdot N_p \cdot v_{2,p} \right) & \frac{dN_6}{dt} &= - \left(W6 \left(\sum_{\varphi=0}^3 N_{\varphi} \right) \cdot N_6 + \sum_{p=4}^7 N_6 \cdot N_p \cdot v_{6,p} \right) \\ \frac{dN_3}{dt} &= - \left(\sum_{i=0}^3 \alpha_{3,i} N_i + V3(N_3) \cdot \left(\sum_{p=4}^7 N_p \right) + \sum_{p=0}^7 N_3 \cdot N_p \cdot v_{3,p} \right) & \frac{dN_7}{dt} &= - \left(W7 \left(\sum_{\varphi=0}^3 N_{\varphi} \right) \cdot N_7 + \sum_{p=4}^7 N_7 \cdot N_p \cdot v_{7,p} \right) \end{aligned}$$

Результаты моделирования с помощью модифицированных уравнений Лотка-Вольтера приведены на рис. 2, 3.

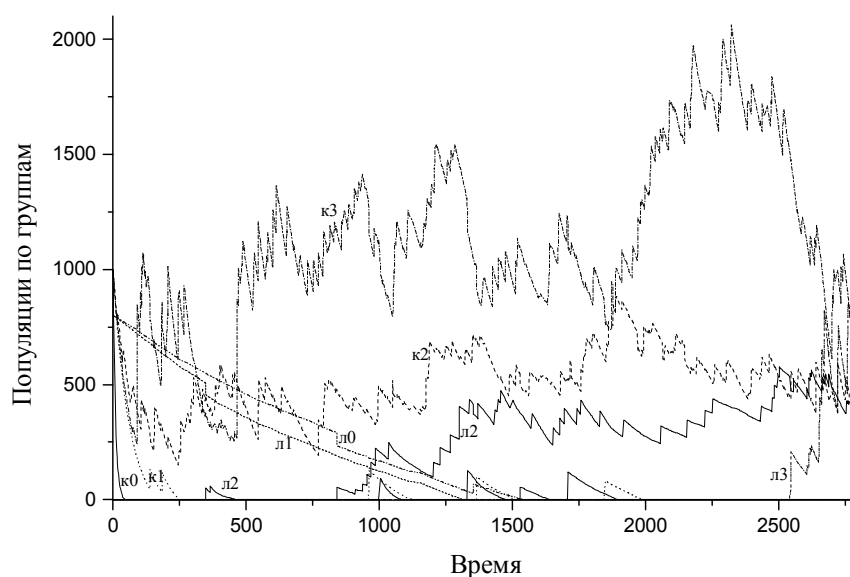


Рис. 2. Изменение численности популяций хищников (л0, л1, л2, л3) и жертв (к0, к1, к2, к3) для каждой группы

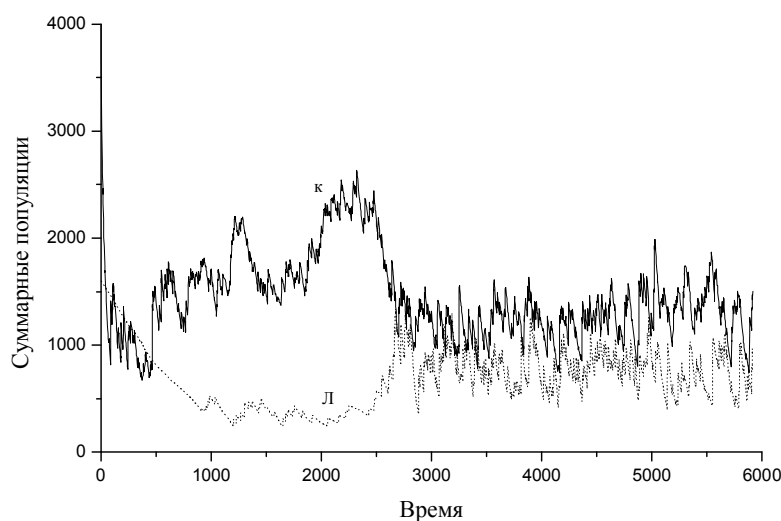


Рис. 3. Общее изменение численности популяций хищников (л) и жертв (к)

Заключение

Классическая модель Лотка-Вольтера описывает колебания численности популяций хищников и жертв, вид графиков определяется начальными данными. Стохастическая модифицированная модель Лотка-Вольтера, предложенная в данной работе, описывает множество деталей процесса взаимодействия популяций. На рис. 2, 3 исследован один из возможных сценариев появления и развития популяции хищников в среде с большим первоначальным числом жертв. Начальное число хищников невелико и заполняет две группы (номер 4 и 5) хищников «плохого» качества. Эти группы имеют отрицательное количество информации и не способны к размножению. Но процессы мутации могут привести к появлению жизнеспособных потомков в популяции (группы 6, 7). Из этих потомков может развиваться жизнеспособная популяция хищников, способная ограничить неконтролируемый рост числа популяции жертв. Детали этого процесса представлены на рис. 2. Когда заканчивается переходный процесс, наблюдается выход на стационарный режим. Популяции жертв и хищников сосуществуют. В силу случайных флуктуаций этот стационарный процесс может прекратиться. Проведение вычислительных экспериментов позволяет определить границы параметров, при которых система устойчиво демонстрирует стабильное поведение. Полученные результаты применимы при исследовании систем дифференциальных уравнений, возникающих в различных областях математической биологии: ферментативной кинетике, нейрофизиологии, иммунологии и т.п.

Литература

1. Ярушкина Н.Г. Основы теории нечетких и гибридных систем. М: Финансы и статистика. – 2004.
2. Базыкин А.Д. Математическая биофизика взаимодействующих популяций. М: Наука. – 1985.
3. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. Пер. с польск. И.Д. Рудинского. – М.: Горячая линия. – Телеком. – 2004. – 452 с.: ил.
4. Плотинский Ю.М. Модели социальных процессов. Изд. 2-е, перераб. и доп. – М.: Логос. – 2001.
5. Свирежев Ю.М., Логофет Д.О. Устойчивость биологических сообществ. М.: Наука. – 1978.

О ПОВЫШЕНИИ ЗАЩИЩЕННОСТИ ПЕРЕКОДИРОВАННЫХ ИЗОБРАЖЕНИЙ

Ю.А. Соколова

Научные руководители:

д.ф.-м.н., профессор Г.П. Мирошниченко, д.ф.-м.н., профессор И.Ю. Попов

Описывается способ преобразования одного изображения в другое с использованием комбинированного подхода для передачи информации. Особенностью процесса трансформирования изображения является то, что с помощью матрицы преобразования исходный рисунок заменяется другим выбранным пользователем объектом. Цель применения комбинированного метода состоит в разделении передачи данных на два потока, один из которых достаточно объемен и использует классические каналы связи, а другой очень мал и пользуется дорогостоящим, но очень защищенным квантовым каналом.

Ключевые слова: преобразование изображений, передача данных

Введение

С развитием коммуникационных систем передача информации в современном обществе набирает все большие и большие объемы. Для множества организаций и обычных пользователей гораздо удобнее обмениваться не только открытыми, но и конфиденциальными данными посредством глобальной сети. Поэтому особое внимание уделяется шифрованию данных и надежной передаче их посредством каналов связи. В настоящее время разработаны достаточно устойчивые и надежные алгоритмы для шифрования секретных данных, используемые в первую очередь крупнейшими разработчиками информационных технологий. К классическим методам шифрования относятся симметрические криптоалгоритмы, которые делятся на два вида: блочные и поточные шифры. Идея большинства итерационных блочных шифров состоит в построении стойкой криптографической системы путем последовательного применения достаточно простых преобразований и дальнейшей передачей ключей по секретному каналу. Первым зарегистрированным стандартом был алгоритм Data Encryption Standard (DES), разработанный фирмой IBM. После был придуман более надежный алгоритм Rijndael, ставший основой стандарта AES (Advanced Encryption Standard) и использующийся до сих пор. Основная идея поточного шифрования состоит в том, что разным преобразованиям подвергается не блок данных, а каждый символ в отдельности. Такие шифры почти всегда работают быстрее и требуют гораздо меньше программного кода для реализации. Примером данного шифра может служить RC4. Второй разновидностью классических криптоалгоритмов являются асимметрические шифры, стойкость которых основывается на трудности вычисления определенных математических задач (например, факторизации большого числа на простые множители). Наиболее известными являются RSA, Knapsack Cryptosystem, EGCS. В таких системах отпадает необходимость секретного канала для обмена ключами, а проблема аутентификации пользователей решена с помощью электронно-цифровых подписей [1]. Однако как только решится проблема вычисления NP-полных задач, станет возможно взломать системы, использующие алгоритмы шифрования с открытым ключом, в считанные минуты.

Существует также отдельное направление кодирования данных, такое как стеганография, а именно сокрытие секретной информации в цифровые объекты, например, изображения, с небольшой потерей качества, то есть незначительным искажением объекта. Особенность такого подхода заключается в скрывании самого факта шифрования данных. В настоящее время наиболее распространенным, но наименее стойким является метод замены наименьших значащих битов или LSB-

метод. Он заключается суть этого метода заключается в замене последних значащих битов в изображении, аудио или видеозаписи на биты скрываемого сообщения. Разница между пустым и заполненным объектом не вызывает значительной трансформации изображения и не обнаруживается визуально [2]. Другим методом встраивания сообщений является использование особенностей форматов данных, использующих сжатие с потерей данных (например, JPEG). Этот метод (в отличие от LSB) более стоек к геометрическим преобразованиям и обнаружению канала передачи, так как имеется возможность в широком диапазоне варьировать качество сжатого изображения, что делает невозможным определение происхождения искажения [3]. Алгоритм JPEG основан на дискретном косинусоидальном преобразовании (ДКП), которое является разновидностью кодирования изображения с помощью преобразования Фурье. Оно применяется к матрице изображения для получения некоторой новой матрицы коэффициентов. А именно изображение можно представить в виде ограниченной функции двух пространственных переменных, заданной на ограниченной прямоугольной области, причем оси X и Y совпадают с шириной и высотой картинка, а по оси Z откладывается значение цвета соответствующего пикселя изображения. При преобразовании мы получаем матрицу, в которой многие коэффициенты либо близки, либо равны нулю. Кроме того, благодаря несовершенству человеческого зрения, можно аппроксимировать коэффициенты более грубо без заметной потери качества изображения. При этом чтобы получить исходное изображение достаточно лишь применить обратное преобразование Фурье.

Процедура преобразования изображений

В настоящей работе предлагается некоторый способ, комбинирующий использование классических и квантовых каналов для передачи данных, при этом исходное изображение преобразуется в определенное шифровальщиком другое изображение. Для передачи открытой информации, содержащей перекодированный объект и некоторые дополнительные данные, предлагается использовать классические каналы связи, а для секретной информации – квантовые. Специфика предлагаемого подхода в том, что объем открытого сообщения серьезно увеличивается, а вот размер секретных данных будет достаточно мал, что особенно существенно для дорогостоящего квантового канала.

Предлагаемый способ состоит в следующем. Рассмотрим исходное изображение размера $n \times m$ как множество точек (пикселей), у каждой из которых существует набор характеристик таких, как цвет, яркость, насыщенность и т.д. Задача состоит в том, чтобы данное изображение (будем называть его истинным) преобразовать в некоторое выбранное шифровальщиком изображение (будем называть его перекодированным). Указанные характеристики точки задаются некоторыми числами, поэтому, изменяя эти числа, можно добиться совершенно другого отображения пикселя. При этом по данной числовой матрице изображение восстанавливается единственным образом. Итак, в результате анализа истинного и перекодированного изображений мы получаем матрицы чисел, задающих цветовые характеристики пикселей двух объектов. Преобразуем истинную числовую матрицу аддитивным образом в числовую матрицу, соответствующую изображению, которое мы хотим получить. Задающая такое преобразование матрица (матрица преобразования) и будет зашифрована. Пользователь, получив только новое изображение и не зная матрицы преобразования, никак не сможет восстановить его. Полученную матрицу преобразования мы кодируем описанным ниже способом и после шифрования передаем по открытому каналу. Для шифрования используем некоторую монотонную функцию с несколькими свободными параметрами, например,

$$f(x) = \sqrt{ax + b},$$

где a и b – неотрицательные параметры.

В качестве аргументов в функцию подставляются последовательно элементы матрицы преобразования. Причем для каждого элемента случайным образом генерируются значения параметров a и b . По открытому каналу связи будет происходить передача числовой матрицы, задающей преобразованное изображение, зашифрованной матрицы преобразования, а также наборов значений параметров функции a и b , то есть объем передаваемых данных возрастает примерно в 4 раза по сравнению с объемом матрицы изображения. По надежному квантовому каналу пересылается лишь вид функции, с помощью которой возможна дешифровка матрицы, являющейся ключом к получению исходного секретного изображения. В качестве частного случая изображение можно преобразовывать не в строго определенный объект, а в шум (случайный набор характеристик пикселей).

Заметим, что предлагаемая процедура подходит и для других способов задания изображений. В частности, если изображение задается с помощью разложений в ряды Фурье, то истинной и преобразованной матрицами являются соответствующие наборы значений коэффициентов Фурье, которые переходят друг в друга с помощью матрицы преобразования, процесс же шифрования матрицы не меняется.

Рассмотрим предлагаемый подход на конкретном примере.

Истинное изображение (рис. 1) необходимо преобразовать в перекодированное (рис. 2).



Рис. 1. Истинное изображение



Рис. 2. Перекодированное изображение

Каждый пиксель изображения задается с помощью характеристик цвета, чаще всего используется модель RGB (Red – красный, Green – зеленый, Blue – синий). Распространены также и другие цветовые модели представления пикселей, например, CMYK (Cyan – голубой, Magenta – пурпурный, Yellow – желтый, black – черный) наиболее часто используется в полиграфии и HSB (каждый цвет описывается цветовым тоном – Hue, насыщенностью – Saturation и яркостью – Brightness) удобная модель для редактирования рисунков и восприятия человеком. Мы будем пользоваться первой из вышеописанных цветовых моделей в силу ее простоты и широкого применения для компьютерной техники.

Итак, сначала истинное и перекодированное изображения с помощью программных средств (J2SE) нужно представить в виде массивов чисел, описывающих цвет и прозрачность каждой точки. Для этого используется класс PixelGrabber и его метод grabPixels(). Полученные числа (выражаются 32-битными целыми числами) нужно модифицировать, а значит необходимо понимать, как определить красную, синюю и зеленую составляющие, а также прозрачность из такого числа. В данном случае биты с 24 по 31 отвечают за прозрачность, с 16 по 23 – за красный цвет, с 8 по 15 – за зеленый, а с 0 по 7 отвечают за синий цвет. Таким образом, можно

перекодировать изображение в произвольные шумы, меняя случайным образом все или отдельные компоненты цвета.

Так как основная задача состоит в нахождении матрицы преобразования исходного изображения в перекодированное, то можно не рассматривать каждую составляющую цвета в отдельности, а использовать все числовое значение целиком. Пусть x_{ij} – это целое число, задающее цвет пикселя в исходном изображении, а y_{ij} – в перекодированном. Тогда определим операцию преобразования таким образом:

$$z_{ij} = y_{ij} - x_{ij},$$

где z_{ij} – элемент числовой матрицы преобразования.

Зная массив элементов z_{ij} можно совершить обратное преобразование и получить значения цветов пикселей, соответствующих истинному изображению. Полученную матрицу мы кодируем с помощью описанной ранее функции, предварительно случайно генерируя набор коэффициентов.

Заключение

Предложенная схема требует передачи по квантовому каналу вида функции для успешного дешифрования перекодированного изображения. Если передавать вид функции посимвольно, принимая, что каждый символ занимает около 1 байта, то в данном примере на это потребуется около 10 байт. То есть объем секретной информации очень мал.

В результате передачи сообщения по открытым каналам могут произойти искажения или появиться шумы, т.е. часть элементов матрицы преобразования будет утеряна. Но так как преобразование локально (действует на каждый пиксель в отдельности), только некоторое количество точек будет искажено. При этом возможно восстановление изображения стандартными способами, например методом усреднения с помощью соседних пикселей.

Литература

1. Авдошин С.М., Савельева А.А. Криптотехнологии Microsoft // Приложение к журналу «Информационные технологии». – 2008. – №9. – С. 2–6.
2. Кустов В.Н., Федчук А.А. Методы встраивания скрытых сообщений. // Защита информации. Конфидент. – 2000. – №3. – С. 34.
3. Быков С.Ф. Алгоритм сжатия JPEG с позиции компьютерной стеганографии. // Защита информации. Конфидент. – 2000. – №3. – С. 26.

МОДЕЛЬ РАСПРЕДЕЛЕНИЯ ЗАДАЧ ПРИ КЛАУД-ВЫЧИСЛЕНИЯХ И КВАНТОВО-КОМПЬЮТЕРНЫЕ АНАЛОГИИ

И.В. Блинова, С.И. Попов

(Санкт-Петербургский государственный университет информационных технологий, механики и оптики),

А.И. Попов (Санкт-Петербургский государственный университет)

Научный руководитель – д.ф.-м.н., профессор И.Ю. Попов

Предложена модель для описания распределения задач в системе при клауд-вычислениях. Осуществляя воздействие на клауд, меняющее поле вероятности распределения задач, в принципе, возможно, выполнить «супервычисление» и реализовать аналоги квантовых алгоритмов.

Ключевые слова: клауд-вычисления, квантовые алгоритмы

Введение

В последнее время все большее распространение получила технология клауд-вычислений («cloud computing»). К настоящему времени еще не появилось устоявшегося русского термина, поэтому мы будем здесь говорить о клауде и клауд-вычислениях. Эту технологию часто определяют как образ компьютерных вычислений, при котором ИТ-ресурсы с высокой степенью масштабируемости предоставляются множеству внешних пользователей «как услуга» через интернет. Услуги масштабируются без вмешательства пользователя на основе динамичной и эластичной архитектуры. Говоря кратко, этот подход позволяет нам не использовать ненужную нам часть данных, а работать только с необходимой нам ее частью. Основой в этом служит интернет (облако – метафорическое название Интернета). Именно он предоставляет нам возможность использовать крупную, отвлеченную ИТ-инфраструктуру, которая распределяет задачи между пользователями, выделяя им необходимые данные и аппаратные средства. Услуги и данные, размещаемые в совместно используемых, динамически масштабируемых пулах ресурсов, базируются на технологиях виртуализации и/или прикладных средах с горизонтальным масштабированием. С этой технологией исчезает необходимость инсталляции ПО и аппаратных средств, достаточно войти в систему. Сегодня cloud computing в основном используются для предоставления услуг частным лицам, например для поиска информации в Интернете, ведения персонального почтового ящика, участия в социальных сетях, а также для работы других приложений Web 2.0. Однако не следует забывать, что с помощью него можно решать гораздо более серьезные задачи, при меньших усилиях и затратах.

При данном подходе каждая из решаемых задач поступает в сеть, состоящую из тысяч элементов, и направляется на один из компьютеров, включенных в нее. При этом централизация в системе отсутствует, и компьютер-получатель задачи определяется в достаточной степени случайно, точнее, этот процесс не контролируется тем, кто загружает задачу. Использование клауд-вычислений позволяет повысить эффективность и скорость решения задач [1, 2]. Большинство работ по данной тематике посвящено именно указанным вопросам. Распределение задач в клауде и способы управления им еще исследованы недостаточно. Нас же сейчас будет интересовать как раз не решение конкретных задач, а их распределение в системе, воздействуя на которое определенным весьма специфическим образом, можно выполнить некоторые операции (не связанные с данными задачами, а использующие их распределение как среду), в частности, реализовать алгоритмы, аналогичные квантовым.

Постановка задачи

Для каждой поступающей задачи есть поле вероятностей попадания на различные компьютеры. Таким образом, в пространстве компьютеров имеется распределение вероятностей. В настоящее время еще не разработано моделей, позволяющих корректно описывать указанное распределение. Возможны две основные версии: диффузионная и волновая. В первом случае при описании с помощью уравнения диффузии имеется принцип максимума. Однако ряд данных указывает на то, что распределение может иметь максимумы во внутренних точках, что противоречит данному принципу и указывает на возможность интерференционных эффектов, т.е. на волновую модель (либо на ее аналог – модель уравнения Шредингера). В этом случае, как и в квантовой механике, возможно описание с помощью волновой функции, квадрат модуля которой есть плотность вероятности попадания задачи в данную точку компьютерного пространства, т.е. на данный компьютер (это аналог чистого состояния в квантовой механике). Измерение, как и в квантовом случае, есть применение оператора проектирования (то есть определение локализации задачи на конкретном компьютере). Аналогом квантовых гейтов (вентилей) являются унитарные операторы в компьютерном пространстве. При этом аналог квантового бита реализуется на паре связанных компьютеров. Состоянию $|0\rangle$ отвечает локализация задачи на одном из компьютеров, а $|1\rangle$ – на другом. А состояние задачи для пары компьютеров («кубита») есть $\alpha|0\rangle + \beta|1\rangle$, $|\alpha|^2 + |\beta|^2 = 1$ (вероятность локализации задачи на первом компьютере – $|\alpha|^2$, а на втором – $|\beta|^2$). Если клауд состоит из 2^n компьютеров, то можно выполнять и многокубитовые операции, вплоть до n – кубитовых, что необходимо для реализации алгоритмов, аналогичных квантовым. Для осуществления этих алгоритмов необходимо иметь возможность осуществлять такое воздействие на клауд, которое играет роль унитарного оператора в m – кубитовом пространстве, т.е. соответствующим образом меняет поле вероятностей распределения задач по клауду (заметим, что оператор действует не на сами вероятности, а на волновую функцию). Например, рассмотрим, какие операторы нужны для реализации алгоритма Гровера поиска в базе данных [3, 4]. Элементы базы (пусть их количество $m = 2^n$) кодируются двоичным кодом. Алгоритм состоит из следующих шагов.

1. Приготовление начального состояния, в котором все базисные состояния присутствуют с одинаковыми амплитудами (равновероятны). Это состояние легко реализуется в клауде.

2. Вторая операция повторяется циклически определенное число раз. Она состоит из двух этапов.

А) Контролируемое изменение фазы, при котором фазы всех состояний, кроме того, которое кодирует искомым элемент базы, не меняются, а указанное состояние меняет знак (фаза меняется на π).

Б) Применение к полученному состоянию оператора инверсии относительно среднего: $D = -I + 2P$, где I – единичная матрица размером $m \times m$, P – матрица, все элементы которой одинаковы.

Последней операцией является измерение, что в случае клауда есть определение локализации задачи на конкретном компьютере. Ясно, что сложность представляет реализация в клауде второго этапа алгоритма. Для этого необходимо иметь возможность эффективно влиять на распределение задач в клауде, чтобы обеспечить выполнение требуемого унитарного оператора, что представляется в принципе

осуществимым. Аналогичная ситуация наблюдается и с другими квантовыми алгоритмами.

Заключение

Таким образом, при выполнении сформулированных предположений с помощью клауда возможно выполнение операций, которые можно назвать супервычислениями, поскольку отдельные компьютеры решают конкретные задачи, а мы реализуем некоторый совершенно другой алгоритм, наблюдая за распределением задач по системе и осуществляя воздействие на это распределение.

Литература

1. Hwang K. Massively distributed systems: from grids and P2P to clouds//. Lect. Notes Comp. Sci. – 2008. – V. 5036. – PP. 1–12.
2. Cryans J.-D., April A., Abran A. Criteria to compare cloud computing with current database technology// Lect. Notes Comp. Sci. – 2008. – V. 5338. – PP. 114–126.
3. Нильсен М., Чанг И. Квантовые вычисления и квантовая информация. М.: Мир. – 2006. – 822 с.
4. Попов И.Ю. Квантовый компьютер и квантовые алгоритмы. СПб: СПбГУ ИТМО – 2007. – 88 с.

ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МЕТОДИКА ПРОЕКТИРОВАНИЯ СИСТЕМ С ЭЛЕМЕНТАМИ, ОБЩИМИ ПО ФУНКЦИОНАЛЬНОСТИ, НО РАЗНОПАРАМЕТРИЧЕСКИХ (НА ПРИМЕРЕ СОСТАВНОГО ЗЕРКАЛА)

А.Ю. Рабыш, Ю.В. Беликов

Научный руководитель – д.т.н., профессор А.В. Демин

В статье на основе основных принципов объектно-ориентированного программирования была предложена объектно-ориентированная методика построения систем с элементами, сходными по функциональности и различающимися только параметрами. Данный подход был проиллюстрирован на примере проектирования системы составного зеркала: построена иерархия классов и разработан алгоритм позиционирования зеркальных элементов.

Ключевые слова: объектно-ориентированный, САПР, системы, составное зеркало

Введем класс систем с элементами, сходными по функциональности и различающимися только параметрами.

Необходимо предложить методику проектирования и классифицирования системы данного класса, а так же переводить их на язык программирования, что очень важно для создания программных комплексов САПР. И на примере проектирования системы составного зеркала произвольной формы вращения, проиллюстрировать полученную методику.

Принимается решение для создания на основе объектно-ориентированного программирования методики для проектирования систем с элементами, общими по функциональности, но различными по параметрам [1].

Объектно-ориентированная технология основывается на так называемой объектной модели. Основными ее принципами являются: *абстрагирование, инкапсуляция, модульность, иерархичность, типизация, параллелизм и сохраняемость*. Каждый из этих принципов сам по себе не нов, но в объектной модели они впервые применены в совокупности.

Объектно-ориентированный анализ и проектирование отражают эволюционное, а не революционное развитие проектирования; новая методология не порывает с прежними методами, а строится с учетом предшествующего опыта.

Понятие объекта является центральным во всем, что относится к объектно-ориентированной методологии.

Каждый объект является экземпляром какого-либо класса; класс может порождать любое число объектов. В большинстве практических случаев классы статичны, т.е. все их особенности и содержание определены изначально. Из этого следует, что любой созданный объект относится к строго фиксированному классу. Объекты же этих классов, напротив, динамичны.

Объектно-ориентированное проектирование – это методология проектирования, соединяющая в себе процесс объектной декомпозиции и приемы представления логической и физической, а также статической и динамической моделей проектируемой системы.

На основе основных принципов объектно-ориентированного программирования введем методику построения иерархии классов системы. Иерархия классов строится исходя из принципа «от общего к частному» и определяется наследованием.

1. За базовый класс, стоящий в самом верху иерархического дерева выбираем самые общие понятия, определяющие проектируемую систему.
2. Каждый базовый класс разбивается на ряд более узких классов, и определяются связи между ними.

3. Затем каждый из этих классов тоже разбивается на ряд классов. Таким образом, иерархические деревья классов доходят до тех понятий, которые определяют проектируемую систему. Данный уровень иерархии определяется как рабочие классы. Определяются отношения между рабочими классами.
4. Строится полная диаграмма классов проектируемой системы.
5. В каждом из рабочих классов вводятся свойства, определяющие параметры элемента системы, и методы, определяющие операции над этими элементами системы.
6. Строится конструктор с определенным набором свойств. Конструктор создает элементы с заданными параметрами.

Данная методика позволяет проектировать системы (с элементами, сходными по функциональности, но различающимися по параметрам) любого уровня сложности, четко отслеживать все связи между элементами системы и легко переводить на язык программирования для проектирования программных комплексов САПР.

Рассмотрим задачу построения систем составных зеркал с точки зрения ООП и построим диаграмму классов

В начале, рассмотрим пространство.

Назовем (n -мерной) «точкой» систему из n вещественных чисел: сами числа являются координатами «точки» M . Множество всех мыслимых n -мерных «точек» составляет n -мерное «пространство» (которое иногда называют арифметическим).

С точки зрения ООП, « n -мерное пространство» является базовым классом для иерархии классов «Пространства». Подклассами класса mn -мерное пространство» являются классы:

- «Одномерное пространство»;
- «Двумерное пространство»;
- «Трехмерное пространство»;
- «Четырехмерное пространство» и др.

Для решения задачи построения систем составных зеркал целесообразно выбрать в качестве рабочего класс «Трехмерное пространство». Заметим, что подклассы класса « n -мерное пространство» определяют свойства для иерархий классов «Точки» и «Множества точек».

Определим класс «Трехмерная точка» как подкласс базового « n -мерная точка». А класс «Трехмерное множество точек» подклассом « n -мерное множество точек».

Определим как подкласс класса «Трехмерное множество точек» класс «Трехмерная поверхность».

Класс «Трехмерное пространство» определяет объекты классов «Трехмерная точка» и «Трехмерное множество точек».

Поверхностью называют множество последовательных положений линий, перемещающихся в пространстве. Эта линия может быть прямой или кривой и называется **образующей** поверхности. Если образующая кривая, она может иметь постоянный или переменный вид. Перемещается образующая по **направляющим**, представляющим собой линии иного направления, чем образующие. Направляющие линии задают закон перемещения образующим. При перемещении образующей по направляющим создается **каркас** поверхности, представляющий собой совокупность нескольких последовательных положений образующих и направляющих. Рассматривая каркас, можно убедиться, что образующие l и направляющие m можно поменять местами, но при этом поверхность получается одна и та же.

На практике из всех возможных способов образования поверхности выбирают наиболее простой.

В зависимости от формы образующей все поверхности можно разделить на **линейчатые**, у которых образующая прямая линия, и **нелинейчатые**, у которых образующая кривая линия.

В зависимости от формы образующей и закона ее перемещения в пространстве поверхности можно разделить на отдельные группы.

- Линейчатые поверхности (поверхности, которые могут быть образованы с помощью прямой линии.).

- Нелинейчатые поверхности (поверхности, которые могут быть образованы только с помощью кривой линии).

Определим эти поверхности как подклассы класса «Трёхмерная поверхность». Для решения задачи построения систем составных зеркал целесообразно выбрать класс «Нелинейчатые поверхности» в качестве рабочего.

Определим классы:

- с постоянной образующей;
- с переменной образующей, как подклассы класса «Нелинейчатые поверхности». Для решения задачи построения систем составных зеркал целесообразно выбрать класс «С постоянной образующей» в качестве рабочего.

Определим классы:

- поверхности вращения;
- циклические (трубчатые) поверхности, как подклассы класса «Нелинейчатые поверхности». Для решения задачи построения систем составных зеркал целесообразно выбрать класс «Поверхности вращения» в качестве рабочего. От данного класса будут наследоваться подклассы.

- Сфероид.
- Гиперболоид.
- Параболоид.
- Эллипсоид.

Все эти классы будут использованы в качестве рабочих для задачи построения систем составных зеркал.

Трёхмерное пространство в системе будет представлено декартовой системой координат $oxyz$. Расположим поверхность вращения таким образом, чтобы её вершина совпадала с началом координат, а ось вращения совпадала с положительной полуосью oz .

В класс «Поверхности вращения» введем свойство «Граница», определяющее ограничение для функции формы поверхности составного зеркала. Это свойство будет наследоваться в каждый из подклассов данного класса.

Функция формы поверхности $f(x,y,z)$ равномерно ограничена на множестве S «точек» (x,y) , если абсолютная величина (модуль) функции f как функция от (z) имеет верхнюю границу, не зависящую от (x,y) на S .

Система составного зеркала, представленная ограниченной поверхностью вращения, должна быть разбита на отдельные составные элементы, которые в совокупности будут составлять собой поверхность составного зеркала.

Расположим образующие и направляющие линии поверхности таким образом, чтобы они разбивали её на ряд отдельных элементов

Если на поверхности вращения направляющие линии расположены параллельно плоскости oxy , тогда каждые две соседние друг с другом направляющие ограничивают отдельный ряд элементов, имеющих одинаковую форму и отличающихся друг от друга на равную величину в полярной системе координат.

Таким образом, поверхность будет представлена системой, состоящей из подсистем, в каждой из которых элементы отличаются друг от друга только координатами в полярной системе координат.

Распишем элементы, на которые разбита каждая поверхность вращения, с точки зрения ООП.

Введем класс «Элемент», являющийся базовым классом в иерархии «Элементы».

Для упрощения расчетов абстрагируемся от того, что элемент представляет собой часть поверхности неопределенной формы. Каждый из элементов представим как часть плоскости, касательной к поверхности вращения и ограниченной по контуру.

Элементы, выполненные по контуру обладающие центральной симметрией, отличаются более простыми расчетными соотношениями и конструктивной реализацией. Плоскость без зазоров можно представить в виде совокупности треугольников, четырехугольников и шестиугольников (гексагонов).

Тогда от класса «Элемент» наследуются подклассы:

- Треугольник.
- Четырехугольник.
- Гексагон.

Поверхность вращения без зазоров может быть представлена в виде совокупности элементов треугольной и гексагональной формы.

Наиболее приемлемыми для компоновки составного зеркала следует признать элементы, выполненные по контуру как гексагональные, т.к. в этом случае составленная из подобных элементов поверхность не намного отличается по контуру от поверхности вращения.

Таким образом, в качестве рабочего выбираем класс «Гексагон». Класс «Поверхности вращения» порождает экземпляры класса «Гексагон»

В качестве рабочих классов были выбраны класс «Гексагон» и, в зависимости от формы расчетной поверхности, один из подклассов класса «Поверхность вращения».

На основе полученной иерархии классов строится универсальная расчетная математическая модель составного зеркала.

К полученной модели необходимо свести все параметры реальной физической модели составного зеркала с некоторым допуском, определяемым точностью оптических датчиков положения и шагом двигателей, приводящих в движение элементы составного зеркала.

Поскольку поверхность элемента определяется кривизной поверхности и нормалью к ней, то можно показать, что положение ЗС в пространстве эквивалентно положению касательной плоскости, построенной относительно центра симметрии этого элемента в выбранной системе координат. Тогда алгоритм позиционирования ЗС на НП определим как алгоритм совмещения каждой касательной плоскости, определяющей ЗС, с касательными плоскостями, построенными к точкам на расчетной поверхности в её системе координат. С точками на расчетной поверхности должны совпадать центры симметрии ЗС. Расчетная поверхность определяет и несущую поверхность. При этом касательная плоскость в пространстве определяется тремя координатами либо декартовыми, либо сферическими. Таким образом, алгоритм позиционирования ЗС на НП может быть представлен в следующем виде (см. рисунок).

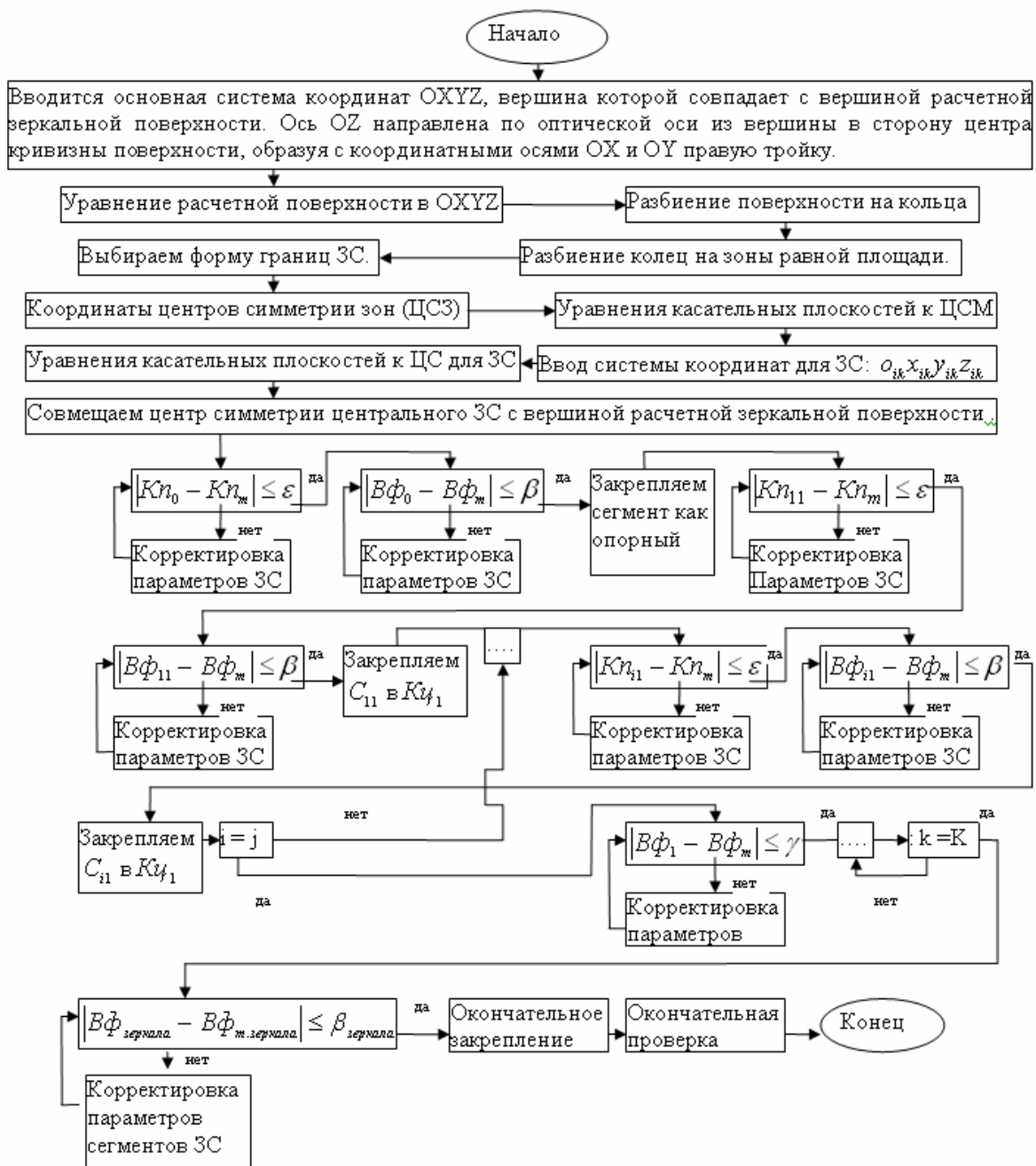


Рисунок. Алгоритм позиционирования зеркальных элементов

На рисунке приняты следующие обозначения: Kn_0 – параметры касательной плоскости центрального ЗС, Kn_m – расчетные параметры основной поверхности, $B\phi_0$ – реальный волновой фронт центрального ЗС, $B\phi_m$ – расчетный волновой фронт ЗС, Kn_{ik} – параметры касательной плоскости i -го ЗС k -го кольца, $B\phi_{ik}$ – реальный волновой фронт i -го ЗС k -го кольца, $B\phi_k$ – реальный волновой фронт k -го кольца, $B\phi_{зеркала}$ – реальный волновой фронт составного зеркала, $B\phi_{т.зеркала}$ – расчетный волновой фронт составного зеркала, ϵ – поле допуска для корректировки положения ЗС в OXYZ, β – поле допуска волнового фронта ЗС, $\beta_{зеркала}$ – поле допуска волнового фронта составного зеркала.

Входными данными для алгоритма позиционирования ЗС на НП являются:

- уравнение расчетной зеркальной поверхности (РЗП) для не составного зеркала;

- уравнение волнового фронта;
- координаты точек на расчетной поверхности, с которыми должны быть совмещены центры симметрии ЗС;
- уравнения касательных плоскостей, построенных к точкам позиционирования центров симметрии ЗС;
- допуск на положение ЗС относительно РЗП;
- допуск на отклонение волнового фронта составного зеркала от расчетного значения для РЗП.

Заключение

В статье на основе основных принципов объектно-ориентированного программирования был предложена объектно-ориентированная методика построения систем с элементами, сходными по функциональности и различающимися только параметрами. Данный подход был проиллюстрирован на примере проектирования системы составного зеркала: построена иерархия классов и разработан алгоритм позиционирования зеркальных элементов.

В следующих статьях данная методика будет расширена. А так же будет построена расчетная модель составного зеркала на её основе.

Литература

1. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++. – Второе изд. – СПб: Невский Диалект. – 2000. – 637 с.
2. Мастаченко В.Н. Проектирование САПР из типовых компонентов. Учебное пособие. – М.: МИИТ. – 2004. – 96 с.
3. Канатников А.Н., Крищенко А.П. Аналитическая геометрия: Учебник для вузов. 4-е изд., испр. /Под ред. В.С. Зарубина, А.П. Крищенко. – М.: Изд-во МГТУ им. Н.Э. Баумана. – 2005. – 392 с.
4. Теория систем. Математические методы и моделирование. Сборник статей Пер. с англ. – М.: Мир. – 1989. – 384 с.

НАЧАЛЬНАЯ ВЫСТАВКА И КАЛИБРОВКА ГИРОГОРИЗОНТКОМПАСА НА БАЗЕ БЕСКАРДАННОГО ЭЛЕКТРОСТАТИЧЕСКОГО ГИРОСКОПА И БЛОКА МИКРОМЕХАНИЧЕСКИХ ДАТЧИКОВ

А.В. Лочехин

Научный руководитель – д.т.н., профессор Г.И. Емельянецв

Рассматриваются алгоритмы начальной выставки и калибровки бескарданного гироскопа на базе электростатического гироскопа и блока микромеханических датчиков. Особенностью рассматриваемого решения начальной выставки и калибровки в пуске является оценивание суммарных дрейфов электростатического гироскопа. В качестве эталонной информации привлекаются данные приемной аппаратуры спутниковой навигационной системы с разнесенными антеннами и инерциального блока на микромеханических датчиках. При этом формируется виртуальный («идеальный») электростатический гироскоп, используемый в качестве опорного. Алгоритмы строятся на базе обобщенного фильтра Калмана с обратной связью по всему вектору состояния.

Ключевые слова: Инерциальные системы, гироскопия, калибровка

Одно из важных требований, предъявляемых к современным системам ориентации и навигации на сегодняшний день, является время готовности системы [3]. Для ГГК на базе электростатического гироскопа (ЭСГ) калибровка КМУ занимает в среднем сутки, что неприемлемо для практической реализации. Это связано с необходимостью разделения соответствующих коэффициентов, приводящих к наличию кратных частот с периодом от 24 до 4 часов.

Особенностью рассматриваемого решения задачи начальной выставки и калибровки в пуске является оценивание суммарных дрейфов электростатического гироскопа. В качестве эталонной информации привлекаются данные приемной аппаратуры спутниковой навигационной системы (ПА СНС) с разнесенными антеннами и инерциального блока (ИБ) на микромеханических датчиках. При этом формируется виртуальный («идеальный») электростатический гироскоп, используемый в качестве опорного.

На основании аналитических решений формируются алгоритмы функционирования гироскопа на базе бескарданного электростатического гироскопа и блока микромеханических датчиков в режиме калибровки.

Формирование измерений

$$z_{avn} = \cos \theta^R - \cos \theta^S; \quad (1)$$

$$z_{1_2} = \bar{h}_{int1_2}^R - \bar{h}_{int1_2}^{et};$$

где $\cos \theta^S = h_{11}^b h_{12}^b + h_{21}^b h_{22}^b + h_{31}^b h_{32}^b$; $\cos \theta^R = h_{11}^{in} h_{12}^{in} + h_{21}^{in} h_{22}^{in} + h_{31}^{in} h_{32}^{in}$; $\bar{h}_{int_i}^R, \bar{h}_{int_i}^{et}$ (здесь $i = 2$) – расчетные \bar{h}^R (прогнозируемые) и эталонные \bar{h}^{et} , сформированные по внешним данным, значения ортов \bar{h}_i ЭСГ в проекциях на оси квазиинерциальной системы координат $int_1 int_2 int_3$, совпадающей в моменты коррекции положения ЭСГ с осями ортогонального гироскопического трехгранника $q_1 q_2 q_3$, построенного на ортах ЭСГ_i по данным их СУ – $\bar{h}_{kn_i}^S$.

Положение орта $\bar{h}_{in_i}^R$ кинетического момента реального гироскопа в ИСК вычисляется из решения следующего векторного уравнения при известной модели ухода

$$\begin{aligned}\frac{d\bar{h}_{in_i}^R}{dt} &= \bar{\omega}_{in_i} \times \bar{h}_{in_i}^R; \\ \bar{h}_{in_i}^R(t_0) &= C_{in}^{c-et} \cdot C_c^{kn-i} \cdot \bar{h}_{kn_i}^S(t_0).\end{aligned}\quad (2)$$

Здесь $\bar{h}_{kn_2}^S$ – вектор измеренных значений направляющих косинусов орта кинетического момента реального ЭСГ₂ в его корпусных осях; C_{in}^{c-et} – эталонные значения матрицы ориентации объекта относительно ИСК, которая формируется по внешним данным (ИСОИ) [4].

По значениям

$\bar{h}_{in_1}^R, \bar{h}_{in_2}^R(j+1)$ – находим матрицу $C_{in}^q(j+1)$ и значения $\cos\theta^R(j+1), \sin\theta^R(j+1)$ по прогнозируемым значениям направляющих косинусов для данной пары ЭСГ;

Положение квазиинерциальной системы координат относительно ИСК определяется матрицей C_{in}^{int} , равной значению матрицы C_{in}^q в моменты коррекции положения ЭСГ.

Тогда необходимые для формирования измерений расчетные значения ортов ЭСГ_i будут равны:

$$\bar{h}_{int_i}^R = C_{int}^{in} \cdot \bar{h}_{in_i}^R.$$

Эталонные значения ортов ЭСГ_i формируются по внешним данным C_{in}^{c-et} (данным ИСОИ) и данным $\bar{h}_{kn_i}^S$ СУ ЭСГ следующим образом:

$$\begin{aligned}\bar{h}_{int_i}^{et} &= C_{int}^{in} \cdot C_{in}^{c-et} \cdot C_c^b \cdot C_b^{kn-i} \cdot \bar{h}_{kn_i}^S; \text{ или} \\ \bar{h}_{int_i}^{et} &= C_{int}^{in} \cdot C_{in}^{h-et} \cdot C_h^c \cdot C_c^{kn-i} \cdot \bar{h}_{kn_i}^S.\end{aligned}$$

Расчетная модель погрешностей

Модель погрешностей $\Delta\bar{h}_{int_i}^R$ интегрированной системы при прогнозировании текущих значений ортов $\bar{h}_{int_i}^R$ ЭСГ_i в проекциях на оси квазиинерциальной системы координат $int_1 int_2 int_3$, требуемая для обработки в ФК измерений (1), была получена варьированием уравнений (2). Линеаризация матрицы динамики системы осуществлялась относительно следующих значений ортов ЭСГ_i [2]

$$\bar{h}_{int_1o}^R = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \bar{h}_{int_2o}^R = \begin{bmatrix} 0 \\ \cos\theta \\ \sin\theta \end{bmatrix}.$$

Упрощенную модель погрешностей уходов реального ЭСГ-2 (учитывая, что калибровка КМУ ЭСГ и привязка его измерительных осей к корпусу ЭСГ произведена при калибровке на стенде завода-изготовителя) можно принять в следующем виде:

$$\begin{aligned}\Delta\dot{h}_{int1}^R &= -(\omega_{int3} + \omega_{int2} \cos\theta / \sin\theta) \cdot \Delta h_{int2}^R + \sin\theta \cdot \Delta\omega_{int2} - \cos\theta \cdot \Delta\omega_{int3}, \\ \Delta\dot{h}_{int2}^R &= \omega_{int3} \cdot \Delta h_{int1}^R + (\omega_{int1} \cos\theta / \sin\theta) \cdot \Delta h_{int2}^R - \sin\theta \cdot \Delta\omega_{int1}, \\ \Delta\dot{\omega}_{int1} &= 0 + q_1, \\ \Delta\dot{\omega}_{int2} &= 0 + q_2, \\ \Delta\dot{\omega}_{int3} &= 0 + q_3,\end{aligned}\quad (3)$$

где $\Delta\omega_{int1}, \Delta\omega_{int2}, \Delta\omega_{int3}$ – суммарные остаточные дрейфы ЭСГ-2 в осях int , характеризующие погрешности калибровки КМУ ЭСГ на заводе, изменчивость КМУ от

запуска к запуску и в данном запуске, описываемые винеровскими процессами с заданными параметрами.

Можно показать, что измерения вблизи точки линеаризации могут быть описаны в следующем виде:

Можно показать, анализируя матрицу

$$C_{int_pr}^{int_et} = C_{int_pr}^q \cdot C_q^{int_et} = E - \delta C_{int_et}^{int_pr},$$

где $\delta C_{int_et}^{int_pr} = \begin{bmatrix} 0 & -\Lambda_{int3} & \Lambda_{int2} \\ \Lambda_{int3} & 0 & -\Lambda_{int1} \\ -\Lambda_{int2} & \Lambda_{int1} & 0 \end{bmatrix}$, что вблизи точки линеаризации

$$\vec{h}_{int_1o}^R = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \vec{h}_{int_2o}^R = \begin{bmatrix} 0 \\ \cos \theta \\ \sin \theta \end{bmatrix}$$

имеют место следующие приближенные соотношения:

$$\begin{aligned} \Lambda_{int1} &= -\Delta h_{int3_1}^R, \\ \Lambda_{int2} &= -\frac{1}{\sin \theta} \Delta h_{int1_2}^R + ctg \theta \cdot \Delta h_{int1_1}^R, \\ \Lambda_{int3} &= \Delta h_{int1_1}^R, \end{aligned} \quad (4)$$

где $\Delta h_{int_j_i}^R$ – составляющие ($j=1,2,3$) векторов погрешностей прогнозирования уходов ЭСГ_i ($i=1,2$) в проекциях на оси квази-ИСК.

Учитывая (3) и соотношения (3), получим для реального ЭСГ-2 вблизи точки линеаризации следующие приближенные выражения:

$$z_{1_2} = \cos \theta \cdot \Lambda_{int3} - \sin \theta \cdot \Lambda_{int2} - \Delta h_{int1_2}^{et};$$

или, учитывая (11), получим

$$z_{1_2} = \Delta h_{int1_2}^R - \Delta h_{int1_2}^{et}.$$

Можно показать, что для автономного измерения z_avt будем иметь,

$$z_avt = \cos \theta \cdot \Delta h_{int2_1}^R + \sin \theta \cdot \Delta h_{int3_1}^R + \Delta h_{int2_2}^R - \Delta(\cos \theta^S).$$

Так как $\Delta h_{int2_1}^R = 0$, $\Delta h_{int3_1}^R = 0$, (опорный ЭСГ-1 – идеальный ЭСГ), то

$$z_avt = \Delta h_{int2_2}^R - \Delta(\cos \theta^S),$$

где $\Delta(\cos \theta^S) = \sin \theta \cdot \Delta h_{int3_1}^S + \Delta h_{int2_2}^S$.

Здесь

$$\vec{h}_{c_1}^S = C_c^{in_et} \cdot \vec{h}_{in_1}^R = C_c^{h_et} C_h^{in_et} \vec{h}_{in_1}^R;$$

$$\Delta \vec{h}_{c_1}^S = \Delta C_c^{h_et} \cdot C_h^{in_et} \cdot \vec{h}_{in_1}^R + C_c^{h_et} \cdot \Delta C_h^{in_et} \cdot \vec{h}_{in_1}^R + C_c^{in_et} \cdot \Delta \vec{h}_{in_1}^R.$$

Так как $\Delta \vec{h}_{in_1}^R = 0$, то

$$\Delta \vec{h}_{c_1}^S = \Delta C_{h,c}^{et} \cdot C_{in,h}^{et} \cdot \vec{h}_{in_1}^R + C_{h,c}^{et} \cdot \Delta C_{in,h}^{et} \cdot \vec{h}_{in_1}^R. \quad (5)$$

Тогда $\Delta h_{int3_1}^S$ – это погрешности $\Delta \vec{h}_{c_1}^S$ ЭСГ-1 в проекциях на ось int3.

Т.е. $\Delta \vec{h}_{c_1}^S = (C_c^{int})^T \cdot \Delta \vec{h}_{c_1}^S$; $\Delta h_{int2_2}^S$ – это погрешности $\Delta \vec{h}_{kn_2}^S$ СУ ЭСГ-2 в проекциях на ось int2, а также включает и погрешности «геометрии» ЭСГ-2 в проекциях на ось int2.

Тогда, учитывая (5), имеем

$$\Delta \bar{h}_{int_1}^S = (C_c^{int})^T \cdot \Delta \bar{h}_{c_1}^S = C_{int}^{in} \cdot C_{in}^h \cdot C_c^c (\Delta C_c^{h-et} \cdot C_h^{in} \cdot \bar{h}_{in_1}^R + C_c^h \cdot \Delta C_h^{in-et} \cdot \bar{h}_{in_1}^R);$$

или

$$\Delta \bar{h}_{int_1}^S = C_{int}^h \cdot C_h^c \cdot \Delta C_c^{h-et} \cdot C_h^{in} \cdot \bar{h}_{in_1}^R + C_{int}^{in} \cdot C_{in}^h \cdot \Delta C_h^{in-et} \cdot \bar{h}_{in_1}^R;$$

или

$$\Delta \bar{h}_{int_1}^S = C_{int}^h \cdot \delta C_h^c \cdot \bar{h}_{in_1}^R - C_{int}^{in} \cdot \delta C_h^{in} \cdot \bar{h}_{in_1}^R, \text{ где } \bar{h}_{in_1}^R = C_h^{in} \cdot \bar{h}_{in_1}^R;$$

Итак, искомые измерения имеют вид

$$z_{1_ESG} = z_{_avt} = \Delta h_{int2_2}^R - \Delta(\cos \theta^S);$$

$$z_{2_ESG} = z_{1_2} = \Delta h_{int1_2}^R - \Delta h_{int1_2}^{et};$$

где $\Delta(\cos \theta^S) = \sin \theta \cdot \Delta h_{int3_1}^S + \Delta h_{int2_2}^S$,

$$\Delta h_{int_1}^S = \begin{bmatrix} \Delta h_{int1_1}^S \\ \Delta h_{int2_1}^S \\ \Delta h_{int3_1}^S \end{bmatrix};$$

$$\Delta \bar{h}_{int_1}^S = C_{int}^h \cdot \delta C_h^c \cdot \bar{h}_{in_1}^R - C_{int}^{in} \cdot \delta C_h^{in} \cdot \bar{h}_{in_1}^R;$$

$\Delta h_{int2_2}^S$ – это шумы измерений (погрешности $\Delta \bar{h}_{kn_2}^S$ СУ реального ЭСГ-2 в проекциях на ось int2, полагая, что погрешности «геометрии» ЭСГ-2 в проекциях на ось int2 малы); $\Delta \bar{h}_{int1_2}^{et}$ – погрешности формирования эталонных значений орта ЭСГ₂ по внешним данным, полученные из векторного соотношения для

$$\Delta \bar{h}_{int_2}^{et} = \begin{bmatrix} \Delta h_{int1_2}^{et} \\ \Delta h_{int2_2}^{et} \\ \Delta h_{int3_2}^{et} \end{bmatrix};$$

Согласно алгоритму $\bar{h}_{int_2}^{et} = C_{int}^{in} \cdot C_{in}^h \cdot C_c^c \cdot C_c^{kn} \cdot \bar{h}_{kn_2}^S$, имеем:

$$\Delta \bar{h}_{int_2}^{et} = C_{int}^h \cdot \delta C_h^{in} \cdot \bar{h}_{in_2}^S - C_{int}^h \cdot \delta C_h^c \cdot \bar{h}_{in_2}^S + C_{int}^{kn} \cdot \Delta \bar{h}_{kn_2}^S.$$

Отметим, что вблизи точки линеаризации

$$\Delta \bar{h}_{int3_2} = -(\cos \theta / \sin \theta) \cdot \Delta \bar{h}_{int2_2};$$

$$\Delta \bar{h}_{in_2} = (C_{int}^{in})^T [\Delta \bar{h}_{int1_2}; \Delta \bar{h}_{int2_2}; \Delta \bar{h}_{int3_2}].$$

Линеаризованная расчетная модель погрешностей

Таким образом, линеаризованная упрощенная расчетная модель погрешностей интегрированной системы и измерения может быть представлена в следующем виде

$$x_{k+1} = \Phi_{k+1/k} \cdot x_k + \Gamma_{k+1} \cdot w_k,$$

$$z_{k+1} = H_{k+1} \cdot x_{k+1} + v_{k+1},$$

где $x^T = [\Delta h_{int1_2}^R \quad \Delta h_{int2_2}^R \quad \Delta \omega_{int1} \quad \Delta \omega_{int2} \quad \Delta \omega_{int3} \quad \Delta h_{kn1} \quad \Delta h_{kn2} \quad \Delta h_{kn3}]$ – вектор состояния;

$\Phi_{k+1/k} \cong E_{n \times n} + F(t_k) \cdot dT + \frac{1}{2} \cdot [F(t_k) \cdot dT]^2$ – переходная на шаге dT матрица динамической системы; $F(t_k)$ – матрица динамики системы; $\Gamma_{k+1} \cong \Phi_{k+1} \cdot dT$ – матрица, определяющая влияние вектора входных шумов w_k с ковариациями Q_k ; H – матрица измерений.

Формирование переходной матрицы $\Phi_{k/k+1}$ состояния системы для интервала Tz :

j – оператор дискретности на рабочей частоте; $\Delta t_j = t_{j+1} - t_j = dT$;

k – оператор дискретности на частоте измерений; $\Delta t_k = t_{k+1} - t_k = Tz$;

$$\Phi_{k+1/k} \cong E_{n \times n} + F(t_k) \cdot dT + \frac{1}{2} \cdot [F(t_k) \cdot dT]^2;$$

$$\Phi_{k/j+1} = \Phi_{j/j+1} \cdot \Phi_{k/j};$$

при $j = k$

$\Phi_{k/j+1} = \Phi_{k/k+1}$; – искомое значение переходной матрицы $\Phi_{k/k+1}$;

$$\Phi_{k/j=0} = E(8 \times 8);$$

Ковариационный канал ФК:

$$P_{k/k+1} = \Phi_{k/k+1} P_k \Phi_{k/k+1}^T + \Gamma_{k/k+1} Q_k \Gamma_{k/k+1}^T,$$

$$K_{k+1} = P_{k/k+1} H_{k+1}^T [H_{k+1} P_{k/k+1} H_{k+1}^T + R_{k+1}]^{-1},$$

$$P_{k+1} = [E_{n \times n} - K_{k+1} H_{k+1}] P_{k/k+1} [E_{n \times n} - K_{k+1} H_{k+1}]^T + K_{k+1} R_{k+1} K_{k+1}^T.$$

$$(P_{k+1} = [E_{n \times n} - K_{k+1} H_{k+1}] P_{k/k+1}).$$

Оценочный канал ФК

$\hat{\epsilon}_{k+1} = \Phi_{k/k+1} \hat{\epsilon}_k + K_{k+1} z_{k+1}$ – для составляющих вектора состояния, оценки которых на каждом шаге подаются в обратной связи в выходные сигналы соответствующих измерителей [2].

Численное моделирование осуществлялось в среде MATLAB с использованием пакета SIMULINK. Начальные значения погрешностей принимались следующие:

По углам качки (случайные и систематические составляющие погрешностей бескарданного инерциального измерительного модуля):

$$\Delta \psi = 0,3 \text{ град.}, \Delta \bar{\psi} = 0,3 \text{ град.};$$

$$\Delta \theta = 0,3 \text{ град.}, \Delta \bar{\theta} = 0,3 \text{ град.};$$

По курсу (ПА СНС):

$$\Delta K = 0,5 \text{ град.}, \Delta \bar{K} = 0,3 \text{ град.};$$

При оценивании суммарных дрейфов электростатического гироскопа время готовности системы снижается до получаса, при этом точность выставки вектора кинетического момента ЭСГ имеет порядок меньше 0,4 град. Что касается роста суммарных дрейфов в режиме без привлечения внешних данных о курсе, то это предмет рассмотрения автономного режима работы, который будет исследован в ближайшее время.

Литература

1. Анучин О.Н., Гусинский В.З., Емельянцеv Г.И., Литманович Ю.А. Начальная выставка и калибровка бескарданного измерительного модуля для систем управления движением судна// Гироскопия и навигация. – 1994. – № 4. – С. 7–13.
2. Анучин О.Н., Емельянцеv Г.И., Бесплатформенные инерциальные системы навигации и ориентации (БИНС и БИСО). Учебное пособие. – СПб: СПбГУ ИТМО. – 1995. – 110 с.
3. Пешехонов В.Г. Ключевые задачи современной автономной навигации// Гироскопия и навигация. – 1996. – №1. – С. 48–55.
4. Анучин О.Н., Емельянцеv Г.И., Интегрированные системы ориентации и навигации морских подвижных объектов. – 2-е изд., перераб. и доп. – СПб.: ГНЦ РФ – ЦНИИ «Электронприбор». – 2003. – 390 с.

АППРОКСИМАЦИЯ МНОГОМЕРНОЙ ЗАВИСИМОСТИ С ПОМОЩЬЮ ГРАДИЕНТНОГО УСИЛЕНИЯ РЕГРЕССИОННОЙ НЕЙРОННОЙ СЕТИ И ЕЕ ПРИМЕНЕНИЕ В БЫСТРОМ МЕТОДЕ АЭРОДИНАМИЧЕСКОГО РАСЧЕТА ДЛЯ ЗАДАЧ ПРОЕКТИРОВАНИЯ

М.Г. Беляев, П.В. Приходько

**(Московский физико-технический институт (государственный университет)),
Е.В. Бурнаев (Институт проблем передачи информации им. А.А. Харкевича РАН)
Научный руководитель – д.ф.-м.н., профессор А.В. Бернштейн
(Институт системного анализа РАН)**

Разработан алгоритм оценки параметров Искусственной Нейронной Сети (ИНС), который имеет меньшее по сравнению со стандартными алгоритмами обучения время работы, а также позволяет уменьшить влияние недостатков стандартных алгоритмов обучения ИНС. Предлагаемый алгоритм использует комбинацию регрессионных методов и эластичного распространения ошибки с последующим построением ансамбля из ИНС на основе градиентного усиления.

Ключевые слова: аппроксимация, многомерная зависимость, искусственная нейронная сеть, линейная регрессия, бустинг, аэродинамические характеристики самолета

Введение

Задача определения аналитического описания множества данных возникает во многих разделах науки и связана с моделированием данных и/или идентификацией систем. Искусственные нейронные сети (ИНС) являются удобным средством моделирования данных, поскольку представляют собой универсальные аппроксиматоры, которые можно настроить для описания данных путем обучения на примерах [12, 11]. За счет параллельной структуры ИНС легко обучаются отображению многомерных нелинейных функций.

Как уже упоминалось, задача моделирования многомерной зависимости очень распространена. Например, в [6] рассмотрено применение ИНС для прогнозирования зависимости свойств металла от его состава и структуры. В [7] ИНС используется для аппроксимации зависимостей аэродинамических характеристик самолета от его компоновки и параметров режима полета. В [8] ИНС применяется для идентификации и управления динамических систем за счет отображения представления входа-выхода неизвестной системы и, в некоторых случаях, закона управления этой системы.

Однако в указанных выше работах для аппроксимации используются ИНС с весьма простой структурой, например, двухслойные ИНС, у которых передаточная функция нейронов первого скрытого слоя представляет собой гиперболический тангенс, а передаточная функция нейронов второго скрытого слоя является линейной. Для краткости такие ИНС будем называть простыми.

Целью работы является описание общей методологии получения аппроксимации многомерной зависимости, которая бы превосходила по точности и скорости построения методы, основанные на простых ИНС и использующие стандартные алгоритмы обучения с помощью градиентного спуска. Предложенная методология, успешно примененная при решении задачи предсказания аэродинамических характеристик, состоит в совместном использовании технологии нейронных сетей, различных регрессионных методов и методов объединения ансамблей искусственных нейронных сетей с помощью градиентного усиления.

Постановка задачи аппроксимации многомерных зависимостей

Рассмотрим некоторую многомерную зависимость

$$y = f(\mathbf{X}), \quad \mathbf{X} \in R^M (M \gg 1), \quad y \in R^1. \quad (1)$$

Пусть известна выборка $\Sigma = \{y_i, \mathbf{X}_i\}_{i=1}^N$ значений зависимости (1). Случайным образом разделим выборку Σ на две части $\Sigma = \Sigma_{estim} \cup \Sigma_{test}$, где объем подвыборки Σ_{test} составляет некоторую долю (обычно 10–20%) от объема выборки Σ . Задача аппроксимации многомерной зависимости (1) заключается в том, чтобы, используя подвыборку Σ_{estim} , построить такую многомерную зависимость $y = \tilde{f}(\mathbf{X})$, $\mathbf{X} \in R^M$, $y \in R^1$, для которой некоторая функция ошибки $D(f, \tilde{f} | \Sigma_{test}) \leq \varepsilon$, где ε – заданное число. В качестве функции ошибки часто используют функцию вида $D(f, \tilde{f} | \Sigma_{test}) = \sqrt{\sum_{\{y, \mathbf{X}\} \in \Sigma_{test}} (y - \tilde{f}(\mathbf{X}))^2}$. Методология построения аппроксимаций многомерных зависимостей состоит из нескольких частей, каждая из которых описана в разделе с соответствующим названием.

Гибридный алгоритм обучения нейронной сети

В основе разработанного аппроксиматора многомерной зависимости лежит простая ИНС. Для обучения простой ИНС обычно используется алгоритм обратного распространения ошибки [2]. Известно, что этот метод зачастую медленно сходится. Были разработаны методы, позволяющие ускорить процесс обучения, например, методы второго порядка, методы с адаптивным изменением скорости обучения и т.д. [1, 2, 10]. В данном разделе описывается гибридный алгоритм обучения простой ИНС, сочетающий в себе как методы градиентного спуска, так и методы регрессионного анализа. Применение методов регрессионного анализа для оценки весов выходного слоя позволяет существенно сократить время обучения по сравнению со временем, которое требуется при применении стандартных методов обучения. Этот эффект объясняется тем, что на каждой итерации гибридного алгоритма найденные значения весов выходного слоя являются оптимальными.

Опишем, каким образом регрессионный анализ можно использовать для оценки весов выходного слоя в простой ИНС. Простая ИНС реализует следующее преобразование вектора \mathbf{X}

$$y_{ANN}(\mathbf{X}) = \sigma(\mathbf{X} \cdot \mathbf{W}^T + \mathbf{d}) \cdot \mathbf{V} + b, \quad (2)$$

где $\mathbf{X} \in R^M$ – входной вектор-строка, $\mathbf{W} \in R^{m \times M}$ и $\mathbf{d} \in R^m$ – матрица весов и вектор-строка свободных членов скрытого слоя соответственно, $\mathbf{V} \in R^m$ и $b \in R^1$ – вектор-столбец весов и свободный член выходного слоя соответственно, $\sigma(x) = 1/(1 + e^{-x})$ – сигмоидная функция активации, действующая на каждую из компонент вектора-строки $\mathbf{X} \cdot \mathbf{W}^T + \mathbf{d}$ по отдельности, m – количество нейронов в скрытом слое.

Разобьем случайным образом обучающее множество Σ_{estim} на множества S_{train} и S_{val} . Первое множество будем использовать для оценки параметров ИНС, а второе – для контроля обобщающей способности ИНС. Веса простой ИНС определяются как решение задачи минимизации

$$E = E(\mathbf{W}, \mathbf{d}, \mathbf{V}, b) = \sum_{\{y, \mathbf{X}\} \in S_{train}} (y - y_{ANN}(\mathbf{X}))^2 \rightarrow \min_{\mathbf{W}, \mathbf{d}, \mathbf{V}, b}. \quad (3)$$

Допустим, что значения весов \mathbf{W} и \mathbf{d} известны. Так как функция $y_{ANN}(\mathbf{X})$ в (2) линейно зависит от весов \mathbf{V} и b , то значения этих весов, для которых величина (3) принимает минимальное значение при каких-то фиксированных \mathbf{W} и \mathbf{d} , могут быть

найлены с помощью метода наименьших квадратов. Пусть $N_{train} = \#(S_{train})$ – объем обучающей выборки S_{train} , $\mathbf{Q} = \mathbf{Q}(S_{train}) \in R^{N_{train} \times m}$ – матрица, строки которой равны $\sigma(\mathbf{X} \cdot \mathbf{W}^T + \mathbf{d})$, $\mathbf{X} \in S_{train}$, $\mathbf{Y} \in R^{N_{train}}$ – вектор-столбец, составленный из значений $y \in S_{train}$, $\bar{y} = \frac{1}{N_{train}} \sum_{y \in S_{train}} y$, $\bar{\mathbf{Q}}$ – вектор-строка, элементы которой равны

выборочным средним, подсчитанным по столбцам матрицы \mathbf{Q} , $\tilde{\mathbf{Q}}$ – «центрированная» матрица \mathbf{Q} (из каждого элемента каждого столбца матрицы \mathbf{Q} было вычтена соответствующая компонента $\bar{\mathbf{Q}}$), аналогично, $\tilde{\mathbf{Y}}$ – «центрированный» вектор \mathbf{Y} . В таком случае

$$\mathbf{v} = (\tilde{\mathbf{Q}}^T \cdot \tilde{\mathbf{Q}})^{-1} \cdot (\tilde{\mathbf{Q}})^T \cdot \tilde{\mathbf{Y}}, \quad b = \bar{y} - \bar{\mathbf{Q}} \cdot \mathbf{v}. \quad (4)$$

Для подстройки весов \mathbf{W} и \mathbf{d} будем использовать градиентный спуск. Как следует из формулы (4), веса выходного слоя теперь являются функциями весов скрытого слоя. Однако оказывается, что эту зависимость можно не учитывать при подсчете градиента функции ошибки (3).

Утверждение. При подсчете градиента функции ошибки (3) можно не учитывать функциональную зависимость весов выходного слоя от весов скрытого слоя.

Доказательство.

С учетом уравнения (2) функция ошибки принимает вид

$$E = \sum_{\{y, \mathbf{x}\} \in S_{train}} (\sigma(\mathbf{X} \cdot \mathbf{W}^T + \mathbf{d}) \cdot \mathbf{v} + b - y)^2 \rightarrow \min_{\mathbf{w}, \mathbf{d}, \mathbf{v}, b}$$

Найдем градиент функции ошибки по весам скрытого слоя \mathbf{W} :

$$\begin{aligned} \frac{\partial E}{\partial \mathbf{W}} &= 2 \cdot \sum_{\{y, \mathbf{x}\} \in S_{train}} (y_{ANN}(\mathbf{X}) - y) \cdot \frac{\partial \sigma(\mathbf{X} \cdot \mathbf{W}^T + \mathbf{d})}{\partial \mathbf{W}} \cdot \mathbf{v} \\ &+ 2 \cdot \sum_{\{y, \mathbf{x}\} \in S_{train}} (y_{ANN}(\mathbf{X}) - y) \cdot \left(\sigma(\mathbf{X} \cdot \mathbf{W}^T + \mathbf{d}) \cdot \left(\frac{\partial \mathbf{v}}{\partial \mathbf{W}} \right) + \frac{\partial b}{\partial \mathbf{W}} \right). \end{aligned} \quad (5)$$

Покажем, что

$$\sum_{\{y, \mathbf{x}\} \in S_{train}} (y_{ANN}(\mathbf{X}) - y) \cdot \sigma(\mathbf{X} \cdot \mathbf{W}^T + \mathbf{d}) \cdot \left(\frac{\partial \mathbf{v}}{\partial \mathbf{W}} \right) \equiv 0$$

Для этого найдем градиент функции ошибки по весам выходного слоя \mathbf{v}

$$\frac{\partial E}{\partial \mathbf{v}} = 2 \cdot \sum_{\{y, \mathbf{x}\} \in S_{train}} (y_{ANN}(\mathbf{X}) - y) \cdot \sigma(\mathbf{X} \cdot \mathbf{W}^T + \mathbf{d}) \equiv 0, \quad (6)$$

$$\frac{\partial E}{\partial \mathbf{v}} \equiv 0$$

поскольку по определению метода наименьших квадратов.

Тогда с учетом полученного тождества (6) упростим уравнение (5), обнулив второе слагаемое в скобках:

$$\frac{\partial E}{\partial \mathbf{W}} = 2 \cdot \sum_{\{y, \mathbf{x}\} \in S_{train}} (y_{ANN}(\mathbf{X}) - y) \cdot \left(\frac{\partial \sigma(\mathbf{X} \cdot \mathbf{W}^T + \mathbf{d})}{\partial \mathbf{W}} \cdot \mathbf{v} + \frac{\partial b}{\partial \mathbf{W}} \right). \quad (7)$$

Аналогично, из равенства нулю производной функции ошибки по b следует, что

$$\frac{\partial E}{\partial \mathbf{W}} = 2 \cdot \sum_{\{y, \mathbf{x}\} \in S_{train}} (y_{ANN}(\mathbf{X}) - y) \cdot \left(\frac{\partial \sigma(\mathbf{X} \cdot \mathbf{W}^T + \mathbf{d})}{\partial \mathbf{W}} \cdot \mathbf{v} \right). \quad (8)$$

Используя те же рассуждения, получим аналогичную формулу для подсчета градиента функции ошибки по свободному члену скрытого слоя:

$$\frac{\partial E}{\partial \mathbf{d}} = 2 \cdot \sum_{\{y, \mathbf{X}\} \in S_{train}} (y_{ANN}(\mathbf{X}) - y) \cdot \left(\frac{\partial \sigma(\mathbf{X} \cdot \mathbf{W}^T + \mathbf{d})}{\partial \mathbf{d}} \cdot \mathbf{v} \right) \quad (9)$$

Из формул (8) и (9) следует утверждение.

Таким образом, гибридный алгоритм обучения простой ИНС с учетом доказанного выше утверждения сводится к следующей последовательности действий.

- (а) Инициализируются веса скрытого слоя случайным образом (с помощью алгоритма Нгуена-Видроу [9] и подсчитываются веса выходного слоя с помощью метода наименьших квадратов.
- (б) Рассчитывается градиент функции ошибки по формулам (8) и (9).
- (в) Веса скрытого слоя обновляются согласно алгоритму эластичного распространения ошибки [10].
- (г) Веса выходного слоя вычисляются по формулам (4).
- (д) Если за заданное число последовательных итераций алгоритма ошибка на множестве S_{val} не опустилась ниже ранее достигнутого минимального значения и/или выполнен один из определенных в алгоритме эластичного распространения ошибки критериев останова, то обучение прекращается. В противном случае происходит переход к пункту (б) алгоритма.

Ансамбль аппроксиматоров

Рассмотренный выше алгоритм обучения ИНС имеет преимущества в скорости обучения и точности аппроксимации по сравнению с традиционными алгоритмами обучения [1, 2, 10]. Однако этому алгоритму также свойственны некоторые недостатки стандартных алгоритмов обучения ИНС – результаты работы алгоритма зависят от случайной начальной инициализации, для подстройки параметров используются не все имеющиеся выборочные данные. Для решения этой проблемы предлагается использовать мета-алгоритм градиентного усиления для построения ансамбля из аппроксиматоров (ИНС), основанный на идеях методов бустинга (boosting) и беггинга (bagging), предложенных для задачи регрессии в [3, 4, 5]. Приведем описание алгоритма градиентного усиления:

(1) Положим начальное значение выхода ансамбля равным $F_0(\mathbf{X}) = 0$

(2) Для $B = 1, 2, \dots$:

- (а) Определим новую выборку данных, равную $\Sigma_{estim, B} = \{y^B, \mathbf{X}\}$, где $y^B = B \cdot y - (B-1) \cdot F_{B-1}(\mathbf{X})$ при $\{y, \mathbf{X}\} \in \Sigma_{estim}$. Разобьем случайным образом обучающее множество $\Sigma_{estim, B}$ на множества $S_{train, B}$ и $S_{val, B}$.
- (б) Обучим по выборке $\Sigma_{train, B} = \{y^B, \mathbf{X}\}$, используя гибридный алгоритм из предыдущего раздела, B -й аппроксиматор ансамбля $f_B(\mathbf{X})$. Для подстройки параметров будем использовать множество $S_{train, B}$. Остановка алгоритма обучения будет происходить, если за заданное число последовательных итераций алгоритма ошибка на множестве $S_{val, B}$ не опустилась ниже ранее достигнутого минимального значения и/или выполнен один из определенных в алгоритме эластичного распространения ошибки критериев останова.
- (в) Полагаем значение выхода ансамбля равным $F_B(\mathbf{X}) = ((B-1) \cdot F_{B-1}(\mathbf{X}) + f_B(\mathbf{X})) / B$.
- (г) Остановка алгоритма градиентного усиления происходит в том случае, если за заданное число последовательных итераций алгоритма значение функции

ошибки $D(f, F_B | \Sigma_{train})$ не опустилось ниже ранее достигнутого минимального значения.

Использование ансамбля ИНС, построенных на основе предложенного алгоритма позволяет подстроиться под все обучающее множество данных Σ_{estim} и уменьшить случайный разброс результатов обучения, связанный со случайной инициализацией весов, поскольку в силу специфики алгоритма каждый аппроксиматор будет исправлять ошибки предыдущих аппроксиматоров ансамбля.

Некоторое улучшение при предложенном подходе по сравнению с применением только Boost.Reg (оригинального алгоритма бустинга, предложенного в [4]) для исследуемых данных подтверждается приведенными на рисунке результатами эксперимента по аппроксимации аэродинамических характеристик самолета.

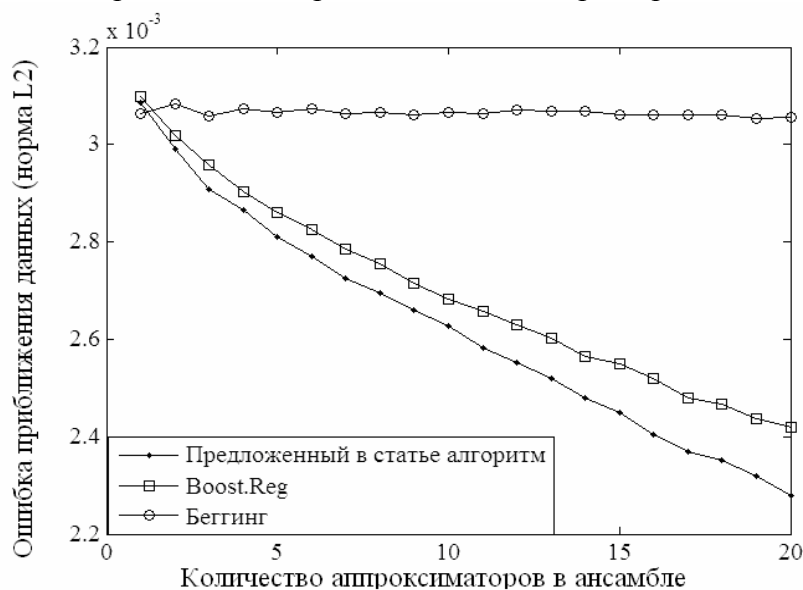


Рисунок. Сравнение различных алгоритмов построения ансамблей

Экспериментальные данные

Для проверки эффективности алгоритма были поставлены эксперименты по аппроксимации аэродинамических характеристик самолета в зависимости от его компоновки. В таблице приведены различные ошибки приближения данных для аппроксиматора, построенного по разработанной методологии, и простой ИНС.

Таблица. Ошибки приближения аэродинамических данных

		ИНС		Предложенный алгоритм	
		Обучающее множество	Проверочное множество	Обучающее множество	Проверочное множество
Абсолютные ошибки	по норме L2	0.000410	0.000559	0.000210	0.000310
	максимальная	0.005780	0.005901	0.004612	0.004774
	квантиль 99%	0.001161	0.001761	0.000628	0.000902
	квантиль 95%	0.000780	0.001017	0.000404	0.000591
Относительные ошибки (%)	по норме L2	3.09	3.75	1.58	2.21
	максимальная	24.24	27.55	23.23	25.88
	квантиль 99%	2.29	2.78	1.15	1.61
	квантиль 95%	2.29	2.78	1.15	1.61

Данные были разделены на два множества в отношении четыре к одному. На большем множестве (обучающем множестве) аппроксиматоры обучались, а на меньшем (проверочном множестве) проверялась их эффективность. Из таблицы видно, что предложенный алгоритм превосходит ИНС по всем параметрам, имея при это меньшее время обучения. На обучение одной регрессионной нейронной сети (с помощью гибридного алгоритма) было затрачено в среднем 85 секунд, в то время как на обучение ИНС такого же размера методом эластичного распространения ошибки [10] тратится в среднем 278 секунд.

Заключение

В работе представлена методология построения многомерных зависимостей, являющаяся развитием методов построения аппроксиматоров на основе ИНС и улучшения их работы с помощью бустинга. Эта методология позволяет уменьшить время обучения аппроксиматора, уменьшить зависимость результатов от случайного разбиения данных на обучающее и проверочное множество и от случайной начальной инициализации.

Работа выполнена при частичной поддержке гранта РФФИ 09-01-00584-а.

Литература

1. Тархов Д.А. Нейронные сети. Модели и алгоритмы. Кн. 18. – М.: Радиотехника. – 2005. – 256 с.
2. Хайкин С. Нейронные сети. Полный курс. – М.: Вильямс. – 2006. – 1104 с.
3. Breiman L. Bagging Predictors // Technical Report № 421, Department of Statistics, University of California. – 1994.
4. Friedman J.H. Greedy function approximation: a gradient boosting machine // Technical Report, Department of Statistics, Stanford University. – 1999.
5. Friedman J.H., Hastie T., Tibshirani R. Additive logistic regression: a statistical view of boosting // Technical Report, Department of Statistics, Stanford University. – 1999.
6. Gavard L., Bhadeshia H K.D.H., MacKay D.J.C., Suzuki S. Bayesian neural network model for austenite formation in steels // Mater. Sci. Technol. – 1996. – № 12. – P. 453–463.
7. Linse D., Stengel R.F. Identification of aerodynamic coefficients using computational neural networks // J. Guid. Control Dyn. – 1993. – Vol. 16. – № 6. – P. 1018–1025.
8. Narendra K.S., Parthasarantly K. Identification and control of dynamical systems using neural networks // IEEE Trans. Neural Netw. – 1990. – Vol. 1. – № 1. – P. 4–27.
9. Nguyen D., Widrow B. Improving the Learning Speed of 2-Layer Neural Networks by Choosing Initial Values of the Adaptive Weights // Proc. Intl. Joint Conf. on Neural Networks, San Diego, CA. – 1990. – Vol. III. – P. 21–26.
10. Riedmiller M., Braun H. A direct adaptive method for faster backpropagation learning: the RPROP algorithm // Proceedings of the IEEE International Conference on Neural Networks. – 1993.
11. Sutton R.S., Barto A.G. Reinforcement learning. – Cambridge MA: MIT Press. – 1998.
12. Werbos R.J. Neurocontrol and supervised learning: an overview and evaluation // Handbook of Intelligent Control / Ed. by D. A. White, D. A. Sofge. – New York: Van Nostrand. – 1992. – P. 65–86.

АЛГОРИТМЫ ПОСТРОЕНИЯ ОБЛАСТИ ГЕНЕРАЦИИ МНОГОМЕРНЫХ ОБЪЕКТОВ

А.А. Маланов

(Московский физико-технический институт (государственный университет)),

Д.Е. Ветров

(Международный научно-исследовательский институт проблем управления)

Научный руководитель – д.ф.-м.н., профессор А.В. Бернштейн

(Институт системного анализа РАН)

При проведении массовых расчетов необходимо иметь большое количество исходных данных. Для этого должна быть решена задача генерации объектов, описания которых удовлетворяли бы ограничениям и требованиям некоторой модели. В работе предложен новый зависящий от данных алгоритм построения области генерации многомерных объектов, приводятся результаты вычислительных экспериментов, демонстрирующих свойства предложенного алгоритма.

Ключевые слова: когнитивные технологии, снижение размерности, генерация многомерных данных

В процессе проектирования и создания сложных технических объектов рассматриваются и сравниваются различные технические решения. Эти решения касаются структуры объектов, механизмов их функционирования, выбора параметров и других элементов объекта. Для сравнения решений и выбора оптимального решения создаются основанные на знаниях системы поддержки принятия инженерных решений, в основе которых лежит широкий спектр различных когнитивных технологий.

Одной из таких востребованных технологий является когнитивная технология быстрых расчетов [1–2], позволяющая строить основанные на данных так называемые суррогатные адаптивные модели. Построенные модели фактически имитируют не только источники получения данных об объекте, основанные на некоторой исходной полноразмерной или упрощенной модели, но и сами модели, созданные на основе изучения физических феноменов, описывающих процессы функционирования объектов.

Технология быстрых расчетов предлагает схему исследования поведения объекта, состоящую из нескольких этапов.

Этап 1. На первом этапе строится модель изучаемого объекта. Например, характеристика объекта z , определяющая свойства объекта в некоторых условиях, может быть описана в виде функциональной зависимости $z = F(x, y)$, где переменная x описывает сам объект, а переменная y задает условия функционирования объекта (параметры управления объектом, параметры внешней среды).

Этап 2. Функция F является неизвестной, и для ее вычисления проводятся натурные или вычислительные эксперименты. В результате формируется некоторое количество измерений $\Phi = \{X=\{x_i\}, Y=\{y_i\}, Z=\{z_i\}, i=1,2,\dots\}$, где значение $z_i = F(x_i, y_i)$ исследуемой характеристики получено для объекта, имеющего описание x_i в условиях функционирования y_i . Предполагается, что имеющиеся измерения имеют приемлемую точность.

Этап 3. По известному множеству Φ с использованием тех или иных математических методов анализа и обработки данных строится функция $F_s(x, y)$, значения которой принимаются в качестве приближенного значения характеристики z для объекта с описанием x в условиях функционирования y . Такие функции принято называть

аппроксиматорами, поскольку они производят вычисления с некоторой погрешностью. Но если зависимости F_S и F близки друг к другу в некоторой метрике, то построенная функция F_S может рассматриваться как «заменитель» (суррогат) функции исходной функции F . Методы вычисления характеристик с использованием, таким образом, построенных функций носят название «суррогатные модели».

Этап 4. На этом этапе к построенной суррогатной модели применяют различные математические методы для решения экстремальных и оптимизационных задач. Экстремальные и оптимизационные задачи требуют проведения массовых расчетов. Сами аппроксиматоры строятся так, чтобы вычисления производились очень быстро. Но для проведения расчетов необходимо большое количество исходных данных. Другими словами, должна быть решена задача генерации объектов, описания которых удовлетворяли ограничениям и требованиям модели, построенной на этапе 1.

В работе приводится формальная постановка задачи генерации новых объектов, формулируются требования к формированию области генерации, предложен новый алгоритм построения области генерации сложной формы и приводятся результаты вычислительных экспериментов, демонстрирующих свойства предложенного алгоритма.

Базовые определения

Прежде чем перейти к формальной постановке задачи генерации многомерных объектов, введем несколько определений и понятий.

Модель описания объекта. Рассмотрим обобщенное описание объекта b , представленное вектором $W(b) = \{w_1, \dots, w_n\}$, состоящим из параметров, существенно влияющих на значение исследуемой характеристики $F(x, y)$. Значения этих параметров являются предметом принятия решений при выборе варианта построения объекта.

Для целей технологии вектор $W(b)$ должен иметь небольшую (порядка десятков) размерность n . Поэтому при построении аппроксиматоров первоочередной задачей является задача снижения размерности. В работе мы будем придерживаться формальной постановки задачи снижения размерности, предложенной в [3, 4].

Задача снижения размерности описывается набором $T = \{n, m, K, K^{-1}\}$, состоящим из следующих компонентов:

- n – размерность исходного вектора данных;
- m – размерность сжатого вектора данных;
- K – процедура сжатия данных $K: W \rightarrow K(W) \in \mathbf{R}^m$, которая по заданному вектору описания W размерности n строит сжатое описание $\lambda = \lambda(W)$ меньшей размерности m ;
- K^{-1} – процедура восстановления данных $K^{-1}: \lambda \rightarrow K^{-1}(\lambda) = W \in \mathbf{R}^n$, которая по заданному сжатому описанию λ восстанавливает «полноразмерный» вектор описания $W = W(\lambda)$.

Пусть есть множество $\Sigma = \{W(b_i), i=1,2,\dots,N\}$, состоящее из описаний объектов, для которых известно значение аппроксимируемой функции. В силу высокой размерности входного вектора, множество аргументов функции Σ является очень «разреженным» и не обладает какой-либо регулярной структурой типа «решетки». Поэтому при построении аппроксиматоров, как правило, исходные данные вначале сжимаются до меньшей размерности m с помощью некоторой процедурой сжатия K . Именно сжатые описания объектов и являются входными данными аппроксимирующей функции.

Формальная постановка задачи генерации многомерных объектов

Входными данными процедуры генерации описаний являются:

- выборка векторов $\Sigma = \{W_1, \dots, W_N\}$, где каждый вектор W_i есть вектор размерности n , представляющий собой описание объекта i ($i=1, \dots, N$);
- процедура снижения размерности $T = \{n, m, K, K^{-1}\}$;
- алгоритм генерации сжатого вектора описания (A) и параметры $\theta(A)$ выбранного алгоритма.

Выходом процедуры генерации является случайно сгенерированный вектор. Тем самым, процедура генерации профилей является процедурой генерации случайных объектов, и, следовательно, содержит внутри себя некоторый механизм формирования случайных чисел.

Генерация векторов проводится в несколько этапов.

Этап 1. Вычисляется множество m -мерных векторов

$$\Lambda_N = \{\lambda_1, \lambda_2, \dots, \lambda_N\} \subset \mathbf{R}^m, \quad (1)$$

состоящее из результатов применения преобразования сжатия K к векторам из множества Σ :

$$\lambda_i = K(W_i), \quad i = 1, 2, \dots, N.$$

Этап 2. Вычисляется m -мерный вектор λ

$$\lambda = A(\Lambda_N | \theta, \varepsilon),$$

являющийся результатом применения алгоритма $A = A(\cdot | \theta, \varepsilon)$ к множеству Λ_N .

Этап 3. Вычисляется полноразмерный вектор описания объекта

$$W = W(A, \theta, \varepsilon) = K^{-1}(\lambda),$$

являющийся результатом применения преобразования восстановления K^{-1} к сгенерированному сжатому вектору λ .

Алгоритм генерации сжатого вектора $A = A(\cdot | \theta, \varepsilon)$ в качестве входных данных использует:

- множество сжатых векторов Λ_N (1),
- параметры $\theta = \theta(A)$ алгоритма,
- значение случайного элемента ε .

и состоит из двух шагов

- Шаг 1. Построение области генерации,
- Шаг 2. Генерации в построенной области.

Построение области генерации

Естественное требование, предъявляемое к задаче генерации, состоит в том, чтобы новые объекты сохраняли физические свойства реальных объектов. Другими словами, необходимо, чтобы новые вектора генерировались в достаточной близости от реальных прототипов – множества векторов Σ в полноразмерном пространстве и множества векторов Λ в сжатом. Но при переходе к сжатому пространству теряется информация о физических особенностях объектов и трудно оценить свойства множества прототипов. Поэтому чаще всего в качестве области генерации выбирают:

- кубе, построенному по минимальным и максимальным значениям координат векторов из множества Λ ;
- некоторым образом построенную выпуклую оболочку множества векторов Λ .

Оба варианта не учитывают структуру множества прототипов Λ . В обоих случаях существуют конфигурации исходного множества в сжатом пространстве, данные методы построения областей оказываются неэффективными. На рис. 1 приведены примеры таких конфигураций для первого и второго варианта областей соответственно.

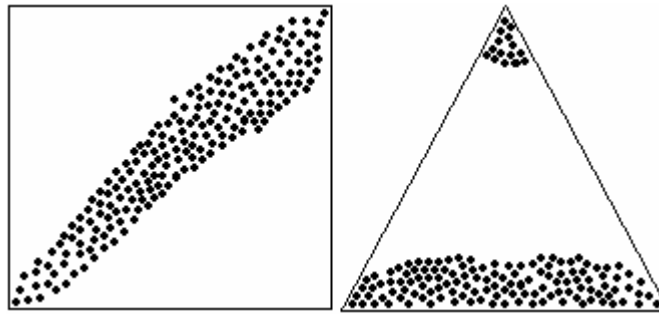


Рис. 1. Часто используемые области генерации

В обоих случаях часть сгенерированных точек будет приходиться на области, далеко отстоящие от прототипов.

Мы предлагаем новый алгоритм формирования области генерации, который существенно учитывает структуру множества прототипов.

Генерация сжатого вектора от одного прототипа

Рассмотрим случай, когда множество прототипов состоит из одного элемента. Тогда параметрами $\theta = (\lambda^*, d)$ алгоритма генерации сжатого вектора «от одного прототипа» являются:

- заданный вектор $\lambda^* \in \Lambda$;
- радиус окрестности d .

Случайный элемент $\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m)$ состоит из m независимых равномерно распределенных на отрезке $[0, 1]$ случайных чисел.

Областью генерации алгоритма является шар в \mathbf{R}^m :

$$S(\lambda^*, d) = \{\lambda \in \mathbf{R}^m: \|\lambda - \lambda^*\| \leq d\}. \quad (2)$$

Тогда генерируемый случайный вектор $\lambda = \lambda(\varepsilon) = \{\lambda_1(\varepsilon), \lambda_2(\varepsilon), \dots, \lambda_m(\varepsilon)\}$, вычисляемый по случайному элементу ε , равен:

$$\lambda_1(\varepsilon) = \lambda_1^* + \varepsilon_m \times d \times \sin(2\pi\varepsilon_1),$$

$$\lambda_2(\varepsilon) = \lambda_2^* + \varepsilon_m \times d \times \cos(2\pi\varepsilon_1) \times \sin(2\pi\varepsilon_2),$$

...

$$\lambda_k(\varepsilon) = \lambda_k^* + \varepsilon_m \times d \times \cos(2\pi\varepsilon_1) \times \cos(2\pi\varepsilon_2) \times \dots \times \cos(2\pi\varepsilon_{k-1}) \times \sin(2\pi\varepsilon_k),$$

...

$$\lambda_{m-1}(\varepsilon) = \lambda_{m-1}^* + \varepsilon_m \times d \times \cos(2\pi\varepsilon_1) \times \cos(2\pi\varepsilon_2) \times \dots \times \cos(2\pi\varepsilon_{m-2}) \times \sin(2\pi\varepsilon_{m-1}),$$

$$\lambda_m(\varepsilon) = \lambda_m^* + \varepsilon_m \times d \times \cos(2\pi\varepsilon_1) \times \cos(2\pi\varepsilon_2) \times \dots \times \cos(2\pi\varepsilon_{m-2}) \times \cos(2\pi\varepsilon_{m-1}).$$

Генерация сжатого вектора от множества прототипов

Параметрами $\theta = (\Lambda, d, s)$ алгоритма генерации сжатого вектора «от многих прототипов» являются:

- заданное множество сжатых прототипов $\Lambda = \{\lambda_1, \dots, \lambda_N\}$;

- радиус окрестности d ;
- решающее правило выбора прототипа.

Областью генерации алгоритма является объединение шаров в \mathbf{R}^m :

$$S(\Lambda, d) = \cup_{(i=1, \dots, N)} S(\lambda_i, d) = \{\lambda \in \mathbf{R}^m: \|\lambda - \lambda^*\| \leq d\}.$$

где $S(\lambda_i, d)$ есть шар (2) с центром в прототипе λ_i . В общем случае шары могут пересекаться.

Введем дополнительный набор характеристик области генерации. Для этого каждому прототипу λ_i , $i=1, \dots, N$, поставим в соответствие число s_i , равное количеству соседей (других прототипов из множества Λ), попавших в шар с центром в этом прототипе. Будем использовать построенный вектор $S = (s_1, \dots, s_N)$ в качестве дополнительного параметра в алгоритме генерации.

Алгоритм генерации состоит из 2-х шагов.

Шаг 1. При помощи решающего правила и вектора параметров S выбирается прототип λ_j .

Шаг 2. По прототипу λ_j и радиусу окрестности d генерируется случайный вектор при помощи алгоритма генерации от одного прототипа.

Примеры решающих правил для выбора прототипа

В качестве решающего правила при выборе прототипа было рассмотрено два правила: правило усиления плотности, правило выравнивания плотности.

В соответствии с правилом усиления плотности вероятность выбора вектора-прототипа λ прямо пропорциональна его характеристике $s(\lambda)$, равной количеству других прототипов, попавших в его d -окрестность. Другими словами, чем больше плотность d -окрестности некоторой точки в сжатом пространстве, тем чаще в ней генерируется новая точка.

Для данного правила был введен дополнительный параметр процедуры *Outlier*, который указывал, следует ли исключать из прототипов выбросы, то есть далеко отстоящие друг от друга точек сжатого пространства. Если параметра задан, то относительно каждого вектора $\lambda \in \Lambda$ вначале принимается решение, является ли он выбросом. Если вектор признан выбросом, то в его окрестности новые вектора не генерируются. Для остальных векторов вероятность их выбора прямо пропорциональна количеству соседей в их d -окрестности.

В случае выравнивания плотности механизм случайной генерации точки в сжатом пространстве работает так, что после генерации достаточно большого количества векторов, количество векторов в d -окрестности каждого из исходных векторов-прототипов должно стать приблизительно одинаковым.

Пример реализации разработанных алгоритмов

Предложенные алгоритмы были реализованы для решения задачи генерации аэродинамических профилей. Полноразмерное описание профиля было представлено вектором размерности 57. В качестве процедуры снижения размерности была выбрана процедура, построенная с помощью метода анализа главных компонент. Размерность сжатого пространства была выбрана равная 6. Множество прототипов состояло из 200 аэродинамических профилей.

На рисунках ниже представлены результаты вычислительных экспериментов. На

рис. 2 показано множество Λ сжатых векторов-прототипов, представленное в виде проекции на плоскость, образованную первыми двумя координатными осями. Рис. 3 показывает результаты генерации 200 новых векторов (черные квадратики – прототипы, серые точки – сгенерированные вектора). На левой части рис. 3 представлены результаты генерации с использованием правила выравнивания плотности, а на правой – с использованием правила усиления плотности.

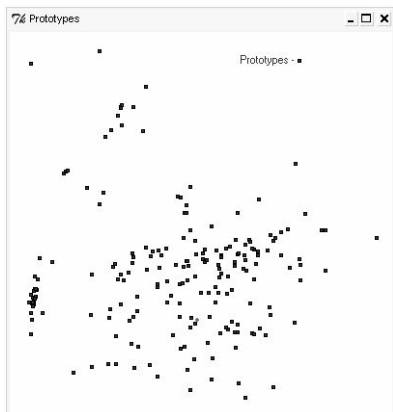


Рис. 2. Прототипы

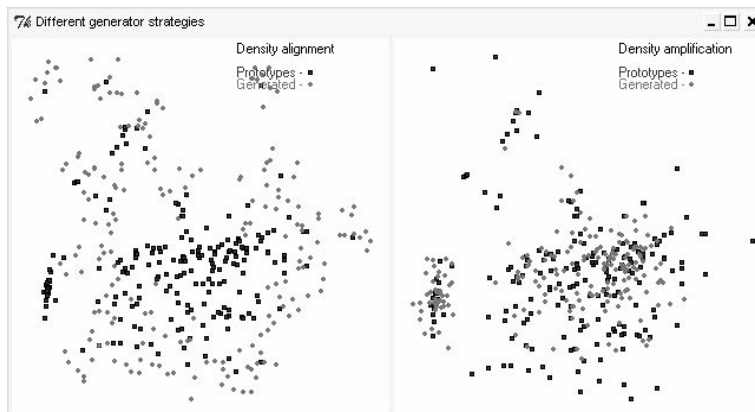


Рис. 3. Выравнивание и усиление плотности

Выводы

Предложенный алгоритм построения области генерации обладает следующими свойствами:

- существенно зависит от данных;
- учитывает структуру и плотность множества прототипов;
- гарантирует близость сгенерированного вектора к множеству прототипов, т.е. для каждого сгенерированного вектора в его d -окрестности найдется хотя бы один прототип.

Литература

1. Кулешов А.П. Информационные технологии в основанных на знаниях инженерных системах. // Материалы Международной научно-технической конференции «Информационные технологии и математическое моделирование систем 2006–2008». М.: Радиотехника. – 2008. – С. 98–100.
2. Кулешов А.П. Информационные технологии в проблеме создания моделей сложных объектов // Вторая Международная конференция «Системный анализ и информационные технологии» САИТ-2007 (10–14 сентября 2007 г., г. Обнинск, Россия): Труды конференции. – Т. 1. – 2007. – С. 14–16.
3. Кулешов А.П. Когнитивные технологии в основанных на данных адаптивных моделях сложных объектов // Информационные технологии и вычислительные системы. – Вып. 1. – 2008. – С. 95–106.
4. Кулешов А.П. Технология быстрого вычисления характеристик сложных технических объектов // Информационные технологии. Прил. – 2006. – № 3. – С. 4–11.

МОДЕЛИРОВАНИЕ ПРОМЫВКИ ГЛАЗА ПРИ ХИРУРГИИ КАТАРАКТЫ

К.Г. Вачнадзе

(Московский физико-технический институт (государственный университет))

Научный руководитель – к.ф.-м.н. А.А. Аксенов

(ООО «ТЕСИС»)

Данная работа посвящена моделированию течения эмульсии в передней камере глаза во время операции по лечению катаракты методом факоэмульсификации. Цель работы – уменьшение вероятности травмирования глаза путем оптимизации положений хирургических инструментов. Рассмотрены несколько видов факоэмульсификации катаракты на глазу, не оперированном ранее, и на глазу с имплантированной факичной интраокулярной линзой. В результате компьютерного моделирования получены распределения давления и скорости при промывании передней камеры глаза, исследована зависимость силы трения и давления на роговице от расположения хирургических инструментов, найдены оптимальные положения инструментов.

Ключевые слова: эмульсия, катаракта, факоэмульсификация, интраокулярная линза, компьютерное моделирование

1. Введение

Катаракта – помутнение хрусталика глаза. Существует ряд способов лечения катаракты. Одним из современных радикальных методов является лечение с помощью хирургического вмешательства методом факоэмульсификации [1]. Метод факоэмульсификации заключается в механическом дроблении хрусталика с помощью специальных инструментов, вводимых в переднюю камеру глаза через малые разрезы, и удалении полученных фрагментов вместе с внутриглазной жидкостью путём её аспирации. Одновременно осуществляется подача замещающей жидкости во избежание коллапса передней камеры под действием разряжения. Существуют два альтернативных метода факоэмульсификации: коаксиальная и бимануальная. При коаксиальной факоэмульсификации дробление, аспирация и ирригация осуществляются одним прибором, состоящим из двух коаксиальных трубок. При бимануальной факоэмульсификации используются аспирационный и ирригационный приборы, которые в процессе операции разносятся на некоторое расстояние и располагаются под углом друг к другу.

В хирургии катаракты существует ряд гидродинамических задач. Одна из них – уменьшение травмирующего гидродинамического воздействия на ткани глаза. Травма может вызываться повышенным давлением жидкости, трением на поверхности роговицы, засасыванием в аспирационную трубку задней капсулы хрусталика, коллапсом передней камеры глаза вследствие большого перепада внешнего и внутреннего давления. Бытующий в настоящее время эмпирический подход при конструировании офтальмологических инструментов, субъективизм в клинической оценке преимуществ тех или иных их видов, затрудняют создание средств, обеспечивающих минимальную травму тканей и наибольшую эффективность соответствующей техники лечения. Таким образом, анализ с помощью компьютерного моделирования альтернативных методик операций и конструкций применяемых инструментов является весьма актуальной проблемой. Её решение востребовано как практическими врачами, так и разработчиками хирургического инструментария.

Данная работа посвящена моделированию этапа промывки глаза в нескольких задачах: во время операции по лечению катаракты методом бимануальной факоэмульсификации и по лечению катаракты методом коаксиальной

факоэмульсификации. В последней задаче рассмотрены случаи с не оперированным ранее глазом и с имплантированной ранее искусственной линзой ФИОЛ (факичная интраокулярная линза). Цель работы – уменьшение вероятности травмирования глаза в центральной области роговицы путём оптимизации положений хирургических инструментов, сравнение полученных результатов в различных методиках операции.

2. Постановка задачи

В данной работе использовался программный комплекс FlowVision HPC [3], с помощью которого моделировалось ламинарное течение ирригационной жидкости. Ирригационная жидкость рассматривается как несжимаемая вязкая среда, находящаяся в условиях стационарного течения. Это предположение справедливо даже при наличии колебаний расхода ирригационной жидкости в широком диапазоне частот. Частоту f , начиная с которой следует учитывать нестационарные волновые процессы, можно оценить по формуле (1).

$$f = \frac{c}{L}, \quad (1)$$

где c – скорость звука в ирригационной жидкости, L – характерный размер области течения. Применительно к глазу: $L = 10$ мм, следовательно, пороговое значение f равно не менее 105 Гц, и рассматриваемые процессы укладываются в этот диапазон.

Будем рассматривать одну фазу – ирригационную жидкость с параметрами аналогичными параметрам внутриглазной жидкости: плотность $\rho = 1000$ кг/м³, вязкость $\nu = 0.014$ Па·с [2].

Определим число Рейнольдса, чтоб проверить, не является ли течение турбулентным:

$$Re = \frac{\rho \cdot L \cdot V}{\nu}, \quad (2)$$

где ρ – плотность жидкости, V – скорость течения, ν – коэффициент динамической вязкости, L – характерный размер. В случае истечения ирригационной жидкости, L – диаметр трубки, который может достигать до 3 мм. Тогда по формуле (2) имеем, что число Рейнольдса не превышает 90, что меньше критического значения $Re = 2300$, свойственного течением в трубах. Следовательно, ирригационный поток жидкости является ламинарным. В случае течений эмульсии в операционном поле, значение L имеет значение порядка 10 мм, тогда из формулы (2) $Re = 300$, что тоже далеко от порогового значения. Следовательно, в нашей задаче мы имеем дело только с ламинарными потоками жидкости.

В качестве модели будем использовать дифференциальные уравнения Навье-Стокса, которые описывают гидродинамику исследуемых потоков. Они будут решаться численными методами, с учётом имеющихся граничных условий. В результате будут определены искомые параметры течения (давление, скорость) в различных точках исследуемой области.

2.1. Метод бимануальной факоэмульсификации

В данном методе используются ирригационный и аспирационный хирургические инструменты. Расстановка граничных условий представлена на рис. 1. На поверхности передней камеры, хрусталика, хирургических инструментов задан тип границы «стенка» с граничным условием для скорости «прилипание» (т.е. равна нулю). На

торцевой поверхности ирригационного прибора задано относительное давление¹ 0 Па. На торцевой поверхности аспирационного прибора задана скорость откачки 0.4 м/с.

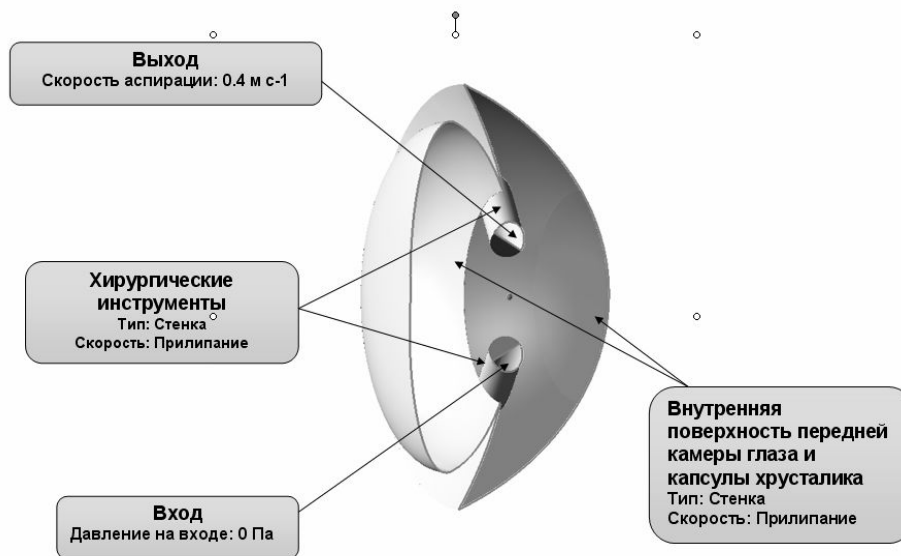


Рис. 1. Расстановка граничных условий (бимануальная факоемульсификация)

2.2. Метод коаксиальной факоемульсификации

В данном методе используется хирургический инструмент с коаксиальным наконечником, в котором совмещены функции аспирации и ирригации жидкости. Постановка граничных условий задачи не отличается от метода бимануальной факоемульсификации (рис. 2).

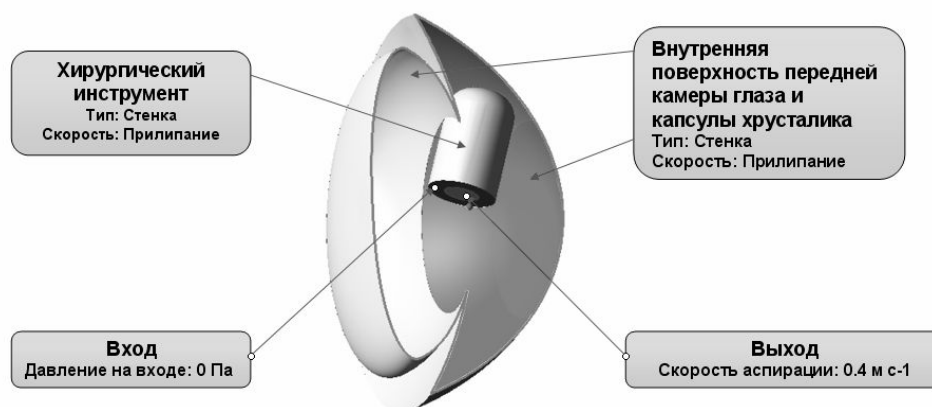


Рис. 2. Расстановка граничных условий (коаксиальная факоемульсификация)

2.3. Метод коаксиальной факоемульсификации под ФИОЛ

В данном методе рассматривается операция промывки глаза методом коаксиальной факоемульсификации на глазу, с ранее имплантированной искусственной линзой (ФИОЛ). Постановка граничных условий не отличается от предыдущих случаев (рис. 3).

¹ Относительное давление отсчитывается от абсолютного давления, равного 101000 Па (1атмосфере)

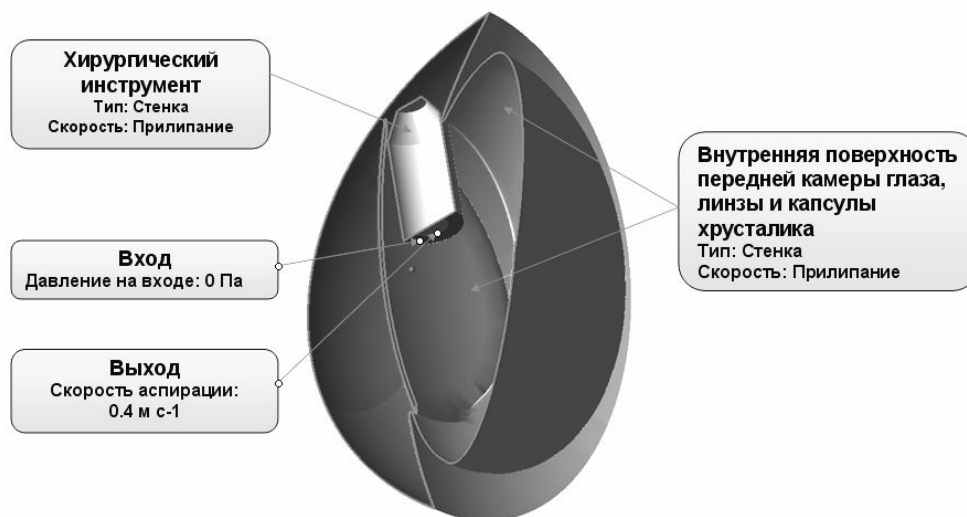


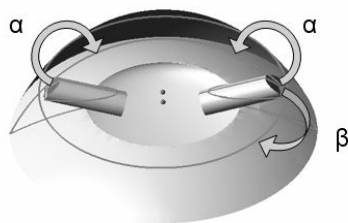
Рис. 3. Расстановка граничных условий (коаксиальная факоемульсификация под ФИОЛ)

3. Результаты моделирования

Расчётная сетка, используемая во Flowvision HPC, является ортогональной, адаптивно локально измельчённой и с подсеточным разрешением геометрии. Это позволяет адаптировать криволинейные поверхности прямоугольными сетками. В данной работе генерировалась определённая начальная сетка, которая затем адаптировалась в местах высоких градиентов переменных. Было проведено исследование сходимости по сетке. Рассматривались адаптации различных уровней и с различным числом слоёв. В качестве критерия сходимости по сетке было использовано усреднённое давление в центральной области роговицы. Количество расчётных ячеек оказалось равным 123877, 188882 и 155149 в методах бимануальной факоемульсификации, коаксиальной факоемульсификации и коаксиальной факоемульсификации под ФИОЛ соответственно.

3.1. Метод бимануальной факоемульсификации

Было проведено моделирование течения при варьировании различных параметров конфигурации приборов: углов перпендикулярных плоскости радужной оболочки глаза, углов параллельных плоскости радужки и глубины проникновения инструментов. Рассматривались 9 различных ситуаций, из которых были выделены наилучшие и наихудшие положения. В качестве критерия для выбора использовались давление на центральной поверхности роговицы (относительно давления на входе жидкости ирригационного прибора) и средняя скорость потока в центральной области роговицы (мера силы трения на роговице). Наилучший результат был получен, когда приборы располагались симметрично и были направлены в центр хрусталика. Наихудшие результаты были получены, когда инструменты располагались под небольшим углом друг к другу и имели различную глубину проникновения. Для дальнейшей оптимизации выбраны два параметра. Первый параметр – угол α в плоскости, перпендикулярной плоскости радужки. По нему оба инструмента будут симметрично двигаться в определённых пределах. Второй параметр – угол β в плоскости, параллельной плоскости радужки. Он будет зафиксирован у ирригационного прибора, а у аспирационного – варьироваться в заданных пределах (рис. 4).



- α от 16.6° до 23.6°
- β от 0° до 120°
- Минимизация силы трения на центральной поверхности роговицы

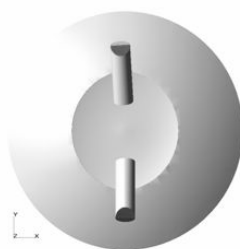
Рис. 4. Параметры для оптимизации

3.2. Метод коаксиальной факоэмульсификации

Было проведено моделирование течения при варьировании глубины проникновения коаксиального прибора. Было рассмотрено 5 различных положений инструмента. Отслеживалась средняя скорость потоков жидкости в центральной области роговицы и давления в центральной области роговицы (относительно давления на входе жидкости ирригационного прибора). Было установлено, что при уменьшении глубины проникновения инструмента, силы трения на роговицу со стороны жидкости линейно растут, а перепад давлений в глазу резко увеличивается при минимальной глубине проникновения.

3.3. Оптимизация положения приборов в методе бимануальной факоэмульсификации

С помощью программного комплекса IOSO NM [4] была проведена оптимизация положений приборов в методе бимануальной факоэмульсификации по выбранным параметрам α и β . Было рассмотрено около 100 различных вариантов и выделен наилучший с точки зрения минимальной средней скорости потоков жидкости в центральной области роговицы (минимальной силы трения). Минимальная скорость оказалась равной $0,0032$ м/с при следующих углах: $\alpha = 23^\circ$, $\beta = 3.4^\circ$ (рис. 5).



- $\alpha=23^\circ$
- $\beta=3.4^\circ$
- $V_{min}=0.0032$ м/с

Рис. 5. Результаты оптимизации

3.4. Определение времени операции промывки глаза

В данной задаче также было проведено исследование времени промывки глаза в методах бимануальной и коаксиальной факоэмульсификации. Промывка считалась завершённой в момент, когда все участки исследуемой области (передняя камера и

капсула хрусталика глаза) заполнялись ирригационной жидкостью, то есть осуществлялась их полная промывка. В результате исследования было получено, что время промывки в методе бимануальной факоэмульсификации равняется 554 секундам, а в методе коаксиальной факоэмульсификации (случай без линзы) – 771 секундам. Также был рассмотрен случай коаксиальной факоэмульсификации под ФИОЛ, в котором на момент времени 832 секунды были промыты полностью капсула хрусталика, а также труднодоступные для промывки участки глаза. Ирригационная жидкость проникала в переднюю камеру глаза, где в данный момент времени промывки не попадала в центральную область роговицы.

4. Заключение

В данной работе было проведено трехмерное численное моделирование операции факоэмульсификации процедуры промывки глаза при удалении катаракты. Были найдены оптимальные положения инструментов при бимануальной и коаксиальной факоэмульсификации, при которых сила трения на центральной поверхности роговицы минимальна. Также было проведено моделирование промывки передней полости глаза в ходе операции коаксиальной факоэмульсификации катаракты при ФИОЛ. Было получено, что время промывки при коаксиальной факоэмульсификации больше времени бимануальной.

Результаты этой работы были доложены в МНТК Микрохирургии глаза им Федорова и получили положительный отзыв. В дальнейшем планируется продолжать работу исследованием операции факоэмульсификации катаракты в глазу, перенесшем тотальную витрэктомию (с удалённым стекловидным телом), учитывая деформации тканей глаза вследствие течений жидкости при промывке.

Литература

1. Зимина Т.Ю., Спирочкин Ю.К., Плетнев И.В., Трубилин В.Н. О решении гидродинамических проблем в хирургии катаракты // Сборник трудов конференции «Биомеханика глаза». – 2005. – Т.5. – С. 179–186.
2. Малюгин Б.Э. Медико-технологическая система хирургической реабилитации пациентов с катарактой на основе ультразвуковой факоэмульсификации с имплантацией интраокулярной линзы // Диссертация на соискание учёной степени докт. мед. наук. – М.: ГУ МНТК «Микрохирургия глаза» им. акад. С.Н. Фёдорова. – 2002.
3. FlowVision HPC Руководство пользователя, Версия 3.06. – 2007.
4. Справка IOSO NM, Версия 1.1. – 2008.

СИНТЕЗ ФРАКТАЛЬНЫХ ДИАГРАММ НАПРАВЛЕННОСТИ АНТЕНН

А.А. Кобликов

(Московский государственный университет им. М.В. Ломоносова)

Научный руководитель – д.ф.-м.н., профессор А.Н. Боголюбов

(Московский государственный университет им. М.В. Ломоносова)

Рассматривается решение задачи синтеза фрактальных диаграмм направленности антенн (ДНА). В основе теории фрактального синтеза заложена идея масштабной инвариантности (реализации характеристик излучения с повторяющейся структурой на произвольных масштабах). Это позволяет создавать новые режимы во фрактальной электродинамике и получать объекты с принципиально новыми свойствами.

Ключевые слова: фрактальная электродинамика, синтез диаграмм направленности, диаграмма направленности антенны, антенны

Введение

В данной работе рассматривается решение задачи синтеза фрактальных диаграмм направленности антенн (ДНА). Сама по себе теория взаимодействия электромагнитных волн с фракталами создана недавно и интенсивно развивается. Применительно к ней в мировой практике сформировался устойчивый термин «фрактальная электродинамика» [1, 2].

В отличие от традиционных методов, когда синтезируются гладкие ДНА, в основе теории фрактального синтеза заложена идея масштабной инвариантности (реализации характеристик излучения с повторяющейся структурой для произвольных масштабов) [3]. Это позволяет создавать новые режимы во фрактальной электродинамике и получать объекты с принципиально новыми свойствами. Например, размещение фрактальных элементов на корпусе объекта может существенно исказить радиолокационный портрет. Сам по себе спектр применения фрактальных антенн весьма широк: телекоммуникации, нелинейная радиолокация, системы поиска, определение местоположения источников радиоизлучения и др.

Синтез фрактальных диаграмм направленности антенн

Синтез фрактальных диаграмм направленности проводится с помощью функции Вейерштрасса:

$$f(x) = \sum_{n=1}^{\infty} \eta^{(D-2)n} g(\eta^n x),$$

где $D = -\log(N)/\log(r)$ – модифицированная фрактальная размерность $1 < D < 2$; N – число элементов в одной подрешетке, $r = r_1/r_2$; r_1 – среднее расстояние между элементами подрешетки (генератора); r_2 – среднее расстояние между элементами случайного возбудителя, $\eta > 1$; g – ограниченная периодическая функция.

При помощи трех переменных (распределение излучателей по пространству, амплитуда и фаза тока возбуждения решетки), можно управлять ДНА. В качестве примера рассмотрим следующую симметричную антенную решетку, показанную на рис. 1.

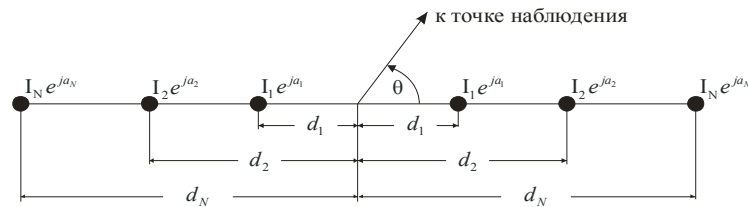


Рис. 1. Антенная решетка с заданным распределением тока возбуждения $I_n = \eta^{(D-2)n}$, $k d_n = a \eta^n$, при $u = \cos \theta$, (где $k = 2\pi/\lambda$ – волновое число)

Получаем, что фрактальная размерность ДНА может контролироваться распределением тока по решетке. По мере уменьшения D , главный лепесток диаграммы направленности расширяется, а соответствующие значения коэффициента направленного действия решетки уменьшаются.

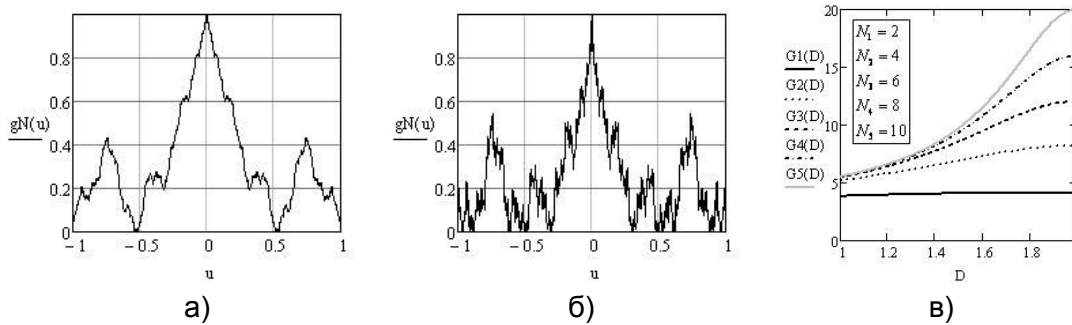


Рис. 2. Нормализованные множители для решетки Вейерштрасса для различных D . (а) – $D=1,1$; (б) – $D=1,5$; (в) – коэффициент направленного действия в зависимости от D при различных N

Рассмотрим непрерывную линейную излучающую систему длиной L с непрерывным изменением тока $I(z)$. Проведя процедуру синтеза с генераторными обобщенными функциями вида (а) $g(\theta) = 1 - |1 - \cos \theta|$, $0 \leq \theta \leq \pi$, (б) $g(\theta) = \sin^2 \theta$, $0 \leq \theta \leq \pi$, были получены синтезированные диаграммы направленности антенн, которые представлены на рис. 3.

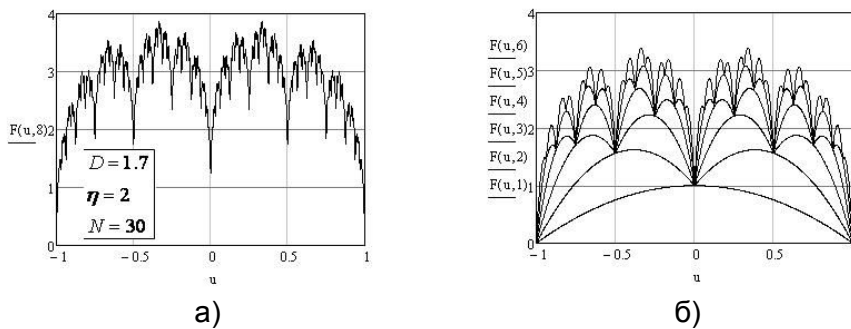


Рис. 3. (а) – синтезированные ДНА; (б) – первые шесть стадий $N=1-6$ в синтезе фрактальных ДНА

Проведенные исследования показали, что использование фрактальных антенн позволяет создавать новые режимы и получать принципиально новые свойства, улучшая эксплуатационные характеристики подобных устройств, открывая тем самым обширную область для применения такого рода антенн. В настоящее время именно математическое моделирование является одним из основных методов исследования подобных структур, позволяющим априори установить их оптимальные параметры. Работа выполнена при поддержке РФФИ (проект № 06-01-00146).

Литература

1. Jaggard D.L. Fractal Electrodynamics: Wave Interaction with Discretely Self-Similar Structures //Symmetry in Electrodynamics /Ed. by C.Baum, H.N.Kritikos. – London: Taylor & Francis. – 1995.
2. New Fractal Antennas for Compact and Versatile Telecommunication Services //Microwave. – 2000. – V.43. – №1. – P. 196–204.
3. Боголюбов А.Н., Кобликов А.А., Петухов А.А., Шапкина Н.Е. Фракталы, их моделирование и применение //XVI международная конференция «Радиолокация и радиосвязь» и XVI международная конференция по спин-электронике и гиромagnetизму (секция XVIII Международной конференции «Электромагнитное поле и материалы»). Москва-Фирсановка. 11–16 ноября 2008 г. – С. 91–95.

ПОДХОД К МОДЕЛИРОВАНИЮ ЭФФЕКТИВНЫХ АВТОМАТИЗИРОВАННЫХ СИСТЕМ НА КЛАСТЕРАХ

**М.Ю. Нестеренко, Д.В. Леонов, М.Д. Быков
(Оренбургский государственный университет)**

Одним из подходов повышения эффективности автоматизированных систем и их подсистем защиты информации является их разработка на основе высокопроизводительных и параллельных технологий. В работе предлагается подход к моделированию функционирования автоматизированных систем на основе кооперативных игр. Решение кооперативных игр – вычислительно емкая задача и может быть решена с помощью высокопроизводительных технологий.

Ключевые слова: системы защиты информации, высокопроизводительные вычисления, автоматизированные системы

Введение

Известно, что к подсистеме защиты информационного и программного обеспечения АСУ предъявляется ряд требований: функциональные, экономические, технические, организационные. Для коммерческих структур одним из главных требований являются экономические. Поэтому построение системы защиты информации (СЗИ) проводится с максимальной экономической эффективностью. В литературе оптимальные по экономическим показателям системы рассматриваются с двух крайних позиций: при заданных ресурсах максимизировать уровень безопасности информации или при заданном уровне безопасности минимизировать затраты на ресурсы, выделяемые для обеспечения безопасности информационных технологий.

При таких подходах в процессе разработки СЗИ АСУ возникает задача определения оптимального объема ресурсов для реализации подсистемы и выбора оптимального уровня защиты информационного и программного обеспечения АСУ. В [1, 2] предлагается определять требования к обеспечению безопасности на основе экспертных оценок по совокупности таких факторов как характер и объем информационного и программного обеспечения, продолжительность пребывания информации на объекте обработки информации (ООИ), структура ООН и др.

В [3, 4] оценка объема ресурсов, которые следует выделять для СЗИ, проводится исходя из того, что стоимость средств обеспечения безопасности должна быть меньше, чем размер возможного ущерба.

В [5] рассматривается некоторый достаточный уровень безопасности. Обеспечение такого уровня безопасности, когда стоимость его преодоления становится больше стоимости получаемой при этом информации (достижимого эффекта), или когда за время получения информации она обесценивается настолько, что усилия по ее получению теряют смысл. В [5] авторы только ставят задачу определения оптимального уровня безопасности: важно правильно выбрать тот достаточный уровень защиты, при котором затраты, риск и размер возможного ущерба были бы приемлемыми (задача анализа риска).

Аналізу рисков при разработке системы защиты информации для построения стратегии защиты уделено значительное внимание. Анализ риска используется для выбора наиболее реальных угроз информационному и программному обеспечению и целесообразных способов защиты от них. Однако, этап анализа рисков подразумевает только оценку реальных затрат и выигрыша от применения предполагаемых мер защиты, и на основе величины выигрыша предлагается принимать решение о выборе стратегии безопасности. Величина выигрыша может иметь как положительное, так и отрицательное значение. В первом случае это означает, что использование подсистемы

обеспечения безопасности информации приносит очевидный выигрыш, а во втором – лишь дополнительные расходы на обеспечение собственной безопасности.

Очевидно, такие подходы не учитывают экономическую эффективность СЗИ, под которой понимается соотношение возможных потерь до внедрения и после внедрения подсистемы с учетом стоимости самой подсистемы. «Злоумышленником» будем называть людей сознательно или несознательно нарушающих безопасность функционирования информационного и программного обеспечения АСУ и неблагоприятное влияние окружающей среды на информационное и программное обеспечение (ПО) АСУ.

Согласно [1–3, 5, 6] принятие оптимальных решений в условиях риска и неопределенности требуют большого объема вычислений. Это обусловлено ограниченностью по объему и номенклатуре массива статистических данных, характеризующих работу реальных СЗИ и учитываемых при выработке оптимальных решений. Такой недостаток информации резко увеличивает множество вариантов развития исходных ситуаций, и каждый из которых должен быть проанализирован, что становится возможным с использованием кластерных технологий.

Если существует и известна функциональная зависимость критерия оптимизации от входных управляемых параметров, тогда задача оптимизации сводится к отысканию таких значений факторов, при которых целевая функция достигает экстремума. Если на объект воздействуют аддитивные помехи, то зависимость выражает не функциональную, а регрессионную зависимость, которая образует поверхность отклика. Для отыскания экстремума существует два принципиально различных подхода:

1) если есть возможность найти n -факторную математическую модель в той части факторного пространства, где расположен экстремум, то задача решается аналитическим или численным методами;

2) если математическое описание не получено, то осуществляется экспериментальный поиск области оптимума.

1. Игровая модель системы защиты информации

Построение модели взаимодействия злоумышленников и СЗИ может быть выполнено на основе кооперативной игры двух или более игроков, которая эффективно решается с использованием высокопроизводительных технологий. Первым игроком в разрабатываемой модели будет система защиты информации (СЗИ) предприятия. Стратегии СЗИ представляют собой набор методов и средств, обеспечивающих информационную безопасность, которые в свою очередь делятся на административные, технические и аппаратно-программные. Остальными $n-1$ игроками (если общее число игроков при моделировании принять за n) являются злоумышленники. Стратегии злоумышленников – реализация угрозы.

Из теории игр известно [7], что кооперативные игры получаются в тех случаях, когда в игре n игроков разрешается образовывать определённые коалиции (в рассматриваемой модели образование коалиции между СЗИ и злоумышленниками исключается). Обозначим через N множество всех игроков, $N = \{1, 2, \dots, n\}$, а через K – любое его подмножество. Пусть игроки (злоумышленники) из K договариваются между собой о совместных действиях и, таким образом, образуют одну коалицию. Образовав коалицию, множество игроков K действует как один игрок против остальных игроков, и выигрыш этой коалиции зависит от применяемых стратегий каждым из n игроков.

Для описания кооперативной игры необходимо определить характеристическую функцию v , ставящую в соответствие каждой коалиции K наибольший, уверенно

получаемый ею выигрыш $v(K)$. Мощность множества коалиций, на котором определена характеристическая функция равна 2^n , причем коалиция с номером 0, является пустой коалицией и ее выигрыш по условию, всегда равен нулю.

Для решения кооперативной игры следует построить оптимальный дележ $X = \{x_1, x_2, \dots, x_n\}$, распределяющий выигрыши игроков с учетом следующих условий:

- 1) $x_i \geq v(i)$, $i \in N$ (принцип индивидуальной рациональности),
- 2) $\sum_{i \in N} x_i = v(N)$ (принцип коллективной рациональности).

В бескоалиционных играх исход формируется в результате действий игроков, которые в этой ситуации получают свои выигрыши. Исходом в кооперативной игре является делёж, возникающий не как следствие действий игроков, а как результат их соглашений. Поэтому в кооперативных играх сравниваются не ситуации, как это имеет место в бескоалиционных играх, а дележи, и сравнение это носит более сложный характер.

Возможность задания кооперативной игры множеством биматричных игр рассмотрена И.Д. Протасовым [8], однако не приводится описание способа построения биматричных игр. Мы предлагаем следующий алгоритм задания кооперативной игры. Коалиционная игра задается в виде $n \cdot (n-1)$ матриц размера $k \times k$, где n – количество игроков, k – количество стратегий у игроков, и имеющих вид:

$$A_{xy} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \dots & \dots & \dots & \dots \\ a_{k1} & a_{k2} & \dots & a_{kk} \end{pmatrix},$$

где a_{ij} ($i, j = \overline{1, k}$) – выигрыши игрока x в игре с игроком y . Стратегии коалиции определяются выбором игроков-участников коалиций. Таким образом, коалиция из m игроков будет иметь k^m чистых стратегий, где k – число выборов у каждого игрока.

Далее необходимо найти матрицы выигрышей всех коалиций в игре с остальными участниками. Каждая матрица размера $k^m \times k^{n-m}$, где m – количество игроков в текущей коалиции, имеет вид:

$$A_{\overline{XY}} = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1k^{n-m}} \\ b_{21} & b_{22} & \dots & b_{2k^{n-m}} \\ \dots & \dots & \dots & \dots \\ b_{k^m 1} & b_{k^m 2} & \dots & b_{k^m k^{n-m}} \end{pmatrix}, \text{ при } 0 < m \leq n,$$

где b_{ij} – выигрыш коалиции \overline{X} в игре с коалицией \overline{Y} , причем i и j – комбинированные стратегии коалиций.

Каждая биматричная игра описывается двумя матрицами, которые сводятся к задаче линейного программирования и решаются с помощью симплекс-метода. В результате решения получаем характеристическую функцию v .

Для нахождения дележа используется принцип оптимальности распределения выигрыша между игроками в задачах теории кооперативных игр, называемый вектором Шепли.

Представим кооперативную игру СЗИ и злоумышленников в виде совокупности биматричных игр следующего вида:

$$A_{XY_k} = \begin{pmatrix} -a_{11} & -a_{12} & \dots & -a_{1k} \\ -a_{21} & -a_{22} & \dots & -a_{2k} \\ \dots & \dots & \dots & \dots \\ -a_{k1} & -a_{k2} & \dots & -a_{kk} \end{pmatrix},$$

где X – это СЗИ, Y_k – k -й злоумышленник (или коалиция злоумышленников), $-a_{ij}$ ($i, j = \overline{1, k}$) – включает в себя затраты на СЗИ, реализующую i -ую стратегию, и потери от действия злоумышленника, реализующего j -ую угрозу;

$$B_{XY_k} = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1k^{n-m}} \\ b_{21} & b_{22} & \dots & b_{2k^{n-m}} \\ \dots & \dots & \dots & \dots \\ b_{km_1} & b_{km_2} & \dots & b_{km_k^{n-m}} \end{pmatrix},$$

где b_{ij} – доход от преодоления системы защиты информации, реализующей i -ую стратегию, за вычетом затрат на реализацию угрозы, i – номер стратегии СЗИ, j – номер угрозы.

Матрицы взаимных платежей злоумышленников выглядят следующим образом:

$$A_{Y_s Y_p} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \dots & \dots & \dots & \dots \\ a_{k1} & a_{k2} & \dots & a_{kk} \end{pmatrix}, B_{Y_s Y_p} = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1k^{n-m}} \\ b_{21} & b_{22} & \dots & b_{2k^{n-m}} \\ \dots & \dots & \dots & \dots \\ b_{km_1} & b_{km_2} & \dots & b_{km_k^{n-m}} \end{pmatrix},$$

где a_{ij} (b_{ij}) – платеж злоумышленника p (соответственно s) злоумышленнику s (соответственно p) при выборе ими j -ой и i -ой угрозы.

2. Анализ подходов к распараллеливанию алгоритма решения кооперативной игры для вычислительного кластера

Для решения кооперативных игр применяется сведение игр к множеству стратегических игр и решение каждой из них симплекс-методом, для нахождения выигрыша каждой коалиции. Как было отмечено раньше, всего коалиций может быть 2^n , где n – количество игроков, следовательно, необходимо решить 2^n стратегических игр симплекс-методом, сложность алгоритма которого составляет $O(n^2)$. При достаточно больших n это занимает длительное время.

Кроме того, большую вычислительную сложность имеет процесс формирования матриц, которыми задаются коалиции. Трудоемкость этого процесса составляет $O(n^2)$. Так как, матрицы, описывающие коалиции не зависят друг от друга, а зависят только от матриц, задающих саму игру, то генерация матриц может осуществляться параллельно.

Исходя из вышесказанного, можно сделать вывод, что формирование матриц, характеризующих коалиции и их расчет симплекс-методом являются независимыми участками кода, которые можно выполнять параллельно (подробный анализ алгоритма рассматривается в статье [9]).

Заключение

В результате проведенной работы были получены следующие результаты.

1. Построена модель взаимодействия злоумышленников и СЗИ на основе кооперативной игры двух или более игроков.
2. Рассмотрена возможность сведения кооперативных игр к множеству биматричных игр для игроков с большим числом стратегий.
3. Предложен параллельный алгоритм решения кооперативной игры для высокопроизводительных кластерных систем.
4. Экспериментальное исследование на кластере ОГУ программной реализации данного алгоритма демонстрирует хорошую масштабируемость.

Литература

1. Малюк А.А. Теоретические основы формализации прогнозной оценки уровня безопасности информации в системах обработки данных [Текст] / Малюк А.А. – М.: МИФИ. – 1998. – 40 с.
2. Герасименко В.А. Основы защиты информации [Текст] / Герасименко В.А., Малюк А.А. – М.: МИФИ. – 1997. – 537 с.
3. Семкин С.Н. Основы безопасности объектов обработки информации [Текст] / Семкин С.Н., Семкин А.Н. – М.: Москва. – 2000.
4. Романец Ю.В. Защита информации в компьютерных системах и сетях [Текст] / Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф. М.: Радио и связь. – 2001.
5. Гайкович В.Ю. Основы безопасности информационных технологий [Текст] / Гайкович В.Ю., Ершов Д.В. – М.: МИФИ. – 1995.
6. Гайкович В.Ю. Безопасность электронных банковских систем [Текст] / Гайкович В.Ю., Першин А.Ю. – М.: Единая Европа. – 1994.
7. Крушевский А.В. Теория игр [Текст] / Крушевский А.В.. – Киев, Издательское объединение «Вища школа». – 1977. – 216 с.
8. Протасов И.Д. Теория игр и исследование операций: Учебное пособие [Текст] / Протасов И.Д. – М.: Гелиос АРВ. – 2003. – 368 с.
9. Нестеренко М.Ю., Леонов Д.В., Болгова Е.В., Кириллов А.С. Разработка программного обеспечения для моделирования конкурентного рынка на кластерных системах / СПб: Научно-технический вестник СПбГУ ИТМО. – №54. – 2008. – С. 156–161.

ВЕЙВЛЕТ-ФРАКТАЛЬНАЯ ОБРАБОТКА ИЗОБРАЖЕНИЯ

А.А. Виноградова, П.П. Коваленко, Г.А. Недоцука

Научный руководитель – д.т.н., профессор В.М. Мусалимов

Данная статья показывает важность применения обработки изображений. В ней описаны два вида обработки: вейвлет-обработка и фрактальная обработка. Описано применение двумерных дискретных вейвлет-преобразований. Наглядно показано использование двумерных пакетных вейвлетов для двумерного разложения и реконструкции изображения. Статья также даёт понятие о применении фрактальной обработки.

Ключевые слова: вейвлеты, фракталы, обработка, изображения

Введение

В жизни человечества существует необходимость к улучшению качества изображений. Не маловажным данный вопрос становится в медицине, при изучении или определении болезни врач должен иметь на руках четкий снимок пациента, а также военном деле, науке, искусстве и др. Поэтому встал вопрос об обработке и улучшении качества изображений. С хорошей стороны в данном вопросе зарекомендовали себя как вейвлет-обработка так и фрактальная обработка изображений.

Из целого ряда весьма полезных и специфических применений вейвлетов, на первом месте стоит редактирование изображений, включающее вывод изображения на экран дисплея, раскрашивание и корректировка изображений, учёт граничных условий и техника дробных разрешений. Во всех этих операциях применение вейвлет-технологии даёт явный выигрыш в сравнении с ранее известными средствами машинной графики.

При серьезных графических работах объем изображений становится огромным и поиск нужных картинок и их просмотр превращается в весьма утомительную процедуру. И здесь применение вейвлетов становится оправданным и перспективным. По существу речь идет о новых средствах построения базы данных для изображений и средств управления ими.

Так же фракталы, эти красивые образы динамических систем, ранее использовались в машинной графике в основном для построения изображений неба, листьев, гор, травы. Красивое и, что важнее, достоверно имитирующее природный объект изображение могло быть задано всего несколькими коэффициентами. Неудивительно, что идея использовать фракталы при обработке изображений возникла и раньше, но считалось практически невозможным построить соответствующий алгоритм, который подбирал бы коэффициенты за приемлемое время. В последствии алгоритм был найден.

Вейвлет-обработка

В системе MATLAB при помощи вейвлет пакетов можно оперировать качеством изображений [1]. Немаловажно в данном вопросе очистка изображений от шумов (рис. 1).

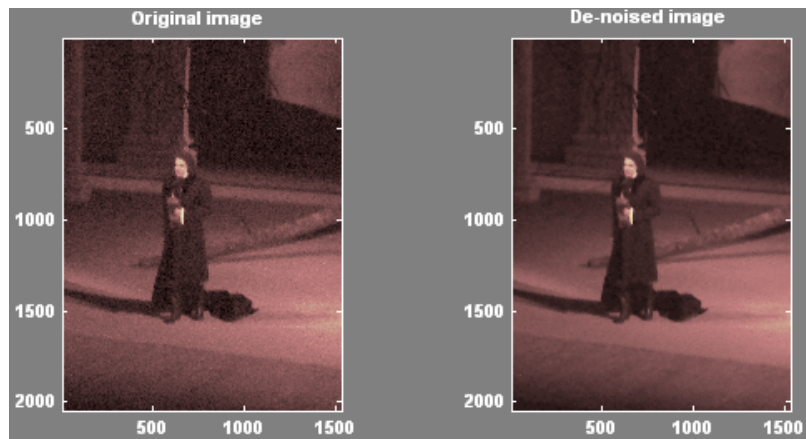


Рис. 1. Очистка изображения от шумов

Растровые изображения являются двумерным дискретным сигналом, к нему применимы двумерные дискретные вейвлет-преобразования. Один шаг двумерного вейвлет-преобразования выделяет одну низкочастотную и три высокочастотных компоненты исходного сигнала-изображения. Если не производить никаких дополнительных действий с этими компонентами, то по ним с помощью шага обратного вейвлет-преобразования можно полностью восстановить исходное изображение.

Специальным образом обрабатывая низко- и высокочастотные компоненты такого разложения и осуществляя (если нужно) обратные преобразования, можно добиваться различных результатов обработки.

Если, выполняя обратное вейвлет-преобразования, все вейвлет-коэффициенты одного или нескольких уровней детализации умножить на некоторое неотрицательное число, то можно изменить четкость изображения. Если число будет больше 1, то четкость возрастет, если меньше 1 или даже равно 0, то четкость понизится (изображение окажется размытым).

Двумерные пакетные вейвлеты также могут успешно использоваться для двумерного разложения и реконструкции изображений с использованием для этого деревьев декомпозиции и реконструкции (рис. 2).

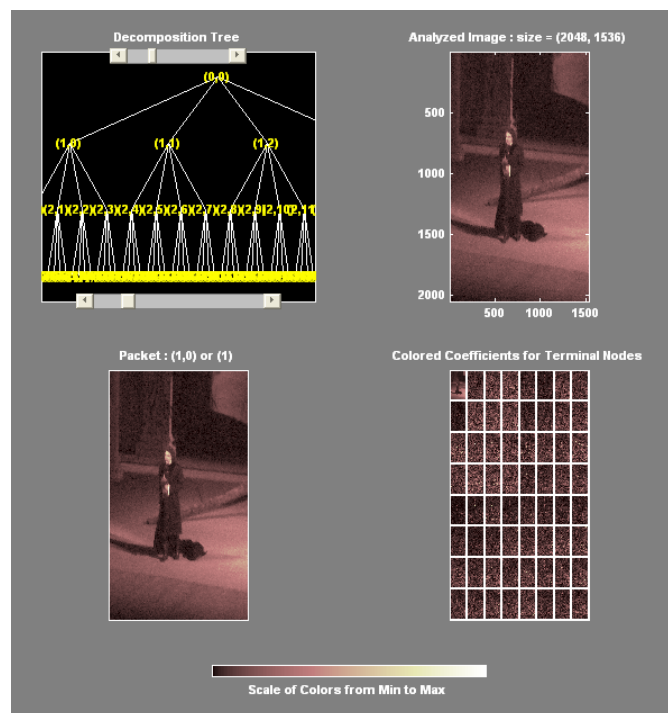


Рис. 2. Разложение изображения с использованием деревьев декомпозиции

Элементы в правой части окна позволяют выбирать различные типы дерева, осуществлять анализ, компрессию и очистку изображения от шума (если загруженное изображение имеет шум).

Для компрессии изображения и его очистки от шума применение вейвлет-обработки предоставляет ряд возможностей. Активизировав в нужном окне необходимую кнопку, можно очистить изображения от шума и заодно провести его компрессию. Степень очистки изображения и компрессии во многом определяется установкой порога. Можно выбрать дополнительно метод очистки и получить остаток (рис. 3).



Рис. 3. Исходное изображение и очищенное от шума с проведением компрессии

Фрактальная обработка

Математическое понятие фрактала выделяет объекты, обладающие структурами различных масштабов, как больших, так и малых, и, таким образом, отражает иерархический принцип организации. В основе этого понятия содержится одна очень важная идеализация действительности: фрактальные объекты самоподобны. Т.е. фрактал – это структура, состоящая из частей, которые в каком-то смысле подобны целому.

Самые простые и известные фракталы – это кривая Коха (рис. 4) и ковер Серпинского (рис. 5).

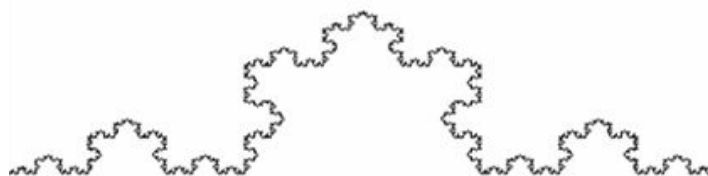


Рис. 4. Кривая Коха

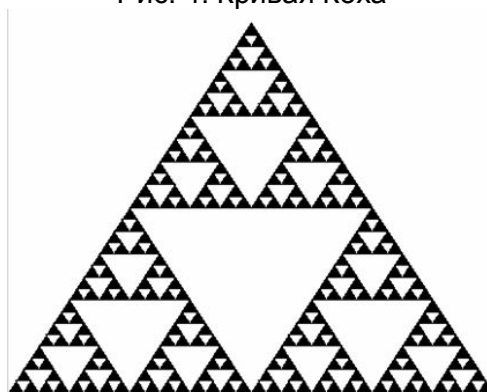


Рис. 5. Ковер Серпинского

Существует критический показатель (фрактальная размерность) $D_H = \lim_{r \rightarrow 0} \frac{\ln N}{\ln(1/r)}$ [2].

Отличительной особенностью фрактальной размерности является то, что она характеризует степень заполнения пространства, в которой существует фрактальная система. Большому значению фрактальной размерности соответствует большая степень заполнения изображения (его трехмерного представления). Для совершенно черного изображения фрактальная размерность будет равна $D = 2$, т.е. совпадать с топологической размерностью плоскости, а для изображения имеющего одинаковую яркость всех пикселей – $D = 3$ (топологическая размерность объема). Изображения в градациях серого будут иметь дробную фрактальную размерность.

В результате можно четко определить, например, границы льда и воды (рис. 6).

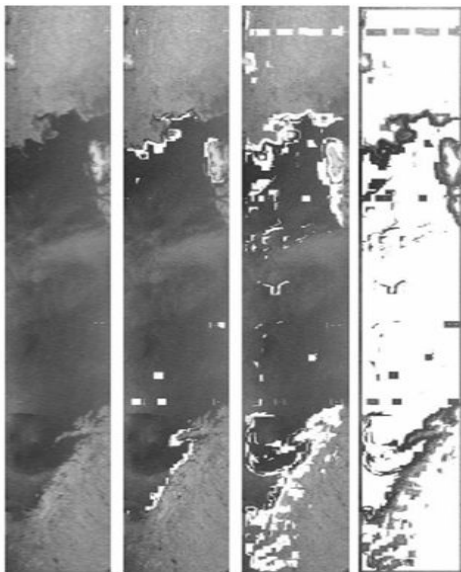


Рис. 6. Границы льда и воды при разных фрактальных размерностях: $D=2,7-2,8$, $D=2,8-2,9$, $D=2,9-2,995$

Количественная оценка

Качество изображения связано с нормами [3], которые впервые были применены Е.А. Воронцовым и С.В. Орловым. В качестве критерия степени деградации использовались нормы матриц изображения. Количественная оценка позволяет автоматизировать обработку изображений, исключая работу оператора.

Заключение

В статье были описаны два вида обработки изображений: вейвлет-обработка и фрактальная обработка. Также показано, каким образом, возможно, проводить данные операции.

Фракталы и вейвлеты прекрасно зарекомендовали себя при обработке изображений, улучшении качества и четкости изображения. В дальнейшем планируется совместить вейвлет-обработку и фрактальную обработку изображений, что должно намного улучшить распознавание объектов на фотографиях и других изображениях. В таком случае работа с обработкой будет намного легче в любой области её применения.

Литература

1. Дьяконов В.П. Вейвлеты. От теории к практике: 2-е изд., перераб. и доп. – М.: СОЛОН – Пресс. – 2004. – 399 с.
2. Шредер М. Фракталы, хаос, степенные законы: Ижевск: НИЦ «РХД», Ижевская республиканская типография. – 2001. – 528 с.
3. Орлов С.В. Циклическое ударно-фрикционное взаимодействие чеканочного инструмента с монетной заготовкой. Автореферат, СПбГУ ИТМО. – 2007. – 20 с.

ИСПОЛЬЗОВАНИЕ НЕЧЕТКОЙ ЛОГИКИ ДЛЯ АНАЛИЗА И ОБРАБОТКИ ИЗОБРАЖЕНИЙ

Г.А. Недоцука, П.П. Коваленко, А.А. Виноградова

Научный руководитель – д.т.н., профессор В.М. Мусалимов

В данной работе проводится краткий обзор основных понятий нечеткой логики. Рассматривается создание нечеткой модели с использованием нечеткого логического вывода Мамдани. В качестве основных входных переменных выступают факторы, которые оказывают влияние на качество изображения. Само качество обрабатываемого изображения, является искомой выходной переменной.

Ключевые слова: нечеткая логика, нечеткие множества, функция принадлежности, качество изображения, обработка изображений

Введение

Нечеткая логика и теория нечетких множеств – раздел математики, являющийся обобщением классической логики и теории множеств. Понятие нечёткой логики было введено профессором Лютфи Заде в 1965 г. в статье «Fuzzy Sets» в журнале «Information and Control». Заде расширил понятие множества допущением, что функция принадлежности элемента к множеству может принимать любые значения в интервале от 0 до 1, а не только 0 или 1, как это было раньше. Такие множества он назвал нечеткими. Также он предложил различные логические операции над нечеткими множествами и ввел понятие лингвистической переменной [1].

Сейчас нечеткое моделирование является одной из наиболее перспективных направлений прикладных исследований в области управления и принятия решений. Нечеткое моделирование активно используется в описании технических систем и бизнес-процессов, когда в них присутствует неопределенность, которая затрудняет или даже исключает применение точных количественных методов и подходов. Нечеткие системы, т.е. системы, используемые нечеткие множества, успешно работают в различных областях инженерии, экономики, медицины, биологии, психологии, политики и др. [2].

Построение нечеткой модели

Характеристикой нечеткого множества выступает функция принадлежности (Membership Function), которая позволяет для произвольного элемента универсального множества вычислить степень его принадлежности нечеткому множеству. Степень принадлежности – это число из диапазона $[0,1]$. Чем выше степень принадлежности, тем больше элемент универсального множества соответствует свойствам нечеткого множества. Значение функции принадлежности равно 0 означает отсутствие принадлежности к множеству, 1 – полную принадлежность. Лингвистической переменной называется переменная, значениями которой могут быть слова или словосочетания некоторого естественного языка, например, «плохое качество изображения», «хорошая резкость», «много шумов». Термом называется любой элемент терм-множества, которое является множеством всех возможных значений лингвистической переменной. Для нечетких множеств, как и для обычных, определены основные логические операции. Самыми основными, необходимыми для расчетов, являются пересечение (нечеткое "И"), объединение (нечеткое "ИЛИ") и дополнение.

Нечеткая переменная описывается набором (N, X, A) , где N – это название переменной, X – универсальное множество (область рассуждений), A – нечеткое множество на X . База правил, содержащая нечеткие высказывания в форме "Если-то" и

функции принадлежности для соответствующих лингвистических термов выступают в качестве основы для проведения операции нечеткого логического вывода. При этом должно существовать хотя бы одно правило для каждого лингвистического термина выходной переменной и любой терм входной переменной имеет хотя бы одно правило, в котором этот терм используется в качестве предпосылки. Иначе имеется неполная база нечетких правил.

При создании нечеткой модели в качестве входных переменных были взяты основные факторы, от которых зависит качество изображения (рис. 1).

1. Резкость (Sharpness) определяется количеством деталей, которые передаются изображением.

2. Шум (Noise) – случайные погрешности в каждой точке изображения.

3. Контрастность (Contrast) можно определить как диапазон тонов средней яркости, при увеличении контрастности диапазон яркости средних тонов сужается, а при уменьшении контрастности наоборот увеличивается, так при увеличении контрастности пиксели, имеющие яркость выше некоторого значения, приобретают еще большую яркость, а пиксели, имеющие яркость ниже некоторого значения, становятся еще темнее.

4. Цветовая четкость (Color accuracy) как всего изображения в целом, так и всех отдельных пикселей изображения.

5. Дисторсия (Distortion) – геометрическое искажение прямых линий, существуют два вида дисторсии – подушкообразная, если прямые стали вогнутыми, и бочкообразная, если выпуклыми.

6. Виньетирование (Light falloff) – это падение яркости от центра к краям изображения, как правило, в центре нет затемнения, оно четко видно на углах.

7. Задний и передний фокус (Focus) – это когда четкость смещена в сторону камеры или в обратную от камеры сторону.

8. Хроматические aberrации (Chromatic aberration) – явление вызванное дисперсией света (разложение луча света на составляющие) проходящего через объектив, потому что лучи с разной длиной волны (разного цвета) преломляются под разными углами, поэтому из белого пучка образуется радуга, хроматические aberrации приводят к снижению чёткости изображения и появлению цветных контуров (особенно на контрастных объектах).

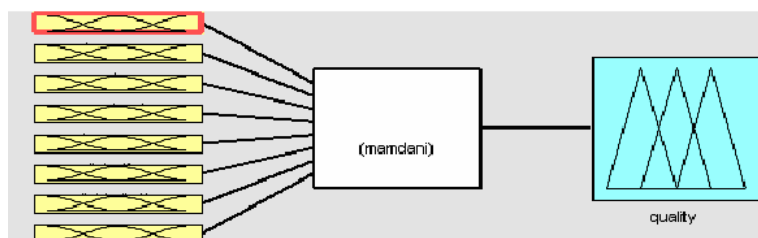


Рис. 1. Модель оценки качества изображения с 8 входными переменными

Для каждой входной переменной были заданы треугольные функции принадлежности. Для лингвистической оценки каждой переменной были использованы три термина «плохой (bad)», «хороший (good)», «отличный (excellent)». Причем, например, для входной переменной шум (Noise) терм «отличный (excellent)» означает, что шумы практически отсутствуют и не ухудшают качество изображения, а входная переменная хроматические aberrации (Chromatic aberration) по аналогии имеет терм «отличный (excellent)», если также aberrаций нет. Для выходной переменной, которая обозначает качество изображения (quality), используем пять термов (bad, worse, good, better, excellent) с гауссовыми функциями принадлежности. На этой основе была создана база знаний, описывающий основные правила (рис. 2). Например,

If (Sharpness is good) and (Noise is good) and (Contrast is good) and (Color accuracy is good) and (Distortion is good) and (Light falloff is good) and (Focus is good) and (Chromatic aberration is good) then (quality is good)

If (Sharpness is bad) and (Noise is bad) and (Contrast is bad) and (Color accuracy is bad) and (Distortion is bad) and (Light falloff is bad) and (Focus is bad) and (Chromatic aberration is bad) then (quality is bad)

If (Sharpness is bad) and (Noise is bad) and (Contrast is bad) and (Color accuracy is bad) and (Distortion is good) and (Light falloff is good) and (Focus is good) and (Chromatic aberration is good) then (quality is worse)

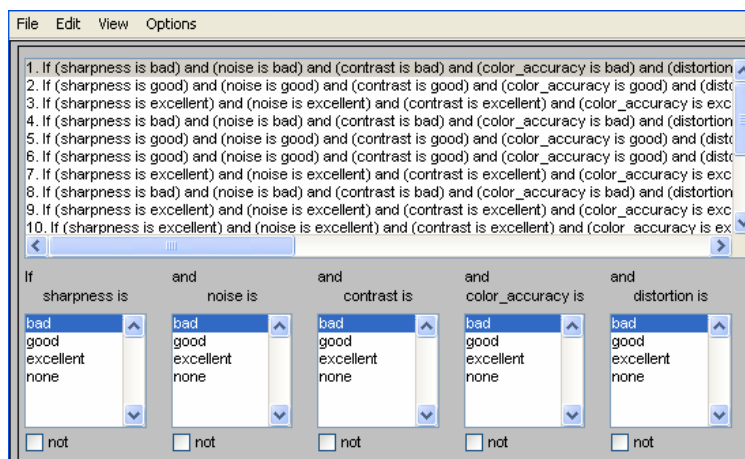


Рис. 2. Логистические правила

При составлении логических правил, авторы исходили из того, что качество изображения ухудшается при большом наличии шумов, плохой резкости и контрастности, цветовой нечеткости, наличия дисторсии и виньетирования, а также хроматических aberrаций и переднего и заднего фокуса. При отсутствии этих факторов или их снижении качество изображения улучшается. Графическое представление основных логических правил представлено на рис. 3.

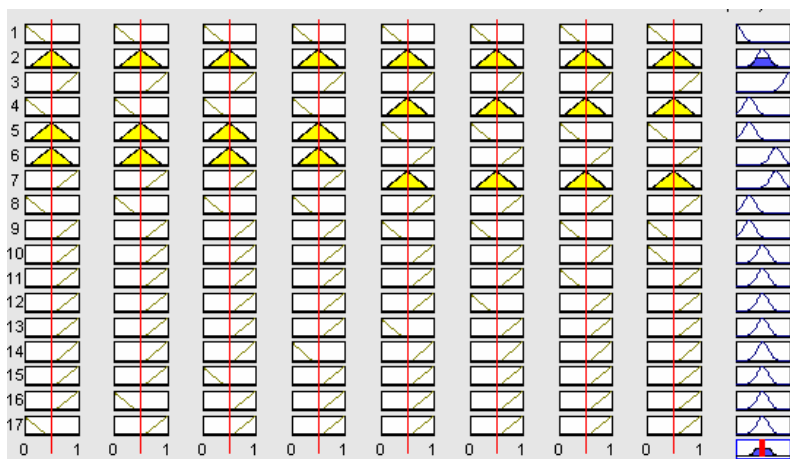


Рис. 3. Графическое представление логических правил

Алгоритмы нечеткого вывода различаются главным образом видом используемых правил, логических операций и разновидностью метода дефазификации. Сейчас наиболее известны модели нечеткого вывода Мамдани, Сугено, Ларсена, Цукамото.

Модель оценки качества изображения основывается на нечетком логическом выводе Мамдани, в котором все значения входных и выходных переменных заданы нечеткими множествами. Базу знаний Мамдани можно трактовать как разбиение пространства влияющих факторов на подобласти с размытыми границами, внутри

которых функция отклика принимает нечеткое значение. Ниже представлены некоторые полученные результаты (рис. 4).

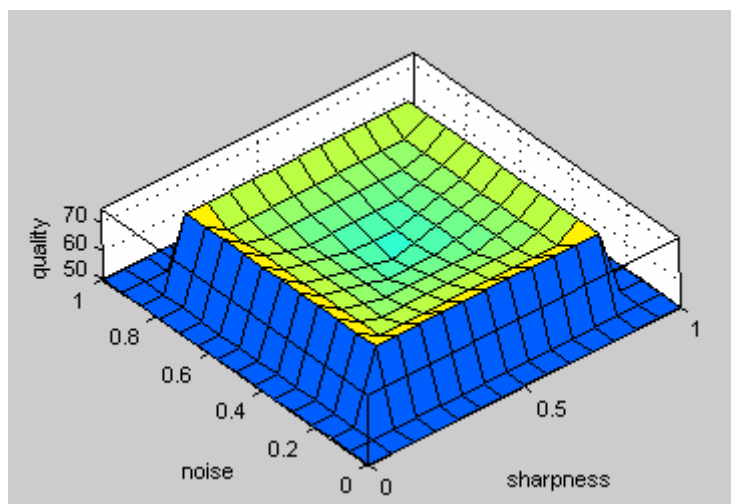


Рис. 4. Визуализация нечеткого вывода Мамдани

Вывод

Популярность нечетких множеств для моделирования сложных процессов объясняется тем, что нечеткие системы разрабатываются быстрее четких аналогов, они получаются проще и дешевле, также нечеткие алгоритмы более понятны для человека. Это направление активно развивается и представляет большой интерес. Мы также планируем расширять построенную нами нечеткую модель. Сейчас ведется работа по увеличению количества входных переменных, которые являются факторами, оказывающими влияние на качество изображения, более глубоко оценивается степень их взаимовлияния и разрабатывается модель, основанная на нечетком логическом выводе Сугено.

Литература

1. Штовба С.Д. Проектирование нечетких систем средствами MATLAB. – М.: Горячая линия – Телеком. – 2007. – 228 с.
2. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб: БХВ – Петербург. – 2005. – 736 с.

НЕЛИНЕЙНОЕ СНИЖЕНИЕ РАЗМЕРНОСТИ С ПОМОЩЬЮ МНОГООБРАЗИЯ ПРОЕКТИРОВАНИЯ И ЕГО ПРИМЕНЕНИЕ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ОБСЧЕТА ПРОФИЛЕЙ КРЫЛА САМОЛЕТА

**Е.В. Бурнаев (Институт проблем передачи информации им. А.А. Харкевича РАН),
П.Д. Ерофеев
(Московский физико-технический институт (государственный университет))
Научный руководитель – д.ф.-м.н., профессор А.В. Бернштейн
(Институт системного анализа РАН)**

В данной работе исследуется применение процедур нелинейного снижения размерности в задачах восстановления пропущенных значений в наборах однородных данных. Предложенный метод обобщает метод восстановления пропущенных значений на основе анализа главных компонент за счет учета нелинейных связей между элементами выборки данных. Метод может быть применен к задаче обратного проектирования профиля крыла самолета, когда по желаемым аэродинамическим характеристикам (в данном случае – распределению давления) восстанавливается профиль крыла. Результаты моделирования на выборке реальных профилей крыла и распределений давления на них показали, что использование нелинейности позволяет значительно уменьшить ошибку восстановления.

Ключевые слова: восстановление пропущенных данных, снижение размерности данных, профиль крыла

Введение

В прикладных задачах математики часто возникает проблема восстановления пропущенных данных. Предполагается, что отсутствие некоторых значений параметров объекта вызвано исключительно техническими проблемами, но не изменением состояния объекта. Так, пропуски могут появиться в результате недоступности или сложности проведения некоторых измерений, из-за грубых ошибок при подготовке данных, неисправности измерительных приборов, а также при исключении резко выделяющихся данных. В зависимости от решаемой задачи, исследователю может потребоваться оценить сами пропущенные значения, либо оценить некоторые другие данные, на основе восстановленных значений. В обоих случаях используются методы восстановления, основанные на предположении о существовании некоторой избыточной информации, возникающей вследствие наличия связей между признаками.

При построении алгоритмов восстановления пропущенных данных может быть использовано два подхода [1, 4]. Первый предполагает вероятностное происхождение выборки [4]. Т.е. процесс получения значений может быть многократно повторен, хотя бы мысленно. Но есть задачи, в которых говорить о вероятностной природе данных не имеет смысла. К ним можно отнести, например, задачу проектирования профиля крыла самолета по его аэродинамическим характеристикам. В этом случае следует применять другой подход – алгебро-геометрический. Этот подход не содержит каких-либо дополнительных предположений о том, как была порождена выборка, и основан лишь на ее геометрической структуре. В дальнейшем будет использоваться именно этот подход.

В данной статье рассматривается метод восстановления пропущенных значений на основе снижения размерности данных. Метод заключается в том, что недостающие данные в многомерном векторе, содержащем пропуски, заполняются значениями его проекции на многообразие, рядом с которым (в смысле геометрической близости) находится обучающая выборка. Многообразие имеет более низкую размерность, чем исходное пространство векторов выборки. В [1] описан метод, реализующий эту идею

и основанный на анализе главных компонент. Но, в силу линейности последнего, в [1] не были учтены возможные нелинейные связи в исходных данных. Целью данной работы является обобщение метода восстановления пропущенных данных на случай, когда в данных присутствуют нелинейные связи. Применение разработанного метода проиллюстрировано на примере решения задачи облета крыла самолета.

Работа имеет следующую структуру: во втором разделе описана общая постановка задачи снижения размерности, в третьем – общий подход к восстановлению пропущенных данных, в четвертом – специально разработанный метод нелинейного снижения размерности с помощью многообразий проектирования и, наконец, в пятом разделе приведены результаты применения предложенного метода для решения задачи облета крыла самолета.

Общая постановка задачи снижения размерности

Пусть исследуемые объекты описываются векторами n -мерного пространства:

$$\mathbf{Y} = (Y_1, Y_2, \dots, Y_n) \in R^n. \quad (1)$$

Процедура снижения размерности Σ выполняет две основные функции [3]:

- по заданному вектору профиля \mathbf{Y} размерности n строит сжатое описание $\lambda = \lambda(\mathbf{Y})$ меньшей размерности;
- по заданному сжатому описанию λ восстанавливает «полноразмерный» вектор $\mathbf{Y} = \mathbf{Y}(\lambda)$.

Процедура снижения размерности $\Sigma = \{m, K_m, K_m^-\}$ состоит из:

- размерности m сжатого описания;
- преобразования сжатия $K_m : \mathbf{Y} \rightarrow \lambda = K_m(\mathbf{Y}) \in R^m$, преобразующего вектор \mathbf{Y} в m -мерный сжатый вектор $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m) \in R^m$;
- преобразования восстановления $K_m^- : \lambda \rightarrow \mathbf{Y} = K_m^-(\lambda) \in R^n$, преобразующего m -мерный сжатый вектор λ в полноразмерный вектор $\mathbf{Y}(\lambda)$.

Обозначим через

$$\mathbf{Y}^* = K_m^-(K_m(\mathbf{Y})) = (Y_1^*, Y_2^*, \dots, Y_n^*), \quad (2)$$

восстановленный вектор, получающийся применением преобразования восстановления K_m^- к результату сжатия $K_m(\mathbf{Y})$ исходного вектора \mathbf{Y} . Пусть

$$\Delta(\mathbf{Y}) = \Delta(\mathbf{Y} | \Sigma) = \mathbf{Y}^* - \mathbf{Y} = (\Delta_1, \Delta_2, \dots, \Delta_n) \quad (3)$$

вектор, состоящий из ошибок восстановления, и

$$\rho(\Delta) = \sqrt{\sum_{r=1}^n (\Delta_r)^2} \quad (4)$$

расстояние между исходным вектором \mathbf{Y} (1) и восстановленным вектором \mathbf{Y}^* (2).

Расстояние $\rho(\Delta(\mathbf{Y} | \Sigma))$ (4) характеризует качество процедуры снижения размерности Σ , примененной к конкретному вектору \mathbf{Y} .

Для заданной выборки объема N

$$D_N = \{\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_N\}, \quad (5)$$

интегральной выборочной характеристикой процедуры снижения размерности Σ , вычисленной по этой выборке, является величина

$$R = R(\Sigma | D_N) = \frac{1}{N} \sum_{r=1}^N \rho(\Delta(\mathbf{Y}_r | \Sigma)),$$

равная результату усреднения множества расстояний

$$\rho(\Delta(D_N | \Sigma)) = \{\rho(\Delta(\mathbf{Y}_1 | \Sigma)), \rho(\Delta(\mathbf{Y}_2 | \Sigma)), \dots, \rho(\Delta(\mathbf{Y}_N | \Sigma))\}.$$

Восстановления пропущенных данных на основе процедуры снижения размерности

Пользуясь некоторой процедурой снижения размерности, построим по заданной выборке (5) многообразию

$$\Omega_m = \{\mathbf{Y} \in R^n : \mathbf{Y} = K_m^-(\boldsymbol{\lambda}), \boldsymbol{\lambda} \in \Lambda \subset R^m\}, \quad (6)$$

максимально приближающее выборку.

Теперь предположим, что вектор \mathbf{Y} состоит из двух векторов меньшей размерности:

$$\mathbf{Y} = (\mathbf{X}, \mathbf{Z}) = (X_1, X_2, \dots, X_s, Z_1, Z_2, \dots, Z_{n-s}), \mathbf{X} \in R^s, \mathbf{Z} \in R^{n-s}$$

и преобразование восстановления представим в виде

$$K_m^-(\boldsymbol{\lambda}) = (K_{\mathbf{X},m}^-(\boldsymbol{\lambda}), K_{\mathbf{Z},m}^-(\boldsymbol{\lambda})), K_{\mathbf{X},m}^-(\boldsymbol{\lambda}) \in R^s, K_{\mathbf{Z},m}^-(\boldsymbol{\lambda}) \in R^{n-s}$$

Пусть $\tilde{\mathbf{Y}} = (\tilde{\mathbf{X}}, \tilde{\mathbf{Z}})$ – новое наблюдение из того же класса, что и наблюдения из выборки (5), и пусть последние $n-s$ компонент вектора $\tilde{\mathbf{Y}}$ неизвестны. Спроектируем вектор $\tilde{\mathbf{Y}}$ на многообразии Ω_m (6) и тем самым оценим неизвестные компоненты этого вектора. Фактически, проектирование является решением задачи минимизации:

$$(\boldsymbol{\lambda}^*, \mathbf{Z}^*) = \arg \min_{\boldsymbol{\lambda}, \mathbf{Z}} (\|\tilde{\mathbf{X}} - K_{\mathbf{X},m}^-(\boldsymbol{\lambda})\| + \|\tilde{\mathbf{Z}} - K_{\mathbf{Z},m}^-(\boldsymbol{\lambda})\|). \quad (7)$$

Очевидно, что пара

$$\mathbf{Z}^* = K_{\mathbf{Z},m}^-(\boldsymbol{\lambda}^*), \boldsymbol{\lambda}^* = \arg \min_{\boldsymbol{\lambda}} \|\tilde{\mathbf{X}} - K_{\mathbf{X},m}^-(\boldsymbol{\lambda})\|,$$

– решение задачи минимизации (7).

Нелинейное снижение размерности с помощью многообразий проектирования

В данном разделе будет описано семейство процедур снижения размерности $\Sigma(\text{NLPCA})$, основанных на построении нелинейного многообразия Ω_m размерности m , называемом многообразием проектирования, наиболее близко расположенного к векторам заданной выборки D_N (5).

Описание процедуры снижения размерности $\Sigma(\text{NLPCA})$

Процедура снижения размерности $\Sigma(\text{NLPCA}) = \{m, K_{\text{NLPCA}}, K_{\text{NLPCA}}^-\}$ сжимает данный вектор \mathbf{Y} (1) в сжатый вектор $\boldsymbol{\lambda}(\mathbf{Y})$ с помощью двух преобразований.

- преобразование сжатия $K_{\text{NLPCA}} : \mathbf{Y} \rightarrow \boldsymbol{\lambda} = K_m(\mathbf{Y}) \in R^m$,
- преобразование восстановления $K_{\text{NLPCA}}^- : \boldsymbol{\lambda} \rightarrow \mathbf{Y} = K_m^-(\boldsymbol{\lambda}) \in R^n$.

В соответствии с работой [5], преобразование восстановления определяет многообразии проектирования Ω_m , а преобразование сжатия заключается в проектировании заданного вектора \mathbf{Y} на многообразии проектирования.

Нелинейное многообразии Ω_m строится посредством нелинейной аппроксимации векторов остатков, лежащих в ортогональном дополнении к плоскости, образованной m первыми главными компонентами метода главных компонент [3]. При построении этого многообразия решается задача аппроксимации нелинейных зависимостей [2].

Конкретные процедуры $\Sigma(\text{NLPCA}_m)$ индексируются параметром m равным количеству выбранных главных компонент, координаты которых параметризуют нелинейное многообразии Ω_m .

Описание базового метода построения многообразия проектирования

Пусть $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$ – n -мерные собственные векторы, соответствующие собственным значениям $\{\mu_1, \mu_2, \dots, \mu_n\}$ выборочной ковариационной матрицы $S(D_N)$, построенной по выборке (5). При этом, $\mu_1 \geq \mu_2 \geq \dots \geq \mu_n$.

Опишем метод снижения размерности, использующий анализ главных компонент [3]. Пусть преобразование $K_{PCA,n} : \mathbf{Y} \rightarrow \boldsymbol{\lambda} = K_n(\mathbf{Y}) \in R^n$ преобразует каждый вектор \mathbf{Y}_r выборки (5) в n -мерный сжатый вектор:

$$\boldsymbol{\lambda}^{(r)}(\mathbf{Y}) = (\lambda_1^{(r)}(\mathbf{Y}), \lambda_2^{(r)}(\mathbf{Y}), \dots, \lambda_n^{(r)}(\mathbf{Y})),$$

где $\lambda_i^{(r)} = ((\mathbf{Y}_r - \mathbf{Y}_{\text{mean}}), \mathbf{e}_i)$, $i = 1, 2, \dots, n$, а $\mathbf{Y}_{\text{mean}} = \frac{1}{N} \sum_{r=1}^N \mathbf{Y}_r$, $r = 1, \dots, N$.

Предполагая, что данные имеют внутреннюю размерность m , $m \ll n$, следует ожидать, что компоненты вектора $\boldsymbol{\lambda}^{(r)}$ с индексами $i > m$ являются некоторыми функциями от переменных $\lambda_1^{(r)}, \lambda_2^{(r)}, \dots, \lambda_m^{(r)}$. Т.е., для любого r

$$\lambda_{m+1}^{(r)} = g_{m+1}(\lambda_1^{(r)}, \lambda_2^{(r)}, \dots, \lambda_m^{(r)});$$

...

$$\lambda_n^{(r)} = g_n(\lambda_1^{(r)}, \lambda_2^{(r)}, \dots, \lambda_m^{(r)}).$$

Эти зависимости определяют некоторую m -мерную поверхность Ω_m , расположенную в n -мерной плоскости, являющейся линейной оболочкой первых m главных компонент:

$$\Omega_m = \{\omega(\lambda_1, \lambda_2, \dots, \lambda_m) \mid \lambda_i \in R, i = 1, 2, \dots, m\}, \quad (8)$$

где $\omega(\lambda_1, \lambda_2, \dots, \lambda_m) = (\lambda_1, \lambda_2, \dots, \lambda_m, g_{m+1}(\lambda_1, \lambda_2, \dots, \lambda_m), \dots, g_n(\lambda_1, \lambda_2, \dots, \lambda_m))$. (9)

Функции $g_k(\lambda_1, \lambda_2, \dots, \lambda_m)$, $k = m+1, m+2, \dots, n$, подлежат восстановлению по исходной выборке данных D_N (5), для этого требуется решать задачу аппроксимации многомерных нелинейных зависимостей. Следует заметить, что функции g_k обязательно должны искажаться в классе нелинейных зависимостей, так как в противном случае мы получили бы простое переопределение базиса главных компонент.

Для аппроксимации многомерных нелинейных зависимостей применяются специальным образом построенные ансамбли искусственных нейронных сетей (ИНС) [6, 7, 8], где в качестве ИНС используется сеть специального вида – персептрон с одним скрытым слоем. Неизвестные нелинейные зависимости ищутся в следующем виде:

$$g_i(\lambda_1, \lambda_2, \dots, \lambda_m) = \sum_{k=1}^H h_k^i \cdot \sigma \left(\sum_{j=1}^m c_{k,j}^i \lambda_j + d_k^i \right) + h_0^i,$$

где H – количество нейронов в скрытом слое ИНС, а $h_k^i, c_{k,j}^i, d_k^i$, $k = \overline{1, H}$; $j = \overline{1, m}$; $i = \overline{m+1, n}$, есть веса ИНС, которые должны быть выбраны в процессе построения зависимостей. В качестве функции $\sigma(x)$ обычно используется сигмоидная функция $\sigma(x) = (1 + \exp(-x))^{-1}$.

Веса ИНС определяются как решения оптимизационных задач

$$\sum_{r=1}^N (\lambda_i^{(r)} - g_i(\lambda_1^{(r)}, \lambda_2^{(r)}, \dots, \lambda_m^{(r)}))^2 \rightarrow \min$$

для $i = \overline{m+1, n}$, которые можно решить, например, с помощью стандартного алгоритма обратного распространения ошибки [6, 7]. Критерием остановки работы алгоритма является возрастание ошибки на проверочном множестве. Значение параметра H можно

определить с помощью, так называемых конструктивных алгоритмов настройки числа нейронов в скрытом слое ИНС [7].

После того как задача аппроксимации функций $g_k(\lambda_1, \lambda_2, \dots, \lambda_m)$ решена, мы получаем параметрическое описание m -мерного многообразия Ω_m (8), (9) в координатах первых m главных компонент.

В исходных координатах многообразие Ω_m описывается вектор-функцией

$$Y(\lambda) = Y(\lambda_1, \lambda_2, \dots, \lambda_m) = Y_{\text{mean}} + \left(\sum_{i=1}^m \lambda_i \times e_i + \sum_{i=m+1}^n g_i(\lambda_1, \lambda_2, \dots, \lambda_m) \times e_i \right)$$

от переменных $(\lambda_1, \lambda_2, \dots, \lambda_m)$, где $\{e_1, e_2, \dots, e_n\}$ – векторы PCA базиса.

Результаты моделирования

Для проверки предложенного метода была использована выборка 5418 контуров профилей крыла самолета и распределения давления на них. Контурные профили представляют собой 58-мерные вектора координат (по оси ординат) в фиксированных точках (по оси абсцисс). Расположение профилей выбирается так, чтобы ординаты двух крайних точек отличались только знаком. Таким образом, остается 57 значимых координат. Распределения давления представляют собой 57-мерные вектора значений давления в точках, определенных контурами профилей.

Объем обучающей выборки составлял 90% от всей выборки, проверочной – 10%. При моделировании по очереди предполагалось, что данные распределения давления или профили крыла в проверочном множестве неизвестны. С помощью процедур, описанных в предыдущих разделах, эти данные восстанавливались по известным значениям. Количество используемых главных компонент в (8), (9) определялось по минимальной среднеквадратичной ошибке восстановления пропущенных данных на обучающей выборке. Количество аппроксимируемых главных компонент выбиралось следующим образом: аппроксиматоры добавлялись до тех пор, пока относительная ошибка на обучающем множестве при добавлении нового аппроксиматора уменьшалась более чем на 1%.

Результаты для среднеквадратичных ошибок на проверочном множестве приведены в табл. 1 и 2.

Таблица 1. Ошибки восстановления профилей крыла по распределению давлений

	7 главных компонент	8 главных компонент	7 главных компонент, один аппроксиматор
ошибка среднеквадратичная	0,0029	0,0049	0,0026

Таблица 2. Ошибки восстановления распределений давления по профилям крыла

	7 главных компонент	8 главных компонент	7 главных компонент, один аппроксиматор
ошибка среднеквадратичная	1,14e-6	1,48e-6	1,11e-6

Заключение

В работе рассмотрен геометрический подход к восстановлению пропущенных данных. Подход основан на совместном использовании нелинейной процедуры снижения размерности с помощью многообразий проектирования и процедуры ортогонального проектирования на заданное параметрическое многообразие. Применение разработанных численных алгоритмов к решению задачи восстановления

описания профиля крыла самолета по заданному распределению давления на профиле позволило восстановить профиль с высокой точностью. В дальнейшем планируется обобщить предложенный метод на случай, когда в обучающей выборке также содержатся пропуски.

Работа выполнена при частичной поддержке гранта РФФИ 09-01-00584-а.

Литература

1. Айвазян С.А., Енюков С.А., Мешалкин Л.Д. Прикладная статистика. Основы моделирования и первичная обработка данных. – М.: Финансы и статистика. – 1983. – 472 с.
2. Айвазян С.А., Енюков И.С., Мешалкин Л.Д. Прикладная статистика. Исследование зависимостей. М.: Финансы и статистика. – 1985. – 487 с.
3. Айвазян С.А., Бухштабер В.М., Енюков С.А., Мешалкин Л.Д. Прикладная статистика. Классификация и снижение размерности. М.: Финансы и статистика. – 1989. – 607 с.
4. Little R.J.A., Rubin D.B. Statistical analysis with missing data. Wiley. – 1987.
5. Бернштейн А.В., Бурнаев Е.В., Дорофеев Е.А., Свириденко Ю.Н., Чернова С.С. Каскадные процедуры снижения размерности // Труды одиннадцатой национальной конференции по искусственному интеллекту с международным участием КИИ-08 (28 сентября – 3 октября 2008 г., г. Дубна, Россия): Труды конференции. – Т.1. – М.: ЛЕНАНД. – 2008. – С. 241–250.
6. Дорофеев Е.А., Свириденко Ю.Н. Введение в нейроинформатику // Труды Центрального аэрогидродинамического института им. проф. Н.Е. Жуковского, Выпуск №2678 «Применение искусственных нейронных сетей в задачах прикладной аэродинамики». – Москва. – 2008.
7. Хайкин С. (2006) Нейронные сети. Полный курс. – М.: Вильямс. – 2006. – 1104 с.
8. Бернштейн А.В., Кулешов А.П., Бурнаев Е.В. Об одной методологии построения аппроксимации многомерных зависимостей. Труды IV Международной конференции «Параллельные вычисления и задачи управления». Москва. – 2008.

ИТЕРАТИВНЫЕ АЛГОРИТМЫ РАНЖИРОВАНИЯ С ПРИМЕНЕНИЕМ К НАУКОМЕТРИИ И ИНТЕРНЕТУ

А.В. Тимонина

(Московский физико-технический институт (государственный университет))

Научный руководитель – д.т.н., профессор Б.Т. Поляк

(Институт проблем управления РАН)

Изучаются системы большой размерности на примере сети Internet. Исследование проводится с целью разработки алгоритма, который позволит наиболее точно находить PageRank Internet-страницы. В работе рассмотрена упрощенная модель сети Internet, на примере которой демонстрируется наличие существенной ошибки между точным значением «веса» страницы и значением, полученным из алгоритма PageRank. Найден способ точного вычисления PageRank.

Ключевые слова: собственный вектор, алгоритм, PageRank, точный метод

Введение

Рассматривается сеть Internet и ранжирование сайтов в порядке значимости (по их весам). Известен алгоритм PageRank [1], используемый Google для нахождения веса Internet-страницы. Данный алгоритм основан на приближенном поиске собственного вектора матрицы $P = (p_{ij})$, элементами которой p_{ij} являются вероятности перехода с сайта i на сайт j за 1 шаг. В данной работе на примере модели Internet-сети будет показано наличие существенной ошибки между точным и приближенным собственными векторами. Найденны формулы для точного расчета собственного вектора, элементами которого будут являться веса сайтов.

Тестовая задача

Рассмотрим следующую задачу:

Пусть задана модель Internet-сети (рис. 1, рис. 2).

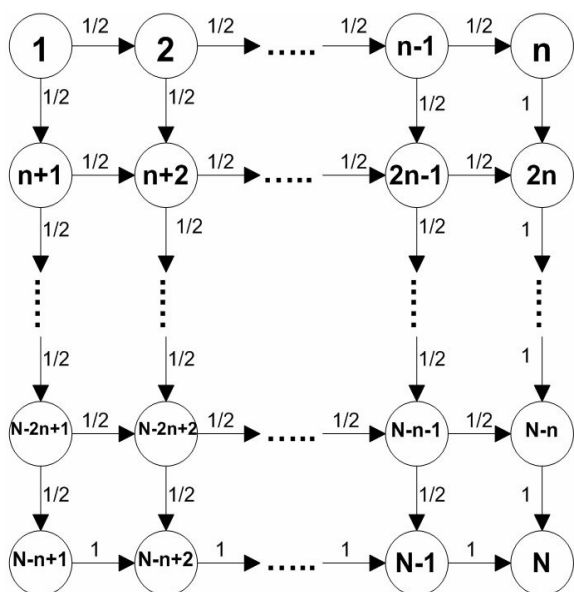


Рис. 1. Граф переходов между состояниями

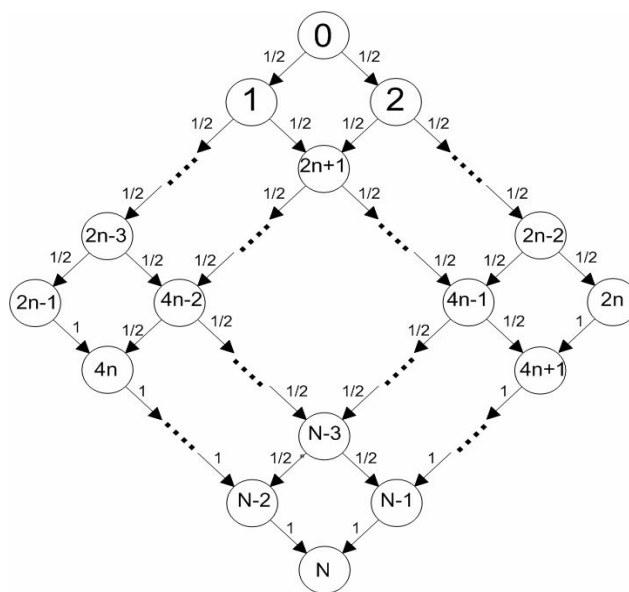


Рис. 2. Граф переходов между состояниями с измененной нумерацией сайтов

Изначально, на сайте 1 находится человек (рис. 1). С каждого сайта ведут ссылки, показанные стрелками. При этом сайт $N=n^2$ ссылается на все другие сайты в сети. Пусть человек ходит по этой сети. Вероятности перейти с одного сайта на другой показаны над стрелками. При этом с сайта $N=n^2$ человек может попасть на любой другой либо остаться с вероятностями $\frac{1}{n^2}$.

Нас интересует вероятность нахождения человека на сайте (i, j) через большое число шагов.

Вершине (i, j) поставим в соответствие число x_{ij} , так чтобы выполнялось $x^*_{ij} = \frac{x_{ij}}{\sum_{i,j} x_{ij}}$, и чтобы число x^*_{ij} являлось вероятностью нахождения человека на сайте

(i, j) . Если P – матрица переходов между сайтами (т.е. ее элементами p_{ij} являются вероятности перехода с сайта i на сайт j за 1 шаг), то вектор x^* , составленный из чисел x^*_{ij} , будет являться собственным вектором матрицы P : $x^* P = x^*$, принадлежащим собственному числу $\lambda = 1$ [2].

Заметим сразу, что на каком бы сайте не находился человек, через определенное число шагов он окажется на сайте $N=n^2$ с вероятностью 1, при этом в вершину $(1, 1)$ человек может попасть только из вершины $N=n^2$ с вероятностью $\frac{1}{n^2}$, откуда нетрудно заключить, что вероятность нахождения человека в вершине (n, n) в n^2 раз больше, чем вероятность нахождения человека в вершине $(1, 1)$:

$$x^*_{11} = \frac{1}{n^2} x^*_{nn}, \text{ а значит выполняется } x_{11} = \frac{1}{n^2} x_{nn}.$$

Зададим число x_{nn} произвольно и посчитаем все другие числа x_{ij} . Пусть $x_{nn} = n^2$.

Ясно, что тогда $x_{11} = 1$. Теперь запишем правила нахождения чисел x_{ij} :

$$x_{i1} = \frac{1}{2} x_{i-11} + 1, \quad i = 2, \dots, n, \quad x(1,1) = 1 \quad (1)$$

$$x_{ij} = \frac{1}{2} x_{i-1j} + \frac{1}{2} x_{ij-1} + 1, \quad i, j = 2, \dots, n-1 \quad (2)$$

$$x_{nj} = x_{n-1j} + \frac{1}{2} x_{nj-1} + 1, \quad j = 2, \dots, n-1 \quad (3)$$

$$x_{nn} = x_{n-1n} + x_{nn-1} + 1 = n^2. \quad (4)$$

Составим из чисел $a(i, j) = 2^i x(i, j)$ таблицу, изменив нумерацию строк так, как показано в таблице; и сравним получившуюся таблицу с треугольником Паскаля [3].

Так как элементами треугольника Паскаля являются числа C_i^j , то введем обозначения для элементов разности: $b_{ij} = a_{ij} - C_i^j$.

Теперь можно записать формулы для a_{ij} и b_{ij} , учитывая тождество $C_i^j = C_{i-1}^{j-1} + C_{i-1}^j$:

$$a_{ij} = a_{i-1j-1} + a_{i-1j} + 2^i \quad (5)$$

$$a_{i0} = 2a_{i-10} + 1 = 2^{i+1} - 1$$

$$b_{ij} = b_{i-1j-1} + b_{i-1j} + 2^i \quad (6)$$

$$b_{i0} = 2b_{i-10} + 2 = 2^{i+1} - 2.$$

Таблица чисел $a(i, j) = 2^i x(i, j)$								Треугольник Паскаля							Разность этих двух таблиц								
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
							255								1								254
							127								1								126
							63								7								62
							31								6								30
							15								21								14
							7								15								6
							3								35								22
							10								20								8
							3								35								22
							7								15								6
							15								21								14
							31								6								30
							63								7								62
							127								1								126
							255								1								254

Видим, что формулы (5) и (6) на вид одинаковы, но из таблицы видно, что числа b_{ij} , четные, тогда как a_{ij} есть как четные, так и нечетные.

Будем искать b_{ij} в виде:

$$b_{ij} = \sum_{k=1}^i P_{ij}(k) 2^k.$$

Подставим данное равенство в уравнение (6) и проверим выполняется ли оно:

$$P_{ij}(i) 2^i + \sum_{k=1}^{i-1} [P_{ij}(k) 2^k - P_{i-1, j-1}(k) 2^k - P_{i-1, j}(k) 2^k] = 2^i$$

Это равенство верно в случае, если верна система:

- $P_{ij}(i) = 1$;
- $P_{ij}(k) = P_{i-1, j-1}(k) + P_{i-1, j}(k)$.

Данные равенства выполняются для биномиальных коэффициентов $P_{ij}(k) = C_{i-k}^j$ при $k \leq i - j$ и для $P_{ij}(k) = C_j^{i-k}$ при $i - j < k \leq i$.

Отсюда запишем формулу для b_{ij} :

$$b_{ij} = \sum_{k=1}^{i-j} C_{i-k}^j 2^k + \sum_{k=i-j+1}^i C_j^{i-k} 2^k$$

Отсюда следует выражение для a_{ij} :

$$a_{ij} = \sum_{k=1}^{i-j} C_{i-k}^j 2^k + \sum_{k=i-j+1}^i C_j^{i-k} 2^k + C_i^j$$

тогда, отсюда и из формулы бинома Ньютона $(a + b)^n = \sum_{k=0}^n C_n^k a^k b^{n-k}$ имеем:

$$x_{ij} = \frac{\sum_{k=1}^{i-j} C_{i-k}^j 2^k + \sum_{k=i-j+1}^i C_j^{i-k} 2^k + C_i^j}{2^i} = \frac{\sum_{k=1}^{i-j-1} C_{i-k}^j 2^k + 2^{i-j} 3^j + C_i^j}{2^i}. \quad (7)$$

Из формулы (7) можно точно вычислить элементы x_{ij} вектора x , а значит и элементы x^*_{ij} собственного вектора матрицы P для $\lambda = 1$.

Для удобства ранжирования можно сделать так, чтобы элементы вектора x^* возрастали монотонно с возрастанием номера элемента. Этого можно добиться, изменив нумерацию сайтов так, как показано на рис. 2.

Приведем результаты моделирования сходимости степенного метода $x_k = x_{k-1}P$ [4]. Как видно из рис. 3–6 для Internet-сети размера $n \times n$ нормы $\|x_k - x^*\|_{1,2,\infty} \rightarrow 0$.

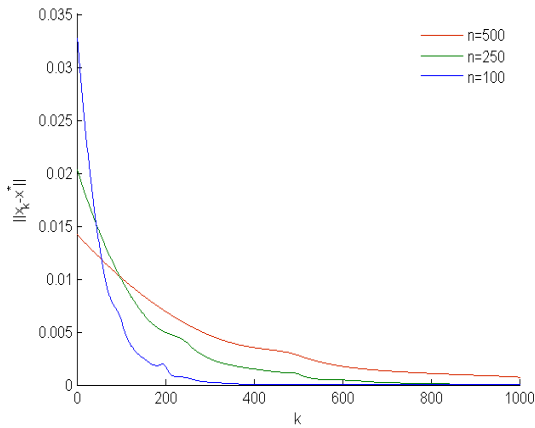


Рис. 3. Сходимость $\|x_k - x^*\|$ для разных n

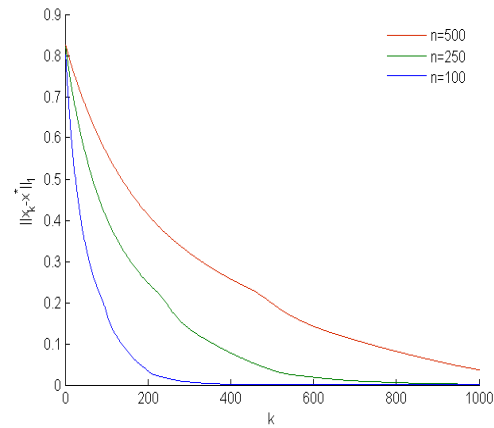


Рис. 4. Сходимость $\|x_k - x^*\|_1$ для разных n

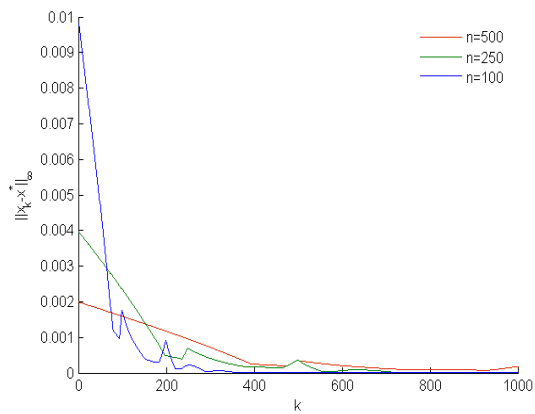


Рис. 5. Сходимость $\|x_k - x^*\|_\infty$ для разных n

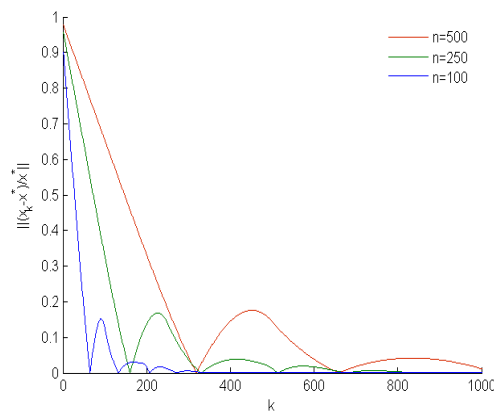


Рис. 6. Сходимость $\left\| \frac{x_k - x^*}{x^*} \right\|$ для разных n

Теперь покажем, что собственный вектор, найденный по алгоритму PageRank, существенно отличается от точного собственного вектора.

Исследование степенного метода и M_α

Рассмотрим матрицу $M = \alpha P + (1 - \alpha)S$, где S – матрица размера $n^2 \times n^2$ вида:

$$\begin{pmatrix} \frac{1}{n^2} & \cdots & \frac{1}{n^2} \\ \vdots & \ddots & \vdots \\ \frac{1}{n^2} & \cdots & \frac{1}{n^2} \end{pmatrix}.$$

В алгоритме PageRank ищется собственный вектор матрицы M и считается, что он будет близок к собственному вектору матрицы P при $\alpha \approx 0.85$ [5].

Найдем правило, по которому можно найти собственный вектор матрицы M точно. Обозначим этот вектор через y , тогда должно выполняться

$$yM = \alpha yP + (1 - \alpha)yS = y.$$

Запишем это уравнение в координатном виде:

$$\alpha \sum_{i=1}^{n^2} y_i p_{ij} + \frac{1-\alpha}{n^2} \sum_{i=1}^{n^2} y_i = y_j. \quad (8)$$

В матрице P известны все ненулевые элементы.

- в столбце $2 \leq j \leq n$ ставим $\frac{1}{2}$ на $j-1$ место;
- в столбце $nk+1 \leq j \leq n(k+1)-1$, где $k=1, \dots, n-2$, ставим $\frac{1}{2}$ на $j-1$ место и $\frac{1}{2}$

на $j-n$ место;

- в столбце $n^2 - n + 1 \leq j \leq n^2 - 1$ ставим 1 на $j-1$ место;
- в столбце $j = nk$, где $k=2, \dots, n-1$, ставим 1 на $j-n$ место;
- в столбце $j = n^2$ ставим 1 на $j-1$ и $j-n$ место;
- во всех столбцах ставим $\frac{1}{n^2}$ на n^2 место.

Теперь, в формулу (8) подставим ненулевые элементы матрицы P , откуда получим правила нахождения элементов вектора y :

Так как в первом столбце ненулевой элемент только $\frac{1}{n^2}$ в строке n^2 , то

$$y_1 = \frac{\alpha}{n^2} y_{n^2} + \frac{1-\alpha}{n^2} \sum_{i=1}^{n^2} y_i,$$

далее получаем

$$y_j = \frac{\alpha}{2} y_{j-1} + y_1, \quad j = 2, \dots, n,$$

$$y_j = \frac{\alpha}{2} y_{j-n} + y_1, \quad j = kn+1, \quad k = 1, \dots, n-1,$$

$$y_j = \frac{\alpha}{2} y_{j-n} + \frac{\alpha}{2} y_{j-1} + y_1, \quad nk+2 \leq j \leq n(k+1)-1, \quad k = 1, \dots, n-2,$$

$$y_j = \frac{\alpha}{2} y_{j-n} + \alpha y_{j-1} + y_1, \quad j = n^2 - n + 1, \dots, n^2 - 1,$$

$$y_j = \alpha y_{j-n} + \frac{\alpha}{2} y_{j-1} + y_1, \quad j = nk, \quad k = 2, \dots, n-1,$$

$$y_{n^2} = \alpha y_{n^2-n} + \alpha y_{n^2-1} + y_1.$$

Так же можно переписать эти уравнения в обозначениях уравнений (1)–(4):

$$y_{i1} = \frac{\alpha}{2} y_{i-11} + y_{11}, \quad i = 2, \dots, n$$

$$y_{ij} = \frac{\alpha}{2} y_{i-1j} + \frac{\alpha}{2} y_{ij-1} + y_{11}, \quad i, j = 2, \dots, n-1$$

$$y_{nj} = \alpha y_{n-1j} + \frac{\alpha}{2} y_{nj-1} + y_{11}, \quad j = 2, \dots, n-1$$

$$y_{nn} = \alpha y_{n-1n} + \alpha y_{nn-1} + y_{11}.$$

Так как при $\alpha = 1$ должны получиться уравнения (1)–(4), то $y_{11}(\alpha = 1) = 1$. Пусть $y_{11} = \alpha$.

Получим уравнения:

$$y_{i1} = \alpha \left(\frac{1}{2} y_{i-11} + 1 \right), \quad i = 2, \dots, n$$

$$y_{ij} = \alpha \left(\frac{1}{2} y_{i-1j} + \frac{1}{2} y_{ij-1} + 1 \right), \quad i, j = 2, \dots, n-1$$

$$y_{nj} = \alpha \left(y_{n-1j} + \frac{1}{2} y_{nj-1} + 1 \right), \quad j = 2, \dots, n-1$$

$$y_{nn} = \alpha (y_{n-1n} + y_{nn-1} + 1).$$

Действуя аналогично случаю $\alpha = 1$, получим числа $q_{ij} = \frac{2^i}{\alpha^i} y_{ij}$, для которых

$$q_{ij} = q_{i-1j-1} + q_{i-1j} + \frac{2^i}{\alpha^i}.$$

Для этих чисел верно равенство:

$$q_{ij} = \sum_{k=1}^{i-j} C_{i-k}^j \frac{2^k}{\alpha^k} + \sum_{k=i-j+1}^i C_j^{i-k} \frac{2^k}{\alpha^k} + C_i^j.$$

Отсюда следует выражение для y_{ij} :

$$y_{ij} = \frac{\alpha^i}{2^i} \left(\sum_{k=1}^{i-j} C_{i-k}^j \frac{2^k}{\alpha^k} + \sum_{k=i-j+1}^i C_j^{i-k} \frac{2^k}{\alpha^k} + C_i^j \right) = \frac{\alpha^i}{2^i} \left(\sum_{k=1}^{i-j-1} C_{i-k}^j \frac{2^k}{\alpha^k} + 2^{i-j} \left(1 + \frac{2}{\alpha} \right)^j + C_i^j \right).$$

Из этой формулы следует наличие существенной ошибки между вектором x^* и вектором $y^* = \frac{y}{\sum_{i,j} y_{i,j}}$. При этом ошибка увеличивается при увеличении (i, j) .

Заключение

Путем исследований доказано наличие существенной ошибки между точным значением веса Internet-страницы и значением, полученным из алгоритма PageRank.

Главным результатом работы являются формулы, которые позволяют точно находить элементы собственного вектора матрицы переходов между сайтами, а значит и веса сайтов.

Литература

1. Langville A.N., Meyer C.D. Google's PageRank and Beyond: The Science of Search Engine Ranking. – Princeton University. – 2006. – 256 p.
2. Чистяков В.П. Курс теории вероятностей: Учебник для вузов. – 5-е изд., перераб. и доп. – М.: Агар. – 2000. – 256 с.
3. Успенский В.А. Треугольник Паскаля. – М.: Наука. – 1979. – 50 с.
4. Фадеев Д.К., Фадеева В.Н. Вычислительные методы линейной алгебры. – М.: ФИЗМАТГИЗ. – 1960. – 656 с.
5. Bryan K., Leise T. The \$25,000,000,000 Eigenvector: The Linear Algebra behind Google //Siam Review. – 2006. – №3. – PP. 569–581.

НЕЙРОНЕЧЕТКОЕ УПРАВЛЕНИЕ ЭЛЕКТРОМЕХАНИЧЕСКИМИ ОБЪЕКТАМИ С УПРУГИМИ СВЯЗЯМИ

В.К. Фьонг, Н.К. Чьен

(Санкт-Петербургский государственный электротехнический университет)

Научный руководитель – д.т.н., профессор В.В. Путов

(Санкт-Петербургский государственный электротехнический университет)

Исследуются методы построения и обучения нейронечетких регуляторов с правилами TSK для управления упругими электромеханическими объектами. Оценивается эффективность подавления упругих колебаний и точность управления в условиях изменения параметров в упругих объектах.

Ключевые слова: адаптивное управление, нечеткое управление, нейронное управление, нейронечеткая сеть, электромеханическая система

Введение

Применение нейронечетких систем является новым подходом для решения задач управления многомассовыми нелинейными упругими электромеханическими объектами. Нейронечеткие системы основаны на синтезе методов нечеткого управления и нейронных сетей. Нейронечеткие системы используют нечеткие правила и нейронный механизм реализации этих правил, обладая тем самым преимуществами как нейронных сетей, так и нечетких систем управления [1, 2].

Среди различных типов существующих нейронечетких регуляторов наиболее эффективным, на наш взгляд, является регулятор с нечеткими правилами TSK (Takagi-Sugeno-Kang) [1]. В регуляторах этого типа выполняется параметрическое обучение, т.е. настройке подлежат функции принадлежности и другие параметры нейронечеткой сети управления, архитектура которой устанавливается предварительно на основе опыта или экспертных знаний.

Нейронечеткие регуляторы

Главной особенностью нечетких правил TSK является то, что их заключения есть линейные комбинации входных переменных вида

$$R_j: \text{если } u_1 \text{ есть } A_{1j} \text{ и } u_2 \text{ есть } A_{2j} \text{ и } \dots \text{ и } u_n \text{ есть } A_{nj}, \quad (1)$$

то $y = f_j = b_{0j} + b_{1j}u_1 + b_{2j}u_2 + \dots + b_{nj}u_n$,

где u_i – входные переменные, y – выходная переменная, $A_{i,j}$ – лингвистические значения входных переменных с функциями принадлежности $\mu_{A_{ij}}(u_i)$, b_{ij} – коэффициенты линейных уравнений $f_j = (u_1, u_2, \dots, u_n)$, $j = 1, 2, \dots, m$, $i = 1, 2, \dots, n$.

Для простоты предположим, что нечеткая система управления имеет два входа u_1, u_2 , один выход y , а база правил содержит два следующих нечетких правила TSK

$$R_1: \text{если } u_1 \text{ есть } A_{11} \text{ и } u_2 \text{ есть } A_{21}, \text{ то } y = f_1 = b_{01} + b_{11}u_1 + b_{21}u_2, \quad (2)$$

$$R_2: \text{если } u_1 \text{ есть } A_{12} \text{ и } u_2 \text{ есть } A_{22}, \text{ то } y = f_2 = b_{02} + b_{12}u_1 + b_{22}u_2. \quad (3)$$

Выходное управляющее воздействие y^* определяется по формуле

$$y^* = \frac{\alpha_1 f_1 + \alpha_2 f_2}{\alpha_1 + \alpha_2}, \quad (4)$$

где α_j – уровень истинности правила R_j , $j = 1, 2$, определяется как

$$\alpha_j = \min(\mu_{A_{1j}}(u_1), \mu_{A_{2j}}(u_2)), j = 1, 2. \quad (5)$$

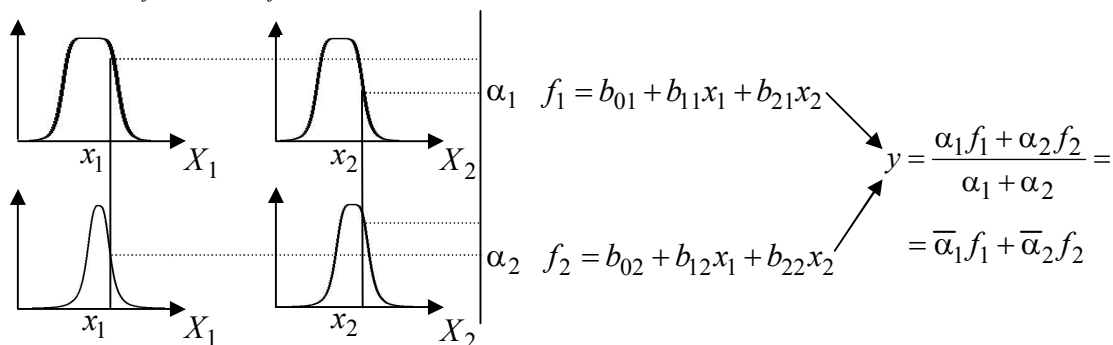


Рис. 1. Структура системы нечеткого вывода

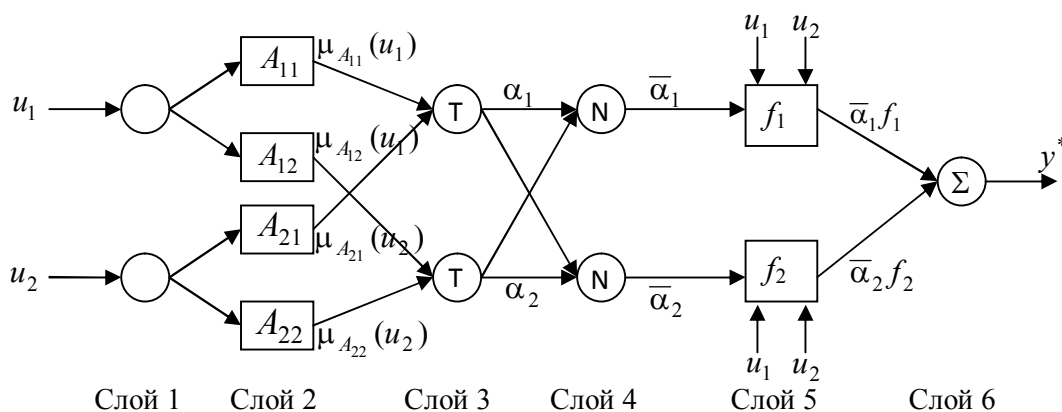


Рис. 2. Структура нейронечеткого регулятора с правилами TSK

Архитектура рассмотренной нечеткой системы управления представлена на рис. 2, которая состоит из шести слоев и имеет вид нейронной сети [1]. Каждый узел сети характеризуется узловой функцией. Если узловая функция зависит от параметров узла, то узел является настраиваемым и обозначается в виде квадрата. Параметры этих узлов подлежат настройке методами параметрического обучения. Если узловая функция фиксирована, то узел является фиксированным и обозначается в виде круга.

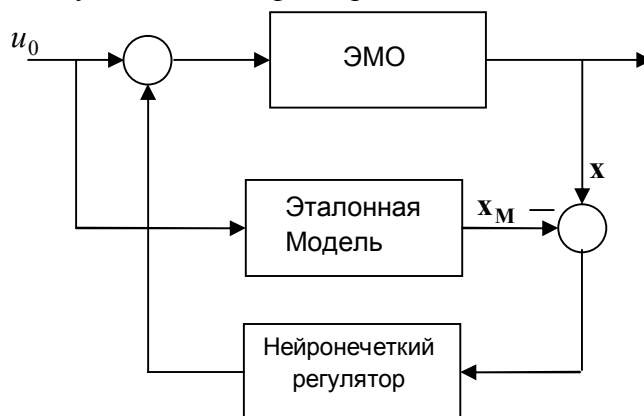


Рис. 3. Нейронечеткое управление с эталонной моделью и обратной связью по ошибке

Для подавления упругих колебаний и обеспечения заданного быстродействия переходных процессов электромеханического объекта (ЭМО) в условиях неопределенности его параметров и наличия нелинейностей в виде зазоров в упругих связях и сухого трения вводится нейронечеткий регулятор с правилами TSK. Рассмотрим нейронечеткую систему с эталонной моделью (рис. 3) для управления двухмассовым упругим электромеханическим объектом.

Объект управления

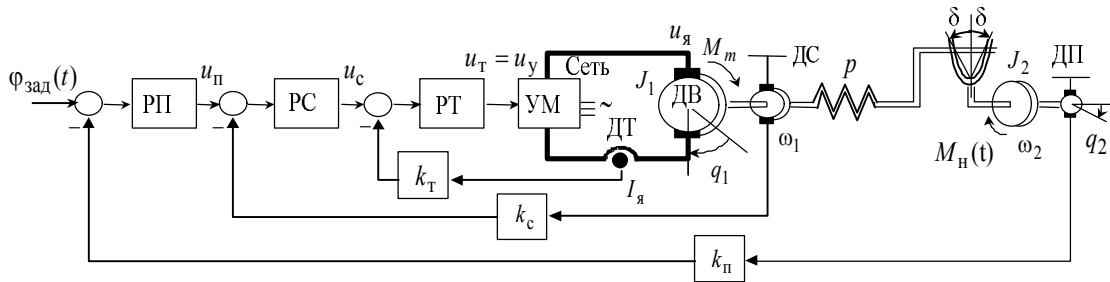


Рис. 4. Трехконтурная следящая система с подчиненным управлением и двухмассовым упругим нелинейным электромеханическим объектом

Рассмотрим в качестве исходного двухмассовый электромеханический упругий объект с крутильными колебаниями, схема которого представляется на рис. 4 [4], где J_1 – момент инерции исполнительного привода с частью жестко соединенных с ним инерционных звеньев механизма; J_2 – момент инерции остальной части механической конструкции, приведенной к вращению исполнительного привода; p – коэффициент упругости упругой связи между дисками J_1, J_2 ; δ – угловая величина, равная половине зазора, имеющегося в сочленениях; ДВ – двигатель; ДП – датчик углового положения; ДС – датчик угловой скорости; ДТ – датчик тока; УМ – усилитель мощности; РТ – контурный регулятор тока; РС – контурный регулятор угловой скорости; РП – контурный регулятор углового положения; $\Phi_{зад}(t)$ – программное задающее воздействие; q_1, q_2 – угловые положения первого и второго дисков соответственно; ω_1, ω_2 – угловые скорости первого и второго дисков соответственно; u_T, u_c, u_{π} – выходные напряжения контурных регуляторов РТ, РС, и РП соответственно; k_T, k_c, k_{π} – постоянные коэффициенты передачи датчиков обратных связей по току ДТ, скорости ДС, и положению ДП соответственно; M_m – электромагнитный момент электропривода постоянного тока, являющийся управляющим воздействием, приложенным к механическому объекту и формируемым в рамках рассматриваемой электромеханической следящей системы с подчиненным управлением; $M_H(t)$ – возмущающий момент нагрузки; u_y – управляющее входное напряжение управляемого электронного выпрямителя-усилителя мощности УМ, подключенного к якорной обмотке исполнительного привода постоянного тока.

Пренебрегая электромагнитной инерцией электропривода и звенья РТ, ДТ, k_T контура тока, представим математическое описание упругого электромеханического объекта с подчиненным управлением в виде системы дифференциальных уравнений вида

$$\left. \begin{aligned} \dot{\omega}_1 &= J_1^{-1}(-m_y + M_m); \dot{\omega}_2 = J_2^{-1}(m_y); \dot{m}_y = p(\omega_1 - \omega_2); \dot{\phi} = \omega_2; \\ M_m &= k_m / R_y (k_y u_c - k_e \omega_1); u_c = \beta_c [(u_{\Sigma} - k_{\pi} \phi) \beta_{\pi} - k_c \omega_1]; u_{\Sigma} = \Phi_{зад} + u_{\pi}, \end{aligned} \right\} (6)$$

которую перепишем в векторно-матричной форме:

$$\left. \begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{b}u_{\Sigma}; \mathbf{y} = \mathbf{c}^T \mathbf{x}; \\ \mathbf{A} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & a_1 & 0 \\ 0 & -a_2 & 0 & a_2 \\ a_3 & 0 & a_4 & a_5 \end{bmatrix}; \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ b \end{bmatrix}; \mathbf{c} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ c \end{bmatrix}; \mathbf{x} = \begin{bmatrix} \phi \\ \omega_2 \\ m_y \\ \omega_1 \end{bmatrix}, \end{aligned} \right\} (7)$$

где

$$\left. \begin{aligned} a_1 = J_2^{-1}; a_2 = p; a_3 = -J_1^{-1}R_{я}^{-1}k_mk_yk_{п}\beta_c\beta_{п}; a_4 = -J_1^{-1}; \\ a_5 = -J_1^{-1}R_{я}^{-1}k_m(k_e + k_yk_c\beta_c); b = -J_1^{-1}R_{я}^{-1}k_mk_y\beta_c\beta_{п}, \end{aligned} \right\} \quad (8)$$

где $\beta_c, \beta_{п}$ – коэффициенты усиления регуляторов РС и РП.

Нейронечеткое управление двухмассовым упругим электромеханическим объектом с эталонной моделью

Для решения задачи подавления упругих колебаний и обеспечения заданного быстродействия переходных процессов в качестве эталонной модели выбирается замкнутая система по модальному управлению. Модальный регулятор для линеаризованного объекта (6) имеет вид обратной связи по состоянию

$$u_{п} = \mathbf{kx} = k_1\varphi + k_2\omega_2 + k_3m_y + k_4\omega_1. \quad (9)$$

Тогда эталонная модель имеет вид

$$\dot{\mathbf{x}}_{м} = \mathbf{A}_{м}\mathbf{x}_{м} + \mathbf{b}_{м}u^0, \quad (10)$$

где матрица $\mathbf{A}_{м}$ и вектор $\mathbf{b}_{м}$ определяются из замкнутой системы обратных связей по состоянию линеаризованного объекта (7)

$$\mathbf{A}_{м} = \mathbf{A} + \mathbf{bk}, \mathbf{b}_{м} = \mathbf{b}. \quad (11)$$

Обучение параметров нейронечеткого регулятора с правилами TSK и эталонной моделью (рис. 3) предлагается осуществлять на основе обучающей аналитической адаптивной системы с сигнальной настройкой. Алгоритм сигнальной адаптации настройки имеет вид [3, 4]

$$u_a(t) = -h_0 \text{sign} \mathbf{b}_{м}^T \mathbf{P}(\mathbf{e} - \mathbf{e}_{м}), \quad (12)$$

где h_0 – положительный коэффициент усиления, \mathbf{P} – симметричная положительно определенная матрица, удовлетворяющая уравнению Ляпунова вида

$$\mathbf{A}_{м}^T \mathbf{P} + \mathbf{P} \mathbf{A}_{м} = -\mathbf{G}, \mathbf{G} = \mathbf{G}^T > 0. \quad (13)$$

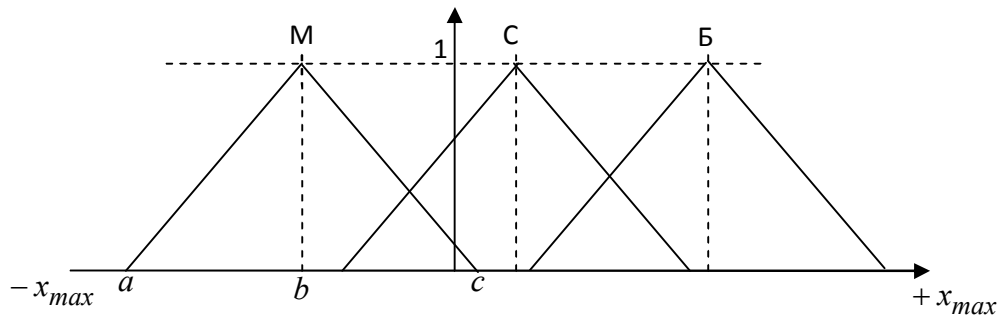


Рис. 5. Функции принадлежности (термы) для каждого входа

Архитектура нейронечетких регуляторов с правилами TSK выбрана следующим образом: каждый вход обоих регуляторов имеет 3 функции принадлежности треугольного вида, представленные на рис. 5, где М (малое), С (среднее), Б (большое) – лингвистические значения входных переменных; a, b, c – координаты вершин треугольников. Нейронечеткая архитектура сети регулятора с двумя входами (рис. 3) будет иметь 6 узлов во втором слое, 9 узлов в третьем, четвертом и пятом слоях и 18 связей между вторым и третьим слоями. Параметры пятого слоя являются коэффициентами линейных уравнений первого порядка от двух входов вида $f = b_0 + b_1x_1 + b_2x_2$. Выбранная архитектура нейронечетких регуляторов с правилами TSK в редакторе ANFIS пакета MATLAB представлена на рис. 6.

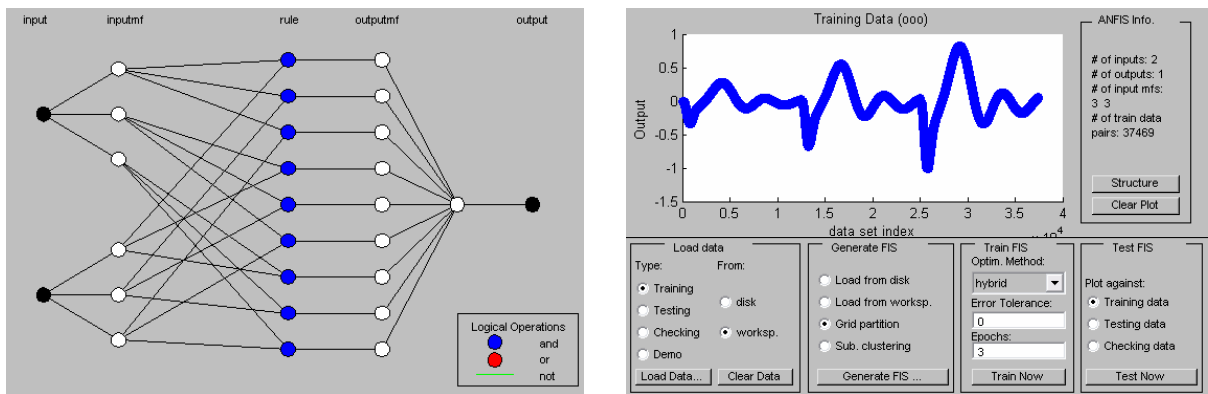


Рис. 6. Архитектура нейронечетких регуляторов с правилами TSK в редакторе ANFIS пакета MATLAB

Результаты моделирования

В Matlab – Simulink построена программа предлагаемого адаптивного обучения нейронечеткого регулятора с правилами TSK. Данные для обучения получены на входе и на выходе адаптивного регулятора (12) в процессе его работы с разными задающими воздействиями u_0 и с изменением параметров объекта. В работе проведено моделирование нейронечеткой системы управления с рассматриваемым двухмассовым упругим электромеханическим объектом с эталонной моделью. Результаты исследования эффективности работы адаптивной нейронечеткой системы управления (ННСУ) (рис. 7) при изменении параметров объекта сопоставляются с результатами работы адаптивной системы управления с сигнальной настройкой (АСУСН) (рис. 8) таким же объектом.

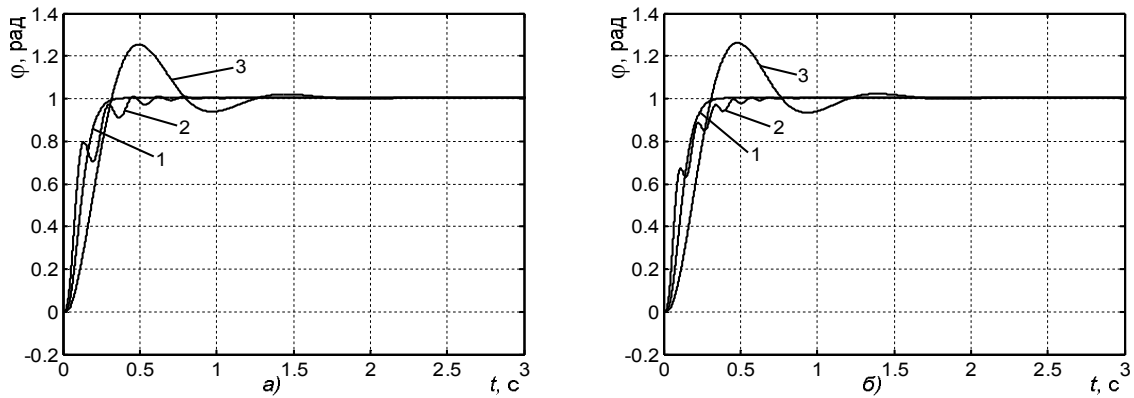


Рис. 7. Переходные процессы в ННСУ с наблюдателем состояния

На рис. 7а и рис. 8а показаны графики переходных процессов угла поворота нагрузки при работе ННСУ и АСУСН соответственно при изменении коэффициента упругости p , где кривая 1 – при $p = p_0$; кривая 2 – при $p = 3p_0$; кривая 3 – при $p = p_0/3$. На рис. 7б и рис. 8б показаны графики переходных процессов угла поворота нагрузки при работе ННСУ и АСУСН соответственно при изменении момента инерции второго диска J_2 , где кривая 1 – при $J_2 = J_{02}$; кривая 2 – при $J_2 = J_{02}/3$; кривая 3 – при $J_2 = 3J_{02}$.

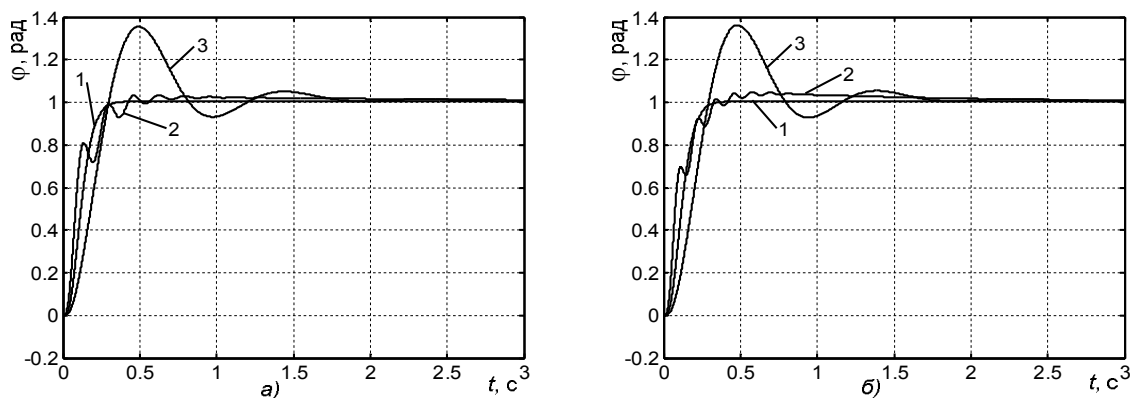


Рис. 8. Переходные процессы в АСУСН с наблюдателем состояния

Заключение

При девятикратном изменении параметров p или J_2 работоспособность нейронечеткой системы управления с наблюдателем состояния сохраняется при удовлетворительном качестве переходных процессов (кривые 2, 3 на рис. 7). Сравнение переходных процессов на рис. 7 и 8 позволяет говорить о том, что хотя нейронечеткая система обучена на основе обучающей аналитической адаптивной системы с сигнальной настройкой, но в силу эвристичности интеллектуальной системы приобретаемые ею адаптивные свойства могут превосходить адаптивные свойства самой обучающей системы, и эффективность нейронечеткой адаптивной системы при глубоких изменениях параметров всегда выше эффективности адаптивной системы с сигнальной настройкой.

Литература

1. Chin-Teng Lin C.S. George Lee. Neural Fuzzy Systems – A Neuro-Fuzzy Synergism to Intelligent Systems: Prentice-Hall International, Inc. – 1996.
2. Поляхов Н.Д., Приходько И.А. Нечеткие системы управления: Учеб. пособие. СПб: Изд-во СПбГЭТУ «ЛЭТИ». – 2003.
3. Путов В.В. Адаптивное управление динамическими объектами: беспоисковые системы с эталонными моделями: Учеб. пособие. СПб: Изд-во СПбГЭТУ «ЛЭТИ». – 2001.
4. Путов В.В., Шелудько В.Н. Адаптивные и модальные системы управления многомассовыми нелинейными упругими механическими объектами. СПб: ООО «Техномедиа». Изд-во «Элмор». – 2007. – 244 с.

АЛГОРИТМ ИДЕНТИФИКАЦИИ ПАРАМЕТРОВ СИСТЕМЫ ДВИГАТЕЛЬ – ДВУХМАССОВЫЙ МЕХАНИЗМ

С.В. Арановский

Научный руководитель – д.т.н., профессор А.А. Бобцов

Статья посвящена проблеме идентификации параметров системы двигатель – двухмассовый механизм при известной структуре объекта и неизвестных параметрах передаточной функции. Идентификация проводится по результатам эксперимента на готовой к работе установке. Предложенный подход использует измерения только выходной переменной, не требует информации о медианных значениях параметров и не накладывает ограничений на дрейф параметров и позволяет получить высокую скорость оценивания параметров. В статье приводится пример, иллюстрирующий основные шаги предложенного алгоритма. Так же приведены результаты математического моделирования, иллюстрирующие работоспособность рассматриваемого подхода.

Ключевые слова: двигатель, двухмассовый механизм, идентификация, адаптивный идентификатор, метод наименьших квадратов

Введение

В настоящее время для решения широкого круга задач используются оптико-электронные комплексы, телескопы которых наводятся и длительное время удерживаются на объектах наблюдения с суммарной погрешностью в единицы угловых секунд. Основные функциональные характеристики таких комплексов в значительной степени определяются реальной точностью слежения за наблюдаемыми объектами. В решении подобных задач важную роль играет опорно-поворотное устройство (ОПУ), оси которого представляются в процессе моделирования и проектирования электроприводов, как правило, двухмассовыми механизмами.

При разработке подобных комплексов важной задачей является полная автоматизация процессов настройки электроприводов с требуемой точностью. Причем, эти настройки производятся как во время сборки и монтажа телескопа на испытательном стенде, так и на полигоне, в условиях случайно изменившихся вследствие транспортировки и новой сборки ОПУ параметров системы в целом. К таким параметрам относятся: люфты и зазоры, смещение центров масс нагрузок ОПУ, моменты инерции навесного оборудования, электрические параметры двигателя и др. Решение этой задачи возможно с использованием процедуры идентификации параметров подсистемы «двигатель-ось ОПУ». Суть процедуры идентификации в рамках поставленной нами задачи сводится к получению реальных, определенных в результате эксперимента на готовой к работе установке, параметров электрического двигателя, его нагрузки с учетом упругих связей, например, частот механических резонансов, моментов инерции. В дальнейшем результаты процедуры идентификации используются для автоматизации расчета параметров регуляторов. Кроме того, для придания замкнутой системе свойства адаптивности к изменениям параметров объекта следует выбрать алгоритм, позволяющий проводить идентификацию в реальном времени и на основании результатов вносить коррективы в работу регулятора.

Существуют различные подходы к решению задачи параметрической идентификации [1–3], в том числе метод локальной адаптации, метод скоростного градиента и его вариации, а так же ряд других. Несмотря на простоту реализации метода локальной адаптации, его недостатком является обеспечение сходимости только при малом отклонении идентифицируемых параметров от истинных значений, что требует повышенного внимания к начальным условиям. Так же метод локальной адаптации функционирует только в условиях квазистационарности и сильно зависит от режимов работы системы. Градиентные методы являются более развитыми, и как

следствие, более распространенными, однако они так же обладают рядом недостатков: сложность разработки, зависимость процесса идентификации и времени схождения от выбора функции качества. Кроме того, использование градиентных методов зачастую требует измерения всех переменных состояния объекта или же накладывает ограничения на объект.

Кроме указанных методов существует широкий класс подходов, основанных на предположении об известности медианных значений идентифицируемых параметров и ограниченности их дрейфа, причем данные подходы варьируются в зависимости от «широты» области дрейфа.

Предложенный в данной работе подход является развитием идей, представленных в [4, 5] и их приложением к вопросу идентификации параметров двухмассового ЭМО в реальном времени. Предложенный подход обладает следующими преимуществами:

- используются измерения только выходной переменной;
- высокое быстродействие;
- не требуется знать медианные значения параметров;
- нет ограничений на область дрейфа параметров и начальные условия за исключением их конечности;
- простота реализации.

Постановка задачи

В данной работе в качестве рассматриваемого объекта выступает система «двигатель – двухмассовый механизм». Обобщенная структурная схема такой системы приведена на рис. 1 [6].

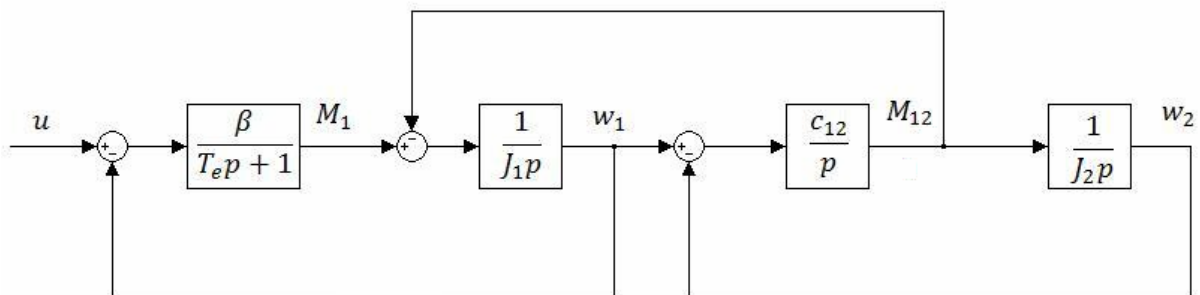


Рис. 1. Обобщенная структурная схема

Здесь w_1 – угловая скорость первой массы оси, w_2 – угловая скорость второй массы оси, u – обобщенное управляющее воздействие, имеющее размерность скорости холостого хода, M_1 – электромагнитный момент двигателя, M_{12} – момент, обусловленный нежесткостью конструкции оси, c_{12} – коэффициент жесткости, J_1 , J_2 – моменты инерции соответственно первой и второй масс, β – жесткость механической характеристики двигателя, T_e – электрическая постоянная времени двигателя.

Данное структурное описание легко может быть преобразовано к передаточной функции вида, где выходом является измеряемая угловая скорость первой массы:

$$y(t) = w_1(t) = \frac{b_2 p^2 + b_0}{p^4 + a_3 p^3 + a_2 p^2 + a_1 p + a_0} u(t), \quad (1)$$

где

$$\begin{aligned}
p &= \frac{d}{dt}, \\
b_2 &= \beta / (J_1 T_e), \\
b_0 &= c_{12} \beta / (J_1 J_2 T_e), \\
a_3 &= 1 / T_e, \\
a_2 &= (T_e (J_1 + J_2) c_{12} + J_2 \beta) / (J_1 J_2 T_e), \\
a_1 &= c_{12} / (J_1 T_e), \\
a_0 &= c_{12} \beta / (J_1 J_2 T_e).
\end{aligned}$$

Сформулируем задачу идентификации параметров объекта как задачу синтеза алгоритма идентификации, обеспечивающего для любых параметров $b_2, b_0, a_3, a_2, a_1, a_0$ формирование таких оценок этих параметров, что

$$\begin{cases} \lim_{t \rightarrow \infty} |b_i - \hat{b}_i| = 0, & i = 0, 2, \\ \lim_{t \rightarrow \infty} |a_j - \hat{a}_j| = 0, & j = \overline{0, 3}. \end{cases} \quad (2)$$

Основной результат

Перепишем выражение (1) следующим образом:

$$a(p)y = b(p)u, \quad (3)$$

где $a(p) = p^4 + a_3 p^3 + a_2 p^2 + a_1 p + a_0$, $b(p) = b_2 p^2 + b_0$.

Введем в рассмотрение гурвицев полином четвертой степени с единичным коэффициентом при старшей степени:

$$\phi(p) = (p + \alpha)^4 = p^4 + \phi_3 p^3 + \phi_2 p^2 + \phi_1 p + \phi_0, \quad (4)$$

где $\alpha > 0$.

Так же сформируем следующий полином:

$$\begin{aligned}
a_1(p) &= \phi(p) - a(p) = \\
&= (\phi_3 - a_3)p^3 + (\phi_2 - a_2)p^2 + (\phi_1 - a_1)p + (\phi_0 - a_0).
\end{aligned} \quad (5)$$

Подставив (5) в (3) можно записать:

$$\begin{aligned}
\phi(p)y &= a_1(p)y + b(p)u, \\
y &= \frac{a_1(p)}{\phi(p)}y + \frac{b(p)}{\phi(p)}u.
\end{aligned} \quad (6)$$

Введем дополнительные переменные, представляющие собой пропущенные через фильтр $\frac{1}{\phi(p)}$ сигналы y и u :

$$\xi_1 = \frac{1}{\phi(p)}y, \quad \xi_2 = \frac{1}{\phi(p)}u. \quad (7)$$

Подставляя (5) и (7) в (6) получим:

$$\begin{aligned}
y &= a_1(p)\xi_1 + b(p)\xi_2 = \\
&= (\phi_3 - a_3)\xi_1^{\ddot{\cdot}} + (\phi_2 - a_2)\xi_1^{\dot{\cdot}} + (\phi_1 - a_1)\dot{\xi}_1 + \\
&+ (\phi_0 - a_0)\xi_1 + b_2\dot{\xi}_2 + b_0\xi_2 = \psi^T \theta,
\end{aligned} \quad (8)$$

где

$$\begin{cases} \psi = \text{col}\{\xi_1^{(3)}, \xi_1^{(2)}, \xi_1^{(1)}, \xi_1, \xi_2^{(2)}, \xi_2\} \\ \theta = \text{col}\{(\phi_3 - a_3), (\phi_2 - a_2), (\phi_1 - a_1), (\phi_0 - a_0), b_2, b_0\}. \end{cases} \quad (9)$$

В выражении (9) в векторе-столбце ψ собраны доступные измерению переменные, а в векторе-столбце θ – неизвестные параметры.

Замечание. Нетрудно видеть, что для идентификации параметров объекта достаточно решить задачу идентификации параметров вектора θ .

Для идентификации неизвестных параметров в модели (8) могут быть использованы различные алгоритмы. Например, в работах [5, 6] был предложен алгоритм вида

$$\begin{aligned} \hat{\theta}(t) &= K\psi(t)\psi^T(t)(\theta - \hat{\theta}(t)) = \\ &= K\psi(t)y(t) - K\psi(t)\psi^T(t)\hat{\theta}(t), \end{aligned} \quad (10)$$

где $K = K^T > 0$ – диагональная матрица постоянных коэффициентов. Для данного алгоритма была доказана его применимость для решения рассматриваемой задачи.

Так же для идентификации параметров в модели (8) может быть использован подход, известный как метод наименьших квадратов. Этот метод отличается существенно более высокой вычислительной сложностью, однако он обладает высокой точностью и робастностью по отношению к возмущениям в измерениях. Рассмотрим этот метод подробнее. Предположим, что имеется N отсчетов. Тогда для выражения (8) можно записать:

$$[y_1 \ y_2 \ \dots \ y_N]^T = [\psi^T_1 \ \psi^T_2 \ \dots \ \psi^T_N]^T \theta$$

или

$$Y = \Psi^T \theta. \quad (11)$$

Тогда получаем:

$$\hat{\theta} = (\Psi\Psi^T)^{-1}\Psi Y. \quad (12)$$

С ростом числа отсчетов N выражение (12) будет обеспечивать сходимость параметров к их истинным значениям в том случае, если матрица $\Psi\Psi^T$ не вырождена. Это условие будет выполнено, если входной управляющий сигнал является «частотно богатым» (см. подробнее в [7]).

Для проведения идентификации в реальном времени может использоваться итеративная версия алгоритма (12), позволяющая обновлять оценку параметров при получении каждого нового отсчета и не требующая хранения значений всех отсчетов.

Пример

Для иллюстрации работоспособности алгоритма проведем моделирование. В качестве идентифицируемого объекта выберем систему двигатель – двухмассовый механизм со следующими параметрами:

$$c_{12} = 5.46 \cdot 10^8, J_1 = 415, J_2 = 83685, \beta = 16 \cdot 10^4, T_e = 16 \cdot 10^{-4}.$$

Данный объект можно представить в виде (1), вычислив соответствующие параметры:

$$a_3 = 625, a_2 = 1.563 \cdot 10^6, a_1 = 8.223 \cdot 10^8, a_0 = 1.572 \cdot 10^9, b_2 = 2.409 \cdot 10^5, b_0 = 1.572 \cdot 10^9.$$

Введем гурвицев полином 4-ой степени $\phi(p) = (p+1)^4 = p^4 + 4p^3 + 6p^2 + 4p + 1$. Тогда в соответствии с (5):

$$a_1(p) = (4 - a_3)p^3 + (6 - a_2)p^2 + (4 - a_1)p + (1 - a_0).$$

Теперь сформируем вектор неизвестных и подлежащих идентификации параметров:

$$\theta = \text{col}\{(4 - a_3), (6 - a_2), (4 - a_1), (1 - a_0), b_2, b_0\}.$$

Далее в соответствии с выражением (7) запишем

$$\begin{cases} \dot{\xi}_1 = \xi_2 \\ \dot{\xi}_2 = \xi_3 \\ \dot{\xi}_3 = \xi_4 \\ \dot{\xi}_4 = -\xi_1 - 4\xi_2 - 6\xi_3 - 4\xi_4 + y \end{cases} \quad \begin{cases} \dot{\zeta}_1 = \zeta_2 \\ \dot{\zeta}_2 = \zeta_3 \\ \dot{\zeta}_3 = \zeta_4 \\ \dot{\zeta}_4 = -\zeta_1 - 4\zeta_2 - 6\zeta_3 - 4\zeta_4 + u \end{cases}$$

и сформируем вектор измеряемых параметров

$$\psi = \text{col}\{\xi^{(3)}, \xi^{(2)}, \xi^{(1)}, \xi, \zeta^{(2)}, \zeta\}.$$

Для идентификации неизвестных параметров будем использовать метод наименьших квадратов: проведем моделирование объекта, представленного передаточной функцией вида (1) и параметрами, представленными выше. При моделировании будем сохранять отсчеты, соответствующие значениям ψ и y с шагом в 0.001 секунды. В качестве задающего сигнала выберем меандр с амплитудой 1 и периодом 1 секунда, время моделирования выберем равным 1 секунде. Таким образом, по окончании моделирования мы имеем 1000 отсчетов. Сформировав в соответствии с (11) матрицу Ψ^T и вектор Y , получим оценку $\hat{\theta}$ согласно выражению (12). Проведенные моделирование и вычисления показали, что все оценки параметров сходятся к своим истинным значениям, причем скорость сходимости достаточно велика. В связи с ограниченным форматом статьи приведем только два графика из шести. На рис. 2 и 3 приведены графики для параметров a_0 и b_0 соответственно, истинные значения параметров отмечены пунктиром. В целом, по результатам эксперимента можно считать, что оценка параметров достигает истинного (расчетного) значения за время, равное 50 отсчетам, т.е. в данном примере – 0.05 секунды.

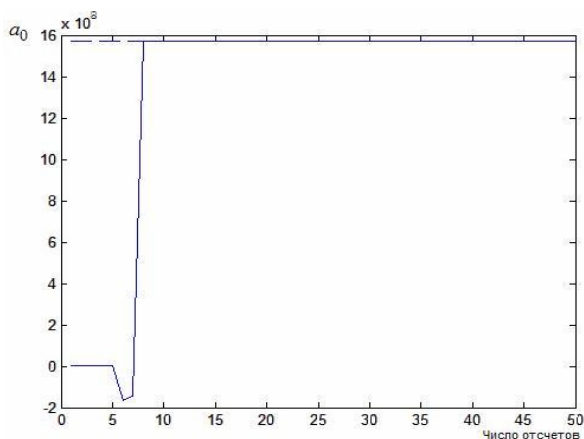


Рис. 2. Оценка параметра a_0

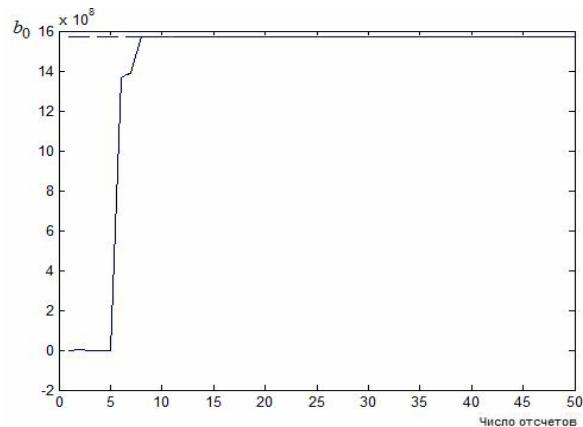


Рис. 3. Оценка параметра b_0

Заключение

В работе был предложен подход, позволяющий проводить идентификацию параметров системы «двигатель – двухмассовый механизм» при известной структуре системы по результатам эксперимента на готовой к работе установке. Результатом идентификации являются параметры передаточной функции объекта. Предложенный

подход обладает следующими преимуществами: в работе используются измерения только выходной переменной объекта, но не ее производных, высокая скорость оценивания параметров, отсутствует ограничение на область дрейфа параметров, не требуется знать медианные значения параметров. Работоспособность предложенного алгоритма подтверждается как аналитическими выкладками, так и результатами моделирования.

Литература

1. Борцов Ю.А., Поляхов Н.Д., Путов В.В. Электромеханические системы с адаптивным и модальным управлением. – Л.: Энергоатомиздат. Ленингр. отд-ние. – 1984 – 216 с.
2. Борцов Ю.А., Соколовский Г.Г. Автоматизированный электропривод с упругими связями. – Л.: Энергоатомиздат. – 1992. – 288 с.
3. Фрадков А.Л. Адаптивное управление в сложных системах. – М.: Наука. – 1990.
4. Арановский С.В., Бобцов А.А., Кремлев А.С., Лукьянова Г.В. Робастный алгоритм идентификации частоты синусоидального сигнала. //Известия РАН. Теория и системы управления. – 2007. – №3.
5. Арановский С.В., Бобцов А.А., Кремлев А.С., Лукьянова Г.В., Николаев Н.А. Идентификация частоты смещенного синусоидального сигнала. //АиТ. – 2008. – №9.
6. Ключев В.И. Теория электропривода: Учебник для вузов. – М: Энергоатомиздат. – 2001. – 704 с.
7. Мирошник И.В., Никифоров В.О., Фрадков А.Л. Нелинейное и адаптивное управление сложными динамическими системами. – СПб.: Наука. – 2000. – 549 с.

ПРИМЕНЕНИЕ МЕТОДА СТОХАСТИЧЕСКОЙ АППРОКСИМАЦИИ С ПРОБНЫМ ОДНОВРЕМЕННЫМ ВОЗМУЩЕНИЕМ НА ВХОДЕ В ЗАДАЧАХ СТЕРЕОЗРЕНИЯ И ОПРЕДЕЛЕНИЯ ДВИЖЕНИЯ

А.Т. Вахитов, Д.В. Павленко

(Санкт-Петербургский государственный университет)

Научный руководитель – д.ф.-м.н., профессор О.Н. Граничин

(Санкт-Петербургский государственный университет)

Задачи стереозрения и определения движения являются близкими. Обычно задача такого типа формулируется в виде поиска вектора расхождения для данной пары изображений, на одном из которых выделен искомый объект. Популярным алгоритмом решения такой задачи является алгоритм Лукаса-Канаде, основывающийся на оптимизации некоторого функционала качества. В этой работе обосновывается использование рандомизированных алгоритмов стохастической аппроксимации (РАСА) для оптимизации функционала качества.

Ключевые слова: стереозрение, стохастическая аппроксимация, определение движения, пробное возмущение

Введение

Задачи стереозрения и определения движения являются близкими. Обе задачи являются актуальными для робототехники, сенсорных систем, систем видеонаблюдения и др. Рынок технических охранных систем в РФ составляет более \$1 млрд. В мире камеры наблюдения имеют первостепенную важность при разработке промышленных роботов, используемых сегодня на производстве автомобилей, бытовой техники и многих других товаров. Предлагаемый в работе метод может сократить требуемые ресурсы для работы систем обработки видеопотока, тем самым увеличивая эффективность использования этих систем.

Обычно задача такого типа формулируется в виде поиска вектора расхождения для данной пары изображений и выбранного объекта, локализованного на одном из изображений. Задача может ставиться либо в виде поиска карты расхождений для данной пары изображений, либо в виде определения расстояния до выделенного на изображении объекта. В этом исследовании используется второй вариант постановки задачи: известны два изображения (заданные в виде матриц интенсивностей $L(i,j)$, $R(i,j)$), известно, где на одном из изображений находится объект (задано подмножество индексов $\{(i',j')\}=I$, такое что $\{L(i',j')\}$ является изображением объекта), требуется найти объект на втором изображении и определить расстояние до объекта и его линейный размер.

Основной алгоритм

В 1981 году для решения описанной задачи Б. Лукасом и Т. Канаде был предложен метод, основанный на псевдоградиентной оптимизации функционала качества. Рассмотрим разность $F(i,j,h_1,h_2) = (P(i+h_1, j+h_2) - Q(i,j))^2$, являющуюся расхождением интенсивности пикселей изображений P и Q с расхождением h_1, h_2 . Авторы метода предлагают искать вектор расхождения $h = (h_1, h_2)$, обеспечивающий минимальное агрегированное значение расхождения по всем пикселям изображения объекта: $f(h_1, h_2) = \sum_{(i,j) \in I} F(i,j, h_1, h_2) \rightarrow \min$ (Lucas & Kanade 1981). По расхождению можно определить расстояние до объекта, используя формулу $d = \text{baseline} * f / |h|$, где baseline – размер стереобазы, f – фокусное расстояние камеры, $|h|$ – евклидова норма вектора.

После появления метода Лукаса-Канады были предложены некоторые его модификации, полученные использованием других методов оптимизации в той же задаче (Baker & Matthews 2004).

В своей статье Лукас и Канада используют собственную оценку градиента функции E , полученную с использованием знаний о виде этой функции. Для построения следующей оценки по методу такого типа требуется $O(q^2N+q^3)$ вычислений, где $N=|I|$ – число пикселей в изображении объекта, q – размерность аргумента E , в нашем случае $q=2$ (Baker & Matthews 2004). На каждом шаге алгоритм вычисляет kqN раз значение функции F , где k – число измерений функции, необходимое для построения аппроксимации градиента, для конечноразностной аппроксимации первого порядка $k=2$. Если рассматривать более общий вариант приближения градиента, типа использующегося в методе Кифера-Вольфовица, то число вычислений функции F будет таким же.

В последнее время интенсивно развивается область рандомизированных алгоритмов стохастической аппроксимации (РАСА) с пробным одновременным возмущением на входе (Граничин 1989, Spall 1992), которые позволяют сократить количество вычислений на итерации по сравнению с упомянутым методом Кифера-Вольфовица. Особенности РАСА позволяют резко уменьшить число необходимых измерений F на итерации алгоритма в рассматриваемой в этой работе задаче. Перечислим основные преимущества РАСА:

Во-первых, РАСА основаны на приближении градиента по случайно выбранному направлению. За счет этого число вычислений функции F на итерации может быть сокращено в q раз.

Во-вторых, f может рассматриваться как математическое ожидание F :

$$\frac{1}{N}f = \sum_{(i,j) \in I} \frac{1}{N} F(i,j, h_1, h_2) = E_{(i,j)},$$

где (i,j) распределены равномерно на множестве I . Сходимость метода РАСА обоснована для случая оптимизации функционала среднего риска типа f при наличии наблюдений эмпирического функционала, такого как F . Таким образом, на одной итерации алгоритма РАСА требуется всего k измерений F .

Алгоритм типа РАСА, выбранный нами, выглядит следующим образом (Граничин 2003):

$$\hat{\theta}^{(n)} = \hat{\theta}^{(n-1)} - \frac{\alpha_n}{2\beta_n} \Delta_n (F(w_{2n-1}, \hat{\theta}^{(n-1)} + \beta_n \Delta_n) - F(w_{2n}, \hat{\theta}^{(n-1)} - \beta_n \Delta_n))$$

Здесь α_n, β_n – последовательности, определяющие скорость сходимости и точности конечноразностной аппроксимации градиента, Δ_n – бернуллиевский вектор, играющий роль пробного возмущения, w_n – неопределенность, связанная со случайным выбором пикселя (соответствует паре i,j), θ_n^* – искомая оценка минимума. Подход, основанный на применении РАСА, является предметом этого исследования. Для проведения экспериментов было выбрано стандартная тестовая пара изображений из набора, описанного в (Scharstein & Szeliski 2001). Были определены значения параметров алгоритма $\alpha_n=100/n^{0.5}$, $\beta_n=2$. Размер объекта в пикселях $N=7360$. Эксперимент производился на компьютере под управлением операционной системы Linux с использованием встроенного генератора случайных чисел. Результаты сравнения алгоритмов приведены в табл. 1.

Таблица 1. Сравнение производительности алгоритмов

	Алг. Лукас-Канаде	РАСА (в среднем по 10 запускам)
Число измерений F	73600	112

Проведенные эксперименты показывают значительно сокращение времени, необходимого на поиск h , по сравнению с применением изначального метода Б. Лукаса и Т. Канаде.

Заключение

Проект находится сейчас на стадии прототипирования и тестирования на готовых наборах, подготовленных зарубежными исследователями и доступных в Интернет. Следующим этапом станет тестирование алгоритма для обработки видеопотока с веб-камеры, затем – с двух камер. Результатом проекта будет программно-аппаратная система из 2-х камер и персонального компьютера с установленным программным обеспечением для получения данных о размерах движущихся объектов и дальности до них в режиме реального времени.

Литература

1. Lukas B., Kanade T., An Iterative Registration Technique with an Application to Stereo Vision / B. Lukas, T. Kanade // In Proc. Imaging Understanding Workshop. – 1981. – С. 121–130.
2. Baker S., Matthews I. Lukas-Kanade 20 years on: A unifying framework: Part 1. / S. Baker, I. Matthews // Technical Report CMU-RI-TR-02-16 Carnegie Mellon University Robotics Institute. – 2002.
3. Граничин О.Н. Об одной стохастической рекуррентной процедуре при зависимых помехах в наблюдении, использующей на входе пробные возмущения / О.Н. Граничин // Вестник Ленингр. ун-та. – 1989. – Сер.1. – Вып. 1(4). – С. 19–21.
4. Spall J.C. Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation / J.C. Spall // IEEE Transactions on Automatic Control. – 1992. – Vol. 37. – С. 332–341.
5. Scharstein D., Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. / D. Scharstein, R. Szeliski // International Journal of Computer Vision. – 2002. – 47(1/2/3). – С. 7–42.

КОМПЕНСАЦИЯ НЕИЗВЕСТНОГО СМЕЩЕННОГО ГАРМОНИЧЕСКОГО ВОЗМУЩЕНИЯ ДЛЯ НЕЛИНЕЙНОГО ОБЪЕКТА С ЗАПАЗДЫВАНИЕМ ПО УПРАВЛЕНИЮ

С.А. Колюбин, А.А. Пыркин

Научный руководитель – д.т.н., профессор А.А. Бобцов

Статья посвящена развитию методов компенсации гармонических возмущений по измерениям выходной переменной объекта. В работе предлагается новый алгоритм компенсации параметрически не определенных смещенных гармонических возмущений. В отличие от известных результатов, в данной работе рассматривается случай компенсации возмущений, когда относительная степень объекта может быть любой, измеряются только выходная переменная и сигнал управления, и канал управления характеризуется запаздыванием.

Ключевые слова: идентификация частоты, компенсация возмущения, управление по выходу

1. Введение

В данной работе предлагается новый алгоритм компенсации параметрически не определенного смещенного гармонического возмущения, действующего на нелинейный объект управления. Достаточно большое число работ посвящено управлению в условиях действия неизвестного возмущающего воздействия $\delta(t) = \sigma_0 + \sigma \sin(\omega t + \varphi)$ по измерениям только выходной переменной и сигнала управления (см. [1–7]).

В данной статье развиваются подходы, опубликованные в [1–7]. В [1, 2, 5–7] предлагаются методы компенсации возмущения, действующего на линейный объект управления в отличие от [3, 4]. В [3, 4] рассматриваются минимально-фазовые объекты управления, а в [5, 6] объекты с единичной относительной степенью. В [6] решена задача компенсации неизвестного возмущения в случае, когда канал управления характеризуется запаздыванием.

В данной статье предлагается новый адаптивный алгоритм компенсации параметрически не определенного смещенного гармонического возмущения, действующего на нелинейный объект управления, относительная степень которого может быть любой, а канал управления характеризуется запаздыванием в отличие от [1–7].

2. Постановка задачи

Рассмотрим нелинейный строго реализуемый объект вида:

$$\dot{x}(t) = Ax(t) + Bu(t - h) + Gf(y(t)) + E\delta(t), y(t) = C^T x(t), \quad (1)$$

где вектор переменных состояния $x(t) \in R^n$ не измеряется, $u(t - h)$ – скалярный сигнал управления, $h > 0$ – известное постоянное запаздывание, $y(t)$ – скалярная регулируемая переменная, $f(y(t))$ – известная нелинейная функция.

Входное возмущение $\delta(t)$ представлено в виде функции $\delta(t) = \delta_1(t) + \delta_2(t)$, где $\delta_2(t) = \sigma \sin(\omega t + \varphi)$ – неизвестное смещение, $\delta_1(t) = \sigma_0$ – синусоида с неизвестными амплитудой σ , частотой ω и фазовым сдвигом φ .

Наряду с моделью вход-состояние-выход (1) рассмотрим модель вход-выход вида

$$a(p)y(t) = b(p)u(t - h) + g(p)f(y(t)) + e(p)\delta(t), \quad (2)$$

где $p = d / dt$ – оператор дифференцирования, $a(p) = p^n + a_{n-1}p^{n-1} + \dots + a_1p + a_0$,

$b(p) = b_m p^m + \dots + b_1 p + b_0$, $g(p) = g_l p^l + \dots + g_1 p + g_0$ и $e(p) = e_r p^r + \dots + e_1 p + e_0$, – соответствующие полиномы, полученные в результате перехода от модели вход-состояние-выход к модели вход-выход

$$\frac{b(p)}{a(p)} = C(pI - A)^{-1}B, \frac{g(p)}{a(p)} = C(pI - A)^{-1}G, \frac{e(p)}{a(p)} = C(pI - A)^{-1}E.$$

Представим уравнение (2) в виде

$$y(t) = \bar{u}(t) + \bar{f}(t) + \bar{\delta}_1(t) + \bar{\delta}_2(t), \quad (3)$$

где $\bar{u}(t) = \frac{b(p)}{a(p)} u(t-h)$, $\bar{f}(t) = \frac{g(p)}{a(p)} f(y(t))$, $\bar{\delta}_1(t) = \frac{e(p)}{a(p)} \delta_1(t)$ и $\bar{\delta}_2(t) = \frac{e(p)}{a(p)} \delta_2(t)$.

Рассмотрим следующие допущения относительно системы (1), (2).

Допущение 1. Будем полагать, что измеряются только сигналы $y(t)$ и $u(t)$.

Допущение 2. Коэффициенты матриц A, B, C и G известны.

Допущение 3. Тройка A, B, C полностью управляема и наблюдаема.

Допущение 4. Полином $a(p)$ гурвицев.

Допущение 5. Полином $b(p)$ не имеет корней $\pm j\omega$.

Допущение 6. Для объекта управления справедливо предельное равенство $\lim_{\bar{u}(t) \rightarrow \bar{\delta}(t)} y(t) = 0$, где $\bar{\delta}(t) = \bar{\delta}_1(t) + \bar{\delta}_2(t)$.

Требуется найти закон управления, обеспечивающий целевое условие

$$\lim_{t \rightarrow \infty} y(t) = 0. \quad (4)$$

3. Синтез закона управления

В силу допущения 6 цель (4) будет достигнута при таком законе управления, при котором справедливо соотношение:

$$\bar{u}(t) = -\bar{\delta}_1(t) - \bar{\delta}_2(t).$$

С учетом допущений 4 и 5 имеем

$$\bar{\delta}_1(t) = L_{\delta_1} \cdot \delta_1(t) + \varepsilon_1 \text{ и } \bar{\delta}_2(t) = L_{\delta_2} \cdot \delta_2(t + \frac{\varphi_{\delta_2}}{\omega}) + \varepsilon_2, \quad (5)$$

где $L_{\delta_1} = \frac{e_0}{a_0}$ – положительный коэффициент передачи, $L_{\delta_2} = \left| \frac{e(j\omega)}{a(j\omega)} \right|$ и $\varphi_{\delta_2} = \arg \frac{e(j\omega)}{a(j\omega)}$ – соответственно положительный коэффициент передачи и фазовый сдвиг при воздействии гармонического воздействия с частотой ω на объект управления, ε_1 и ε_2 – экспоненциально затухающие составляющие [7].

Введем в рассмотрение функции:

$$L_1 = \frac{b_0}{a_0}, L_2 = \left| \frac{b(j\omega)}{a(j\omega)} \right| \text{ и } \varphi_2 = \arg \frac{b(j\omega)}{a(j\omega)}.$$

Пренебрегая экспоненциальными составляющими ε_1 и ε_2 , выберем $u(t)$ следующим образом

$$u(t) = -\frac{1}{L_1} \hat{\delta}_1(t+h) - \frac{1}{L_2} \hat{\delta}_2(t+h - \frac{\hat{\varphi}_2}{\hat{\omega}}), \quad (6)$$

где функции $\hat{\delta}_1(t+h)$ и $\hat{\delta}_2(t+h - \frac{\hat{\varphi}_2}{\hat{\omega}})$ являются оценками $\bar{\delta}_1(t+h)$ и $\bar{\delta}_2(t+h - \frac{\varphi_2}{\omega})$

соответственно, $\hat{L}_2 = \left| \frac{b(j\hat{\omega})}{a(j\hat{\omega})} \right|$ – оценка коэффициента L_2 , $\hat{\varphi}_2 = \arg \frac{b(j\hat{\omega})}{a(j\hat{\omega})}$ – оценка фазового сдвига φ_2 , $\hat{\omega}$ – оценка частоты ω . Если

$$\bar{\omega} = \omega, \quad \widehat{\delta}_1(t+h) = \bar{\delta}_1(t+h), \quad \widehat{\delta}_2(t+h - \frac{\widehat{\varphi}_2}{\bar{\omega}}) = \bar{\delta}_2(t+h - \frac{\varphi_2}{\omega}), \quad \widehat{L}_2 = L_2, \quad \widehat{\varphi}_2 = \varphi_2, \quad (7)$$

$$\text{то } u(t) = -\frac{1}{L_1} \widehat{\delta}_1(t+h) - \frac{1}{L_2} \widehat{\delta}_2(t+h - \frac{\widehat{\varphi}_2}{\bar{\omega}}) = -\frac{1}{L_1} \bar{\delta}_1(t+h) - \frac{1}{L_2} \bar{\delta}_2(t+h - \frac{\varphi_2}{\omega}).$$

Так как функция $\bar{\delta}_1(t)$ является постоянной, а $\bar{\delta}_2(t)$ – гармонической с частотой ω , то справедлива цепочка соотношений:

$$\bar{u}(t) = \frac{b(p)}{a(p)} u(t-h) = -\frac{b(p)}{a(p)} \left(\frac{1}{L_1} \bar{\delta}_1(t) \right) - \frac{b(p)}{a(p)} \left(\frac{1}{L_2} \bar{\delta}_2(t - \frac{\varphi_2}{\omega}) \right) = -L_1 \left(\frac{1}{L_1} \bar{\delta}_1(t) \right) - L_2 \left(\frac{1}{L_2} \bar{\delta}_2(t - \frac{\varphi_2}{\omega} + \frac{\varphi_2}{\omega}) \right) = -\bar{\delta}_1(t) - \bar{\delta}_2(t).$$

Таким образом, закон управления вида (6) обеспечивает полную компенсацию возмущения и, с учетом допущения 6, достижение цели управления (4).

Далее требуется построить устройство оценки, обеспечивающее выполнение соотношений (7).

4. Идентификация возмущения

В этом разделе будем решать задачу идентификации параметров неизвестного возмущения. Временно предположим, что сигнал $\bar{\delta}(t) = \bar{\sigma}_0 + \bar{\sigma} \sin(\omega t + \bar{\varphi})$ измеряется, где $\bar{\sigma}_0$ – смещение, $\bar{\sigma}$ – амплитуда и $\bar{\varphi}$ – фазовый сдвиг. Известно, что для генерирования сигнала $\bar{\delta}(t)$ можно использовать дифференциальное уравнение вида

$$(p^3 - \theta p) \bar{\delta}(t) = 0, \quad (8)$$

где $\theta = -\omega^2$ – постоянный параметр.

Перепишем уравнение (8):

$$p^3 \bar{\delta}(t) = \bar{\theta} p \bar{\delta}(t), \quad (9)$$

Введем в рассмотрение вспомогательный линейный фильтр:

$$\xi(t) = \frac{\alpha^2}{(p + \alpha)^2} \bar{\delta}(t). \quad (10)$$

Нетрудно показать, что:

$$\alpha^2 \dot{\bar{\delta}}(t) - 2\alpha \ddot{\xi}(t) - \alpha^2 \dot{\xi}(t) = \bar{\theta} \dot{\xi}(t) + \varepsilon_\delta(t). \quad (11)$$

Предположим, что функция $\dot{\bar{\delta}}(t)$ доступна измерению. Тогда, пренебрегая экспоненциально затухающим слагаемым $\varepsilon_\delta(t)$, запишем идеальный алгоритм идентификации следующим образом

$$\dot{\hat{\theta}}(t) = k \dot{\xi}^2(t) (\theta - \hat{\theta}(t)) = k \dot{\xi}(t) z(t) - k \dot{\xi}^2(t) \hat{\theta}(t), \quad (12)$$

где функция $z(t) = \alpha^2 \dot{\bar{\delta}}(t) - 2\alpha \ddot{\xi}(t) - \alpha^2 \dot{\xi}(t)$ и число $k > 0$.

Утверждение. Пусть оценка параметра θ настраивается следующим образом

$$\dot{\hat{\theta}}(t) = k \dot{\xi}^2(t) (\theta - \hat{\theta}(t)), \quad (13)$$

тогда $\lim_{t \rightarrow \infty} |\theta - \hat{\theta}(t)| = 0$.

Замечание. При увеличении коэффициента k функция $\hat{\theta}(t)$ будет быстрее сходиться к параметру θ .

Однако алгоритм (12)

технически нереализуем, так как содержит неизмеримый сигнал $z(t) = \alpha^2 \dot{\bar{\delta}}(t) - 2\alpha \ddot{\xi}(t) - \alpha^2 \dot{\xi}(t)$, и, следовательно, (12)

имеет вид

$$\begin{aligned} \dot{\hat{\theta}}(t) &= k \dot{\xi}^2(t) (\theta - \hat{\theta}(t)) = k \dot{\xi}(t) z(t) - k \dot{\xi}^2(t) \hat{\theta}(t) = \\ &= k \dot{\xi}(t) (\alpha^2 \dot{\bar{\delta}}(t) - 2\alpha \ddot{\xi}(t) - \alpha^2 \dot{\xi}(t)) - k \dot{\xi}(t) \hat{\theta}(t) \end{aligned} \quad (14)$$

Рассмотрим переменную:

$$\chi(t) = \hat{\theta}(t) - k\alpha^2 \dot{\xi}(t) \bar{\delta}(t). \quad (15)$$

Дифференцируя (15), получим

$$\dot{\chi}(t) = \dot{\hat{\theta}}(t) - k\alpha^2 \ddot{\xi}(t) \bar{\delta}(t) - k\alpha^2 \dot{\xi}(t) \dot{\bar{\delta}}(t) = k\dot{\xi}(t)(-2\alpha\ddot{\xi}(t) - \alpha^2\dot{\xi}(t)) - k\dot{\xi}^2(t)\dot{\bar{\theta}}(t) - k\alpha^2\dot{\xi}(t)\dot{\bar{\delta}}(t) \quad (16)$$

Из (15) получим

$$\hat{\theta}(t) = \chi(t) + k\alpha^2 \dot{\xi}(t) \bar{\delta}(t) \quad (17)$$

На основе (3) для (10), (16) и (17) имеем

$$\xi(t) = \frac{\alpha^2}{(p + \alpha)^2} (y(t) - \bar{f}(t) - \bar{u}(t)). \quad (18)$$

$$\dot{\chi}(t) = k\dot{\xi}(t)(-2\alpha\ddot{\xi}(t) - \alpha^2\dot{\xi}(t)) - k\dot{\xi}^2(t)\dot{\hat{\theta}}(t) - k\alpha^2\dot{\xi}(t)(y(t) - \bar{f}(t) - \bar{u}(t)) \quad (19)$$

$$\hat{\theta}(t) = \chi(t) + k\alpha^2 \dot{\xi}(t)(y(t) - \bar{f}(t) - \bar{u}(t)), \quad (20)$$

и, следовательно, при таком формировании функции $\xi(t)$ алгоритм является технически реализуемым. (12)

Частоту возмущения найдем из (8)

$$\bar{\omega}(t) = \sqrt{|\hat{\theta}(t)|}. \quad (21)$$

Так как фильтр (10) линейный, то справедливо соотношение:

$$\xi(t) = \xi_1(t) + \xi_2(t), \quad (22)$$

где $\xi_1(t) = \frac{\alpha^2}{(p + \alpha)^2} \bar{\delta}_1(t)$ и $\xi_2(t) = \frac{\alpha^2}{(p + \alpha)^2} \bar{\delta}_2(t)$.

Аналогично (5) можем записать следующие соотношения:

$$\xi_1(t) = L_{\xi_1} \cdot \bar{\delta}_1(t) \text{ и } \xi_2(t) = L_{\xi_2} \cdot \bar{\delta}_2(t + \frac{\varphi_{\xi_2}}{\omega}), \quad (23)$$

где $L_{\xi_1} = \frac{\alpha^2}{\alpha^2} = 1$, $L_{\xi_2} = \left| \frac{\alpha^2}{(j\omega + \alpha)^2} \right|$, $\varphi_{\xi_2} = \arg \frac{\alpha^2}{(j\omega + \alpha)^2} < 0$.

На основе (23) получим выражения для оценки функций $\bar{\delta}_1(t)$ и $\bar{\delta}_2(t)$:

$$\hat{\delta}_1(t) = \frac{1}{L_{\xi_1}} \cdot \xi_1(t) = \xi_1(t) \text{ и } \hat{\delta}_2(t) = \frac{1}{L_{\xi_2}} \cdot \xi_2(t - \frac{\hat{\varphi}_{\xi_2}}{\hat{\omega}}), \quad (24)$$

где $\hat{L}_{\xi_2} = \left| \frac{\alpha^2}{(j\hat{\omega} + \alpha)^2} \right|$ – оценка коэффициента L_{ξ_2} , $\hat{\varphi}_{\xi_2} = \arg \frac{\alpha^2}{(j\hat{\omega} + \alpha)^2}$ – оценка фазового сдвига φ_{ξ_2} , а функции $\xi_1(t)$ и $\xi_2(t)$ являются постоянной и синусоидальной.

Дифференцируя (22) 2 раза, получим:

$$\ddot{\xi}(t) = \ddot{\xi}_1(t) + \ddot{\xi}_2(t) = \theta \xi_2(t), L_{\xi_2}, \text{ откуда имеем } \xi_2(t) = \frac{\ddot{\xi}(t)}{\theta}.$$

Устройство оценки для функций $\xi_1(t)$ и $\xi_2(t)$ построим в виде:

$$\hat{\xi}_1(t) = \xi(t) - \hat{\xi}_2(t) \text{ и } \hat{\xi}_2(t) = \frac{\ddot{\xi}(t)}{\hat{\theta}(t)}. \quad (25)$$

5. Реализуемый закон управления

В силу того, что выполнено равенство $\lim_{t \rightarrow \infty} |\theta - \bar{\theta}(t)| = 0$, то справедливы соотношения (7), и как следствие закон управления (6) обеспечивает цель управления (4). С учетом (24) закон управления (6) примет вид:

$$u(t) = -\frac{1}{L_1} \widehat{\xi}_1(t+h) - \frac{1}{\widehat{L}_2} \cdot \frac{1}{\widehat{L}_{\xi_2}} \cdot \widehat{\xi}_2(t+h - \frac{\widehat{\varphi}_2}{\widehat{\omega}} - \frac{\widehat{\varphi}_{\xi_2}}{\widehat{\omega}}). \quad (26)$$

Введем в рассмотрение функции и φ_{u2} такие, что

$$L_{u2} = L_2 \cdot L_{\xi_2} = \left| \frac{b(j\omega)}{a(j\omega)} \right| \cdot \left| \frac{\alpha^2}{(j\omega + \alpha)^2} \right| \text{ и } \varphi_{u2} = \varphi_2 + \varphi_{\xi_2} = \arg \frac{b(j\omega)}{a(j\omega)} + \arg \frac{\alpha^2}{(j\omega + \alpha)^2}.$$

Легко показать, что функции L_{u2} и φ_{u2} могут быть вычислены по формулам:

$$L_{u2} = \left| \frac{b(j\omega)\alpha^2}{a(j\omega)(j\omega + \alpha)^2} \right| \text{ и } \varphi_{u2} = \arg \frac{b(j\omega)\alpha^2}{a(j\omega)(j\omega + \alpha)^2}.$$

Для формирования закона управления введем функции

$$\widehat{L}_{u2} = \left| \frac{b(j\widehat{\omega})\alpha^2}{a(j\widehat{\omega})(j\widehat{\omega} + \alpha)^2} \right| \text{ и } \widehat{\varphi}_{u2} = \arg \frac{b(j\widehat{\omega})\alpha^2}{a(j\widehat{\omega})(j\widehat{\omega} + \alpha)^2}. \quad (27)$$

Перепишем (26) в виде

$$u(t) = -\frac{1}{L_1} \widehat{\xi}_1(t+h) - \frac{1}{\widehat{L}_{u2}} \cdot \widehat{\xi}_2(t+h - \frac{\widehat{\varphi}_{u2}}{\widehat{\omega}}).$$

Далее необходимо построить устройство упреждающей оценки переменных $\xi_1(t+h)$ и $\xi_2(t+h - \frac{\varphi_{u2}}{\omega})$. Воспользуемся тем, что функция $\xi_1(t)$ – постоянная, а $\xi_2(t)$ – периодическая.

Пусть $h_0 = h - \frac{\varphi_{u2}}{\omega} - qT$, где $h_0 \in [0; T)$, $T = \frac{2\pi}{\omega}$ – период функции $\xi_2(t)$, q – целое число.

$$\text{Тогда } \xi_2(t+h - \frac{\varphi_{u2}}{\omega}) = \xi_2(t+h_0 + qT) = \xi_2(t+h_0).$$

В силу того, что функция $\xi_1(t)$ – постоянная, то справедливо соотношение:

$$\xi_1(t+h) = \xi_1(t).$$

Для получения упреждающей оценки для $\xi_2(t+h_0)$ введем в рассмотрение переменную $\bar{\xi}_2(t)$ и линейный оператор $W_u(p) = k_p + k_d p$:

$$\bar{\xi}_2(t) = -W_u(p)\xi_2(t) = -(k_p + k_d p)\xi_2(t). \quad (28)$$

Так как $\xi_2(t)$ имеет синусоидальную форму, то

$$\bar{\xi}_2(t) = -|W_u(j\omega)|\xi_2(t + \frac{\arg W_u(j\omega)}{\omega}) + \varepsilon_u, \text{ где } |W_u(j\omega)| \text{ – коэффициент передачи,}$$

$\arg W_u(j\omega)$ – фазовый сдвиг, ε_u – экспоненциально затухающая составляющая.

Пренебрегая составляющей ε_u , выберем коэффициенты k_p и k_d в (28) так, чтобы выполнилось соотношение $\bar{\xi}_2(t) = -\frac{1}{L_{u2}}\xi_2(t+h_0)$:

$$k_p = \frac{\cos(h \cdot \omega - \varphi_{u2})}{L_{u2}}, k_d = \frac{\sin(h \cdot \omega - \varphi_{u2})}{\omega L_{u2}} \quad (29)$$

Перепишем (29) в реализуемой форме:

$$k_p = \frac{\cos(h \cdot \widehat{\omega} - \widehat{\varphi}_{u2})}{\widehat{L}_{u2}}, k_d = \frac{\sin(h \cdot \widehat{\omega} - \widehat{\varphi}_{u2})}{\widehat{\omega} \widehat{L}_{u2}}, \text{ где } \widehat{\omega} \neq 0 \text{ и } \widehat{L}_{u2} \neq 0. \quad (30)$$

Таким образом, закон управления выглядит следующим образом:

$$u(t) = -\frac{1}{L_1} \hat{\xi}_1(t) - k_p \cdot \hat{\xi}_2(t) - k_d \cdot \dot{\hat{\xi}}_2(t). \quad (31)$$

Дифференцируя (22), получим $\dot{\hat{\xi}}(t) = \dot{\hat{\xi}}_1(t) + \dot{\hat{\xi}}_2(t) = \dot{\hat{\xi}}_2(t)$, откуда построим наблюдатель сигнала $\dot{\hat{\xi}}_2(t)$:

$$\dot{\hat{\xi}}_2(t) = \dot{\hat{\xi}}(t). \quad (32)$$

Так как выполнены соотношения (7), то закон управления (6) обеспечивает цель управления (4).

6. Пример

Для иллюстрации работоспособности алгоритма управления (18)–(21), (27), (30)–(32) рассмотрим задачу стабилизации нелинейного стационарного объекта управления, описываемого уравнением

$$(p^2 + 5p + 6)y(t) = (p + 3)u(t - h) + (p + 4)f(y) + (p + 1)\delta(t),$$

где $h = 2$, $f(y) = -y \sin(y)$, $\delta(t) = 1 + 2 \sin(3t + 4)$.

Из рис. 1, 2 видно, что предложенный алгоритм управления обеспечивает выполнение поставленной задачи. Меняя параметры закона управления k и α , можно регулировать скорость сходимости выходной переменной $y(t)$ к нулю.

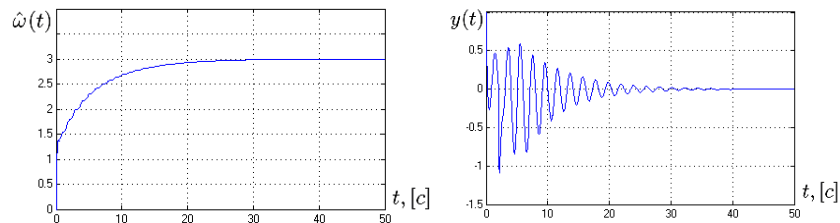


Рис. 1. Временная диаграмма для $\hat{\omega}(t)$ и $y(t)$ при $k = 2$ и $\alpha = 2$

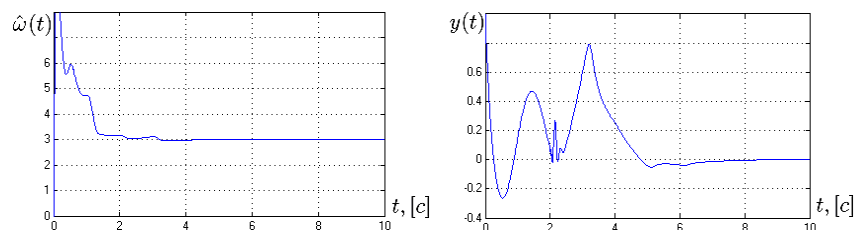


Рис. 2. Временная диаграмма для $\hat{\omega}(t)$ и $y(t)$ при $k = 100$ и $\alpha = 3$

7. Заключение

В данной статье предложен новый алгоритм компенсации параметрически неопределенного конечномерного мультигармонического возмущения, действующего на нелинейный объект управления. В развитии результатов [1, 2, 5–7] допускалось, что возмущение действует на нелинейный объект, в отличие от [5, 6] относительная степень объекта управления может быть любой, в отличие от [1–6] предполагается, что канал управления характеризуется запаздыванием. Развивая подходы [5–7], рассматривается случай смещенного гармонического возмущающего воздействия.

Литература

1. Marino R., Santosuosso G.L., Tomei P. Regulation of Linear Systems with Unknown Additive Sinusoidal Sensor Disturbances // 17th IFAC World Congress, Seoul. – 2008.
2. Marino R., Tomei P. Adaptive Regulator for Uncertain Linear Minimum Phase Systems with Unknown Undermodeled Exosystems Regulation of Linear Systems with Unknown

- Additive Sinusoidal Sensor Disturbances // 17th IFAC World Congress, Seoul. – 2008.
3. Nikiforov V.O. Adaptive nonlinear tracking with complete compensation of unknown disturbances // European Journal of Control. – V. 4. – №. 1. – 1998. – PP. 132–139.
 4. Никифоров В.О. Адаптивное и робастное управление с компенсацией возмущений. – СПб: Наука. – 2003. – 282 с.
 5. Бобцов А.А. Алгоритм управления по выходу с компенсацией гармонического возмущения со смещением // АиТ. – 2008. – №8.
 6. Бобцов А.А., Пыркин А.А. Компенсация гармонического возмущения в условиях запаздывания по управлению // Известия РАН. Теория и системы управления. – 2008. – № 4. – С. 19–23.
 7. Бобцов А.А., Пыркин А.А. Компенсация неизвестного синусоидального возмущения для линейного объекта любой относительной степени // Автоматика и Телемеханика. – 2009. – № 3. – С. 114–122.

АДАПТИВНАЯ СТАБИЛИЗАЦИЯ МАЯТНИКА С ИНЕРЦИОННЫМ МАХОВИКОМ В ПОЛОЖЕНИИ НЕУСТОЙЧИВОГО РАВНОВЕСИЯ

С.А. Колюбин, А.А. Пыркин

Научный руководитель – д.т.н., профессор А.А. Бобцов

Работа посвящена развитию методов адаптивного управления однозвенным маятником с инерционным маховиком. Целью управления является стабилизация маятника в перевернутом положении. Задача решается в условиях полной параметрической неопределенности объекта и при наличии возмущений, а также при ограниченном управляющем воздействии. Особенностью работы является расширенный алгоритм стабилизации, реализованный на базе ПД-регулятора. Предложенный подход позволяет проводить настройку параметров регулятора в режиме «он-лайн».

Ключевые слова: адаптивное управление, идентификация, стабилизация, гибридное управление, мехатронный комплекс

Введение

Маятниковые системы являются прекрасным аналогом для многих естественных, технологических и физических процессов. Задачи исследования динамики и управления подобными системами вызывают в настоящее время значительный интерес, и об этом свидетельствует многообразие посвященных данной тематике работ [1–7]. Наиболее популярными объектами исследований являются маятники с подвижным основанием (маятник Капицы, маятник Фуруты), многозвенные маятники и гиродинные системы.

В современных научных работах в области систем автоматического управления активно используются мехатронные исследовательские комплексы, представляющие собой компактные и высокотехнологичные программно-аппаратные комплексы. Подобные установки могут быть использованы как в образовательных целях, так и при внедрении современных подходов при решении различных прикладных задач.

В данной работе представлены результаты развития адаптивных подходов в управлении маятниковой системой, представляющей собой маятник с инерционным маховиком. Описание подобной системы встречается в ряде публикаций [1–3], [6–7]. Исследования были выполнены с использованием мехатронного комплекса фирмы Mechatronics Systems Inc. Таким образом, авторам удалось проверить работоспособность полученных алгоритмов управления на реальном объекте, отличающемся рядом специфических особенностей, которые было бы сложно учесть, ограничиваясь только математическим моделированием.

Стоит отметить, что до сих пор адаптивным подходам при решении подобных задач было уделено недостаточно внимания, несмотря на очевидную перспективность их применения. Так в работах [1] и [2] параметры объекта принимаются априорно достоверно известными. В работе [3] поставленные задачи решаются при неизвестных параметрах объекта, однако предлагаемый идентификационный подход и метод неявной настраиваемой модели требуют предварительного проведения активного эксперимента для идентификации неизвестных параметров. Такая постановка зачастую оказывается неприемлемой при решении практических задач. Поэтому авторами была поставлена задача разработки алгоритмов управления, позволяющих настраивать параметры регуляторов в процессе работы системы на основании оценок системных параметров. Такой подход позволит значительно расширить область задач, в решении которых могут быть востребованы полученные результаты.

В данной работе в отличие от [4] предложен расширенный алгоритм стабилизации с использованием метода модального управления и стандартных полиномов, реализованный на базе ПД-регулятора.

1. Постановка задачи

Объектом управления является однозвенный маятник с инерционным маховиком. Маятник пассивно закреплен на оси вращения. Точка подвеса расположена на неподвижном массивном основании. На противоположной точке подвеса конце маятника находится инерционный маховик. Схематично система представлена на рис. 1.

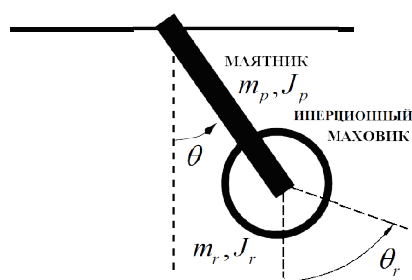


Рис. 1. Схематичное изображение маятниковой системы

Управление колебаниями маятника осуществляется за счет изменения направления и скорости вращения инерционного маховика, управляемого двигателем постоянного тока.

Целью управления является стабилизация маятника в перевернутом положении неустойчивого равновесия.

Математическая модель маятниковой системы, составленная с использованием метода Лагранжа, может быть записана следующей системой дифференциальных уравнений:

$$\begin{cases} \ddot{\theta} + a \sin \theta - b_p u, \\ \ddot{\theta}_r = b_r u, \end{cases} \quad (1)$$

где θ – угол крена маятника относительно вертикали, θ_r – угол поворота инерционного маховика относительно маятника, $a = mgl / J$, $b_p = k_u / J$, $b_r = k_u / J_r$ – обобщенные системные параметры, m – суммарная масса маятника и маховика, l – расстояние от точки подвеса до общего центра масс маятниковой системы, J – суммарный момент инерции маятниковой системы, J_r – момент инерции маховика относительно собственного центра масс, k_u – коэффициент передачи электродвигателя, g – ускорение свободного падения, $|u| < 10$ – ограниченное по модулю управляющее воздействие.

Цель управления может быть выражена предельным равенством:

$$\lim_{t \rightarrow \infty} (\theta(t) - \theta^*) = 0, \quad (2)$$

где $\theta(t)$ – текущее значение угла крена маятника, а $\theta^* = \pi$ – соответствует верхней точке траектории вращения маятника.

Задача решается при полной параметрической неопределенности объекта управления, при этом измерению доступны только выходные переменные θ , θ_r и управление u .

2. Синтез алгоритмов управления

Так как ограниченным по модулю управлением стабилизировать маятник в перевернутом положении при произвольных начальных условиях крайне сложно, последовательно решаются частные подзадачи раскачки и стабилизации.

2.1. Алгоритм раскачки и приведения маятника в заданную область

Для синтеза алгоритма используем метод скоростного градиента [8, 9] с энергетической целевой функцией.

В работе [3] показано, что в задаче управления маятником с инерционным маховиком целесообразно оперировать не гамильтонианом системы, а парциальной энергией, характеризующей движение маятника без учета вращения маховика.

Выражение для целевой энергетической функции записывается в виде:

$$G(\theta, \dot{\theta}) = \frac{1}{2} \Delta H^2,$$

где $\Delta H = H(\theta, \dot{\theta}) - H^*$, $H(\theta, \dot{\theta}) = mgl(1 - \cos \theta(t)) + 0.5J\dot{\theta}^2(t)$ – текущая парциальная энергия системы, $H^* = 2mgl$ – желаемый уровень энергии маятника, соответствующий перевернутому положению.

Полученный из конечной формы алгоритма скоростного градиента, пропорциональный алгоритм управления в общем виде описывается выражением:

$$u = \gamma \Delta H \sigma,$$

где γ – положительный коэффициент, $\sigma = \frac{\partial H(\theta, \dot{\theta})}{\partial u}$.

Применительно к исследуемой системе перепишем его в виде:

$$u(t) = \gamma(0.5J\dot{\theta}^2 - mgl(1 + \cos \theta)) \cdot b_p J \dot{\theta}. \quad (3)$$

Преобразуем закон управления (3):

$$u(t) = \gamma(0.5\dot{\theta}^2 - \frac{mgl}{J}(1 + \cos \theta)) \cdot b_p J \dot{\theta} = Kb_p(0.5\dot{\theta}^2 - a(1 + \cos \theta))\dot{\theta}, \quad (4)$$

где $K = \gamma J^2$ – положительный коэффициент, выбираемый разработчиком.

На основе (4) построим реализуемый алгоритм раскачки:

$$u(t) = Kb_p(0.5\hat{\theta}^2 - \hat{a}(1 + \cos \theta))\hat{\theta}, \quad (5)$$

где \hat{a}, \hat{b}_p – оценки параметров a и b_p соответственно, $\hat{\theta}$ – оценка скорости вращения маятника.

2.2. Алгоритм стабилизации маятника в положении неустойчивого равновесия

В окрестности перевернутого положения для маятника справедливо выражение:
 $\sin \theta \approx (\pi - \theta) \bmod 2\pi$.

Следовательно, локально линеаризованная модель системы (1) представима в виде:

$$\begin{cases} \ddot{\theta} = a\bar{\theta} - b_p u, \\ \dot{\theta}_r = b_r u, \end{cases}$$

где параметр $\bar{\theta} = (\theta - \pi) \bmod 2\pi$ обозначает отклонение маятника от перевернутого положения.

В работе [1] показано, что если при синтезе стабилизирующего алгоритма не учитывать динамику вращения маховика, то после фиксации в перевернутом положении при наличии момента трения маховик будет вращаться с постоянным

ускорением, достаточно быстро произойдет насыщение по управлению, и маятник упадет.

Таким образом, при стабилизации помимо выполнения основного целевого условия (2) надо требовать ограничения скорости вращения маховика в установившемся режиме $\lim_{t \rightarrow \infty} \dot{\theta}_r(t) = 0$.

Также в [1] показано, что при управлении только по переменным $\theta, \dot{\theta}$ и $\dot{\theta}_r$ в случае неточности калибровки энкодера по углу крена маятника в установившемся режиме маховик будет вращаться с постоянной скоростью, а при наличии возмущающего момента – с постоянным ускорением.

Поэтому при работе с реальной системой стабилизирующий алгоритм целесообразно выбирать в форме:

$$u = -k_1 \bar{\theta} - k_2 \hat{\theta} - k_3 \theta_r - k_4 \hat{\theta}_r. \quad (6)$$

Если в качестве желаемого характеристического полинома системы выбрать канонический полином 4-го порядка:

$$A(s) = s^4 + 2(\xi_1 \omega_1 + \xi_2 \omega_2) s^3 + 2(\xi_1 \omega_2 + \xi_2 \omega_1) \omega_1 \omega_2 s + \omega_1^2 \omega_2^2,$$

где $\xi_1, \xi_2, \omega_1, \omega_2 > 0$ – постоянные коэффициенты, влияющие на качество управления.

Тогда согласно методу модального управления коэффициенты регулятора (6) в случае параметрической неопределенности объекта должны вычисляться на основании соотношений:

$$k_1 = -\frac{\hat{a}(\omega_1^2 + 4\xi_1 \xi_2 \omega_1 \omega_2 + \omega_2^2) + \omega_1^2 \omega_2^2 + \hat{a}^2}{\hat{a} \hat{b}_p}, \quad k_2 = -\frac{2\hat{a}(\xi_1 \omega_1 + \xi_2 \omega_2) + 2(\xi_1 \omega_2 + \xi_2 \omega_1) \omega_1 \omega_2}{\hat{a} \hat{b}_p},$$

$$k_3 = -\frac{\omega_1^2 \omega_2^2}{\hat{a} \hat{b}_r}, \quad k_4 = -\frac{2(\xi_1 \omega_2 + \xi_2 \omega_1) \omega_1 \omega_2}{\hat{a} \hat{b}_r},$$

где $\hat{a}, \hat{b}_p, \hat{b}_r$ - оценки соответствующих системных параметров.

При этом оценки угловых скоростей маятника $\hat{\theta}$ и маховика $\hat{\theta}_r$ можно получить посредством наблюдателя Льюенбергера.

3. Идентификации неизвестных параметров объекта

Выполним параметризацию модели объекта управления (1) для получения регрессионной формы.

Введем в рассмотрение функцию $\mathcal{G} = \sin \theta$ и вспомогательные фильтры второго порядка $\xi_1(t) = \frac{\lambda^2}{p^2 + 2\lambda + \lambda^2} \theta(t)$, $\xi_2(t) = \frac{\lambda^2}{p^2 + 2\lambda + \lambda^2} \theta_r(t)$, $\xi_3(t) = \frac{\lambda^2}{p^2 + 2\lambda + \lambda^2} \mathcal{G}(t)$, $\xi_4(t) = \frac{\lambda^2}{p^2 + 2\lambda + \lambda^2} u(t)$. Здесь $\xi_i(t)$ – выходные сигналы фильтров, p – оператор дифференцирования и число $\lambda > 0$.

Запишем модель объекта управления в регрессионной форме:

$$\begin{cases} z_1(t) = \varphi_1(t)^T \Psi_1, \\ z_2(t) = \varphi_2(t)^T \Psi_2, \end{cases}$$

где $z_1(t) = \ddot{\xi}_1$ и $z_2(t) = \ddot{\xi}_2$ – известные функции, $\varphi_1^T(t) = [\xi_3(t) \quad \xi_4(t)]$ и $\varphi_2(t) = [\xi_4(t)]$ – доступные измерению регрессоры, а $\Psi_1^T = [-a \quad -b_p]$ и $\Psi_2^T = [b_r]$ – векторы неизвестных параметров.

Предположим, что на вход объекта поступают сигналы $\{u(1) \ u(2) \ \dots \ u(N)\}$, порождающие последовательность наблюдаемых выходов $\{\theta(1) \ \theta(2) \ \dots \ \theta(N)\}$ и $\{\theta_r(1) \ \theta_r(2) \ \dots \ \theta_r(N)\}$. Для параметризованной модели системы можно выписать массивы данных для N снятых точек $Z_1^T = [z_1(1) \ z_1(2) \ \dots \ z_1(N)]$, $Z_2^T = [z_2(1) \ z_2(2) \ \dots \ z_2(N)]$ по измеримым сигналам и $\Omega_1 = [\varphi_1(1) \ \varphi_1(2) \ \dots \ \varphi_1(N)]$, $\Omega_2 = [\varphi_2(1) \ \varphi_2(2) \ \dots \ \varphi_2(N)]$ – по регрессорам. Значения запоминаются последовательно на каждом шаге функционирования системы.

Тогда алгоритм идентификации представим в виде:

$$\begin{cases} \widehat{\Psi}_1 & (\Omega_1 \Omega_1^T)^{-1} \Omega_1 Z_1, \\ \widehat{\Psi}_2 & (\Omega_2 \Omega_2^T)^{-1} \Omega_2 Z_2, \end{cases} \quad (7)$$

где $\widehat{\Psi}_1$ и $\widehat{\Psi}_2$ – векторы оценок системных параметров.

Замечание: Алгоритм идентификации (7) сойдется, если входной управляющий сигнал $u(t)$ будет «частотно богат».

Итак, реализуемый рекуррентный алгоритм оценки неизвестных параметров имеет вид:

$$\begin{cases} \widehat{a}(i) = -(W_3(i)V_1(i) - W_2(i)V_2(i)) / \Delta_1(i), \text{ если } \Delta_1(i) \neq 0, \text{ иначе } \widehat{a}(i) = \widehat{a}(i-1), \\ \widehat{b}_p(i) = (W_2(i)V_1(i) - W_1(i)V_2(i)) / \Delta_1(i), \text{ если } \Delta_1(i) \neq 0, \text{ иначе } \widehat{b}_p(i) = \widehat{b}_p(i-1), \\ \widehat{b}_r(i) = V_3(i) / \Delta_2(i), \text{ если } \Delta_2(i) \neq 0, \text{ иначе } \widehat{b}_r(i) = \widehat{b}_r(i-1), \end{cases}$$

где $V_1(i) = V_1(i-1) + \xi_3(i)z_1(i)$, $V_1(0) = 0$, $V_2(i) = V_2(i-1) + \xi_4(i)z_1(i)$, $V_2(0) = 0$,
 $V_3(i) = V_3(i-1) + \xi_4(i)z_2(i)$, $V_3(0) = 0$, $W_1(i) = W_1(i-1) + \xi_3(i)\xi_3(i)$, $W_1(0) = 0$,
 $W_2(i) = W_2(i-1) + \xi_3(i)\xi_4(i)$, $W_2(0) = 0$, $W_3(i) = W_3(i-1) + \xi_4(i)\xi_4(i)$, $W_3(0) = 0$,
 $\Delta_1(i) = W_1(i)W_3(i) - W_2(i)W_2(i)$, $\Delta_2(i) = W_3(i)$.

4. Коммутация алгоритмов управления

Выбор и обоснование способа коммутации описаны в [3, 4]. Основное значение имеет подбор условий коммутации так, чтобы избежать нежелательных переходных процессов в системе во время переключения.

Закон управления для гибридной системы имеет вид:

$$u(t) = \begin{cases} u_1(t), \text{ если } |\theta - \pi| \leq \theta_s, \\ u_2(t), \text{ если } |\theta - \pi| > \theta_s, \end{cases}$$

где $u_1(t)$ и $u_2(t)$ – обозначают соответственно алгоритм стабилизации (6) и алгоритм раскачки (5), $\theta_s = 0,5$ рад – угол крена маятника, при котором происходит переключение регуляторов.

5. Результаты эксперимента

На рис. 2 представлены результаты реализации полученных алгоритмов на мехатронном исследовательском комплексе фирмы Mechatronics System.

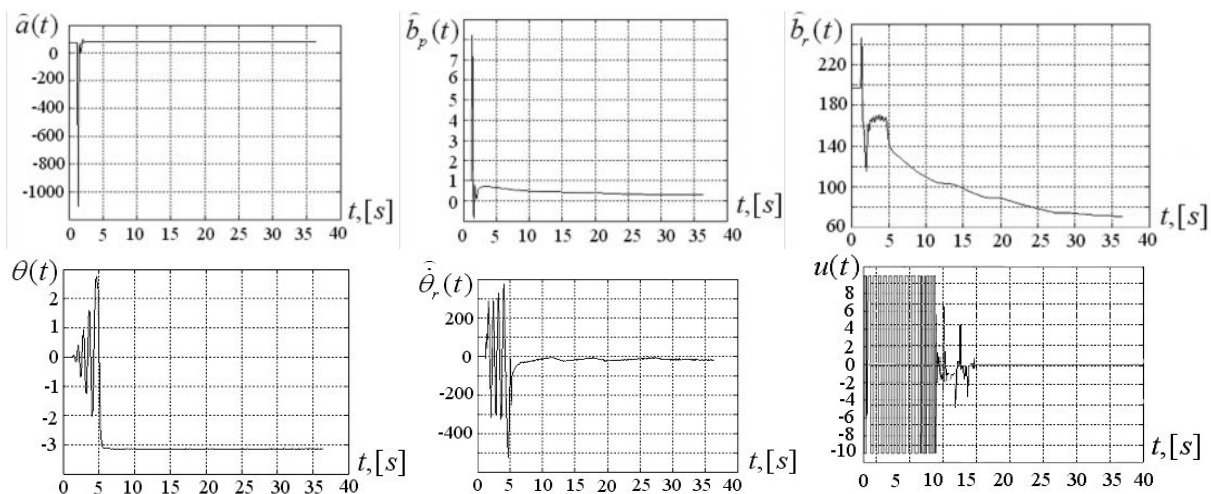


Рис. 2 – Идентификация неизвестных параметров и переходные процессы в реальной системе при параметрах $\lambda = 5$, $K = 5$, $\omega_1 = 2$, $\omega_2 = 1$, $\xi_1 = 0.707$, $\xi_2 = 1$, $\theta_s = 0.5$

Заключение

В работе была получена гибридная система управления на базе адаптивных регуляторов, обеспечивающая стабилизацию маятника с инерционным маховиком в положении неустойчивого равновесия. Полученные алгоритмы являются робастным по отношению к внешним возмущениям, погрешностям измерений и начальным условиям. Предложенный подход обеспечивает решение задачи стабилизации в условиях полной параметрической неопределенности объекта ограниченным управлением, при этом настройка параметров регуляторов выполняется одновременно с работой алгоритмов раскачки и стабилизации.

Литература

1. Astrom K.J, Block D.J., Spong Mark W. The Reaction Wheel Pendulum. Synthesis Lectures on Controls and Mechatronics. – Morgan and Claypool Publishers. – 2007. – 105 p.
2. Spong M.W., Corke P., Lozano R. Nonlinear Control of the Inertia Wheel Pendulum. – Automatica. – 2001. – V. 37. – PP. 1845–1851.
3. Андриевский Б.Р. Стабилизация перевернутого маятника с инерционным маховиком в качестве движителя. Управление в физико-технических системах. Под ред. А.Л. Фрадкова. – СПб: Наука. – 2004. – С. 52–71.
4. Kolyubin S.A., Pyrkin A.A. Adaptive control of a reaction wheel pendulum //Preprints 12th International Student Olympiad on Automatic Control. – 2008. – PP. 23–27.
5. Гришин А.А., Ленский А.В., Охоцимский Д.Е. и др. О синтезе управления неустойчивым объектом. Перевернутый маятник // Изв. РАН. Теория и системы управления. – 2002. – № 5. – С. 14–24.
6. Astrom, K.J., Hybrid Control of Inverted Pendulums, In Yamamoto and Hara, Eds., Learning, Control, and Hybrid Systems. Springer. – 1999. – PP. 150–163.
7. Lozano R., Fantoni I., Block D., Stabilization of the Inverted Pendulum around its Homoclinic Orbit. Systems and Control Letters. – V. 40. – 2000. – PP. 197–204.
8. Мирошник И.В., Никифоров В.О., Фрадков А.Л. Нелинейное и адаптивное управление сложными динамическими системами. – СПб: Наука. – 2000. – 549 с.
9. Фрадков А.Л. Кибернетическая физика: Принципы и примеры. – СПб: Наука. – 2003. – 208 с.

ИССЛЕДОВАНИЕ НАПРЯЖЕННО-ДЕФОРМИРОВАННОГО СОСТОЯНИЯ СОЕДИНЕНИЙ ОПТИЧЕСКИХ ДЕТАЛЕЙ ТРАНСЛЯТОРОВ ИЗОБРАЖЕНИЯ

К.Г. Секарин

Научный руководитель – к.т.н., доцент В.А. Трофимов

Бесклеевые соединения часто описывают фрагментарно с учетом области их применения, особенностей их оптического производства и используемых методов исследования их оптических характеристик, не учитывая при этом влияние поверхностного слоя (ПС) деталей, физико-химические свойства которого определяются технологией ее обработки и могут оказывать существенное влияние на качество самого оптического соединения (ОС) элементов оплотехники и функциональные возможности оптического узла в целом.

Ключевые слова: поляризация света, эллипсометрия, оптический контакт, напряженно-деформированное состояние детали в оптическом узле

Введение

Здесь рассматриваются новые поляризационные методы анализа напряженно-деформированного состояния элементов в их оптических соединениях и эллипсометрические методы технологического контроля качества оптического узла, учитывающего физико-химические свойства неоднородных поверхностных слоев элементов, предназначенных для создания бесклеевых соединений [1, 5].

Основная часть

В интерференционных методах технологического контроля напряженно-деформированного состояния прозрачного изотропного диэлектрика, как правило, используется упрощенная модель плоскопараллельной пластинки одноосного кристалла, оптическая ось которого перпендикулярна световому пучку света и находится в плоскости ее параллельных поверхностей. В пределах границ применимости теории упругой деформации и напряженного состояния оптической детали, изменение показателя преломления $\Delta n(\sigma)$ для одностороннего «растяжения» или «сжатия», в первом приближении, будет пропорционально величине напряжений σ , определяемых по формулам:

$$n - n_{pN} = \Delta n_{pN} = \pm C_1 \cdot \sigma, \quad n - n_{sN} = \Delta n_{sN} = \pm C_2 \cdot \sigma, \quad (1)$$

где n – показатель преломления силикатного стекла; n_{pN} , n_{sN} – показатели преломления для поляризованного света с электромагнитными колебаниями параллельными (р) и перпендикулярными (s) направлению действия напряжений; C_2 и C_1 – фотоупругие постоянные стекла для тех же направлений электромагнитных колебаний; σ – величина нормальных напряжений. В теории упругой деформации детали напряжения «сжатия» принято считать отрицательными, а напряжения «растяжения» положительными.

Метод эллипсометрии обладает высокой чувствительностью к изменению двулучепреломления $\Delta n = n_{pN} - n_{sN}$, вызванного напряженно-деформированным состоянием детали в оптическом узле и позволяет определить направление воздействия напряжений q (или механической нагрузки F) в области зоны контакта оптического соединения деталей (ЗОС). Для прозрачных изотропных диэлектриков, находящихся в напряженно-деформированном состоянии, наибольшей точности измерения поляризационных параметров можно достичь при использовании метода «нулевой» (компенсационной) эллипсометрии, сущность которого состоит в том, что

азимуты поляризующих элементов прибора (поляризатора P, компенсатора C, анализатора A) фиксируются в момент «гашения» светового пучка, прошедшего оптическую схему прибора и объект измерения S (рис. 1а). Тогда поляризационные характеристики объекта S можно определить в результате решения нелинейного уравнения:

$$-\operatorname{tg} A_1 = \rho_s \cdot [\rho_c + \operatorname{tg} C_1 \cdot \operatorname{tg} P_1] \cdot [\rho_c \cdot \operatorname{tg} C_1 - \operatorname{tg} P_1]^{-1}, \quad (2)$$

$$A_1 = A - q, \quad C_1 = C - q, \quad P_1 = C - P,$$

$$\rho_s = \left| \frac{T_s^{(p)}}{T_s^{(s)}} \right| \cdot \exp i \delta_s, \quad \rho_c = \left| \frac{T_c^{(p)}}{T_c^{(s)}} \right| \cdot \exp i \delta_c. \quad (3)$$

Здесь $T_c(p,s), T_s(p,s)$ и δ_c, δ_s – коэффициенты пропускания и разность фаз между p- и s-поляризацией светового пучка соответственно для компенсатора C и объекта S; q – ориентация главной плоскости падения светового пучка в анизотропной оптической системе S, где значение угла q определяет направление действия сил F «растяжения» или «сжатия» относительно нормали N к поверхности элементов в оптическом соединении (рис. 1а, линия FF').

Из условия равенства нулю мнимой части уравнения (2), при азимуте ориентации осей компенсатора относительно плоскости падения светового пучка (плоскость XOZ) $C = \pm \pi/4$ и значении $\rho_c = i$, основные эллипсометрические параметры объекта исследования (δ_s, q) при нормальном падении светового пучка вдоль зоны ОС деталей определяются соотношениями:

$$q = (0,5 \cdot \operatorname{tg}[-\cos 2A \cdot (\sin 2A + \sin 2P)^{-1}], \delta_s = \operatorname{arctg}[-(\operatorname{tg} 2P \cdot \cos 2q)^{-1}]. \quad (4)$$

Для модели напряженно-деформированного состояния изотропного прозрачного диэлектрика, представленной в виде плоскопараллельной пластинки одноосного кристалла, фазовый сдвиг определяется как $\delta_s = \delta_{s,0} + \delta_1$, где поправка δ_1 учитывает многократные отражения от поверхностей объекта исследования [4]. При малой величине $\Delta n_p N$ и $\Delta n_s N$ значение фазового параметра $\delta_{s,0} \gg \delta_1$, тогда для определения величины напряжений σ в оптическом соединении деталей можно использовать следующие соотношения:

$$\Delta n_{p,s} = n_{pN} - n_{sN} = \pm (C_1 - C_2) \cdot \sigma = B \cdot \sigma = \delta_{s,0} / (k_0 \cdot L), \quad k_0 = 2\pi / \lambda, \quad (5)$$

где B – оптический коэффициент напряжений; λ – длина волны излучения; L – длина объекта измерений S вдоль оси X (рис. 1а).

В частности, на рис. 1 показаны два типа соединений оптических элементов, полученных методом ОК и различающиеся характером напряженно-деформированного состояния. Если в первом случае (рис. 1б, в) характерным изменением величины нормальных напряжений $\sigma(y)$ в оптическом элементе является то, что на одном краю ОС деталей наблюдаются напряжения «сжатия» (« $-\sigma$ »), а на другом краю – напряжения «растяжения» (« $+\sigma$ »), то для другого оптического соединения элементов – в центральной зоне наблюдаются напряжения «сжатия» ($\sigma \leq 0$), а концах – напряжения «растяжения» ($\sigma > 0$) (рис. 1г, д). В последнем случае такое изменение $\sigma(y)$ может иметь место при сопряжении двух поверхностей деталей, имеющих отклонение от плоскостности ($N, \Delta N$) характерное для выпуклой поверхности деталей (рис. 1г), где после релаксации исходного напряженно-деформированного состояния элементов, возникающего при получении оптического контакта (ОК) путем сжимающих силовых нагрузок $F_{сжим.}$, упругие внутренние напряжения приведут к появлению растягивающих оптическое соединение элементов сил $F_{раст.}$

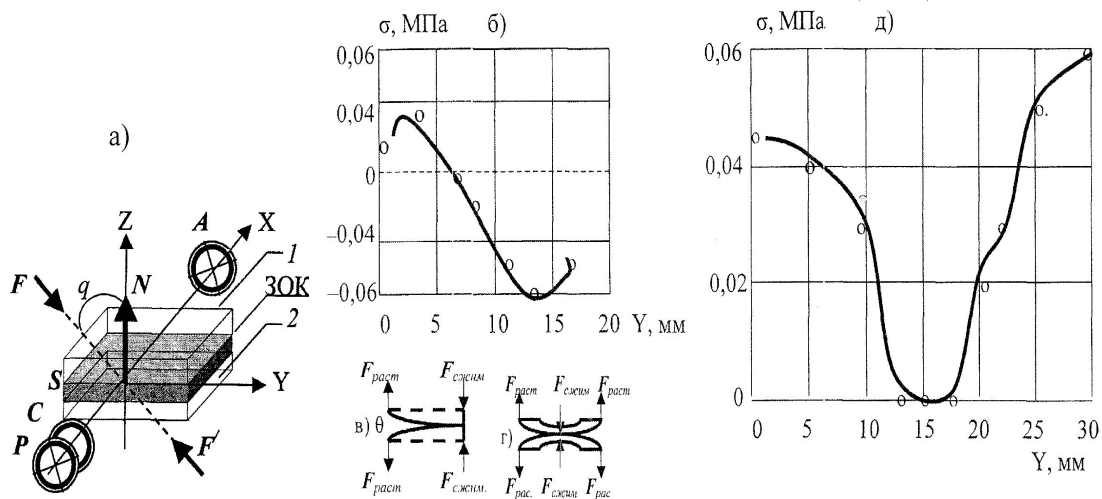


Рис. 1. Схема эллипсометрического анализа напряженно-деформированного состояния оптического соединения деталей (а) и распределение нормальных напряжений в оптических элементах (б, д) после получения оптического контакта при различном направлении прикладываемой нагрузки (в, г). Обозначения: P – поляризатор; C – компенсатор; S – объект измерений; A – анализатор; ЗОК – зона оптического контакта; 1, 2 – оптические элементы; FF' – направление действия напряжений (или внешних сил F) «растяжения» или «сжатия» относительно нормали N к поверхности элементов под углом q в оптическом соединении деталей

Знакопеременный характер изменения $\sigma(y)$ можно отнести также к оптическим соединениям деталей, поверхности которых имеют клиновидность θ (рис. 1в). Естественно, что напряженно-деформированное состояние оптических соединений будет определяться не только макро- и микрогеометрией поверхности деталей, но и физико-химическим состоянием ПС, которое оценивается по поляризационно-оптическим параметрам, определяемым по методике эллипсометрического анализа многослойной системы, описанной в [4].

Успешное применение эллипсометрических методов технологического контроля качества оптических соединений элементов оптоэлектроники связано с решением целого ряда взаимосвязанных задач, в том числе: выбором метода физико-математического моделирования напряженно-деформированного и физико-химического состояния многослойной отражающей системы: «однородная подложка (ОП₁) – неоднородный слой (ПС₁) – зона оптического контакта (ЗОК) поверхности элементов (1,2) – неоднородный слой (ПС₂) – однородная подложка (ОП₂)» (рис. 1а) и методики определения ее поляризационно-оптических характеристик, которые, в общем случае, должны решаться в рамках теории эллипсометрии неоднородных анизотропных оптических систем [4].

Для оценки качества ОС элементов оптоэлектроники, при различных технологических условиях их изготовления и оптического узла в целом, определение поляризационно-оптических характеристик отражающей неоднородной системы можно проводить двумя путями: методом секционирования [4] и методом физико-математического моделирования структуры неоднородной оптической системы, используя при этом модели «эффективной подложки» или «эффективного слоя». В методе «эффективной подложки» многослойная система «ПС₁–ЗОК–ПС₂–ОП₂»

(рис. 1а, луч 2) описывается эффективным показателем преломления n_{oc}^* , который для анизотропной однородной подложки при $q \approx 0$ можно определить из условия равенства произведений адмиттансов для р- и s-компонент поляризованного света однородной анизотропной и изотропной подложки как

$$[(U_0^{(s)}) \cdot (U_0^{(p)})]_{\text{ан.}} = [(U_0^{(s)}) \cdot (U_0^{(p)})]_{\text{изот.}}, \quad n_{OC}^* = \sqrt{n_{pN} \cdot n_{sN}} \quad (6)$$

В методе «эффективного слоя» для системы «ПС₁–ЗОК–ПС₂» вводятся эффективный показатель преломления n_{OK}^* и толщина слоя d_{OK}^* неоднородной оптической системы (рис. 2а, в). Физический смысл этих параметров состоит в том, что модель однородного слоя с параметрами n_{OK}^* и d_{OK}^* и оптическое соединение деталей (ОС) по своим поляризационно-оптическим свойствам эквивалентны. В рамках этой модели ОК можно описывать анизотропные свойства оптического соединения деталей, используя при этом методы эллипсометрических измерений элементов нормированной матрицы отражения объекта измерений $M=[\rho_{pp}, \rho_{ps}, \rho_{sp}]$.

В общем случае, эллипсометрические параметры элементов нормированной матрицы отражения $\rho_{pp}, \rho_{ps}, \rho_{sp}$ будут отличаться от измеренных значений $\rho_{изм}$ (рис. 2а, луч 2) за счет преломления светового пучка на границах раздела «внешняя среда–ОП₁» и «ОП₁–внешняя среда» на величину эллипсометрического отношения $\rho(T)=T(p)/T(s)$.

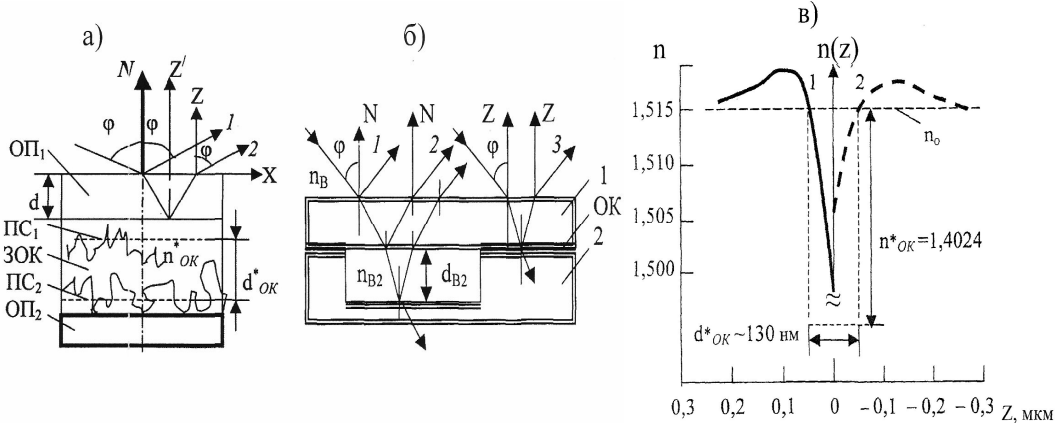


Рис. 2. Модель многослойной отражающей системы оптического контакта (ОК), метод эллипсометрических измерений характеристик оптического соединения элементов (б) и изменение показателя преломления $n(z)$ в оптическом контакте деталей 1 и 2 (в).

Обозначения: ОП_{1,2} – однородная подложка; ПС_{1,2} – поверхностный слой однородной подложки; ЗОК – зона оптического контакта оптических элементов; n_{OK}^* , d_{OK}^* – эффективный показатель преломления и толщина ОК; n_{B2} , d_{B2} – показатель преломления и толщина воздушного промежутка в оптическом соединении деталей 1 и 2

При малой величине анизотропии $\Delta n_{p,s} \ll 10^{-6}$ однородной подложки ОП₁ и измеренном значении коэффициента эллиптичности χ отраженного светового пучка от однородной подложки ОП₁ (рис. 2а, луч 1), эллипсометрические соотношения для нахождения параметров $\rho_{pp}, \rho_{ps}, \rho_{sp}$ имеют вид:

$$\rho_{pp} \cdot (\rho^{(T)})^2 = \rho_{pp}^{изм.}, \quad \rho_{ps} \cdot \rho^{(T)} = \rho_{ps}^{изм.}, \quad \rho_{sp} \cdot \rho^{(T)} = \rho_{sp}^{изм.} \quad (7)$$

$$\text{tg}\Psi^{(T)} = \left(\frac{n_o \cdot (U_o^{(s)} + U_B^{(s)})}{(n_B \cdot U_o^{(s)} + n_o^2 \cdot \cos \varphi)} \right) \Delta^{(T)} = \pm 4\chi \frac{n_B (n_o^2 - n_B^2) \sin^2 \varphi}{\sqrt{n_B^2 + n_o^2 (n_o^2 \cos^2 \varphi + n_B \cdot U_o^{(s)})}} \quad (8)$$

$$U_B^{(s)} = \sqrt{\varepsilon_B} \cdot \cos \varphi, \quad U_o^{(s)} = (\varepsilon_o - \varepsilon_B \cdot \sin^2 \varphi)^{1/2} \quad (9)$$

где n_B, n_o – показатель преломления внешней среды и однородной подложки; φ – угол падения светового пучка; χ – эллиптичность светового пучка отраженного от ОП₁ при угле Брюстера $\varphi_B = \arctg(n_o/n_B)$; при изменении угла падения светового пучка $\varphi > \varphi_B$ значение $\Delta(T) > 0$ и при $\varphi < \varphi_B$ значение $\Delta(T) < 0$ [2]. В том случае, когда главная плоскость анизотропной системы совпадает с плоскостью падения светового пучка, т.е. при $q \approx 0$, то параметры $\rho_{ps} = \rho_{sp} = 0$, а эллипсометрическое отношение ρ_s от объекта измерения S определяется как

$$\rho_s = \rho_{pp} = R^{(p)} / R^{(s)} = \text{tg}\Psi \cdot \exp(i\Delta), \quad \text{tg}\Psi = \text{tg}\Psi^{изм.} / (\text{tg}\Psi^{(T)})^2, \quad \Delta = \Delta^{изм.} \mp \Delta^{(T)} \quad (10)$$

При изготовлении низкочастотных модуляторов оптического излучения, в которых с помощью пьезокерамического модулятора производится изменение толщины воздушного промежутка в оптическом соединении деталей (рис. 2б) требуется прецизионный контроль толщины этого промежутка dB_2 , величина которого мала $dB_2 < \lambda$ и составляет несколько долей длины волны модулируемого излучения. Обычно такие модуляторы построены по схеме нарушенного полного внутреннего отражения (НПВО). В эллипсометрии контроль оптических параметров однородной пленки, заключенной между двумя однородными подложками носит название «метод свободной пленки», особенности которого подробно изложены в [4]. Однако, при наличии поверхностного слоя подложка становится неоднородной и определение оптических параметров пленки (или слоя) между двумя неоднородными подложками по ранее используемому «методу свободной пленки» для однородных сред может давать ошибочный результат.

Для детального анализа поляризационно-оптических свойств соединений элементов оптоэлектроники в оптических узлах, целесообразно использовать метод секционирования многослойной оптической системы, сущность которого состоит в том, что оптические параметры каждого j -ого слоя в системе $ПС_1$ –ЗОК– $ПС_2$ – $ОП_2$ определяются по параметрам ρ_{j-1} и ρ_j измеряемым в процессе синтеза многослойной оптической системы. Например, до и после нанесения пленки (клеевого слоя) на однородную подложку с диэлектрической проницаемостью ϵ_0 или до и после получения оптического контакта.

Ранее полученное уравнение эллипсометрии для системы «неоднородный ПС – неоднородная подложка» непосредственно связывает оптические характеристики j -ого слоя с эллипсометрическими параметрами ρ_j и ρ_{j-1} и для его решения не требуется априорной информации о виде оптического профиля $\epsilon(z)$ нижележащей неоднородной отражающей системы [2]:

$$\Phi_j(p) = (ik_0 A_0 \rho_0)^{-1} \cdot [(\rho_j - \rho_0) - (\rho_{j-1} - \rho_0) \cdot \exp(i2k_0 U_0^{(s)} d_j)] \quad (11)$$

$$\Phi_j(p) = \int_0^{d_j} (\epsilon_j(z) - \epsilon_0) \cdot (\epsilon_j(z) - \epsilon_B) \cdot \epsilon_j^{-1}(z) \cdot \exp(-i2k_0 U_0^{(s)} z) dz. \quad (12)$$

Применение метода секционирования позволяет избежать методической ошибки при расчете оптических параметров однородного слоя nc и dc , находящегося на неоднородной подложке с профилем $\epsilon(z)$.

В табл. 1 приведены поляризационно-оптические параметры в различных зонах оптических узлов элементов (рис. 2б, лучи 1–3), полученных из стекла К8 ($n_0=1,51466$), где, до получения ОК деталей, пучком ионов Ag^+ сделана выборка материала, глубина dB_2 которой контролировалась методом интерферометрии на приборе МИИ-4 с погрешностью $Sd \leq 0,03$ мкм и равна $d_{инт} \approx 0,3-0,4\lambda$. Оптические узлы I и II отличаются физико-химическим состоянием поверхности элемента 2 в зоне выборки материала (рис. 2в, луч 2), т.е. наблюдается изменение состояния поляризации отраженного светового пучка: для I характерна левоэллиптическая поляризация ($\pi < \Delta(\varphi) < 2\pi$), а для II – правоэллиптическая поляризация ($0 < \Delta(\varphi) < \pi$).

Поляризационные параметры Δ и Ψ измерялись на приборе ЛЭФ–3М при $\lambda=0,6328$ мкм, где погрешность измерения азимутов поляризующих элементов составила $SP, A=1/$, а установка угла падения светового пучка $S\varphi=0,5/$. Учет поляризационно-оптических свойств $ПС_2$ элемента 2 приводит к уточнению значений параметров nB_2 и dB_2 , где изменение глубины dB_2 находится в пределах ошибки измерения параметров Δ и Ψ , а при $nB_2 \rightarrow nB$. Средние значения оптических параметров nB_2 и dB_2 выборки материала согласуются со значениями показателя внешней среды $nB_2 \approx nB=1,0003$ и толщиной $dB_2 \approx d_{инт}$, При этом значение показателя преломления в

системе «ПС₁–ЗОК–ПС₂–ОП₂» равно $n_{OK}=1,4024$, что согласуются с данными работы [3].

Детальный анализ поляризационно-оптических свойств ОС показывает, что полученные значения параметров – показатель преломления n^*_{OK} и толщина d^*_{OK} – дают информацию не только о ЗОК, толщиной $d_{ЗОК} \sim 30\text{--}60 \text{ \AA}$ и показателем преломления $n_{OK} \sim 1,40 \div 1,43$ [3], но и сильно градиентной области ПС элементов, прилегающей к зоне непосредственного оптического контакта поверхностей деталей 1 и 2 (рис. 2в).

Таблица 1. Поляризационно-оптические характеристики элементов оптических соединений

Зона ОС	Элемент ОС	Угол измерения	Поляризационные параметры		Оптические параметры	
		φ	Δ	Ψ	n	d , мкм
1	ПС ₁	50°	179°17'	10°09'	1,5211	0,197
		60°	1°05'	5°14'	1,5192	0,183
2	I (ПС ₂)	50°	180°17'	10°13'	1,5181	0,096
	I (1–2)	50°	185°18'	11°05'	1,0342	0,190
2	II (ПС ₂)	60°	1°55'	5°16'	1,5058	0,064
	II (1–2)	60°	9°53'	6°18'	0,9785	0,235
Средние значения $n_{B2}=1,0063$, $d_{B2}=0,213$ мкм						
3	II (ПС ₁)	60°	1°21'	5°13'	1,4975	0,064
	II (ОС)	60°	358°53'	12°30'	1,4024	0,131

Примечание: для II (ПС₁, ПС₂) значения n и d приведены для модели неоднородного ПС до получения ОС, где $n=nq$ и $d=dq$, для I (ПС₁, ПС₂) – для модели однородного ПС.

Вывод

Таким образом, разработанный метод эллипсометрического контроля оптических соединений элементов оптоэлектроники позволяет определять обобщенный показатель качества оптического узла, количественная оценка которого определяется по эффективным параметрам однородного слоя: показателю преломления $n=nq$ и толщине слоя $d=dq$.

Литература

1. Архипова Л.Н., Ивашевский С.Н., Карапетян Г.О. и др. Градиентная оптика для медицинских эндоскопов // Оптический журнал. – 1994. – №12. – С. 51–54.
2. Прокопенко В.Т., Храмцовский И.А., Землянский В.С., Лисицын Ю.В., Секарин К.Г. Эллипсометрия оптических соединений элементов оптоэлектроники // Приборостроение. Изв. вузов. – 2008. – Т.51. – № 10. – С. 57–65.
3. Абаев М.И., Лисицын Ю.В., Путилин Э.С. Исследование зоны оптического контакта стеклянных поверхностей методом эллипсометрии // Письма в ЖТФ. – 1984. – Т.4. – Вып.24. – С. 1505–1507.
4. Эллипсометрия: теория, методы, приложение // Под ред. А.В. Ржанова и Л.А. Ильина, Новосибирск, «Наука». – 1987. – 192 с.
5. Левинок В.Е. Влияние условий полимеризации однокомпонентного оптического клея на свойства склеивающего слоя // ОМП. – 1980. – №6. – С. 24–26.

МОДЕЛИРОВАНИЕ ПОЛЕЙ ТЕМПЕРАТУР И ТЕРМОНАПРЯЖЕНИЙ В ПРОЦЕССЕ НАГРЕВА АЛЮМИНИЕВЫХ ЗАГОТОВОК, ВРАЩАЮЩИХСЯ В МАГНИТНОМ ПОЛЕ ПОСТОЯННОГО ТОКА

С.Е. Коршиков, Н.В. Заикина, Г.С. Рыбалко
(Самарский государственный технический университет)
Научный руководитель – к.т.н., доцент Ю.Э. Плешивцева
(Самарский государственный технический университет)

Работа посвящена моделированию, исследованию и оптимальному по быстродействию управлению новой инновационной энергосберегающей технологией индукционного нагрева алюминиевых заготовок, вращающихся в магнитном поле постоянного тока. Приведены результаты численного расчета температурных полей, а также полей термонапряжений, возникающих в заготовке в процессе нагрева.

Ключевые слова: моделирование, индукционный нагрев, энергосберегающая технология, температурное поле, термонапряжение

Введение

Индукционный нагрев алюминия, обладающего высокой удельной проводимостью, является одним из самых крупномасштабных электротехнологических процессов, применяющихся в цветной металлургии, особенно в технологиях горячей обработки металлов давлением (прессование, прокат, штамповка). Именно этот вид нагрева в большинстве случаев является наиболее распространенным, не смотря на довольно низкий КПД, составляющий 55–60% [1].

При традиционном способе индукционного нагрева металлических заготовок в переменном электромагнитном поле КПД процесса зависит от величины потерь в индукторе. Даже при оптимальном дизайне индуктора (разбиение индуктора на оптимальное число секций с целью снижения плотности тока) КПД установки обычно не превышает 60% [1].

Для повышения энергетической эффективности процесса индукционного нагрева в последние годы была разработана принципиально новая технология, которая заключается в использовании сверхпроводящих материалов для возбуждения постоянного магнитного поля высокой интенсивности [2]. Данная технология основана на способности сверхпроводящих материалов проводить постоянный ток практически без потерь. Нагрев в таких установках осуществляется за счет токов, индуцируемых в заготовке при ее вращении в постоянном магнитном поле. Коэффициент полезного действия при таком методе нагрева теоретически может достигать 90%.

В работе исследуется актуальная проблема моделирования и управления описанной энергосберегающей технологией нагрева.

Инновационная технология нагрева алюминиевых заготовок перед прессованием

При вращении заготовки с постоянной скоростью в направленном перпендикулярно оси её вращения (рис. 1) постоянном магнитном поле, магнитный поток изменяется по синусоидальному закону, в результате чего и производится нагрев изделия индуцируемыми вихревыми токами. Конструктивные параметры системы нагрева представлены на рис. 2–3.

Для обеспечения охлаждения и выдержки сверхпроводящей катушки при низкой температуре, её размещают в криостате, в котором рабочий объём поддерживается при низкой температуре за счёт постороннего источника холода, в качестве которого используют жидкие газы с низкими температурами конденсации (азот, водород, гелий и др.) [2].



Рис. 1. Принцип индукционного нагрева заготовки, вращающейся в магнитном поле сверхпроводника

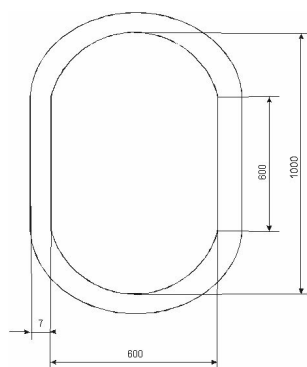


Рис. 2. Конфигурация индуктора

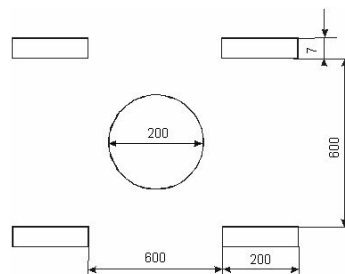


Рис. 3. Поперечное сечение заготовки и индуктора

В процессе нагрева алюминиевых цилиндрических заготовок индукционным методом появляется перепад между температурами поверхности и центра, в результате чего поверхностные слои заготовок оказываются более прогретыми в отличие от слоев, расположенных около центра. Поэтому для обеспечения требуемого равномерного температурного распределения по сечению заготовки, а также для предотвращения возникновения термических напряжений благодаря температурным перепадам необходимо решить задачу оптимального управления процессом нагрева.

Численная двумерная модель температурных полей и полей термонапряжений

Для исследования процесса индукционного нагрева использовалась двумерная численная модель, разработанная в среде программного пакета ANSYS, которая позволяет моделировать вихревые токи внутри алюминиевой заготовки, вращающейся в магнитном поле постоянного тока [2].

Математическая модель системы нагрева алюминиевых заготовок вращением включает взаимосвязанный расчет электромагнитных и тепловых полей в поперечном сечении заготовки в предположении, что длина заготовки много больше её радиуса.

Для моделирования исследуемого процесса был выбран метод конечных элементов. При моделировании процесса учитывается, что теплофизические свойства материала заготовки существенно зависят от температуры и от неравномерности магнитного поля. Алгоритм решения задачи моделирования включает последовательную итерационную процедуру, включающую гармонический электромагнитный анализ и переходный тепловой анализ.

Моделирование поля температур в процессе оптимального по быстродействию нагрева

На первом этапе была сформулирована и решена задача оптимального по быстродействию управления процессом индукционного нагрева, который рассматривался в качестве объекта оптимизации с распределенными параметрами. Состояние объекта однозначно определяется пространственно-временным распределением температуры нагреваемого тела $\theta(l, \tau)$ во времени τ и по радиальной координате $l \in [0, R]$, где R – радиус поперечного сечения заготовки, в пренебрежении температурной неравномерностью по её длине. Холодные заготовки поступают в установку с начальным распределением температуры $\theta_0(l)$ и нагреваются в процессе вращения до заданной температуры $\theta^*(l)$. Индукционная установка должна обеспечивать нагрев металлической заготовки до заданной температуры с допустимым температурным отклонением по сечению заготовки ε_0 . Изменение частоты вращения заготовки во времени $f(\tau)$, позволяющее целенаправленно изменять температурное поле заготовки, можно рассматривать в качестве сосредоточенного управляющего воздействия. В качестве критерия оптимальности для обеспечения максимальной производительности установки рассматривается минимальное время нагрева τ^0 .

Тогда задача оптимального по быстродействию управления процессом нагрева заготовки, вращающейся в магнитном поле постоянного тока, может быть сформулирована следующим образом: необходимо найти такое переменное во времени управляющее воздействие $f(\tau) = f_{opt}(\tau)$, которое обеспечивает перевод нагреваемого изделия с начальным распределением температуры $\theta_0(l) = \theta_0 = const$ в заданное состояние $\max_{l \in [0, R]} |\theta(l, \tau^0) - \theta^*(l)| \leq \varepsilon_0$ за минимально возможное время τ_{min}^0 при выполнении ограничения $f_{min} \leq f(\tau) \leq f_{max}$.

Применительно к исследуемому в настоящей работе классу задач оптимизации индукционная установка должна обеспечивать нагрев заготовок с приемлемой для типовых технологий точностью $\varepsilon_0 = 10 \div 15$ °С.

При использовании численной модели температурного поля были получены результаты расчета оптимального по быстродействию процесса нагрева алюминиевых цилиндрических заготовок до температуры 500°С диаметром 215 мм путем их вращения в постоянном магнитном поле при токе источника питания 323,2 кА при $f(\tau) = f_{max}$, приведенные в табл. 1.

Таблица 1. Результаты моделирования температурных полей в процессе одноинтервального нагрева при $f(\tau) = f_{max}, \forall \tau \in [0; \tau^0]$

Частота вращения f_{max} , об/с	Время нагрева τ_{min}^0 , с	Максимальное температурное отклонение на выходе, °С	Максимальная температура в процессе нагрева, °С
8	536,137	7,834	507,834
16	346,964	15,357	515,357
25	264,18	23,056	523,056
35	214,849	30,966	530,966
50	172,229	41,403	541,402

Эти результаты показывают, что допустимое температурное отклонение $\varepsilon_0=12.5^\circ\text{C}$ ($\pm 2.5\%$) может быть получено только при частоте вращения заготовки меньшей, чем 16 об/с, что приводит к существенному увеличению времени нагрева. Отклонение $\varepsilon_0=41.4^\circ\text{C}$, получаемое при максимальной частоте вращения, равной 50 об/с, означает неравномерность нагрева, которая обычно недопустима для последующего процесса горячего прессования.

Поэтому необходимо применение двухинтервального оптимального алгоритма управления. В этом случае на первом этапе происходит нагрев с максимальной частотой вращения f_{max} , на втором этапе при отключенной частоте вращения заготовки ($f = 0$) происходит выравнивание температуры по сечению заготовки и достижение температурной однородности на выходе, требуемой для дальнейшей обработки давлением.

Переход к двухстадийному процессу с форсированным нагревом на первой стадии и выравниванием температурного перепада при отключенной мощности – на второй, всегда приводит, как показывает анализ, к меньшему времени нагрева.

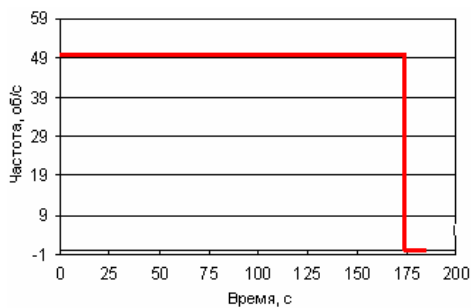


Рис. 4. Двухинтервальный оптимальный алгоритм управления частотой

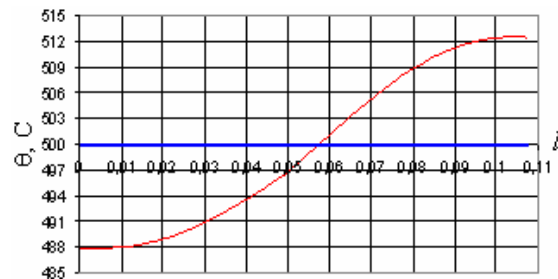


Рис. 5. Температурный профиль по сечению заготовки в конце оптимального процесса двухинтервального нагрева

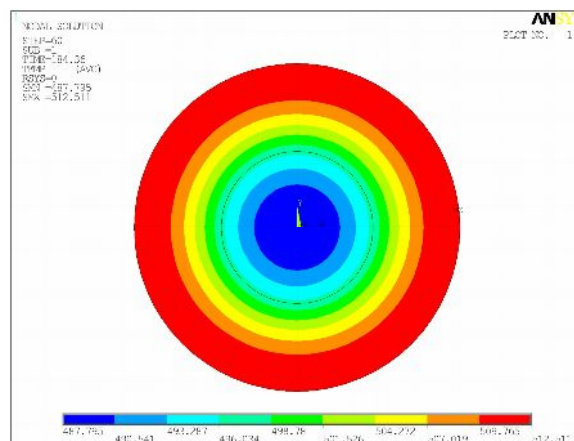


Рис. 6. Температурное распределение на выходе при оптимальном двухинтервальном нагреве

Решение задачи поиска оптимального по быстродействию двухинтервального управления для $f_{max} = 50$ об/с представлено на рис. 4–6. На рис. 4 показана зависимость оптимальной частоты вращения заготовки от времени. Длительность интервала нагрева с максимальной частотой вращения составляет 173 с, продолжительность интервала выравнивания температуры – 11 с, максимальное температурное отклонение на выходе $\varepsilon_0=12.5^\circ\text{C}$. Оптимальное температурное распределение по сечению заготовки на выходе из нагревателя приведено на рис. 5. На рис. 6

продемонстрированы результаты моделирования оптимального процесса нагрева с помощью прикладного пакета ANSYS.

Моделирование полей термических напряжений в процессе нагрева

При нагреве заготовки помимо температуры необходимо также учитывать и уровень возникающих термических напряжений.

В данной работе за критическое значение температурного напряжения был взят предел упругости (значение напряжения, при котором в металле начинают возникать пластические деформации), который для алюминия составляет 30 МПа.

Используемая численная двумерная ANSYS-модель позволяет рассчитывать напряжения, возникающие из-за температурных перепадов по сечению нагреваемой заготовки.

Для анализа были выбраны наиболее опасные растягивающие радиальные термонапряжения, определяемые выражением:

$$\sigma_r = \frac{\alpha E}{1-\nu} \left(\frac{1}{b^2} \int_0^b \theta r dr - \frac{1}{r^2} \int_0^r \theta r dr \right), \quad (1)$$

где ν – коэффициент Пуассона; α – коэффициент линейного расширения; E – модуль упругости или модуль Юнга; b – радиус цилиндра; θ – функция распределения температуры.

Было проведено сравнение результатов аналитического расчета и численного моделирования полей радиальных термонапряжений, возникающих внутри заготовки при ее нагреве. Для выбранного температурного состояния были рассчитаны теоретические значения термонапряжения, согласно (1), также получено поле термонапряжений с использованием модели ANSYS. Полученные результаты, представленные в табл. 2, свидетельствуют о высокой сходимости результатов численного моделирования и аналитических расчетов.

Таблица 2. Результаты сравнения аналитического и численного расчетов

	х, mm	0	0.025	0.05	0.0823	0.1075
σ _r , Па	Численное моделирование	5.8058·10 ⁷	5.5978·10 ⁷	4.7158·10 ⁷	2.3629·10 ⁷	6.3239·10 ⁶
	Аналитический расчет	5.851·10 ⁷	5.582·10 ⁷	4.716·10 ⁷	2.398·10 ⁷	6.816·10 ⁶

Для исследования поведения термонапряжения в процессе нагрева были использованы результаты оптимального по быстрдействию нагрева, приведенные в табл. 1. На рис. 7–8 представлено изменение термонапряжений в одностадийном процессе нагрева.

Как видно из приведенных результатов, максимальное термонапряжение возникает в центре заготовки. Также из рис. 7 следует, что нагрев изделия при максимальной частоте вращения приводит к существенному превышению допустимого значения термонапряжения. Поэтому для удержания термонапряжений на максимально допустимом уровне должна быть сформулирована и решена задача оптимального по быстрдействию нагрева с учетом технологических ограничений.

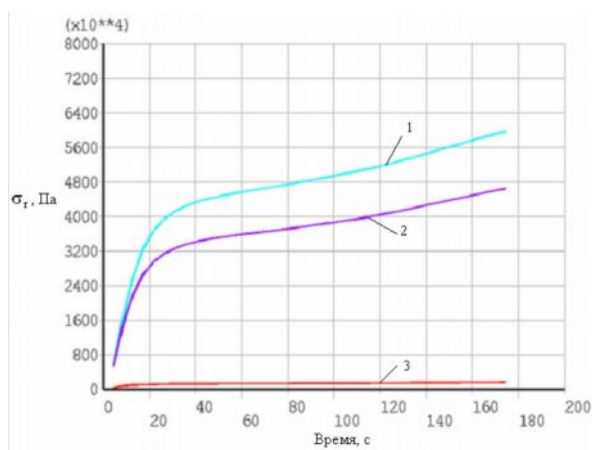


Рис. 7. Изменение термонапряжения в заготовке в процессе одноинтервального оптимального по быстродействию управления при частоте 50 об/с: 1 – в центре заготовки; 2 – в промежуточной точке; 3 – на поверхности заготовки

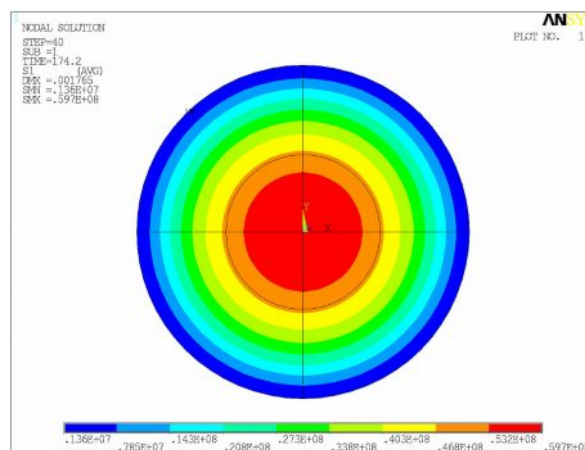


Рис. 8. Поле термонапряжений в конце процесса нагрева

Заключение

В данной работе рассматривается новая энергосберегающая технология процесса индукционного нагрева алюминиевых заготовок перед прессованием путем их вращения в магнитном поле постоянного тока. Путем численного моделирования был проведен анализ поведения температурного поля при оптимальных одноинтервальном и двухинтервальном режимах нагрева для различных значений максимально допустимой частоты вращения в диапазоне от 8 до 50 об/с. Из представленных результатов следует, что только применение двухинтервального управления позволяет обеспечивать заданную точность нагрева заготовок при минимальном значении времени процесса.

При помощи численной ANSYS-модели исследованы поля термонапряжений, возникающих в результате неравномерного нагрева заготовок по сечению. Проведено моделирование и исследование наиболее опасных растягивающих радиальных термонапряжений. В качестве примера рассмотрено поведение полей термонапряжений в процессе оптимального управления при частоте вращения 50 об/с, результаты которого иллюстрируют необходимость учета технологических ограничений уже на этапе формулировки задачи оптимального управления, что позволит обеспечить требуемое качество нагрева при максимальной производительности технологического комплекса.

Литература

1. Рапопорт Э.Я. Оптимизация процессов индукционного нагрева металла. – М.: Металлургия. – 1993. – 279 с.
2. Труды Международного симпозиума «Heating by electromagnetic sources» HES-07, Padua, Italy, Июнь 19–22. – 2007.

ИССЛЕДОВАНИЕ И МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКИХ ПРОЦЕССОВ В ГЛАВНОМ ПРИВОДЕ РАБОЧЕГО РОЛЬГАНГА

А.А. Пушкин

(Донбасская государственная машиностроительная академия)

Научный руководитель – к.т.н., доцент Н.А. Задорожний

(Донбасская государственная машиностроительная академия)

В работе определены наилучшие соотношения параметров регуляторов тока и скорости главного привода рабочего рольганга, реализованного на принципе подчиненного регулирования, предложены способы достижения данных значений.

Ключевые слова: электромеханическая система, демпфирование, упругие связи, математическое моделирование, переходные процессы

Введение

Металлургия является ведущей отраслью промышленности Украины, а, следовательно, производительность данной отрасли и качество продукции в значительной степени определяют состояние и темпы развития экономики нашего государства. Прокатное производство – одна из важнейших составляющих металлургической промышленности. В современных электроприводах прокатных станов большое внимание уделяется надежности, производительности и энергопотреблению электрооборудования. Надежность прокатных станов во многом определяется значительными динамическими нагрузками колебательного характера, которые возникают из-за наличия упругих механических звеньев [1]. Подавление упругих колебаний, возникающих в электроприводах при пуске и ударных нагрузках, существенно увеличит срок службы передаточных устройств, вызовет снижение динамических нагрузок, что приведет к уменьшению потребления энергии на бесполезные колебательные движения, а также уменьшит тепловые нагрузки двигателя и управляемого преобразователя.

Анализ современного состояния вопроса

Условия эксплуатации прокатных станов предъявляют ряд требований к переходным процессам по колебательности, времени регулирования, перерегулированию, динамическому падению скорости, статическому падению скорости и т.д. В связи с этим возникла проблема демпфирования упругих колебаний путем изменения параметров системы и подбора соответствующих регуляторов. Выявление наиболее рациональных методов достижения минимума колебательности даст возможность с минимальными затратами улучшить качество переходных процессов и тем самым продлить срок службы оборудования. Наиболее распространенными способами гашения упругих механических колебаний являются.

1. Технологический, за счет управления технологическим процессом [2].
2. Конструктивный, основанный на выборе оптимальных конструктивных параметров ЭМС в процессе проектирования приводов.
3. Использования дополнительных демпфирующих устройств: гидравлических, пневматических, механических, на основе пар трения [3].
4. Электротехнический, который в свою очередь подразделяется на:

а) электромеханический способ, который основан на демпфировании самим двигателем [4, 5];

б) электродинамический способ, который основан на свойствах вихревых токов.

Наиболее перспективным и рациональным способом демпфирования упругих колебаний является электротехнический, поскольку он сравнительно просто реализуется, легко может использоваться для любой электромеханической системы и не требует никаких дополнительных материальных затрат.

Для анализа электромеханической системы (ЭМС) зачастую применяют обобщенные оценки влияния упругих звеньев механической передачи, которые имеют различную степень влияния на демпфирующую способность электропривода и определенные границы существенности влияния вышеуказанных параметров на подавление упругих механических колебаний [3–5].

В работах [5] произведено исследование влияния динамических нагрузок на режимы работы прокатного оборудования, на основе моделирования переходных процессов в среде MATLAB Simulink установлены основные негативные факторы их воздействия. Таким образом, для улучшения режимов работы прокатного оборудования подавление упругих механических колебаний является очень важным аспектом.

Принцип подчиненного регулирования – один из самых распространенных принципов, используемых при построении систем автоматического управления.

Постановка задачи исследования

Целью работы является исследование и математическое моделирование динамических процессов в главном приводе рабочего рольганга с системой подчиненного регулирования (СПР), функциональная схема которого приведена на рис. 1, и расчет параметров данной системы, необходимых для минимизации колебательности в системе.

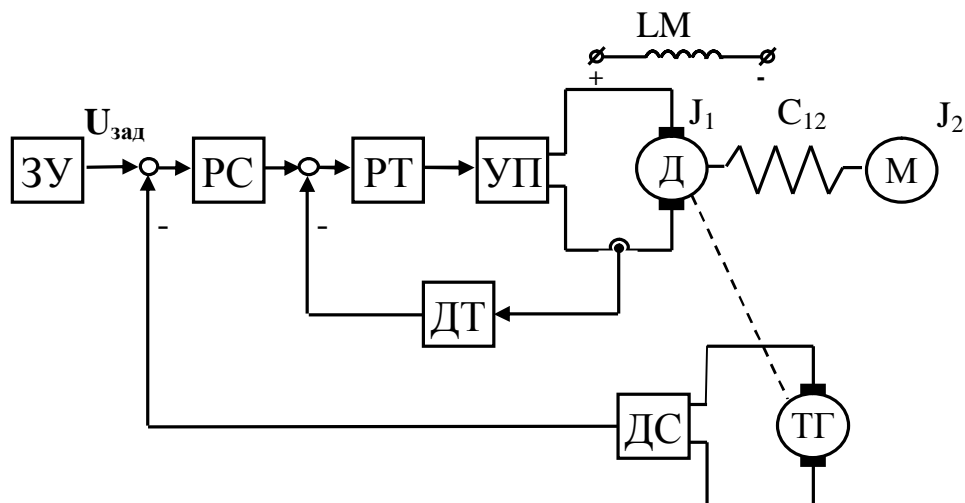


Рис. 1. Функциональная схема главного привода рабочего рольганга

На рис. 1 приняты следующие обозначения: ЗУ – задающее устройство; Д – двигатель постоянного тока с моментом инерции ротора J_1 ; УП – управляемый преобразователь; РТ – регулятор тока; РС – регулятор скорости; ДТ – датчик тока в цепи обратной связи по току; ДС – датчик скорости в цепи обратной связи по скорости; ТГ – тахогенератор для преобразования угловой скорости двигателя в пропорциональное ей напряжение; C_{12} – жесткость упругого звена; М – механическая часть с приведенным моментом инерции вращающихся частей J_2 .

Материалы исследований

Параметры системы, полученные из каталожных данных, приведены в табл. 1.

Таблица 1. Параметры главного привода рабочего рольганга

№	Наименование параметров	Ед. изм.	Значение
Электродвигатель			
1	Номинальная мощность двигателя, P_n	кВт	200
2	Номинальное напряжение на зажимах якоря двигателя, U_n	В	440
3	Номинальный ток якоря двигателя, I_n	А	788
4	Номинальная частота вращения двигателя, n_n	об/мин	500
6	Момент инерции двигателя, J_1	кг·м ²	46,5
7	Электромагнитная постоянная времени, T_3	с	0,0175
Тиристорный преобразователь			
1	Номинальный ток преобразователя, $I_{дн}$	А	800
2	Действующее значение фазной ЭДС, $E_{2\phi}$	В	600
3	Номинальное напряжение преобразователя, $U_{дн}$	В	600
Система автоматического регулирования			
1	Некомпенсированная постоянная времени контура тока, $T_{\mu i}$	с	0,005
2	Постоянная времени фильтра в цепи ОС по скорости, T_ϕ	с	0,01
3	Напряжение источника питания элементов САР, $U_{пит}$	В	12

На основании функциональной схемы, изобразим структурную схему (рис. 2).

Предварительно охарактеризуем постоянные времени контуров тока и скорости.

а) Постоянная времени интегрирования разомкнутого контура тока:

$$T_i = VT_{\mu i}, \quad (1)$$

где $V=2$, согласно модульному оптимуму; $T_{\mu i}$ – суммарная некомпенсируемая малая постоянная времени контура тока:

$$T_{\mu i} = \tau_b + \tau_\phi + T_{oy} + T_{дт}, \quad (2)$$

где τ_b – постоянная времени, обусловленная инерционностью силовой части управляемого выпрямителя; τ_ϕ – постоянная времени, обусловленная инерционностью фильтра на входе системы импульсно-фазового управления выпрямителем; T_{oy} – постоянная времени аналоговых элементов контура тока; $T_{дт}$ – постоянная времени, обусловленная инерционностью датчика тока в цепи обратной связи по току.

б) Постоянная времени интегрирования разомкнутого контура скорости:

$$T_c = VT_{\mu c}; \quad (3)$$

$$T_{\mu c} = 2T_{\mu i} + T_{\phi c} + T_{oy}, \quad (4)$$

где $T_{\phi c}$ – постоянная времени фильтра на выходе датчика скорости; T_{oy} – постоянная времени аналоговых элементов входящих в контур скорости.

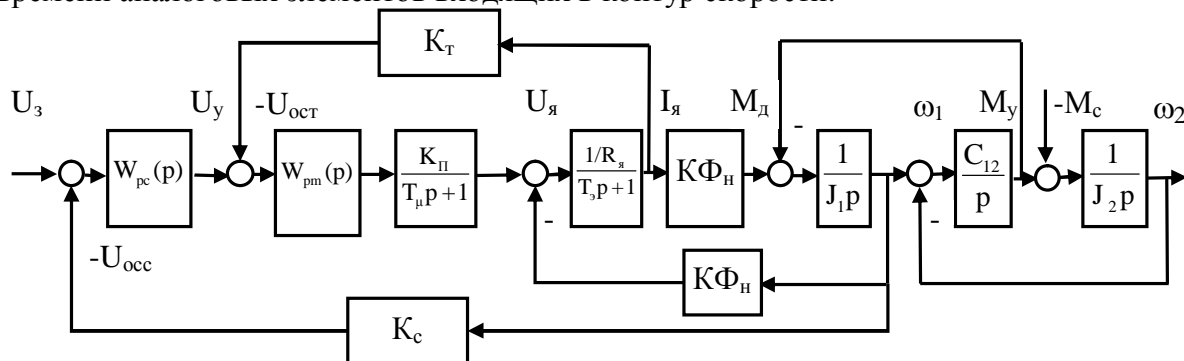


Рис. 2. Структурная схема главного привода рабочего рольганга

На рис. 2 приняты следующие обозначения: ω_1 – скорость двигателя; ω_2 – скорость рабочего рольганга; U_3 – напряжения задания; $I_я$ – ток якоря двигателя; $M_д$ – момент развиваемый на валу двигателя; $M_с$ – момент сопротивления на валу двигателя; M_y – момент упругих колебаний; $T_э$ – электромагнитная постоянная времени; $K_с$ – коэффициент обратной связи по скорости; $K_т$ – коэффициент обратной связи по току; $K_п$ – коэффициент передачи преобразователя; $KФ_н$ – коэффициент пропорциональности ЭДС скорости ω_1 ; C_{12} – приведенный коэффициент жесткости; $T_μ$ – некомпенсированная постоянная времени контура тока; $R_я$ – суммарное сопротивление якорной цепи двигателя.

Передачные функции регуляторов:

Тока: $W_{рт}(p) = (T_э p + 1) / (T_т K_п K_т (1/R_я) p)$; скорости: $W_{рс}(p) = T_{м1} \gamma / (T_с (R_я / KФ_н) K_с (1/K_т))$.

Выполнив преобразование структурной схемы, получаем передачные функции:

$$W_1(p) = \frac{M_y(p)}{M_c(p)} = \frac{D(p)}{Q(p)}; \quad W_2(p) = \frac{I_я(p)}{M_c(p)} = \frac{C(p)}{Q(p)}; \quad W_3(p) = \frac{\omega_2(p)}{M_c(p)} = \frac{R(p)}{Q(p)}.$$

Демпфирующие свойства электропривода определяются знаменателем передаточной функции – характеристическим уравнением, которое имеет вид:

$$Q(p) = \frac{T_с T_т}{\Omega_{12}^2} p^4 + \frac{T_с}{\Omega_{12}^2} p^3 + \frac{\gamma + T_с T_т \Omega_{12}^2}{\Omega_{12}^2} p^2 + T_с p + 1 = 0. \quad (5)$$

Для выявления закономерностей электромеханического демпфирования более удобно пользоваться нормированием характеристического уравнения по обобщенным показателям (6), что дает возможность определить наилучшие с точки зрения минимизации колебательности параметры ЭМС [4]:

$$Q(p) = \gamma K_в T_y^4 p^4 + 2\gamma \sqrt{K_в} \xi_d T_y^3 p^3 + \gamma(1 + K_в) + 2\gamma \sqrt{K_в} \xi_d T_y p + 1 = 0; \quad (6)$$

$$K_в^* = 1; \quad \xi_d^* = \sqrt{\gamma - 1}, \quad (7)$$

где $K_в^*$ – коэффициент электромеханического взаимодействия; ξ_d^* – коэффициент демпфирования.

$$\xi_э = \xi_м = \frac{\sqrt{\gamma - 1}}{2}, \quad (8)$$

где $\xi_э$, $\xi_м$ – коэффициенты демпфирования колебаний в электрической и механической подсистемах соответственно.

Перейдем от обобщенных параметров к постоянным времени привода:

$$T_т^* = \sqrt{K_в^*} \frac{T_y}{2\xi_d^*} = \frac{1}{2\sqrt{\gamma - 1}\Omega_{12}}; \quad T_с^* = 2\sqrt{K_в^*} \xi_d^* T_y = \frac{2\sqrt{\gamma - 1}}{\Omega_{12}}. \quad (9)$$

По виду полученных соотношений можно сделать важные выводы о свойствах ЭМС: при выполнении условий (9) достигается предельное значение демпфирования упругих колебаний в полной ЭМС, улучшается качество переходных процессов; демпфирующие свойства ЭМС, зависят не только от значений постоянных времени, но и от их отношений; зависимости (9) представляют собой обобщенные требования к конструкции упругой механической передачи машин с минимальной колебательностью процессов и учетом взаимосвязи электрических, механических и технологических параметров; из соотношений (2) и (4) видно, что достижение необходимых значений постоянных времени (9) возможно за счет соответствующего выбора выпрямителей, фильтров, аналоговых элементов, датчиков тока и скорости на этапе проектирования СПР, а также подбором коэффициентов передачи регуляторов.

Составим математическую модель главного привода рабочего рольганга, реализованную в среде MATLAB Simulink (рис. 3.). На её основе определим показатели качества ЭМС до и после оптимизации, сравним их с требованиями, предъявляемыми

технологическим процессом. Графики переходных процессов по основным координатам электропривода приведены на рис. 4–5. Полученные результаты сведем в табл. 2.

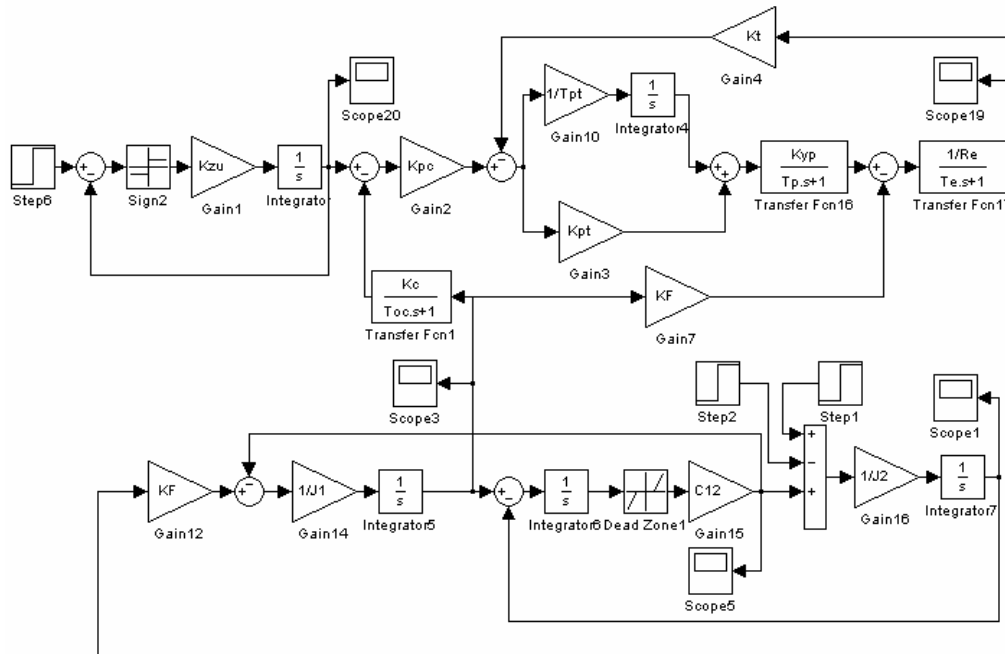


Рис. 3. Математическая модель главного привода рабочего рольганга в среде MATLAB Simulink

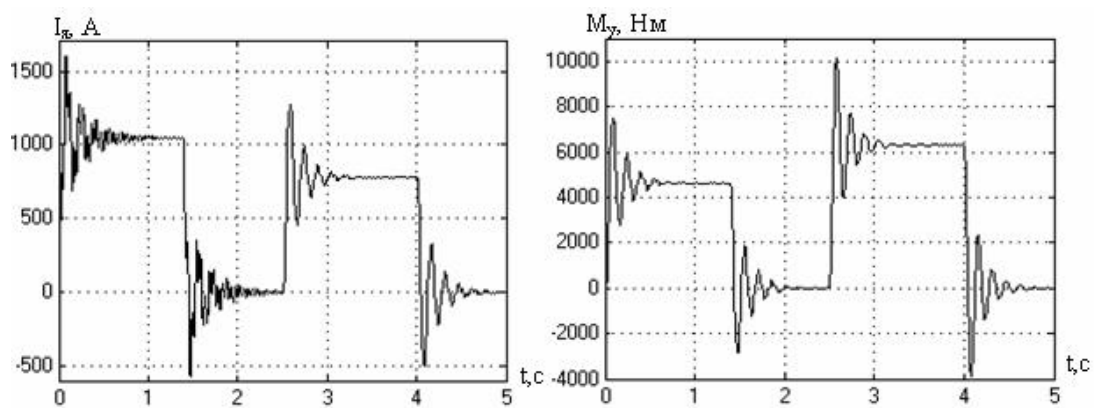


Рис. 4. Переходные процессы в ЭМС при стандартных настройках

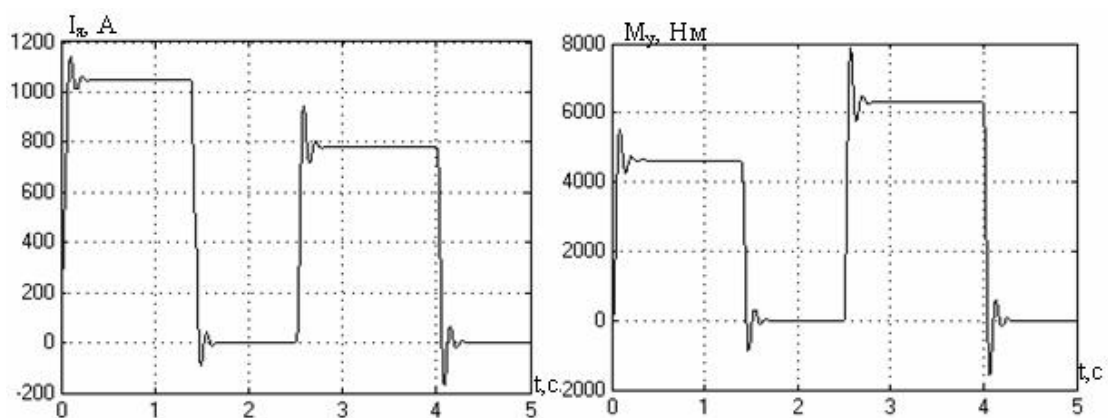


Рис. 5. Переходные процессы в ЭМС при настройках на минимум колебательности

Таблица 2. Динамические показатели качества работы главного привода рабочего рольганга

Показатели качества	λ	η	μ	σ , %	$t_{пр}$, с	$\Delta\omega_c$, %	$\Delta\omega_d$, %	$\Delta\omega_c$, %	$\Delta\omega_d$, %	K_d
Требуемые	3,14	19,14	2	34	0,3	3,5	7	4	8	1,5
Реальные	1,8	11,41	3,49	40,7	0,343	1,9	4,9	1,92	5,25	1,8
После усовершенствования	4,11	34,39	1,53	12,8	0,114	2,5	6,0	2,5	6,1	1,2

Из табл. 2 видно, что многие реальные показатели качества работы главного привода рабочего рольганга до усовершенствования не удовлетворяют требованиям, предъявляемым технологическим процессом, соответственно в результате усовершенствования СПР – удовлетворяют.

Выводы

1) Определили необходимые соотношения параметров регуляторов тока и скорости главного привода рабочего рольганга, реализованного на принципе подчиненного регулирования, предложили способы достижения данных значений.

2) Составили математическую модель главного привода рабочего рольганга и, на её основе определили показатели качества работы электромеханической системы до и после усовершенствования, сравнили их с требованиями, предъявляемыми технологическим процессом и установили:

- реальная СПР не удовлетворяет требованиям и требует усовершенствования;
- усовершенствованная СПР удовлетворяет всем динамическим показателям качества работы главного привода рабочего рольганга.

3) Данный способ может быть использован для электроприводов с целью энергосбережения и выполнения требований технологического процесса на основе демпфирования упругих колебаний.

Литература

1. Бычков В.П. Электропривод и автоматизация металлургического производства. – М: Высшая школа. – 1977 – 392 с.
2. Смирнов В.В. Механика привода прокатных станов/ В.В. Смирнов, Р.А. Яковлев. – М.: Металлургия. – 1977. – 216 с.
3. Задорожний Н.А. Элементы теории электромеханического взаимодействия в двухмассовых системах электропривода с упругими механическими связями: Учебное пособие по дисциплине «Теория электропривода» для студентов специальности «Электромеханические системы автоматизации и электропривод» дневной формы обучения. – Краматорск: ДГМА. – 2006. – 72 с.
4. Пушкин А.А. Определение путей достижения минимума колебательности в электроприводе с упругими связями. – Электротехника и электромеханика: материалы Всеукраинской научно-технической конференции студентов, аспирантов, молодых ученых с международным участием. – Николаев: НУК. – 2007 – С. 77–82.
5. Пушкин А.А. Оптимизация динамических режимов в электроприводах переменного тока при учете упругих механических связей // Вісник кафедри «Електротехніка» за підсумками наукової діяльності студентів. – Донецьк, ДонНТУ. – 2008 – С. 106–107.

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ИМИТАЦИОННОЙ ПОВЕРКИ ПОГРУЖНЫХ ВИХРЕВЫХ РАСХОДОМЕРОВ

А.С. Фролов

(Сибирский государственный технологический университет)

Научный руководитель – д.т.н., профессор М.С. Лурье

(Сибирский государственный технологический университет)

В данной работе рассматривается имитационный способ поверки вихревых расходомеров на специальных имитационных стендах. Рассмотрена конструкция стенда и его математическая модель. Показана возможность реализации имитационной модели стенда в программе Matlab, используя пакет Simulink.

Ключевые слова: погружной расходомер, водосчетчик, поверка, проливная установка, имитационный стенд, герметичная камера, модель

Поверка погружных вихревых расходомеров

В промышленности и коммунальном хозяйстве, существует потребность достоверного измерения расхода воды или различных растворов. В последнее время в России появился новый класс вихревых расходомеров и водосчетчиков, называемый погружными кондуктометрическими. В них, тело обтекания (ТО) устанавливается непосредственно в рабочий трубопровод. Приемник-преобразователь вихревых колебаний (ППВК) помещается в специальном канале тела обтекания, который называется каналом обратной связи (КОС). Он представляет собой два электрода, один из которых колеблется в такт с частотой вихреобразования, при этом частота изменения электрического сопротивления межэлектродного промежутка является выходным сигналом несущим информацию о расходе. Для поверки таких водосчетчиков и расходомеров существуют два метода поверки: проливной и беспроливный (имитационный) метод.

Проливной метод осуществляется на соответствующей проливной установке – это устройство (эталон), предназначенное для поверки и регулировки счетчиков жидкости.

Достоинством проливных установок является возможность поверки на них расходомеров и водосчетчиков практически любых конструкций. В большинстве случаев они позволяют испытывать приборы при «нормальных» условиях: измеряемая среда – вода, оссимметричный стационарный поток, трубопровод протяженный, отсутствие сторонних помех и т.п.

Недостатком проливных установок, является то, что их недостаточно в России, в особенности на диаметры условного прохода более 150 мм и как следствие этого они не могут обеспечить своевременной поверкой все существующие расходомеры. Кроме того, обслуживание проливной установки процесс очень дорогостоящий, трудоемкий, установке требуется отдельное помещение, что весьма удорожает поверку приборов.

В настоящее время практика имитационной периодической поверки преобразователей расхода широко распространена но, к сожалению, не подходит ко всем существующим типам расходомеров.

Беспроливный (имитационный) метод поверки осуществляется на специальных компактных имитационных стендах. Для вихревых кондуктометрических расходомеров, суть этого метода поверки заключается обычно в электрическом моделировании напряжения на электродах первичного преобразователя расходомера, являющегося, носителем информации о величине расхода [1].

Так же условия испытания на существующих имитационных стендах весьма далеки от рабочих. Обычно они ограничиваются подачей на вход прибора заранее

заданных электрических сигналов и наблюдения за реакцией на выходе, т.е. эти методики позволяют проверить только электрический тракт прибора при сигналах, форма которых может быть далека от реального [2].

Имитационная поверка вихревых расходомеров

Качество имитационной поверки можно повысить, если создать стенды, которые будут имитировать не электрические сигналы, а реальные гидродинамические явления, происходящие в расходомере. Стенд, который соответствует этому принципу работы, изображен на рис. 1. Он представляет собой цилиндрическую камеру 1, торцевые стенки которой представляют собой мембраны 2 и 3. Камера заполняется жидкостью через патрубки 8 и 9. Через пробку 10 из нее выпускается остаточный воздух. Поперек камеры устанавливается перегородка 4, в которую вставляется тело обтекания испытываемого расходомера 5. Боковые стенки тела обтекания с отверстиями канала обратной связи (КОС) должны быть направлены параллельно перегородке [3].

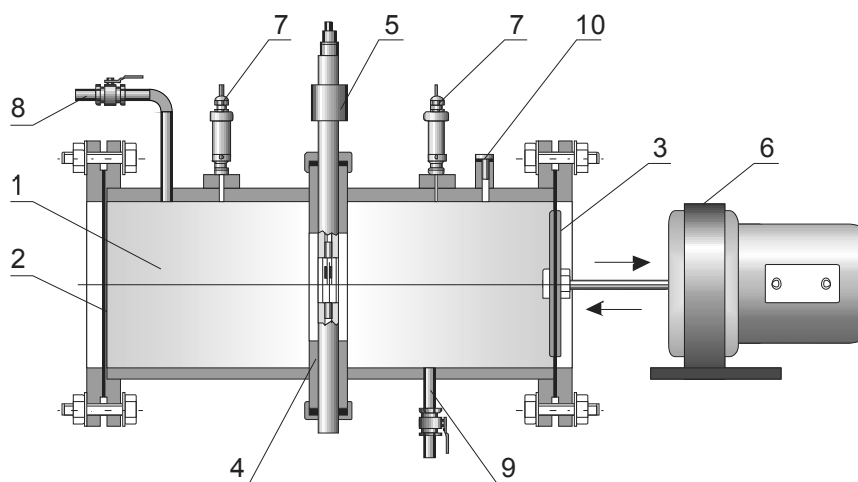


Рис.1. Стенд для испытаний погружных вихревых расходомеров

Одна из мембранных стенок камеры снабжена жестким центром 3, приводится в колебательное движение специальным электромагнитным приводом 6. При этом на торцевые стенки ТО, т.е. на срез канала обратной связи ТО будет действовать знакопеременный перепад давления, имитирующий вихреобразование. Камеру снабжает датчиком перепада давления на перегородке. Им служат два измерительных преобразователя давления SITRANS P (фирмы Siemens) 7, включенные по дифференциальной схеме.

Задачей стенда является создание на срезе КОС (боковых поверхностях тела обтекания) перепада давления по величине и форме соответствующему реальному, который действует в вихревом расходомере. При этом внутри КОС установится знакопеременный переток жидкости по форме и величине также соответствующий реальному. Таким образом, на стенде будет имитироваться реальные гидродинамические процессы, протекающие в КОС и взаимодействие их с чувствительным элементом расходомера – кондуктометрическим ППВК.

Для решения данной задачи на вход системы управления приводом стенда охваченного отрицательной обратной связью по давлению подается сигнал, воспроизводящий изменение перепада давления на срезе КОС. Но сформировать данный сигнал из теоретических предпосылок трудно, а измерить реальный перепад давления на работающем приборе практически невозможно, так как введение в КОС каких либо датчиков, трубок и т.п. искажает гидродинамику и влияет на взаимодействие вихревой формации с ППВК. Поэтому задающий сигнал формируется косвенным образом.

В основе работы стенда лежит допущение о том, что для получения электрического сигнала определенной формы и величины с ППВК необходимо воздействовать на него перепадом давления также строго определенной формы. Если на выходе ППВК будет формироваться сигнал близкий к реально действующему в приборе на заданной скорости потока (то есть, на заданном расходе), то и гидродинамический режим работы КОС и ППВК будут близки к реально существующему.

Однако метод, описанный выше, имеет ряд недостатков, из которых:

- использование датчиков давления увеличивает стоимость стенда;
- применение в имитационном стенде датчиков давления требует их предварительной калибровки и поверки, что усложняет эксплуатацию стенда;
- поскольку перепад давления на чувствительном элементе пропорционален квадрату скорости перетока жидкости через КОС поверяемого прибора, для имитации работы расходомера в реальном диапазоне изменения расхода (до 80), диапазон измерения датчиков давления должен быть до 6400, что сложно осуществить на практике.

Нами разработан стенд, в котором указанные выше недостатки преодолены. Такой стенд изображен на рис. 2.

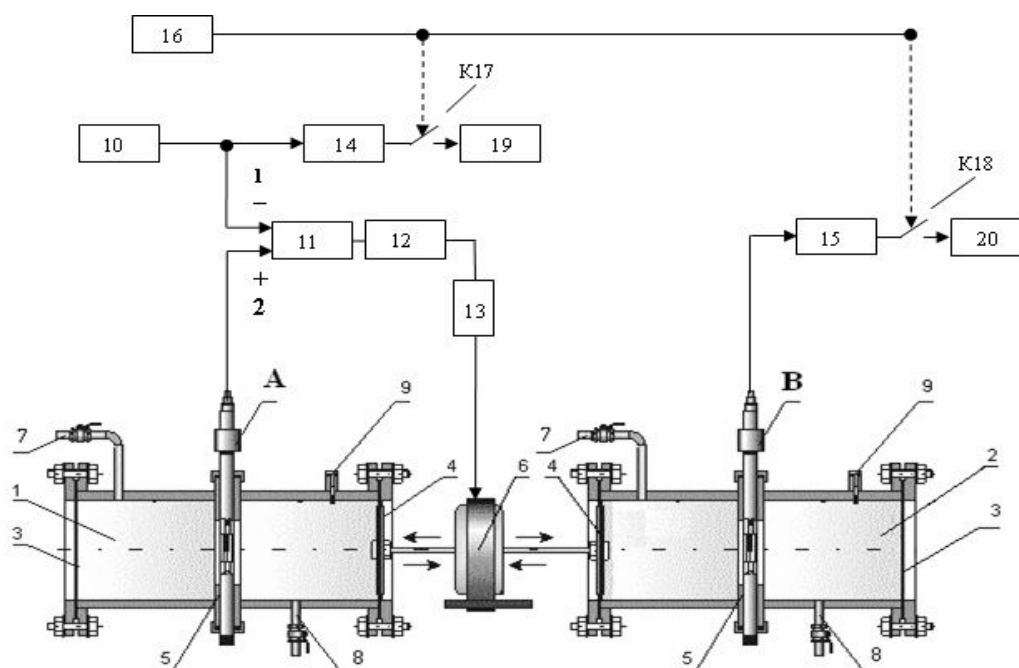


Рис. 2. Модернизированный стенд для испытаний погружных вихревых расходомеров

Стенд состоит из двух идентичных камер, аналогичных показанным на рис. 1, имеющих общий привод, но без датчиков давления. В перегородку 5 левой камеры 1 вставляют эталонный водосчетчик (А) так, чтобы чувствительный элемент располагался параллельно перегородке 5. Таким же образом в герметичную камеру 2 вставляют поверяемый водосчетчик (В). Через патрубки 7 в каждую из камер набирают воду и после выхода остаточного воздуха закрывают пробки 9.

Эталонным сигналом для стенда, который подается с генератора 10, является предварительно записанный на проливной установке электрический сигнал с выхода эталонного водосчетчика на определенном эталонном расходе. Электропривод стенда охвачен отрицательной обратной связью по выходному сигналу эталонного водосчетчика А. В цепь обратной связи входят: блок сравнения сигналов 11; ПИД регулятор 12; усилитель мощности 13; электромагнитный привод 6.

В результате, при работе стенда на мембранах 4 обеих из камер создаются одинаковые механические колебания, амплитуда, форма и частота которых определяются эталонным сигналом. В обеих камерах 1 и 2 создаются знакопеременные перепады давления, возникает знакопеременный переток жидкости в КОС через чувствительные элементы эталонного (А) и поверяемого (В) водосчетчиков. Таким образом, при одинаковых геометрических размерах герметичных камер 1 и 2, в обеих из камер протекают одинаковые гидродинамические явления. Это позволяет считать, что подавая от генератора 10 сигналы, записанные на реальной проливной установке для различных поверяемых расходов, мы будем имитировать на стенде работу водосчетчика в условиях различных расходов.

Управление поверкой осуществляется реле времени 16. Оно задает интервал времени поверки на заданном расходе. Ключи К17 и К18 подключают на время поверки счетчики импульсов 19 и 20 к выходу электронных блоков расходомеров 14 и 15. По окончании интервала времени поверки, счет импульсов прекращается. Сравнивая показания счетчиков 19 (эталонного прибора) и 20 (поверяемого) можно судить о погрешности измерения на данном поверочном расходе.

Для анализа работы и оптимизации характеристик стенда следует создать его математическую модель, а затем на ее основе имитационную модель стенда, например в программе Matlab, используя пакет Simulink.

Для того чтобы описать движение жидкости в герметичных камерах нужно описать движение жидкости в КОС расходомера (1).

Движение жидкости в КОС можно записать в форме, [1]:

$$\frac{dV_{КОС}}{dt} = \frac{1}{a_{II}m_K} (S \cdot \Delta P - F_T) = \frac{S}{a_{II}m_K} (f_1(t) - \text{sign}(V_{КОС}) \xi_{СVM}(\text{Re}) \rho \frac{V_{КОС}^2}{2}), \quad (1)$$

где $V_{КОС}$ – скорость движения жидкости в КОС, м/с; a_{II} – корректирующий коэффициент, учитывающий, что скорость потока изменяется по сечению. Для ламинарного течения $a_{II} = 1,33$, а для турбулентного $a_{II} = 1,03 \div 1,04$; m_K – масса жидкости в контрольном объеме, кг; ΔP – перепад давления, Па; $f_1(t)$ – зависимость перепада давления ΔP от времени; $F_T = \xi_{СVM}(\text{Re}) \rho \frac{V_{КОС}^2}{2}$ – силы сопротивления движению жидкости; S – площадь поперечного сечения КОС, м².

В стенд включен электромагнитный привод создающий силу $F_{ЭМ}$, которая будет действовать на мембрану камеры (2). Для линейной конструкции привода она будет такова

$$F_{ЭМ} = BLI, \quad (2)$$

где B – индукция магнитного поля в зазоре электромагнита привода, Т; L – длина проводника катушки, м; I – ток в катушке электромагнита, А.

Для имитационной модели блок электропривода, в первом приближении, можно задать в виде передаточной функции (3).

$$W_{Эпр}(p) = \frac{K}{Tp + 1}, \quad \text{где } K = \frac{1}{R_K} \cdot BL, \quad \text{а } T = \frac{L_K}{R_K}, \quad (3)$$

где L_K – индуктивность катушки, Гн; R_K – сопротивление катушки, Ом.

Учитывая, что в конструкции камеры две мембраны сила их сопротивления движению электропривода F_M , будет (4)

$$2 \cdot F_M = 2 \cdot K_M \cdot l_{неп} = 2 \cdot \frac{K_M}{S_M} \cdot S \int V_{КОС} \cdot dt, \quad (4)$$

где K_M – коэффициент упругости, Н/м²;

$l_{пер}$ - путь перемещения центра мембраны, м;

S_M - площадь поперечного сечения мембраны, м².

Полученную силу сопротивления мембраны вставим в ранее выведенную формулу для движения жидкости в КОС и получим (5):

$$\begin{aligned} \frac{dV_{КОС}}{dt} &= \frac{1}{a_{II}m_K} (S \cdot \Delta P - F_T - 2F_M) = \\ &= \frac{S}{a_{II}m_K} (f_1(t) - \text{sign}(V_{КОС}) \xi_{СУМ}(\text{Re}) \rho \frac{V_{КОС}^2}{2} - 2 \frac{K_M}{S_M} \int V_{КОС} dt). \end{aligned} \quad (5)$$

Если в качестве входного сигнала камеры стэнда принять напряжение, подаваемое на электромагнитный привод U , а выходной величиной считать скорость движения жидкости в канале обратной связи $V_{КОС}$, то модель одной из камер стэнда в среде Simulink можно представить схемой, показанной на рис. 3.

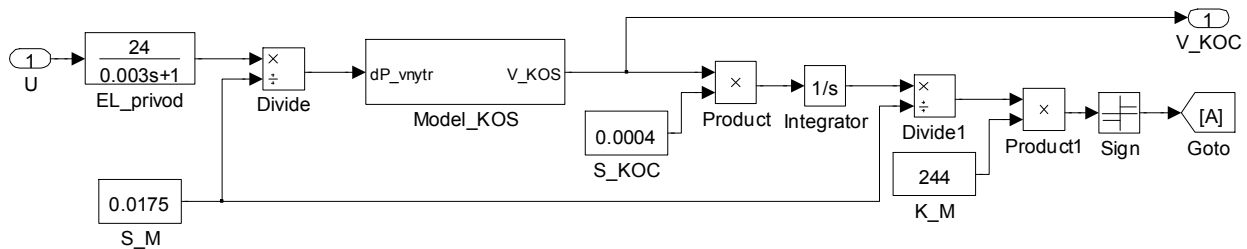


Рис. 3. Имитационная модель одной из камер стэнда

В модели блоки S_M, S_KOC, K_M задают значения S_M , S , K_M , соответственно.

Результаты имитационного моделирования

В модели камеры, использован блок Model_KOS приведенный в работе [1], выполнен в виде маскированной подсхемы с окном параметров, учитывающим все геометрические и гидродинамические характеристики КОС.

Разработанная нами модель камеры имитационного стэнда послужила основой для модели всего стэнда, с учетом обратных связей, показанных на рис. 1.

В качестве примера результатов моделирования, на рис. 4 приведен график скорости потока жидкости в КОС вихревого погружного кондуктометрического расходомера «Фотон» при диаметре условного прохода 80 мм, температуре 22°C и расходе воды 2 м³/с.

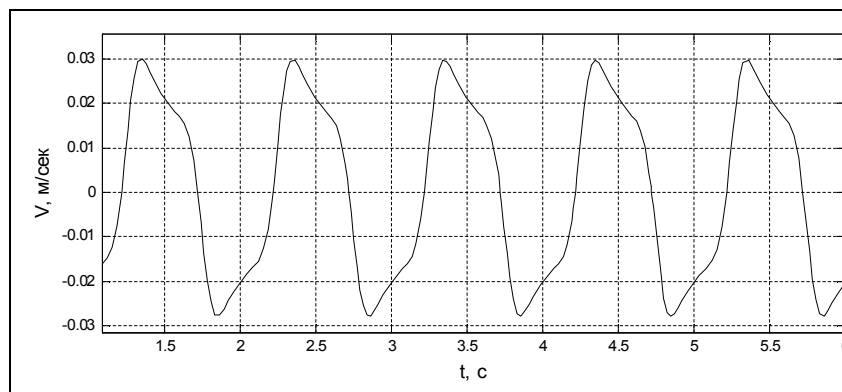


Рис. 4. Скорость движения воды в КОС

Исследование на модели стэнда, позволили оптимизировать характеристики электропривода и настройки ПИД регулятора стэнда в различных условиях.

Разработанная модель может быть использована и для исследования статических и динамических погрешностей стенда в процессе эксплуатации и разработке методов их снижения.

Испытания на стенде серийного вихревого погружного водосчетчика «Фотон», показали хорошее совпадения результатов поверки, в частности зависимость погрешности от расхода, с данными полученными на проливной установке типа МПСР.

Литература

1. Лурье М.С. Имитационное моделирование вихревых расходомеров / М.С. Лурье, С.П. Жуков // Вестник КрасГАУ. – 2005. – №7. – С. 254–258.
2. Лурье М.С. Имитационная поверка погружных кондуктометрических вихревых расходомеров / М.С. Лурье, В.Н. Волынкин, Э.М. Шейнин // Измерительная техника. – 2005. – №12. – С. 38–41.
3. Пат. 2282830 РФ, МПК G 01 F 25/00. Имитационный стенд для поверки вихревых расходомеров, Лурье Михаил Семенович, Плотников Сергей Михайлович, 24.01.2005. Зарегистрирован в Государственном реестре изобретений РФ 27.08.2006

ТЕХНОЛОГИИ ПРОИЗВОДСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

УДК 004.41

УНИВЕРСАЛЬНЫЙ ПАКЕТ ПОДДЕРЖКИ ИНТЕЛЛЕКТУАЛЬНЫХ ВЫЧИСЛЕНИЙ GANS

Д.О. Конончук, Ю.С. Окуловский

(Уральский государственный университет им. А.М. Горького)

Научный руководитель – д.ф.-м.н., профессор В.Ю. Попов

(Уральский государственный университет им. А.М. Горького)

Рассмотрены проблемы реализации интеллектуальных систем – нейронных сетей, генетических алгоритмов и алгоритмов коллективного разума – для решения прикладных задач. Исследованы недостатки существующих реализаций. Представлен универсальный комплекс поддержки указанных типов интеллектуальных вычислений GANS, изложены его архитектурные основания. Выполнен обзор реализаций трех типов интеллектуальных вычислений в комплексе GANS.

Ключевые слова: интеллектуальные системы, нейронные сети, генетические алгоритмы, алгоритмы коллективного разума

Интеллектуальные системы широко применяются в самых разных областях: финансовая аналитика, распознавание изображений, управление робототехническими системами, компьютерные игры, решение сложных вычислительных задач, и многое другое. Под интеллектуальными системами мы понимаем в основном имитационные алгоритмы – нейронные сети, моделирующие нервную систему человека [1], генетические алгоритмы, имитирующие эволюционный процесс [2], и алгоритмы коллективного разума, являющиеся моделью поведения улья [3] или иммунных систем [4]. Для применения интеллектуальных систем следует либо написать исходный код системы самостоятельно, либо воспользоваться одним из готовых решений. Качественная разработка интеллектуальной системы является весьма трудоемким процессом, что связано с объемом кодирования, необходимостью оптимизации и существенными трудностями в отладке интеллектуальных систем. Поэтому самостоятельная разработка интеллектуальных систем часто не оправдывает себя.

Готовые реализации интеллектуальных систем представляют собой шаблоны интеллектуальных алгоритмов, которые требуют небольшой доработки для решения конкретной задачи. Использование существующих готовых решений также вызывает трудности. Не существует ни одной распространенной системы, которая бы поддерживала все три основных типа интеллектуальных вычислений. Наиболее широко представлены реализации нейронных сетей, существенно реже – генетические алгоритмы, крайне редко нейронные системы и генетические алгоритмы совмещены в одном пакете, а для алгоритмов коллективного разума нам не известно ни одной шаблонной реализации. В то же время, совместное использование различных типов интеллектуальных вычислений часто позволяет легко решить задачи, решение которых одним типом затруднительно. Кроме того, для некоторых задач разные типы интеллектуальных систем взаимозаменяемы – так, некоторые задачи распознавания изображений могут быть решены как с помощью нейронных сетей, так и с помощью алгоритмов коллективного разума. Отсутствие пакета, содержащего реализации этих систем, приводит к существенным сложностям в создании гибридных систем, поскольку для этого требуется создание интерфейса между двумя различными программными пакетами, которые могут быть написаны на разных платформах, и

всегда содержат идеологические различия. Кроме того, затрудняется и выбор оптимального типа интеллектуальных вычислений для решения задачи, поскольку для сравнения требуется изучение нескольких различных пакетов. Другими недостатками многих готовых решений являются: закрытая архитектура, которая препятствует внесению в систему нового кода с реализациями нестандартных алгоритмов обучения; сложности в интеграции, в результате которых осложняется использование интеллектуальные системы как составной части разрабатываемого программного продукта; закрытые исходные коды и платные реализации.

Мы представляем пакет GANS (Genetic algorithms and neural systems), свободный от перечисленных недостатков. Пакет содержит общую архитектуру и реализации на ее основе нейронных сетей, генетических алгоритмов и алгоритмов коллективного разума. За счет особенностей архитектуры достигаются широкие возможности внесения в пакет дополнительного кода с реализациями нестандартных алгоритмов, а также реализации новых типов интеллектуальных систем. Реализация пакета как библиотеки .NET существенно упрощает встраивание интеллектуальных систем в создаваемые приложения. Кроме того, .NET обеспечивает кросс-платформенность пакета, а также возможность работы с ним на разных языках программирования. Пакет распространяется с открытым исходным кодом по открытому лицензионному соглашению (GPL) GNU, являясь, в упрощенном понимании, программным обеспечением, свободным для некоммерческого использования. GANS разрабатывался в течение более чем четырех лет, использовался в решение множества прикладных задач, и на сегодняшний момент представляет собой готовый к использованию программный продукт. В настоящей работе мы изложим общую архитектуру GANS и некоторые особенности реализации интеллектуальных систем. Полное описание пакета и его функций доступно в [5].

Компонентная модель GANS

Основной архитектурной особенностью GANS является построение интеллектуальных систем в виде дерева компонент. Компонентом является, например, нейронная сеть, алгоритм ее обучения или целевая функция. Сборка из компонент позволяет обеспечить локальность разработки: так, для разработки нового алгоритма обучения нужно закодировать лишь этот алгоритм, после чего можно интегрировать его с имеющимися компонентами нейронной сети и целевой функции. Кроме того, облегчается документирование систем: библиотека фактически представляет собой набор компонент, к каждому из которых пишется локальная документация, а сборка системы из этих компонент происходит по известным правилам.

Также оказывается возможной разработка специального языка описания интеллектуальных систем, который бы описывал дерево компонент. Такой язык может иметь простую структуру, в частности, являться языком разметки, поскольку теговые языки хорошо подходят для описания древовидных структур. Последующая компиляция описания порождает исходный код на .NET языке, запуск которого создает и сохраняет интеллектуальную систему в файл. Этот подход был реализован на базе языка Thorn и системы кодогенерации Thornado [6].

Компонентный подход также позволяет обеспечить распараллеливание процесса обучения. Компонента может выполнять действия самостоятельно, а может вместо этого рассылать задачи узлам вычислительной сети, собирать от них информацию и представлять как результат собственных вычислений. При этом система в целом не знает, что какая-то часть ее является параллельной, и остальные компоненты работают без изменений [7].

Также разработан драйвер обучения, который обеспечивает автоматическое сохранение системы, ведение и последующий просмотр лога обучения, и позволяет легко добавлять визуализацию процесса обучения. Драйвер работает с произвольными деревьями компонент, и может быть адаптирован под вычисления, не связанные с искусственным интеллектом, в частности, переборным алгоритмам.

Нейронные системы

Нейронная система представляет собой интеллектуальную систему на базе нейронной сети. В системе GANS подобные системы представляются следующим набором компонент.

- Нейронная сеть, поведение которой определяется фиксированными параметрами, такими, как топология или базисная функция, и адаптивным параметром – вектором весов, которые изменяются для обучения сети.

- Алгоритм обучения сети, вычисляющий для обучающего примера вектор направления, т.е. вектор, на который должны быть изменены веса нейронной сети для обучения.

- Целевая функция, т.е. мера ошибки нейронной сети на обучающем примере.

- Генератор примеров, который создает обучающие примеры для нейронной системы. Генератор используется для интеграции нейронной системы с решаемой задачей. В частности, обучающие примеры могут быть получены из изображения, снятого с видеокамеры, из текстовых файлов, и так далее. Соответственно, необходимо программировать новый генератор примеров для каждого нового источника обучающей информации, чтобы преобразовывать эту информацию в векторный вид. Также разработаны универсальные генераторы примеров, которые принимают уже подготовленные векторы из текстового файла или консоли. Такие генераторы позволяют избежать кодирования, однако требуют предварительной подготовки обучающих векторов вручную.

Основной проблемой реализации нейронных сетей является многообразие различных топологий сетей и методов их обучения. В рамках компонентной модели нами предложен способ декомпозиции топологии сети и алгоритмов на более простые. Нейронная сеть и алгоритмы обучения в этом подходе являются составными сущностями, и также построены из компонент. Компонента нейронной сети представляет собой некую подсеть с заданным количеством входов и выходов. Элементарной компонентой является слой нейронов одного вида. Компоненты могут связываться различными способами. На рисунке изображены примеры последовательного, параллельного и рекуррентного соединения компонент. Данные операции позволяют получить из слоев самые разные архитектуры нейронных сетей: многослойный персептрон, сеть прямого распространения сигнала, составленную из разных типов нейронов, а также рекуррентные сети (например, сети Эльмана или RTRN [1]). Для всех построенных сетей будет автоматически реализован метод обратного распространения ошибки, являющийся важным методом обучения нейронных сетей.

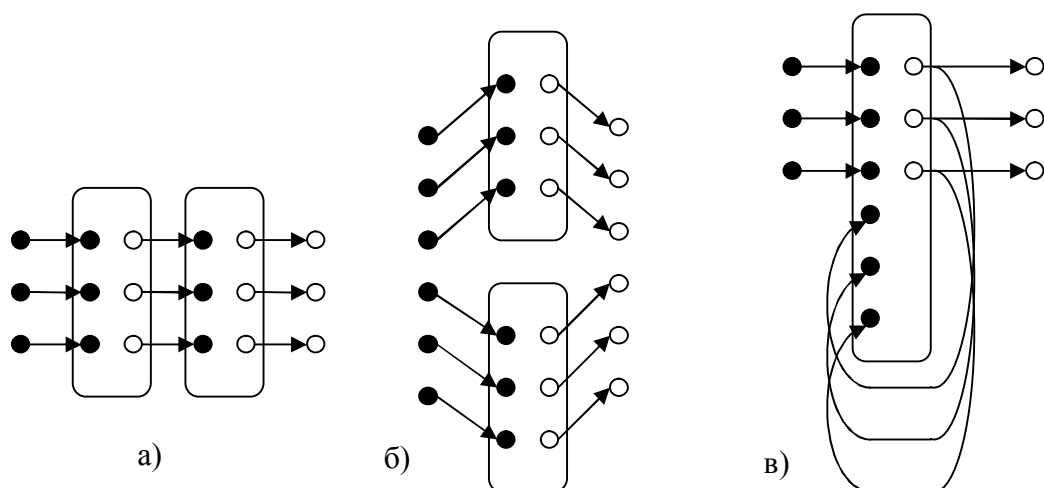


Рисунок. Различные типы соединения компонент нейронных сетей: а) последовательное соединение; б) параллельное соединение; в) рекуррентное замыкание компоненты

Схожий подход применяется для декомпозиции алгоритмов. Каждый из алгоритмов принимает направление изменение весов и уточняет его. Алгоритм, являющийся последовательной композицией двух алгоритмов, подает принятый вектор на вход первому алгоритму. Вектор уточняется первым алгоритмом и поступает на вход второму алгоритму для аналогичной процедуры. В параллельной композиции алгоритмы уточняют вектор одновременно, после чего результаты складываются. Вектор, подаваемый самому первому алгоритму в этой цепочке, равен нулю. Предполагается, что первый в цепочке алгоритм должен игнорировать начальное значение вектора, и вместо уточнения определять его. За счет декомпозиции алгоритмов оказывается возможным свести сложные эвристические алгоритмы обучения, такие как RPROP [1], к комбинации простых алгоритмов. Эти и другие особенности реализации нейронных сетей более подробно изложены в [8].

Генетические алгоритмы

Генетические алгоритмы представляют собой модель эволюционного процесса и сводятся к манипулированию генами – кодами, представляющие возможные решения задачи – с целью максимизации функции приспособленности гена, которая является качеством решения. Традиционные описания генетических алгоритмов [2] предполагают, что ген является последовательностью дискретных значений, чаще всего – логического типа. Система GANS не налагает этого ограничения на гены, и геном может считаться фактически любой объект. Это связано с практическими соображениями – было, например, установлено, что применение в качестве генов последовательностей чисел с плавающей точкой хорошо подходит для решения некоторых прикладных задач [9].

Основной проблемой реализации генетических алгоритмов является огромная вариативность методов изменения генов. Компонентный подход позволяет решить эту проблему и создать по-настоящему универсальный шаблон генетического алгоритма, который можно было бы настроить произвольно. Число компонентов в генетическом алгоритме достигает 16 компонентов. Мы опишем основные из них.

- Генератор. Создает новые гены. Гены могут создаваться случайно, или подбираться каким-либо алгоритмом приближенного решения задачи с целью последующей оптимизации генетическим алгоритмом.

- Мутатор. Обеспечивает точечные изменения гена. Изменения могут быть случайными, или являться применениями какого-либо алгоритма оптимизации задачи.
 - Выборщик пар. Выбирает пары для скрещивания, например, по пропорциональной схеме или по схеме элитарного скрещивания.
 - Кроссер. Обеспечивает скрещивание двух генов случайным образом (кроссовер), или детерминировано.
 - Оценщик. Оценивает новые гены на соответствие задачи и вычисляет их функцию приспособленности.
- Чистильщик. Удаляет из пула гены с недостаточным значением оценочной функции.

Другие компоненты, полностью описанные в [5], контролируют количество генов, производимых мутатором, кроссером, генератором, а также выполняют другие вспомогательные функции.

В пакете выполнена реализация генов, представляющие собой массивы какого-либо типа (логического, целочисленного, чисел с плавающей точкой и т.д.). При использовании таких типов, единственным компонентом, требующим кодирования, является оценщик, поскольку для каждой задачи необходимо заново определять соответствие гена задаче. Остальные компоненты стандартные. В случае, если пользователь хочет использовать нестандартные гены (например, деревья, графы, списки переменной длины), ему необходимо написать и компоненты для работы с ними. Также дополнительные компоненты должны быть написаны в случае, если мутация или скрещивание выполняется не случайно, а детерминировано, в зависимости от задачи.

Алгоритмы коллективного разума

Алгоритмы коллективного разума (АКР) представлены двумя основными направлениями: алгоритмами муравейника [3], и искусственными иммунными системами [4]. Подходы имеют отличия, однако воплощают общую идею решения задачи коллективом из множества простых агентов, которые работают в некоем «мире», представляющем задачу, и при этом обладают локальным зрением, т.е. видят не весь мир, а лишь некоторую его часть.

Для реализации мы выделяем в АКР следующие сущности:

1. Карта – весь «мир», который является контейнером для всех остальных объектов.
2. Окрестности – область карты, обзораемая неким существом, предназначенная для использования координат относительно каждого агента, и для ограничения области видимости.
3. Ячейка – элементарная единица карты, к которой привязываются агенты и дополнительные свойства.
4. Агенты – активные, изменяющиеся со временем объекты, собственно выполняющие решение задачи. В муравьиных алгоритмах агентами являются «муравьи» и оставляемые ими феромоны, в моделях иммунных систем – клетки иммунитета. Взаимодействие агентов с окружающей средой локально.
5. Эфир – вспомогательная сущность для широковещательных сообщений агентов, используемая в некоторых моделях иммунной системы.

АКР предполагают использование большого числа агентов, поэтому необходима оптимизация по времени. Оптимизация выполнена за счет делегатов – одного из отличий платформы .NET от других объектно-ориентированных языков, которое представляет собой указатель на функцию, оформленный в соответствии с правилами ООП. Процесс работы АКР выглядит следующим образом. Карта из внешнего мира

получает команду на выполнение одной итерации. Далее последовательно вызываются следующие делегаты.

1. Начало хода.

2. Собственно ход. Этот делегат принимает сведения об окружающем мире: его окрестность, радиус которой задан свойством существа и с центром в самом этом существе; «доску объявлений». Делегат возвращает – специальную структуру, описывающую, каким образом агент хочет изменить окружающий мир: куда передвинуться в локальных координатах его окрестности, какие объекты и куда добавить, какие сообщения добавить в эфир.

3. Конец хода. В этот момент все изменения предыдущего этапа уже применены. Объект на данном этапе может проанализировать сложившуюся в его ячейке ситуацию, и сообщить о том, как он хочет изменить свое состояние:

- обновить требования к окружающему миру (делегаты и радиус окрестности);
- «умереть», т.е. окончательно исчезнуть с карты;
- никак не изменять состояние.

Поскольку действия одного объекта могут существенно влиять на действия других в этом же ходе, была организована система транзакций, благодаря которой агенты сначала вырабатывают желаемые изменения, которые затем применяются. Например: если агент А «убивает» агента Б на некотором ходе, то у агента Б в случае отсутствия системы транзакций не будет возможности совершить ход, однако с этой системой возможно «взаимное» «убийство» агентов. Таким образом, система транзакций моделирует одновременные действия.

Различные типы агентов могут по-разному использовать делегаты, или не использовать их вовсе. Так, феромонный след в муравьиных алгоритмах использует только третью стадию хода, для того, чтобы объединять все феромоны одного типа находящиеся в одной клетке в одно целое, все остальные стадии хода игнорируются.

Приведенная схема работы АКР позволяет реализовать распараллеливание алгоритмов коллективного разума, как на системы с общей памятью, так и на распределенные вычислители. Для реализации первого способа используются распараллеливание вычислений в пределах одной транзакции средствами .NET framework. Для реализации второго предоставлены абстрактные классы «фрагментируемой карты» и «фрагмента карты». С их помощью вся карта разбивается на непересекающиеся фрагменты, которые могут обрабатываться на различных узлах кластера, при этом узлы будут взаимодействовать только для синхронизации небольших пограничных областей.

В настоящее время реализованы следующие объекты и вспомогательные классы.

1. Карты и окрестности для работы в двумерном и трехмерном пространствах, где ячейки связаны в параллельную решетку. Эти карты могут применяться для обработки изображений, их последовательностей или голографических изображений.

2. Карта и окрестность для пространства, связность ячеек которого задается произвольным графом

3. Стандартный класс ячейки и класс эфира, подходящие для большинства задач.

4. Абстрактные классы для упрощения дальнейших разработок: «агент», который соответствует поведению как муравьев в муравьиных алгоритмах, так и клеток в искусственных иммунных системах, и «феромон», использующийся в муравьиных алгоритмах.

Примененная модель алгоритмов коллективного разума успешно применена для задачи поиска углов и границ на изображении. Планируется дальнейшее тестирование алгоритмов коллективного разума на других задачах. Возможно использование реализации алгоритмов коллективного разума для построения моделей компьютерных игр, поскольку поведение игровых единиц и их взаимодействие друг с другом и с

картой полностью идентично агентам в АКР. Алгоритмы коллективного разума также являются моделью описания взаимодействия групп роботов. Особенно это справедливо для нанороботов, которые в силу своих физических характеристик полностью соответствуют агентам АКР: в силу малых размеров они могут выполнять лишь сравнительно простые операции, восприятие мира и сигналов от других роботов осуществляется локально.

Литература

1. Осовский С. Нейронные сети для обработки информации / С. Осовский. М. – Финансы и статистика. – 2002.
2. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, Л. Рутковский, М. Пильинский. – М. Горячая линия – Телеком. – 2004.
3. Dorigo M. Ant Colony Optimization / M. Dorigo, T. Stützle. MIT Press. – 2004
4. Dasrputa D. Immunity-based systems: A survey / D. Dasrputa, N. Ottob-Okine // IEEE International Congerence on Systems, Man and Cybernetics. – 1997.
5. <http://ai.math.usu.ru/wiki/GANS>
6. Деев Д.В. Система кодогенерации Thornado / Деев Д.В., Окуловский Ю.С. // Прикладные аспекты моделирования и разработки систем информационно-аналитической поддержки и принятия решений. – Выпуск 4. – Екатеринбург. – 2008. – С. 209–216.
7. Морнев М.Л. Комплекс распределения интеллектуальных вычислений на кластер под управлением операционной системы Windows / Морнев М.Л., Окуловский Ю.С. // Четвертая сибирская школа-семинар по параллельным и высокопроизводительным вычислениям. Программа и тезисы. – Томск. – 2007. – С. 49–50.
8. Окуловский Ю.С. A model and implementation of universal engine for neural systems. / Ю.С. Окуловский // 9-й международная конференция «Интеллектуальные системы и компьютерные науки», труды конференции, II том. – Москва. – 2006. – С. 21–24.
9. Скрипов П.В. Estimation of thermophysical properties for pulse heated substances. Experiment and modelling. / Скрипов П.В., Смотрицкий П.В., Старостин А.А., Окуловский Ю.С. // Тезисы XXIII международной конференции «Уравнения состояния вещества». – Эльбрус. – 2008. – С. 66–68.

МОДЕЛЬ СЕТЕВОГО ВЗАИМОДЕЙСТВИЯ ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

С.М. Платунова

Научный руководитель – к.т.н., доцент В.А. Костеж

Статья содержит описание модели сетевого взаимодействия прикладного программного обеспечения, установленного в корпоративной компьютерной сети. Описание модели осуществляется на базе систематизации информации принятой в модели OSI. Кроме того, используются элементы теории массового обслуживания. В качестве коммутационного оборудования рассматривается коммутатор 3-го уровня.

Ключевые слова: модель, очереди, система массового обслуживания

Введение

В работе рассматривается модель сетевого взаимодействия прикладного программного обеспечения. Графоаналитическая модель сетевого взаимодействия прикладного программного обеспечения включает две рабочие станции, под управлением сетевых операционных систем с сетевым прикладным программным обеспечением, соединенных через коммутатор второго или третьего уровня посредством UTP. Модель теории массового обслуживания М/М/1 моделирует сеть с коммутацией пакетов одно обслуживающее устройство, марковское или пуассоновское распределение интервалов поступления заявок и марковское распределение значений времени обслуживания. С помощью модели М/М/1 предполагается моделировать сеть с коммутацией пакетов.

Для исследования сетевого взаимодействия используется эталонная модель взаимодействия открытых систем OSI (Open Systems Interconnection) – международную стандартизованную архитектуру, определяющую способ декомпозиции задачи взаимодействия по вертикали [1].

Графоаналитическая модель сетевого взаимодействия прикладного программного обеспечения

Рассмотрим и опишем модель сегмента корпоративной компьютерной сети, в которой организована установка сетевого прикладного программного обеспечения (СППО). Описание модели сегмента корпоративной сети необходимо для дальнейшего исследования характеристик сети.

В работе предполагаем, что сетевая среда взаимодействия сетевого прикладного программного обеспечения корректно функционирует вплоть до 3-го уровня модели OSI. Т.е. взаимодействие между сетевыми интерфейсами, использующими IP-адреса, происходят корректно, что подтверждается успешным завершением сетевых команд (ping и др.). Физическая модель сегмента корпоративной компьютерной сети представлена на рис. 1. В ней показаны рабочие станции под управлением сетевой операционной системы с установленным сетевым программным обеспечением, соединенные неэкранированной витой парой UTP, обеспечивающей передачу со скоростью 100 Мбит/с по технологии Fast Ethernet и соединенных в виртуальную локальную сеть посредством коммутатора третьего уровня.

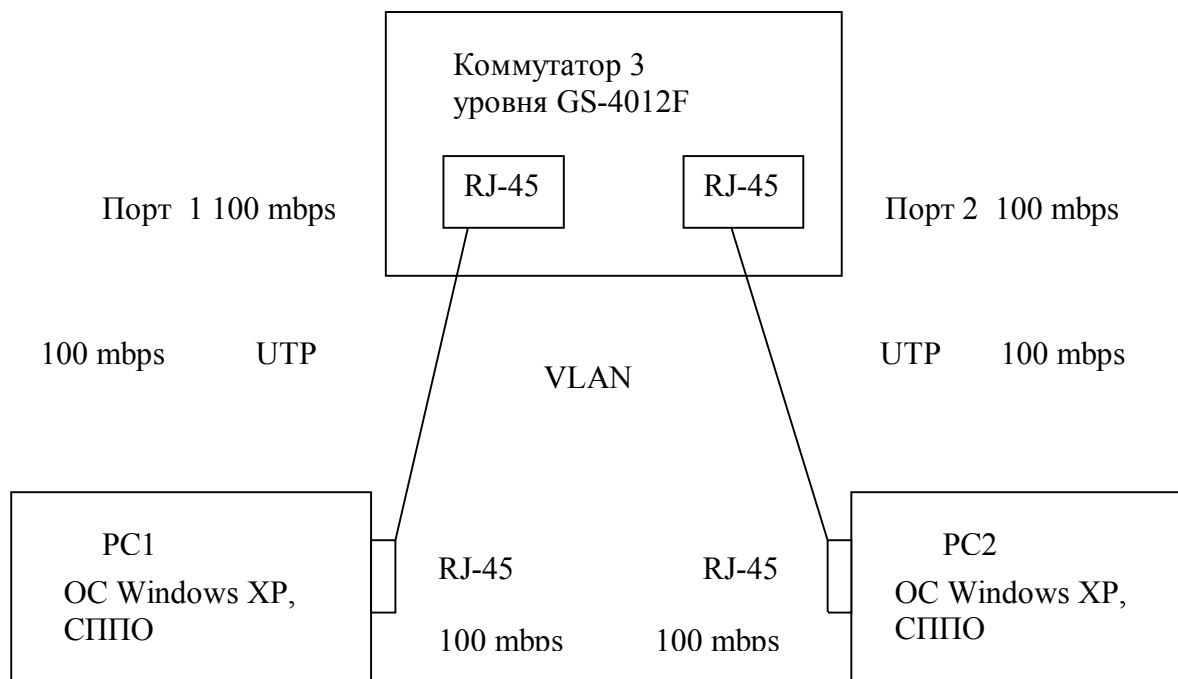


Рис. 1. Физическая модель сегмента корпоративной компьютерной сети

В сегменте корпоративной компьютерной сети могут решаться группы сетевых задач (СППО).

1. Удаленный доступ.
Использование протоколов telnet , SSH и др.
2. Совместное использование файловых систем различных компьютеров сети.
Использование протоколов smb, NFS.
3. Доставка файлов.
Использование протоколов ftp, http.
4. Базы данных.
Технологии клиент-сервер.
5. Прикладные сетевые задачи.
ERP-системы.

Все группы задач могут создавать поток запросов, различных для различных уровней модели OSI, описываемый на основе теории массового обслуживания. Вариантов описания по уровням модели может быть несколько.

Физическая модель сегмента корпоративной компьютерной сети (рис. 1) может быть представлена в виде графа (рис. 2).

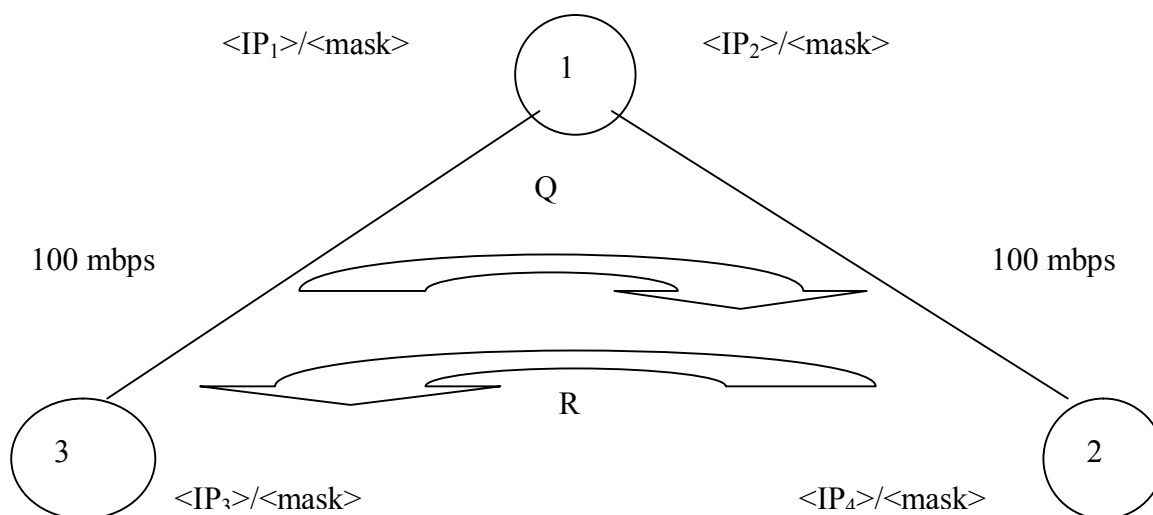


Рис. 2 Графическая модель сегмента корпоративной компьютерной сети: 1 – коммутатор 3-го уровня GS-4012F; 2 – серверное сетевое прикладное программное обеспечение; 3 – клиентское сетевое прикладное программное обеспечение; Q – поток запросов клиентского сетевого прикладного программного обеспечения к серверу; R – поток ответов серверного сетевого прикладного программного обеспечения клиенту

Потоки запросов и ответов носят сложный характер, обусловленный особенностями различных уровней модели OSI и групп сетевых задач.

Поток запросов клиентского сетевого прикладного программного обеспечения может играть роль входного потока. Понятие буфер модели массового обслуживания может носить сложный характер, зависящий от уровня рассмотрения модели OSI и рассматриваемой группы сетевых задач в качестве основной и в качестве фоновой. Понятие обслуживающего устройства модели массового обслуживания конкретизируется в соответствии с уровнями модели OSI.

На физическом и канальном уровнях потоки Ethernet кадров сетевых задач из среды домена коллизий обращаются к сетевому адаптеру на обслуживание и соответствуют входному потоку СМО, сетевой адаптер, обслуживающий кадры, соответствует обслуживающему устройству, а кадры ответов, выдаваемые адаптером в физические каналы, соответствуют выходному потоку.

На транспортном уровне пакеты TSP(UDP)/IP запросов сетевых задач соответствуют запросам на обслуживание, входной интерфейс, имеющий IP адрес, соответствует обслуживающему устройству СМО, выходной интерфейс, имеющий IP адрес, соответствует выходным потоком обслуженных заявок.

На прикладном уровне клиенты сетевых задач сетевого прикладного программного обеспечения являются входным потоком заявок, программы-серверы соответствует обслуживающему устройству, ответы сервера соответствуют выходному потоку СМО.

Соответствующим образом могут быть сформированы характеристики свойств процесса сетевого взаимодействия прикладного программного обеспечения и критерии их оценки.

Графическая модель сегмента корпоративной компьютерной сети может быть представлена в виде еще более абстрактном, представленном на рис. 3.

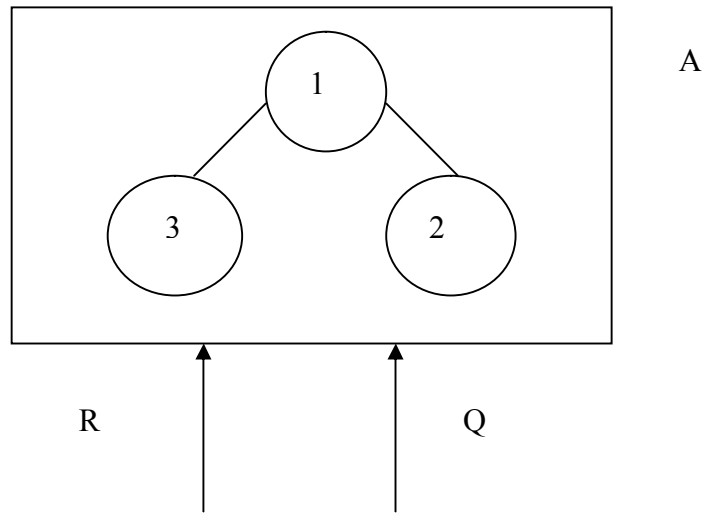


Рис. 3. Модель сегмента корпоративной компьютерной сети: 1 – коммутатор; 2 – серверное сетевое прикладное программное обеспечение; 3 – клиентское сетевое прикладное программное обеспечение; А – структура сегмента корпоративной сети; Q – поток запросов клиентского сетевого прикладного программного обеспечения к серверу; R – поток ответов серверного сетевого прикладного программного обеспечения клиенту

Модель теории массового обслуживания

Модель M/M/1 – наиболее простая модель теории массового обслуживания ТМО [2]. В ней моделируется одно обслуживающее устройство, первая буква M означает марковское распределение интервалов поступления заявок или пуассоновское, вторая буква M означает марковский тип распределения значений времени обслуживания. Однородный стационарный поток без последствий является простейшим, пуассоновским потоком. Пуассоновский поток заявок удобен при решении задач теории массового обслуживания. Простейшие потоки редки на практике, однако многие моделируемые потоки допустимо рассматривать как простейшие.

Рассмотрим модель M/M/1.

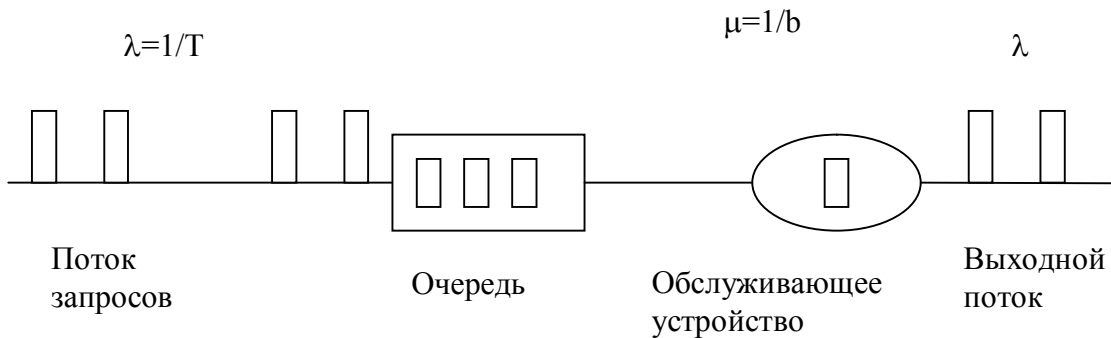


Рис. 4. Модель M/M/1

Основными элементами модели являются:

1. входной поток абстрактных заявок на обслуживание;
2. буфер (очередь);
3. обслуживающее устройство;
4. выходной поток обслуженных заявок.

Заявки поступают на вход буфера в случайные моменты времени. Если в момент поступления заявки буфер пуст и обслуживающее устройство свободно, то заявка

передается в это устройство для обслуживания. Обслуживание также длится случайное время [3].

Если в момент поступления заявки буфер пуст, но обслуживающее устройство занято обслуживанием ранее поступившей заявки, то заявка ожидает его завершения в буфере. Как только обслуживающее устройство завершает обслуживание заявки, она передается на выход, а прибор выбирает из буфера заявку (если буфер не пуст). Выходящие из обслуживающего устройства заявки образуют выходной поток. Буфер считается бесконечным, то есть заявки никогда не теряются из-за того, что исчерпана емкость буфера, прибывшая заявка застаёт буфер не пустым, то она становится в очередь и ожидает обслуживания. Заявки выбираются из очереди в порядке поступления, то есть соблюдается дисциплина обслуживания «первым пришел – первым обслужен (First-In, First-Out, FIFO)». Теория очередей позволяет оценить для этой модели среднюю длину очереди и среднее время ожидания заявки в очереди в зависимости от характеристик входного потока и времени обслуживания.

С помощью модели M/M/1 можно моделировать сеть с коммутацией пакетов. Входной поток заявок играет роль потока пакетов, поступающих на вход интерфейса коммутатора, буфер модели соответствует буферу входного интерфейса коммутатора, а абстрактное обслуживающее устройство моделирует процессор, обрабатывающий пакеты и направляющий их на выходной интерфейс, следовательно, среднее время обслуживания заявки соответствует среднему времени продвижения пакета из входного буфера в выходной канал. Приведённая модель очень упрощенно описывает процессы, происходящие в сети. Она не учитывает многих особенностей. Например, обработки пакетов, конечного размера буфера коммутатора, ненулевого времени поступления пакета в буфер и других. Тем не менее, она очень полезна для основных факторов, влияющих на величину очереди.

Заключение

В работе была рассмотрена и описана схема сетевого взаимодействия прикладного программного обеспечения. Предложена модель для исследования сетевого взаимодействия прикладного программного обеспечения с использованием теории массового обслуживания и модели взаимодействия открытых систем OSI.

Литература

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. 3-е изд. СПб: Питер. – 2007. – 957 с.
2. Клейнрок Л. Вычислительные системы с очередями. – М.: Мир. – 1979. – 600 с.
3. Клейнрок Л. Теория массового обслуживания. – М.: Машиностроение. – 1979. – 432 с.

АНАЛИЗ НЕЯВНОГО ПОТОКА УПРАВЛЕНИЯ И ЕГО ВЛИЯНИЕ НА ВЫБОР СТРУКТУРНЫХ КРИТЕРИЕВ ТЕСТИРОВАНИЯ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ ПРОГРАММ

А.М. Карпушинский

Научный руководитель – к.т.н., доцент Т.А. Павловская

Механизм обработки исключений в современных объектно-ориентированных языках предоставляет возможность писать защищенный и надежный код посредством включения блоков обработки всевозможных исключений и ошибок выполнения программы. Однако, зачастую программы, перегруженные такими конструкциями, становятся сложными для понимания, а также тестирования, так как они вводят новые структурные элементы (неявный поток управления, новые пути в управляющем графе), которые должны быть учтены при выборе критериев структурного тестирования. В работе рассматривается проблема определения класса таких критериев в присутствии неявного потока управления. Ключевые слова: структурное тестирование, обработка исключений, критерии

Введение

Структурное тестирование (проводится по принципу «белого ящика») – процесс выявления ошибок в реализации алгоритмов, основанный на знании внутренней структуры программы. Структурное тестирование применяется при модульном тестировании (на уровне отдельно взятых модулей, функций или классов) и при интеграционном тестировании (тестирование части системы, состоящей из двух и более модулей). Структурное тестирование характеризуется степенью, в которой тесты выполняют или покрывают логику программы (исходный текст). Тестовые наборы строятся на основе анализа *потока управления* и *потока данных*.

Анализ потока управления подразумевает определение для каждой команды *s* в программе набор команд, которые могут выполняться непосредственно за *s*. Многие методы анализа программного кода, такие как исследование потока данных или зависимостей по управлению, а также структурное и регрессионное тестирование, используют информацию о потоке управления. Эти методы предполагают отправной точкой построение *управляющего графа* (УГ) программы, которую необходимо проанализировать или протестировать. Вершины управляющего графа соответствуют либо элементарным операторам, либо базовым блокам, а дуги – возможным передачам управления между блоками (операторами). Базовый блок представляет собой линейную последовательность операторов, такую, что выполнение любого оператора из этой последовательности означает выполнение и всех остальных ее операторов.

Если программа содержит обработку исключений, неявный поток управления, привносимый такими конструкциями, должен быть учтен при построении УГ (представлен явно), а также при составлении критериев структурного тестирования.

Построение иерархии критериев структурного тестирования подразумевает определение *отношения включения*: критерий *C1* включает в себя критерий *C2*, если для всех программ и тестовых наборов выполнение *C1* подразумевает выполнение *C2*. Упорядочив критерии тестирования в соответствии с отношением включения, можно построить иерархию критериев, представляющую относительную «силу» и «слабость» критериев в смысле строгости требований к покрытию элементов программы.

Критерии потока управления и потока данных

Структурные критерии потока управления базируются на основных элементах УГ, операторах, ветвях и путях.

1. Условие критерия тестирования *команд* (критерий C0) – набор тестов в совокупности должен обеспечить прохождение каждой команды не менее одного раза. Это слабый критерий, он, как правило, используется в больших программных системах, где другие критерии применить невозможно.
2. Условие критерия тестирования *ветвей* (критерий C1) – набор тестов в совокупности должен обеспечить прохождение каждой ветви не менее одного раза. Это достаточно сильный и при этом экономичный критерий, поскольку множество ветвей в тестируемом приложении конечно и не так уж велико. Данный критерий часто используется в системах автоматизации тестирования.
3. Условие критерия тестирования *путей* (критерий C2) – набор тестов в совокупности должен обеспечить прохождение каждого пути не менее 1 раза.

Критерии потока данных формулируются с использованием следующих понятий [1].

Определение переменной (definition, def) – это оператор или выражение, меняющее значение переменной. Это может быть оператор присваивания, вызов процедуры или функции.

Использование переменной (use) – появление переменной в правой части оператора присваивания (c-use, computational use) или в составе логического предиката (p-use, predicate use).

Для данного набора тестов определим P как множество полных путей, пройденных на этих тестах. Будем рассматривать пути от определения переменной до ее использования, которые не содержат внутри себя повторных определений переменной.

Выделяют следующие критерии потока данных.

Покрытие определений (all-defs): критерий выполняется, если P включает в себя все пути от каждого определения переменной до какого-либо ее использования.

Покрытие def-use-цепочек (all-uses): критерий выполняется, если P включает в себя все пути до каждого использования переменной от какого-либо определения.

Покрытие def – c-use цепочек (all-c-uses): критерий выполняется, если P включает в себя все пути от каждого определения переменной до ее c-use-использования.

Покрытие def – p-use цепочек (all-p-uses): критерий выполняется, если P включает в себя все пути от каждого определения переменной до ее p-use-использования.

Критерий all-c-uses/some-p-uses: критерий выполняется, если P включает в себя все пути от каждого определения переменной до ее c-use-использования. Если определению переменной не соответствует ни одного c-use-использования, то P должен включать в себя путь до некоторого p-use-использования этой переменной.

Критерий all-p-uses/some-c-uses: критерий выполняется, если P включает в себя все пути от каждого определения переменной до ее p-use-использования. Если определению переменной не соответствует ни одного p-use-использования, то P должен включать в себя путь до некоторого c-use-использования этой переменной.

Критерий all-du-path выполняется, если P включает в себя все пути от определения переменной до ее использования, т.е. если для конкретного определения и конкретного использования переменной существует несколько соединяющих их путей, все они должны быть покрыты.

Критерии тестирования обработки исключений

В объектно-ориентированной программе (в частности, написанной на языке Java) обработка исключений в общем виде представляется конструкцией try, состоящей из обязательного блока try и необязательных блоков catch и finally. Блок try содержит

исполняемый код, в котором могут выбрасываться исключения с помощью оператора `throw`, за ним следует один или несколько блоков `catch` или блок `finally`, либо и блоки `catch`, и `finally`. Блок `try` выполняется либо полностью, либо пока не выбросится исключение. Если в процессе выполнения команд блока `try` выбрасывается исключение определенного типа, управление передается в соответствующий этому типу блок `catch`, затем оператору, следующему за блоком `try`. Если для соответствующего исключения не было найдено ни одного блока `catch`, оно выбрасывается на блок выше и далее согласно стеку вызовов, пока не найдется соответствующий этому исключению обработчик. Если определен блок `finally`, он выполняется после блока `try`, с которым связан. При этом передача управления в блок `finally` осуществляется как в случае нормального завершения выполнения блока `try` (нормальный контекст), так и в случае возникновения исключения (исключительный контекст). На рис. 1 представлены различные варианты передачи управления при обработке исключений.

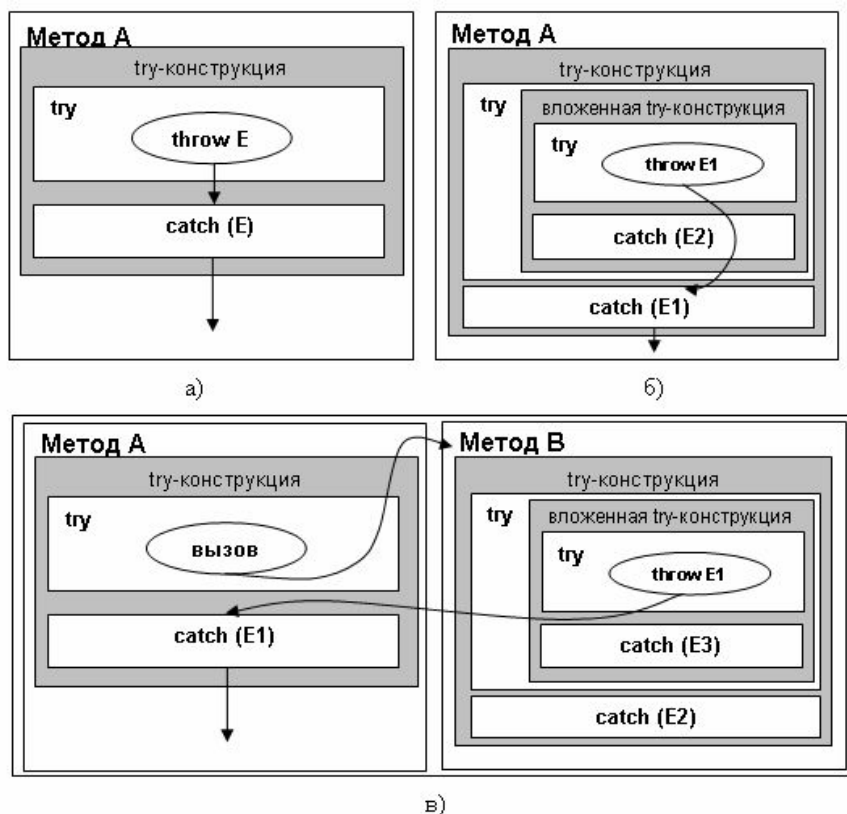


Рис. 1. Передача управления при обработке исключений: в случае одиночной try-конструкции (а), в случае вложенной try-конструкции (б), в случае продвижения исключения (в)

Критерии тестирования обработки исключений формулируются с использованием следующих специфичных для неявного потока управления понятий.

Тип исключения (e-type) – класс, экземпляры которого могут быть выброшены оператором `throw`, несет информацию от места, где было выброшено исключение до места, где исключение было обработано. В Java любой класс, являющийся подклассом `java.lang.Throwable`, является типом исключения.

Объект-исключение (eobj) – это экземпляр класса-типа исключения.

Переменная-исключение (evar) – переменная, являющаяся экземпляром класса-типа исключения.

Временная переменная-исключение (evar_i) – переменная-исключение, сопоставленная оператору `throw`, в котором указан вызов метода либо конструктора.

Объект-исключение становится *активным объектом-исключением*, когда он выбрасывается оператором `throw`. В любой момент выполнения программы активным может быть только один объект-исключение, однако в каждый конкретный момент объектов-исключений может существовать несколько. Таким образом, можно определить уникальную переменную $evar_{active}$, которая отслеживает активный объект-исключение: в каждый момент выполнения программы значение $evar_{active}$ либо не определено, если в данный момент нет ни одного необработанного исключения, либо равно $eobj_k$, если $eobj_k$ является активным объектом исключения в этот момент.

Как уже было сказано, критерии потока данных связаны с такими понятиями, как определение и использование переменных. Для обработки исключений специфично появление переменных-исключений, которые так же необходимо связать с вершинами УГ.

Для каждой вершины i в УГ множество определений исключений $e-def(i)$ содержит переменные-исключения, которые могут быть определены в вершине i . Вершина i определяет переменную-исключение, если:

1. в вершине i происходит присваивание значения переменной-исключению v ($e-def(i)$ содержит v);
2. i является вершиной `catch` ($e-def(i)$ содержит переменную, указанную в инструкции `catch`);
3. i является вершиной `throw`, такой, что в соответствующем `throw`-операторе указан вызов метода или конструктора ($e-def(i)$ содержит временную переменную-исключение $evar_i$);
4. i является вершиной `throw` ($e-def(i)$ содержит $evar_{active}$).

Для каждой вершины i в УГ, множество использований исключений $e-use(i)$, содержит переменные-исключения, которые используются в операторе, представляемом вершиной i . Вершина i использует переменную-исключение, если:

1. i имеет доступ к значению этой переменной v ($e-use(i)$ содержит v);
2. i является вершиной `catch` ($e-use(i)$ содержит $evar_{active}$);
3. i является вершиной `throw`, такой, что выражение, указанное в соответствующем операторе `throw`, является либо вызовом метода, либо конструктором ($e-use(i)$ содержит временную переменную-исключение $evar_i$, которая определена в той же вершине).

Для каждой вершины i в УГ множество *деактиваций исключений* $e-undef(i)$ содержит переменные-исключения, которые в этой вершине деактивируются. Каждая `catch`-вершина деактивирует $evar_{active}$. Таким образом, множество $e-undef$ связано только с `catch`-вершиной и содержит один элемент – $evar_{active}$.

Чистым от определений путем ($def-clear$ путь) по отношению к переменной-исключению v является путь $(i, n_1, n_2, \dots, n_m, j)$ в УГ, $m \geq 0$, такой, что нет ни одного определения или деактивации v в n_1, n_2, \dots, n_m .

Для генерации тестовых данных необходимо выявить определение переменной-исключения v в вершине i , множество вершин j , которые используют значение, присвоенное v в i (*exception def-use* множество или $e-du$). Множество $e-du$ содержит для каждого определения переменной-исключения все достижимые использования этой переменной.

В отличие от своего нормального состояния, объект-исключение может находиться в активном состоянии. Оператор `throw` активирует объект-исключение, в то время как в оператор `catch` либо `finally` деактивируют его. Процессы активации и деактивации схожи с процессами определения и использования.

Для каждой вершины i в УГ множество активаций исключений $e-act(i)$ содержит объекты-исключения, которые могут быть активированы в этой вершине. Множество

деактивации исключений $e-deact(i)$ для вершины i определяется аналогично. Активация объекта-исключения $eobj_m$ деактивирует любой другой активный объект $eobj_n$.

По аналогии с критериями потока данных, которые требуют покрытия определений и использований переменных, критерии тестирования обработки исключений требуют покрытия определений и последовательных использований переменных-исключений. Отличием является возможность покрытия альтернативного набора команд, связанных с активацией и деактивацией объектов-исключений. Однако, симметрия этих двух наборов позволяет ввести два параллельных подкласса критериев: $def-use$ (du) и $act-deact$ (ad).

На рис. 2 справа представлена иерархия критериев тестирования обработки исключений. Из рисунка видно, что некоторые критерии потока данных включают симметричные им критерии потока исключений.

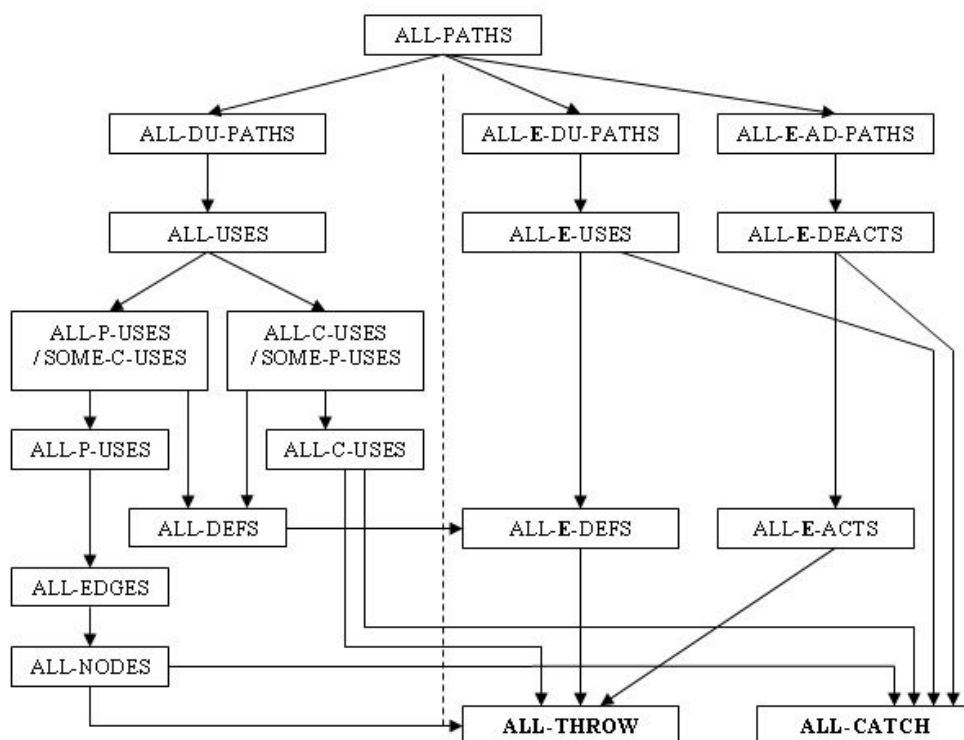


Рис. 2. Иерархия критериев потока данных (слева) и критериев потока исключений (справа)

Некоторые из критериев потока исключений ($all-e-defs$, $all-e-uses$) основаны на определении и использовании переменных-исключений и схожи с соответствующими критериями потока данных. Критерий $all-e-du-paths$ предполагает покрытие всех путей от определения переменной-исключения до всех достижимых использований.

Критерий $all-e-acts$ предполагает покрытие каждой активации исключения до одной из достижимых деактиваций, в то время как критерий $all-e-deacts$ – каждой активации до всех достижимых деактиваций. Критерий $all-e-ad-paths$ требует покрытия всех путей от каждой активации объекта-исключения до всех достижимых деактиваций.

Заключение

Рассмотрены проблемы, связанные с тестированием объектно-ориентированных программ, содержащих обработку исключений. Предложен способ анализа потока управления таких программ, а также иерархия критериев, которые могут быть использованы для структурного тестирования.

Литература

1. Sinha S., Harrold M.J. Criteria for Testing Exception-Handling Constructs in Java Programs // Technical Report OSU-CISRC-6/99-TR16. – June. – 1999.
2. Shujuan Jiang, Yuanpeng Jiang. An analysis approach for testing exception handling programs // SIGPLAN Notices. «ACM». – NY, USA. – Vol. 42. – 2007.
3. Palsberg J., Schwartzbach M. Object-oriented type inference // In Proceedings of Object-Oriented Programming Systems, Languages and Applications. – 1991. – P. 146–161.
4. Plevyak J., Chien A. Precise concrete type inference for object-oriented languages // In Proceedings of Object-Oriented Programming Systems, Languages and Applications. – 1994. – P. 324–340.

ЭФФЕКТИВНЫЙ РАЗБОР XML НА ОСНОВЕ АВТОМАТНОГО ПОДХОДА

А.А. Владыкин

Научный руководитель – д.т.н., профессор А.А. Шалыто

Предложен непроцедурный язык для описания обработчиков XML-документов, который целесообразно применять для разбора этих документов. Обработчики реализованы на основе конечных автоматов, что упрощает их структуру и понимание.

Ключевые слова: XML, SAX-обработчики, автоматы

Введение

В настоящее время язык XML [1] широко используется для разнообразных целей: от хранения настроек программ до описания векторных изображений. Поэтому все больше программ требуют поддержки этого языка, а программистам все чаще приходится писать код, который считывает XML-документ и выделяет из него фрагменты, представляющие тот или иной интерес.

Нередко встречаются XML-документы размером в десятки и сотни гигабайт. Один из таких документов – полная база данных Википедии, доступная для скачивания в виде XML-файла [2]. Размер этого файла, включающего все статьи Википедии с историей их изменений, в сжатом архиватором bzip2 виде составляет 148 гигабайт.

Как прочитать такой документ и извлечь из него полезную информацию, например, число перекрестных ссылок между статьями? Единственным вариантом в данном случае является использование технологии Simple API for XML (SAX) [3] и написание соответствующего обработчика, который позволит решить поставленную задачу. Известны также и другие технологии доступа к XML-данным, например Document Object Model (DOM) [3] или Java API for XML Binding (JAXB) [4], но они требуют загрузки всего документа в оперативную память. Это при таких размерах документов не представляется возможным.

Недостатком технологии SAX, затрудняющим его использование, является сложность написания кода для реализации указанных обработчиков. В данной работе предложена технология, позволяющая значительно упростить написание обработчиков этого класса. При этом предложен непроцедурный язык описания поведения SAX-обработчика. Обработчик реализуется на основе автоматного подхода, его код генерируется автоматически.

Simple API for XML

SAX – это стандартный способ разбора XML, при котором XML-разборщик (XML parser) последовательно читает XML-документ и уведомляет пользовательский код (SAX-обработчик, SAX handler) о каждом встречающемся открывающем или закрывающем теге и о текстовом контенте между ними.

Стандартный SAX-обработчик на языке Java расширяет класс `DefaultHandler` и переопределяет следующие три метода:

```
void startElement(String uri, String localName, String qName,  
    Attributes attrs) throws SAXException;
```

```
void endElement(String uri, String localName, String qName)  
    throws SAXException;
```

void characters(char[] ch, int start, int length)
throws SAXException;

Метод `startElement` вызывается разборщиком, когда тот обнаруживает открывающий тег, а метод `endElement` – в ответ на закрывающий тег, и, наконец, метод `characters` – для текстового контента между тегами.

SAX-обработчик как автоматизированный объект управления

Попробуем разобраться в причинах сложности SAX-обработчиков и устранить эти причины. Этой цели можно добиться с использованием автоматного подхода [5].

SAX-обработчик получает уведомления от XML-разборщика и выполняет всю работу по преобразованию потока событий в понятные приложению команды и/или структуры данных. Он должен отслеживать текущее положение разборщика в структуре XML-документа, и в зависимости от этого положения по-разному реагировать на поступающие события.

SAX-обработчик является сущностью со сложным поведением [5], так как для него характерна зависимость от предыстории. При этом SAX-обработчик можно разделить на две части:

1. **управляющая часть** (автомат или система автоматов), которая отслеживает текущее положение разборщика в структуре XML-документа и выбирает соответствующие выходные воздействия на объект управления;
2. **объект управления** – преобразователь потока воздействий со стороны управляющей системы в понятные приложению команды и/или структуры данных.

Наибольшую сложность обычно составляет управляющая (поведенческая) часть обработчика, поскольку отслеживание текущего положения разборщика в структуре XML-документа при традиционном подходе к программированию требует введения и учета многочисленных флагов либо других хитростей. При этом объект управления, несмотря на относительную простоту, теряется в управляющем коде.

В соответствии с парадигмой автоматного программирования, сущность со сложным поведением следует представлять в виде автоматизированного объекта управления [5]. Отделив объект управления (управляемую часть) от системы управления (управляющей части), можно значительно повысить качество кода XML-обработчика. Более того, можно автоматически генерировать управляющую часть на основе его описания на предлагаемом непроцедурном текстовом языке. Это делает написание SAX-обработчиков значительно более простым, а структуру более наглядной.

Текстовый язык описания SAX-обработчиков

Рассмотрим задачу извлечения списка персоналий из таблицы, находящейся в XHTML-документе. XHTML – это HTML, соответствующий всем правилам XML. Поэтому для разбора XHTML-документа можно применить технологию SAX. Эта задача выбрана в качестве примера, поскольку на ней можно продемонстрировать сложность написания SAX-обработчика при традиционном подходе к программированию.

Алгоритм извлечения списка персоналий выглядит следующий алгоритм:

- найти таблицу (`<table>`) с атрибутом `class="header"`;
- найти таблицу без атрибута `class`;
- в таблице без атрибута пропустить первую строку (`<tr>`) и все строки, у которых первая ячейка имеет атрибут `class="banner"`;

- из оставшихся строк взять вторую и четвертую ячейки (<td>) как фамилию и имя соответственно.

В соответствии с предлагаемым подходом, опишем SAX-обработчик на непроцедурном текстовом языке, специально разработанном для этой цели.

Язык этого описания состоит из следующих элементов:

- открывающий и закрывающий теги, являющиеся для обработчика входными воздействиями. Открывающий тег может иметь ограничение на значения атрибутов (например, <td class=="banner">). Выходное воздействие задается после тега в фигурных скобках (<td> { object.newPerson(); });
- скобки, используемые для группировки тегов. Со скобками могут использоваться следующие операции:
- альтернатива (...|...);
- повторение (...)*
- опциональность (...)?

Описание на предлагаемом языке SAX-обработчика, реализующего приведенный выше алгоритм, имеет вид:

```
<table class=="header"> </table>
<table class==null>
<tr> </tr>
(
  <tr>
  (
    <td class=="banner"> </td> |
    <td> { object.newPerson(); } </td>
    <td> { capture(); } </td> { object.setLastName(captured()); }
    <td> </td>
    <td> { capture(); } </td> { object.setFirstName(captured()); }
  )
  </tr>
)*
</table>
```

В соответствии с автоматной парадигмой, обработчик будет состоять из двух частей: объекта управления и автомата управления.

Объект управления предоставляет управляющей системе некоторый интерфейс. В этом примере – функции newPerson(), setLastName() и setFirstName(). Эти функции программист пишет *вручную*.

Автомат управления вызывает функции объекта управления в соответствии с выходными воздействиями, заданными в описании SAX-обработчика. При этом могут использоваться два вспомогательных метода, предоставляемых автоматом управления: capture() и captured(). Метод capture() инициирует захват поступающего текстового контента во временный буфер автомата. Метод captured() возвращает захваченный с предыдущего вызова capture() контент и очищает буфер.

Алгоритм построения автомата управления

Генерация управляющей части SAX-обработчика требует выделения управляющих состояний и построения переходов между ними.

Выделение состояний является достаточно простой задачей, так как состояния уже неявно выделены при составлении текстового описания XML-обработчика.

Приведем программу построения управляющего автомата в целом для данного примера (реализация используемых в программе классов для экономии места не приводится):

```
import java.util.List;
public class StateGenerator {
    /**
     * @param input описание обработчика в виде последовательности элементов:
     *             открывающих и закрывающих тегов и групп
     * @return сгенерированная таблица переходов
     */
    public static TransitionTable generate(List<InputElement> input) {
        TransitionTable table = new TransitionTable();
        generate(input, table, new StateSet(0), table.addState());
        return table;
    }

    private static StateSet generate(List<InputElement> input,
        TransitionTable table, StateSet initState, int finalState) {
        StateSet states = initState.clone();
        for (int i = 0; i < input.size(); ++i) {
            InputElement elem = input.get(i);
            int localFinalState = (i + 1 == input.size()) ?
                finalState : table.addState();
            switch (elem.getType()) {
                case START_TAG:
                case END_TAG: {
                    for (Integer source : states.getAll()) {
                        table.addTransition(source, (Tag) elem, localFinalState);
                    }
                    states.setAll(localFinalState);
                    break;
                }
                case GROUP: {
                    Group group = (Group) elem;
                    StateSet finalStates = generate(
                        group, table, states, localFinalState);
                    switch (group.getQuantifier()) {
                        case NONE:
                        case QUESTION:
                            states = finalStates;
                            break;
                        case ASTERISK:
                            table.copyTransitions(
                                states.getPrimary(), localFinalState);
                            states.add(finalStates, localFinalState);
                            break;
                    }
                    break;
                }
                default:
                    throw new IllegalArgumentException();
            }
        }
    }
}
```

```

    }
  }
  return states;
}

private static StateSet generate(Group group, TransitionTable table,
    StateSet initState, int finalState) {
    StateSet states = null;
    for (int j = 0; j < group.getAlternativeCount(); ++j) {
        StateSet finalStates = generate(
            group.getAlternative(j), table, initState, finalState);
        if (states == null) {
            states = finalStates;
        } else {
            states.add(finalStates);
        }
    }
    return states;
}
}
}

```

Неформально говоря, каждый открывающий или закрывающий тег переводит обработчик в новое состояние. Единственное исключение – тег в конце группы, заключенной в скобки. В этом случае новое состояние добавляется не после данного тега, а «выносится за скобки» группы.

Воздействия, не предусмотренные описанием SAX-обработчика, игнорируются и не изменяют его состояние. Например, обнаружение тегов `th` внутри первой строки таблицы вполне закономерно, и не должно влиять на поведение обработчика.

Автоматическая генерация кода автомата SAX-обработчика

Имея описание управляющего автомата в виде множества состояний и переходов между ними, можно выполнить автоматическую генерацию кода автомата SAX-обработчика для произвольного языка программирования. В рамках данной работы была реализована кодогенерация автомата SAX-обработчика на языке Java.

Результатом кодогенерации является класс, который расширяет `DefaultHandler` и реализует указанные выше методы `startElement`, `endElement` и `characters`. При этом все переходы автомата по открывающим тегам размещаются в методе `startElement`, а по закрывающим – в методе `endElement`. Автомат имеет всего две внутренние переменные: текущее состояние и буфер для временного хранения текстового контента. Таким образом, расход памяти минимален.

Заключение

В данной работе предложен непроедурный текстовый язык описания SAX-обработчиков и система автоматической генерации кода на его основе. Входными данными для системы является простое и наглядное текстовое описание обработчика. На основе этого описания строится управляющий автомат обработчика, который затем преобразуется в код на том или ином языке программирования. На настоящий момент реализовано преобразование автомата в код на языке Java. Остальной код обработчика, соответствующий объекту управления, пишется вручную. Предложенный подход отличается одновременно простотой написания кода и эффективностью использования ресурсов компьютера.

Описанный подход был использован при разработке приложения, с помощью которого извлечены данные из тысяч HTML-страниц, содержащих несколько типов таблиц сложной структуры.

Литература

1. Official website for SAX [Электронный ресурс]. Режим доступа: <http://www.saxproject.org/>, свободный. Загл. с экрана. Яз. англ.
2. Wikipedia Database Download [Электронный ресурс]. Режим доступа: http://en.wikipedia.org/wiki/Wikipedia_database, свободный. Загл. с экрана. Яз. англ.
3. W3C Document Object Model [Электронный ресурс]. Режим доступа: <http://www.w3.org/DOM/>, свободный. Загл. с экрана. Яз. англ.
4. JAXB Reference Implementation [Электронный ресурс]. Режим доступа: <https://jaxb.dev.java.net/>, свободный. Загл. с экрана. Яз. англ.
5. Поликарпова Н.И., Шалыто А.А. Автоматное программирование. – СПб: Питер. – 2009. – 176 с.

УТОНЧЕНИЕ ИЗОБРАЖЕНИЙ СИМВОЛОВ НА ОСНОВЕ КЛЕТОЧНЫХ АВТОМАТОВ

Д.И. Суясов

Научный руководитель – д.т.н., профессор А.А. Шалыто

В статье предлагаются два подхода к решению задачи утончения изображений символов на основе клеточных автоматов. Первый из них построен на основе циклического клеточного автомата, а второй – использует метки как расширение классического клеточного автомата. Эти подходы решают также и дополнительную задачу – обеспечивают фильтрацию точек шума на изображениях символа.

Ключевые слова: распознавание, фильтрация, предобработка, клеточный автомат

Введение

Задача утончения символов в изображениях является составной частью задачи распознавания символов. Многие алгоритмы [1, 2] предполагают использование алгоритма утончения в качестве предобработки изображения. При этом данная предобработка либо влияет на производительность и скорость работы, либо напрямую влияет на качество результата распознавания.

Под алгоритмом утончения в данной статье понимается уменьшение толщины линий и отрезков в изображении символов до толщины в одну точку раstra.

Работа над алгоритмами утончения ведется уже несколько десятилетий и продолжается в настоящее время. Существующие алгоритмы [3–7] позволяют решить задачу утончения на основе сложного анализа изображения. Некоторые из них используют элементы теории клеточных автоматов, но формально строят алгоритмы на неочевидных преобразованиях.

Цель данной статьи – создание простого подхода по утончению изображений символов на основе теории клеточных автоматов [8, 9]. В статье предложено два алгоритма, которые удовлетворяют критериям, предъявляемым при решении задачи утончения символов.

Критерии решения задачи утончения символов

Для решения задачи утончения символов необходимо выделить основные требования, предъявляемые к исследуемому алгоритму [3].

- (1) Сохранение связности отрезков изображения символа.
- (2) Сохранение конечных точек.
- (3) Толщина линий изображения символа должна равняться одной точке. При этом соседними точками в линии считаются только те точки, которые связаны вдоль осей.
- (4) Изотропность – изображение символа должно сохранять стилевое соответствие с необработанным вариантом, при этом не должно возникать дополнительных наклонов и кривых.
- (5) Независимость от шума. Точка изображения символа считается точкой шума, если она выступает на изображении линии символа на одну клетку.

Все перечисленные требования могут быть предъявлены к алгоритму утончения для выявления успешности выполнения задачи.

Представленные требования не являются универсальными для всех алгоритмов утончения. Основная цель их введения – упрощение задачи и адаптация алгоритма под требования системы распознавания текста, частью которой данный алгоритм и является.

Первый подход на основе циклических клеточных автоматов

Для решения задачи утончения символов предлагается использование клеточного автомата, заданного на двумерной плоскости [8].

Переформулировав определение в начале статьи, можно отметить, что утончение или выделение скелета (остова) символа предполагает замену его линий толщиной в несколько точек линиями толщиной в одну точку. Расположение сформированных линий должно соответствовать символу и способствовать адекватному восприятию результирующего изображения символа. Данному требованию удовлетворяет положение обработанной линии символа в центре, внутри исходной необработанной линии (рис. 1).

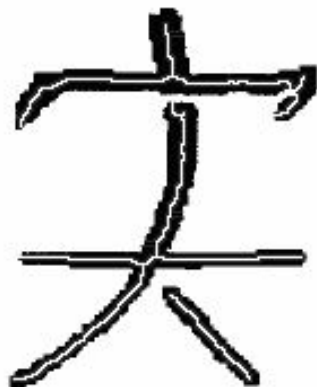


Рис. 1. Изображение символа с исходной толщиной (черными чернилами) и результирующей толщиной символов (белыми чернилами на изображении исходных линий)

Определение положения обработанной линии символа может быть выполнено клеточным автоматом, который выделяет граничные точки символа и удаляет их. На рис. 2 выделены четыре типа граничных точек символа: угловые, крайние, внутренние и точки, не входящие в предыдущие группы. Угловые точки – это точки, граничащие с остальными точками символа лишь с одной или двух соседних сторон. Крайние точки – точки, граничащие с точками символа с трех или двух несмежных сторон. Внутренние точки окружены точками символа.

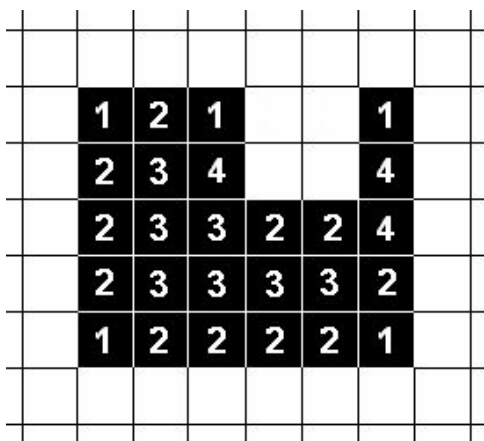


Рис. 2. Типы точек, из которых состоит изображение символа: 1 – угловые; 2 – крайние; 3 – внутренние; 4 – остальные символы

Таким образом, в процессе утончения клеточный автомат должен выделять угловые и крайние точки и удалять их. Для этих целей предложено использовать классический клеточный автомат в двумерной среде, который определяется на основе правил переходов. На рис. 3 выделено восемь правил переходов автомата. Каждое

правило определяет переход состояния центральной клетки на основе ее текущего состояния и состояний ее соседей; если клетка заштрихована, то начальное состояние клетки не влияет на выполнение правила.

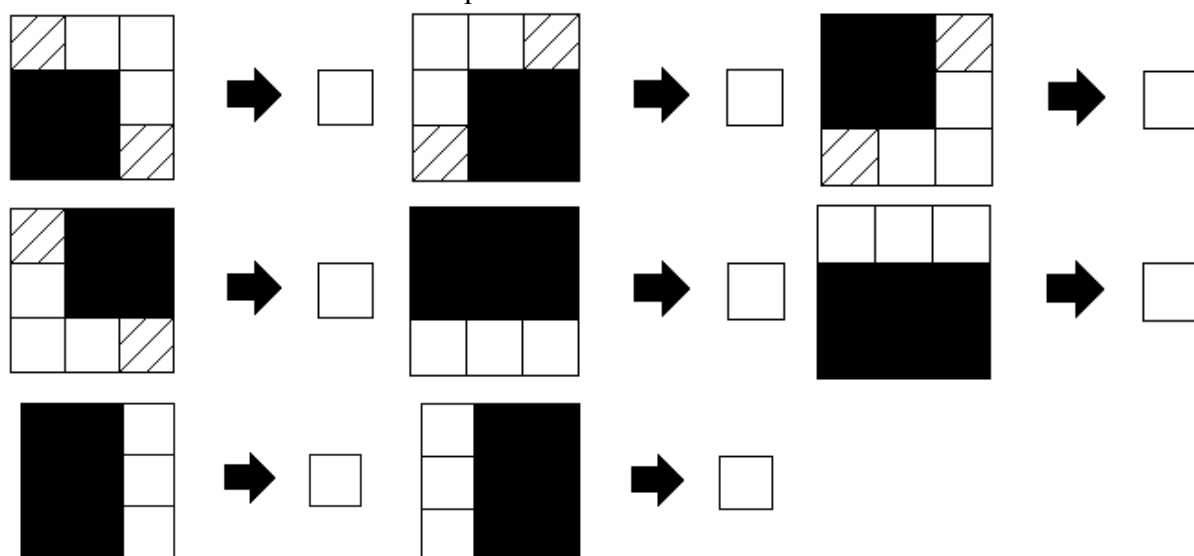


Рис. 3. Восемь правил выделения и удаления крайних точек символа. Заштрихованные точки в правилах указывают на неограниченное состояние клетки (она может быть окрашена как в черный, так и в белый цвет)

Стоит отметить, что одновременное выполнение приведенных правил одного клеточного автомата не приводит к желаемому результату: линии изображения символа с толщиной в четное число клеток не утончаются, а исчезают полностью. Ситуацию может исправить использование циклического клеточного автомата [8], где правила применяются ко всему изображению поочередно, т.е. в каждый момент времени для изменения состояния поля используется только одно из правил. После применения последнего из них клеточный автомат начинает цикл заново.

Данный автомат успешно выполняет утончение символов, но при этом сильно подвержен влиянию шумов. Любая выделяющаяся точка символа воспринимается автоматом как элемент символа. Поэтому она является причиной появления лишних линий на результирующем утонченном изображении символа (рис. 4б).

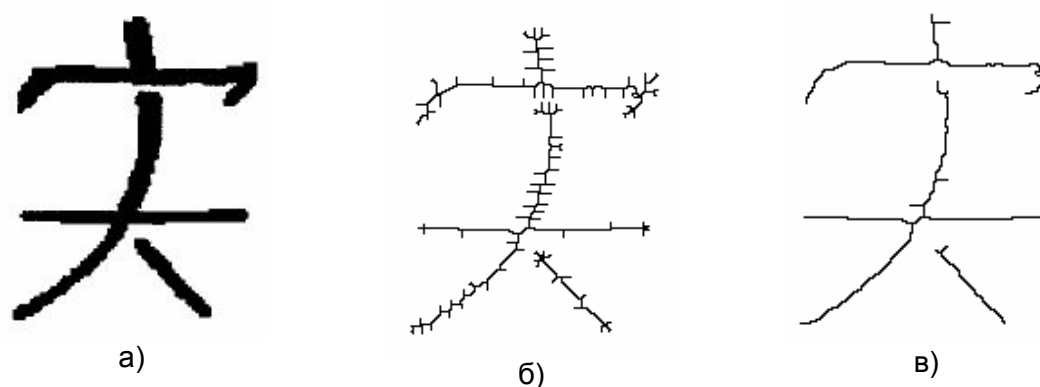


Рис. 4. Результат утончения символа при применении циклического клеточного автомата (а) исходное изображение символа; (б) обработанное изображение на основе клеточного автомата с правилами из рис. 3; (в) обработанное изображение на основе клеточного автомата с правилами из рис. 3 и рис. 5

Для решения задачи удаления шума в клеточный автомат были введены дополнительные правила, которые удаляют точки шума на каждой итерации работы автомата (рис. 5).

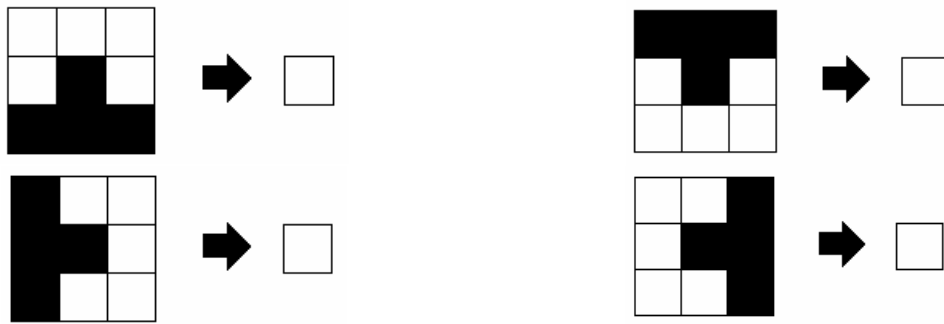


Рис. 5. Правила клеточного автомата для удаления точек шума

В результате применения шумопонижающих правил работа алгоритма становится значительно эффективнее – влияние шума сводится к минимуму (рис. 4в).

Второй подход на основе клеточных автоматов с разделением процесса утончения

Первый подход требует многократного обхода изображения для поочередного применения правил, но системы распознавания символов, в составе которых часто используются алгоритмы утончения, имеют некоторые ограничения по производительности.

Для оптимизации процесса утончения предлагается второй подход, являющийся двухэтапным процессом утончения. На первом этапе помечаются крайние и угловые точки изображения символа (рис. 2), а на втором этапе удаляются только те помеченные точки, которые граничат с непомеченными точками изображения символа.

Для реализации данного процесса за основу взяты клеточные автоматы с метками [10]. В этом случае выделение крайних и угловых точек изображения означает присвоение метки граничной клетке (граничной по фон Нейману). На рис. 6 представлен полный список правил клеточного автомата с метками.

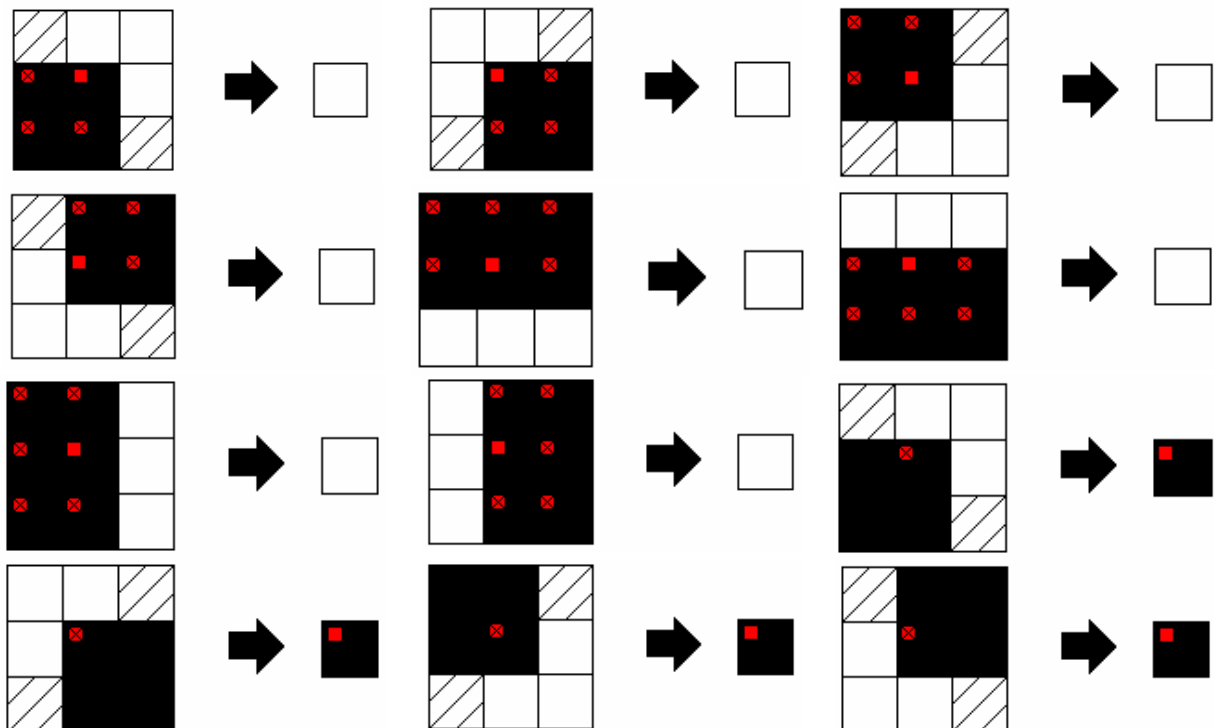


Рис. 6. Правила выделения и удаления крайних точек символа на основе разделения процесса утончения. Заштрихованные точки в правилах указывают на неограниченное состояние клетки (она может быть окрашена как в черный, так и в белый цвет), красный квадрат в клетке указывает на метку, зачеркнутый красный квадрат – отсутствие метки в клетке

Описанный клеточный автомат позволяет сократить толщину линий символа до двух точек. Завершение процесса утончения производится на основе первого подхода.

Пример работы алгоритма с разделением процесса утончения представлен на рис. 7.



Рис. 7. Результат утончения символа при применении двухэтапного клеточного автомата на базе меток, (а) исходное изображение символа; (б) результирующее изображение символа после работы алгоритма

Второй подход имеет лучшие характеристики производительности по сравнению с первым подходом, суммарное количество итераций клеточных автоматов оказывается меньшим. Последний факт позволяет провести дополнительную оптимизацию выполнения правил клеточного автомата.

Оценки производительности и качества

Исследование предложенных подходов проводилось на основе программы [10] по моделированию работы последовательностей клеточных автоматов. Результаты исследования представлены в таблице ниже.

В таблице представлены число итераций и время работы алгоритмов на тестовых данных.

Таблица. Сравнение алгоритмов на базе циклического клеточного автомата и клеточного автомата с метками по числу итераций автомата и времени выполнения

Тестовое изображение	Первый подход на основе клеточных автоматов	Второй подход на основе клеточных автоматов с разделением процесса утончения
Японский символ, изображение 160×190 точек	100 итераций, 4.4 сек.	31 итерация, 3.0 сек.
Китайский символ, изображение 280×250 точек	172 итерации, 9.8 сек.	45 итераций, 7.1 сек.
Символ «е», изображение 33×42 точки	46 итераций, 0.08 сек.	18 итераций, 0.07 сек.
Символ «ж», изображение 155×155 точек	82 итераций, 1.7 сек.	26 итераций, 1.1 сек.
Русский алфавит, изображение 455×55 точек	37 итераций, 0.6 сек.	17 итераций, 0.4 сек.
Текст на английском языке (63 символа), изображение 600×40 точек	37 итераций, 0.7 сек.	17 итераций, 0.4 сек.
Текст на русском языке (28 символов), изображение 950×70 точек	73 итерации, 4.2 сек.	23 итерации, 3.4 сек.

Сравнение показывает, что второй подход является более эффективным как по количеству итераций, так и по времени выполнения. Стоит отметить, что время может быть дополнительно сокращено при оптимизации реализации алгоритмов.

Заключение

В данной статье предложены два подхода по утончению символов на изображениях. Первый из них основан на циклическом клеточном автомате, а второй решает задачу с помощью клеточного автомата с метками. Алгоритмы, положенные в основу подходов, показывают эффективность решения задачи утончения символов с помощью теории клеточных автоматов и позволяют производить дальнейшие исследования и оптимизации данного направления.

Задачи утончения решаются с учетом присутствия шума на изображениях символов, решение основано на 12 циклических правил (рис. 3, 5) в первом алгоритме и того же числа правил с метками во втором случае (рис. 6). Производительность данных решений имеет прямую зависимость от начальной толщины символов на изображении, а также от размеров изображения.

Дальнейшая оптимизация может быть основана на объединении соседних точек изображения и работе с ними как с одним целым.

Литература

1. OCR&ICR Technology White Paper. Аналитика [Электронный ресурс] / Аналитика. Электрон, дан. – М.: Инф. портал DOCFLOW, 2005. Режим доступа: http://www.docflow.ru/analytic_full.asp?param=30387 свободный – Загл. с экрана. – Яз. рус.
2. Maher Ahmed, Rabab Ward, A Rotation Invariant Rule-Based Thinning Algorithm for Character Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence. Dec. – 2002. – Vol. 24. – №. 12. – PP. 1672–1678.
3. Levaire J.-L., Ubeda S.A multiscale parallel thinning algorithm // Lecture Notes in Computer Science, Springer Berlin / Heidelberg. – 1992. – PP. 595–600.
4. Olszewski J. A flexible thinning algorithm allowing parallel, sequential, and distributed application //ACM New York. – 1992. – Vol. 18. – Issue 1. – PP. 35–45.
5. Kwok P. A thinning algorithm by contour generation // Univ. of Calgary. – 1988. – Vol. 31. – Issue 11. – PP. 1314–1324.
6. Guo Z., Hall R.W. Parallel thinning with two-subiteration algorithms //ACM New York. – 1989. – Vol. 32. – Issue 3. – PP. 359–373
7. Huang L., Wan G., Liu C. An Improved Parallel Thinning Algorithm /International Conference on Document Analysis and Recognition (ICDAR). – 2003.
8. Wolfram S.A New Kind of Science. Wolfram Media. – 2002.
9. Наумов Л.А. Разработка среды и библиотеки CAME&L для решения задач с использованием клеточных автоматов. СПбГУ ИТМО. – 2003. – <http://is.ifmo.ru/papers/camel>
10. Суясов Д.И. Разработка алгоритмов распознавания текста на основе клеточных автоматов. СПбГУ ИТМО. – 2007. – http://is.ifmo.ru/papers/28_07_2007_Sujasov

ОБЗОР МАРШРУТОВ ПРОЕКТИРОВАНИЯ ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПЛИС/ASIC/SOC НА ОСНОВЕ ЯЗЫКОВ C/C++

Е.И. Колесников

Научный руководитель – д.т.н., профессор А.А. Шалыто

Рассмотрены современные маршруты проектирования прикладного программного обеспечения для программируемых логических интегральных микросхем (ПЛИС), специализированных заказных микросхем (*ASIC*), систем на кристалле (*СнК* или *SoC*) на основе языков *C/C++*; произведено сравнение маршрутов проектирования и показаны отличия их; рассмотрены аппаратные версии языков *C/C++* и проведено их сравнение с языками описания аппаратуры по охвату уровней проектирования.

В статье основное внимание уделяется аппаратным диалектам языков *C/C++* и исследуется целесообразность их использования в маршрутах проектирования ПО для *ПЛИС/ASIC/SoC*; указываются проблемы при работе с традиционными языками описания аппаратуры (*HDL*). Рассматривается возможность построения мультязычной системы проектирования на основе связи языков *SystemC–SystemVerilog*. Показаны преимущества и недостатки предлагаемых сегодня маршрутов проектирования *ПЛИС/ASIC* (в том числе систем на кристалле) и диалектов языков *C/C++* описания аппаратуры для данной области разработок.

Ключевые слова: маршруты проектирования, ПЛИС, *ASIC*, *SoC*, языки *C/C++*, *HDL*, *SystemC*, *SystemVerilog*, *ESL-проектирование*

Введение

Современный уровень проектирования быстродействующих алгоритмически сложных цифровых систем на базе языков описания аппаратуры (*HDL – Hardware Description Language*), например, *VHDL*, *Verilog* и др. испытывает серьезные проблемы с производительностью труда инженеров – это постоянно растущая сложность разрабатываемых проектов и необходимость в универсальном способе описания, пригодном для всех уровней проектирования и независимом от формы реализации проектов [1]. Поэтому все время ведутся разработки новых средств проектирования, которые обеспечат более удобные и эффективные способы ввода и отладки описаний алгоритмов функционирования цифровых систем, автоматически генерируя корректные «по построению» *HDL*-описания.

Существует два маршрута проектирования *ПЛИС/ASIC/ SoC*.

- 1) Традиционный нисходящий маршрут проектирования.
- 2) Маршрут с абстракцией проекта на системном уровне (*ESL-проектирование*).

Маршрут проектирования (*design flow*) определяет этапы проектных процедур, используемых на всех стадиях разработки, — от выработки и формализации идеи до тестирования готовых образцов.

Разработка средств проектирования *ПЛИС/СБИС* системного уровня (*Electronic System Level, ESL*) – одно из самых актуальных направлений развития индустрии САПР электронной техники. Размеры электронных систем растут, традиционные методы проектирования *ПЛИС/СБИС* становятся неэффективными. Распространение *ПЛИС/SoC* делает современные микроэлектронные технологии доступными для разработчиков, не знакомых с особенностями проектирования *СБИС*.

Недостаток нисходящего проектирования заключается в том, что с увеличением сложности проекта увеличивается опасность появления ошибок и затрудняется процесс их поиска. То, насколько удовлетворяет разрабатываемая *СБИС* предъявляемым к ней требованиям, становится ясно только в самом конце процесса проектирования. Ошибки, обнаруженные в конце той или иной стадии проектирования, ведут к повторному ее выполнению, что в ряде случаев влечет неоднократный выпуск

прототипов, значительно замедляя сроки выполнения проекта и резко повышая его стоимость [2].

ESL-проектирование или проектирование «сверху вниз» основано на имитационном моделировании. Ключ к реализации *ESL*-подхода – моделирование на более высоких уровнях абстракции, нежели на уровне регистровых передач (*RTL*-уровень или *Register Transfer Level*). Задачи *ESL*-средств – обеспечить широкому кругу пользователей удобную и понятную среду проектирования и сократить сроки разработки [3].

В последние годы разработчики, как ПЛИС, так и САПР стали больше внимания уделять физическому и системному (*ESL*) уровню проектирования. Это обусловлено переходом на новые технологии (90 нм и 65 нм) и, следовательно, необходимостью учета новых физических эффектов. Потребность же в средствах проектирования системного уровня вызвана увеличением размерности проектов и тенденцией создания систем на кристалле.

При разработке сложных электронных систем, как например, ПЛИС/*ASIC* и систем на кристалле, возникает необходимость разработки программной и аппаратной части. При этом, как правило, вначале разрабатывается модель системы, которая описывает поведение системы на верхнем структурном уровне. Эта модель предназначена для проверки заложенных в систему алгоритмов и совместно с моделями физических уровней (радиолинии, проводной связи, объекта управления и т.д.) используется для моделирования всей системы. В настоящее время ввиду своей популярности и универсальности для программного моделирования широко используется язык *C++*. После завершения моделирования система разбивается на аппаратную и программную составляющие, и разработка идет по двум направлениям. Для описания аппаратной части проекта используются языки описания аппаратуры (*HDL*): *Verilog* или *VHDL* (также могут использоваться низкоуровневые языки *HDL* и схемотехнический ввод). В результате такого подхода возникает проблема связи между исходной поведенческой моделью верхнего уровня и более детальным описанием аппаратной части проекта, теряется возможность моделировать весь проект целиком, и либо последующие изменения в аппаратной части проекта повторяются в высокоуровневой модели (требуется удвоенное время на внесение изменений как в *HDL*, так и в *C++*-модель), либо высокоуровневая модель устаревает и не используется. После разработки аппаратной части требуется полностью переписывать программную часть, так как система должна работать в реальном времени и пользоваться специфическими возможностями *RTOS* (многозадачность, прерывания, детерминированное время отклика), также возможно, что реальная аппаратная часть сильно отличается от исходной модели. Возникает необходимость в универсальном языке программирования, как для описания поведенческих моделей системы, так и для описания аппаратно-программной реализации [4].

Таким образом, задача состоит в том, чтобы обеспечить автоматизированный переход от эталонной программной модели системы к ее программно-аппаратной реализации.

В качестве начальной задачи можно рассматривать разработку языка, позволяющего описывать аппаратуру, используя синтаксис *C/C++*. А при наличии поведенческой модели на *C++* возможно использование готового кода с минимальными изменениями.

Преимущества автоматического синтеза *RTL*-описаний из исходного описания алгоритмов на *C/C++* [5].

- Устраняется разрыв между алгоритмическим уровнем описания системы (*MATLAB*, *C/C++*, *SystemC*) и *RTL*-описанием, поскольку язык *C/C++*, как

правило, хорошо знаком как системным инженерам, так и специалистам, работающим с *RTL*-описанием (*VHDL* и/или *Verilog*).

- Основная нагрузка по отладке и верификации проекта переносится на более высокий уровень абстракции, а на этап синтеза поступает проект, уже практически полностью отлаженный на системном уровне. Поскольку верификация модели на *C/C++* выполняется на несколько порядков быстрее, чем *RTL*-модели, сроки проектирования могут быть кардинально сокращены.
- Анализ альтернативных вариантов реализации алгоритмов на *C/C++* и внесение возможных изменений выполняется быстро и эффективно.
- Описание проекта на *C/C++* не привязано к выбору микроархитектуры и конкретной технологии реализации. Это преимущество является ключевым, поскольку позволяет полностью абстрагироваться от особенностей реализации, не перегружая исходное описание дополнительными операторами настройки на конкретную элементную базу. В зависимости от потребностей текущей разработки одно и то же описание может быть использовано как для реализации проекта на базе ПЛИС различных типов, так и в виде *ASIC*. Настройка и оптимизация *RTL*-описания под конкретную элементную базу осуществляется средствами *C/C++*-программы в процессе синтеза *RTL*-кода из описания на языке *C/C++*.

Развитие использования *C/C++* для описания архитектуры в настоящее время идет по трем направлениям [6].

1. Разработка библиотек на *C/C++* — *SystemC* (www.systemc.org) и *Cynlib* (www.ForteDS.com), которые должны использоваться совместно со стандартным компилятором *C++*.
2. Разработка специального языка, основанного на синтаксисе *ANSI C* и имеющего специальные конструкции для моделирования аппаратуры — *Handel-C* (www.celoxica.com) и *SpecC* (www.specc.org).
3. Разработка языка на базе *Verilog* и *C++* — *SystemVerilog* (www.systemverilog.org).

Наиболее перспективными языками являются *SystemC* и *System Verilog*. На сегодняшний день стандарт языков *SystemC* и *SystemVerilog* в той или иной мере поддерживается всеми ведущими производителями САПР микроэлектроники. Активно развиваются средства проектирования на их основе. У этих языков есть хорошие шансы в ближайшее время стать базовыми средствами проектирования СБИС и систем на кристалле [7].

1. Маршруты проектирования ПЛИС

Рассмотрим современные маршруты проектирования прикладного программного обеспечения для ПЛИС/*ASIC* (в том числе *SoC*), в которых применяются языки *C/C++* на основе двух подходов: традиционного *HDL*-проектирования и нового *ESL*-подхода.

Традиционный маршрут проектирования

Типовой проект ведется следующим образом (рис. 1). Проект начинается с проработки *технического задания* на проектирование и создания *спецификаций проекта*. Далее выполняется поведенческое описание проекта.

В общем случае исходными компонентами проектируемой системы могут быть текстовые спецификации, программы и модели на языках *C/C++*, высокоуровневые аппаратные спецификации на языке *SystemC*, функциональные модели *Matlab/Simulink*, а также готовые программные и аппаратные (синтезируемый код *VHDL/Verilog* на уровне регистровых передач *RTL*) и другие блоки.

На этом этапе моделируется то, как проект будет взаимодействовать с внешней средой.

Создание общей модели предполагает интеграцию всех этих разнородных компонентов в единую систему и ее верификацию как программно-аппаратного комплекса. На этом этапе проектирования проводят верификацию смешанного кода *C/C++/Handel-C/SystemC* и совместное моделирование с системами логического моделирования (например, *Active-HDL* и *ModelSim*) и *Matlab/Simulink*.

После того, как этот этап выполняется, разработчики переходят к одному из наиболее существенных этапов – *RTL*-описанию. На этом этапе производится моделирование работы устройства на уровне передачи информации между регистрами.

Далее производится компиляция проекта под конкретную технологию, проверяются временные соотношения сигналов, и производится окончательная трассировка на физическом уровне [8, 9, 10].



Рис. 1. Традиционный маршрут проектирования ПЛИС/ASIC/SoC на основе *HDL*-языков проектирования

Маршрут проектирования с использованием общесистемного уровня

Общесистемный уровень *ESL* (Entire System Level) состоит из трех основных этапов: спецификация системы (Service + Specification), уровень сообщений (Message Level) и уровень передач (Transaction Level), связь между которыми отображена на рис. 2 [11].

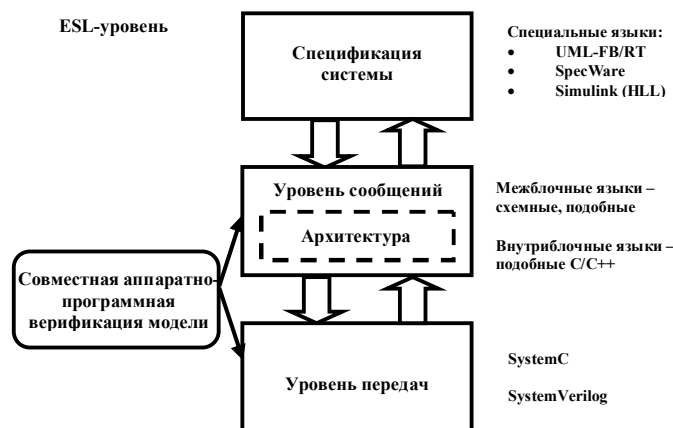


Рис. 2. Общесистемный уровень разработки

На этапе спецификации системы (Service + Specification):

- осуществляется постановка задачи без разделения на аппаратную и программную части;
- описывается система на уровне крупных функциональных блоков с использованием специальных языков (рис. 2);
- отлаживается взаимодействие между блоками и корректность спецификации системы.

На уровне сообщений (Message Level):

- формируется архитектура системы;
- описываются блоки на стандартных языках программирования без разделения на аппаратную и программную части;
- формируются прообразы ПО и функционального описания аппаратной части;
- верифицируется программная модель на более детальном уровне.

В случае невозможности реализации выполняется коррекция спецификации системы, в случае успешного результата проводится вручную разделение задач на программную и аппаратную части.

На уровне передач (Transaction Level):

- отлаживается совместимость программной и аппаратной частей системы;
- все блоки описываются на специальных языках;
- формируется среда верификации для аппаратной части;
- проверяется совместимость со стандартными операционными системами;
- продолжается определение оптимального разделения на аппаратную и программную части.

Для моделирования *Verilog*- и *VHDL*-кодов предназначены специальные симуляторы. В частности, это позволяет, разработав системную модель сложного устройства, довести отдельный его блок до вентилярного и даже до транзисторного уровня и верифицировать системную модель, используя уже готовый блок. В этом случае может выясниться, например, что какие-либо параметры, заложенные в блок на системном уровне, не реализуются, и тогда можно изменить саму системную модель.

Для перехода с системного уровня к непосредственному проектированию СБИС необходимо описание проекта на уровне регистровых передач (*RTL*). Переход от системной модели к *RTL* возможен двумя путями.

Традиционный подход представляет собой ручное написание соответствующего кода на языках *Verilog/VHDL*. Описывать *RTL*-модели позволяет и *SystemC*, причем для этого не надо заново переписывать код – достаточно лишь дополнить существующее описание определенными конструкциями.

Другой подход, предоставляемый производителями САПР ПЛИС, – это генерация так называемой поведенческой модели на *SystemC* (синтезируемый код *SystemC*). При этом формируется некая модель в виде «черного ящика», у которой определены интерфейсы, временные характеристики, выполняемые функции, однако внутренняя структура не детализирована. Из такой модели можно синтезировать логическое описание проекта.

Иными словами, система сама синтезирует внутреннюю микроархитектуру СБИС.

Технология логического синтеза на основе поведенческой *RTL*-модели позволяет быстро получить ответ на вопрос, реализуема ли в принципе та или иная система и какие аппаратные ресурсы СБИС (число вентиляей, площадь кристалла, энергопотребление, частотные характеристики и т.д.) для этого нужны. Причем синтезируемые компилятором САПР структуры если и превышают по числу вентиляей разработанные вручную, то в среднем не более чем на 10% [4, 8, 9].

Далее разработка частей системы (в части микроэлектроники, аппаратуры и ПО) ведется параллельно, по традиционным маршрутам проектирования.

2. Сравнение аппаратных языков C/C++ и HDL-языков

Фундаментальное отличие различных подходов к проектированию на основе аппаратных языков C/C++ от HDL-языков, определяется охватом этими языками определенных стадий проектирования (рис. 3).

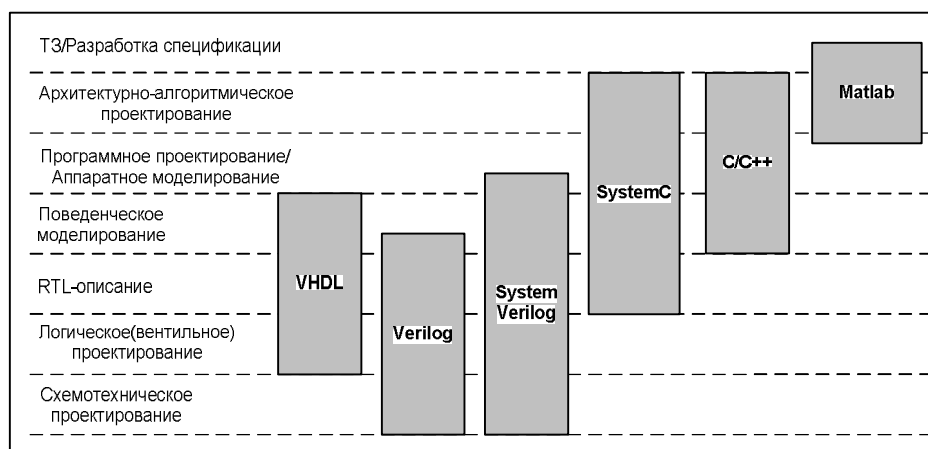


Рис. 3. Сравнение аппаратных языков C/C++ и HDL-языков по охвату стадий проектирования [13, 15, 16]

Три ключевых момента можно увидеть на рис. 3.

- *SystemVerilog* не заменяет *SystemC* – у каждого языка есть уникальные возможности.
- *SystemVerilog* соединяет промежуток между *Verilog* и *SystemC*.
- Перекрывание между *SystemVerilog* и *SystemC* облегчает смешивать модели, написанные на разных языках в пределах того же самого моделирования [16].

SystemC — язык проектирования и верификации моделей системного уровня, реализованный в виде C++-библиотеки с открытым исходным кодом [13, 14, 15].

Аппаратные диалекты языков C/C++ преимущественно используется сегодня для встроенного системного программного обеспечения. Языки описания аппаратных средств *VHDL* и *Verilog* используются для того, чтобы моделировать и синтезировать цифровые схемы. *SystemVerilog* — новый язык, который расширяет язык *Verilog* для того, чтобы обратиться ко многим проблемам проектирования аппаратно-ориентированных систем [19, 20]. *MATLAB* используется в качестве средства высокоуровневого описания, моделирования и исследования проектного пространства встроенной системы, а затем автоматически генерирует синтезируемое HDL-описание, которое может быть загружено в соответствующую ПЛИС/ASIC/SoC [15].

Из рис. 3 видно, что языки C/C++ и *SystemC* подняли уровень абстракции, при которой проектировщик определяет проект ближе к концептуальной модели. При таком подходе проект прогрессирует от вполне функциональных технических требований до структурной реализации за несколько шагов. Это приводит к снижению времени разработки, более эффективному использованию пространства проекта и позволяет снизить время перепроектирования [2, 18].

Также из рис. 3 можно сделать вывод, что эффективнее будет смешанное использование языков *SystemC* и *SystemVerilog*, чем применение одного определенного универсального языка описания аппаратуры, так как вышеназванные языки в совокупности охватывают все стадии проектирования СБИС, и каждый из них эффективен на определенных стадиях проектирования. На системном уровне работаем с *SystemC*, а затем с помощью *SystemVerilog* «конкретизируем» содержимое уровня реализации. Одним из преимуществ мультязычных систем проектирования и тестирования заключается в том, что можно использовать тестовые воздействия языка

C/C++ или *SystemC* при работе средств моделирования на уровнях абстракции *RTL* и вентилей. И понадобится лишь незначительная модификация кода вместо полного ручного переноса тестовых воздействий из *C/C++* в *VHDL/Verilog* [16, 17].

Заключение

Внедрение автоматического синтеза ПЛИС/*ASIC/SoC* непосредственно из *C*-описаний может расширить круг разработчиков аппаратного обеспечения, повысить их производительность труда, сократив тем самым сроки и стоимость разработки.

Конкурентные преимущества подхода к проектированию с применением диалектов языков *C/C++* в маршрутах проектирования с *ESL*-подходом:

- создание ПЛИС/*ASIC/SoC*, аппаратуры и ПО, изначально адаптированных к требованиям системы и согласованных друг с другом;
- разработка и отладка ПО, разработка ПЛИС/*ASIC/SoC* и разработка аппаратуры проводятся параллельно под контролем общесистемного уровня;
- к выпуску прототипов ПЛИС/*ASIC/SoC* готовы средства тестирования, отладки, а также работоспособное прикладное и системное ПО и макет аппаратуры.

Следовательно, применение маршрута с общесистемным уровнем и систем мультязычного проектирования на основе языков *SystemC–SystemVerilog* в современных маршрутах проектирования обеспечивают создание высококачественной системы в кратчайшие сроки. Для всего цикла разработки встраиваемой системы использовать единый язык программирования нецелесообразно из-за того, что все языки описания аппаратуры имеют ограничения по охвату уровней проектирования (рис. 3) и по уровню абстракции [17].

Описанные выше принципы проектирования на общесистемном уровне могут быть положены в основу исследований по разработке конкретной методологии проектирования прикладного ПО для ПЛИС/*ASIC/SoC* в рамках автоматного программирования. Методы автоматного программирования позволят использовать формализованные методы проектирования, уже начиная со спецификации. Разработка и внедрение таких методов является необходимым условием для эффективного использования методов высокоуровневого проектирования.

Литература

1. Долинский М. Обзор современных подходов и средств к «программисткой» разработке аппаратного обеспечения алгоритмически сложных цифровых систем // Компоненты и технологии. – 2004. – № 1. – С. 2–15.
2. Стешенко В. Проектирование СБИС. Стили и этапы проекта. // Компоненты и технологии. – 2003. – № 4. – С. 2–5.
3. Строганов А. Проектирование цифровых автоматов с использованием системы MATLAB/Simulink // Компоненты и технологии. – 2008. – № 4. – С. 1–4.
4. Бухтев А. Средства ESL-проектирования компании Celoxica. Программно-ориентированный подход. // Электроника: Наука, Технология, Бизнес. – 2006. – № 8. – С. 1–3.
5. Лохов А., Селиванов И. Система CATAPULT С SYNTHESIS: C++ в FPGA // Электроника: Наука, Технология, Бизнес. – 2005. – № 5. – 1 с.
6. Емец С. Использование языков C и C++ для HDL-описания — будущий стандарт или неудачный эксперимент? // Компоненты и технологии. – 2002. – № 9. – С. 1–6.
7. Пеженков А. Радченко Д. Язык SystemVerilog: проектирование СБИС и систем // Электроника: Наука, Технология, Бизнес. – 2006. – № 4. – С. 1–5.
8. Кравченко В., Радченко Д. САПР компании Synopsys. Основные средства и возможности. // Электроника: Наука, Технология, Бизнес. – 2003. – № 5. – С. 1–3.

9. Лохов А., Рабовалюк А. Средства проектирования FPGA компании Mentor Graphics // Электроника: Наука, Технология, Бизнес. – 2004. – № 4. – С. 1–4.
10. Verilog HDL: A Guide to Digital Design and Synthesis, Second Edition. Samir Palnitkar. Prentice Hall PTR. – 2003. – 396 с.
11. Маршрут проектирования СнК: www.ipmce.ru/custom/ekb/projects/
12. Кравченко В., Радченко Д. Виртуальное прототипирование для аппаратно-программной верификации СБИС // Электроника: Наука, Технология, Бизнес. – 2003. – № 7. – С. 1–4.
13. Знакомство с SystemC: www.systemc.dax.ru/book/
14. IEEE Std 1666-2005 SystemC Language Reference Manual / The Institute of Electrical and Electronics Engineers, Inc. 3 Park Avenue, New York, NY 10016-5997, USA. – 2008.
15. Black D. C., Donovan J. Eklectic Ally, Inc. – SystemC: From the Ground Up – Kluwer Academic Publishers, Boston. – 2004. – 244 с. – Яз. англ.
16. Sutherland S. Integrating SystemC Models with Verilog and SystemVerilog Models Using the SystemVerilog Direct Programming Interface // SNUG Europe 2004 // http://www.sutherland-hdl.com/papers/2004-SNUG-Europe-paper_SystemVerilog_DPI_with_SystemC.pdf
17. Максфилд К. Проектирование на ПЛИС. Курс молодого бойца. – М.: Издательский дом «Додэка-XXI», 2007. – 408 с.: (Серия «Программируемые системы»).
18. Verkest D., Kunkel J., Schirmeister F. System Level Design Using C++ // ACM 2000, Paris, France. <http://delivery.acm.org/10.1145/350000/343709/p74-verkest.pdf>
19. “IEEE Std. 1364-2001 standard for the Verilog Hardware Description Language”, IEEE, Piscataway, New Jersey. – 2001.
20. “SystemVerilog 3.1a, draft 6: Accellera’s Extensions to Verilog”, Accellera, Napa, California. – 2004.

РЕАЛИЗАЦИЯ КОНЕЧНЫХ АВТОМАТОВ НА ФУНКЦИОНАЛЬНЫХ ЯЗЫКАХ ПРОГРАММИРОВАНИЯ

Я.М. Малаховски

Научный руководитель – к.т.н., доцент Г.А. Корнеев

В статье обсуждаются проблемы реализации конечных автоматов «по Шалыто» на функциональных языках программирования. На примерах небольших систем предлагаются варианты решений, имеющие преимущества перед реализациями на традиционных языках программирования (C, Pascal, ...).

Ключевые слова: конечные автоматы, функциональное программирование, *haskell*

1. Введение

В данной работе описаны несколько методов реализации конечных автоматов [1] в функциональных языках программирования. Работа носит практический характер и предлагает по-новому взглянуть на привычные вещи.

Для примеров в этой работе используется язык *Haskell* [2–5], так как он достаточно популярен в академической среде; в отличие от иных подобных языков, близок к *типизированному лямбда исчислению* (отсутствие побочных эффектов); не нарушает функциональные концепции для осуществления ввода-вывода; имеет несколько качественных компиляторов.

Основное отличие функциональных программ от императивных заключается в том, что функциональная программа представляет собой некоторое выражение (в математическом смысле), а выполнение программы означает вычисление значения этого выражения (в данном случае слово «вычисление» вовсе не означает, что операции производятся только над числами).

При сравнении функционального и императивного (традиционного) подхода к программированию можно отметить следующие свойства функциональных программ.

- (1) Функциональные программы не используют переменные в том смысле, в котором это слово употребляется в императивном программировании.
- (2) В частности, в функциональных программах не используется оператор присваивания.
- (3) Как следствие из предыдущего пункта, в функциональных программах нет циклов.
- (4) Выполнение последовательности команд в функциональной программе бессмысленно, поскольку одна команда не может повлиять на выполнение следующей.
- (5) Вместо циклов функциональные программы широко используют рекурсивные функции.

Из вышесказанного следует, что в функциональных программах зачастую не удается напрямую применять способы реализации конечных автоматов, используемые в традиционных языках программирования.

Тем не менее, в настоящее время не сложно найти упоминания использования конечных автоматов при разработке функциональных программ, однако область их применения, как правило, ограничена системами генерации парсеров по контекстно-свободным грамматикам и регулярным выражениям, как, например, [6, 7], а большинство этих автоматов являются «чистыми» функциями и не генерируют выходных воздействий.

Кроме того, авторы данной работы не смогли найти ни одного упоминания построения конечных автоматов в функциональных программах через написание (или

генерацию) программного кода по диаграмме состояний, хотя полученные функции могли бы валидироваться самим компилятором, а не отдельными утилитами.

Таким образом, основными задачами реализации конечных автоматов на функциональных языках программирования в данной работе являются: удобное и безопасное представление автоматов с выходными воздействиями, валидация функций переходов автоматов в процессе компиляции.

2. Построение функции переходов по диаграмме состояний

Суть предлагаемого подхода к реализации функции переходов в функциональных языках программирования заключается в представлении событий при помощи алгебраических типов данных, а состояния автомата – кортежем или структурой из всех его переменных. Продемонстрируем данный подход на примере.

2.1. Реализация на примере счетного триггера

Пусть требуется реализовать «счетный триггер», реагирующий на события. Другими словами, требуется построить конечный автомат с двумя состояниями, кодируемыми нулем и единицей, который управляется кнопкой. Каждое нажатие кнопки порождает событие x . К выходу z конечного автомата подключена лампа. Каждое событие x переводит автомат в состояние $(1 - y)$, где y – текущее состояние. В состоянии 0 выход z равен нулю, а в состоянии 1 – единице. Соответственно, каждое нажатие кнопки будет приводить то к включению лампы, то к ее выключению.

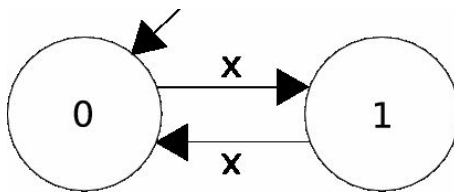


Рис. 1. Диаграмма состояний счетного триггера, 0 – состояние с выключенной лампой; 1 – с включенной; x – нажатие кнопки

Исходный код, демонстрирующий предлагаемый подход к реализации на данном примере, приведен в листинге 1.

Листинг 1. Реализация счетного триггера на Haskell

```
data Event = ButtonClick deriving Show -- Событие: «Нажатие на кнопку».
data State = LampOff | LampOn deriving Show -- Состояния: «Выключено» и «Включено».

-- Функция переходов.
gotEvent :: State -> Event -> State
gotEvent state event = case state of
  LampOff -> case event of
    ButtonClick -> LampOn
  LampOn -> case event of
    ButtonClick -> LampOff

-- Функция, вызываемая системой. Начальное состояние – LampOff.
main = putStrLn $ show $ gotEvent LampOff ButtonClick
```

Предлагаемый вариант записи функции переходов не является единственно возможным. Можно, например, использовать конструкции *pattern matching*, однако такой исходный код обычно получается длиннее, поскольку требует частичного дублирования. Поэтому в примерах используется оператор *case*.

2.2. Последовательности событий

Для того чтобы добавить возможность применения последовательности событий к начальному состоянию введем функцию *applyEvents*. Дублируемая часть кода из листинга 1 не приводится.

Листинг 2. Функция *applyEvents* и ее использование

```
-- Функция, применяющая список событий к начальному состоянию,  
-- для получения результирующего состояния.  
applyEvents :: State -> [Event] -> State  
applyEvents state [] = state  
applyEvents state (e:es) = applyEvents (gotEvent state e) es  
  
-- Функция, вызываемая системой.  
main = putStrLn $ show $ applyEvents LampOff [ButtonClick, ButtonClick]
```

2.3. Обобщение на произвольные типы состояний и событий

Если бы в реализуемом примере было более одного конечного автомата, то пришлось бы писать несколько функций, аналогичных *applyEvents*. Поэтому можно выделить общую их часть в библиотечный код (листинг 3), пригодный для написания других конечных автоматов.

Листинг 3. Библиотечная функция *applyEvents* и ее использование

```
-- Абстрактный тип функции переходов.  
type SwitchFunc state event = state -> event -> state  
-- Функция, применяющая список событий к начальному состоянию автомата  
-- при помощи функции переходов.  
applyEvents :: SwitchFunc state event -> state -> [event] -> state  
applyEvents _ state [] = state  
applyEvents switchFunc state (event:eventsTail) = applyEvents switchFunc  
  (switchFunc state event) eventsTail  
-----  
data TriggerEvent = ButtonClick deriving Show -- Типы событий  
data TriggerState = LampOff | LampOn deriving Show -- и состояний для счетного триггера.  
  
-- Функция переходов для счетного триггера.  
triggerSwitchFunc state event = case state of  
  LampOff -> case event of  
    ButtonClick -> LampOn  
  LampOn -> case event of  
    ButtonClick -> LampOff  
  
-- Функция, вызываемая системой.  
main = putStrLn $ show $ applyEvents triggerSwitchFunc LampOff [ButtonClick, ButtonClick]
```

При этом отметим, что новая версия функции *applyEvents*, является левой сверткой (одна из стандартных операций функционального программирования) списка событий по функции переходов, то есть можно заменить все определение функции *applyEvents* на *applyEvents = foldl*.

2.4. Преимущества функциональной реализации

Алгебраические типы данных позволяют производить более строгие проверки, чем конструкции сходные с конструкцией *enum* языка C. Например, при использовании алгебраических типов данных невозможно случайно проверить на равенство элемент множества состояний с элементом множества событий. Еще одним преимуществом данного подхода является то, что компилятор способен самостоятельно проверить

полноту и непротиворечивость веток оператора *case*, тем самым, осуществляя валидацию функции переходов на этапе компиляции.

3. Выходные воздействия

Функции переходов предыдущих примеров самостоятельно не производили выходных воздействий, а только возвращали результирующее состояние, которое печаталось на консоль функцией *main*. На практике же автоматам требуется воздействовать на окружающую среду значительно более широким классом выходных воздействий, например: выводить сообщения на экран, отправлять пакеты данных в сеть, обмениваться событиями.

В качестве примера реализации автоматов с выходными воздействиями рассмотрим счетный триггер с четырьмя состояниями. Его отличие от предыдущего триггера заключается в том, что лампа будет включаться или выключаться только после отпускания кнопки, а повторное нажатие или отпускание кнопки не будет производить никакого эффекта.

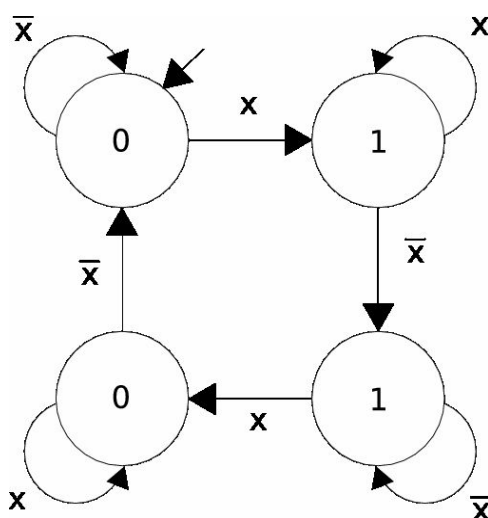


Рис. 2. Диаграмма переходов счетного триггера с четырьмя состояниями, 0 – состояния с выключенной лампой; 1 – с включенной; \bar{x} – нажатие кнопки

Общую часть реализаций примеров этой секции для экономии места вынесем в отдельный листинг 4.

Листинг 4. Общая часть реализаций счетного триггера с четырьмя состояниями

```
data TriggerEvent = ButtonDown -- Типы событий
  | ButtonUp deriving Show
data TriggerState = LampOffButtonUp -- и состояний для счетного триггера.
  | LampOffButtonDown
  | LampOnButtonUp
  | LampOnButtonDown deriving Show
```

3.1. Возврат списка выходных воздействий

Один из вариантов реализации выходных воздействий – изменить функцию переходов так, чтобы она возвращала не только новое состояние автомата, но и список выходных воздействий некоторого типа. Это может быть как арифметический тип данных, представляющий собой множество возможных выходных воздействий, так и непосредственно *IO ()*. В таком случае ответственность за выполнение выходных воздействий лежит на той части кода, которая вызывает функцию переходов автомата. Преимуществом данного подхода является его гибкость.

Листинг 5. Функция переходов, возвращающая список выходных воздействий

```
-- Абстрактный тип функции переходов,  
-- возвращающей новое состояние и список выходных воздействий.  
type SwitchFunc state input output = state -> input -> (state, [output])  
  
-- Функция applyEvents, реализованная через свертку.  
applyEvents :: SwitchFunc state input output -> state -> [input] -> (state, [output])  
applyEvents switchFunc state events = foldl (switchToAcc switchFunc) (state, []) events  
  
-- Функция «конвертирующая» функцию переходов в аккумулятор выходных воздействий.  
switchToAcc :: SwitchFunc state input output -> (state, [output]) -> input -> (state, [output])  
switchToAcc switchFunc (state, output) event = (nstate, output ++ noutput)  
  where (nstate, noutput) = switchFunc state event  
  
-- Функция переходов для счетного триггера.  
triggerSwitchFunc state event = case state of  
  LampOffButtonUp -> case event of  
    ButtonDown -> (LampOffButtonDown, [])  
    _ -> (state, [])  
  LampOnButtonUp -> case event of  
    ButtonDown -> (LampOnButtonDown, [])  
    _ -> (state, [])  
  LampOffButtonDown -> case event of  
    ButtonUp -> (LampOnButtonUp, [putStrLn "LampOn"])  
    _ -> (state, [])  
  LampOnButtonDown -> case event of  
    ButtonUp -> (LampOffButtonUp, [putStrLn "LampOff"])  
    _ -> (state, [])  
  
-- Функция, вызываемая системой.  
main = do  
  sequence_ o  
  putStrLn $ show $ s  
  where  
    (s, o) = applyEvents triggerSwitchFunc LampOffButtonUp  
      [ButtonDown, ButtonUp, ButtonDown, ButtonUp]
```

3.2. Непосредственное использование IO в функции переходов

Вторым предлагаемым решением проблемы реализации выходных воздействий у автоматов является модификация функции переходов так, чтобы она в качестве возвращаемого значения имела *IO state*, где *state* – тип состояния автомата, а сами выходные воздействия выполнялись бы непосредственно самой функцией переходов (листинг 6).

Такой подход предоставляет больше свободы по отношению к предыдущему, однако функция переходов уже не является чистой функцией. Данный способ реализации является аналогом подхода используемого в традиционных языках программирования, поскольку все вычисления выполняются строго в контексте *IO*.

4. Заключение

В представленной работе были предложены несколько наиболее удачных, по мнению автора, подходов к реализации конечных автоматов «по Шалыто» на языке *Haskell*. Преимуществом данных подходов перед реализациями на традиционных языках программирования является строгая типизация составных частей конечного автомата и валидация функции переходов компилятором.

```

Листинг 6. Функция переходов, возвращающая список выходных воздействий
-- Абстрактный тип функции переходов, возвращающей новое состояние и осуществляющей IO.
type SwitchFunc state input output = state -> input -> IO state

-- Функция applyEvents, реализованная через монаду IO
applyEvents :: SwitchFunc state input output -> state -> [input] -> IO state
applyEvents switchFunc state [] = return state
applyEvents switchFunc state (event:eventsTail) = do
    newstate <- switchFunc state event
    applyEvents switchFunc newstate eventsTail

-- Функция переходов для счетного триггера.
triggerSwitchFunc state event = case state of
    LampOffButtonUp -> case event of
        ButtonDown -> return LampOffButtonDown
        _ -> return state
    LampOnButtonUp -> case event of
        ButtonDown -> return LampOnButtonDown
        _ -> return state
    LampOffButtonDown -> case event of
        ButtonUp -> do
            putStrLn "LampOn"
            return LampOnButtonUp
        _ -> return state
    LampOnButtonDown -> case event of
        ButtonUp -> do
            putStrLn "LampOff"
            return LampOffButtonUp
        _ -> return state

-- Функция, вызываемая системой.
main = do
    result <- applyEvents triggerSwitchFunc LampOffButtonUp
        [ButtonDown, ButtonUp, ButtonDown, ButtonUp]
    putStrLn $ show $ result

```

Дальнейшие исследования по данной тематике предполагается проводить в направлении построения автоматов, пригодных для повторного использования, путем их декомпозиции.

Литература

1. Поликарпова Н.И., Шалыто А.А. Автоматное программирование. Учебно-методическое пособие. – СПб: СПбГУ ИТМО. – 2007.
2. Bird R. Introduction to Functional Programming using Haskell. Prentice Hall. – New York. – 1998.
3. Davie A. Introduction to Functional Programming System Using Haskell. – Cambridge University Press, Cambridge. – 1992.
4. Paul Hudak, John Peterson and Joseph H. Fasel. A Gentle Introduction to Haskell 98. Haskell [Электронный ресурс] / Haskell Wiki – Электрон, дан. – Режим доступа: <http://www.haskell.org/tutorial/> свободный. – Загл. с экрана. – Яз. англ.
5. Евгений Кирпичев. Монады//RSDN Magazine – 2008. – №3.
6. Parsec [Электронный ресурс] / Haskell Wiki – Электрон, дан. – Режим доступа: <http://www.haskell.org/haskellwiki/Parsec> свободный. – Загл. с экрана. – Яз. англ.
7. The Parser Generator for Haskell [Электронный ресурс] / Haskell Wiki – Электрон, дан. – Режим доступа: <http://www.haskell.org/happy/> свободный. – Загл. с экрана. – Яз. англ.

ПРИМЕНЕНИЕ ШАБЛОНОВ ДЛЯ РАЗРАБОТКИ КОМПОНЕНТОВ ПРОГРАММНЫХ КОМПЛЕКСОВ СИСТЕМ ФИЗИЧЕСКОЙ ЗАЩИТЫ

А.В. Филимонов

(Московский физико-технический институт (государственный университет))

Научный руководитель – Д.Г. Ткаченко

(ЗАО «Компания Безопасность»)

В статье описывается подход к автоматизации процесса разработки программного обеспечения, позволяющий снизить затраты на создание служебного кода, необходимого для внедрения новых компонентов в существующую систему. В основе подхода лежит формализация описания компонентов и использование шаблонов для преобразования описаний в программный код.

Ключевые слова: шаблон, язык, автоматизация

Введение

При разработке программных комплексов для управления системами физической защиты (видеонаблюдение, контроль доступа, охранная и пожарная сигнализация и т.п.) часто применяется прием, когда отдельные технические средства представляются специализированными компонентами, и создается некоторая инфраструктура, обеспечивающая их взаимодействие.

В данном случае инфраструктуру программного комплекса, к которой можно отнести механизмы связи компонентов, взаимодействие с базами данных, отображение элементов пользовательского интерфейса, можно рассматривать как набор некоторых правил. Для того чтобы управляющее программное обеспечение смогло работать с новым техническим средством, требуется разработать компонент, соответствующий принятым ограничениям. С одной стороны это упрощает разработку, но зачастую объем служебного программного кода в несколько раз превышает реализацию содержательной части.

Целью работы является разработка механизма, позволяющего снизить затраты на создание компонентов программного комплекса путем автоматической генерации инфраструктурного кода на примере сетевого взаимодействия с другими частями системы.

Для решения подобного рода задач часто применяют подход метапрограммирования, который предполагает разработку специализированной программы-кодогенератора. Входными данными такой программы является формализованное описание компонента, а выходными – программный код. При этом правила преобразования описания в код оказываются жестко зафиксированы.

Подход, предлагаемый в данной работе, предполагает формализацию не только описания компонента, но и правил и ограничений, что позволяет не только решить поставленную задачу, но и существенно упростить разработку компонентов в условиях эволюционирующей инфраструктуры программного комплекса.

Автоматизация разработки компонентов программного обеспечения с использованием шаблонов

В работе решается задача автоматизации процесса разработки кода для компонентов, представляющих технические средства в рамках управляющего программного обеспечения систем физической защиты.

Работы по внедрению нового технического средства в программный комплекс

обычно начинаются со сбора требований и составления спецификации компонента, который предстоит реализовать. Если спецификация фиксируется неформально, то дальнейшая работа по ее преобразованию в программный код ложится целиком на плечи программиста. Формализация спецификации компонента позволяет использовать инструменты для генерации реализации компонента [1].

Помимо формализации описания разрабатываемого компонента необходимо каким-либо образом зафиксировать преобразование, которое нужно применить к спецификации для получения программного кода.

Итак, общий подход к решению задачи заключается в формализации описания компонента и правил и ограничений, которые порождаются реализацией программного комплекса. Формальные описания позволяют разработать инструмент, который будет создавать код компонента.

Исходную задачу можно разбить на три части.

- (1) Разработка формального описания компонентов.
- (2) Разработка формального описания правил и ограничений инфраструктуры программного комплекса.
- (3) Реализация программы, генерирующей по описанию компонента и набору правил программный код.

Описание компонента

Широкое применение для формального описания компонентов нашли языки моделирования общего назначения, в особенности унифицированный язык моделирования (Unified Modeling Language – UML). Такой подход на первый взгляд кажется удачным, но сопряжен со следующими трудностями [1].

- (1) Язык UML не имеет четко определенной семантики.
- (2) Определение семантических правил затруднено.

Более естественным представляется использование специализированно проблемно-ориентированного языка, выраженного в терминах предметной области систем физической защиты. Преимущества этого подхода заключаются в следующем.

- (1) Создание формального описания происходит в терминах, наиболее близких к тем, которыми выражены требования к компоненту.
- (2) Язык обладает ясной семантикой.
- (3) Выражения языка более конкретны по сравнению с языками общего назначения.

Все это позволяет снизить затраты на разработку формальной спецификации за счет приближения ее к неформальным описаниям в предметной области [1].

Описание компонента на языке, созданном в рамках данной работы, состоит из следующих частей.

- (1) Перечисление конфигурационных параметров технического средства. Включает в себя название параметра, его тип и параметры доступа.
- (2) Перечисление параметров состояния, которые отражают текущее состояние технического средства.
- (3) Перечисление команд, которые воспринимает устройство.
- (4) Перечисление сигналов, которые устройство может передавать по своей инициативе.

Шаблоны

Для создания инструмента генерации реализации по формальному описанию требуется зафиксировать способы преобразования описания в сущности программного кода. Для этой цели может использоваться язык шаблонов [2].

Согласно Александру, шаблон – это отношение между контекстом, силами, часто встречающимися в этом контексте, и стратегией разрешения этих сил [2, 3]. Шаблон

описывает конкретную проблему проектирования, которая часто встречается в специфичном контексте, и предлагает обобщенный подход к ее решению. Подход специфицирует роли и распределение ответственности между сущностями, которые будут использованы в обеспечении решения [4].

Контекст – это обычно некоторая предметная область, в которой часто встречается повторяющаяся задача. В данной работе для примера используется контекст взаимодействия клиентского и серверного компонента посредством инфраструктуры, предоставляемой программным комплексом.

В рассматриваемой области типичными повторяющимися задачами являются.

(1) «Чтение/модификация» – чтение и модификация параметров технического средства.

(2) «Изменение состояния» – уведомление об изменении параметра состояния.

(3) «Команда» – вызов команды устройства.

(4) «Уведомление» – получение сигнала от устройства.

Решение каждой из этих задач можно зафиксировать в виде шаблона. Описание шаблона представляет собой параметризованный прототип программного кода. Значения параметров могут быть установлены во время применения шаблона. Схематичный пример описания шаблона представлен на рис. 1:



Рис. 1. Описание шаблона «Чтение/Модификация»

Согласно Шмидту, существует тенденция в сообществе авторов и пользователей шаблонов, направленная на определение языков шаблонов вместо отдельно стоящих изолированных шаблонов [5]. Язык шаблонов – это набор шаблонов и отношений между ними, который может быть использован для систематического решения крупно гранулированных проблем [1]. Любой язык определяет набор абстракций, формирующий его словарь, и набор правил для комбинации абстракций в выражения.

Приведенные выше небольшие шаблоны могут быть использованы для определения шаблонов более высокого уровня, таких как:

(1) «конфигурируемый объект». Описывается как набор шаблонов «Чтение/Модификация», использующих конфигурационные параметры технического средства;

(2) «управляемый объект». Описывается как набор шаблонов «Команда», использующих набор команд технического средства.

Аналогично можно определить шаблоны «Объект с состоянием», «Объект-датчик» и др. Описание шаблона «Конфигурируемый объект» может иметь вид, представленный на рис. 2:

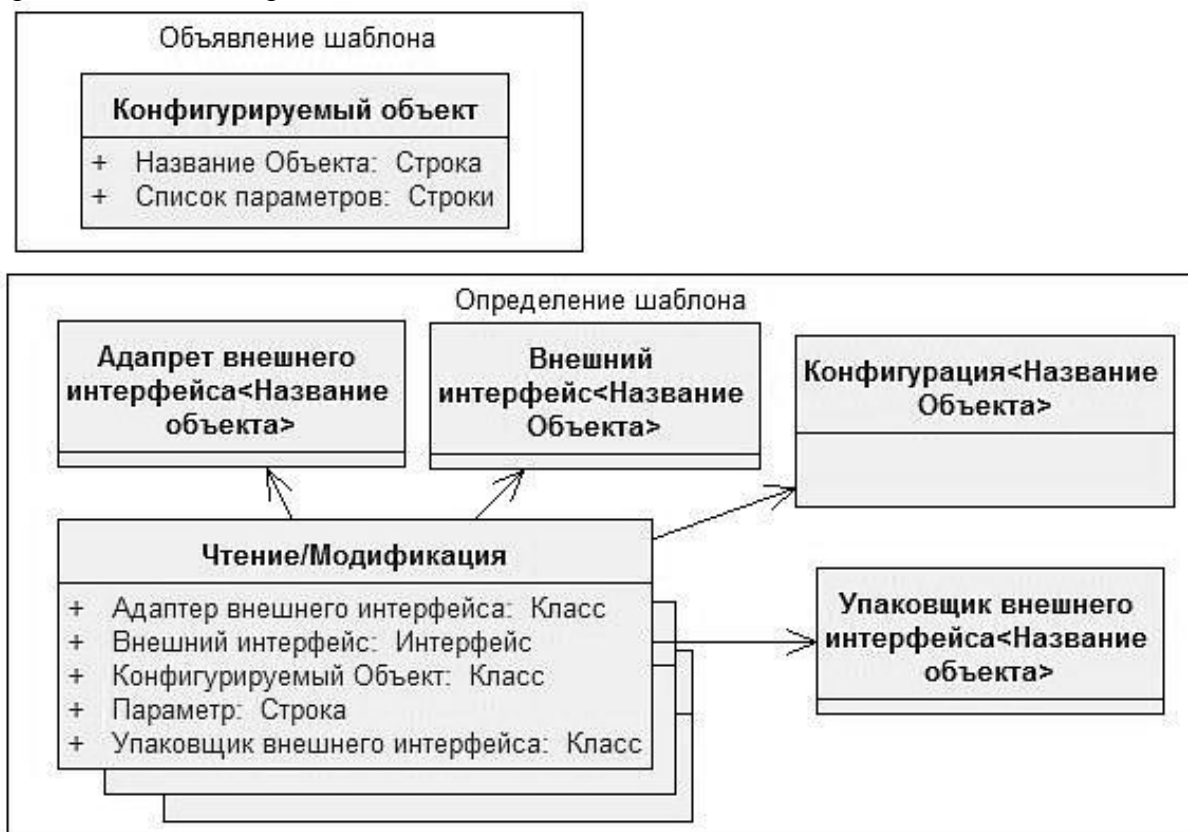


Рис. 2. Шаблон «Конфигурируемый объект» определяется через шаблон «Чтение/Модификация»

Комбинируя шаблоны высокого уровня можно определить шаблон, которые будет описывать компонент, представляющий техническое средство в программном комплексе. Так как шаблоны содержат фрагменты программного кода, требуемые для реализации тех или иных правил и ограничений, накладываемых инфраструктурой, то язык шаблонов может быть использован в качестве формального описания указанных правил. Кроме того, язык шаблонов может быть расширен путем добавления новых базовых абстракций, а также модифицирован, что позволяет снизить затраты на адаптацию его к изменениям в программном комплексе.

Применение языка описания компонентов и языка шаблонов

Изложенный подход применяется при внедрении новых технических средств в программные комплексы, разрабатываемые ЗАО «Компания Безопасность».

Компоненты, реализующие технические средства, разделяются на клиентскую и серверную части. Взаимодействие между ними осуществляется через специализированную подсистему внешних интерфейсов и контур сообщений. Подсистема внешних интерфейсов выступает в роли инфраструктуры, формирующей

правила для реализации компонентов.

Для организации чтения и модификация конфигурационных параметров по сети требуется реализовать следующие сущности.

(1) Объект, содержащий конфигурационные параметры, который может быть синхронизирован с хранилищем данных.

(2) Внешний интерфейс объекта.

(3) Адаптер внешнего интерфейса.

(4) Упаковщик внешнего интерфейса.

Для объекта типа «Видеокамера», описываемого такими конфигурационными параметрами, как название, идентификатор коммутирующего устройства, номер входа и логический номер, указанные сущности составляют около 1000 строк программного кода. В то же время описание камеры на разработанном языке составляет 10 строк. Еще 14 строк составляет выражение на языке шаблонов, описывающее создаваемый компонент. Таким образом, количество кода, который требуется создать программисту, уменьшается на порядки, что резко повысило эффективность разработки.

Заключение

В рамках работы была решена задача частичной автоматизации процесса разработки реализации программного кода компонентов программного комплекса, управляющего системой физической защиты. Для этого был предложен подход, заключающийся в создании формального проблемно-ориентированного языка для описания требований к компонентам и языка шаблонов для формализации правил преобразования спецификаций компонента в программный код.

Полученные формальные описания были использованы для создания инструмента, генерирующего программный код компонентов для подсистемы сетевого взаимодействия в рамках программного комплекса «Интеграл». Применение инструмента позволило на порядок уменьшить объем служебного кода, требующегося для внедрения новых технических средств в систему.

Разработанный механизм служит основой для проектирования проблемно-ориентированного языка для описания систем безопасности, который позволит описывать не только структуру компонентов, но и их взаимодействие в терминах предметной области систем физической защиты. Такой подход позволит повысить уровень абстракции в ходе разработки такого рода программных продуктов, что повлечет дальнейшее снижение затрат и повышение уровня качества.

Литература

1. Гринфилд Д., Шорт К., Кук С., Кент С., Купри Д. Фабрики разработки программ: потоковая сборка типовых приложений, моделирование, структуры и инструменты. Пер. с англ. – М.: ООО «И.Д. Вильямс». – 2007. – 592 с.
2. Alexander C., Ishikawa S., Silverstein M. A Pattern Language. – Oxford University Press. – 1977.
3. Alexander C. The Timeless Way of Building. – Oxford University Press. – 1979.
4. Buschmann F., Meunier R., Rohnert H., Sommerlad P., Stal M. Pattern-Oriented Software Architecture, Volume 1. A System Of Patterns. – Wiley Publishing, Inc. – 2000.
5. Schmidt D., Stal M., Rohnert H., Buschmann F. Pattern-Oriented Software Architecture, Volume 2. Patterns for Concurrent and Networked Objects. – Wiley Publishing, Inc. – 2000.

ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ

УДК 004.715

ТЕХНОЛОГИЯ ВИРТУАЛЬНОЙ ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ ЕСИМО

П.С. Лобачев, Б.М. Айтуганов

(Всероссийский научно-исследовательский институт гидрометеорологической информации – мировой центр данных)

Научный руководитель – к.т.н. В.М. Шаймарданов

(Всероссийский научно-исследовательский институт гидрометеорологической информации – мировой центр данных)

Виртуальная сеть – это новый уровень услуг сетевого взаимодействия в области интеграции территориально распределенных информационных ресурсов, систем, технологий и локальных сетей для их объединения в единую логическую сеть на основе использования сетей общего пользования.

Ключевые слова: Интернет, виртуальная сеть, передача данных, защита информации

Введение

Исторически, проблему организации надёжного канала для передачи данных решали при помощи прокладки прямого кабеля, физически защищённого от перехвата данных, например за счёт расположения его в труднодоступных местах или установки датчиков объёма, контролирующих доступ к нему. Но стоимость подобных решений была слишком высока даже для соединения близкорасположенных объектов, а в случае соединения объектов из разных частей страны или разных стран стоимость приближалась к астрономическим цифрам.

С другой стороны, конец XX века был отмечен лавинообразным распространением Интернета: в геометрической прогрессии росли скорости доступа, охватывались все новые и новые территории, практически между любыми двумя точками в мире можно было установить быструю связь через Интернет. Но передача информации не была безопасной, злоумышленники могли перехватить, украсть, изменить её. В это время стала набирать популярность идея организации надёжного канала, который для связи будет использовать общедоступные коммуникации, но за счёт применения криптографических методов защитит передаваемые данные. Стоимость организации такого канала была во много раз меньше стоимости прокладки и поддержания выделенного физического канала, таким образом организация защищённого канала связи становилась доступной средним и малым предприятиям, и даже частным лицам [1].

Виртуальная телекоммуникационная сеть ЕСИМО позволяет организациям различных ведомств, занятых морской деятельностью и географически удаленных друг от друга, создать единую логическую сеть взаимодействия различных компонент единой информационной системы, как виртуальную частную сеть VPN (*Virtual Private Network* – виртуальная частная сеть), которая обладает совокупностью средств защиты информации.

Таким образом, основным назначением технологии виртуальной телекоммуникационной сети ЕСИМО является защита информации, передаваемой по открытым каналам связи, а также защита сегментов ЛВС от проникновения извне.

Основные задачи технологии

Изначально стратегия разработки виртуальной телекоммуникационной сети ЕСИМО была направлена на решение следующих основных задач:

- (1) создание виртуальной телекоммуникационной сети ЕСИМО на базе существующих сетей общего пользования с заданными характеристиками транспортного потока для взаимодействия основных информационных служб и систем различных ведомств;
- (2) включение информационно-вычислительных ресурсов участников ЕСИМО в единое информационное пространство;
- (3) обеспечение доступа пользователей к открытым информационным ресурсам ЕСИМО через Российский ИНТЕРНЕТ;
- (4) создание единой системы мониторинга узлов виртуальной сети.

Вид деятельности технологии

Технология VPN – это броня, которая защищает информацию от несанкционированного доступа. Прежде всего, информация передается в зашифрованном виде. Прочитать полученные данные может лишь обладатель ключа к шифру. Наиболее часто используемыми алгоритмами шифрования являются алгоритм 3DES, который обеспечивает тройное шифрование с использованием трех разных ключей, а так же алгоритм шифрования ГОСТ 28147-89 и длиной ключа 256 бит.

Подтверждение подлинности включает в себя проверку целостности данных и идентификацию пользователей, задействованных в VPN. Первая гарантирует, что данные дошли до адресата именно в том виде, в каком были посланы. Самые популярные алгоритмы проверки целостности данных – SHA1 и ГОСТ 28147-89. Далее система проверяет, не были ли изменены данные во время движения по сетям, по ошибке или злонамеренно. Таким образом, построение VPN предполагает создание защищенных от постороннего доступа туннелей между несколькими локальными сетями или удаленными пользователями.

Для построения виртуальной сети необходимо иметь на обоих концах линии связи программы шифрования исходящего и дешифрования входящего трафиков. Они могут работать как на специализированных аппаратных устройствах, так и на ПК с такими операционными системами как Windows, Linux, FreeBSD.

Управление доступом, аутентификация и шифрование – важнейшие элементы защищенного соединения.

Состав виртуальной сети ЕСИМО

На начало 2009 года в состав телекоммуникационной сети ЕСИМО входят 13 центров различных ведомств РФ.

- ГУ «ВНИИГМИ –МЦД» Росгидромет;
- ГУ «ГОИН» Росгидромета;
- ГУ «Гидрометцентр России» Росгидромета;
- ГУ «ААНИИ» Росгидромета;
- ГУ «ДВНИГМИ» Росгидромета;
- ФГУП «ГНИНГИ» Минобороны России;
- 373-ВМФ Минобороны России;
- ФГУ «РФИ» МПР России;
- ФГУП «Морсвязьспутник» Минтранс России;
- ЗАО ЦНИИМФ;

- ФГУП «ЦИТиС» Минобрнауки;
- НЦУКС МЧС России;
- НТЦ ОМЗ филиал ФГУП «РНИИ КП» Роскосмос;

Общая схема функционирования технологии

Виртуальная сеть (VPN) применяется для объединения территориально разнесенных компьютерных сетей в единую логическую сеть без прокладки или аренды дорогостоящих каналов связи. Созданная на основе VPN сеть позволяет передавать и обмениваться данными с гарантированной скоростью и их защитой от несанкционированного доступа.

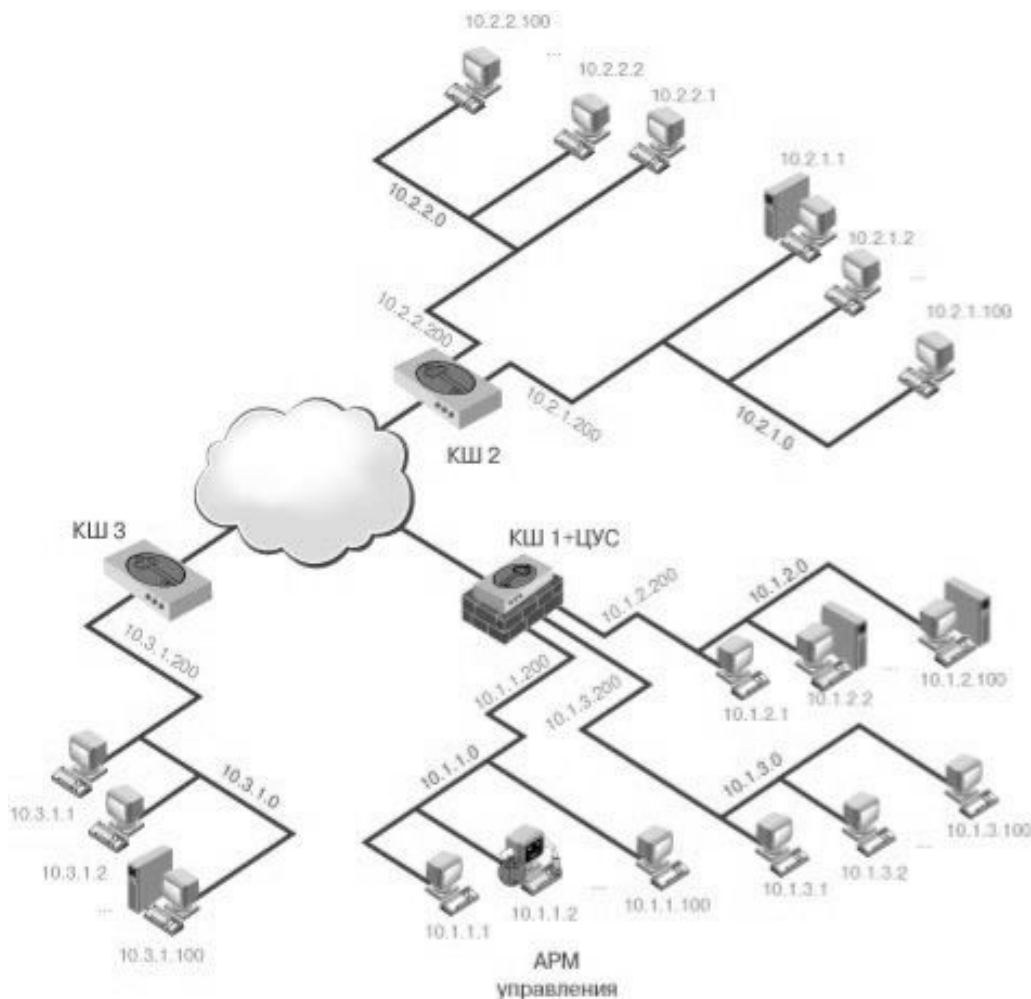


Рис. 1. Схематическое представление технологии взаимодействия в виртуальной сети (где КШ – криптографический шлюз, он же межсетевой экран)

Виртуальная телекоммуникационная сеть ЕСИМО состоит из двух частей:

- *защищенные* внутренние сети организации;
- внешняя сеть (Интернет), по которой проходит *защищенное* соединение.

Между внешней сетью и внутренней располагается межсетевой экран (на рис. 1 КШ), который обеспечивает защиту локальных сетей организации в соответствии с принятой политикой безопасности от возможных угроз исходящих из внешней сети. Несмотря на то, что коммуникации осуществляются по публичным сетям, с использованием небезопасных протоколов, за счёт криптографической защиты данных создаются закрытые от посторонних лиц каналы обмена информацией между составными частями VPN.

Технологии виртуальной сети ЕСИМО относятся к классу общесистемных технологий и поэтому для конечных пользователей и прикладных процессов, взаимодействующих в ЕСИМО, все действия криптошлюза остаются прозрачными, т.е. незаметными. Технология настраивается на аппаратных средствах телекоммуникационного узла организации системным администратором или специалистом по информационной безопасности.

Общая структура технологии

Топология виртуальной сети ЕСИМО – «звезда». Топология «звезда» характеризуется тем, что не допускает возможности двум компьютерам в виртуальной сети обмениваться данными иначе, чем как с помощью посредника – центрального сервера (центра интеграции). Исходя из этого, структура виртуальной сети ЕСИМО предполагает наличие VPN туннелей по принципу «каждый – с центром» и, соответственно, «центр – со всеми».

Технология виртуальной сети основана на обработке IP-пакетов и организации туннеля для соединения. Обработка IP-пакетов осуществляется программно-аппаратным комплексом, включающим в себя межсетевой экран и криптографический шлюз. Режим обработки IP-пакетов зависит, прежде всего, от статуса абонента сети по отношению к данному криптошлюзу (рис. 2).

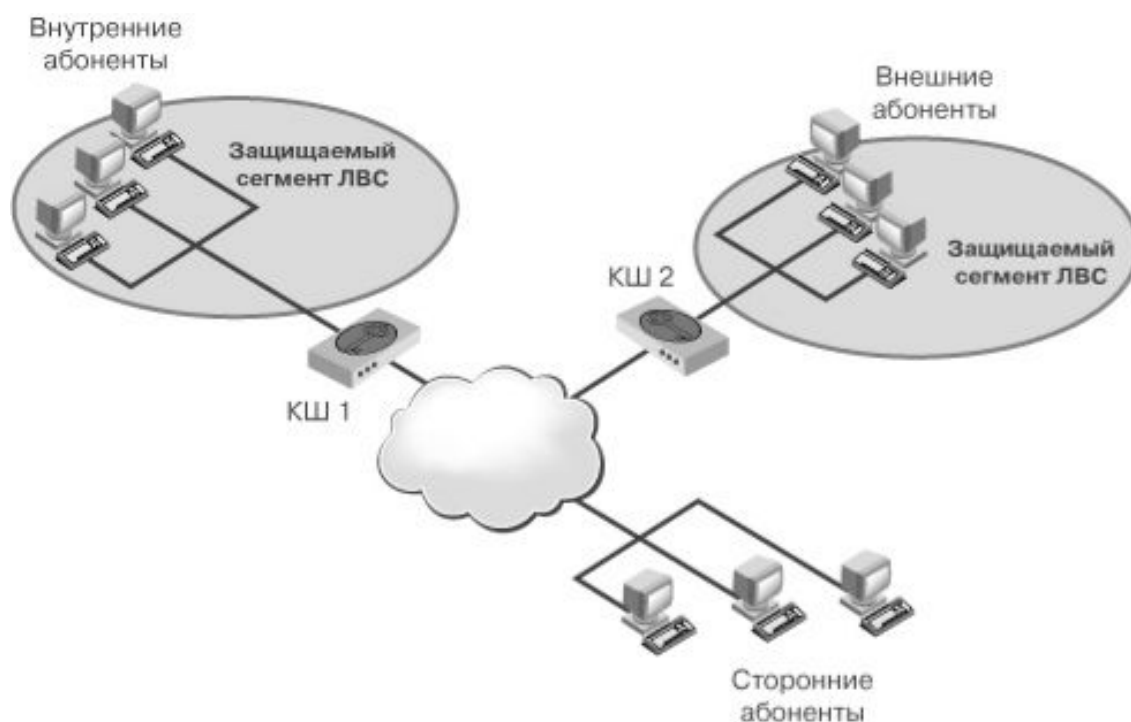


Рис. 2. Статус абонентов сети

Где «Абонент» — любой компьютер, отправляющий и получающий IP-пакеты. Абоненты IP-сети классифицируются следующим образом (на рисунке — по отношению к криптошлюзу 1):

- внутренний абонент, если он входит в состав сегмента сети, защищаемого данным криптошлюзом;
- внешний абонент, если он относится к сегменту сети, защищаемому любым другим (отличным от данного) криптошлюзом комплекса;
- сторонний абонент – любой абонент IP-сети, не входящий в состав защищаемых сегментов виртуальной сети.

Заключение

Виртуальная телекоммуникационная сеть ЕСИМО представляет собой технически сложную, территориально-распределенную на огромной территории структуру. В ее состав входит как аппаратная, так и программная составляющие. Виртуальная телекоммуникационная сеть ЕСИМО служит для объединения всех участников ЕСИМО в одну распределенную сетевую систему. Благодаря наличию этой сети участники ЕСИМО могут взаимодействовать как одно целое, что облегчает работу каждого из них, повышая при этом ее уровень.

На начало 2009 года были реализованы три основные задачи технологии. Четвертая задача, создание единой системы мониторинга узлов виртуальной сети, находится в стадии проектирования и реализации.

Литература

1. Virtual Private Network [Электронный ресурс] / Режим доступа: <http://ru.wikipedia.org>, свободный. – Яз. рус.
2. Документация на информационные технологии ЕСИМО. Требования к содержанию и оформлению. – Обнинск: ГУ «ВНИИГМИ-МЦД». – 2006.
3. Аппаратно-программный комплекс шифрования Континент. Версия 3. УВАЛ 00300-72 91 1.
4. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования данных. ГОСТ 28147-89.
5. VPN и IPSec на пальцах [Электронный документ] / Dru Lavigne. – Режим доступа: <http://www.nestor.minsk.by/sr/2005/03/050315.html>, свободный. – Яз. рус.

ОБЕСПЕЧЕНИЕ КАЧЕСТВА ОБСЛУЖИВАНИЯ ГЕТЕРОГЕННОЙ СЕТИ

С.М. Платунова

Научный руководитель – к.т.н., доцент В.А. Костеж

Статья содержит описание обеспечения качества обслуживания гетерогенной сети на примере коммутаторов второго и третьего уровня, установленных в корпоративной компьютерной сети. Показаны возможности повышения качества обслуживания за счет перераспределения ресурсов. Вариант перераспределения ресурсов зависит от решений реализованных в коммутаторах.

Ключевые слова: качество, гетерогенные сети, QoS

Введение

Обеспечение качества обслуживания (QoS) сетей является необходимостью с точки зрения выполнения целевых задач сетевого прикладного программного обеспечения. Обеспечение качества требует знания параметров, характеризующих топологию сети, особенности функционирования программно-аппаратного комплекса определяющих скорость передачи данных, задержку и потерю пакетов. Практика построения корпоративных сетей в настоящее время подразумевает создание неоднородной (гетерогенной) среды, которые состоят из различных рабочих станций, операционных систем и приложений, а для реализации взаимодействия между компьютерами используют различные протоколы. В работе под гетерогенной сетью понимаем информационная сеть, в которой физически реализованы и взаимодействуют разнородные программно – аппаратные средства.

Распространенным способом повышения качества обслуживания является программно-аппаратные способы перераспределения ресурсов.

Разнообразие всех компонентов, из которых строится сеть, порождает множество проблем проектирования, администрирования и управления такой гетерогенной интернет-сетью. В идеале это объединение неоднородных сетей должно быть прозрачным для пользователя. Без нужного уровня качества обслуживания сети невозможна корректная работа целого ряда типовых для организации программ поддержания документооборота и т.д.

Основная часть

К аппаратным способам повышения качества обслуживания можно отнести способы перераспределения ресурсов. В частности, указанным целям соответствует применение коммутаторов при создании структуры сети. В отличие от концентраторов коммутаторы имеют дополнительные технические ресурсы, заключающиеся в технически поддерживаемых решениях [1]. Например, ограничение полосы пропускания, агрегирование каналов, резервирование линий, применение виртуальных локальных сетей, применения различных средств маршрутизации между виртуальными локальными сетями.

Применение различных операционных сред при решении сетевых задач тоже при определенных условиях приводит к повышению качества обслуживания за счет специализации при выполнении сетевых задач («заточка» операционной среды под задачу). Например, коммутаторы имеют свою операционную среду, качество функционирования, включая ресурсоемкость, сетевых сервисов увеличивается в зависимости от операционной среды.

Под операционной средой будем понимать операционную систему с соответствующими настройками и дополнительными программами, при необходимости.

Ограничение полосы пропускания

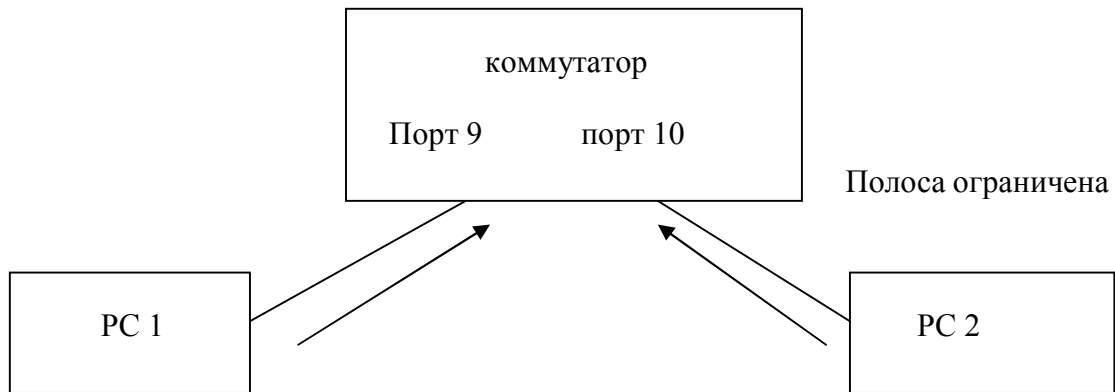


Рис. 1. Ограничение полосы пропускания коммутатором второго или третьего уровня

Коммутаторы второго и третьего уровней могут за счет перераспределения ресурсов влиять на качество обслуживания фрагмента сети.

Динамически изменяемая функция ограничения пропускной способности заключается в установке требуемой пропускной способности порта.

При уменьшении пропускной способности одного фрагмента сети, например, уменьшение пропускной способности между портом 10 и PC 2 (рис. 1), т.е. уменьшения качества обслуживания соответствующего фрагмента сети, потенциально увеличивается качество обслуживания другого фрагмента сети между портом 9 и PC 1 (рис. 1).

Реально качество обслуживания фрагмента сети (порт 9 и PC 1) увеличивается при определенных условиях распределения сетевого трафика. Указанные условия тесно связаны с неравномерностью разделения ресурсов между разными классами трафика.

Агрегирование каналов

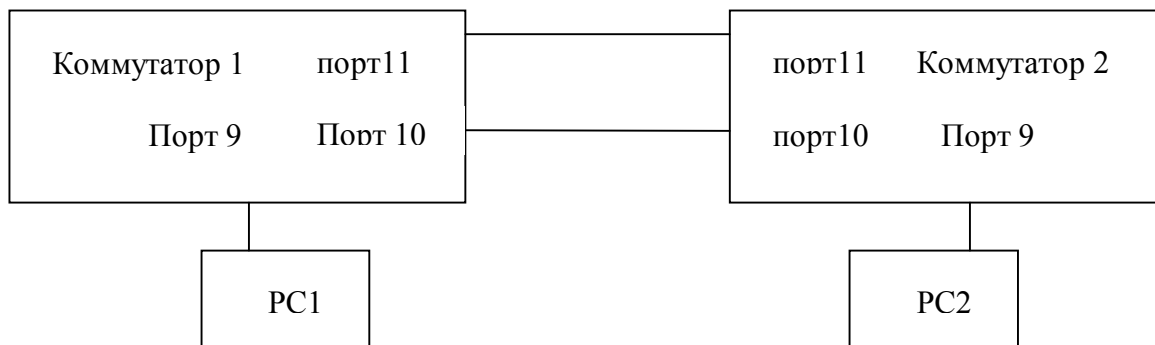


Рис. 2. Агрегирования портов в один канал

Коммутаторы могут влиять на качество обслуживания сети в целом за счет агрегирования портов (рис. 2). В этом случае происходит более существенное объединение ресурсов коммутаторов и повышается надежность каналов и поэтому, сети в целом. Например, качество обслуживания канала связи между PC1 и PC2 (рис. 2) увеличивается при агрегировании каналов между коммутаторами 1 и коммутатором 2 (рис. 2). Агрегирование каналов заключается в создании параллельных соединений (порт 11 коммутатор 1 порт 11 коммутатор 2), (порт 10 коммутатор 1 порт 10 коммутатор 2).

Резервирование линий

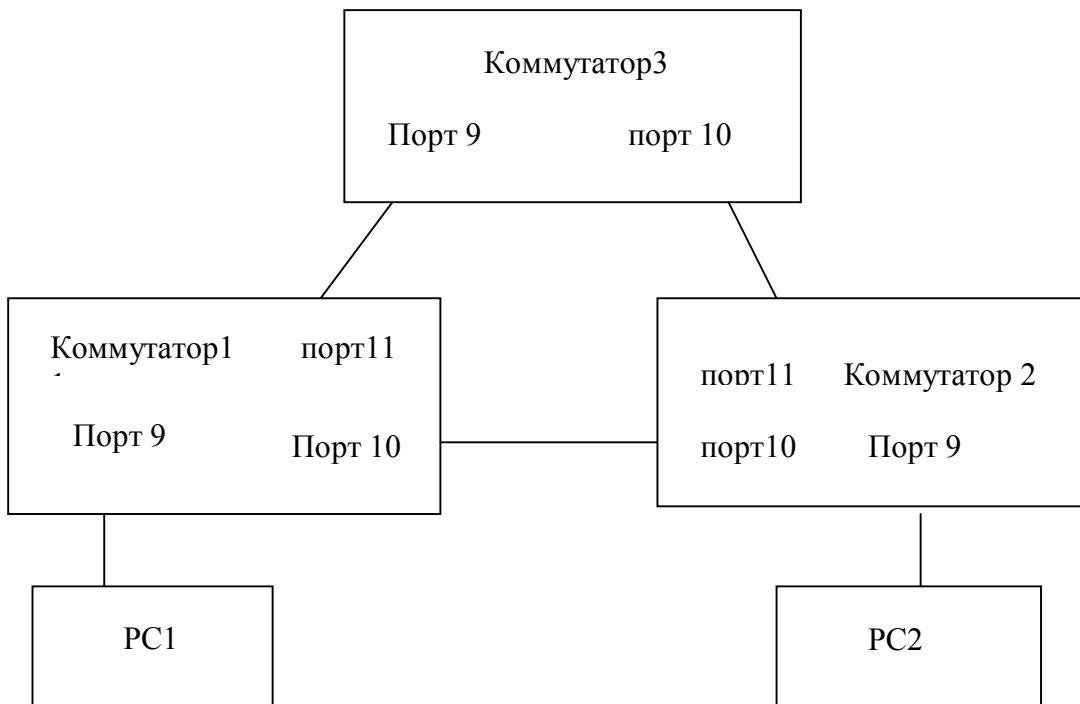


Рис. 3. Резервирование линий с помощью протокола RSTP

Аналогично с агрегированием каналов изменяется качество обслуживания сети в целом за счет протокола RSTP (рис. 3). Использование протокола покрывающего дерева RSTP (Rapid Spanning Tree Protocol) заключается в создании дополнительных резервных связей (порт 10 коммутатор 1 – порт 10 коммутатор 2) при обрыве основной (порт 11 коммутатор 1 – порт 9 коммутатор 3) или (порт 11 коммутатор 2 – порт 10 коммутатор 3).

Применение виртуальных локальных сетей

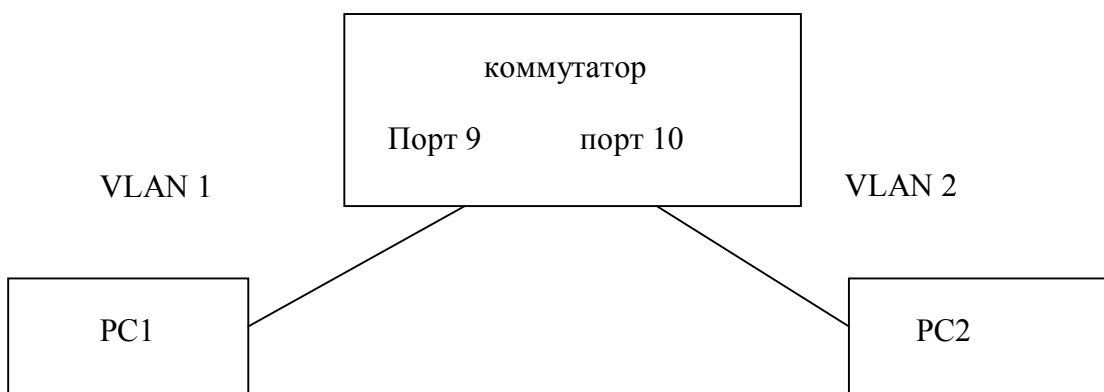


Рис. 4. Применение виртуальных локальных сетей

Качество обслуживания сети в целом при определенных условиях распределения сетевого трафика может быть увеличено за счет сегментирования сети. В качестве сегмента может выступать виртуальная локальная сеть (VLAN). На рис. 4 PC1 и PC2 относятся к разным виртуальным локальным сетям (VLAN 1 и VLAN 2).

Виртуальные локальные сети ограничивают возникновение широковещательных штормов.

Применение различных средств маршрутизации между виртуальными локальными сетями

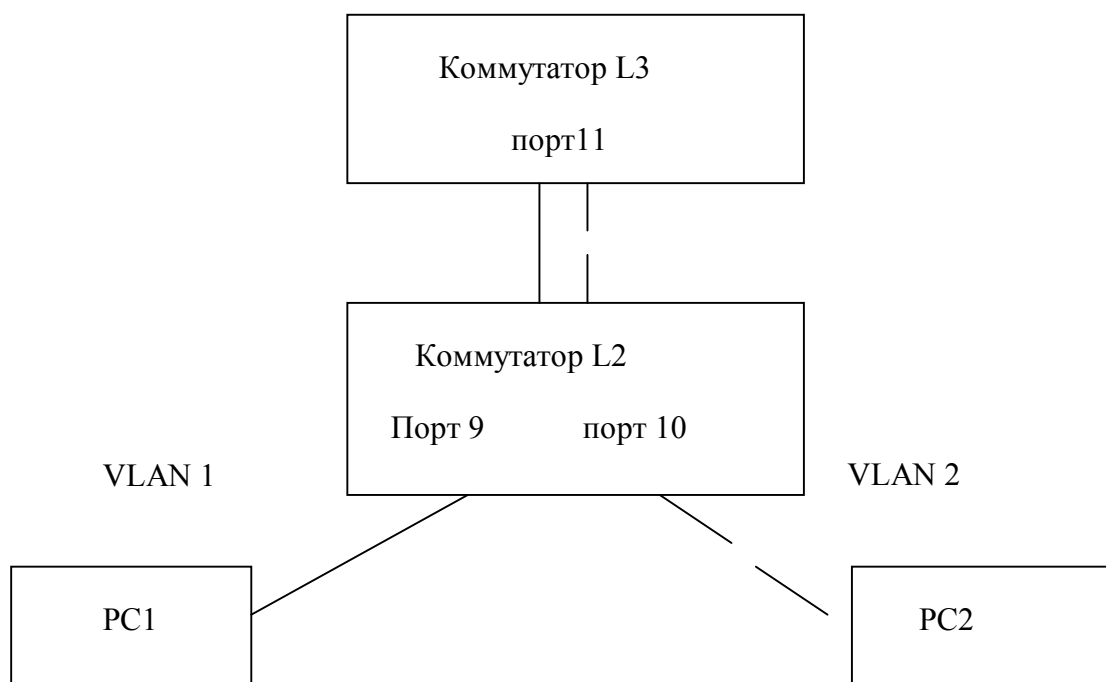


Рис. 5. Применение средств маршрутизации виртуальных локальных сетей

На рис. 5 показан один из возможных вариантов связи между виртуальными локальными сетями VLAN 1 и VLAN 2. Выбор варианта соединения виртуальных локальных сетей зависит от требований к качеству обслуживания и особенностей сетевого трафика.

Обеспечение качества обслуживания достигается использованием различных средств маршрутизации между виртуальными локальными сетями. В частности, при настроенном тегированном канале между коммутаторами второго и третьего уровней возможна связь между изолированными VLAN 1 и VLAN 2 с преодолением ограничений VLAN.

Обеспечение качества обслуживания использует различные алгоритмы управления очередями, резервирования и обратной связи, позволяющие снизить негативное влияние очередей до приемлемого уровня.

Алгоритмы управления очередями нужны для работы в периоды временных перегрузок, когда сетевое устройство не справляется с передачей пакетов на выходной интерфейс в темпе поступления [1].

Алгоритм приоритетного обслуживания применяются и для преимущественной по сравнению с другими обработки одного класса трафика. Механизм приоритетного обслуживания основан на разделении всего сетевого трафика на классы и последующего назначения каждому классу приоритета. Один из возможных вариантов обеспечения приоритетов может быть обеспечен за счет ограничения пропускной способности, показанной на рис.1. Другой вариант предлагает IEEE 802.1P.

Стандарт IEEE 802.1P определяет до 8 различных типов трафика путем добавления признака в кадр MAC уровня [2]. В соответствии с **Priority bit** в тегированном кадре IEEE 802.1Q, коммутатор классифицирует кадр в различные очереди. Рабочий процесс IEEE 802.1P аналогичен процессу IEEE 802.1Q и состоит из трех этапов **Ingress**

Process (Входной процесс), Forwarding Process (Процесс пересылки) и Egress Process (Выходной процесс). При поступлении кадра в коммутатор с включенной поддержкой 802.1P, вначале Ingress Process классифицирует полученный кадр. Затем кадр передается Forwarding Process. После Forwarding Process он поступает на Egress Process и затем отправляет получателю.

Ingress Process.

- Кадры без признака не несут никакой информации о VID и Priority. При поступлении такого кадра, Ingress Process добавляет поле признака в Ethernet кадр с значениями PVID и Default Priority.
- При получении кадра с признаком он уже содержит информацию о VID и Priority. Ingress Process пропускает такой кадр без изменения на Forwarding Process.

После Ingress Process все кадры имеют 4 байтовое поле признака и перенаправляются на Forwarding Process.

- коммутаторы ZyXEL поддерживают 8 Очередей;
- очереди и значения приоритета связываются пользователем;
- очередь с большим номером более приоритетна;
- существует два типа очередей SPQ и WFQ.

802.1P-Strict Priority Queue (SPQ) строгая очередь приоритетов работает по следующему принципу:

Данные из более низкой очереди не будут отправлены, пока более высокие очереди не будут пустыми. Таким образом, возможна потеря определенных данных, которые попали в самую низкоприоритетную очередь при загрузке более высокоприоритетных очередей.

802.1P-Weighted Fair Queue (WFQ)- тип очереди работает по следующему принципу:

Из каждой очереди последовательно отправляется определенный процент сообщений от общего числа сообщений в очередях.

Существует вид взвешенного обслуживания, как *взвешенное справедливое обслуживание* (Weighted Fair Queuing, WFQ) – пропускная способность ресурса делится между всеми потоками поровну, то есть «справедливо». Взвешенное обслуживание обеспечивает требуемые соотношения между интенсивностями трафика различных очередей только в периоды перегрузок, когда каждая очередь постоянно заполнена. Если же какая-нибудь из очередей пуста (то есть для трафика данного класса текущий период не является периодом перегрузки), то при просмотре очередей она пропускается, и ее время обслуживания распределяется между остальными очередями в соответствии с их весом. Поэтому в отдельные периоды трафик определенного класса может обладать большей интенсивностью, чем соответствующий процент от пропускной способности выходного интерфейса.

Построение сетей на основе коммутаторов позволяет использовать приоритезацию трафика, причем делать это независимо от технологии сети. Эта возможность является следствием того, что коммутаторы буферизуют кадры перед их отправкой на другой порт. Коммутатор обычно ведет для каждого входного и выходного порта не одну, а несколько очередей, причем каждая очередь имеет свой приоритет обработки.

Основным вопросом при приоритетной обработке кадров коммутаторами является вопрос назначения кадру приоритета [3]. Наиболее распространенный способ – приписывание приоритета портам коммутатора. При этом способе коммутатор помещает кадр в очередь кадров соответствующего приоритета в зависимости от того, через какой порт поступил кадр в коммутатор. Способ несложный, но недостаточно гибкий - если к порту коммутатора подключен не отдельный узел, а сегмент, то все узлы сегмента получают одинаковый приоритет. Более гибким является назначение приоритетов кадрам в соответствии со стандартом IEEE 802.1p. Этот стандарт разрабатывался

совместно со стандартом 802.1Q. В обоих стандартах предусмотрен общий дополнительный заголовок для кадров Ethernet, состоящий из двух байт.

Практическая реализация описанных выше технологий существует, например, в коммутаторах ZyXEL 3-го уровня GS4012F и 2-го уровня ES-3124.

Заключение

В работе рассмотрены варианты влияния на обеспечения качества обслуживания корпоративной гетерогенной сети фирменных решений на базе коммутаторов второго и третьего уровней.

В частности, рассмотрены варианты решений: ограничение полосы пропускания, агрегирование каналов, резервирование линий, применение виртуальных локальных сетей, применения средств маршрутизации между виртуальными локальными сетями

Литература

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – 3-е изд. СПб: Питер. – 2007. – 957 с.
2. Столлингс В. Современные компьютерные сети, 2-е издание Энциклопедия. – СПб: Питер. – 2003. – 782 с.
3. Таненбаум Э. Компьютерные сети, 4-е издание, Классика computer science. – Питер. – 2008. – 992 с.

ИССЛЕДОВАНИЕ И РАЗРАБОТКА СИСТЕМЫ ПЕРЕДАЧИ ЦИФРОВЫХ ДАННЫХ ЧЕРЕЗ ТЕЛО ЧЕЛОВЕКА

К.А. Павлов

(Московский государственный институт электронной техники)

Научный руководитель – д.ф.-м.н., профессор С.В. Селищев

(Московский государственный институт электронной техники)

Известно, что тело человека является проводником с сопротивлением, зависящим от частоты проходящего через него тока. При использовании частот порядка десятков и сотен кГц, тело можно считать проводником с сопротивлением $\approx 10 \div 100$ кОм. Таким образом, существует возможность передачи электрического сигнала через тело человека. Технология передачи данных через тело в настоящее время находится на начальной стадии развития и исследуется во многих странах. В статье представлен обзор существующих систем передачи цифрового сигнала через тело человека, а также описана разработанная нами система.

Ключевые слова: передача данных, тело человека, беспроводная технология, Body Area Network, Индивидуальная Беспроводная Связь

Введение

Назовём технологию передачи цифрового сигнала через тело человека Индивидуальной Беспроводной Связью (ИБС). Будем определять ИБС как способ беспроводной связи, при котором человеческое тело используется в качестве передающей среды для электрического сигнала. Этот способ связи позволяет передавать данные между устройствами, находящимися вблизи тела человека (в карманах, бумажнике и т.д.), также данные могут быть переданы, когда человек дотрагивается до совместимого устройства или путём рукопожатия с другим человеком, имеющим при себе аналогичное устройство.

По сравнению с технологиями беспроводной радиосвязи, например Bluetooth и ZigBee, ИБС позволяет использовать относительно низкие частоты (в пределах 1 МГц), сократить энергопотребление портативных устройств и повысить безопасность передачи данных, исключив возможность несанкционированного перехвата передаваемой информации [1].

В ноябре 2007 г. Институтом инженеров в области электроники и электротехники (IEEE) была сформирована рабочая группа 802.15.6 по разработке стандарта для технологии Body Area Network (BAN). В настоящее время лишь сформулированы технические требования предъявляемые к технологии, среди которых: максимальное энергопотребление, количество устройств в сети, количество сетей на m^2 , пропускная способность сети, время запуска, задержка, конфиденциальность и безопасность передачи данных [2].

Методы передачи данных через тело человека

Что касается практических разработок технологии ИБС, то в открытой литературе описаны и исследованы макеты опытных образцов, основанных на различных методах передачи данных через тело человека.

На рис. 1 представлены основные методы передачи данных через тело человека, описанные в литературе: простая электрическая цепь, электрическая цепь с ёмкостной связью в обратной цепи, волноводная связь и гальваническая связь. В первом методе (см. рис. 1 (а)) тело человека используется в качестве сигнального провода, цепь замы-

кается вне тела через провод между приёмником и передатчиком. Этот принцип используется, например, в приборах для измерения жировых отложений организма.

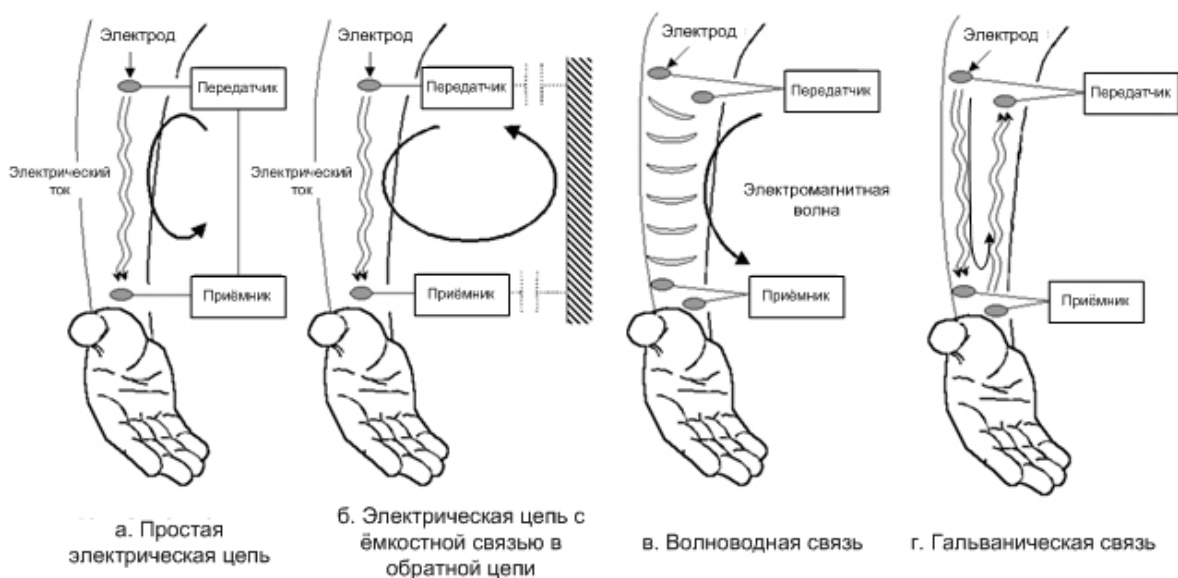


Рис. 1. Методы передачи данных через тело человека

Системе с ёмкостной связью в обратной цепи (см. рис. 1 (б)) не требуются внешние провода, но как и любой электрической цепи, системе необходим обратный путь тока для того чтобы цепь была замкнута [3]. Этот обратный путь осуществляется через ёмкостную связь с «землёй», под которой подразумеваются все окружающие предметы, находящиеся в непосредственной близости с приёмником и передатчиком. Качество связи при таком методе передачи зависит от окружающей среды, размеров и расположения электродов обратной цепи.

В литературном источнике [4] представлены конструктивные подходы к решению проблемы затухания в обратной цепи:

1. Расположение сигнального электрода на ступне, а электрода обратной цепи – на подошве обуви для заземления. Этот подход обеспечивает малое затухание в обратной цепи при условии заземления совместимого устройства.

2. Расположение устройства на запястье. В этом случае малое затухание будет обеспечено, если электрод обратной цепи электрически изолирован от тела и находится достаточно близко к электроду обратной цепи совместимого устройства. Такой способ может быть применен при передаче данных через рукопожатие.

3. Расположение электрода обратной цепи на поясе, для обеспечения большой площади поверхности.

Перечисленные конструктивные подходы позволяют уменьшить затухание сигнала в системе, но они имеют ограничения по расположению устройств на теле, а следовательно, имеют достаточно узкие применения.

В третьем методе (см. рис. 1 (в)) тело человека используется в качестве волновода для передачи радиочастотных электромагнитных волн в диапазоне частот от 1 МГц до 10 МГц. В этом случае качество передачи не зависит от окружающей среды. В работе [5] описаны эксперименты, для которых был создан фантом руки человека из поливинилхлоридной оболочки, наполненной 0,9% физиологическим раствором (см. рис. 2). Было выяснено, что характеристики фантома как передающей среды аналогичны характеристикам руки человека на частоте 10 МГц, затухание сигнала в обоих случаях составляет 26 дБ.

Электроды передатчика и приёмника на фантоме в эксперименте разнесены на 280 мм. В статье [5] нет данных о том, как будет работать система при расположении

устройств на большем расстоянии (например, в левой и правой руке). Кроме того, при использовании высоких частот передатчик создаёт электромагнитное излучение, которое уже не ограничивается телом человека. На частотах порядка нескольких сотен мегагерц длина волны сопоставима с размерами тела человека, в этом случае тело будет являться антенной для передающего устройства, излучая сигнал в пространство, а не пропуская его через себя. Для устранения этого эффекта целесообразно использовать частоты до 1 МГц [6].

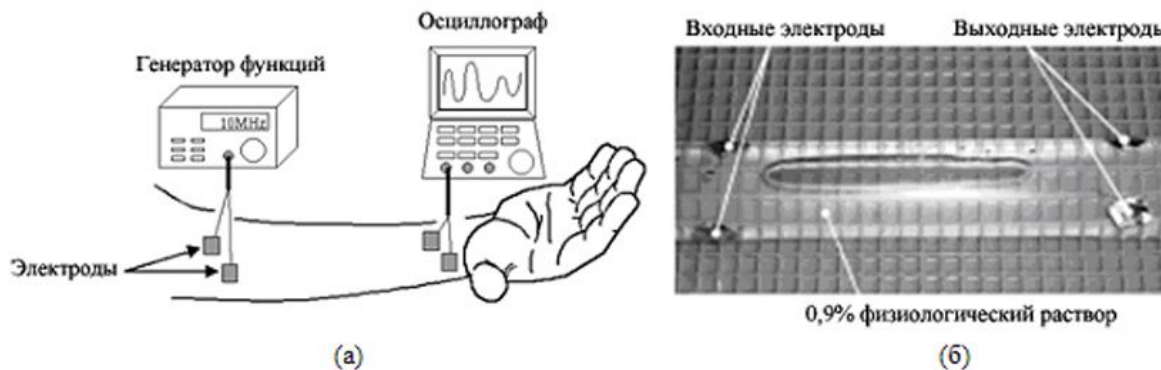


Рис. 2. Расположение электродов в системе BAN с волноводной связью на руке (а) и на фантоме руки (б)

В работе [7] представлен метод передачи данных через тело посредством гальванической связи между устройствами (см. рис. 1 (г)). Этот метод позволяет использовать более низкие частоты от 10 кГц до 1 МГц. В статье описаны эксперименты по измерению затухания сигнала в системе ИБС с различным расположением передающих и приёмных электродов. В ходе исследований было выяснено, что затухание сигнала прошедшего через сустав увеличивается в среднем на 3 дБ. Также исследования показали, что площадь приёмных электродов имеет незначительное влияние на затухание сигнала, тогда как затухание уменьшается при увеличении площади передающего электрода.

Недостатком последнего метода является высокое затухание сигнала при больших расстояниях между передатчиком и приёмником: при увеличении расстояния между электродами на руке на 5 см, затухание увеличивается на 6 – 9 дБ [7]. Такой подход может быть использован для реализации системы сбора данных с медицинских датчиков расположенных на теле человека.

Японская компания NTT DOCOMO разработала свою технологию передачи данных через тело человека под названием RedTacton. Система RedTacton задействует электромагнитное поле, создаваемое небольшим передатчиком вокруг тела человека, как вокруг проводника. Кожа соприкасается с передатчиком, испускающим слабый электрический сигнал. Поскольку его величина очень мала, для снятия сигнала на другом участке кожи используется чувствительный оптический датчик, который с помощью лазера интерпретирует флуктуации электрического поля. Эффект флуктуации лазером передается на кристалл, который является составной частью приемника. Таким образом система регистрирует пересылаемые данные. В настоящее время компания уже выпускает устройства со скоростью передачи данных 230 кб/с.

Недостатком системы RedTacton является высокая стоимость, которая вероятно связана с использованием оптического датчика. Кроме того, из-за высокой чувствительности приёмника, система RedTacton может передавать информацию не только через кожу, но и через поверхности окружающих предметов, что может привести к ложному срабатыванию приёмного устройства без контакта с телом человека.

Другая японская компания Matsushita Electric Industrial также начала выпуск устройств передачи данных через тело человека, но с меньшей скоростью – 3,7 кб/с. Пере-

датчик закрепляется у пользователя на запястье и позволяет прикосновением пальца до кассового аппарата оплачивать товар со своего банковского счёта.

Результаты наших исследований

В ходе исследований технологии ИБС были разработаны опытные образцы устройств системы ИБС, в которой затухание сигнала, как показали эксперименты, практически не зависит от окружающей среды и расположения устройств на теле. Система предназначена для передачи цифровых данных через тело со скоростью 9,6 кб/с на несущей частоте 500 кГц.

Передатчик размером 45×25×20 мм имеет 5 кнопок, при нажатии которых микроконтроллер формирует соответствующий код размером 100 байт и путём частотной манипуляции модулирует частоту 500 кГц. Приёмник размером 45×50×10 мм имеет 5 светодиодов, загорающихся при прохождении через тело сигнала, сформированного нажатием одной из кнопок передатчика. Оба устройства имеют по одному сигнальному электроду, причём электрод приёмника может находиться на одежде без контакта с кожей. Амплитуда напряжения передаваемого по телу сигнала составляет 5 В, чувствительность приёмника – 50 мкВ.

На рис. 3 представлены фрагменты демодулированных приёмником сигналов: переданного через тело (а) и переданного по воздуху (без контакта передающего электрода с кожей) (б). Как видно, при передаче сигнала по воздуху отношение сигнал/шум в системе слишком мало и в приёмник поступает лишь шумовой сигнал.

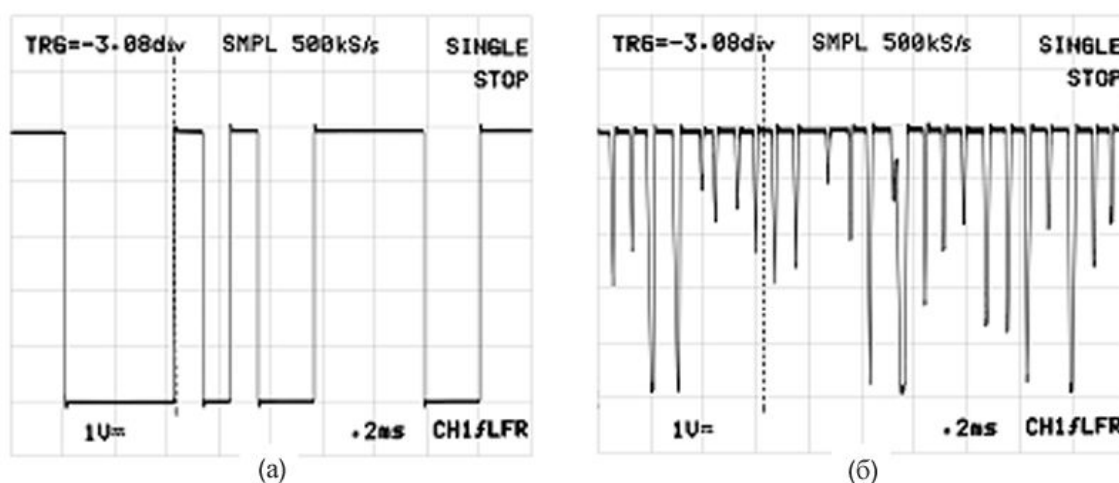


Рис. 3. Осциллограммы демодулированных сигналов: прошедшего через тело (а) и по воздуху (б)

Для оценки коэффициента затухания сигнала в системе, был проведён эксперимент с использованием селективного микровольтметра. В ходе эксперимента передатчик формировал непрерывный электрический сигнал постоянной частоты, площадь передающего электрода составляла 0,8 см². Человек рукой дотрагивался до щупа микровольтметра, передатчик постепенно перемещали по руке и далее по телу от щупа и записывали значения напряжения, измеренные прибором. Затем эксперимент повторялся для сигнала прошедшего по воздуху. Измерение проводилось при четырех частотах передаваемого сигнала: 30, 100, 500 и 1000 кГц в полосе 1 кГц.

На рис. 4 представлены графики зависимости коэффициента затухания сигнала от расстояния до передающего электрода. Коэффициент затухания рассчитывался по формуле (1), где $U_{ПРД}$ – амплитуда напряжения передаваемого сигнала, $U_{ПРМ}$ – напряжение создаваемое на щупе микровольтметра.

$$K_{зат} = 20 \cdot \lg \left(\frac{U_{ПРМ}}{U_{ПРД}} \right). \quad (1)$$

Кривые A1, B1, C1, D1 (см. рис. 4) показывают зависимость коэффициента затухания от расстояния при передаче через тело сигналов частотой 30, 100, 500 и 1000 кГц соответственно. Кривые A2, B2, C2, D2 – при передаче тех же сигналов по воздуху.

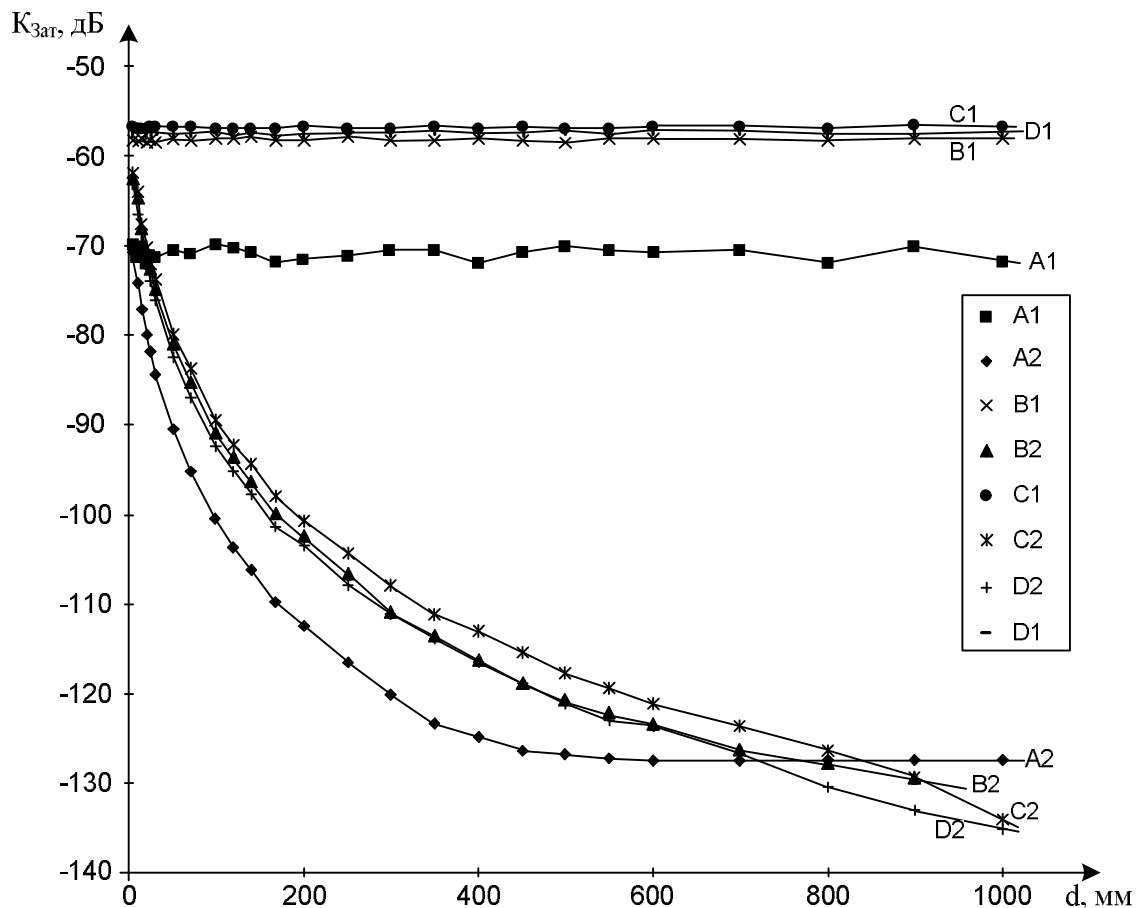


Рис. 4. Зависимость напряжения на приёмном электроде от расстояния до передающего электрода при передаче сигнала частотой 500 кГц через воздух (1) и через тело (2)

Как видно из рис. 4 коэффициент затухания при передаче сигнала через тело (кривые A1, B1, C1) остаётся практически постоянным с увеличением расстояния от передающего до приёмного электрода. По нашему мнению резкое возрастание коэффициента затухания сигнала при увеличении расстояния между передатчиком и приёмником в некоторых из описанных в данной статье систем [5, 7] связано с тем, что сигнал в них распространяется как в обычной электрической цепи, в приёмнике сигнал лишь усиливается. В нашем же случае используются частотные фильтры, 2 каскада усилителей и частотный детектор, таким образом, благодаря «полноценному» приёмнику удалось добиться постоянного значения коэффициента затухания сигнала практически между любыми двумя точками тела человека.

Выводы

Проведя обзор открытой литературы, посвященной передаче данных через тело человека, было выяснено, что существующие системы ИБС имеют крайне узкие области применения (например, компания NTT DOCOMO выпускает систему отпирания дверей касанием руки). Разработка универсальной системы ИБС, способной передавать

данные, звук, управлять электронными устройствами, является сложной научной, схемотехнической и конструкторской задачей.

В разработанной системе ИБС удалось минимизировать влияние окружающей среды на качество передачи, а также добиться независимости от расположения устройств системы на теле человека. Кроме того, устройства в разработанной системе имеют только по одному электроду, в отличие от большинства описанных в литературе систем.

В дальнейшем мы планируем оптимизировать параметры передатчика и приёмника: улучшить чувствительность приёмника, снизить амплитуду напряжения передаваемого сигнала и энергопотребление устройств. Как следствие, маломощные передатчики и приёмники с высокой чувствительностью позволят уменьшить влияние системы BAN на здоровье человека. Такая система может получить широкое применение в медицине, спорте, системах доступа и т.д.

Литература

1. P. Gupta, "Personal Area Networks: Say It And You Are Connected!," Wireless Developer Network Website Article. Режим доступа: <http://www.wirelessdevnet.com>, свободный.
2. S. Drude, "Project: IEEE P802.15 Working Group for Wireless Personal Area Networks", doc.: IEEE 802.15-06-0331. – July 2006.
3. T.G. Zimmerman, "Personal area network (PAN): Near-field intrabody communication", IBM Systems Journal. – Vol. 35. – 1996. – PP. 609–618.
4. K. Partridge, B. Dahlquist, A. Veiseh, A. Cain, A. Foreman, J. Goldberg, and G. Borriello, "Empirical measurements of intra-body communication performance under varied physical configurations," in Proc. ACM Symp. User Interface Software Technol. – 2001. – PP. 183–190.
5. K. Hachisuka, A. Nakata, T. Takeda, Y. Terauchi, K. Shiba, K. Sasaki, H. Hosaka, and K. Itao, "Development and performance analysis of an intra-body communication device", in Proc. 12th Int. Conf. Solid-State Sensors, Actuators Microsyst. (Transducers), 2003. – Vol. 2. – PP. 1722–1725.
6. N. Matsushita, S. Tajima, Y. Ayatsuka, J. Rekimoto, "Wearable Key: Device for Personalizing nearby Environment", ISWC 00.
7. M.S. Wegmueller, A. Kuhn, J. Froehlich, M. Oberle, N. Felber, W. Fichtner, N. Kuster, "An attempt to model the human body as a communication channel", IEEE transactions on biomedical engineering. – Vol. 54. – No. 10. – October 2007.

ИССЛЕДОВАНИЕ МЕТОДОВ МАРКИРОВАНИЯ ИЗОБРАЖЕНИЙ ЦИФРОВЫМИ ВОДЯНЫМИ ЗНАКАМИ

Ван Цзянь

Научный руководитель – д.т.н., профессор А.Ю. Тропченко

В статье рассматриваются методы маркирования изображений цифровыми водяными знаками. В основу предлагаемого метода маркирования положен метод *Согvi*, однако помещение вносимого цифрового водяного знака производится в спектральной области. Тем самым обеспечивается повышенная устойчивость внесенного знака при сжатии маркированного изображения или при его преобразовании. В процессе исследования маркированию подвергалось изображение типа портрет размером 480×480 точек. Маркируемое изображение подвергалось 3-х уровневому преобразованию Хаара. В качестве логотипа использовалось изображение кота размером 32×32 точки. Коэффициент α брался равным 20, что обеспечивало компромисс между невосприимчивостью ЦВЗ и его устойчивостью к атакам.

Ключевые слова: цифровой водяной знак, маркирование, устойчивость

Введение

Проблема защиты авторского права на мультимедиа информацию привела к необходимости разработки технологии защиты авторского права и технологии защиты от копирования мультимедиа информации. Одной из таких технологий является цифровое маркирование данных. При этом необходимо создать алгоритмы, которые позволяли бы «подписывать» или маркировать мультимедийные данные без потери информативности и качества воспроизведения, однако при этом в любой момент можно было бы определить, кому принадлежат авторские права на тот или иной мультимедийный продукт. Данные, скрыто встроенные в мультимедийные продукты называют «цифровым водяным знаком» – ЦВЗ «1,2,3,4,5,6,7,8,9,10».

Задача встраивания и выделения сообщений из другой информации решается стегосистемой, которая состоит из следующих основных элементов:

- прекодера – устройства, предназначенного для преобразования скрываемого сообщения к виду, удобному для встраивания в сигнал-контейнер. (Контейнером называется информационная последовательность, в которой прячется сообщение);
- стегокодера – устройства, предназначенного для осуществления вложения скрытого сообщения в другие данные с учетом их модели;
- стегодетектора – устройства, предназначенного для определения наличия стегосообщения;
- декодера – устройства, восстанавливающего скрытое сообщение.

Цель маркирования заключается в определении в дальнейшем:

- владельца объекта маркирования;
- изменений, произведённых над объектом маркирования;
- легальности прав на использование объекта маркирования.

В работе рассматриваются так называемые прозрачные ЦВЗ, которые обладают следующими свойствами [2]:

- устойчивостью;
- невидимостью;
- криптостойкостью;
- малой вероятностью ошибки;
- инвертируемостью;
- извлечением ЦВЗ без сравнения с оригиналом;

- большим числом внедряемых бит;
- вычислительной эффективностью.

В общем виде проблема маркирования изображений рассматривается как проблема передачи слабого сигнала малой мощности (ЦВЗ) в широкополосном сигнале (изображении) таким образом, чтобы быть визуально не восприимчивым и устойчивым к искажениям, которые могут появиться в процессе передачи информации.

Известно достаточно много методов добавления цифровых «водяных» знаков. В частотной области изображения ЦВЗ могут формироваться с использованием дискретно косинусного преобразования (DCT), преобразования Фурье (DFT) или wavelet преобразования [4, 5, 7, 9, 10] (рис. 1).

Обычно водяной знак (W) можно представить как сигнал, который накладывается на изображение через процедуру добавления [2]:

$$W = \begin{cases} w(k) \\ w(k) \in U \end{cases}, \text{ где } k \in W^d$$

W^d определяет размерность водяного знака: $d=1,2,3$ (как правило, $d=1$ для аудиосигналов, $d=2$ для неподвижных изображений, $d=3$ для видеоданных).

ЦВЗ гораздо меньше изображения, поэтому он многократно добавляется в трансформированные области изображения. Наиболее информативная часть спектра логотипа помещается в восприимчиво значимую часть сигнала, что обеспечивает устойчивость ЦВЗ. Схема маркирования изображений на основе вейвлет-преобразования показана на рис. 1.

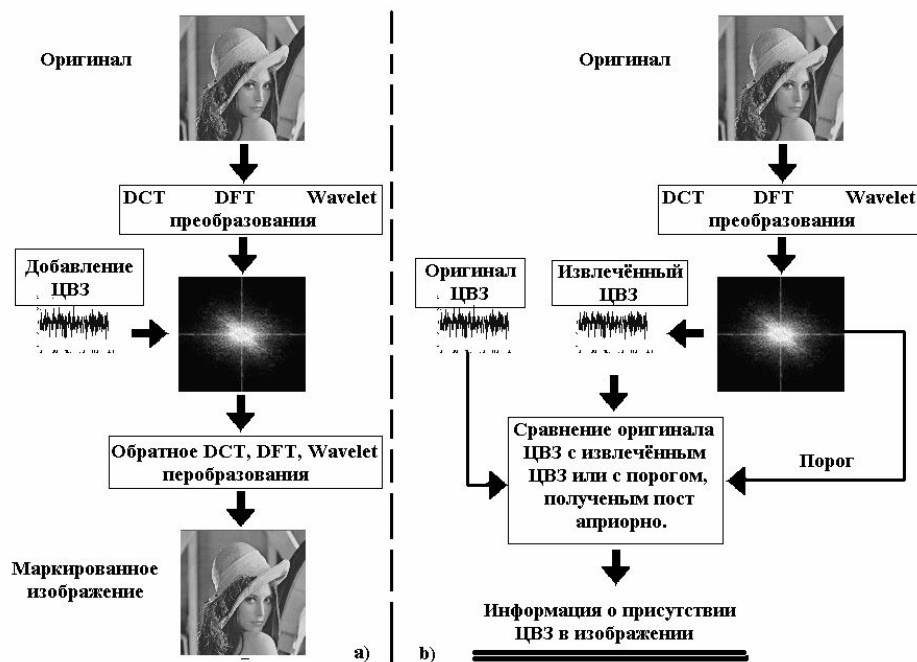


Рис. 1. Маркирование в частотной области изображения
а) добавление ЦВЗ; б) извлечение ЦВЗ

Процесс маркирования представляется тремя стадиями:

- исходное изображение подвергается вейвлет-преобразованию 3-го уровня;
- субполосы разделяются на непересекающиеся блоки и для каждого из них вычисляется величина «всплеска» S , который представляет собой числовое значение восприимчивости каждого из блоков;
- к изображению применяется обратное вейвлет преобразование 3-го уровня для получения маркированного изображения.

Модифицированный метод маркирования Corvi

Для осуществления маркирования в низкочастотной области сигнала цифровым водяным знаком в виде логотипа разработан модифицированный метод маркирования на основе методов Corvi и миксинг системы [3, 6]. Данный метод маркирования разрабатывался с целью его совмещения с методами сжатия изображений на основе вейвлет преобразований.

В начале рассмотрим метод преобразования черно-белого логотипа, выступающего в качестве ЦВЗ, в шум или в псевдослучайную последовательность. Обычно логотип представляет собой картинку, которая неоднородна, т.е. имеются большие области чёрного и белого, и при добавлении в изображение он может исказить его. Поэтому необходимо распределить ЦВЗ по всей поверхности изображения, превращая его в «шум» [3].

Для этого можно использовать пространственное преобразование плоской области U , определяемое формулой:

$$r' = Ar \pmod{M}, \begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} \pmod{M}, \quad (1)$$

где $a_{ij} \in \mathbb{Z}$, $\det A = 1$ и $\lambda_{1,2} \notin \{-1, 0, 1\}$ – собственные значения матрицы A ; x, y – координаты точки в логотипе; r – положение точки в логотипе до преобразования; r' – положение точки в логотипе после преобразования; M – размер логотипа.

Итерационное воздействие матрицей « A » на точку $r_0 \in U$ образует динамическую систему. Эту систему можно представить следующим образом:

$$r_{n+1} = Ar_n \pmod{M}, \quad (2)$$

где $n=0, 1, 2, \dots$. Рассмотрим одно из таких преобразований с матрицей « A » вида: $(a_{11}=a_{12}=a_{21}=1, a_{22}=2)$. В качестве логотипа используется изображение «кота».

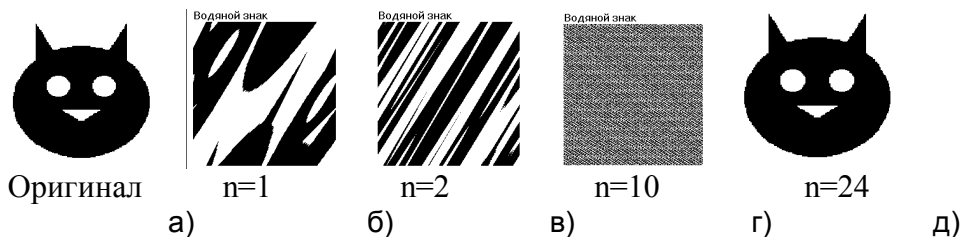


Рис. 2. Динамическое изменение логотипа под воздействием матрицы A

На рис. 2 изображены различные виды одного логотипа после итерационного воздействия на него матрицей A . Из первоначального состояния (рис. 2(а)) после первой итерации логотип преобразуется в рис. 2(б). После десятой итерации логотип преобразуется в псевдослучайную последовательность (рис. 2(г)). Если преобразование итерационно повторить 24 раза, то логотип вернётся в первоначальное состояние (рис. 2(д)).

Обычно в преобразовании подобного типа используют одну из стандартных матриц A , записываемую в виде:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} \pmod{M}, \quad (3)$$

где $k \in [1, M) \subset \mathbb{Z}$. Подобные преобразования являются периодическими, т.е. существует номер итерационного цикла T , такой, что $r_0 = r_T$ (период подбирается опытным путём для каждого логотипа).

В предложенном ниже методе маркирования данный способ преобразования плоской области используется для равномерного распределения логотипа по всей маркируемой плоскости. В процессе маркирования, маркируемое изображение подвергает-

ся 3-х уровневому вейвлет преобразованию Хаара. Маркируется только низкочастотная, аппроксимирующая составляющая (LL подуровень).

Цифровой знак представляет собой чёрно-белый логотип. Размер логотипа не должен превышать размеры аппроксимирующей составляющей вейвлет преобразования. Если размеры логотипа меньше размеров LL подуровня, то размер холста логотипа увеличивается до размеров LL подуровня, а сам логотип остается в центре этого холста.

К каждой точке полученного ЦВЗ, применяется алгоритм перемешивания, описанный выше. Преобразование 3 применяется к ЦВЗ n раз. Число n подбирается экспериментально, (преобразование повторяется до тех пор, пока спектральная плотность логотипа не станет постоянной). В итоге, получается изображение вида:



Рис. 3. Равномерно распределённый ЦВЗ

Процесс маркирования можно представить следующим образом:

$$f'(x, y) = f_{mean} + [(f(x, y) - f_{mean}) + ((1 + w(x, y)) \cdot \alpha)] \quad (4)$$

где f_{mean} – среднее значение коэффициентов LL подуровня; $f'(x, y)$ – маркированный коэффициент LL подуровня в позиции (x, y) ; $f(x, y)$ – исходный коэффициент LL подуровня в позиции (x, y) ; $w(x, y)$ – отчёт цифрового знака в позиции (x, y) ; α – коэффициент, определяющий силу добавления ЦВЗ.

Затем, для получения маркированного изображения, к полученным коэффициентам применяется обратное вейвлет преобразование.

При извлечении ЦВЗ требует оригинал LL подуровня. Процедура извлечения ЦВЗ обратна процедуре добавления.

После извлечения ЦВЗ $w(x, y)$, к нему применяется преобразование (3) T - n раз. Где n – число «перемешиваний» применяемое к логотипу перед маркированием, а T – количество итерационных циклов (постоянная величина для данного логотипа).

Предложенный метод маркирования показывает хорошую устойчивость к атакам различного типа. Хотя для восстановления ЦВЗ метод требует оригинал изображения, хранить этот оригинал целиком нет необходимости. Поскольку маркируется только аппроксимирующая составляющая вейвлет преобразования, то и для восстановления ЦВЗ потребуется оригинал только аппроксимирующей составляющей. Этот факт позволяет значительно сэкономить на хранении оригиналов изображений, поскольку даже при 3-х уровне преобразовании размер аппроксимирующей составляющей меньше полного размера изображения в 64 раз.

Устойчивость к атакам модифицированного метода Corvi

В процессе исследования маркированию подвергалось изображение типа портрет размером 480×480 точек. Маркируемое изображение подвергалось 3-х уровневому преобразованию Хаара. В качестве логотипа использовалось изображение кота размером 32×32 точки. Коэффициент α брался равным 20, что обеспечивало компромисс между невосприимчивостью ЦВЗ и его устойчивостью к атакам.

Качество восстановленного ЦВЗ из маркированного изображения, сжатого JPEG-ом с различным значением quality factor.



Рис. 4. Сравнение качества извлечения ЦВЗ из маркированного изображения, сжатого JPEG алгоритмом различным качеством
 а) QF=10, б) QF=30, в) QF=40, г) QF=50, д) QF=60, ж) QF=70, з) QF=80

Из рис. 4 видно, что при сжатии JPEG-ом до значения QF=30, логотип уверенно извлекается и визуально различим. Сжатие изображений с QF меньшим 30 приводит к достаточно сильному искажению изображения и широко не применяется в коммерческих целях. Поэтому можно заключить, что модифицированный метод Corvi устойчив к JPEG сжатию на достаточно широком интервале QF.

Качество восстановленного ЦВЗ из маркированного изображения, зашумлённого Гауссовым шумом.

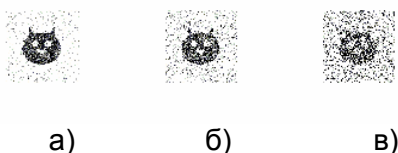


Рис. 5. Сравнение качества извлечения ЦВЗ из маркированного изображения, зашумлённого Гауссовым шумом:
 а) с уровнем шума 1, б) с уровнем шума 2, в) с уровнем шума 5

Из рис. 5 видно, что до уровня шума равного 5 ЦВЗ хоть и извлекается достаточно зашумленным, но визуально различим. Нужно отметить, что при зашумлении уровнем шума от 5 и более изображение уже теряет свою коммерческую ценность. Поэтому можно заключить, что модифицированный метод Corvi устойчив к зашумлению маркированного изображения при уровнях шума до 5.

Качество восстановленного ЦВЗ из маркированного изображения, при его масштабировании.



Рис. 6. Сравнение качества извлечения ЦВЗ из маркированного изображения, при масштабировании маркированного изображения (сжатие) на: а) 10%, б) 15%, в) 25%, г) 50%

Из рис. 6 видно, что ЦВЗ извлекаем и визуально различим на достаточно широком диапазоне масштабирования.

Выводы

Для совмещения реализованных алгоритмов сжатия изображений на основе вейвлет преобразований с алгоритмами защиты авторских прав, наиболее подходят методы вейвлет маркирования подобные модифицированному методу Corvi. Именно в этом

методе цифровая подпись помещается в восприимчиво значимую часть преобразованного изображения, которую рассмотренные алгоритмы сжатия стараются сохранить наиболее полно, а также при восстановлении ЦВЗ требуется не оригинал изображения, а малая его часть. Другие же алгоритмы маркирования помещают цифровую подпись в шумовые составляющие преобразованного изображения. Эти составляющие сжимаются наиболее сильно, что приводит к существенной потере информации в них, так что даже алгоритмы маркирования, извлекающие ЦВЗ на основе статистических расчётов, допускают ошибки в определении цифрового знака.

Таким образом, можно заключить, что алгоритмы маркирования подобные модифицированному методу Corvi, скрывающие цифровую подпись в восприимчиво значимой части преобразованного изображения на основе вейвлет преобразования обладают достаточной устойчивостью к методам цифровой обработки сигнала, и могут быть адаптированы для применения совместно с алгоритмами сжатия изображений.

Литература

1. Deepa Kundur and Dimitrios Hatzinakos. "Digital watermarking using multiresolution wavelet decomposition."// In Proceedings of IEEE ICASSP '98. – Volume 5. – PP. 2969–2972, Seattle, WA, USA. – May 1998.
2. George Voyatzis and Ioannis Pitas "The use of watermarks in the protection of digital multimedia products."// Proceedings of the IEEE, Special Issue on Identification and Protection of Multimedia Information, 87(7):1197–1207. – July 1999.
3. George Voyatzis and Ioannis Pitas. "Digital image watermarking using mixing systems."// Computer & Graphics, 22(4): 405–416. – August 1998.
4. Ingemar J. Cox, Joe Kilian, Tom Leighton, and Talal G. Shamoan. "Secure spread spectrum watermarking for multimedia."// Technical report, NEC Research Institute, Princeton, USA. – October 1995.
5. Martin Kutter, Frederic Jordan, and Frank Bossen. "Digital signature of color images using amplitude modulation."// In Ishwar K. Sethi, editor, Proceedings of the SPIE Conference on Storage and Retrieval for Image and Video Databases. – Volume 2952. – PP. 518–526, SanJose, USA. – 1997.
6. Peter Meerwald "Digital image watermarking in the wavelet transform domain"// Salzburg, am 11. Janner 2001.
7. Po-Chyi Su, C.-C. Jay Kuo "An Image watermarking scheme to resist generalized geometrical transformation"// Integrated Media System Center and Department of electrical engineering System.
8. Scott Craver, Nasir Memon, Boon-Lock Yeo, and Minerva M. Yeung. "Can invisible watermarks resolve rightful ownerships?"// Technical Report 20509, IBM Research Report, July 1996.
9. Vicius Licks "On digital image watermarking robust to geometric transformation"// THESIS; B.S., Control Engineering, Pontificia University.
10. Walter Bender, Daniel Gruhl, Norishige Morimoto, and Anthony Lu. "Techniques for data hiding."// IBM Systems Journal, 35:313–336. – 1996.

АЛГОРИТМЫ ОБРАБОТКИ ИНФОРМАЦИОННЫХ ПОТОКОВ В РАСПРЕДЕЛЕННОЙ СИСТЕМЕ ХРАНЕНИЯ ДАННЫХ

Н.М. Лукьянов, А.А. Дергачев

Научный руководитель – к.т.н., профессор В.В. Кириллов

В настоящей работе описываются алгоритмы и методы обработки информационных потоков, применяемые для работы распределенной системы хранения данных в составе Интернет сервисов. Рассмотрена последовательность движения пользовательских данных внутри системы, а также описан метод выбора оптимального узла хранения с использованием ранжирования весовыми коэффициентами.

Ключевые слова: хранилище, распределенные, данные, алгоритмы, интернет, сервис, модуль

Введение

Повседневное распространение информационных и развлекательных интернет систем, ставит новые задачи перед их разработчиками. Эта статья является продолжением исследований в области обработки и хранения данных в современных Интернет сервисах. В предыдущих публикациях были приведены результаты всесторонних исследований распределенных хранилищ данных, проведен анализ качественных и количественных показателей их функционирования.

Были тщательно исследованы вопросы производительности системы, построены имитационные модели, просчитаны необходимые объемы и структура хранилища, особое внимание уделено надежности данных, с помощью вероятностной модели просчитана вероятность выхода из строя носителей информации в процессе их эксплуатации, а также рассмотрены способы резервирования дискового пространства системы [1].

В результате 1,5 годовых исследований был построен и испытан прототип проектируемой распределенной системы хранения данных (РСХД), которая была использована в открытой социальной сети студентов факультета Высшей Школы Менеджмента СПбГУ. Наблюдение за работой системы в реальных условиях позволило более тонко настроить алгоритмы работы с данными, а также предложить более продвинутые способы обработки информации.

Постановка задачи

Объём информации в Интернет постоянно растёт, усложняются алгоритмы обработки, анализа и отображения. Для оптимизации хранения и обработки данных создаются специальные системы хранения и управления данными. Они обеспечивают такие функции как одновременный доступ сотен и тысяч пользователей к массиву данных, проверку хранимых данных на корректность и целостность, разрешение или запрещение операций с данными в зависимости от прав пользователей.

Задачей распределенного файлового хранилища, как составного элемента централизованной Интернет системы, является хранение программ и данных и предоставление доступа к ним по мере необходимости со стороны сетевых пользовательских сервисов. Задачи подобного масштаба требуют единого подхода к организации работы с данными во всех узлах распределенной системы, начиная от приёма данных, сохранения и заканчивая предоставлением необходимых данных потребителю.

Вертикальное масштабирование и применение систем хранения бизнес уровня для Интернет сервисов не дает приемлемого соотношения цены/производительности [2]. В свою очередь именно программные решения с применением несложных аппарат-

ных компонентов позволяют обеспечить открытость, гибкость и масштабируемость системы.

Архитектура системы

Представляемую систему хранения данных можно рассматривать как совокупность распределенных программно-аппаратных компонентов.

Программная реализация является наиболее важной составляющей системы, поскольку идеология данной РСХД предлагает использовать не сложных аппаратных решений, а заменить их на программные аналоги (см. рисунок).

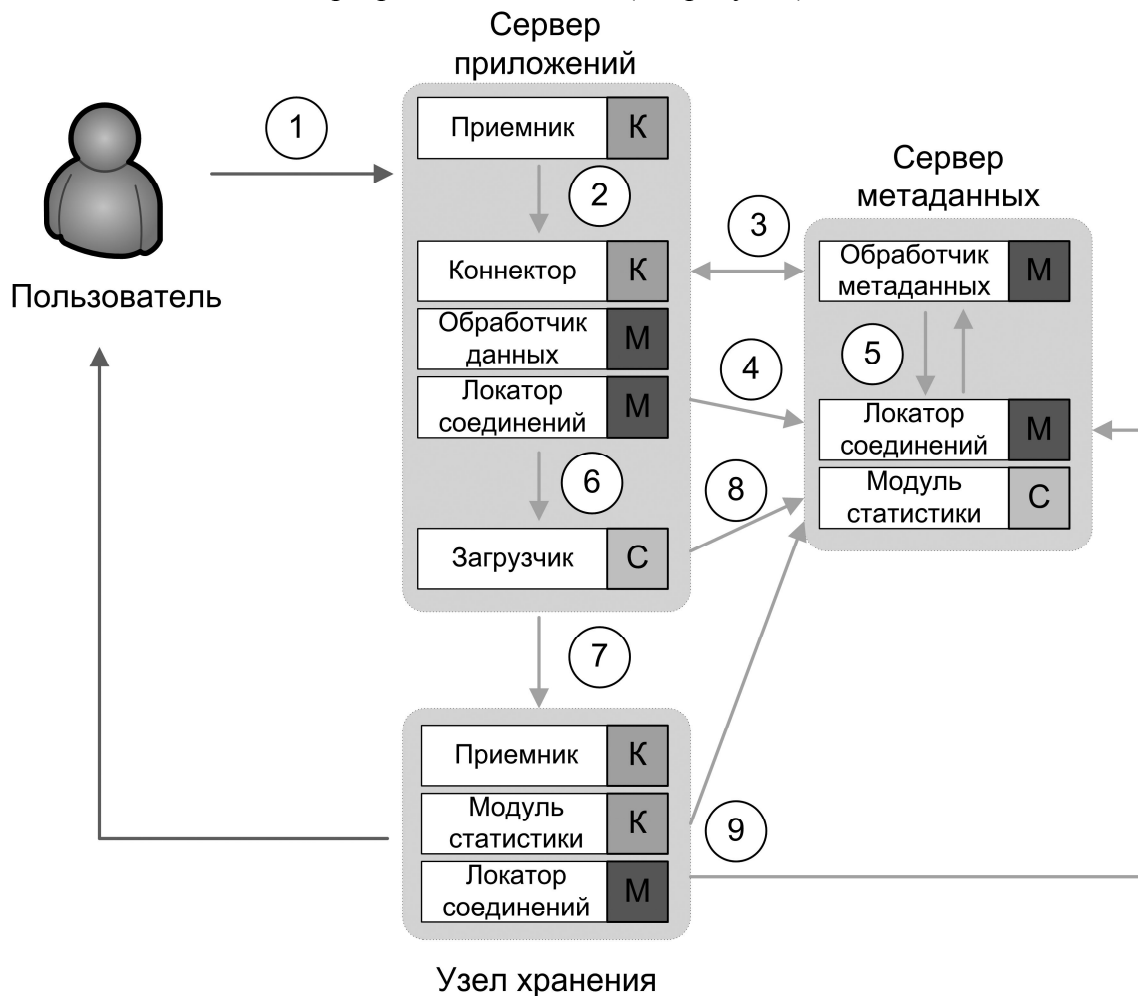


Рисунок. Программные модули обработки данных распределенной системы хранения данных

Условно всю РСХД можно разделить на 5 составляющих/

- Пользователь.
- Сервер приложений.
- Сервер метаданных.
- Узлы хранения.
- Сервер производительности (нет на рисунке).

Подобный подход является гибридным и использует составляющие всех трех разновидностей файловых систем: параллельных, симметричных и с API доступом [1].

Взаимодействие пользователя с системой происходит в двух точках: с сервером приложений, производящим прием и обработку данных для хранения, а также непосредственно с узлами хранения, предоставляющим пользователю ранее сохраненные

данные. Алгоритмы и способы работы с информацией, применяемые в представляемой распределенной системе хранения данных схематично изображены в виде модулей, которые подразделяются по своему типу на:

- клиентские модули (К) – являются сборщиками информации, для ее сохранения в БД;
- серверные модули (С) – основываясь на данных от клиентских модулей, выполняют заранее определенные действия;
- обособленные модули (М) – похожи на серверные модули, но для своей работы не требуют клиентской части.

Отдельные алгоритмы, реализованные в виде модулей, позволяют обрабатывать разнородную информацию, в зависимости от ее типа, среды передачи и условий внутри системы хранения. Происходит интеллектуальная, а не конвейерная обработка данных. Благодаря модульности, формализовав правила ввода/вывода данных, становится возможным модернизировать последовательность обработки информации путем добавления новых модулей.

Наиболее важными модулями, проводящими наибольшую часть работы по обработке данных, являются модули статистики и загрузки/приема данных.

Модуль статистики

Ключевым в работе и Интернет систем является их масштабируемость и быстрое действие. Чтобы избежать узких мест в условиях постоянных нагрузок, необходимо обеспечивать меры по оптимизации и распределению нагрузки внутри системы. Именно модуль статистики берет на себя эту задачу.

В процессе эксплуатации все узлы хранения периодически сообщают системе о своих нагрузочных параметрах. Их модуль статистики каждые 3 минуты передает в систему следующие характеристики:

- количество обработанных запросов;
- среднее время обработки запроса;
- нагрузка на центральный процессор с момента последнего опроса;
- загрузка канала передачи данных;
- емкость свободного места на дисках.

На стороне узла совершаются лишь простые действия по обработки статистических данных. Затем данные упаковываются и передаются серверу метаданных в БД.

Серверная часть модуля статистики, выполняющая большую часть операций по обработке статистических данных, располагается на сервере метаданных. Обрабатываются показатели серверов за краткосрочный период времени (последние 48 часов), поскольку не имеет смысла учитывать более старые показатели для системы, обрабатывающей текущие запросы в реальном времени.

Основными параметрами быстродействия серверов являются.

- Коэффициент нагрузки: $k_l = \frac{q_{зап} \times t_{cp}}{t_{nep}}$, где $q_{зап}$ – количество запросов, t_{cp} – средн. время обработки, t_{nep} – период опроса.
- Нагрузка процессора.
- Нагрузка канала.
- Свободное место на диске.

Первые три параметра обрабатываются путем присвоения каждому весовых коэффициентов, с учётом которых далее рассчитывается общий индекс [3]. Весовые коэффициенты, применяемые в настоящей методике, приведены в таблице:

Таким образом, путем выставления рейтинга составляется упорядоченный список готовых к приему и предоставлению данных пользователю. Именно согласно этой таблице модуль загрузки на сервере приложений размещает данные на доступных узлах хранения.

Параметр	Весовой коэффициент
Коэффициент нагрузки	0,75
Нагрузка процессора	0,1
Нагрузка канала	0,15

Таблица. Весовые коэффициенты параметров быстродействия

В случае отсутствия отчета от узла хранения за положенный промежуток времени, считается, что сервер утратил связь с системой: вышел из строя или сильно загружен. Тогда этот узел изымается из списка до тех пор, пока не будет восстановлена его нормальная работа.

В случае уменьшения лимита свободного пространства на дисках узла хранения до 10%, узел автоматически переводится в режим Только Чтение и участвует в выборе подходящего для пользователя узла хранения только для чтения данных.

Лимит в 10% на свободное место на узлах хранения введен для возможного их использования в случаях крайней необходимости, например при отсутствии свободного места на других узлах системы, а также для записи служебных данных при перемещении информации с одного узла хранения на другой.

Загрузка/прием данных

Загрузку данных в систему можно условно разделить на 4 этапа:

- прием данных от пользователя;
- обработка данных и определение узлов хранения;
- загрузка данных на узлы.

Прием данных от пользователя осуществляется по HTTP протоколу через сервер приложений. Затем с использованием данных о состоянии узлов хранения, предоставляемых сервером статистики, определяется 3 узла хранения, на которые будут загружены 3 копии данных, полученных от пользователя.

Обработчик метаданных генерирует уникальный строковый идентификатор файла, по которому можно будет однозначно определять его независимо от того, на скольких серверах он находится.

Модуль загрузки поочередно соединяется с узлами хранения. Для аутентификации на узлах хранения используются специальные ключи безопасности, которые позволяют однозначно определить сервер метаданных. После успешной аутентификации файл загружается по протоколу HTTP, и загрузчик переходит к следующему узлу хранения.

В случае отказа одной (или двух) из выбранных целей хранения выполнение задания все равно считается успешным. Информация о необходимости загрузки дополнительных копий файла передается Серверу производительности в модуль загрузки, который в дальнейшем, аналогично модулю загрузки сервера метаданных, обеспечивает необходимый уровень повторяемости данных.

Локаатор соединения

Локаатор соединений – модуль, реализующий интеллектуальный выбор узлов хранения, обеспечивающих наибольшее быстродействие обмена данными с конкрет-

ным пользователем. Так, например, в случае, если пользователь будет находиться в США, то наиболее быстрыми скоростными характеристиками обмена данными с ним, вероятно, будут обладать узлы хранения системы, также расположенные в США. Именно для определения наилучшего узла хранения для пользователя и предназначен этот модуль.

Подобные, но более статичные алгоритмы выбора зеркал уже применяются в различных Интернет системах [4]. Принцип их работы основан на определении месторасположения пользователя по IP адресу, а не на определении наилучшей скорости соединения с ними, что может привести к ошибкам системы, поскольку не всегда можно однозначно определить местоположения пользователя по его IP адресу [5].

Локатор соединения расположен на сервере приложений и узлах хранения. Статистические данные о скорости соединения пользователя с системой передаются во время загрузки данных на сервер приложений, а также при получении пользователем данных с узла хранения.

Вначале для получения данных пользователь направляется на один из наименее загруженных узлов хранения. Данные о скорости загрузки направляются на сервер метаданных. В процессе работы пользователя с системой собирается статистика и скорость соединения его IP адреса с различными узлами хранения, составляется рейтинг скорости загрузки.

В дальнейшем, при формировании ответа пользователю, происходит выборка не только наименее загруженного узла хранения, но и имеющего наилучшие скоростные характеристики обмена данными для данного пользователя.

Заключение

Применение программных реализаций для решения задач хранения данных показало свою состоятельность и перспективность. С помощью языков программирования были реализованы ранее недоступные аппаратные реализации алгоритмов резервирования данных.

В качестве основы алгоритма выбора оптимального узла хранения используется метод ранжирования с весовыми коэффициентами. Рассмотрена последовательность движения пользовательских данных внутри системы.

Гибкость и простота модульной системы позволила без труда наращивать функциональность компонентов и всей системы в целом. Так, уже на этапе эксплуатации РСХД начал внедряться новый модуль – Локатор соединений, реализующий интеллектуальный выбор доступных зеркал данных на различных узлах хранения.

Открытые интерфейсы, а также широко используемые языки программирования позволили осуществить легкую интеграцию распределенной системы хранения данных с проектом социальной сети.

В дальнейшем, по завершению всей запланированной функциональности, распределенная система хранения данных будет интегрирована с одной из популярных систем управления контентом. Так же в ближайшее время будет опубликован код системы, что позволит осуществлять совместную с открытым сообществом работу над улучшением системы, а так же сделает доступным распределенную систему хранения данных любому желающему.

Литература

1. Лукьянов Н.М. Анализ факторов, влияющих на качественные и количественные показатели функционирования систем распределенного хранения данных // Научно-технический вестник СПбГУ ИТМО №56. – 2008. – 9 с.

2. Hoff T. Google Architecture // материалы сайта HighScalability.com. – 2008. – <http://highscalability.com/google-architecture>
3. Корников В.В. Байесовская модель обработки нечисловой, неточной и неполной информации о весовых коэффициентах // Санкт-Петербургский государственный университет – 2000. – <http://www.inftech.webservis.ru/it/conference/scm/2000/session3/kornikov.htm>
4. Michael Vogt, James Troup. Dynamic Mirror Decisions // материалы компании Ubuntu. – 2006. – <https://wiki.ubuntu.com/DynamicMirrorDecisions>
5. Yokota H. A proposal of DNS-based adaptive load balancing method for mirror server systems and its implementation // Advanced Information Networking and Applications – 2004. – 208 с.

ЕДИНЫЙ РЕГИСТР НОРМАТИВНЫХ ПРАВОВЫХ АКТОВ РОССИЙСКОЙ ФЕДЕРАЦИИ

С.В. Сорокин

(Московский физико-технический институт (государственный университет))

Научные руководители:

д.ф.-м.н., профессор В.К. Исаев, д.ф.-м.н., профессор Ю.Е. Нестерихин,

д.т.н., профессор К.К. Глухарев

(Московский физико-технический институт (государственный университет))

Единый регистр нормативных правовых актов Российской Федерации – это комплекс программно-технических средств, включающий в себя набор информационных систем и информационных хранилищ по сбору, обработке, регистрации и предоставления доступа к нормативным правовым актам всех уровней: федерального, регионального и муниципального. Рассматриваются основные аспекты создания, организационная структура и архитектура решения, обосновывается значимость данной системы для информатизации органов государственной власти, построения вертикали власти в стране, механизмов эффективного правового и демократического общества.

Ключевые слова: нормативные правовые акты, федеральный регистр нормативных правовых актов субъектов Российской Федерации, муниципальный регистр нормативных правовых актов, раскрытие правовой информации

Введение

Одним из необходимых условий развития правового государства является формирование единого информационно-правового пространства, обеспечивающего полную правовую информированность граждан, юридических лиц, всех государственных и общественных структур, доступность правовых актов для всех заинтересованных лиц. В современных условиях, единое информационно-правовое пространство абсолютно необходимо для обеспечения согласованности, непротиворечивости издаваемых в стране правовых актов.

Проблема состоит в том, что государство не только не предоставляет гражданам в необходимом объеме возможность получать информацию о действующем законодательстве, но и само не располагает достаточно эффективными информационными системами по работе с правовой информацией.

На реализацию данной цели направлено создание Единого регистра нормативных правовых актов Российской Федерации (далее – Регистр).

Целью создания и развития Регистра является построение общенациональной правовой информационной системы, базирующейся на вертикальной и горизонтальной интеграции имеющихся и вновь создаваемых в стране информационных систем правовой информации различного уровня.

Опыт создания аналогичных систем

В зоне Евросоюза разработана и успешно функционирует система Eur-Lex для предоставления публичного интернет доступа к законодательству Европейского союза.

В соответствии с постановлением Правительства РФ от 29.11.2000 № 904 «Об утверждении Положения о порядке ведения федерального регистра нормативных правовых актов субъектов Российской Федерации» был разработан федеральный регистр нормативных правовых актов субъектов Российской Федерации [1, 2].

В ходе реализации первой очереди работ были продемонстрирована важность и необходимость создания общенациональной информатизации правовой системы Рос-

сийской Федерации и обеспечения публичного доступа к законам страны в соответствии с Конституцией России, осознана необходимость дальнейшего развития системы.

Для реализации этих целей в 2007-2008 году на базе Минюста России были проведены работы по модернизации существующего «Федерального регистра» и по созданию на его базе Единого регистра нормативных правовых актов с включением в банк данных кроме актов уровня субъектов РФ, документов федерального уровня: федеральное законодательство, актов федеральных органов исполнительной власти; и муниципального уровня в виде федерального муниципального регистра. В соответствии с постановлением Правительства РФ от 10.09.2008 № 657 «О ведении федерального регистра муниципальных нормативных правовых актов» с 1 января 2009 было начато формирование единого банка данных муниципальных нормативных правовых актов.

Цели регистра

Основной целью создания и развития Регистра является создание единой информационной правовой системы органов государственной власти, обеспечивающей единообразный подход к формированию, сбору, хранению, обработке, передаче и использованию правовой информации.

Указанная цель включает в себя:

- формирование и ведение автоматизированного учета нормативных правовых актов федерального, регионального и муниципального уровня с использованием единой программно-технической платформы;
- обеспечение актуальности, полноты и достоверности информации за счет своевременной фиксации внесенных изменений в акты, включенных в Систему, и оперативности пополнения баз данных;
- повышение качества услуг по предоставлению нормативной правовой информации всем категориям заинтересованных лиц и организаций;
- предоставление технологических условий для систематизации и обеспечения непротиворечивости законодательства Российской Федерации.

Объектом автоматизации является деятельность структурных подразделений Министерства юстиции Российской Федерации:

- на федеральном уровне – Центральный аппарат Минюста России, ФГУ НЦПИ при Минюсте России;
- на окружном уровне – управления Минюста России по федеральным округам;
- на региональном уровне – территориальные управления Минюста России в субъектах РФ, управления Администраций субъектов РФ.

Организационная структура системы

Работы по созданию системы были поделены на ряд связанных проектов по разработке изолированных функциональных блоков, что отразилось на организационной структуре решения. Регистр включает в себя следующие подсистемы:

- Подсистема нормативных правовых актов единой системы информационно-телекоммуникационного обеспечения Минюста России (шифр «ПС НПА ЕСИТО»). Данная подсистема обеспечивает ведение регистра нормативных правовых актов Федерального законодательства, федеральных органов исполнительной власти, субъектов РФ, а также актов, заключенных в рамках международного сотрудничества, уставов муниципальных образований.
- Федеральный муниципальный регистр (шифр «Муниципальный регистр»). В рамках данной подсистемы организовано взаимодействие с региональными

базами данных муниципальных актов в целях формирования на федеральном уровне единого банка данных муниципальных нормативных правовых актов.

- Система раскрытия информации (шифр «АИС УП. Раскрытие»). Данная подсистема представлена открытым Интернет-порталом, который обеспечивает публичный доступ ко всем нормативным правовым актам, включенным в Регистр.

Для реализации требуемой функциональности Система взаимодействует с иными информационными системами Минюста России (рис. 1). Сопряжение с данными системами позволяет передавать данные от одной системы к другой, что позволяет оптимизировать работу пользователей, за счет использования на последующих этапах данных, введенных на предыдущих этапах процесса, что позволяет избежать повторного ввода информации.

Регистр взаимодействует со следующими системами:

- Электронный архив (КСА-ЦОД).
- Ведомственный документооборот («Кодекс»).
- Муниципальный АРМ ввода НПА («Мини АРМ МО»).
- Информационно-поисковая система «Эталон Плюс».

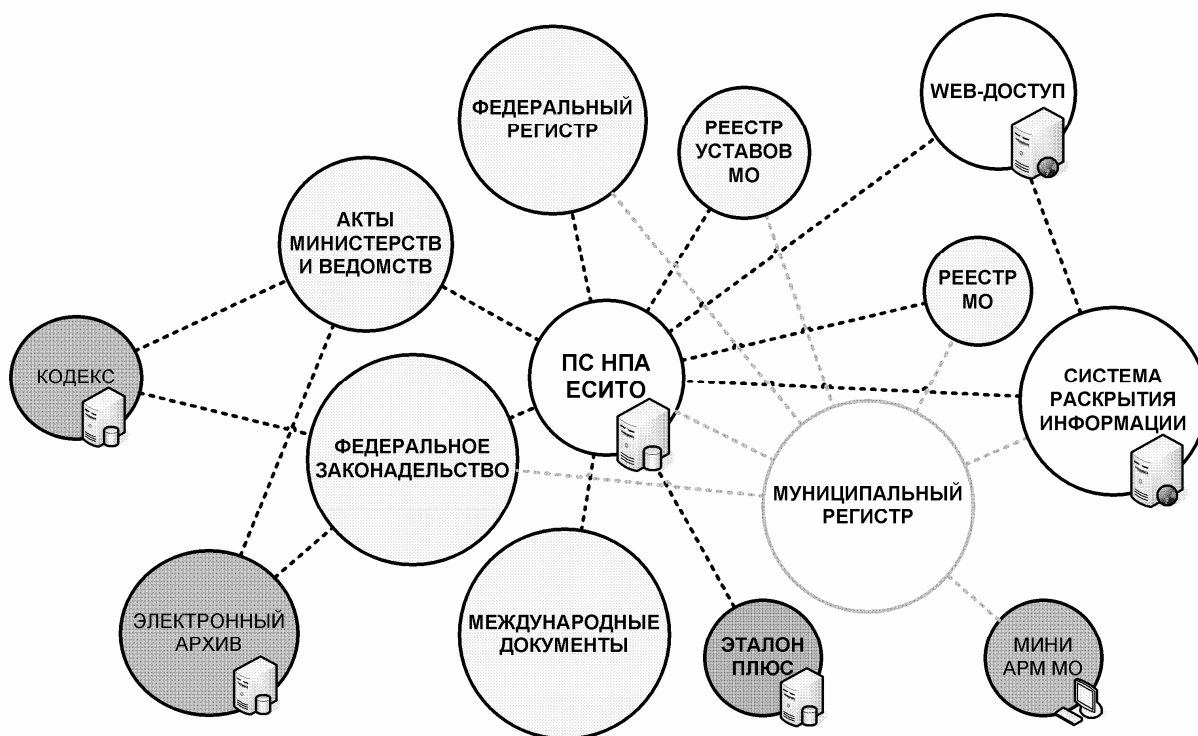


Рис. 1 Схема межсистемного взаимодействия в Едином регистре НПА

Архитектура системы

Система состоит из федерального и регионального уровней. Работа пользователя с системой включает в себя работу со специализированными автоматизированными рабочими местами (АРМ) пользователя. На каждом уровне работа с системой обеспечивается с использованием 3 АРМ-ов: АРМ ввода, АРМ аналитики, АРМ администрирования. В зависимости от уровня функционирования меняются доступные функции ПО (рис. 2). Для регионального уровня:

- АРМ ввода: ввод и ведение НПА уровня субъектов РФ, обработка НПА уровня субъектов РФ, юридическая экспертиза, регистрация уставов муниципальных образований и ведения реестра муниципальных образований;
- АРМ администрирования: ограниченное ведение классификаторов, формирование пакетов обновлений для межуровневого взаимодействия;
- АРМ аналитики, статистики и мониторинга: информационно-поисковые средства.

Для федерального уровня:

- АРМ ввода: ведение НПА Федерального законодательства, ведение ведомственных НПА, ведение международных НПА, межгосударственный обмен;
- АРМ аналитики: сопряжение с Электронным архивом, сопряжение с Электронным документооборотом, технический мониторинг и диагностика, аналитика и статистика, мониторинг, информационно-поисковые средства;
- АРМ администрирования: управление межуровневым взаимодействием, управление публикациями, ведение классификаторов.

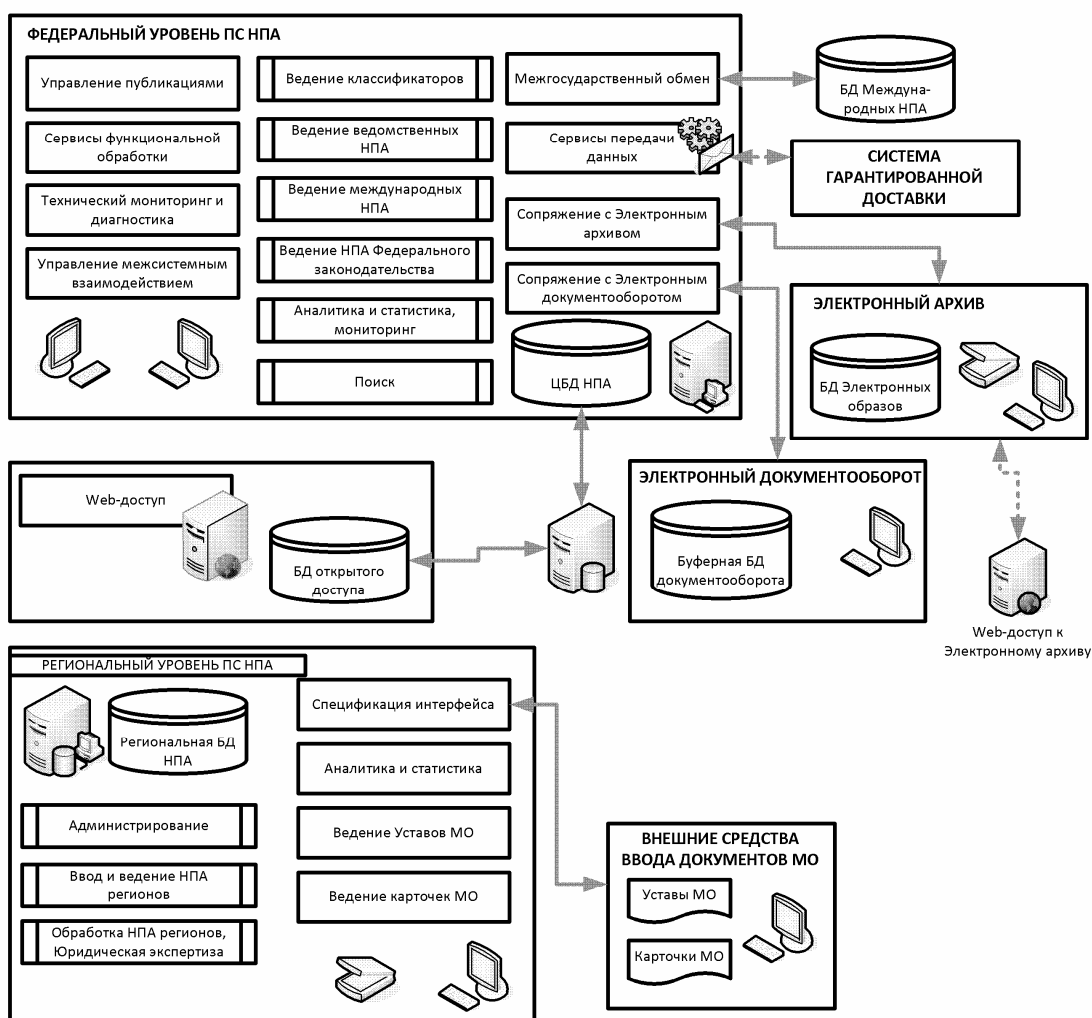


Рис. 2 Перечень и деление в системе

Система позволяет хранить нормативную правовую информацию федерального, регионального и муниципального уровней в рамках единого хранилища данных и формировать единую точку доступа к этой информации. Централизованная передача дан-

ных по уровням и средства контроля сбора данных позволяют обеспечить достоверность и целостность информации.

В рамках автоматизации процесса государственной регистрации НПА разработаны правила формирования номеров государственной регистрации по единому принципу для различных типов НПА. Система поддерживает все стадии, которые проходит регистрируемый документ с момента его поступления из документооборота, фиксирующего входящую документацию, до присвоения ему номера государственной регистрации [3].

Заключение

В течение 2007–2008 годов были разработаны и введены в опытную эксплуатацию основные подсистемы Регистра. Как было указано выше, с 1 января 2009 года в соответствии с постановлением Правительства РФ один из модулей системы – Федеральный муниципальный регистр – был введен в промышленную эксплуатацию на территории всей страны. Во втором квартале 2009 года планируется ввод в промышленную эксплуатацию всех остальных модулей Регистра: «ПС НПА ЕСИТО» и «АИС УП. Раскрытие».

Литература

1. Гукасов Н.В. О Федеральном регистре нормативных правовых актов субъектов Российской Федерации // Бюллетень Министерства юстиции Российской Федерации. – 2001. – № 6. – С. 19–22.
2. Бурмистров А.В. Федеральный регистр как средство обеспечения единого правового пространства России // Конституция Российской Федерации и развитие отраслевого законодательства. Материалы Межрегиональной научно-практической конференции, 10 декабря 2003 г. – Белгород: Изд-во Белгород. юрид. ин-та МВД России, 2003. – С. 18–21.
3. Григорьев В.Г. О проблеме дублирования норм нормативных правовых актов, принятых (изданных) на федеральном уровне, в нормативных правовых актах, принимаемых (издаваемых) на уровне субъектов РФ // Вестник Волжского университета им. В.Н. Татищева. – 2003. – № 29. – С. 63–70.

ПОВЫШЕНИЕ ОПЕРАТИВНОСТИ ИНФОРМАЦИОННОГО ОБМЕНА В СПУТНИКОВОЙ СИНГУЛЯРНОЙ КОНВЕРГЕНТНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЕ ЗА СЧЕТ ИНТЕЛЛЕКТУАЛЬНОЙ ОБРАБОТКИ ИНФОРМАЦИИ

Е.И. Ледовской, М.Ю. Татарков

(Северо-Кавказский государственный технический университет)

Научный руководитель – д.т.н., профессор Г.В. Слюсарев

(Северо-Кавказский государственный технический университет)

Представлено техническое решение, за счет которого повышается оперативность информационного обмена в спутниковой сингулярной конвергентной информационной системе путем интеллектуальной обработки информации.

Ключевые слова: сингулярность, конвергенция, оперативность, система, информация, обработка, интеллект

Введение

Повышение оперативности информационного обмена в сингулярных конвергентных инфокоммуникационных системах [1] является актуальной и важной научно-технической задачей. Процесс сингулярной конвергенции инфокоммуникационных технологий приводит к ускорению информационного обмена в таких системах. Объединение различных инфокоммуникационных технологий, в конечном счете, усиливает человеческий интеллект, придавая этому процессу собственно свойство сингулярности [2–4].

Современные достижения в области инфокоммуникационных технологий позволяют решать задачу повышения оперативности информационного обмена в сингулярных конвергентных инфокоммуникационных системах более эффективно. К ним относятся разрабатываемые в настоящее время проекты в области искусственного интеллекта такие как «Examining Obstacles to Software Developer Adoption of Statistical Machine Learning», «CueFlik» [5] и др., в рамках которых ведётся изучение методов статистического обучения интеллектуальных систем, интерактивного обучения при поиске изображений и т.п. Так, например, известны интеллектуальные системы [6, 7], основанные на технологии Open IE (Information Extraction), выполняющие такие функции как преобразование фраз в логические выражения, обеспечивающие ускоренный и более точный, за счет интеллектуальной обработки, поиск информации в веб-службе.

Вследствие этого возможно модернизировать инфокоммуникационную систему, введя в нее блок интеллектуальной обработки, взаимодействующий с веб-службой. В функции, выполняемые этим блоком, входит анализ поступающей на веб-службу информации, её классификация и размещение в соответствующих разделах базы данных веб-службы с наиболее точным соответствием тематике раздела; при извлечении информации из базы данных веб-службы блок интеллектуальной обработки позволяет наиболее чётко определить тематику и суть запроса, классифицировать результаты поиска по степени соответствия запросу, чтобы передать абоненту именно только нужную информацию.

При этом повышается скорость поиска и размещения информации. Особенно это важно, например, в экстремальной обстановке, когда у пользователя нет возможности долго осуществлять поиск нужной информации или соответствующего раздела для размещения информации в базе данных веб-службы. Такие ситуации характерны, например, для военнослужащих, пилотов самолётов, сотрудников МЧС и т.п., т.е. там, где

важна высокая скорость реакции пользователя инфокоммуникационной системы. Таким образом, повышается оперативность информационного обмена в спутниковой сингулярной конвергентной информационной системе за счёт интеллектуальной обработки информации.

Техническое решение

Техническое решение, реализующее данный подход разработано на базе известной спутниковой информационной системы [8].

Данная спутниковая информационная система состоит из N наземных приёмопередающих центров, включающих в себя последовательно соединенные локальные центры формирования информации, содержащие демодулятор, декодер, терминал с дисплеем, подвижные абонентские терминалы, передатчик, приёмник, приёмопередающую антенну, искусственный спутник Земли, включающий в себя приёмопередающую антенну, соединённую с ретранслятором, содержащим разделительный фильтр, первый преобразователь частоты, усилитель промежуточной частоты, второй преобразователь частоты, усилитель мощности, демодулятор, декодер, веб-службу с чат-роботом, кодер, модулятор, преобразователь частоты обратного тракта, усилитель промежуточной частоты обратного тракта, демодулятор обратного тракта, декодер обратного тракта, кодер обратного тракта, модулятор обратного тракта, второй преобразователь частоты обратного тракта и усилитель мощности обратного тракта, диспетчерский пункт, наземный приёмный пункт, включающий в себя последовательно соединённые приёмную антенну, приёмник, локальную цепь кабельного телевидения, телевизионный приёмник (рис. 1).

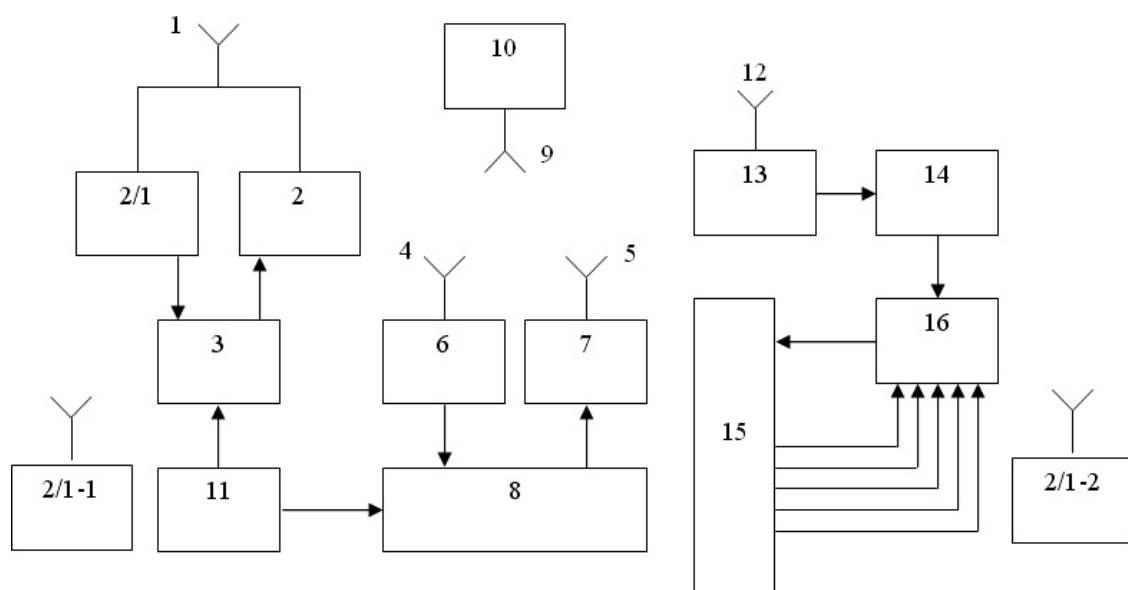


Рис. 1. Спутниковая информационная система: 1 – приёмопередающая антенна; 2 – передатчик; 2/1 – приемник, 2/1-1...2/1-2 – подвижные абонентские терминалы; 3 – локальный центр формирования информации; 4 – приемная антенна; 5 – передающая антенна; 6 – приемник; 7 – передатчик; 8 – центр обработки данных; 9 – приёмопередающая антенна; 10 – ретранслятор; 11 – система единого времени; 12 – приемная антенна; 13 – приемник; 14 – локальная сеть кабельного телевидения; 15 – блок выделения информации сигнала; 16 – телевизионный приёмник

Для решения задачи повышения оперативности информационного обмена в этой системе введем в ретранслятор блок интеллектуальной обработки информации 29/7 (рис. 2), двусторонне взаимодействующий с веб-службой и состоящий из

последовательно соединённых оперативного запоминающего устройства 54, блока интеллектуального анализа информации 55 и блока формирования запроса 56 (рис. 3).

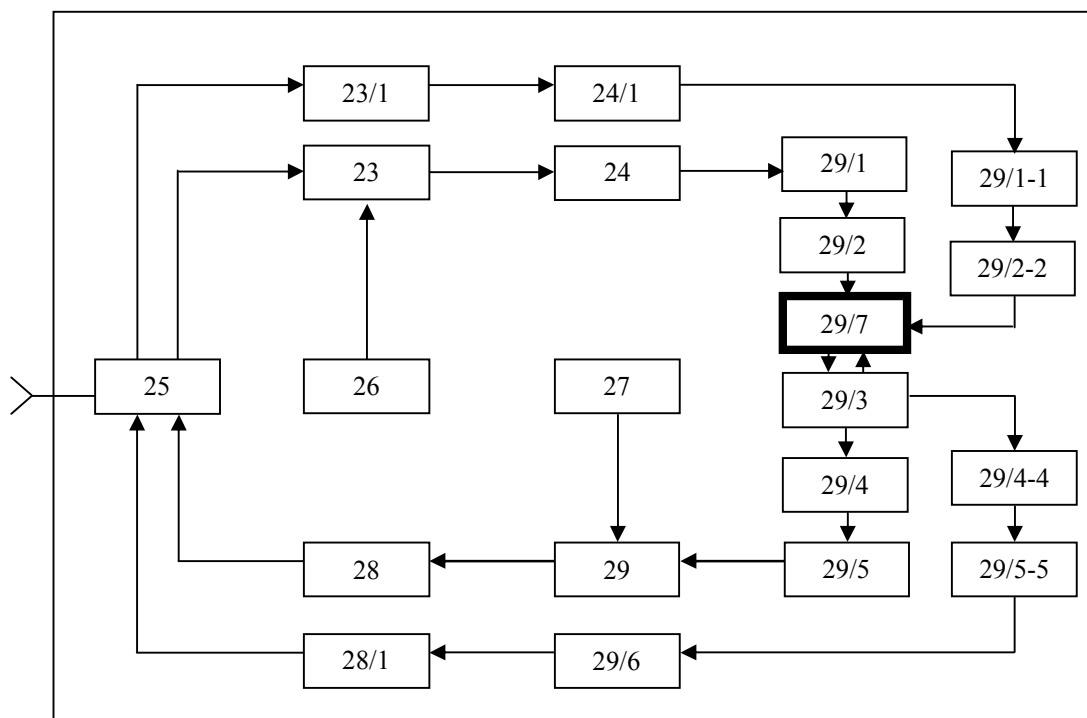


Рис. 2. Ретранслятор: 23 – преобразователь частоты; 26, 27 – гетеродин; 24 УПЧ, 25 – разделительный фильтр; 29 – преобразователь частоты; 28 – усилитель мощности; 29/1 – демодулятор; 29/2 – декодер; 29/3 – веб-служба; 29/4 – кодер; 29/5 – модулятор; 23/1 – преобразователь частоты обратного тракта; 24/1 - усилитель промежуточной частоты обратного тракт; 29/1-1 – демодулятор обратного тракта; 29/2-2 – декодер обратного тракта; 29/4-4 – кодер обратного тракта; 29/5-5 – модулятор обратного тракта; 29/6 – преобразователь частоты; 28/1 – усилитель мощности; 29/7 – блок интеллектуальной обработки информации

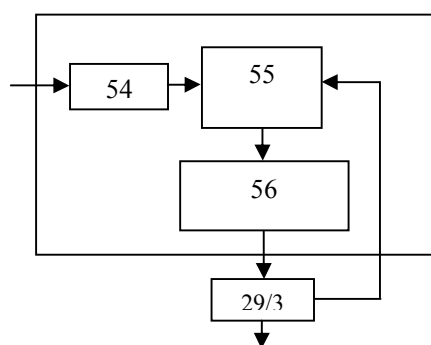


Рис. 3. Блок интеллектуальной обработки информации: 54 – оперативное запоминающее устройство (ОЗУ); 55 – блок интеллектуального анализа информации; 56 – блок формирования запроса; 29/3 – веб-служба

Работает данный блок следующим образом: информация поступает на оперативное запоминающее устройство 54, откуда передаётся на блок интеллектуального анализа информации 55, где происходит классификация информации, её анализ и уточнение. Далее в блоке формирования запроса 56 генерируется соответствующий запрос на по-

лучение или добавление информации, который поступает на веб-службу 29/3. В свою очередь, полученный ответ веб-службы поступает на блок интеллектуального анализа информации 55, где производится проверка соответствия результата запроса начальным требованиям и выделение из него наиболее значащей информации. При несоответствии результата запроса начальным требованиям блок интеллектуального анализа информации 55 через блок формирования запроса 56 передаёт на веб-службу с чат-роботом 29/3 повторный запрос для уточнения результата первого запроса.

Рассматриваемая система работает следующим образом. Текущая информация в различных регионах собирается в локальных центрах формирования информации. Сигнал поступает в устройства 17 преобразования информации в код и ЗУ 18, где из множества информационных сигналов, каждый из которых имеет метку, формируется блок информации для передачи. Далее информация сжимается по объему и в виде дискретных сигналов поступает на модулятор 21 канала передачи. Радиосигнал через передающее устройство, антенну поступает в разделительный фильтр 25, далее через первый преобразователь частоты 23 и усилитель промежуточной частоты 24 радиосигнал поступает на демодулятор 29/1, декодер 29/2, информация с декодера 29/2 поступает в блок интеллектуальной обработки информации 29/7 (рис. 3) в оперативное запоминающее устройство 54, передаётся на блок интеллектуального анализа информации 55, блок формирования запроса 56, поступает на веб-службу 29/3. Кодер 29/4 преобразует информацию из веб-службы 29/3 в радиосигнал, далее через модулятор 29/5, преобразователь частоты 29, усилитель мощности 28 и аппаратуру 10 искусственного спутника земли сигнал ретранслируется на диспетчерский пункт центра обработки информации.

В результате абонент этой системы для размещения информации в базе данных веб-службы может не затрачивать времени (особенно, это важно в экстремальных условиях) на подготовку к размещению информации, связанную с определением темы размещаемой информации и поиском соответствующего раздела базы данных веб-службы. Так же при получении важной оперативной информации, у абонента отсутствует необходимость просматривать ненужную информацию. В свою очередь блок интеллектуальной обработки, размещённый на ретрансляторе и взаимодействующий с веб-службой, осуществляет необходимые для размещения и поиска информации действия, описанные выше.

Заключение

Таким образом, за счет включения блока интеллектуальной обработки информации в ретранслятор, удалось качественно изменить спутниковую информационную систему так, что возросла оперативность передачи и приема информации за счет того, что информация из локальных центров формирования информации и из центра обработки данных может быстро без дополнительных затрат времени и излишней концентрации внимания абонента размещаться на веб-службе. В дальнейшем, на основе современных результатов теории массового обслуживания, будет проведен количественный анализ повышения оперативности информационного обмена между абонентами спутниковой информационной системы и, как следствие, повышения производительности спутниковой информационной системы в целом.

Литература

1. Слюсарев Г.В., Анашкин Р.В. Перспективные сингулярные конвергентные инфокоммуникационные системы. // XIII Всероссийская научно-методическая конференция «Телематика '2006» // http://tm.ifmo.ru/tm2006/db/doc/get_thes.php?id=1

2. Сток Г. Метачеловек: слияние людей и машин в глобальный сверхорганизм. – Simon & Schuster. – 1993.
3. Вернор Виндж Технологическая сингулярность // <http://www.computerra.ru/think/35636/>
4. Майкл Диригг Рассвет сингулярности (Пер. с английского Павел Васильев) // http://transhumanism.org/languages/russian/dawnofsingularity/Deering.htm#_Точ83817277
5. Пол Крилл Искусственный интеллект улучшит качество поиска в Интернете // <http://www.jobmarket.com.ua/news/internet/377129.html>
6. Turing Center Open Information Extraction // <http://www.kreate.ru/newsit/2/view/149>
7. Oren Etzioni A Softbot-based Interface to the Internet // <http://www.cs.washington.edu/homes/etzioni/papers/cacm.pdf>
8. Слюсарев Г.В., Анашкин Р.В., Романько Д.В. Патент №2332794 RU, МПК 7 H04B7/185. Спутниковая информационная система – 2008.03.20.

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ СЧИТЫВАНИЯ ПОКАЗАНИЙ АКТИВНОСТИ ГОЛОВНОГО МОЗГА ДЛЯ ПОВЫШЕНИЯ ОПЕРАТИВНОСТИ ИНФОРМАЦИОННОГО ОБМЕНА В СПУТНИКОВОЙ СИНГУЛЯРНОЙ КОНВЕРГЕНТНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЕ

Е.И. Ледовской, М.Ю. Татарков

(Северо-Кавказский государственный технический университет)

Научный руководитель – д.т.н., профессор Г.В. Слюсарев

(Северо-Кавказский государственный технический университет)

Представлено техническое решение, за счет которого повышается оперативность информационного обмена в спутниковой сингулярной конвергентной информационной системе за счет считывания показаний активности головного мозга.

Ключевые слова: сингулярность, конвергенция, оперативность, система, информация, активность, мозг

Введение

В настоящее время в мире проводится значительное количество исследований в области считывания показаний активности головного мозга. Подобные исследования ведутся учеными таких институтов как Advanced Telecommunications Research Institute International [1, 2], Маахтинский университет [3, 4] и др. В рамках этих исследований найдены способы преобразования мысли человека в визуальную информацию, для чего разрабатываются алгоритмы перевода показаний активности головного мозга в особые шаблоны, определяющие конкретный звук или голос, а также созданы системы управления роботами при помощи мысли человека [5]. Подобные технологии смогут найти применение в компьютерных системах для ускоренного ввода информации в базы данных веб-служб.

На этой основе возможно модернизировать инфокоммуникационную систему, вводя в нее блок считывания показаний активности головного мозга. В функции, выполняемые этим блоком, входит считывание показаний мозговой активности человека, их анализ и приведение информации к текстовому или графическому виду.

При этом повышается скорость ввода и размещения информации. Особенно это важно, например, в экстремальной обстановке когда у пользователя нет возможности долго осуществлять ввод нужной информации для размещения информации в базе данных веб-службы. Такие ситуации характерны, например, для военнослужащих, пилотов самолётов, сотрудников МЧС и т.п., т.е. там, где важна высокая скорость реакции пользователя инфокоммуникационной системы. Таким образом, повышается оперативность информационного обмена в спутниковой сингулярной конвергентной информационной системе [6–8] за счёт считывания показаний активности головного мозга.

Техническое решение

Техническое решение, реализующее данный подход разработано на базе известной спутниковой информационной системы [9]. Данная спутниковая информационная система состоит из N наземных приёмопередающих центров, включающих в себя последовательно соединенные локальные центры формирования информации, содержащие демодулятор, декодер, терминал с дисплеем, подвижные абонентские терминалы, передатчик, приёмник, приёмопередающую антенну, искусственный спутник Земли, включающий в себя приемопередающую антенну, соединенную с ретранслятором, со-

державшим разделительный фильтр, первый преобразователь частоты, усилитель промежуточной частоты, второй преобразователь частоты, усилитель мощности, демодулятор, декодер, веб-службу с чат-роботом, кодер, модулятор, преобразователь частоты обратного тракта, усилитель промежуточной частоты обратного тракта, демодулятор обратного тракта, декодер обратного тракта, кодер обратного тракта, модулятор обратного тракта, второй преобразователь частоты обратного тракта и усилитель мощности обратного тракта, диспетчерский пункт, наземный приемный пункт, включающий в себя последовательно соединенные приемную антенну, приемник, локальную цепь кабельного телевидения, телевизионный приемник (рис. 1, 2).

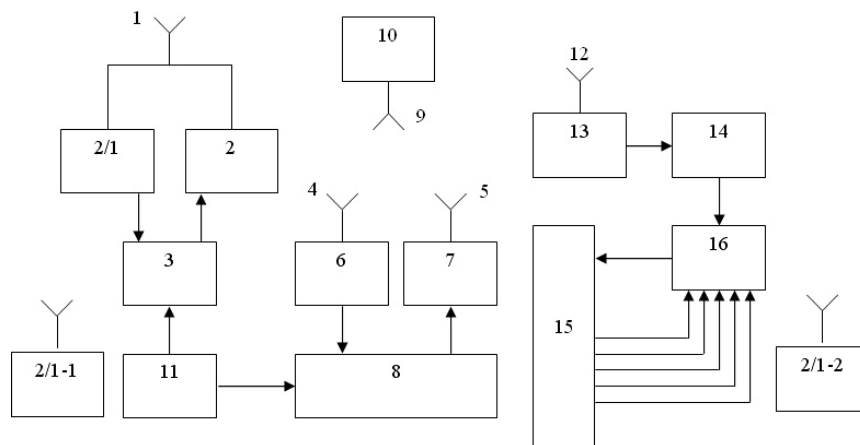


Рис. 1. Спутниковая информационная система: 1 – приемопередающая антенна; 2 – передатчик; 2/1 – приемник; 2/1-1...2/1-2 – подвижные абонентские терминалы; 3 – локальный центр формирования информации; 4 – приемная антенна; 5 – передающая антенна; 6 – приемник; 7 – передатчик; 8 – центр обработки данных; 9 – приемопередающая антенна; 10 – ретранслятор; 11 – система единого времени; 12 – приемная антенна; 13 – приемник; 14 – локальная сеть кабельного телевидения; 15 – блок выделения информации сигнала; 16 – телевизионный приёмник

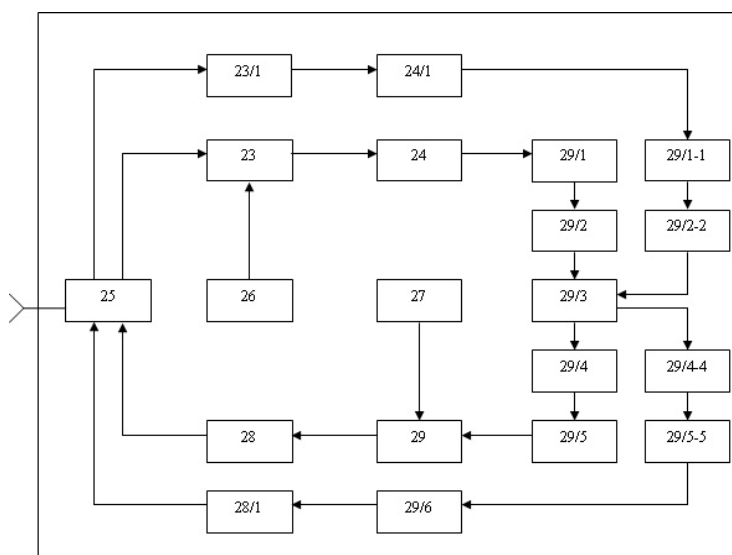


Рис. 2. Ретранслятор: 23 – преобразователь частоты; 26, 27 – гетеродин; 24 УПЧ; 25 – разделительный фильтр; 29 – преобразователь частоты; 28 – усилитель мощности; 29/1 – демодулятор; 29/2 – декодер; 29/3 – веб-служба; 29/4 – кодер; 29/5 – модулятор; 23/1 – преобразователь частоты обратного тракта; 24/1 – усилитель промежуточной частоты обратного тракта; 29/1-1 – демодулятор обратного тракта; 29/2-2 – декодер обратного тракта; 29/4-4 – кодер обратного тракта; 29/5-5 – модулятор обратного тракта; 29/6 – преобразователь частоты; 28/1 – усилитель мощности

Для решения задачи повышения оперативности информационного обмена в этой системе введем в локальные центры формирования информации блоки считывания показаний активности головного мозга 60/1-60/N (рис. 3), состоящие из последовательно соединённых N датчиков регистрации показаний активности головного мозга 61, блока усиления сигнала 62, блока преобразования сигнала 63, оперативного запоминающего устройства 64 и устройства управления 65 (рис. 4).

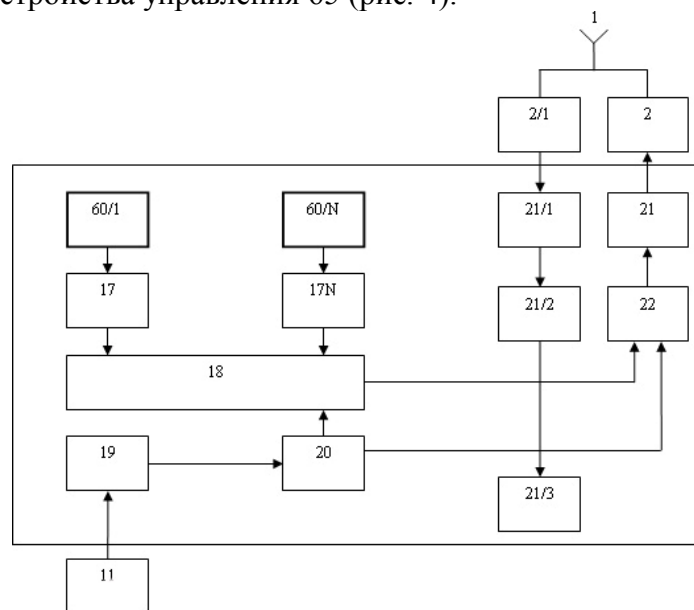


Рис. 3. Локальный центр формирования информации: 17/1-17/N – устройства преобразования информации в код; 18 – запоминающее устройство; 19 – таймер; 20 – блок управления; 21 – модулятор канала передачи; 22 – блок сжатия информации; 21/1 – демодулятор; 21/2 – декодер; 21/3 – терминал (дисплей); 60/1-61/N – блоки считывания показаний активности головного мозга; 11 – система единого времени

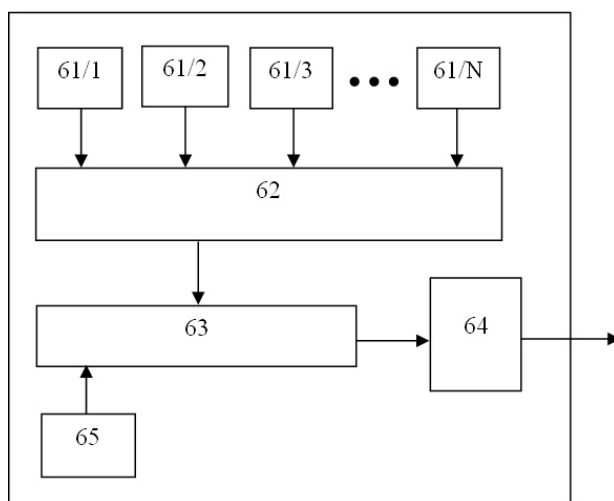


Рис. 4. Блок считывания показаний активности головного мозга: 61/1-61/N – датчики регистрации показаний активности головного мозга; 62 – блок усиления сигнала; 63 – блок преобразования сигнала; 64 – оперативное запоминающее устройство; 65 – устройство управления

Работает данный блок следующим образом: сигналы об активности головного мозга считываются с N датчиков регистрации электрических импульсов головного мозга 61, далее через блок усиления сигнала 62 поступают на блок преобразования сигнала

63, где происходит преобразование сигнала в текст, либо в графическое изображение. Информация с блока преобразования сигнала сохраняется в оперативном запоминающем устройстве 64, откуда передаётся на устройства 17 преобразования информации в код. Устройство управления 65 осуществляет управление преобразованием сигнала об активности головного мозга в текстовую или графическую информацию.

Рассматриваемая система работает следующим образом. Текущая информация в различных регионах собирается в локальных центрах формирования информации. Сигналы об активности головного мозга считываются с N датчиков регистрации электрических импульсов головного мозга 61, поступают на блок усиления сигнала 62, через блок преобразования сигнала в символы 63 поступают в запоминающее устройство 64. При этом устройство управления 65 осуществляет управление преобразованием сигналов об активности головного мозга в текстовую или графическую информацию. Далее сигнал поступает в устройства 17 преобразования информации в код и ЗУ 18, где из множества информационных сигналов, каждый из которых имеет метку, формируется блок информации для передачи. Далее информация сжимается по объему и в виде дискретных сигналов поступает на модулятор 21 канала передачи. Радиосигнал через передающее устройство, антенну поступает в разделительный фильтр 25, далее через первый преобразователь частоты 23 и усилитель промежуточной частоты 24 радиосигнал поступает на демодулятор 29/1, декодер 29/2, информация с декодера 29/2 поступает на веб-службу 29/3. Кодер 29/4 преобразует информацию из веб-службы 29/3 в радиосигнал, далее через модулятор 29/5, преобразователь частоты 29, усилитель мощности 28 и аппаратуру 10 искусственного спутника земли сигнал ретранслируется на диспетчерский пункт центра обработки информации.

В результате абонент этой системы для размещения информации в базе данных веб-службы может не затрачивать времени (особенно, это важно в экстремальных условиях) на подготовку к вводу и ввод информации, связанную с использованием таких устройств ввода, как клавиатура, графический сканер и т.п. В свою очередь блок считывания показаний активности головного мозга, размещённый в локальных центрах формирования информации осуществляет необходимые для быстрого ввода информации действия, описанные выше.

Заключение

Таким образом за счет включения блока считывания показаний активности головного мозга в локальные центры формирования информации удалось качественно изменить спутниковую информационную систему так, что возросла оперативность передачи и приема информации за счет того, что информация в локальных центрах формирования информации может быстро без дополнительных затрат времени введена для передачи и размещения в базе данных на веб-службы. В дальнейшем будет проведен количественный анализ повышения оперативности информационного обмена между абонентами спутниковой информационной системы и, как следствие, повышения производительности спутниковой информационной системы в целом.

Литература

1. Первый в мире удачный опыт по визуализации мыслей человека // <http://science.compulenta.ru/388495/>
2. Advanced Telecommunications Research Institute International // http://www.atr.co.jp/index_e.html
3. Активность мозга позволяет подслушать разговор // <http://science.compulenta.ru/380559/>

4. Universiteit Maastricht // <http://www.unimaas.nl/default.asp?taal=EN>
5. Мысленный интерфейс между мозгом человека и роботом // <http://www.membrana.ru/lenta/?5930>
6. Слюсарев Г.В., Анашкин Р.В. Перспективные сингулярные конвергентные инфокоммуникационные системы. // XIII Всероссийская научно-методическая конференция «Телематика '2006» // http://tm.ifmo.ru/tm2006/db/doc/get_thes.php?id=1
7. Вернор Виндж Технологическая сингулярность // <http://www.computerra.ru/think/35636/>
8. Майкл Диригг Рассвет сингулярности (Пер. с английского Павел Васильев) // http://transhumanism.org/languages/russian/dawnofsingularity/Deering.htm#_Toc83817277
9. Слюсарев Г.В., Анашкин Р.В., Романько Д.В. Патент №2332794 RU, МПК 7 H04B7/185. Спутниковая информационная система – 2008.03.20.

СЧИТЫВАНИЕ ДВИЖЕНИЙ КИСТИ РУКИ ДЛЯ УСКОРЕНИЯ ИНФОРМАЦИОННОГО ОБМЕНА В СПУТНИКОВОЙ СИНГУЛЯРНОЙ КОНВЕРГЕНТНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЕ

Е.И. Ледовской, М.Ю. Татарков

(Северо-Кавказский государственный технический университет)

Научный руководитель – д.т.н., профессор Г.В. Слюсарев

(Северо-Кавказский государственный технический университет)

Разработано техническое решение, за счет которого повышается оперативность информационного обмена в спутниковой сингулярной конвергентной информационной системе путем считывания движений кисти руки.

Ключевые слова: сингулярность, конвергенция, оперативность, система, информация, движение, кисть, рука

Введение

Среди разрабатываемых в настоящее время в мире инфокоммуникационных устройств перспективными являются блоки считывания движений кисти руки, сконструированные в виде bluetooth-перчаток. Так, например, компания Semetech представила прототип bluetooth-перчатки «Clove2» [1], способной выполнять функции обычной клавиатуры. Также компания Fibre Structures LTD разработала bluetooth-перчатку «Control Glove» [2], предназначенную для управления различными инфокоммуникационными устройствами, промышленными, военными приложениями и т.п. Наряду с этими разработками bluetooth-перчатки находят широкое применение в повседневной жизни человека. Подтверждением этому является разработка компанией Swanny bluetooth-перчаток для горнолыжников и сноубордистов [3] и т.п. Их использование может обеспечить ускоренный ввод информации в частности и в базы данных веб-служб.

Это позволяет модернизировать инфокоммуникационную систему, введя в нее блок считывания движений кисти руки. В функции, выполняемые этим блоком, входит считывание движений кисти руки, преобразование полученных сигналов в символы и передача информации для размещения в базе данных веб-службы.

Это ускорит ввод и размещение информации. Например, находясь в экстремальной обстановке, пользователь как правило не имеет возможности осуществлять ввод нужной информации, используя клавиатуру, для размещения информации в базе данных веб-службы, т.к. для ввода текста с клавиатуры ему необходимо тратить время на то, чтобы приготовить клавиатуру, снять перчатки экипировки и т.п. Такие ситуации характерны, например, для военнослужащих, пилотов самолётов, сотрудников МЧС и т.п., т.е. там, где важна высокая скорость реакции пользователя инфокоммуникационной системы. Таким образом, повышается оперативность информационного обмена в спутниковой сингулярной конвергентной [4–6] информационной системе за счёт считывания движений кисти руки.

Техническое решение

Техническое решение, реализующее данный подход разработано на базе известной спутниковой информационной системы [7]. Данная спутниковая информационная система состоит из N наземных приёмопередающих центров, включающих в себя последовательно соединенные локальные центры формирования информации, содержащие демодулятор, декодер, терминал с дисплеем, подвижные абонентские терминалы,

передатчик, приёмник, приёмопередающую антенну, искусственный спутник Земли, включающий в себя приёмопередающую антенну, соединённую с ретранслятором, содержащим разделительный фильтр, первый преобразователь частоты, усилитель промежуточной частоты, второй преобразователь частоты, усилитель мощности, демодулятор, декодер, веб-службу с чат-роботом, кодер, модулятор, преобразователь частоты обратного тракта, усилитель промежуточной частоты обратного тракта, демодулятор обратного тракта, декодер обратного тракта, кодер обратного тракта, модулятор обратного тракта, второй преобразователь частоты обратного тракта и усилитель мощности обратного тракта, диспетчерский пункт, наземный приёмный пункт, включающий в себя последовательно соединённые приемную антенну, приемник, локальную цепь кабельного телевидения, телевизионный приёмник (рис. 1, 2).

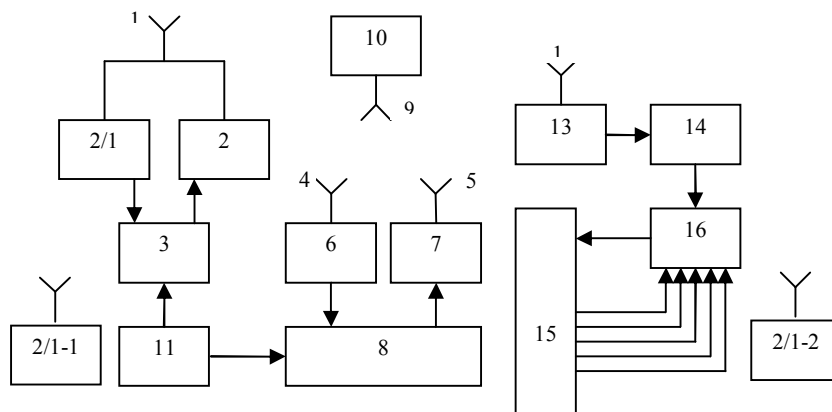


Рис. 1. Спутниковая информационная система: 1 – приёмопередающая антенна; 2 – передатчик; 2/1 – приёмник; 2/1-1...2/1-2 – подвижные абонентские терминалы; 3 – локальный центр формирования информации; 4 – приемная антенна; 5 – передающая антенна; 6 – приёмник; 7 – передатчик; 8 – центр обработки данных; 9 – приёмопередающая антенна; 10 – ретранслятор; 11 – система единого времени; 12 – приемная антенна; 13 – приёмник; 14 – локальная сеть кабельного телевидения; 15 – блок выделения информации сигнала; 16 – телевизионный приёмник

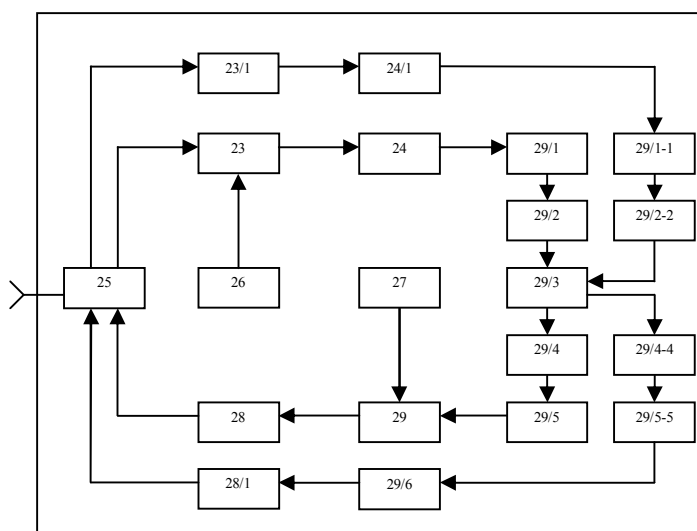


Рис. 2. Ретранслятор: 23 – преобразователь частоты; 26, 27 – гетеродин; 24 УПЧ; 25 – разделительный фильтр; 29 – преобразователь частоты; 28 – усилитель мощности; 29/1 – демодулятор; 29/2 – декодер; 29/3 – веб-служба; 29/4 – кодер; 29/5 – модулятор; 23/1 – преобразователь частоты обратного тракта; 24/1 – усилитель промежуточной частоты обратного тракта; 29/1-1 – демодулятор обратного тракта; 29/2-2 – декодер обратного тракта; 29/4-4 – кодер обратного тракта; 29/5-5 – модулятор обратного тракта; 29/6 – преобразователь частоты; 28/1 – усилитель мощности

Для решения задачи повышения оперативности информационного обмена в этой системе введем в локальные центры формирования информации блоки считывания движений кисти руки 54/1-54/N (рис. 3), состоящие из последовательно соединённых N контактных площадок 55, блока преобразования сигнала в символы 56, оперативного запоминающего устройства 57, bluetooth-передатчика 58 и bluetooth-приемника 59 (рис. 4). Работает данный блок следующим образом: движения кисти руки считываются с N контактных площадок 55. Полученные сигналы преобразуются в блоке преобразования сигнала в символы 56 в текстовую информацию и переходят в оперативное запоминающее устройство 57. Далее через bluetooth-передатчик 58 сигналы передаются на bluetooth-приёмник 59.

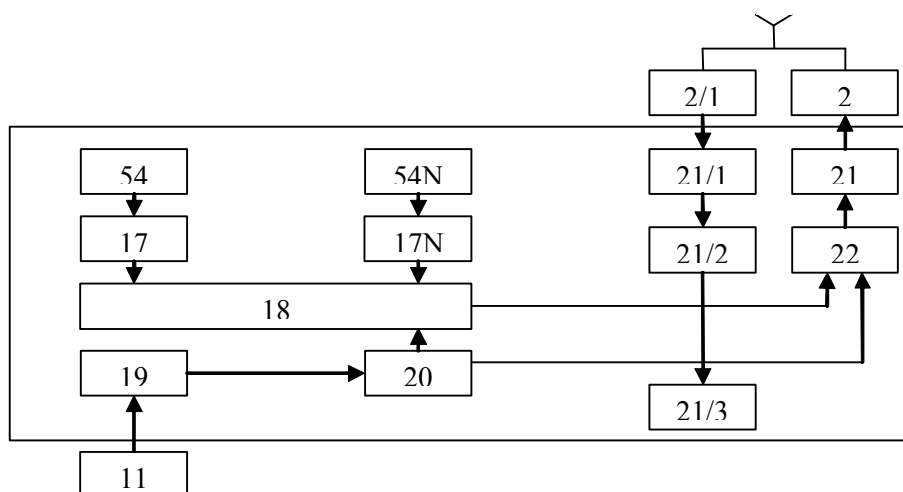


Рис. 3. Локальный центр формирования информации: 17/1-17/N – устройства преобразования информации в код; 18 – запоминающее устройство; 19 – таймер; 20 – блок управления; 21 – модулятор канала передачи; 22 – блок сжатия информации; 21/1 – демодулятор; 21/2 – декодер; 21/3 – терминал (дисплей); 54/1-54/N – блоки считывания движений кисти руки; 11 – система единого времени

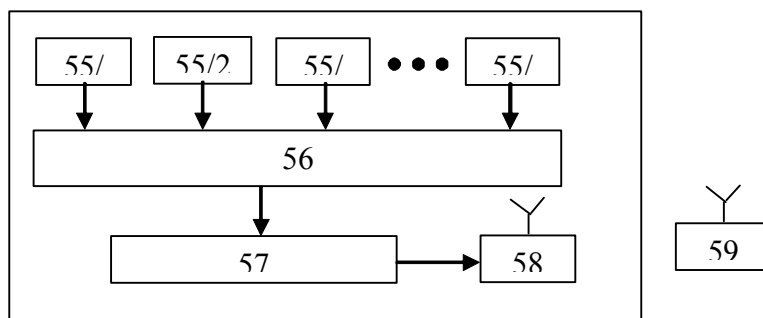


Рис. 4. Блок считывания движений кисти руки: 54/1-54/N – контактные площадки; 56 – блок преобразования сигнала в символы; 57 – оперативное запоминающее устройство; 58 – bluetooth-передатчик; 59 – bluetooth-приемник

Рассматриваемая система работает следующим образом. Текущая информация в различных регионах собирается в локальных центрах формирования информации. Сигнал считывается с N контактных площадок 55, поступает на блок преобразования сигнала в символы 56 и запоминающее устройство 57. Далее через bluetooth-передатчик 58 и bluetooth-приёмник 59 блока считывания движений кисти руки 54 сигнал поступает в устройства 17 преобразования информации в код и ЗУ 18, где из множества информационных сигналов, каждый из которых имеет метку, формируется блок информации

для передачи. Далее информация сжимается по объему и в виде дискретных сигналов поступает на модулятор 21 канала передачи. Радиосигнал через передающее устройство, антенну поступает в разделительный фильтр 25, далее через первый преобразователь частоты 23 и усилитель промежуточной частоты 24 радиосигнал поступает на демодулятор 29/1, декодер 29/2, информация с декодера 29/2 поступает на веб-службу 29/3. Кодер 29/4 преобразует информацию из веб-службы 29/3 в радиосигнал, далее через модулятор 29/5, преобразователь частоты 29, усилитель мощности 28 и аппаратуру 10 искусственного спутника земли сигнал ретранслируется на диспетчерский пункт центра обработки информации.

В результате абонент этой системы для размещения информации в базе данных веб-службы может не затрачивать дополнительное время на подготовку к вводу информации, связанную с необходимостью снимать перчатки экипировки, чтобы воспользоваться клавиатурой (особенно, это важно в экстремальных условиях), установку клавиатуры и подготовку места для нее. В свою очередь блок считывания движений кисти руки, размещенный в локальных центрах формирования информации, осуществляет необходимые для быстрого ввода информации действия, описанные выше.

Заключение

Таким образом, за счет включения блока считывания движений кисти руки в локальные центры формирования информации, удалось качественно изменить спутниковую информационную систему так, что возросла оперативность передачи и приема информации за счет того, что информация в локальных центрах формирования информации может быстро без дополнительных затрат времени введена для передачи и размещения в базе данных на веб-службы. В дальнейшем будет проведен количественный анализ повышения оперативности информационного обмена между абонентами спутниковой информационной системы и, как следствие, повышения производительности спутниковой информационной системы в целом.

Литература

1. Cemetech Bluetooth Dataglove Clove 2 // <http://www.cemetechnet/projects/item.php?id=16>
2. Fibre Structures LTD Control Glove // <http://www.fibrestructures.com/innovations.html>
3. Swanny Bluetooth-перчатки для горнолыжников и сноубордистов // http://www.ski.ru/static/357/4_27586.html
4. Слюсарев Г.В., Анашкин Р.В. Перспективные сингулярные конвергентные инфокоммуникационные системы. // XIII Всероссийская научно-методическая конференция «Телематика '2006» // http://tm.ifmo.ru/tm2006/db/doc/get_thes.php?id=1
5. Верноп Виндж Технологическая сингулярность // <http://www.computerra.ru/think/35636/>
6. Майкл Диринг Рассвет сингулярности (Пер. с английского Павел Васильев) // http://transhumanism.org/languages/russian/dawnofsingularity/Deering.htm#_Toc83817277
7. Слюсарев Г.В., Анашкин Р.В., Романько Д.В. Патент №2332794 RU, МПК 7 H04B7/185. Спутниковая информационная система – 2008.03.20.

ВСТРАИВАЕМОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫСОКОСКОРОСТНОЙ СЕТЕВОЙ КАМЕРЫ ДЛЯ ОПТИЧЕСКОЙ ДИАГНОСТИКИ БЫСТРОПРОТЕКАЮЩИХ ПРОЦЕССОВ

С.А. Федоров

(Санкт-Петербургский государственный политехнический университет)

Научный руководитель – к.т.н., доцент В.В. Амосов

(Санкт-Петербургский государственный политехнический университет)

Оптическая диагностика быстропротекающих процессов имеет широкое научное и практическое значение. В данной работе приводится реализованный подход к разработке сетевой высокоскоростной камеры, имеющей в своём составе встраиваемую систему, работающую под управлением операционной системы GNU/Linux. Подход основан на использовании технологии Система на кристалле, сетевых технологий и высокой интеграции необходимых элементов, позволяет обойтись без дорогостоящих плат захвата изображения и свободен от основных недостатков решений, основанных на специальном сетевом интерфейсе Channel Link. В работе также определены перспективы и ещё нерешённые задачи.

Ключевые слова: встраиваемое программное обеспечение, сетевая камера, обработка и передача данных, встраиваемая система на кристалле, Linux, скоростная видеосъёмка

Введение

Создание высокоскоростных устройств детектирования, хранения и передачи оптической информации, поступающей в результате различных физических явлений (взрывы, вспышки, разрушения, баллистика и т.п.), позволяет разрабатывать новые методики исследования и контрольно-измерительную аппаратуру этих процессов. Стоит отметить, что среди большого числа возможных применений высокоскоростных видеокамер наиболее сложные задачи возникают в исследованиях физики плазмы, что связано со следующими условиями и требованиями:

- короткое время существования плазменного разряда (≈ 100 мс);
- большие перепады яркости (до 10^8) во времени и в различных участках детектируемого изображения;
- обязательное интегрирование в существующие диагностические комплексы;
- жесткая синхронизация работы с моментом вспышки;
- работа в условиях сильных электромагнитных помех;
- необходимость передачи данных на удаленные центры мониторинга и обработки данных.

Анализ тенденций развития рынка высокоскоростных видеокамер показал, что последние достижения ведущих компаний-производителей в настоящее время связаны не столько с совершенствованием детекторов, сколько с разработкой принципиально новых подходов к их «электронному обрамлению» и создания, так называемых, «сетевых камер», в том числе, устройств с возможностью реконфигурирования через Internet непосредственно в системе.

Скоростные цифровые камеры ведущего производителя отечественного рынка - фирмы FastVideo обычно поставляются с платой видеозахвата на PCI-Express x1 с интерфейсом Camera Link для подключения камеры к настольному компьютеру или серверу, что накладывает ряд ограничений, связанных, в первую очередь, с использованием протокола Camera Link (см. рис. 1).

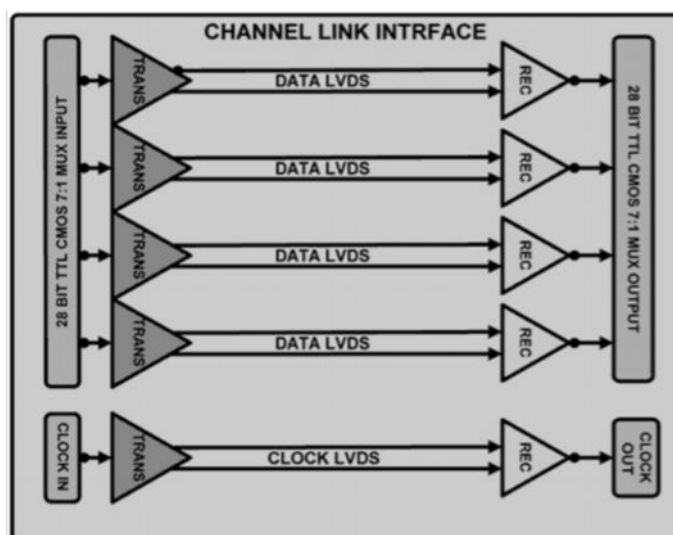


Рис. 1. Интерфейс Channel Link

- Малая длина кабеля. Длина кабеля Camera Link обычно 3 м, 5 м или 10 метров, но последнее справедливо только для конфигурации Base Camera Link. Существуют также повторители (repeaters), которые могут увеличивать максимальную длину кабеля. Однако для работы на максимально высоких частотах такие длины кабелей совершенно не подходят и повторители не используются. Для конфигурации Full CameraLink рекомендуется пара кабелей длиной от 1,5 до 3 метров, причём эти кабели обязательно должны быть одинаковыми.
- Низкая скорость передачи данных для управления камерой до 57600 бод.
- Стоимость: соединение камеры с компьютером требует дорогостоящих специализированных плат видеозахвата.
- Соединения всегда имеют вид точка-точка, что затрудняет работу при подключении нескольких камер одновременно.

Таким образом, тема, связанная с разработкой специального ПО для высокоскоростных оптических детекторов [1], несомненно, является актуальной и имеющей большое научное практическое значение, так как позволит разрабатывать на ее основе новую аппаратуру и методики исследований не только в области плазменного эксперимента, но и в таких важных направлениях как баллистика и крэш-контроль материалов и деталей, диагностика горения и взрывов.

Постановка задачи

Анализ рынка выпускаемых к настоящему времени высокоскоростных видеокамер показал, что, обеспечивая в принципе необходимую скорость регистрации данных, они не удовлетворяют совокупности перечисленных выше требований. В связи с этим в ФТИ им. А. Ф. Иоффе РАН была разработана высокоскоростная камера на базе высокоскоростного КМОП-сенсора фирмы DALSA, программируемых логических матриц и системы на кристалле (СнК, System On Chip, SOC), позволяющая реализовать все необходимые функции для обеспечения различных диагностических методов в научных экспериментах по исследованию физики плазмы.

Основной задачей является исследование подходов и разработка специального встраиваемого программного обеспечения для данной высокоскоростной камеры, в том числе для реализации скоростной передачи данных, получаемых в результате проведения оптической диагностики в экспериментах.

Подход к организации структуры высокоскоростной камеры

Альтернативным решением проблем протокола Camera Link, указанных выше, является встраивание платы видеозахвата, модуля с оперативной памятью и модуля для связи с компьютером в высокоскоростную видеокамеру.

Таким образом, предлагаемая концепция построения реконфигурируемой сетевой высокоскоростной видеокамеры, конструктивно выполненной на одной печатной плате на базе программируемой вентиляционной матрицы Xilinx FPGA и встроенной системы на кристалле ADM5120 фирмы Infineon, полностью отвечает современным тенденциям построения высокоскоростных контрольно-измерительных систем различного применения. Такой подход к аппаратной реализации видеокамеры позволяет реализовать в едином блоке функциональные модули управления сенсором (КМОП или ПЗС), аналого-цифрового преобразования, кадровой памяти, интерфейс локального пользователя (TFT LCD и клавиатура) и контроллер передачи данных в PC, функционирующие под управлением центрального процессора (CPU).

Разработанный загрузчик системы на кристалле

Для старта системы необходим загрузчик. С помощью разработанного программатора его можно запрограммировать во flash-память блока камеры.

При написании загрузчика очень важно было изучить систему на кристалле и процессор в ней для получения следующей платформи-зависимой информации, которая существенным образом будет влиять на исходный код:

- размер flash-памяти – 4 Мб;
- порядок байтов от младшего к старшему;
- flash-память архитектуры NOR;
- адрес загрузки 0x1FC00000.

Загрузчик написан на языке C с некоторым ассемблерным кодом для инициализации процессора. Он основан на стандартном загрузчике – имеет в своём составе:

- драйвер RS232 для интерфейса пользователя через консоль (COM-порт);
- программа обновления загрузчика через протокол X-Modem;

Загрузчик расширен и имеет ряд дополнительных функций (подпрограмм):

- драйвер IEEE 802.3 Ethernet;
- TFTP-клиент, обеспечивающий загрузку загрузчика и ядра Linux вместе с приложениями по сети;
- программа автоматического запуска ядра Linux, если к камере не обращаются через консоль или не проводят в ней (консоли) никаких действий
- программа обновления ядра Linux через протокол X-Modem;
- программа обновления ядра Linux по сети (Fast Ethernet).

Загрузчик предназначен для инициализации процессора, обновления встроенного ПО и запуска ядра операционной системы GNU/Linux.

Способы высокоскоростной передачи данных

Изучение архитектуры системы и её анализ дали три возможных способа реализации высокоскоростной передачи данных.

Программирование процессора. Есть возможность встроить все необходимые функции прямо в код загрузчика, расширив тем самым его функциональность. После инициализации он запускает циклическую программу опроса сетевого порта.

Преимущества подхода:

- сравнительная простота реализации;

- быстродействие, т.к. выполняется всего один процесс.

Но у этого подхода есть и серьезные недостатки:

- необходимость подключаться к консоли в случае задания параметров;
- отсутствие гибкости задания сетевых настроек и параметров скачивания;
- необходимость устанавливать на компьютере дополнительное ПО;
- плохая переносимость.

Встраивание Linux и использование интерфейса sockets. Можно встроить Linux в СнК, а загрузчику поручить её запуск. Тогда все необходимые сетевые средства будут в Linux. Передачу же файлов реализовать с помощью интерфейса sockets.

Преимущества подхода:

- подключение только по сети (без консоли);
- гибкость задания сетевых настроек.

Недостатки:

- сложность реализации;
- быстродействие, т.к. выполняется несколько процессов;
- отсутствие гибкости задания параметров скачивания;
- необходимость устанавливать на компьютере дополнительное ПО;
- плохая переносимость из-за использования интерфейса sockets.

Встраивание Linux и использование веб-интерфейса. Этот подход похож на предыдущий, но не содержит его основных недостатков. Передачу файлов в данном случае предлагается реализовать через веб-интерфейс.

Преимущества подхода:

- подключение только по сети (без консоли);
- гибкость задания сетевых настроек;
- гибкость задания параметров скачивания;
- не нужно устанавливать на компьютере дополнительное ПО - необходимо наличие любого веб-браузера;
- переносимость из-за использования протокола http.

Недостатки:

- сложность реализации;
- быстродействие, т.к. выполняется несколько процессов.

Реализация сетевых функций камеры

Автором был использован третий подход, обладающий неоспоримыми преимуществами. Чтобы его реализовать, необходимо поднять всю сетевую инфраструктуру Linux и написать веб-страницы, реализовав тем самым сервер на камере.

Для конфигурирования Linux была выбрана серия ядер Linux 2.6 [2, 3] и дистрибутив OpenWrt Kakimaze 8.09. В конфигурационном файле были выставлены все необходимые сетевые сервисы, указана архитектура процессора MIPS32 [4, 5], СнК ADM5120, объём установленной оперативной памяти – 16 Мб. Для сборки Linux использовались инструменты кросс-платформенной компиляции (toolchain 0423-1) и библиотека uclibc.

Архитектура встроенного ПО

Архитектура встроенного ПО основана на самом ядре Linux и нескольких процессах (приложениях), выполняющих различные задачи. Каждый процесс запрашивает для своей работы различные ресурсы – кванты процессорного времени, память, сетевое подключение. Ядро Linux обрабатывает их запросы. Таким образом, компоненты

встроенного ПО можно поделить на пространство пользователя и пространство ядра в соответствии с блок-схемой на рис. 2.

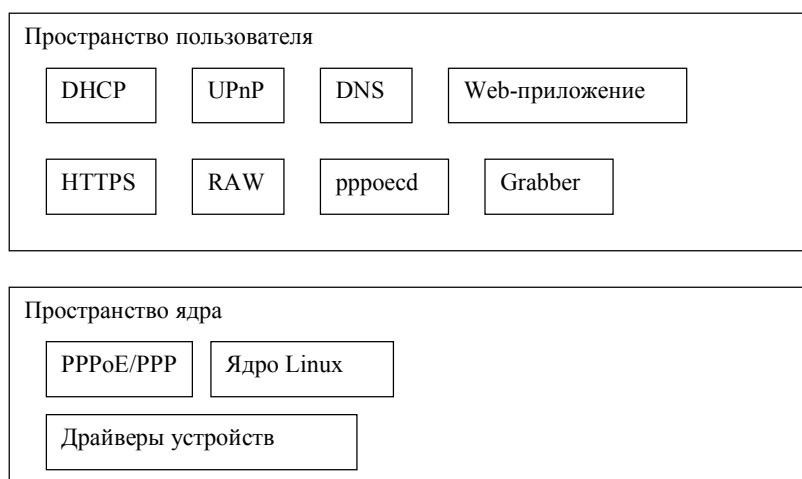


Рис. 2. Структура встроенного ПО Linux

Основные результаты

Были изучены элементы и устройства скоростной камеры, куда встраивалась система. Изучены устройства и характеристики встраиваемой системы на кристалле (СнК) ADM5120 [6], а также архитектура её процессора MIPS32 4Кс. Реализован программатор на основе интерфейса JTAG, позволяющий связывать СнК с компьютером и тем самым программировать её. Разработан загрузчик для встраиваемой СнК. Проведено тестирование СнК-загрузчика. Сняты метрики загрузчика.

Исследованы и проанализированы возможные способы реализации высокоскоростной передачи данных для встраиваемой системы на кристалле. Создано встраиваемое программное обеспечение, которое позволяет запускать и инициализировать встраиваемую систему, работать камере как сетевому устройству. Встраиваемое ПО обеспечивает автоматическую настройку ее сетевых параметров и подключенного к ней компьютера. Камера, как сетевое устройство, работает под управлением встроенной операционной системы Linux, которая была сконфигурирована для её архитектуры.

Заключение

Подход к разработке сетевой камеры, изученный и реализованный автором, убедительно подтвердил преимущества по сравнению с другими решениями, в частности, использующими протокол Camera Link. Нет необходимости устанавливать в компьютер, в котором собирается и изучается видеoinформация, дорогостоящее оборудование, вроде плат захвата изображения. Сетевая камера использует технологию Fast Ethernet, которая широко распространена и реализована почти во всех южных мостах у современных наборов микросхем. Не требуется также устанавливать дополнительное ПО – для работы с камерой достаточно наличие веб-браузера. Это обеспечивает широкую переносимость, как среди аппаратных платформ, так и среди операционных систем, ведь при таких требованиях камера подключается без предварительных действий ко всем компьютерам с сетевым интерфейсом.

Встроенный сервер DHCP обеспечивает простоту подключения к камере компьютера и их автоматическую сетевую настройку. Сервер DNS даёт удобное сетевое имя камере, что позволяет обращаться к ней быстрее и универсальной.

Реализованный подход полностью оправдывает себя в областях научных исследований. Благодаря использованию технологии Fast Ethernet компьютер может находиться удалённо в другом безопасном и защищённом от действия изучаемого процесса помещении. И по той же причине к нему может подключаться несколько сетевых камер.

Благодаря тому, что встраиваемое ПО ориентировано на использование веб-технологий, его основной исходный код, отвечающий именно за функциональность, является архитектурно-независимым. Это позволит не только в дальнейшем быстро разрабатывать новые сетевые камеры на других кристаллах, но и с лёгкостью перейти на ещё более скоростные технологии, например IEEE 802.3z Gigabit Ethernet со скоростью 1000 Мбит/с.

Возможности сетевой камеры и реализованные в ней средства передачи информации по сети полностью отвечают жёстким требованиям, предъявляемым к ней как устройству со встроенной автоматизированной системой научных исследований быстро протекающих или опасных процессов.

В данной работе реализован не имеющий аналогов на отечественном рынке подход сетевой камеры, когда для связи с компьютером не используются интерфейс Channel Link и дорогостоящие платы захвата изображения. В ФТИ им А.Ф. Иоффе РАН планируется расширить функциональность встроенного ПО. Важно выделить несколько важных и интересных решений и путей развития, имеющих несомненные перспективы:

- разработка web-приложения, позволяющего скачивать файл данных с необходимой информацией по сети, и специального модуля ядра Linux [7, 8], обеспечивающего чтение файлов с видеoinформацией без использования дорогостоящих плат видеозахвата;
- возможность копирования отснятого изображения на flash-носитель;
- просмотр отснятого материала через веб-интерфейс в замедленном с нужной частотой кадров скоростью, а также в режиме реального времени;
- встраивание сенсорной панели;
- использование для настроек и управления собственного LCD-монитора камеры;
- переход на скорости Gigabyte Ethernet;
- настройка фокуса и экспозиции камеры ввиду особенностей наблюдаемых процессов при научных исследованиях;
- интеллектуальный анализ изображений, автоматический выбор параметров и продолжительности съёмки, накопление информации, обучение и адаптация к особенностям типа изучаемого эксперимента.

Литература

1. Федоров С.А. Диссертация на соискание учёной степени магистра: Высокоскоростная передача данных для встраиваемой АСНИ на кристалле. – СПбГПУ. – 2008. – 101 с.
2. Лав Р. Разработка ядра Linux. СПб.: Вильямс. – 2006. – 448 с.
3. Чезати М. Ядро Linux. СПб: БХВ-Петербург. – 2007. – 1104 с.
4. MIPS32 4Kc Processor Core Data Sheet. MD00039-2B-4Kc-DTS-01.07. 28 с.
5. MIPS32 4K Processor Core Family Integrator's Guide. MD00036-2B-4K-ING-01.10. – 56 с.
6. ADM5120 Network Processor V1.16 Data Sheet. ADM5120_V1.16_DS_Rev1.1. – 196 с.
7. Salzman P.J., Burian M., Pomerantz O. The Linux Kernel Module Programming Guide. 2007-05-18 ver 2.6.4. – 82 с.
8. Corbet J., Rubini A., Kroah-Hartman G. Linux Device Drivers. O'Reilly. – 2005. – 636 с.

ИСТОРИЯ И ТЕНДАНЦИИ РАЗВИТИЯ МОБИЛЬНЫХ СРЕДСТВ ПЕРЕДАЧИ ИНФОРМАЦИИ

Н.С. Токалов, С.Е. Антонов

Научный руководитель – А.П. Ищенко

В статье рассмотрены основные вехи и некоторые тенденции развития мобильных средств передачи информации, включая карманные переносные компьютеры, органайзеры и смартфоны, в сравнительном аспекте рассмотрены их достоинства и недостатки, а также эволюционные пути интеграции различных портативных устройств.

Ключевые слова: мобильные средства, карманные переносные компьютеры, органайзеры, смартфоны

Введение

Современный период развития цивилизации характеризуется тем, что основным товаром является информация в различных ее проявлениях. К настоящему времени в мире накоплено столько информации, что люди уже не в состоянии сами воспользоваться этим благом в полном объеме. На помощь приходят компьютеры и глобальные сети, объединяющие множество компьютеров. Поэтому, появление и развитие компьютеров – это необходимая составляющая процесса информатизации общества. Одной из основных тенденций с момента появления компьютеров было стремление к уменьшению их размеров. Целый этаж здания, комната, шкаф, настольный компьютеры, ноутбуки, карманные переносные компьютеры (КПК) – вот примерный путь развития средств передачи информации. В результате стремительного развития в России рынка мобильных устройств их продажи за последние десять лет возросли более чем сто раз (рис. 1). В настоящей работе будут рассмотрены отдельные вехи истории и некоторые тенденции развития мобильных средств передачи информации [10].

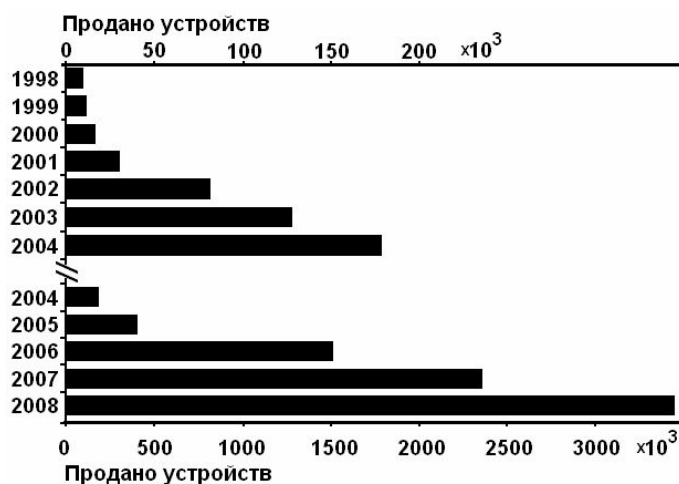


Рис. 1. Объемы продаж мобильных устройств [1–9]

КПК

Промышленный выпуск КПК начался в 1984 г. с выходом Psion Organizer. Позже (1989) на рынке появились КПК от компании Palm. Можно сказать, что последние сформировали рынок КПК. Появились вполне доступные устройства с простительными недостатками. В 1996 г. появились устройства от Microsoft, послужившие прообразом

для последующих мобильных платформ компании. Можно вспомнить и Linux-решения. Это известные некогда Sharp Zaurus различных версий. На них впервые появился встроенный жесткий диск. К началу нового столетия популярность интернет заставляет всех производителей внедрять в свои КПК поддержку интернет приложений (WWW, E-mail, SMS). Появляется возможность выходить с КПК в Интернет (через мобильный телефон), отправлять почту. Стремительная конвергенция вычислительных функций и средств связи вместе с развитием и популяризацией беспроводных сетей и мультимедийных технологий привели к тому, что в фокусе внимания потребителей оказались гибридные устройства: коммуникаторы и смартфоны.

Коммуникаторы. Название «коммуникатор» ввела в обиход компания Nokia. Nokia 9000, вышедшая в 1996 г. представляла собой очень дорогой громоздкий кирпич. Ни о каких сторонних приложениях или мультимедийности не было и речи. Зато коммуникатор обладал полноразмерной QWERTY-клавиатурой и монохромным экраном (640 x 200). А главное, по нему можно было звонить. За 9000-ой последовали модели 9000i, 9110, 9210/9210i. Последняя модель, вышедшая в 2002 г., стала культовой. Она обладала отличным на тот момент экраном, могла работать с офисными документами, архивами, графикой, интернет-страницами. Однако у коммуникаторов Nokia был недостаток – отсутствие сенсорного экрана. Это отличало их от традиционных КПК. Прочие компании пошли по другому пути. В 2002 году альянс Sony Ericsson представил модель P800 с сенсорным экраном и QWERTY-клавиатурой. Компания Handspring выпустила коммуникаторы Treo, Palm – устройство Tungsten W. Модель провалилась, результатом чего стало слияние в 2003 г. компаний Palm и Handspring. Коммуникаторы Palm Treo 600, 650 стали очень успешными. Их можно было охарактеризовать как КПК с GSM-модулем. Достоинства КПК – сенсорный экран, рукописный ввод – дополнялись возможностью просмотра интернет-страниц и работы с электронной почтой. С развитием операционной системы Pocket PC стали появляться Windows-коммуникаторы. Однако наибольший вклад в развитие Windows-устройств и коммуникаторов внесла тайваньская компания HTC (High Tech Computer Corp). Именно HTC создала сегодняшний облик коммуникаторов. Первые устройства HTC по габаритам не отличались от КПК. В 2001 г. появился HTC Wallaby. Спустя два года вышла платформа HTC Himalayas с мощным процессором, объемной памятью и свежей ОС Windows Mobile 2003 Phone Edition. Появление коммуникаторов обнажило главный недостаток КПК – отсутствие голосовой связи. С выходом в 2005 году платформы HTC Magician, коммуникаторы стали массовыми, они еще больше отдалились от КПК и, соответственно, еще больше сблизилась со смартфонами и телефонами.

Смартфоны. Есть мнение, что коммуникаторы – это КПК с телефонными функциями, а смартфоны – это телефоны с функциями КПК. Такое объяснение довольно примитивно, но отчасти верно. Внешний вид смартфонов не очень отличается от телефонов. Первым «умным телефоном» (smartphone – интеллектуальный, умный телефон) стала модель Ericsson R380s. В отличие от громоздких КПК, она имела приемлемые габариты (130 x 50 x 26 мм). Вес также был сравнительно небольшим (169 г). По внешнему виду аппарат был похож на другие модели Ericsson с флипами (R320s). Именно за флипом скрывалась главная особенность R380s – сенсорный экран. Хотя по современным меркам аппарат нельзя было считать полноценным смартфоном. Он не позволял устанавливать сторонние приложения, ОС была закрытой. Гораздо ближе понятие смартфона было для модели Nokia 7650. Многие считают ее первым «умным телефоном» на рынке. Аппарат работал на Symbian OS, позволял устанавливать сторонние приложения. Были и минусы. Например, малый объем памяти (4 Мб), отсутствие слота для карт памяти. Последовавшие смартфоны Nokia быстро стали популярными. Что выявило еще одно отличие «умных телефонов» от коммуникаторов. Смартфоны не продвигались как интеллектуальные устройства. Они продавались наравне с обычными

телефонами. По другому пути пошла компания Sony Ericsson. Ее модели P800, P900 и другие, были слишком специфичны для обычных телефонов. Эти устройства находились между коммуникаторами и смартфонами. Кроме Nokia и Sony Ericsson, Symbian-решения представляли другие производители. Siemens, Sendo, Samsung. Можно вспомнить Linux-смартфоны. Например, аппараты Motorola. С выходом ОС Windows Mobile for Smartphone появились первые Windows-смартфоны. Пробным шаром стал вышедший в 2000 г. Avenger. В 2001 г. появился Sendo Z100, в следующем году – HTC Canary. В 2003 году появилось сразу несколько смартфонов на Windows Mobile. Это HTC Tanager и Voyager, Red-E SC110, Samsung SCH-i600, Mio 8380, Motorola MPx200. «Неубиваемый мотор» стал первым официально поставляемым на российский рынок Windows Mobile-смартфоном. Где и оправдал свое прозвище. Во многом благодаря прочному корпусу и неприхотливости. С 2004 г. развитие Windows Mobile-смартфонов стало лавинообразным. Как и среди коммуникаторов, лидером была компания HTC. Ее платформы HTC Typhoon, Hurricane, Tornado и другие, распространялись под многочисленными торговыми марками. Смартфоны перестали восприниматься как просто навороченные телефоны. Возросшая функциональность ставила их в один ряд с коммуникаторами. Многие стали относить смартфоны и коммуникаторы к одному типу устройств. Это мнение усилилось после выхода линейки так называемых мультимедийных компьютеров N-series от Nokia. А также с развитием корпоративных QWERTY-смартфонов. Эти устройства были наследниками BlackBerry от канадской компании RIM. Но нельзя однозначно приравнять смартфоны и коммуникаторы. Это пока различные устройства. Они обладают как сходными, так и отличительными чертами.

Сравнение смартфонов и коммуникаторов. Внешний вид смартфонов и коммуникаторов чрезвычайно разнообразен. Для коммуникаторов пестрота дизайна объясняется различием устройств ввода. Это может быть рукописный ввод, QWERTY-клавиатура, виртуальная клавиатура на экране, цифровая клавиатура. Типичный современный коммуникатор – это моноблок или сайд-слайдер с выдвинутой QWERTY-клавиатурой. У него большой сенсорный экран (порядка 2,8 дюйм), размеры и вес чуть больше среднего для телефонов. Большинство коммуникаторов не имеют цифровой клавиатуры. Наличие QWERTY-клавиатуры также не обязательно. Главное отличие коммуникаторов от смартфонов – наличие сенсорного экрана. Есть и исключения. Это коммуникаторы Nokia – 9300, 9300i, 9500, E90. Последний, кстати, построен на «смартфонной» платформе Symbian OS S60. Среди смартфонов также есть обладатели сенсорных экранов. Это продукты Sony Ericsson – P990i, M600i, W950i. Хотя их трудно однозначно причислить к смартфонам. Это гибридные, смежные с коммуникаторами устройства. Типичный смартфон – это устройство сравнимых с телефонами габаритов. Оно обладает крупным дисплеем (2–2,4 дюйма) и цифровой клавиатурой. Наиболее часты типичные форм-факторы. Это моноблок, раскладушка, слайдер.

Коммуникаторы предназначены для работы с сенсорным экраном. Отсюда необходимость использования стилуса, перьевого ввода. Следовательно, при работе необходимо задействовать обе руки. Со смартфонами можно управиться и одной рукой. Джойстик или навигационная клавиша дополняются цифровой клавиатурой. На ее кнопки можно назначить различные действия. Что облегчает управление. В последних моделях коммуникаторов также облегчена работа одной рукой. Расширено использование аппаратных клавиш, других элементов управления. Это еще один довод в пользу неизбежности слияния смартфонов и коммуникаторов.

Основа комфортной работы с мобильным устройством – производительный процессор и емкая память. Большинство современных смартфонов и коммуникаторов оснащены 64 Мб ОЗУ и 128 Мб ПЗУ. При наличии 64 Мб памяти и процессоре 200 МГц комфортная работа обеспечена. Хотя наращивание характеристик продолжается. Уже в 2007 году коммуникаторы смогут похвастать памятью 128 Мб

ОЗУ и 256 ПЗУ (линейка Ultimate от компании i-mate). Ее всегда можно расширить с помощью карт памяти. Ситуация для Symbian-устройств схожа с Windows Mobile. С той разницей, что объем доступной пользователю памяти различен. Он может составлять от нескольких десятков мегабайт (бюджетные смартфоны Nokia) до 160 Мб (Nokia N95). Или еще больше, если используется жесткий диск или емкая флэш-память (Nokia N91, Sony Ericsson W950i). Таким образом, по объему памяти смартфоны догнали коммуникаторы. Увеличение объемов будет идти параллельно для обоих типов устройств.

Процессоры, установленные в смартфонах и коммуникаторах, разнообразны. Сейчас нельзя сказать, какой тип устройств мощнее. В коммуникаторах в основном используются три типа процессоров: Texas Instruments (TI) OMAP 850 с тактовой частотой 200 МГц, Intel XScale PXA272 416 МГц и Samsung SC32442 400 МГц. Новинки 2007 года располагают более мощными процессорами. Такими, как Intel XScale PXA 272 520 МГц. Рекордсмен на сегодняшний момент – HTC X7500 Advantage. Частота его процессора – целых 624 МГц. Единственный коммуникатор на Symbian OS S60 Nokia E90 также обладает шустрым процессором. Это TI OMAP 2420 330 МГц. В смартфонах в основном используются два типа процессоров. В многочисленных аппаратах Nokia – TI OMAP 1710 220 МГц. В большинстве Windows Mobile-смартфонов – TI OMAP 850 200 МГц. Модели Sony Ericsson оснащались процессорами Philips с тактовой частотой 208 МГц. Топовые смартфоны Nokia (N93/N93i, N95) перешли на производительный процессор TI OMAP 2420 330 МГц, который позволяет снимать видео в VGA-качестве и играть в трехмерные игры. Самым мощным процессором среди Symbian-устройств обладает Nokia 6110 Navigator. Это 369 МГц-решение от Freescale. Производительный процессор позволяет обеспечить комфортную работу с GPS-модулем. Среди Windows Mobile-смартфонов можно выделить Samsung i600, оснащенный Intel XScale PXA 272 416 МГц. Такому могут позавидовать многие владельцы коммуникаторов. Однако производительность зависит от тактовой частоты процессора нелинейно. 330 МГц процессор в Nokia N95 по быстродействию превосходит иные 520 МГц-решения в коммуникаторах. Не последнюю роль играет оптимизация ОС, размеры и разрешение экранов. Таким образом, по производительности или объему памяти оба типа мобильных устройств похожи. Конечно, если рассматривать аппараты со сравнимым позиционированием.

Можно сказать, что смартфоны догнали по функциональности коммуникаторы. Как и среди обычных телефонов, преобладают аппараты с GPRS/EDGE. Среди обоих типов устройств есть модели с поддержкой UMTS. Наличие Bluetooth – обязательное условие для смартфонов и коммуникаторов. За последние два года появилось много смартфонов с поддержкой Wi-Fi. Их количество только возрастает. В этом году появились первые смартфоны с GPS-модулем (Nokia N95, 6110 Navigator). Безусловными лидерами продаж на рынке смартфонов и коммуникаторов остаются продукты Nokia (рис. 2) – второй год подряд их доля на рынке составляет более 50%.

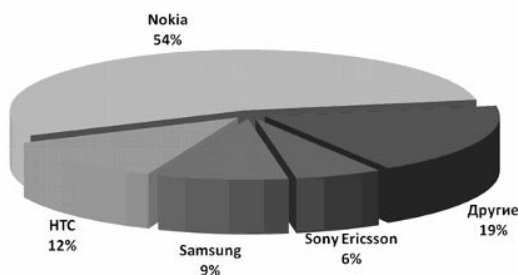


Рис. 2. Количество продаж смартфонов и коммуникаторов разных производителей [8]

В нишу КПК, кроме смартфонов и коммуникаторов, начинают вторгаться и другие мобильные устройства. Например, мультимедийные GPS-навигаторы, различные компактные мультимедийные плееры. А также – в более отдаленной перспективе – субноутбуки UMPC. Сейчас это довольно неуклюжие и дорогие устройства весом в полкило. Однако с уменьшением габаритов они способны занять место КПК.

Нельзя забывать и о последних событиях в мире телефонов. В частности, выходе iPhone. Один из продуктов Apple, iPod, уже изменил рынок mp3-плееров. iPhone также по силам внести свежую струю в облик будущих «умников», если мы дождемся его второй версии. Уже сейчас появляются устройства с сенсорным экраном и пальцевым вводом, без использования стилуса. Особенностью таких аппаратов является переработанный интерфейс стандартных ОС для более удобной работы с устройством. Возможно, будущий смартфон/коммуникатор будет напоминать HTC Touch? Будущее покажет. А пока остается следить за существующими смартфонами и коммуникаторами. Ведь среди них множество интересных моделей, чьи возможности могут удовлетворить любые запросы.

Заключение

Таким образом, сравнение показало, что смартфоны и коммуникаторы не сильно отличаются друг от друга. Конструктивные особенности касаются в основном типа экрана (сенсорный или нет). Есть предпосылки для дальнейшего объединения обоих типов устройств в один класс. Уже сейчас различия лежат, скорее, в области маркетинга, позиционирования устройств. Фактически это встречные эволюционные пути, возникшие на рынке с маркетинговой подачи ряда производителей, и, хотя в будущем на пересечении коммуникаторов, смартфонов и ультракомпактных ноутбуков наверняка возникнет некий объединенный класс устройств, пока что все они мирно сосуществуют друг с другом на рынке.

Литература

1. Басина Н. (2001) Российский рынок карманных компьютеров: Состояние. Особенности. Развитие, ИнфоБизнес, N1 (146), 16 января 2001, <http://offline.ibusiness.ru>
2. Хавжу Д (2002) Несколько поправок к маркетинговому исследованию, <http://www.hpc.ru/lib/arts/1821/printable.shtml>
3. Итоги исследования российского рынка КПК за 2002 г., <http://www.handy.ru/a/2003/03/25/1003.html>
4. Итоги исследования российского рынка КПК за 2003 г., <http://www.handy.ru/a/2004/01/19/1799.html>
5. Российский рынок КПК и смартфонов в 2004 году, <http://www.handy.ru/a/2004/12/22/2416.html>
6. SmartMarketing объявляет оценки рынка КПК и смартфонов за 2005 г, http://www.smartmarketing.ru/pda_smartphone_y2005
7. Российский рынок смартфонов, коммуникаторов и КПК в 2006 г., <http://www.handy.ru/a/2007/03/06/6750.html>
8. КПК, коммуникаторы и смартфоны в России. 2007 год, <http://www.handy.ru/a/2008/03/20/7277.html>
9. Рынок смартфонов и коммуникаторов в России: 3,4 млн. проданных устройств в 2008 году <http://www.handy.ru/a/2009/01/13/7470.html>
10. Федоров М. КПК: исторический экскурс, <http://www.ixbt.com/pda/pda-history.shtml>
11. История Карманных Персональных Компьютеров <http://informika.net.ru/jel/txt/127.php>
12. Как появились карманные компьютеры, <http://www.ihand.ru/history.shtml>
13. Эволюция карманных компьютеров на примере Palm OS, <http://kpk-user.ru/articles>
14. PDA Компьютеры, <http://www.pda-book.ru/p2.html>

ЦЕЛОЧИСЛЕННОЕ РЕШЕНИЕ ЗАДАЧ ОПТИМИЗАЦИИ КОМПЬЮТЕРНЫХ СИСТЕМ В СРЕДЕ MATHCAD

А.В. Осипов

**(Санкт-Петербургский государственный инженерно-экономический университет)
Научный руководитель – д.т.н., профессор В.А. Богатырев**

В работе исследованы возможности среды MathCAD для решения задач оптимального надежностного проектирования вычислительных систем. Разработан алгоритм целочисленного решения оптимизационной задачи средствами программирования в MathCAD.

Ключевые слова: надежность, оптимизация, отказоустойчивость, MathCAD, целочисленность

Введение

Одной из основных проблем построения вычислительных систем остается задача обеспечения их продолжительного функционирования. Эта задача имеет три составляющих: надежность, готовность и удобство обслуживания. Все эти три составляющих предполагают, в первую очередь, борьбу с неисправностями системы, порождаемыми отказами и сбоями в ее работе.

Повышение надежности основано на принципе предотвращения неисправностей путем снижения интенсивности отказов и сбоев за счет применения электронных схем и компонентов с высокой и сверхвысокой степенью интеграции, снижения уровня помех, облегченных режимов работы схем, обеспечения тепловых режимов их работы, а также за счет совершенствования методов сборки аппаратуры.

Отказоустойчивые системы имеют в своем составе избыточную аппаратуру для всех функциональных блоков, включая процессоры, источники питания, подсистемы ввода/вывода и подсистемы дисковой памяти. В отказоустойчивых системах избыточные аппаратные средства можно использовать для распараллеливания вычислительных работ.

Надежность информационной системы зависит от простоты архитектуры, надежности элементов, наличия избыточности, управляемости.

Проектирование компьютерных систем связано с решением задачи их структурной и параметрической оптимизации. Решение оптимизационной задачи с использованием средств MathCAD сопряжено с трудностями получения целочисленного решения, когда искомые параметры оптимизации находятся над знаком суммы целевой функции.

Оптимизационная задача проектирования вычислительной системы

В работе ставится задача исследования возможностей среды MathCAD для решения задач оптимального надежностного проектирования вычислительных систем.

В качестве объекта оптимизации рассмотрим многоуровневую компьютерную систему [1].

Исследуемая система представлена на рис. 1. Имеется m объектов управления, которые управляются контроллерами. Коммутаторы связывают контроллеры с управляющими компьютерами верхнего уровня. Кратность резервирования контроллеров – n_1 . Кратность резервирования коммутаторов – n_2 , а компьютеров верхнего уровня – n_3 . Будем считать, что для функционирования системы необходимо функционирование хотя бы одного контроллера для каждого объекта управления, хотя бы одного коммутатора и одного компьютера верхнего уровня. Заданы стоимость и вероятность безотказ-

ной работы контроллера, коммутатора и компьютера верхнего уровня, а также лимит стоимости всей системы. Требуется найти такую конфигурацию системы, которая обеспечит максимальную вероятность безотказной работы.

В результате оптимизации находится число узлов на различных уровнях, при котором достигается максимум надежности системы при соблюдении ограничения ее общей стоимости.

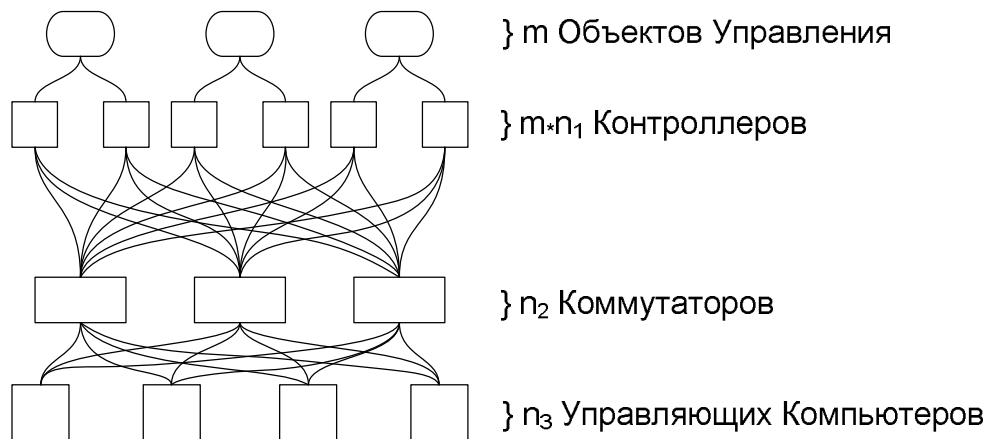


Рис. 1. Организация системы управления

Решение оптимизационной задачи

Исходя из перечисленных условий, вероятность безотказной системы определяется следующей функцией:

$$P(n_1, n_2, n_3) = \left(\sum_{i=1}^{n_1} C_{n_1}^i P_1^i (1-P_1)^{n_1-i} \right)^m \left(\sum_{i=1}^{n_2} C_{n_2}^i P_2^i (1-P_2)^{n_2-i} \right) \left(\sum_{i=1}^{n_3} C_{n_3}^i P_3^i (1-P_3)^{n_3-i} \right),$$

где P_1 – вероятность безотказной работы контроллера; P_2 – вероятность безотказной работы коммутатора; P_3 – вероятность безотказной работы управляющего компьютера; n_1 – количество контроллеров; n_2 – количество коммутаторов; m – количество объектов управления; n_3 – количество управляющих компьютеров.

Для соблюдения ограничения стоимости системы необходимо выполнение неравенства:

$$m \cdot n_1 c_1 + n_2 c_2 + n_3 c_3 \leq c_0,$$

где c_1 – стоимость одного контроллера; c_2 – стоимость одного коммутатора; c_3 – стоимость одного хранилища; c_0 – лимит общей стоимости управляющего компьютера.

Задача состоит в максимизации функции вероятности безотказной работы системы.

Пакет MathCAD обладает встроенными средствами решения задач оптимизации. Для этих целей в MathCAD имеются функции:

- Minimize(f, x₁, x₂, ...);
- Maximize(f, x₁, x₂, ...).

Эти функции возвращают значения переменных x_1, x_2, \dots , которые удовлетворяют условиям в блоке Given и при которых функция f принимает свое минимальное или максимальное значение соответственно.

Однако эти средства обладают ограниченными возможностями, а именно:

1. Встроенные средства MathCAD не позволяют найти целочисленное решение оптимизационных задач.

2. Если аргумент x оптимизируемой функции находится над знаком суммы $(\sum_{i=i_0}^x y)$, то MathCAD не производит оптимизацию по этому аргументу.

В статье предлагается алгоритм и программа для целочисленного решения поставленной оптимизационной задачи средствами MathCAD.

MathCAD- программа поиска максимума целевой функции:

$$P(n_1, n_2, n_3) = \left(\sum_{i=1}^{n_1} C_{n_1}^i P_1^i (1-P_1)^{n_1-i} \right)^m \left(\sum_{i=1}^{n_2} C_{n_2}^i P_2^i (1-P_2)^{n_2-i} \right) \left(\sum_{i=1}^{n_3} C_{n_3}^i P_3^i (1-P_3)^{n_3-i} \right).$$

Поиск решения:

$$\begin{array}{l}
 P_0 \leftarrow 0 \\
 \text{for } i_1 \in 1, 2, \dots, \left\lfloor \frac{c_0}{mc_1} \right\rfloor \\
 \quad \text{for } i_2 \in 1, 2, \dots, \left\lfloor \frac{c_0 - mc_1 i_1}{c_2} \right\rfloor \text{ if } \left(\frac{c_0 - mc_1 i_1}{c_2} \right) \geq 1 \\
 \quad \quad \text{for } i_3 \in 1, 2, \dots, \left\lfloor \frac{c_0 - mc_1 i_1 - c_2 i_2}{c_3} \right\rfloor \text{ if } \left(\frac{c_0 - mc_1 i_1 - c_2 i_2}{c_3} \right) \geq 1 \\
 \quad \quad \quad \text{if } P_0 < P(i_1, i_2, i_3) \\
 \quad \quad \quad \quad y_0 \leftarrow i_1 \\
 \quad \quad \quad \quad y_1 \leftarrow i_2 \\
 \quad \quad \quad \quad y_2 \leftarrow i_3 \\
 \quad \quad \quad \quad P_0 \leftarrow P(i_1, i_2, i_3) \\
 y
 \end{array}$$

Пример расчета

Приведем пример расчета при следующем блоке начальных условий:

$$P_1 = 0,50 \quad c_1 = 200 \quad n_1 = 1$$

$$P_2 = 0,50 \quad c_2 = 200 \quad n_2 = 1$$

$$P_3 = 0,50 \quad c_3 = 200 \quad n_3 = 1$$

$$m = 5 \quad c_0 = 10000$$

Результат:

$$y = \begin{pmatrix} 7 \\ 5 \\ 3 \end{pmatrix}.$$

$$c_0 - (my_0c_1 + y_1c_2 + y_2c_3) = 0.$$

$$P(y_0, y_1, y_2) = 0,953.$$

Таким образом, получено целочисленное решение поставленной задачи оптимизации.

Анализ результатов

Результаты проведенных с помощью предлагаемой программы исследований сведены в таблицу.

Таблица. Результаты расчетов

Кол-во ОУ	Исходные условия							Результат расчетов				
	Стоимость элементов системы			Вероятность безотказной работы элементов			Лимит стоимости системы	Количество элементов			Вероятность безотказной работы	
	c_1	c_2	c_3	P_1	P_2	P_3	c_0	n_1	n_2	n_3	P	
5	200	300	500	0.50	0.75	0.80	10000	7	5	3	0.953	
3	400	320	550	0.50	0.80	0.60	11000	6	3	5	0.937	
10	150	430	750	0.50	0.75	0.80	12000	4	5	5	0.984	

Результаты исследования представлены графиками на рис. 2–4. Рис. 2 показывает, как изменяется число контроллеров, коммутаторов и управляющих компьютеров, соответствующих максимуму целевой функции, в зависимости от вероятности безотказной работы контроллеров. Из рисунка следует, что при меньшей вероятности безотказной работы контроллеров следует выбирать конфигурацию системы с большим количеством контроллеров, чтобы компенсировать их низкую надежность.

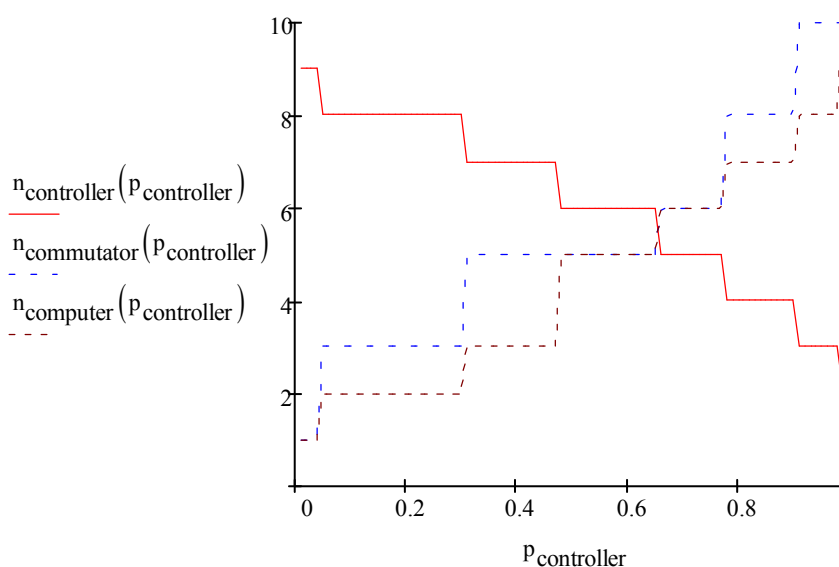


Рис. 2. Зависимость оптимального числа элементов системы от вероятности безотказной работы контроллеров

На рис. 3 отражена зависимость оптимального количества контроллеров, коммутаторов и управляющих компьютеров от стоимости коммутаторов. Рис. 3 показывает, что при повышении стоимости коммутаторов их количество в оптимальной конфигурации системы уменьшается, так как использование дополнительных коммутаторов становится неэффективным вследствие увеличения их стоимости.

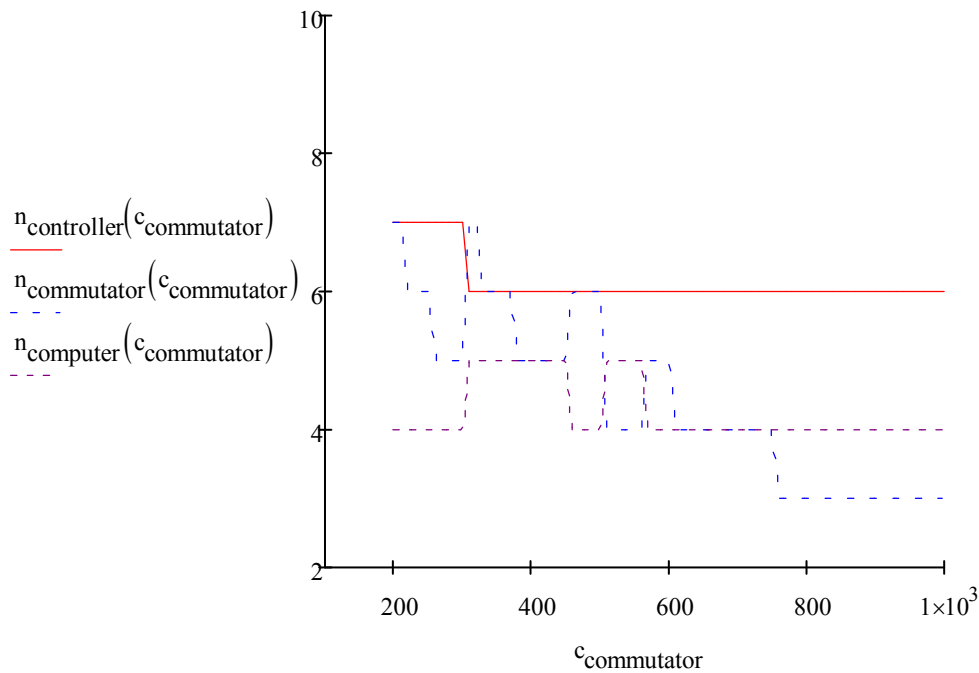


Рис. 3. Зависимость оптимального числа элементов системы от стоимости коммутаторов

Рис. 4 иллюстрирует, что разработанный алгоритм позволяет получать результат за приемлемое время (ожидание не более 5 секунд) при небольшом количестве элементов в системе (менее 50 элементов).

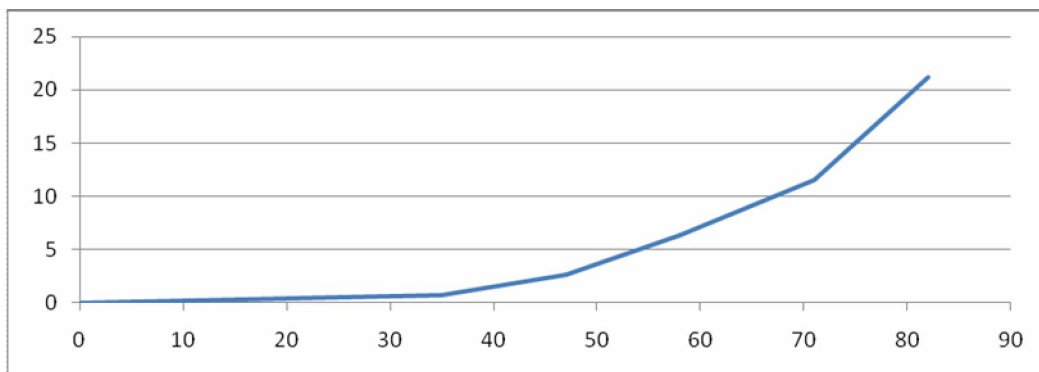


Рис. 4. Зависимость времени выполнения алгоритма (в секундах) от общего количества элементов в системе

Заключение

Таким образом, предложен алгоритм и его программная реализация по решению оптимизационной целочисленной задачи проектирования вычислительных систем, в среде MathCAD, что в отличие от использования традиционных для решения задач оптимизации функций $\text{Minimize}(f, x_1, x_2, \dots)$ / $\text{Maximize}(f, x_1, x_2, \dots)$ позволяет найти не только целочисленное решение, но и решение, если искомые переменные находятся над знаком суммы.

Литература

1. Богатырев В.А., Богатырев С.В. Векторная оптимизация структуры кластера //Сборник научных трудов \ \ Информационные системы и технологии: теория и практика. ЛТА. – 2008. – С. 19–27.
2. Богатырев В.А. Оценка надежности и оптимальное резервирование кластерных компьютерных систем// Приборы и системы. Управление, контроль, диагностика \ \ 2006. – №10. – С. 18–21.
3. Макаров Е.Г. Mathcad: учебный курс – СПб: Питер. – 2009. – 384 с.

ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ В МОБИЛЬНЫХ ТЕХНОЛОГИЯХ

Н.С. Токалов

Научный руководитель – А.П. Ищенко

В статье рассмотрены вопросы, касающиеся обеспечения безопасности в мобильных технологиях с учетом исторического развития стандартов беспроводных сетей. Описаны основные меры применимые для защиты беспроводных сетей.

Ключевые слова: безопасность, беспроводные сети, 802.11, IEEE, защита, стандарты связи

Введение

За последние несколько лет мобильные технологии получили широкое распространение во всём мире. Однако, уже сама их распространенность порождает проблему их безопасности. Из-за общедоступного характера радиоспектра мобильных сетей они подвержены не только уже хорошо известным уязвимостям, но также и уникальным вирусам, характерным только для мобильной области (рис. 1), и, кроме того, могут пасть жертвой непосредственных атак со стороны хакеров или преступных группировок (табл. 1).

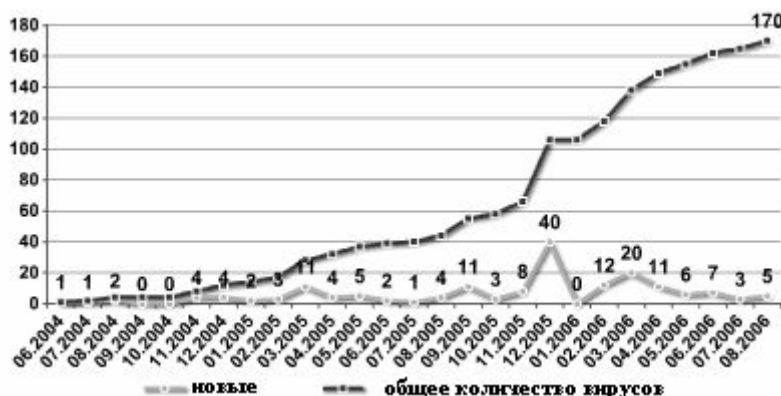


Рис. 1. Рост количества разновидностей известного мобильного вредоносного ПО [1]

Основная часть

Чтобы подслушивать сообщения в проводной сети, необходим физический доступ к такому сетевому компоненту, как точка подсоединения устройства к локальной сети. Для беспроводной сети нужен только приемник, такой как обычный сканер частот. В настоящей работе будут рассмотрены основные механизмы обеспечения безопасности в беспроводных сетях.

Остановимся теперь на технологиях, которые обеспечивают защиту от вышеперечисленных атак. Изначально для защиты использовался лишь SSID (имя сети). Но, вообще говоря, защитой такой способ можно называть с большой натяжкой – SSID передается в открытом виде и никто не мешает злоумышленнику его подслушать, а потом подставить в своих настройках нужный. Не говоря уже о том, что (это касается точек доступа) может быть включен широковещательный режим для SSID, т.е. он будет принудительно рассылаться в эфир для всех слушающих. Поэтому возникла потребность именно в шифровании данных.

Тип атаки	Атакуемые элементы сети	Цель атакующего
Черви, вирусы, трояны, спам посредством SMS/MMS	Другие пользователи сети, сервера с мобильным контентом	Отказ в обслуживании (DoS).
DoS, SYNflood, атаки на уровне приложений (на сервера RADIUS, переполнение буфера, RTP flooding, и т.д.)	HLR, AAA, сервера с контентом, модуль обработки вызовов (signaling node)	Сделать невозможной работу сети
Overbilling attack	Элементы управления сетью (AAA, HLR, VLR, и т.д.)	Мошенничество
Подделка PDP контекста	User-sessions	Незаконное получение доступа к услугам
Атаки на сигнальном уровне (SIGTRAN, SIP)	Signaling nodes	Сделать невозможной работу сети

Таблица 1. Типы атак, которым подвергаются мобильные операторы [8]

В настоящее время стандарты беспроводных сетей 802.11 a/b/g предусматривают несколько механизмов обеспечения безопасности: режим аутентификации и шифрования данных по протоколу WEP (Wired Equivalent Privacy); режим аутентификации и шифрования данных по протоколу WPA (Wi-Fi Protected Access); фильтрация по MAC адресам, режим скрытого идентификатора сети.

Все современные мобильные устройства поддерживают WEP, который был изначально заложен в спецификацию беспроводных сетей IEEE 802.11. Шифрование по протоколу WEP осуществляется с помощью 44 или 104-битного ключа (поточное шифрование с использованием алгоритма RC4 на статическом ключе). Ключ представляет собой набор ASCII-символов длиной 5 или 13 символов, которые переводятся в последовательность шестнадцатеричных цифр. Кроме ASCII-фразы (статической составляющей ключа) есть еще Initialization Vector (IV). Он служит для рандомизации оставшейся части ключа. IV выбирается случайным образом и динамически меняется во время работы. Длина IV равна 24 битам (примерно 16 миллионов комбинаций), поэтому общая длина ключа получается равной 64 (40+24) или 128 (104+24) бит. Используемый алгоритм шифрования не является стойким поскольку ключ можно подобрать перебором. Другая уязвимость WEP связана с тем, что в реальной работе все возможные варианты ключей будут использованы за промежуток от десяти минут до нескольких часов (для 40-битного ключа). После этого вектора начнут повторяться. Злоумышленнику стоит лишь набрать достаточное количество пакетов и найти эти повторы. После этого подбор статической составляющей ключа (ASCII-фразы) не занимает много времени. Но это еще не все. Существуют так называемые «нестойкие» вектора инициализации. Использование подобных векторов в ключе дает возможность злоумышленнику практически сразу приступить к подбору статической части ключа. В настоящее время некоторые производители беспроводного оборудования предлагают «расширенные варианты» алгоритма WEP – в них используются ключи длиной более 128 бит. Но в этих алгоритмах увеличивается лишь статическая составляющая ключа. Длина IV остается той же самой со всеми вытекающими отсюда последствиями.

В 2003 году был предложен новый алгоритм шифрования WPA, главной особенностью которого является технология динамической генерации ключей шифрования данных на базе протокола TKIP (Temporal Key Integrity Protocol). По протоколу TKIP сетевые устройства работают с 48-битовым IV и реализуют правила изменения последовательности его битов, что исключает повторное использование ключей. Предусмот-

рена генерация 128-битного ключа для каждого передаваемого пакета. Кроме того, контрольные криптографические суммы рассчитываются по методу MIC (Message Integrity Code). Достоинства TKIP не покрывают его основной недостаток – используемый алгоритм RC4. Поэтому на замену TKIP приходит AES (Advanced Encryption Standard), который отличается более стойким, по сравнению с протоколами WEP и TKIP, криптоалгоритмом.

Фильтрация MAC-адресов, которая поддерживается всеми современными точками доступа и беспроводными маршрутизаторами, хотя и не является составной частью стандарта 802.11, тем не менее позволяет повысить уровень безопасности беспроводной сети. Для реализации данной функции в настройках точки доступа создается таблица MAC-адресов беспроводных адаптеров клиентов, авторизованных для работы в данной сети.

Еще одна мера предосторожности, которую часто используют в беспроводных сетях, – это режим скрытого идентификатора сети. Каждой беспроводной сети назначается свой уникальный идентификатор (SSID), который представляет собой название сети. При попытке пользователя войти в сеть драйвер беспроводного адаптера прежде сканирует эфир на предмет наличия в ней беспроводных сетей. При использовании режима скрытого идентификатора (Hide SSID) сеть не отображается в списке доступных и подключиться к ней можно только в том случае, если, во-первых, точно известен ее SSID, а во-вторых, заранее создан профиль подключения к этой сети.

Функции аутентификации возлагаются на протокол EAP, который сам по себе является лишь каркасом для методов аутентификации. Методов же аутентификации, которых существует довольно много: EAP-SIM, EAP-AKA – используются в сетях GSM мобильной связи; LEAP – проприетарный метод от Cisco systems; EAP-MD5 – простейший метод (не стойкий); EAP-MSCHAP V2 – метод аутентификации на основе логина/пароля пользователя в MS-сетях; EAP-TLS – аутентификация на основе цифровых сертификатов; EAP-SecureID – метод на основе однократных паролей; EAP-TTLS и EAP-PEAP, которые образуют TLS-туннель между клиентом и сервером аутентификации, что повышает безопасность аутентификации.

Объем мирового рынка ПО для обеспечения информационной безопасности за 2007г. вырос до 10,4 млрд долл., что на 19,8% выше показателя 2006г.

Какие же меры можно применить для защиты беспроводной сети? Прежде всего, смените настройки умолчанию: пароль администратора, имя пользователя, имя сети на точке доступа, отключите доступ к ее настройкам через беспроводное соединение. Обновите прошивку и, если нужно, оборудование. Отключите широковещание SSID. Выключайте сеть, когда не работаете. Снижение мощности передачи. Шифрование (WEP, WPA или WPA2). Можно включить аутентификацию пользователей по паролю и фильтрацию по MAC адресу. Дополнительным средством защиты беспроводной сети от взлома может послужить установка VPN (Virtual Private Network), которые позволяют формировать защищенные туннели для обмена данными. Как минимум, нужно использовать обновлённые антивирусные решения и регулярно обновлять базы. Шифрование файлов с использованием криптостойких алгоритмов. Из минусов подобной защиты можно отметить, что она требует ресурсов процессора, а это может существенно замедлить работу с файлами.

Заключение

Безопасности беспроводных сетей стоит уделять особое внимание. К счастью, в настоящее время существуют множество различных способов защиты, а, при условии правильной настройки, вполне возможно построить на базе стандартов серии 802.11x защищенную и устойчивую к атакам беспроводную сеть и быть уверенным в обеспече-

нии необходимого уровня безопасности. Если вы не сделаете защиту самостоятельно, то никто не сделает это за вас. Так что взвесьте все "за" и "против" и защитите свою сеть!

Литература

1. Gostev A. (2006) Mobile Malware Evolution: An Overview, Part 1., Kaspersky Lab, <http://www.viruslist.com/en/analysis?pubid=200119916>
2. Ou G. (2005) Understanding the updated WPA and WPA2 standards, <http://blogs.zdnet.com>
3. Владимиров А. (2006) Атакуем клиентские устройства на Wi-Fi сетях, <http://www.inattack.ru>
4. Гайкович В., Ершов Д. (1997) Гибкое управление средствами защиты - необходимое условие их успешного применения, Системы безопасности, связи и телекоммуникаций.
5. Гордейчи С., Дубровин В. Безопасность беспроводных сетей». – М: Горячая Линия-Телеком. – 2008. – 288 с.
6. Зайцев Е. (2005) Защита беспроводных сетей, WPA: теория и практика www.ixbt.com/comm/prac-wpa-eap.shtml
7. Мерритт М., Поллино Д. Безопасность беспроводных сетей. Пер. с англ. Семенова А.В., М. – Andair. – 2008.
8. Нечай О. (2006) Безопасность беспроводной сети, www.terralab.ru
9. Подворчан Н. (2007) Безопасность в мобильных сетях 3го поколения: угрозы, возникающие при передаче данных через IP <http://www.re.mipt.ru/infse>
10. Стейнхарт М.Д. (2007) Безопасность сетей и беспроводных устройств <http://www.pcmag.ru>
11. Феллинг Д. (2006) Безопасность сетей 802.11g – это просто, <http://www.osp.ru>
12. Ференец В. (2008) Средства защиты информации и бизнеса 2008, <http://www.cnews.ru>

УПРАВЛЕНИЕ РАБОТОЙ КОМПЬЮТЕРНОГО КЛАССА, ПОСТРОЕННОГО НА ОСНОВЕ ТЕРМИНАЛЬНОЙ СИСТЕМЫ

А.А. Скшидлевский

Научный руководитель – к.т.н., доцент А.В. Лямин

В статье описана централизованная система управления работой компьютерного класса, построенного на основе терминальной системы Sun Ray. Система управления автоматизирует процесс распределения ограниченного количества компьютеров между большим количеством пользователей, что позволяет повысить эффективность терминальной системы и снизить затраты, связанные с контролем регламента работы пользователей.

Ключевые слова: терминальная система Sun Ray, компьютерный класс, система управления

Введение

Терминальная система Sun Ray от компании Sun Microsystems выполнена по технологии «тонкого клиента». Она обеспечивает доступ ко всем приложениям, исполняемым на платформе SPARC/Solaris. Система Sun Ray реализует архитектуру Sun Ray Hot Desk, являющуюся очень экономичной и безопасной вычислительной средой. В этой архитектуре вся вычислительная нагрузка сконцентрирована на одном или нескольких серверах, в то время как клиентская сторона используется лишь для покадрового отображения графической сессии и пересылки на сервер вводимой пользователем информации. На настольной системе не устанавливается ни операционная система (ОС), ни клиентское программное обеспечение. Графическая сессия передается по специальному протоколу, позволяющему использовать на клиентской системе недорогой процессор и минимальный объем оперативной памяти. Для обеспечения описанного взаимодействия на сервере устанавливается специальное программное обеспечение. Оно выполняет функции аутентификации и управления сессиями пользователей. Производительность современных серверов и технология Hot Desk позволяют эффективно использовать разделяемые ресурсы сервера, обеспечивая при этом высочайший уровень надежности и безопасности.

Достоинствами терминальной системы являются:

- длительной срок эксплуатации;
- минимальная стоимость поддержки;
- централизованное администрирование;
- высокий уровень доступности и надежности.

Тем не менее, использование такой архитектуры в общественных компьютерных классах затруднено из-за отсутствия эффективной централизованной системы управления подобными комплексами. Эффективность использования терминальной системы может быть значительно повышена за счет совершенствования механизмов управления и контроля:

- управление включением и выключением терминалов;
- управление продолжительностью работы пользователей;
- контроль за действиями пользователей во время работы и управление доступом к ресурсам.

Эти функции решает централизованная система управления терминальными системами, разработанная в Санкт-Петербургском государственном университете информационных технологий, механики и оптики (СПбГУ ИТМО). Для защиты от несанкционированного доступа в системе реализовано:

- изолированная среда для работы с системой;

- привязка пользователей к рабочей станции;
- учет и протоколирование использования рабочих станций.

В СПбГУ ИТМО система управления используется с 2003 года в компьютерном классе, рассчитанном на одновременную работу шестидесяти пользователей. Основная задача класса – обеспечение самостоятельной работы студентов в среде системы дистанционного обучения [1, 2] и проведение компьютеризированных аттестаций [3, 4]. Эксплуатация системы в течение длительного времени показала, что она является эффективным инструментом управления терминальной системой в условиях, когда количество потенциальных пользователей значительно превосходит количество рабочих мест, и позволяет снизить затраты, связанные с контролем регламента работы пользователей.

В статье представлено описание централизованной системы управления работой компьютерного класса, построенного на основе терминальной системы Sun Ray. Перечислены основные функции системы, описана схема системы управления и приведены правила ее администрирования.

Функции системы управления

Особенности аппаратной реализации терминала Sun Ray позволяют ограничить несанкционированное перемещение информации и установку пользователями программных средств: в клиентской станции отсутствует возможность несанкционированного подключения устройств ввода-вывода. Все драйверы и все программное обеспечение устанавливается на сервере системным администратором, который управляет правами доступа и может централизованно контролировать поток данных в сети. Операционная система Solaris, управляющая работой настольных систем, не позволяет пользователям нарушить работу всей системы в целом, например, занесением программ-вирусов, как это может произойти в случае использования персональных компьютеров. В традиционной системе, работающей под управлением ОС Solaris, внесение вируса может привести лишь к потере информации самого пользователя, занесшего вирус [5].

Централизованная система управления работой компьютерных классов заточена на работу в учебных заведениях и поэтому имеет ряд специфических для этой области возможностей. Вся типовая работа с системой ведется через веб-интерфейс. Для начала работы с системой необходимо пройти авторизацию. Для каждого заранее зарегистрированного пользователя назначается одна из трех ролей: студент, инструктор или администратор.

Роль «студент» имеет наименьшие права и используется для аренды терминала в классе. Для этого в веб-интерфейсе «студента» есть ссылка «подать заявку», с помощью которой пользователь может арендовать для себя терминал на указанные день и час. У каждого пользователя с ролью «студент» есть свой лимит времени, который измеряется в количестве сеансов по 40 минут. Пользователь с этой ролью может оставлять сообщения инструктору.

Пользователи с ролью «инструктор» выполняют функции дежурных, которые отмечают на входе в класс пришедших пользователей и распределяют их по терминалам. Для этого у них есть веб-интерфейс, где день разделен на сеансы, а в каждом сеансе находится список пользователей, которые были записаны на это время, и номера их терминалов. На рис. 1 представлен интерфейс инструктора для управления пользователями. Пользователи, выделенные жирным шрифтом присутствуют в классе и для них включен компьютер, то есть инструктор их уже отметил как пришедших. В верхней части на светлом фоне находится список пользователей, которые записаны по заявке на текущий сеанс. В нижней части списка на темном фоне находятся пользователи с прошлых сеансов, для которых время сеанса закончилось, но они еще не зафиксировали

свой уход. Такая структура интерфейса инструктора помогает быстро обслуживать пользователей и не создает путаницы. Пришедший в класс пользователь обращается к инструктору, тот находит пользователя в списке и отмечает его прибытие флажком в интерфейсе. После чего пользователь проходит за свой терминал, который автоматически включается, и может начинать работу. Время окончания сеанса пользователя отображается на экране в виде таймера с обратным отсчетом. По окончании работы за терминалом инструктор отмечает уход пользователя флажком в интерфейсе, терминал автоматически выключается. Также терминал автоматически отключается при окончании времени аренды терминала.

Расписание аттестаций

Список студентов, Вторник (30.12.2008), 16:00

№	Фамилия, И., О.	Группа	Начало	Окончание	№ комп.	Вошёл	Вышел
1	Абакумова К. Л.	2707	14:00	17:20	23	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	Абросимов Д. Е.	1165	13:20	18:00	16	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	Андасов Р. Э.	3320	14:40	18:00	53	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Афанасьев П. И.	2652	16:00	18:00	39	<input type="checkbox"/>	<input type="checkbox"/>
-- вырезано --							
15	Гобарев С. Г.	1651	11:20	16:40	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	Смирнов П. И.	2743	15:20	16:00	31	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	Шапошник А. А.	1750	15:20	16:00	29	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Количество студентов в классе: 6

Текущий сеанс Обновить Вернуться назад

Рис. 1. Интерфейс управления пользователями для роли «инструктор»

Веб-интерфейс для роли «инструктор» имеет следующий функционал. Инструктор может изменять статус терминала, используя ссылку в интерфейсе «статус компьютера». Каждый терминал имеет пометку об исправности. Если терминал по какой-либо причине недоступен, то его можно пометить как «сломан» до указанного числа, тем самым исключить его номер из списка назначаемых номеров при записи на сеансы.

Инструктор может управлять включением и выключением терминалов без предварительной записи и ограничений по времени. Для этого в интерфейсе есть ссылка «управление компьютерами». Загружать терминалы можно под одним из заранее созданных профилей на сервере, например, есть профили «гостевой» и «олимпиадный». Для загрузки терминала по предварительной записи используется профиль «тестовый». Эти профили представляют собой конфигурацию пользовательского окружения в операционной системе, которое автоматически применяется при загрузке терминала.

Помимо этого в интерфейсе инструктора есть и другие специфические для учебных заведений инструменты:

- студенты без доступа – список пользователей с ролью «студент», для которых доступ к терминалам по предварительной записи заблокирован;
- просмотр замечаний – список сообщений, которые могут оставлять пользователи под учетной записью «студент»;
- отчет о посещаемости – генерирует отчеты по посещаемости класса по предварительной записи для каждого пользователя или группы пользователей с ролью «студент»;
- карточка студента – дает возможность просмотреть заявки всех пользователей на аренду терминалов, а также управлять ими.

Веб-интерфейс пользователей с ролью «администратор» расширяет функционал роли «инструктор» и добавляет следующие функции:

- редактирование лимитов – позволяет задавать лимиты времени работы за терминалом для групп пользователей с ролью «студент»;
- график работы – позволяет задавать режим работы класса, интервал времени и рабочие дни класса;
- настройка – задает основные параметры работы веб-интерфейса, такие как количество сеансов за день, продолжительность одного сеанса, число терминалов в классе, IP-адрес управляющего компьютера или сети и другие;
- просмотр статистики – генерация отчетов о посещаемости, о нагрузке на класс, о сеансах пользователей, о географии обращений, об успеваемости.

Если не углубляться в нюансы управления и настройки системы, то вся работа системы управления работой компьютерных классов заключается в следующем:

- пользователь с ролью «студент» через веб-интерфейс, например и из интернета, арендует для себя компьютер в классе на определенные время и дату;
- придя в указанные им день и час в класс, он отмечается у дежурного, который работает с веб-интерфейсом под ролью «инструктор»;
- пользователь проходит за указанный дежурным компьютер;
- компьютер автоматически загружается, работает отведенное на аренду время и автоматически выключается;
- после окончания работы пользователь еще раз отмечается у дежурного.

Схема системы управления

Вся система управления работой компьютерных классов делится на три основных блока: веб-сервер, база данных (БД) и система управления терминалами [6].

Под управлением веб-сервера Apache с модулем PHP работает веб-интерфейс системы. Веб-интерфейс занимается обновлением записей в БД, так как практически все параметры системы управления хранятся в ней.

База данных находится под управлением системы управления базой данных (СУБД) Oracle. С базой данных работает как веб-интерфейс, так и система управления терминалами.

Система управления терминалами отвечает за централизованное управление доступом к терминальным сессиям. После записи пользователя на сеанс, к нему прикрепляется номер компьютера. Когда пользователь приходит в класс, он должен отметить на регистрационном компьютере и пройти в класс за выделенный ему компьютер. Пока пользователь идет к своему рабочему месту, на его терминале автоматически загружает сеанс. При загрузке сеанса инициализируется все необходимое рабочее окружение. Сеанс автоматически завершается при окончании времени аренды компьютера. Также сеанс завершается, когда на регистрационном компьютере делается пометка о выходе.

Система управления терминалами работает на терминальных серверах, к которым подключены терминалы компьютерного класса. Терминальные серверы работают под управлением ОС Solaris. На рис. 2 изображена схема взаимодействия компонентов системы управления терминалами.

Система управления терминалами работает следующим образом. После запуска операционной системы стартует командный сценарий (скрипт) dtlogon, который собирает данные о номерах терминалов и их дисплеях (DISPLAY), записывает их в файл dtinfo.dat, циклически запускает скрипт dtautologin, используя в качестве параметра один из дисплеев, и запускает java-приложение de_shell.jar, после чего скрипт заканчивает свою работу. Скрипт dtautologin запускает X Window System на указанном дисплее и отображает фоновую заставку на терминале, используя указанный дисплей. Скрипт dtautologin остается постоянно загруженным в памяти и позволяет восстанавливать работу X Window System в случае её завершения. Приложение de_shell.jar постоянно за-

гружено и опрашивает базу данных с заданным в файле config.cfg интервалом времени, формирует список «номер терминала – учетная запись», записывает его в файл logusers.dat и запускает скрипт terminit.sh. Скрипт terminit.sh на основе данных из файлов dtinfo.dat и logusers.dat ищет на сервере сессии, которые должны быть закрыты, а потом запускает скрипт dlcsession на тех дисплеях терминалов, которые должны быть загружены под определенной учетной записью.

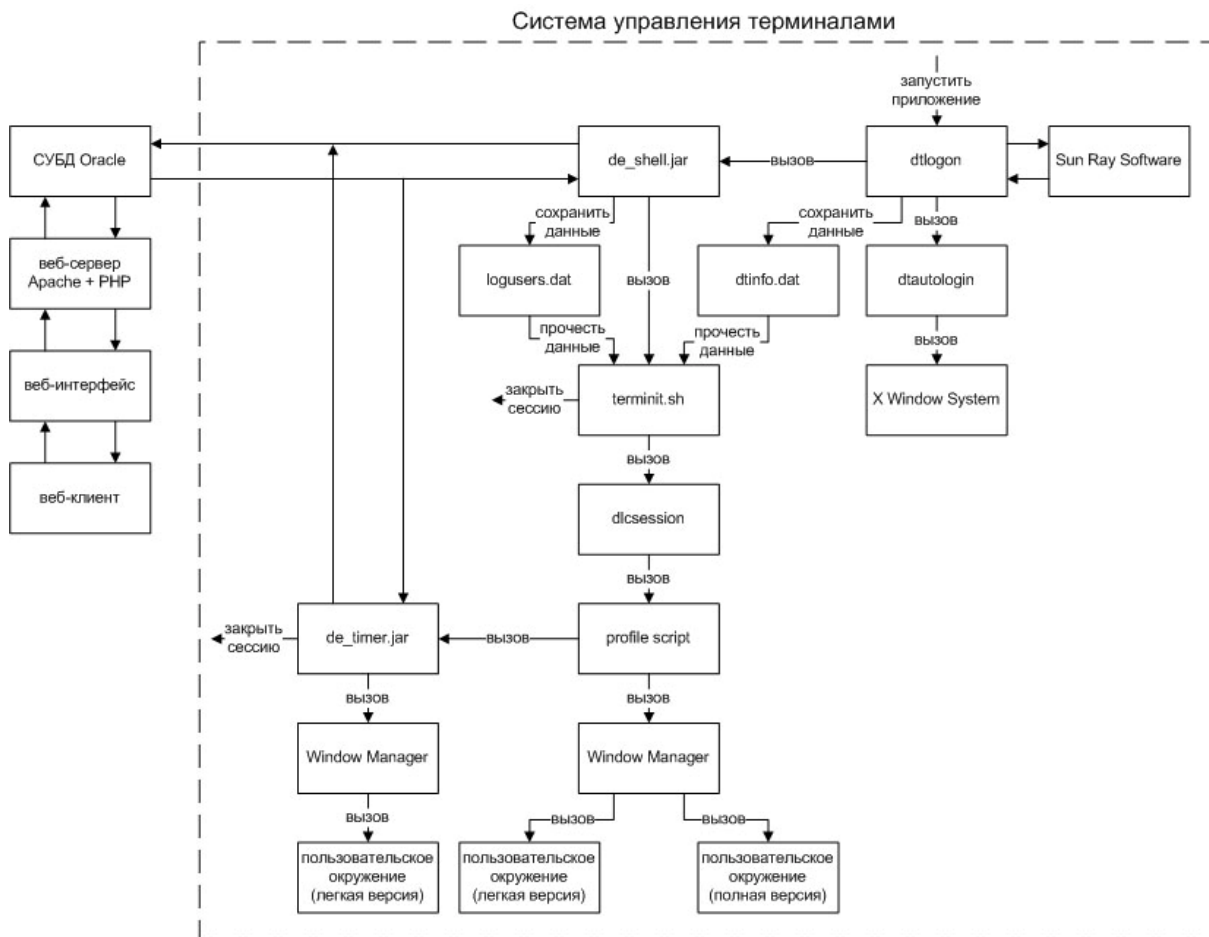


Рис. 2. Схема взаимодействия компонентов системы управления терминалами

Скрипт dlcsession запускает «profile script» из загружаемой учетной записи и в зависимости от типа учетной записи запускает либо java-приложение de_timer.jar (для аттестаций), либо Window Manager и пользовательское окружение полной или сокращенной версии. Java-приложение de_timer.jar запускает Window Manager и пользовательское окружение сокращенной версии и работает, пока есть время, отведенное для работы с этим терминалом. Когда время кончается, приложение закрывает сессию текущего пользователя.

Администрирование системы

Система работает под управлением ОС Solaris 10, а для работы требуется пакеты Sun Java SE 6 и Sun Ray Software 3. Для настройки веб-интерфейса достаточно настроить работу веб-сервера Apache, подключить к нему модуль PHP и разместить файлы веб-интерфейса на сервере, указав в файле config.php параметры подключения к базе данных Oracle. В базе данных необходимо создать схему и загрузить в нее дамп, к которой будет подключаться веб-интерфейс и система управления терминалами. Специ-

ально созданный пакет вспомогательных программ позволяет управлять работой как терминальных серверов и терминалов, так и всей системы в целом.

Заключение

Рассмотренное в статье решение позволяет автоматизировать процесс управления компьютерным классом, построенным по технологии «тонкого клиента». За основу взята система Sun Ray Hot Desk, позволяющая перенести всю нагрузку с клиентских рабочих станций на сторону сервера и сократить расходы на администрирование каждой рабочей станции. Система управления компьютерным классом автоматизирует процесс распределения ограниченного количества компьютеров между большим количеством пользователей, позволяя пользователям самим выбирать удобное для них время работы с компьютером.

Литература

1. Васильев В.Н., Лямин А.В., Чежин М.С. Система дистанционного обучения второго поколения // Научно-технический вестник СПбГУ ИТМО. – 2007. – Выпуск 45. – С. 148–157.
2. Васильев В.Н., Лямин А.В., Чежин М.С. Средства организации самостоятельной работы студентов в сетевой информационно-образовательной среде AcademicNT // Материалы межвузовской научно-методической конференции "Проблемы разработки учебно-методического обеспечения перехода на двухуровневую систему в инженерном образовании" / МИСиС. – М., 2008. – С. 224–232.
3. Васильев В.Н., Лисицына Л.С., Лямин А.В. Технология проведения ЕГЭ по информатике в компьютерной форме // Научно-технический вестник СПбГУ ИТМО. – 2007. – Выпуск 45. – С. 126–143.
4. Васильев В.Н., Лисицына Л.С., Лямин А.В. Сетевая технология проведения вступительных испытаний по информатике в режиме on-line // Сборник научных трудов "Использование информационно-коммуникационных технологий в процессе оценки качества образования". – СПб., 2008. – С. 55–70.
5. Душкин А.В. Распознавание и оценка угроз несанкционированного воздействия на защищенные информационно-телекоммуникационные системы // Информационные технологии. – 2008. – Выпуск 139. – С. 71–75.
6. Коханенко И.К. Правила распределения моделей по узлам вычислительной сети // Информационные технологии. – 2008. – Выпуск 138. – С. 36–41.

МЕТОДЫ УВЕЛИЧЕНИЯ ЭФФЕКТИВНОСТИ ИСПОЛЬЗОВАНИЯ ОРБИТАЛЬНО-ЧАСТОТНОГО РЕСУРСА ГЕОСТАЦИОНАРНОЙ ОРБИТЫ ДЛЯ СИСТЕМ СПУТНИКОВОЙ СВЯЗИ И ВЕЩАНИЯ

М.В. Прокопьев

(Московский физико-технический институт (государственный университет))

Научный руководитель – д.т.н., профессор Л.Я. Кантор

(Научно-исследовательский институт радио)

В работе рассматриваются основные проблемы эффективного использования орбитально-частотного ресурса для геостационарных спутниковых систем связи и вещания. Проблема обоснованно привлекает внимание специалистов, поскольку на сегодняшний день существуют серьезные затруднения в размещении на орбите новых спутниковых систем. Проводится оценка емкости рассматриваемого ресурса, и предлагаются возможные методы повышения эффективности использования радиочастотного спектра для спутников, размещенных на геостационарной орбите.

Ключевые слова: геостационарная орбита, орбитально-частотный ресурс, спутник

Введение

Число позиций для спутников на геостационарной орбите и количество полос частот относятся к ограниченным ресурсам, которыми располагает человечество, и в соответствии с международными соглашениями является общечеловеческой собственностью. Потребности мирового сообщества в передаче все больших объемов информации посредством радиосвязи очевидны, поэтому проблема радиочастотного обеспечения геостационарных спутниковых систем становится одной из наиболее острых на всех этапах создания и эксплуатации.

В свою очередь уникальность геостационарной орбиты состоит в том, что размещенный на ней космический аппарат будет вращаться с угловой скоростью, равной угловой скорости вращения Земли вокруг собственной оси. Другими словами, спутник, размещенный на геостационарной орбите, будет неподвижен относительно любой точки земной поверхности (сигнал со спутника может быть принят на стационарную антенну, применение следящих антенн не требуется).

Очевидно, что на геостационарной орбите можно разместить лишь ограниченное количество спутниковых систем, поскольку при нахождении друг от друга на излишне близком расстоянии их радиоэлектронные системы будут испытывать взаимные помехи. Следовательно, для того чтобы увеличить емкость геостационарной орбиты, необходимо минимизировать угловой разнос между спутниками.

Ниже приводятся методы, с помощью которых можно попытаться разрешить эту задачу, а также некоторые количественные оценки.

Предполагаемые методы

▪ Увеличение пространственной избирательности антенны

Избирательностью антенны называют способность антенны выделять полезный сигнал, ослабляя при этом действие мешающих сигналов (помех). Основана она на использовании тех или иных различий в характеристиках полезных и мешающих сигналов: направлении прихода, поляризации, амплитуды, частоты и фазы. Именно пространственная избирательность определяет необходимый угловой разнос между соседними спутниками на орбите. Проведя несложные вычисления, несложно заметить, что отношение амплитуды главного лепестка диаграммы направленности антенны к амплитуде огибающей боковых лепестков, которое определяет коэффициент пространствен-

ной избирательности, растет с ростом диаметра антенны и угол разноса двух соседних спутников соответственно может быть уменьшен.

- **Использования узких лучей на антеннах спутника**

Угловая ширина луча антенны обратно пропорциональна ее диаметру, следовательно, чем больше диаметр антенны, тем уже ее луч. Увеличение диаметра приводит к увеличению стоимости антенны и уменьшению зоны обслуживания. Это что касается нежелательных последствий. Что касается преимуществ, выигрыш, во-первых, в уменьшении углового разноса спутников, во-вторых, в уменьшении мощности передатчика, в-третьих, в возможности многократного использования полосы частот.

- **Применение методов приема/передачи радиосигналов с высокой помехоустойчивостью**

При передаче дискретных сигналов применяются методы помехоустойчивого кодирования, когда благодаря введению некоторой избыточности на передаче и соответствующего метода приема появляется возможность обнаружить и исправить ошибки, т.е. выиграть в помехоустойчивости, а также несмотря на необходимость передачи дополнительных символов, повысить пропускную способность системы связи при некоторой спектральной плотности шумов и заданной достоверности передачи. Конкретный анализ показывает, что необходимо создание кодов, приближающих пропускную способность линии связи к теоретическому пределу. Такой метод позволяет применить более низкие требования к отношению сигнал/помеха. Упущением является небольшое уменьшение скорости передачи информации. На сегодняшний день метод адаптированного кодирования и модуляции сигнала, который используется в стандарте DVB-S2, позволяет в зависимости от условий трансляции сигнала динамически изменять способ кодирования и формат модуляции, создавая при этом необходимое отношение сигнал/шум у абонента.

- **Разделение сигналов по поляризации**

Поляризация – явление нарушения симметрии распределения возмущений (векторов электрического и магнитного полей) в поперечной волне относительно направления ее распространения. Разделение сигналов по поляризации достигается при излучении двух сигналов с ортогональной поляризацией, либо линейной в двух взаимно перпендикулярных плоскостях, либо круговой с противоположными направлениями вращения. Во всех случаях при идеальном разделении может быть достигнуто удвоение емкости геостационарной орбиты. Дополнительным фактором, влияющим на разделение ортогонально поляризованных сигналов, является поворот плоскости поляризации на трассе распространения – явление, имеющее место в атмосфере.

- **Увеличение нормы допустимых помех**

При увеличении нормы допустимых взаимных помех пропускная способность каждой системы падает, поскольку приходится уменьшать собственные (тепловые и нелинейные) шумы системы, но при этом спутники могут располагаться ближе друг к другу, из-за чего общая пропускная способность геостационарной орбиты возрастает. Оптимальное значение доли помех зависит от конкретных параметров спутниковой системы. Для реальных параметров обычно оптимальное значение близко к единице, т.е. на помехи следует выделить почти всю норму шумов в линии передачи.

- **Использование однородности систем**

Две системы считают однородными, если помеха системы 1 от системы 2 равна некоторому допустимому значению при некотором угловом разносе, и при этом же угловом разносе помеха системы 2 от системы 1 также равна тому же допустимому значению. В простейшем случае критерием однородности является максимальная схожесть технических параметров двух систем. В частности, если две системы однородны, то и чувствительность к принимаемым мешающим сигналам у них одинаковая. Отсюда

можно сделать вывод, что при использовании максимально однородных систем емкость орбиты будет тоже максимальна.

- **Исключение расширения зоны покрытия за зону обслуживания**

Зона обслуживания – географическая зона, в которой гарантируются необходимые энергетические соотношения, достаточные для обеспечения устойчивой связи с абонентами. В системах с космическими аппаратами на геостационарной орбите зона покрытия (и соответственно зона обслуживания) незначительно изменяется со временем по мере движения спутника по орбите в силу неидеальности земной поверхности, ее размеры определяются характеристиками антенной системы спутника. Соответственно и изменяется зона обслуживания. Задача же состоит в устремлении отношения площади зоны обслуживания к площади зоны покрытия к единице. Это позволит улучшить электро-магнитную совместимость двух спутниковых систем и высвободить дополнительный орбитально-частотный ресурс.

- **Освоение высокочастотных диапазонов (более 30 ГГц)**

Если рассматривать общую емкость геостационарной орбиты, то очевидным, теоретически неограниченным способом увеличения емкости геостационарной орбиты является освоение новых диапазонов частот выше 30 ГГц и методов корректной работы на таких частотах. Основной проблемой, возникающей при передаче высокочастотного радиосигнала, является его значительное затухание в атмосферных осадках. Потери энергии радиоволн в таких условиях увеличиваются по мере увеличения ее частоты. Две основные причины, из-за которых происходит рассеяние – это поглощение газами (водяными парами и кислородом) и поглощение в воде. Бороться с этой проблемой пытаются по-разному: можно воспользоваться уже упоминавшимся выше методом адаптивного кодирования и модуляции сигнала (DVB-S2), который весьма эффективен в сочетании со способом сжатия телевизионного сигнала MPEG-4, а можно воспользоваться методом адаптивного динамического перераспределения мощности сигнала, предложенным японцами (о нем более подробно можно узнать на сайте Международного Союза Электросвязи <http://www.itu.int>).

- **Введение постоянной платы за закрепленный орбитально-частотный ресурс**

Такая мера должна будет вынудить структуры, за которыми сейчас закреплены неиспользуемые частоты, отказаться от них. Большинство пользователей относятся к категории спецпотребителей, в том числе спецпотребителем является Министерство Обороны РФ. При этом многие из владельцев часто неэффективно используют выделенный ресурс, потому что он для них бесплатный. По этой причине крупные телекоммуникационные операторы пока не могут полноценно развернуть свои высокотехнологичные проекты (например, 3G-связь, Wi-MAX). В частности, спектр частот, необходимый для полноценной эффективной работы 3G-связи, используется космической группировкой, различными охранными структурами, а также Администрацией Президента. На ум приходит идея конверсии частот, используемых военным сектором, а освободившиеся частотные диапазоны распределить между телекоммуникационными операторами.

- **Ограничение числа «бумажных» спутниковых систем**

Тенденция заявления странами – членами МСЭ орбитальных позиций с целью не развертывания собственных спутниковых систем, а превращения их в предмет купли-продажи наметилась с середины 1980-х гг. Наиболее яркий пример проблемы «бумажных спутников» – заявление в 1992 г. Королевством Тонга 31 геостационарной позиции, причем использование этих позиций, как выяснилось позднее, вообще не планировалось. Впоследствии Королевство Тонга уменьшило количество заявленных позиций до шести. Из них две позиции были проданы с аукциона, одна сдана в аренду американской компании, две были заняты приобретенными королевством спутниками, од-

на осталась неиспользуемой. В настоящее время в Международном Союзе Электросвязи идет интенсивный процесс выработки международно-правовых механизмов, не допускающих заявления позиций на ОГС для «бумажных спутников» и искоренения такого рода «коммерции».

Проведенная оценка

В Международном Союзе Электросвязи существует рабочая группа, занимающаяся эффективным использованием геостационарной орбиты и радиочастотного спектра. Результаты ее исследований публикуются на сайте <http://www.itu.int> в виде вкладов. В одном из недавних вкладов предлагается при проектировании новых антенн использовать обновленную формулу для огибающей функции не менее 90% боковых лепестков диаграммы направленности:

$$G = 29 - 25 \log \varphi \text{ dBi.}$$

(G –ослабление относительно изотропной антенны, φ – угол отклонения от оси основного лепестка диаграммы направленности). До недавнего времени при проектировании антенн использовали другую формулу:

$$G = 32 - 25 \log \varphi \text{ dBi.}$$

В приведенной ниже таблице посчитан выигрыш, измеряемый в количестве спутниковых систем, который следует ожидать в случае принятия решения об изменении эталонной формулы, описывающей огибающую функцию боковых лепестков диаграммы направленности антенны.

	C/I = 20 дБ	C/I = 25 дБ	C/I = 30 дБ	
45	29	18		D = 1.2 м
68	43	27		D = 2.0 м
87	55	35		D = 2.7 м

(Рабочая частота $F = 11$ ГГц, варьируется минимальное отношение сигнал/помеха, диаметр антенны)

Результаты сходятся с физическими предположениями: с одной стороны, чем больше диаметр антенны, тем уже ширина луча антенны, тем лучше становится пространственная избирательность антенны, соответственно тем меньше помех она создает окружающим антеннам; с другой стороны, увеличивая минимальное отношение сигнал/помеха, мы уменьшаем допустимую помеху, задаем более сильные условия для совместимости спутников, соответственно выигрываем в плотности размещения спутниковых систем на геостационарной орбите, что и демонстрируют полученные результаты.

Заключение

Спутниковая связь и вещание – это единственный на сегодняшний день коммерчески выгодный вид космической деятельности. Основой данного направления являются геостационарные спутники. При этом следует особо отметить проблему эффективного использования орбитально-частотного ресурса геостационарной орбиты. На сегодняшний день ей занимаются десятки ученых, исследователей, специалистов. Процесс международной координации спутниковых сетей в последние годы стал весьма затруднительным, требующим значительных временных и материальных ресурсов (как пра-

вило, координация растягивается на несколько лет) и далеко не всегда приводящим к положительным результатам. Поэтому необходимо приложить большие усилия и как можно быстрее максимально эффективно решить эту проблему, открыв тем самым дополнительные возможности использования космической связи.

Литература

1. Кантор Л.Я., Тимофеев В.В. Спутниковая связь и проблема геостационарной орбиты. – М.: Радио и связь. – 1988.
2. Бартенев В.В., Болотов Г.В., Быков В.Л. и др. Справочник по спутниковой связи. – М.: Радио и связь. – 1997.

РОЛЬ ФЕДЕРАЛЬНЫХ УНИВЕРСИТЕТОВ В РАЗВИТИИ СФЕРЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Н.А. Новоселов

(Южный федеральный университет)

Научный руководитель – ст. преподаватель М.Б. Файн

(Южный федеральный университет)

Данная статья поднимает проблемы отсутствия специалистов в области информационных технологий, а также недостатки недавно образованных Федеральных Университетов (ФУ). Дается способ взаимовыгодного решения: общие ИТ факультеты. Создание таких факультетов с единой материальной и информационной базой, направленных на комплексное обучение сотрудников и студентов, позволит насытить сферу квалифицированными кадрами и трудоустроить 100% обучающихся. Метод реализуется кредитно-модульной системой в пределах одного дня в неделю стажировки на ИТ факультете.

Ключевые слова: Информационные технологии, федеральные университеты, образование, работа по специальности, социальная роль, специалист, квалификация, инновации, мотивация

1. Введение. Постановка проблемы

1.1. Сфера информационных технологий

В последнее время наблюдается дефицит специалистов различного уровня ИТ сферы: квалифицированных, нишевых со смежными знаниями и студентов с соответствующими базовыми компетенциями без опыта, желающих развиваться в данной сфере. По прогнозам к 2012 году общее количество выпускников составит от 2.7 до 6.5 раз меньше годовой емкости данной сферы в целом [1]. Даже с учетом перехода специалистов со смежных областей, возвращением из-за границы в связи с кризисом рынок труда сферы ИТ будет далек от насыщения.

В соответствии с разработанными моделями сценариев развития ИТ рынка до 2012 года (рисунок) можно проследить динамику роста объемов рынка ИТ (млн. долл.). Согласно сценарию 1 произойдет сохранение главенства добывающих отраслей, рынок будет развиваться по законам сырьевой экономики. При сценарии 2 осуществится движение к экономике, основанной на знаниях, повышение конкурентоспособности экономики и, соответственно, рост аутсорсинга и в целом сектора услуг.

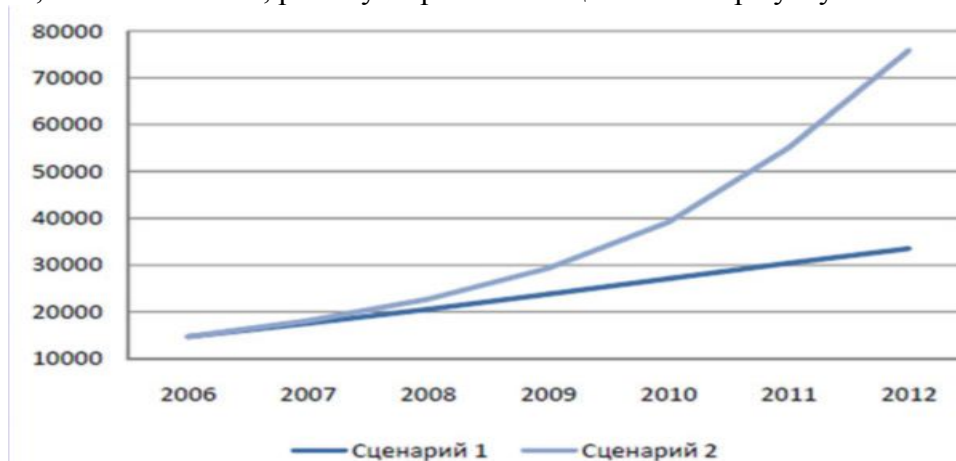


Рисунок. «Моделирование сценариев развития ИТ рынка»

Сопоставление полученных в ходе исследования результатов с данными проведенного по заказу АП КИТ исследования «Анализ контингента студентов инженерно-

коммуникационно-технологических специальностей в высших и средних специальных учебных заведениях Российской Федерации в 2006 году» по приему в учебные заведения свидетельствует о существенной недостаточности бюджетных учебных мест по ИТ специальностям. В 2006–2007 году суммарный выпуск ссузов и вузов по ИТ специальностям был на уровне 69 тыс. при потребности 188 тыс. – т.е. составлял всего около 40%. В 2012 году прогнозируется усугубление существующей ситуации: выпуск 85 тыс. человек обеспечит рынок на 35 или 16 процентов соответственно [1]. В дополнение, существует серьезная проблема несоответствия большинства программ обучения международным стандартам и текущим потребностям локального и глобального бизнеса [2]. Изучение мировых данных по занятости в ИТ сфере показало значительное отставание (в среднем втрое) по доле ИТ специалистов среди трудоспособного населения по сравнению с развитыми странами.

Последний аспект имеет примеры успешного решения: организация новых направлений в ведущих университетах страны, создание объединений, стандартизирующих курсы обучения, развитие научной деятельностью грантами и конференциями, создание специализированных кафедр и факультетов совместно с бизнес структурами [3]. Однако государственная поддержка слаба и не систематична: положительные изменения носят локальный характер. Происходит насыщение конкретных компаний специалистами, ведутся единичные разработки и тестирования собственного программного обеспечения. В глобальном смысле происходит коррекция существующих программ обучения, меняются тенденции в области развития ИТ сферы. Существенного изменения трендов рынка или определения общего направления развития не происходит. На текущий момент разработаны методы исправления ситуации, более того, вышеупомянутый союз «АП КИТ» активно участвует в продвижении образовательных проектов и развитии сферы в целом. Просчитаны основные варианты развития, определен потенциал рынка труда, конкретные методики для разрешения ситуации. Один из основных способов – переквалификация или дополнительное образование для людей, желающих работать в данной сфере [2, 4, 5].

1.2. Федеральные университеты

В настоящий момент на территории России существуют два федеральных университета – Южный Федеральный и Сибирский Федеральный университеты. В них одновременно проходит обучение более 100 тысяч студентов. Количество абитуриентов – более 10 тысяч каждый год [2]. Данные показатели не являются предельными для университетов. Также федеральные университеты имеют обширное финансирование, суммарно составляющее 6 млрд. рублей в год, в то время как Московский государственный университет имени Ломоносова получает 0.5 млрд. рублей [2,6]. При этом федеральные университеты были созданы путем присоединения к крупным государственным университетам классического типа, что сказалось на функционировании. Значительно пострадал процесс обучения: увеличилось количество нагрузки на преподавателей (создание методических указаний, изучение новых возможностей интерактивных материалов); потерян важный элемент корпоративного духа классических университетов - близость преподавательского состава и студентов, что наилучшим образом стимулировало научную деятельность; усложнена структура университета, что повлияло на повышение уровня бюрократии [7, 8].

До процесса объединения показатели точного соответствия специальности обучения и будущей профессии были низки: 5–7% на некоторых направлениях Архитектурной академии, 10–15% для гуманитарных и точных специальностей. В целом по стране коэффициент соответствия составляет менее 50% [9, 10]. Соответственно, в последствии данный показатель будет падать за счет снижения качества образования и мотивации к обучению.

2. Решение. Создание факультета Общего IT образования

2.1. Использование знаний каждого студента для блага общества

Решение выше обозначенных проблем возможно путем создания общеуниверситетского факультета, специализированного на обучении информационно-коммуникационным технологиям по всем направлениям деятельности федерального университета. То есть студент, магистр, научный сотрудник или выпускник имеет возможность дополнительного изучения IT дисциплин, связанных с его основной специализацией, углубления существующих навыков, их актуализации или частичной переквалификации в IT сфере. Факультет предполагает обязательное прохождение основных дисциплин, направленных на общее повышение компьютерной грамотности, и предполагает наличие таких курсов для всех участников образовательного процесса Федерального университета.

Также предполагается на основе факультета производить подготовку и переподготовку не относящегося к Федеральному университету населения: выполнение образовательных заказов Администрации и Правительства и преподавание курсов на коммерческой основе для сотрудников компаний.

Факультет Общего IT образования – отдельный факультет, не выпускающий самостоятельно специалистов, функционирующий в рамках всего федерального университета, обеспечивающий обучение всех студентов и сотрудников информационным технологиям. Формы обучения: дополнительная квалификация, углубление существующей в зависимости от специальности основного факультета, прохождение отдельных курсов с получением сертификатов.

2.2. Рассмотрение частного случая

В качестве примера может быть рассмотрено любое направление. Например, биология/медицина и биологический факультет. При этом основными должностными позициями, занимаемыми выпускниками данного факультета, будут являться:

- 1) научные сотрудники по обеспечению функционирования приборов и современного программного обеспечения;
- 2) операторы на медицинское оборудование;
- 3) сотрудники, занимающиеся разработкой инновационного научного технического и программного оборудования;
- 4) сотрудники, занимающиеся разработкой инновационного медицинского технического и программного оборудования;
- 5) аналитики, способные работать с базами данных;
- 6) преподаватели, со знаниями современных методик работы с информационно-показательными системами и их программным обеспечением;
- 7) административный персонал: санэпидемстанция, эксперты со знанием современных методов работы, технологий и программного обеспечения.

Современная наука в целом обязывает специалиста любой области быть образованными в IT сфере. Повышение эффективности работы научных сотрудников и качества преподавания и управления напрямую зависит от степени знания информационно-коммуникационных технологий. В свете текущего развития мирового общества можно сказать, что указанные умения имеют критическую важность для развития человечества.

На указанные выше должностные позиции идет меньшая часть выпускников университетов: такова особенность российского классического образования. В западных системах образования характерна подготовка конкретных специалистов, где формирование набора знаний определяется самим обучаемым или тьютором. Раньше, система советского образования включала в себя институт распределения по специальности по-

сле окончания учебного заведения. Благодаря этому, студент всегда знал, что работой он будет обеспечен, надо лишь учиться. Сейчас в России система такова, что университеты также готовят специалистов-теоретиков для дальнейшего распределения, которого уже нет. В результате, меньшая часть проходит отбор, и начинают работать по основной специальности. Отсеянные кандидаты не имеют возможности применить свои знания и находят непрофильную работу.

2.3. Реализация социальных обязательств университета на 100 процентов

Суть реализации социальных обязательств такова, чтобы направить естественный отсев в сферу ИТ. В пределах федерального университета это тысячи качественно подготовленных людей, которые могут одновременно получать современные знания о технологиях. В итоге, университет насыщает научную и административную отрасль, информационный рынок специалистами и профессионалами, выполняя свой социальный долг почти на 100%. В контексте примера, получаем и менеджеров по продаже медицинского оборудования, и программистов медицинского обеспечения. Оба направления нишевые, требуют обязательного медицинского или биологического образования. По аналитике агентства «Head Hunter», самой прибыльной и востребованной областью продаж является именно продажа данного оборудования. Средний заработок специалиста продаж при условии профильного образования и опыта работы 4 года равняется 70 – 200 тыс. рублей [11].

Выше описанные данные относятся ко всем факультетам Федерального университета кроме тех, которые имеют внутренние специальности и включают в себя использование ИТ, например, математическое моделирование в экономике. Также существуют факультеты, занимающиеся полностью данным направлением. Для них также возможно обучение на Общем факультете с целью расширения и углубления ИТ кругозора, фокусирования развития на определенных направлениях.

ИТ сфера включает в себя большое количество основных внутренних направлений, среди которых: руководители, программисты, ИТ-менеджеры, веб-программисты, системные администраторы, дизайнеры, программисты C++, администраторы 1С, тестировщики программного обеспечения, 3D дизайнеры, сборщики персональных компьютеров, продакт-менеджеры, программисты Java, верстальщики, контент-менеджеры, аниматоры, программисты баз данных, системные аналитики, специалисты по интернет-рекламе, разнообразные специалисты продаж и работы с программным обеспечением в узких областях. Рынок труда обладает большой емкостью, следовательно, внутри него вероятны переходы между смежными специализациями. Например, перекавалификация из дизайнера (кафедра промышленного дизайна) в аниматора 3D моделей и сцен (в рамках ЮФО только платные курсы, нет профильного обучения) или переход из программиста Java в контент-менеджера. Сферы, которые не обеспечиваются образованием, емки, количество их велико. Они насыщаются за счет самоучек и сотрудников, которых переучивает компания за свои деньги [2]. Такое положение не дает России выход на мировые рынки в связи с невысоким уровнем конечного продукта. Открытый факультет позволит студентам быть мобильными в выборе сферы интересов и фокусирования обучения, государству – разрешить острые социальные вопросы. В целом, обеспечится формирование мотивации и активной жизненной позиции граждан, что приведет к росту профессионализма и появлению реальной конкуренции в самой емкой и важной сфере развития России.

2.4. Практическое применение

Практическая реализация заключается в выделении дня для посещения факультета и обязательным количеством часов, которые студент сам распределяет, исходя из своих интересов и планов. Наряду с этим существует необходимый набор занятий, обя-

зательный для изучения. То есть один день в неделю все студенты проводят на другом факультете, находящемся в отдельном здании, необходимое количество времени, определенное нормами нагрузки. За отведенное время они проходят набор курсов, состоящие из нескольких обязательных и нескольких выбранными ими в начале семестра. Здесь возможна реализация кредитной программы, заключающейся в доступности выбора определенных курсов при наличии требуемого уровня успеваемости и знания базовых предметов.

Данный метод позволяет экономить на постройке и использовании новых корпусов, приобретении нового оборудования и программного обеспечения: необходимо обеспечить оборудованием и преподавательским составом один корпус без необходимости студентам или сотрудникам за время учебы покидать его. Также обеспечивается смешение групп обучающихся внутри университета, что благотворно сказывается на социализации студентов. Также система создания Общего факультета будет содействовать активному привлечению крупных спонсоров, таких как Microsoft, Abby, Autodesk, заинтересованных обучением широкого круга продвинутых пользователей своему программному обеспечению, что позволит сделать обучение на факультете актуальным реальным потребностям общества и снизить расходы на программное обеспечение и разработки методических пособий.

2.5. Перспективы развития системы с использованием общего IT факультета

1. Данная система после введения позволяет осуществить переход к университетам с обучением модульного типа: каждый курс идет в форме блока и программа обучения каждого студента формируются в индивидуальном порядке. Переход к западной унифицированной системе будет происходить мягко и с учетом особенностей развития отечественной системы образования [12, 8].

2. В рамках данной системы реализуется одно из перспективных направлений Минобрнауки: превращение обучения по системе бакалавров в основное для обучения специалистов и переход на лимитированную по времени магистратуру, как источник научных кадров России. Четкое деление обеспечится свободой выбора и высоким уровнем мотивации студентов [12].

3. Глобализация и концентрация системы образования: возрастает роль перспективного направления – федеральных университетов. Абитуриенты и компании будут равно заинтересованы в дипломах университетов мирового уровня, позволяющих работать в реальном секторе сразу после выпуска. Роль второстепенных учебных заведений широкого профиля будет ослабевать, что позволит уменьшить их количество [12, 7, 13].

2.6. Особенности развития системы с использованием общего IT факультета

Рынок информационных технологий:

- 1) станет более прозрачным, гибким для всех участников;
- 2) спровоцирует взаимодействие реального сектора бизнеса и науки с поставщиком специалистов – университетом, что обеспечит дополнительное финансирование науки;
- 3) позволит России в целом приблизиться к ведущим странам по развитию и культуре сферы информационных технологий;
- 4) спровоцирует всплеск интереса к обучению в федеральных университетах, как к способу реализации амбиций абитуриентов и нахождения интересной работы;
- 5) уменьшит количество безработных и низкооплачиваемых граждан, что благоприятно скажется на уровне жизни и стабильности состояния в стране.

3. Заключение

На основании вышерассмотренных аспектов, можно сделать вывод, что следующей задачей является разработка методики интеграции Общего ИТ факультета в существующие федеральные университеты. Основной задачей является конечное создание факультета и поддержание его актуальности.

Организация отдельного Общего факультета ИТ в Федеральном университете позволит решить широкий круг проблем:

- 1) насыщение сферы ИТ и развитие до европейского уровня, как по количественному, так и по качественному показателю;
- 2) выполнение инновационными Федеральными университетами социальной роли, как посредника в сфере высоких и информационных технологий между работодателем и работником;
- 3) решение проблемы низкого уровня мотивации современных студентов, рост самосознания, постоянная социализация и уменьшение влияния социальных стереотипов;
- 4) подготовка к переходу на кредитно-модульную систему обучения.

Литература

1. ИТ кадры в российской экономике. – Центр социологических и психологических технологий в образовании МИЭМ. – М. 2007. – 44 с.
2. Сухомлин В. Создание системы ИТ образования — это совместный проект бизнеса и образования. – ж. "ИТ-Спец" от 08.2007.
3. Официальный сайт МФТИ (ГУ) [Электронный ресурс]. Каталог кафедры.– Электрон, дан. – Список с экр. – <http://www.mipt.ru/facultets/>
4. Приказ: РФ N 4175 новое направление подготовки бакалавров и магистров 511900 «Информационные технологии». – 29.11.2002.
5. ИТ – Образование в России, сайт посвященный вопросам развития ИТ образования в России. – Электрон, дан. – www.it-edu.ru
6. Кузьмин Владимир. В России появится сеть федеральных университетов. – «Российская Газета».– 24.04.2008.
7. Аналитический обзор [Электронный ресурс]: В России появится сеть федеральных университетов. Интервью с В. Захаревичем. – 2007. – Электрон, дан. – Список с экр. – <http://www.taganrog.su/node/3375>
8. Стенографический отчет о встрече с преподавателями и студентами Сибирского и Южного федеральных университетов [Электронный ресурс]. – Красноярск. – 13.11.2007. – Электрон, дан. – с экр. – <http://www.kremlin.ru/text/appears/2007/11/151258.shtml>
9. Вахрушева Дарья. Дипломы на ветер. – Газета «Труд». – № 036 за 03.03.2009.
10. Обзор: Александр Старовойтов. В омских училищах до сих пор готовят невостребованных специалистов. – "Российская газета" Омск. – №4512 8.10.2007. – Электрон, дан. – <http://www.rg.ru/2007/11/08/reg-irtysh/ptu.html>
11. Тенденции рынка медицины и фармацевтики [Электронный ресурс]. – «Планеты HR». – Электрон, дан. – Список с экр. – <http://planetahr.ru/publication/2426/91>
12. Новости. Информационно-образовательный портал Фаэтон [Электронный ресурс]. – Phaethon 2005–2008. – Электрон, дан. – С экр. – <http://faito.ru/>
13. Анатолий Баранов. Аналитический анализ: Получатся ли элитные университеты в российской провинции? Анатолий Баранов, гл. редактор ФОРУМа.мск. – 22.12.2006. – Электрон, дан. – С экр. –<http://forum.msk.ru/material/region/17750.html>

ТОЧНОСТЬ ПАРАМЕТРОВ ДЕМОДУЛЯТОРА В СИСТЕМАХ С МОДУЛЯЦИЕЙ КАМ

А.Ю. Янушковский

(Санкт-Петербургский государственный университет кино и телевидения)

Научный руководитель – д.т.н., профессор А.В. Кривошейкин

(Санкт-Петербургский государственный университет кино и телевидения)

В статье предлагается метод нахождения допусков на параметры демодулятора в цифровых системах связи, применяющих модуляцию КАМ (например, система цифрового кабельного телевидения DVB C). Дан вывод выражения, связывающего отклонение параметра демодулятора и вероятность ошибки. Приведён пример расчёта допуска на параметр модуляции 64 КАМ в системе DVB C.

Ключевые слова: отклонение параметров, номинальные значения, отклонение вероятности ошибки

Введение

Основные результаты классической теории помехоустойчивости [1] получены в предположении того, что причиной появления ошибок в канале связи является присутствие в нём аддитивных или мультипликативных помех. Между тем существуют и другие причины, в частности, наличие дестабилизирующих факторов в процессе производства и эксплуатации приёмопередающей аппаратуры. Они приводят к отклонению параметров аппаратуры от номинальных значений и, как следствие, к отклонению вероятности ошибки от значения, полученного в теории помехоустойчивости [1]. В настоящее время широкое применение нашли системы многоуровневой модуляции от 32 КАМ (квадратурная амплитудно-фазовая модуляция) до 1024 КАМ (квадратурная амплитудно-фазовая модуляция), чувствительные к этим факторам. Поэтому возникает необходимость установить зависимость отклонения вероятности ошибки от отклонения параметров аппаратуры и найти допуск на параметры аппаратуры при заданном допуске на отклонение вероятности ошибки. Решение этих задач применительно к демодулятору системы КАМ рассматривается в предлагаемой статье.

Основные расчётные соотношения

Вероятность ошибочного приёма сигнала с m КАМ модуляцией находится по формуле:

$$P_{ош} = \sum_{i=1}^n \left[p(s_i) \sum_{\substack{j=1 \\ j \neq i}}^m p(s_j / s_i) \right], \quad (1)$$

где $p(s_i)$ – вероятность передачи i -го сигнала s_i , $p(s_j / s_i)$ – условная вероятность ошибочного приёма сигнала s_j при условии, что был передан сигнал s_i , m – число уровней сигнала, то есть число возможных сигналов в поле сигналов КАМ. Здесь под вероятностью ошибочного приёма сигнала понимается вероятность ошибочного приёма символа, содержащего $k = \frac{\ln m}{\ln 2}$ бит.

Считая, что все сигналы равновероятны, т.е. $p(s_i) = 1/m$, запишем (1) в виде:

$$P_{ouu} = \frac{1}{m} \sum_{i=1}^m \left[\sum_{\substack{j=1 \\ j \neq i}}^m p(s_i / s_j) \right]. \quad (2)$$

Следуя [1], примем, что наибольший вклад в вероятность ошибки вносит ошибочный приём четырёх соседних сигналов, ближайших к сигналу s_i . Поэтому во внутренней сумме в (2) пренебрежём всеми составляющими за исключением четырёх ближайших:

$$P_{ouu} = \frac{1}{m} \sum_{i=1}^m [p(s_{i+1} / s_i) + p(s_{i+2} / s_i) + p(s_{i+3} / s_i) + p(s_{i+4} / s_i)]. \quad (3)$$

Степень обоснованность пренебрежения будет рассмотрена в Приложении.

Вероятность ошибочного приёма одного сигнала при передаче другого зависит от расстояния между сигналами [2]. В системе КАМ расстояния между соседними сигналами одинаковые. Поэтому справедливы соотношения:

$$\begin{aligned} p\left(\frac{s_{i+1}}{s_i}\right) &= p\left(\frac{s_2}{s_1}\right) & p\left(\frac{s_{i+2}}{s_i}\right) &= p\left(\frac{s_3}{s_1}\right) & p\left(\frac{s_{i+3}}{s_i}\right) &= p\left(\frac{s_4}{s_1}\right) \\ p\left(\frac{s_{i+4}}{s_i}\right) &= p\left(\frac{s_5}{s_1}\right) & i &= 1 \dots m \end{aligned} \quad (4).$$

Следовательно, без потери общности можно принять, что передавался сигнал s_1 . Подставляя (4) в (3) получим выражение:

$$P_{ouu} = p(s_2 / s_1) + p(s_3 / s_1) + p(s_4 / s_1) + p(s_5 / s_1). \quad (5)$$

На рисунке представлен фрагмент поля сигналов в окрестности передаваемого сигнала s_1 .

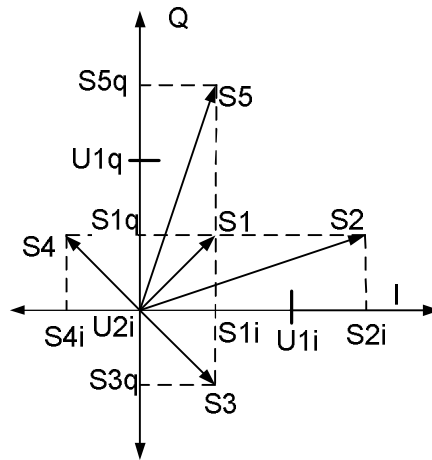


Рисунок. Фрагмент поля сигналов

Формирование КАМ сигнала на передаче и демодуляция сигнала на приёме производится по квадратурным каналам I и Q. Выразим (5) через квадратурные составляющие сигналов:

$$\begin{aligned} p\left(\frac{s_2}{s_1}\right) &= p\left(\frac{s_{2i}}{s_{1i}}\right) + p\left(\frac{s_{2q}}{s_{1q}}\right) & p\left(\frac{s_3}{s_1}\right) &= p\left(\frac{s_{3i}}{s_{1i}}\right) + p\left(\frac{s_{3q}}{s_{1q}}\right) \\ p\left(\frac{s_4}{s_1}\right) &= p\left(\frac{s_{4i}}{s_{1i}}\right) + p\left(\frac{s_{4q}}{s_{1q}}\right) & p\left(\frac{s_5}{s_1}\right) &= p\left(\frac{s_{5i}}{s_{1i}}\right) + p\left(\frac{s_{5q}}{s_{1q}}\right) \end{aligned}$$

При передаче сигнала s_1 и приёме его в канале I в соответствии с рисунком может быть принято правильное решение о передаче сигнала s_1 и возможны только два оши-

бочных решения – передан сигнал s_2 , либо передан сигнал s_4 . Поэтому вероятности ошибочных решений о передаче сигнала s_3 , либо сигнала s_5 равны нулю, то есть $p\left(\frac{s_{5i}}{s_{1i}}\right) = p\left(\frac{s_{3i}}{s_{1i}}\right) = 0$. Рассуждая подобным образом о приёме сигнала в канале Q, за-

пишем аналогичные равенства $p\left(\frac{s_{2q}}{s_{1q}}\right) = p\left(\frac{s_{4q}}{s_{1q}}\right) = 0$.

Подставим полученные соотношения в (5):

$$P_{ou} = p\left(\frac{s_{2i}}{s_{1i}}\right) + p\left(\frac{s_{4i}}{s_{1i}}\right) + p\left(\frac{s_{3q}}{s_{1q}}\right) + p\left(\frac{s_{5q}}{s_{1q}}\right). \quad (6)$$

Границами между сигналами в каналах, в соответствии с рисунком, являются следующие пороговые уровни U_{1i} , U_{2i} , U_{1q} , U_{2q} [3]:

$$U_{1i} = \frac{S_{2i} + S_{1i}}{2}, \quad U_{2i} = \frac{S_{4i} + S_{1i}}{2}, \quad U_{1q} = \frac{S_{5q} + S_{1q}}{2}, \quad U_{2q} = \frac{S_{3q} + S_{1q}}{2}. \quad (8)$$

При отклонении пороговых уровней от их номинальных значений (8) возникает отклонение вероятности ошибки от теоретического значения.

В теории допусков [1] при небольших отклонениях принято использовать линейную часть ряда Тейлора. Поэтому связь между отклонением вероятности ошибки и отклонениями пороговых уровней определяется следующей формулой:

$$d(\ln P_{ou}) = S_{U_{1i}}^{P_{ou}} d(\ln U_{1i}) + S_{U_{2i}}^{P_{ou}} d(\ln U_{2i}) + S_{U_{1q}}^{P_{ou}} d(\ln U_{1q}) + S_{U_{2q}}^{P_{ou}} d(\ln U_{2q}), \quad (9)$$

где $S_U^{P_{ou}} = \frac{U}{P_{ou}} \frac{dP_{ou}}{dU}$ – чувствительность вероятности ошибки к изменению порогового уровня [1].

Выведем формулы для чувствительностей, для этого прежде всего установим выражение для вероятности ошибки P_{ou} . Обозначим сигналы, появляющиеся в каналах I и Q, соответственно I_s и Q_s . Из-за наличия шумов в сигнале s_1 каждый из этих сигналов является случайным с математическим ожиданием, соответственно, S_{1i} и S_{1q} , имеет дисперсию σ^2 и распределён по нормальному закону.

Рассмотрим первоначально чувствительность вероятности ошибки к отклонению порогового уровня U_{1i} .

При нормальном законе распределения шумов вероятность ошибочного приёма сигнала S_{2i} при переданном сигнале S_{1i} , есть вероятность того, что сигнал I_s превосходит пороговое значение U_{1i} , то есть:

$$p\left(\frac{s_{2i}}{s_{1i}}\right) = p(I_s > U_{1i}) = -\frac{1}{\sqrt{2\pi}} \int_{\infty}^{(S_{1i}-U_{1i})/\sigma} e^{-\frac{t^2}{2}} dt. \quad (10)$$

Перейдём в (10) к производной:

$$\frac{d p(S_{2i}/S_{1i})}{dU_{1i}} = \frac{1}{\sqrt{2\pi}} \frac{1}{\sigma} \exp\left(-\frac{(S_{1i}-U_{1i})^2}{(2\sigma)^2}\right). \quad (11)$$

Используя (11) и определение чувствительности, получим формулы:

$$S_{U_{1i}}^{p(S_{2i}/S_{1i})} = \frac{U_{1i} * \exp\left(-\frac{h^2}{2}\right)}{\sigma \int_{\infty}^h \exp\left(-\frac{t^2}{2}\right) dt}, \quad (12)$$

$$h^2 = \frac{(S_{1i} - U_{1i})^2}{(\sigma)^2}. \quad (13)$$

Обозначим $d = s_{1i} - U_{1i}$ – расстояние между сигналом и его пороговым уровнем в канале I. Расстояние между соседними сигналами в этом же канале равно $2d$.

Поэтому параметр h^2 пропорционален отношению мощности сигнала разности соседних сигналов в канале I к мощности шума. Эта величина в реальных системах передачи намного больше единицы. Упростим (12), устремив h к бесконечности. Для упрощения возникающей при этом неопределённости вида $\frac{0}{0}$ используем правило Лопиталя.

Окончательное выражение имеет вид:

$$S_{U_{1i}}^{p(S_{2i}/S_{1i})} = -\frac{U_{1i} * h}{\sigma}.$$

Так как в (6) только первое слагаемое зависит от порогового уровня U_{1i} , то формула чувствительности вероятности ошибки к отклонению параметра U_{1i} имеет вид:

$$S_{U_{1i}}^{P_{oш}} = \frac{p(s_{2i}/s_{1i})}{P_{oш}} S_{U_{1i}}^{p(S_{2i}/S_{1i})}$$

В силу эквидистантности точек поля сигналов КАМ все слагаемые в (6) равны друг другу и формула для чувствительности записывается в виде:

$$S_{U_{1i}}^{P_{oш}} = -\frac{1}{4} \frac{U_{1i} * h}{\sigma}. \quad (14)$$

Повторяя проведенные преобразования для пороговых уровней U_{2i} , U_{1q} , U_{2q} , получим следующие соотношения:

$$S_{U_{2i}}^{P_{oш}} = \frac{1}{4} \frac{U_{2i} * h}{\sigma}, \quad S_{U_{1q}}^{P_{oш}} = -\frac{1}{4} \frac{U_{1q} * h}{\sigma}, \quad S_{U_{2q}}^{P_{oш}} = \frac{1}{4} \frac{U_{2q} * h}{\sigma}. \quad (15)$$

В каждом из квадратурных каналов I и Q системы КАМ с m уровнями может появиться один из n равновероятных сигналов ($n = \sqrt{m}$). Мощность в квадратурном канале P , полученная усреднением мощностей по всем равновероятным сигналам, находится по формуле [3]:

$$P = \frac{n^2 - 1}{3} d^2. \quad (16)$$

Прибавляя к разностному сигналу d среднее значение напряжения \sqrt{P} в квадратурном канале, получим среднее значение порогового уровня:

$$U_{nop} = d \left(1 + \sqrt{\frac{n^2 - 1}{3}} \right).$$

С учётом (13), заменим в (15) значения пороговых уровней их средним значением:

$$S_{U_{1i}}^{P_{oш}} = -S_{U_{2i}}^{P_{oш}} = S_{U_{1q}}^{P_{oш}} = -S_{U_{2q}}^{P_{oш}},$$

$$S_{U_{1i}}^{P_{oш}} = -\frac{d * \left(1 + \sqrt{\frac{n^2 - 1}{3}} \right) * h}{4 * \sigma} = -\frac{\left(1 + \sqrt{\frac{n^2 - 1}{3}} \right) * h^2}{4}. \quad (16a)$$

Выразим параметр h через параметр $S/N = P/\sigma$ – отношение мощности сигнала к мощности шума. В соответствии с (13): $h^2 = d^2/\sigma^2$.

Из (16) следует, что $d = \sqrt{\frac{3P}{n^2 - 1}}$. Таким образом выражение для параметра h_2 имеет вид:

$$h^2 = \frac{3P}{(n^2 - 1) \cdot \sigma^2} = \frac{3 \cdot S / N}{n^2 - 1}. \quad (17)$$

Все чувствительности в (16а) равны по абсолютной величине и отличаются только по знаку. В дальнейшем нам понадобится абсолютная величина чувствительности S_{abs} , выражение для которой с учётом (17) имеет вид:

$$S_{abs} = \frac{3 \cdot (1 + \sqrt{\frac{m-1}{3}}) \cdot S / N}{4 \cdot (m-1)}, \quad (18)$$

где $m = n_2$ – число уровней сигнала в системе КАМ.

Расчёт допусков

Так как пороговые уровни в квадратурных каналах I и Q не зависят друг от друга, то и их отклонения в (9) являются независимыми случайными величинами. С учётом этого обстоятельства, применяя выражение (18), перейдём в (9) к дисперсиям:

$$\sigma^2(d(\ln P_{ou})) = (S_{abs})^2 \cdot [\sigma^2(d(\ln U_{1i})) + \sigma^2(d(\ln U_{2i})) + \sigma^2(d(\ln U_{1q})) + \sigma^2(d(\ln U_{2q}))]. \quad (19)$$

Для нормального закона распределения с вероятностью 0,997 выражение (19) по правилу «3σ» записывается в виде:

$$\Delta(\ln P_{ou}) = 3 \cdot S_{abs} \cdot \sqrt{\sigma^2(d(\ln U_{1i})) + \sigma^2(d(\ln U_{2i})) + \sigma^2(d(\ln U_{1q})) + \sigma^2(d(\ln U_{2q}))},$$

где $\Delta(\ln P_{ou})$ – граница изменения, то есть допуск на случайную величину отклонения вероятности ошибки от своего номинального значения. Применяя правило «3σ» к отклонениям пороговых уровней, получим следующее выражение для допусков:

$$\Delta(\ln P_{ou}) = S_{abs} \cdot \sqrt{\Delta^2(\ln U_{1i}) + \Delta^2(\ln U_{2i}) + \Delta^2(\ln U_{1q}) + \Delta^2(\ln U_{2q})}, \quad (20)$$

где символ Δ , относящийся к параметру, означает допуск на величину этого параметра.

Как следует из (16) вероятность ошибки равночувствительна к отклонениям пороговых уровней от своих номинальных значений. Поэтому примем допуски на все пороговые уровни равными друг другу, т.е.

$$\Delta(\ln U_{1i}) = \Delta(\ln U_{2i}) = \Delta(\ln U_{1q}) = \Delta(\ln U_{2q}) = \Delta(\ln U).$$

Подставим это условие в (20) и получим следующую формулу для расчёта допусков:

$$\Delta(\ln U) = \frac{\Delta(\ln P_{ou})}{2 \cdot S_{abs}}. \quad (21)$$

Пример расчёта

Рассчитаем допуск на точность установки пороговых уровней в системе 64 КАМ, используемой в системе цифрового кабельного телевидения DVB-C. Здесь $m=64$, $k=6$, $n=8$. В соответствии с [2] на выходе демодулятора необходимо обеспечить вероятность ошибочного приёма бита в сигнале не более, чем 10^{-4} . Для обеспечения этого требования с учётом неточной установки пороговых уровней зададим в качестве номинального значения вероятность ошибочного приёма бита равным 10^{-5} . Таким образом, установим допуск на отклонение вероятности $\Delta(\ln P_{ou}) = \ln(10^{-4}) - \ln(10^{-5}) = 2,3$. По кривой помехоустойчивости для системы 64 КАМ [2] находим, что для вероятности ошибоч-

ного приёма бита 10^{-5} отношение S/N равно 17,5 dB (56,2). Так как число бит в символе равно 6, то отношение сигнал/шум для символа в 6 раз больше и равно $S/N = 337,2$. Подставим это значение, а также значение $m = 64$ в (18), получим величину чувствительности:

$$Sabs = \frac{3 * (1 + \sqrt{\frac{m-1}{3}}) * S / N}{4 * (m-1)} = 22,4 .$$

Подставим полученное значение чувствительности в (21) и найдём допуск на отклонение порогового уровня:

$$\Delta(\ln U) = \frac{\Delta(\ln Pou)}{2 * Sabs} = 0,051 = 5,1 \% .$$

Литература

1. Гехер К. Теория чувствительности и допусков электронных цепей. – М.: Сов. радио. – 1973. – 200 с.
2. ETSI TR 101 290 V1.2.1 (2001-05) Digital Video Broadcasting (DVB); Measurement guidelines for DVB systems.
3. Боккер П. Передача данных. Техника связи в системах телеобработки данных. – М.: Связь. – 1980. – 264 с.

РАЗРАБОТКА СИСТЕМЫ ВИЗУАЛЬНОГО УПРАВЛЕНИЯ КОМПЬЮТЕРОМ

С.Е. Антонов

Научный руководитель – к.т.н. В.В. Повышев

Человек получает большую часть информации о мире с помощью глаз. Современные представления об управлении компьютерной техникой можно довольно сильно изменить, если научить компьютер видеть и понимать человека, окружающие предметы, события. Целью данного исследования является разработка базиса библиотеки обработки и распознавания видео-данных, которую в дальнейшем можно будет применить для создания полноценной системы визуального управления компьютером.

Ключевые слова: программирование, распознавание объектов, преобразование видео

Введение

Существуют различные методы управления компьютерной техникой. Их можно разделить на несколько типов:

Управление с помощью вынесенных манипуляторов — наиболее старый и наиболее широко используемый метод управления. В его основе лежит использование дополнительных устройств, физическое воздействие на которые интерпретируется как управляющее воздействие. При этом само устройство управления находится на расстоянии от устройства воспроизведения (монитора, проектора). К данному виду управления можно отнести использование клавиатур, мышек, джойстиков, пультов ДУ, тачпадов, и др. Использование данного метода управления не совсем естественно для человека, так как воздействие на один объект приводит к изменению другого, не связанного с ним объекта. Например, психологически, факт того что движение мышкой приведет к изменению координат курсора не явен, и требует тренировки от пользователя. Однако, при долгосрочном использовании манипуляторов данного типа происходит привыкание и события начинают казаться естественными и идеально спроектированными для использования человеком.

Управление с помощью манипуляторов визуально взаимодействующих непосредственно с устройством воспроизведения информации. Данный метод управления более понятен человеку, так как происходит непосредственный контакт манипулятора и объекта управления. К устройствам, в которых применяется данный метод, можно отнести так называемые «touch screen» дисплеи (в них применяется как управление с помощью стилусов, так и просто с помощью пальцев рук), системы управления с помощью лазерной указки. Метод ввода информации в системах подобного типа удобен людям даже далеким от компьютера, а использование нейро-сетевых программ распознавания почерка пользователя компьютера делает процесс ввода и оцифровки информации ещё более удобным.

Визуальное управление – предполагает управление компьютером через видео или веб-камеру. С камеры изображение поступает в программу, которая интерпретирует его в управляющие сигналы для компьютера. Таким образом, для управления компьютером может использоваться различный набор предметов и действий. Например, можно управлять компьютером с помощью жестов, использовать для управления подручные предметы, удовлетворяющие некоторым требованиям, программировать поведение компьютера на изменение освещенности, на появление или исчезновение движения. В результате, «общение» с компьютером может стать более удобным, понятным и приятным. Реализация же и внедрение таких идей как «виртуальные окна», «трекеры», идеи дополненной реальности могут в будущем довольно сильно изменить системы управления ряда программных средств. Несомненным плюсом данной технологии является

простота и низкая стоимость её внедрения на компьютеры пользователей, особенно если учесть, что большая часть ноутбуков уже сейчас комплектуется веб-камерами. Данный метод позволяет сделать менее четкой границу между виртуальным и реальным миром, дополнить реальность.

Системы виртуальной реальности – обычно реализуются с использованием специальных шлемов, очков, датчиков, которые крепятся на голову, руки, тело. Системы виртуальной реальности позволяют достичь эффекта полного погружения в виртуальный мир. При использовании специальных шаров и подобных устройств, в виртуальной реальности можно реально ходить, бегать. А если добавить к этому системы воспроизведения трехмерного изображения, путем создания собственного видео ряда для каждого глаза, то можно добиться очень реалистичного эффекта. Основным недостатком данных систем является высокая сложность и цена, малая распространенность.

Комбинированные системы – объединение нескольких типов управления, например, систем визуального управления и традиционных манипуляторов.

В рамках данной работы будут рассмотрены основные моменты создания системы визуального управления компьютером на примере разработки библиотеки для работы с видео-данными. Система визуального управления выбрана, как наиболее перспективная и наименее затратная система для исследований.

Постановка задачи

Требуется разработать систему, способную распознавать определенные объекты и интерпретировать их перемещение в управляющие сигналы. Для простоты рассмотрим систему, способную распознавать геометрические фигуры, которые при различных условиях освещенности могут находиться на разном расстоянии от камеры. В дальнейшем полученную программу можно доработать для распознавания более сложных объектов или движений. В качестве средства разработки использовалась IDE Codegear RAD Studio.

Видеозахват

Для видеозахвата был выбран драйвер «avicap» – avicap.lib [1]. Видеозахват происходит по следующей схеме:

- инициализация web-камеры – создание окна видеозахвата (capCreateCaptureWindowA(...)), его позиционирование, масштабирование, инициализация драйвера (WM_CAP_DRIVER_CONNECT);
- установка параметров видеозахвата — включает получение графического идентификатора окна, настройку параметров драйверов (выбор режима цветности, установка разрешения, проверка работоспособности установленных параметров — WM_CAP_GET_VIDEOFORMAT, WM_CAP_SET_VIDEOFORMAT);
- установка «callback»-функции. Данная функция вызывается каждый раз, когда на драйвер камеры приходит новая команда «считать следующий кадр». В этой функции осуществляются основные действия по обработке видео-изображения (WM_CAP_SET_CALLBACK_FRAME);
- установка таймеров (SetTimer(...)). Таймеры нужны для периодической отправки драйверу команды WM_CAP_GRAB_FRAME, по которой происходит считывание очередного фрейма с камеры и, соответственно, вызов «callback»-функции. Также они применяются для создания детектора движения, который будет использован для оптимизации использования процессорных ресурсов системой распознавания.

Преобразование видео

Для преобразования видео был создан специальный класс WebCam, в котором

описываются все основные действия над видео-потокком. Также дополнительно были созданы: класс `PictureArea`, описывающий работу с определенными областями изображения; структура `BitmapPoint`, позволяющая легко получать доступ к цветам каждого конкретного пикселя в изображении. В итоге, для первоначальной работы с кадрами видео использовался массив из элементов типа `BitmapPoint`. Все рассмотренные ниже методы преобразования изображения обычно производятся одновременно, чтобы уменьшить количество обходов картинки.

Объединение по маске

Первое что было сделано с изображением – это объединение его по определенной маске. Была выбрана квадратная маска с переменным размером. Смысл преобразования заключается в том, что все изображение «проходится» выбранной маской. Пиксели, попавшие под определенный шаг маски усредняются и записываются в изображение в виде одного пикселя. Пример работы данного преобразования приведен на рис. 1.

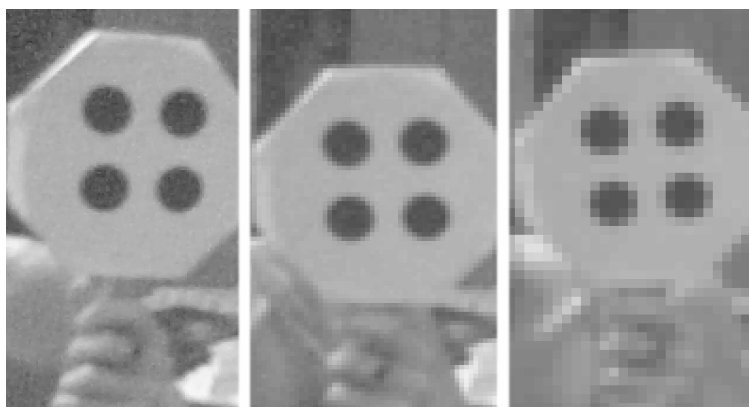


Рис. 1. Усреднение изображение по маске различных размеров

Основная цель данного преобразования – подавление шумов, уменьшение картинки, которая будет использоваться для дальнейших преобразований. Также, как будет рассмотрено далее, манипулируя размером маски можно распознавать объекты, находящиеся на разном расстоянии от веб-камеры.

Реализация данного преобразования.

- Выбирается размер маски.
- Картинка проходится с шагом в ширину маски, и на каждом шаге пиксели, попавшие под маску, усредняются и записываются в целевое изображение. В итоге мы получаем картинку меньшего размера и с уменьшенными шумами.

Подавление шумов

Для подавления шумов используется алгоритм, основанный на предположении о том, что нас интересуют только выпуклые объекты. Таким образом – наша задача сделать полученные области выпуклыми и убрать впадины. Реализуем поглощение большими областями меньших областей, на микроуровне мы таким образом уберем шумы, нежелательные линии, узкие области. Простейший алгоритм, который можно применить для такого шумоподавления: проходим по картинке и сливаем пиксели, которые окружены пикселями равными по цвету между собой, но отличными от исходного пикселя. Если добавить допущение по пикселям равное двум, увеличить количество проходов и применить этот алгоритм к областям, которые мы получим на следующих этапах, то можно добиться довольно неплохих результатов при работе в помещениях с плохой освещенностью.

Выделение областей движения

Выделение областей движения может существенно облегчить в дальнейшем задачу поиска объектов распознавания, так как таким образом мы можем отсечь значительную часть изображения, которое не меняется с течением времени [2]. Исходя из предположения о том, что мы распознаем динамические объекты, можно строить различные логические последовательности распознавания, используя информацию от датчика движения. Например, если мы нашли объект в определенном месте, а в следующие несколько моментов времени движения не было, или оно было незначительно, то можно не искать объект заново, а предположить, что он все ещё находится в том же самом месте. Или, если мы не смогли найти объект в некоторый момент времени, то если далее не было движения, то нам и не стоит искать его в это время.

Для реализации датчика движения был заведен таймер, который инициирует раз в пол секунды запись основного кадра, полученного с веб-камеры [3]. Далее, до следующего срабатывания таймера, все полученные кадры сравниваются с основным, с определенным допущением – чувствительностью. Чувствительность определяет максимальную допустимую степень отклонения цвета пикселей, при котором они считаются не изменившимися. Таким образом, составляется маска движения кадра, в которой 0 значит, что пиксель не изменился, 1 – что пиксель изменился. Далее, циклически, волновым методом в этой маске выделяются области движения, и производится фильтрация слишком маленьких областей движения. В результате мы получаем области кадра, в которых произошли существенные изменения.

Выделение областей схожих по цвету

Цель данного преобразования – объединение групп пикселей схожих по цвету, определение размера этих областей, их положения, цвета. Преобразование реализуется волновым методом. В ходе преобразования, при попадании указателя на крайнюю границу «волны», информация о новых точках, не включенных ни в какую область, попадает в созданный стек точек. Когда текущий обход области заканчивается, новый обход начинается с последней точки стека. Опустошение стека свидетельствует об окончании преобразования. На рис. 2 приведен пример применения данного преобразования с различным допустимым отклонением цвета соседних пикселей.

По ходу преобразования собирается статистика об областях, в которую включается площадь, положение крайних точек, цвет. Найденные области фильтруются, и отсекаются те, размер которых нас не интересует.

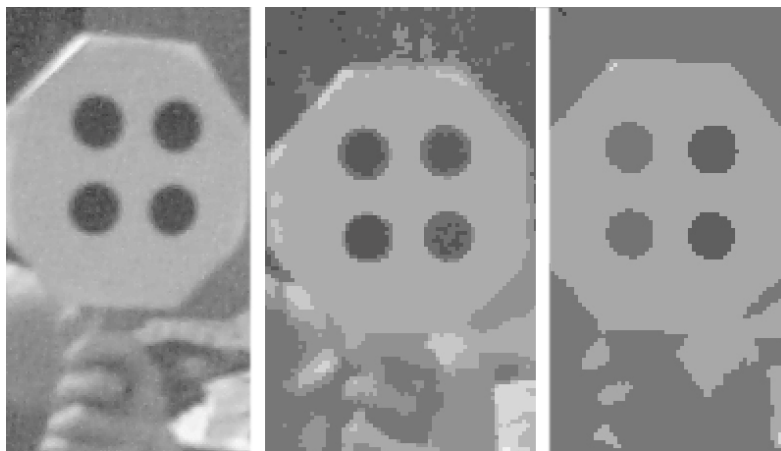


Рис. 2. Выделение областей схожих по цвету

Распознавание

Результатом процедуры распознавания является информация о наличии или отсутствии объекта в пределах видимости, информация о его координатах, ориентации в пространстве.

Анализ областей

Рассмотрим распознавание простых геометрических объектов, например круга или группы кругов, с использованием полученной информации об областях движения и областях схожих по цвету.

Зададим некоторый размер искомого круга. Произведем фильтрацию областей таким образом, чтобы выделить области, которые наиболее вероятно содержат этот круг. Для того, чтобы сделать распознавание менее жестким, введем коэффициент, определяющий точность распознавания. Он будет определять допустимую степень отклонения реальной площади области от площади искомого круга. Нас интересуют области, которые удовлетворяют нескольким критериям: во-первых, площадь области находится в допустимых пределах; во-вторых, геометрические размеры области не превышают размеры круга. Также выполняется проверка того, что минимальный размер (ширина или высота, в зависимости от ситуации) не меньше диаметра искомого круга. Таким образом, мы можем отфильтровать как слишком маленькие, так и слишком большие области.

Допустим, мы получили несколько областей, которые удовлетворяют нашим критериям. Для того, чтобы с большей вероятностью распознать необходимый объект, можно разделить его на несколько примитивных объектов, которые мы распознавать уже умеем [2]. Например, используем квадрат из четырех кругов. Далее табличным методом зададим полный взвешенный граф из полученных на этапе распознавания областей. В данном графе циклически удалим все вершины, которые находятся на недопустимом расстоянии от трех любых других вершин. Если в итоге остались вершины, которые могут образовать квадрат, то это скорее всего искомым объектом. Если в результате такого поиска не было найдено подходящих объектов (или их было найдено несколько), то следует, например, ограничить поиск только областями, где было зафиксировано движение, или попытаться поискать данный объект с различными размерами пиксельных масок, с различной чувствительностью распознавания. Пример подсветки результата работы системы распознавания приведен на рис. 3.



Рис. 3. Подсветка найденного объекта

В результате данных преобразований можно повысить качество распознавания объекта и сделать систему адаптивной к увеличению или уменьшению объекта, к изменению освещенности, увеличению шумов.

Заключение

В ходе выполнения работы были рассмотрены основные аспекты разработки базового модуля программного продукта для визуального управления компьютером. Был проведен обзор следующих этапов преобразования изображения: видеозахват, подавление шумов, объединение по маске, выделение областей движения, областей схожих по цвету. Принципы распознавания единичных и составных объектов, изученные на этапах выполнения работы, были успешно применены при разработке программного обеспечения для визуального управления компьютером. Полученное программное обеспечение, знания и навыки могут быть применены при создании «трекеров», «виртуальных окон», систем дополненной реальности.

Литература

1. Microsoft Developer Network [Электронный ресурс] / Официальный сайт подразделения компании Microsoft по взаимодействию с разработчиками – Режим доступа: <http://msdn.ru>, свободный. – Яз. рус., англ., нем.
2. Russian Software Developer Network [Электронный ресурс] / Сообщество русскоговорящих разработчиков программного обеспечения; ред. RSDN Magazine – Режим доступа: <http://rdsn.ru>, свободный. – Яз. рус., англ.
3. Delphi Kingdom [Электронный ресурс] / Виртуальный клуб программистов; ред. Александр Малыгин – Режим доступа: <http://www.delphikingdom.com/>, свободный. – Яз. рус.

РАСЧЕТ ПОМЕХОУСТОЙЧИВОСТИ УЧАСТКА ВОЛС С ОПТИЧЕСКИМИ УСИЛИТЕЛЯМИ

Д.Н. Багаветдинов

Научный руководитель – к.ф.-м.н., доцент С.А. Миронов

Статья посвящена анализу накопления шумов на участке волоконно-оптической линии связи, содержащем оптические усилители. Изложена методика оценки помехоустойчивости в спектральном канале DWDM сигнала, исходя из заданного коэффициента ошибок. Проведен расчет для участка ВОЛС протяженностью $L = 1000$ км, содержащего 9 оптических усилителей.

Ключевые слова: ВОЛС, оптические усилители, BER, помехоустойчивость, коэффициент шума

Введение

Для увеличения пропускной способности в современных ВОЛС применяется метод спектрального уплотнения каналов – технология DWDM. Поскольку в оптическом волокне происходит ослабление сигнала, через определенное расстояние требуется его восстановление. Для этой цели в системах с DWDM используются оптические усилители (ОУ). Наибольшее распространение получили эрбиевые волоконно-оптические усилители (EDFA).

Оптические усилители являются аналоговыми устройствами и, обеспечивая усиление мощности передаваемого оптического сигнала, накапливают искажения (шумы усилителя, хроматические и поляризационные дисперсионные искажения оптического кабеля и т.д.), возникающие на различных участках многопролетной ВОЛС содержащей несколько ОУ, что приводит к снижению отношения сигнал/шум на фотоприемнике в конце магистрали. Накопление шумов – это фундаментальное свойство аналоговых усилителей.

Поэтому, необходимо предусмотреть такой исходный запас помехоустойчивости для каждого спектрального канала, который позволил бы обеспечить требуемую помехозащищенность переданного информационного сигнала в конце ВОЛС.

Определение требуемого значения отношения сигнал-шум

Функционирование цифровых систем передачи оценивается коэффициентом битовых ошибок BER, определяемым числом ошибок на заданное количество переданных битов информации, т.е. вероятностью появления ошибок. Для расчета BER, достаточно знание одного параметра – Q-фактора, который определяется следующим выражением [1]:

$$Q = \frac{|\mu_1 - \mu_2|}{\sigma_1 + \sigma_2}, \quad (1)$$

где μ_0, μ_1 – уровни сигналов, соответствующие 0 и 1; σ_0, σ_1 – их среднеквадратичные отклонения.

Если задано требуемое значение BER, то, пользуясь выражением (2), можно найти требуемое значение Q-фактора.

$$BER(Q) = \frac{1}{2} \cdot \operatorname{erfc}\left(\frac{Q}{\sqrt{2}}\right). \quad (2)$$

Рассчитаем Q-фактор, исходя из типичного для современных ВОЛС коэффициента ошибки $BER = 10^{-10}$, подставляя его в выражение (2), получаем требуемое значение $Q = 6,36$.

Помимо вышеизложенных параметров качество оптического сигнала характеризуют величиной, которую принято называть оптическим отношением сигнал/шум (OSNR). OSNR равно отношению мощности полезного сигнала к мощности шума в спектральном интервале $\Delta\nu$ оптического канала. По мере распространения сигнала между регенераторами значение OSNR может только убывать.

Допустимый уровень OSNR [2], который обеспечивает заданное значение Q-фактора, определяется выражением (3).

$$OSNR = 4 \cdot Q^2. \quad (3)$$

Подставляя рассчитанное значение Q в выражение (3), определяем минимальный допустимый уровень оптического отношения сигнал-шум на выходе ВОЛС $OSNR = 161,87$ (или 22,09 дБ – в логарифмических единицах).

Расчет коэффициента усиления ОУ

Коэффициент усиления ОУ должен обеспечить компенсацию оптических потерь на участке усиления:

$$g = \alpha_{ov} + \alpha_{oc}, \quad (4)$$

где α_{ov} – потери в оптическом кабеле (ОК), дБ; α_{oc} – потери в соединениях, дБ.

Потери в оптическом кабеле определяются как:

$$\alpha_{ov} = \alpha \cdot \sum L_i,$$

где L_i – средняя строительная длина ОК, км; α – значение коэффициента затухания оптического волокна, дБ/км;

Потери в соединениях складываются из потерь на разъёмных и неразъёмных соединениях.

$$\alpha_{oc} = \alpha_{nc} \cdot N_{nc} + \alpha_{pc} \cdot N_{pc}, \quad (5)$$

где α_{nc} – потери на неразъёмных соединениях, $\alpha_{nc} = 0,05$ дБ; N_{nc} – количество неразъёмных соединений; α_{pc} – потери на разъёмных соединениях, $\alpha_{pc} = 0,25$ дБ; N_{pc} – количество разъёмных соединений.

Количество неразъёмных соединений зависит от строительной длины оптического кабеля и от общей протяженности линии:

$$N_{nc} = \frac{L}{L_i} - 1,$$

где N_{nc} – количество неразъёмных соединений; L – общая протяженность линии.

Рассчитаем коэффициент усиления ОУ, считая заданными следующие параметры: длина усилительного участка $L_y = 100$ км; волокно G.655 $\alpha = 0,21$ дБ; средняя строительная длина ОК $L_i = 5$ км; количество разъёмных соединений $N_{pc} = 2$ (на входе и выходе участка усиления).

Потери в кабеле:

$$\alpha_{ov} = \alpha \cdot L_y = 0,21 \cdot 100 = 21 \text{ дБ.}$$

Количество неразъёмных соединений:

$$N_{nc} = \frac{L_y}{L_i} - 1 = \frac{100}{5} - 1 = 19.$$

Тогда по формуле (5) $\alpha_{oc} = \alpha_{nc} \cdot N_{nc} + \alpha_{pc} \cdot N_{pc} = 0,05 \cdot 19 + 0,25 \cdot 2 = 1,45$ дБ.

В итоге коэффициент усиления ОУ в рассматриваемой ВОЛС $g = 22,45$ дБ.

Расчет отношения сигнал-шум на выходе РУ

Шумовые свойства оптического усилителя оцениваются с помощью коэффициента шума, показывающего, насколько ухудшилось отношение сигнал/шум на выходе усилителя по сравнению с величиной отношения сигнал/шум на входе [1].

Коэффициент шума (или шум-фактор) одиночного оптического усилителя NF равен:

$$NF = (P_{c.вх} / P_{ш.вх}) / (P_{c.вых} / P_{ш.вых}) = (P_{c.вх} \cdot P_{ш.вых}) / (P_{ш.вх} \cdot P_{c.вых}), \quad (6)$$

где $P_{c.вх}$ – мощность сигнала на входе ОУ; $P_{ш.вх}$ – мощность шума на входе ОУ;

$P_{c.вых}$ – мощность сигнала на выходе ОУ; $P_{ш.вых}$ – мощность шума на выходе ОУ.

Так как $P_{c.вх} / P_{c.вых} = 1/G$, где G – коэффициент усиления ОУ, то выражение (6) можно представить в виде:

$$NF = P_{ш.вых} / GP_{ш.вх}. \quad (7)$$

Рассмотрим, как изменяется коэффициент шума для многопролетной линии связи, содержащей ОУ. Примем для упрощения, что:

- все пролеты (усилительные участки) ВОЛС имеют одинаковые потери оптического сигнала A ($\Sigma\alpha = 10lg(A)$);
- каждый ОУ имеет коэффициент усиления G , причем $G = A$, т.е. усиление ОУ полностью компенсирует потери передаваемого сигнала в каждом пролете;
- система регулирования оптических усилителей обеспечивает режим стабилизации выходной мощности в каждом спектральном канале, величина которой одинакова для всех ОУ.

На рис. 1 показан фрагмент многопролетной ВОЛС с оптическими усилителями и диаграммы уровней для сигнала и шума.

Минимальная мощность шума, действующего на входе оптического усилителя в полосе передаваемого оптического сигнала, определяется квантовым шумом [3]:

$$P_{ш.вх} = h\nu\Delta\nu_0 = N_0, \quad (8)$$

где $h = 6,6252 \cdot 10^{-34}$ Дж·с – постоянная Планка; ν – оптическая частота; $\Delta\nu_0$ – спектральная полоса канала.

Шумы на выходе оптического усилителя складываются из усиленных квантовых шумов и шумов усиленного спонтанного усиления ASE (Amplified Spontaneous Emission) [4].

Мощность шумов усиленного спонтанного излучения ASE, действующих на выходе ОУ в полосе передаваемого информационного сигнала, обозначим $ASE_{\Delta F}$.

Тогда полная мощность шумов на выходе первого оптического усилителя ОУ-1 равна:

$$P_{ш.вых1} = ASE_{1\Delta F} + GN_0.$$

Коэффициент шума усилителя ОУ-1 определяется как:

$$NF_1 = P_{ш.вых1} / G \cdot P_{ш.вх} = (ASE_{1\Delta F} + GN_0) / G \cdot N_0 = 1 + ASE_{1\Delta F} / GN_0. \quad (9)$$

Мощность шумов, действующих на входе следующего оптического усилителя (ОУ-2), равна:

$$P_{ш.вх2} = ASE_{1\Delta F} / A + GN_0 / A = ASE_{1\Delta F} / A + N_0.$$

На выходе ОУ-2 мощность шумов составит:

$$P_{ш.вых2} = ASE_{1\Delta F} \cdot G / A + GN_0 + ASE_{2\Delta F}.$$

Так как уровень передачи одинаков для всех оптических усилителей в ВОЛС, то принимаем, что мощность шумов ASE, вносимых каждым ОУ, также одинакова, т.е. $ASE_{1\Delta F} = ASE_{2\Delta F} = ASE_{\Delta F}$, тогда:

$$P_{ш.вых2} = 2 \cdot ASE_{\Delta F} + GN_0.$$

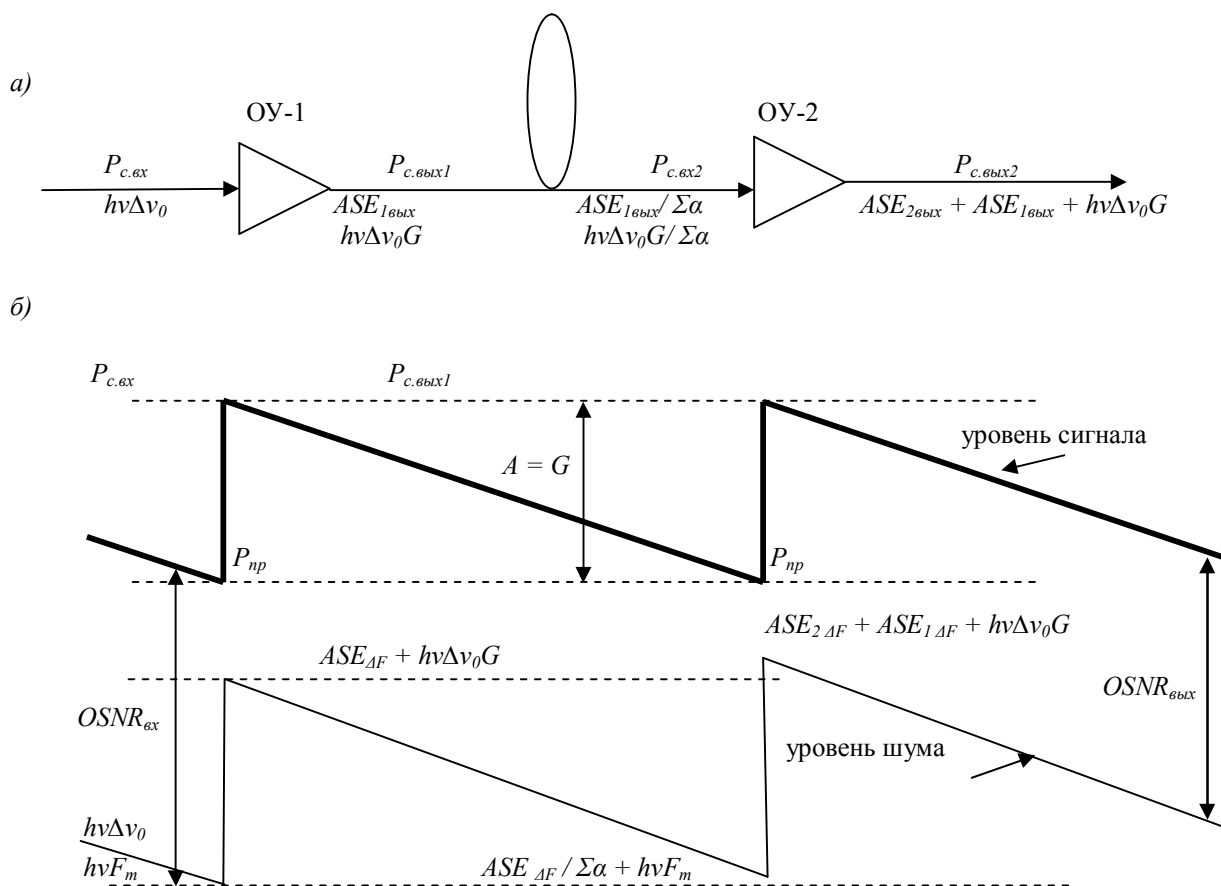


Рис. 1. Накопление шумов в многопролетной ВОЛС с оптическими усилителями
 а) – фрагмент линии с оптическими усилителями;
 б) – диаграмма уровней сигнала и шумов для фрагмента ВОЛС

На выходе линии, содержащей k последовательно включенных оптических усилителей, мощность шума будет равна:

$$P_{ш.выхk} = k \cdot ASE_{\Delta F} + GN_0.$$

Тогда коэффициент шума для такой ВОЛС $NF_{ВОЛС}$ определится как:

$$NF_{ВОЛС} = P_{ш.выхk} / GN_0 = 1 + k \cdot ASE_{\Delta F} / GN_0.$$

Учитывая соотношение (9), можно выразить величину коэффициента шума для ВОЛС через коэффициент шума одного оптического усилителя NF_1 .

$$\begin{aligned} NF_{ВОЛС} &= (1 + k \cdot ASE_{\Delta F} / GN_0) + (k - 1) \cdot ASE_{\Delta F} / GN_0 = NF_1 + (k - 1) \cdot (NF_1 - 1) = \\ &= NF_1 + k \cdot NF_1 + NF_1 - k + 1 = k \cdot NF_1 - (k - 1). \end{aligned}$$

Ухудшение отношения сигнал/шум на выходе ВОЛС (в децибелах) по отношению ко входному OSNR составит:

$$\Delta OSNR_{dB} = 10 \lg(k \cdot NF_1 - k + 1). \quad (10)$$

Мощность шумов ASE практически не зависит от мощности шумов, действующих на входе оптического усилителя, и каждый ОУ просто добавляет «квант» шумов ASE, который проходит на выход ВОЛС практически без изменений (усиление этих шумов в каждом ОУ только компенсирует затухание предшествующего участка линии).

В итоге на выходе ВОЛС отношение сигнал/шум будет равно:

$$OSNR_{выхdB} = p_{пер} - \Sigma \alpha - N_0 - 10 \lg(k \cdot NF_1 - k + 1). \quad (11)$$

Произведем расчет участка ВОЛС протяженностью $L = 1000$ км, содержащего 9 последовательно включенных оптических усилителей через каждые $L_y = 100$ км, для

одного оптического канала со скоростью передачи $V = 2,5$ Гбит/с (STM-16). При этом OSNR в конце линии уменьшится на:

$$\Delta OSNR = 10 \lg(k \cdot NF_1 - k + 1) = 10 \lg(9 \cdot 3,16 - 9 + 1) = 13,12 \text{ дБ}$$

($k = 9$ – число усилителей; $NF = 3,16$ раза (5 дБ) – типичное значение коэффициента шума одного ОУ типа EDFA).

При этом входное OSNR должно быть не менее, чем:

$$OSNR_{\text{вх}} = OSNR + \Delta OSNR = 22,09 + 13,12 = 35,21 \text{ дБ.}$$

Мощность шума, действующего на входе первого оптического усилителя в полосе передаваемого оптического сигнала, определяется квантовым шумом N_0 . Тогда для обеспечения полученной помехозащищенности уровень принимаемой оптической мощности определяется:

$$p_{np} = OSNR_{\text{вх}} - N_0 = OSNR_{\text{вх}} - 10 \lg(h\nu\Delta\nu_0 \cdot 10^3) = -29,74 \text{ дБм}$$

($\nu = 1,934 \cdot 10^{14}$ Гц оптическая частота несущей канала ($\lambda = 1550$ нм); $\Delta\nu_0 = 2,5 \cdot 10^9$ Гц спектральная полоса канала).

Таким образом, для одного канала:

$$p_{np} = -29,74 \text{ дБм.}$$

Уровень передачи будет равен:

$$p_{nep} = p_{np} + g = -29,74 + 22,45 = -7,29 \text{ дБм.}$$

На основании полученных результатов, построим диаграмму уровней участка ВОЛС протяженностью 1000 км, содержащего 9 оптических усилителей для одного оптического канала (рис. 2).

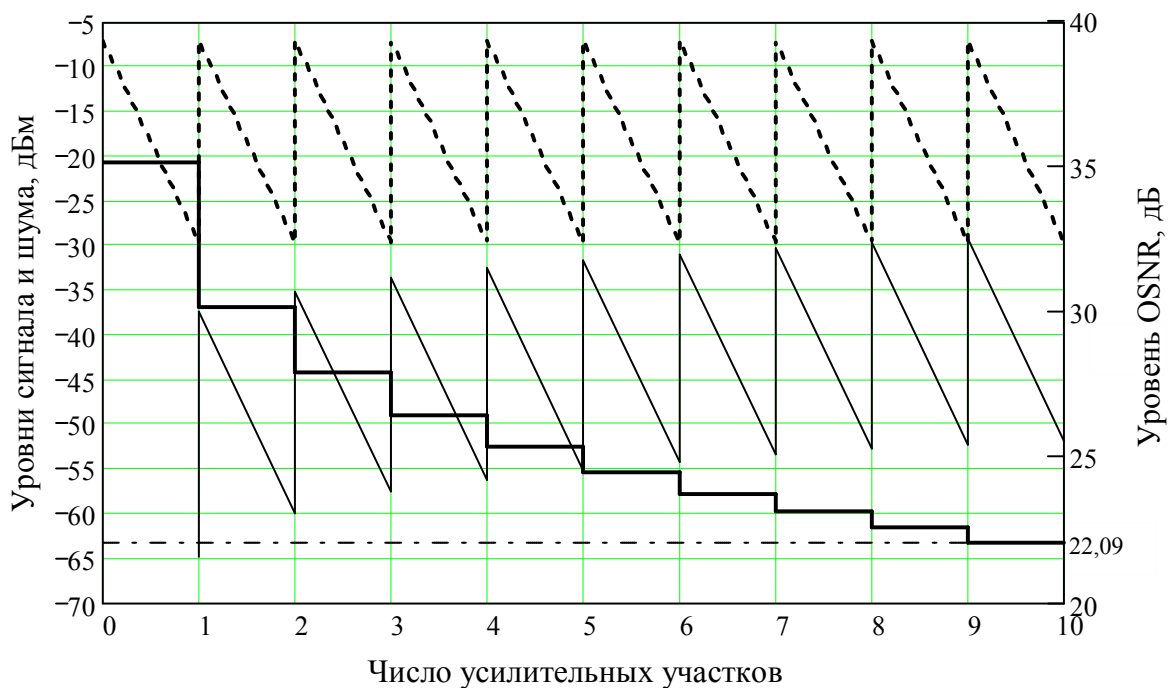


Рис. 2. Диаграмма уровней участка ВОЛС протяженностью 1000 км содержащего 9 ОУ

— уровень шума; уровень сигнала; — OSNR

Также на основании вышеизложенных расчетов при такой же мощности передачи оптических сигналов ($p_{nep} = -7,29$ дБм) определены предельные длины усилительных участков рассматриваемой ВОЛС, при различных значениях скоростей передачи синхронной цифровой иерархии и коэффициентов ошибок. Результаты представлены в таблице.

Стоит отметить, что в современных ОУ уровень выходной оптической мощности может достигать до плюс (20–27) дБ; это позволяет повысить оптическую мощность на входе усилителя, чтобы устранить воздействие других источников шумов и помех в ОУ, в оптическом кабеле и фотоприемном устройстве.

BER \ B	L _y , км			
	STM-1 155 Мбит/с	STM-4 622 Мбит/с	STM-16 2,5 Гбит/с	STM-64 10 Гбит/с
10 ⁻⁹	159,95	131,22	102,45	73,78
10 ⁻¹⁰	157,52	128,78	100	71,34
10 ⁻¹¹	155,34	126,6	97,83	69,17
10 ⁻¹²	153,36	124,62	95,85	67,18

Таблица. Предельные длины усилительных участков ВОЛС, содержащей 9 ОУ, при различных скоростях передачи, B и коэффициентах ошибок, BER

Заключение

Результаты расчета показывают, что использование оптических усилителей в ВОЛС ведет к снижению отношения сигнал-шум передаваемых информационных сигналов. Основным источником шумов ОУ является усиленное спонтанное излучение (ASE), которое влияет на ограничение минимальной мощности в канале. Проведенные оценки снижения помехоустойчивости системы связи при различных параметрах позволяют выбрать соответствующее приемное и передающее оборудование, обеспечивающее требуемую помехозащищенность передаваемых сигналов.

Литература

1. Убайдуллаев Р.Р. Протяженные ВОЛС на основе EDFA //Lightwave Russian Edition. – 2003. – № 1. – С. 22–28.
2. Desurvire E. Erbium-doped fiber amplifiers: principles and applications – NY: John Wiley & Sons, 1994. – 770 p.
3. Heffner H. The Fundamental Noise Limit of Linear Amplifiers //Proceedings of the IRE. – July 1962. – Vol. 50. – Iss. 7. – P. 1604–1608.
4. Фриман Р. Волоконно-оптические системы связи. – 2-е изд., перераб. и доп. – М.: Техносфера, 2006. – С. 512.

СИСТЕМНЫЙ АНАЛИЗ, МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И УПРАВЛЕНИЕ В ТЕХНИЧЕСКИХ СИСТЕМАХ	3
Неугодникова Л.М. (Уфимский государственный авиационный технический университет). Система поддержки принятия решений при оценке типовых траекторий полета	3
Беляев А.В., Гаврилов М.И., Ситников А.Н. Расчет коэффициентов прохождения и отражения в системе связанных квантовых волноводов	9
Петтай П.П. Решение матричных игр в нечетких смешенных стратегиях.....	15
Ямалетдинова А.Р. Математическое моделирование в задачах биологической эволюции	23
Соколова Ю.А. О повышении защищенности перекодированных изображений.....	28
Блинова И.В., Попов С.И. (Санкт-Петербургский государственный университет информационных технологий, механики и оптики), Попов А.И. (Санкт-Петербургский государственный университет). Модель распределения задач при клауд-вычислениях и квантово-компьютерные аналогии	32
Рабыш А.Ю., Беликов Ю.В. Объектно-ориентированная методика проектирования систем с элементами, общими по функциональности, но разнопараметрических (на примере составного зеркала).....	35
Лочехин А.В. Начальная выставка и калибровка гироскопа на базе бескарданного электростатического гироскопа и блока микромеханических датчиков.....	41
Беляев М.Г., Приходько П.В. (Московский физико-технический институт (государственный университет)), Бурнаев Е.В. (Институт проблем передачи информации им. А.А. Харкевича РАН). Аппроксимация многомерной зависимости с помощью градиентного усиления регрессионной нейронной сети и ее применение в быстром методе аэродинамического расчета для задач проектирования.....	46
Маланов А.А. (Московский физико-технический институт (государственный университет)), Ветров Д.Е. (Международный научно-исследовательский институт проблем управления). Алгоритмы построения области генерации многомерных объектов	52
Вачнадзе К.Г. (Московский физико-технический институт (государственный университет)). Моделирование промывки глаза при хирургии катаракты.....	58
Кобликов А.А. (Московский государственный университет им. М.В. Ломоносова). Синтез фрактальных диаграмм направленности антенн.....	64
Нестеренко М.Ю., Леонов Д.В., Быков М.Д. (Оренбургский государственный университет). Подход к моделированию эффективных автоматизированных систем на кластерах.....	67
Виноградова А.А., Коваленко П.П., Недоцука Г.А. Вейвлет-фрактальная обработка изображения.....	72
Недоцука Г.А., Коваленко П.П., Виноградова А.А. Использование нечеткой логики для анализа и обработки изображений.....	77
Бурнаев Е.В. (Институт проблем передачи информации им. А.А. Харкевича РАН), Ерофеев П.Д. (Московский физико-технический институт (государственный университет)). Нелинейное снижение размерности с помощью многообразия проектирования и его применение для решения задачи обсчета профилей крыла самолета.....	81

Тимонина А.В. (Московский физико-технический институт (государственный университет)). Итеративные алгоритмы ранжирования с применением к наукометрии и интернету.....	87
Фьонг В.К., Чьен Н.К. (Санкт-Петербургский государственный электротехнический университет, ЛЭТИ). Нейронечеткое управление электромеханическими объектами с упругими связями.....	93
Арановский С.В. Алгоритм идентификации параметров системы двигатель – двухмассовый механизм	99
Вахитов А.Т., Павленко Д.В. (Санкт-Петербургский государственный университет). Применение метода стохастической аппроксимации с пробным одновременным возмущением на входе в задачах стереозрения и определения движения.....	105
Колюбин С.А., Пыркин А.А. Компенсация неизвестного смещенного гармонического возмущения для нелинейного объекта с запаздыванием по управлению.....	108
Колюбин С.А., Пыркин А.А. Адаптивная стабилизация маятника с инерционным маховиком в положении неустойчивого равновесия	115
Секарин К.Г. Исследование напряженно-деформированного состояния соединений оптических деталей трансляторов изображения.....	121
Коршиков С.Е., Заикина Н.В., Рыбалко Г.С. (Самарский государственный технический университет). Моделирование полей температур и термонапряжений в процессе нагрева алюминиевых заготовок, вращающихся в магнитном поле постоянного тока.....	127
Пушкин А.А. (Донбасская государственная машиностроительная академия). Исследование и математическое моделирование динамических процессов в главном приводе рабочего рольганга.....	133
Фролов А.С. (Сибирский государственный технологический университет, Красноярск). Математическое моделирование имитационной поверки погружных вихревых расходомеров.....	139
ТЕХНОЛОГИИ ПРОИЗВОДСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	145
Конончук Д.О., Окуловский Ю.С. (Уральский государственный университет им. А.М. Горького). Универсальный пакет поддержки интеллектуальных вычислений GANS	145
Платунова С.М. Модель сетевого взаимодействия прикладного программного обеспечения.....	152
Карпушинский А.М. Анализ неявного потока управления и его влияние на выбор структурных критериев тестирования объектно-ориентированных программ.....	157
Владыкин А.А. Эффективный разбор XML на основе автоматного подхода.....	163
Суясов Д.И. Утончение изображений символов на основе клеточных автоматов.....	169
Колесников Е.И. Обзор маршрутов проектирования прикладного программного обеспечения для ПЛИС/ASIC/SoC на основе языков C/C++.....	175
Малаховски Я.М. Реализация конечных автоматов на функциональных языках программирования	183
Филимонов А.В. (Московский физико-технический институт (государственный университет)). Применение шаблонов для разработки компонентов программных комплексов систем физической защиты.....	189

ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ	194
Лобачев П.С., Айтуганов Б.М. (Всероссийский научно-исследовательский институт гидрометеорологической информации – мировой центр данных, Обнинск). Технология виртуальной телекоммуникационной сети ЕСИМО	194
Платунова С.М. Обеспечение качества обслуживания гетерогенной сети.....	199
Павлов К.А. (Московский государственный институт электронной техники). Исследование и разработка системы передачи цифровых данных через тело человека	205
Ван Цзянь. Исследование методов маркирования изображений цифровыми водяными знаками	211
Лукьянов Н.М., Дергачев А.А. Алгоритмы обработки информационных потоков в распределенной системе хранения данных	217
Сорокин С.В. (Московский физико-технический институт (государственный университет)). Единый регистр нормативных правовых актов Российской Федерации.....	223
Ледовской Е.И., Татарков М.Ю. (Северо-Кавказский государственный технический университет, Ставрополь). Повышение оперативности информационного обмена в спутниковой сингулярной конвергентной информационной системе за счет интеллектуальной обработки информации	228
Ледовской Е.И., Татарков М.Ю. (Северо-Кавказский государственный технический университет, Ставрополь). Использование технологии считывания показаний активности головного мозга для повышения оперативности информационного обмена в спутниковой сингулярной конвергентной информационной системе	233
Ледовской Е.И., Татарков М.Ю. (Северо-Кавказский государственный технический университет, Ставрополь). Считывание движений кисти руки для ускорения информационного обмена в спутниковой сингулярной конвергентной информационной системе.....	238
Федоров С.А. (Санкт-Петербургский государственный политехнический университет). Встраиваемое программное обеспечение высокоскоростной сетевой камеры для оптической диагностики быстропротекающих процессов.....	242
Токалов Н.С., Антонов С.Е. История и тенденции развития мобильных средств передачи информации.....	248
Осипов А.В. (Санкт-Петербургский государственный инженерно-экономический университет). Целочисленное решение задач оптимизации компьютерных систем в среде Mathcad.....	253
Токалов Н.С. Обеспечение безопасности в мобильных технологиях	259
Скшидлевский А.А. Управление работой компьютерного класса, построенного на основе терминальной системы	263
Прокопьев М.В. (Московский физико-технический институт (государственный университет)). Методы увеличения эффективности использования орбитально-частотного ресурса геостационарной орбиты для систем спутниковой связи и вещания.....	269
Новоселов Н.А. (Южный федеральный университет, Ростов-на-Дону). Роль федеральных университетов в развитии сферы информационных технологий	274
Янушковский А.Ю. (Санкт-Петербургский университет кино и телевидения). Точность параметров демодулятора в системах с модуляцией КАМ	280
Антонов С.Е. Разработка системы визуального управления компьютером	286
Багаветдинов Д.Н. Расчет помехоустойчивости участка ВОЛС с оптическими усилителями.....	292

Сборник трудов конференции молодых ученых, Выпуск 4.
МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И ПРОГРАММНОЕ
ОБЕСПЕЧЕНИЕ / Главный редактор д.т.н., проф. В.Л. Ткалич. - СПб: СПбГУ
ИТМО, 2009. - 301 с.

СБОРНИК ТРУДОВ КОНФЕРЕНЦИИ МОЛОДЫХ УЧЕНЫХ
Выпуск 4

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И ПРОГРАММНОЕ
ОБЕСПЕЧЕНИЕ

Главный редактор доктор технических наук,
профессор В.Л. Ткалич Дизайн обложки Т.В.
Точилина Редакционно-издательский отдел СПбГУ
ИТМО Зав. РИО Н.Ф. Гусарова Лицензия ИД №
00408 от 05.11.99. Подписано в печать 31.03.09.
Заказ 2101. Тираж 100 экз.