

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ
ТОЧНОЙ МЕХАНИКИ И ОПТИКИ (ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ)**

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК

Выпуск 6

**ИНФОРМАЦИОННЫЕ,
ВЫЧИСЛИТЕЛЬНЫЕ
И УПРАВЛЯЮЩИЕ СИСТЕМЫ**



**САНКТ-ПЕТЕРБУРГ
2002**

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

Главный редактор

ВАСИЛЬЕВ Владимир Николаевич – доктор технических наук, профессор, заслуженный деятель науки и техники Российской Федерации, ректор

Члены редколлегии

ГУРОВ Игорь Петрович – доктор технических наук, профессор, зам декана факультета компьютерных технологий и управления

КАРАСЕВ Вячеслав Борисович – кандидат технических наук, профессор, проректор

КОЛЕСНИКОВ Юрий Леонидович – доктор физико-математических наук, профессор, проректор

МУСАЛИМОВ Виктор Михайлович – доктор технических наук, профессор кафедры мехатроники

НИКИФОРОВ Владимир Олегович – доктор технических наук, профессор, проректор

ПОДЛЕСНЫХ Виктор Иванович – кандидат экономических наук, доцент, зам. декана гуманитарного факультета

ПУТИЛИН Эдуард Степанович – доктор технических наук, профессор, зав. кафедрой оптических технологий

СТАФЕЕВ Сергей Константинович – доктор технических наук, профессор, декан естественнонаучного факультета

СТУДЕНИКИН Леонид Михайлович – зам. начальника научно-исследовательской части

ТАРЛЫКОВ Владимир Алексеевич – доктор технических наук, профессор кафедры квантовой электроники и биомедицинской оптики

ЯКОВЛЕВ Евгений Борисович – доктор технических наук, профессор кафедры лазерных технологий и экологического приборостроения

Секретариат

ГУСАРОВА Наталия Федоровна – кандидат технических наук, доцент кафедры технологий профессионального обучения, ученый секретарь

КАЗАР Людмила Николаевна – начальник отдела охраны интеллектуальной собственности, ответственный секретарь

ПОЛУФИНАЛЬНЫЕ СОРЕВНОВАНИЯ ЧЕМПИОНАТА МИРА ПО ПРОГРАММИРОВАНИЮ 2001-2002 гг.

**В.Н. Васильев, М.А. Казаков, Г.А. Корнеев,
В.Г. Парфенов, А.С. Станкевич**

Командный чемпионат мира по программированию является самым престижным интеллектуальным состязанием молодой программистской элиты в мировом компьютерном сообществе. Представлены материалы по организации и проведению основных этапов чемпионата мира по программированию и результаты полуфинальных соревнований чемпионата 2001-2002 гг.

Ежегодный студенческий командный чемпионат мира по программированию среди сборных команд высших учебных заведений (ACM International Collegiate Programming Contest) проводится наиболее авторитетной в компьютерном мире международной организацией ACM – Association for Computing Machinery ("the First Society in Computing") – с 1977 года. Эти соревнования являются самыми престижными интеллектуальными состязаниями молодой программистской элиты в мировом компьютерном сообществе. Соревнования рассматриваются ведущими компьютерными фирмами как источник их пополнения наиболее квалифицированными кадрами и традиционно поддерживаются крупнейшими компьютерными корпорациями. В последние годы четыре сезона подряд генеральным спонсором чемпионата была корпорация Microsoft, вложившая в организацию соревнований 27 миллионов долларов. Начиная с сезона 1997/98 гг., генеральным спонсором стал главный конкурент Microsoft – корпорация IBM, вступившая с ней в жесткую борьбу за лучшие головы планеты. В сезоне 2001/2002 гг. в 26-м чемпионате принимают участие около 3000 команд из более 1500 ведущих вузов 70 стран мира.

Согласно выработанным за более чем двадцатилетнюю историю правилам проведения соревнований команде, состоящей из трех участников, предоставляется один компьютер и предлагается в течение пяти часов решить максимальное число из предложенных задач, число которых обычно лежит в диапазоне от 7 до 9. Побеждает команда, решившая наибольшее число задач, а в случае равенства числа решенных задач – команда, затратившая меньше времени.

Схема организации чемпионата мира включает три этапа: четвертьфинальный, полуфинальный и финальный. На полуфинальных этапах команды соревнуются в сезоне 2001/2002 гг. в 29 региональных группах, среди которых распределены вузы стран всех континентов. Команды-победительницы полуфинальных состязаний выходят в финал, который состоится 20-24 марта 2002 г. в Гонолулу, Гавайские острова, США. В финале примут участие 60 команд из ведущих университетов мира. По результатам финальных соревнований определится команда – чемпион мира, а также команды – чемпионы континентов. В силу огромной конкуренции в финале используется специальная схема награждения. Командам, решившим одинаковое с чемпионом мира число задач, но уступившим ему по времени, присуждаются золотые медали, решившим на одну меньше — серебряные, на две – бронзовые.

Россия впервые получила право на организацию собственной полуфинальной Северо-восточной Европейской группы в сезоне 1996/97 гг. Организацию соревнований Директорат чемпионата поручил Санкт-Петербургскому государственному институту точной механики и оптики (техническому университету), студенты которого успешно выступают в чемпионате. Северо-восточный европейский

регион включает страны бывшего СССР. В этом году полуфинальные соревнования прошли в Санкт-Петербурге в шестой раз.

С самого начала в течение пяти лет в качестве генерального регионального спонсора состязаний выступает выставочное объединение "Рестэк", предоставившее молодым российским программистам весомую материальную поддержку при организации их поездок на финал. Полуфинальные соревнования являются важной частью научно-образовательной программы известной выставки информационных технологий "Инвеком", проводимой компанией "Рестэк". В этом году спонсором состязаний выступают также Консорциум петербургских софтверных компаний "Форт Росс", а также известная корпорация Вестерн Юнион.

Итоги отборочных четвертьфинальных состязаний сезона 2001/2002 гг. показали огромный интерес вузов к этим соревнованиям. В частности, в этом году было организовано восемь четвертьфинальных региональных групп, охвативших всю территорию России и стран ближнего зарубежья: Дальневосточная (вуз-организатор – Дальневосточный ГУ), Восточно-Сибирская (Красноярский ГТУ), Западно-Сибирская (Новосибирский ГТУ), Уральская (Пермский ГУ), Южно-Поволжская (Саратовский ГУ), Центральная (Рыбинская государственная авиационная технологическая академия), Северо-западная (СПбГИТМО(ТУ)), Западная (Белорусский ГУ). В прошлом сезоне была организована также новая Закавказская полуфинальная группа, включающая вузы Армении, Азербайджана и Грузии. Всего в четвертьфинальных и полуфинальных соревнованиях приняла участие 301 команда из 148 вузов, представляющих практически все ведущие классические и технические университеты России, Азербайджана, Армении, Белоруссии, Грузии, Киргизии, Литвы и Эстонии.

Таким образом, к настоящему времени эти соревнования превратились в крупнейший за все время по географии и числу участников творческий научно-технический студенческий форум России и стран ближнего зарубежья. Северо-восточная Европейская региональная полуфинальная группа стала самой большой в чемпионате мира по числу участвующих команд, что позволило увеличить за прошедшие шесть лет число мест в финале для этой группы с двух до семи.

В этом году одновременно с соревнованиями студентов в Санкт-Петербурге проходила и Вторая Всероссийская командная олимпиада школьников по программированию, организуемая по правилам олимпиад ACM.

Как и в предыдущие годы, в рамках соревнований была проведена беспрецедентная акция по обеспечению участия в состязаниях в реальном масштабе времени команд сибирских вузов с использованием сети RUNNet. Команды ведущих вузов Барнаула, Новосибирска, Томска, Омска, Красноярска, Владивостока и других вузов Сибири и Дальнего Востока (32 команды) разместились во время проведения петербургского тура (в Аничковом дворце собрались 58 команд) в узле федеральной университетской компьютерной сети RUNNet, расположенном в Алтайском государственном техническом университете, и соревновались под наблюдением представителей жюри, которые специально прибыли в Барнаул. В Грузинском техническом университете в Тбилиси собрались почти 20 команд. Такая же схема соревнований была использована и для проведения олимпиады школьников. Полуфинальные соревнования дали яркий пример использования сети RUNNet для формирования единого научно-образовательного пространства России и, как следствие, укрепления единства и целостности Российского государства.

По результатам состязаний были определены семь команд – участниц финала чемпионата мира, а также команда – чемпион России по программированию 2001 года.

В 1996, 1997, 1999 и 2000 гг. чемпионами России становились петербургские команды: Государственного университета (трижды) и Института точной механики и оптики (технического университета). Эти же команды имеют лучшие достижения в

финалах мирового первенства: 2001 г. – СПбГУ – первое место и звание чемпионов мира, СПбГИТМО(ТУ) – золотые медали, третье место в абсолютном мировом зачете; 2000 г. – СПбГУ – первое место и звание чемпионов мира, СПбГИТМО(ТУ) – серебряные медали, четвертое место в абсолютном зачете, 1999 г. – СПбГИТМО(ТУ) – золотые медали, третье место в абсолютном зачете, 1998 г. – СПбГУ – золотые медали, второе место в абсолютном зачете. В 1998 г. чемпионом России стала команда МГУ.

В Санкт-Петербурге соревнования студентов и школьников прошли в Аничковом дворце с 24 по 29 ноября 2001 г.

Полуфинальные соревнования чемпионата мира по программированию отличались традиционно острым соперничеством московской и петербургской школ программирования, закончившимся победой петербуржцев, а также мощным наступлением команд провинциальных вузов, три из которых – Саратовский ГУ, Орловский ГТУ и Петрозаводский ГУ – впервые пробились в финал первенства.

Предварительные прогнозы были неутешительны для Санкт-Петербурга. Две великие петербургские команды, мировые лидеры прошедших двух сезонов, двукратные чемпионы мира – студенты Санкт-Петербургского государственного университета и серебряные-2000 и золотые-2001 призеры студенты Санкт-Петербургского государственного института точной механики и оптики – закончили свои выступления (в финале студент может выступить только два раза независимо от результата). В связи с этим резко возросли шансы на завоевание первого места у команд Московского государственного университета, которые уже предвкушали взятие близкого реванша у петербургских команд – за прошедшие пять сезонов москвичам удалось отобрать у петербуржцев звание чемпионов России только один раз. К тому же на этих соревнованиях "большую" тройку вузов – СПбГУ, СПбГИТМО, МГУ, занимавших три первые места в рейтинге достижений вузов полуфинальной группы за последние три года, "покинул" СПбГУ. К сожалению, молодые команды Санкт-Петербургского государственного университета, сформированные в основном из первокурсников, не выдержали напряженнейшей пятичасовой борьбы. В итоге лучшая команда СПбГУ оказалась лишь на 35 месте.

На старте соревнований команды дружно "набросились" на задачу С. Первыми ее решили на 24 минуте первая команда Уральского ГУ и команда Самарского муниципального университета. За ними на 26 минуте последовали третья команда Саратовского ГУ и четвертая команда СПбГИТМО. Неожиданностью соревнований стал быстрый старт третьей команды Саратовского государственного университета, которая на 48 минуте сдала вторую задачу В. На 53 минуте вторую задачу D решила и четвертая команда первокурсников СПбГИТМО, которая в начале ноября на четвертьфинальных соревнованиях одержала сенсационную победу, выиграв первое место и звание чемпиона Санкт-Петербурга. Отметим, что этот успешный старт петербургских первокурсников оказался их последним успехом на соревнованиях. Не выдержав испытания внезапно открывшимися "грандиозными перспективами", они не сумели сдать за оставшиеся четыре часа ни одной задачи. После истечения первого часа соревнований две команды решили по две задачи и 16 команд – по одной. В число этих 18 команд входили 3 команды Саратовского ГУ, по 2 команды МГУ и СПбГИТМО и только четыре команды (Саратовский ГУ-3, СПбГИТМО-1, Петрозаводский ГУ-2, Орловский ГТУ-2), которые в итоге вышли в финал. Будущий победитель – команда СПбГИТМО-1 – занимала 12 место.

На втором часе борьбы на 63 минуте вторую задачу сдала команда Петрозаводского ГУ-2, на 76 минуте – МГУ-1, на 82 минуте – СПбГИТМО-1, на 85 минуте – МГУ-4, на 88 минуте – СПбГИТМО-3, на 89 минуте – Орловского ГТУ-2. После полутора часов сформировалась лидирующая группа из 8 команд, решивших по две задачи. Во второй половине второго часа команды начали сдавать третьи задачи.

Первой это сделала на 93 минуте команда Саратовского ГУ-3, за ней последовали команда СПбГИТМО-1 на 96 минуте и команда МГУ-1 на 102 минуте. Эти команды и составили лидирующую тройку. В конце второго часа активизировалась вторая команда СПбГИТМО. Сдав две задачи на 68 и 101 минутах, она, наконец, с шестой попытки сдала свою третью задачу С на 111 минуте. На втором часу соревнований пошли вперед и не сдавшие за первый час ни одной задачи команды Белорусского ГУ – все три его команды решили по две задачи. В итоге на исходе второго часа 4 команды решили по три задачи и 12 команд – по две задачи. В число этих 16 команд входили 4 команды СПбГИТМО, 3 команды МГУ и 3 команды БГУ.

В самом начале третьего часа на 121 минуте третью задачу сдала команда БГУ-1 и вышла на четвертое место. На 129 минуте свою четвертую задачу А решила первая команда СПбГИТМО и надолго заняла первую строчку турнирной таблицы. На 130 и 139 минутах две задачи сдала первая команда Орловского ГТУ, на которую тренеры возлагали основные надежды, и вышла с тремя решенными задачами на шестое место. Однако, работая в дальнейшем в течение почти трех часов над тремя задачами, ребята из этой команды так и не смогли сдать хотя бы еще одну задачу. На 160 минуте четвертую задачу решила вторая команда СПбГИТМО и вышла на второе место, правда, проигрывая первой команде своего вуза больше 200 минут штрафного времени. На 161 минуте третью задачу сдала команда Орловского ГТУ-2 и обошла первую команду своего вуза. На 163 и 166 минутах третьей задачи сдали четвертая команда МГУ, переместившаяся на пятое место, и третья команда первокурсников СПбГИТМО. На 169 и 172 минутах свои четвертые задачи решили первая и третья команды МГУ и вышли соответственно на второе и четвертое места. На 175 минуте третью задачу решила вторая команда Петрозаводского ГУ и поднялась на седьмое место, заявив о своих претензиях на место в финале. Таким образом, к исходу третьего часа по две команды СПбГИТМО и МГУ решили по четыре задачи, и 8 команд решили по 3 задачи. Третьей команде Саратовского ГУ, которая сдала свою третью задачу еще на 93 минуте, не удалось решить за прошедшие с того момента почти полтора часа ни одной задачи. Стало казаться, что все призовые места разыграют между собой многочисленные команды СПбГИТМО и МГУ.

Третий час соревнований стал чрезвычайно успешным для команды Башкирского ГУ, которая на 130, 147 и 181 минутах сдала три задачи и вошла в желанную финальную семерку. Кроме студентов Башкирии, на исходе третьего часа в семерку входили команды СПбГИТМО, МГУ, Саратовского ГУ, Петрозаводского ГУ, БГУ и Орловского ГТУ. В решающие заключительные два часа соревнований к борьбе за выход в финал были готовы также и находившиеся в непосредственной близости от лидирующей группы с двумя решенными задачами команды Уральского ГУ, Новосибирского ГУ и Красноярского ГТУ.

В начале четвертого часа на 194 минуте четвертую задачу решила и поднялась на пятое место третья команда первокурсников СПбГИТМО. Наконец, на 195 минуте свое долгое молчание прервала третья команда Саратовского ГУ, которая решила-таки свою четвертую задачу и вышла на второе место, "прорвавшись сквозь строй" команд СПбГИТМО и МГУ. В это время у тренеров команд СПбГИТМО начало вызывать опасения поведение их лидирующей первой команды, поскольку после сдачи на 129 минуте четвертой задачи она проявила себя только двумя неудачными заходами (на 170 и 193 минутах) на задачу Н. Тренеры с облегчением вздохнули только на 203 минуте, когда была, наконец, добыта эта задача. Дальше события стали приобретать захватывающий характер. На 206 минуте последовал новый успех команды Башкирского ГУ, она сдала четвертую задачу и вышла на шестое место. На 208 минуте свою пятую задачу решила команда Саратовского ГУ, сделав весомую заявку на призовое место. На 214 минуте четвертую задачу решила команда ОрГТУ-2 и вышла на

пятое место, поставив в весьма сложное положение первую команду своего вуза, которой для выхода в финал теперь стало необходимо решить не менее пяти задач. На 219 минуте вторая команда Белорусского ГУ решила четвертую задачу и опередила первую команду Белорусского ГУ, имевшую три решенные задачи. Третий час соревнований надолго запомнится и команде Новгородского ГУ. В течение этого часа ребята из Новгорода сдали на 196, 214, 219 и 230 минутах 4(!) задачи и переместились из группы команд, не решивших ни одной задачи и располагавшихся ниже 65 места, на 10 место, получив реальные шансы на выход в финал, которые, к сожалению, так и не удалось реализовать. Наконец, под занавес четвертого часа на 236 минуте пятую задачу сдала вторая команда СПбГИТМО и поднялась на третье место. В замороженной таблице результатов три команды решили по 5 задач и 7 команд по 4 задачи. В финальную семерку, наряду с командами СПбГИТМО, Саратовского ГУ, МГУ, БГУ, Орловского ГТУ, вошли на тот момент времени также и команды Башкирского и Новгородского ГУ, вытеснившие из числа финалистов команду Петрозаводска.

В начале пятого часа на 244 минуте пятую задачу решила первая команда МГУ, и вследствие этого вторая команда СПбГИТМО опустилась с третьего на четвертое место. На 250, 252 и 260 минутах свои четвертые задачи решили команды Петрозаводского ГУ, Новосибирского ГУ и первая команда Белорусского ГУ, которые отеснили команды Башкирии, Новгорода и вторую команду Белорусского ГУ из финальной семерки благодаря меньшему числу штрафных минут. На 266 минуте команда Саратовского ГУ решила шестую задачу и вернулась на первое место, через минуту шестую задачу сдала первая команда МГУ и перешла на второе место. Лидер соревнований команда СПбГИТМО-1 за прошедшее с начала пятого часа время сделала неудачную попытку сдачи одной из двух самых сложных задач F и никак не проявила себя в задаче E, которую решили уже многие команды. Сидящие в жюри руководители команд СПбГИТМО испытывали малоприятные переживания – терялись казавшиеся такими близкими первое место и звание чемпиона России. Однако, к счастью для них, эти переживания оказались недолгими – на 270 минуте со второго захода первая команда СПбГИТМО сдала задачу F и снова вернула себе первое место. Стало ясно, что она сможет решить и семь задач, поскольку после сдачи пятой задачи для решения задачи E у нее было вполне достаточно времени. Это предположение оправдалось – за тринадцать минут до истечения контроля времени первая команда СПбГИТМО решила седьмую задачу и одержала блестящую чистую победу. Борьба за места в первой шестерки продолжалась до последнего момента времени. На 280 минуте шестую задачу решила вторая команда СПбГИТМО и, поднявшись на третье место, впервые за шестилетнюю историю полуфиналов вытеснила первую команду МГУ из призовой тройки. Сдав на 274 и 284 минутах две задачи, вышла на почетное пятое место с пятью решенными задачами опытнейшая четвертая команда МГУ – финалист Орландо-2000. И, наконец, на 283 минуте пятую задачу решила первая команда Белорусского ГУ, обеспечив себе шестое место.

Основные результаты полуфинальных соревнований (первые 20 мест) представлены в таблице с указанием числа решенных задач и штрафного времени (в мин.).

1. СПбГИТМО(ТУ)-1	7	1190
2. Саратовский ГУ	6	956
3. СПбГИТМО(ТУ)-2	6	1136
4. МГУ-1	6	1184
5. МГУ-4	5	856
6. Белорусский ГУ-1	5	1002
7. Орловский ГТУ-2	4	579
8. Петрозаводский ГУ-2	4	595
9. МГУ-3	4	634

10. СПбГИТМО(ТУ)-3	4	647
11. МГУ-2	4	667
12. Новосибирский ГУ-1	4	697
13. Белорусский ГУ-2	4	749
14. Башкирский ГУ	4	764
15. Уральский ГУ-1	4	870
16. Университет Тарту	4	897
17. Новгородский ГУ	4	1179
18. Белорусский ГУ транспорта	3	445
19. Орловский ГТУ-1	3	500
20. Красноярский ГТУ-1	3	523

Таким образом, первое место и звание чемпиона России, спустя пять лет после первого подобного успеха, завоевала первая команда СПбГИТМО. Новые чемпионы России – студенты знаменитой кафедры компьютерных технологий факультета информационных технологий и программирования СПбГИТМО(ТУ) Тимофей Бородин, Александр Штучкин и Евгений Южаков – закончили школы в Костроме, Саратове и Котласе и были приглашены на кафедру за успехи на Всероссийских олимпиадах школьников по информатике. Отметим, что в настоящее время на кафедре компьютерных технологий СПбГИТМО собраны в большой концентрации победители Всероссийских олимпиад школьников по точным наукам. Среди примерно 180 студентов шести курсов обучается около 60 студентов, получивших в школьные годы дипломы на Всероссийских олимпиадах школьников по информатике, что составляет почти третью часть от всех дипломантов Всероссийских олимпиад. По общей численности дипломантов кафедра сопоставима только с математическими факультетами МГУ и СПбГУ. Тренировал команду студент четвертого курса этой же кафедры, серебряный медалист Орландо-2000 и золотой медалист Ванкувера-2001 Андрей Станкевич.

В команду института точной механики и оптики, занявшую третье место, входили братья Андрей и Олег Пестовы и Иван Прокушкин, которые в школьные годы учились в Вятке и занимались в известном центре, возглавляемом одним из лучших российских педагогов Станиславом Михайловичем Окуловым. Интересно, что Евгений Южаков и Олег Пестов уже начали свою трудовую деятельность в известной петербургской фирме "Аркадия" в группе, руководимой председателем технического комитета соревнований, золотым медалистом Эйндровена-99 и тренером команды СПбГИТМО – золотого призера Ванкувера-2001 Матвеем Казаковым.

Неплохо выступила и команда первокурсников ИТМО, занявшая достаточно престижное 10 место. Результаты соревнований – это большой успех Центра подготовки одаренных программистов, работающего в институте точной механики и оптики и получившего в настоящее время без преувеличения мировое признание.

Впервые выиграв путевку в финал мирового первенства, блестящую победу одержала занявшая второе место и завоевавшая звание вице-чемпиона России команда Саратовского государственного университета в составе Андрея Лазарева, Михаила Мирзаянова и Ильи Эльтермана. Команду готовили к соревнованиям Наталия Львовна Андреева и Антонина Гавриловна Федорова. Успех саратовской школы подготовки программистов тем более значителен, что в первых командах СПбГИТМО и МГУ выступали Александр Штучкин и Максим Бабенко, которые тоже закончили саратовские школы. Этот успех – достойный итог многолетней работы, включающей, в частности, и проведение четвертьфинальных соревнований Южного (Поволжского) региона.

Традиционно сильно выступили и прошли в финал неоднократные участники финалов – команды Московского и Белорусского государственных университетов.

Отличительной особенностью прошедшего полуфинала было успешное выступление команд нестоличных вузов. Жаркая борьба за путевки в финал шла в основном между командами провинциальных российских вузов. Кроме СПбГИТМО и МГУ, ни один вуз из Москвы и Санкт-Петербурга не принял в ней участие – команда МИФИ оказалась на 24 месте, МАИ – на 46, МФТИ – на 68, команды СПбГУ – на 35, 36 и 65. Да и сильнейшие команды МГУ и СПбГИТМО "сражались друг против друга" иногородними студентами. Указанные обстоятельства наводят на определенные размышления об изменениях в распределении компьютерного интеллектуального студенческого потенциала России. В результате захватывающей борьбы путевки в финал впервые завоевали команды Орловского ГТУ и Петрозаводского ГУ, а также финалист Орландо-2000 команда Новосибирского ГУ, выступавшая в Сибирской полуфинальной группе в Барнауле и вернувшая себе звание чемпиона Сибири и Дальнего Востока.

По решению жюри соревнований дипломы АСМ с указанием занятого места были присуждены 17 командам, решившим четыре и более задач. Дипломы Всероссийской командной олимпиады по программированию были вручены командам, решившим 7, 6 и 5 задач, 2 степени – решившим 4 задачи, 3 степени – 3 задачи.

Следующий этап чемпионата – финал – пройдет в конце марта 2002 г. в городе Гонолулу на Гавайских островах.

Успехи петербургских студентов в двух последних финалах 2000 и 2001 гг. внесли, по общему мнению, огромный вклад в улучшение образа Санкт-Петербурга, да и всей России, в области информационных технологий. Пожелаем же новому поколению продолжить эту замечательную традицию.

Более подробную информацию можно получить у декана факультета информационных технологий и программирования СПбГИТМО(ТУ), члена Международного организационного комитета чемпионата мира по программированию Владимира Глебовича Парфенова, т. 233-42-98, 232-43-18, parfenov@mail.ifmo.ru

МЕТОДЫ УПРАВЛЕНИЯ ТРАФИКОМ
В МУЛЬТИСЕРВИСНЫХ СЕТЯХ

Р.Т. Алиев

Эффективность функционирования вычислительных сетей в значительной степени определяется временными задержками при передаче данных между пользователями сети. Минимизация задержек в сетях с однородным трафиком обеспечивается на этапе проектирования с использованием математических моделей массового обслуживания с однородным потоком заявок [1].

В настоящее время все большее распространение получают мультисервисные сети, характерной особенностью которых является неоднородность трафика [2, 3]. Неоднородность трафика заключается в передачи по телекоммуникационной сети пакетов нескольких типов (видео– и аудиопакетов, речевых пакетов, текстовых пакетов и т.д.), к которым предъявляются различные требования [4]. Эти требования формулируются в виде ограничений на время доставки пакетов разных типов, при этом ограничения могут быть двух видов:

1) вероятностные в виде допустимой вероятности γ_i^* превышения заданного ограничения τ_i^* на время задержки τ_i пакетов в телекоммуникационной сети:

$$P(\tau_i > \tau_i^*) < \gamma_i^* \quad (i = \overline{1, n});$$

2) средние в виде ограничения τ_i^* на среднее время $\overline{\tau_i}$ задержки: $\overline{\tau_i} < \tau_i^* \quad (i = \overline{1, n})$, где n – количество типов пакетов в сети.

Указанные ограничения могут быть выполнены за счет применения специальных методов управления трафиком, позволяющих эффективно распределить пропускную способность канала связи между пакетами разных типов, в частности, за счет оптимального распределения приоритетов. При этом актуальной становится задача оценки эффекта от введения приоритетов, предоставляемых пакетам, критичным к задержкам в телекоммуникационной сети. Решение этой задачи требует использования моделей с неоднородным потоком заявок, позволяющих выполнить анализ свойств приоритетных систем передачи данных и сформулировать рекомендации для проектирования приоритетных сетей, в частности, оценить требуемую пропускную способность каналов связи.

Приоритизация имеет смысл, когда маршрутизаторы или коммутаторы способны различать разные типы трафика. Для указания приоритета пакета может использоваться специальный байт "тип сервиса" (Type of Service — ToS). В IPv4 три первых бита (0-2) поля ToS позволяют устанавливать восемь уровней приоритета [4]:

- 111 — управление сетью (Network Control);
- 110 — межсетевое управление (Internetwork Control);
- 101 — CRITIC/ECR;
- 100 — сверхсрочный (Flash Overrider)
- 011 — срочный (Flash);
- 010 — неотложный (Immediate);
- 001 — приоритетный (Priority);
- 000 — обычный (Routine).

Для оценки эффективности приоритетных методов управления трафиком в мультисервисных сетях в качестве базовой модели канала связи будем использовать

систему массового обслуживания с неоднородным потоком пакетов n типов, поступающих в канал связи с интенсивностями $\lambda_1, \dots, \lambda_n$. Обозначим: V – пропускная способность канала связи, L_i – средняя длина пакета i -го типа. Для случая, когда пакеты одного класса имеют одинаковую длину, а потоки пакетов являются простейшими, средняя задержка пакета i -го типа при использовании метода управления трафиком на основе относительных приоритетов определяется по формуле

$$\tau_i = \frac{\sum_{j=1}^n \lambda_j L_j^2}{(V - \sum_{j=1}^{i-1} \lambda_j L_j)(V - \sum_{j=1}^i \lambda_j L_j)} + \frac{L_i}{V} \quad (i = \overline{1, n}). \quad (1)$$

В табл. 1 представлены результаты расчета временных задержек пакетов в канале связи для четырех уровней приоритетов и двух способов назначения приоритетов при использовании каналов связи с разными пропускными способностями.

Таблица 1

$V_{КС}$ (Кбит/с)			128	192			128	192
Код приоритета	λ_i (с ⁻¹)	L (байт)	τ_i (мс)		λ_i (с ⁻¹)	L (байт)	τ_i (мс)	
011	20	100	60,7	27,3	2	1500	152,4	86,7
010	10	500	128,1	53,4	5	1000	179,8	77,9
001	5	1000	401,4	101,4	10	500	539,6	90,1
000	2	1500	3143,7	175,5	20	100	4072,9	127,4
Загрузка КС			0,9375	0,625			0,9375	0,625

Анализ представленных результатов позволяет сформулировать следующие выводы.

1. Для обеспечения минимальной задержки пакетов всех типов приоритет необходимо предоставлять коротким пакетам.
2. При большой загрузке канала связи низкоприоритетные пакеты имеют практически недопустимые задержки, превышающие для речевых пакетов ограничения 150–300 мс [4].
3. При увеличении пропускной способности канала связи в 1,5 раза (со 128 до 192 Кбит/с) задержки для высокоприоритетных пакетов уменьшились более чем в два раза, а низкоприоритетных – более чем на порядок.

Одной из важных задач, решаемых на этапе проектирования корпоративной вычислительной сети, является определение требований к пропускным способностям каналов связи. Очевидно, что эти требования существенно зависят от нагрузки, создаваемой передаваемыми пакетами данных, и ограничений, налагаемых на величину задержки высокоприоритетных пакетов, к которым относятся, например, речевые пакеты.

Рассмотрим случай, когда ограничения заданы на среднее время задержки речевых пакетов, имеющих наивысший (первый) приоритет, в виде: $\bar{\tau}_1 < \tau_1^*$. Положим, что размеры пакетов всех типов одинаковы ($L_i = L$ для всех $i = \overline{1, n}$), а доля речевых пакетов в общей нагрузке составляет k ($0 \leq k \leq 1$), т.е.: $\lambda_1 = k\Lambda$. Тогда, решая квадратное неравенство $\bar{\tau}_1 < \tau_1^*$ с учетом (1), получим, что пропускная способность канала связи должна выбираться из условия

$$V > \frac{L}{2} \left\{ \frac{1}{\tau_1^*} + k\Lambda + \left[\left(k\Lambda - \frac{1}{\tau_1^*} \right)^2 + \frac{4\Lambda}{\tau_1^*} \right]^{\frac{1}{2}} \right\}. \quad (2)$$

Выражение в правой части неравенства (2) представляет собой нижнюю границу пропускной способности V_k канала связи, требующейся для передачи речевых пакетов с заданным качеством в сети с приоритетным управлением.

В процессе проектирования корпоративной сети обычно достаточно сложно задать долю k речевых пакетов в общей нагрузке. В то же время эта доля в течение суток может изменяться в значительных пределах. В связи с этим предлагается оценивать требуемую пропускную способность для всего диапазона изменения k . Для этого рассмотрим предельные случаи, когда $k \rightarrow 0$ и $k \rightarrow 1$. Тогда из (2) получим нижнюю и верхнюю границу пропускной способности канала связи:

$$V_{\min} = \frac{L}{2} \left\{ \frac{1}{\tau_1^*} + \left[\frac{1}{\tau_1^{*2}} + \frac{4\Lambda}{\tau_1^*} \right]^{\frac{1}{2}} \right\};$$

$$V_{\max} = L \left(\frac{1}{\tau_1^*} + \Lambda \right).$$

Можно показать, что последнее выражение для $V_{\max}$ соответствует требуемой пропускной способности при беспriorитетном управлении трафиком в узлах сети. Тогда эффект от введения приоритетного управления составит $\delta = (V_{\max} - V_k) / V_k$.

В табл. 2 представлены результаты расчета пропускной способности канала связи при различных значениях параметров нагрузки и ограничений на задержку пакетов. Для качественной передачи речевого трафика допустимая задержка составляет $\tau_1^* = 150$ –300 мс. Если для передачи речевого трафика задействовано N последовательных каналов связи, то, очевидно, что задержка в каждом канале не должна превышать τ_1^* / N .

Таблица 2

Доп. задержка, τ_1^* , мс	Интен- сивность, Λ , с ⁻¹	Доля речевых пакетов, k	Пропускные способности			Эффект
			V_k , кбит/с	$V_{\min}$, кбит/с	$V_{\max}$, кбит/с	δ , %
150	10	0,1	6,4	6,2	8,5	32,8
		0,3	11,3	9,2	18,8	66,4
		0,1	12,4	11,2	29,0	133,9
		0,3	15,2	11,2	29,0	90,8
		0,5	18,6	11,2	29,0	55,9
		0,99	28,8	11,2	29,0	0,7
300	10	0,1	4,1	3,9	6,8	65,9
		0,2	6,0	5,1	12,0	100,0
		0,3	8,5	6,0	17,1	101,2
		0,5	13,3	6,8	22,2	66,9
		0,99	27,1	7,5	27,3	0,7
			Длина пакетов $L = 64$ байт			

Анализ полученных результатов показывает, что введение приоритетного управления трафиком позволяет снизить требования к пропускным способностям каналов связи, при этом:

- уменьшение допустимой задержки в α раз требует увеличения пропускной способности канала менее чем в α раз;
- с увеличением нагрузки (интенсивности Λ) увеличивается эффект от введения приоритетного управления трафиком;
- с уменьшением доли k речевых пакетов в общей нагрузке этот эффект увеличивается.

Таким образом, предлагаемая модель приоритетного управления в канале связи мультисервисной сети позволяет определить пропускную способность канала и оценить эффект, достигаемый за счет использования приоритетного управления трафиком.

Литература

1. Клейнрок Л. Вычислительные системы с очередями. М.: Мир, 1979.
2. Кульгин М. Технология корпоративных сетей. СПб: Питер, 2000.
3. Кульгин М. Введение в систему управления трафиком // LAN. 1998. № 11.
4. Холл Э. Приоритизации трафика в сетях IP // Сети и системы связи. 1988. №11 (33).

АНАЛИЗ МЕТОДОВ ПЕРЕДАЧИ ВИДЕОИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ СЕТЯХ

В.Н. Васильев, Ю.В. Гугель, И.П. Гуров

Выполнен анализ проблемы передачи видеоинформации в телекоммуникационных сетях. Рассмотрены особенности методов управления параметрами передачи видеоинформации и кодирования видеоинформации с различной степенью сжатия и коррекцией ошибок. Проведен сравнительный анализ вариантов высокопроизводительных сетей, стандартов передачи и представления видеоинформации. Рассмотрены практические аспекты решения проблемы передачи видеоинформации

Введение

Передача видеоинформации становится все более актуальной для различных сфер жизнедеятельности – телевидения, научных исследований, медицины, современных технологий дистанционного обучения и др. В связи с активным развитием информационных технологий появились широкие технические возможности для реализации различных методов обработки и высококачественной передачи видеоинформации. Передача видеоинформации через сеть Интернет является важнейшей составляющей информационного потока для многих современных мультимедиа-приложений. При этом передача видеоинформации в реальном времени предъявляет повышенные требования к ширине полосы частот, задержкам передачи и допустимым потерям данных. В настоящее время сеть Интернет не всегда обеспечивает гарантированное качество обслуживания (Quality of Service – QoS). Кроме этого, неоднородность структуры сетей и характеристик систем передачи и приема видеоинформации затрудняет передачу в режиме распределенного доступа. Освоение наиболее подходящих стандартов сжатия, преобразования и представления видеоинформации, разработка протоколов и методов передачи составляют важную проблему в области развития информационных технологий. Для решения этой проблемы необходимо контролировать ситуации превышения пропускной способности каналов, осуществлять адаптивную подстройку характеристик сжатия и кодирования видеоинформации и обеспечивать динамическое управление уровнем допустимых ошибок передачи видеоинформации за счет предотвращения ожидаемых ошибок, дублирующей передачи данных, обеспечения малой заметности ошибок.

Использование новых возможностей информационных технологий становится более эффективным при оптимизации структуры информационных сетей, отдельных узлов и их взаимодействия. В наиболее совершенных системах критические характеристики распределяются в известной мере согласованно в структурной части, характеристиках оборудования и программном обеспечении, что повышает качество и надежность работы и расширяет функциональные возможности системы в целом.

В настоящей статье рассмотрены подходы к оптимизации условий передачи видеоинформации, предотвращения и коррекции ошибок передачи, а также выполнен анализ характеристик основных стандартов и протоколов, используемых при передаче видеоинформации в компьютерных сетях.

Особенности передачи видеоинформации в компьютерных сетях

Для обеспечения требуемого качества передачи видеоинформации в реальном времени требуется определенная ширина полосы или пропускная способность канала. Существуют два принципа потоковой передачи звука и изображения в Интернет – одноадресная передача данных (Unicast) и многоадресная передача (Multicast).

В режиме одноадресной передачи данных (см. рис. 1, а) сервер-отправитель информации формирует для каждого клиента отдельный поток данных, а компьютер пользователя периодически отправляет на сервер подтверждение о доставке

информационных пакетов. При этом требуемые мощность сервера и полоса пропускания канала связи прямо пропорциональны количеству клиентов. Как видно из примера рис. 1, а, пять копий потока данных пересылаются через участок 1 и три копии – через участок 2. Одноадресная передача данных используется, в основном, в системах "видео по запросу" (video-on-demand). Она удобна для работы с видеоархивами единичных пользователей и для распространения вещания на абонентов, подключенных по низкоскоростным коммутируемым линиям связи.

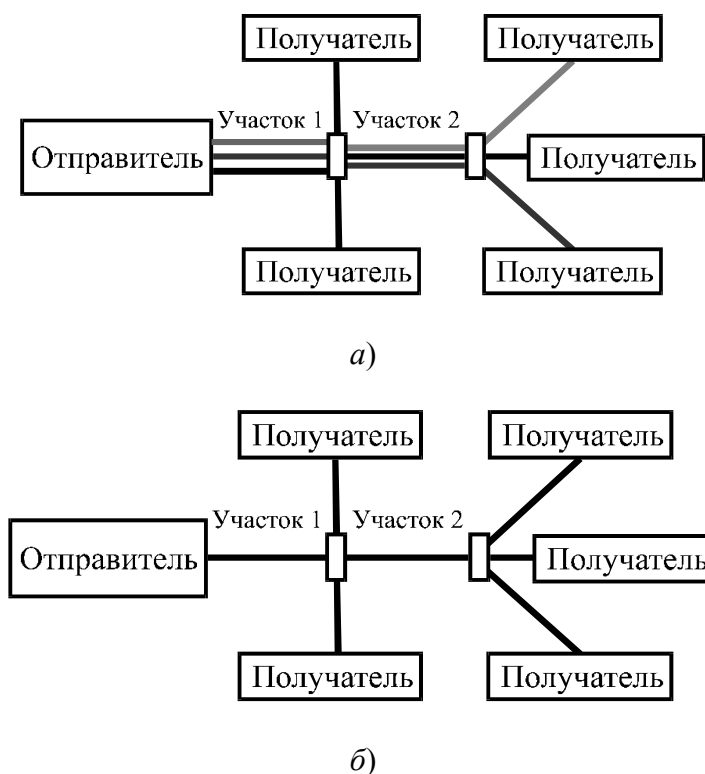


Рис. 1. Схемы одноадресной (а) и многоадресной (б) передачи данных

В режиме многоадресной передачи данных (рис. 1, б) сервер формирует один поток данных, к которому могут подключаться по сети различные группы клиентов. Здесь мощность сервера и полоса пропускания канала не зависят от количества получателей информации. Многоадресная передача удобна, например, для Интернет-телевидения и видеоконференций, поскольку каждый пользователь имеет возможность подключения к источнику информации. Однако это преимущество достигается ценой отсутствия подтверждения о доставке информации и ручной настройки каждого маршрутизатора или коммутатора на всем сетевом пути от сервера-отправителя до группы получателей информации. Технология многоадресной передачи данных применяется для новостийных и обучающих программ в корпоративных сетях. Для российского сегмента Интернет, где ощущается дефицит высокоскоростных соединений сетевых узлов, сдерживающий широкое использование вещательных технологий, применение многоадресной передачи данных особенно актуально.

Обычные маршрутизаторы не обеспечивают контроля переполнения или превышения пропускной способности канала связи и, следовательно, гарантированного качества передачи. Другая проблема состоит в том, что при передаче непрерывной видеoinформации каждый информационный пакет должен вовремя достигать получателя, быть декодирован и отображен без пауз. Задержки передачи пакетов нарушают непрерывность отображения, что неприятно для визуального восприятия, а в ряде случаев является неприемлемым. В настоящее время сеть Интернет не

обеспечивает гарантированной доставки информационных пакетов в реальном времени и без потерь. Потери пакетов считаются в среднем допустимыми, если их относительный уровень составляет не более долей процента. В ситуациях превышения пропускной способности потери информации могут быть гораздо большими, и качество передачи резко снижается. Неоднородность структуры сети и характеристик отдельных систем передачи и приема приводят к изменчивости значений задержек и потерь, которые в результате не могут быть скомпенсированы. Этот недостаток особенно характерен для многоадресной передачи видеoinформации.

При решении указанных проблем могут быть использованы два основных подхода. Первый из них состоит в обеспечении требуемого QoS за счет поддержки всех необходимых параметров сети при глобальном усовершенствовании сетевого оборудования. Второй подход основан на использовании на стороне приема видеoinформации таких систем, которые обеспечивали бы приемлемое качество при малой зависимости от характеристик сети. Этот подход рассматривается как более предпочтительный, поскольку не требует кардинального изменения параметров сети.

Повышение качества передачи видеoinформации в компьютерных сетях может быть достигнуто в двух направлениях: либо за счет совершенствования методов управления, обработки и передачи безотносительно к характеру видеoinформации, либо на основе использования методов сжатия, учитывающих семантику изображений на различных уровнях компрессии.

Управление параметрами передачи видеoinформации

Управление параметрами сети осуществляется тремя основными способами: варьированием скорости передачи, адаптивным кодированием видеoinформации, характеристики которого определяются скоростью передачи, и локальным ускорением передачи на коротких интервалах за счет повышения степени сжатия. Первый способ реализуется на уровне передачи информации, второй основывается на специальных методах сжатия видеoinформации, третий включает в себя управление передачей и сжатием видеoinформации.

В качестве примера на рис. 2 представлена структурная схема управления, включающая перечисленные процедуры.

Система со структурой рис. 2 ориентирована на передачу видеoinформации в реальном времени, однако она может быть использована для стационарного потока видеоданных в случае отключения блоков адаптивного кодирования в зависимости от скорости передачи. В схеме реализуется управление скоростью передачи на стороне источника информации (отправителя). В динамическом режиме передачи после адаптивного сжатия поток видеоданных обрабатывается в блоке локального ускорения передачи и далее преобразуется на уровнях RTP/UDP/IP перед поступлением в сеть.

Транспортный протокол RTP (Real-Time Transport Protocol) обеспечивает передачу данных между конечными узлами сети, работает поверх протокола IP и является альтернативой широко используемому протоколу TCP [1]. Хотя протокол TCP и гарантирует доставку пакетов данных в нужной последовательности, трафик при этом очень неравномерен (пакеты испытывают всевозможные задержки). Обладая способностью распознавать содержимое пакетов (например, различать видеоданные, соответствующие спецификациям MPEG и H.261), а также средствами обнаружения потери данных, протокол RTP снижает задержки до уровня, необходимого для успешной передачи потоков интегрированных аудио-, видео- и цифровых данных. Протокол RTP обеспечивает идентификацию типа и номера пакета, устанавливает в него метку синхронизации. На основе этой информации приемный терминал синхронизирует данные и осуществляет их последовательное и непрерывное

воспроизведение. Корректное функционирование RTP возможно при наличии в абонентских терминалах механизмов буферизации принимаемой информации.

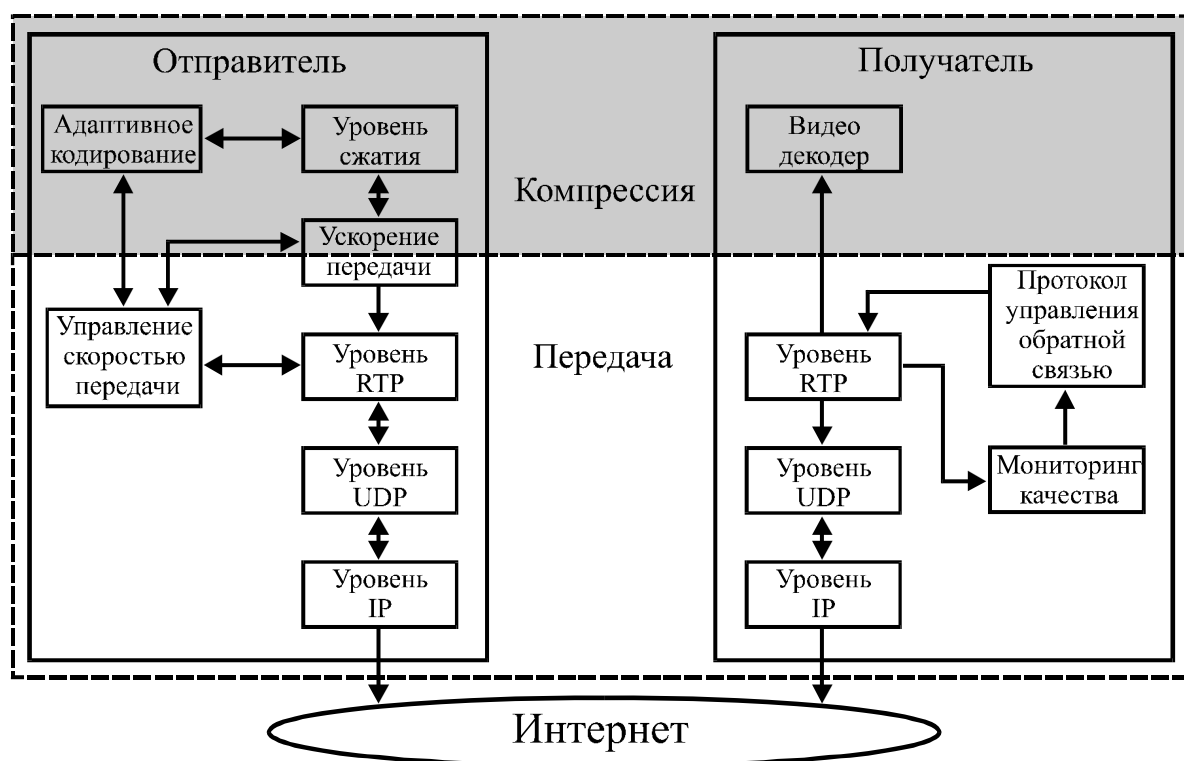


Рис. 2. Иерархия системы передачи видеoinформации в реальном времени

Протокол UDP (User Datagram Protocol) используется в сетях с негарантированной полосой пропускания с целью минимизации задержек и максимального использования имеющейся полосы пропускания для передачи видеопотоков [2]. Этот протокол реализует упоминавшийся выше механизм многоадресной рассылки (IP Multicast) для негарантированной доставки аудио- и видеoinформации определенному числу пользователей. Поверх IP Multicast работает RTP, который создает необходимые условия для нормального воспроизведения полученных потоков данных на абонентских терминалах. Однако протокол UDP не обеспечивает контроля превышения пропускной способности канала, поэтому такой контроль должен быть обеспечен на более высоких уровнях в схеме рис. 2.

На стороне получателя информационные пакеты преобразуются на уровнях IP/UDP/RTP перед их декодированием. В схеме рис. 2 предусмотрен динамический мониторинг показателя QoS, основывающийся на свойствах полученных пакетов, а именно на контроле потерь и задержек при их получении. Используя эту информацию, блок управления обратной связью обеспечивает оценку требуемой ширины полосы для выбора подходящего режима адаптивного кодирования и локального ускорения скорости передачи за счет локального повышения степени сжатия.

Управление скоростью передачи видеoinформации

Для успешной передачи видеoinформации можно использовать два подхода, а именно, учитывать (подстраивать) характеристики сети передачи либо обеспечивать управление степенью сжатия видеoinформации.

Методы управления, предназначенные для предотвращения превышения пропускной способности сети при передаче видеoinформации, можно подразделить на два основных вида: управление временным окном [3] и непосредственное управление скоростью передачи [4]. В первом случае (которому соответствует, например, протокол

ТСР) имеющаяся ширина полосы проверяется за счет медленного увеличения продолжительности временного интервала, который соответствует ситуации превышения пропускной способности сети, чтобы определить, каковы при этом потери данных. При обнаружении потери пакетов данных длительность временного окна резко уменьшается, что предотвращает сбой передачи. Недостатком метода являются возможные задержки в передаче информации, поскольку его осуществление связано с повторной передачей информационных пакетов.

При непосредственном управлении скоростью передачи используются сведения об имеющейся ширине полосы, в соответствии с которой и устанавливается скорость передачи. Если оценка ширины полосы достаточно точна, то можно эффективно предотвращать сбой в передаче данных, поэтому такой метод предпочтителен при передаче видеоинформации в реальном времени.

Существующие методы управления скоростью передачи в реальном времени подразделяются на методы управления характеристиками источника информации, управления на принимающей стороне и комбинированного управления. Соответственно этому производится согласование исходного потока видеоинформации с имеющейся шириной полосы, регулируется скорость приема видеоинформации за счет добавления (отключения) каналов связи, либо используется комбинирование обоих методов.

Метод управления характеристиками источника видеоинформации может использоваться при одноадресной [2] и многоадресной [5] передаче данных (рис. 1). При одноадресной передаче устанавливается такая скорость передачи, чтобы достигнуть заданного значения показателя QoS, т.е. чтобы относительные потери пакетов p не превышали порогового уровня P_0 . Этот уровень определяется допустимой заметностью искажений видеоинформации. Возможный нелинейный алгоритм управления определяется следующими выражениями [6]:

$$\begin{aligned} & \text{if } (p \leq P_0) \\ & \quad r = \min \{ (r + \Delta r), \max r \} \\ & \text{else} \\ & \quad r = \max \{ \alpha r, \min r \}, \end{aligned}$$

где r – скорость передачи источника видеоинформации, Δr – аддитивное увеличение скорости, α – коэффициент мультипликативного уменьшения скорости. Скорость передачи можно установить также расчетным путем [7] при известных параметрах соединения ТСР, максимального объема информационного пакета, временного интервала соединения в сети и значения коэффициента p .

В случае управления источником при многоадресной передаче отправитель может использовать один канал или одну многоадресную IP-группу для передачи потока видеоданных к получателям информации. При этом осуществляется одноканальная многоадресная передача на основе представленного выше нелинейного алгоритма управления и достигается высокая эффективность использования ширины полосы, однако в ущерб гибкости в обслуживании различных получателей информации. С другой стороны, многоадресная передача на основе использования одноадресных потоков данных обеспечивает высокую гибкость ценой неэффективности использования ширины полосы. Такие предельные ситуации иллюстрируют необходимость в дополнительных методах управления на стороне получателей информации.

Одним из возможных подходов является "многослойная" многоадресная передача при различном качестве в зависимости от доступной ширины полосы. Основная часть видеоинформации передается на нижнем уровне пропускной способности канала связи и обеспечивает некоторое приемлемое качество приема. При повышении пропускной

способности подключаются более высокие уровни, обеспечивающие улучшенное качество приема. Управление уровнями осуществляется на основе общих подходов, аналогичных рассмотренным выше применительно к управлению источниками видеoinформации.

Рассмотрим методы управления второго вида, а именно, управления степенью сжатия видеoinформации.

Адаптивное кодирование видеoinформации с варьируемой степенью сжатия

Возможности кодирования видеoinформации с адаптацией к скорости передачи исследовались для различных стандартов и приложений, в частности, для стандартов H.261 и H.263 [8, 9], MPEG-1/2 [10, 11] и MPEG-4 [2, 12]. При этом задача заключается в обеспечении высокого качества визуального восприятия видеoinформации при заданной скорости передачи, которая может быть фиксированной, либо динамически изменяться в зависимости от пропускной способности сети. Реализация адаптивного кодирования достигается изменением параметра дискретизации видеoinформационных сигналов и/или изменения скорости передачи видеокадров.

В традиционных кодерах видеoinформации, например, в стандартах H.261 и MPEG-1/2, используется изменение параметра дискретизации при постоянной скорости передачи видеокадров. Это вызывает резкое снижение качества, особенно для динамических сцен, даже при незначительном снижении скорости передачи. В стандартах H.263 и MPEG-4 предусмотрена возможность обеспечения достаточно высокого качества при изменениях скорости передачи видеокадров за счет пропуска отдельных видеокадров, которые не могут быть декодированы за допустимое время. При этом кодовые последовательности предыдущих видеокадров передаются в течение временного интервала пропущенного кадра, что предотвращает переполнение буфера кодера. В стандарте MPEG-4 осуществляется сегментация видеокадров на так называемые видеообъекты, т.е. части изображения, которые кодируются отдельно. Это обеспечивает высокую гибкость адаптивного кодирования различных видеообъектов с различным уровнем сжатия видеoinформации в зависимости от степени важности видеообъекта для получателя видеoinформации. Например, видеoinформация в виде интересующего объекта на неизменном фоне передается с более высоким качеством по сравнению с фоном.

Важной проблемой является определение количественного критерия качества при заданной скорости передачи видеoinформации. Зависимость степени искажений от скорости передачи может быть получена на основе модельного подхода, при котором учитываются статистические характеристики видеопотока для определенных параметров дискретизации видеоданных. При этом решение задачи обеспечения требуемого качества передачи основывается на математических методах теории оптимизации [2, 13]. С другой стороны, возможна реализация методов адаптации с учетом конкретных характеристик оборудования, когда стратегия минимизации искажений строится в соответствии с имеющейся пропускной способностью сети. Оптимальные решения задачи могут быть получены методами теории целочисленного программирования [9, 11, 14, 15].

Искажения видеoinформации возникают также вследствие ошибок при передаче/приеме с использованием реальных каналов связи. Для повышения качества получаемой видеoinформации используются различные методы корректирования ошибок, рассматриваемые в следующем разделе.

Методы корректирования ошибок передачи видеoinформации

Для некоторых типов видеоданных (например, для текста) потери пакетов данных недопустимы, хотя задержки в получении видеоинформации являются приемлемыми. Для других типов данных, содержащих динамические фрагменты изображения, некоторые потери данных являются приемлемыми, тогда как задержки получения видеоинформации недопустимы. В случае потери пакета данных восстановление информации может быть обеспечено традиционными методами корректирующего (канального) кодирования, либо при повторной передаче потерянных данных.

Особенности передачи видеоданных в реальном времени требуют использования ряда специальных методов контроля ошибок передачи, которые не применялись при передаче обычных данных. К таким методам, помимо корректирующего кодирования и повторной передачи, можно отнести снижение чувствительности (критичности) к ошибкам, т.е. видоизменение ошибок, и маскирование ошибок. Первые три упомянутых метода могут быть реализованы на передающей и приемной стороне, тогда как маскирование ошибок осуществляется при приеме видеоинформации.

На рис. 3 показана схема, иллюстрирующая взаимодействие различных уровней контроля ошибок. Повторная передача позволяет восстанавливать утраченные пакеты при их транспортировке через сеть, тогда как видоизменение ошибок при снижении чувствительности к ним и маскирование ошибок обеспечиваются обычно за счет адаптации степени сжатия видеоинформации.

Использование корректирующего кодирования имеет преимущество в малой задержке передачи. Этот метод основывается на добавлении дополнительной информации к сжатому потоку видеоданных, что позволяет реконструировать потерянные пакеты данных. При канальном корректирующем кодировании видеопоток разделяется на сегменты данных, каждый сегмент упаковывается в k пакетов и с помощью обычного блочного кодирования генерируется блок из n пакетов, $n > k$. При этом канальный кодер размещает k пакетов в группу и создает из них дополнительные пакеты, что повышает устойчивость к потере пакетов. В случае приема не менее k пакетов из переданных n можно полностью восстановить переданную видеоинформацию. Поскольку восстановление осуществляется на принимающей стороне, канальное кодирование реализуется для произвольного количества получателей при многоадресной передаче видеоинформации. Вследствие избыточности $\Delta k = n - k$ пакетов некоторые из них могут отбрасываться при ограничении пропускной способности канала без потери качества передачи видеоинформации. Это особенно важно для сетей передачи с неоднородной структурой и различными параметрами.

Недостатками канального кодирования являются повышенные требования к скорости передачи информации в n/k раз и увеличение задержек при передаче, поскольку декодер должен получить k пакетов прежде, чем будут выделены Δk избыточных пакетов, после чего возможно отображение принятой видеоинформации. Кроме этого, канальное кодирование является наиболее эффективным при неизменных характеристиках потерь информации, соответствующих заданной степени избыточности.

В настоящее время разрабатываются методы кодирования с приоритетом, когда пользователь получает возможность устанавливать различную степень коррекции ошибок для различных сегментов видеопотока. Такой подход является удобным, в частности, для распространенного стандарта MPEG, в котором I -кадр имеет приоритет по сравнению с P -кадром, который, в свою очередь, является более важным по сравнению с B -кадром [15].

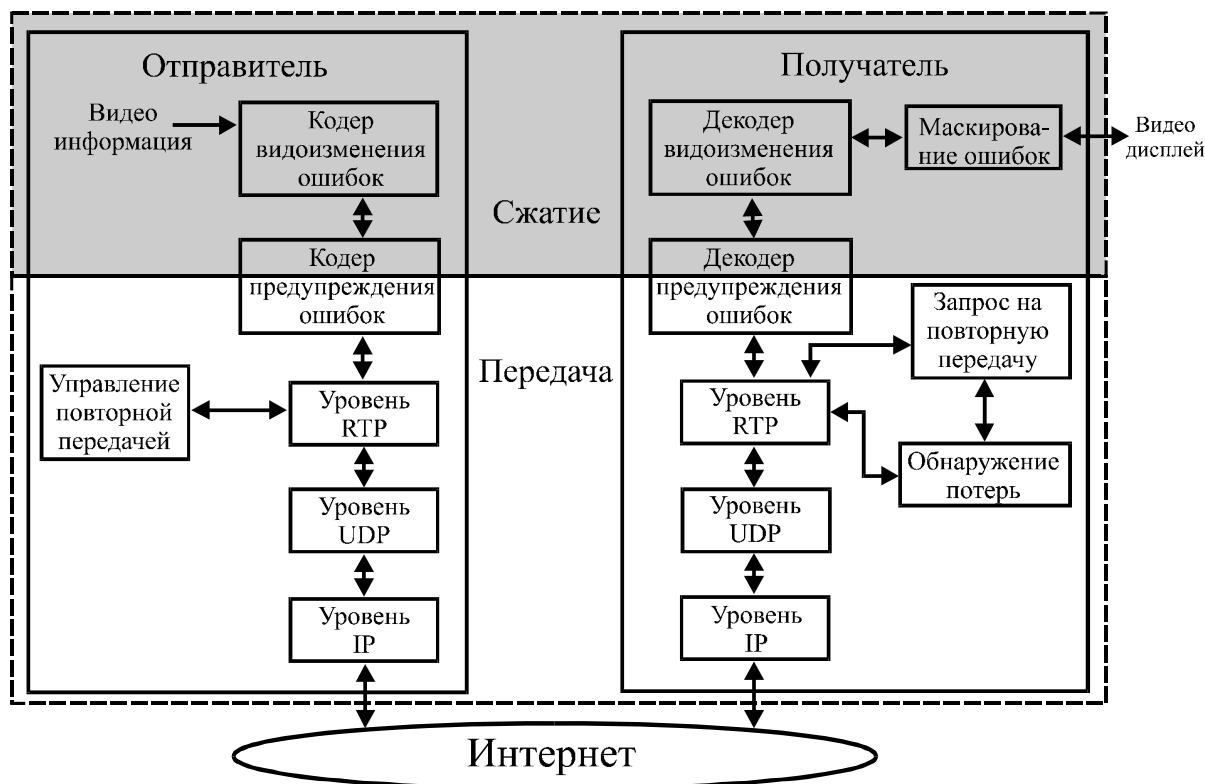


Рис. 3. Схема контроля ошибок при передаче/приеме видеоинформации

Наряду с методами канального кодирования используются методы кодирования источника видеоинформации. В отличие от канального кодирования, при котором избыточность вводится блочными кодами безотносительно к виду информации, кодирование источника осуществляется за счет добавления в данные видеоинформации с более высокой степенью сжатия. При этом в случае потери пакетов происходит восстановление видеоинформации с некоторой (допустимой) потерей качества. Преимуществом кодирования источника является снижение задержек по сравнению с канальным кодированием.

В схеме рис. 3 предусмотрен режим повторной передачи, при котором в случае потери пакетов осуществляется режим обратной связи с источником в форме запроса на повторную передачу [16]. Недостатком повторной передачи является примерно трехкратное увеличение времени передачи информации по сети (исходная передача – запрос на повторную передачу – повторная передача). Поэтому повторная передача наиболее эффективна в случае малого времени доставки видеоинформации (например, в локальных сетях) или при наличии информационного буфера большого объема на принимающей стороне. Управление режимом повторной передачи обычно предусматривает отмену повторной передачи тех пакетов, которые не могут быть отображены в течение имеющегося временного интервала. Такое управление может осуществляться как на передающей, так и на приемной стороне [6]. В случае многоадресной передачи ограничение повторной передачи должно устанавливаться только для пользователей с большим временем доставки видеоинформации [17–19].

Снижение заметности ошибок передачи видеоинформации является наиболее эффективным на уровне сжатия информации, при этом оказывается возможным предотвращение распространения ошибок и ограничение степени искажений видеоинформации при потере информационных пакетов. Это достигается за счет "многослойной" структуры представления видеоинформации с различной степенью сжатия в каждом слое, что обеспечивает восстановление видеоинформации приемлемого качества даже при потерях части информации для отдельного слоя.

Подобные методы рассмотрены в [20] с точки зрения оптимизации степени сжатия и устойчивости к потерям информационных пакетов.

В случае потерь информации заметность искажений может быть снижена заменой потерянного полного видеокadra на предыдущий, а также заменой отдельного искаженного блока изображения на блок из предыдущего кадра либо на блок с экстаполяцией данных на основе динамики изменения изображения.

Реализация методов передачи видеoinформации в телекоммуникационных сетях

При проектировании сетей, предназначенных для работы с видеоданными, необходимо особенно тщательно выбирать методы сжатия информации и аппаратные средства применительно к типу информационной сети.

Первое узкое место сети – это серверы. Способность большинства серверов передавать видеоизображения в реальном времени ограничивается главным образом скоростью обмена с дисковыми накопителями. Видеосигнал представляет собой непрерывный поток данных, и накопитель, считывающий информацию отдельными фрагментами, как правило, не способен поддерживать такой поток. Скорость поступления данных в центральный процессор и, соответственно, непрерывность потока видеоданных определяются средней установившейся скоростью считывания с диска, т.е. той, с которой производится последовательное считывание файлов. На пропускную способность отрицательно влияют также задержки, вызванные необходимостью коррекции ошибок считывания. Это препятствие можно обойти, отказавшись от отдельных пакетов данных, что мало сказывается на качестве видеоизображения. Другой вариант – направить видеотрафик в обход центрального процессора. Передача стандартного видеосигнала не требует вычислений, и пропускная способность сервера может быть увеличена за счет непосредственной пересылки данных с диска на устройство ввода/вывода.

Еще одна существенная задача при работе с видеоданными состоит в обеспечении требуемого объема и быстродействия внешних запоминающих устройств (ВЗУ). Как правило, в вычислительных системах применяются первичные (где хранятся часто используемые данные) и вторичные (служашие для резервного копирования и хранения редко используемых файлов) ВЗУ. Чтобы обеспечить максимальную пропускную способность локальных сетей, в качестве первичных ВЗУ применяются накопители с интерфейсом SCSI (Small Computer Systems Interface). Они рассчитаны на работу с 8-разрядной или 16-разрядной шиной SCSI (что теоретически обеспечивает вдвое большую скорость обмена). Имеется несколько типов таких накопителей: SCSI-1 и SCSI-2 имеют скорость обмена 5 Мбайт/с, Fast SCSI-2 – 10 Мбайт/с, Fast Wide SCSI-2 – 20 Мбайт/с и, наконец, Fast-20 SCSI – 40 Мбайт/с. Накопители SCSI, способные хранить значительные (измеряемые гигабайтами) объемы информации, производятся в больших количествах. К одному контроллеру может быть подключено до семи 8-разрядных и до пятнадцати 16-разрядных накопителей (контроллер SCSI представляет собой мост между шиной SCSI и шиной сервера, такой как PCI, VME, EISA, ISA и др.).

Еще одна возможная технология хранения данных основана на использовании матриц дисковых накопителей с избыточностью (RAID), в которых несколько дисков рассматриваются как единое целое. Применение накопителей типа RAID позволяет кардинальным образом улучшить показатели запоминающего устройства, увеличивает степень защищенности от ошибок и допускает наращивание емкости накопителя путем добавления новых дисководов. Передача видеоданных по локальной сети требует реализации технологии сжатия данных для сокращения объема передаваемой и хранимой информации. В частности, при работе с видеоданными обычно уменьшают разрешение (количество элементов изображения на экране) и частоту следования кадров.

При видеомонтаже наиболее популярен метод Motion JPEG, несмотря на то, что он не позволяет работать с аудиосигналом. Первоначально эта технология была разработана для сжатия неподвижных изображений, но, поскольку динамическое видеоизображение по сути представляет собой последовательность неподвижных изображений, следующих одно за другим с частотой 30 кадров/с, упомянутая технология может применяться для преобразования в цифровую форму отдельных кадров, последовательное воспроизведение которых имитирует полномасштабное видеоизображение.

Стандарты MPEG-1 и MPEG-2 предполагают сжатие данных с использованием программных или аппаратных средств. В методе MPEG избыточность информации, содержащейся в кадрах, устраняется путем дифференциального кодирования в последовательности видеок кадров. В результате каждый отдельный кадр не хранится в виде самостоятельного элемента данных, и для восстановления одного кадра необходимо иметь только информацию об отличиях предыдущих кадров. Поскольку при использовании метода MPEG эффективность сжатия значительно выше, чем при применении JPEG, требования к пропускной способности канала оказываются намного мягче. С другой стороны, сжатие методом MPEG обычно сопряжено с большей потерей информации, что влияет на качество изображения.

Стандарт MPEG-1 используется при скорости передачи от 1 до 1,5 Мбит/с и позволяет получать степень сжатия до 200:1, но удовлетворительное качество изображения сохраняется при соотношении не более 100:1. Обеспечивается разрешение 352×240 точек в стандарте NTSC при частоте 30 кадров в секунду, что сравнимо с качеством изображения VHS, получаемого с помощью бытовых видеосистем. Стандарт MPEG-1 находит применение в основном при передаче мультипликационных изображений и обычных видеоизображений. MPEG-2 поддерживает качество изображения на уровне вещательного стандарта и требует пропускной способности канала передачи данных 7–8 Мбит/с.

Технология Wavelet, как и JPEG, предусматривает раздельное кодирование данных для разных кадров. При этом, в отличие от JPEG, видеок кадр при кодировании не разбивается на отдельные блоки пикселей. В результате дефекты, обычно возникающие при использовании этого метода, отличаются от дефектов, которые иногда появляются при применении метода JPEG с большой степенью сжатия. Недостатком технологии Wavelet является заметное снижение качества изображения по краям объектов.

Технологии Intel Indeo и TrueMotion S реализуют программные методы кодирования и декодирования данных, позволяющие воспроизводить полноэкранное видеоизображение с частотой 30 кадров/с на стандартных ПК. Сжатие методом Indeo требует относительно небольших по сравнению с другими методами затрат. Качество изображения получается не очень высоким, но, если это не существенно, экономия средств обычно оправдывает эту потерю. Метод TrueMotion S обеспечивает более высокое качество изображений.

Метод Apple Video прекрасно подходит для компьютерной мультипликации, а для обычных видеоизображений – не очень. Radius Cinepak (совместная разработка Apple и SuperMac Technology) представляет собой программный кодек, используемый в основном в мультимедиа-продуктах на CD-ROM. Он очень медленно работает при сжатии данных, но обеспечивает значительную степень сжатия, высокую скорость воспроизведения и прекрасное качество изображения.

Декодирование и воспроизведение видеоданных осуществляется программными или аппаратными средствами. Программные методы все еще значительно медленнее, чем аппаратные, позволяющие декодировать видеосигнал в реальном времени, но

использование методов TrueMotion S и MPEG-1 можно считать шагом в правильном направлении.

Для повышения производительности локальных сетей при передаче видеоданных обычно применяют управление потоками данных на основе двух протоколов. Первый – MTP (Media Transport Protocol) фирмы Starlight Networks – обеспечивает необходимую для передачи видеоданных пропускную способность сети с помощью программирования путем присвоения видеоданным высокого приоритета. Второй подход разработан фирмой Novell для мультимедиа-систем и предусматривает отбрасывание отдельных кадров при перегрузке сети.

Не менее существенным препятствием при передаче мультимедиа-данных в локальных сетях без приоритета узлов или с архитектурой клиент/сервер является сама сеть. В настоящее время не решен окончательно вопрос, какой тип линий связи наилучшим образом подходит для передачи видеоинформации. Принято считать, что наиболее перспективными высокоскоростными сетями являются сети АТМ (Asynchronous Transfer Mode). АТМ с его масштабируемой пропускной способностью и гарантией качества услуг представляет собой идеальную инфраструктуру для поддержки мультимедийных приложений с жесткими требованиями к пропускной способности и задержке. Однако сегодня многие компании не испытывают потребности или не располагают достаточными средствами для развертывания АТМ либо другой высокоскоростной технологии для магистралей. Сети АТМ обеспечивают скорость передачи данных до 622 Мбит/с. Сеть АТМ – самое дорогостоящее, но и самое эффективное решение проблемы передачи мультимедиа-данных по сети.

Для увеличения пропускной способности локальных и глобальных сетей разработан ряд технологий передачи информации. Технология коммутируемой Ethernet, вытесняющая традиционную технологию Ethernet с разделяемым доступом, на сегодняшний день реализуется в двух вариантах – 10 Мбит/с и 100 Мбит/с. В последнее время значительное внимание привлекла к себе еще одна разновидность Ethernet – Gigabit (1 Гбит/с).

Технология Fast Ethernet является эволюционным развитием классической технологии Ethernet. Ее основными достоинствами являются: увеличение пропускной способности сегментов сети до 100 Мб/с, сохранение метода случайного доступа Ethernet, сохранение звездообразной топологии сетей и поддержка традиционных сред передачи данных с использованием витой пары и оптоволоконного кабеля.

Указанные свойства позволяют осуществлять постепенный переход от сетей 10Base-T – наиболее популярного на сегодняшний день варианта Ethernet – к скоростным сетям, сохраняющим значительную преемственность с хорошо освоенной технологией. При этом Fast Ethernet не требует коренного переобучения персонала и замены оборудования во всех узлах сети.

Технология Gigabit Ethernet является перспективной с точки зрения простоты и экономичности для высокоскоростной передачи информации при сохранении достоинств Ethernet, но не освобождает от недостатков последней. Это препятствует использованию Gigabit Ethernet для передачи потоков данных в реальном времени в силу непредсказуемости задержек и отсутствия поддержки требуемых классов обслуживания.

Соединение коммутаторов Fast Ethernet по Gigabit Ethernet позволяет резко повысить пропускную способность магистрали локальной сети и в результате поддерживать большее число как коммутируемых, так и разделяемых сегментов Fast Ethernet. Установка сетевой платы Gigabit Ethernet на сервер дает возможность повысить характеристики канала с сервером и тем самым увеличить производительность пользователей мощных рабочих станций, в том числе при передаче и обработке видеоинформации.

В системах, где требуется передавать видеоизображения в реальном времени с качеством, соответствующим вещательному стандарту, наилучшим решением был бы канал связи с высокой пропускной способностью. Поскольку реальная скорость трафика не может превосходить пропускную способность используемого участка сети, сбои в передаче видеоизображения с качеством на уровне телевизионного возникают даже в выделенных линиях АТМ. В то же время узкополосный сигнал, получаемый при сжатии видеоданных методом MPEG-1, может передаваться с очень малой задержкой по сети Ethernet. Таким образом, выбор оптимальных решений должен осуществляться на основе использования различных технологий и методов, рассмотренных в статье.

Заключение

Методы передачи видеоинформации основываются на положениях теории информации, теории связи и теории обработки изображений, которые активно разрабатывались в течение нескольких последних десятилетий. Успехи современных высоких технологий позволили реализовать и оптимизировать различные методы и алгоритмы преобразования и обработки изображений применительно к телекоммуникационным сетям.

Технологии Интернет обеспечивают широкую доступность мультимедиа-информации, однако при этом требуется решать ряд проблем, обусловленных особенностями передачи видеоинформации в реальном времени. Использование новых возможностей информационных технологий становится более эффективным при оптимизации структуры информационных сетей, отдельных узлов и их взаимодействия, широкого использования методов управления параметрами сети и режимов адаптации, рассмотренных в настоящей работе. В наиболее совершенных системах критические характеристики распределяются согласованно в структурной части, характеристиках оборудования и программном обеспечении, что повышает качество и надежность работы и расширяет функциональные возможности систем передачи видеоинформации.

Литература

1. Shulzrinne H., Casner S., Frederick R. and Jacobson V. RTP: a transport protocol for real-time applications //RFC 1889. Internet Engineering Task Force. 1996.
2. Wu D., Hou Y.T., Zhu W., et al. On end-to-end architecture for transporting MPEG-4 video over the Internet //IEEE Trans. Circ. Syst. Video Technol. 2000. V.10. P.923-941.
3. Jacobson V. Congestion avoidance and control //Proc. ACM SIGCOMM'88. 1988. P. 314-329.
4. Tirletti T., Huitema C. Videoconferencing on the Internet // IEEE/ACM Trans. Netw. 1996. V.4. P. 340-351.
5. Bolot J.-C., Tirletti T. and Wakeman I. Scalable feedback control for multicast video distribution in the Internet // Proc. ACM SIGCOMM'94. 1994. P.58-67.
6. Wu D., Hou Y.T. and Zhang Y.-Q. Transporting real-time video over the Internet: challenges and approaches // IEEE Proc. 2000. V.88. P.1855-1875.
7. Floyd S., Fall K. Promoting the use of end-to-end congestion control in the Internet // IEEE/ACM proc. Netw. 1999. V.7. P.458-472.
8. Martins F.C., Ding W. and Feig E. Joint control of spatial quantization and temporal sampling for very low bit-rate video //Proc. IEEE Int. Conf. ICASSP'96. 1996. V.4. P.1062-1072.
9. Weigand T., Lightstone M. and Mukherjee D., et al. Rate-distortion optimized mode selection for very low bit-rate video coding and the emerging H.263 standard //IEEE Trans. Circ. Syst. Video Technol. 1996. V.6. P.182-190.

10. Ding W. Joint encoder and channel rate control of VBR video over ATM networks // IEEE Trans. Circ. Syst. Video Technol. 1997. V.7. P.266-278.
11. Hsu C.Y., Ortega A. and Reibman A.R. Joint selection of source and channel rate for VBR transmission under ATM policing constraints //IEEE J. Select. Areas Commun. 1997. V.15. P. 1016-1028.
12. Vetro A., Sun H. and Wang Y. MPEG-4 rate control for multiple video objects //IEEE Trans. Circ. Syst. Video Technol. 1999. V.9. P.186-199.
13. Chiang T., Zhang Y.-Q. A new rate control scheme using quadratic rate distortion model //IEEE Trans. Circ. Syst. Video Technol. 1997. V.7. 246-250.
14. Lee J. and Dickenson B.W. Rate-distortion optimized frame type selection for MPEG encoding // IEEE Trans. Circ. Syst. Video Technol. 1997. V.7. P.501-510.
15. Sun H., Kwok W., Chien M. and Ju C.H.J. MPEG coding performance improvement by jointly optimizing coding mode decision and rate control //IEEE Trans. Circ. Syst. Video Technol. 1997. V.7. P.449-458.
16. Podolsky M., Vetterli M. and McCanne S. Limited retransmission of real-time layered multimedia //Proc. IEEE Workshop Multimedia Signal Processing. 1998. P.591-596.
17. Li X., Paul S. and Ammar M.H. Layered video multicast with retransmissions: evaluation of error recovery schemes //Proc. IEEE Int. Workshop NOSSDAV'97. 1997. P.161-172.
18. Maxemchuk N., Padmanabhan K. and Lo S. A cooperative packet recovery proptocol for multicast video //Proc. IEEE Int. Conf. ISNP'97. 1997. P.259-266.
19. Xu X.R., Myers R.C., Zhang H. and Yavatkar R. Resilent multicast support for continuous-media applications //Proc. IEEE Int. Workshop NOSSDAV'97. 1997. P.183-194.
20. Wang Y., Orchard M.T. and Reibman A.R. Multiple description image coding for noisy channels by pairing transform coefficients //Proc. IEEE Workshop Multimedia Signal Processing. 1997. P. 419-424.

АНАЛИЗ ВЛИЯНИЯ ХАРАКТЕРА ИНФОРМАЦИОННЫХ ПОТОКОВ НА КАЧЕСТВО ФУНКЦИОНИРОВАНИЯ ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ

Л.А. Муравьева-Витковская

Анализируется влияние непуассоновских информационных потоков на качество функционирование телекоммуникационной сети при распределении сетевых ресурсов в соответствии с заданными приоритетами. Определены области параметров нагрузки, при которых приближенные методы, разработанные в предположении о пуассоновском характере потоков, позволяют получить приемлемые для инженерной практики результаты.

Интенсивное развитие экономики связано с совершенствованием технологии и организации производства на базе широкого применения вычислительной техники и средств телекоммуникаций. Одной из форм повышения эффективности производства является использование компьютерных сетей для управления многопрофильными структурами, сложными энергетическими системами, гибкими производственными системами, телефонным и телеграфным оборудованием в системах связи, для автоматизации измерений и в других целях. Успешная работа многих организаций и компаний сегодня напрямую зависит от средств телекоммуникаций (СТК). Большую роль в деловой жизни стали играть Интернет и мультимедиа. В настоящее время из-за интенсивного роста числа пользователей и различных приложений в телекоммуникационных сетях существуют десятки разновидностей трафика [1]. Это обстоятельство необходимо учитывать администратору сети для повышения эффективности использования ресурсов сети. Одним из способов распределения сетевых ресурсов является распределение в соответствии с существующими на данный момент приоритетами.

Для успешного применения СТК необходимо располагать моделями и инженерными методами, позволяющими на основе данных измерений оценивать качество функционирования СТК, прогнозировать характеристики их работы при изменении технических и программных средств, способов диспетчеризации. Кроме того, важной задачей является разработка методов проектирования СТК с приоритетными способами диспетчеризации, обеспечивающими реализацию заданных функций при ограничении на время реакции.

В качестве моделей СТК целесообразно использовать модели массового обслуживания: системы (СМО) и сети (СеМО) массового обслуживания [2]. При этом часто предполагается, что потоки заявок, поступающие в систему, являются простейшими. В случае произвольных потоков расчет средних значений характеристик обслуживания заявок обычно проводится на основе аппроксимации закона распределения интервалов времени между заявками в потоке с учетом первых двух моментов. Такой подход особенно широко используется при разработке приближенных методов расчета неэкспоненциальных СеМО [3]. Однако, как показывают исследования, если для расчета характеристик функционирования системы на уровне средних значений, в частности, среднего времени ожидания заявок в очереди, достаточно задать только два момента длительности обслуживания, то при описании интервалов времени между заявками в потоке этого оказывается недостаточно. Другими словами, на средние значения характеристик обслуживания заявок оказывают существенное влияние моменты более высокого порядка.

В настоящей работе приводятся результаты исследований влияния третьего момента интервалов времени между заявками во входном потоке на среднее время ожидания в одноканальных СМО следующих классов:

- с однородным потоком заявок и экспоненциальной длительностью обслуживания в приборе;

- с однородным потоком заявок и длительностью обслуживания, распределенной по произвольному закону;
- с неоднородным потоком заявок.

Рассмотрим одноканальную СМО с однородным потоком заявок, интервалы времени между которыми распределены по произвольному закону с плотностью $a(\tau) = A'(\tau)$, где $A(\tau)$ – функция распределения интервалов. Пусть длительность обслуживания распределена по экспоненциальному закону. Тогда, согласно [4], функция распределения времени ожидания определяется как

$$W(\tau) = 1 - \sigma^{-\mu(1-\sigma)^\tau} \quad (\tau \geq 0),$$

а среднее время ожидания – как

$$w = \int_0^{\infty} \tau dW(\tau) = \sigma / \mu / (1 - \sigma),$$

где μ – интенсивность обслуживания заявок, а σ – единственный в области $0 \leq \sigma < 1$ корень уравнения $\sigma = A^*(\mu - \mu\sigma)$. Здесь $A^*(s)$ – преобразование Лапласа плотности распределения $a(\tau)$:

$$A^*(s) = \int_0^{\infty} e^{-s\tau} a(\tau) d\tau.$$

В качестве законов распределения $a(\tau)$ интервалов времени между заявками в потоке были выбраны:

- равномерный и Эрланга для потоков с коэффициентами вариации интервалов времени между заявками $v_a < 1$;
- различные формы представления гиперэкспоненциального закона для потоков с $v_a > 1$, различающиеся значениями третьих моментов.

Для описания третьего момента интервалов времени между заявками в потоке использовался коэффициент асимметрии $\gamma = \beta^{(3)} / [\beta^{(2)}]^{3/2}$, где $\beta^{(l)}$ – центральный момент l -го порядка ($l = 2, 3, \dots$):

$$\beta^{(l)} = \int_0^{\infty} (\tau - 1/\lambda)^l dA(\tau).$$

Здесь $1/\lambda$ – средний интервал времени между заявками.

В таблице представлены некоторые результаты проведенных исследований для потоков заявок с коэффициентами вариаций интервалов времени между заявками $v_a = 0,58; 2,0; 5,0$ при загрузке $\rho = \lambda/\mu$, изменяемой в пределах от 0,1 до 0,99.

Для одного и того же коэффициента вариации v_a в широких пределах изменялся коэффициент асимметрии γ , характеризующий третий момент интервалов времени между заявками в потоке. В таблице для наглядности приведены лишь крайние значения γ , при которых проводились исследования, и соответствующие им средние времена ожидания заявок, отнесенные к средней длительности обслуживания при различных значениях загрузки ρ системы.

Анализ полученных результатов свидетельствует о существенном влиянии третьего момента интервалов времени между заявками потока, задаваемого в виде коэффициента асимметрии γ , на среднее время ожидания заявок в системе, причем с увеличением коэффициента асимметрии среднее время ожидания заявок уменьшается. Эта зависимость особенно сильно проявляется при малых загрузках системы и уменьшается с ее увеличением. Так, при значениях загрузки $\rho = 0,1$ времена ожидания при разных γ различаются в несколько раз, а при $\rho = 0,99$ эта разница составляет несколько процентов. В области значений загрузки от 0,3 до 0,7, наиболее характерной для СТК, эта разница достаточно значительна и составляет десятки и сотни процентов,

причем растет с увеличением коэффициента вариации v_a интервалов времени между заявками в потоке.

Таблица

Среднее время ожидания заявок

v	γ	ρ				
		0,1	0,3	0,5	0,7	0,99
0,58	0	0,056	0,224	0,564	1,42	66,1
	300	0,013	0,152	0,494	1,36	65,8
2,0	-1,5	1,80	2,60	4,00	7,30	249
	110	0,113	0,438	1,03	2,46	199
5,0	-0,5	13,4	17,6	25,0	42,3	1299
	50	0,131	0,535	1,38	4,22	1219

Для СМО с произвольно распределенной длительностью обслуживания и с неоднородным потоком заявок исследования проводились на имитационных моделях. Полученные при этом результаты имеют аналогичный характер.

Таким образом, выполненные исследования позволили выявить существенное влияние третьего момента интервалов времени между поступающими в систему заявками на качество ее функционирования, которое особенно необходимо учитывать в тех случаях, когда система работает в области малых загрузок или характеризуется большими значениями коэффициента вариации интервалов времени между заявками, поступающими в систему. Полученные результаты можно использовать при администрировании телекоммуникационной сети для повышения эффективности функционирования при увеличении объема сетевого трафика, вызванного внедрением новых информационных технологий, использованием различных приложений: Интернет, IP-телефонии (VoIP), видеоконференц-связи, планирования ресурсов предприятия (ERP), управления взаимоотношениями с заказчиками (CRM) и др.

Литература

1. Якубович Д. Оптимизация сетевого трафика // Сети и системы связи. 2001. №10. С.92–97.
2. Алиев Т.И., Кругликов В.К., Муравьева Л.А. Анализ сетей передачи данных с неоднородными сообщениями // Изв. вузов СССР. Приборостроение. 1989. № 1. С.33–39.
3. Алиев Т.И., Муравьева Л.А. Расчет характеристик вычислительных систем на основе разомкнутых сетевых моделей с приоритетами // Архитектура и проектирование вычислительных систем. Автоматизация проектирования. Рига: Риж. политехн. ин-т, 1985. С.97–109.
4. Клейнрок Л. Вычислительные системы с очередями. М.: Мир, 1979. 600 с.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ НА ОСНОВЕ РАСШИРЕННОЙ ИЕРАРХИИ УРОВНЕЙ OSI-МОДЕЛИ

Н.А. Рубина, Ю.Г. Кирчин

Рассматриваются некоторые аспекты проблемы эффективности предоставления информационного сервиса на основе компьютерных и сетевых технологий. Предлагается расширение структурной иерархии OSI-модели введением дополнительного уровня, формализующего информационные потоки и структуры. Рассматривается взаимодействие этого дополнительного уровня с логическими компонентами поддерживаемого проекта, с одной стороны, и поддерживающих традиционных информационных технологий, с другой стороны. Предлагается сфокусировать внимание на требованиях к информационному обороту поддерживаемого проекта в противовес традиционному формированию требований к программному продукту, исследовать выделенную информационную задачу с точки зрения ее разложения по базису элементарных информационных задач, предложить заказчику в аренду динамически компонуемую информационную услугу. Рассматриваются достигнутые практические результаты.

Введение

Повышение эффективности использования информационных технологий позволяет улучшить деловую результативность различных видов деятельности и, в частности, обеспечить увеличение прибыли учреждений и предприятий, более точное выполнение ими своей роли [1]. Ключевой проблемой при этом является соответствие информационного обеспечения предъявляемым к нему требованиям. Специфика производства программного продукта и информационных услуг определяет сложность, а во многих случаях и невозможность прямого переноса опыта по управлению соответствием изделия требованиям заказчика, накопленного в традиционных областях, на область информационного производства.

Традиционные области деятельности, представленные на рис. 1 под этим общим названием, развившиеся и сформировавшие свои технологии, правила, схемы в докомпьютерную эпоху, в определенный момент потребовали резкого увеличения производительности в области информационного оборота, чем вызвали появление и последующий бурный рост информационных технологий.

С развитием и профессионализацией как традиционных областей деятельности, так и поддерживающих их информационных технологий наблюдается эффект их взросления¹, заключающийся в том, что новые решения базируются на ранее найденных как на очевидных [2]. Процесс этот многократно повторяется; все труднее становится редуцировать современные профессиональные решения до уровня здравого смысла, очевидного представителю другой профессии. Сложность однозначного взаимного отображения внутренних логик поддерживаемых традиционных областей деятельности и поддерживающих информационных технологий, логический разрыв между ними является критическим фактором эффективности применения информационных технологий [3–5].

Определенную сложность при обсуждении проблемы взаимодействия традиционных областей деятельности и поддерживающих их информационных технологий представляет отсутствие устоявшейся терминологии. Поскольку эта проблема занимает пограничную область, ни термины традиционных областей, ни термины информационных технологий не могут в полной мере обеспечить необходимый словарь. В английском языке наиболее часто употребляются следующие термины: *Business* (или *Real World*) – для обозначения поддерживаемой традиционной области деятельности, проекта; *The Gap* – для обозначения логического разрыва;

¹ Русский аналог емкого английского термина *maturity* – зрелость, описывающего уменьшение изменчивости области знаний с течением времени при увеличении числа решенных в этой области проблем.

Information Technology – для обозначения поддерживающих информационных технологий и *Bridging the Gap* (или *Navigating the Gap*) – для обозначения действий по преодолению логического разрыва. В пределах этой статьи мы будем использовать следующую терминологию, соответственно: аббревиатуру *RL* (Real Life) – для обозначения поддерживаемой области, *The Gap* в диаграммах и синонимично *разрыв* в тексте – для обозначения логического разрыва, аббревиатуру *IT* (Information Technology) – для обозначения поддерживающих технологий. Постоянно подчеркивая протяженный характер разрыва, в ряде случаев мы будем использовать более точное обозначение логического перехода между RL- и IT-областями, той черты, за которой непосредственное действие логики каждой из них прекращается. В этом случае мы будем использовать термин *граница*.

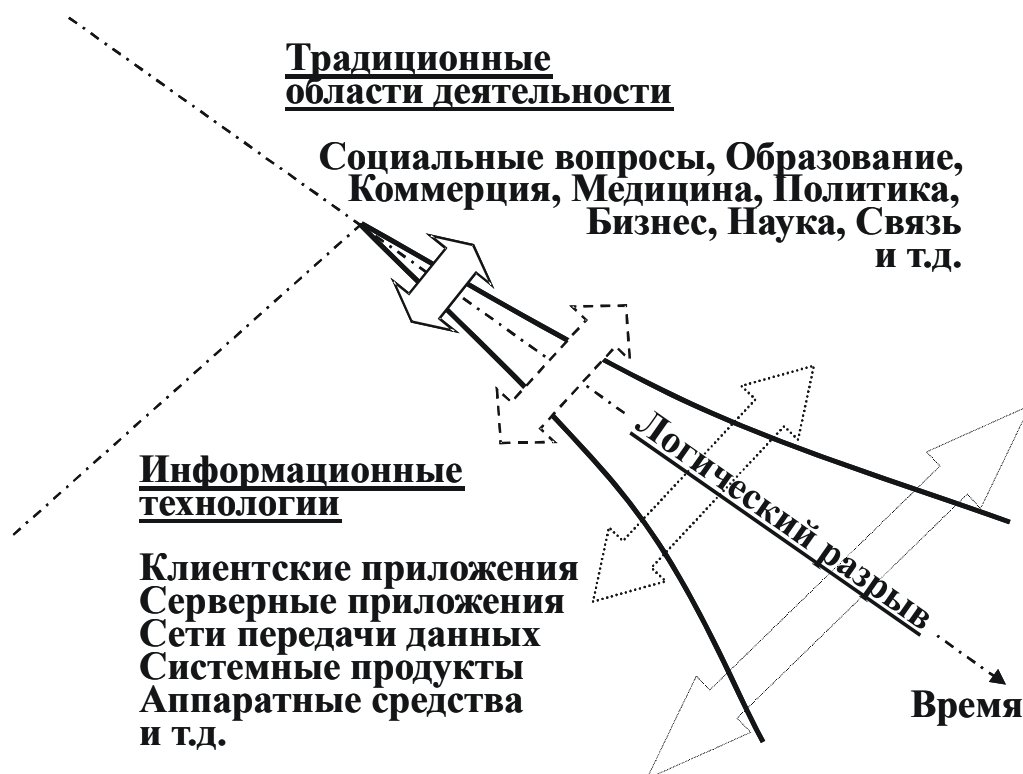


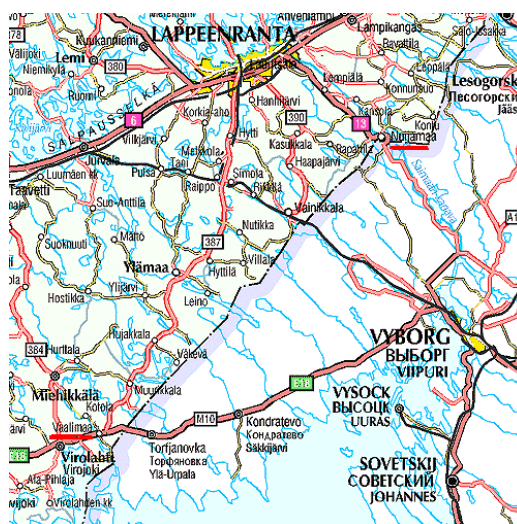
Рис. 1. Логический разрыв между традиционными областями деятельности и информационными технологиями увеличивается со временем, ослабляя их связь и возможность эффективного взаимодействия

Суммируя введенные обозначения, предложим простейшую модель области исследования:

RL | The Gap | IT.

Естественно предположить, что для преодоления разрыва необходимо исследовать примыкающие к нему логические слои RL- и IT-областей и предложить нечто, непрерывно их соединяющее.

Хорошей аналогией исследуемой проблемы является государственная граница двух стран. Подобно соседствующим странам, RL- и IT-области имеют разные законы, языки, степень освоения приграничных областей (рис. 2, а). Приграничные области соседствующих стран, именуемые пограничными зонами, имеют специальный режим и могут служить прекрасной иллюстрацией исследуемого нами разрыва. Представив RL- и IT-области как две различные страны, предстоит исследовать приграничные области и рационально организовать режим пересечения границы: проложить дороги и создать пункты пересечения границы (рис. 2, б).



а)

б)

Рис. 2. Государственная граница² является аналогом исследуемой области, а пограничные переходы – аналогом предлагаемого решения

1. RL-деятельность и логическое структурирование

Невозможно проанализировать все виды RL-деятельности на предметном уровне, во всем многообразии их специфик. С другой стороны, нашей задачей является обнаружение и исследование лишь их пограничных областей, т.е. рассмотрение лишь с точки зрения применения ИТ. Для этого будет достаточно сколь угодно абстрактная, но внутренне непротиворечивая модель, применимая и к бизнесу, и к образованию, и к медицине и т.д., адекватная с точки зрения решаемой задачи. Такую модель предлагает метод логического структурирования (Logical Frameworking Method) [6].

Суть метода (рис. 3) состоит в следующем:

- при разработке проекта определяется его общая цель (Overall Objective), возможно, весьма сложная и абстрактная;
- общая цель разбивается на ряд составляющих целей (Constituting Objectives), которые, совместно достигнутые, в совокупности дают достижение общей цели;
- составляющие цели разбиваются на ряды результатов (Outcomes), допускающих бинарную оценку "достигнуто / не достигнуто" и в совокупности дающих достижение каждой составляющей цели;
- для достижения каждого результата планируются ряды необходимых действий (Activities);
- для выполнения каждого действия определяются необходимые средства (Inputs);
- для исследования вероятностной компоненты поведения планируемого проекта вводится категория предположений и рисков (Assumptions and Risks).

Для краткого обозначения этого метода, его применения, предлагаемых моделей, а также объективного действия используемых причинно-следственных закономерностей будем использовать символ π^3 . Например, желая сказать "иерархическая логическая структура, направленная на достижение цели посредством

² Графический материал [http://www.tieh.fi]

³ Экзотический символ π (читается «тет», по первой, справа, букве слова תוכנית – план, программа) используется, чтобы возможно более наглядно, в том числе и графически, отделить понятие, заимствованное из сложной внешней области знания. Используемый далее подобно математическому оператору, он является скорее указателем на сложную группу операторов, большинство свойств которых не анализируется.

взаимодействия составляющих эту структуру объектов, число и природа которых определяется целью, взаимодействующих как непосредственно в пределах одного логического слоя, так и с объектами ближайшего логического слоя, причем межслойное взаимодействие носит характер обслуживания со стороны более удаленных от цели слоев, а объекты разных слоев не могут быть выражены друг через друга, находясь в причинно-следственной связи", будем писать " π -структура".

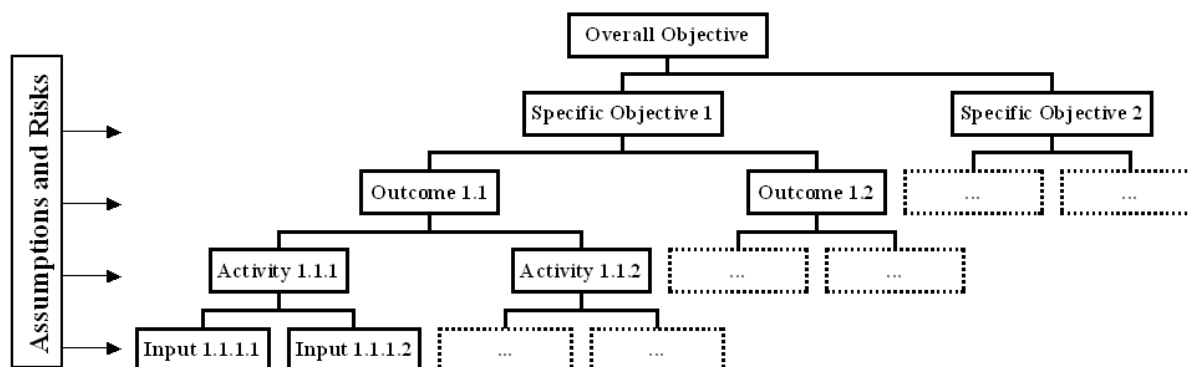


Рис. 3. Модель LFM: логическая декомпозиция планируемого проекта с точки зрения определения причинно-следственных связей и с учетом вероятностной компоненты

Как правило, допустимы множественные π -разложения общей цели, а следовательно, и множественные проектные π -структуры, ведущие к одной общей цели. Поэтому возможна π -оптимизация проекта по какому-либо критерию, как правило, по критериям минимизации средств, минимизации сроков достижения общей цели, или максимизации вероятности достижения общей цели.

Важно отметить, что одинаково действенны как нисходящий π -анализ, так и восходящая π^{-1} -реконструкция. В первом случае общая цель является заданным объектом и определению подлежит ведущая к ней π -структура, во втором случае скрытая общая цель оценивается по видимым элементам в общем случае скрытой π -структуры⁴.

Мы дополнили классический метод логического структурирования двумя понятиями, важными для решаемой задачи:

1. *ценности* (Values) – ограничивающие положения, находящиеся вне π -логики проекта, соблюдение которых обязательно при π -планировании и последующем выполнении проекта (например, законодательство, моральные принципы, традиции, внешние обязательства и т.п.) [7];
2. *каскадирование* π -структуры, т.е. возможность породить вторичную π' -структуру, общей целью которой будет промежуточный π -объект первичной π -структуры (например, выполнение бизнес-плана является общей целью первичной π -структуры предприятия и одновременно ценностью вторичной π' -структуры, скажем, цеха) [8].

Пользуясь модифицированным методом, мы предлагаем следующую формулу, ведущую к искомому решению:

- если из π -планирования проекта следует определяемая вторичной информационной π_i -структурой ($\pi_i \prec \pi$) необходимость информационного обмена, демонстрирующего π_i -определенные свойства, которые находятся в π -логике всего проекта и, по определению, способствуют достижению общей/составляющей цели проекта, то эти свойства с необходимостью должны быть описаны и зафиксированы.

⁴ π^{-1} -реконструкция применяется в стратегическом и тактическом анализе, ее суть весьма точно иллюстрирует фраза "Если на небе зажигаются звезды, значит это кому-нибудь нужно".

На основе π^{-1} -реконструкции справедливо и обратное:

- если свойства информационного обмена не зафиксированы и допускают вариацию, то реально получаемое достижение проекта отклоняется от намеченной цели; при этом величина отклонения от цели определяется как величиной вариации свойств информационного обмена, так и чувствительностью π_i^{-1} -реконструкции к этой вариации (рис. 4).

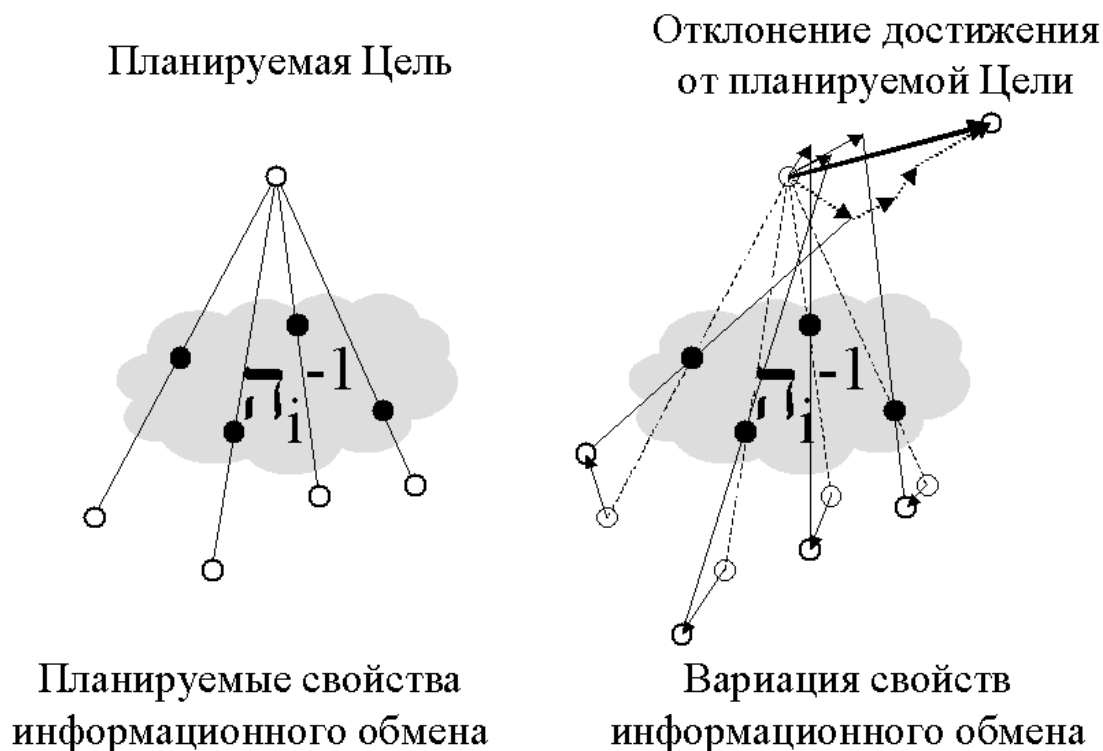


Рис. 4. Вследствие вариации параметров информационного обмена достигаемый проектом результат отклоняется от намеченной цели

Следует отметить, что аналитическая, в том числе и экономическая, оценка функционала взаимного отображения вариации достижений проекта и вариации свойств информационного обмена является отдельной новейшей областью исследования⁵. Возможность оценить допуски вариации свойств информационного обмена и чувствительность к ним всей π -структуры сосредоточена внутри π -логики RL-проекта. Это означает, что проект должен в рамках процесса π -планирования породить фиксированные и π_i -увязанные с целью правила обращения с информацией. Эти правила мы называем информационной политикой (Information Policy) проекта [9].

В общем случае информационная политика содержит предписывающие (например, с информацией типа А следует поступать по схеме В), запрещающие (например, с информацией типа С запрещается поступать по схеме D), а также специфические для проекта правила⁶. Являясь внутренним объектом проекта, информационная политика пишется на естественном языке проекта, однако уже есть предложения по унифицированным языкам для написания информационных политик. Например, в [10] рассматривается Ponder – описательный объектно-ориентированный язык для спецификации политик безопасности и управления для распределенных объектных систем; в этой же работе содержится библиография, представляющая

⁵ В такой постановке эта тема была рассмотрена на семинаре с участием авторов в университете De Montfort [<http://www.dmu.ac.uk>] в декабре 2001 г.

⁶ Некоторые авторы выделяют такие компоненты как авторизация, делегирование, фильтрация, уклонение, долженствование, запрет – в разных комбинациях.

результаты 10-летних исследований, проводимых в Imperial College, Лондон, по управлению на основе политик (policy-based management).

Применив рассмотренный выше модифицированный метод логического структурирования, мы получили модель RL-проекта с точки зрения информационного обмена, которая в терминах бизнес-планирования представлена на рис. 5. Здесь π_i -определенный информационный обмен, являющийся одним из π -определенных бизнес-процессов, принимается за цель вторичной π' -структуры, которая строится обычным образом, принимая первичную π -структуру за ценность.



Рис. 5. Модель RL-проекта с точки зрения информационного обмена. Заполнение логического разрыва между информационным обменом и поддерживающими проект информационными технологиями.

Информационная политика, принадлежащая RL-области, занимает слой результатов (достигнуто / не достигнуто, выполнено / не выполнено) и оказывается логическим слоем, наиболее формализованным и близким к логическому слою поддерживающих технологий. При этом технологии не специфицируются: воплощать информационную политику можно ручными средствами (например, телефон, факс, бумажное делопроизводство), средствами массовой информации, средствами ИТ – в зависимости от результатов π -анализа и планирования: технологии становятся для π -структуры проекта вторичной, π' -структурой. Согласно π -модели, между средствами и результатами позиционируются действия, заключающиеся в применении средств для достижения результатов. действия, включая как технологические, так и организационные, направленные на обеспечение выполнения информационной политики, мы называем информационной поддержкой.

Информационная политика выступает π -связанным с целью проекта перекрытием разрыва со стороны RL:

Цель | π_i -структура | Инф. политика | $\frac{1}{2}$ The Gap | ИТ.

В случае использования ИТ информационная поддержка позиционируется уже в области ИТ и рассматривается в следующем разделе.

2. IT и OSI-модель

В разделе 1 мы провели подготовку к перекрытию разрыва со стороны RL, особое внимание было уделено непрерывности логической связи с целью поддерживаемого RL-проекта. Теперь нам предстоит произвести аналогичные встречные действия со стороны IT, т.е. со стороны поддерживающих технологий. Информационная поддержка, оказываемая средствами IT, явится перекрытием разрыва со стороны IT:

Цель | π -структура | Инф. политика | Инф. поддержка | IT,

При этом мы будем предполагать, что эта поддержка оказывается распределенными, сетевыми средствами; использование локального поддерживающего компьютера может быть представлено как частный случай.

Существенным упрощением задачи является то обстоятельство, что в области IT существуют универсальные π -логика и π -структуры. Такими структурами для компьютерных сетей являются OSI-модель [11] и ее вариации или подобию, в частности широко используемая на практике TCP/IP модель [12]. Предложенное выше π -каскадирование позволит нам создать непрерывную π -структуру от целей RL-проекта до любого логического уровня OSI-подобной модели.

Тот факт, что OSI-подобные модели являются π -структурами, позволяет нам в этом контексте использовать предельно упрощенную модель:

$$IT = APPL \mid NET \mid PHU,$$

где APPL и PHU – обычные для OSI- и TCP/IP моделей логические слои в исходном значении, а NET – все остальные логические слои исходных моделей, например, NET = TCP | IP для TCP/IP модели; и записать:

Цель | π -структура | Политика | Поддержка | APPL | NET | PHU,

что является искомым решением. При этом, подчеркнем еще раз, уже частично выполнены условия π -непрерывности; дополнительно необходимо выполнить условия π -непрерывности следующих отображений:

1. Информационная политика | Информационная поддержка,
2. Информационная поддержка | APPL.

Первое условие является, по сути, формулировкой определения информационной поддержки; но его практическая реализуемость является ключом ко всей проблеме и рассматривается в следующем разделе.

Второе условие наиболее просто выполнить, продолжив развитие OSI-подобной модели вверх и введя дополнительный логический слой с соблюдением принципов OSI и "разместив" в этом слое информационную поддержку.

Логический слой информационных потоков и структур, названный INFO, был предложен нами [13] для заполнения указанной ниши. В соответствии с принципами OSI (и π -непрерывности) он опирается на предшествующий слой APPL, который является для него обслуживающим. Это означает, что для осуществления действий в рамках информационной поддержки (слой INFO) создается программный продукт (слой APPL), который в свою очередь опирается на стандартные сети передачи данных (слой NET и далее PHU). Важнейшим обстоятельством является тот факт, что созданный программный продукт находится в π -непрерывной связи с целями поддерживаемого проекта, что является решением поставленной задачи по преодолению разрыва:

Цель | π -структура | Политика | INFO | APPL | NET | PHU.

Таким образом, нам удалось предложить непрерывную логику от (общей) цели поддерживаемого RL-проекта вплоть до аппаратных средств, используемых для его информационной поддержки.

Конечно, практическая результативность предложенного подхода будет зависеть как от качества предпроектного 7-планирования, так и от принципиальной возможности учесть влияющие факторы и оценить их динамику, что для сложных проектов никак не может считаться простой задачей.

Отметим интересный побочный результат введения и активного использования логического слоя INFO. Поскольку логический слой INFO следует принципам OSI – т.е. населяется протоколами "горизонтального" взаимодействия и аналогом PDU⁷ для "вертикального" взаимодействия, можно надеяться на феномен совместимости информационных поддержек разных RL-проектов. Действительно, скажем на уровне NET, одни и те же сети используются для передачи текста, графики и т.п., причем совершенно произвольного смыслового наполнения. Аналогично, возможен случай, когда одна и та же информационная поддержка будет применима (а стало быть, и применена) для обслуживания нескольких различных по целям и сути RL-проектов. А это означает не что иное, как бизнес-возможность, новый сегмент рынка – аутсорсное⁸ предоставление информационной поддержки средствами Интернет. В свою очередь это означает рост рынка транзитных информационных услуг⁹ [14]. Следовательно, весьма быстро провайдер¹⁰ таких услуг окажется перед соблазном предложить на рынок агрегированную информацию по исходным фактурам¹¹, доверенным ему клиентами, и станет снижать цены аутсорсного обслуживания в обмен на право производить агрегирование по истечении меньшего периода конфиденциальности. Рынок сегментируется: аутсорсно и дешево будет обслуживаться информация, для которой важность скорости и качества обработки доминирует над конфиденциальностью, стоимость закрытых корпоративных информационных систем будет увеличиваться при относительном уменьшении их числа.

3. Практические результаты и Инициатива RISBANK

В разделах 1 и 2 мы провели перекрытие разрыва, пользуясь формальными методами, и определили ряд условий, соблюдение которых необходимо для достижения успеха на практике. Теперь рассмотрим практический подход и полученные результаты. Задача состоит в отображении информационных политик поддерживаемых проектов на логический слой INFO расширенной OSI-подобной модели.

В 1997 г. авторы сконцентрировали внимание на переходе от производства программного продукта к производству поставляемой по сети информационной услуги. В 1998 г. была сформулирована концепция информационной поддержки в Интернет (Internet Information Support), а в январе 1999 г. – принцип аренды информационной поддержки в Интернет¹² и наименование соответствующего бизнеса, а также

⁷ PDU – англ. Protocol Data Unit – агрегат данных/информации, приспособленный для передачи между соседними логическими слоями.

⁸ Англ. outsource – внешний, расположенный снаружи, приходящий извне; передача части бизнес-процесса стороннему исполнителю, тип услуги.

⁹ Транзитные информационные услуги – услуги, заключающиеся в предоставлении клиенту возможности манипулировать собственной информацией (например, телефония), в отличие от терминальных информационных услуг, когда суть услуги состоит в поставке информации (например, справочник, классический Web сайт).

¹⁰ Англ. provider – поставщик услуг.

¹¹ Исходная фактура – фактический материал, ценность и внутренние взаимосвязи которого не оценены, коллекция разрозненных фактов; исходный материал для аналитики.

¹² Синонимом является вводимый в настоящее время английский термин *Service Oriented Approach*.

предложена аббревиатура RISBANK (Rent of Information Support Based on Advanced Network Knowledge) [15]. В настоящее время вся деятельность по RISBANK объединена наименованием Инициатива RISBANK (RISBANK Initiative)¹³ [16].

Если рассмотреть информационные задачи различных RL-проектов, представленные в виде (формализованных) информационных политик, то они окажутся уникальными лишь на первый взгляд; при более внимательном анализе оказывается, что их разнообразие определяется в первую очередь использованием синонимов. Пациент поликлиники, участник конференции, покупатель в магазине, клиент банка – все люди, но рассматриваются разными проектами с разных точек зрения. Аналогично, и информационные действия, которые эти люди осуществляют в рамках различных проектов, в значительной мере совпадают, будучи рассмотрены с точки зрения информационных потоков и структур. В реальности уникальные в области RL проекты в большинстве случаев komponуют свои информационные задачи из ограниченного ряда элементарных информационных задач, в полном соответствии с π -принципом¹⁴. Таким образом, если решить элементарные информационные задачи, а затем осуществить их динамическую π ⁻¹-реконструкцию, то информационная задача проекта будет решена.

На основании предположения о неуникальности, а как следствие – о декомпозируемости информационных задач RL мы предложили разложение информационных политик по некоторому единому базису элементарных информационных задач. Затем элементарные информационные задачи, принадлежащие области RL, изоморфно отображаются на объекты уровня INFO, принадлежащие области IT. Изоморфное отображение разбивает общую проблему преодоления разрыва на ряд малых – и легко решаемых на практике – проблем преодоления разрыва ортами информационного пространства (Information Space). Это обстоятельство важно и напоминает рассмотренную в начале организацию государственной границы (рис. 2), на которой на обеих сторонах организуются пункты пересечения границы, принадлежащие разным странам, но расположенные попарно и работающие согласованно. Так и в этом случае информационные поддержки средствами IT превращаются в простые линейные комбинации (ортогональных) стандартных и заблаговременно решенных информационных задач из стандартных модулей уровня INFO – RISBANK классов (RISBANK Class), поддерживаемых соответствующим программным обеспечением¹⁵ [17], плюс, возможно, дополнительный внебазисный член, представляющий новую(ые), ранее не решенную(ые) информационную(ые) задачу(и):

$$Y = \sum_{i=1}^n a_i x_i + \varphi,$$

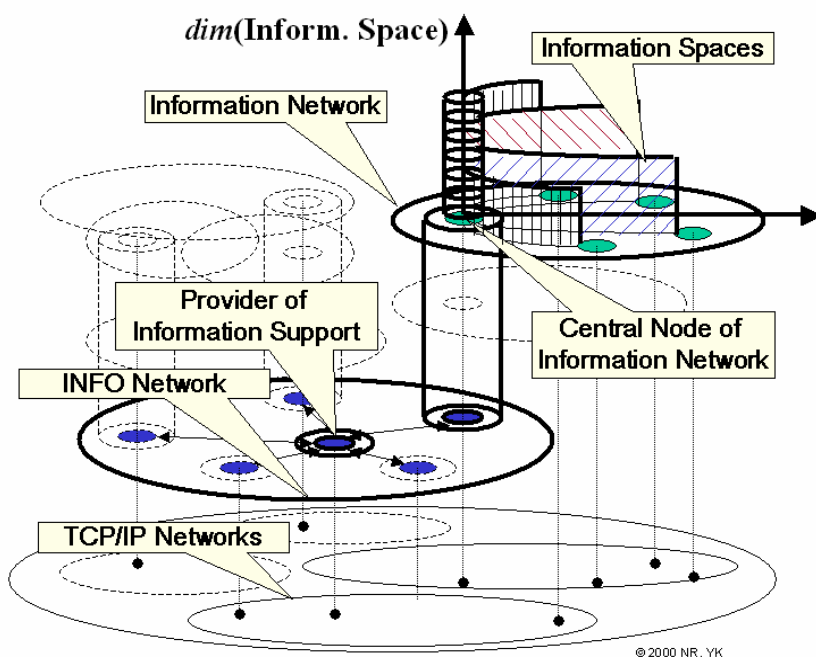
где Y – информационная поддержка отдельного проекта, x – базисный элемент (RISBANK Класс) единого информационного пространства провайдера, n – размерность этого пространства, φ – специфичная для данного проекта информационная задача (Project specific Class), неразложимая по существующему базису.

¹³ Термины и словосочетания, выделенные в этом разделе курсивом, а также наименования RISBANK Классов – © RISBANK Initiative.

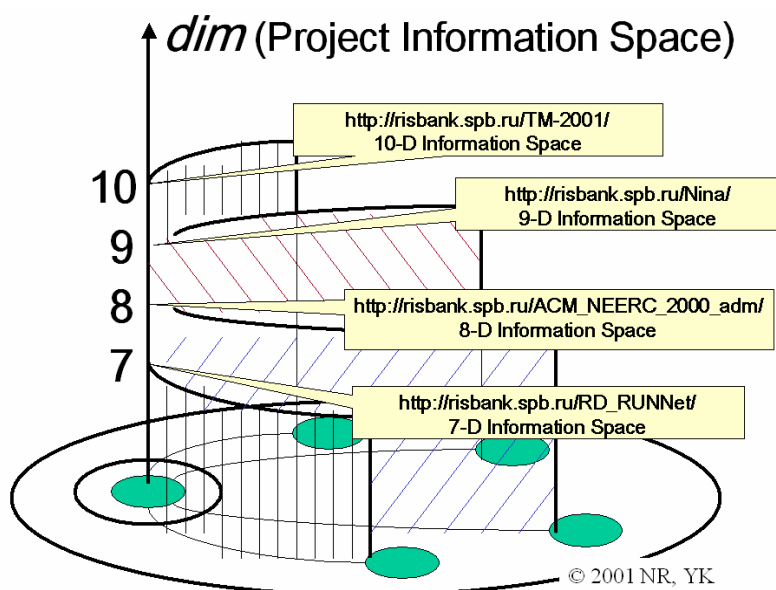
¹⁴ Аналогичный принцип лежит в основе объектно-ориентированных языков, в том числе и языка написания политик Ponder, отмеченного выше.

¹⁵ Важно отметить: требуя ортогональности на уровнях информационной политики и INFO, мы не требуем изоморфизма отображения этого единого базиса на уровень APPL, а также ортогональности объектов уровня APPL, что наблюдается и на практике, позволяя организовать при производстве поддерживающего программного продукта спиральный цикл Бозма в бесконечной версии.

Коэффициент a несет здесь повышенную смысловую нагрузку, описывая информационную сеть (Information Network) (рис. 6) проекта, т.е. не только количество вызовов некоторого RISBANK класса x , но и топологический порядок вызовов, и права доступа к вызову класса.



а)



б)

Рис. 6. В иерархии сетей (снизу вверх): PHY | NET | APPL | INFO над верхним слоем располагаются информационные сети поддерживаемых проектов (а). Информационное пространство проектов обладает размерностью (б).

В настоящее время мы рассматриваем следующую иерархию сетей (снизу вверх)¹⁶: сети физической связности (слой PHY) | сети передачи данных (слои NET или

¹⁶ На рисунке слои PHY и APPL не показаны.

TCP/IP) | сети приложений (слой APPL, например, Oracle SQL*Net) | сети информационной поддержки (слой INFO, например, RISBANK) | информационные сети поддерживаемых проектов¹⁷ (область RL). Информационное пространство RL-проекта графически позиционируется перпендикулярно условной плоскости его информационной сети и определяет номенклатуру типов элементарных информационных задач, решаемых проектом; размерность информационного пространства становится исчисляемой величиной и мерой сложности поддерживаемого информационного обмена в проекте (рис. 7).

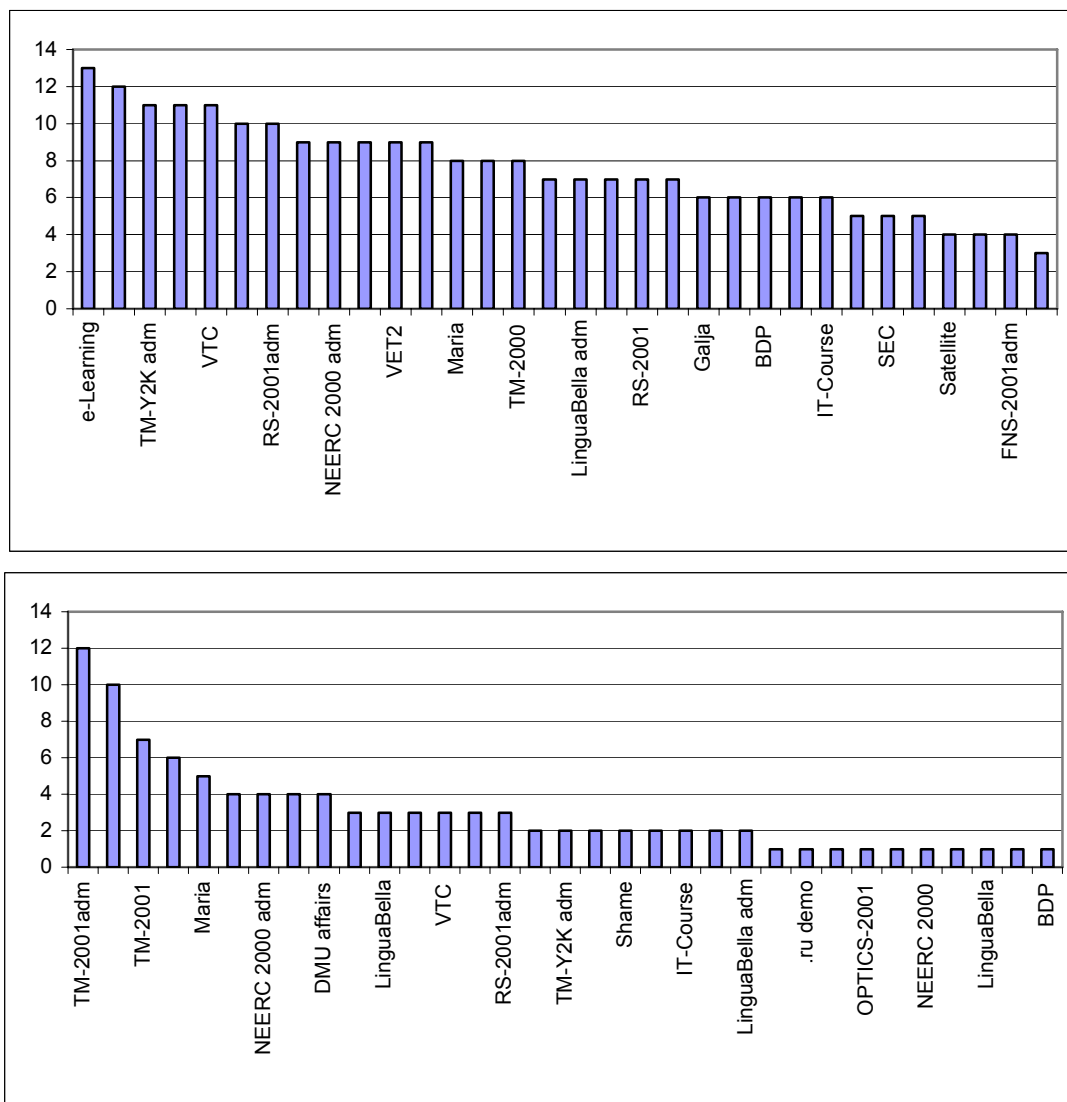


Рис. 7. Размерности и группы доступа поддерживаемых проектов, принадлежащих пространству сети RISBANK

На рис. 8 приведены примеры взаимодействия RISBANK классов: а) каскадный вызов для формирования одного макро-класса в пределах одного RISBANK-сервера (поддержка оформления европроектов Bistro Tacis), б) структура дочернего RISBANK-узла RISBANK-сети, контролируемого головным RISBANK-узлом (экспериментальная сеть СПО РФ CORNET).

¹⁷ В этом контексте строгое значение получают такие словосочетания как, например, "Информационная Сеть предприятия", "Информационные Сети образования и науки".

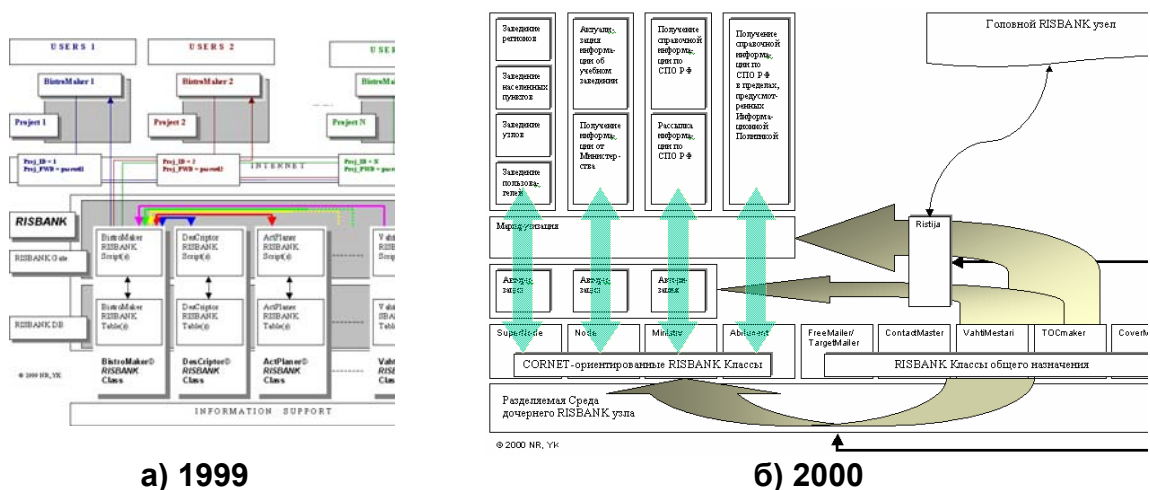


Рис. 8. Примеры взаимодействия RISBANK классов: а) каскадный вызов для формирования одного макро-класса в пределах одного RISBANK-сервера; б) структура дочернего RISBANK-узла RISBANK-сети, контролируемого головным RISBANK-узлом.

Важной чертой RISBANK является постоянное развитие его базиса¹⁸ (или базиса RISBANK-сервера), которую мы сейчас рассматривать не будем; заинтересованный читатель может обратиться, например, к работе [18], где этот процесс описан детально.

К настоящему времени формально насчитывается 33 RISBANK класса, но далеко не все они используются равномерно, а часть из них является версиями. Приведем статистику использования наиболее часто вызываемых RISBANK классов. Подчеркнем, что это не статистика обращений клиента к задаче (как например, Web лог), а число подключений стандартной информационной задачи к поддерживаемым проектам.

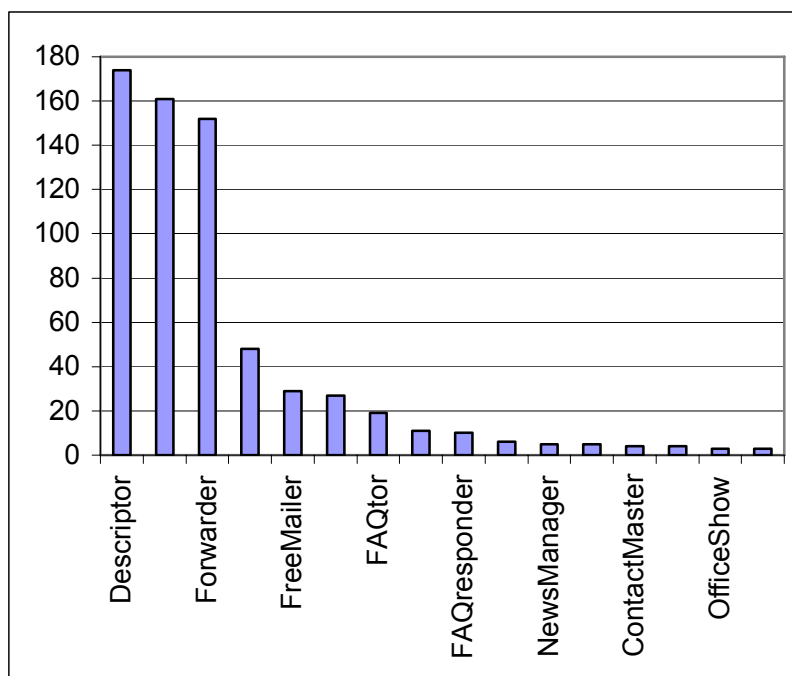


Рис. 9. Статистика вызовов RISBANK-классов

¹⁸ Развитие RISBANK Класа представимо циклом Бозма: прототип – улучшение – улучшение и т.д., но без прекращения функциональности.

Классы Descriptor ("статическая"¹⁹ информация: текст, изображения, статические Web страницы), TOCmaker (механизм навигации и скрытой безопасности – средство создания информационной сети, динамическая навигационная схема сайта), Forwarder (подключение приложений²⁰, обычно переход на специфические для проекта задачи вне базиса), являются наиболее востребованными. Значительно реже подключаются "более транзитные" классы VahtiMestari (персональная авторизация пользователей и слежение за использованием закрытыми областями информационных сетей проектов), FreeMailer (обращение клиента к провайдеру непосредственно, без использования внешней электронной почты), OfficeShow (генерируемые сервером слайды с возможностью управления воспроизведением со стороны пользователя) и FAQtor (открытые вопросы и авторизованные ответы). Это естественно: эти классы пока редко подключаются к поддерживаемому проекту более одного раза; никаких ограничений на их подключение нет, но их число свидетельствует о реальной сложности поддерживаемого RL-проекта, которая в настоящее время обычно ограничена возможностями пользователя поддерживать разветвленную (собственную) информационную сеть. Однако в самом начале 2002 г. был создан проект-специфический класс, автоматически порождающий "информационные подсети"²¹ и информационные подпространства, что ведет к быстрому увеличению счета классов TOCmaker и VahtiMestari.

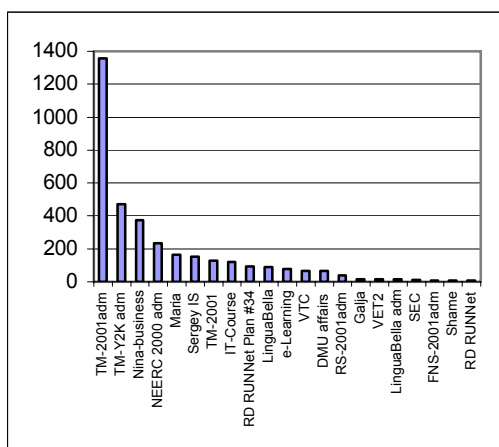
Все информационные поддержки более чем 20-ти проектов различных областей деятельности (программирование, IT, Web-образование, радиоактивная безопасность, персональная деятельность), различной гранулярности (обслуживание в международном или всероссийском масштабе, корпоративное или персональное), долгосрочные, краткосрочные или постоянные по времени жизни проекта, – все они имеют единую, прозрачную организационную и технологическую структуру. Все они, действуя параллельно, используя одно и то же ядро информационной системы и даже, зачастую, одну и ту же таблицу распределенной базы данных, воспринимаемые различными пользователями как выделенные информационные поддержки, являются в действительности информационными подпространствами единого информационного пространства, демонстрируя таким образом предсказуемое существование базовых элементов базового пространства – в различных комбинациях и пропорциях. Подобная структура позволяет автоматически получать различные статистики по любому числу поддерживаемых проектов на определенный момент времени.

На рис. 10 в качестве примера представлены а) статистика посещения пользователями закрытых областей поддерживаемых проектов; б) пример реконструкции путей трех пользователей между проектами в едином информационном пространстве за период около 20 дней. Крупные треугольники и индекс проекта 301 соответствуют поведению "стандартного" пользователя, изредка посещающего одну поддержку. Квадраты и сплошная линия представляют путь пользователя, активно работающего в нескольких проектах одновременно. Ромбы и пунктирная линия иллюстрируют характерный путь разработчика.

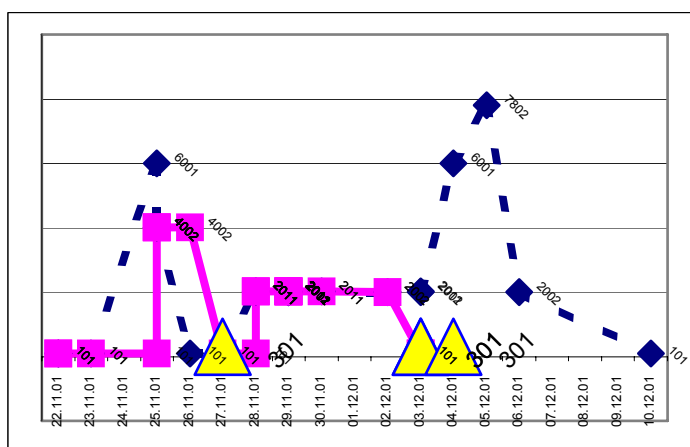
¹⁹ RISBANK не содержит статического HTML материала в принципе; заключение в кавычки подчеркивает аналогию.

²⁰ Специальный класс для перехода с уровня INFO на уровень APPL, т.е. к традиционным файлам и скриптам. Невозможность организовать непосредственно из RISBANK обычную HTML ссылку своеобразно подтверждает принадлежность RISBANK к иному, нежели APPL, логическому слою.

²¹ Разговорный, но выразительный термин "подсеть" используется в менеджменте компьютерных сетей для обозначения подчиненных сетевых объектов.



а)



б)

Рис. 10. Статистики пользовательской деятельности: а) статистика посещения пользователями закрытых областей поддерживаемых проектов; б) пример реконструкции путей трех пользователей между проектами в едином информационном пространстве

Для управления информационным пространством используется специальный инструмент RISBANK TOC Builder, предоставляющий Web интерфейс для выполнения следующих технологических операций (рис. 11):

- менеджмент информационных сетей поддерживаемых проектов – создание, модификация, ликвидация навигационных схем и управление правами доступа, включая механизмы скрытой и неявной авторизации;
- менеджмент информационных пространств поддерживаемых проектов – подключение/отключение RISBANK классов, управление параметрами вызова (прообраз протоколов уровня INFO).

Дополнительное удобство представляют автоматические справочники прецедентов использования классов. В настоящее время управление RISBANK практически полностью организовано через Web, т.е. настройка информационной поддержки может производиться «из офиса заказчика».

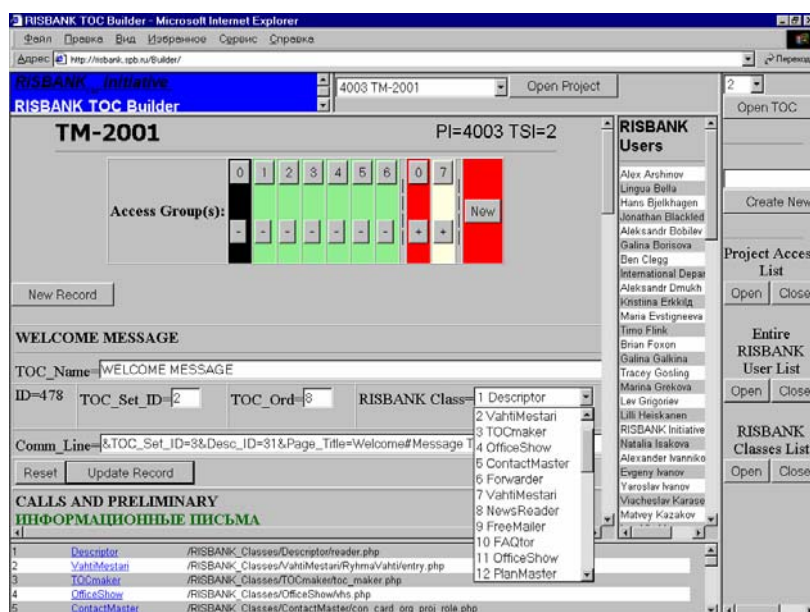


Рис. 11. Web панель управления единым информационным пространством поддерживаемых проектов

Информационная поддержка проектов имеет статус научных исследований. Тем не менее, высокая степень ответственности реальных поддержанных проектов явилась принципиальным фактором при практическом тестировании вышеизложенных моделей и концепций. Воплощенные информационные поддержки успешно прошли проверку на параллельную работу, сохранность информации, скорость доступа, период безотказной работы. Отдельным достоинством явилась возможность разместить сервисы на экономичном оборудовании: Pentium 100 / 128М / 1+4Г / WinNT4.0 / Oracle7.3 / PHP3.0 / IIS4.

Заключение

Проводимые научные исследования принадлежат области развития информационных технологий, которая на сегодняшний день является одной из наиболее актуальных, что, к большому удовольствию авторов, было подтверждено коллегами из Финляндии, Южной Африки, Великобритании. Это направление признается приоритетным на ближайшие 10 лет, в перспективе у английских коллег – специализированная международная ежегодная конференция, а у авторов статьи – издание книги. В этой книге мы постараемся отразить и проанализировать специфику исследуемой области уровня INFO. Прежде всего, при интуитивной очевидности, еще не существует стандартов, протоколов и прочих атрибутов этого уровня. По-видимому, на новом витке повторяется процесс, который мы все совсем недавно наблюдали, когда впервые широко обсуждались принципы Открытых Систем, создавались логические слои теории и анализа компьютерных сетей, ныне именуемые традиционными. С другой стороны, степень "открытости" систем уровня INFO на порядок выше, им предстоит "открыть" границу между реальной жизнью и информационными технологиями, соединить две области, не только различные, а во многом и антагонистические, но уже в положительном смысле обреченные на долгое взаимодействие. Готовящаяся к выпуску летом 2002 г. монография более подробно раскроет вопросы, кратко представленные в этой статье, коснется ряда смежных тем, которые вышли за рамки этой статьи.

Литература

1. N.Mullen. Developing an Enterprise Data Strategy. DM Review, Vol. 11, N 10, 2001, pp. 16, 55.
2. A.Finkelstein, J.Kramer. Software Engineering: A Roadmap. *In*: The Future of Software Engineering 2000. 22nd International Conference on Software Engineering. Editor A.Finkelstein. ACM, NY, 2000, pp. 3-22.
3. D.J.Flynn. Information Systems Requirements: Determination and Analysis. 2nd Edition. The McGraw-Hill Companies, 1998.
4. R.J.Gascoyne, K.Ozcubukcu. Corporate Internet Planning Guide. Aligning Internet Strategy With Business Goals. John Wiley & Sons, Inc., 1997.
5. STRL. Publications, 1995-2001. On-line catalogue, <http://www.cse.dmu.ac.uk/STRL/research/publications/pubs/index.html>
6. The United Nations Development Programme. UNDP Programming Manual. Chapter 4. Formulating Programmes and Projects. Task 6. Using the logical framework. December 2000, p.13. <http://www.undp.org/osg/pm/progm4.pdf>
7. Рубина Н.А., Кирчин Ю.Г. . Глава 4 годового отчета Вузтелекомцентра. МО РФ, СПб, 1999, http://risbank.spb.ru/risbank2/Presentations/Report_1999/1210_Report.doc.
8. Nina Rubina. Internet for the Project Management. PhD Work Plan, 2000, http://risbank.spb.ru/risbank2/CD_Ready.
9. Кирчин Ю.Г., Рубина Н.А.. Информационная Поддержка. Web статья, январь 1999, <http://risbank.spb.ru/risbank2/papers/InfoSupport/InfoSupportNote.htm>.

10. N.Damianou, N.Dulay, E.Lupu, M.Sloman. The Ponder Policy Specification Language. Proc. Policy 2001: Workshop on Policies for Distributed Systems and Networks, Bristol, UK, 29-31 Jan. 2001, pp.18-39.
11. D.Wetteroth. OSI Reference Model for Telecommunications. McGraw-Hill Professional Publishing, 2001.
12. M.A.Miller. Troubleshooting TCP/IP. Third Edition. Hungry Minds, Inc., 1999.
13. Кирчин Ю.Г., Рубина Н.А., Танаев А.В. Информационный оборот предприятия. Web статья, декабрь 1997, <http://risbank.spb.ru/risbank2/papers/InfoSupport/Infoexchange.htm>
14. B.T. Clegg, Y.G. Kirchin, N.A. Rubina. Marketing Network Information Services. // Proc. of Telematics-2000 International Conference, SPb, 2000, http://risbank.spb.ru/tm2000/info/tm_paper_present.php? Proj_ID=4001&Paper_ID=251.
15. N.Rubina, Y.Kirchin. RISBANK – Information Support over the Internet. Web paper, April 1999, http://risbank.spb.ru/risbank2/papers/InfoSupport/RISBANK_ADBIS.htm.
16. Private Research and Development Initiative on over-Application Internet Information Services by Nina Rubina and Yury Kirchin. RISBANK Initiative. Web site, <http://risbank.spb.ru>.
17. Y.G.Kirchin, N.A.Rubina. Information Spaces. // Proc. of International Conference on Telematics and Web-Based Education. SPb, 18-21 June 2001, International Volume, pp. 58-61.
18. Иванов Е.В., Славянский О.Е., Кирчин Ю.Г., Рубина Н.А.. Информационная Поддержка системы среднего профессионального образования Российской Федерации Отчет о НИР, часть 4, МО РФ, Петровский Колледж, СПб, 2000, с. 36-37, http://risbank.spb.ru/risbank2/CORNET/report_4.zip.

ПОСТРОЕНИЕ СИСТЕМ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА НА УРОВНЕ ИНФОРМАЦИОННОЙ СЕТИ ПРЕДПРИЯТИЯ

А.Г. Баженов

Рассмотрены подходы к построению систем, обеспечивающих информационную поддержку электронного документооборота. Уточнено понятие информационной сети, предложен метод построения информационной сети электронного документооборота. Приведен пример реализации информационной сети для решения задач управления инженерными данными.

Введение

На сегодняшний день интенсивное развитие средств телекоммуникаций и глобальных компьютерных сетей позволяет строить распределенные информационные системы для осуществления информационной поддержки различных производственных процессов на современном предприятии. Поскольку указанные процессы разнообразны по содержанию, для каждого предприятия и для каждой задачи они уникальны, то решение проблемы в области предоставления универсального механизма построения информационных систем является актуальным. Одним из вариантов решения проблемы является использование при проектировании концепции информационных сетей [2, 3]. В настоящей статье предлагается вариант построения информационной сети электронного документооборота и управления инженерными данными.

1. Уточнение понятия "информационная сеть" и его применение для решения задач электронного документооборота

Перед рассмотрением вопроса применения информационных сетей для поддержки электронного документооборота необходимо конкретизировать термин "информационная сеть". В настоящее время понятие "информационная сеть" имеет очень широкую трактовку, начиная от определения информационных потоков в экономических и социологических науках и заканчивая синонимом глобальной компьютерной сети Internet.

Рассмотрим проблему информационной поддержки процессов, протекающих на предприятии. С одной стороны, указанные процессы определяются с помощью набора документов, посредством которых происходит обмен информацией между участниками процесса, и документов, регламентирующих правила обмена. С другой стороны, имеются стандартизованные средства обмена информацией в электронном виде, базирующиеся на теории построения компьютерных сетей. Возникает вопрос, как обеспечить взаимодействие между процессами, протекающими на предприятии, и существующими средствами обмена информацией (см. рис. 1, а). В настоящее время этот вопрос решается путем разработки специализированного программного обеспечения (СПО) для каждого процесса (см. рис. 1, б). Основным недостатком такого подхода является невозможность использования СПО для информационной поддержки других процессов, что обуславливает высокую стоимость разработки СПО и сложность адаптации СПО при изменении характеристик производственного процесса.

Недостатки существующего подхода в организации взаимосвязи между процессами, протекающими на предприятии, и существующими средствами обмена информацией ставят задачу создания нового метода организации указанной взаимосвязи. Предлагаемый подход заключается в выделении уровня программного обеспечения, позволяющего решать ряд типовых задач информационной поддержки и сетевого обмена электронной информацией, которые могут присутствовать в различных производственных процессах на предприятии (см. рис. 1, в).



Рис. 1. Структура обмена информацией

В настоящей статье под информационной сетью понимается организация среды обмена информацией между процессами, которая обладает следующими свойствами:

- предоставляет стандартизованный интерфейс для поддержки процессов, протекающих на предприятии, с помощью набора типовых действий (если рассмотреть реализацию типового процесса, то типовое действие будет представлять собой СПО для данного процесса (см. рис. 1, б));
- предоставляет механизм для поддержки специализированных действий с целью использования их совместно с типовыми действиями, что позволяет организовать информационную поддержку любого процесса реального мира;
- предоставляет стандартизованный интерфейс для организации обмена информацией между действиями, как типовыми, так и специализированными.

Исходя из предложенного определения, для построения информационной сети, поддерживающей электронный документооборот предприятия, необходимо решить три задачи, а именно:

- определить множество типовых действий;
- определить порядок взаимодействия типовых действий со специализированными;
- описать механизм обмена информацией, который включает в себя определение формата передаваемой информации и определение порядка обмена информацией.

Перейдем теперь к более подробному рассмотрению путей решения перечисленных задач.

2. Построение информационной сети, поддерживающей электронный документооборот предприятия

При построении любой системы, в том числе и информационной сети, необходима формализация задачи, что позволяет разработать определенные методы ее решения.

На рис. 2 приведена обобщенная схема построения информационной сети. Для формализации задач в области электронного документооборота целесообразно использовать метод логического структурирования (Logical Frameworks) [6]. Непосредственное применение указанного метода для формирования структуры информационной сети подробно описано в работе [3]. В результате применения указанного метода, обозначенного на рис. 2 (шаг а) оператором L, получим множество частных задач $\{A_j\}: j \in [1..J], j \in Z$, где J — количество частных задач, результат решения которых может быть описан двумя состояниями — "истинно" или "ложно". Множество $\{A_j\}$ не содержит информации о конкретной технологии решения задачи и является техническим заданием на разработку системы.

Для выбора оптимального метода решения задачи необходимо учесть особенности технологии, на основе которой обеспечивается решение поставленной задачи. В нашем случае особенности технологии учитываются характеристиками информационной сети и электронного представления информации.

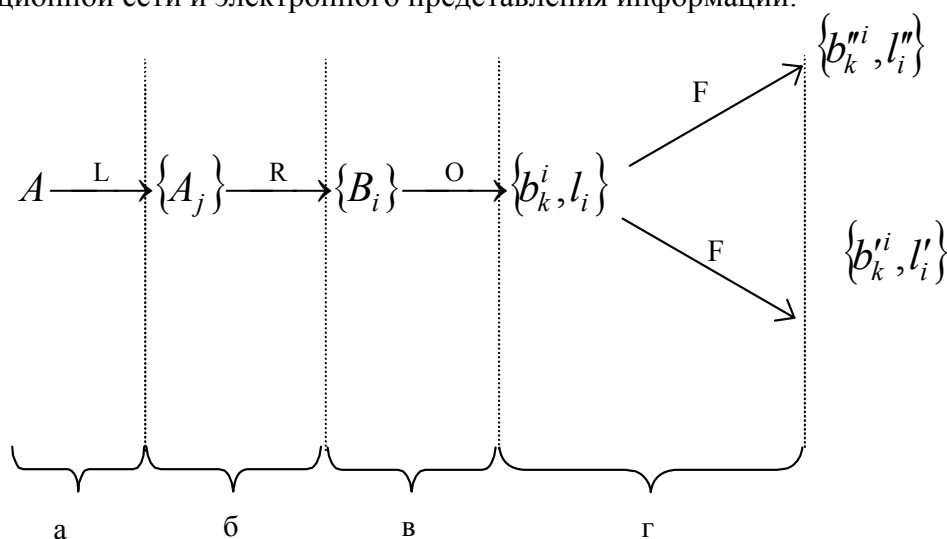


Рис. 2. Схема построения информационной сети

Перед проведением дальнейшей детализации модели информационной сети необходимо отметить, что многие задачи A_j могут иметь варианты решения с использованием других технологий, и возникает естественное желание перенести этот вариант решения на новую технологию. Использование идей старых технологий далеко не всегда удачно и часто требует пересмотра. Процесс пересмотра идеи реализации решения частной задачи называется реинжинирингом. В результате проведения реинжиниринга (рис. 2 шаг б) получаем новое множество частных задач $\{B_i\} = R(\{A_j\}): i \in [1..I]; j \in [1..J]; i, j \in Z$, где I — количество частных задач, полученных после реинжиниринга; R — отображение, описывающее процесс реинжиниринга.

Проведение реинжиниринга позволяет скорректировать исходную постановку задачи в части ее реализации с помощью электронного представления информации.

Для получения структуры информационной сети предлагается использовать методы лингвистического анализа и объектно-ориентированного анализа (обозначенного на рис. 2, шаг в, как отображение О) [5]. В результате для каждой задачи B_i получим множество действий $\{b_k^i\}: k \in [1..K]; i \in Z$, и правило взаимосвязи между ними l_i , которые совместно решают задачу B_i .

Множество $\{l_i\}: i \in [1..I]; i \in Z$ описывает уровень информационного обмена проектируемой сети (рис. 1, в). В работе [3] введен термин "информационная политика", описывающий рассмотренное множество. В свою очередь множество $\{b_k^i\}: i \in [1..I]; k \in [1..K]; i, k \in Z$ включает набор действий, которые необходимо совершить.

Для получения окончательной структуры информационной сети необходимо сформулировать критерий F, определяющий типовое действие из множества $\{b_k^i\}$. Решающим моментом при определении типа действия (типовое или специализированное) является частота его использования для описания элементов множества $\{b_k^i\}$. Если частота использования превышает единицу, то элемент множества следует отнести к множеству типовых действий. Полученное множество типовых действий $\{b_k^{ti}\}$ определяет соответствующий уровень информационной сети, $\{l_i'\}$ описывает взаимосвязи между ними. Соответственно, множество $\{b_k^{ni}\} = \{b_k^i\} \setminus \{b_k^{ti}\}$ является множеством специализированных действий (рис. 2, шаг г), $\{l_i''\}$ описывает взаимодействие между ними.

Таким образом, модель информационной сети можно представить в обобщенном виде:

$$F(O(R(L(A)))) = \{b_k^{ti}, l_i'\} \cup \{b_k^{ni}, l_i''\}, \quad (1)$$

где A – цель проекта; F – критерий выделения типовых действий; O – отображение, описывающее применение методов лингвистического и объектно-ориентированного анализа; R – отображение, описывающее процесс реинжиниринга; L – отображение, описывающее применение методов логического структурирования.

Перейдем теперь к рассмотрению применения полученной модели к построению систем управления инженерными данными.

3. Система управления инженерными данными как фрагмент информационной сети предприятия

Применяя подход, описанный выражением (1), к построению систем управления инженерными данными, получим трехуровневую модель информационной сети, как это иллюстрируется на рис. 3. Верхний уровень модели определяет уровень внешних приложений, или, в веденной терминологии, – уровень специализированных действий. Второй уровень относится к системе управления инженерными данными (СУИД) и представляет собой уровень типовых действий. В табл. 1 приведен перечень функциональных модулей соответствующего уровня, которые составляют множество типовых действий в информационной сети (в таблице использовано сокращенное обозначение: ЭД – электронный документ).

Уровень систем управления бизнес-процессами (СУБП) реализует информационный обмен в проектируемой сети. В качестве модели уровня СУБП предлагается использовать результаты работ Workflow Management Coalition [4], где информационный обмен определяется моделью бизнес-процесса обращения инженерной документации, разработанной в электронном виде [1], и моделью единицы информационного обмена, в рассматриваемом случае это модель электронного документа. Можно выделить три класса электронных документов:

- документы, содержащие структурированные (понятные для системы) данные (спецификация, технологическая оснастка и т. д.), полностью обрабатываются системой управления;
- документы, содержащие данные по изделию, которые полностью не задействованы в процессе управления (например, графическое описание деталей);
- документы, описывающие текущее состояние изделия или его частей.

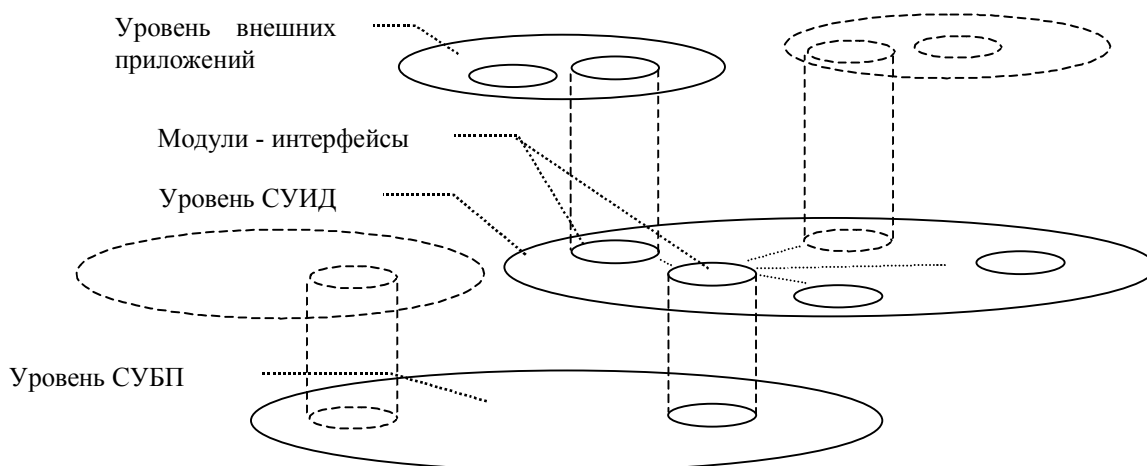


Рис. 3. Модель информационной сети предприятия

Таблица 1

Функциональные модули уровня СУИД

№	Наименование модуля
1	Оформление ЭД
2	Согласование ЭД
3	Интерфейс с хранилищем данных
4	Интерфейс к внешним приложениям
5	Интерфейс к СУБП
6	Регистрация потребителей
7	Помещение в хранилище
8	Удаление из хранилища
9	Поисковая машина
10	Доступ в систему
11	Доступ к компоненту
12	Доступ к параметрам компонента
13	Изменение доступа к объекту
14	Управление бизнес-процессом
15	Управление хранилищем
16	Управление пользователями
17	Управление компонентами

Применение предложенной модели информационной сети позволяет: организовать поддержку бизнес-процесса обращения инженерных данных с учетом особенностей электронного представления документа; решать задачи управления инженерными данными в пределах одного уровня за счет многоуровневой архитектуры информационной сети; эффективно организовать информационные потоки в сети на основе предложенной классификации электронных документов; полностью обеспечить жизненный цикл документации при помощи предложенной структуры уровня СУИД.

Заключение

Развитие технологии информационных сетей позволит в дальнейшем предоставить качественно новый вид услуг в области телекоммуникационной и сетевой поддержки различных распределенных процессов человеческой жизнедеятельности. Указанная услуга может быть востребована при проектировании виртуальных предприятий, систем дистанционного обучения, электронной коммерции и других распределенных приложений.

Литература

1. Баженов А.Г. Влияние компьютерных технологий на бизнес-процесс обращения конструкторской документации // Компьютерное и математическое моделирование в естественных и технических науках: Материалы III Всероссийской научной internet-конференции (сентябрь – ноябрь 2001). Тамбов: ТриАда, 2001. С. 49–55.
2. Кирчин Ю.Г., Рубина Н.А. и др. Информационная поддержка системы среднего профессионального образования Российской Федерации. Заключительный отчет. СПб.: RUNNET, 2000
3. Kirchin Y, Rubina N. Information spaces // International Conference on Telematics and Web-Based Education. Telematica-2001. SPb., IFMO, 2001, p. 58-61.
4. The Workflow Management Coalition Specification. Workflow Management Coalition: The Workflow Reference Model, Document Number TC00-1003, Issue 1.1, November 1994, Brussels, Belgium
5. Yourdon E., Argila C. Case Studies in Object-Oriented Analysis and Design, Yourdon press, Prentice Hall Building Upper Saddle River, 1996 Эдвард Йордон, Карл Аргила, Структурные модели в объектно-ориентированном анализе и проектировании/ Под ред. В. Алеева. М.: ЛОРИ, 1999

Дополнительные источники информации

6. Jackson B. Designing Projects and Project Evaluations Using The Logical Framework Approach. <http://iucn.org/themes/eval/english/lfa.htm>

АНАЛИЗ НАГРУЗКИ И КЛАССИФИКАЦИЯ ПРИЛОЖЕНИЙ В ЛОКАЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЯХ

С.Н. Воротягин

Эффективность локальных вычислительных сетей (ЛВС) определяется соответствием структурно–функциональной организации сети и нагрузки, создаваемой приложениями в рабочих станциях и в сети передачи данных. Вычислительная сеть, предназначенная в основном для передачи больших объемов данных, должна иметь высокоскоростные каналы связи, в то время как сеть, предназначенная для реализации приложений, требующих больших объемов вычислений, должна иметь мощные рабочие станции.

Нагрузка на рабочие станции и канал связи в основном зависит от используемых приложений. Проектирование ЛВС связано с разработкой адекватных моделей, отображающих процесс функционирования сети. Адекватность моделей определяется достоверностью параметров, описывающих как структурно–функциональную организацию сети, так и нагрузку, создаваемую приложениями. При проектировании ЛВС априори может быть задан состав приложений, которые будут выполняться в сети. При этом возникает задача классификации приложений в зависимости от нагрузки, создаваемой в рабочих станциях и канале связи различными приложениями.

Нагрузка в ЛВС делится на вычислительную нагрузку, создаваемую непосредственно в рабочих станциях и серверах, и сетевую в канале связи. Для определения вычислительной нагрузки предлагается измерять загрузку основных ресурсов рабочих станций, таких как оперативная память и центральный процессор. Сетевая нагрузка также играет важную роль в работе сети. Следовательно, при проектировании новой или модернизации уже существующей сети часто возникает необходимость в количественном измерении некоторых сетевых характеристик (например, интенсивность потоков данных по сетевым линиям связи, задержки, возникающие на различных этапах обработки пакетов, время реакции на запросы того или иного вида, частота возникновения определенных событий). Для определения сетевой нагрузки необходимо измерять количество и типы передаваемых пакетов, а также количество байт, передаваемых за единицу времени.

Для решения задач диагностики сети и мониторинга ее производительности существует множество различных средств и инструментов, которые можно разделить на несколько классов [1]:

- системы управления сетью;
- средства управления системой;
- встроенные системы диагностики и управления;
- анализаторы протоколов;
- оборудование для диагностики и сертификации кабельных систем;
- экспертные системы;
- многофункциональные устройства анализа и диагностики.

Для формирования нагрузки, создаваемой различными типами приложений, можно измерить отдельно каждое приложение, а затем, зная время работы в каждом из приложений в течение определенного промежутка времени, создавать смеси и получать значения нагрузки, приближенные к реальным. Эти значения могут использоваться для формирования нагрузочных параметров моделей ЛВС.

Измерения проводились в ЛВС Fast Ethernet, содержащей 9 рабочих станций и один файл-сервер.

Измерительные программные средства дополнительно загружают сеть, что искажает реальную нагрузку, создаваемую приложениями. Для повышения

достоверности результатов измерений использовались различные измерительные средства. Измерения проводились в операционной системе Windows 98 с помощью встроенного средства "Системный монитор", а также с помощью пакета Norton Utilities. В среде WindowsNT Terminal Server измерения проводились с помощью встроенной в WinNT утилиты Performance Monitor [2]. Результаты измерений представлялись в виде графиков и временных диаграмм

На основе анализа полученных результатов выполнена классификация приложений по степени загруженности ресурсов системы (центрального процессора, основной памяти и канала связи) при различных видах работы с приложением, а именно:

- открытие – Open – процесс открытия приложения с момента запуска приложения до момента окончания его загрузки;
- работа – Work – выполнение набора определенных действий, характерных для данного приложения;
- сохранение – Save – сохранение в стандартном формате файла, созданного в результате работы в приложении;
- выход – Quit – включает в себя как выход из приложения, так и автоматическое закрытие уже сохраненного файла.

Для классификации использовалось следующее обозначение приложений: Open / Work / Save / Quit, где Open – степень загрузки ЦП и ОП при открытии приложения, Work – степень загрузки ЦП и ОП при работе в приложении, Save – степень загрузки ЦП и ОП при сохранении файла, Quit – степень загрузки ЦП и ОП при выходе из приложения. В классификации параметры Open, Work, Save, и Quit могут принимать значения от 1 до 4 (табл. 1).

Таблица 1

Признаки классификации OWSQ

Значение параметра классификации	Степень загрузки ресурса, %	Название степени загрузки
1	0 – 25	Малая
2	26 – 50	Средняя
3	51 – 75	Высокая
4	76 – 100	Критическая

В табл. 2 представлены обобщенные данные измерений, полученные для 16 исследованных приложений по степени загрузки ресурсов.

Таблица 2

Данные классификации OWSQ рассмотренных приложений

№ п/п	Название приложения	Обозначение	Тип используемого ресурса	
			ЦП	ОП
1	MS Word 97	99_70 / 100_74 / 100_75 / 32_59	4 / 4 / 4 / 2	3 / 3 / 3 / 3
2	MS Excel 97	13_69 / 3_72 / 4_76 / 8_72	1 / 1 / 1 / 1	3 / 3 / 4 / 3
3	MS Power Point 97	8_73 / 10_79 / 14_73 / 13_62	1 / 1 / 1 / 1	3 / 4 / 3 / 3
4	MS Access 97	100_76 / 100_82 / 100_81 / 14_64	4 / 4 / 4 / 1	4 / 4 / 4 / 3
5	MS Office 97	51_76	3	4
6	PROMT 98	74_65 / 72_63 / 8_64 / 100_75	3 / 3 / 1 / 4	3 / 3 / 3 / 3
7	Photoshop 3.0	38_56 / 70_71 / 45_74 / 14_71	2 / 3 / 2 / 1	3 / 3 / 3 / 3

8	Premiere 5.5	50_80 / 75_91 / 14_80 / 24_59	2 / 3 / 1 / 1	4 / 4 / 4 / 3
9	3D MAX R2	55_100 / 86_97 / 8_100 / 23_55	3 / 4 / 1 / 1	4 / 4 / 4 / 3
10	БД «Адреса и телефоны»	91_84 / 100_85 / - / 99_54	4 / 4 / - / 4	4 / 4 / - / 3
11	Edc.exe	99_61 / 99_55 / - / 99_54	4 / 4 / - / 4	3 / 3 / - / 3
12	Dos Navigator v1.50	99_59 / 99_61 / - / 98_61	4 / 4 / - / 4	3 / 3 / - / 3
13	N Commander for Win95	23_68 / 9_66 / - / 14_64	1 / 1 / - / 1	3 / 3 / - / 3
14	Far v1.61	42_58 / 100_58 / - / 9_59	2 / 4 / - / 1	3 / 3 / - / 3
15	WinNavigator v1.65	70_60 / 45_55 / - / 28_54	3 / 2 / - / 2	3 / 3 / - / 3
16	Файловые менеджеры	62_60	3	3

В табл. 3 приведена классификация приложений в соответствии с кодами, учитывающими степень загрузки ЦП и ОП (см. табл. 4).

Таблица 3

Классификация OWSQ рассмотренных приложений

Коды	Приложение	OWSQ	
		ЦП	ОП
1.	1.1 Access	4/4/4/1	4/4/4/3
	1.2 БД «Адреса и тел.»	4/4/-/4	4/4/-/3
2.	2.1 Word	4/4/4/2	3/3/3/3
	2.2 edc.exe	4/4/-/4	3/3/-/3
	2.3 DN		
3.	Office (по 25%)	3	4
	5.1 3D MAX	3/4/1/1	4/4/4/3
4.	6.1 PROMT 98	3/3/1/4	3/3/3/3
	6.2 FAR	2/4/-/1	3/3/-/3
	Shells (по 25%)	3	3
5.	Premiere 5.5	2/3/1/1	4/4/4/3
6.	10.1 Photoshop 3.0	2/3/2/1	3/3/3/3
	10.2 WN	3/2/-/2	3/3/-/3
7.	Power Point 97	1/1/1/1	3/4/3/3
8.	14.1 Excel 97	1/1/1/1	3/3/4/3
	14.2 NC	1/1/-/1	3/3/-/3

Таблица 4

Коды классификации

	Степень загрузки ОП:			
Степень загрузки ЦП:	Критическая	Высокая	Средняя	Малая
Критическая	1	2	3	4
Высокая	5	6	7	8
Средняя	9	10	11	12
Малая	13	14	15	16

В табл. 5 представлены результаты измерения сетевой нагрузки, где все виды работ с приложением выполняются на рабочей станции, но с сервера. Открывается файл также с сервера и сохраняется опять на сервер.

Таблица 5

Результаты измерения сетевой нагрузки

Приложение	Processor: % ProcessorTime (в процентах)	Server: BytesTotal/sec (Kb/s)	Время передачи (в сек)
Word:			
1) открытие приложения	3	117	8
2) открытие файла	3	1958	5
3) сохранение файла	15	7386	4
4) закрытие приложения	2	65	2
Access:			
1) открытие приложения	12	3078	4
2) открытие файла	3	3184	3
3) создание БД	11	7431	22
4) закрытие приложения	2	198	2
Excel:			
1) открытие приложения	3	117	6
2) открытие файла	7	8828	6
3) сохранение файла	5	6857	4
4) закрытие приложения	1,5	11,7	2
Power Point:			
1) открытие приложения	2,5	13,2	2
2) открытие файла	7	6751	3
3) сохранение файла	5	7366	4
4) закрытие приложения	2,5	11	1.5
3D Studio MAX:			
1) открытие приложения	30	6191	37
2) открытие файла	17	7251	4
3) сохранение файла	28	8315	4
4) закрытие приложения	12	1238	2
Dos Navigator:			
1) открытие приложения	2,5	1380	2
2) закрытие приложения	2	1096	3,5
Win Navigator:			
1) открытие приложения	17	5176	5
2) закрытие приложения	1	216	2

Анализируя полученные результаты, можно сделать следующие выводы:

- наиболее ресурсоемкими являются приложения баз данных, векторные и графические редакторы;

- наименьшую загрузку создают файловые менеджеры и электронные таблицы;
- базы данных предъявляют высокие требования и к оперативной памяти, и к центральному процессору, в то время как для векторных приложений более важным параметром является мощность процессора, для графических редакторов – размер оперативной памяти.

На основе полученных данных могут формироваться смеси приложений в зависимости от времени работы в каждом из них, и, следовательно, можно определить общую загруженность системы в течение дня, недели, месяца.

Смеси приложений можно представить следующим образом. Предположим, что на открытие и закрытие приложения уходит в среднем по 2% общего времени работы с приложением, на сохранение результатов работы – 14% и непосредственно на работу в приложении – 82% времени. Тогда, например, загрузка рабочих станций приложениями MS Office составит:

Word

$$\text{Загрузка ЦП} = 0,02 \times 99 + 0,82 \times 94 + 0,14 \times 90 + 0,02 \times 32 = 92,3$$

$$\text{Загрузка ОП} = 0,02 \times 70 + 0,82 \times 72 + 0,14 \times 75 + 0,02 \times 58 = 72,1$$

Excel

$$\text{Загрузка ЦП} = 0,02 \times 13 + 0,82 \times 3 + 0,14 \times 4 + 0,02 \times 8 = 3,44$$

$$\text{Загрузка ОП} = 0,02 \times 69 + 0,82 \times 72 + 0,14 \times 76 + 0,02 \times 72 = 72,5$$

Access

$$\text{Загрузка ЦП} = 0,02 \times 100 + 0,82 \times 100 + 0,14 \times 100 + 0,02 \times 14 = 98,8$$

$$\text{Загрузка ОП} = 0,02 \times 76 + 0,82 \times 82 + 0,14 \times 81 + 0,02 \times 64 = 81,38$$

Power Point

$$\text{Загрузка ЦП} = 0,02 \times 8 + 0,82 \times 10 + 0,14 \times 14 + 0,02 \times 13 = 10,6$$

$$\text{Загрузка ОП} = 0,02 \times 73 + 0,82 \times 79 + 0,14 \times 73 + 0,02 \times 62 = 77,7$$

Предположим, что средний пользователь работает с MS Word 50% общего времени работы за компьютером, с MS Excel – 40%, с MS Access и Power Point – по 5%. Тогда общая загрузка при работе пользователя будет определяться следующим образом:

$$\text{Загрузка ЦП} = 0,5 \times 92,3 + 0,4 \times 3,44 + 0,05 \times 98,80 + 0,05 \times 10,6 = 53$$

$$\text{Загрузка ОП} = 0,5 \times 72,1 + 0,4 \times 72,5 + 0,05 \times 81,38 + 0,05 \times 77,7 = 73$$

Таким образом, предложенный подход заключается в измерении рабочей нагрузки, создаваемой различными типами приложений в уже существующих сетях, анализе и структуризации полученных данных, дальнейшей классификации рабочей нагрузки, разработке методики применения полученных результатов в качестве данных для моделей при проектировании вычислительных сетей.

Литература

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. СПб: Питер, 1999.
2. Андерсон Криста, Минаси Майк. Локальные сети. Полное руководство. СПб: КОРОНА, 1999.

КОНЦЕПТУАЛЬНЫЕ МОДЕЛИ В БАЗАХ ЗНАНИЙ

К.К. Боярский, Е.А. Каневский, Г.В. Лезин

Рассматривается подход к построению формализованного описания предметной области методами концептуального моделирования текстов. Текущее состояние описания предметной области представлено в базе знаний сетью отношений между образующими ее объектами (понятиями и их экземплярами). Сеть задается значениями предикатов, сопоставляемых каждому из объектов. Допустимые значения объектных предикатов определяются на основе фактов, извлекаемых из концептуальных моделей текстов и правил. Правила, управляя поиском информации в концептуальной памяти, являются вместе с тем неотъемлемой частью описания предметной области.

Введение

Новейшие информационные технологии создают предпосылки новых методологических подходов к процессу анализа текстовых источников. Становится возможным представить содержание текстов в виде моделей, доступных для автоматизированной обработки. В совокупности модели текстов и алгоритмы их анализа могут оформляться в виде базы знаний о предметной области исследования. Во второй половине 90-х гг. исследования в этом направлении были начаты в СПб ЭМИ РАН и СПбГИТМО (ТУ). Отправной точкой исследований было создание системы управления базой знаний, позволяющей накапливать исходные тексты, их формальные описания, указатели к этим текстам и словари понятий, а также строить сеть ассоциативных связей между текстами. Была разработана исходная версия экспериментальной системы **MAZE** [1–3], предназначенной для поддержки различного рода текстологических исследований. Система позволяет накапливать и обрабатывать информацию трех типов:

- тексты и формальные описания их в текстовой форме,
- концептуальную информацию,
- словари и предметные указатели к текстам.

Информация двух первых типов в картотеке представляется совокупностью ее карточек [1]. Структура карточки, способ разбиения отображаемой ею информации на отдельные элементы определяются ее макетом. Для каждого из элементов макета указываются его обозначение, положение на карточке и тип фиксируемой в нем информации: текст или концептуальная информация. Возможны следующие виды отношений между карточками:

- карточке может быть сопоставлен набор других карточек, т. е. определена рубрика;
- карточка может входить в различные рубрики;
- карточки могут образовывать сеть с именованными и направленными ребрами.

В любом тексте на карточке можно комбинацией цвета, подчеркивания и начертания графем выделить то или иное сочетание знаков для уточнения интерпретации этого текста, которые могут быть зафиксированы в словарях системы.

Рассматриваемые в этой работе средства общения с базой знаний определены в виде языка концептуального моделирования текстов.

Концептуальной моделью (схемой) данных обычно называют абстрактное описание фрагмента реальной действительности, которое получается в результате выделения свойственных этому фрагменту понятий, указания их атрибутов и возможных связей между понятиями [4, 5]. Концептуальной моделью описывается структура данных, подлежащих накоплению в информационной системе. Для традиционных баз данных концептуальная модель разрабатывается на начальном этапе проектирования информационной системы, затем служит основой для ее физической реализации и остается неизменной в процессе накопления и обработки данных.

Построение формализованного описания предметной области гуманитарного исследования нуждается в совершенно иной технологии концептуального моделирования. Формирование понятийного аппарата предметной области здесь является вполне

самостоятельной задачей гуманитарного исследования. Стирается технологическая грань между информацией о понятиях области исследований и конкретными данными, представляющими эти понятия. Информация о новых понятиях и уточнение ранее введенных понятий поступают в систему в одном потоке с конкретными данными и становятся вполне равноправными с ними объектами обработки.

В нашем подходе к определению языка мы руководствовались следующим блоком требований к технологическому оформлению средств взаимодействия с базой знаний.

- Картотека интегрирует в единый комплекс средства гипертекста, филологический инструментарий (построение тезаурусов и лексических указателей, определение семиотических сопоставлений) и средства описания семантики накапливаемых в ней текстов.
- Концептуальная модель предметной области гуманитарного исследования собирается системой из концептуальных моделей отдельных текстов на карточках картотеки.
- Определения понятий, необходимых для моделирования семантики конкретного текста, строятся и уточняются по мере необходимости в процессе поступления этих текстов.

Концептуальные модели текстов, написанные на формальном языке, содержат фактографическую информацию и правила ее обработки. Поступающая в сообщениях информация накапливается в концептуальной памяти системы. Обработка полученной информации инициализируется запросами к концептуальной памяти. Тексты формализованных сообщений представлены в концептуальной памяти в нормализованной (канонической) форме: концептуальная память, наряду со сведениями, извлекаемыми из сообщений, фиксирует также место сообщения в картотеке.

Далее в работе рассматриваются структура и методы обработки формализованных сообщений. В качестве примера будем анализировать смысл следующего предложения русского языка:

Президент Азербайджана посетит штаб-квартиру ООН.

1. Объекты

Формальное сообщение записывается последовательностью предложений формального языка. Предложение может содержать утверждения об отношениях между объектами концептуальной модели, иметь форму правила, устанавливающего зависимости между отдельными классами утверждений, либо представлять собой запрос к базе знаний. Предложения отделяются одно от другого знаком `;`.

Объектами в концептуальной модели представляются сущности моделируемой предметной области. В сообщении объекты задаются или непосредственно своими обозначениями, или автономно, в соответствии с соглашениями о некоторых видах записи отношений между ними. Сообщения, будучи локализованными на карточках, тем не менее, представляют собой фрагменты общей модели предметной области. В связи с этим предусмотрена возможность использования общих и локальных обозначений объектов. Одно и то же общее обозначение в разных сообщениях указывает один и тот же объект. Это же обозначение, будучи локальным, в разных сообщениях задает объекты, различающиеся между собой и отличные от объекта, заданного общим обозначением. Обозначения задаются цепочкой букв и цифр алфавита системы, включая знак '_'. Локальное обозначение образуется из общего добавлением суффикса '#'. На объект, заданный локальным обозначением, можно ссылаться в других сообщениях, используя для этой цели общее составное обозначение **объект#имя_карточки**.

Примеры обозначений: **дом; Визит; B1; B1#**.

Первые три обозначения – общие. Обозначение 'B1' использовано для указания двух объектов: один из них задан общим обозначением, другой – локальным. Допускается определение следующих видов объектов: понятия (классы, множества), экземпляры понятий, связи, карточки, содержащие формальные сообщения, числа. Вид объекта явно

указывается в его обозначении. Принято соглашение, в соответствии с которым общие обозначения используются для указания понятий, а локальными задаются объекты-экземпляры. Текущее состояние объектов концептуальной памяти описывается наборами предикатов. Обозначение переменной в предикате отличается от обозначения объекта наличием префикса '!'. В обозначении переменной явным образом указывается область ее определения, например:

- **!обозначение** – числа, понятия и экземпляры концептуальной памяти;
- **!обозначение#** – экземпляры.

2. Отношения

Атрибуты. Объект-понятие представляет в концептуальной памяти множество элементов, имеющих общие атрибуты. Записью **Е (Е1)** мы утверждаем факт наличия атрибута **Е1** у объекта **Е**. Утверждение **Е (Е1,...,Еk)** задает для объекта **Е** список атрибутов. В качестве атрибутов могут указываться лишь объекты типа "понятие":

Визит (Кто, Куда)

Отметим, что список атрибутов никогда не считается завершенным и может пополняться по мере поступления новых сообщений. Текущее состояние списка атрибутов понятия **Е** определяется предикатом **Е (!X)**. В предыдущем примере, в частности, предикат

Визит (!X)

истинен для **!X = Кто, Куда**. Предикат **!X (Е)** позволяет установить наличие в концептуальной памяти понятий, для которых **Е** определено в качестве атрибута. В нашем примере **!X (Кто)** истинен для **!X = Визит**.

Конкретизации. Факт принадлежности понятия **Е1** или экземпляра **V#** понятию **Е** устанавливается отношением **Е:Е1** и, соответственно, **Е:V#**. Далее мы будем говорить в таких случаях, что объект **Е1** конкретизирует объект **Е** (является его подклассом), объект **V#** – конкретный пример, экземпляр класса **Е**; само отношение мы будем называть отношением конкретизации.

Одно понятие может конкретизировать несколько других понятий, и факты, устанавливающие отношение конкретизации, могут поступать в разных сообщениях. Предикаты **!X:Е** и **Е:!X** определяют текущее состояние отношения конкретизации для заданного понятия **Е** в концептуальной памяти. Отношение конкретизации транзитивно, но не ассоциативно: если **Е1:Е2**, **Е2:Е3**, то **Е3** конкретизирует **Е1**. Объект **Е1**, конкретизируя **Е**, "наследует" его атрибуты. Конкретизация любого понятия – это результат уточнения его атрибутов. Записями **Е (Е1.Е2)** мы утверждаем факт, согласно которому:

- понятие **Е** конкретизирует некое, не указанное в записи и, может быть, отсутствующее в концептуальной памяти, понятие с атрибутом **Е1**;
- атрибут **Е1** для всех конкретизаций понятия **Е** конкретизирован значением **Е2**.

Ниже приведен пример объекта, имеющего атрибут с конкретизацией:

Президент (Страна.Азербайджан)

Участие понятия **Е** в отношениях, уточняющих значения атрибутов, характеризуется предикатами **Е (!X.!Y)**, **!X (Е.!Y)** и **!X (!Y.Е)**.

Связки. Предусмотрена возможность пополнения исходного списка отношений между объектами. Примерами могут быть разновидности отношений "причина-следствие" (**Е1 "влечет" Е2**) или временные отношения между объектами (**Е1 "затем" Е2**). Для любого заданного объекта **Е** концептуальной памяти в ее текущем состоянии предикаты **Е !"X" !Y** и **!Y !"X" Е** определяют связки (значения **!"X"**), операндом которых является объект **Е**, и объекты, связанные с **Е** этими связками. Отношения, заданные связками, не ассоциативны.

Концептуализации. Часто бывает необходимо описать структуру элементов, составляющих понятие, указав объекты, образующие эту структуру, и отношения

между ними. В других случаях бывает важно рассматривать наборы утверждений об объектах как целостный объект с присущей любому объекту способностью вступать в отношения с другими объектами. Конструкция

Объект {набор предложений}

предназначена для моделирования именно таких ситуаций. Набор предложений в фигурных скобках мы, следуя Шенку [6], будем называть концептуализацией соответствующего объекта, а отношения, заданные набором, – отношениями вхождения в концептуализацию. Предикаты типа $E \{!X\}$, $E \{!X:!Y\}$, $E \{!X (!Y.!Z)\}$ определяют объекты и отношения, входящие в концептуализацию E , а предикаты типа $!X \{E\}$, $!X \{E:!Y\}$, $!X \{E (!Y.!Z)\}$ и т. п., истинны для концептуализаций, содержащих отношения, связанные с E .

3. Визуализация семантических отношений

На формальном языке смысл анализируемого нами предложения может быть отображен через понятие 'Визит' в виде:

Визит: B1 (Кто.Президент, Куда.ООН);

Президент (Страна.Азербайджан);

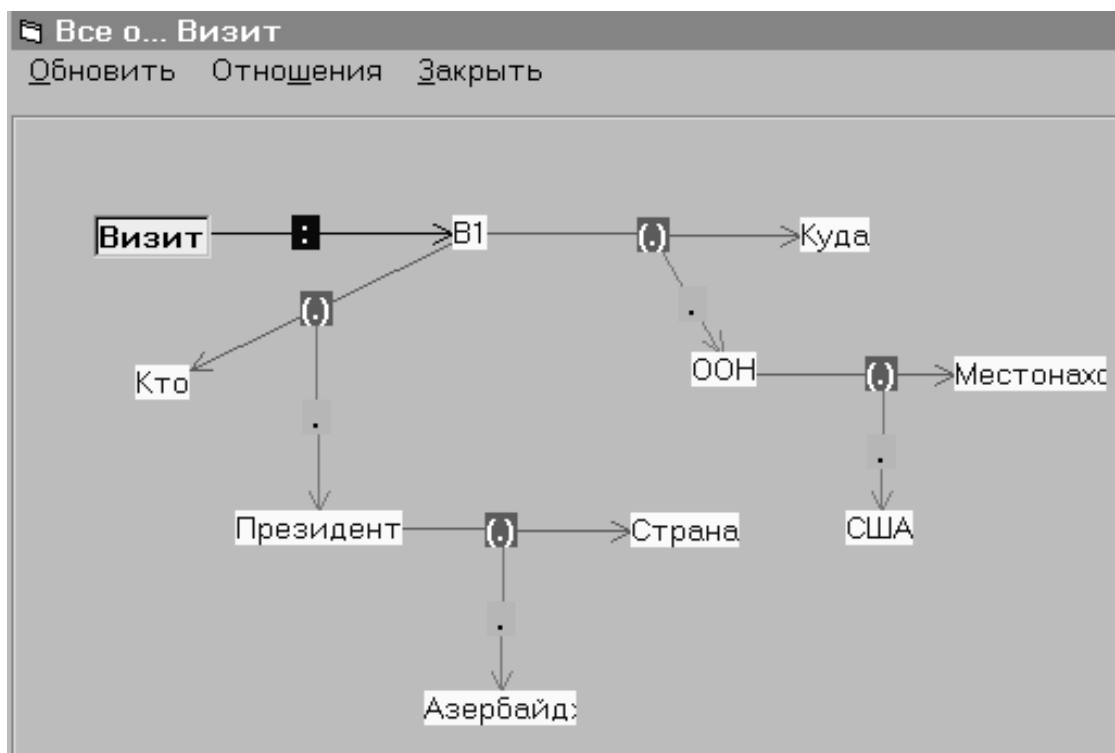


Рис. 1. Семантическая сеть

Дополним эту информацию формальным сообщением, описывающим данную предметную область:

ООН (Местонахождение.США)

В системе **MAZE** предусмотрена возможность визуального отображения содержимого концептуальной памяти в виде семантической сети объектов и их отношений. Поскольку полная сеть может содержать сотни и тысячи объектов, построение сети начинается с задания исходного объекта и формирования непосредственно примыкающего к нему фрагмента сети. В дальнейшем возможно наращивание отображаемого участка сети путем указания детализируемых объектов. В нашем случае полная семантическая сеть, построенная от вершины 'Визит', имеет вид, изображенный на рис. 1.

4. Запросы и правила

Запросы к концептуальной памяти имеют форму предложений, помеченных знаком '?'. Например, предложение

? Президент

интерпретируется как вопрос к концептуальной памяти: имеется ли в ее составе объект, обозначенный как 'Президент'. На этот вопрос возможен ответ лишь в форме "да" или "нет". Более интересен запрос, имеющий форму предиката, устанавливающего отношения между объектами. Имея в концептуальной памяти модель указанного выше текста можно, в частности, поинтересоваться, кто и из какой страны наносил какой-нибудь визит:

? Визит: !X (Кто.!Y); !Y (Страна.!Z);

В качестве ответа на этот запрос получаем следующие значения переменных:

!X = В1, !Y = Президент, !Z = Азербайджан.

Правила. Далеко не всегда ответ на интересующий нас запрос может быть получен в результате непосредственного вычисления указанных в нем объектных предикатов. Нас может интересовать, например, название страны, в которую совершался визит. Для ответа на этот вопрос необходимо усложнить поиск, указав правило, следуя которому можно установить соответствие между организацией, в которую наносится визит и соответствующей страной:

** !X (Куда.!Z) <- (!Y (Местонахождение.!Z); !X (Куда.!Y))

Имея в концептуальной памяти это правило, мы по запросу

? В1 (Куда.!Z);

получаем два ответа:

!Z = ООН; !Z = США.

При этом второй ответ получается при использовании правила, в процессе исполнения которого меняется цель поиска со следствия правила на его посылку.

Отработка правил и переключение целей поиска производится системой логического вывода [7]. Используя прологopodobные механизмы вывода, мы базируемся на совместном хранении фактов и пользовательских правил в общей концептуальной памяти. Общий список констант, на основе которого строятся факты и правила концептуальной памяти, объединяет обозначения понятий, их экземпляров и введенных пользователем бинарных отношений. Атомарный элемент хранения в концептуальной памяти – это либо константное значение отношения, либо предикат отношения в том или ином правиле пользователя. Правила операционной семантики отношений в виде образцов, независимых от конкретного наполнения базы знаний, являются принадлежностью системы логического вывода. Построение конкретных правил на основе образцов и их последующее применение осуществляются в процессе доказательства запроса к базе знаний. Возможность применения этих правил определяется фактами, хранящимися в концептуальной памяти.

Изложенный здесь подход позволяет:

- обрабатывать запросы к базе знаний в двух режимах: либо используя только правила пользователя (область поиска доказательства существенно сужается, а быстродействие столь же существенно возрастает), либо с учетом семантики отношений;
- отделить работу по определению понятий и правил пользователя от работы по определению формальной семантики отношений;
- снизить затраты на представление информации в базе знаний для тех задач, в которых объем информации по определению понятий и правил соизмерим с объемом информации о принадлежности экземпляров понятиям.

Заключение

Согласно рассмотренной технологии построения формализованного описания предметной области гуманитарного исследования понятийное пространство формируется и

уточняется на основе концептуальных моделей текстов и по мере их поступления. Текущее состояние описания предметной области представлено сетью отношений между образующими ее объектами (понятиями и их экземплярами). Сеть отношений задается допустимыми значениями предикатов, сопоставляемых каждому из объектов. Допустимые значения объектных предикатов определяются на основе фактов, извлекаемых из концептуальных моделей текстов и правил. Правила, управляя поиском информации в концептуальной памяти, являются вместе с тем неотъемлемой частью описания предметной области.

Разработанный для системы MAZE язык концептуального моделирования отличается рядом важных новых возможностей [1, 3].

- Техника логического программирования на концептуальных схемах данных в этом языке определена как средство фиксации частных теорий и подборки иллюстрирующих теорию фактов. Фактами в предлагаемом подходе к программированию являются отношения между самими понятиями, а также между понятиями и конкретными объектами (индивидами). Алгоритмы обработки данных задаются наборами правил, устанавливающих зависимости между понятиями и их отношениями.
- В основе языка лежит фиксированный набор терминальных отношений между понятиями и индивидами. Каждому из понятий и конкретных объектов сопоставлено уникальное обозначение. Предикатом отношения называется функция, принимающая значение “истина” или “ложь” на кортежах значений аргументов этого отношения. Аргумент предиката определяется либо на общем множестве обозначений понятий и индивидов, либо на множестве индивидов. Правило имеет вид $C \leftarrow A$, где C и A – конъюнкции предикатов отношений. Значение предиката отношения может быть задано соответствующим фактом, либо вычислено в результате применения правил. В системе реализован нисходящий метод SLD-резолюции [8].
- Каждому из понятий или конкретных объектов в концептуальной памяти сопоставлены все факты и правила, в которых соответствующий объект упоминается. Обозначения объектов служат ключами при подборке фактов и правил, что позволяет существенно ускорить процедуры поиска.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований, грант № 00-06-80468.

Литература

1. Лезин Г.В., Боярский К.К., Каневский Е.А., Попова А.И. Анализ текстов: представление и обработка концептуальной информации // Труды Международного семинара Диалог'97 по компьютерной лингвистике и ее приложениям. М.: РосНИИ ИИ, 1997. С. 170–174.
2. Lezin H.V., Boyarsij K.K., Kanevskij E.A., Popova A.I. Programming of texts conceptual treatment // Manuscripta Orientalia, 1997, v.3, №2, P. 42-48.
3. Боярский К.К., Каневский Е.А., Лезин Г.В., Попова А.И. Формализация знаний в гуманитарных исследованиях // Экономико-математические исследования: математические модели и информационные технологии. СПб.: Наука, 2000. Вып. 1. С. 248–264.
4. Цаленко М.Ш. Моделирование семантики в базах данных. М.: Наука, 1989. 287 с.
5. Цикридис Д., Лоховски Ф. Модели данных. М.: Финансы и статистика, 1985. 344 с.
6. Шенк Р. Обработка концептуальной информации. М.: Энергия, 1980. 360 с.
7. Лезин Г.В., Мамедниязова Н.С. О представлении семантики концептуальных моделей в базах знаний // Труды Международного семинара Диалог'2000 по компьютерной лингвистике и ее приложениям. М.: РосНИИ ИИ, 2000. С. 235–242.
8. Боярский К.К., Мамедниязова Н.С., Попова А.И. Представление и обработка концептуальной информации в системе MAZE // Экономико-математические исследования: математические модели и информационные технологии. СПб.: Наука, 2000. С. 264–279.

РАЗРАБОТКА БАЗЫ ДАННЫХ ПО МЕТАЛЛИЧЕСКИМ МАТЕРИАЛАМ ДЛЯ УНИВЕРСИТЕТА

К.Е. Вурья, О.А. Приходько

Формируется база данных по металлическим материалам, в основу которой заложен принцип поиска материалов по их назначению. В базе предусмотрена возможность ее пополнения и трансформации с целью более удобного и информативного представления данных, а также расширения классификации материалов. База построена с использованием языка и среды программирования Delphi, относящейся к классу RAD- (Rapid Application Development - Средство быстрой разработки приложений) средств CASE - технологии.

В связи с постоянным ростом объемов используемой информации всегда существует проблема организации записи, систематизации, хранения и использования данных. В области приборной техники эта проблема связана с необходимостью накопления экспериментальных данных, получаемых в ходе научного исследования, данных о контролируемых параметрах технологических процессов, данных, полученных при, различного рода экспертизах, мониторинге окружающей среды и т. д. Накопление и эффективный поиск данных при проектировании и изготовлении приборов позволяют ускорить разработку конструкций и выбор технологических процессов. В частности, в процессе проектирования всегда возникает проблема выбора материалов, в том числе и конструкционных. Сведения о материалах, присутствующие в справочниках, часто неудобны для использования. Существующие компьютерные базы данных (БД) по материалам отсутствуют в бесплатном доступе, построены на основе устаревших программных средств и слабо ориентированы на выбор материала, исходя из его назначения и условий работы. Поэтому есть необходимость в создании базы данных по материалам, использующей современные программные средства, удобной для пользователя, ориентированной на выбор материала, исходя из условий применения, имеющей возможность контролируемого пополнения, удобной для использования в учебном процессе университета.

Использование значительных объемов хранимой информации, помимо развития системных устройств, средств передачи данных, памяти, потребовало совершенствования средств обеспечения диалога "человек – ЭВМ", которые позволяют пользователю вводить запросы, читать файлы, модифицировать хранимые данные, добавлять новые данные или принимать решения на основании хранимых данных. Для обеспечения этих функций созданы специализированные средства – системы управления базами данных (СУБД), которые поддерживают все более сложные уровни абстрактных данных, заданных пользователем, и обеспечивают взаимодействие компонентов, распределенных в глобальных сетях и постепенно интегрирующихся с телекоммуникационными системами. СУБД являются многопользовательскими системами, специализирующимися на управлении массивом информации одним или множеством одновременно работающих пользователей.

Современные СУБД обеспечивают:

- набор средств, используемых для поддержки таблиц и отношений между связанными таблицами;
- развитый пользовательский интерфейс, который позволяет вводить и модифицировать информацию, выполнять поиск и представлять информацию в графическом или текстовом режиме;
- средства программирования высокого уровня, с помощью которых можно создавать собственные приложения.

Последним шагом в этом направлении стала объектно-ориентированная технология. Объектно-ориентированный подход позволяет упаковывать данные и код для их обработки вместе. Используемые сегодня реляционные базы данных приобрели механизм запросов пользователя, предоставив СУБД самой оптимальным образом найти результат, используя

динамическую индексацию. Одним из самых современных средств программирования, позволяющих создавать базы данных, является Delphi – язык и среда программирования, относящаяся к классу RAD (Rapid Application Development – Средство быстрой разработки приложений) средств CASE-технологии. Интерфейс Windows обеспечивает полное перенесение CASE-технологий в интегрированную систему поддержки работ по созданию прикладной системы на всех фазах жизненного цикла работы и проектирования системы [1].

Среда устраняет необходимость программировать такие компоненты Windows общего назначения, как метки, пиктограммы и даже диалоговые панели. Это наглядная реализация применений CASE-технологий в современном программировании приложений. Та часть, которая непосредственно связана с программированием интерфейса пользователя системой, получила название визуального программирования

Визуальное программирование дает возможность изображать объекты на экране монитора до выполнения самой программы. Благодаря средствам визуальной разработки можно работать с объектами, держа их перед глазами и получая результаты практически сразу. Способность видеть объекты такими, какими они появляются в ходе исполнения программы, снимает необходимость проведения множества операций вручную, что характерно для работы в среде, не обладающей визуальными средствами, вне зависимости от того, является она объектно-ориентированной или нет. После того, как объект помещен в форму среды визуального программирования, все его атрибуты сразу отображаются в виде кода, который соответствует объекту как единице, исполняемой в ходе работы программы. Данная среда программирования и была использована при создании базы данных по материалам на основе справочной литературы, согласованной с государственной службой стандартных справочных данных [2–4].

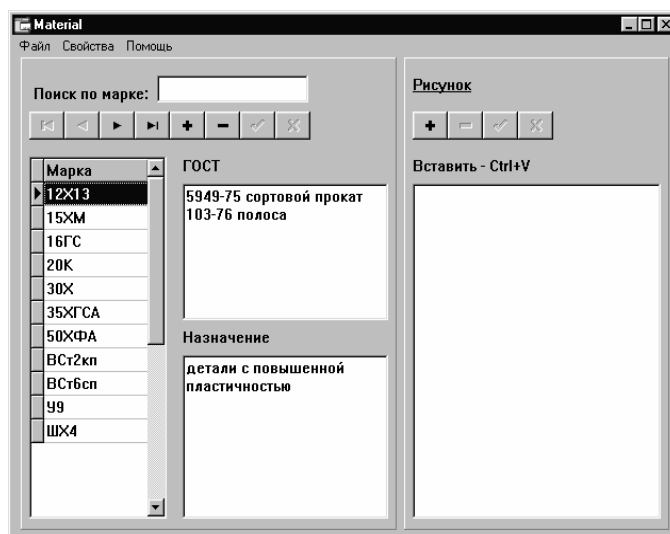


Рис. 1. Окно "Управляющее меню"

После загрузки базы данных по материалам открывается окно программ, имеющее свойства, присущие другим окнам Windows: управляющее меню (рис. 1), полоса заголовка, кнопки максимизации, минимизации и закрытия окна в верхнем правом углу. "Управляющее меню" содержит, в свою очередь, несколько дочерних меню, таких как: "Файл", "Свойства" и "Помощь". Меню "Файл" обеспечивает доступ к основным командам базы данных. В него включены команды "Печать" и "Выход". Меню "Свойства" включает в себя набор дочерних окон, таких как "Эксплуатационные свойства", "Физико-химические свойства", "Технологические свойства", "Химический состав" и "Поиск". Меню "Помощь" в данном программном комплексе содержит только раздел "О программе". В дальнейшем планируется создать полное меню "Помощь" с полным объяснением правил пользования программным комплексом, а также способы устранения некоторых возможных неполадок. Кроме этого, в окне программ располагаются: диалоговое окно поиска по марке

металлического материала и рабочие окна, в которые выводится информация о марке материала, ГОСТы (вида поставок, сортамента и др.), его назначение [2–4]. Имеется поле, в котором может располагаться какое-либо изображение, фотография или рисунок, например, микроструктуры. Также предусмотрена панель инструментов, на которой расположены клавиши управления окнами (добавление, удаление, перемещение к первому, последнему, предыдущему и последующему объекту и сохранение введенных данных).

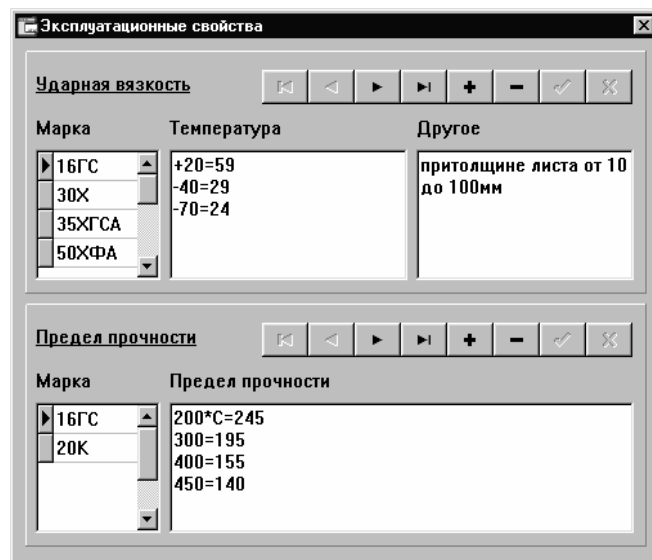


Рис. 2. Окно меню "Эксплуатационные свойства"

Меню "Свойства" содержит следующие дочерние окна.

- "Физико-химические свойства" – окно, содержащее физико-химические свойства металлических материалов, такие как упругие константы, теплопроводность, теплоемкость, коэффициент теплового расширения и другие. Дочернее окно, также как и главное, имеет панель инструментов.
- "Эксплуатационные свойства" – окно (рис. 2), содержащее эксплуатационные свойства, такие как механические свойства ($\sigma_{0,2}$ – предел текучести, σ_b – временное сопротивление (предел прочности при растяжении), δ_5 – относительное удлинение после разрыва, ψ – относительное сужение после разрыва, НВ – твердость по Бринеллю и КСU, V, T – ударная вязкость, зависимость прочности от температуры), надежность, долговечность и др.
- "Технологические свойства" – здесь (рис. 3) размещаются сведения о технологических свойствах материала: марка, температураковки, свариваемость, обрабатываемость резанием, флокеночувствительность, склонность к отпускной хрупкости, а также сортамент и цена. Это окно также имеет панель инструментов.

По желанию пользователя в окне "Химический состав" (рис. 4) выводится информация о химических элементах содержащихся в материале в процентах. Там же содержится информация о марке, ГОСТе.

В окне "Поиск" (рис. 5) производится поиск материалов по назначению и ГОСТам по всей базе данных. В дальнейшем предполагается включить поиск по механическим, физико-химическим и технологическим свойствам, а также по химическому составу

Анализ представленной работы позволяет сделать следующие выводы:

- находящаяся в стадии разработки и заполнения база данных по металлическим материалам на основе Delphi явится эффективным средством выбора материалов;

- использование базы данных позволит улучшить информационное качество учебного процесса как на этапе изучения курса материаловедения, так и при выполнении курсовых и дипломных проектов;
- в дальнейшем необходимо провести усовершенствование данного программного комплекса с целью расширения его возможностей и по вводимой информации, и по поиску материалов, уделяя основное внимание выбору по назначению.

Марка	Температура ковки	Свариваемость
12X13	начала 1230°C конца 850°C	ограниченно свариваемая
15ХМ		
16ГС		
20К		
30Х		
35ХГСА		
50ХФА		
ВСт2кп		
ВСт6сп		
У9		

Обрабатываемость резанием
в закаленном и отпущенном состоянии при HB 235

Флокеночувствительность

Склонность к отпускной хрупкости
склонна

Цена

Рис. 3. Окно меню "Технологические свойства"

Марка	C	Si	Mn	Cr	S	P	Cu	Ni	As
12X13	0,09-0,8	0,8	12-14	0,02-0,03	0,3	0,6			
15ХМ	0,11-0,17	0,4-0,8	0,8-1,1	0,03-0,035	0,3	0,3			
16ГС	0,12-0,44	0,9-1,3	0,04	0,035	0,3	0,3	0,08		
20К	0,16-0,15	0,35-0,3	0,04	0,04	0,3	0,3	0,08		

Рис. 4. Окно меню "Химический состав"

Поиск

Поиск

Параметр

☐ По ГОСТу

☐ По назначению

Марка

Рис. 5. Окно меню "Поиск"

Литература

1. Оузьер Д., Гробман С. Самоучитель по Delphi 3. М., 1998.
2. Раскатов В.М. Краткий справочник по машиностроительным материалам. М.: Машиностроение, 1980.

3. Сорокин В. Г. Марочник сталей и сплавов. М.: Машиностроение, 1989.
4. Арзамасов Б.Н. Справочник по конструкционным материалам. М.: Машиностроение, 1990.

ТЕХНОЛОГИЯ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ В ВИЗУАЛИЗАЦИИ СОБРАНИЙ ДАННЫХ

М.Г. Рождественская, И.В. Рождественский

Введение

Цель "добычи данных" (data mining) и "поиска знаний" (knowledge discovery) состоит в "получении полезных знаний из больших собраний данных" (Mannila, 1997). Пользовательский интерфейс собрания данных должен удовлетворять совокупности противоречивых требований: унификация различных форматов представления данных, поддержка эвристического поиска, поддержка внутренней иерархии, алгоритмируемость, высокая плотность информации, поддержка существующих Интернет-стандартов и др. Традиционные системы на базе текста и простой двумерной графики решают эти проблемы только частично. Виртуальная реальность (ВР) имеет значительный потенциал в области визуализации научной информации. Основными преимуществами ВР (Waterworth, 1996) являются: естественное пространственное поведение при поиске информации; подсознательная ориентация; автоматизм; компромисс между общим видом и детализацией; визуализация связей; мнемонизация поиска; индивидуальная настройка ВР-представления; переносимость; масштабируемость; богатство уже существующих метафор. Целью настоящей работы является описание концепции визуализации генетической базы данных в виде коллекции объектов на виртуальном ландшафте.

SequenceWorld: ВР-представление генетической базы данных

Целью проекта SequenceWorld (Rojdestvenski et al., 2000, Рождественская и др., 2000) является применение способности пользователя к навигации в пространстве и сенсомоторного восприятия к работе с генетической базой данных. В рамках данного проекта создан новый подход к анализу данных в генетической базе данных с использованием элементов технологии и философии ВР – виртуальный ландшафт данных. Записи базы данных представляют собой 1106 EST последовательностей из БД Populus. Последовательности аннотированы по ближайшим гомологам из БД SwissProt (SwissProt, 1995). В качестве ландшафта была использована плоскость в трехмерном пространстве, для кластеризации данных был применен метод Монте-Карло с постепенным охлаждением (Simulated Annealing). Для каждой пары записей введено искусственное "взаимодействие", параметром которого является мера гомологической близости двух генетических последовательностей. После этого система взаимодействующих "частиц" со статистикой Максвелла-Больцмана "охлаждалась" до достижения конденсации подобных частиц в кластеры.

Система кодирования цветом и формой базируется на аннотациях генетических последовательностей. Форма объекта, представляющего определенную генетическую последовательность осины, описывает функцию гена из БД SwissProt, гомологически наиболее близкого к данной последовательности осины. Цвет зарезервирован для кодирования организма происхождения этого гена. С демонстрационной версией генетической базы данных в ВР – SequenceWorld можно познакомиться на <http://130.239.110.23>. SequenceWorld представляет собой ограниченно функциональную БД, снабженную простыми средствами поиска и просмотра текстовой информации. Однако главной характерной чертой системы является представление базы в виде системы объектов на виртуальном ландшафте, по которому пользователь может перемещаться. Путешествие по виртуальному миру осуществляется с помощью просмотрщика VRML-файлов, наиболее удобным из которых является CosmoPlayer (компании CosmoSoftware). На рис. 1 приведена копия экрана SequenceWorld.

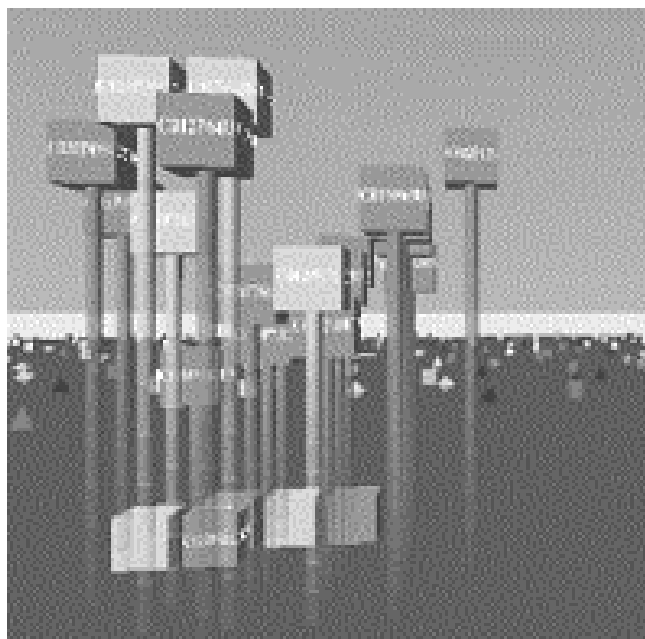


Рис. 1. "SequenceWorld" – внутри базы данных

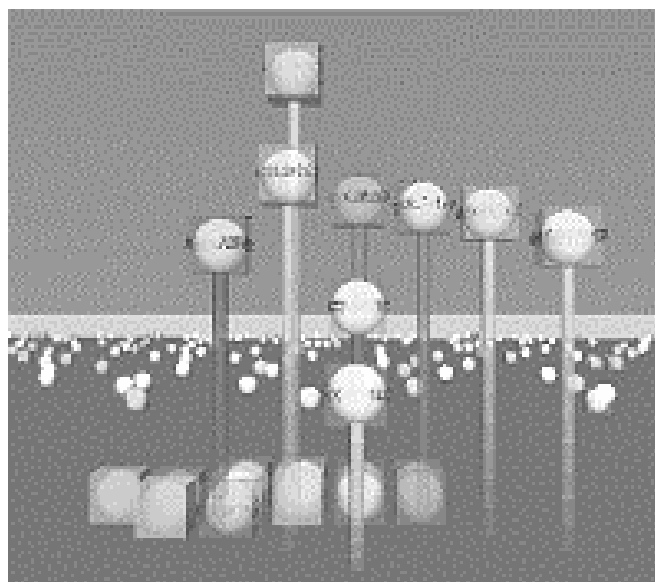


Рис. 2. Визуализация достоверности аннотирования в версии "Fading Annotations"

Для демонстрации некоторых возможностей концепции SequenceWorld была также разработана система автоматического аннотирования последовательностей. Основная идея – использование прозрачности как меры уверенности в правильности приписываемых последовательностям аннотаций. Подробное описание системы приводится в работах (Rojdestvenski et al., 2000, Рождественская и др., 2000). Концепция, продемонстрированная системой SequenceWorld, может оказаться эффективным для ряда традиционных задач, для решения которых используются базы данных. Например, исследователю можно предоставить возможность "запустить" неизвестную последовательность в виртуальный мир и наблюдать ее "движение" к "ее" кластеру. В результате такого "движения" по виртуальному миру последовательность может остановиться и вблизи нескольких кластеров сразу. Это будет означать, что

данная последовательность гомологична сразу нескольким "семьям" генов. Такого рода информация уже требует длительного поиска стандартными "текстовыми" методами.

VRML визуализатор систем биохимических реакций

При проектировании системы визуализации сетей биохимических реакций MNV (Metabolic Network Visualizer) мы основывались на постулате, что для адекватной визуализации некоторой системы необходимо, чтобы алгебра объектов и правил, которая описывает данную систему, взаимно однозначно соответствовала алгебре визуальных образов и правил визуализации. Для описания сетей биохимических реакций нами разработан формальный SGML-подобный язык MNV, гомологичный подмножеству недавно разработанного языка разметки системной биологии SBML (Bologni и др., 2000). Каждый тип объекта ассоциируется с HTML-подобным тегом, атрибуты которого описывают свойства объекта. Объект "compartment" описывает место, где находятся вещества и происходят реакции. Объект "species" (метаболит) описывает вещества, участвующие в реакциях, которые, в свою очередь, описываются объектом "reaction". Объект "transporter" (транспортер) введен для реакции по вводу метаболитов в область реакции и выводу метаболитов из нее. Для алгоритмизации трансляции описания сети реакций на MNV в трехмерный VR-мир мы применяем жесткие правила визуализации, с которыми можно познакомиться на <http://130.239.110.23>, также как и с демонстрационной версией системы.

Пример визуализации упрощенного цикла Кальвина приведен на рис.3.

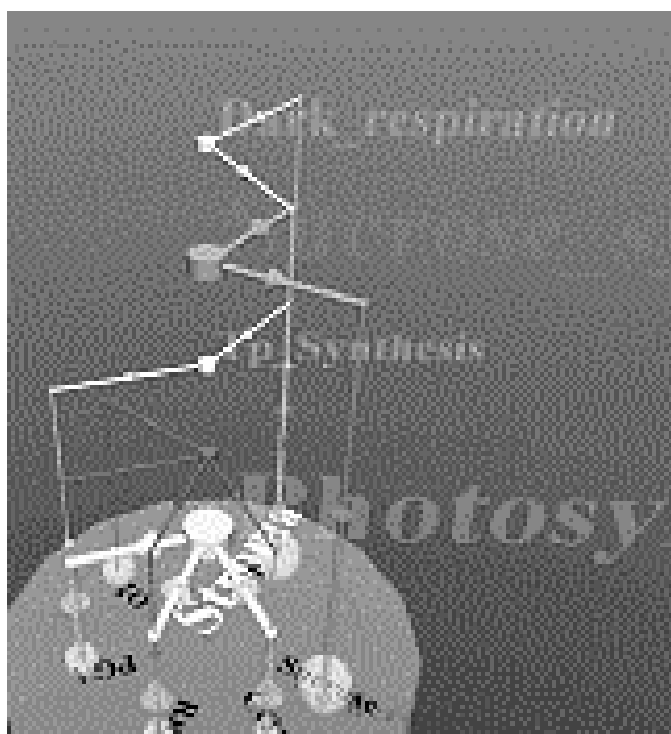


Рис. 3. Визуализация упрощенного цикла Кальвина

Пользователь может вращать картинку в трех измерениях, передвигаться и даже попадать "внутрь" схемы. Дополнительная функциональность достигается использованием гиперссылок, встроенных в VRML-мир. При сравнении с традиционными двумерными схемами сетей биохимических реакций становятся видны некоторые потенциальные преимущества предложенного нами подхода с использованием VRML. Во-первых, использование третьего измерения приводит к экономии поля зрения, необходимого для представления сети реакций, в сравнении с двумерными изображениями. Использование таких визуальных признаков, как форма,

цвет, размер и прозрачность, распараллеливает восприятие информации, "уплотняет" его. Во-вторых, подвижность пользователя в виртуальном мире обеспечивает возможность непрерывного сдвига фокуса внимания, а также компромисс между общим видом и детализацией. В-третьих, использование стандартного для Интернет языка VRML 2 делает работу по визуализации сетей биохимических реакций легко распространяемой и публикуемой.

В настоящее время системы SequenceWorld и MNV внедряются на кафедре физиологии растений университета города Умео, Швеция. При помощи MNV в рамках проекта Populus планируется создать наиболее полное виртуальное описание сетей биохимических реакций обмена веществ в клетках осины.

Литература

1. H. Mannila (1997). Methods and problems in data mining (a tutorial) // Proceedings of International Conference on Database Theory (ICDT'97), Delphi, Greece, January 1997, F. Afrati and P. Kolaitis (ed.), p. 41–55.
2. Waterworth J (1996). Personal Spaces: 3D Spatial Worlds for information Exploration, Organization and Communication / 3D on the Internet: Information, Images and Interaction, eds: R.Earnshaw, J.Vince, New York: Academic Press, pp. 97–118
3. Rojdestvenski, I., D. Modjeska, F. Pettersson, M. Rojdestvenskaia and P. Gustafsson, (2000). Sequence World: A Genetics Database in Virtual Reality // Information Visualization 2000, Лондон, Великобритания.
4. М. Рождественская, И. Рождественский, Д. Можека (2000). SequenceWorld: генетическая база данных в среде виртуальной реальности // Региональная Информатика -2000, С.-Петербург, декабрь 2000.
5. SwissProt Database WWW Gateway (1995), <http://www.expasy.ch>
6. Bolouri, Hamid, Andrew Finney, Michael Hucka, Herbert Sauro, John Doyle, Hiroaki Kitano (2000). The ERATO Systems Biology Workbench: an integrated environment for multiscale & multi-theoretic simulations of molecular biology // The First International Conference on Systems Biology (ICSB), Japan, Nov.14-16, 2000.

**ИСПОЛЬЗОВАНИЕ АППАРАТНЫХ ИНТЕРПРЕТАТОРОВ
ДЛЯ УПРАВЛЕНИЯ НЕОДНОРОДНЫМИ КОМПОНЕНТАМИ
МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ****Р.Р. Ковязин, А.Г. Чистяков**

В статье рассматривается способ проектирования встроенных систем на базе микропроцессорного ядра, включающего заказную схему и стандартный микропроцессор. Отдельное внимание уделено функциональности заказной структуры на базе ПЛИС, где в качестве управляющего ядра использован аппаратный интерпретатор. Приводятся характеристики реального устройства, разработанного на основе рассмотренных принципов.

Введение

Особенностью специализированных микропроцессорных систем, применяемых в промышленной автоматике (встроенные управляющие системы), является наличие неоднородных компонент [1]. В таких системах управление сводится к взаимодействию с гетерогенным набором объектов, куда входят: цифроаналоговые модули (ЦАП/АЦП), сетевые интерфейсы, каналы передачи данных и т.д. Информационное взаимодействие вычислительного ядра микропроцессорной системы с этими объектами в большинстве случаев осуществляется посредством специализированных интегральных контроллеров. Набор интерфейсов этих модулей довольно разнообразен, поэтому доступ к ним со стороны вычислительного ядра усложнен, что сказывается на общей производительности системы.

Существует два традиционных подхода к проектированию встроенных систем в вопросе реализации вычислительного ядра [2] – использование микроконтроллеров или микропроцессора совместно с заказной интегральной схемой.

Создание вычислительного ядра встроенной системы представляет собой сложную, слабо формализованную на сегодняшний день задачу. Усилия специалистов направлены на создание методик системного проектирования таких ядер [3]. Разработка подобных методик требует накопления опыта использования новой программируемой элементной базы. В статье приводится анализ результатов экспериментального проекта приборного контроллера. Проект выполнялся в рамках второго подхода к проектированию вычислительного ядра.

Организация встроенной системы на базе заказных схем

Рассмотрим процесс проектирования встроенных систем на базе заказных интегральных схем ASIC (Application Specific Integrated Circuits) совместно с микропроцессором. Вычислительное ядро системы представляет собой микропроцессор и специализированный периферийный расширитель на базе ASIC (см. рис. 1). Микропроцессор предназначен для реализации основного системного алгоритма управления, а периферийный расширитель занимается интерфейсным согласованием неоднородных компонент системы и предоставляет высокоуровневые функции обмена для микропроцессора.

Для реализации ASIC удобно использовать ПЛИС с архитектурой FPGA (Field Programmable Gate Array), которые с развитием проекта могут заменяться на более дешевые заказные структуры [4].

Для проектирования заказных интегральных схем сегодня существует довольно много инструментальных средств, например фирм Cadence и Mentor Graphics, которые

позволяют описывать сложные проекты с использованием поведенческих конструкций, сообразно программным алгоритмам. Средства автоматизированного синтеза и моделирования позволяют относительно просто получить работающую структуру заказного устройства в рамках заданного элементного базиса ПЛИС или подготовить технологические материалы для изготовления заказного кристалла.

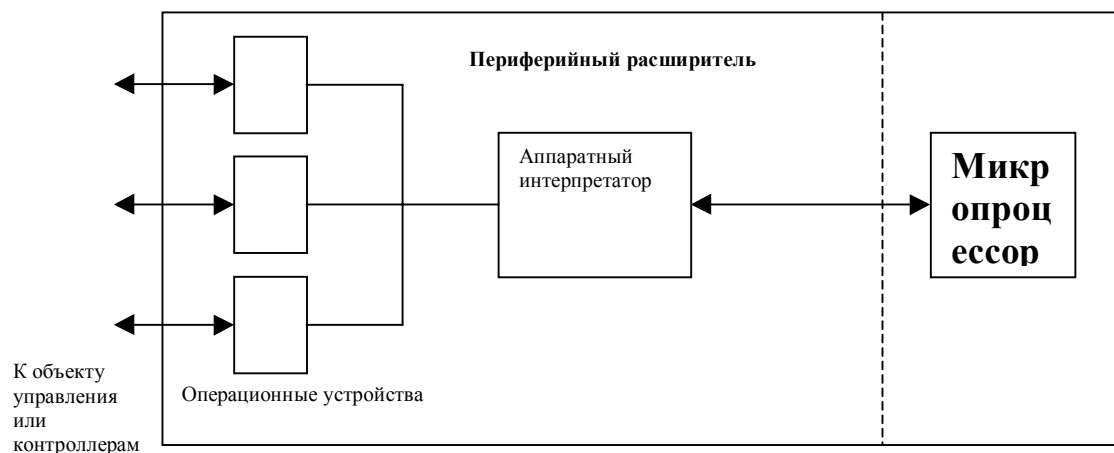


Рис. 1. Архитектура встроенной системы на базе микропроцессора с периферийным расширителем.

Рассмотрим архитектурные аспекты специализированного периферийного расширителя на базе заказной схемы, который является неотъемлемой частью вычислителя на базе микропроцессора:

- интерфейс периферийного расширителя с микропроцессором;
- структура операционных устройств (ОУ) периферийного расширителя;
- модули управления ОУ и организация информационных потоков целевого алгоритма.

Интерфейс связи микропроцессора с периферийным расширителем должен быть организован таким образом, чтобы он был "прозрачен" для выполняемого алгоритма. Это обеспечивает эффективное программное взаимодействие с периферией за счет снижения затрат на реализацию сложных режимов доступа. Требования к интерфейсу следующие:

- параллельный интерфейс (системная шина микропроцессора);
- доступ к периферийному расширителю сообразно доступу к памяти с произвольным доступом;
- развитые (гибкие) механизмы синхронизации работы интерфейса (минимизация взаимодействий в режиме запрос/ответ по системной шине для обмена данными, гибкая система прерываний, наличие механизмов прямого доступа (DMA) и другие).

ОУ представляют собой проблемно-ориентированные модули, которые взаимодействуют с объектом управления непосредственно или через программируемые интегральные контроллеры. В зависимости от размерности задачи управления число таких модулей в составе специализированного вычислителя может быть довольно большим. Они работают параллельно друг относительно друга и, как правило, занимаются низкоуровневым интерфейсным взаимодействием (синхронное / асинхронное параллельно-последовательное преобразование, фильтрация, шифрация/дешифрация сигналов или данных и другие). Перечислим основные свойства описанных ОУ:

- относительно простая структурная организация (реализация в виде автомата на "жесткой" логике);

- низкий уровень программируемости;
- единообразный интерфейс;
- функционирование в режиме запрос/ответ.

Для управления ОУ в составе периферийного расширителя должны присутствовать программируемые управляющие модули. Количество таких устройств невелико (один или два). Главной функцией этих модулей является обеспечение работоспособности ОУ – подготовка и/или предварительная обработка данных объекта и их буферизация. Основные параметры таких модулей:

- довольно емкая архитектурная организация на базе автоматов с "программируемой" (хранимой в памяти) логикой;
- система команд фиксированная, отсутствие сложных режимов адресации, одинаковое время выполнения команд, отсутствие сложных команд, требующих реализации сложных функциональных блоков (умножение, деление и другие);
- программируемость на уровне алгоритма (возможность перепрограммирования микропрограммы в работающей системе);
- согласованность интерфейса с ОУ.

В качестве управляющего модуля был использован проблемно-ориентированный аппаратный интерпретатор (АИ) [5,6], архитектура которого ориентирована на работу с ОУ, удовлетворяющими требованиям скорости обработки, объему адресуемых данных и имеющими однородный интерфейс. Можно гибко варьировать режим работы ОУ, изменяя микропрограммы АИ микропроцессором.

Для синтеза микропрограмм АИ должны быть разработаны специализированные инструментальные средства, куда входят:

- языки описания управляющих алгоритмов;
- оптимизирующие компиляторы и ассемблеры;
- средства отладки и анализа микрокода.

Интерфейс АИ с ОУ менее развит, чем интерфейс периферийного расширителя с микропроцессором. Основной характеристикой этого интерфейса является жесткая модель синхронизации, которая определяет некоторые структурные особенности АИ.

Реализация приборного контроллера

На базе описанной архитектуры была разработана система, состоящая из приборного контроллера и персонального компьютера: периферийный расширитель был реализован в ПЛИС, функции микропроцессора выполнял персональный компьютер. В качестве заказной структуры была использована ПЛИС фирмы Altera семейства ACEX EP1K100. Периферийный расширитель был размещен в приборном контроллере, подсоединяемом к компьютеру по интерфейсу USB. Для связи периферийного расширителя и микропроцессора использовался коммуникационный процессор на базе распространенной архитектуры x86 от фирмы AMD am186CC, работающий на тактовой частоте 40МГц. Структурная схема разработанной системы показана на рис. 2.

АИ имеет следующие архитектурные особенности:

- гарвардская архитектура;
- обработка 16-ти разрядных данных с использованием аккумулятора;
- имеется 4-х битный регистр флагов и группа команд условного перехода;
- все команды имеют одинаковую длину и унифицированный формат.

Работа с ОУ идет в режиме опроса, так как отсутствует система прерываний. АИ поддерживает не более 256 ОУ. Кроме этого, у него есть 256 ячеек под данные микропрограммы (локальные и временные переменные).

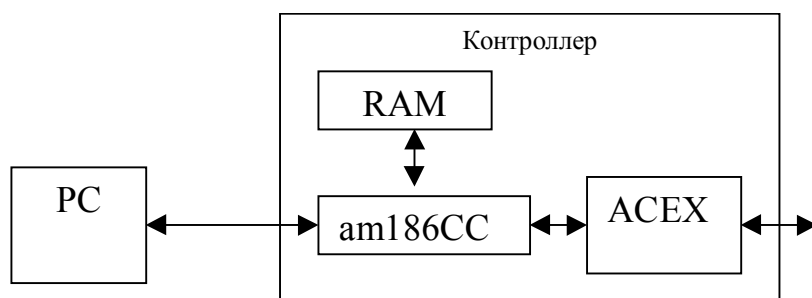


Рис. 2. Структура системы с приборным контроллером

Коммуникационный микропроцессор является главным звеном в рамках приборного контроллера. Он выполняет следующие функции:

- обмен данными с компьютером в соответствии с его запросами;
- обмен данными с АИ;
- управление работой АИ;
- слежение за ходом работы АИ.

В разработанной системе компьютер является главным узлом. Он инициирует обмен данными, запуск микропрограммы на исполнение, а также может ее завершить. Все свои функции компьютер выполняет через коммуникационный микроконтроллер:

- управление периферийным расширителем;
- обмен данными с периферийным расширителем;
- анализ данных, полученных от периферийного расширителя.

Интерфейс между периферийным расширителем и компьютером можно разделить на две части: интерфейс обмена данными и интерфейс управления, который включает как управление АИ и коммуникационным микропроцессором, так и получение и анализ результатов их работы. Каждый из участников интерфейса работает асинхронно.

В системе есть три потока данных: два входных (PC → АИ) и один выходной (АИ → PC). Понятно, что скорость работы АИ выше скорости интерфейса, но из-за того, что микропрограмма производит обмен данными не постоянно, то можно избежать ситуаций ожидания данных, буферизируя их. Для каждого потока реализовано три буфера: буфер в АИ, буфер в статической памяти микропроцессора и буфер в памяти компьютера. Направление потоков и связь буферов показана на рис. 3.

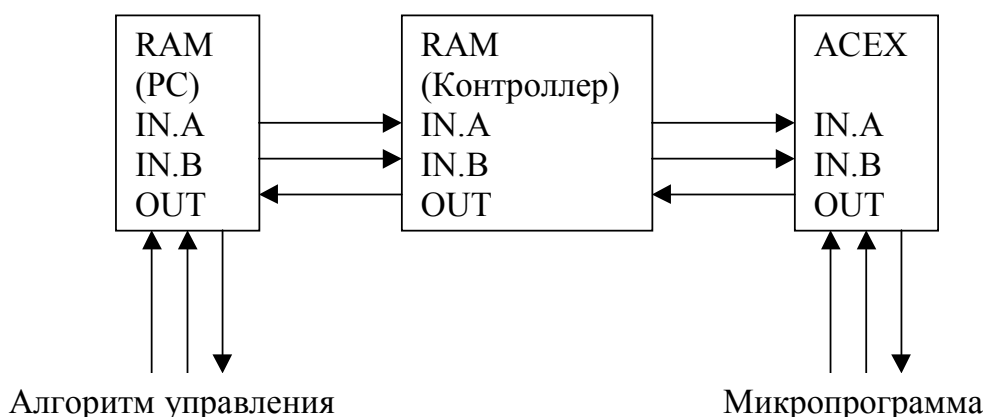


Рис. 3. Буферы и потоки данных

Буферы, расположенные в ACEX и организованы по принципу FIFO. У каждого из буферов есть контроль количества данных (count) и граница (threshold), при пересечении которой исполнение микропрограммы приостанавливается.

Микропроцессор содержит свои буферы в статической памяти контроллера. Он, когда не занят обменом данными с компьютером, производит обмен между своими

буферами и буферами АИ. Кроме этого, получая сигналы о приостановке работы АИ, он возобновляет его работу, добавляя данные из своих входных буферов или читая данные в свой выходной буфер. Разумеется, все это происходит в случае наличия данных во входных буферах и свободного места в выходных буферах. Буферы микропроцессора организованы по принципу FIFO, но программным образом.

Последние буферы расположены в памяти компьютера и содержат непосредственные данные пользователя.

Среди всех описанных буферов самые маленькими являются буферы ACEX (512 байт на каждый буфер), а самыми большими – буферы, содержащие данные пользователя (размер почти неограничен). В то же время размер буферов в микропроцессоре и компьютере можно менять, а в ACEX он фиксированный. В оптимальном варианте нужно, с одной стороны, чтобы буферы не были слишком большими, а с другой стороны – чтобы АИ работал непрерывно, т.е. в буферах постоянно был необходимый объем данных (свободного места).

В рамках проекта был разработан набор инструментальных средств, включающий:

- компилятор языка высокого уровня (типа C). Он создает бинарные образы для их исполнения АИ, оптимизирует их по скорости исполнения, по объему кода и данных;
- компилятор языка Ассемблер. Он позволяет воспользоваться всеми возможностями, предоставляемыми АИ. С помощью него можно создавать файлы для симуляции выполнения микропрограммы в среде MAX+PLUS II;
- система удаленных вызовов функций, выполняющихся микроконтроллером. На их основе построен API обмена данными между ним и компьютером;
- начальный загрузчик, позволяющий программировать ПЛИС и настраивать контроллер.

Заключение

Разработка полнофункционального варианта системы велась с использованием языка Verilog HDL и средств автоматического синтеза/размещения от фирмы Altera. Ресурсы периферийного расширителя занимают 80% кристалла ПЛИС. Применение АИ не было первой реализацией поставленной задачи. Сначала был предложен и реализован программный интерпретатор. По возможностям он был функциональнее АИ: он содержал 17 инструкций (АИ поддерживает 7 инструкций). Он поддерживал операции умножения, деления, которые были исключены из языка для АИ. Недостатками программного интерпретатора были объем кода и скорость его работы. Исполнение одной инструкции составляло несколько миллисекунд, в то время как у АИ – 100 нс.

Литература

1. S. Prakash and A. Parker. SOS: synthesis of application-specific heterogeneous multiprocessor systems // Journal of Parallel and Distributed Computing. 1992.V..16. PP.338 – 351.
2. Hardware-Software Codesign // IEEE Design & Test of Computers, January – March 2000. PP. 92 – 99.
3. W. H. Wolf. Hardware-Software Co-Design of Embedded Systems // Proceedings of the IEEE. V. 82. № 7. PP. 967 – 989. July 1992.
4. D. Wingard and A. Kurosawa. Integration Architecture for System-on-a-Chip Design // Proc. of the 1998 Custom Integrated Circuit Conference. PP. 85 – 88. May 1998.
5. R. Gonzalez and Xtensa, A Configurable and Extensible Processor // IEEE Micro. 20(2). March/April 2000.
6. J. A. Fisher. Customized instruction-sets for embedded processors // 36th DAC. PP. 253 – 257. June 1999.

АРХИТЕКТУРНЫЕ АБСТРАКЦИИ В ТЕХНОЛОГИИ СКВОЗНОГО ПРОЕКТИРОВАНИЯ ВСТРОЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

А.Е. Платунов

В работе рассматриваются вопросы высокоуровневого проектирования встроенных вычислительных систем (embedded systems). В рамках создания архитектурных моделей таких систем анализируются перспективные абстракции, как известные, так и предлагаемые автором. Формулируется задача создания "действующей" модели встроенной вычислительной системы, инвариантной к способу аппаратно-программной реализации.

Введение

Проектирование встроенных вычислительных систем (ВсС) [1] представляет собой сложную техническую задачу. Процесс проектирования направлен на выбор приемлемого (в смысле целевого критерия) решения при существовании потенциально огромного числа вариантов и ограниченном количестве пробных реализаций. На практике число анализируемых разработчиком вариантов, включая прототипы, ограничивается единицами. Это определяется сжатыми сроками и бюджетами разработок на фоне высокой сложности проектируемой системы, отсутствием эффективных технологий и инструментальных средств [2].

Общая задача в области проектирования ВсС состоит в создании сквозной технологии проектирования ВсС, основанной на формальном представлении системы от этапа исходных спецификаций до реализации.

Задача носит комплексный характер. Ее решение в значительной мере "буксует" из-за отсутствия эффективных моделей архитектурного уровня, которые адекватно отображали бы особенности ВсС [3]. Существующие модели носят последовательный характер и слабо учитывают фактор времени. Они применимы для вычислительных систем (ВС) общего назначения. Использование таких моделей при создании ВсС порождает проблемы в части реактивных свойств системы, ее функциональной надежности, возможности дальнейшей модификации.

В работе анализируются подходы к проектированию ВсС, вводится и обсуждается понятийная база технологии сквозного проектирования ВсС.

Технологии проектирования ВсС

Традиционные подходы к проектированию ВсС. Ключевыми чертами традиционного процесса проектирования микропроцессорных вычислительных систем следует считать:

- ручное разбиение системы на аппаратную и программную части (на основе опыта разработчика) на начальном шаге проектирования;
- раздельное моделирование и последовательное проектирование аппаратуры и программы;
- ручную интеграцию аппаратной и программной частей проекта;
- исправление (компенсацию) выявившихся в процессе отладки ошибок за счет изменения программы (с ухудшением характеристик системы);
- повторное выполнение цикла проектирования при невозможности компенсировать ошибки за счет программной части.

Особенность данного процесса состоит в раннем делении системы на аппаратную и программную составляющие с последующим изолированным их проектированием.

Такой подход приводит к высокой избыточности реализации, выявлению ошибок только в конце – на этапе объединения аппаратуры и программы, устранению ошибок путем практически полного повторного проектирования.

Распространенные технологии проектирования ВcC можно разделить на группы по степени влияния применяемой готовой элементной базы на архитектуру создаваемой системы. Можно говорить о следующих группах: а) проектирование в рамках высокоуровневой системной платформы (на готовой вычислительной системе); б) проектирование на основе набора вычислительных модулей (полузаказное); в) проектирование в условиях свободного выбора элементной базы (заказное).

На практике сегодняшнее массовое проектирование ВcC сводится к полузаказному (модульному) способу или к использованию готовой вычислительной системы.

Заказное проектирование ВcC применяется в случаях создания систем с "нестандартными" характеристиками и при создании образцов для дальнейшего серийного и массового производства.

Дальнейшее применение морально устаревших традиционных технологий проектирования ВcC делает разработки уже в сегодняшних условиях неконкурентоспособными.

Альтернативой являются потенциально реализуемые в современных условиях технологии сквозного параллельного проектирования. Результатом их применения должна стать экономия вычислительных ресурсов (на порядки величины) в сочетании с повышением надежности проектирования и сокращением сроков.

Технология сквозного проектирования ВcC. Центральной идеей сквозного проектирования ВcC является перенос основного объема проектирования на абстрактный уровень. Соответствующая технология может складываться из: а) иерархии архитектурных моделей ВC, инвариантных к аппаратно-программной реализации; б) методики проектирования, направленной на перенос основного объема проектирования на уровень абстрактных вычислительных механизмов; в) инструментальной инфраструктуры, обеспечивающей корректное и эффективное преобразование формальных описаний различного уровня и назначения.

В основе методики сквозного проектирования ВcC [1] лежат следующие положения:

- унификация представления аппаратной и программной частей проекта;
- унификация процесса проектирования аппаратной и программной частей проекта;
- использование сквозного формализованного описания проекта;
- оценка качества проекта на уровнях его абстрактного представления;
- выполнение основного объема отладки проекта на этапах абстрактного представления;
- дополнение архитектурной модели рядом нетрадиционных для этого уровня представления параметров;
- использование расширенной трактовки вычислительной элементной базы.

Важной особенностью технологии сквозного проектирования ВcC является ее принадлежность к заказной модели проектирования в рамках предложенной выше классификации. При этом вторичность выбора элементной базы присутствует даже в тех случаях, когда элемент представляет собой высокоуровневую системную платформу и фактически целиком "закрывает" задачу.

Архитектурные абстракции сквозного проектирования ВcC

Далее мы рассмотрим ряд ключевых абстракций и некоторые важные модели, перспективные, по мнению автора, в контексте развития и конкретизации технологии сквозного проектирования ВcC. Варианты методики сквозного проектирования и инструментальной инфраструктуры оставлены за рамками данной работы.

Вычислительная платформа (ВП). Термин *платформно-ориентированное проектирование* (platform-based design) был введен совсем недавно специалистами в

контексте технологии Codesign [4, 5]. Существуют его различные трактовки. Мы этим термином будем обозначать фиксированную функционально завершенную часть архитектуры ВсС, инвариантную к способу реализации.

Специфицируя ВП на уровне технического задания (ТЗ) или используя этот объект на уровне элементной базы, разработчик может в значительной степени формализовать процесс проектирования ВсС.

ВП можно рассматривать, как: а) класс элементной базы, наряду с микропроцессорами (МП), интерфейсами, операционными системами реального времени (ОСРВ) и другими объектами; б) способ обеспечения программной и аппаратной совместимости; в) базу для повторного использования разработанных компонент; г) абстрактное ядро ряда совместимых ВсС.

Расширенный взгляд на элементную базу ВсС. Привычное понятие элементной базы ВсС в контексте сквозного проектирования нуждается в пересмотре. Повышая уровень абстракции проектирования, разработчик должен сосредоточиться на тех классах элементов, которые значимы, в первую очередь, в рамках архитектурного представления ВсС.

Необходимо изменить подходы к представлению и оценке параметров элементов. На передний план должны выноситься системные параметры, которые позволяют оценить такие стороны использования элемента, как функциональность, системная производительность, надежность, переносимость, степень стандартности. Большое значение приобретают "не технические" в обычном понимании параметры, позволяющие учесть организационные и экономические риски применения элемента, оценить необходимую для применения инфраструктуру.

В качестве новых классов элементной базы ВсС следует рассматривать алгоритмы, программные компоненты (например, драйверы устройств), операционные системы, протоколы, интерфейсы, вычислительные платформы.

Задача описания элементов для использования на высоких уровнях абстракции предполагает унификацию представления независимо от того, к какой из традиционно выделяемых групп – аппаратной или программной, они относятся. Более того, важнейшее значение приобретает категория виртуальных элементов, которые можно определить как параметризуемые объекты с заданной функциональностью и незафиксированным способом реализации.

Встроенное программное обеспечение (embedded software). Важным шагом в области проектирования ВсС явилось определение термина *встроенное программное обеспечение* [3], который фиксирует особенности этой отрасли программирования. Как отмечает один из идеологов этого направления Э.А. Ли в своей работе [6], встроенное программное обеспечение (ПО) предназначено не для трансформации данных (как ПО "обычных" ВС), но для взаимодействия ВС с реальным миром.

Рассматривая организацию встроенного ПО на уровне архитектуры, Э.А. Ли выделяет в качестве основных абстракций программирования своевременность (timeliness), параллельность, живучесть (оригинальное liveness) и неоднородность (heterogeneity).

Существует проблема "программистского перекоса" в проектировании ВсС. Взгляд на проектирование систем реального времени только в виде традиционного программирования (пускай и со всеми этапами, начиная от архитектуры программной надстройки), безусловно, вреден.

Современные аппаратные возможности позволяют реально распространить "программистские" технологии на проектирование аппаратной компоненты ВсС, что является предпосылкой для унификации процесса проектирования системы в целом.

Проектирование систем на кристалле (system on chip - SoC). Одной из составляющих технологии создания ВсС является проектирование сложных цифровых

интегральных микросхем. В этой сфере в наибольшей степени ощущается потребность в технологиях повторного использования разрабатываемого продукта.

Ведущие фирмы в области создания цифровых микросхем и схемотехнических САПР представили ряд стандартов в рамках reuse-технологий [7, 8]. Такие технологии в своей основе содержат иерархию моделей для описания функциональных элементов на последовательных стадиях создания.

Следует отметить, что в reuse-стандартах на сложные аппаратные элементы в наибольшей степени проработаны схемы специфицирования, высокоуровневые модели функционирования и интерфейсные модели. Введены понятия виртуальных компонент и вычислительных моделей, что позволяет разработчикам накапливать не реализации, а механизмы функционирования в абстрактном виде с различной степенью детализации.

Технология создания SoC тесно смыкается с технологией проектирования ВсС. Анализ reuse-технологий этой области выявляет недостаточное развитие в них средств обслуживания программной компоненты. Результатом является "ущемленное положение" программных абстракций по сравнению с аппаратными в инструментarii проектировщика. Это проявляется в виде поиска решения преимущественно в навязанной инструментом "аппаратной" области.

Программирование как универсальный инструмент проектирования ВсС. Поднять эффективность проектирования ВсС возможно за счет формализации высокоуровневых этапов проектирования и унификации вычислительных абстракций. Перспективным следует считать распространение хорошо развитых приемов традиционного программирования на весь цикл проектирования ВсС.

Работы в этом направлении в мире ведутся очень активно. Прежде всего следует отметить, что уже сегодня сколько-нибудь серьезное аппаратное проектирование выполняется языковыми средствами, однако эти средства заметно отличаются от языков высокого уровня для процессоров с последовательной интерпретацией команд. Известны, например, реализации САПР аппаратуры, в которых входное описание очень близко к языку Си. Однако стандартом де-факто является представление результата работы САПР системного уровня в виде совокупности Си и VHDL – текстов, соответственно, для описания программной и аппаратной частей проекта.

Большой интерес представляют в этом смысле синхронный язык Esterel [9] и язык многопланового системного описания SLDL [10], предоставляющие разработчику конструкции, не ориентированные на тот или иной способ реализации непосредственно.

Архитектурные агрегаты (АА). Архитектурная модель ВсС в рамках методики сквозного проектирования [1] может формироваться в двух вариантах: в терминах функциональных преобразователей и виртуальных машин (Ф-модель); в терминах архитектурных агрегатов (А-модель).

Ф-модель может быть представлена в структурном или автоматном стиле и позволяет отразить алгоритм функционирования ВсС, но без многих важных аспектов реализации.

Более полное представление о проектируемой ВсС можно получить посредством А-модели. Базис АА позволяет сгруппировать объекты различных классов по трем направлениям: непосредственно функциональные преобразователи; технические приемы (алгоритмы или механизмы); "не вычислительные" свойства.

Рассмотрим более подробно элементы, лежащие в основе Ф- и А-модели ВсС. Функциональный преобразователь можно определить как объект, который имеет информационные входы, информационные выходы, управляющие входы и характеризуется законом преобразования. Можно различать абстрактные и реальные ФП.

Абстрактный ФП может быть определен как абстрактный вычислительный механизм. Примеры: умножитель, механизм прерывания вычислительного процесса, сторожевой таймер.

Виртуальной машиной будем называть ФП, который позволяет решать логически завершенную задачу пользователя или разработчика ВС. В качестве примеров ВМ можно рассматривать Java-машину, диспетчер вычислительных процессов ОС.

Архитектурные агрегаты – одно из центральных понятий предлагаемой методики. Определенной близостью к АА обладают понятия IP – Intellectual Property modules и VC – Virtual Components [2, 7]. АА рассматривается как абстрактное представление технического решения, которое обеспечивает определенное качество или свойство ВМ. Примеры: арифметико-логический блок, память команд/данных, избыточность кодирования, моноблочная конструкция.

Абстрактные ФП представляют собой одну (центральную) из групп АА. Функциональная классификация ФП, таким образом, рассматривается как базовая классификация "вычислительных" АА. В рамках А-модели базовая классификация дополняется группами, которые позволяют непосредственно отобразить в проекте важные "не вычислительные" аспекты.

Практическое использование А-модели нуждается в формальном описании АА и методике получения метрик. Однако использование А-модели даже без должной степени формализации позволяет разработчику ВС получить интересные результаты. Далее кратко будут описаны результаты двух проектов, в которых неформальным образом использовались обсуждающиеся в данной работе абстракции и модели.

Построение "действующих" моделей ВС, инвариантных к способу реализации. Уровень представления (детализации) архитектурной модели ВС непосредственно определяет ряд важных характеристик проекта. Например, в зависимости от выбранного базиса (конкретный набор АА) будет определяться фактический объем архитектурного этапа проектирования и степень "привязки" результирующей модели к стилю реализации.

Предлагается создавать детальное представление проекта в терминах стандартных механизмов вычислительной системы (стандартные АА). Такое представление может быть получено в результате эволюции архитектурной модели ВС до определенного уровня детализации многими известными методами.

Целевой функцией такого преобразования является декомпозиция исходной спецификации на элементы базиса вычислительных механизмов с определенными свойствами, с указанием для каждого из элементов конкретного набора параметров. Элементы такого "свободного" базиса должны обладать "прозрачностью" функционирования (т.е. механизмы должны быть стандартными, предполагающими хотя бы одну известную реализацию), но не навязывать конкретный стиль (аппаратный или программный в традиционном понимании) реализации.

В качестве способа описания свободного базиса может быть принят верхний уровень одного из стандартов reuse-технологий или формальное представление АА в рамках А-модели. Формирование такого базиса является на сегодня открытой задачей.

Внедрение "действующих" А-моделей в практику проектирования ВС предполагает наряду с формальным описанием АА создание плеера (симулятора) для "прогона" свободных базисных моделей ВС, обладающего широкими возможностями в части параллелизма и временного масштабирования работы модели. Частичным прототипом такой системы можно считать продукты серии "Virtual Component Co-Design" (VCC) фирмы Cadence, однако отсутствие развернутой информации о них не позволяет делать точные оценки.

При всей масштабности задач автоматизации архитектурного проектирования ВcС автору кажется вполне реальным создание экспериментальных САПР этого уровня силами ограниченного научного коллектива.

Опыт использования архитектурных абстракций в проектировании ВcС

Положения и элементы технологии сквозного проектирования ВcС отрабатываются в рамках научно-производственной деятельности фирмы "ЛМТ" (<http://lmt.cs.ifmo.ru>) при непосредственном участии автора на протяжении ряда лет. Рассмотрим кратко два проекта, о которых дополнительную информацию можно найти на интернет-сайте фирмы и в литературе.

Коммуникационный вычислитель МЗМ с высокой безопасностью функционирования. Исходные условия проектирования данного изделия формулировались следующим образом: создать одноплатный вычислитель с безопасным процессорным ядром, широким набором двоированных последовательных интерфейсов для сопряжения с модулями верхнего и нижнего уровней иерархии, возможностью ограниченного расширения системных ресурсов, возможностью функционирования стандартной ОСРВ. При этом параметры целевой задачи (например, интенсивности информационных потоков и сложность их обработки) определялись приблизительно.

Было принято решение разработать гетерогенную многопроцессорную ВП со значительными возможностями перераспределения вычислительной нагрузки между "программными" и "аппаратными" процессорами, оснащенную базовой инструментальной инфраструктурой.

Созданная аппаратная ВП, именуемая МЗМ, содержит три процессора для встроенных применений с архитектурой X86, два массива оперативно программируемой логики фирмы Altera, выделенные CAN-контроллеры и TCP-стеки, ряд массивов оперативной и Flash-памяти, слоты интерфейсов расширения с программируемыми функциями, набор стандартных устройств поддержки реального времени, сервисный процессор питания и термоконтроля.

В основу инструментальной инфраструктуры ВП положен принцип вложенной отладки распределенных ВС, чем обеспечивается эффективный транзитный доступ ко всем программируемым элементам с инструментальной консоли (например, с ПК).

Предложенная аппаратная ВП позволила прикладным проектировщикам реализовать в ограниченные сроки ряд прототипных и целевых вариантов центрального вычислителя линейного пункта системы диспетчерской централизации железнодорожного движения [11]. Следует отметить серьезное разнообразие архитектур разработанных версий вычислителя. Например, объем исходных алгоритмов, реализуемых на аппаратных акселераторах, по отношению к общему объему алгоритмов в различных реализациях изменялся от 10 до 50%.

ВП МЗМ, дополненная ОС QNX, продолжает активно использоваться в экспериментах по ряду направлений, в частности, на ней отрабатываются адаптивные механизмы надежности и безопасности функционирования.

Многофункциональный контроллер для встроенных применений MEC5091/ADEPT. Экспериментальный проект по созданию такого контроллера начат в 2001 г. Одной из основных задач проекта является практическая проверка эффективности положений и приемов технологии сквозного проектирования ВcС.

Разработана архитектурная модель контроллера, в значительной степени представленная средствами языка UML. Выбран состав периферийных интерфейсов и варианты ресурсов вычислительного ядра. Определены параметры информационных потоков в контроллере. Разработана модель инструментальной инфраструктуры.

Обеспечить создание архитектуры контроллера, независимой в полной мере от доступной элементной базы, к сожалению, в рамках проекта не удалось. Однако перечень вопросов, решенных на этапе архитектурного проектирования, был значительно расширен.

Контроллер создан в виде набора из двух модулей формата PC104 с шифрами MEC5091 и ADEPT. В сумме они представляют собой масштабируемую в широких пределах системную ВП с набором резидентных и внешних (для работы на ПК) программных драйверов, оснащенную развитой инструментальной инфраструктурой.

В настоящее время на ВП MEC/ADEPT реализован контроллер сканирующего зондового микроскопа. Ведется разработка еще ряда целевых систем на базе ВП MEC/ADEPT.

Заключение

Растущая потребность в ВcC в мире определяет направленность усилий специалистов на создание промышленной технологии их проектирования.

На повестке дня стоят задачи формализации ранних стадий проектирования ВcC, совершенствования и развития абстракций такого проектирования.

К сожалению, активность отечественных специалистов в данном направлении, судя по публикациям, отсутствует.

Автор надеется, что высказанные в работе мысли будут способствовать как решению конкретных технических проблем, так и интеграции российских специалистов в процесс создания передовых технологий не только в области офисного программного обеспечения.

Литература

1. Архитектурная модель цифровых вычислительных систем для встроенных применений. / Платунов А.Е. // Изв. вузов. Приборостроение. 2001. Т.44. №3. С. 8–15.
2. Hardware-Software Codesign. // IEEE Design & Test of Computers, January – March 2000, pp. 92 – 99.
3. Embedded Software – An Agenda for Research. / E.A. Lee // Technical Memorandum UCB/ERL M99/63, University of California, Berkeley, CA 94720, December 15, 1999.
4. System Design: Traditional Concepts and New Paradigms. / A. Ferrari, A. Sangiovanni-Vincentelli // Proceedings of the 1999 Int. Conf. On Comp. Des., Austin, Oct. 1999.
5. System Level Design: Orthogonalization of Concerns and Platform-Based Design. / K. Keutzer, S. Malik, R. Newton, J. Rabaey, A. Sangiovanni-Vincentelli // IEEE Transactions on Computer-Aided Design of Circuits and Systems, Vol. 19, No. 12, December, 2000.
6. Embedded Software. / E.A. Lee // Technical Memorandum UCB/ERL M01/26, University of California, Berkeley, CA 94720, November 1, 2001.
7. VSI Alliance. Model Taxonomy. Version 2.1 / URL: <http://www.vsi.org>.
8. URL: <http://www.gigascale.org>.
9. The Esterel synchronous programming language: Design, semantics, implementation. / G. Berry, G. Gonthier // Science of Programming, 1992, vol. 19, № 2, pp. 87 – 152.
10. Get a handle on design languages. / G. Moretti // URL: <http://www.ednmag.com/ednmag/reg/2000/06052000/12cs.htm>
11. Комплекс технических средств для систем железнодорожной автоматики / Гавриков В.О., Платунов А.Е., Никифоров Н.Л. // Автоматика, телемеханика и связь. 1998. №11. С. 5–10.

ДИНАМИЧЕСКИЕ ИНСТРУМЕНТАЛЬНЫЕ КОМПОНЕНТЫ

Н.П. Постников

В статье предлагается архитектура инструментальных компонент встроенных систем, обеспечивающая возможность динамического распределения функций инструментального комплекса между целевой встроенной системой и инструментальной машиной. Специфика представленного подхода к организации инструментальных компонент заключается в том, что они рассматриваются как независимые множества данных и методов их обработки. В статье очерчивается класс встроенных систем, для которых возможно применение динамических инструментальных компонент. Рассматриваемая реализация инструментария была апробирована при разработке приборного контроллера на базе коммуникационного микропроцессора фирмы AMD Am186CC [1].

Введение

Повышение сложности проектов, использование новых технологий, большие объемы памяти, высокопроизводительные интерфейсы, распределенная структура, использование сложных программных компонент усиливают требования к инструментарию встроенных систем [2]. Сложившаяся в настоящее время ситуация требует учета инструментальных аспектов на этапе архитектурного проектирования встроенной системы, а собственно разработка инструментальных компонент может оказаться нетривиальной задачей. Разнообразие инструментальных требований и их специфика ставят перед разработчиком, на первый взгляд, взаимоисключающие задачи, например, обеспечение развитой функциональности и минимизация используемых ресурсов, полный доступ к ресурсам системы и обеспечение безопасности функционирования системы.

Ниже предлагается способ организации инструментальных компонент и взаимодействия с инструментальной машиной, позволяющий создавать масштабируемые инструментальные комплексы поддержки отладки современных встроенных систем.

Описание проблемы

Задачи инструментального комплекса. Основной задачей инструментального комплекса является предоставление разработчику системы (на этапе проектирования) или обслуживающему персоналу (на этапе эксплуатации) возможности решения инструментальных задач, среди которых:

- отладка системного и прикладного программного обеспечения целевой встроенной системы;
- обновление программной компоненты встроенной системы;
- верификация и тестирование блоков и узлов;
- пассивный или активный мониторинг состояния системы;
- сбор и обработка отладочной информации с последующим ее анализом.

Сам инструментальный комплекс следует рассматривать как комплекс программно-аппаратных средств, предназначенный для решения поставленных выше задач. В настоящее время все большую популярность завоевывают инструментальные средства на базе механизмов BDM и JTAG [3, 4]. Такие средства обеспечивают доступ в целевую систему на уровне регистров / инструкций, что может быть использовано для построения мощных интерактивных отладчиков (в традиционном понимании), но неприменимо для обработки значительных объемов данных или проведения автономного мониторинга целевой системы.

Наиболее распространенной архитектурой инструментального комплекса на данном этапе развития встроенных систем является распределенная архитектура с выделенной инструментальной машиной / консолью и специализированным каналом связи. Варианты реализации в этом случае отличаются только распределением

инструментальных функций между целевой встроенной системой (резидентные инструментальные компоненты) и инструментальной машиной (внешние инструментальные компоненты). Несомненно, описанная реализация имеет свои преимущества – унификация инструментальной машины, потенциально малая ресурсоемкость резидентных инструментальных компонент и др. Однако эта реализация имеет значительный недостаток – наличие выделенного (физического или логического) независимого инструментального канала передачи данных между целевой системой и инструментальной машиной, связывающего резидентные и внешние инструментальные компоненты. Большое количество функций и различный характер передаваемых данных делают задачу выбора инструментального протокола непростой. Кроме того, перенос инструментального канала на новый физический интерфейс может привести к полному реинжинирингу существующей инструментальной машины и протокола обмена.

Ядро инструментального комплекса. В рамках выбранной парадигмы инструментального комплекса (резидентные и внешние компоненты, связанные выделенным интерфейсом) предлагается способ организации инструментальных компонент для определенного класса встроенных систем, позволяющий упростить разработку резидентного инструментария и повысить эффективность повторного использования инструментального комплекса. Как было сказано выше, инструментальный комплекс (ИК) представляет собой множество инструментальных задач, выполняющих обработку данных определенного типа. При этом очевидно, что в конкретной целевой системе в конкретный момент времени активированы далеко не все заложенные в нее инструментальные возможности.

Каждая из инструментальных задач является достаточно крупной, и ее можно представить как набор более мелких инструментальных функций. Одни из них могут быть выполнены только на борту целевой системы, другие – только на инструментальной машине. Часть же инструментальных функций жестко не привязана ни к целевой, ни к инструментальной системе. Жесткая “привязанность” той или иной функции определяется в основном природой обрабатываемых ею данных, ресурсоемкостью производимых при этом вычислений и требованиями к ресурсам памяти. Исходя из сказанного выше, в качестве важной составляющей ИК можно выделить собственно обрабатываемые данные. Одни данные должны располагаться в целевой системе, другие на инструментальной машине. ИК в конечном итоге решает определенную инструментальную задачу, функционально распределенную между целевой системой и инструментальной машиной. При этом состав и распределение элементов множеств между инструментальной машиной и целевой системой являются непостоянными и зависят от таких факторов, как, например, доступные ресурсы целевой и инструментальной машин, решаемая задача, производительность инструментального канала связи. Критерием оценки качества (оптимальности) конкретного разделения функций между внешними и резидентными компонентами могут служить скорость работы, загрузка инструментального канала, минимизация использованных ресурсов целевой системы и т.д. Конкретные способы расчета оптимальных распределений выходят за рамки данной статьи.

Для создания ИК предлагаемой архитектуры необходимо решить следующие задачи:

- удобное и гибкое распределение инструментальных функций между инструментальной и целевой машинами;
- манипулирование отладочными данными инструментальной и целевой машин;
- активизация инструментальных функций;
- обмен “инструментальными” данными между целевой и инструментальной машинами.

На рис. 1 представлена обобщенная структура ИК. В минимальной конфигурации ИК состоит только из распределенного ядра, скрывающего в себе конкретный инструментальный интерфейс. Соккрытие реализации инструментального канала в ядре ИК является важным моментом, т.к. остальные (собственно инструментальные) функции комплекса становятся независимыми от канала связи. Наращивание функциональности ИК происходит благодаря возможности ядра создавать резидентные компоненты и обрабатываемые ими данные. Динамическое создание резидентных инструментальных функций (в узком смысле – инструментальных агентов, target agents [5]) и данных позволяет значительно сэкономить ресурсы целевой системы и повысить безопасность ее функционирования.

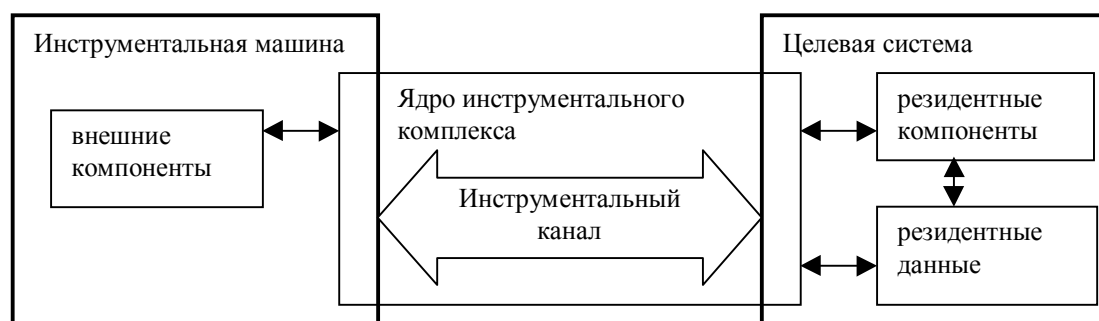


Рис 1. Структура инструментального комплекса

Видно, что часть ядра ИК функционирует в целевой системе, и это предъявляет определенные требования к целевой системе для успешной реализации ядра ИК. На этапе архитектурного проектирования встроенной системы достаточно обеспечить реализуемость функций ядра ИК, чтобы впоследствии решать различные инструментальные задачи.

Перечень функций ядра ИК

Регистрация резидентных инструментальных компонент. Функция предназначена для динамического изменения состава резидентных компонент ИК. Резидентная функция может быть зарегистрирована как со стороны инструментальной машины, так и со стороны целевой системы. При регистрации функции необходимо указать список ее формальных параметров.

Регистрация резидентных данных. Инструментальная машина, регистрируя данные, указывает их тип. В дальнейшем инструментальная машина имеет возможность обрабатывать данные с помощью резидентных функций. При необходимости инструментальная машина может удалить зарегистрированные данные, тем самым освободить ресурсы памяти целевой системы.

Определение фактических параметров резидентных функций. В общем случае для решения инструментальной задачи необходимо обеспечить сложное взаимодействие между инструментальной машиной, резидентными функциями и резидентными данными (см. рис. 2). Ядро ИК предоставляет возможности динамического изменения фактических параметров резидентных функций, назначения нескольким функциям одних и тех же данных, создания нескольких экземпляров одной функции с разными наборами фактических параметров (полезно при одновременном решении нескольких инструментальных задач). Ядро ИК осуществляет контроль типов параметров резидентных функций.

Обмен резидентными данными. Инструментальная машина в любой момент имеет полный доступ к зарегистрированным резидентным данным.

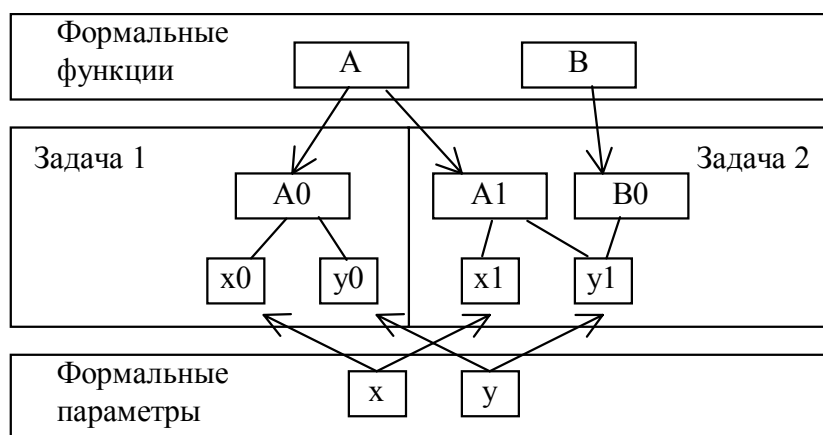


Рис. 2. Взаимодействие резидентных данных и функций

Для создания работоспособного ядра ИК к целевой системе предъявляются следующие требования.

1. Обеспечение выделенного (физического или логического) инструментального канала.
2. Предоставление возможностей управления памятью данных. Для обеспечения гибкой функциональности ИК необходимо существование механизма управления ресурсами памяти целевой системы. В случае отсутствия такого механизма в целевой системе его можно реализовать внутри ядра ИК, но при этом ресурсы целевой системы будут использоваться неэффективно.
3. Предоставление возможностей управления памятью кода. При отсутствии таких средств становится невозможно обеспечить динамику резидентной компоненты (в том смысле, что инструментальный код всегда будет присутствовать в целевой системе), но функциональность ИК по-прежнему может быть обеспечена, за счет статических резидентных компонент.
4. Механизм запуска инструментальных функций. По требованию ядра ИК целевая система должна обеспечить выполнение указанной инструментальной функции с тем или иным приоритетом относительно целевой задачи.

Значительным преимуществом предлагаемой архитектуры ИК является обеспечение доступа к различным структурам данных целевой системы (структуры RTOS, данные системного менеджера памяти и т.д.). Представленная технология позволяет реализовать механизм “инструментальных мостов”. Суть механизма заключается в том, что для интересующих сугубо целевых функций создаются простейшие функции-wrapper’ы, которые “импортируют” системную функцию в ИК, тем самым превращая ее в резидентную инструментальную. Такие инструментальные мосты могут быть с легкостью динамически “наведены” практически в любую точку целевой системы и уничтожены после выполнения инструментальной задачи.

Реализация

Представленная архитектура ядра ИК была реализована при разработке приборного контроллера на базе микропроцессора Am186CC (коммуникационный процессор фирмы AMD на базе x86 ядра) [6].

В качестве инструментального канала использовался интерфейс USB [7], который обеспечил высокую скорость передачи данных (до 600кБайт/с) и простоту организации выделенного логического канала. Операционная среда целевой системы поддерживала управление памятью, поэтому распределение ресурсов целевой системы удалось обеспечить оптимальным образом. Имелась возможность одновременного асинхронного запуска нескольких инструментальных функций с приоритетом фоновой

задачи целевой системы. Т.о. целевая система удовлетворяла всем требованиям для реализации ядра ИК в полном объеме.

Заключение

Представленная в статье архитектура инструментальных средств позволяет создавать мощные инструментальные комплексы. Простота использования, гибкость и относительная нетребовательность к ресурсам целевой системы позволяют применять этот подход для обеспечения инструментальной поддержки широкого класса систем. Предлагаемая архитектура позволяет учесть большинство инструментальных требований на этапе архитектурного проектирования и формально оценить оптимальность распределения инструментальных функций между инструментальной машиной и целевой системой.

Апробация описанной архитектуры ядра ИК показала его высокую универсальность и эффективность при разработке достаточно сложных встроенных систем.

Литература

1. MEC5091 – многофункциональный контроллер для встроенных применений. <http://lmt.cs.ifmo.ru/mec5091.html>.
2. Постников Н.П. Вложенная отладка распределенных систем. <http://embedded.ifmo.ru/up/uprs.htm>.
3. The Nexus 5001 Forum™ Standard for a Global Embedded Processor Debug Interface. <http://www.ieee-isto.org/Nexus5001/standard2.html>
4. Craig A. Haller, Macraigor System Inc. The ZEN of BDM. <http://www.macraigor.com/zenofbdm.pdf>
5. Glossary of Terms for Embedded Technology and Related Issues. <http://www.windriver.com/press/html/glossary.html>
6. AMD. Am186CC/CH/CU User's Manual. <http://www.amd.com/epd/processors/2.16bitcont/3.am186cxf/a21914/21914.pdf>
7. USB 2.0 Specification. http://www.usb.org/developers/data/usb_20.zip

ВЫБОР ОПТИМАЛЬНОГО АЛГОРИТМА ПЛАНИРОВАНИЯ ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

О.Г. Самохвалова

Настоящая работа посвящена вопросам из области планирования процессов в системах реального времени. Задачей исследований являлось: создание методики выбора приемлемого алгоритма планирования на основе анализа сведений об алгоритмах, предполагаемой модели задач и структурных характеристиках будущей системы, формирование теоретической базы для дальнейшего создания программных инструментов проектирования планировщиков систем реального времени. Статья кратко излагает результаты исследований и разработок автора по данному направлению. Предлагаемая в статье методика адресована разработчикам программного обеспечения управляющих систем.

Управление процессом предоставления ресурсов системы задачам, нитям, процедурам обработки прерываний и т.д. является одной из основных функций любой операционной системы и осуществляется при помощи такого механизма, как планирование. Данный механизм обеспечивает системе возможность параллельного выполнения нескольких задач. В системах реального времени планирование должно также гарантировать предсказуемое поведение, безопасность, возможность длительной, безотказной работы, выполнение задач к поставленному сроку. От метода планирования во многом зависит успешная работа системы в целом [1, 2].

Создание программного обеспечения управляющих систем может производиться различными способами, в том числе и с использованием "готовых" операционных систем. Однако, как правило, универсальные ОС РВ поддерживают лишь ограниченное число алгоритмов планирования (2-3), которые являются очень простыми и имеют ограниченные возможности. При построении заказных ИУС, специализированных ОС и т.п. коллектив разработчиков часто вынужден создавать свои планировщики [3–5].

При создании планировщика основной проблемой, которую необходимо решить разработчикам, является нахождение оптимального алгоритма планирования для данной системы, в соответствии с которым вычислительные ресурсы предоставляются задачам. (С точки зрения автора, алгоритм считается оптимальным, если его параметры соответствуют требованиям разрабатываемой системы, а характеристики, связанные с его практической реализацией и эффективностью функционирования, являются наилучшими для данного случая.)

Часто решение этого вопроса сопряжено с различными трудностями. Это ограниченная материальная и информационная база, сжатые сроки разработки, невозможность проведения предварительной НИР. Наличие методик, на базе которых возможен указанный выбор, существенно облегчило бы труд разработчиков систем реального времени.

Задачами проведенных исследований являлось:

- создание методики выбора приемлемого алгоритма планирования на основе анализа сведений об а) алгоритмах; б) предполагаемой модели задач; в) структурных характеристиках будущей системы;
- формирование теоретической базы для создания программных инструментов проектирования планировщиков систем реального времени.

Для решения поставленных задач было проведено изучение различных вопросов в области планирования в системах реального времени – политик планирования, их модификации, влияния аппаратных средств на механизмы планирования и т.д. Сформирован ряд критериев для сравнительного анализа стратегий планирования. Некоторые из них, например, коэффициент эффективности планировщика, предложены для оценки алгоритмов впервые. Также была разработана, приведена к

полуформальному виду и оформлена в виде методики, кратко рассмотренной ниже, последовательность действий выбора оптимального алгоритма.

Согласно Дж. Фоксу [6], процесс разработки программного обеспечения систем, в том числе и планировщиков, может быть разбит на шесть этапов: определение требований, проектирование, написание команд, компоновка, тестирование и документирование.

Разработанная методика охватывает первые два этапа, а дальнейшее ее усовершенствование позволит включить в этот список и стадию написания команд. Рис. 1 и 2 показывают, как изменяется процесс разработки ПО (планировщиков) с использованием «автоматизированного разработчика планировщиков». Схема методики отражена на рис. 3.

Первым шагом при создании планировщика, согласно схеме, является формирование на основе технического задания или иного описания системы сведений о структурных характеристиках элементов системы, задачах и окружающей среде. Эти данные являются исходными для проводимого далее анализа. Методика подробно описывает, каким образом можно получить интересующие нас данные и привести к виду, пригодному для дальнейшего использования.

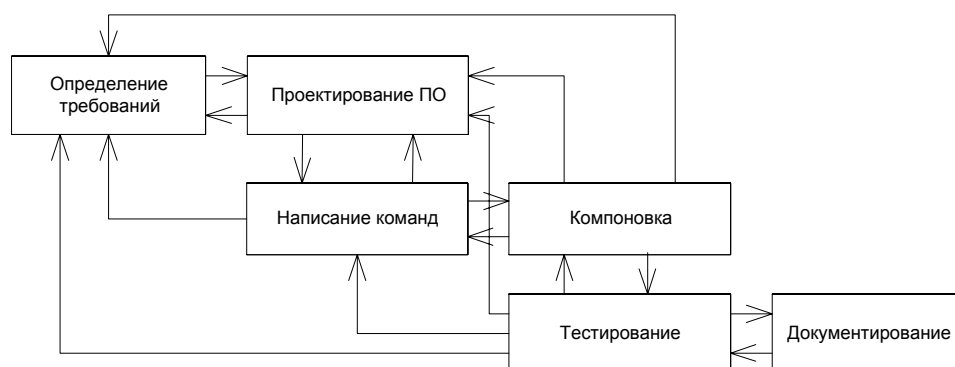


Рис. 1. Традиционный процесс разработки ПО (планировщиков)

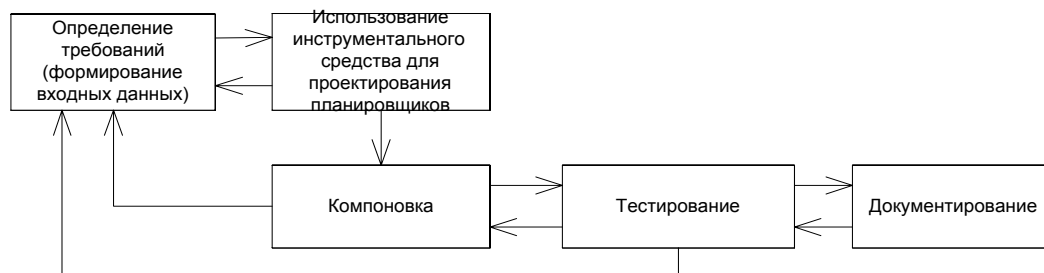


Рис. 2. Процесс разработки с использованием инструментального средства

Вторая часть методики охватывает этап проектирования, который включает в себя, как минимум, четыре стадии – проведение НИР, анализ и принятие решения, структуризацию и описание.

Использование методики позволяет опустить некоторые из названных пунктов или значительно формализовать и упростить их. Так, сведения об алгоритмах планирования в качестве входных параметров и их блок-схемы снимают необходимость в проведении предварительной исследовательской работы и описании. Сравнительные таблицы и методы выбора наилучшего решения сводят работу по

анализу, принятию решений и структуризации к набору указанных, формальных действий, снижая время на проектирование.

Часто в результате некорректных решений, возникновения тупиковых ситуаций и т.д. приходится возвращаться к более ранним стадиям разработки. Этот факт нашел свое отражение в методике в виде этапа коррекции исходных данных, где указаны пути внесения изменений во входные параметры, в том числе и без глобальных изменений в структуре и функциональности системы.

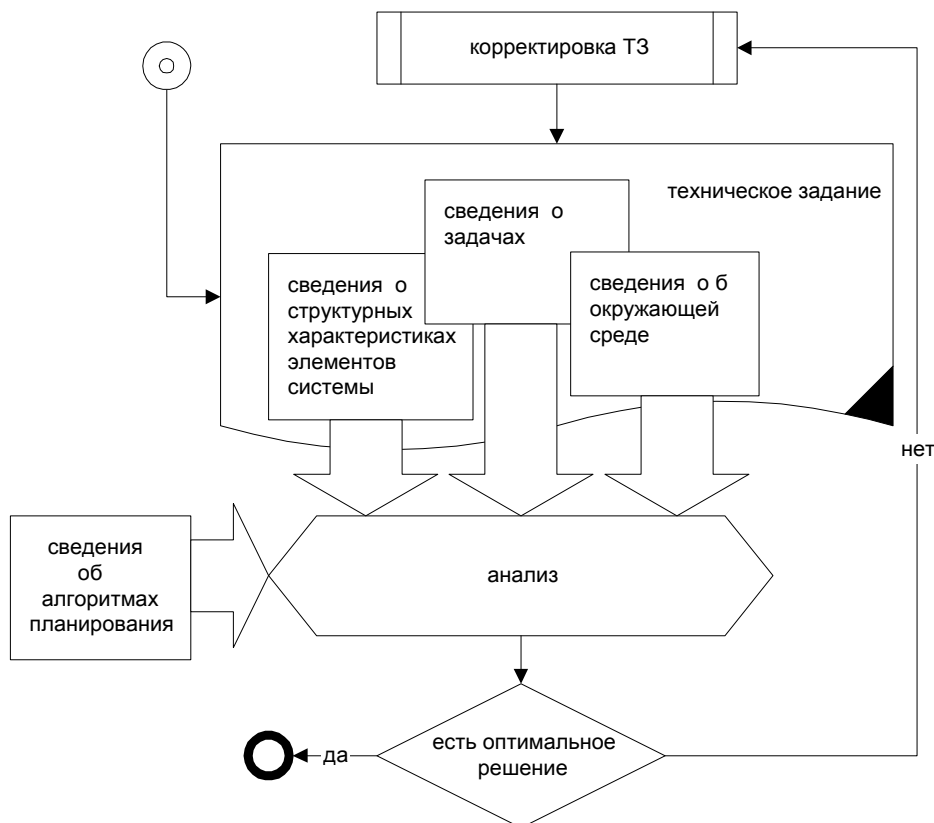


Рис. 3. Схема методики

Анализ данных и выбор алгоритма планирования разбит на 4 этапа. В первую очередь происходит проверка совместимости параметров системы. Она необходима, поскольку может быть выявлено, что при имеющихся структурных характеристиках системы невозможно поддерживать и эффективно обслуживать заданный набор задач. Для этого необходимо сравнить данные о системе с таблицей зависимостей, где указаны возможное быстродействие, прерываемость заданий и их количество в зависимости от типа процессора. После прохождения первой ступени мы можем перейти к следующей – проверке соответствия самих алгоритмов планирования входным параметрам системы. Стратегии, по каким-либо причинам не удовлетворяющие условиям разрабатываемой системы, отсеиваются на данном этапе.

После этого мы получаем некоторое множество алгоритмов, которое необходимо сравнить между собой по указанным характеристикам, что и производится на третьей стадии.

Характеристики, по которым производится сравнение, также сведены в таблицы. Анализируя их, мы отбираем алгоритм с наилучшими характеристиками по всем позициям. Если такого не существует, отсеиваем наихудшие по всем пунктам и переходим к четвертому этапу.

Здесь, применяя методы принятия оптимального решения – ранжирование критериев выбора по важности и присвоение относительных весовых коэффициентов, мы отбираем наилучший алгоритм среди дошедших до данной ступени. Сведения о предпочтительности того или иного фактора можно выявить из технического задания или иного описания системы.

В результате мы должны получить единственную стратегию планирования, которая будет максимально соответствовать заданным условиям. Поскольку блок-схемы политик планирования приведены в методике, после выбора алгоритма можно приступить непосредственно к написанию программного кода.

Разработанная методика предоставляет информацию по планированию разработчикам систем реального времени, значительно сокращает сроки на разработку планировщиков, проведение НИР, снижает стоимость системы за счет. Она также является методической базой для дальнейшего создания программных инструментов для проектирования планировщиков систем реального времени.

Методика адресована разработчикам программного обеспечения управляющих систем.

Литература

1. C.W. Mercer. An Introduction to Real –Time Operation Systems: Scheduling Theory // Technical Report, Carnegie Mellon University (USA), 1992.
2. E.A. Lee. Embedded software // To appear in Advances in Computers, vol.56, Academic Press, London, 2002.
3. S.A. Aldarmi, A. Burns. Time-Cognizant Value Function for Scheduling Real -Time System // Technical Report, University of York (England), 1998.
4. Bestavros. Scheduling. Boston University (USA), 1995.
5. M. Gonzales Harbour, L. Sha. An Application-Level Implementation of the Sporadic Server // Technical Report, Carnegie Mellon University (USA), 1991.
6. Фокс Дж. Программное обеспечение и его разработка. М.: Мир, 1985.

ИССЛЕДОВАНИЕ ОПЕРАЦИОННЫХ СРЕД ПРИКЛАДНЫХ ЗАДАЧ В ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМАХ

В.В. Окулевич

Настоящая статья посвящена вопросам создания системного программного обеспечения для информационно-управляющих систем. В статье введено определение операционной среды прикладных задач, предложен взгляд на системное программное обеспечение как на промежуточный слой между вычислительной платформой и прикладной задачей. Так же в статье изложены основные принципы разработанной автором методики проектирования операционных систем реального времени на основе архитектурных шаблонов.

В настоящее время при разработке информационно-управляющих систем (ИУС) значительной сложности используют разделение программного обеспечения на уровни различной иерархии. Как правило, в архитектуре программного обеспечения укрупненно можно выделить уровни системного и прикладного (или функционального) назначения. Далее эти уровни могут снова разделяться на новые подуровни, число которых в архитектуре определяется сложностью создаваемой системы, с одной стороны, и требованием к ясности архитектуры для разработчиков, с другой.

Для ИУС характерны высокие требования к надежности и безопасности. Им должны удовлетворять как программные, так и аппаратные компоненты системы. Влияние этих требований на системное программное обеспечение выражается, в частности, в необходимости поддержки распределенных вычислений, функционировании в реальном масштабе времени, проверке правильной работы системы, корректном восстановлении после сбоев и т.п. [1].

При создании ИУС разработчики находятся перед выбором: следует ли реализовывать системный уровень программного обеспечения самостоятельно или использовать в качестве его основы ту или иную готовую операционную систему. Во многих случаях разработчик выбирает второй путь. Действительно, использование операционных систем позволяет сократить время разработки, уменьшить объем низкоуровневого программирования, с меньшими затратами поддерживать широкий спектр устройств. С другой стороны, усложняется анализ временных характеристик системы, увеличивается избыточность конечного изделия, появляются проблемы лицензирования программного обеспечения. Разработчик вынужден искать свой путь, основываясь на компромиссе между имеющимися ресурсами (средства, время, квалификация исполнителей), с одной стороны, и ресурсами вычислительной платформы (перечень поддерживаемых устройств, аппаратные ресурсы) – с другой.

Автор предлагает при исследованиях системного уровня программного обеспечения исходить из следующих рассуждений:

С одной стороны, существует функциональность, определяемая решаемой задачей предметной области. С другой стороны, имеется аппаратная платформа, на основе которой должна быть реализована система. Таким образом, системный уровень программного обеспечения должен соединять аппаратную платформу с алгоритмами решения прикладной задачи [1].

Вне зависимости от того, используется ли готовая операционная система в разработке или нет, системный уровень программного обеспечения должен предоставить базис для эффективного решения прикладных задач системы [2].

Автор считает необходимым ввести определение операционной среды. Операционная среда – это набор структур данных и операций над ними, которые используются при решении прикладных задач. В зависимости от метода решения конкретной задачи к операционной среде предъявляются специфические требования.

Кроме того, операционная среда должна так же учитывать специфику ИУС и позволять достичь необходимых показателей надежности и безопасности. Данные

требования должны выполняться одновременно, что служит дополнительной сложностью при создании системного программного обеспечения [1, 3].

Таким образом, функция системного уровня программного обеспечения состоит в создании операционной среды для функционирования прикладных задач.

Рассмотрим подробнее случай, когда при разработке системы используется операционная система. В данном случае задача создания операционной среды состоит в адаптации (доработке) сервисов, предлагаемых операционной системой, к тем структурам и механизмам, которые могут быть эффективно использованы для решения прикладных задач. Так, например, решение системы дифференциальных уравнений, описывающих движение реальных объектов, требуется связать с асинхронно поступающими данными от датчиков положения.

Отметим, что часто даже для сравнительно небольших встроенных систем опытный разработчик создает подобие операционной системы, которая позволяет сравнительно небольшими усилиями организовывать псевдопараллельное выполнение типовых задач. В таких случаях можно говорить о том, что разработчик сам создает проблемно-ориентированную операционную среду.

В настоящее время существует достаточно большое количество модульных операционных систем. В случае их использования появляется задача изучения и конфигурирования модулей.

Поэтому во всех случаях представляется актуальной задача проектирования и разработки оптимальных операционных сред в зависимости от функций, выполняемых прикладными задачами.

Часть новых материалов, полученных автором в рамках решения этой проблемы, была представлена в магистерской диссертации "Проектирование операционных систем реального времени на основе архитектурных шаблонов". В данной работе разработаны элементы методики построения архитектур операционных систем на основе анализа классов решаемых задач, типов поддерживаемого оборудования и требований к целевой системе.

Далее кратко представлена суть новых принципов, разработанных в диссертации.

В соответствии с ними при проектировании архитектуры операционной системы реального времени (ОС РВ) определяется набор базовых представлений. Для каждого такого представления на основании анализа решаемой задачи выполняется преобразование 1-го уровня, которое состоит в определении некоторого базового набора возможных архитектур разрабатываемой ОС РВ. После этого переходят к выполнению предварительных оценок, цель которых состоит в определении соответствия полученной архитектуры требованиям решаемой задачи. Оно основано на экспертных оценках с использованием целевых функций, построенных на основании принципа ранжирования по важности или присвоения относительных весовых коэффициентов.

После получения совместных оценок для каждого из исследуемых представлений системы необходимо получить значения общей целевой функции для всего множества представлений. Для этого предложено также использовать методы ранжирования по важности или присвоения относительных весовых коэффициентов.

Цель преобразования 2-го уровня состоит в дальнейшей конкретизации и уточнении архитектуры ОС РВ. На данном этапе происходит определение особенностей реализации тех или иных механизмов и выдачи рекомендаций по устранению узких мест в зависимости от результатов оценок, полученных на 1-ом уровне. При этом шаблоны, входящие в состав представления архитектуры, изменяются в зависимости от того, какие характеристики системы нуждаются в улучшении. Принципы модификации шаблонов включают в себя возможность

реализации аппаратной поддержки, а сами шаблоны не привязаны к способу реализации.

В дальнейшем автор планирует продолжить исследования, посвященные определению оптимальных операционных сред для прикладных задач. В них будет рассматриваться не только процесс проектирования архитектуры ОС РВ, но и тесно связанные с ним системные части прикладного программного обеспечения. Даже если используемая в ИУС операционная система является покупной и только конфигурируется разработчиком, то системная часть прикладного программного обеспечения создается для конкретной решаемой задачи с учетом ее специфики.

Данная разработка, как предполагается, позволит проектировать и разрабатывать архитектуру системного программного обеспечения в зависимости от прикладной задачи и получать качественное решение для всей системы в целом.

Литература

1. Фокс Д. Программное обеспечение и его разработка /. Пер. с англ. М.: Мир, 1985. 368 с.
2. Buschmann F., Meunier R., Rohnert H., Sommerlad P., Stal M. Pattern Oriented Software Architecture - A System of Patterns. John Wiley & Sons, 1996.
3. Burns A., Wellings A.J. Designing the Hard Real-time Systems. Ada: Moving Towards 2000, 11th Ada-Europe Conference. Springer-Verlag.

ОРГАНИЗАЦИЯ ТЕСТОВОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ВСТРОЕННЫХ СИСТЕМ

А.Н. Лукичев, А.О. Ключев

Рассмотрены некоторые типовые варианты организации и даны рекомендации по созданию программного обеспечения для тестирования встроенных вычислительных систем различной сложности: от простых контроллеров с ограниченной вычислительной мощностью (тестовое меню, простейший интерпретатор) до сложных распределенных информационно-управляющих систем (вложенное тестирование).

Введение

При создании встроенных вычислительных систем возникает проблема их отладки на разных этапах разработки. Для каждого этапа существуют свои методы решения проблемы отладки, т.е. выявления, локализации и исправления ошибок. Эти методы, в общем, зависят от сложности системы, ее структурных и функциональных особенностей.

По завершении разработки и при производстве неизбежна проблема создания системы контроля качества выпускаемых изделий. Особенно остро эта проблема встает при серийном производстве. Разработка системы автоматизированного тестирования – зачастую довольно сложный и трудоемкий (а значит, и дорогостоящий) процесс.

В данной работе внимание будет уделено организации тестирования встроенных систем на этапах разработки и производства [1]. Будут рассмотрены несколько типовых вариантов организации тестового программного обеспечения и их особенности, а также проанализирована оправданность их выбора. Вопросы аппаратного обеспечения процесса тестирования, а также разработки методов тестирования конкретных механизмов данная работа затрагивать не будет.

Под системой тестирования (или тестовой системой) будем понимать программное обеспечение тестирования встроенных систем, ориентированное на проверку качества их работы, а также на обнаружение ошибок (и до некоторой степени точную их локализацию).

Обзор вариантов построения тестовых систем

Простейшим вариантом тестовой системы является набор тестовых модулей в виде отдельных независимых программ.

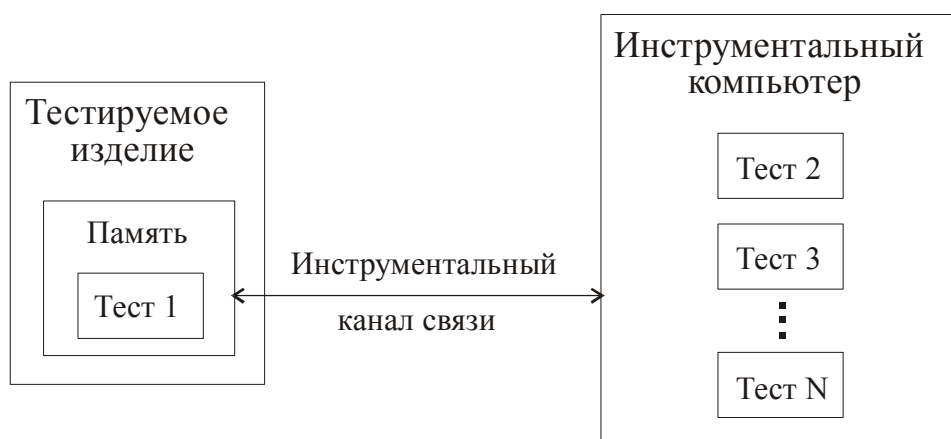


Рис. 1. Простейший вариант организации тестирования

Модули поочередно доставляются в тестируемое изделие (контроллер) с помощью средств ICP (In-Circuit Programming, программирование на борту системы): либо аппаратно предусмотренных, либо обеспечиваемых резидентными

инструментальными программами [2]. После того, как тест пройден, на его место в памяти записывается новый.

Этот вариант, очевидно, является самым дешевым и наименее трудоемким в изготовлении. Его целесообразно использовать при начальной отладке опытного образца изделия на этапе разработки. Однако очевидным его недостатком является высокая трудоемкость тестирования при большом количестве тестов (и/или тестируемых подсистем).

Выходом в данном случае является тестовая система, объединяющая в себе все разработанные тесты и доставляемая в тестируемое изделие в виде единой программы (загрузочного модуля). При запуске этой программы на исполнение в инструментальный канал связи из изделия выдается список тестов, и оператор проводит тестирование, выбирая из этого списка нужные ему. Вся информация о процессе тестирования передается в инструментальный канал связи.

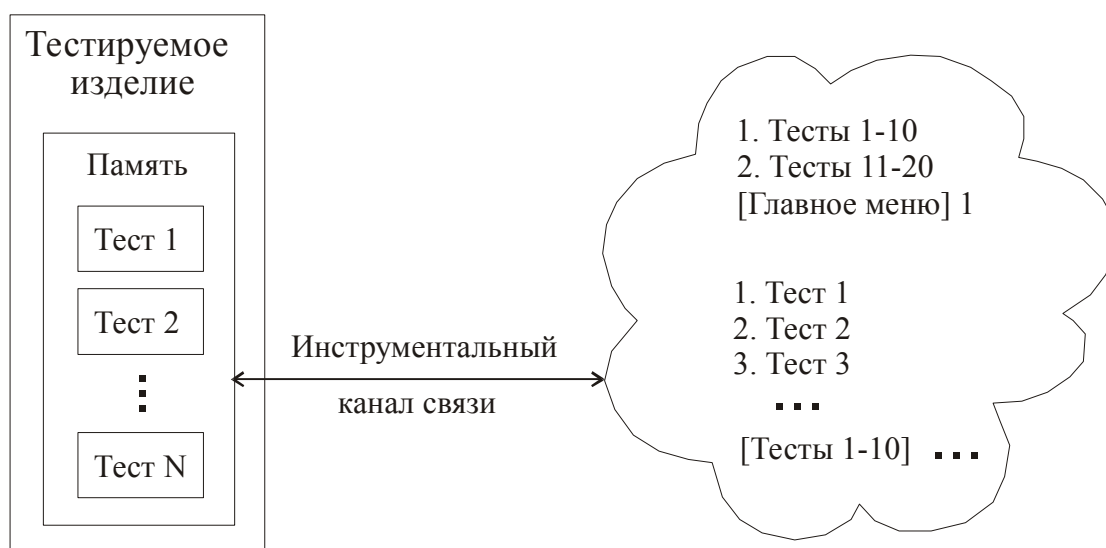


Рис. 2. Программа, объединяющая тесты в систему меню

Тесты можно сгруппировать по различным признакам в отдельные подменю (например, по тестируемым подсистемам: "Тестирование последовательных каналов", "Тесты подсистемы контроля питания", "Тесты управления дискретными выходами" и т.д.). Процесс тестирования сводится к выбору категории тестов, выполнении всех тестов из выбранной категории и переходу к следующей категории. Такой подход, по сравнению с предыдущим вариантом, немного более трудоемок в изготовлении, однако помогает систематизировать тестирование и существенно упрощает сам процесс. Область его применения также существенно ограничена, т.к. не позволяет автоматизировать процесс тестирования: для проведения тестов требуется оператор, выбирающий тесты из списка.

Однако для того, чтобы в такой системе изменить или дополнить набор тестов или изменить логику работы какого-либо из них, необходимо будет заново скомпилировать программу и получить из нее загрузочный модуль. Затем нужно будет доставить этот новый модуль в память тестируемого изделия. Этот существенный недостаток присущ и первому рассмотренному варианту построения тестовой системы. Хотелось бы иметь систему, которая позволяет изменять процесс тестирования без перекомпиляции всей программы. В этом случае можно описать тестовую систему в виде интерпретатора языка сценария экспериментов, принимающего команды с различными параметрами через инструментальный интерфейс. В качестве вызываемых интерпретатором при исполнении команды функций можно описать как специально разработанные тесты, так и отдельные функции системного API (Application Program Interface; здесь под API

понимается набор низкоуровневых сервисов, позволяющих прикладным программам использовать аппаратные возможности контроллера).

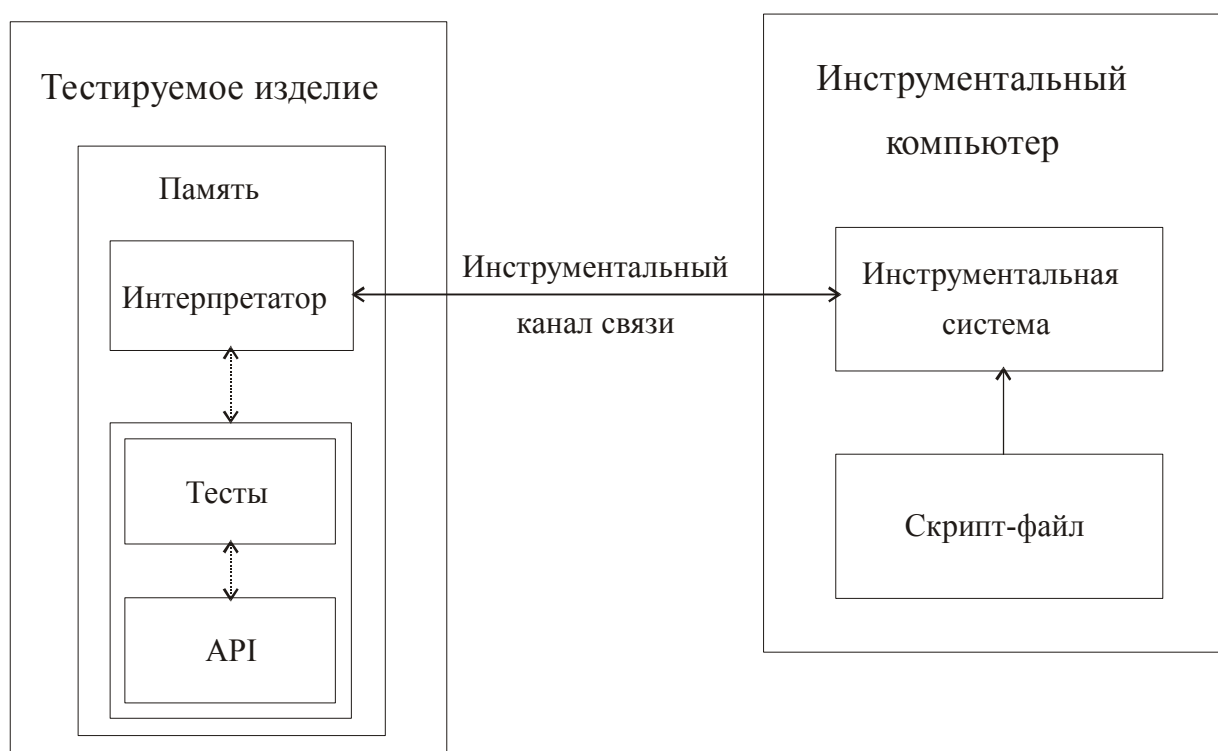


Рис. 3. Система с интерпретатором тестовых команд

На инструментальном компьютере составляется сценарий эксперимента, а затем последовательно, команда за командой, пересылается на резидентную консоль тестовой системы. Чтобы изменить или дополнить ход эксперимента, достаточно просто изменить тестовый сценарий, как правило, представляющий собой обычный текстовый файл. Такой подход можно использовать как при тестировании работоспособности подсистем изделия, так и при тестировании системного API (например, на устойчивость при различных наборах входных данных, на качество работы при повышенной нагрузке на систему). Также этот подход целесообразно применять для тестирования партии изделий, т.к. тестирование здесь в некоторой степени автоматизировано: достаточно подготовить один сценарий тестирования и запускать его, подключая изделия из партии к инструментальному компьютеру. Целесообразно при этом реализовать автоматическое составление отчетов для каждого экземпляра на инструментальном компьютере.

При таком варианте организации тестовой системы существенным недостатком является то, что сценарий тестирования обычно представляет собой линейную программу, т.е. набор команд без возможности ветвления. Что если потребуется изменить ход теста в зависимости от результата исполнения предыдущей команды? Или использовать этот результат в виде аргументов последующих команд? В этом случае необходимо усовершенствовать резидентный интерпретатор, предусмотрев возможность промежуточного сохранения данных, а также возможность условного исполнения блоков команд. Простейшим решением, на наш взгляд, является организация интерпретатора в виде стековой виртуальной машины. При исполнении какой-либо команды часть ее аргументов может помещаться в стек, результат также помещается в стек. Условные операторы извлекают значение из стека и, в зависимости от результата проведенного сравнения, исполняют или не исполняют следующий блок команд.

В случае использования пакетных интерфейсов в качестве инструментальных каналов связи с тестируемым контроллером (таких как Ethernet или CAN), целесообразно кодировать команды интерпретатора числами из фиксированного набора. Это значительно упростит процедуру их интерпретации и перенесет процесс синтаксического анализа команд сценария на инструментальный компьютер. То же самое касается и числовых аргументов команд. При этом существенно расширяется область применения такой тестовой системы, т.к. для специализированных процессоров с ограниченными ресурсами (вычислительная мощность, память) иногда довольно сложно разработать синтаксический анализатор. При тестировании таких систем целесообразно минимизировать вычислительную нагрузку на контроллер.

При пакетной передаче можно также в одном пакете пересылать целый блок команд сценария, если используемый интерфейс предусматривает достаточный для этого размер пакетов. Можно вообще передавать на борт изделия целиком весь сценарий для исполнения. Это может быть особенно важным, когда инструментальный канал связи имеет малую скорость передачи, а скорость выполнения тестового сценария критична.

Рассмотренные варианты организации тестового программного обеспечения рассчитаны на случаи, когда инструментальный интерфейс изделия является доступным для компьютера, с которого проводится тестирование. А как быть, когда внешним интерфейсом изделия является другой интерфейс, недоступный для инструментального компьютера? Или в случае многопроцессорного контроллера, когда тест требуется провести на процессоре не связанном непосредственно с внешними интерфейсами, а только через другой процессор (или их цепочку)? В этом случае необходимо организовать транзитную пересылку команд сценария через дополнительный контроллер или коммуникационный процессор (в случае многопроцессорной системы). При этом необходимо предусмотреть возможность адресации тестируемого объекта, так как промежуточный узел может быть связан еще с несколькими объектами. Архитектура такой системы, обеспечивающей так называемое вложенное тестирование, представлена на рис. 4.

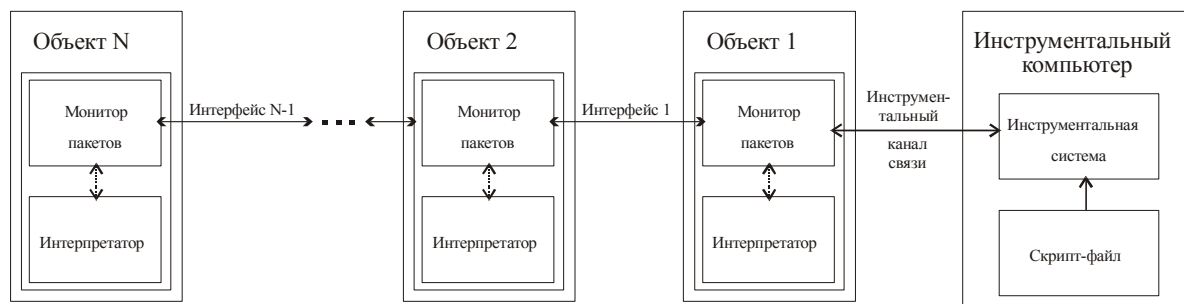


Рис. 4. Система, обеспечивающая вложенное тестирование компонентов контроллерной сети

Каждый из объектов, изображенных на рис. 4 (Объект 1, ..., Объект N), может быть доступен для тестирования с инструментального компьютера через другие. Резидентный монитор пакетов на каждом из них обеспечивает транзитную пересылку команд на другие объекты или передачу управления собственному интерпретатору [3].

Изготовление такой системы представляется более трудоемким и дорогим по сравнению с предыдущими рассмотренными вариантами, однако это оправдывается более высокой стоимостью тестируемых объектов: сложных многомодульных и/или многопроцессорных систем.

Заключение

Коллектив лаборатории микропроцессорной техники кафедры ВТ СПб ГИТМО (ТУ) занимается данной тематикой уже много лет. Все рассмотренные выше подходы к организации тестового программного обеспечения для встроенных систем опробованы на практике в рамках достаточно большого количества НИР. Информацию о некоторых из них можно получить в публикациях [4–6].

Наилучшие результаты с точки зрения качества тестирования были получены при использовании технологии вложенной отладки (рис. 4) и удаленного исполнения команд (рис. 3). Практика показывает, что применение этих двух технологий дает положительный эффект при тестировании сложных многопроцессорных распределенных встроенных систем.

Простейшие системы тестирования (система меню, простой текстовый интерпретатор) хорошо вписываются во встроенные системы на базе 8-ми разрядных однокристальных микро-ЭВМ, таких как Intel MCS51, Atmel AVR, Microchip PIC, и с точки зрения бюджета разработки и с точки зрения ресурсоемкости тестового ПО.

В настоящее время коллективом лаборатории производятся работы по совершенствованию средств тестирования встроенных систем широкого спектра сложности. В работе над новыми технологиями преследуются следующие цели и задачи:

- уменьшение трудоемкости и увеличение качества тестирования за счет автоматизации максимального количества этапов и исключения человеческого фактора;
- построение комплексных систем автоматизированного тестирования (функциональное, климатическое, механическое, электромагнитное тестирование);
- создание и исследование гомогенной инструментальной среды на базе инструментальных серверов для распределенных многопроцессорных встроенных систем.

Литература

1. Распределенные системы управления / Ключев А.О., Кустарев П.В., Платунов А.Е. // СПб.: Сб. тезисов ДИМЭБ-97, 1997
2. Использование интерфейса JTAG для отладки встраиваемых систем / Ключев А.О., Коровьякова Т.А., Платунов А.Е. // Изв. вузов. Приборостроение. 1998. Т. 41. № 5
3. Инструментальный сервер / Ключев А.О., Платунов А.Е. // СПб.: Сб. тезисов ДИМЭБ-96, 1996
4. Механизмы обеспечения функциональной безопасности транспортных информационно-управляющих систем / Гавриков В.О., Платунов А.Е., Шубинский И.Б. // СПб.: Тез. VII Международной конференции "Региональная информатика 2000". Часть 1. СПб, СПОИСУ, 2000. С. 121.
5. Микропроцессорная система диспетчерской централизации "Тракт" нового поколения. / Гавриков В.О. Григорьев В.В. Платунов А.Е. Шубинский И.Б. // Тезисы докладов конференции "Информационные технологии на железнодорожном транспорте". Санкт-Петербург, 1997, с.163-173.
6. Восьмиразрядные микроконтроллеры в системах автоматического управления / А. Ключев, П. Кустарев, А. Платунов // Компоненты и технологии. 2001. №1. С.23–24.

МЕТОД ОБЕСПЕЧЕНИЯ ЗАМКНУТОСТИ ПРОГРАММНОЙ СРЕДЫ ПУТЕМ РАЗГРАНИЧЕНИЯ ПРАВ ДОСТУПА К ФАЙЛОВОЙ СИСТЕМЕ

И.П. Павличенко

Статья посвящена проблемам реализации для ОС семейства MS Windows одного из основных механизмов защиты: обеспечению замкнутости программной среды, т.е. обеспечению невозможности запуска пользователем несанкционированных процессов. В ней освещаются основные недостатки существующих реализаций этого механизма, и предлагается новый подход, позволяющий устранить эти недостатки.

Обеспечение замкнутости программной среды является, согласно руководящим документам Гостехкомиссии РФ, обязательным механизмом в системах защиты информации (СЗИ): "КСЗ должен контролировать доступ наименованных субъектов (пользователей) к наименованным объектам (файлам, программам, томам и т. д.)" [1]. Это необходимо для исключения возможности запуска пользователем разного рода снифферов, программ взлома или подбора паролей, инструментальных средств и т.п.

ОС семейства MS Windows не имеют встроенных средств обеспечения замкнутости программной среды, удовлетворяющих РД Гостехкомиссии РФ [2]. Существующие в данное время реализации этого механизма в различных СЗИ опираются на список разрешенных пользователю для запуска процессов, т.е. каждому пользователю ставится в соответствие свой собственный список разрешенных программ. Такой подход прост, но имеет некоторые недостатки: необходимо перечислять в этих списках все процессы, разрешенные к запуску пользователем, в том числе и процессы, порождаемые разрешенными процессами, так как ни одна из существующих СЗИ не различает, кто запустил процесс – пользователь или другой процесс. Это приводит к тому, что списки разрешенных процессов становятся очень громоздкими, существенно снижается производительность системы. За счет больших затрат времени на анализ таких списков при каждом запуске программы (процесса) сильно усложняется администрирование подобной СЗИ, так как администратору безопасности приходится составлять такие списки для каждого пользователя, а при добавлении новых программ изменять списки у каждого пользователя, пользующегося этой новой программой. Соответственно, при удалении некоторой программы приходится удалять из списков все процессы, порождаемые удаляемой программой. Кроме того, возникает проблема с введением в список разрешенных процессов отладочных и иных программ, которые необходимо запускать эксплуатационным службам при работе некоторого пользователя в системе, а этому пользователю запускать эти программы запрещено. Для исключения подобной ситуации требуется различать субъект запуска – пользователь или процесс – и назначать процессу свои собственные списки программ, а также динамически изменять список разрешенных процессов у пользователя. Ни одна из существующих СЗИ для ОС семейства MS Windows не позволяет этого.

Предлагаемый метод позволяет устранить (в некоторой степени) все перечисленные недостатки, а также добавить некоторые новые возможности в реализуемые механизмы защиты. Основу метода составляет использование для решения задачи обеспечения замкнутости программной среды механизма разграничения доступа к файловой системе – подсистема разграничения дискреционным методом прав доступа (РПД) пользователей к файловой системе. Вышеприведенная формулировка требований РД не включает процесс в число субъектов доступа. В ОС семейства MS Windows следование этому приводит к невозможности определения для процесса собственных прав доступа и, соответственно,

невозможности разрешения описанных выше тупиковых ситуаций. Поэтому предлагается добавить в подсистему РПД возможность определять, какой процесс ведет обращение к файловой системе, возможность рассматривать его (процесс) как отдельный субъект доступа, а также возможность определять обращение к ресурсам файловой системы с целью запуска программ.

Основу реализации метода составляет выполнение следующих действий средствами подсистемы разграничения прав доступа.

- Выделить для каждого пользователя (или группы пользователей) каталоги, из которых разрешен запуск программ. Это требует от администратора СЗИ проводить установку пользовательских программ только в такие каталоги.
- Запретить всем пользователям операции записи в эти каталоги.
- Запретить запуск программ из всех каталогов, куда пользователю (или группе пользователей) разрешена запись.

Настроенная таким образом подсистема РПД будет, наряду со своими непосредственными функциями, обеспечивать целый ряд новых возможностей. Это позволит не только удобно управлять составом программ (т.к. нет необходимости корректировки списков прав пользователей, или такая корректировка сводится к добавлению всего одной строки), но также исключает необходимость перечисления всех процессов, которые запускает новая программа, в списке разрешенных. В альтернативном варианте необходимость подобного перечисления приводила к тупикам: программе для работы требовался некоторый процесс, а разрешать запускать этот процесс пользователю нельзя. Так как любой процесс рассматривается подсистемой РПД как еще один самостоятельный субъект доступа, становится возможным управлять именно его (процесса) правами. Поэтому вместо внесения программ, необходимых для работы некоторому процессу, в список пользователя, вызвавшего процесс, эти программы добавляются только в список этого процесса. Это позволяет избежать упомянутых выше тупиковых ситуаций. Кроме того, использование этого механизма обеспечивает возможность обработки некоторых файлов только с помощью специальных программ, а также запуск пользователем некоторых программ только с помощью других программ, что невыполнимо в существующих реализациях СЗИ. Наконец, такая подсистема РПД позволяет разграничивать права доступа процессам без учета пользователей, что позволяет выделять привилегированные процессы, т.е. процессы, имеющие доступ к файлам и программам, недоступным для пользователя, что позволяет закрыть для операций записи пользователю системные каталоги ОС. В существующих системах подобное ограничение приводило к невозможности функционирования самой ОС.

Литература

1. Гостехкомиссия России. Руководящий документ "Средства вычислительной техники, Защита от несанкционированного доступа к информации, Показатели защищенности от несанкционированного доступа к информации".. Москва, 1992.
2. Рихтер. Windows для профессионалов. М.: MS Пресс, 1999.

РЕШЕНИЕ СПЕЦИФИЧЕСКИХ ЗАДАЧ АУДИТА БАЗЫ ДАННЫХ

Г.Ю. Громов, Д.А. Дорохин, В.С. Черемухин

Рассматриваются новые средства СУБД Oracle и предлагается их использование для расширения стандартных возможностей, связанных с механизмом сбора данных по функционированию информационной системы, обработкой исключительных ситуаций и повышению производительности системы.

Вопросы, связанные с аудитом базы данных, в СУБД Oracle решены давно и вполне успешно. В СУБД Oracle существует стандартный, довольно громоздкий, но при этом обладающий большими возможностями механизм сбора информации аудита базы данных. На любом уровне детализации можно отследить события, происходящие в базе данных, вплоть до конкретных запросов, выполняемых определенным пользователем. Однако использование средств аудита, помимо естественно возникающих накладных расходов, связанных с дополнительной загрузкой базы данных, вызывает и ряд административных сложностей. Обязательно должна быть тщательным образом разработана и отлажена стратегия аудита базы данных, которая должна предусматривать своевременную очистку системной таблицы SYS.AUD\$, размещенную в табличном пространстве SYSTEM. Если не уделить этому вопросу должного внимания, постоянное изменение размера системной таблицы SYS.AUD\$ может привести к фрагментации табличного пространства SYSTEM, что просто недопустимо, так как избавиться от него можно только пересозданием базы данных. Наилучшим решением этой проблемы было бы перемещение системной таблицы SYS.AUD\$ в другое табличное пространство, однако Oracle официально заявляет о непредсказуемых последствиях данного шага.

К тому же, несмотря на все возможности по сбору информации аудита, предоставляемые стандартными средствами Oracle, возникают специфические задачи, решение которых с использованием стандартных средств невозможно.

Однако в Oracle, начиная с версии 8i, появилось средство, которое и должно помочь администраторам и разработчикам решить большинство проблем, возникающих при сборе специфической информации аудита. Это средство называется "триггера событий".

В отличие от "классических" триггеров базы данных, срабатывающих в случае выполнения операций INSERT, UPDATE, DELETE, триггера событий срабатывают при возникновении как-либо системных событий, среди которых SERVERERROR, AFTER LOGON, BEFORE LOGOFF, AFTER/ BEFORE ALTER/CREATE/DROP, AFTER/ BEFORE GRANT/REVOKE и так далее. По сути, данный тип триггеров позволяет, помимо сбора информации аудита, выполнять набор необходимых действий, связанных с тем или иным системным событием. Триггера событий могут создаваться на двух уровнях – на уровне базы данных и на уровне конкретной схемы. Также только в триггерах событий доступен набор атрибутов этих событий, например, стек ошибок (ora_server_error) для события SERVERERROR или ip-адрес компьютера пользователя (ora_client_ip_address) для события AFTER LOGON.

В качестве простейшего примера можно привести два триггера, выполняющих сбор информации аудита в созданную администратором базы данных таблицу. Таблица создается не в табличном пространстве SYSTEM, и помимо информации, которую можно собрать стандартными средствами (время начала и окончания работы с базой данных), в нее заносится информация о том, какая программа используется пользователем для работы, и ip-адрес компьютера, с которого он работает.

```
DROP TABLE adm_connections;  
CREATE TABLE adm_connections
```

```

(s_id NUMBER ,
 user_name    VARCHAR2(30),
 host_ip      VARCHAR2(200),
 progr        VARCHAR2(200),
 begin_date   DATE DEFAULT sysdate,
 end_date     DATE)
TABLESPACE tools;

CREATE OR REPLACE TRIGGER OnLogon AFTER LOGON ON DATABASE
DECLARE
  -- описываем служебную переменную
  pr VARCHAR2(64);
BEGIN
  BEGIN
    -- выбираем из представления v$session информацию о программе пользователя
    SELECT program INTO pr FROM sys.v_$session t where audsid =
sys_context('USERENV','SESSIONID');
  EXCEPTION
    WHEN NO_DATA_FOUND THEN NULL;
  END;
  -- вставляем всю необходимую информацию в нашу таблицу
  INSERT INTO adm_connections(s_id,user_name,host_ip,progr)
    VALUES (sys_context('USERENV','SESSIONID'), ora_login_user,
ora_client_ip_address, pr);
  END;

CREATE OR REPLACE TRIGGER OnLogoff BEFORE LOGOFF ON DATABASE
BEGIN
  -- обновляем нашу таблицу сведениями об окончании работы с базой данных
данного пользователя
  UPDATE adm_connections SET end_date = sysdate WHERE s_id =
sys_context('USERENV','SESSIONID');
  END;

```

Триггера событий также очень полезны и разработчикам. При использовании многоуровневых архитектур не так просто организовать детальный сбор информации о возникающих при работе приложения ошибок. Здесь также неоценимую помощь оказывают триггера событий. Вместо того, чтобы включать программный код каждой процедуры/функции/пакета в обработчик исключений команды по сохранению возникшей ошибки, достаточно написать триггер на событие SERVERERROR на уровне схемы или базы данных. С его помощью возможно, например, сохранять все ошибки возникающие при работе приложения на уровне базы данных.

```

CREATE OR REPLACE TRIGGER OnError AFTER SERVERERROR ON
DATABASE
BEGIN
  INSERT INTO adm_errors(user_name,host_ip, error_nom)
    VALUES (ora_login_user, SYS_CONTEXT('USERENV','IP_ADDRESS'),
SYS_CONTEXT('USERENV','CLIENT_INFO'), 'ORA-
'||TO_CHAR(ora_server_error(1),'00000'));
  END;

```

Рассмотренные примеры использования триггеров событий лишь частично иллюстрируют их возможности. Использование их в полном объеме позволяет администраторам и разработчикам находить простые и гибкие решения практически любых специфических задач, связанных с аудитом базы данных.

Литература

1. Васильев В.Н. Принципы создания интегрированной информационно-аналитической системы управления вузом // Тезисы докладов Международной конференции "Интернет, Общество, Личность - ИОЛ-2000" Институт "Открытое общество", Санкт-Петербург, 2000.
2. Кириллов В.В., Громов Г.Ю. Применение CASE-технологии при создании интегрированной информационной системы университета // Тез. докл. Юбилейной научно-технической конференции профессорско-преподавательского состава, посвященной 100-летию университета, 29-31 марта 2000 года, Санкт-Петербург, 2000.
3. Кириллов В.В., Громов Г.Ю., Штивельман Я.Е. Первый опыт проектирования несколькими вузами подсистемы "Учебный процесс" // Тезисы докладов Международной конференции "Интернет, Общество, Личность - ИОЛ-2000" Институт "Открытое общество", Санкт-Петербург, 2000.
4. Кириллов В.В., Громов Г.Ю., Черемухин В.С., Вотинцев А.А., Романенко В.А. Работа с одним репозиторием Oracle Designer территориально распределенных разработчиков // Тезисы докладов Международной научно-методической конференции "Телематика'2000", Санкт-Петербург, 2000.

ПРОБЛЕМА ИНТЕГРАЦИИ ЛОКАЛЬНЫХ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ В ОБРАЗОВАТЕЛЬНОЙ СФЕРЕ ДЕЯТЕЛЬНОСТИ

Г.Ю. Громов, Д.А. Дорохин, В.В. Кириллов, В.С. Черемухин

На примере опыта разработки "Интегрированной информационно-аналитической системы управления вузом" и "Информационной системы управления вузом" производится обзор возможных подходов к проектированию автоматизированных систем управления и дается рекомендация по выбору их рациональной структуры.

Сегодня во многих российских вузах существуют разработанные и (или) приобретенные информационные системы и подсистемы оперативной обработки данных, реализованные на самой различной аппаратной и программной основе. Это и автономные файловые системы, и системы, использующие локальные или сетевые версии персональных систем управления базами данных (СУБД), и системы, построенные на серверах реляционных баз данных. Однако все они создавались для автоматизации деятельности сотрудников отдельных подразделений, и поэтому им трудно (а часто и невозможно) обмениваться данными, трудно выдать непредусмотренную справку, трудно "узнать", существуют ли в соседних системах данные, затребованные кем-либо из руководства вуза. Естественно, что такие данные часто противоречивы, и руководство университета не может получить полной и достоверной информации на многие актуальные вопросы [1].

Вот несколько примеров противоречивости:

- один и тот же сотрудник существует какое-то время под разными табельными номерами и оформлен в разных подразделениях;
- встречаются случаи назначения на стипендию неуспевающих студентов;
- не совпадают сведения о численности студентов и преподавателей, поступающие из подсистем отдела кадров, учебной части, деканатов, кафедр.

Непрерывно нарастающее число подобных противоречий в существующих системах и явилось основным побудительным моментом необходимости их замены.

Аналогичная проблема возникает и при разработке автоматизированных систем управления (АСУ) министерства образования РФ или небольших автономных АСУ филиала университета, факультета, кафедры, секции, а также любого подразделения образовательного характера.

На разных уровнях управления эти локальные АСУ имеют неодинаковый характер и преследуют различные цели. Однако в них необходимо хранить частично пересекающуюся информацию (например, кафедры, факультеты и университет хранят в своих АСУ общие сведения об одних и тех же студентах и преподавателях). Как было отмечено выше, такое дублирование данных, как правило, приводит к появлению противоречивых сведений, не позволяющих получать достоверную информацию.

Дублирование можно исключить при создании единой интегрированной базы данных, используемой всеми АСУ. Однако создание единой базы данных для больших и территориально разделенных систем затруднено из-за большого объема информации и удаленного доступа (нереально объединение АСУ министерства с АСУ входящих в него образовательных учреждений).

Поэтому возникает задача исключения дублирования информации локальных АСУ единой корпоративной образовательной системы.

Одним из решений этой задачи является выявление общих данных для различных уровней управления и интегрирование их (при исключении дублирования) в базах данных более высокого уровня. При этом частные данные размещаются (без дублирования) на более низких уровнях. Например, в университетской системе интегрируются сведения о бухгалтерской, хозяйственной, учебной деятельности

университета, санкционированную часть этой информации (сведения о сотрудниках, студентах, подразделениях и т.д.) могут использовать кафедры. В свою очередь АСУ университета может использовать информацию, предоставляемую кафедрами (списки трудов сотрудников, научные разработки, участие в конференциях, сведения о выпускниках, сведения о НИР, публикации и т.д.).

Для обмена информацией между уровнями управления необходимо создать и использовать стандарты, реализуя их в базе данных и приложениях.

В случае, когда одна из систем не предоставляет информации, нужной партнеру, тот вынужден хранить эту информацию в своей базе данных, т.е. создавать дубликат. Например, отдел кадров не дает выпускающей кафедре доступа к полной информации о ее сотрудниках и студентах. Если кафедра собирает и хранит часть такой информации, то она может не совпадать с данными отдела кадров. Для исключения противоречий можно создать процедуры сравнения данных и выявления различий.

В системе может быть предусмотрено умышленное дублирование информации, необходимое для обеспечения контроля правильности данных – например, ввод экзаменационных оценок в деканате и на кафедре. При этом будут выявляться ошибки ввода информации из экзаменационных ведомостей и листов.

Таким образом, при проектировании корпоративных АСУ необходимо по возможности стремиться к созданию единой интегрированной базы данных с использованием всех средств обеспечения целостности, непротиворечивости и безопасности данных. В случае, если не удастся спроектировать единую систему, необходимо создавать систему взаимосвязанных баз данных, а также стандартов и механизмов обмена информацией между этими базами данных.

Литература

1. Васильев В.Н. Принципы создания интегрированной информационно-аналитической системы управления вузом // Тезисы докладов Международной конференции "Интернет, Общество, Личность - ИОЛ-2000" Институт "Открытое общество", Санкт-Петербург, 2000.
2. Кириллов В.В., Громов Г.Ю. Применение CASE-технологии при создании интегрированной информационной системы университета // Доклад на Юбилейной научно-технической конференции профессорско-преподавательского состава, посвященная 100-летию университета 29-31 марта 2000 года, Санкт-Петербург, 2000.
3. Кириллов В.В., Громов Г.Ю. Штивельман Я.Е., Первый опыт проектирования несколькими вузами подсистемы "Учебный процесс" // Тезисы докладов Международной конференции "Интернет, Общество, Личность - ИОЛ-2000" Институт "Открытое общество", Санкт-Петербург, 2000.
4. Кириллов В.В., Громов Г.Ю., Черемухин В.С., Вотинцев А.А., Романенко В.А. Работа с одним репозиторием Oracle Designer территориально распределенных разработчиков // Тезисы докладов Международной научно-методической конференции "Телематика'2000", Санкт-Петербург, 2000.

УНИВЕРСАЛЬНАЯ МОДЕЛЬ ПОСЛЕДОВАТЕЛЬНОСТНЫХ СХЕМ В САПР

А.Г. Зыков, О.Ф. Немолочнов, В.И. Поляков

Предлагается и описывается универсальная модель последовательностной схемы для САПР, адекватно отображающая поведение логической схемы. Определяется основной класс задач, решаемых с использованием предложенной модели

Для решения задач САПР требуется модель, адекватно отображающая поведение логической схемы и удобная при машинной реализации. Для дальнейшего рассуждения определимся в основных понятиях. Логическая схема называется комбинационной (КС), если значения ее выходов зависят только от значений ее входов. Логическая схема называется последовательностной (ПС), если значения ее выходов зависят как от значений ее входов, так и от их предыстории, другими словами, ПС – это схема с обратными связями.

На основании проведенных исследований и практики проектирования и анализа цифровых систем в качестве модели логической схемы предлагается модель, представленная на рисунке. В отличие от классической модели Хаффмана [1, 2], предлагаемая модель учитывает современные тенденции в схемотехнике, в частности, применение двухступенчатых триггеров (M-S) [3]; синхронизацию (при этом если синхросигнал не отличать от других входов схемы, то модель будет отображать основной класс асинхронных схем); понятие сигнала системного сброса (начальной установки) и др. Это значительно уточняет модель при использовании машинно-ориентированных методов проектирования и анализа схем.

Модель схемы условно разбита на комбинационную и последовательностную (П) части. Здесь вектор X есть независимые входы схемы, из которых особо выделены сигнал синхронизации T и сигнал начальной установки элементов памяти R , который может быть как асинхронным сбросом, так и синхронизированным тактом T , что в общем случае не является принципиальным. Значения Y' определены как булевы функции $Y' = Y'(X, Y)$. Значение сигналов обратной связи Y разбито на два подмножества Y^M и Y^S , соответственно $Y' = Y^M \cup Y^S$. Значения выходных сигналов также определены как $Z = Z(X, Y)$ для автомата Мили и как $Z = Z(Y)$ для автомата Мура. Для схемы также определены некоторые промежуточные сигналы Z_j как выходы КС, которые являются вспомогательными и введены как следствие разбиения КС на одновыходные комбинационные подсхемы или с целью уменьшения размерности решаемых задач. Значение вектора сигналов Y определяет предыдущее значение состояния схемы, а значение вектора Y' – последующее состояние. В общем случае векторы Y и Y' задают некоторые классы состояний. Любой сигнал y и y' может принимать три значения: 0, 1 и x , где x есть безразличное состояние, которое произвольным образом можно доопределить в значение 1 или 0, если это не вызывает противоречия при переходе схемы из состояния Y в состояние Y' .

Соответствие между значениями сигналов Y и Y' определяется по интервальным кубическим покрытиям $C(y') = C^x(y') \cup C^p(y')$, где $C^x(y')$ есть область установки, а $C^p(y')$ – область хранения. Если некоторое значение y' вычислено кубом $C \in C^x(y')$, то $y = x$ и $y' = p$, если же значение y' вычислено кубом $C \in C^p(y')$, то значение $y = y' = p$.

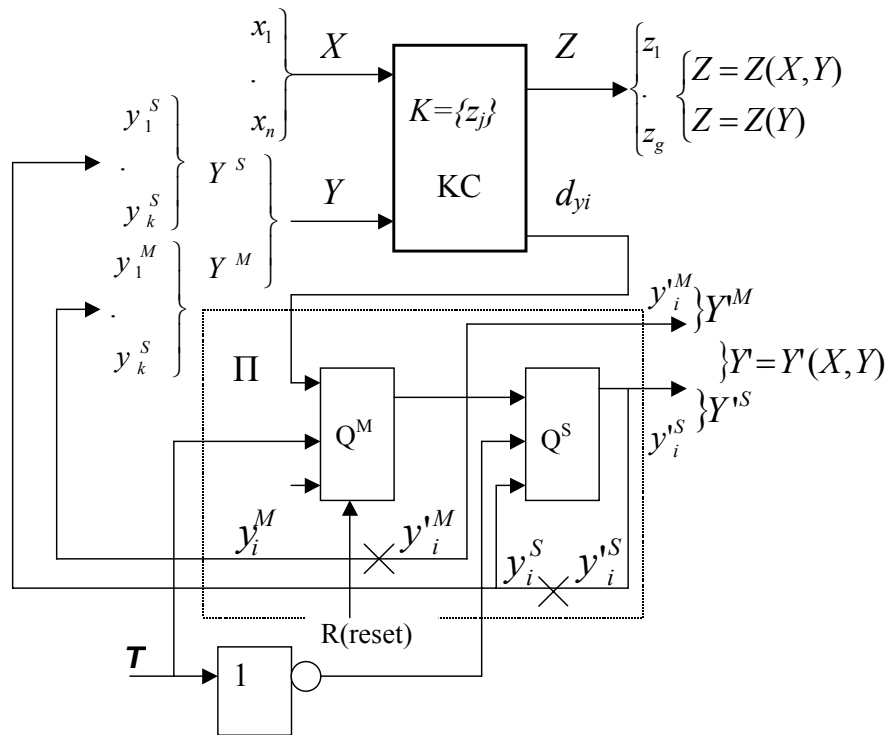


Рис. Универсальная модель последовательностной схемы

В соответствии с предложенной универсальной моделью схемы введём понятие куба единого формата $\Phi_0 = X Y^M Y^S Z_K Y'^M Y'^S Z$. Данный формат в векторном виде описывает все значения сигналов и их разбиение на функциональные классы. Если не делать различия между классами Y^M , Y^S , Y'^M и Y'^S , то можно ограничиться форматом $\Phi_1 = X Y Z_K Y' Z$. Далее, если Z_K отсутствуют, можно ограничиться форматом $\Phi_2 = X Y Y' Z$. Наконец, для случая, когда рассматриваются только переходы схемы, можно перейти к формату $\Phi_3 = X Y Y'$, что является вполне допустимым при теоретическом рассмотрении состояний и переходов схемы, так как значения Z_K и Z являются импликациями от независимых входов X и сигналов обратной связи Y или Y' , и их можно вычислить после построения переходов.

По предложенной универсальной модели, в отличие от модели функциональных неисправностей [4], могут быть решены не только задачи синтеза тестовых последовательностей, но и следующие задачи САПР:

- моделирование;
- анализ состязаний;
- построение переходов схемы в прямом и в обратном направлении;
- верификации иерархических моделей;
- другие задачи анализа.

Литература

1. Huffman D. The design and use of hazard tree switching network // J. Of ACM. 1957. V. 4. №1. P. 37–40.
2. Чжен Г., Меннинг Е., Метц Г. Диагностика отказов цифровых вычислительных систем. М.: Мир, 1972.
3. Баранов С.И. Синтез микропрограммных автоматов (граф-схемы и автоматы). Л.: Энергия, 1979.
4. Немолочнов О.Ф. Щупак Ю.А. Модель функциональных неисправностей для автоматного представления первичных элементов логической схемы // Автоматика и телемеханика. 1993. №5. С. 167–179.

ПОСТРОЕНИЕ КОМПЛЕКСНОГО ПОКРЫТИЯ ПОСЛЕДОВАТЕЛЬНОСТНЫХ СХЕМ МЕТОДОМ ПЕРЕСЕЧЕНИЯ ПОКРЫТИЙ СИСТЕМ БУЛЕВЫХ ФУНКЦИЙ

А.Г. Зыков, О.Ф. Немолочнов, В.И. Поляков

Предлагается и описывается новый метод построения моделей последовательностных схем в виде комплексного покрытия. Определяется круг задач, решаемых на предложенной модели в системах автоматизированного проектирования.

Процесс проектирования логических схем представляет собой последовательность этапов синтеза и анализа иерархических моделей вплоть до конкретной реализации схемы в заданном элементном базисе. Важным фактором достоверной реализации проекта является соблюдение адекватности иерархических моделей на различных стадиях проектирования, что достигается решением задач анализа и верификации. Для решения задач анализа закон функционирования схемы описывается тем или иным способом. Для систем автоматизированного проектирования одним из наиболее компактных и информационных является описание схем в виде кубических покрытий [1–3]. Схема декомпозируется на одновыходные подсхемы, каждая из которых описывается вырожденными покрытиями $C(f_i)$ в локальном формате $\Phi_l = \{X, Y, Y', Z\}$, где X – независимые входы, Y – сигналы обратной связи, определяющие исходное состояние схемы, Y' – сигналы обратной связи, определяющие последующее состояние схемы, Z – выходные или внутренние значения комбинационных элементов (подсхем) схемы. Так как значения Z являются импликациями от независимых входов X и сигналов обратной связи Y и Y' , то их можно вычислить после построения переходов схемы. Поэтому можно перейти к локальному формату $\Phi_{l1} = \{X, Y, Y'\}$, что является допустимым при теоретическом рассмотрении состояний и переходов схемы.

Построение комплексного покрытия всей схемы позволяет получить полную модель исправной схемы для решения требуемых задач анализа, что выгодно отличает её от моделей, предложенных в [2]. Комплексное покрытие (КС) строится в глобальном формате $\Phi_{кс} = \{X, Y, Y', Z\}$. Любой куб комплексного покрытия $k_i \in КС$ трактуется как некоторое состояние логической схемы. КС строится методом пересечения множеств кубов покрытий подсхем на основе полного перебора, т.е. всех возможных сочетаний. Таким образом:

$$КС = \bigcap_{i=1}^n C(f_i) - \{a \mid a \subseteq b; a, b \in КС\},$$

где n – количество покрытий подсхем.

Само КС есть множество простых кубов, адекватно отображающих все состояния и переходы последовательностной схемы. Простой куб $кс_i$ есть такой куб, в котором ни одну координату со значением $p = 1$ или 0 нельзя заменить на x без нарушения импликации $y'_i = p_i$ и $z_j = p_j$.

Сокращение перебора может быть достигнуто методами импликации значений $C(y'_i)$ и $C(Z)$, а также методами локальной и глобальной оптимизации на основе операции поглощения кубов.

Рассмотрим на примере построение КС последовательностной схемы с глобальной обратной связью.

Пусть имеется схема пересчёта (рис. 1), состоящая из двух триггеров: DT1 – синхронный DT триггер и DT2 – синхронный DT – триггер со сбросом, после декомпозиции и являющиеся одновыходными подсхемами.

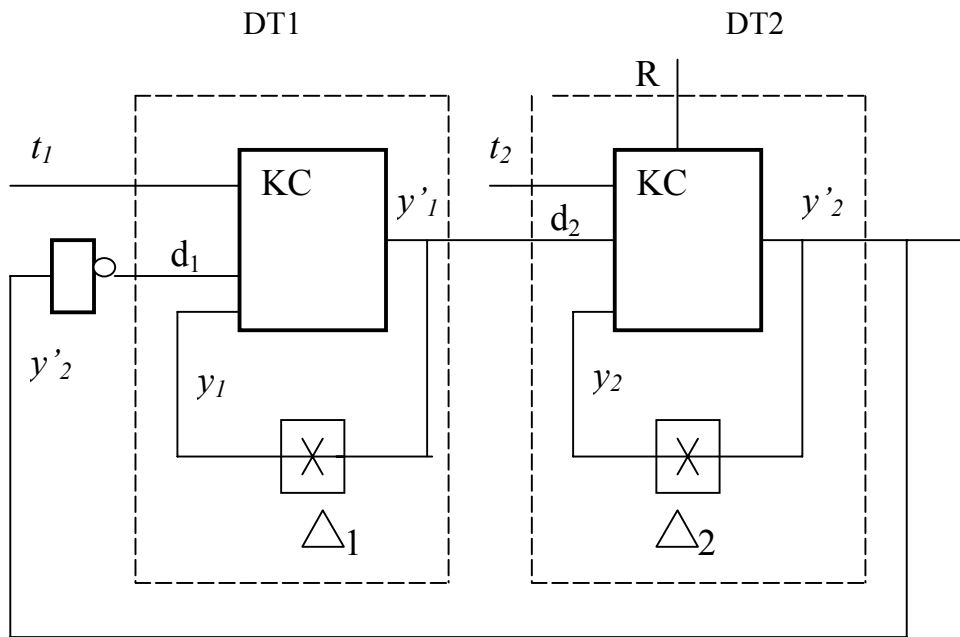


Рис.1. Схема пересчёта

Вырожденные покрытия $C(y'_1)$ и $C(y'_2)$ этих подсхем представлены на рис. 2.

В соответствии с предложенным алгоритмом пересечения и методами сокращения перебора построим комплексное покрытие схемы пересчёта в глобальном формате $\Phi_{\text{КС}} = \{R, t_1, t_2, y_1, y_2, y'_1, y'_2\}$. Отметим некоторые моменты при построении КС. Покрытия в пересечение могут вступать в любом порядке при условии, что выбранный куб из некоторого покрытия должен быть пересечён со всеми кубами из других покрытий.

$$C(y'_1) = \begin{array}{c|ccccc} t_1 & d_1 & y_1 & y'_1 & \\ \hline 1 & 1 & x & 1 & c_1 \\ x & 1 & 1 & 1 & c_2 \\ 0 & x & 1 & 1 & c_3 \\ 1 & 0 & x & 0 & c_4 \\ x & 0 & 0 & 0 & c_5 \\ 0 & x & 0 & 0 & c_6 \end{array}$$

$$C(y'_2) = \begin{array}{c|ccccc} R & t_2 & d_2 & y_2 & y'_2 & \\ \hline 0 & 1 & 1 & x & 1 & c_7 \\ 0 & x & 1 & 1 & 1 & c_8 \\ 0 & 0 & x & 1 & 1 & c_9 \\ 1 & x & x & x & 0 & c_{10} \\ x & 1 & 0 & x & 0 & c_{11} \\ x & x & 0 & 0 & 0 & c_{12} \\ x & 0 & x & 0 & 0 & c_{13} \end{array}$$

Рис. 2. Вырожденные покрытия подсхем DT1 и DT2

Приведём примеры пересечения некоторых кубов из покрытий $C(y'_1)$ и $C(y'_2)$ с учётом, что $d_1 = \overline{y'_2}$ и $d_2 = y'_1$.

$$\begin{array}{c|ccccccc} R & t_1 & t_2 & y_1 & y_2 & y'_1 & y'_2 & \\ \hline c_1 & x & 1 & x & x & x & 1 & 0 \\ \cap c_{10} & 1 & x & x & x & x & x & 0 \\ \hline 1 & 1 & x & x & x & 1 & 0 & - k_1 - \text{куб комплексного покрытия.} \end{array}$$

$$\begin{array}{c} c_2 \\ \cap c_{10} \end{array} \begin{array}{|ccccccc|} \hline x & x & x & 1 & x & 1 & 0 \\ \hline 1 & x & x & x & x & x & 0 \\ \hline \end{array}$$

$$\begin{array}{|ccccccc|} \hline 1 & x & x & 1 & x & 1 & 0 \\ \hline \end{array} - k_3 - \text{куб комплексного покрытия.}$$

$$\begin{array}{c} c_3 \\ \cap c_9 \end{array} \begin{array}{|ccccccc|} \hline x & 0 & x & 1 & x & 1 & x \\ \hline 0 & x & 0 & x & 1 & x & 1 \\ \hline \end{array}$$

$$\begin{array}{|ccccccc|} \hline 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline \end{array} - k_m - \text{данный куб поглощается кубом } k_6 \text{ (таблица 1)}$$

$$R \quad t_1 \quad t_2 \quad y_1 \quad y_2 \quad y'_1 \quad y'_2$$

$$\begin{array}{c} c_3 \\ \cap c_{10} \end{array} \begin{array}{|ccccccc|} \hline x & 0 & x & 1 & x & 1 & x \\ \hline 1 & x & x & x & x & x & 0 \\ \hline \end{array}$$

$$\begin{array}{|ccccccc|} \hline 1 & 0 & x & 1 & x & 1 & 0 \\ \hline \end{array} - k_j - \text{данный куб поглощается кубом } k_3$$

$(k_3 \supset k_j)$ и из дальнейшего рассмотрения исключается. Отметим также, что, например, пересечение кубов c_1 и $c_2 \in C(y'_1)$ с кубами c_7, c_8 и c_9 пустое ($= \emptyset$) по координате y'_2 с учётом того, что $d_1 = \overline{y'_2}$. Выполнив всю процедуру построения комплексного покрытия, получим покрытие всей схемы пересчета, представленное в таблице 1.

Таблица 1

$$KC = \begin{array}{c} R \quad t_1 \quad t_2 \quad y_1 \quad y_2 \quad y'_1 \quad y'_2 \\ \hline \left\{ \begin{array}{ccccccc|c} 1 & 1 & x & x & x & 1 & 0 & k_1 \\ x & 1 & 0 & x & 0 & 1 & 0 & k_2 \\ 1 & x & x & 1 & x & 1 & 0 & k_3 \\ x & x & 0 & 1 & 0 & 1 & 0 & k_4 \\ 0 & 0 & 1 & 1 & x & 1 & 1 & k_5 \\ 0 & 0 & x & 1 & 1 & 1 & 1 & k_6 \\ 0 & 1 & 0 & x & 1 & 0 & 1 & k_7 \\ 0 & x & 0 & 0 & 1 & 0 & 1 & k_8 \\ 1 & 0 & x & 0 & x & 0 & 0 & k_9 \\ x & 0 & 1 & 0 & x & 0 & 0 & k_{10} \\ x & 0 & x & 0 & 0 & 0 & 0 & k_{11} \end{array} \right. \end{array}$$

Построенное комплексное покрытие является алгебро-топологической моделью схемы и позволяет решать установочную задачу, задачи моделирования, анализа состязаний, синтеза тестовых последовательностей, построения переходов и состояний схемы в прямом и обратном направлении при верификации проекта.

Отметим, что предложенный метод представления и построения модели логической схемы на основании единственной операции пересечения кубов покрытий

позволяет автоматизировать соответствующие этапы проектирования цифровых систем.

Литература

1. Проектирование цифровых вычислительных машин./ Под ред. С.А. Майорова. М.: Высш. школа, 1972.
2. Чжен Г., Меннинг Е., Метц Г. Диагностика отказов цифровых вычислительных систем. М.: Мир, 1972.
3. Немолочнов О.Ф. Методы технической диагностики. Методические указания к курсовой работе. Л.: ЛИТМО, 1976.

ОЦЕНКА СЛОЖНОСТИ ГРАФА ФУНКЦИОНАЛЬНОЙ ПРОГРАММЫ

А.В. Лаздин, О.Ф. Немолочнов

Рассматриваются сравнительные характеристики схемы алгоритма функциональной программы и графа, представляющего ее. Доказывается, что сложность графовой формы представления функциональной программы сопоставима со сложностью нормализованной схемы ее алгоритма.

В соответствии с определением жизненного цикла программного обеспечения [1] программа как описание алгоритма решения некоторой задачи последовательно представляется такими формами, как спецификация, схема алгоритма, исходные тексты на различных языках программирования, исполняемый код. Под исполняемым кодом в данной работе понимается программа, предназначенная для выполнения на процессоре определенного типа и под управлением конкретной операционной системы. В идеале программа должна быть инвариантна всем этим формам представления, однако на практике, в ходе трансляции одной формы представления в другую, возможно внесение ошибок и, как результат, появление ненадежных программ. Для контроля корректности преобразований широко применяется метод верификации – доказательство соответствия описания алгоритма на разных уровнях абстракции [2].

Необходимо отметить, что главной целью создания программного продукта является получение исполняемого кода (ИК). Ненадежность ИК может быть результатом, например, следующих причин:

- ошибки на этапе кодирования;
- ошибки компилятора (к негативным воздействиям при формировании ИК можно отнести оптимизацию кода в ходе компиляции в тех случаях, когда в этом либо нет необходимости, либо оптимизация недопустима, например, при формировании программных задержек с использованием циклов);
- ошибки компоновки, которые возможны при сборке программы из нескольких исходных модулей, особенно в тех случаях, когда исходные модули разрабатывались на различных языках программирования.

ИК должен быть не только надежным, но и безопасным, т.е. не содержать таких разрушающих средств, как компьютерные вирусы и программные закладки [3].

Следовательно, ИК как итоговая форма представления исходного алгоритма должен пройти те же контрольные мероприятия (анализ, тестирование, верификация), что и другие формы представления программы.

Для проведения подобных мероприятий ИК должен быть преобразован в удобную для обработки форму представления. Наиболее естественной представляется графовая модель. В соответствии с [4] вершинами графа ИК будут команды управления последовательностью выполнения команд, а дугам соответствовать либо линейные участки последовательных команд обработки данных, либо пустые дуги, соответствующие безусловным переходам.

В данной работе дается оценка степени сложности такого представления в сравнении со схемой алгоритма исходной задачи. Известно, что схема любого алгоритма может быть представлена с помощью трех типов вершин: функциональной, предикатной и объединяющей (рис. 1).

Для одного алгоритма может существовать множество эквивалентных ему схем. Доказано, что любая схема алгоритма может быть представлена структурной схемой, элементами которой могут быть линейная последовательность функциональных вершин, альтернатива и итерация [5].

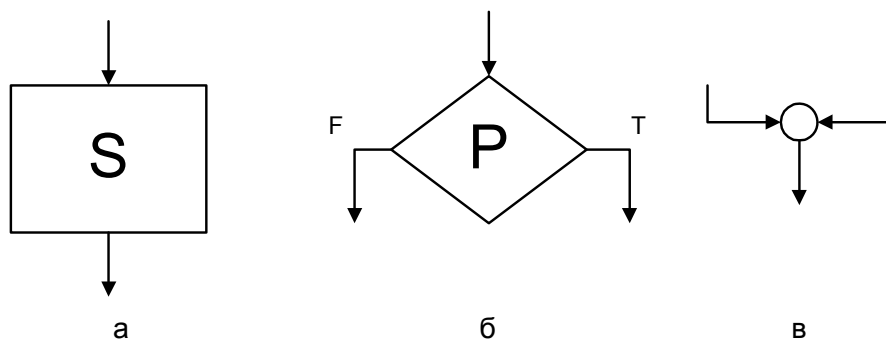


Рис. 1. Типы вершин схемы алгоритма:
а) функциональная; б) предикатная; в) объединяющая

Лемма 1 (о композиции функциональных вершин).

Алгоритм может быть представлен структурной схемой так, что путь между любыми двумя предикатными и(или) объединяющими вершинами будет проходить не более чем через одну функциональную вершину.

Справедливость данной леммы основана на том, что любую линейную последовательность функциональных операторов можно объединить в составной оператор (блок) без потери семантики. С позиций визуального формализма алгоритмов данная лемма проиллюстрирована на рис. 2.

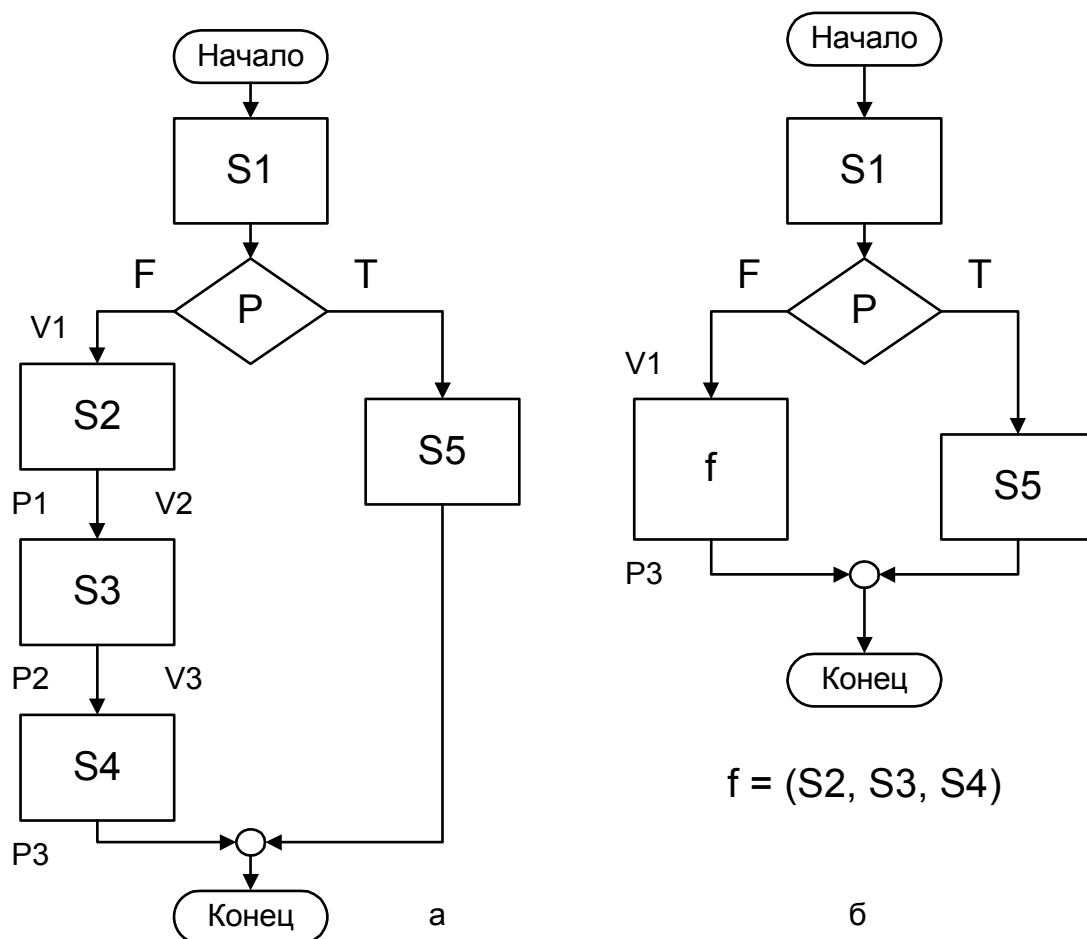


Рис. 2. Две эквивалентные формы представления алгоритма

Доказательство. Тождественность алгоритмов на рисунках 2,а и 2,б может быть доказана в терминах пред- и постусловий для функциональных вершин [6]. Пусть $V1$,

$V2$ и $V3$ – предусловия операторов (или последовательностей операторов), связанных с функциональными вершинами $S2$, $S3$ и $S4$, соответственно, а $P1$, $P2$ и $P3$ – соответствующие этим вершинам постусловия. Связь функциональной вершины $S2$ с соответствующими ей пред- и постусловием принято записывать в следующей форме:

$$\{V1\} S2 \{P1\} \quad (1)$$

Постусловие $P1$ совпадает с предусловием $V2$. В соответствии с этим

$$\{V1\} S2 \{V2\} \quad (2)$$

Следовательно,

$$V1 \Rightarrow V2 \ \& \ \{V2\} S3 \{P2\} \quad (3)$$

Выражение (3) эквивалентно

$$\{V1\} (S2, S3) \{P2\}. \quad (4)$$

Очевидно следующее обобщение (4), для любых n идущих подряд функциональных вершин:

$$\{V1\} (S1, S2, \dots, Sn) \{Pn-1\}. \quad (5)$$

Пусть f – составной оператор такой, что последовательно выполняются операторы из списка $S1, S2, \dots, Sn$. Тогда

$$\{V1\} f \{Pn-1\}. \quad (6)$$

Лемма доказана. ■

Будем называть схему алгоритма нормализованной, если в ней любая композиция операторов представлена одной функциональной вершиной в соответствии с леммой 1.

Возможные способы комбинирования предикатной и функциональных вершин, (альтернатива) показаны на рис. 3. Пусть ИК, соответствующий каждому из приведенных фрагментов схемы алгоритма, строится для процессора, набор команд которого включает условные двухадресные переходы (например, семейство процессоров Intel80x86). Тогда после перевода алгоритмов на язык ассемблера будут получены последовательности команд, приведенные в таблице 1. Очевидно, что в обоих случаях реализация ветвления требует помещения в ИК не более двух команд условного и безусловного переходов (в таблице мнемоники соответствующих команд выделены). Следовательно, каждому ветвлению будет соответствовать одна или две вершины графа ИК.

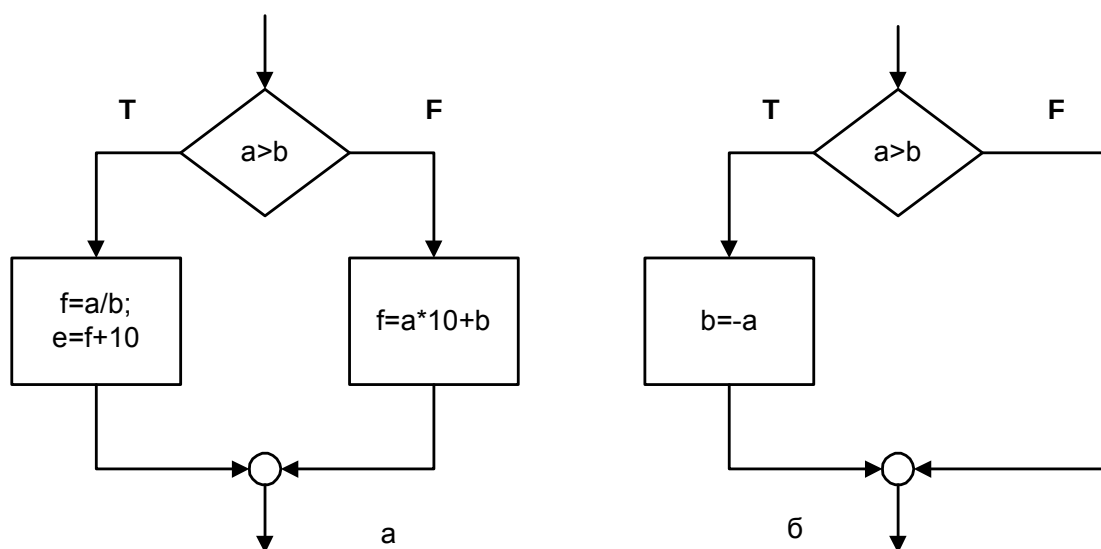


Рис. 3. Два способа связи предикатной и функциональных вершин.

Комбинация предикатных и функциональных вершин типа "итерация" могут быть представлены одним из вариантов, показанных на рис. 4. В таблице 2 приведены соответствующие им последовательности команд. В соответствии с полученной

программной реализацией схемам итерации будет соответствовать одна или две вершины графа ИК. Определим сложность графа как число его вершин. Назовем функциональной программой ИК, полученный при трансляции нормализованной схемы алгоритма.

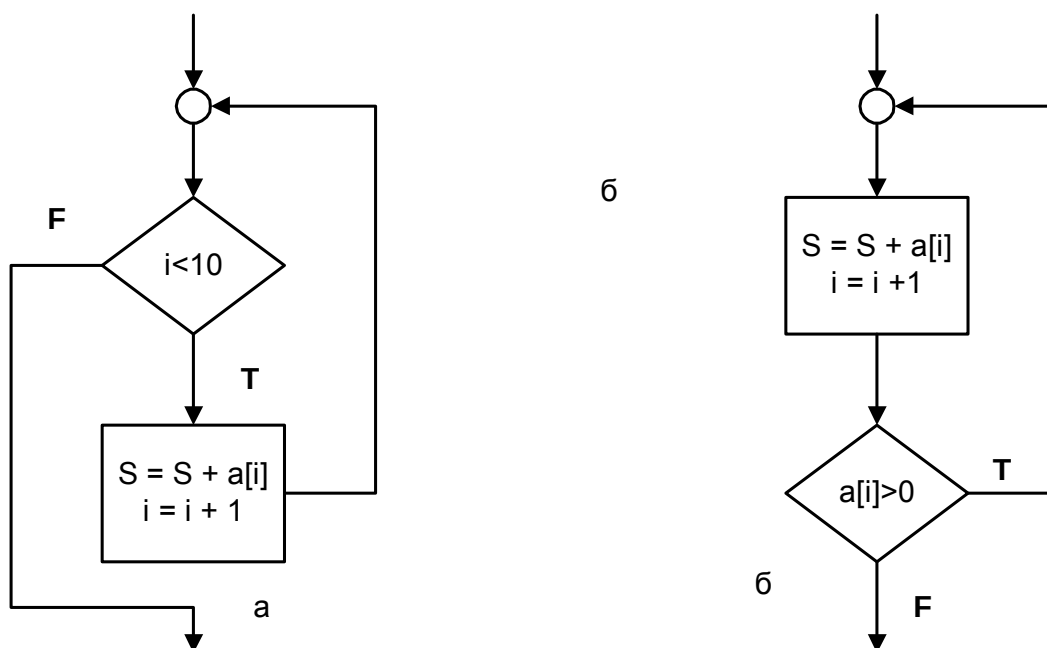


Рис. 4. Схемы итерации

Таблица 1

Реализация ветвления с двумя альтернативами.	Реализация ветвления с одной альтернативой
<pre> mov ax,a cmp ax,b jng metka1 ;последовательность команд ;реализующих первую альтернативу mov ax,a imulax,10 add ax,b mov f,ax jmp metka2 ;последовательность команд ;реализующих вторую альтернативу metka1: mov ax,a cwd div b mov f,ax add ax,10 mov e,ax metka2: ;последующие команды </pre>	<pre> mov ax,b cmp ax,b jng metka1 ;последовательность команд ;реализующих функциональную ;вершину mov ax,a neg ax mov b,ax metka1: ; последующие команды </pre>

Теорема. Сложность графа функциональной программы имеет тот же порядок, что и сложность нормализованной схемы исходного алгоритма описания задачи.

Доказательство. В соответствии с данными из табл. 3 число вершин $N_{НС}$ для нормализованной схемы алгоритма может быть определено следующим образом:

$$N_{НС} = 3 + 3N_{B1} + 2N_{B2} + 2N_{ИПр} + 2N_{ИПост} . \quad (7)$$

Число вершин $N_{ГФП}$ для графа функциональной программы:

$$N_{ГФП} = 2 + 2N_{B1} + N_{B2} + 2N_{ИПр} + N_{ИПост} , \quad (8)$$

где N_{B1} – число операторов выбора первого типа; N_{B2} – число операторов выбора второго типа; $N_{ИПр}$ – число итераций с предусловием; $N_{ИПост}$ – число итераций с постусловием.

Из анализа (7) и (8) очевидно, что число вершин графа функциональной программы не больше числа вершин нормализованной схемы алгоритма. Следовательно, они имеют один и тот же порядок сложности. ■

Полученные в статье результаты показывают, что обратное проектирование (reverse engineering) для исполняемого кода может быть программно автоматизировано, поскольку число вершин графа функциональной программы "обозримо" и зависит не столько от объема программы, сколько от числа ветвлений в ней.

Таблица 2

Реализация схемы итерации с предусловием.	Реализация схемы итерации с постусловием.
<pre> next: cmp i,10 jge exit mov ax,s mov si,i shl si,1 add ax,a[si] add i,10 jmp next exit: ; последующие команды </pre>	<pre> mov ax,s next: mov si,i shl si,1 add ax,a[si] add i,1 cmp a[si+2],0 jg next ; последующие команды </pre>

Таблица 3

№ п/п	Тип фрагмента программы	Число вершин	
		нормализованная схема алгоритма (без учета вершины типа «объединение»)	граф функциональной программы
1	линейная программа	3	2
2	оператор выбора первого типа	3	2
3	оператор выбора второго типа	2	1
4	итерация с предусловием	3	2
5	итерация с постусловием	3	1

Литература

1. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. М.: Финансы и статистика, 1998.176 с.: ил

2. Дейкстра Э. Дисциплина программирования. М.: Мир, 1978. 275с.: ил.
3. Основы верификационного анализа безопасности исполняемого кода программ. / Матвеев В.А., Молотков С.В., Зегжда Д.П., Мешков А.В., Семьянов П.В., Шведов Д.В. Под редакцией проф. Зегжды П.Д. СПб.: СПбГТУ, 1994. 58 с.
4. Лаздин А.В. Построение графа функциональной программы // Диагностика веществ, изделий и устройств: Материалы всероссийской научно-технической конференции. Орел. Изд-во Орл.гос.тех.ун-та, 1999. 189с., ил.
5. Гудман С., Хидетниemi С. Введение в разработку и анализ алгоритмов. М.: Мир. 1981. 366с.:ил.
6. Бейбер Л.Р.. Программное обеспечение без ошибок. М. Радио и связь. 1996. 173 с. ил.

МЕТОД ПОСТРОЕНИЯ ГРАФА ФУНКЦИОНАЛЬНОЙ ПРОГРАММЫ ДЛЯ РЕШЕНИЯ ЗАДАЧ ВЕРИФИКАЦИИ И ТЕСТИРОВАНИЯ

А.В. Лаздин, О.Ф. Немолочнов

Предложен метод формирования графового представления исполняемого кода, ориентированный для работы в системах автоматизированного анализа программного обеспечения.

Считается, что программа как способ представления алгоритма инвариантна формам своего представления. Однако в ходе кодирования возможно внесение непреднамеренных ошибок на каждой стадии жизненного цикла создания программы [1]. Нельзя полностью гарантировать, что в текст программы не будут внесены так называемые “закладки”, направленные на разрушение операционной среды, в которой запущена программа. В соответствии моделями жизненного цикла программ переход от одного уровня абстракции представления исходного алгоритма (или спецификации) к следующему подразумевает такие технологические действия, как тестирование и верификация. Причем в настоящее время верификации уделяется все большее внимание, поскольку этот метод контроля позволяет при меньших временных затратах получать результаты имеющие большую ценность. [2]

Исполняемый код (ИК) и как цель любой программной разработки, и как продукт одного из этапов жизненного цикла программного обеспечения должен проходить те же процедуры контроля (тестирование и верификация), что и алгоритмы, полученные в результате обработки спецификации, и тексты программ, полученные как результат кодирования алгоритмов.

Существует широко распространенное представление программы в виде графа, основанное на понятии "состояние программы". Под состоянием программы понимают содержимое регистров процессора и ячеек памяти (переменных) на некоторый зафиксированный момент. Такое состояние программы ассоциируют с вершиной графа (функциональная вершина), а переходы между ними отображают дугами. Однако такое представление программы отражает скорее состояние аппаратуры, выполняющей ее, и может быть использовано для отладки программы, но не для ее анализа.

В данной работе предлагается метод построения графового представления ИК, ориентированное не столько на отладку программ, сколько на их анализ (обратное проектирование).

Исполняемый код состоит из последовательности машинных команд конкретного процессора. Все команды данного процессора можно разделить на две категории: команды обработки данных (команды пересылки, арифметические команды, команды сдвигов, команды обработки десятичных данных и т.д.) и команды управления последовательностью исполнения (безусловные и условные переходы, команды вызова процедур и команды возврата). С точки зрения выполнения программы, действия процессора по обработке последовательности команд обработки данных – линейный участок – строго детерминированы: команды выполняются одна за другой. Для линейной программы в этом смысле можно выделить всего два состояния: точку запуска и точку останова. Аналогично можно сказать, что любой линейный участок ИК выполняется либо до точки останова программы, либо до команды управления последовательностью команд (КУПК). Точки запуска, останова и КУПК будем называть точками принятия решения.

Определение: функциональная программа – это ИК, решающий некоторую задачу, полученный для конкретного типа процессора и определенной операционной системы. Выполнение функциональной программы (ФП) не зависит от других

программ, т.е. ФП не является драйвером и не включает в себя обработчики прерываний.

При построении программы определяется точка входа – адрес первой выполняемой команды. Выход из ФП (завершение работы) может быть реализован произвольным количеством операторов возврата, вызовов функций `exit()`, `abort()`, использованием механизма исключительных ситуаций. Таким образом, можно говорить о некотором множестве (конечном) точек выхода из программы.

ФП может быть представлена в виде ориентированного графа, вершинами которого будут: точка входа в ФП; множество точек выхода; команды передачи управления, не являющиеся элементами множества точек выхода. С каждой вершиной ассоциируется структура, хранящая адрес этого узла, и адрес перехода из этого узла (если адрес перехода существует).

Дугам этого графа соответствуют линейные участки программы, состоящие из команд обработки данных. Частным случаем линейного участка является пустой оператор. Граф ФП не может содержать петель и параллельных дуг.

Формирование графа ФП начинается с загрузки образа ФП в память и настройки всех абсолютных адресов переходов и вызовов процедур. Выполнение данного этапа соответствует действиям загрузчика ОС по предварительной обработке исполняемого модуля для его последующего запуска. На этом же шаге производится анализ типа исполняемого модуля. Если его формат не соответствует принятому для данной ОС, то обработка прекращается. Эти действия могут быть представлены следующим алгоритмом.

Алгоритм 1.

Шаг 1. Прочитать управляющую информацию из заголовка исполняемого модуля (ИМ). Определить размер ИМ. Определить загрузочное смещение ИМ.

Шаг 2. Выделить память под ИМ. Запись ИМ в память.

Шаг 3. Установить счетчик обработанных перемещаемых элементов в нуль.

Шаг 4. Принять адрес выделенного блока памяти в качестве адреса начального сегмента программы (АНСП).

Шаг 5. Прочитать очередной элемент из таблицы перемещений (ТП).

Шаг 6. Вычислить адрес перемещаемой ссылки как сумму АНСП и значения из ТП.

Шаг 7. Считать из памяти по вычисленному адресу значение относительной адресной ссылки. Прибавить к считанному значению АНСП (выполнение привязки сегмента).

Шаг 8. Записать полученное значение в память.

Шаг 9. Инкрементировать счетчик обработанных ссылок. Если не все ссылки обработаны, перейти к шагу 5.

Результатом работы данного алгоритма будет образ ФП, загруженный в память с настроенными абсолютными адресами. Полная информация о размере и составе кода ФП может быть получена только после выполнения всех действий в ходе построения ГП.

Далее, начиная с точки входа в программу, формируется список вершин графа ФП в соответствии со следующим алгоритмом.

Алгоритм 2.

Шаг 1. Текущий адрес инициализируется адресом начала обработки A_N . (При первом проходе алгоритма A_N инициализируется адресом точки входа.

Шаг 2. Если текущий адрес меньше адреса завершения (A_K), перейти к шагу 3, иначе перейти к шагу 9.

Шаг 3. Определить длину и тип команды.

Шаг 4. Если обрабатываемая команда соответствует точке принятия решения, то определить соответствующий адрес перехода (A_P) и перейти к шагу 5, иначе – к шагу 8.

Шаг 5. Сформировать очередной элемент списка вершин графа.

Шаг 6. Если значение адреса перехода текущей команды меньше A_N перейти к шагу 7, иначе – к шагу 8

Шаг 7. $A_K := A_N$; $A_N := A_P$; Перейти к шагу 1.

Шаг 8. Увеличить значение адреса обработанной команды на ее длину; перейти к шагу 2.

Шаг 9. Выход из алгоритма.

В качестве параметров алгоритм получает адрес начала обработки и конечный адрес просмотра (для первого запуска алгоритма он принимает значение последнего байта программы). Возвращаемое значение (адрес перехода, если $A_P < A_N$, адрес начала обработки в противном случае) заносится в переменную, хранящую A_N .

Данный алгоритм может быть представлен следующим образом:

$$P(A_N, A_K) \equiv \text{if } (A_P < A_N) \text{ then } \Psi[S; P(A_P, A_N)], \quad (1)$$

где P – процедура формирования списка вершин графа; A_N – адрес начала обработки (для первого вызова функции – адрес точки входа.); A_K – конечный адрес просмотра; A_P – адрес перехода; Ψ – композиция базовых операторов S (операторы формирования списка вершин, не содержащие обращения к процедуре P) и процедуры P .

В соответствии с [3] алгоритм, определенный таким образом, является рекурсивным. В [4] показано, что глубина рекурсии для приведенного алгоритма конечна. Применение этого алгоритма обусловлено тем, что в общем случае точка входа в программу не является самым младшим адресом ФП. Однако начинать обработку программы с целью поиска вершин графа возможно только от точки входа, так как по самому младшему адресу (адресу загрузки программы) может находиться, например, сегмент данных, декодирование которого бессмысленно. Следовательно, если в процессе формирования списка вершин графа будет обнаружена команда, адрес перехода которой будет младше адреса точки входа в ФП, то возникнет необходимость обработки команд, начиная от найденного младшего адреса до адреса точки входа.

Следующий этап формирования графа ФП – упорядочивание списка вершин в соответствии с их абсолютными адресами с помощью любого известного алгоритма сортировки. Формирование списка вершин графа ФП закончено. На основании этого списка вершин графа строятся матрицы смежности и(или) инцидентности.

Предложенный в данной работе метод формирования графа ФП позволяет формировать список вершин графа за один проход и хранить только отсортированный список вершин, поскольку вся информация о дугах графа может быть получена на основании анализа адресов переходов, хранимых для каждой вершиной графа.

Граф ФП, построенный в соответствии с предложенным методом, может быть использован для решения задач:

- анализа ИК – например, построение транзитивного замыкания матрицы смежности позволяет находить "висящие" или недостижимые вершины;
- тестирования – для тестирования по методу "белого ящика" важно иметь структуру программы [5], в данном случае структура программы задается графом;
- верификации – в ходе доказательства корректности сегмента программы (в простейшем случае – линейного участка), когда необходимо получить предусловие для заданного постусловия по отношению к данному ИК [6].

Литература

1. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. М.: Финансы и статистика, 1998. 176 с.: ил.

2. Коул Дженнифер, Горэм Томас, МакДональд Марк, Спарджеон Роберт. Принципы тестирования ПО // Открытые системы (Изд-во “Открытые системы”). 1998. № 2.
3. Вирт Н. Алгоритмы + структуры данных = программы: Пер. с англ. М.: Мир, 1985. 406с.: ил.
4. Логунова О.А., Лаздин А.В., Пухтеева Е.А. Рекурсивный алгоритм формирования списка вершин графа переходов функциональной программы. // Труды молодых ученых и специалистов. Сборник научных статей. Выпуск 1. Часть 2. СПб:СПбГИТМО(ТУ), 2000. 112с.
5. Майерс Г. Искусство тестирования программ. М.: Мир, 1982. 212 с
6. Бейбер Л.Р.. Программное обеспечение без ошибок. М.: Радио и связь. 1996. 173с. ил.

РАСЧЕТ РАЗМЕЩЕНИЯ СХЕМ В ЛОГИЧЕСКИХ МАТРИЦАХ

Д.А. Гончаров, В.И. Скорубский

Проектирование схем в программируемых логических матрицах (ПЛИС) – трудоемкая и ответственная задача, оптимальное решение которой существенно определяет качество бортовых и встроенных цифровых систем. Для решения задачи используется фирменная технология, принципы реализации которой неизвестны, и разработчик вынужден интуитивно к ней приспосабливаться. Предлагается модель реализации логики в ПЛИС, которая позволяет выполнить как ручные, так и машинные предварительные расчеты для выбора оптимальной реализации и снижения риска отказа при размещении схем в БИС.

Прогресс в технологии изготовления дискретных схем стал возможен с развитием и совершенствованием технологии проектирования. Современные схемы программируемых логических матриц (ПЛИС или CPLD) содержат до 1.000.000 вентиляей. При таком уровне сложности не ощущаются реальные ограничения, накладываемые структурой матриц при размещении схем большой сложности.

Системы проектирования, предлагаемые фирмами, предоставляют разработчику различные формы описания схем, основанные на языках проектирования VHDL и AHDL. Технология позволяет проектировать схему в графическом виде и редактировать размещение схемы по логическим ячейкам.

Описание схемы компилятор преобразует в файлы, которые могут быть использованы при симуляции или загрузке в ПЛИС. Несмотря на то, что пакет MAX+PLUS имеет, по-видимому, достаточно эффективную алгоритмическую базу, результат автоматического проектирования зависит от некоторой фиксированной стратегии и алгоритма распределения схемы по логическим ячейкам и не предсказуем. В случае неудачного или неудовлетворительного исхода размещения приходится либо вмешиваться в процесс распределения и предсказывать, как будет развиваться процесс при изменении условий, либо самому детально выполнять распределение. В связи с этим полезно иметь доступные стратегии и алгоритмы расчета и модели размещения, которые могут быть использованы для автоматизации и предварительной оценки ресурсов БИС.

На рис. 1 приведена типовая структура логической ячейки ПЛИС.

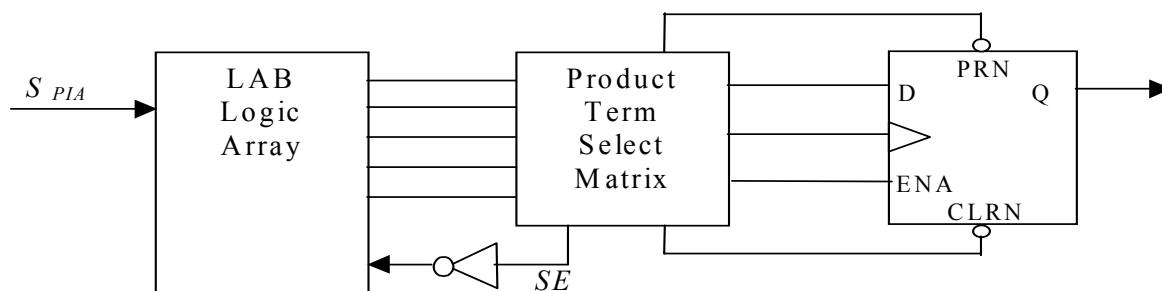


Рис. 1. Структура логической ячейки

Каждая ячейка индивидуально конфигурируется для выполнения определенной логической операции и состоит из четырех элементов.

1. Логический массив (Logic Array) формирует термы для каждой макроячейки и включает 36 сигналов программируемого массива межсоединений и 16 общедоступных расширителей.
2. Матрица распределения термов (Product-Term Select Matrix) формирует функцию OR или XOR из термов, которые поступают из логического массива. Функция может состоять из 5 основных термов, количество которых можно увеличить за счет использования параллельных расширителей. Если имеются свободные термы, то

они могут использоваться для получения дополнительных функций, которые поступают в логический массив через общедоступные расширители.

3. Программируемый регистр (Programmable Register) используется для хранения значения. Может быть любого типа (D, T, JK, RS). Имеется возможность управления синхросигналом (CLK), разрешением (ENA), сбросом (CLR N) и установкой (PR N). Значения битов данных, сохраняемых в регистрах, могут поступать как из матрицы распределения, так и непосредственно из блока контроля выводами. Вместе с тем регистр может быть не использован в том случае, когда для расширения выбирается обходной путь.
4. Формирователи сигналов управления программируемым регистром - CLK, ENA, CLR N и PR N. Сигналы могут быть сформированы термами матрицы расширения для управления локальным регистром в ячейке или являются специальными глобальными сигналами для всей схемы.

Более подробно логическая ячейка представлена на рис. 2.

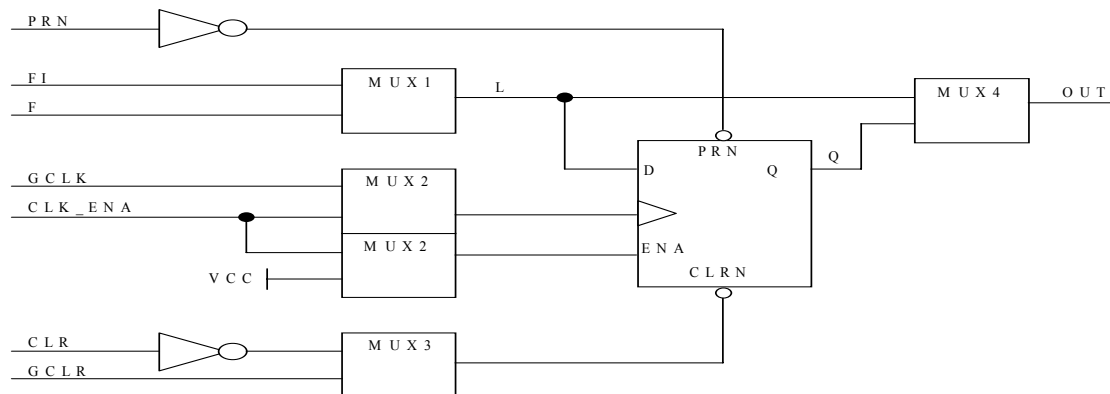


Рис. 2. Подробная структура логической ячейки

Функция выхода элемента может быть представлена общей формулой:

$$OUT = [(F +_1 FI) \& (GCLK \& CLK_ENA +_2 CLK_ENA) \vee PRN] \& \overline{(CLR +_3 GCLR)} +_4 (F +_1 FI).$$

В формуле приняты следующие обозначения.

PRN – Асинхронная установка. Сигнал выходит из матрицы распределения термов. (Product-Term Select Matrix)

FI – Быстрый вход от ножки данной ячейки.

F – Функция данных, формируемая матрицей распределения термов.

GCLK – Глобальный синхросигнал.

CLK_ENA – Функция сигнала синхронизации/разрешения, формируемая матрицей распределения термов.

CLR – Асинхронный сброс. Сигнал выходит из матрицы распределения термов.

GCLR – Глобальный сигнал сброса.

OUT – Выход логической ячейки.

+₁ - Операция выбора на мультиплексоре MUX1 между функцией данных F и быстрым входом FI.

+₂ - Операция выбора на сдвоенном мультиплексоре MUX2. В первой половине мультиплексора происходит выбор между функцией синхронизации/разрешения CLK_ENA и сигналом глобальной синхронизации GCLK, а во второй половине – выбор между функцией синхронизации/разрешения CLK_ENA и единицей VCC.

+₃ - Операция выбора на мультиплексоре MUX3 между функцией асинхронного сброса CLR и сигналом глобального сброса GCLR.

+₄ - Операция выбора на мультиплексоре MUX4 между триггерной и комбинационной ячейками.

Операции выбора конструктивны и не являются алгебраическими, что затрудняет оценку и сравнение реальных функций, размещаемых в ячейке. Для конструктивного описания можно использовать язык регулярных выражений, для распознавания которых существуют эффективные автоматные модели алгоритмов [1].

Логическую ячейку можно представить регулярным выражением

$$OUT = [(f + q) \& (g \& c + c) \vee s] \& (r + e) + (f + q)$$

в языке L с алфавитом

f = F;

q = FI;

g = GCLK;

c = CLK_ENA;

s = PRN;

r = CLR;

e = GCLR.

Пример допустимых в языке L схем:

$$OUT = (f \& c)$$

$$OUT = (f \& c \vee s) \& r$$

$$OUT = (f \& g \& c \vee s) \& e$$

Далее в тексте для обозначения логической операции "или" будет использоваться знак " $\sum$ ", для обозначения логической операции "и" знак " Π ".

Функции управления ячейками могут быть записаны в виде системы уравнений:

$$P^k = T_1 \& \overline{\prod_{j \in Sq} T_j},$$

$$R^k = T_2 \& \overline{\prod_{j \in Sq} T_j},$$

$$C^k = T_3 \& \overline{\prod_{j \in Sq} T_j},$$

$$F^k = \sum_{i \in S_k} T_i \& \overline{\prod_{j \in Sq} T_j} + T_5 \& \overline{\prod_{j \in Sq \setminus T_5^k} T_j} + F^{k-1}.$$

Здесь использованы следующие обозначения.

P – функция асинхронной установки.

R – функция асинхронного сброса.

C – функция синхронизации/разрешения.

F – функция данных.

$k \geq 1$

$S_k = \{T_1^k, T_2^k, T_3^k, T_4^k\}$ – множество конъюнктивных термов.

$S_q = \{T_5^1, T_5^2, \dots, T_5^N\}$ – множество конъюнктивных термов расширения.

N – число логических ячеек в блоке. $N=16$.

T – конъюнктивный терм входных сигналов из массива межсоединений.

$T = \tilde{x}_1 \& \tilde{x}_2 \& \dots \& \tilde{x}_M$

Символ " $\sim$ " обозначает прямое или инверсное значение переменной.

M – число сигналов, поступающих в блок из массива межсоединений, $M \leq 36$.

$\overline{\prod_{j \in Sq} T_j}$ обозначает последовательное применение расширителей с инверсией.

Так, при $\overline{T_1} \& T_2$ используется один расширитель, при $\overline{\overline{T_1} \& T_2} \& T_3$ – два

расширителя, при $\overline{\overline{T_1 \& T_2 \& T_3 \& T_4}}$ - три расширителя и т.д. Расширители позволяют сократить число термов, используемых для реализации функций. Например,

$$\overline{\overline{T_1 \& T_2}} \rightarrow (\overline{x_1 \vee x_2 \vee \dots \vee x_n}) \& T_2 = D \& T,$$

где D - дизъюнктивный терм. Если не использовать расширители, то для реализации соответствующей функции необходимо n термов вместо одного. Количество термов расширения в ПЛИС не более 16.

Можно привести все выражения подобного вида к формуле, содержащей только символы T и D .

Система выражений, заменяющих формулы расширителей с инверсиями, имеет вид

$$\begin{aligned} \overline{\overline{T_1 \& T_2}} &\rightarrow D_1 \& T_2 \\ \overline{\overline{T_1 \& T_2 \& T_3}} &= (\overline{T_1 \vee \overline{T_2}}) \& T_3 = (T_1 \vee D_2) \& T_3 \\ \overline{\overline{\overline{T_1 \& T_2 \& T_3} \& T_4}} &= (\overline{\overline{T_1 \vee \overline{T_2}}}) \& T_3 \& T_4 = (\overline{T_1 \& T_2 \vee \overline{T_3}}) \& T_4 = \\ &= (D_1 \& T_2 \vee D_3) \& T_4 \\ \overline{\overline{\overline{\overline{T_1 \& T_2 \& T_3 \& T_4} \& T_5}}} &= (\overline{\overline{\overline{T_1 \& T_2 \vee \overline{T_3}}}}) \& T_4 \& T_5 = \\ &= ((\overline{T_1 \vee \overline{T_2}}) \& T_3 \vee \overline{T_4}) \& T_5 = ((T_1 \vee D_2) \& T_3 \vee D_4) \& T_5 \end{aligned}$$

Таким образом, расширители позволяют выполнить в БИС сложные скобочные формулы с расширением как по И, так и по ИЛИ.

В общем виде систему уравнений функций управления можно представить следующими выражениями:

$$\begin{aligned} \theta &= T \& \omega, \\ F^k &= \sum T \& \omega + T \& \omega + F^{k-1}. \end{aligned}$$

Здесь под знаком $\sum$ выбираются $n-1$ термов T из общего их числа n , $+$ обозначает объединение в регулярном выражении, F – рекуррентная функция, причем уровень рекурсии зависит от уровня расширителя.

Один из этих термов используется в расширителе.

$$\theta = \{P, R, C\},$$

$$\omega = [((T + D) \vee D) + (((D \& T) + ((T \vee D) \& T)) \vee D)] \& T,$$

$+$ обозначает объединение предложений в регулярном выражении

Любая формула может быть приведена к регулярной скобочной форме типа ω , θ , F . Возможность реализации всякой формулы проверяется распознавателем выражений в языке L , который определяется регулярными выражениями в алфавите $\{T, D, \&, \vee\}$.

Распознаватель слов языка регулярных выражений может быть построен в виде конечного автомата [1], для которого существует эффективная программа, что позволяет автоматизировать процесс распознавания.

В процессе распознавания фиксируются состояния автомата, и соответствующие ячейки загружаются термами. После того как очередная формула выполнена, корректируется система регулярных выражений и соответствующие автоматы - распознаватели. Таким образом, последовательно заполняются ресурсы БИС. Конечный результат реализации системы функций не зависит от их последовательности и ограничивается только ресурсами CPLD: максимальным количеством расширителей, суммарным количеством переменных в термах, количеством запоминающих ячеек, шириной коммутационных каналов. Часть этих ограничений может быть учтена на этапе распознавания, другая – на этапе преобразования проекта в систему функций.

Заключение:

В рамках статьи рассмотрена задача размещения (реализации) в ПЛИС логических формул, полученных в процессе преобразования цифровых схем проекта. Формулы могут быть приведены с использованием языка регулярных выражений к предложениям, которые согласуются с конструктивным описанием ПЛИС. Таким образом, формулируются алгоритмы распознавания формул, допустимых данной ПЛИС. С использованием известного аппарата [1, 2], применяемого при проектировании БИС, строятся распознаватели, которые решают задачу размещения схем.

Литература

1. Ульман Дж. Д. Вычислительные аспекты СБИС. М.: Радио и связь, 1990. С. 298-306.
2. Ульман Дж. Д. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.
3. MAX 7000. Programmable Logic Device Family. ALTERA.: 2001. С. 1-18.

ОПТИМИЗАЦИЯ ПРОЕКТНЫХ РЕШЕНИЙ В САПР АВТОМАТИЗИРОВАННЫХ ТЕХНОЛОГИЧЕСКИХ КОМПЛЕКСОВ

Ю.А. Гатчин, И.Б. Бондаренко

Современные САПР обладают значительными возможностями и многофункциональностью, которые требуют все больших вычислительных затрат и объемов оперативной памяти. Из-за сложности технологических процессов (ТП), применяемых в химической, оптической, нефтегазовой и др. промышленности, которые представляют собой управление множеством объектов, возникает большое число задач, которые невозможно решить с помощью математического аппарата дифференциальных уравнений. Возрастает и поток информации между компонентами систем, что приводит к необходимости существенного изменения подходов к решению задачи его обработки для повышения эффективности научных исследований и использования их результатов для максимальной автоматизации научных и производственных процессов.

Для осуществления процесса проектирования САПР ТП должны использовать сложные формы знаний и адекватный аппарат их обработки и использования. Этот подход основан на создании САПР не последовательным наращиванием возможностей путем подключения дополнительных программных модулей и созданием требуемых баз данных (БД), а радикальным перераспределением вычислительных работ и концентрацией исследовательских, поисковых задач проектирования в экспертной системе, которая может рассматриваться как особая комплексная подсистема со своей информационной базой и программным обеспечением общего и специального назначения.

Таким образом, современные САПР ТП являются сложными системами со следующими особенностями:

- наличие большого числа подсистем;
- многомерность систем;
- многокритериальность;
- многообразие структуры;
- многообразие природы подсистем.

Несмотря на то, что технический прогресс идет по пути создания все более сложных алгоритмов и систем проектирования, полностью исключить роль человека на определенных этапах не представляется возможным, поэтому для достижения наилучших результатов создаются интерактивные системы проектирования, где человеку предоставляется автоматизированное рабочее место (АРМ). Число АРМ будет определяться уровнем иерархичности определенной ТС и ее сложностью.

Создание АРМ на основе автоматизированных технологических комплексов (АТК) применительно к оптической промышленности представляет особый интерес вследствие малой применяемости и освоенности.

Предметом исследования являются алгоритмы, использующиеся в современных пакетах прикладных программ (ППП) для САПР АТК. При автоматизированном проектировании современных ТП рано или поздно возникает задача оценки критерия качества при неопределенных значениях вектора параметров X ТП. При этом область допустимых значений Ω параметров X_i должна быть такой, чтобы: гарантировалась устойчивость системы; внутри области Ω_i должна гарантироваться определенная величина установившихся ошибок, вызванных предельно допустимыми возмущениями; необходимо учитывать особенности технико-экономического характера, которые не позволяют неограниченно изменять некоторые параметры X_i .

При таких условиях оценка качества однозначно определяется выбранным значением вектора X .

Введем в рассмотрение вектор локальных критериев $\Phi = (\Phi_1, \Phi_2, \dots, \Phi_3)$. Известно, что между глобальным критерием Φ и совокупностью локальных критериев Φ_i существует непосредственная или опосредованная связь, следовательно, оптимизацию множества Φ можно осуществлять с помощью последовательной оптимизации локальных критериев Φ_i на соответствующих этапах проектирования. Особенностью функции качества Φ_i :

$$\Phi = \text{extr} (Q_i(X))$$

для сложных систем оптимизации проектных решений является многоэкстремальность, а из-за неаналитического характера функции $Q(X)$ не удастся получить выражения для частных производных $\partial Q / \partial X_i$ в виде формул, что в свою очередь затрудняет поиск глобального минимума функции. Затраты на поиски глобального минимума функции $Q(X)$ зависят от размеров области Ω в пространстве параметров $X \in \Omega$, в которой производится поиск, а также от требуемой точности определения X^* , доставляющего минимум функции $Q(X)$. Для уменьшения расходов на поиск минимума функции необходимо сузить область Ω и выбрать разумную точность определения X^* .

Сложные ТП для АКМ предполагают многоуровневую процедуру их проектирования, следовательно, проектирование таких ТП включает в себя формирование множества возможных способов ее реализации и последующий выбор оптимального варианта. При переходе на следующий уровень структуры принятия решений каждый из ранее синтезированных вариантов порождает новую совокупность возможных решений. Поэтому по мере проектирования число возможных способов реализации существенно возрастает.

С другой стороны, достоверная оценка оптимальности системы оказывается возможной лишь после окончания ее проектирования, что вызвано либо большим числом факторов, влияние которых нельзя заранее предусмотреть из-за их случайного характера, либо отсутствием достаточной информации о них на отдельном этапе проектирования. В связи с этим возникает необходимость полной проработки значительного числа возможных вариантов, в том числе и тех, которые в дальнейшем окажутся неприемлемыми.

Чтобы уменьшить трудоемкость процедуры проектирования на этапе принятия оптимальных решений, необходимо процесс генерации вариантов сочетать с процессом исключения неприемлемых решений, выявленных в ходе проектирования. Однако при этом появляется опасность выбора неоптимального варианта, как результата оценки на промежуточных этапах проектирования с помощью только субъективных показателей, множество которых отражает предпочтение проектировщика относительно полученного варианта проектного решения.

Достоверность субъективного отбора вариантов зависит от объема объективной информации о свойствах проектируемого ТП. Следовательно, для снижения риска выбора неоптимального решения, необходимо осуществить анализ свойств проектируемого ТП. Причем объем проводимых исследований должен быть достаточным для снижения неопределенности в оценке свойств системы до уровня, позволяющего осуществить обоснованный выбор. Для этого необходимо предусмотреть возможность изменения априорных оценок эффективности системы и отказа от ранее выбранного решения. При такой организации проектирования объем объективной информации о системе будет возрастать на каждом последующем этапе, а оценка свойств системы будет все более достоверной.

Таким образом, процесс проектирования сложных ТП представляется как последовательность уровней, на каждом из которых проводится синтез вариантов

построения ТП, предназначенных для решения заданной совокупности задач управления, и проводится ряд экспериментов с целью дополнительного исследования свойств ТП. С учетом уточненной информации о свойствах ТП отбираются наилучшие варианты. На последующих уровнях проектирования эти варианты прорабатываются с учетом более глубокого анализа, и снова производится отсев неудовлетворительных вариантов.

Для оценки свойств вариантов построения системы на i -м уровне проектирования АТК воспользуемся вектором локальных критериев Φ . Локальные критерии оптимальности строятся на основе глобального критерия и оценивают способность разработанных вариантов ТП выполнять поставленные задания в свойственных данному уровню детализации системы показателях. Таким образом, имеется возможность отсеивать неудовлетворительные варианты построения системы в процессе проектирования, а не после его окончания.

Для предпочтения одного подмножества вариантов другому применим метод последовательного анализа и отсева вариантов, который адекватен специфике задач, возникающих на ранних этапах проектирования сложных ТП. Развитие схемы последовательного анализа вариантов в многошаговые дискретные управляемые процессы позволяет управлять процессом проектирования ТП.

Это достигается следующим образом. Процедура проектирования ТП строится таким образом, чтобы объем сведений об объекте на каждом последующем этапе возрастал, при этом множество допустимых вариантов Ω будет включать такие варианты, которые заранее не будут удовлетворять либо ограничениям на ТП, либо оценочной функции. Тогда задача проектирования сведется к задаче отыскания некоторого ядра

$$X \rightarrow X^* = \max(X, \Phi)$$

модели (X, Φ) , где X – множество конкурирующих вариантов проекта, которое предполагается конечным, а Φ – бинарное отношение сравнительной эффективности на множестве X . Необходимо отметить, что сложные ТП характеризуются как большой мощностью множества X , так и сложностью сравнения конкурирующих вариантов по отношению Φ , при котором приходится решать задачи оптимизации, просчитывать вариант на имитационных моделях системы, что требует ощутимых затрат машинного времени.

Сложностью в применении схем последовательного анализа вариантов является корректная разработка правил отбрасывания неприемлемых вариантов и реализация процедур сравнения эффективности достаточно быстрыми алгоритмами, так как для сложных ТП количество обрабатываемых вариантов может быть очень большим.

Параллельно с методом последовательного анализа возникает задача выбора эффективных или неухудшаемых вариантов параметров ТП (альтернатив) из множества допустимых альтернатив A по множеству целевых функций. Существование эффективных альтернатив связано с наличием оптимума по каждой целевой функции. Для отыскания оптимальных значений функций удобно использовать математический аппарат поисковых методов (см. рис. 1). Сложным технологическим системам присуще свойство многопараметричности и многокритериальности, следовательно, для осуществления поиска оптимума воспользуемся IV-й группой методов. На рисунке методы расположены по возрастанию сложности их программной реализации и вспомогательных параметров работы, но, с другой стороны, при этом возрастает и размерность задач, которые можно решать этими методами.



Рис. 1. Математический аппарат оптимального проектирования

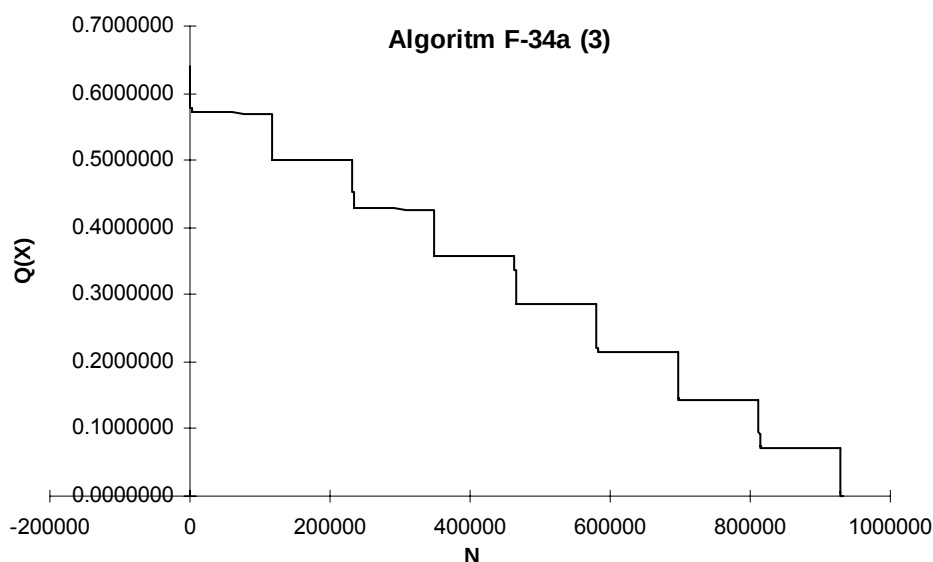


Рис. 2. Результат работы метода многошаговой редукции размерности на равномерной сетке

Метод многошаговой редукции размерности основан на свертке последовательно вложенных друг в друга одномерных задач глобальной минимизации из группы II. В

качестве последних использованы методы перебора на равномерной (рис. 2) и неравномерной сетке узлов (рис. 3). В качестве параметров метода многошаговой редукции использованы отрезок поиска, точность (и параметр изменения шага для каждой переменной для неравномерной сетки). Требуемая точность 0,1% достигается для трехмерного случая функции за тысячи – десятки тысяч итераций алгоритма (за сотни итераций для неравномерной сетки узлов). Несмотря на простоту метода и небольшое число входных данных, с ростом размерности экспоненциально растут общие затраты на поиск экстремума. Поэтому при большом числе переменных метод теряет эффективность.

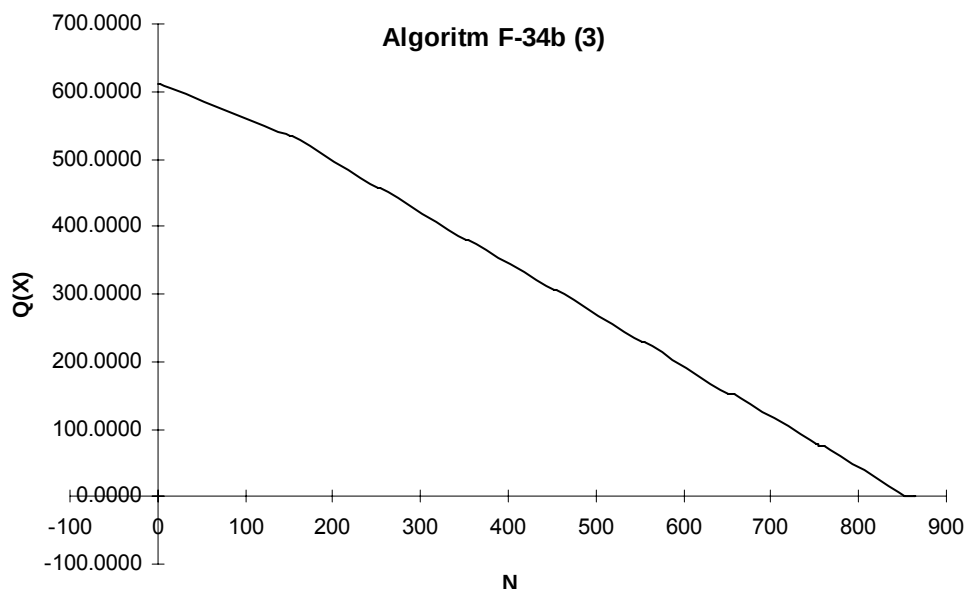


Рис. 3. Результат работы метода многошаговой редукции размерности на неравномерной сетке

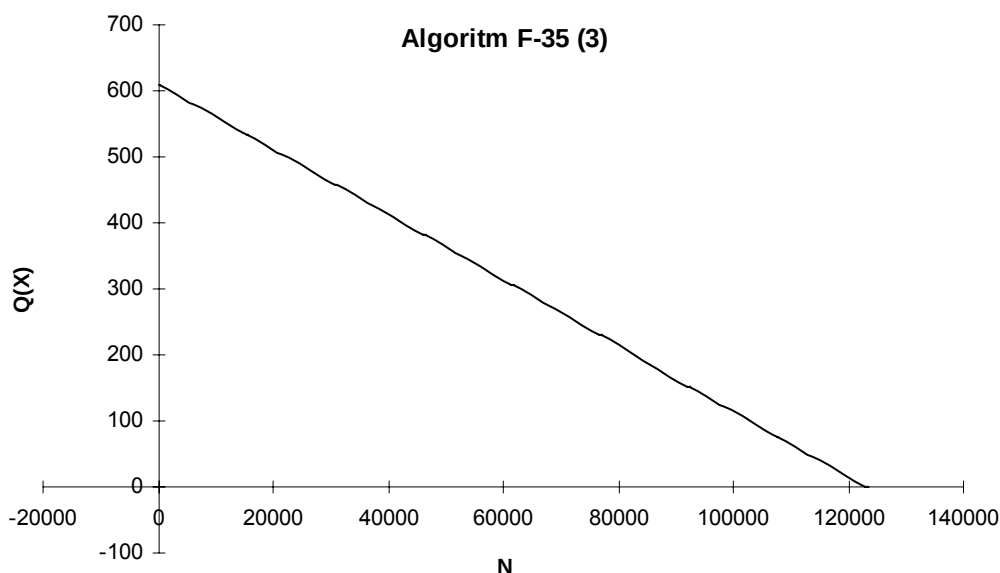


Рис. 4. Результат работы метода глобального поиска по кривой Пеано

Метод глобального поиска по кривой Пеано основан на отображении единичного отрезка вещественной оси в гиперпараллелепипед. Построение отображения

осуществляется с условием непрерывности и однозначности, в результате чего для любой точки единичного отрезка получаем точку одномерной функции. Оптимизация полученной одномерной функции велась с помощью метода редукции размерности. При работе метода наблюдается повышение сложности функции отображения за счет роста числа прообразов точек исходной функции в интервале единичного отрезка. Поэтому метод усложняется с ростом размерности исходной задачи и при усложнении исходной функции. Достоинством метода построения кривой Пеано (рис. 4) является повышение размерности до 7 переменных при сохранении высокой скорости работы метода за счет простоты вычислительных процедур. Для эффективной работы программы требуется вмешательство оператора.

Метод оптимального покрытия прямоугольного параллелепипеда равными кубами подбирает длину ребра оптимальной для покрытия одного из ребер параллелепипеда (рис. 5). При ребрах единичной длины задача оптимального покрытия имеет простое аналитическое решение. Составленная программа формирует массив из альтернативных длин ребер кубов, после чего выбирается наименьшее из этих значений. Найденное значение определит шаг сетки перебора для поиска экстремума на заданном промежутке в гиперпараллелепипеде.

Данный алгоритм работает с достаточной скоростью, но очень восприимчив к входным параметрам, таким как отрезок поиска и параметры, определяющие начало поиска.

Algorithm F-37 (3)

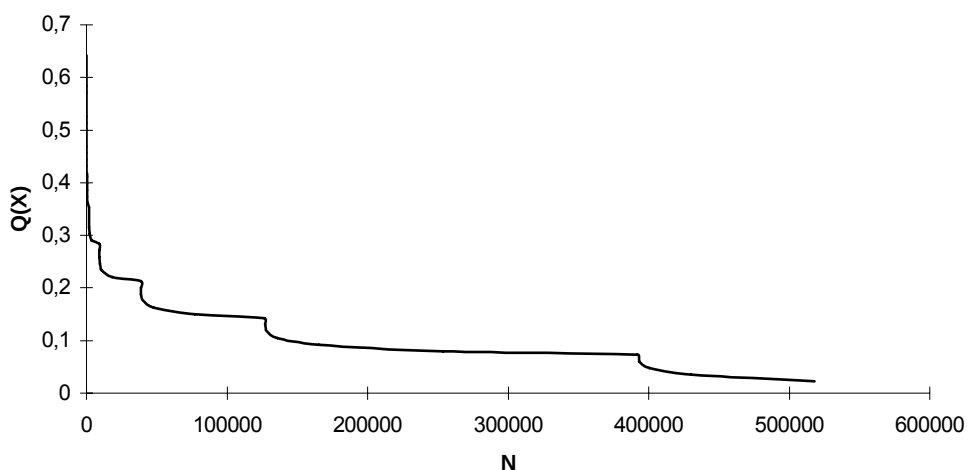


Рис. 5. Результат работы метода оптимального покрытия прямоугольного параллелепипеда равными кубами

Метод глобального поиска с помощью стохастического автомата работает на неравномерной сетке перебора по каждому из параметров. Каждый отрезок сетки характеризуется центральной точкой, а полученное разбиение параллелепипеда ставит в соответствие каждой подобласти состояние стохастического автомата. Очередное испытание регламентируется вероятностью выбора подобласти с помощью датчика случайных чисел. Структура стохастического автомата меняется после каждого испытания. Для выбора оптимума служит параметр максимальной вероятности, с которым сравниваются значения каждого куба разбиения (рис. 6).

Для работы алгоритма большое значение имеют начальные значения функций вероятности выбора состояний автомата, поэтому первая итерация проходит по своему алгоритму. Такое усложнение программы ускоряет поиск и увеличивает вероятность

нахождения экстремума функции. Большую роль играют также вспомогательные параметры программы, от которых зависит нахождение с заданной точностью или пропуск экстремума, а также глубина перебора значений в узлах сетки. Все зависимости параметров друг от друга носят сложный характер, поэтому программа нуждается в диспетчеризации. При определенном образом подобранных параметрах алгоритм позволяет отыскать оптимум с требуемой точностью быстрее, чем перебор на равномерной сетке за счет введения случайности при поиске. Проведенное моделирование показало, что отыскание экстремума производится методом с определенной вероятностью, которая может зависеть и от организации процесса генерирования случайных значений. Метод позволяет значительно расширить круг успешно обрабатываемых функций.

Algoritm F-38 (3x4)

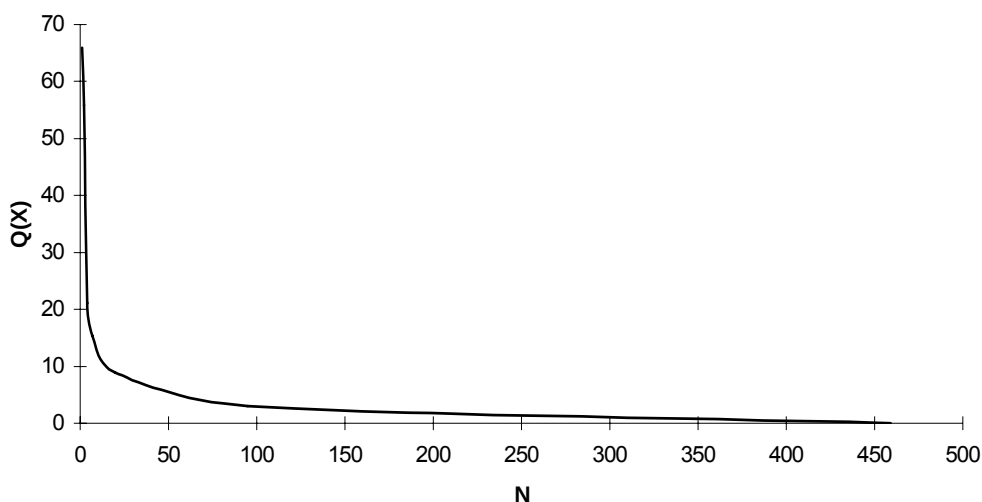


Рис. 6. Результат работы метода глобального поиска с помощью стохастического автомата

В методе глобального поиска с помощью коллектива независимых стохастических автоматов исследуемая функция находится в среде, реагирующей на состояние отдельных автоматов (рис. 7). Входы автоматов независимы друг от друга и отличаются по реакции среды, а действия автоматов определяются их внутренними состояниями. Для каждого автомата введена глубина памяти автомата. Количество независимых автоматов не может быть меньше количества переменных в функции. Запуск такой системы приводит к организации направленного движения к точке экстремума. При продолжении процедуры происходит блуждание в окрестности точки экстремума. Необходимым условием осуществления эффективного поиска является условие разнообразия автоматов по какому-либо их параметру (глубине памяти) или введением случайности в их работу. Эффективность работы метода повышается дублированием автоматов и рассинхронизацией смены внутренних состояний, что связано с большей маневренностью всей системы по разным направлениям поиска. К тому же дублирование повышает надежность работы всей поисковой системы. При числе независимых стохастических автоматов, превышающем размерность задачи в несколько раз, можно регламентировать со случайной вероятностью работу каждого автомата, например, освободить часть из них на некоторое время.

Algorithm F-40 (3)

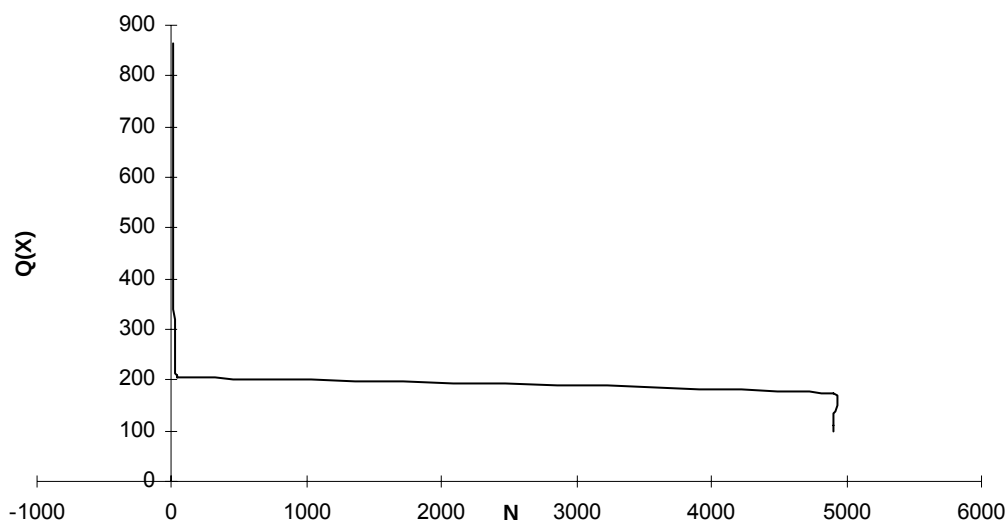


Рис. 7. Результат работы метода глобального поиска с помощью коллектива независимых стохастических автоматов

Такие богатые возможности по варьированию входных параметров делают программу достаточно сложной для алгоритмизации и исследования, а следовательно, для применения. Исследования показывают, что наибольшее число шагов поиска присуще коллективу синхронизированных стохастических автоматов, а для асинхронно работающих стохастических автоматов – наименьшее. Глубина памяти автомата по зависимости от числа шагов алгоритма носит параболический характер и требует подбора ее величины.

Приведенные алгоритмы реализованы в виде пакета программ в едином стиле и позволяют создать мощный интерактивный инструмент для помощи проектировщику при решении широкого круга задач САПР АТК. Реализация методов позволяет находить решение за конечное время поиска с определенной точностью, зависящей от скорости работы вычислительной техники. Размерность задач ограничивается лишь выбором метода и располагаемым временем на расчеты. Для исследования работы методов оптимального проектирования выбрана функция вида

$$Q(X) = \sum_{i=1}^9 (x_i - A \sin \omega t)^2.$$

Графики зависимости достигнутых значений минимизируемой функции от шага приведены на соответствующих рисунках. Такая функция является многомерной и достаточно сложной для нахождения глобального минимума. Приведенные графики отражают свойства методов оптимального проектирования, описанные выше.

На основании полученных результатов разработанные методы позволяют принимать оптимальные проектные решения, исходя из задач и целей проектирования ТП.

Так как принятие решения – завершающая стадия выбора, в которой на основании оценок для последствий проведения каждой операции определяется относительное ее предпочтение по сравнению с ранее предложенными операциями, то оптимизация принятия проектных решений с помощью описанных методов должна ускорить процесс проектирования ТП, особенно при их многомерности.

ХАРАКТЕРИСТИКИ ФУНКЦИОНИРОВАНИЯ ГИБРИДНОЙ САПР

М.Ю. Юдин

В настоящее время проблемы увеличения степени автоматизации процесса проектирования решаются при помощи технологий искусственного интеллекта (ТИИ). Такой подход уже позволил создать принципиально новый класс программного обеспечения – гибридные САПР (ГСАПР) [2].

В основе применения ТИИ в проектировании лежит способность программ, созданных с применением ТИИ, в частности экспертных систем (ЭС), сохранять знания (в нашем случае знания проектировщика), осуществлять с ними работу и принимать решения. В результате пользователи (проектировщики) САПР, имеющей в своей структуре ЭС, имеют информационную или справочную поддержку, основанную на знаниях и опыте об объекте проектирования более квалифицированных специалистов. Такое взаимодействие алгоритмов проектирования с ЭС позволяет автоматизировать те процедуры процесса проектирования, которые решаются на основе знаний проектировщика предметной области (объекта проектирования) и не могут быть представлены в виде алгоритмов. Структурная схема ГСАПР представлена на рис. 1.

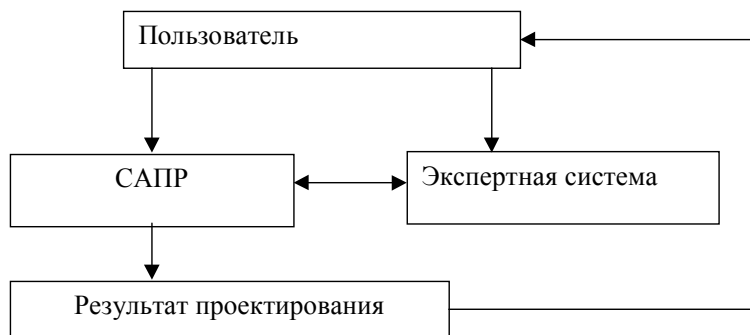


Рис. 1. Структурная схема гибридной САПР (ГСАПР)

По рис. 1 можно сделать вывод, что ГСАПР – программный комплекс, состоящий из множества независимо созданных друг от друга программных блоков, а подчас и работающих независимо друг от друга. В связи с этим определение характеристик функционирования такой сложной системы, включающей в себя как модули, работающие на основе алгоритмов, так и программы, функционирующие на основе логических рассуждений и знаний о предметной области, является необходимым.

Как видно из рис. 1, эффективность работы ГСАПР зависит от взаимодействия пользователя с алгоритмами САПР и ЭС (множества ЭС) при помощи процедур пользовательского интерфейса (на рис. 1 интерфейс показан в виде стрелок). С учётом того, что ЭС является полностью самостоятельным программным модулем и состоит, согласно рис. 2, из базы знаний (БЗ), устройства логического вывода и интерфейса пользователя, то при более детальном анализе эффективности работы ГСАПР и определения характеристик её функционирования необходимо рассмотреть взаимодействие этих 3-х модулей с алгоритмами САПР.

Устройство логического вывода и пользовательский интерфейс являются стандартными компонентами ЭС и их структура, а также их способы реализации практически не оказывают влияния на качество работы ЭС и эффективность её взаимодействия с алгоритмами. В связи с этим дальше эти программные модули рассматриваться не будут. Таким образом, взаимодействие только двух блоков программного обеспечения ГСАПР, а именно БЗ и алгоритмов САПР может определить эффективность работы ГСАПР.

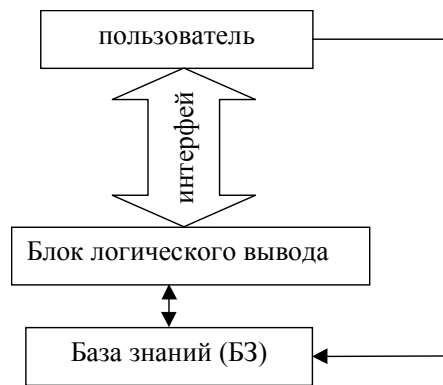


Рис. 2. Структурная схема экспертной системы (ЭС)

Если рассматривать алгоритмы ГСАПР или САПР и эффективность их работы, то параметры и характеристики для качественной оценки программного обеспечения на основе алгоритмов (алгоритмических знаний) довольно хорошо известны и широко применяются для анализа готовых программ [5]. С учётом того, что вывод характеристик эффективности функционирования ГСАПР определяется взаимодействием двух программных модулей (САПР и БЗ), то работу САПР или алгоритмических знаний ГСАПР лучше отображать через характеристики, описывающие САПР как элемент сложной системы. К таким характеристикам можно отнести точность вычислений, время ответа системы и надёжность. Используя эти три характеристики, можно достаточно точно описать ту или иную САПР. Действительно, любая САПР – это математический пакет, работающий с пользователем в диалоговом режиме. Поэтому использование в качестве характеристик, описывающих работу системы, времени ответа и точности вычислений не вызывает сомнений. Применение в качестве характеристики надёжности также необходимо, так как САПР является технической системой. В связи с этим можно сделать вывод, что необходимыми и достаточными характеристиками, описывающими работу САПР, в нашем случае являются точность вычислений, время ответа системы и надёжность.

Теперь рассмотрим БЗ. Любая БЗ представляет собой структуру, состоящую из данных любого типа (базы фактов) и правил их использования (базы правил) [4, 7]. Поэтому при определении характеристик необходимо в первую очередь рассматривать БЗ как структуру. С другой стороны база знаний является неотъемлемой частью ЭС. Совершенно понятно, что область применения ЭС будет оказывать влияние на структуру БЗ и её содержимое. Однако с учётом основных правил и теорем логики, способов логических рассуждений на основе фактов можно сделать вывод, что построение практически любой БЗ можно свести к строгой последовательности действий, т.е. к алгоритму [8]. В связи с этим появляется возможность описать структуру БЗ через те или иные характеристики алгоритма проектирования базы знаний и не учитывать тот факт, что БЗ и их ЭС могут иметь совершенно отличные друг от друга сферы применения.

Основной структурной характеристикой БЗ является глубина вывода, т.е. число уровней правил и фактов, которые образуют так называемое дерево решений (рис. 3). На гипотетическом дереве решений, представленном на рис. 3, изображено 3 уровня выводов или фактов. 1-ый уровень образуют факты или, другими словами, запросы ЭС к пользователю в ходе получения логических выводов. 2-ой уровень – правила образованные фактами 1-го уровня при помощи логических операций [1, 3, 6]. 3-ий уровень – это выводы, сделанные ЭС на основании правил 2-го уровня. Можно предположить, что увеличение уровней дерева решения приведет лишь к детализации выводов 3-го уровня. Например, если взять пример из медицинской сферы, то 3-х уровневое дерево решений будет соответствовать логической цепочке: симптом (1-ый уровень) – болезнь (2-ой уровень) – способ лечения (3-ий уровень). Понятно, что при

использовании 4-го и более уровней дерева решений ЭС будет более детально описывать болезнь, последствия, побочные эффекты лекарств и т.д. В связи с этим можно сделать вывод, что три уровня выводов БЗ (рис. 3) – минимальное число уровней для построения работающей ЭС. Таким образом, число уровней (глубина вывода) БЗ является не только характеристикой, описывающей эффективность работы ГСАПР, но и величиной, определяющей уровень применения ТИИ в том или ином программном продукте класса ЭС.

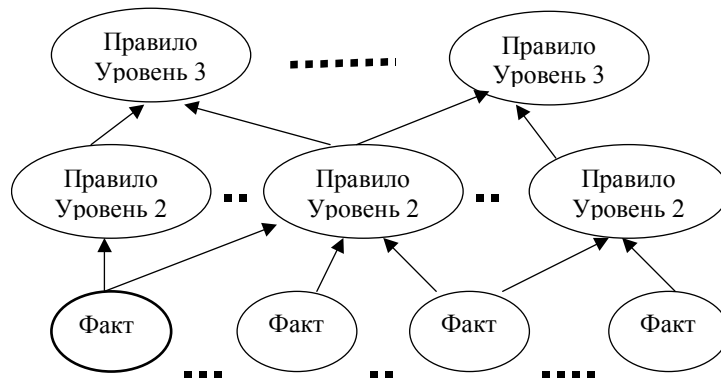


Рис. 3. 3-х уровневое гипотетическое дерево решений

Другой, наиболее часто употребляемой характеристикой БЗ является мощность базы знаний. Под этим термином подразумевают число правил в БЗ. [4, 7]. Однако эта характеристика эффективно может быть использована для описания ЭС или БЗ, работающих в одной предметной области. В других случаях эта характеристика не может быть величиной, пригодной для успешного анализа, сравнения работы ЭС или баз знаний.

Последней характеристикой, которая описывает работу программ, созданных на основе ТИИ, является время ожидания ответа. Понятно, что ЭС является диалоговой системой, и время ответа пользователя зависит от способности блоков ЭС, в первую очередь от БЗ, сформировать и выдать готовое решение.

Таким образом, из анализа характеристик работы блоков ГСАПР видно, что для общего описания функционирования комплекса в целом можно использовать следующие характеристики: время ответа системы, точность вычислений, надёжность и глубина выводов (число уровней правил БЗ).

Литература

1. В.Г. Бондарев. Элементы математической логики. М.: Арт. радиотехн. ак. им. Маршала Говорова, 1962. 128 с.
2. Гибридные экспертные системы в задачах проектирования сложных технических объектов. Материалы краткосрочного научно-технического семинара 24-25 июня 1992 г., под ред. проф. Ю.В. Юдина, с. 1-12, СПб, 1992.
3. Марселус Д.. Программирование ЭС на Турбо Прологе. М.: Фин. и статистика, 1994. 255 с.
4. Уотерман Д. Руководство по экспертным системам / Пер. с англ. под ред. и с предисловием В.Л. Стефанюка. М.: Мир, 1989. 380 с.
5. Липаев В.В. Проектирование программных средств: Учеб. Пособие для вузов по спец. «Автом. сист. обр. информ. и упр.».. М.: Высш. шк., 1990. 303 с.: ил.
6. Математическая логика. Ершов и др. М.: Наука, 1979.. 32 с.
7. Экспертные системы: Инструментальные средства разработки: учебное пособие /Л.А. Керов, А.П. Частиков, Ю.В. Юдин, В.А. Юхтенко; под ред. Ю.В. Юдина. – СПб.: Политехника, 1996. 220 с.
8. Юдин М.Ю. Алгоритмизация процесса проектирования базы знаний гибридной САПР. /ГИТМО (ТУ). СПб, 1998. 21. Деп. В ВИНТИ 29.01.99, № 314-B99 .

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ОПТИЧЕСКИХ ПРИБОРОВ СО ВСТРОЕННОЙ СИСТЕМОЙ СТАБИЛИЗАЦИИ ИЗОБРАЖЕНИЯ

А.В. Демин, Н.С. Копорский, А.В. Сорокин

Изложены вопросы, связанные с имитационным моделированием на ЭВМ оптических систем с управляемой линией визирования в двух плоскостях и со встроенной системой стабилизацией изображения. Приведена математическая модель оптического прибора со встроенной системой стабилизации изображения. Приведены результаты эксперимента.

Моделирование объектов технического проектирования является важным этапом в процессе создания новой техники, так как на этом этапе осуществляется выбор проектного решения, а также возможно принятие решения о стратегии и оптимизации функционирования и верификации выбранного решения или алгоритма функционирования проектируемого технического объекта (ТО). Известны [1] три метода моделирования – физическое, математическое и моделирование с реальной аппаратурой. Не умаляя достоинств первого и третьего методов, заметим, что второй метод, тем не менее, обладает явными преимуществами, хотя бы в плане ресурсных и временных затрат.

Основными требованиями при математическом моделировании является адекватность реальных и моделируемых процессов, обеспечение которой определяется как уровнем ЭВМ, так и уровнем алгоритмизации моделируемого процесса. Известно, что построение математической модели осуществляется на основе физических законов, заложенных в основу функционирования проектируемой системы, а цель математического моделирования – заменить реальный ТО совокупностью более простых устройств, в которых причинно-следственные связи представимы в математической форме, доступной для теоретического исследования. Создание полной математической модели в силу теоремы Тьюринга нецелесообразно, тем самым ТО всегда сложнее его математической модели. Однако следует заметить, что когда ТО настолько сложен и многообразен, то математическая модель его становится слишком сложна и практически не реализуема на ЭВМ без потерь как по точности, так и по достоверности, т.е. с точки зрения отражения ТО в реальном масштабе времени эта модель становится грубым приближением к реальной ситуации. Особенно это характерно для объектов, в состав которого входят устройства, в основе функционирования которых лежат разные физические принципы, но объединенные общей функциональной задачей. В этой связи целесообразно применять имитационное моделирование ТО, суть которого сводится к описанию ТО набором моделей как на функциональном, так и на элементном уровнях. При этом методология и алгоритм построения математической модели ТО для имитационного математического моделирования на ЭВМ включает в себя:

1. анализ функционального состава и структуры моделируемого ТО, типовых режимов функционирования и их взаимосвязи в ТО, влияния внешней среды на функционирование ТО;
2. построение модельных модулей отдельных устройств ТО на основе их поведения; модуля, генерирующего проигрываемую в модели ситуацию; модуля, производящего оценку качества функционирования ТО в моделируемой ситуации.

Рассмотрим всё сказанное применительно к оптическим комплексам, проектирование которых с точки зрения выбора оптимального схмотехнического решения является проблемой.

В данной работе изложены вопросы, связанные с имитационным моделированием на ЭВМ оптических систем с управляемой линией визирования в двух плоскостях и со встроенной системой стабилизацией изображения (ОССИ).

На рис.1 представлена принципиальная схема макета ОССИ, включающая в себя пару подвижных оптических элементов (зеркало и четырехгранная зеркальная призма – ПОЭ) пара оптических каналов (ОС) и пара согласующих неподвижных зеркал, электропривод (ЭП1 и ЭП2), гироскопическое информационно измерительное устройство (ИУ) возмущающих систему внешних воздействий и система управления (СУ). Направления вращения ПОЭ указано стрелками.

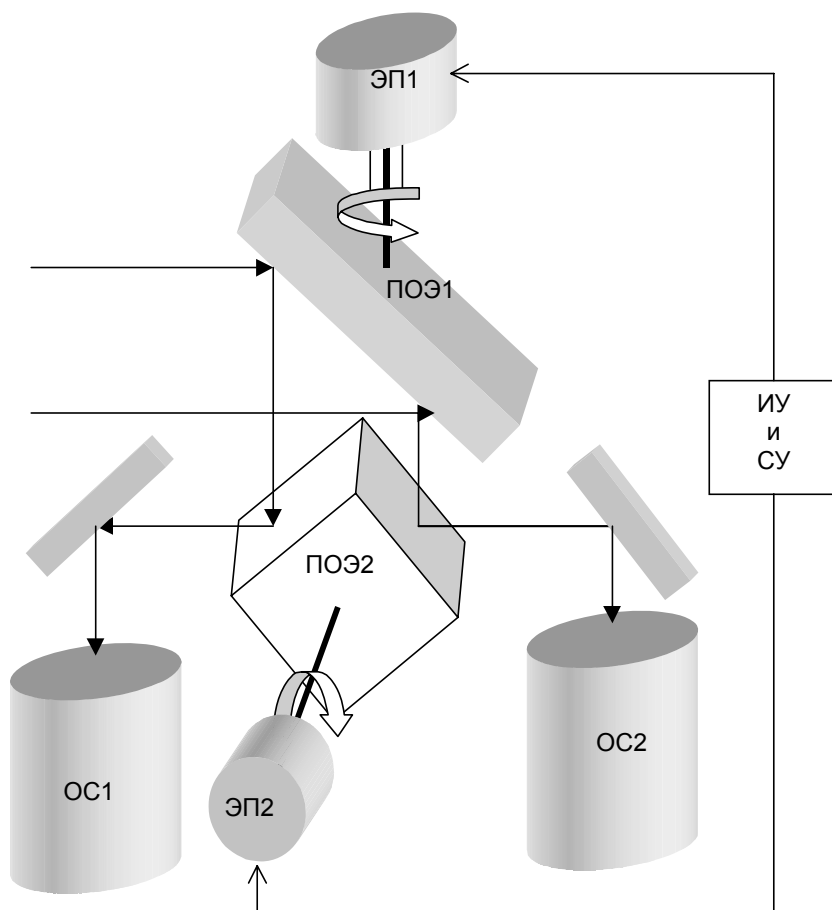


Рис. 1. Принципиальная схема макета ОССИ

Математическая модель ОССИ как сложной системы, включающей в себя как оптическую, так и электромеханическую части, может быть представлена в следующем виде (модель построена на основе работ [2–4]):

$$\begin{bmatrix} \dot{\xi} \\ \dot{x} \\ \frac{I'}{OO'} \end{bmatrix} = \begin{bmatrix} \tilde{A} & 0 & 0 \\ 0 & \tilde{A} & 0 \\ M C_1 + M C_1 \tilde{A} & M C_2 + M C_2 \tilde{A} & 0 \end{bmatrix} \cdot \begin{bmatrix} \xi \\ x \\ \frac{I}{OO} \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ M(C_2 B_2) + B_1 \end{bmatrix} \cdot \xi + \begin{bmatrix} 0 \\ B_3 \\ M C_2 B_3 \end{bmatrix} \cdot U_1 \quad (1)$$

где $\begin{bmatrix} I' \\ OO' \end{bmatrix} = M \begin{bmatrix} I \\ OO \end{bmatrix}$ – матричное уравнение, описывающее действие сложной

оптической системы; $\bar{OO}$ и $\bar{OO}'$ – орты направления оптической оси в пространстве

предметов и изображений; I и I' – лучи в пространстве изображений и предметов, соответственно, $I = [y, V]^T$, y – высота, а V – направляющий косинус луча I ;

$$M = \begin{bmatrix} M_c & 0 \\ 0 & M_{z.} \end{bmatrix}$$

– матрица, представляющая собой оператор действия сложной оптической системы, состоящей из зеркальных и линзовых элементов; M_c – $ABCD$ -матрица действия центрированной системы линз, а $M_{z.}$ – матрица действия системы зеркальных элементов; $\begin{bmatrix} I \\ OO \end{bmatrix} = C_1 \bullet \xi + C_2 \bullet x$ – матрица, учитывающая движение объекта наблюдения и основания системы C_1, C_2 – матрицы постоянных коэффициентов; x – вектор перемещения ОССИ в пространстве; ξ – шум; U – вектор управления оптическими элементами; A_1, B_2, B_3 – матрицы постоянных коэффициентов.

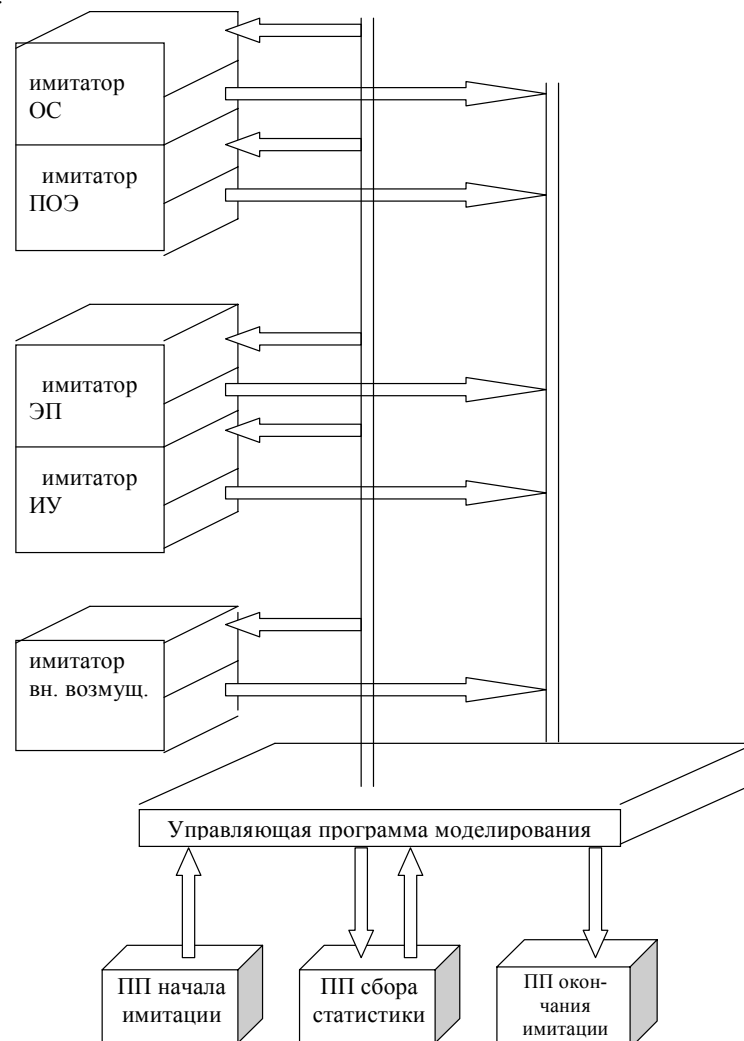


Рис. 2. Имитационная модель ОССИ. (АЛ – алгоритм имитации события; Mt – временной фактор)

Рассматривая ОССИ как объект имитационного моделирования (в соответствии с методикой, предложенной в [1]), можно выделить следующие основные узлы, представленные на рис. 2: ОС – оптическая система; ПОЭ – подвижной оптический элемент; ЭП – электропривод; ИУ – гироскопическое информационно измерительное устройство.

Математически имитационная модель есть не что иное, как совокупность алгоритмов функционирования этих блоков. При этом следует отметить, что для полноты этой модели внешние возмущения (ВВ) тоже должны быть представлены в виде математического блока. Понятно, что одним из переменных факторов при моделировании является временной фактор, который не действует только на параметры телескопической системы. В нашем случае понятие активности [1] следует трактовать как реакция на внешнее возмущение, выражаемое в форме парирования (отработки) ПОЭ этих возмущений, но с отставанием по времени и с амплитудно-частотной характеристикой (АЧХ) максимально приближенной к АЧХ внешних возмущений.

В результате имитационного моделирования на ЭВМ (моделирование проводилось с помощью пакета программ Mat Cad) было установлено, что наиболее "слабое звено" – это ПОЭ. На рис.3 представлен график реальной реакции головного зеркала ОССИ на внешнее возмущение (по оси ординат – мрад; по оси абсцисс – мсек) относительно расчетной (сплошная линия). Реальное текущее положение головного зеркала определялось регистрирующим автоколлимационным устройством (на рис. кривая).

Наименование базы данных : [f51.mat]

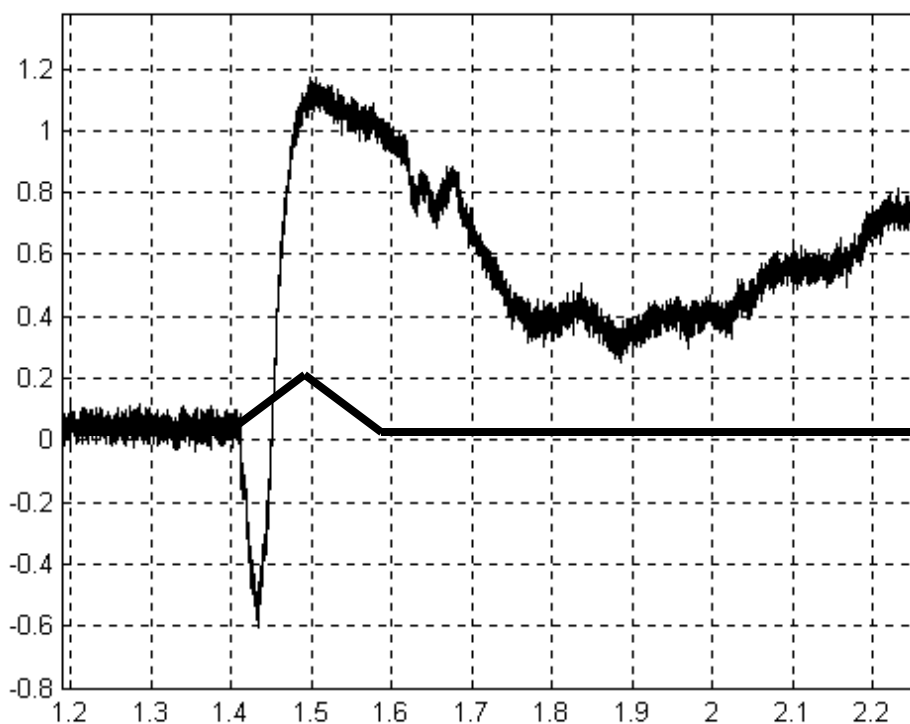


Рис. 3. График реальной реакции головного зеркала ОССИ на внешнее возмущение относительно расчетной реакции (по оси ординат – мрад; по оси абсцисс – мсек)

В качестве исходных параметров динамического режима для моделирования было принято: амплитуда качки ПОЭ ОССИ $\pm 2^\circ$; частота качки 10 Гц. В результате моделирования на ЭВМ и эксперимента были выявлены конструкторско-технологические недостатки макета и определены пути их устранения.

Таким образом, можно сделать следующие выводы:

- предложена математическая модель оптической системы с управляемой линией визирования в двух плоскостях и со встроенной системой стабилизацией изображения, которая может быть положена в основу имитационного моделирования на ЭВМ;

- приведены результаты физического и имитационного моделирования, что говорит о правомерности применения для проектирования технических объектов, в состав которого входят устройства, в основе функционирования которых лежат разные физические принципы, но объединенные общей функциональной задачей.

Литература

1. Максимей И.В. Имитационное моделирование на ЭВМ. М.: Радио и связь, 1988. 232 с.
2. Демин А.В. и др. Серия биноклей со встроенной системой стабилизации изображения. // Оптико-электронные приборы и системы. Сб. научных статей. Выпуск 97 / Под редакцией проф. Э.Д. Панкова и проф. А.В.Демина. СПб., 1997 .С. 8 – 15.
3. Демин А.В. Лапшина И.А. Модель пеленгационного устройства лазерно-информационной системы. // Известия ВУЗов. Приборостроение. 1989. Т. XXIX, № 8.. С.79–84.
4. Демин А.В., Петров И.В. Матричные методы описания действия оптических приборов. // Известия ВУЗов. Приборостроение. 1986. Т. XXIX. № 6. С. 59–66.
5. Демин А.В., Гордеев Д.М. Оптимизация конструкции подвижного зеркала на торсионном подвесе. // Оптико-электронные приборы и системы. Сб. научных статей. Выпуск 97 / Под редакцией проф. Э.Д. Панкова и проф. А.В. Демина СПб., 1997. С. 45 – 49.
6. Демин А.В., Утенков Б.И. Повышение качества изображения в оптических системах со стабилизацией визирной линии. // Оптический журнал. 2001. Т. 68. № 8. С. 21–24.

ПРИНЦИПЫ РАЗРАБОТКИ И МОДЕРНИЗАЦИИ КОМПЛЕКСОВ БОРТОВОГО ОБОРУДОВАНИЯ (КБО) ПЕРСПЕКТИВНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ

А.М. Рогачевский, Ю.Н. Виноградов

Введение

В настоящей статье рассмотрены крайне актуальные на сегодняшний день вопросы разработки, испытания и внедрения бортовой авионики на самолеты и вертолеты гражданского и военного назначения. При этом речь идет о комплексах бортового оборудования (КБО) различного назначения, представляющих собой сложные программно-аппаратные системы.

Круг работ по комплексированию должен охватывать весь спектр гражданских и военных самолетов и вертолетов, в частности, КБО следующих типов объектов:

- гражданских транспортных самолетов;
- гражданских самолетов специального применения (перевозка грузов, исследования земной поверхности, воздействия на атмосферу, самолеты-салоны и т.д.);
- самолетов дальней авиации (ракетоносцы, бомбардировщики, разведчики);
- самолетов фронтовой авиации (штурмовики, бомбардировщики, истребители, разведчики);
- учебно-тренировочных самолетов (УТС);
- самолетов военно-транспортной авиации (ВТА);
- самолетов пограничной авиации;
- самолетов дальнего радиолокационного обнаружения (ДРЛО), разведки;
- вертолетов (ударные, транспортно-десантные, многоцелевые).

В зависимости от назначения объекта и, как следствие, требований ТЗ на комплекс назначение разрабатываемых КБО также различно:

- навигационные (НК);
- пилотажно-навигационные (ПНК);
- индикационные (ИК);
- прицельно-навигационные (ПрНК, ПрВК);
- управления вооружением (СУВ).

Выбор и анализ критерия разработки КБО

Системный подход к проектированию КБО предполагает учет множества факторов, зачастую противоречивых, которые могут быть интегрированы в следующий критерий: "Минимизация суммарной стоимости КБО, отнесенной ко всему его жизненному циклу, с обеспечением возрастающих требований к его эффективности".

Под суммарной стоимостью понимают стоимость приобретения КБО, а также его эксплуатации и модернизации на протяжении всего жизненного цикла.

Стоимость приобретения определяется стоимостью таких составных частей, как разработка и сертификация; а также компонентной и элементной базы.

Стоимость этого этапа жизненного цикла КБО может быть уменьшена за счет стандартизации и унификации аппаратных и программных модулей и, как следствие, возможности использования фрагментов или модификации архитектур базовых разработок. Стоимость компонентной и элементной базы также может быть уменьшена путем ее унификации.

Стоимость эксплуатации определяется в значительной мере затратами на техническое обслуживание (ТО) и ремонт, которые, в свою очередь, зависят от достоверности, знания технического состояния КБО и стоимости ремонта аппаратуры.

Стоимость эксплуатации может быть уменьшена за счет информационной и аппаратурной избыточности КБО и реализации на ее базе свойства отказоустойчивости КБО, т.е. способности работать в условиях отказов и сбоев без снижения ТТХ, а также унификации и быстросъемности элементов КБО, что позволит иметь общий ЗИП для аэродромов базирования ЛА.

Стоимость модернизации, необходимость которой обоснована в разделе 4 настоящей статьи, может быть уменьшена за счет изначальной разработки КБО открытой архитектуры, стандартизации и унификации аппаратных и программных модулей, что позволит легко модернизировать структуру и программное обеспечение (ПО) КБО на протяжении всего жизненного цикла.

Эффективность КБО определяется, в основном, его точностью и надежностью. Обе характеристики задаются в ТЗ на КБО и рассчитываются на основе точностных характеристик датчиков и алгоритмов обработки информации, надежности характеристик аппаратуры и структуры КБО.

Указанные характеристики могут быть достигнуты только путем реализации свойства отказоустойчивости КБО. Данная задача решается введением в КБО информационной и аппаратурной избыточности для обеспечения диагностического контроля, защиты информации, реконфигурации, восстановления.

Таким образом, реализация выбранного критерия разработки комплексов БРЭО для любого ЛА базируется на реализации следующих свойств:

- открытой архитектуры КБО, определяющей возможности дальнейшей модернизации комплекса и реализуемой путем избыточной вычислительной мощности и пропускной способности интерфейсов, стандартизации аппаратных модулей, стандартизации программных модулей с отделением ПО от аппаратуры (загружаемое ПО) и прикладного ПО от системного;
- отказоустойчивости КБО, гарантирующей при работе в условиях отказов элементов, внешних и внутренних помех и сбоев достоверное выполнение своих функций, определяя свое текущее состояние и реконфигурируя, при необходимости, свою архитектуру. Принцип отказоустойчивости КБО реализуется применением информационной и аппаратурной избыточности.

Структура КБО перспективного самолета С-80

В качестве примера рассмотрим систему самолетовождения и индикации ССИ-80, разработанную для нового грузопассажирского самолета АО "ОКБ СУХОГО" С-80. Разработку самолета возглавляет директор программы – главный конструктор АО "ОКБ СУХОГО" Геннадий Александрович Литвинов.

Самолет С-80 и система ССИ-80 в настоящее время находится на этапе летных испытаний.

ССИ-80 разработана в соответствии с требованиями ТЗ и назначением самолета, как бортовой комплекс, решающий широкий круг задач навигации, пилотирования и индикации, в том числе, параметров общесамолетного оборудования.

Таким образом, основным назначением КБО ССИ-80 является:

- автоматизированное, дискретное и ручное самолетовождение с оптимизацией режимов полета по внутренним и зарубежным трассам гражданской авиации;
- управление радиотехническими системами (РТС) в ручном и автономном режимах;
- отображение экипажу пилотажной, навигационной и обзорной информации, а также справочной информации, информации о функционировании бортового оборудования, его отказах и неисправностях, рекомендации по парированию отказов и неисправностей;
- контроль технического состояния бортового оборудования, регистрация, хранение и выдача результатов контроля при техническом обслуживании самолета.

При разработке КБО используются основные элементы ядра комплекса: вычислительная система (ВС) и информационно-управляющее поле (ИУП).

ВС ССИ-80 организована на базе 4 вычислителей и состоит из бортовой цифровой вычислительной машины БЦВМ90-605 (2 шт.) и блока преобразования сигналов БПС-80 (2 шт.) разработки и изготовления ОКБ «ЭЛЕКТРОАВТОМАТИКА».

БЦВМ90-605 является высокопроизводительной вычислительной машиной на RISC - процессоре, предназначенной для решения задач:

- самолетовождения по информации от систем-датчиков;
- обеспечения взаимодействия экипажа с РТС навигации и посадки;
- выдачи на индикаторы необходимой информации;
- отображения на экранах пультов любой знакографической информации.

БПС-80 представляет собой ЦВМ с развитым устройством ввода-вывода, обеспечивающим прием и выдачу информации на индикаторы и потребителям последовательным кодом и в аналоговом виде.

БЦВМ90-605 и БПС-80 состоят из конструктивно функциональных модулей (КФМ), производство которых в настоящее время освоено серийными заводами.

ИУП комплекса является одной из наиболее важных его частей, так как оно обеспечивает связь пилота с КБО и ЛА в целом. В состав ИУП входят информационные системы, а также управляющие органы.

Органы зрения являются наиболее информативной системой человека, поэтому наибольший объем информации передается пилоту с помощью индикационных приборов.

В качестве индикаторов ИУП используются многофункциональные цветные индикаторы МФЦИ-0331 (3 шт.) и индикационная панель пульта управления и индикации ПУИ80С (2 шт.). Управляющая часть ИУП включает в себя кнопочное обрамление МФЦИ-0331, клавиатуру ПУИ80С, а также органы управления на рукоятках и приборной доске ЛА. МФЦИ КБО самолета С-80 представляет собой индикаторы, построенные на базе цветных жидкокристаллических панелей с размерами рабочей области экрана 6×8 дюймов и количеством пикселей 480×640. В состав МФЦИ входит генератор символов, поэтому на экран выводится знакографическая информация, синтезируемая непосредственно в индикаторе по исходным данным вычислителя КБО. На экран МФЦИ выводится также информация от РЛС, которая может быть совмещена со знакографической.

Пульты КБО самолета С-80 построены на базе монохромных электролюминесцентных панелей с размерами рабочей области экрана 3×4 дюйма и количеством пикселей 256×320. На экран пульта выводится знакографическая информация от ВС. Клавиатура включает в себя цифровое и алфавитное наборное поле, а также ряд функциональных и вспомогательных клавиш.

В состав ССИ-80 входит также система ввода информации СВИ. СВИ представляет собой загрузчик полетных данных и предназначен для ввода в ВС аэронавигационной информации и планов полета. Вторым назначением СВИ является регистрация основных событий в полете (отказы аппаратуры, действия летчика и т. д.).

Носителем информации является съемная кассета, выполненная с использованием энергонезависимой FLASH - карты. Подготовка и запись информации на FLASH - карту осуществляется наземными службами с рабочего места оператора СВИ (РМО СВИ) разработки и изготовления ОКБ «ЭЛЕКТРОАВТОМАТИКА». Обработка результатов полета может производиться как на борту с использованием МФЦИ, так и в АТБ на РМО СВИ.

Составной частью ССИ-80 является также спутниковая навигационная система СНС-2.

Таким образом, при разработках КБО для самолета С-80 в достаточной степени учтен основной критерий синтеза КБО, реализованный путем использования:

1. вычислительной системы на базе БЦВМ90, обладающей избыточной на сегодняшний день вычислительной мощностью и пропускной способностью интерфейса;
2. вычислительной системы, имеющей в своем составе 4 вычислителя с реализацией межмашинного обмена;
3. унифицированных КФМ вычислительных средств, средств индикации и управления;
4. стандартных модулей ПО – составных частей прикладного и системного ПО (вычислителей, индикаторов, пультов управления).

Разработка новых и модернизация существующих КБО

При создании новых типов ЛА одновременно производится и разработка комплексов БРЭО. Задача, при этом, решается "с чистого листа": разработка ведется по ТЗ, но ограничения минимальны, поскольку комплекса еще не существует. Примером могут служить ОКР по самолетам С-80, ЯК-130, МИГ-АТ.

Однако продолжительность жизненного цикла ЛА составляет 25–30 лет. За это время, как правило, имеет место существенный прогресс авионики, поэтому модернизации КБО объективно необходимы.

Кроме того, в связи с распадом СССР и последующим затяжным кризисом российской промышленности разработки авиационной техники были фактически прекращены. Постройка новых самолетов и вертолетов происходит в настоящее время достаточно редко. Поэтому задача модернизации бортового оборудования самолетов, в значительной степени устаревшего и не отвечающего современным требованиям, в настоящее время крайне актуальна.

Подтвердим вышесказанное на примере состояния гражданской авиации.

В настоящее время на трассах гражданской авиации пассажиров перевозят, в основном, самолеты ТУ-134 и ТУ-154, на местных воздушных линиях и ближнемагистральных трассах летают АН-24, ЯК-40 и ЯК-42, на дальнемагистральных трассах – ИЛ-86 и ИЛ-62. Кроме того, на трассы выведены ТУ-204 и ИЛ-96.

Базовое БРЭО всех перечисленных самолетов (за исключением ТУ-204 и ИЛ-96) давно устарело и не отвечает современным требованиям. Его состояние таково:

- АН-24, ЯК-40 – набор автономно функционирующих приборов;
- ТУ-134, ТУ-154, ИЛ-62 – комплексы на базе аналоговых вычислителей;
- ЯК-42, ИЛ-86 – первые комплексы на базе цифровых вычислителей.

Даже цифровые комплексы БРЭО ТУ-204 и ИЛ-96, включающие индикаторы на ЭЛТ, ПУИ, КП РТС и цифровые вычислители, разработаны и внедрены достаточно давно.

Необходимость модернизации КБО очевидна.

Возможны два варианта модернизации: “ограниченная” и “полная”. При “ограниченной” модернизации ядро существующего оборудования остается без изменений, кроме того, дорабатывается ПМО и устанавливается дополнительное оборудование, позволяющее довести уровень БРЭО до требуемой с точки зрения состава и круга решаемых задач. При “полной” модернизации существующее оборудование целиком заменяется на современный комплекс БРЭО, за исключением, в некоторых случаях, датчиков пилотажно-навигационной информации.

Первый вариант модернизации был реализован на самолетах ЯК-42, ИЛ-86, АН-72, АН-74 и заключался, в основном, в применении позиционных корректоров бортовых комплексов оборудования (GPS либо аппаратура VOR/DME).

Второй вариант модернизации реализуется в настоящее время на ряде самолетов и вертолетов различного назначения.

Таким образом, задачи разработки КБО для новых типов ЛА и модернизации существующих одинаково актуальны и в равной степени требуют решения.

Заключение

К сожалению, в небольшой статье невозможно подробно осветить все вопросы, связанные с разработкой и модернизацией комплексов бортового оборудования перспективных летательных аппаратов различного назначения. Вне рамок статьи остались вопросы стендового обеспечения с использованием персональных компьютеров в качестве статических и динамических имитаторов систем-датчиков и окружающей среды, принципы реализации модулей ПО и их отработки, вопросы сертификации КБО и ПО, вопросы сопровождения эксплуатации, поддержания эффективности КБО на протяжении жизненного цикла, а также ряд других.

Однако основные принципы разработки и модернизации КБО перспективных летательных аппаратов освещены и проиллюстрированы на примере КБО для современного самолета С-80.

Таким образом, цель настоящей статьи можно считать выполненной.

**УЛУЧШЕНИЕ ИНТЕРФЕРОГРАММ
МЕТОДОМ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ****А.В. Беляков, И.П. Гуров**

Рассмотрены свойства непрерывного многомасштабного анализа (непрерывный вейвлет-анализ). Данный метод реализован применительно к задачам восстановления одномерного интерференционного сигнала из смеси с шумом и улучшения качества картин интерференционных полос на основе использования усредненных по соседним строкам изображения вейвлет-карт.

Введение

Картины полос, получаемые методами муаровой и голографической интерферометрии, часто имеют сложную структуру, локальные особенности и искажены влиянием шума. Для анализа интерференционных картин требуются методы, позволяющие определять не только глобальные пространственно-частотные характеристики интерференционных полос, но и локальные особенности.

Для анализа интерференционных картин широко применяется метод преобразования Фурье, предложенный в работе [1] и основанный на использовании свойств аналитических сигналов [2], теория которых для оптических полей разработана Д. Габором. Преобразование Фурье представляет сигнал, заданный в области независимой переменной, в виде разложения по ортогональным гармоническим базисным функциям, т.е. частотным компонентам. Недостаток преобразования Фурье состоит в том, что эти компоненты вычисляются на основе интегрального преобразования, примененного к реализации сигнала полной протяженности, и поэтому не могут быть локализованы.

В последнее время получил широкое распространение метод анализа сигналов, названный вейвлет-анализом (wavelet analysis) [3, 4]. В литературе данный метод часто называют также многомасштабным (multiscale) анализом [5, 6]. В его основе использованы идеи, сходные с Фурье-преобразованием. Главное отличие состоит в выборе базисных функций или функций, с которыми осуществляется свертка сигнала, а именно, вейвлетов. Вейвлеты – это функции, ненулевая часть которых локализована на ограниченном интервале изменения независимой переменной и являются пространственно масштабируемыми, что позволяет разделять сигналы по различным частотным компонентам и одновременно изучать их локальную структуру с разрешением, соответствующим выбранному масштабу. Именно поэтому свойства вейвлет-преобразования позволяют успешно решать задачи анализа локальных свойств сигналов и изображений.

Английский термин wavelet (“маленькая волна”) хорошо передается русским термином “волновой всплеск”. При вейвлет-преобразовании в пространственной области функция-всплеск свертывается с сигналом при различных масштабах этой функции. В результате оказывается возможным выделять местоположение особенностей сигнала, а не просто осуществлять его глобальное частотное разложение.

В статье рассмотрены свойства непрерывного многомасштабного анализа и представлен пример выделения сигнала из смеси с шумом, а так же улучшения интерференционной картины на основе метода волновых всплесков. Далее по тексту используются термины “всплеск” и “вейвлет”, поскольку в настоящее время терминология не установлена в окончательном виде.

Непрерывный вейвлет-анализ

Всплеск (или вейвлет) - это функция, удовлетворяющая двум условиям: функция должна быстро убывать в области независимой переменной; площадь под графиком функции должна быть равна нулю. Первое условие означает, что существенно ненулевая часть всплеска сосредоточена на отрезке конечной протяженности. Не попавшие в отрезок значения сигнала будут слабо влиять на результат. Следовательно, появляется возможность локализовывать особенности сигнала. Именно это отличает вейвлет-преобразование от преобразования Фурье.

Непрерывное вейвлет-преобразование одномерного сигнала $s(x)$ определяется в форме

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} s(x) h^*\left(\frac{x-b}{a}\right) dx, \quad (1)$$

где $h^*(x)$ является базисной вейвлет-функцией (звездочкой обозначено комплексное сопряжение). Результат преобразования является функцией, зависящей от двух переменных: “текущей” координаты b и масштаба a . При каждом масштабе, согласно (1), исходная функция $h^*(x)$ растягивается по горизонтали и сжимается по вертикали.

Обозначив волнистой линией Фурье-преобразование функции в области частот u , на основе известной теоремы о свертке из (1) получим соотношение

$$\tilde{W}(a,u) = \sqrt{a} \tilde{s}(u) \tilde{h}^*(au). \quad (2)$$

С изменением масштаба a спектр вейвлета $\tilde{h}^*(au)$, рассматриваемый при обработке сигнала как частотная характеристика фильтра, сжимается и расширяется по соответствующим осям, сохраняя при этом неизменной свою форму.

Исходная функция $s(x)$ может быть восстановлена с помощью обратного преобразования

$$s(x) = \frac{1}{C} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{\sqrt{a}} W(a,b) h\left(\frac{x-b}{a}\right) \frac{da db}{a^2}, \quad (3)$$

где C - норма базисной функции, определяемая как

$$C = \int_0^{\infty} \frac{\tilde{h}^2(u)}{u} du.$$

Для восстановления сигнала необходимо, чтобы выполнялось условие $\tilde{h}^2(0) = 0$, и площадь под графиком базисной функции должна быть равна нулю.

Для анализа интерференционных полос удобно использовать вейвлет Морле, который определяется выражением

$$h(x) = \exp[-j\alpha x - x^2/(2\sigma^2)], \quad (4)$$

где α – частотный параметр, σ – параметр масштаба.

На рис. 1 приведена вейвлет-функция Морле. При увеличении параметров α и σ в (4) соответственно возрастает частота “заполнения” и ширина огибающей. Выбор подходящих значений параметров позволяет “адаптировать” свойства вейвлет-преобразования к конкретной задаче обработки интерференционных полос. Спектр вейвлета Морле сконцентрирован в окрестности частоты α_0 , спектр сжатого в n раз вейвлета, согласно известным свойствам Фурье-преобразования (свойству подобия), будет сконцентрирован вокруг частоты $n\alpha_0$ (см. рис. 1, б). Из рис. 1 видно, что использование вейвлета Морле позволяет осуществлять локальную спектральную фильтрацию интерференционных сигналов.

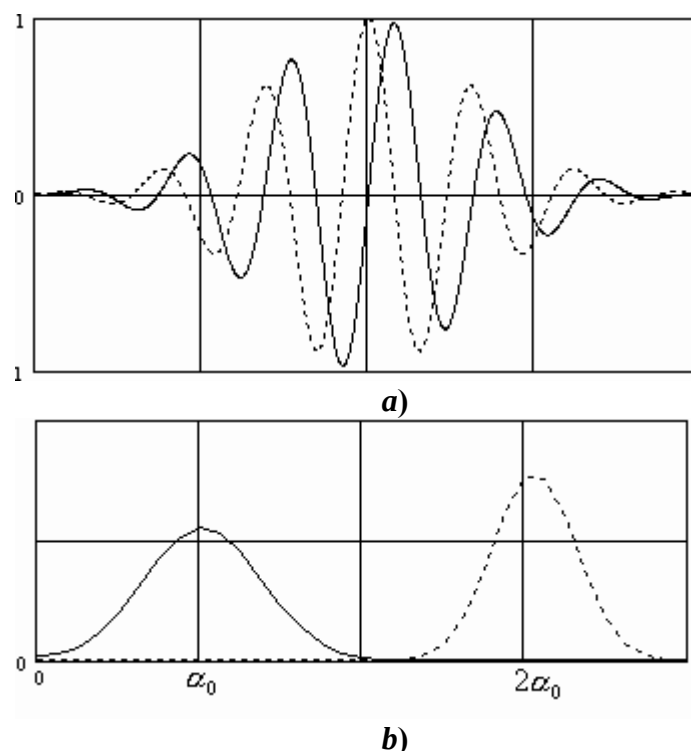


Рис. 1. Комплексный вейвлет Морле (a) и изменение Фурье-спектра при изменении параметра модуляции (b)

Фильтрация сигналов из смеси с шумом

Интерференционные сигналы часто оказываются значительно зашумленными. Получение сигналов, “очищенных” от шумов, является одной из важнейших задач обработки. Рассмотрим способ, использующий для этой цели непрерывное вейвлет-преобразование.

На рис. 2 представлен интерференционный сигнал и карта вейвлет-коэффициентов. Коэффициенты представлены с помощью градации яркости. Масштабы изменяются по вертикали от единичного внизу (соответствующего высокочастотному сигналу) до ста сверху.

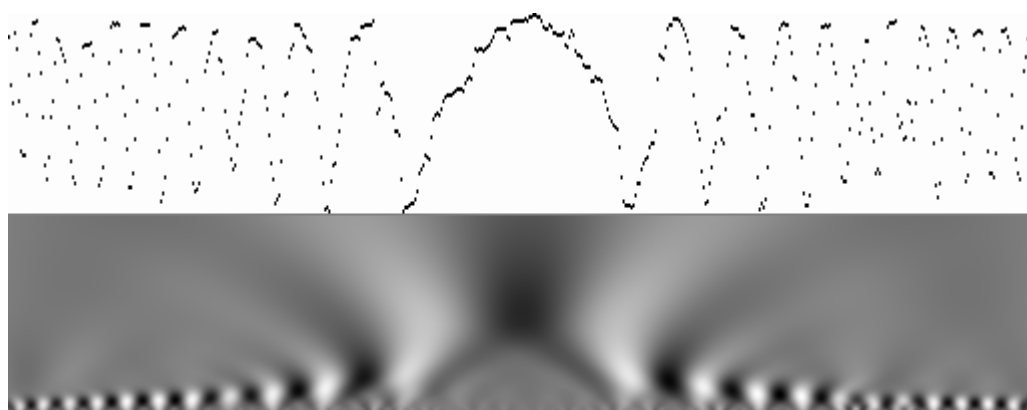


Рис. 2. Вейвлет-преобразование с использованием всплеска Морле. Сигнал (вверху) и карта его вейвлет коэффициентов

На вейвлет-карте хорошо видны масштабы, характерные для данного сигнала. Как видно из рисунка, пики и провалы на вейвлет-карте соответствуют локальному периоду сигнала. Чем больше по абсолютному значению коэффициенты разложения, тем

большую роль они будут играть при обратном преобразовании, как это видно из формулы (3). Остальные коэффициенты можно считать представляющими шум, и их вклад не следует учитывать при обратном преобразовании, так же как вклад коэффициентов из области высоких частот.

Таким образом, следует учитывать коэффициенты соответствующие локальным максимумам и достаточно значительные по модулю, а остальные обнулить. Результат такого преобразования вейвлет-карты, а также результат обратного преобразования представлены на рис. 3. Как видно из рисунка, сигнал стал гладким, и появилась возможность различить изменения сигнала не только в его центральной части, но и по краям. При этом амплитуда сигнала несколько уменьшилась.

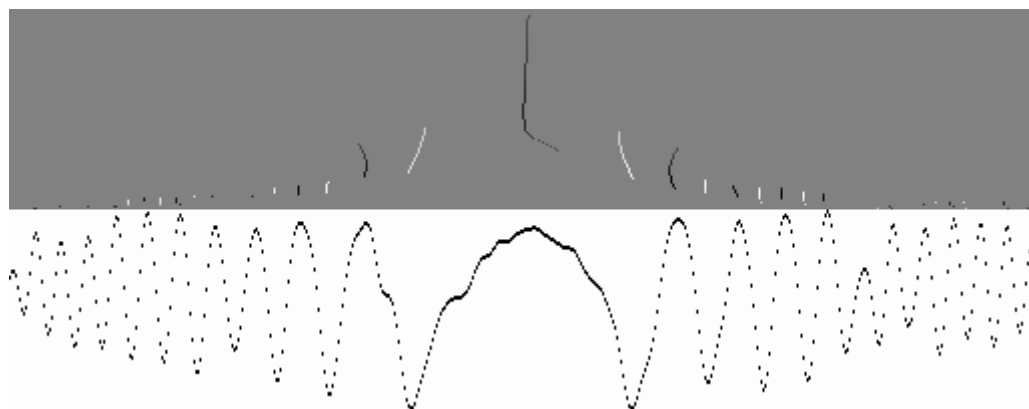


Рис. 3. Прореженная карта вейвлет-коэффициентов (вверху) и данные полученные при обратном преобразовании

Рассмотренный метод дает хорошие результаты для одномерных сигналов, но мало пригоден для построчной обработки интерферограмм, так как при такой обработке информация, содержащаяся в соседних строках, не учитывается.

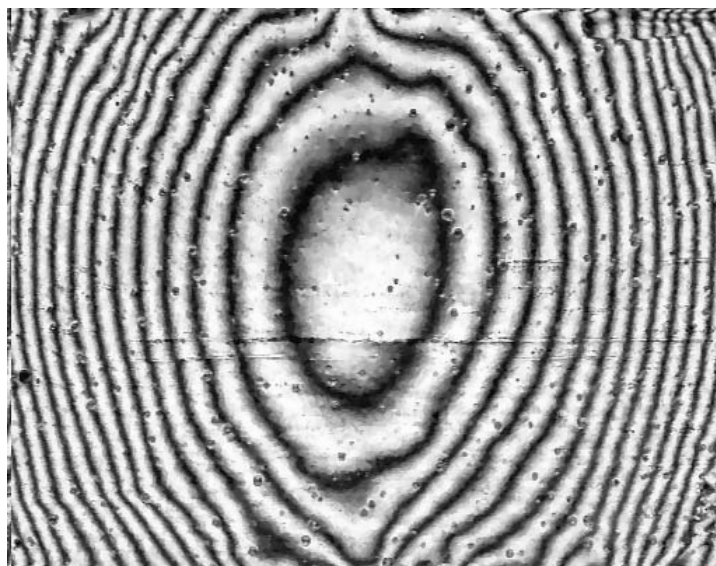


Рис. 4. Исходная интерферограмма

Рассмотрим интерферограмму, приведенную на рис. 4. Данная интерферограмма характеризуется высоким уровнем шумов. Границы интерференционных полос нечетки. Часто встречаются небольшие темные вкрапления размером примерно в пять пикселей по каждой координате.

Для фильтрации картины полос проведем вейвлет-преобразование каждой строки изображения, используя вейвлет Морле. Для соседних строк коэффициенты, находящиеся в одинаковых позициях вейвлет-карт, не должны сильно различаться. Если такое отличие велико, то можно сделать вывод о наличии шума. Различие следует сгладить, тогда при обратном преобразовании соседние строки будут меньше отличаться друг от друга. Для того, чтобы локальные особенности изображения не влияли на результат обратного преобразования, следует использовать статистику по нескольким соседним строкам.



Рис. 5. Улучшенное изображение

На рис. 5 показан результат обработки, полученный усреднением вейвлет-карт семи соседних строк. Значения вейвлет-карт учитываются с весовыми коэффициентами, зависящими от расстояния до “текущей” обрабатываемой строки. Как видно из рисунка, в результате удалось добиться четкой картины полос, свободной от локальных особенностей и шумов. Таким образом, можно заключить, что вейвлет-преобразование с использованием предложенной методики, позволяет улучшить качество интерферограмм, получаемых методами муаровой и голографической интерферометрии.

Литература

1. Takeda M., Ina H., Kabayashi S. Fourier – transform method of fringe-pattern analysis for computer-based topography and interferometry //J. Opt. Soc. Amer. 1982. Vol. 72. №1. P.156-160.
2. Папулис А. Теория систем и преобразований в оптике. М.: Мир, 1971.
3. Добеши И. Десять лекций по вейвлетам. Ижевск: НИЦ “Регулярная и хаотическая динамика”, 2001.
4. Воробьев В. И., Грибунин В. Г. Теория и практика вейвлет-преобразования. СПб.: Изд-во ВУС, 1999.
5. Akansu A.N., Haddad R.A. Multiresolution signal decomposition. Boston: Academic Press, 1992.
6. Mallat S. A theory for multiresolution signal decomposition: the wavelet representation //IEEE Trans. on Pattern Anal. and Mach. Intel. 1989. Vol. 11. № 7. P. 674-693.
7. Angrisany L., Daponte P., D’Apuzzo M. A method for the automatic detection and measurement of transients. //Measurement 1999. Vol. 25. P. 19-40.

ПАРАМЕТРИЧЕСКИЙ АНАЛИЗ ПОЛУТОНОВЫХ ИЗОБРАЖЕНИЙ

К.Г. Коротков, Б.А. Крылов

Прогресс в области вычислительной и телевизионной техники, стремительное развитие методов и алгоритмов автоматизированной обработки и анализа данных породили новые научно-технические направления, одним из которых является исследование изображений с целью извлечения количественной информации об объектах, явлениях или процессах, протекающих в поле изображения. В данном случае изображение выступает объектом исследования.

Подобный подход широко используется при изучении космического пространства, в навигации, биологии и медицине, в промышленности при автоматизации производственных процессов и в других областях.

Исследование изображений основывается на их обработке и анализе. Обработка изображения заключается во внесении в исходное изображение тех или иных изменений по сравнению с оригиналом с целью привести изображение к виду, удобному для дальнейшего анализа или понимания. Таким образом, в результате обработки изображений вновь получается изображение.

Анализ изображения заключается в изучение отдельных характеристик, составных частей, фрагментов или отдельных объектов в поле изображения. К настоящему времени в литературе нет устоявшегося определения термина "анализ изображения". В нашем случае под анализом изображения будем понимать измерительный аспект обработки видеоданных. Следовательно, анализ изображения будет заключаться в определении различных количественных данных изображения – параметров.

В целом параметрический анализ изображения предусматривает вычисление признаков в целом, для отдельных фрагментов или объектов в поле сканирования и определяется задачами исследования. Таким образом, в результате анализа изображения из него выделяется неизобразительная информация.

Формальной моделью полутонového изображения является двумерная матрица $\|B_{ij}\|$, где $i=1,2,\dots,X$; $j=1,2,\dots,Y$; X,Y – размеры изображения, взятые с требуемой точностью, а каждый элемент характеризуется яркостью $b(x,y)$.

При цифровой обработке происходит квантование яркости изображения, т.е. функция B_{ij} принимает значения только из конечного набора чисел $B_{ij}=\{b_1, b_2, \dots, b_k\}$. В нашем случае $B_{ij}=\{0,1,2,3,\dots,255\}$.

Минимальным составляющим элементом изображения является пиксель (точка, дискрет) характеризующийся местоположением (x,y) и яркостью b , $b(x,y) \in \{0,1,2,\dots,255\}$. Пиксель может быть пустым (фон) или значащим (содержащим элемент изображения).

В статье рассматривается комплекс параметров, часть из которых общеупотребительны, часть используется относительно недавно, некоторые предложены в настоящей работе. Информативность и независимость параметров определялась методом корреляционного анализа при обработке конкретных экспериментальных данных.

Все множество количественных параметров можно разбить на несколько групп.

- Интегральные параметры.
- Спектральные параметры.
- Фрактальные параметры.
- Структурные параметры.

Интегральные параметры.

(1) Общая площадь изображения S вычисляется по формуле

$$S = \sum_{i=1}^n \sum_{j=1}^m p_{ij}$$

где $p_{ij} = 1$, если $b(i,j) < L$ $p_{ij} = 0$, если $b(i,j) = L$; L – пороговое значение яркости фона; n, m – размеры матрицы B .

(2) Плотность изображения вычисляется как отношение площади свечения изображения к общей площади региона. Если обозначить $A = |a(x,y)| \in B = |b(x,y)|$ как регион, включающий в себя изображение, то плотность изображения D вычисляется как

$$D = \sum_{i \in A} p_i / \sum_{i \in A} S_i,$$

где $p_i = 1$, если $a(x,y) < L$, $p_i = 0$, если $a(x,y) = L$; L – пороговое значение фона; $S_i = 1$ для всех $i \in A$.

(3) Длина периметра изображения вычисляется как длина огибающей развертки изображения. Рассмотрим процесс отображения изображения в обобщенную систему координат, т.е. построим массив развертки. Развертка изображения осуществляется относительно полюса, который определяется как центр тяжести изображения. Развертка представлена двумерным массивом $M[n \times m]$, в котором n – величина в пикселях максимального размаха изображения, m – количество угловых срезов φ , используемых для получения развертки. При развороте изображения пиксели рассматриваются как точки, имеющие габариты, поэтому при угловом шаге менее 1° для малых радиусов R в массив M записывается один и тот же пиксель несколько раз. Чтобы избежать этих повторов и учитывать пиксель 1 раз, массив M формируется следующим образом:

$$M[i, j] = \begin{cases} 0 \\ 255 \\ b(x, y) + 1 \end{cases}$$

$M[i,j]=0$, если пиксель $b(x,y)$ уже записан в массив; $M[i,j]=255$, если $b(x,y)=251$; $M[i,j]=b(x,y)+1$, если $b(x,y)<251$, где $x = x^\circ + R \cos(\varphi)$; $y = y^\circ + R \sin(\varphi)$; x°, y° – координаты центра тяжести изображения.

Тогда по массиву M легко построить массив радиусов линии, огибающей изображение.

Спектральные параметры.

(4) Интегральная яркость изображения вычисляется по формуле

$$PJ = \sum_{i=0}^n d[i]i / \sum_{i=0}^n d[i];$$

где функция $d=N(b_i)$ задает яркостной спектр изображения, значения которой показывают, сколько пикселей с яркостью b_i содержится в этом изображении.

(5) Ширина яркостного спектра изображения определяет размах яркостного спектра и вычисляется как разность самого светлого значения функции $d = \max \{N(b_i)\}$ в спектре и самого темного $d = \min \{N(b_i)\}$.

Фрактальные параметры.

(6) Коэффициент формы изображения вычисляется как отношение квадрата длины периметра изображения к общей площади свечения.

$$K_f = L^2 / S;$$

(7) Фрактальность (по Мандельброту) f вычисляется как отношение длин периметров изображения свечения, полученных при различных масштабах изображения.

Структурные параметры.

Как было показано выше, изображение может быть представлено как функция $f(x)$ некоторого аргумента x от угла в пределах $[0-2\pi]$. В качестве функции $f(x)$ может выступать максимальная длина радиуса изображения, длина медианы, максимальная яркость или средние величины по радиусу.

В нашем случае функция дискретизировалась по углу поворота радиус-вектора на величину $0,35156^\circ$. В результате для каждой $f(x)$ образуются два вектора $F1^{<J>}$ и $F2^{<J>}$ размером 1028×1 каждый, причем $F1$ характеризует геометрическую компоненту, а $F2$ – яркостную.

Функция $f(x)$ в пределах от 0 до 2π меняется от $f_{\min}$ до $f_{\max}$. Разделим область значений функции $f(x)$ на N равновеликих интервалов и определим $P(f)$ – функцию распределения плотности значений функции $f(x)$ в интервале $[f_{\min}, f_{\max}]$.

(8) Для описания функции можно применить стандартный язык теории вероятности: вычислить математическое ожидание $M(f)$, посчитать дисперсию $D(f)$ и высшие центрированные моменты $M_1(f)$, $M_k(f)$ и т.д..

(9) По аналогии с широко используемым понятием термодинамической энтропии можно определить энтропию функции $f(x)$ следующим образом:

$$E = - \int P(f) \log(P(f)) df, \quad f = [f_{\max}, f_{\min}].$$

Энтропия является мерой отклонения от равновесия: она убывает при стремлении к равновесному состоянию. Для неравновесных функций, имеющих большой разброс $[f_{\min}, f_{\max}]$, значение $P(f)$, обратно пропорциональное $(f_{\max} - f_{\min})$, $P(f) < 1$, что приводит к большим положительным значениям E , в то же время для гладких равновесных функций $P(f) > 1$, что приводит к отрицательным значениям энтропии.

(10) Характеристикой регулярности процесса является функция автокорреляции, которую можно определить следующим образом:

$$K(y) = \int (f(x) - f_0)(f(x+y) - f_0) dx$$

Предполагается, что $f(x+2\pi) = f(x)$. Эта функция равна 1 в точках 0 и 2π , наличие максимумов свидетельствует о регулярности процесса, если $K(y)=0$, корреляции на расстоянии y отсутствуют.

Данный подход реализован в программном комплексе "GDV Technique". На его основе разработан ряд модификаций программно-аппаратных комплексов для исследования параметров газоразрядного свечения объектов различной природы.

В основе метода лежит свойство объектов, помещённых в электромагнитное поле высокой напряженности, инициировать различные виды газового разряда. Изображение свечения газового разряда (ГРВ-грамма) представляет собой сложную двумерную, фрактальную фигуру, пространственные и спектральные характеристики которой несут информацию о структуре и свойствах объекта, индуцировавшего разряд.

Комплекс используется для исследования энергоинформационного обмена в биологии и медицине, для анализа структуры различных объектов в технике и криминалистике.

Программно-аппаратный комплекс сертифицирован комитетом по новой медицинской техники МЗ РФ и выпускается серийно, внедрен в медицинских и исследовательских центрах России, Англии, Германии, Словении, США, Финляндии, Швеции.

ВОССТАНОВЛЕНИЕ ФАЗЫ ИНТЕРФЕРЕНЦИОННЫХ ПОЛОС МЕТОДОМ НЕЛИНЕЙНОЙ ДВУМЕРНОЙ ФИЛЬТРАЦИИ КАЛМАНА

А.С. Захаров

Исследованы характеристики двумерного дискретного нелинейного фильтра Калмана при динамическом оценивании параметров зашумленных интерференционных полос. Представлены результаты обработки двумерных интерферограмм, полученных при отражении света от поверхности арсенида галлия и графита до и после воздействия на поверхность фемтосекундного лазерного импульса.

Введение

При исследовании свойств негладких поверхностей во многих случаях предпочтительны неразрушающие оптические методы. При отражении оптического излучения от исследуемой поверхности происходит изменение формы волнового фронта, характер которого зависит от характеристик поверхности. Фаза интерференционных полос содержит полезную информацию, которую необходимо извлечь с максимальной разрешающей способностью и помехоустойчивостью по отношению к шуму наблюдений.

Одной из актуальных технологических задач является модификация формы поверхности объектов из различных материалов при воздействии интенсивных лазерных импульсов малой длительности. При этом в ряде случаев, включая исследование динамики изменения формы поверхности после воздействия короткого лазерного импульса, важно контролировать изменение формы поверхности бесконтактными методами интерферометрии с повышенным разрешением.

Одним из возможных методов динамической обработки интерференционных картин является нелинейная дискретная фильтрация Калмана [1-3]. Широко используемые методы, основанные на преобразовании Фурье [4], и методы фазового сдвига [1] не могут использоваться в динамическом режиме, поскольку требуют получения всех значений интенсивности для определения фазы, частоты и других параметров полос. Методы линейной фильтрации, например, дискретная линейная фильтрация Калмана [3], не позволяют определить значения параметров, от которых сигнал полос зависит нелинейно, в частности, значения частоты и фазы полос. В настоящей работе рассматривается метод нелинейной дискретной фильтрации Калмана, который может быть использован для обработки одномерных интерференционных сигналов и двумерных картин интерференционных полос с восстановлением развернутой фазы полос, характеризующей форму поверхности.

1. Дискретная нелинейная фильтрация Калмана

Нелинейный фильтр Калмана предназначен для динамического оценивания параметров, нелинейно связанных со значением сигнала. Такой фильтр определяется [2, 3] уравнением наблюдения

$$\mathbf{s}(k) = \mathbf{h}(\boldsymbol{\theta}(k)) + \mathbf{n}(k) \quad (1)$$

и уравнением системы

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + \mathbf{f}(\boldsymbol{\theta}(k)) \Delta x + \mathbf{w}(k). \quad (2)$$

В уравнениях (1) и (2) $\mathbf{s}(k)$ - наблюдаемый сигнал, $\boldsymbol{\theta}(k)$ - вектор параметров, $\mathbf{h}(\cdot)$ и $\mathbf{f}(\cdot)$ - известные нелинейные векторные функции, Δx - шаг дискретизации, $\mathbf{n}(k)$ - шум наблюдений, $\mathbf{w}(k)$ - формирующий шум. Алгоритм фильтрации носит рекуррентный характер и основан на предсказании значений параметров на следующий шаг дискретизации с последующей коррекцией предсказания $\boldsymbol{\theta}_{pr}(k)$ по данным наблюдения. Если удастся найти оценку вектора параметров $\boldsymbol{\theta}_{pr}(k)$, то для определения погрешности

этой оценки $\theta_{er}(k) = \theta(k) - \theta_{pr}(k)$ можно применить известный метод одномерной дискретной линейной фильтрации Калмана [3]. При восстановлении фазы двумерных картин интерференционных полос требуется двумерный нелинейный фильтр Калмана.

2. Обработка двумерных картин полос методом нелинейной фильтрации Калмана

Построчная динамическая фильтрация интерференционной картины полос рассмотрена в [2]. Независимая фильтрация строк может приводить к искажению общей картины полос, поскольку в двух соседних точках столбца погрешности оценки фазы могут значительно отличаться друг от друга, если фильтрация отдельных строк осуществляется независимо, и возможны значительные погрешности в определении пространственной частоты в вертикальном направлении. Квазидвумерная обработка поочередно по строкам и столбцам не решает эту проблему, поскольку не удастся полностью устранить большие скачки фазы. Необходима двумерная обработка интерференционной картины с использованием для предсказания значений параметров не только в предыдущей точке строки, но и в предыдущей точке столбца данных о значениях интенсивности света в картине интерференционных полос.

Для двумерной обработки картин интерференционных полос с помощью дискретного нелинейного фильтра Калмана необходимо оценивать вектор параметров вида $(s_0, s_m, \Phi, u_H, u_V)^T$, где s_0 - фоновая составляющая интерференционной картины, s_m - амплитуда полос, Φ - фаза полос, u_H - пространственная частота в горизонтальном направлении, u_V - пространственная частота в вертикальном направлении. В разработанном варианте двумерного нелинейного фильтра Калмана параметры сигнала в точке с координатами (i, j) предсказывались согласно простым правилам

$$s_0(i, j) = [s_0(i-1, j) + s_0(i, j-1)]/2, \quad (3)$$

$$s_m(i, j) = [s_m(i-1, j) + s_m(i, j-1)]/2, \quad (4)$$

$$\Phi(i, j) = [\Phi(i-1, j) + 2\pi u_V(i-1, j)\Delta y + \Phi(i, j-1) + 2\pi u_H(i, j-1)\Delta x]/2, \quad (5)$$

$$u_H(i, j) = [u_H(i-1, j) + u_H(i, j-1)]/2, \quad (6)$$

$$u_V(i, j) = [u_V(i-1, j) + u_V(i, j-1)]/2, \quad (7)$$

где Δx и Δy - шаги дискретизации в горизонтальном и вертикальном направлениях.

В качестве вектора наблюдений в точке (i, j) использовался вектор вида $[s(i, j), s(i, j+1), s(i+1, j), s(i+1, j+1)]^T$. Для предсказания вектора в каждой точке использовались формулы:

$$s(i, j) = s_0(i, j) + s_m(i, j)\cos[\Phi(i, j)], \quad (8)$$

$$s(i, j+1) = s_0(i, j) + s_m(i, j)\cos[\Phi(i, j) + 2\pi u_H(i, j)\Delta x], \quad (9)$$

$$s(i+1, j) = s_0(i, j) + s_m(i, j)\cos[\Phi(i, j) + 2\pi u_V(i, j)\Delta y], \quad (10)$$

$$s(i+1, j+1) = s_0(i, j) + s_m(i, j)\cos[\Phi(i, j) + 2\pi u_H(i, j)\Delta x + 2\pi u_V(i, j)\Delta y]. \quad (11)$$

Таблица 1

Погрешности восстановления полной фазы полос

Отношение сигнал-шум	СКО фазы, рад	Максимальная погрешность фазы, рад
1,6	0,2	0,99
1,0	0,28	1,34
0,67	0,43	2,25
0,5	0,70	7,12

На рис. 1 приведены примеры обработки двумерной картины интерференционных полос. Отношение сигнал-шум, определяемое как $S/N = s_m/\sigma_n$, было равно 0,5 (рис. 1, а-б) и 1 (рис. 1, в-г) соответственно. Среднеквадратические и максимальные погрешности фазы при двумерной обработке значительно меньше, чем при построчной фильтрации. Кроме этого, двумерная фильтрация позволяет обрабатывать сильно зашумленные

картины полос. Полученные результаты обработки для различных значений отношения сигнал/шум приведены в табл. 1.

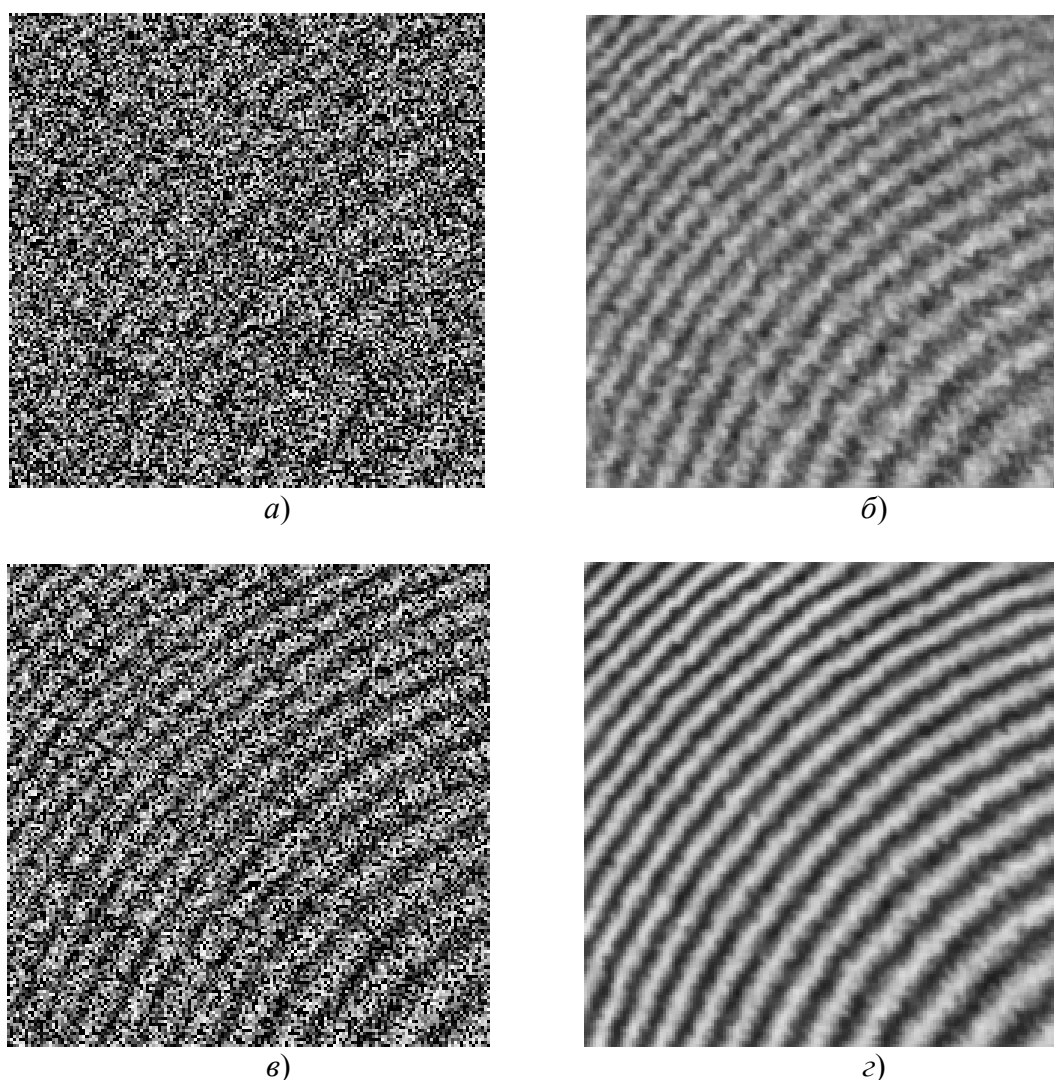


Рис. 1. Исходные картины полос (а), (в) и картины полос, обработанные двумерным фильтром (б), (г).

Метод двумерной дискретной нелинейной фильтрации Калмана был применен для обработки экспериментальных двумерных интерферограмм, полученных при отражении измерительной волны от поверхностей различных материалов, подвергнувшихся воздействию фемтосекундных лазерных импульсов.

На рис. 2 приведены интерферограммы, полученные перед воздействием импульса на образец арсенида галлия (рис. 2, а), после воздействия импульса (рис. 2, б) и после окончания переходных процессов в веществе (рис. 2, в). Восстановленные разности фаз интерферограмм приведены на рис. 2, г-д.

Было установлено, что нелинейный двумерный фильтр Калмана обеспечивает высокую разрешающую способность, поскольку локальное оценивание фазы, используемое в методе нелинейной фильтрации Калмана обеспечивает более высокую разрешающую способность по сравнению с глобальным оцениванием фазы, применяемым при использовании преобразования Фурье в конечных пределах протяженности изображения.

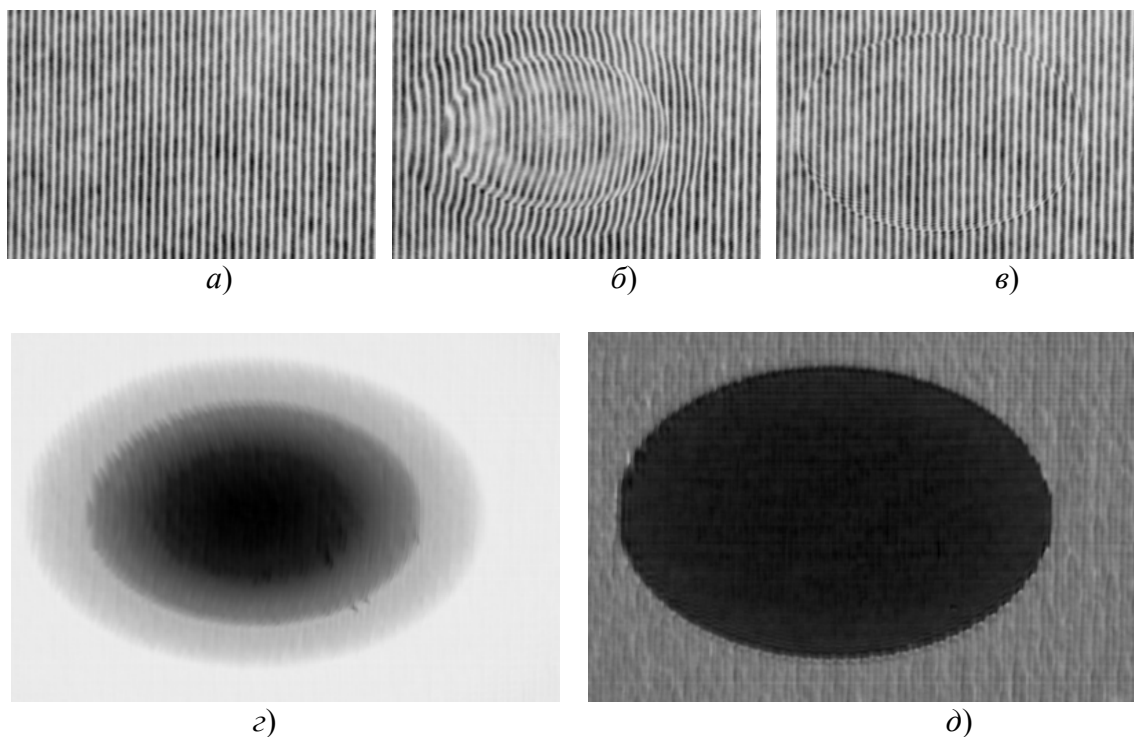


Рис. 2. Исходная (а), промежуточная (б) и итоговая (в) интерферограммы, полученные для арсенида галлия, и восстановленные отклонения фаз в промежуточной (г) и итоговой (д) картинах полос (б, в) от фазы исходной картины полос (а).

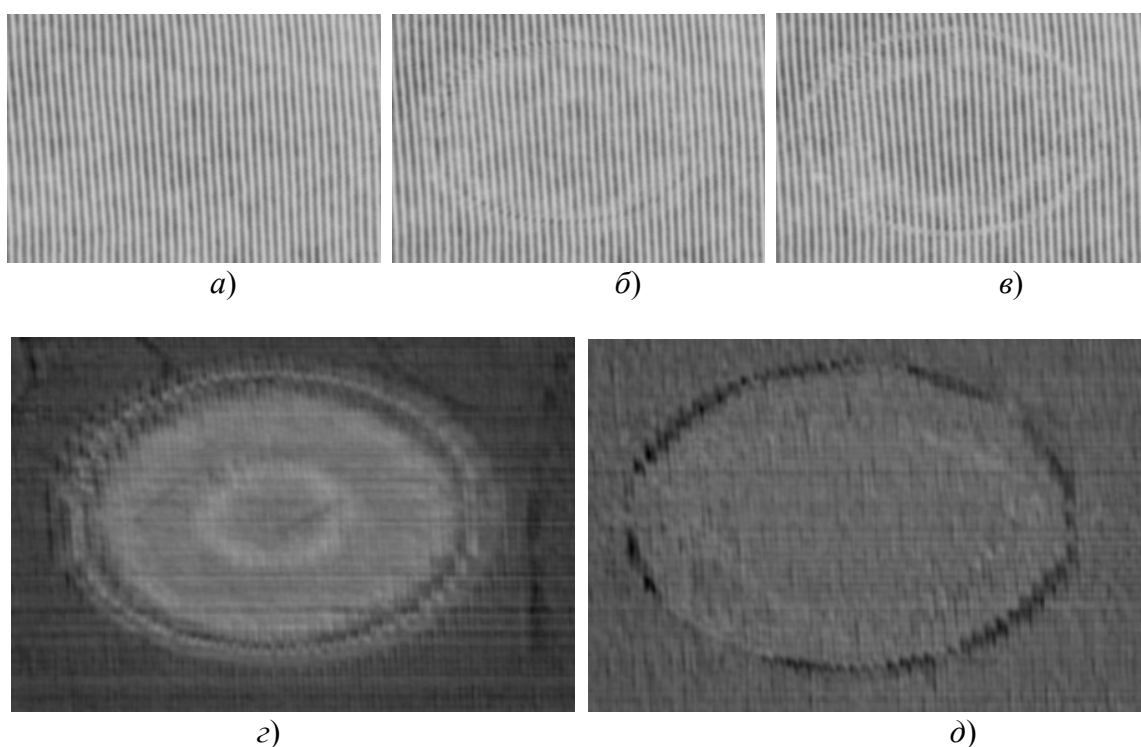


Рис. 3. Исходная (а), промежуточная (б) и итоговая (в) интерферограммы, полученные при модификации поверхности графита, и восстановленные отклонения фаз промежуточной (г) и итоговой (д) картин полос (б, в) от фазы исходной картины полос (а).

На рис. 3 приведены интерферограммы, полученные до воздействия лазерного импульса на образец графита (рис. 3, а), после воздействия импульса (рис. 3, б) и после окончания переходных процессов (рис. 3, в). Приведены восстановленные при двумерной обработке отклонения фаз промежуточной (рис. 3, г) и итоговой (рис. 3, д) картин полос от фазы исходной картины полос.

Заключение

Полученные результаты позволяют сделать вывод о возможности применения нелинейного фильтра Калмана для обработки интерференционных полос с медленно изменяющимися параметрами в условиях влияния интенсивных высокочастотных помех. Важным преимуществом фильтра этого вида является возможность анализа интерференционного поля с получением оптимальных динамических оценок фазы и частоты полос в локальных точках. Предпочтительна двумерная фильтрация, при которой среднеквадратические и максимальные погрешности оценки параметров значительно меньше, чем при построчной обработке. Погрешность нелинейной фильтрации снижается при получении адекватной априорной информации, т.е. при использовании эффективных методов предварительной обработки.

Фильтр Калмана восстанавливает полную фазу полос, тогда как преобразование Фурье позволяет получить только приведенную фазу в пределах $[-\pi, \pi]$. Кроме этого, фильтр Калмана не требует заданного формата входных данных, в отличие от быстрого преобразования Фурье.

Благодарности

Благодарим В. Темнова (Университет Эссена, Германия) за предоставленные экспериментальные интерферограммы.

Литература

1. В.Н. Васильев, И.П. Гуров. Компьютерная обработка сигналов в приложении к интерферометрическим системам. СПб: БХВ-Санкт-Петербург, 1998. 237 с.
2. I. Gurov, A. Zakharov. Dynamic nonlinear analysis of stochastic interference fields. / Presented at International Conference on Coherent and Nonlinear Optics (ICONO 2001) (Minsk, Belarus, 26 June - 1 July, 2001). To be published in Proc. SPIE.
3. Справочник по прикладной статистике /Под ред. Э. Ллойда, У. Лидермана. М.: Финансы и статистика, 1989.
4. M. Takeda, H. Ina, and S. Kobayashi. Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry // J. Opt. Soc. Am. V. 72. pp. 156-160, 1982.

ВОССТАНОВЛЕНИЕ ЛИНИЙ ЭКСТРЕМУМОВ СЛОЖНЫХ КАРТИН ИНТЕРФЕРЕНЦИОННЫХ ПОЛОС

М.В. Волков

Анализ картин интерференционных полос имеет важное значение для задач неразрушающего контроля объектов. Во многих случаях интерференционные полосы искажены влиянием шумов и локальных дефектов, вследствие чего затруднена оценка их параметров. Представленный метод локально-адаптивной нелинейной фильтрации, основанный на модификации локальных гистограмм, обеспечивает помехоустойчивую обработку интерферограмм с восстановлением линий интерференционных экстремумов. Проведенные исследования показали высокую эффективность метода применительно к обработке сложных картин интерференционных полос.

Введение

Исследования свойств поверхностей различных объектов имеют важное значение для многих областей науки и техники. При этом предпочтительны неразрушающие оптические методы, эффективное применение которых требует использования помехоустойчивых алгоритмов обработки картин интерференционных полос. Обработка интерференционных сигналов и картин интерференционных полос необходима в муаровой [1-3] и голографической [4,5] интерферометрии. Во многих случаях интерференционные полосы искажены влиянием шумов, локальных дефектов и могут иметь низкий контраст, вследствие чего затруднена оценка фазы полос, содержащей информацию об исследуемом объекте или процессе. Известные методы обработки интерференционных полос на основе преобразования Фурье [6,7] или аппроксимации интерференционных полос нелинейными полиномами [8,9] часто не обеспечивают необходимой помехоустойчивости и требуют значительных вычислительных ресурсов при анализе реальных интерференционных картин. Таким образом, проблема разработки новых методов обработки и анализа картин полос является весьма актуальной.

Обычно требуется по картине интерференционных полос восстановить фазу полос или линии интерференционных экстремумов, однако это не всегда возможно без предварительной обработки картины полос. Основной задачей такой обработки является устранение шумов и локальных дефектов без снижения контраста полос. В работе исследован метод нелинейной локально-адаптивной обработки сигналов и изображений, основанный на модификации локальных гистограмм значений яркости. Эффективность метода обеспечивается разработанным алгоритмом настройки конфигурации и размеров локальной области фильтрации в зависимости от топологии картины полос. При этом анализ локальных характеристик полос может осуществляться по дисперсиям или приведенным производным сигнала в различных направлениях.

Представленный метод локально-адаптивной нелинейной фильтрации имеет ряд преимуществ перед стандартными методами обработки интерференционных полос. Метод позволяет обеспечить улучшение качества и анализ характеристик весьма сложных интерферограмм, обработка которых при традиционных подходах представляется проблематичной.

Адаптивная настройка параметров обработки позволяет решить задачу построения непрерывных линий экстремумов интерференционных полос. При этом качество восстановленных линий экстремумов превосходит известные алгоритмы отслеживания полос.

Обработка одномерных сигналов на основе модификации гистограмм

Рассмотрим вначале задачу обработки одномерного сигнала $s(x)$, который можно рассматривать как сигнал в сечении двумерной интерференционной картины. Пусть реальный сигнал $\xi(x)$ определяется моделью

$$\xi(x) = s(x) + n(x), \quad (1)$$

где $n(x)$ - аддитивная помеха с нулевым средним, и может принимать дискретные значения (уровни) ξ_i , $i = 0, \dots, N$. Выделим в области независимой переменной некоторый отрезок $[a, b]$ и обозначим его среднюю точку x_c . Будем считать, что оценка идеального сигнала $s(x_c)$ может быть определена по значениям $\xi(x)$ в точках дискретизации x_k , $k = 1, \dots, K$, на выбранном отрезке. Для отрезка $[a, b]$ построим эмпирическую гистограмму $p(\xi_i)$, то есть оценку плотности вероятности, соответствующей закону распределения значений $\xi(x_k)$, $x_k \in [a, b]$, по дискретным уровням ξ_i . Значение гистограммы для некоторого уровня ξ_i равно количеству точек $x_k \in [a, b]$ в которых $\xi(x_k) = \xi_i$. Гистограмма $p(\xi_i)$ позволяет вычислить усредненную величину для выборки $\xi(x_k)$ на заданном отрезке, а именно:

$$\bar{s}(x_c) = \sum_{i=1}^N \xi_i p(\xi_i) / \sum_{i=1}^N p(\xi_i). \quad (2)$$

Сдвигая отрезок на один шаг дискретизации Δx , можно вычислить значение $\bar{s}(x_c + \Delta x)$ и т.д. Согласно (1), операция усреднения в (2) позволяет уменьшить влияние аддитивного шума.

Зададим на отрезке $[a, b]$ некоторую функцию $g(x)$, обеспечивающую уменьшение вклада сигнала $\xi(x)$ в гистограмму при удалении точки x от центра x_c отрезка фильтрации. Построим модифицированную гистограмму с учетом этой функции в форме

$$p(\xi_i) = \left[\sum_{x_k} g(x_k) \right]^\alpha, \quad (3)$$

где $x_k : \xi(x_k) = \xi_i$, $x_k \in [a, b]$ и $\alpha \in [1..3]$.

Таким образом, фильтрация одномерного сигнала методом модификации локальных гистограмм заключается в выполнении следующих операций (см. рис. 1):

- выбирается отрезок фильтрации $[a, b]$ и на нем строится гистограмма (3);
- вычисляется среднее значение $\bar{s}(x_c)$ по формуле (2), задающее значение $\bar{s}(x)$ в средней точке отрезка;
- отрезок $[a, b]$ сдвигается на шаг дискретизации Δx , после чего снова строится гистограмма и т.д.

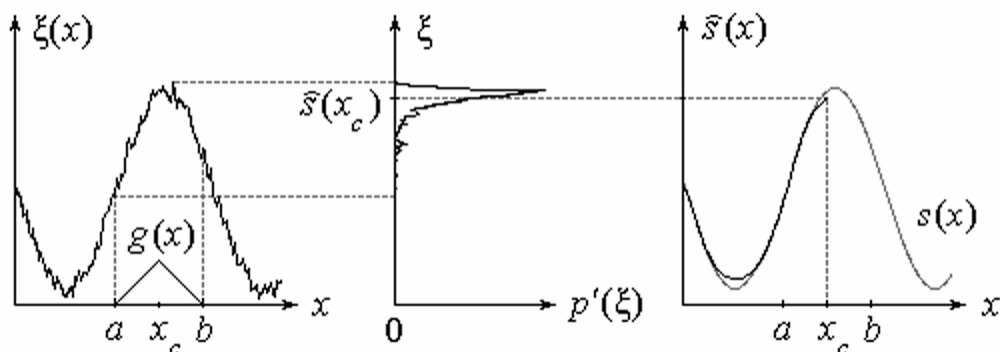


Рис. 1. Восстановление сигнала методом модификации гистограмм с весовой функцией

Особенностью метода (3) является сохранение амплитуды сигнала при итерационной обработке, когда оценка сигнала на n -ой итерации $\bar{s}^n(x)$ вычисляется

на основе данных предыдущей итерации $\hat{s}^{n-1}(x)$, причём $\hat{s}^0(x) = \xi(x)$. Модификация (3) гистограммы возведением ее в степень $\alpha > 1$ увеличивает статистический “вес” точек вблизи экстремумов, что приводит к незначительному изменению формы сигнала, и после нескольких шагов итерационной обработки появляются достаточно заполненные уровни. При этом предельно допустимые отклонения формы сигнала определяются видом весовой функции $g(x)$. В результате, начиная с некоторого шага итерации, полностью прекращается снижение амплитуды сигнала, и дальнейшая обработка приводит лишь к уменьшению влияния помех. После подавления помех изменений сигнала не происходит. Это свойство сходимости итерационного алгоритма очень важно при обработке сигналов, искаженных влиянием пространственно-нестационарных шумов и помех, которые характерны для реальных интерферограмм.

Важной особенностью метода является то, что для сигналов с близкой к нулю производной импульсные помехи в результате почти не оказывают влияния на выходной сигнал. Это объясняется тем, что “вес” каждого значения сигнала $\xi(x_k)$ при модификации гистограммы, согласно (3), зависит от количества и удаленности значений сигнала $\xi(x_k)$, лежащих на том же уровне ξ_i . Значения на более заполненных “стабильных” уровнях являются более устойчивыми, поэтому импульсные помехи оказывают на них очень слабое влияние.

Следует отметить, что увеличение локального периода сигнала приводит к значительному повышению эффективности фильтрации. Это свойство рассмотренного метода может быть использовано для повышения эффективности двумерной обработки картин полос, так как локальный период одномерного сигнала в сечении картины полос, проходящем через некоторую точку, зависит от направления.

Обработка картины полос

При переходе к обработке картин интерференционных полос необходимо обеспечить непрерывность по двум координатам. В простейшем случае значение оценки сигнала в некоторой точке определяется значениями сигнала во всех соседних точках некоторой двумерной области. При итерационной обработке возможно использование значений не во всех точках области, а лишь в некоторых сечениях, проходящих через точку, в которой определяется оценка сигнала. Такой подход позволяет значительно увеличить быстродействие метода, так как двумерная обработка может быть сведена к обработке одномерных сигналов.

Локальные периоды $T(i, j, l)$ одномерных сигналов в различных сечениях l картины полос, проходящих через некоторую точку (i, j) , различны. В случае, когда направление сечения l совпадает с локальным направлением полос для точки (i, j) , период $T(i, j, l)$ максимален, что позволяет с наибольшей эффективностью использовать свойства предложенного метода.

При обработке картины интерференционных полос метод нелинейной фильтрации (3) дает наилучшие результаты, если для каждой дискретной точки (i, j) направление обработки l совпадает с локальным направлением полос. В этом случае помехи в одномерном сигнале устраняются наиболее эффективно, и не происходит снижения амплитуды сигнала. Однако такая обработка приводит к потере непрерывности по направлениям и искажению обработанной картины полос.

Для выполнения условия непрерывности обработки необходимо для каждой точки (i, j) картины полос вычислить оценки $\hat{s}(i, j, l)$ одномерных сигналов во всех сечениях l , $l = 1, 2, \dots, L$, проходящих через обрабатываемую точку, а затем с помощью некоторых “коэффициентов значимости” $C(i, j, l)$ найти усредненное значение полученных оценок, а именно

$$\hat{s}(i, j) = \sum_l C(i, j, l) \hat{s}(i, j, l) / \sum_l C(i, j, l), \quad (4)$$

где $\hat{s}(i, j, l)$ – оценка одномерного сигнала в точке (i, j) в сечении вдоль направления l . Коэффициенты значимости имеют смысл “веса” оценки сигнала в сечении вдоль соответствующего направления. Эффективность нелинейного метода (3) зависит от локального периода сигнала $T(i, j, l)$, следовательно, и от направления обработки l . Таким образом, настройка коэффициентов значимости позволяет повысить эффективность фильтрации. Чем меньше отклонение направления сечения l от локального направления полос, тем больше должен быть соответствующий коэффициент.

Коэффициенты значимости можно оценить по модулям приращений одномерных сигналов вдоль соответствующих направлений. По каждому направлению l для всей картины полос строятся модули приращений $\Delta\xi(i, j, l) = \xi(x_k) = |\xi(x_{k+1}) - \xi(x_{k-1})|$, где x_{k-1} , x_k и x_{k+1} – соседние точки сигнала в сечении вдоль указанного направления l .

Полученные значения $\Delta\xi(i, j, l)$ отражают характер изменения производных картины полос. Зависимость $\Delta\xi(i, j, l)$ от направления l позволяет определить локальное направление полос. Однако, из-за влияния помех возможны значительные колебания параметра $\Delta\xi(i, j, l)$ для соседних точек картины полос, что ведет к погрешности определения локального направления полос. Кроме этого, значение $\Delta\xi(i, j, l)$ как оценка производной сигнала полос зависит от фазы, что может приводить к неправильной оценке направления полос. Так как для большинства реальных интерферограмм изменение направления полос в зависимости от изменения координат рассматриваемой точки достаточно мало, то простая двумерная фильтрация параметра $\Delta\xi(i, j, l)$, например, в форме усреднения внутри некоторой маски A , в общем случае не приводит к значительной потере точности оценки локального направления полос. Размер маски усреднения A необходимо выбирать в зависимости от топологии полос. Увеличение размера маски уменьшает влияние помех, но при этом происходит снижение чувствительности параметра к достаточно быстрым изменениям направления полос. В наиболее простом случае размер маски можно выбрать равным среднему значению шага полос.

В случае, если картина полос не содержит резких изломов, т.е. изменение направления полос мало, размер маски усреднения A может быть в десятки раз больше шага полос. В этом случае за счет использования данных от соседних полос удастся сохранить оценку правильного направления при значительных локальных дефектах, в том числе, при разрыве нескольких полос.

Полученные результаты усреднения значений $\Delta\xi(i, j, l^\perp)$, где $l^\perp$ – номер направления, перпендикулярного l , по некоторой маске A задают значение $C(i, j, l)$. Будем для простоты полагать, что помехи оказывают одинаковое влияние на коэффициенты $C(i, j, l)$, т.е. что влияние помех изотропное. При совпадении направления полос с направлением обработки l в некоторой точке (i, j) значение $C(i, j, l)$ максимально. С другой стороны, если направление сечения перпендикулярно к интерференционным полосам, значение $C(i, j, l)$ минимально.

Задача определения размеров отрезков фильтрации одномерных сигналов в различных направлениях для каждой точки картины полос может быть решена с использованием коэффициентов значимости $C(i, j, l)$.

Размеры отрезков фильтрации, очевидно, должны зависеть от шага интерференционных полос. Для эффективной фильтрации необходимо более широкие интерференционные полосы обрабатывать с большим отрезком фильтрации, более

узкие – с меньшим. Такая обработка позволяет сохранить контрастность полос малой ширины, то есть не приводит к потере информации о картине полос.

Для получения оценок локального шага полос во всех точках интерферограммы, необходимо вначале оценить значение шага полос хотя бы в одной точке. Простейшим случаем такой оценки является задание значения минимального шага полос как исходного глобального параметра интерферограммы. Усредненные по направлениям коэффициенты значимости

$$C(i, j) = \frac{1}{L} \sum_{l=1}^L C(i, j, l) \quad (5)$$

содержат информацию об изменении шага полос. По усредненным коэффициентам значимости можно судить об изменении шага полос в предположении о неизменности амплитуды и уровня помех для всех точек картины полос. Таким образом, с помощью данных коэффициентов можно получить оценки шага полос в каждой точке интерферограммы.

Относительные размеры отрезков фильтрации в различных сечениях для некоторой точки могут быть определены с помощью коэффициентов значимости $C(i, j, l)$. При обработке сигнала в направлении, соответствующему направлению полос, размер отрезка фильтрации может быть достаточно большим, это позволяет лучше подавлять помехи и не приводит к снижению амплитуды полос. В то же время, при обработке сигнала в сечении, перпендикулярном направлению полос, следует выбирать меньший отрезок фильтрации.

Таким образом, для определения размеров отрезков фильтрации по заданному минимальному шагу полос и соответствующему отрезку фильтрации для каждой точки картины полос через коэффициенты значимости находится локальный шаг полос, и по нему определяется средний размер отрезка фильтрации $K(i, j)$. После этого размеры отрезков фильтрации в точке (i, j) вычисляются с помощью коэффициентов значимости по формуле

$$K(i, j, l) = K(i, j) C(i, j, l) / C(i, j),$$

где $C(i, j, l)$ определяются из выражения (5).

Для выполнения одного шага итерации при обработке картины интерференционных полос, необходимо для каждой точки картины полос для всех направлений l определить размеры отрезков фильтрации $K(i, j, l)$, определить коэффициенты значимости, найти оценки $\hat{s}(i, j, l)$ одномерных сигналов и усреднить их по формуле (4).

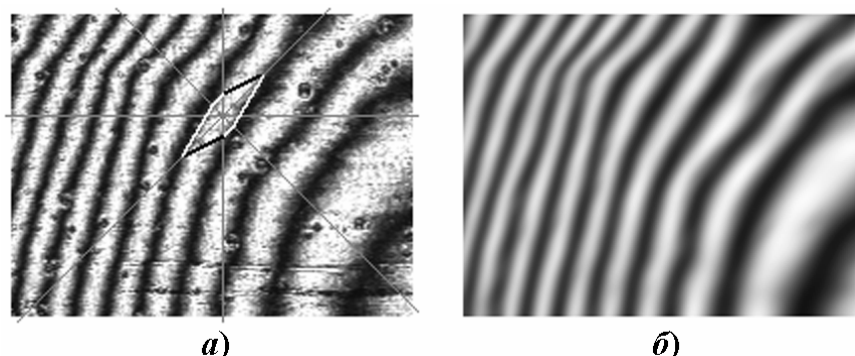


Рис. 2. Пример обработки реальной двумерной интерференционной картины итерационным методом нелинейной фильтрации на основе модификации локальных гистограмм

В качестве примера на рис. 2, а показана искаженная картина интерференционных полос форматом 200×150 точек, квантованная по 256-ти уровням. Для одной из точек

картины полос условно отмечена область фильтрации, ограничивающая размеры отрезков фильтрации, и четыре направления сечений, определяющих результат обработки в данной точке. В результате обработки после 5-ти итераций все импульсные помехи подавлены и контрастность полос уменьшилась незначительно, как это видно из рис. 2, б.

Восстановление линий экстремумов интерференционных полос

Рассмотрим более подробно задачу построения линий экстремумов для зашумленной интерферограммы. В процессе предварительной фильтрации рассмотренным выше методом в результате адаптивной настройки для каждой точки (i, j) становятся известны оценки локальных направлений полос (как направления, соответствующие максимальным коэффициентам значимости $C(i, j, l)$ для каждой точки (i, j)). При этом, для построения линий экстремумов требуется для каждой точки проверить, достигается ли в ней экстремум одномерного сигнала в сечении предварительно улучшенной картины полос, перпендикулярном локальному направлению полос. Задачу определения экстремума упрощает наличие оценки шага полос в каждой точке. Результатом такой обработки для картины полос является построение непрерывных линий экстремумов.

Таким образом, рассмотренный метод может применяться не только для предварительного подавления помех, но и для восстановления линий экстремумов. Существенным преимуществом метода при выделении экстремумов является глобальность обработки, т.е. тот факт, что решения о принадлежности каждой точки картины полос к линии экстремумов принимаются независимо. Такой подход имеет преимущество перед всевозможными алгоритмами отслеживания линий экстремумов, так как при глобальной обработке не происходит потери информации о достаточно коротких отрезках линий экстремумов, возникающих при разрывах интерференционных полос. Такие отрезки линий экстремумов могут содержать существенную информацию о картине полос, и по ним во многих случаях можно восстановить непрерывные линии с помощью алгоритмов сшивания. Качество восстановленных линий экстремумов позволяет эффективно применять различные логические алгоритмы. Это могут быть алгоритмы выравнивания толщины линий, удаления коротких отрезков ложных экстремумов, или алгоритмы сшивания линий.

На рис. 3, а показан результат восстановления экстремумов (минимумов) для картины полос, показанной на рис. 2. Результат обработки восстановленных линий минимумов алгоритмом выравнивания их толщины представлен на рис. 3, б.

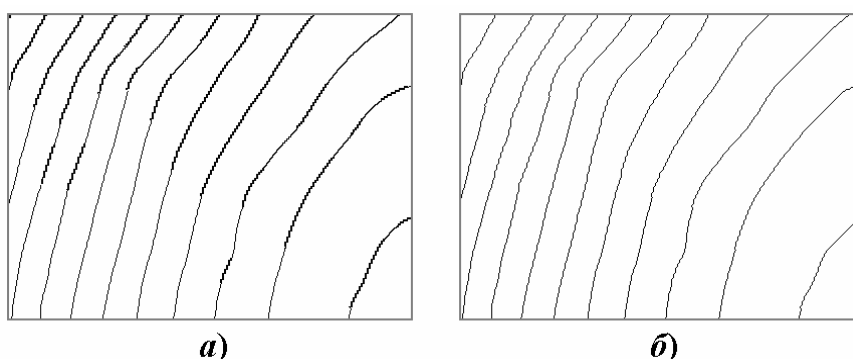


Рис. 3. Восстановленные линии минимумов для картины полос рис. 2: (а) исходная картина линий минимумов и (б) результат выравнивания толщины линий.

Примеры обработки реальных интерферограмм

Как уже отмечалось, обычно требуется по зашумленной картине интерференционных полос восстановить фазу полос или линии интерференционных экстремумов. Решение этих задач не всегда возможно без предварительной обработки (улучшения) картины полос. Предварительная обработка интерферограмм на основе адаптивного метода модификации локальных гистограмм позволяет значительно упростить задачу построения линий экстремумов. Рассмотренный метод может применяться не только для подавления помех, но и непосредственно для восстановления линий экстремумов.

Далее приведены примеры, иллюстрирующие этапы обработки картины интерференционных полос. Полученные результаты позволяют сделать вывод о высокой эффективности рассмотренного метода как для задачи предварительного улучшения зашумленных картин полос, так и для задачи восстановления линий экстремумов. Картина полос анализировалась по восьми направлениям $L=8$, настройка параметров осуществлялась по вычисленным модулям приращений сигналов (см. выше).

На рис. 4 показана искаженная картина полос форматом 512×400 точек, квантованная по 256-ти уровням. В результате обработки после 5-ти итераций все импульсные помехи подавлены, видность полос практически не уменьшилась, как это видно из рис. 5.

На рис. 6 приведен результат восстановления линий минимумов для картины полос рис. 5. Этот рисунок иллюстрирует сложность глобального восстановления линий экстремумов без учета информации о направлении полос.

Адаптивная настройка при предварительной фильтрации позволяет определить направление полос, что дает возможность исключить ложные линии минимумов, как это видно из рис. 7. Линии минимумов на этом рисунке имеют различную толщину, присутствуют короткие отрезки ложных линий, однако качество восстановленных линий минимумов позволяет применить к ним алгоритм выравнивания, устраняющий эти недостатки. Результат обработки линий минимумов рис. 7 показан на рис. 8.

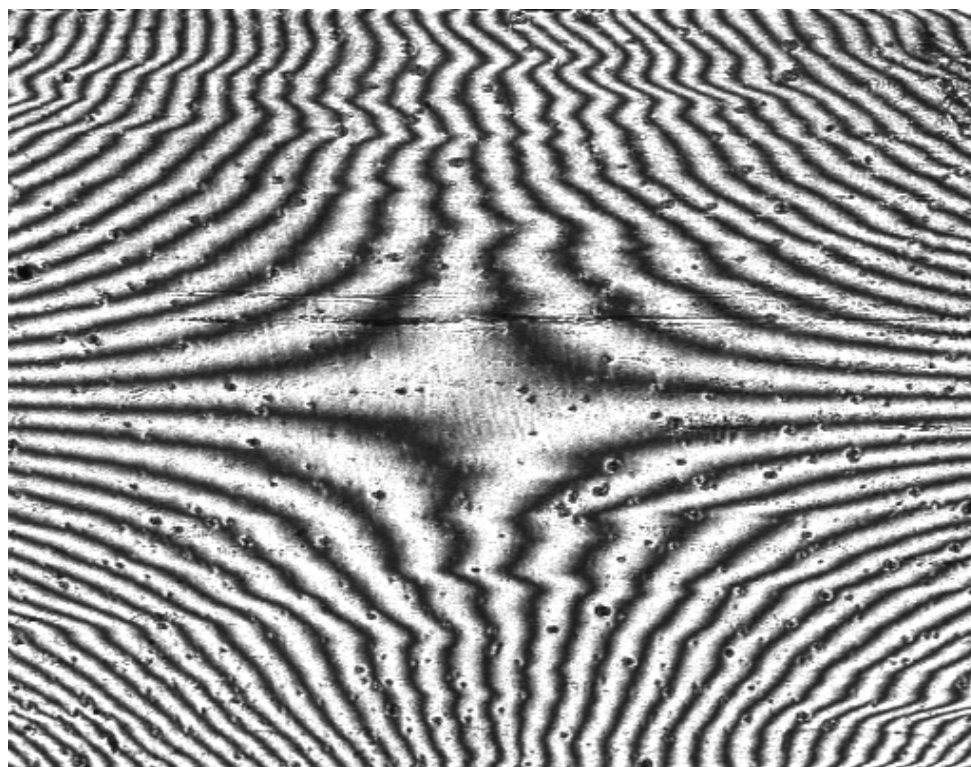


Рис. 4. Исходная интерферограмма

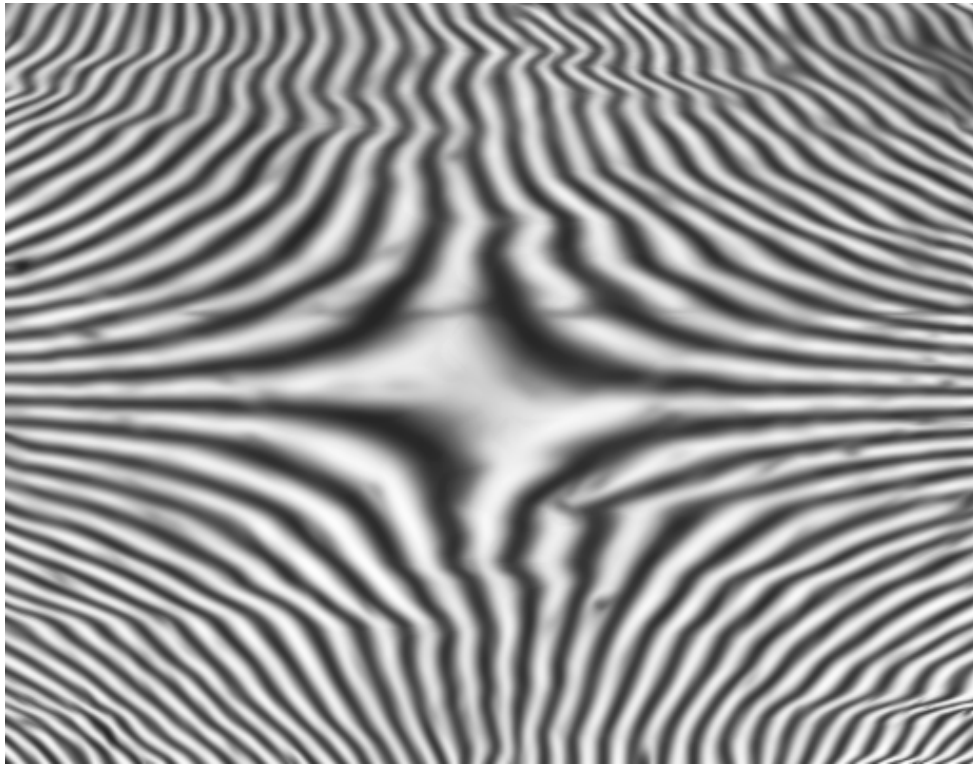


Рис. 5. Пример обработки интерферограммы итерационным адаптивным методом нелинейной фильтрации на основе модификации локальных гистограмм

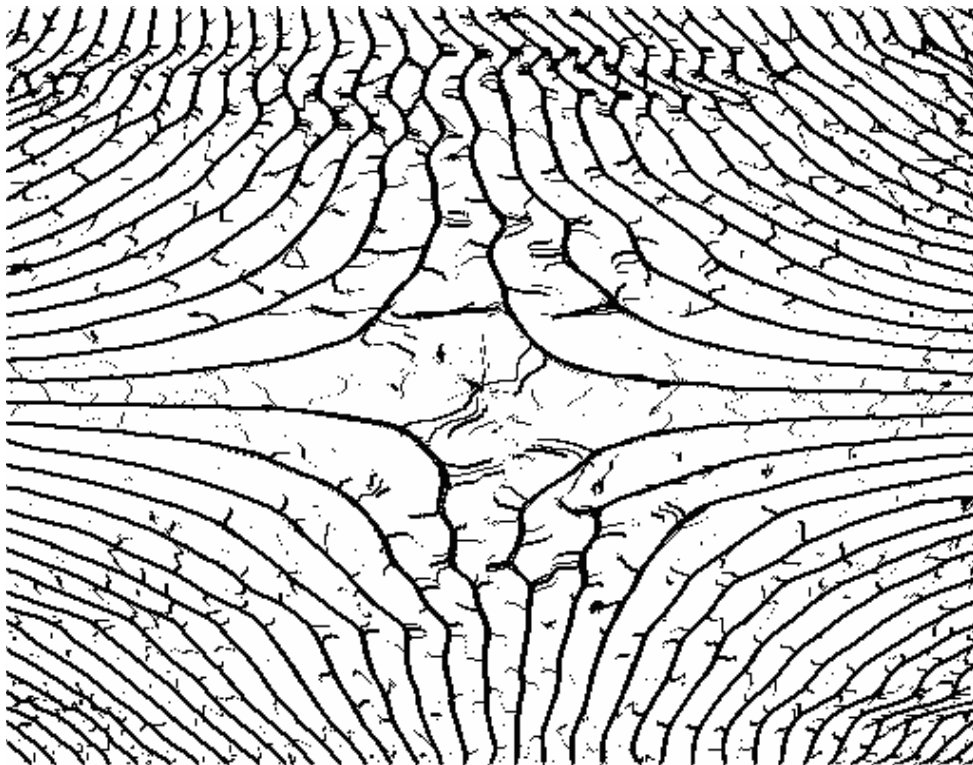


Рис. 6. Восстановление линий минимумов для предварительно обработанной картины полос (рис. 5) без учета локального направления полос

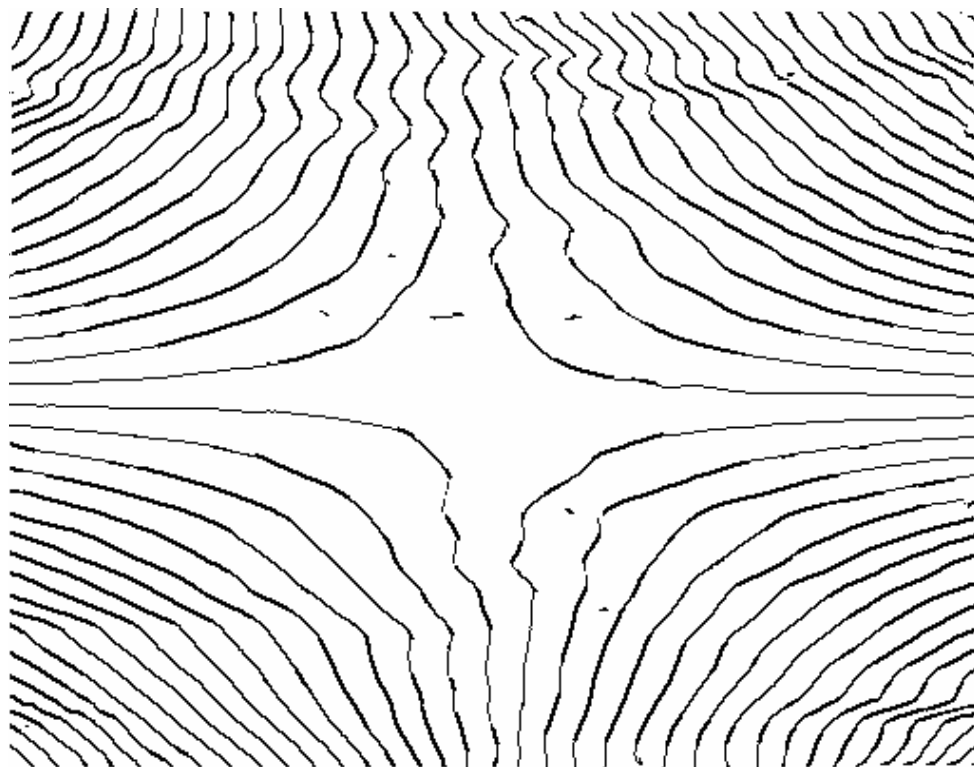


Рис. 7. Восстановление линий минимумов для предварительно обработанной картины полос (рис. 5) с учетом оценок локального направления полос

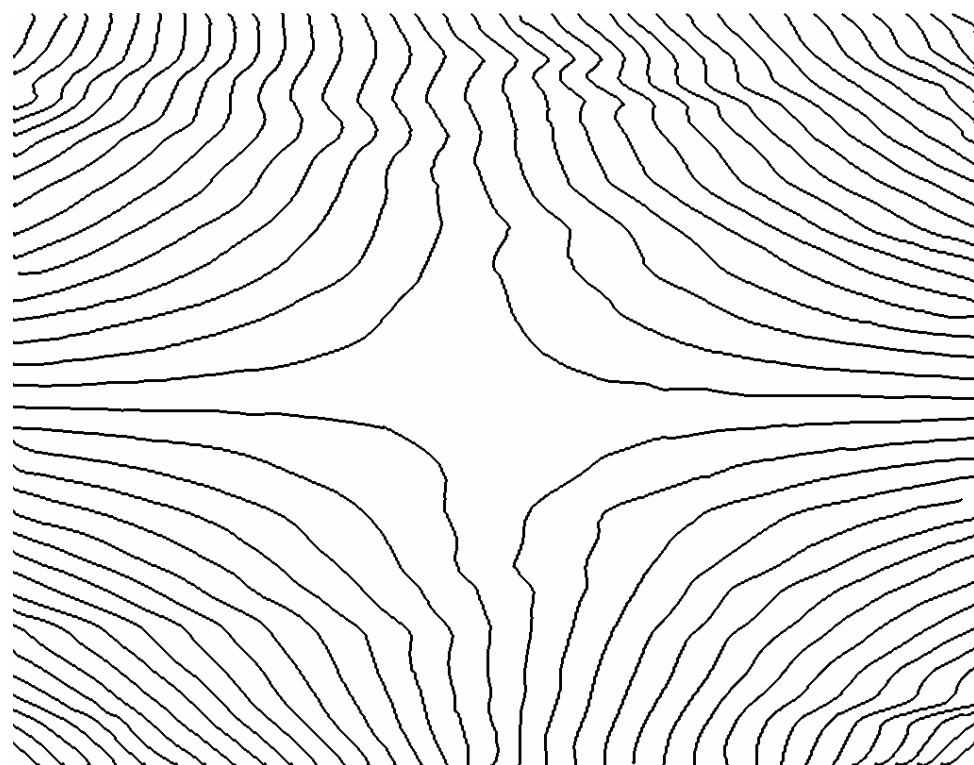


Рис. 8. Результат восстановления линий минимумов

Заключение

Проведенные исследования показали высокую эффективность рассмотренного метода применительно к задачам улучшения и выделения линий экстремумов сильно

зашумленных сложных картин интерференционных полос, обработка которых при традиционных подходах представляется проблематичной. Метод позволяет практически без предварительной информации о параметрах интерференционных полос анализировать топологию полос и автоматически настраивать параметры фильтрации. При этом используется общее свойство полос, заключающееся в том, что они имеют некоторое преимущественное направление в каждой точке картины полос. Адаптивная настройка параметров обработки позволяет устранять значительные дефекты интерферограмм, такие как разрывы полос. При необходимости приведенные результаты могут быть улучшены за счет применения более помехоустойчивых методов адаптивной настройки и увеличения числа направлений обработки.

Литература

1. Post D. *Moiré Interferometry in SEM Handbook on Experimental Mechanics*, A.S. Kobayashi .- New York, 1987.
2. Poon C.Y., Kujawinska M., Ruiz C. Spatial-carrier phase shifting method of fringe analysis for moiré interferometry // *J. Strain Analysis*. 1993. V. 28. P.79-88.
3. De Nicola S., Ferraro P., Gurov I., Koviazin R., Volkov M. Fringe analysis for moiré interferometry by modification of the local intensity histogram and use of a two-dimensional Fourier transform method // *Meas. Sci. Technol*. 2000. V. 11, № 9. P.1328-1334.
4. Jones R., Wykes C. *Holographic and Speckle Interferometry*, Cambridge University Press, Cambridge, 1983.
5. Schnars U., Jüptner W.P.O. Digital recording and reconstruction of holograms in hologram interferometry and shearography // *Appl. Opt*. 1994. V. 33. P. 4373-4377.
6. Takeda M., Ina H., Kobayashi S. Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry // *J. Opt. Soc. Am*. 1982. V. 72. P. 156-160.
7. Kreis T. Digital holographic interference-phase measurement using the Fourier-transform method // *J.Opt.Soc.Am*. 1986. V. 3. P. 847-855.
8. Schemm J.B., Vest C.M. Fringe pattern recognition and interpretation using nonlinear regression analysis // *Appl. Opt*. 1983. V. 22. P. 2850-2853.
9. Cordero-Davila A., Cornejo-Rodrigues A., Cardona-Nunez O. Polynomial fitting of interferograms with Gaussian errors of fringe coordinates. I: Computer simulations // *Appl. Opt*. 1994. V. 33. P. 7339-7342.

ДИНАМИЧЕСКОЕ ОЦЕНИВАНИЕ ЛОКАЛЬНОГО НАПРАВЛЕНИЯ ПОЛОС В НИЗКОКОНТРАСТНЫХ ИНТЕРФЕРОГРАММАХ

Ю.Р. Вознюк

Рассмотрен метод восстановления отклонений формы неровных поверхностей, основанный на обработке картины проецированных интерференционных полос. Обработка включает процедуру улучшения интерференционной картины путем удаления фона, коррекции огибающей и подавления шума. Реализован метод отслеживания линий экстремумов картин полос с использованием стохастического метода предсказания и апостериорной коррекции при поиске локальных направлений полос. Результаты, полученные при обработке данных от реальных объектов, демонстрируют возможность использования разработанного метода в научных исследованиях и промышленности.

Введение

Восстановление карты рельефа шероховатых поверхностей требуется при проведении научных исследований и контроле качества в процессе производства многих видов изделий [1]. При этом бесконтактные оптические методы представляют наибольший интерес. Один из универсальных методов основан на использовании метода проекции интерференционных полос (см., например [2]), который обеспечивает возможность изменения чувствительности в зависимости от выбранного шага полос. Информация о форме поверхности содержится в отклонениях фазы полос интерферограммы, другими словами, в локальном смещении интерференционных полос.

В настоящее время широко применяются известные методы восстановления фазы полос, основанные на преобразовании Фурье [2, 3] и нелинейной аппроксимации интенсивности интерференционных полос [4, 5]. При контроле реальных объектов полосы интерференционных картин часто бывают искажены неравномерным фоном, неравномерной огибающей и влиянием шума. Поэтому сложно обрабатывать картины полос в автоматическом режиме.

В статье представлен метод обработки картин полос, который позволяет получать карту линий экстремумов и восстанавливать форму неровных поверхностей. Процедура улучшения изображения основана на сглаживании и нормировке интенсивности полос с последующим использованием стохастического метода предсказания и апостериорной коррекции при поиске локальных направлений полос улучшенного изображения.

Теоретические основы метода

Проведены исследования и классификация видов искажений картин полос. Типичными искажениями являются малая видность низкоконтрастных полос, влияние шума и разрывы полос. Процедура улучшения изображения обычно включает в себя устранение фоновой составляющей методом низкочастотной фильтрации и вычитания отфильтрованного компонента, подавление случайных шумов и коррекцию огибающей путем локальной нормировки амплитуды полос. Примеры фрагментов искаженных картин полос и улучшенных картин полос на основе описанной выше процедуры, показаны на рис. 1.

Для эффективного устранения искажений необходимо априори иметь информацию о локальных параметрах интерференционной картины. Информацию об амплитуде полос можно получить из статистик локальной интенсивности полос. При этом метод улучшения интерферограммы может быть основан на модификации локальных полутоновых гистограмм, который был успешно применен к обработке муаровых полос [6]. Дополнительно были реализованы процедуры определения краев и

сегментации интерферограмм с использованием регуляризованных дифференциальных операторов [7]. Пример обработанного фрагмента картины полос показан на рис. 2, б.

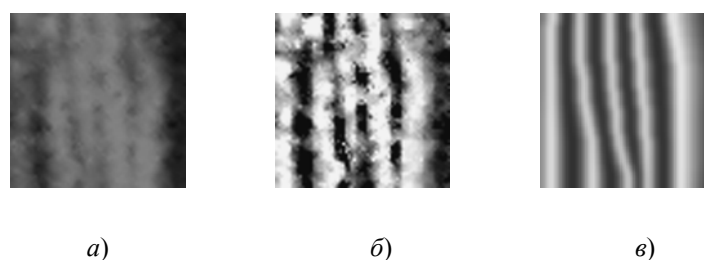


Рис. 1. Первоначальный низкоконтрастный участок картины полос (а), результат, полученный после нормировки амплитуды без сглаживания (б) и результат нормировки при предварительном сглаживании (в)

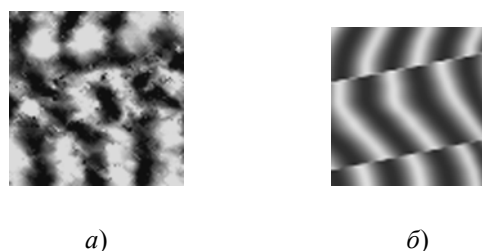


Рис. 2. Первоначальное искаженное изображение картины полос (а) и улучшенный центральный участок (б)

Улучшенное изображение картины полос пригодно для применения метода отслеживания линий экстремумов. Первым шагом является распознавание интерференционных полос внутри поля изображения при анализе данных вдоль произвольно выбранного сечения изображения. Простейшим подходом является выбор порогового значения интенсивности, что позволяет получить предварительную информацию о положении максимумов (минимумов) интенсивности полос. На следующем шаге приблизительно распознаются локальные направления полос с помощью дифференциальных операторов. На третьем шаге выполняется отслеживание линий экстремумов.

Алгоритм отслеживания полос основан на использовании стохастического метода предсказания и апостериорной коррекции при поиске локальных направлений полос и иллюстрируется схемой, показанной на рис. 3.

Анализируются значения интенсивности в окрестности точки O , принадлежащей некоторому начальному сечению изображения. Для каждой точки являются известными предварительно оцененные локальные направления интерференционных полос (оценка направления обозначена на рис. 3 как $\hat{\alpha}_{k-1}$). Для нахождения уточненных локальных направлений полос для следующей k -ой точки дискретизации изображения вычисляются средние значения производных в нескольких фиксированных направлениях. Поправка локального направления на k -ом шаге вычисляется на основе вычисления разности между предполагаемым $\alpha_{k/k-1}$ и расчетным α_k направлениями (углами) со взвешивающим коэффициентом $\beta < 1$, определяемым априорной информацией о допустимых изменениях направления для соседних точек, а именно

$$\hat{\alpha}_k = \alpha_{k/k-1} - \beta (\alpha_k - \alpha_{k/k-1}), \quad (1)$$

где предполагаемое значение $\alpha_{k/k-1}$ может быть определено в простейшем случае как $\alpha_{k/k-1} = \gamma \hat{\alpha}_{k-1}$, $\gamma \leq 1$.

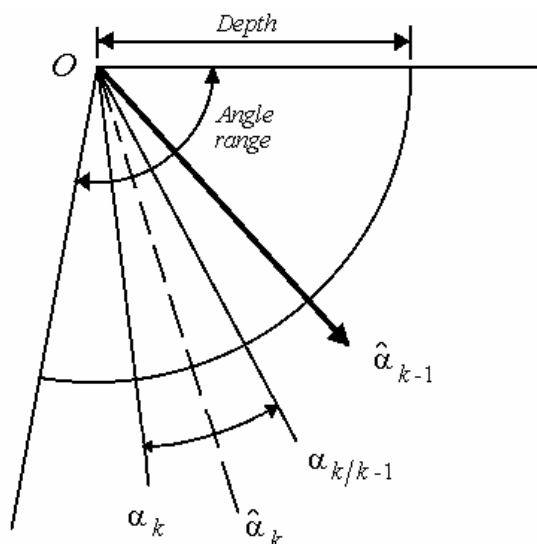


Рис. 3. К алгоритму отслеживания линий экстремумов

Как известно, выражение (1) можно интерпретировать как стохастическое разностное уравнение, которое описывает дискретный стохастический процесс “случайных блужданий” с “окрашенным” спектром мощности $G(u)$. В нашем случае такой подход применяется к локальным изменениям *направления* полос. Частота среза спектра Δu зависит от коэффициента β , как это показано на рис. 4.

Уменьшая величину коэффициента β , можно повысить помехоустойчивость процесса отслеживания линий экстремумов за счет ограничения частоты среза допустимых угловых изменений локального направления полос. Этот коэффициент в простейшем случае вычисляется в форме

$$\beta = A(\sigma_\alpha^2 / \sigma_n^2), \quad (2)$$

где σ_α^2 и σ_n^2 – дисперсия “полезных” угловых отклонений и дисперсия шума (ложных отклонений) соответственно и A является константой (в нашем случае $A=1$).

Выражения (1) и (2) соответствуют процессу дискретной фильтрации Калмана первого порядка [8]. В частности, фильтрация Калмана может быть использована для подавления шума наблюдений в сечении картины полос, как это показано на рис. 5, где $s(k)$ – последовательность дискретных отсчетов сигнала, пропорционального интенсивности полос.

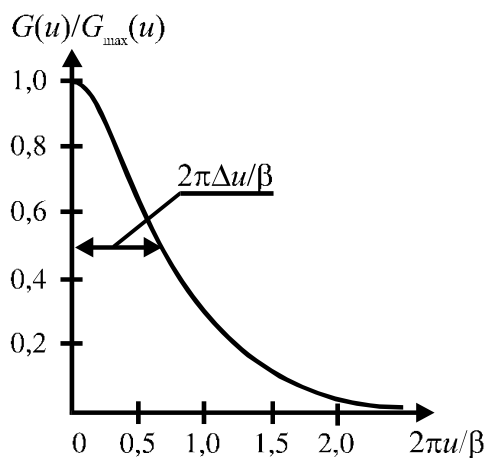


Рис. 4. Спектр мощности “окрашенного” шума

Рассмотренный вид фильтрации использовался для динамической оценки локальных направлений полос. Предполагаемое значение $\alpha_{k/k-1}$ в выражении (1) и дисперсия σ_n^2 в выражении (2) оценивались внутри локального двумерного участка (см. рис. 3). Неоднозначность направлений определялась исходя из разброса данных для всех направлений внутри диапазона угловых отклонений. Это означает, что дискретная стохастическая фильтрация используется как алгоритм уточнения данных о локальном направлении полос после предварительного улучшения изображения и линейной фильтрации на участке, соответствующем установленному диапазону угловых отклонений.

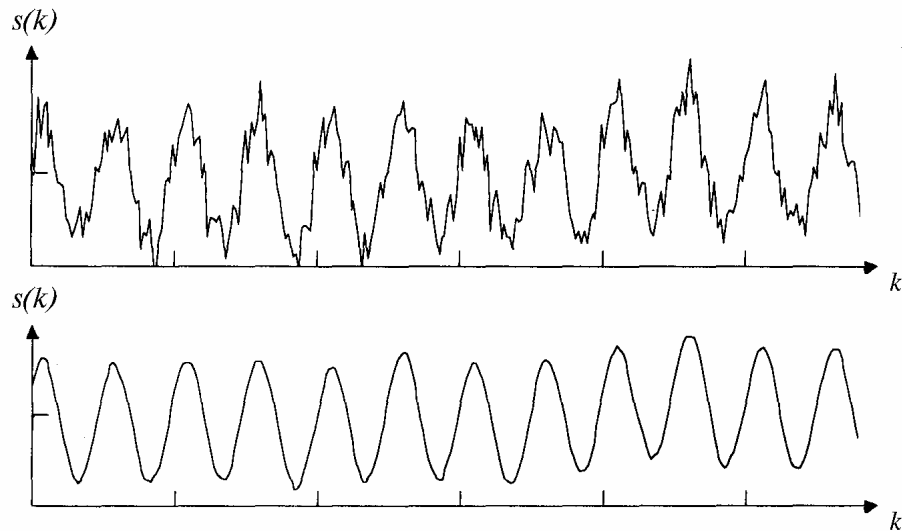


Рис. 5. Фильтрация Калмана зашумленного интерференционного сигнала по интенсивности [9]

После того, как найдены локальные направления полос для всех точек, принадлежащих данному сечению изображения, проверяются однородность (равномерность) направлений и расстояние между соседними полосами. Локальное направление n -ой полосы окончательно корректируется, принимая во внимание направления соседних полос, а именно

$$\alpha_{nk} = [\alpha_{(n-1)k} + \alpha_{(n+1)k}] / 2, \quad (3)$$

$$\alpha'_k = \alpha_k - \beta'(\alpha_k - \alpha_{nk}). \quad (4)$$

Если интерференционная полоса в некоторых точках имеет разрывы, другими словами, отсутствуют локальные данные, положение этой точки определяется как взвешенное положение соседних полос. Это позволяет соединять поврежденные интерференционные полосы. Описанные выше операции применяются ко всем точкам изображения с восстановлением линий экстремумов интерференционных полос, что позволяет получить данные, необходимые для восстановления формы поверхности.

Если картина полос априори содержит несколько систем интерференционных полос как, например, на рис. 2, локальные частные производные полос и направления полос содержат достаточно информации для распознавания границ интерференционных полос.

Экспериментальные результаты

Предложенный метод был использован для восстановления линий экстремумов картины полос с исходным шагом полос, равным 100 мкм. Пример первоначально наблюдаемой картины представлен на рис. 6. Был использован металлический образец,

поверхность которого была деформирована клеймом в виде цифры “4”. На рис. 6 видно, что первоначальное изображение содержит неоднородный фон и существенно изменяющуюся огибающую полос. Картина сильно искажена влиянием шума и низкой видностью полос в нескольких очень темных или светлых участках. При обработке подобных изображений с применением одного из широко распространенных алгоритмов не было получено стабильных результатов. Очевидно, что для нахождения линий экстремумов необходимо произвести операции по улучшению изображения и восстановлению потерянных данных.

В результате применения комбинированного метода, рассмотренного выше, получены верные оценки линий экстремумов полос в автоматическом режиме. Результаты обработки в виде картины линий максимумов и минимумов интенсивности представлены на рис. 7.

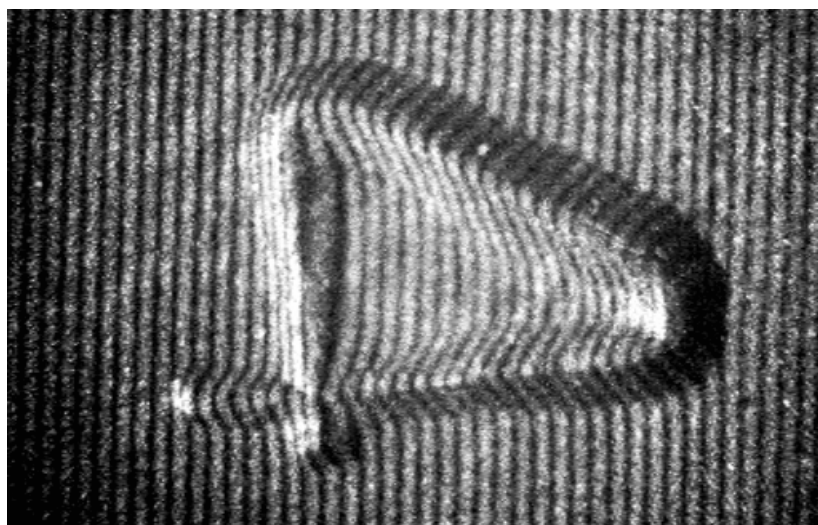


Рис. 6. Первоначальное изображение, полученное методом проекции полос

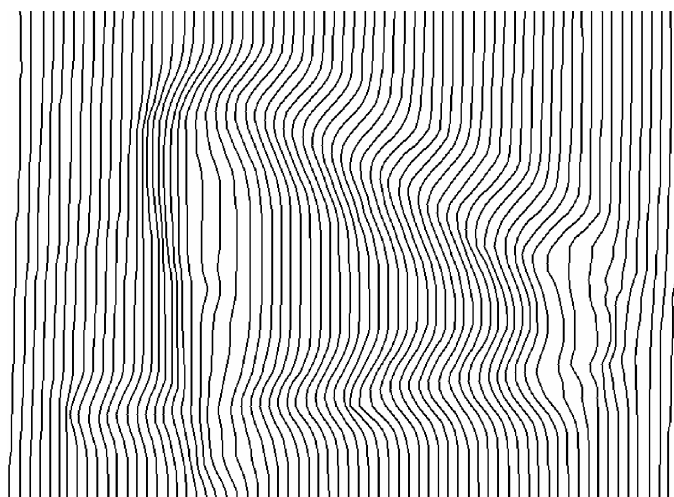


Рис. 7. Картина линий экстремумов полос, полученная при обработке изображения (рис. 6)

Очевидно, что карта линий экстремумов содержит информацию, требуемую для восстановления формы поверхности. Чувствительность метода при контроле отклонений формы определяется шагом полос и для представленного примера составляет около 30 мкм.

Заключение

Предложенный комбинированный метод улучшения изображения и отслеживания линий экстремумов полос реализует основные операции обработки, включая двумерную дискретную фильтрацию, анализ структуры изображения и т. д., что позволяет обрабатывать искаженные интерференционные изображения с высокой помехоустойчивостью. Данный метод может быть использован для бесконтактного контроля отклонений формы поверхностей в автоматическом режиме при проведении научных исследований и в промышленных целях.

Рассмотренный метод обработки изображений и отслеживания линий экстремумов базируется на теоретически обоснованных подходах, реализованных в новой форме. Например, метод фильтрации Калмана (являющийся простейшим случаем общей методологии стохастической фильтрации, подробно рассмотренной в [9, 10]) использован не только для подавления шума наблюдений по интенсивности, но также и для нахождения локальных угловых направлений полос при двумерном предсказании и апостериорной коррекции случайных отклонений их углового направления.

Одно из наиболее важных свойств предложенного метода состоит в его оптимизации с точки зрения быстродействия при обработке картин полос. Основные операции, такие как дифференцирование и рекуррентная фильтрация, являются эффективными при компьютерной реализации.

Чувствительность метода легко варьируется за счет подходящего выбора шага проецируемых интерференционных полос, что позволяет использовать разработанный метод и компьютерные алгоритмы для обработки картин полос не только при контроле малых объектов, но и для объектов с широким диапазоном отклонений геометрических характеристик.

Литература

1. Optical methods in wide scale shape measurements // Automatic Processing of Fringe Patterns, Proc. 3rd Int. Workshop Fringe'97 (Bremen, Germany, Sept. 15-17, 1997), W. Juptner and W. Osten, eds. - Berlin: Akademie Verlag, 1997. P. 147-249.
2. Takeda M. and Mutoh K. Fourier transform profilometry for the automatic measurement of 3-D object shapes // Appl. Opt. 1983. V.22. P. 3977-3982.
3. Takeda M., Ina H. and Kobayashi S. Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry // J. Opt. Soc. Am. 1982. V.72. P. 156-160.
4. Schemm J.B. and Vest C.M. Fringe pattern recognition and interpretation using nonlinear regression analysis // Appl. Opt. 1983. V.22. P. 2850-2853.
5. Cordero-Davila A., Cornejo-Rodrigues A. and Cardona-Nunez O. Polynomial fitting of interferograms with Gaussian errors of fringe coordinates. I: Computer simulations // Appl. Opt. 1994. V.33. P. 7339-7342.
6. De Nicola S., Ferraro P., Gurov I., Koviazin R. and Volkov M. Fringe analysis for moiré interferometry by modification of the local intensity histogram and use of a two-dimensional Fourier transform method // Meas. Sci. Technol. 2000. V.11. P. 1328-1334.
7. Хорн Б.К.П. Зрение роботов. М.: Мир, 1989.
8. Handbook of Applicable Mathematics V.6 "Statistics", E. Lloyd and W. Ledermann eds. Ch. 20. Kalman filtering. Wiley-Interscience Publ., 1984.
9. Васильев В.Н., Гуров И.П. Компьютерная обработка сигналов в приложении к интерферометрическим системам. СПб: БХВ-Санкт-Петербург, 1998.
10. Gurov I. and Sheynihovich D. Interferometric data analysis based on Markov non-linear filtering methodology // J. Opt. Soc. Amer. A. 2000. V.17. P. 21-27.

УЛУЧШЕНИЕ КАЧЕСТВА ИЗОБРАЖЕНИЙ МЕТОДОМ ВАН ЦИТТЕРТА

И.П. Гуров, Д.С. Смирнов

Исследованы свойства алгоритма Ван Циттерта в зависимости от выбора вида импульсного отклика системы, формирующей изображение. Введена пространственно-зависимая функция отклика, определяемая локальной изменчивостью изображения. Предложенная методика реализована применительно к картинам интерференционных полос. Показано преимущество использования функции отклика косинусоидальной формы для повышения контраста интерференционной картины.

Введение

Улучшение качества изображений является актуальной проблемой при исследовании многих физических явлений, в частности, при обработке когерентных изображений. Одной из важных задач является повышение контрастности картины интерференционных полос. Снижение контраста объясняется наличием фоновой составляющей интенсивности, изменяющейся в пределах изображения, и влиянием шума. В предположении, что уменьшение полезной составляющей вызвано "размытием" полос и изменением функции импульсного отклика системы, формирующей изображение, оказывается возможным повышение амплитуды полос при использовании метода Ван Циттерта. Суть метода заключается в реализации итерационного алгоритма противосвертки искаженного изображения с импульсным откликом искажающей системы. Противосвертка является устойчивой операцией при восстановлении истинного изображения из искаженного, но не зашумленного изображения. В условиях воздействия шума возможно возрастание влияния шума и появление ложных деталей изображения.

В данной работе исследовалось влияние формы импульсного отклика системы на процесс улучшения изображений. Показано, что из наиболее часто используемых видов функций импульсного отклика оптимальным для восстановления контраста интерференционных полос является импульсный отклик косинусоидальной формы. Недостаток алгоритма Ван Циттерта, связанный с усилением шумовых составляющих при противосвертке, устранялся использованием данных о локальной изменчивости картины полос и соответствующей модификацией функции импульсного отклика.

Алгоритм Ван Циттерта

Рассмотрим вначале одномерную задачу получения оценки истинного сигнала $s(x)$ из экспериментально зарегистрированного сигнала $f(x)$, который сформирован при влиянии импульсного отклика $h(x)$ системы на исходный сигнал в форме свертки

$$\int_{-\infty}^{+\infty} h(x-x')s(x')dx' = f(x).$$

Влияние импульсного отклика удобно рассматривать в частотной области, используя преобразование Фурье

$$H(\omega) = \int_{-\infty}^{+\infty} h(x)e^{-i\omega x} dx.$$

Алгоритм Ван Циттерта выражается в форме [1, 2]

$$s_0 = f(x), \quad (1a)$$

$$s_{n+1}(x) = s_n(x) + \alpha [f(x) - h(x) \otimes s_n(x)], \quad (1б)$$

где $n = 1, 2, \dots$ – номер итерации, $\otimes$ – знак свертки, или в частотной области

$$S_n(\omega) = [F(\omega)/H(\omega)] \left\{ 1 - [1 - \alpha H(\omega)]^n [1 - H(\omega)] \right\}, \quad (2)$$

где $|H(\omega)| > 0$, $F(\omega)$ – Фурье-образ функции $f(x)$. Из (2) следует, что для сходимости алгоритма (1) необходимо, чтобы на всех частотах выполнялось условие $|1 - \alpha H(\omega)| < 1$, или $0 < \alpha < 2 / \max H(\omega)$.

Продemonстрируем свойства алгоритма Ван Циттерта на конкретном примере. Важной оптической задачей является восстановление границы объекта, при переходе через которую скачком изменяется интенсивность s . При этом в качестве истинного изображения $s(x)$ возьмем элементарную ступенчатую функцию $s(x) = \eta(x)$, где $\eta(x) = 1$ при $x \geq 0$ и $\eta(x) = 0$ при $x < 0$ (см. рис. 1, пунктирная линия).

Предположим, что ступенька размыта импульсным откликом гауссовой формы, т.е.

$$h(x) = (1/\sqrt{\pi}L) \exp(-x^2/L^2). \quad (3)$$

Можно показать, что форма размытой ступеньки определяется в виде

$$f(x) = h(x) \otimes s(x) = [1 + \operatorname{erf}(x/L)]/2, \quad (4)$$

где $\operatorname{erf}(x) = (2/\sqrt{\pi}) \int_0^x e^{-x^2} dx$ – интеграл вероятностей, а ее характерный параметр l_0

определяется величиной размытия L :

$$l_0 = \left(\left. \frac{df(x)}{dx} \right|_{x=0} \right)^{-1} = \sqrt{\pi}L. \quad (5)$$

При достаточно большом числе итераций n параметр l_n уменьшается согласно соотношению

$$l_n = \left(\left. \frac{ds_n(x)}{dx} \right|_{x=0} \right)^{-1} = \frac{\pi}{2\sqrt{\ln(\alpha n)}} L. \quad (6)$$

На рис. 1 приведена кривая, построенная согласно (4), и результат вычислений $s_{10}(x)$ после 10-ой итерации. Видно, что в результате десяти итераций ширина размытой ступеньки уменьшилась примерно вдвое.

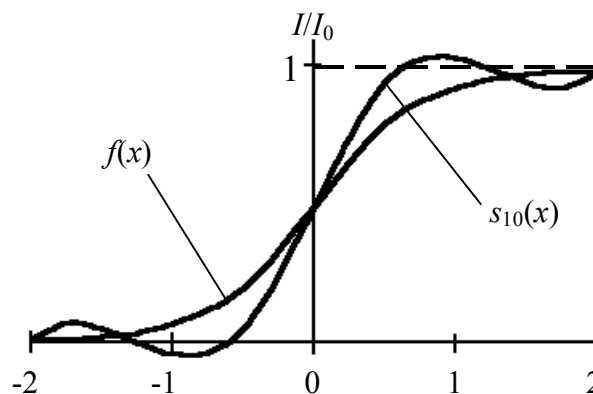


Рис. 1. Итерационное восстановление резкого перепада яркости

Обработка двумерных изображений

Из рассмотренного примера видно, что алгоритм Ван Циттерта осуществляет противосвертку, т.е. реконструкцию исходной функции из "размытой", при известной "искажающей" функции $h(x)$. Таким образом, в общем случае этот алгоритм позволяет выделять детали изображения.

Указанное свойство можно использовать для обработки изображений в виде интерференционных полос при их интерпретации как результата свертки импульсного отклика $h(x)$ косинусоидальной формы с бинарным изображением, определяющим положение полос, например, с линиями экстремумов полос. При этом алгоритм Ван Циттерта в результате позволяет "подчеркнуть" области экстремумов, т.е. повысить контрастность картины полос. Следует отметить, что импульсный отклик в форме полупериода косинусоидальной функции удовлетворяет условиям сходимости алгоритма Ван Циттерта [1, 2].

Из выражения (2) следует, что использование алгоритма Ван Циттерта приводит к изменению спектра изображения, в частности, к усилению высокочастотных составляющих спектра. При наличии шумовой составляющей в изображении это может приводить к усилению влияния шума и возникновению ложных деталей. Следовательно, требуется регуляризация алгоритма, степень которой, с одной стороны, определяется выбором коэффициента α в (1), и, с другой стороны, обеспечивается фильтрацией полезной составляющей перед применением алгоритма и после каждого шага итерации.

Применительно к обработке картин интерференционных полос, следует использовать априорную информацию о характере изображений этого вида. В частности, обычно полосы имеют некоторое преимущественное направление в каждой локальной точке. Кроме этого, необходимо учитывать возможные изменения локального шага полос. Следовательно, при обработке изображений в виде картины полос следует строить двумерную функцию импульсного отклика $h(x,y)$, зависящую от локальных параметров полос.

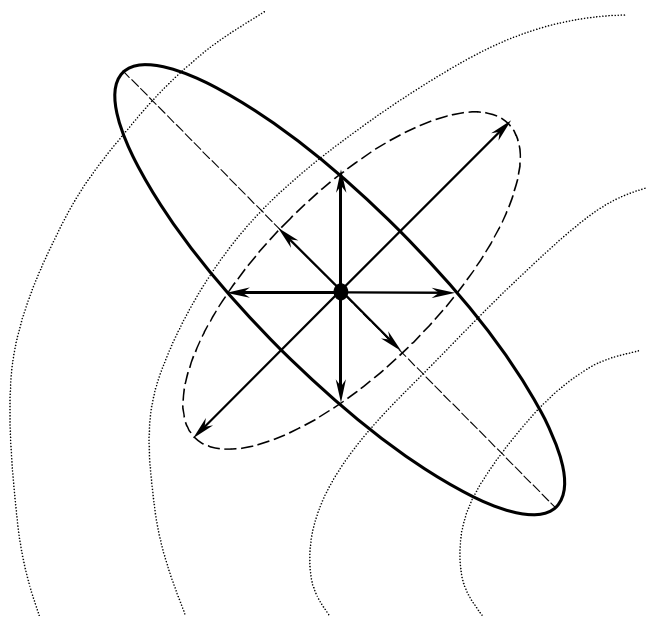


Рис. 2. Модификация области формирования двумерной функции отклика в зависимости от направления полос

Перечисленные задачи решены с помощью модификации функции отклика системы в зависимости от направления полос. Информацию о направлении полос можно получить при дифференцировании изображения по восьми направлениям и вычислении модуля от полученных производных. Отфильтровав эти производные для удаления из них шума, можно получить информацию об "изменчивости" интерференционной картины, а именно, об ориентации и шаге полос [3]. С учетом полученной информации можно локально модифицировать двумерную функцию отклика (см. рис. 2).

На рис. 2 сплошной линией показана область, внутри которой строится оценка "искажающего" импульсного отклика $h(x,y)$ в виде колоколообразной функции с косинусоидальной образующей. Размеры области и ее ориентация определяются соответственно локальным шагом и ориентацией полос. Пунктирной линией выделена область, внутри которой осуществляется сглаживание данных для снижения влияния шума.

На рис. 3 представлен пример обработки картины полос с помощью алгоритма Ван Циттерта. Непосредственное применение этого алгоритма к исходной зашумленной картине (см. рис. 3, а) приводило к подчеркиванию дефектов полос и появлению ложных изменений интенсивности вдоль полос. Для устранения этого недостатка восстановленное изображение подвергалось сглаживанию фильтром, осуществляющим свертку с импульсным откликом, построенным внутри области, вытянутой вдоль полос как показано на рис. 2 пунктирной линией. В результате 10-и итераций получена картина полос (см. рис. 3, б) с существенно повышенным контрастом и незначительным усилением влияния шума.

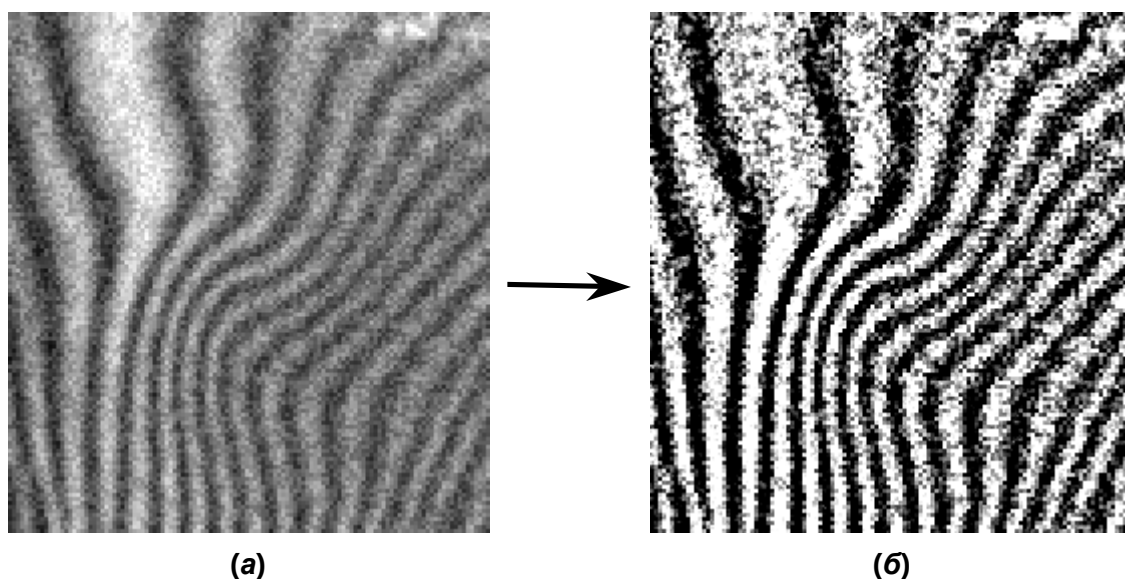


Рис. 3. Преобразование картины с равномерной потерей контраста

Предложенная методика была апробирована применительно к улучшению картин полос с неравномерной по полю потерей контраста (рис. 4, а). Результат обработки после 12-ти итераций показан на рис. 4, б. На рисунке видно значительное повышение контраста картины полос.

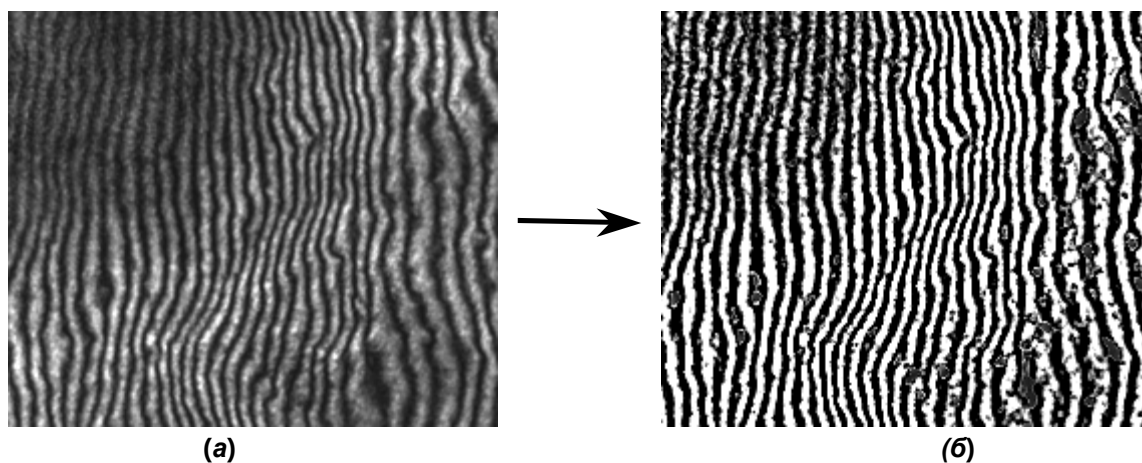


Рис. 4. Преобразование картины с неравномерной потерей контраста

Заключение

Алгоритм Ван Циттерта позволяет улучшить качество картин интерференционных полос, в том числе, уменьшить потери контраста полос. Применение в алгоритме Ван Циттерта двумерной функции отклика косинусоидального вида и ее модификация в зависимости от направления полос позволяют существенно уменьшить тенденцию к усилению шумов по сравнению с обычным немодифицированным двумерным алгоритмом Ван Циттерта. Для картин с сильно изменяющейся шириной полос после использования алгоритма Ван Циттерта целесообразно применять локально-адаптивный сглаживающий фильтр, который устраняет ложные детали, возникающие на таких полосах.

Авторы выражают признательность М.В. Волкову за полезное обсуждение результатов работы.

Литература

1. Hill N.R., Ioup G.E. // J. Opt. Soc. Am. 1966. V. 66. №5. P. 487-489.
2. И.И. Разумова, В.Н. Худик. Прикладные вопросы реализации алгоритма Ван Циттерта в оптико-электронной системе с обратной связью. В кн.: Оптическая обработка изображений. / Под ред. С.Б. Гуревича. Л, Наука, 1985. С. 56-66.
3. Gurov I., and Volkov M. Nonlinear filtering of noisy interference fringes with the 2-D spatially-dependent filter impulse response. In: Int. Conf. on Coherent and Nonlinear Optics ICONO'01 (Minsk, Belarus, 26 June-1 July 2001). To be published in Proc. SPIE, 2001.

ЦИФРОВЫЕ ФОТОАППАРАТЫ: СРАВНИТЕЛЬНЫЕ ХАРАКТЕРИСТИКИ И ПРИМЕНЕНИЕ

О.В. Павлушко

Рассмотрены принципы работы и особенности цифровых фотоаппаратов, а также их преимущества и недостатки по отношению к традиционной технике. Приведены основные сравнительные характеристики цифровых фотоаппаратов различных модификаций и области применения цифровой фотографии.

Введение

Сфера применения цифровых фотоаппаратов довольно разнообразна, они широко используются в полиграфии, научных исследованиях, медицине, геологии, криминалистике. В этих и многих других отраслях довольно часто возникает необходимость получения снимков практически мгновенно с последующей обработкой и пересылкой их на большие расстояния через сеть Интернет. Обширный рынок цифровой фототехники и разнообразные характеристики цифровых фотоаппаратов обуславливают необходимость сравнительного анализа научно-технических решений, представленного в настоящей работе.

Основные виды и характеристики цифровых фотокамер

Для получения изображения невысокого качества достаточно разрешения VGA (640x480). Для того, чтобы изображение было четким, прорисовывались мелкие детали, разрешение должно быть не менее XGA (1024x768). На сегодняшний день существуют камеры, разрешение которых достигает 1600x1200 и выше, вполне сравнимое с профессиональными фотоаппаратами. Разрешение определяется числом светочувствительных элементов (пикселей) на поверхности светочувствительного сенсора. На рынке светочувствительных сенсоров на сегодняшний день конкурируют две технологии: на основе ПЗС-матриц (приборов с зарядовой связью) и КМОП-кристаллов (комплементарных структур металл-окисел-полупроводник).

Кристаллы КМОП значительно дешевле и проще в изготовлении, к тому же они потребляют существенно меньше энергии, чем ПЗС. Светочувствительная микросхема – самый дорогой элемент цифрового фотоаппарата. Стоимость КМОП-датчика, сопоставимого по характеристикам с ПЗС-матрицей примерно вдвое ниже стоимости последней. Кроме этого, приборы ПЗС чаще всего выполняют единственную функцию регистрации интенсивности света. В кристаллах КМОП можно реализовать множество дополнительных функций, таких как аналого-цифровое преобразование, фильтрация сигнала, управление балансом белого и параметрами камеры и др. Плотность КМОП-элементов на кристалле и диапазон представления цвета можно повысить без существенного увеличения цены изделия.

КМОП-кристаллы работают с более низкими напряжениями питания, что уменьшает потребляемую мощность и позволяет строить портативные камеры с более высокими тактовыми частотами.

Одним из главных отличий между КМОП и ПЗС-сенсорами является то, что в КМОП-сенсоре АЦП интегрирован в кристалл, а при использовании ПЗС-сенсора он находится на внешнем чипе. Но по этой же причине КМОП-сенсор имеет более высокий уровень шума.

По указанным причинам в ближайшем будущем почти все камеры “начального уровня” будут выполнены на базе КМОП-приборов, а ПЗС будут применяться только в устройствах среднего и высокого класса.

Поскольку обычный сенсор регистрирует только яркость, то обычный цифровой фотоаппарат может воспроизвести только черно-белое изображение. Профессиональные модели выделяют данные о красном, зеленом и синем цветах

каждого пиксела при помощи трех отдельных сенсоров, причем изображение предварительно разделяется призмой на три компонента, каждый из которых попадает на свой сенсор. Такая технология позволяет предотвратить наложение цветов (когда границы красного, синего и зеленого цвета сдвинуты на изображении). Однако подобные камеры требуют прецизионной технологии изготовления, а из-за наличия призмы они более массивны и менее надежны. Кроме этого, в них должна использоваться улучшенная оптическая система, так что общая цена такой камеры существенно выше.

Другие виды камер используют какую-либо комбинацию главных цветов на каждом сенсоре и сложные алгоритмы для получения цветного изображения. В полупрофессиональных моделях обычно применяется только однокристалльный сенсор, захватывающий только одно значение красного, зеленого или синего цветов для каждого пиксела. После этого камера рассчитывает недостающие значения цветов при использовании цветов в соседних пикселах. Хотя такой подход довольно эффективен, при захвате повторяющихся цветовых узоров могут возникать искажения цветов.

Оптическая система

Помимо характеристик сенсора, важным фактором для качественной работы является оптическая система цифрового фотоаппарата. Она включает в себя высококачественный высокоразрешающий объектив, эффективность вспышки и возможность ручной установки параметров экспозиции.

Более дешевые модели имеют простые объективы и фиксированную или автоматическую фокусировку без возможности ручной настройки. В настоящее время существуют камеры, выполненные на основе зеркального объектива, включающего в себя несколько асферических элементов, что позволяет получить хорошее качество изображения по всей площади кадра, а не только в его центре, даже при крайних положениях диапазона увеличения. Наряду с автоматической фокусировкой у таких камер есть ручная. К моделям рассмотренного класса можно отнести Olympus D-600L, D-620L, Canon Powershot Pro-70, Nikon Coolpix 950, Minolta EX-1500, Kodak DC-260, Minolta Dimage 5(7), Olympus E-10.

Значимость качества оптической системы для цифрового фотоаппарата не меньшая, чем для обычного фотоаппарата. Размер ПЗС-матрицы составляет примерно 1,5 см по диагонали, в то время как диагональ кадра обычной пленки равна приблизительно 4,3 см. При этом обычный фотоаппарат обеспечивает от 30-ти до 60-ти пар черно-белых линий на миллиметр, в то время как цифровой фотоаппарат может обеспечить до 150-ти пар линий на миллиметр. В результате цифровой фотоаппарат очень чувствителен к условиям освещенности и количеству света, прошедшего через оптическую систему. При всем совершенстве современных технологий обработки данных неправильным было бы думать, что они способны обеспечить высокое качество изображения при неправильной экспозиции или недостаточном освещении.

У цифровой и обычной фотографии много общего, и поэтому многие приспособления могут использоваться как с теми, так и с другими фотоаппаратами. Учитывая вышесказанное, рынок профессиональных цифровых фотокамер изначально ориентировался на полную совместимость принадлежностей, однако оказалось, что эта совместимость дается очень дорогой ценой.

Примером оригинального технического решения является разработанный компанией Silicon Film миниатюрный модуль цифровой камеры в виде специальной вставки для 35-мм пленочных камер, которая позволяет перейти на цифровую фотографию, сохранив при этом возможность применения всех своих объективов и функциональность фотоаппарата. Устройство обеспечивает разрешение 1280x1024 и совместимо с камерами Canon EOS 1N, Canon EOS A2/5, Nikon F-5, Nikon N-90/F90 и

Nikon F-3. Конструктивно устройство выполнено как катушка с пленкой. Для съемки цифровых фотографий достаточно вынуть из камеры пленку и установить вместо нее модуль EFS-1 Film. Устройство позволяет сделать 24 снимка, после чего при помощи специального переходника миниатюрную камеру можно подключить к компьютеру для переноса изображений и последующей их обработки. Устройство значительно повышает возможности при фотосъемке, совмещая функциональность зеркального фотоаппарата и цифровой камеры.

В качестве другого примера отметим объектив камеры для научных исследований типа FujiFilm FinePix 4700, характеристики которого очень близки к характеристикам глаза, что позволяет выполнять съемку прямо через окуляр микроскопа, причем изображение занимает почти всю площадь кадра. Обычно приходится подбирать для объектива камеры специфический "окуляр", в рассматриваемом случае можно просто поместить камеру на место глаза наблюдателя и выполнить съемку.

Специфика цифровой технологии подразумевает использование более короткофокусных объективов, и отказ от совместимости позволил создать ряд дешевых моделей, не уступающих по качеству дорогим. Но при этом обнаружилось полное отсутствие каких бы то ни было аксессуаров к этим камерам. Финансовая ситуация, на наш взгляд, сегодня такова. Существует ряд профессиональных камер в ценовом диапазоне от 6000 до 50000 USD, с которыми может использоваться стандартное фотографическое оборудование, а суммарная стоимость доступного ассортимента для каждой из этих камер может составлять сотни тысяч USD. При этом существуют камеры ценой менее 2000 USD, имеющие почти такое же качество изображения, но при отсутствии специальных принадлежностей к ним.

Емкость памяти

Изображения содержат большой объем информации, поэтому особенно важное значение имеют электронные средства записи и архивирования изображений, т.е. "электронная память".

Существуют несколько стандартов, которые используются в цифровых фотокамерах. Это стандарт PCMCIA Type I, II, III, хорошо знакомый владельцам ноутбуков, и стандарты CompactFlash и SmartMedia. Практически во всех полупрофессиональных и любительских камерах в настоящее время используются стандарты CompactFlash и SmartMedia.

Максимальный объем SmartMedia-карточки достигает 256 Mb, хотя наиболее часто используются 32 и 64 Mb. Последняя разработанная CompactFlash-карточка имеет 120 Mb памяти, и максимальный объем постоянно растет.

Выбирая фотоаппарат, необходимо помнить о том, как удобнее будет считывать информацию с карточки. Например, владельцам ноутбуков имеет смысл использовать PCMCIA или CompactFlash (тогда для считывания нужен адаптер). Владельцам обычных персональных компьютеров больше подойдет SmartMedia. При этом требуется обычное устройство считывания с гибких дисков, при помощи которого можно считывать информацию. Существует ряд приспособлений, позволяющих использовать носители PCMCIA или CompactFlash с персональными компьютерами.

Наличие ЖК монитора

Практически все фотокамеры в настоящее время имеют ЖК монитор, поскольку он дает возможность быстрого просмотра снятого материала, выбор наиболее удачных снимков и удаление лишних, т.е. обеспечивает основные преимущества цифровой фотографии. Во многих случаях ЖК монитор имеет совмещенную функцию видеискателя. Но нужно учитывать, что при работе с монитором используется большое количество энергии.

Поскольку на ЖК-панелях отображаются изображения, сохраненные в камере, более предпочтительными считаются большие ЖК-панели. Во многих камерах сохраненные изображения отображаются в виде групп уменьшенных набросков. Например, на 1,8-дюймовой (4,6 см) ЖК-панели Olympus D400 Zoom (размеры панели измеряются по диагонали) размещаются до 16-ти набросков. Для экономии времени работы батарей без перезаряда предусмотрена функция отключения ЖК-панели при съемке.

Светочувствительность

Чувствительность сенсоров цифровых камер обычно представляется как эквивалент чувствительности согласно стандарту ISO определенной пленки. Для камер полупрофессионального класса типично наличие возможности изменения чувствительности ISO от 100 до 400. В моделях с изменяемой чувствительностью, например, DC4800Kodak, при недостаточном освещении сигнал сенсора усиливается, но это может приводить к нежелательному усилению электронного шума сенсора. В настоящее время не существует стандартов в определении эквивалентных значений чувствительности ISO для сенсоров ПЗС. По этой причине, например, компания Casio не приводит эквивалент ISO для своей камеры QV-7000SX.

Увеличение

Оптическое увеличение изменяет размер изображения, сфокусированного на светочувствительной матрице. Менее качественное цифровое увеличение просто увеличивает изображение на основе математических преобразований масштаба и добавления пикселей. У большинства моделей цифровое увеличение функционирует только при съемке с разрешением 640x480 пикселей. Часто в качестве параметра увеличения указывают результат сочетания оптического и цифрового увеличения. Современные модели фотокамер обеспечивают шестикратное оптическое и двукратное цифровое увеличение.

Форматы данных и архивирование изображений

Основными форматами представления цифровых фотографий являются TIFF (Tagged Image File Format) и JPEG (Joint Photographers Expert Group). Первый из них первоначально был разработан для платформы Macintosh и затем дорабатывался и улучшался. Формат TIFF имеет ряд преимуществ, в частности, он поддерживает 32-битную разрядность для цветового CMYK формата, что может быть полезно, например, при прямой печати данных с экрана монитора на принтере.

Формат JPEG совмещает две особенности – полный диапазон оттенков и низкий объем памяти, необходимой для хранения изображения (связанный однако с небольшой потерей качества). Сохранение цифровой фотографии в JPEG требует меньшего объема памяти, чем TIFF формат. Происходит это благодаря тому, что JPEG формат при компрессии распознает некоторые данные как неважные и отбрасывает их при сжатии, причем степень сжатия можно контролировать. Формат JPEG становится все более распространенным из-за удобства его использования при работе в информационной сети.

Удобство работы с цифровым изображением определяется скоростью передачи изображения из фотоаппарата в компьютер. Первоначально фотоаппарат напрямую подключался к компьютеру через последовательный порт RS-232, при этом передача данных (порядка 1 Мб) занимала около 15 мин. В настоящее время существует ряд других способов считывания данных со Smart Card напрямую. Основными устройствами считывания являются PCMCIA-адаптер (MA-2E), который позволяет быстро и удобно считывать данные с карты портативным компьютером; USB-адаптер

(MAUSB-2E), который подключается напрямую к USB – порту и позволяет очень быстро передавать данные на компьютер (средняя скорость передачи 12 Мбит/с, при этом файл емкостью 32Мб можно передать за 60 с); адаптер под 3.5”-дискету (MAFP-2E), в который вставляется Smart Card, и адаптер, в свою очередь, вставляется в компьютер вместо дискеты. Это достаточно удобный способ для считывания информации.

Некоторые принтеры могут считывать информацию напрямую с карты, что тоже является одним из способов передачи данных.

Дополнительные возможности. Печать фотоснимков

Существует большое разнообразие дополнительных функций цифровых фотокамер: возможность работать с объектом съемки в движении, записи звукового сигнала, печати фотографий непосредственно от камеры, наличие видеовыхода и др. Многие изготовители наделяют цифровые камеры возможностями, выходящими за рамки базовых функций, необходимых для получения изображений.

Камеры Ricoh RDC-2E, SoundVision SV mini-209, UMAX MDX 8000 и Vivitar ViviCam 3000 могут использоваться для проведения видеоконференций, хотя низкая частота регенерации (один кадр каждые несколько секунд) не позволяет фиксировать или отображать движущиеся изображения в реальном времени.

При печати напрямую в отсутствие общепринятых интерфейсных стандартов для каждой камеры требуется специальный принтер от фирмы-изготовителя камеры. Модели Casio, Olympus, Ricoh и Sony представляют собой принтеры с термопереносом красителя с форматом отпечатка размером с фотоснимок. В принтерах Fujі и Panasonic используется разработанная компанией Fujі технология Thermo-Autochrome. Модель Epson Stylus Photo - это цветной струйный принтер. Модель Olympus P-200 не только компактна, но и мобильна - при наличии дополнительной батареи можно печатать фотографии даже в полевых условиях. Фотоаппараты, совмещенные с фотопринтером, небогаты ресурсами памяти (не более 1,3 мегапиксела) и функциональными возможностями.

Недавно фирмой Canon были разработаны Bluetooth-модули, присоединяемые к обычной цифровой камере PowerShot S10 и струйному принтеру. С помощью этих модулей появляется возможность производить распечатку фотографий нажатием одной кнопки на камере. Никаких проводов не требуется, ограничен лишь радиус действия Bluetooth-передатчиков, составляющий максимум 10 метров.

Таблица

Сравнительные характеристики некоторых камер

Тип	Canon Powershot G1	Minolta Dimage 7	Nikon Coolpix 995	Olimpus E-10
	Цифровая фотокамера	Цифровая фотокамера	Цифровая фотокамера	Цифровой зеркальный фотоаппарат
Видео	PAL/NTSC	+	NTSC или PAL по выбору	PAL, разъем для подключения внешнего микрофона
Макс. разрешение	2048 x 1536, 1024 x 768, 640 x 480, 320 x 240 Movie (до 30 сек.)	2560x1920	2,048 x 1,536 пикселей; UXGA-размер (1600 x 1200); SXGA-	2240x1680, 1600x1200, 1280x960, 1024x768,

			размер (1280 x 960); XGA-размер (1024 x 768); VGA-размер (640 x 480); 3:2 (2048 x 1360); по выбору	640x480
Память в комплекте	CompactFlash 16mb	CF Type II	CompactFlash 16mb	CompactFlash 16Mb
Связь с компьютером	USB, RS-232C.	USB	USB	USB
Объектив	Фокусное расстояние 7-21мм (34-102мм в эквиваленте для 35мм фотоаппарата), светосила 1:2-2,5. Минимальное значение диафрагмы 8.	Фокусное расстояние 28 - 196 мм (wide, 35мм эквивалент) Материал линз – стекло, диафрагма 2.8 - 9.5, возможно установление временных насадок	4x Zoom Nikkor; f = 8.0~32.0мм [35мм (135) формат – эквивалентно 38~152мм]/F2.6 ~ 5.1 с макродиапазоном; 10 элементов в 8 группах; все элементы сделаны из экологически чистого стекла; Применено суперинтегрированное просветление Nikon Super Integrated Coat	Фокусное расстояние 9-36мм (35-140мм в эквиваленте для 35мм фотоаппарата), светосила 1:2,0-2,4, минимальное значение диафрагмы 11. 14 элементов в 11 группах, все линзы стеклянные, есть асферические элементы и элементы из низкодисперсного стекла.
Светочувствительность	ISO 50, 100, 200, 400, авто	До 800	100, 200, 400, 800, автонастройка; может управляться в любом режиме отработки экспозиции	Автоматическая, эквивалентна ISO 50, 100, 200
Глубина цвета			24 бит	24 бит
Вес	420	595	390 г	575
Размеры	119,7x76,8x63,8	117x91x113 мм	138 x 82 x 40 мм	120x86x152,2
Тип карт расширения памяти	Compact Flash	Compact Flash	CompactFlash	Smart Media/Compact Flash

Применение цифровых фотоаппаратов

В повседневной жизни использование цифровой фотографии имеет значительное преимущество перед традиционными технологиями благодаря скорости получения изображения, удобству обработки и потенциальной возможности быстрой пересылки по Интернету цифрового изображения. В связи с вышеуказанными удобствами

цифровая фотография получила распространение во многих областях. Особо хотелось бы отметить использование цифровых технологий в криминалистике, ортодонтологии, лечении туберкулеза.

Криминалистическая фотография является одной из отраслей криминалистической техники, традиционно применяемая для решения криминалистических задач. Преимущества средств цифровой фотографии - упрощение фиксации улик, гарантированное получение информации и ее достоверность, а также простота обработки изображений. Например, для выявления деталей в тенях и светах используется процедура "Выравнивание гистограммы", изменяющая уровень контраста и динамический диапазон изображения. При наличии фона с регулярной структурой используют операцию "вырезание частотной области", основанную на переходе в частотную область с применением быстрого преобразования Фурье. Данный метод, в основе которого лежит селективная фильтрация в частотной области, признан эффективным способом для выявления малоконтрастных следов пальцев рук. Дальнейшее развитие криминалистической фотографии может быть связано с использованием методов и средств трехмерного моделирования объектов. Для осмотра места происшествия используют полупрофессиональные камеры на ПЗС-матрице с разрешением от 1600x1200 до 2400x1800 элементов, для криминалистических экспертиз - цифровые камеры на ПЗС-линейке с разрешением от 3200x2400.

Использование цифровой фотографии весьма актуально в области медицины. В качестве примера можно упомянуть ортодонтическое лечение – процесс длительный, занимающий месяцы и даже годы. Для врача-ортодонта удобно иметь в своем распоряжении информацию, фиксирующую основные этапы лечения; наглядный план лечебных мероприятий и отметки о его выполнении. Ведение электронных карточек позволяет делать оперативные выборки по диагнозам пациентов, по проведенным мероприятиям, анализировать результаты лечения большого числа пациентов. В клинике «Меди», Санкт–Петербург, используется цифровая камера с максимальным разрешением 1280x960 и с рабочим отрезком от 8 см.

Использование компьютерных технологий для индивидуального учета и формирования групп риска по специально разработанной программе позволило сократить объем флюорографических обследований на 50-55% и выявлять при этом 75-87,9% больных туберкулезом. В Новомосковском городском противотуберкулезном диспансере Тульской области под руководством сотрудников НИИ фтизиопульмонологии ММА им. И.М. Сеченова в 1992 г. разработана и в 1993 г. внедрена система активного выявления туберкулеза на основе компьютерных технологий, включающая базу данных, программу экспресс-определения индивидуальной степени риска заболевания туберкулезом и методику выявления туберкулеза врачами общей лечебной сети по клиническим проявлениям заболевания. Для более эффективного и экономичного выявления туберкулеза интенсивно внедряются в практику методы цифровой бесплочной флюорографии. Преимущества цифровых бесплочных флюорографов перед пленочными заключаются в том, что они позволяют снизить радиационную нагрузку на пациента более чем в 4 раза, избежать расходов на пленку и химреактивы, получать рентгеновское изображение непосредственно по окончании экспозиции на экранах компьютерных мониторов, что исключает возможность брака изображения. Аппараты данного класса выпускает ряд отечественных производителей. Среди них можно выделить цифровую камеру КФЦ «Электрон», где в качестве детектора используется ПЗС-матрица больших размеров и принцип цифрового фотоаппарата, позволяющие добиться разрешения 2,5–2,8 пар линий на мм, при этом принцип сканирования, используемый в большинстве отечественных флюорографов не дает разрешения более 1,5 пар линий на мм. В настоящее время в России находится в эксплуатации более ста отечественных

беспленочных цифровых флюорографов различных моделей.

Заключение

Приобретение цифровой фотокамеры с форматом кадра в 2-3 мегапиксела в настоящее время вполне оправданно, поскольку по функциональным возможностям она не устареет в короткие сроки в отличие от камер предыдущих поколений. Комплект из цифровой фотокамеры и принтера полностью удовлетворяет актуальным потребностям фотографии для профессионального применения и научных исследований.

Литература

1. Шарыгин М.Е.. Сканеры и цифровые камеры. СПб: БХВ - Санкт-Петербург, 2000.
2. Лысов И. Самоучитель по современной фотографии. Самостоятельно от азов к Мастерству. М.: А.Д.& Т, 2000..
3. Панченко В.Г., Яценко А.Н. Убийца фотографии. М., 2000.

Дополнительные источники

4. <http://www.interlink.ru/dp-help-choice.html> - *фотоаппараты, характеристики, ссылки на статьи*
5. <http://www.cosmetolog.ru/is/is3/inf/inf.htm> - *ортодонтология*
6. <http://www.rmj.ru/rmj/t8/n12/492.htm> - *туберкулез*
7. http://www.mvd-expo.ru/CONFER/CRIM_MVD/doc19.htm - *применение в криминалистике.*
8. <http://www.novinki.ru/userlev3.php?id=829> - *мини фотоаппарат*
9. <http://www.3dnews.ru/rewies/peripheral/digitalphoto/index02.htm> – *анатомия цифрового фотоаппарата*
10. <http://www.zdne.ru/id=21786> – *Прорыв в цифровой фотографии?*
11. <http://www.flashcom.ru/html/camera/> - *о цифровых фотоаппаратах*

ОПТИЧЕСКИЕ ВОЛОКОННЫЕ УСИЛИТЕЛИ ИНФОРМАЦИОННЫХ СИГНАЛОВ

Н. С. Макаров

Работа посвящена обзору различных вариантов волоконных оптических усилителей, принципов их работы, преимуществ и недостатков. Также рассматривается предложенный нами метод одновременного стоксового и антистоксового ВКР-усиления в волокнах.

Введение

При передаче оптических сигналов по волоконным линиям связи на большие расстояния, вследствие рассеяния и поглощения, ослабляющих информационный сигнал, необходимо использовать последовательно расположенные усилители-повторители. Известны два типа повторителей – оптико-электронные, основанные на преобразовании свет – электронный сигнал – свет, и полностью оптические, которые в последнее время получили большое распространение. В качестве полностью оптических повторителей используют волоконные усилители, основанные на явлении вынужденной эмиссии в волокнах, активированных ионами редкоземельных металлов, а также рамановские усилители, в которых используется нелинейно-оптический эффект вынужденного комбинационного рассеяния (ВКР) в волокнах. Такие системы усиливают оптические сигналы в относительно широком диапазоне частот, включающем десятки и сотни информационных каналов, мультиплексированных по длинам волн. Они не чувствительны к скорости передачи данных вплоть до 100 Гб/с и работают с любыми форматами передачи данных [1-13]. В данной работе рассматриваются различные варианты волоконных оптических усилителей, принципы их работы, преимущества и недостатки.

Эрбиевые волоконные усилители

В настоящее время наибольшее распространение получили усилительные системы на основе волокна, активированного ионами эрбия Er^{3+} (erbium-doped fiber amplifier – EDFA) и позволяющего эффективно усиливать сигналы в спектральной полосе наименьшего поглощения кварцевого стекла 1535...1565 нм, а также в области 1570...1610 нм [4].

В линиях связи применяются три различных варианта эрбиевых волоконных усилителей (ЭВУ) [4, 5].

В первом варианте *усилитель мощности (УМ)* устанавливается непосредственно после мультиплексирующей системы, суммирующей излучение полупроводниковых лазеров, которые генерируют излучение на различных длинах волн и формируют спектральные информационные каналы. УМ предназначен для усиления широкополосного сигнала до уровня, который не может быть достигнут в отдельных полупроводниковых лазерах. Усиление по мощности в УМ достигает 30 дБ.

Во втором варианте *линейный усилитель (ЛУ)* устанавливается в промежуточных точках протяженных линий связи между регенераторами на расстояниях до 160 км с целью компенсации ослабления сигнала, которое происходит из-за затухания в оптическом волокне. Усиление в ЛУ обычно составляет 18...23 дБ.

В третьем варианте *предварительный усилитель (ПУ)* устанавливается непосредственно перед устройством демультиплексирования по длинам волн и обеспечивает увеличение отношения сигнал/шум на входе электронных каскадов усиления после системы оптоэлектронных детекторов, регистрирующих информационные сигналы на различных длинах волн. Усиление в ПУ составляет 18...23 дБ.

Конструктивно типы ЭВУ различаются по уровням мощности накачки, концентрации ионов активатора и длине самого волокна.

Простой однокаскадный ЭВУ [4, 5] состоит из бобины с намотанным на нее эрбиевым волокном длиной 1...30 м (рис. 1).

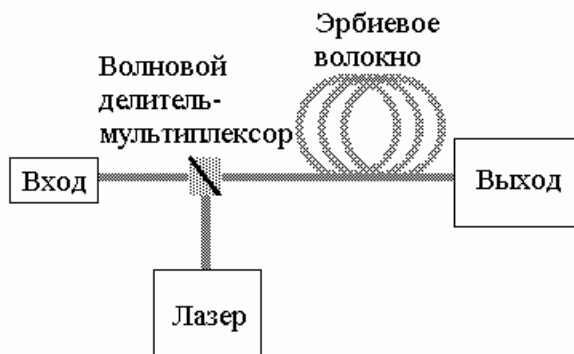


Рис. 1. Принципиальная схема однокаскадного эрбиевого волоконного усилителя

Оптическая накачка волокна осуществляется на длинах волн 980 нм и/или 1480 нм, после чего лазерное излучение с помощью оптического мультиплексора вводится в усилительное волокно. Энергия накачки поглощается ионами эрбия, которыми активировано кварцевое волокно, переводя их в возбужденное состояние. Усиление происходит, когда сигнальный фотон с длиной волны 1550 нм инициирует переход возбужденного иона эрбия в основное состояние с излучением идентичного фотона на длине волны 1550 нм.

Однокаскадный усилитель [9, 10] обычно содержит один или два лазера накачки, при этом для увеличения усиления осуществляется поляризационное или частотное комбинирование большего числа источников накачки. Усиление информационного оптического сигнала возможно при попутном распространении излучения накачки, при встречном, а также при их комбинации.

Простой однокаскадный усилитель не вполне удовлетворяет требованиям современных телекоммуникационных систем, поэтому его дополняют рядом специальных устройств. Для предотвращения попадания излучения накачки или усиленной спонтанной эмиссии эрбиевого волокна в линию передачи на входе и/или выходе усилителя необходимо использовать оптические изоляторы. Для получения информации о мощности и спектре сигнала на входе и выходе используются оптические разветвители с измерительными устройствами. Сигналы этих устройств используются для управления мощностью лазерного излучения накачки, контроля коэффициента усиления, или для оповещения об аварийной ситуации [4, 5].

При передаче большого числа информационных каналов, мультиплексированных по длинам волн (wavelength-division multiplexing – WDM), используется расширенный диапазон частот, в котором сложно обеспечить равномерную частотную характеристику ЭВУ. Частотная зависимость становится острой проблемой в системах большой протяженности, в которых используются несколько последовательно расположенных секций с ЭВУ между ними. Указанный недостаток устраняется за счет применения узкополосных сглаживающих оптических фильтров для ослабления сигналов на частотах, которым соответствуют большие коэффициенты усиления. Для коммерческих ЭВУ с неравномерностью усиления менее 1 дБ в диапазоне рабочих частот фильтры тщательно согласовываются с архитектурой усилителя. Сглаживание частотной характеристики усиления обычно достигается при обеспечении требуемых значений ряда параметров, таких как уровни входной мощности сигнала и накачки, длины волокна и т.д. В однокаскадных усилителях сглаживание частотной характеристики приводит либо к потере выходной мощности, либо к дополнительному ослаблению информационного сигнала до его усиления. Поэтому обычно для сглаживающих усилителей используются многокаскадные схемы [4, 5].

В системах ЭВУ с мультиплексированием по длинам волн обычно необходимы управление усилением, добавление или удаление информационных сигналов на различных длинах волн, а также компенсация дисперсии групповых скоростей, которая возникает при прохождении длинного оптического волокна. Это вносит дополнительные потери, которые могут достигать 10 дБ и более, и могут быть компенсированы только в многокаскадных усилителях [4, 5].

Оптическая схема двухкаскадного ЭВУ приведена на рис. 2. Как правило, в подобных устройствах используется попутный лазер накачки с длиной волны 980 нм в первом каскаде и встречный лазер накачки с длиной волны 1480 нм – во втором, что позволяет избежать попадания лазерного излучения накачки на вход и выход информационной сети.

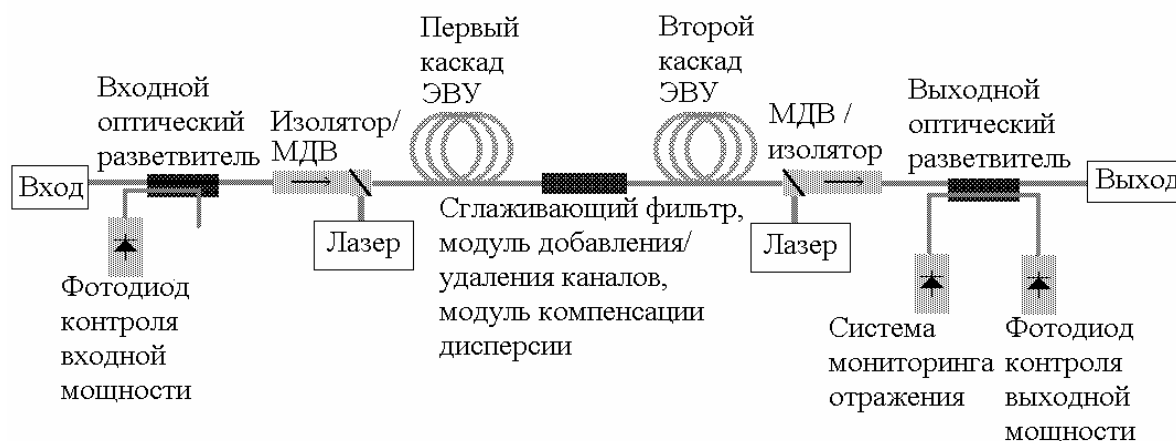


Рис. 2. Многокаскадный эрбиевый усилитель, МДВ – система мультиплексирования по длинам волн

Сглаживающий фильтр, модуль добавления/удаления каналов и модуль компенсации дисперсии вносят существенные потери в двухкаскадном ЭВУ, поэтому их, как правило, располагают между каскадами усиления.

Из кривых спектров усиления многокаскадного ЭВУ, представленных на рис. 3, следует, что применение сглаживающего фильтра приводит к неравномерности усиления менее 0,3 дБ во всей частотной полосе усиления.

До недавнего времени ЭВУ были основным, если не единственным, видом оптических усилителей при передаче данных по волоконным линиям связи на большие дистанции [4]. Возрастающие потребности в скорости передачи данных и количестве спектральных каналов обусловили появление альтернативных технологий.

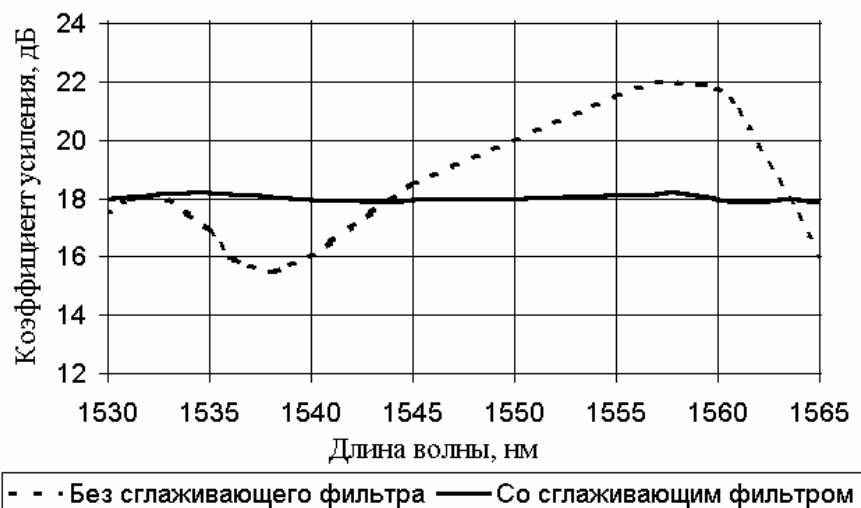


Рис. 3. Спектральные кривые усиления многокаскадного эрбиевого усилителя

Тулиевые волоконные усилители

Усилительные системы на основе волокна, активированного ионами тулия Ti^{3+} (thulium-doped fiber amplifier – TDFA) работают подобно ЭВУ: оптическое усиление сигнала в них происходит в результате преобразования энергии накачки в энергию сигналов при распространении излучения в усилительном волокне [6, 7]. Разница между ЭВУ и тулиевыми волоконными усилителями (ТВУ) заключается в способе усиления, определяемом ионами-активаторами и схемой накачки (рис. 4).

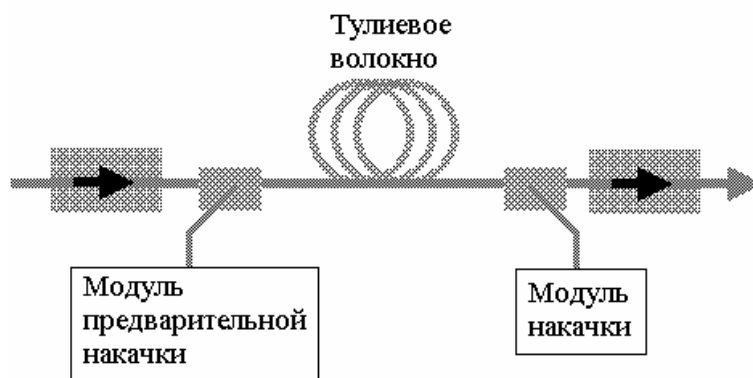


Рис. 4. Принципиальная схема тулиевого волоконного усилителя

Схема энергетических уровней (рис. 5) иллюстрирует процесс оптического усиления в ТВУ. Усиление осуществляется с использованием двух волн накачки с одинаковыми или различными частотами. Фотоны предварительной накачки переводят ионы активатора на энергетический уровень E_2 , являющийся нижним рабочим уровнем, затем фотоны основной накачки используются для заселения верхнего рабочего энергетического уровня E_3 , вынужденный переход с которого на уровень E_2 обеспечивает усиление оптических сигналов.

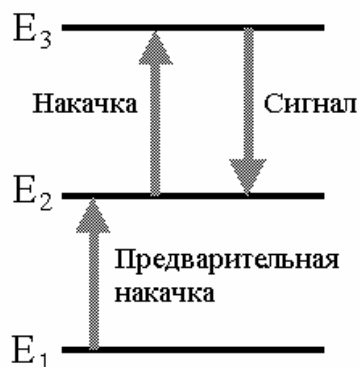


Рис. 5. Схема энергетических уровней тулиевого волоконного усилителя

При использовании указанной схемы накачки центр усиления фтористого, фтористоциркониевого или многокомпонентного кварцевого волокна с высокой концентрацией ионов тулия находится вблизи длины волны 1460 нм (рис. 6). Ширина спектральной полосы усиления с уровнем 20 дБ составляет в таких волокнах примерно 35 нм, а максимальное усиление может достигать 31 дБ [6, 7]. Применяя иные схемы накачки можно также получить усиление в диапазоне длин волн 1480...1510 нм [6].

Основным препятствием широкому распространению ТВУ является недостаточно высокая прочность используемого в усилителе оптического волокна. В отличие от кварцевого волокна, оно более ломкое, поэтому при его производстве, прокладке и обслуживании требуются специальные методики. Другой проблемой является сложность обеспечения высокого качества оптического соединения различных по физико-химическому составу волокна ТВУ и обычного кварцевого волокна. Так как сварка этих волокон невозможна, используются менее надежные способы механического соединения торцов различных волокон или соединения с помощью эпоксидной смолы. Подобные соединения вносят потери более 0,3 дБ, в отличие от потерь 0,05 дБ при сварке [7]. Кроме этого, в состав волокна входит цирконий, что обуславливает высокую себестоимость производства. На сегодняшний день низкая надежность и высокая стоимость ТВУ препятствуют их широкому распространению.

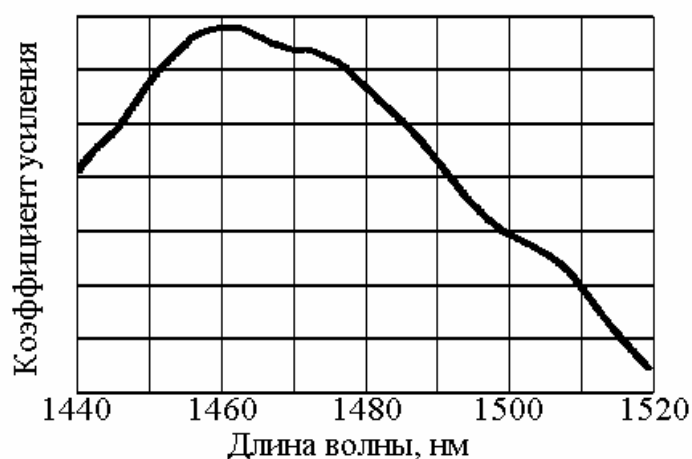


Рис. 6. Спектральная кривая усиления тулиевого волоконного усилителя

ВКР волоконные усилители

Системы, построенные на основе ЭВУ и ТВУ, обладают большим уровнем шумов на частотах информационных сигналов, что обусловлено влиянием спонтанного

излучения ионов активаторов. Усиление на основе ВКР позволяет обеспечить дополнительное повышение уровня сигнала без существенного увеличения шума на частоте сигнала [7].

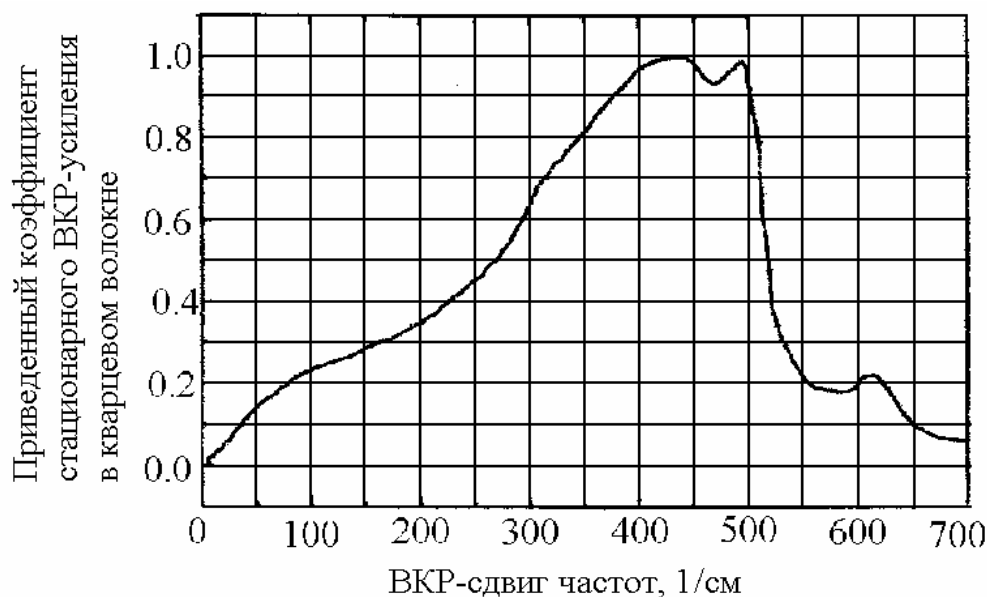


Рис. 7. Зависимость коэффициента ВКР-усиления в кварцевом волокне от сдвига частот

Усиление в процессе ВКР обусловлено нелинейно-оптической перекачкой энергии накачки в энергию Стокса с участием фоновой волны среды. Например, для достижения усиления на длине волны 1550 нм (информационный сигнал), необходима накачка на длине волны 1455 нм, при этом за счет широкой полосы ВКР усиления кварцевого волокна усиление происходит и на соседних частотах (рис. 7) [7, 8]. Коэффициент ВКР-усиления в кварцевом волокне является поляризационно-зависимым, и оптимальное усиление происходит только тогда, когда поляризации сигнала на стоксовой частоте и частоте накачки совпадают. Высокое усиление без поляризационно-зависимых потерь достигается при использовании источников накачки с ортогональными поляризациями (рис. 8).

Процесс ВКР-усиления происходит в любом оптическом волокне при соответствующей накачке и поэтому его можно использовать для улучшения параметров не только новых, но и уже существующих оптических линий передачи информации. Для сглаживания кривой усиления и для обеспечения большей мощности усилителя возможно применение накачки на нескольких длинах волн (рис. 9) [7, 8].

Основным недостатком ВКР-усилителей является их низкая эффективность по сравнению с ЭВУ. Для достижения усиления 30 дБ ЭВУ необходима мощность накачки около 50 мВт, тогда как для аналогичного усиления ВКР-усилителя необходима мощность накачки более 1 Вт [7].

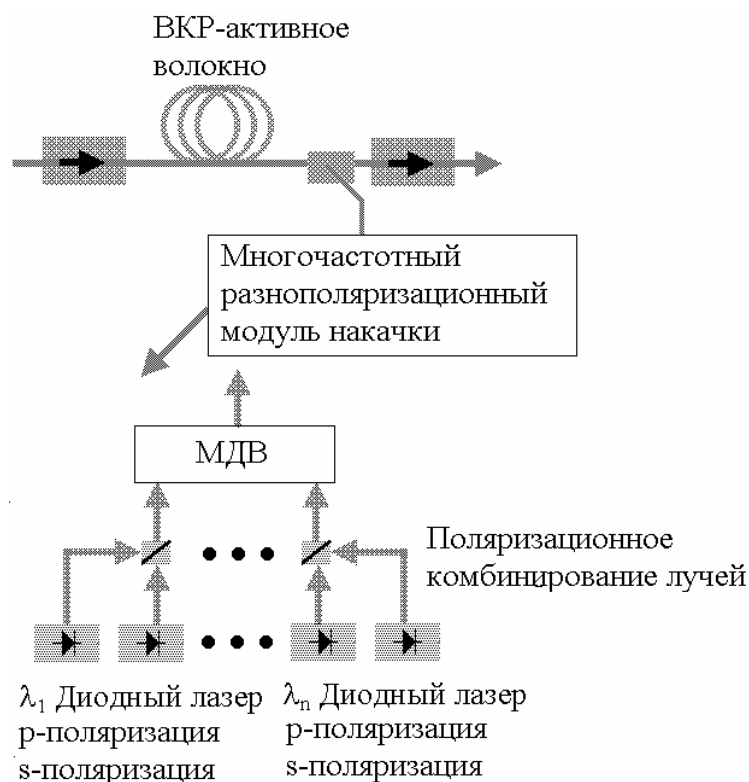


Рис. 8. Схема ВКР-усилителя с использованием накачки на нескольких длинах волн с различной поляризацией

В последнее время [7] появились мощные диодные лазеры с мощностью до 1 Вт и фазированные линейки диодных лазеров с мощностью более 10 Вт, что позволило создать эффективные ВКР-усилители, способные конкурировать с ЭВУ. На этой основе ожидается скорое появление коммерческих ВКР-усилителей.

ВКР-усилители могут быть использованы для усиления в S-полосе (1480...1520 нм), в L-полосе (1570...1610 нм) и в диапазоне от 1642 до 1672 нм [7, 8]. В оптических сетях ВКР-усиление может использоваться как единственный механизм усиления или в комбинации с ЭВУ.

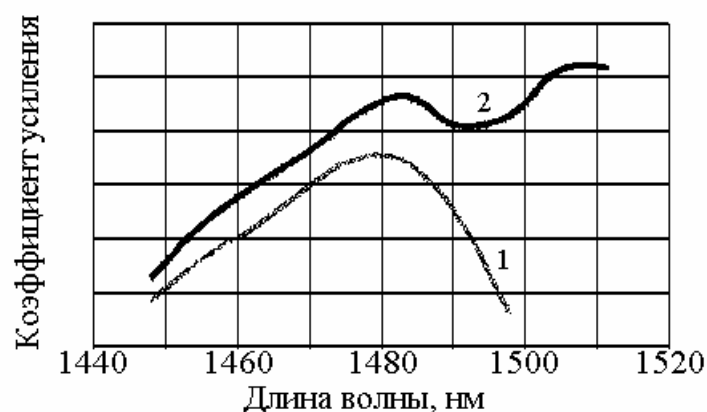


Рис. 9. Спектральные кривые усиления ВКР-усилителя при накачке на одной (1) и на двух (2) длинах волн

Гибридные усилители

Ни один из существующих видов оптических усилителей не может удовлетворить всем современным потребностям телекоммуникационных систем, поэтому в настоящее время активно ведутся разработки новых видов оптических усилителей.

Благодаря тому, что полоса усиления ВКР-усилителя может быть сдвинута в сторону более высоких или низких частот по отношению к полосе усиления ЭВУ [9, 10], при использовании достаточно мощных лазеров накачки можно согласовать усилители таким образом, чтобы результирующая полоса усиления была значительно шире, чем у каждого из усилителей в отдельности. При этом можно значительно сгладить неравномерность кривых усиления отдельных усилителей (рис. 10).

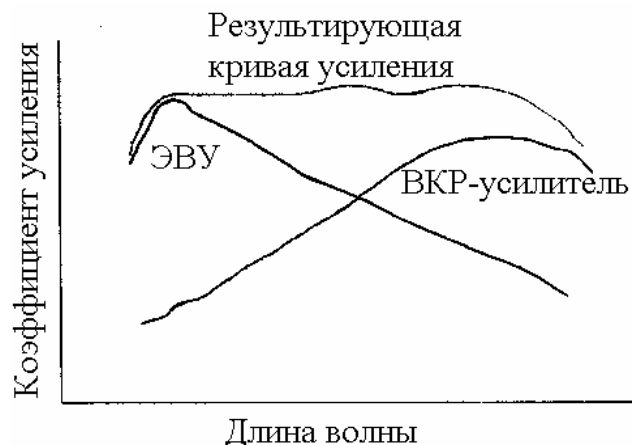


Рис. 10. Сглаживание кривой общего усиления при использовании ВКР-усилителя и ЭВУ

Экспериментальная система, разработанная в лаборатории Нортел Харлоу (Эссекс, Англия), при использовании данной гибридной схемы обеспечила передачу 32-х спектральных каналов на расстояние 1000 км с пропускной способностью каждого канала 40 Гб/с с общей скоростью передачи 1,28 Тб/с [11].

Стокс-антистоксовые ВКР-усилители

Для одновременного усиления в окнах прозрачности 1310 и 1550 нм кварцевого волокна можно использовать комбинированное стоксовое и антистоксовое ВКР-усиление в условиях фазового квазисинхронизма [11-12]. Для обеспечения условия фазового квазисинхронизма волокно должно состоять из чередующихся областей с нелинейными свойствами (активные слои) и без нелинейных свойств (пассивные слои). Толщины слоев должны выбираться таким образом, чтобы в каждом активном слое происходила эффективная перекачка энергии из волны накачки в антистоксовую компоненту ВКР. Основные свойства фазового квазисинхронизма при ВКР были подробно исследованы в [13]. Расчеты, проведенные с использованием данной методики, показали, что длина волокна, на которой усиление достигает 13 дБ, составляет примерно 3 км.

За счет широкой полосы усиления кварцевого волокна ВКР-усиление происходит не на одной частоте, а в некоторой полосе частот (рис. 11) [11-12]. Так как в широком диапазоне длины волны накачки (1400...1500 нм) слоистая структура остается неизменной, для увеличения мощности усилителя, увеличения и сглаживания полос усиления можно использовать накачку на нескольких длинах волн.



Рис. 11. Спектральная кривая стоксового и антистоксового усиления в кварцевом волокне

Одновременное стоксовое и антистоксовое ВКР-усиление обладает всеми преимуществами и недостатками стоксового ВКР-усиления (при этом немного снижается результирующий коэффициент усиления). Основным недостатком данного метода является высокая сложность реализации слоистой структуры, состоящей из большого числа слоев.

Заключение

В настоящее время в волоконно-оптических линиях связи реально используются только эрбиевые полностью оптические усилители, хотя их возможностей явно недостаточно для удовлетворения требований, предъявляемых к телекоммуникационным системам следующего поколения. Для увеличения полосы пропускания (числа спектральных каналов) необходимо использование гибридных усилителей, сглаживающих кривую усиления. Реализация гибридного усилителя с одновременным стоксовым и антистоксовым усилением позволит увеличить полосу пропускания. С ростом скоростей передачи данных относительно низкое быстродействие эрбиевых усилителей вызовет ряд проблем, в результате чего ВКР-усилители займут лидирующие позиции на рынке волоконных оптических усилителей.

Литература

1. Nakagawa K., Nishi S., Yoneda E. // J. Lightwave Technol. 1991. V. 9. P. 198-207.
2. Randy G., Tingyc L. I. // Proc. IEEE. 1996. V. 84. P. 870-883.
3. Urquhart P. // Proc. IEEE. 1988. V. 6. P. 385-407.
4. Trivedi D. A., Strite T., Gerlas van den Hoven // WDM solutions. 2000. № 4. P. 14-20.
5. Daniel C. McCarthy // Photonics Spectra. 2001. № 7. P. 88-98.
6. Yvonne Carts-Powell // WDM solutions. 2001. № 7. P. 9, 20.
7. Islam M., Nietubyc M. // WDM solutions. 2001. № 3. P. 53-62.
8. Hecht J. // Laser Focus World. 2001. № 6. P. 135-140.
9. Ahmed M. H., Shalaby M., Misk F. M. // Pure Appl. Opt. 1998. V. 7. P. 659-666.
10. Savage N. // WDM solutions. 2000. № 4. P. 8.
11. Bains S. // WDM Solution. 2001. № 4. P. 9.
12. Bespalov V. G., Makarov N. S. // Proc. SPIE. 2001. V. 4605. P. 280-285.
13. Беспалов В. Г., Макаров Н. С. // Известия РАН. Серия физическая. 2002. Т. 66. № 3. С. 350-352.
14. Bespalov V. G., Makarov N. S. // Proc. SPIE. 2001. V. 4268. P. 109-116.

АНАЛИЗ АКУСТИЧЕСКИХ СИГНАЛОВ НА ОСНОВЕ МЕТОДА ФИЛЬТРАЦИИ КАЛМАНА

И.П. Гуров, П.Г. Жиганов, А.М. Озерский

Рассматриваются особенности динамической обработки стохастических сигналов с использованием дискретных фильтров Калмана первого и второго порядка. Показана необходимость наличия адекватной априорной информации для эффективного анализа акустических сигналов. Даны примеры обработки акустических сигналов фильтрами Калмана первого и второго порядка.

Введение

Динамический анализ стохастических сигналов имеет важное значение при решении широкого круга научно-технических задач. Важным видом задач обработки является идентификация или определение параметров исследуемых процессов на основе анализа поступающей акустической информации. Для анализа сигналов, как правило, используются методы спектрального и стохастического анализа [1, 2]. Особенностью спектрального подхода является использование всех отсчетов данных, зарегистрированных в анализируемой выборке. Многие типичные исследуемые процессы имеют специфический спектральный состав и занимают характерные спектральные области, что позволяет легко выделять информативные составляющие сигналов. К недостаткам классического спектрального анализа относятся малая чувствительность к локальным свойствам сигналов, недостаточно высокое спектральное разрешение и сравнительно большие вычислительные затраты. Как правило, спектральный анализ проводится по выборке данных достаточно большой протяженности, и короткие локальные изменения не вносят значительного вклада в результирующий спектр сигнала. Вместе с тем, часто именно локальные изменения свойств сигналов, например, при ударных процессах, могут содержать полезную информацию. Большой объем обрабатываемых данных и сложные алгоритмы обработки требуют значительных вычислительных затрат при анализе спектров даже в случае использования быстрых алгоритмов типа быстрого преобразования Фурье (БПФ).

Методы стохастического анализа [3, 4] позволяют анализировать как глобальные, так и локальные свойства сигналов в динамическом режиме и требуют меньших вычислительных затрат. Такие методы основываются на параметрических моделях, в структуре которых учитывается априорная информация об общих свойствах анализируемых сигналов. При использовании методов стохастической фильтрации ведется обработка только той информации, которая необходима в рамках решаемой задачи. Например, если известно, что следует анализировать данные только в определенной полосе частот, можно обойтись без спектрального преобразования и последующего анализа во всей спектральной области, используя фильтр Калмана для выделения в исходных данных необходимых спектральных областей.

В настоящей работе рассмотрены параметрические модели сигналов и исследованы возможности динамической обработки стохастических сигналов фильтром Калмана первого и второго порядка для выделения информативных составляющих реальных акустических сигналов.

Метод фильтрации Калмана

Рассмотрим вначале дискретный линейный фильтр Калмана. Такой фильтр определяется векторным уравнением наблюдения

$$\mathbf{s}(k) = \mathbf{C}(k)\boldsymbol{\theta}(k) + \mathbf{n}(k) \quad (1)$$

и уравнением системы

$$\boldsymbol{\theta}(k+1) = \mathbf{A}(k)\boldsymbol{\theta}(k) + \mathbf{w}(k). \quad (2)$$

В уравнениях (1) и (2) $\mathbf{s}(k)$ - последовательность регистрируемых дискретных значений сигнала, $\mathbf{C}(k)$ - матрица измерений, $\boldsymbol{\theta}(k)$ - вектор параметров, $\mathbf{n}(k)$ - шум наблюдений, $\mathbf{A}(k)$ -

матрица перехода, $\mathbf{w}(k)$ - шум системы, $k=1, \dots, K$ - номер отсчета дискретного сигнала. Обозначим ковариационные матрицы шумов $\mathbf{n}(k)$ и $\mathbf{w}(k)$ как $\mathbf{R}_n$ и $\mathbf{R}_w$ соответственно. Прогноз вектора параметров для шага $k+1$, согласно (2), можно определить в нулевом приближении как $\mathbf{A}(k)\boldsymbol{\theta}(k)$. Невязка между измеренным и прогнозированным значениями равна $\mathbf{v}(k+1)=\mathbf{s}(k+1)-\mathbf{C}(k+1)\mathbf{A}(k)\boldsymbol{\theta}(k)$. Тогда оценка вектора параметров имеет вид

$$\boldsymbol{\theta}(k+1)=\mathbf{A}(k)\boldsymbol{\theta}(k)+\mathbf{P}(k+1)\mathbf{v}(k+1), \quad (3)$$

где $\mathbf{P}(k+1)$ - коэффициент усиления фильтра Калмана.

Обозначим ковариационную матрицу погрешности предсказания вектора параметров $\boldsymbol{\theta}(k+1)$ как $\mathbf{R}_{pr}(k+1)$, а ковариационную матрицу апостериорной оценки - как $\mathbf{R}(k+1)$. Из (3) получим

$$\mathbf{R}(k+1)=[\mathbf{I}-\mathbf{P}(k+1)\mathbf{C}(k+1)]\mathbf{R}_{pr}(k+1)[\mathbf{I}-\mathbf{P}(k+1)\mathbf{C}(k+1)]^T+\mathbf{P}(k+1)\mathbf{R}_n\mathbf{P}^T(k+1), \quad (4)$$

где $\mathbf{I}$ - единичная матрица. Диагональные элементы матрицы $\mathbf{R}(k+1)$ представляют собой дисперсии компонентов вектора параметров, и коэффициент усиления $\mathbf{P}(k+1)$ следует выбирать так, чтобы эти дисперсии минимизировать.

Условие минимума дисперсий можно выразить в форме

$$-\mathbf{R}_{pr}(k+1)\mathbf{C}^T(k+1)+\mathbf{P}(k+1)[\mathbf{C}(k+1)\mathbf{R}_{pr}(k+1)\mathbf{C}^T(k+1)+\mathbf{R}_n]=0,$$

откуда находим коэффициент усиления фильтра в виде

$$\mathbf{P}(k+1)=\mathbf{R}_{pr}(k+1)\mathbf{C}^T(k+1)[\mathbf{C}(k+1)\mathbf{R}_{pr}(k+1)\mathbf{C}^T(k+1)+\mathbf{R}_n]^{-1}. \quad (5)$$

Из (4) и (5) имеем $\mathbf{R}(k+1)=[\mathbf{I}-\mathbf{P}(k+1)\mathbf{C}(k+1)]\mathbf{R}_{pr}(k+1)$.

Вследствие (2), можно записать выражение для ковариационной матрицы погрешности предсказания в форме

$$\mathbf{R}_{pr}(k+1)=\mathbf{A}(k)\mathbf{R}(k)\mathbf{A}^T(k)+\mathbf{R}_w. \quad (6)$$

В результате алгоритм обработки данных на каждом шаге определяется следующими выражениями:

$$\boldsymbol{\theta}(k+1)=\mathbf{A}(k)\boldsymbol{\theta}(k)+\mathbf{P}(k+1)[\mathbf{s}(k+1)-\mathbf{C}(k+1)\mathbf{A}(k)\boldsymbol{\theta}(k)], \quad (7)$$

$$\mathbf{P}(k+1)=\mathbf{R}_{pr}(k+1)\mathbf{C}^T(k+1)[\mathbf{C}(k+1)\mathbf{R}_{pr}(k+1)\mathbf{C}^T(k+1)+\mathbf{R}_n]^{-1}, \quad (8)$$

$$\mathbf{R}(k+1)=[\mathbf{I}-\mathbf{P}(k+1)\mathbf{C}(k+1)]\mathbf{R}_{pr}(k+1). \quad (9)$$

Уравнение (6) определяет эволюцию ковариационной матрицы $\mathbf{R}_{pr}(k+1)$ погрешности предсказания параметров на каждом шаге. Оценка вектора параметров на следующем шаге (7) вычисляется с учетом коэффициента усиления, рассчитанного согласно (8). Ковариационная матрица апостериорной оценки параметров определяется выражением (9). Для применения алгоритма требуется знать начальные условия $\boldsymbol{\theta}(0)$ и $\mathbf{R}(0)$, получаемые на этапе предварительной обработки сигнала.

Фильтр Калмана второго порядка

При необходимости выделения произвольной полосы частот узкополосных процессов использование фильтра Калмана первого порядка не эффективно. Для полосовой фильтрации стохастических сигналов целесообразно использовать фильтр Калмана второго порядка. Такой фильтр для скалярного дискретного случая определяется уравнениями:

$$\boldsymbol{\theta}_1(k-1)=\boldsymbol{\theta}(k-1)-\boldsymbol{\theta}(k-2),$$

$$\boldsymbol{\theta}_1(k)=\boldsymbol{\theta}_1(k-1)-2a\boldsymbol{\theta}_1(k-1)-b^2\boldsymbol{\theta}(k-1)+b^2w(x), \quad (10)$$

где a, b - коэффициенты, $w(x)$ - белый шум.

Такой фильтр позволяет осуществлять фильтрацию узкополосного процесса, спектр мощности которого сосредоточен около некоторой центральной частоты в сравнительно узкой полосе частот. Спектральная плотность случайного процесса, определяемого уравнениями (10), имеет вид [3]:

$$G_{\theta}(u)=\frac{b^4N_{\theta}}{2[(b-2\pi u)^2+a^2][(b+2\pi u)^2+a^2]} \quad (11)$$

при $b^2 \gg a^2$. Функция, определяемая уравнением (11), имеет максимумы на частотах $\pm b/2\pi$, и ширина полосы пропускания фильтра определяется параметром a .

Фильтрация акустических сигналов

Фильтр Калмана первого порядка может быть использован для выделения низкочастотных составляющих сигнала. Частотная характеристика такого фильтра определяется выражением [3]

$$G_\theta(u) = \frac{aN_\theta}{2(a^2 + 4\pi^2 u^2)} \quad (12)$$

и имеет вид, показанный в левой части рис. 1. Выражение (12) определяет полуширину кривой спектральной плотности выходного сигнала при поступлении на вход фильтра гауссовского “белого” шума. Параметр a позволяет установить требуемую ширину полосы: чем больше величина этого параметра, тем более узкой является полоса пропускания фильтра первого порядка.

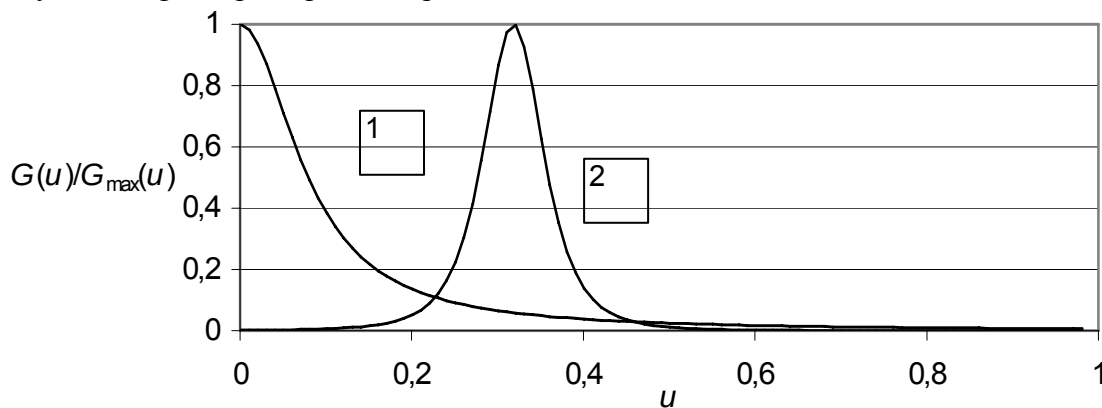


Рис. 1. Полоса пропускания фильтра Калмана первого (кривая 1) и второго (кривая 2) порядков для спектральной плотности стохастических сигналов в зависимости от нормированной частоты

На рис. 2, а представлен пример реализации стохастического процесса в виде акустического сигнала, содержащего локальные участки с повышенными частотами на фоне медленно изменяющейся фоновой составляющей сигнала. Обработка сигналов такого вида при помощи традиционных методов спектрального анализа, как правило, не обеспечивает требуемого спектрального разрешения на локальных участках.

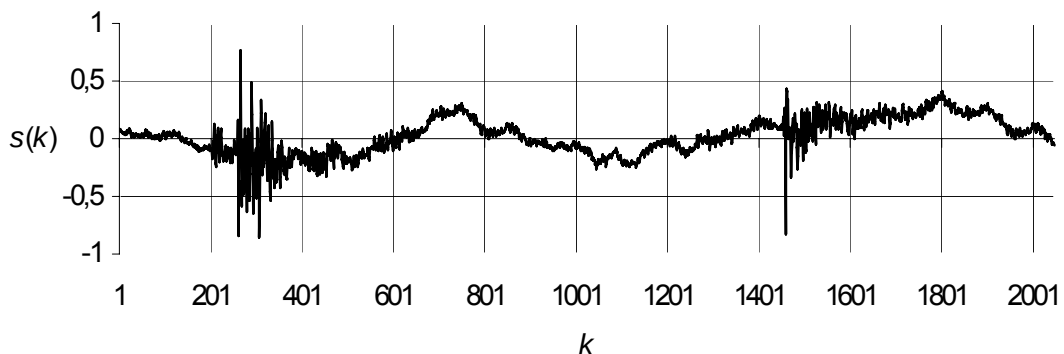


Рис 2, а. Нормированный исходный сигнал

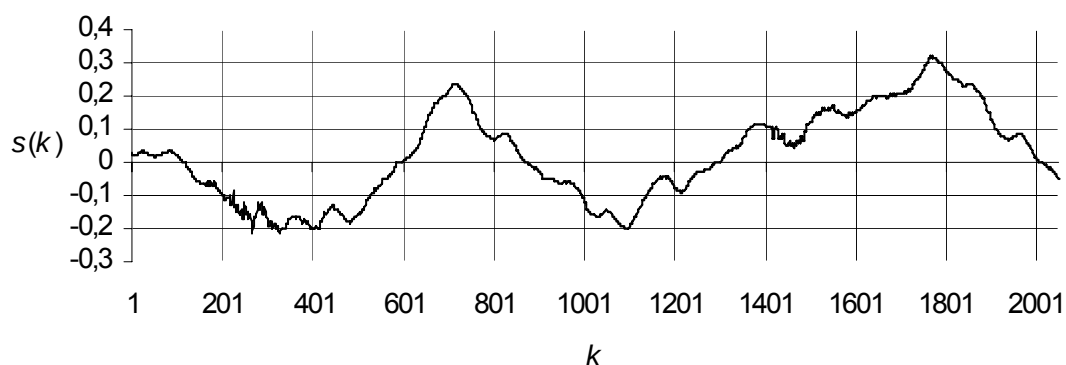


Рис 2, б. Сигнал, полученный на выходе фильтра Калмана первого порядка

Результат выделения низкочастотных составляющих сигнала фильтром Калмана первого порядка представлен на рис. 2, б. Следует отметить, что фильтр Калмана обеспечивает динамическое формирование на его выходе сигнала, наиболее соответствующего заданной кривой спектральной плотности (12) в смысле минимума средней квадратической ошибки, из всех возможных реализаций случайного процесса.

Очевидно, что с помощью фильтра первого порядка возможно также выделение высокочастотных составляющих сигнала путем вычитания из исходного сигнала предварительно отфильтрованных низкочастотных составляющих. Пример обработки этого вида применительно к акустическим сигналам приведен на рис. 3.

Кривые спектров рис. 3 наглядно иллюстрируют результат подавления спектральных составляющих в области нижних частот. Спектры $S(u)$ вычислялись при помощи обычного алгоритма дискретного преобразования Фурье. Для большей наглядности кривые зависимостей модулей амплитудных спектров от значений дискретной частоты u представлены на рис. 3 в логарифмическом масштабе.

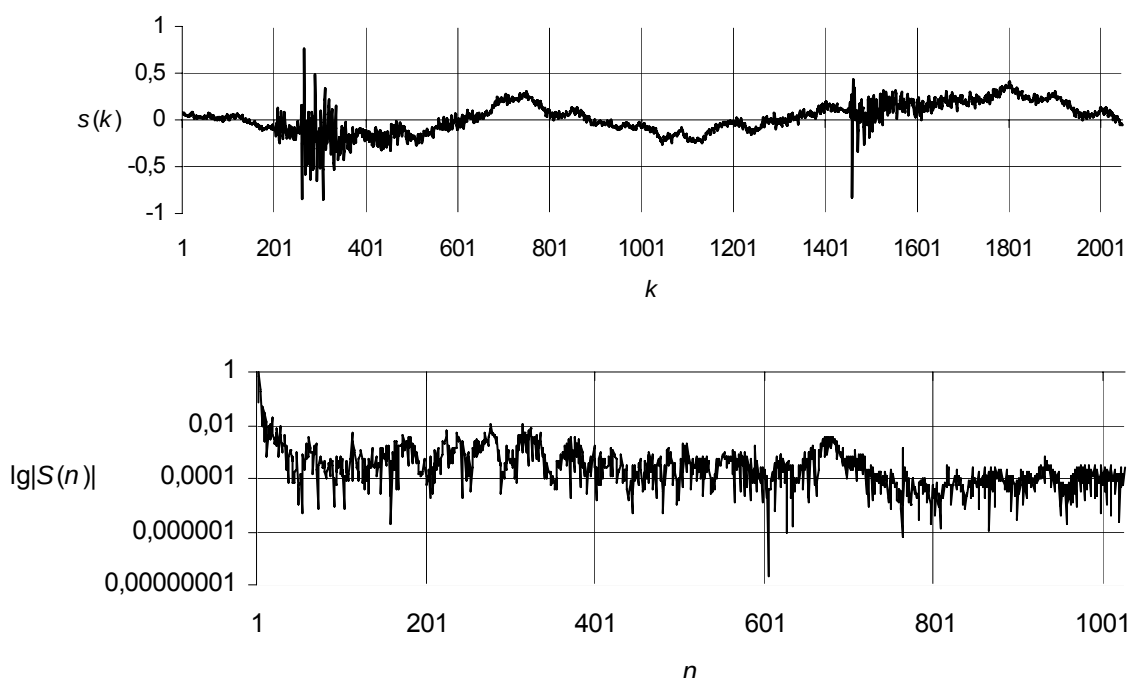


Рис 3, а. Исходный акустический сигнал (вверху) и его спектр

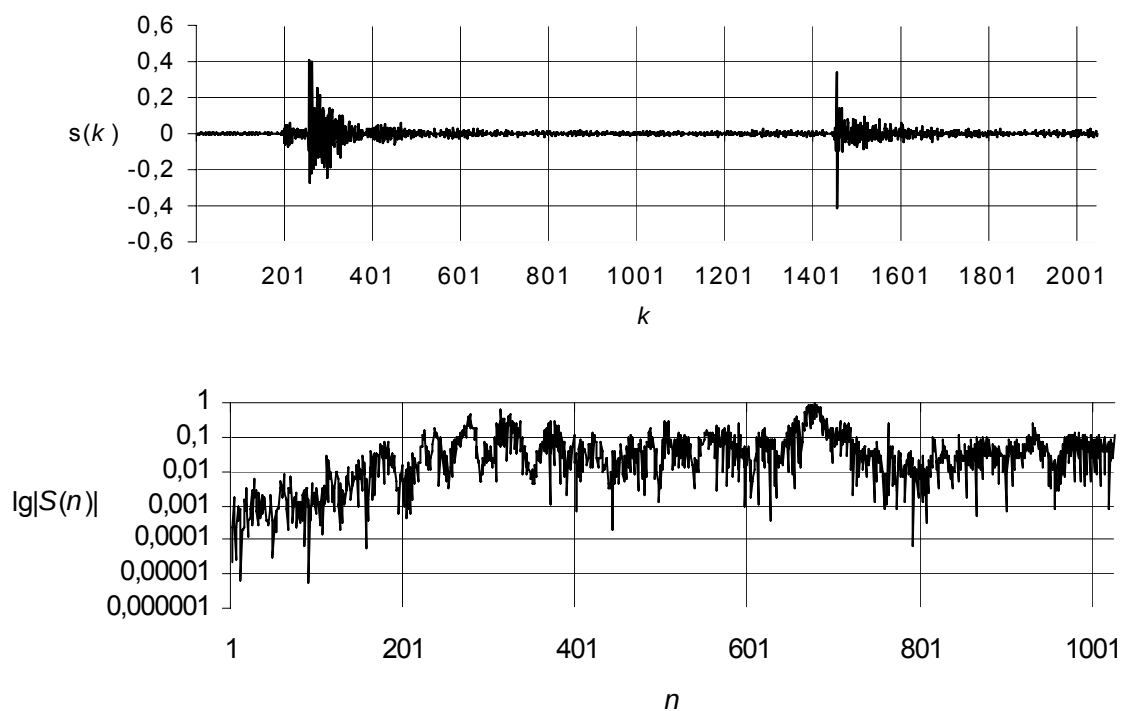


Рис 3, б. Результат выделения высокочастотных составляющих сигнала с использованием фильтра Калмана первого порядка

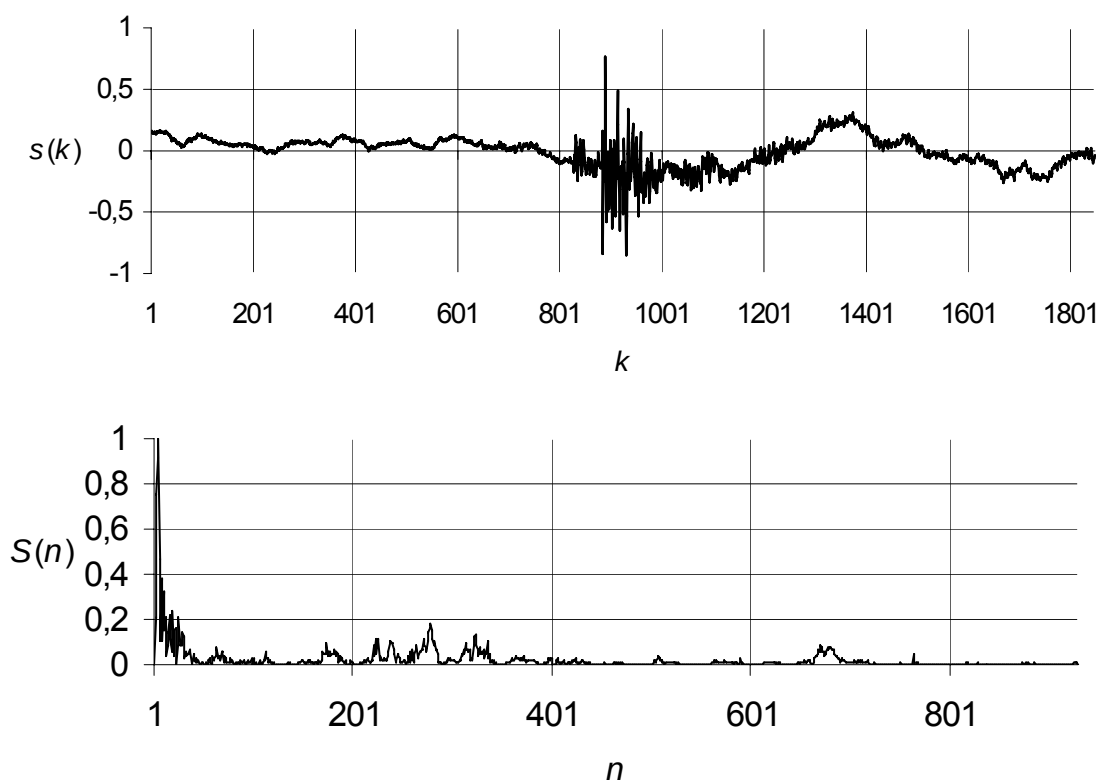


Рис. 4. Сигнал на входе фильтра второго порядка (вверху) и его спектр

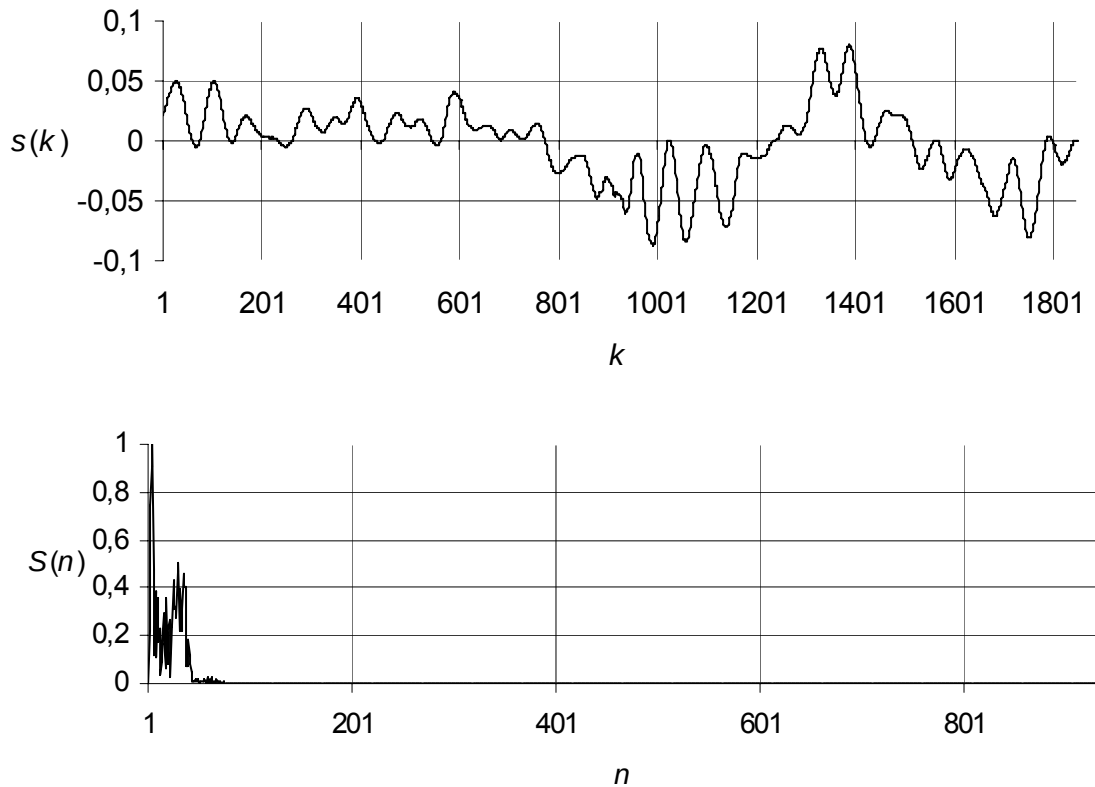


Рис 5. Составляющая сигнала рис. 4 в узкой полосе частот, полученная при обработке данных дискретным фильтром Калмана второго порядка, и ее спектр

Частотная характеристика фильтра Калмана второго порядка определяется выражением (11). Таким образом, фильтр второго порядка осуществляет обработку стохастического сигнала подобно колебательному контуру, возбуждаемому случайным шумом. Центральная частота спектра мощности сигнала на выходе фильтра второго порядка определяется параметром b , при этом ширина спектра, как отмечалось выше, устанавливается при помощи параметра a в (11). Следует отметить, что при обработке сигналов важным фактором является обеспечение достаточно высокой частоты дискретизации для выполнения условий теоремы отсчетов. В противном случае нарушается устойчивость рекуррентных вычислений, используемых в дискретных фильтрах Калмана.

На рис. 4 представлен пример реализации акустического сигнала. На рис. 5 показаны результаты фильтрации этого сигнала, полученные при помощи фильтра Калмана второго порядка.

Отметим, что на рисунке показан модуль амплитудного спектра отдельной реализации сигнала. При усреднении по ансамблю реализаций случайного процесса в заданной полосе частот несложно получить кривую, по форме совпадающую с кривой 2 на рис. 1. При перестройке параметров фильтра возможен динамический анализ сигналов в различных частотных полосах, в том числе при адаптивной подстройке параметров к характеристикам стохастических сигналов.

Заключение

Фильтры Калмана низких порядков позволяют осуществлять динамическую обработку стохастических сигналов при незначительных вычислительных затратах, что особенно важно при анализе акустических сигналов в реальном времени. Фильтр Калмана первого порядка удобен для выделения низких частот в заданном диапазоне

спектра мощности, фильтр второго порядка обеспечивает выделение узкополосных стохастических сигналов в заданной полосе частот.

Проведенные исследования подтвердили эффективность рекуррентных алгоритмов реализации дискретных фильтров Калмана при обработке экспериментально зарегистрированных нестационарных акустических сигналов. Использование фильтров Калмана обеспечивает возможность идентификации локальных изменений сигнала на участках небольшой длительности, которая затруднена при использовании классических методов спектрального анализа.

Литература

1. Марпл С.Л. Цифровой спектральный анализ и его приложения. М.: Мир, 1990.
2. Бендат Дж., Пирсол А. Применения корреляционного и спектрального анализа. М.: Мир, 1983.
3. Васильев В.Н., Гуров И.П. Компьютерная обработка сигналов в приложении к интерферометрическим системам. СПб: БХВ - Санкт-Петербург, 1998.
4. Справочник по прикладной статистике / Под ред. Э. Ллойда, У. Ледермана. М.: Мир, 1990. Т. 2.

МАТЕМАТИКО-СТАТИСТИЧЕСКИЕ МЕТОДЫ В ЗАДАЧЕ ПАРАМЕТРИЗАЦИИ ОПИСАНИЯ КЛАССОВ ЭЭГ-СИГНАЛА

И.О. Жаринов

Использование для анализа сигналов электроэнцефалограммы (ЭЭГ) человека параметрических моделей авторегрессии-скользящего среднего (АРСС), являющихся косвенным описанием признаков классов ЭЭГ-сигналов, находится сейчас в стадии интенсивного развития: совершенствуются вычислительные методы, уточняются статистические аспекты, выясняются границы применимости АРСС анализа.

Основной задачей в проблемах анализа и классификации ЭЭГ является задача выбора признаков классов, которая может быть решена с использованием прямого и косвенного описаний [7].

Прямое описание основано на вероятностных мерах — характеристиках многомерных распределений случайных процессов, моментных характеристиках, оценках корреляционных функций и может быть использовано [2] для анализа и классификации стационарных и нестационарных сигналов.

Косвенное описание использует [6,9] информацию о признаках классов сигналов опосредованно заложенную в коэффициенты их динамических моделей, представленных в виде алгебраических, дифференциальных и интегральных уравнений, а также их дискретными эталонами.

Основной недостаток прямого описания признаков ЭЭГ-сигналов и их классов заключается в необходимости проведения сравнительно длительного статистического эксперимента для его формирования на заключительном этапе — этапе распознавания.

Косвенное описание может быть использовано для коротких выборок стационарных процессов, так как соответствующая статистическая информация заложена в параметры их математических (динамических) моделей на предварительном этапе — этапе обучения.

АРСС анализ базируется на предположении [1,6,8], что текущие значения ЭЭГ имеют существенную статистическую связь с ее предысторией. АРСС модель i -ого квазистационарного участка ЭЭГ представляет значения отсчетов ЭЭГ-сигнала $y^{(i)}[n]$ посредством линейного соотношения вида

$$y^{(i)}[n] + \sum_{k=1}^{p^{(i)}} a_k^{(i)} \cdot y^{(i)}[n-k] = b_0^{(i)} \cdot x[n] + \sum_{k=1}^{q^{(i)}} b_k^{(i)} \cdot x[n-k], \quad i=1,2,\dots,M,$$

где $\{x[n]\}$ - последовательность независимых, нормально распределенных случайных величин с нулевым математическим ожиданием и единичной дисперсией. Эта последовательность называется порождающим процессом. Коэффициенты модели $\{a_1^{(i)}, a_2^{(i)}, \dots, a_{p^{(i)}}^{(i)}\}$ и $\{b_0^{(i)}, b_1^{(i)}, \dots, b_{q^{(i)}}^{(i)}\}$, а также величины $p^{(i)}$ и $q^{(i)}$ являются параметрами модели АРСС. Параметры $p^{(i)}$ и $q^{(i)}$ определяют порядок модели авторегрессии и скользящего среднего соответственно, а величина $(p^{(i)} + q^{(i)})$ определяет порядок АРСС модели. Часто адекватную параметрическую авторегрессионную модель ЭЭГ удается получить, даже полагая коэффициенты $b_1^{(i)}, \dots, b_{q^{(i)}}^{(i)}$ равными нулю

$$y^{(i)}[n] + \sum_{k=1}^{p^{(i)}} a_k^{(i)} \cdot y^{(i)}[n-k] = b_0^{(i)} \cdot x[n], \quad i=1,2,\dots,M.$$

Индекс (i) означает описание моделью i -ого класса из M возможных взаимоальтернативных классов ЭЭС. В последующем, через параметры АРСС модели могут быть выражены оценки частотных и корреляционных функций, определены отрезки стационарности ЭЭГ, решены задачи сегментации и классификации ЭЭГ

[3,6,10]. В частности, при известных параметрах АРСС модели можно получить [10] оценку спектральной плотности i -ого класса ЭЭС.

Оценивание параметров линейного уравнения регрессии по имеющимся данным $\{y[n^*]\}_{n=N}^n$ представляет собой одну из основных процедур прикладного статистического анализа. Для этой цели широко применяется метод наименьших квадратов [5], всесторонне изученный и имеющий несколько теоретических обоснований. Он достаточно просто реализуется в виде специализированных программ расчетов оценок параметров и других характеристик линейной регрессионной связи между объясняющими и объясняемыми переменными. Такие программы для небольшого числа оцениваемых параметров и не очень большого числа используемых наблюдений имеются сегодня в прикладном программном обеспечении любого компьютера.

Удачным примером подхода, сочетающего систематическое изложение теории, математических методов и алгоритмов является подход, основанный на применении традиционных методов анализа стационарных процессов к формированию новых по постановке задачи синтеза признаков классов ЭЭГ-сигналов.

Задача формирования косвенного описания признаков классов ЭЭГ может рассматриваться как известная задача подбора модели определенного класса, адаптированная к стационарным или квазистационарным случайным процессам. Такую задачу принято [7] решать в четыре этапа: концептуальный выбор класса моделей (выбор подходящего описания в пространстве параметрических моделей (использование АР или АРСС моделей)); идентификация порядка модели (определение порядка АР или АРСС модели); оценивание параметров модели (получение оценок параметров и остаточной дисперсии при известном порядке модели); подтверждение качества модели (проверка пригодности (адекватности) полученного описания).

Формирование параметрических описаний признаков классов ЭЭГ базируется на использовании [9,10] метода Юла-Уолкера, метода факторизации спектральной плотности, метода уравнивания z -преобразований текущих АКФ, метода Берга, ковариационных методов и т.д.

Для оценки качественных показателей математических методов косвенного описания реализаций классов ЭЭГ-сигнала была проведена серия статистических экспериментов по обработке реальных данных ЭЭГ $\{y[n^*]\}_{n=N}^n$. Обработка производилась не в реальном масштабе времени на IBM-совместимом ПК при помощи специализированного авторского программного обеспечения, реализующего несколько наиболее распространенных алгоритмических методов получения оценок авторегрессионных компонентов параметрических АР и АРСС моделей по дискретным данным.

Исследования проводились на базе цифровых методов обработки, относящихся к блокам данных, т.е. алгоритмами, предназначенными для обработки целых блоков (последовательностей) накопленных дискретных отсчетов $\{y[n^*]\}_{n=N}^n$ данных некоторого фиксированного объема N . Рассматриваемые методы блочной обработки данных можно кратко описать как алгоритмы с фиксированным временем, рекурсивные относительно порядка формируемых моделей в том смысле, что они применяются к блокам временных дискретных отсчетов $\{y[n^*]\}_{n=N}^n$ фиксированного объема и позволяют рекуррентным образом получать оценки параметров АР моделей более высокого порядка по оценкам параметров АР модели более низкого порядка. Такие алгоритмы целесообразно применять в тех случаях, когда порядок требуемой АР модели заранее не известен, поэтому для выбора АР модели надлежащего порядка необходимо испытывать много таких моделей различных порядков и сравнивать получаемые результаты.

Для численного определения точностных характеристик различных методов формирования косвенного описания классов ЭЭС на базе эталонной АР модели α -ритма методом математического моделирования на ЭВМ [4] была синтезирована последовательность дискретных отсчетов, по которой, впоследствии, определялись оценки весовых коэффициентов АР моделей разными методами для одной и той же последовательности $x[n]$ – нормального порождающего БШ.

Серии статистических экспериментов показали, что параметрические модели, полученные рассматриваемыми методами формирования косвенного описания, обладают приемлемым качеством аппроксимации при сравнительно высоких объемах N анализируемых данных ЭЭГ $\{y[n^*]\}_{n=N}^n$. Начиная с $N \geq 100$, значение ошибки определения коэффициента оказывается инвариантно к методу формирования модели и сходится к оптимальному, нижняя граница дисперсии оценки которого определена в соответствии с неравенством Рао-Крамера [8]. Экспериментально подтверждается, что в случае аппроксимации реализаций $\{y[n^*]\}_{n=N}^n$ классов ЭЭС параметрическими моделями авторегрессии обратная функциональная зависимость величины $\sigma_{a_1^{(\alpha-р.м.)}}$ от величины $\sqrt{N}$, справедливая для классического авторегрессионного анализа, сохраняется при обработке дискретных данных ЭЭГ $\{y[n^*]\}_{n=N}^n$.

Литература.

1. Айвазян С.А. и др. Прикладная статистика: Исследование зависимостей: Справ. изд. / С.А. Айвазян, И.С. Енюков, Л.Д. Мешалкин; / Под ред. С.А. Айвазяна. М.: Финансы и статистика, 1985. 487с., ил.
2. Бендат Дж., Пирсол А. Применение корреляционного и спектрального анализа: /Пер. с англ. М.: Мир, 1983. 312с., ил.
3. Бокс Дж., Дженкинс Г. Анализ временных рядов. Прогноз и управление: / Пер. с англ. М.: Мир, 1974. 408с.
4. Быков В.В. Цифровое моделирование в статистической радиотехнике. М.: Советское радио, 1971. 328с.
5. Губанов В.С. Обобщенный метод наименьших квадратов. Теория и применение в астронометрии. СПб.: Наука, 1997. 318с., ил.
6. Дженкинс Г., Ваттс Д. Спектральный анализ и его приложения. Вып.2М.: Мир, 1972. 287с.
7. Елисеев А.А., Зиятдинов С.И., Изранцев В.В. и др. Управление движущимися объектами: Учеб. пособие / Под ред. А.А. Елисеева и А.А. Оводенко. М.: Изд-во МГАП "Мир книги", 1994. 427с.: ил.
8. Кашьяп Р.Л., Рао А.Р. Построение динамических стохастических моделей по экспериментальным данным: Пер. с англ. М.: Наука, 1983. 384с.
9. Кей С.М., Марпл С.Л. Современные методы спектрального анализа: Обзор. // ТИИЭР. 1981. № 11. Т.69. С.5–51.
10. Марпл.-мл. С.Л. Цифровой спектральный анализ и его приложения: / Пер. с англ. М.: Мир, 1990. 584с., ил.

СВЯЗЬ АЛГЕБРАИЧЕСКИХ СПЕКТРОВ СОБСТВЕННЫХ ЗНАЧЕНИЙ И СИНГУЛЯРНЫХ ЧИСЕЛ В ЗАДАЧЕ ОБЕСПЕЧЕНИЯ СТАБИЛЬНОСТИ ЭЛЛИПСОИДНЫХ ПОКАЗАТЕЛЕЙ КАЧЕСТВА

Т.А. Акунов, А.В. Ушаков

В работе ставится задача установления связи алгебраических спектров собственных значений и сингулярных чисел матрицы состояния динамической системы и ее фундаментальной матрицы. Задача возникает на стыке методов синтеза систем с использованием обобщенного модального управления и оценки качества этих систем с помощью эллипсоидных показателей. Решение задачи сориентировано на проблему обеспечения стабильности эллипсоидных показателей качества посредством обеспечения стабильности собственных значений и собственных векторов матрицы состояния системы.

Введение. Постановка задачи

В настоящее время одним из конструктивных методов синтеза многомерных систем управления, инвариантных относительно размерности вектора входа, является метод модального управления. Его обобщенная версия [1,2], ставящая задачу обеспечения как желаемой структуры собственных значений, так собственных векторов, позволяет доставлять собственным значениям гарантированную стабильность в условиях вариаций или неопределенности параметров матричных компонентов модельного описания исходного объекта. Одновременно в теории многомерных систем типа многомерный вход-выход для оценки качества процессов в последнее время интенсивно разрабатывается аппарат эллипсоидных показателей. Эллипсоидные показатели (оценки) [3] качества представляют собой экстремальные элементы алгебраического спектра сингулярных чисел критериальной матрицы исследуемой системы своей для каждой версии изучаемых процессов (матрицы состояния, грамины управляемости и наблюдаемости, кросс-грамины, фундаментальная и переходная матрицы, матрица ковариаций и т.д.). В этой связи, если ставится задача синтеза многомерных систем управления с эллипсоидными показателями качества гарантированной стабильности, алгоритмически опирающаяся на возможности модального управления, возникает необходимость установления связи между алгебраическими спектрами собственных значений и сингулярных чисел критериальных матриц, банк которых в настоящей работе представлен матрицей состояния динамической системы, ее фундаментальной и переходной матрицами.

Конструирование матриц связей алгебраических спектров собственных значений и сингулярных чисел

Решение поставленной задачи осуществим на примере непрерывной многомерной системы

$$\dot{x}(t) = Fx(t) + Gg(t); x(0), y(t) = Cx(t); \quad (1)$$

полученной агрегированием исходного объекта управления

$$\dot{x}(t) = Ax(t) + Bu(t); x(0), y(t) = Cx(t); \quad (2)$$

и регулятора, реализованного в виде прямой связи по экзогенному воздействию и обратной связи по состоянию

$$u(t) = K_g g(t) + Kx(t). \quad (3)$$

В (1)–(3) x, u, y, g соответственно векторы состояния, управления (входа), выхода и экзогенного воздействия, $x \in R^n, y \in R^m, u \in R^r$; F, G, C, A, B, K_g, K соответственно матрицы состояния системы, входа, выхода, матрицы состояния ОУ, управления, прямой связи по экзогенному воздействию и обратной связи по состоянию, $F, A \in R^{n \times n}, G, C^T \in R^{n \times m}, B, K^T \in R^{n \times r}, K_g \in R^{r \times m}$.

Основные результаты изложены в виде системы утверждений.

Утверждение 1. Векторы $\alpha = \text{col}\{\alpha_i; i = \overline{1, n}\}$ и $\lambda = \text{col}\{\lambda_i; i = \overline{1, n}\}$ составленные из сингулярных чисел α_i и собственных значений λ_i матрицы F состояния системы (1) связаны векторно-матричным соотношением

$$\alpha = \Pi_{\alpha\lambda} \lambda, \quad (4)$$

где матрица связей задается соотношением

$$\Pi_{\alpha\lambda} = \text{col}\{(U_i)^T M \text{diag}\{(M^{-1}V_i)^j; j = \overline{1, n}\}; i = \overline{1, n}\}, \quad (5)$$

в котором U и V соответственно матрицы левого и правого сингулярного базисов сингулярного разложения матрицы F , $(\circ)_i$ – i -ый столбец, $(\circ)^i$ – i -ая строка соответствующих матричных компонентов, M и Λ – матрицы собственных векторов и собственных чисел матрицы F . $\square$

Доказательство. Применим к матрице F процедуру ее сингулярного разложения и процедуру приведения к диагональному виду с помощью матрицы собственных векторов. Тогда для матрицы F получим два представления

$$F = U \Sigma V^T, \quad (6)$$

$$F = M \Lambda M^{-1}, \quad (7)$$

где

$$\Sigma = \text{diag}\{\alpha_i; i = \overline{1, n}\}, \Lambda = \text{diag}\{\lambda_i; i = \overline{1, n}\}. \quad (8)$$

Разрешим соотношение (6) относительно матрицы Σ сингулярных чисел в форме

$$\Sigma = U^T F V. \quad (9)$$

В силу (8) и (9) для сингулярного числа α_i как (ii) -го элемента матрицы Σ можно записать

$$\alpha_i = (U^T)^i F V_i = (U_i)^T F V_i. \quad (10)$$

Связь α_i с элементами λ_i алгебраического спектра собственных значений матрицы F может быть получена, если в (10) подставить (7) и с учетом (8) воспользоваться справедливостью соотношения

$$\text{diag}\{(\circ)_i; i = \overline{1, n}\} \text{col}\{(\circ)^i; i = \overline{1, n}\} = \text{diag}\{(\circ)^i; i = \overline{1, n}\} \text{col}\{(\circ)_i; i = \overline{1, n}\}, \quad (11)$$

которые позволяют записать

$$\alpha_i = (U_i)^T M \text{diag}\{(M^{-1}V_i)^j; j = \overline{1, n}\} \lambda. \quad (12)$$

Формирование столбца α из элементов (12) дает (4) с матрицей (5). $\blacksquare$

Степень близости матрицы (5) к диагональной или сигнатурной матрице определяется степенью близости согласованных элементов левого сингулярного базиса и собственных векторов матрицы F . По мере роста отличия геометрического спектра матрицы F от левого сингулярного базиса увеличивается отличие матрицы $\Pi_{\alpha\lambda}$ от диагонального вида.

Для переноса полученных результатов на случай матричной функции $f(F)$ от матрицы F напомним, что она порождается рядом [5] по степеням скалярной переменной ϑ

$$f(\vartheta) = a_0 + a_1\vartheta + a_2\vartheta^2 + \dots + a_p\vartheta^p + \dots, \quad a_i \in R$$

в форме

$$f(F) = a_0I + a_1F + a_2F^2 + \dots + a_pF^p + \dots.$$

Матричная функция $f(F)$ от матрицы F сохраняет отношение подобия в форме (7) $f(F)M = Mf(\Lambda)$ и геометрический спектр собственных векторов, причем ее собственные значения $\{f(\lambda_i), i = \overline{1, n}\}$ определяются функцией вида $f(\vartheta)$ на спектре $\{\lambda_i, i = \overline{1, n}\}$ матрицы F . К сожалению, матричная функция $f(F)$ от матрицы F не сохраняет геометрические спектры сингулярных базисов и не обладает в общем случае указанной функциональной связью алгебраических спектров сингулярных чисел. Ниже рассматриваются параметризованные непрерывным временем t версии матричной функции $f(F) = f(F, t)$, представленных фундаментальной матрицей $f(F) = \exp(Ft)$ и переходной матрицей $f(F) = F^{-1}(\exp(Ft) - I)G$ системы (1).

Утверждение 2. Векторы $\alpha_f = \text{col}\{\alpha_{fi}; i = \overline{1, n}\}$ и $\lambda_f = \text{col}\{\lambda_{fi}; i = \overline{1, n}\}$ составленные из сингулярных чисел α_{fi} и собственных значений λ_{fi} матричной функции $f(F)$ от матрицы F , $f(F): R^{n \times n} \Rightarrow R^{n \times n}$ связаны векторно-матричным соотношением

$$\alpha_f = \Pi_{\alpha\lambda} \lambda_f, \quad (13)$$

где матрица связей задается соотношением

$$\Pi_f = \text{col}\left\{\left(U_f\right)^T M \text{diag}\left\{M^{-1}V_{fi}\right\}^j; j = \overline{1, n}\right\}; i = \overline{1, n} \quad (14)$$

в котором U_f и V_f соответственно матрицы левого и правого сингулярного базисов в сингулярном разложении матричной функции $f(F)$ от матрицы F $\square$

Доказательство. Применим к матричной функции $f(F)$ от матрицы F процедуру сингулярного разложения и процедуру приведения к диагональному виду с помощью матрицы собственных векторов матрицы F , используя свойство матричной функции от матрицы сохранять отношение подобия в форме $f(F)M = Mf(\Lambda)$, где матрицы F, Λ, M удовлетворяют (7)

Тогда для матричной функции $f(F)$ от матрицы F получим два представления

$$f(F) = U_f \Sigma_f V_f^T, \quad (15)$$

$$f(F) = Mf(\Lambda)M^{-1}, \quad (16)$$

где

$$\Sigma_f = \text{diag}\{\alpha_{fi}, i = \overline{1, n}\}, \quad f(\Lambda) = \text{diag}\{\lambda_{fi}, i = \overline{1, n}\}. \quad (17)$$

Разрешим соотношение (15) относительно матрицы Σ_f сингулярных чисел матричной функции $f(F)$ от матрицы F в форме

$$\Sigma_f = U_f^T f(F) V_f. \quad (18)$$

В силу (17) и (18) для сингулярного числа α_{fi} как (ii) -го элемента матрицы Σ_f можно записать

$$\alpha_{fi} = (U_{fi})^T f(F) V_{fi}. \quad (19)$$

Связь α_{fi} с элементами λ_{fi} алгебраического спектра собственных значений матричной функции $f(F)$ от матрицы F может быть получена, если в (19) подставить (16) и с учетом (17) воспользоваться справедливостью соотношения (11), которые позволяют записать

$$\alpha_{\bar{f}_i} = (U_{\bar{f}_i})^T M \text{diag} \left\{ (M^{-1} V_{\bar{f}_i})^j; j = \overline{1, n} \right\} \lambda_{\bar{f}_i}. \quad (20)$$

Формирование столбца $\alpha_{\bar{f}}$ из элементов (20) дает (13) с матрицей (14). ■

Основные результаты.

Установленная связь алгебраических спектров собственных значений и сингулярных чисел матрицы состояния динамической систем, ее фундаментальной и переходной матриц используется в дальнейшем для решения поставленной задачи обеспечения стабильности эллипсоидных показателей качества. При этом предполагается, что параметрические вариации таковы, что применимы методы теории чувствительности в рамках функций чувствительности первого порядка. Существуют решения задачи анализа параметрической чувствительности собственных значений [1,4] и сингулярных чисел [3] в раздельном виде. В данной работе устанавливается связь функций чувствительности собственных значений и сингулярных чисел и показывается, что контролем чувствительности собственных значений, а также собственных векторов может быть обеспечена стабильность эллипсоидных показателей качества.

Пусть матрица F состояния системы (1) и как следствие матричная функция $f(F)$ от матрицы F зависят от вектора параметров $q \in R^p$ с номинальным значением q_0 в форме $F(q)$ и $f\{F(q)\}$. При этом алгебраические спектры собственных значений и сингулярных чисел и геометрические спектры собственных векторов и сингулярных базисов также обнаруживают зависимость от q так, что для $q \neq q_0$ оказываются справедливыми представления

$$F(q) = U(q) \Sigma(q) V^T(q), \quad (21)$$

$$F(q) = M(q) \Lambda(q) M^{-1}(q), \quad (22)$$

где

$$\Sigma(q) = \text{diag} \{ \alpha_i(q), i = \overline{1, n} \}, \quad \Lambda(q) = \text{diag} \{ \lambda_i(q), i = \overline{1, n} \}. \quad (23)$$

$$\alpha_i(q) = \alpha_i + \Delta \alpha_i(q_0, \Delta q), \quad \lambda_i(q) = \lambda_i + \Delta \lambda_i(q_0, \Delta q). \quad (24)$$

Конечные приращения сингулярных чисел $\Delta \alpha_i$ и собственных чисел $\Delta \lambda_i$ с использованием соответствующих функций чувствительности первого порядка α_{iq_k} и λ_{iq_k} к вариациям k -го элемента q_k вектора параметров q определяются с помощью соотношений

$$\Delta \alpha_i(q_0, \Delta q_k) = \alpha_{iq_k} \Delta q_k, \quad \Delta \lambda_i(q_0, \Delta q_k) = \lambda_{iq_k} \Delta q_k, \quad (25)$$

позволяющих записать для вариаций сингулярных чисел $\Delta \alpha_i(q_0, \Delta q)$ и собственных чисел $\Delta \lambda_i(q_0, \Delta q)$ вызванных вариаций всех элементов Δq_k , $k = \overline{1, p}$ вектора параметров Δq , используя обозначения $(\circ(q))|_{q=q_0} = (\circ)$, $(\circ(q))_{q_k} = \frac{\partial}{\partial q_k} (\circ(q))|_{q=q_0}$.

$$\Delta \alpha_i(q_0, \Delta q) = \text{row} \{ \alpha_{iq_k}; k = \overline{1, p} \} \Delta q, \quad \Delta \lambda_i(q_0, \Delta q) = \text{row} \{ \lambda_{iq_k}; k = \overline{1, p} \} \Delta q, \quad (26)$$

Вычисление функций чувствительности α_{iq_k} и λ_{iq_k} осуществляется с помощью положений следующих утверждений.

Утверждение 3. Функции чувствительности собственных значений λ_{q_k} и сингулярных чисел α_{q_k} связаны соотношением

$$\alpha_{q_k} = \text{col} \{ \Pi^i_{\alpha \lambda q_k} \lambda + \Pi^i_{\alpha \lambda} \lambda_{q_k}; i = \overline{1, n} \}, \quad (27)$$

где матрица-строка $\Pi^i_{\alpha \lambda q_k}$ определяется с помощью соотношения

$$\begin{aligned} \Pi^i \alpha_{\lambda_{q_k}} = & (U_i)^T M \text{diag} \left\{ (M^{-1} V_i)^j; j = \overline{1, n} \right\} + (U_i)^T M_{q_k} \text{diag} \left\{ (M^{-1} V_i)^j; j = \overline{1, n} \right\} - \\ & - (U_i)^T M \text{diag} \left\{ (M^{-1} M_{q_k} M^{-1} V_i)^j; j = \overline{1, n} \right\} + (U_i)^T M \text{diag} \left\{ (M^{-1} V_{i_{q_k}})^j; j = \overline{1, n} \right\} \end{aligned} \quad (28)$$

а вектор из функций чувствительности собственных значений λ_{q_k} к вариациям k -го элемента q_k вектора параметров q определяется [1] с помощью соотношения

$$\lambda_{q_k} = \text{col} \left\{ (M^{-1} F_{q_k} M)_{ii}; i = \overline{1, n} \right\}. \quad \square \quad (29)$$

Доказательство основано на прямом дифференцировании (20) с учетом справедливости соотношения $(U^T)^i = (U_i)^T$ и способа дифференцировании обратной матрицы в форме $(M^{-1})_{q_k} = -(M^{-1}) M_{q_k} M^{-1}$. ■

Аналогичным образом может быть доказано следующее утверждение относительно связи функции чувствительности собственных значений $\lambda_{f_{q_k}}$ и сингулярных чисел $\alpha_{f_{q_k}}$ матричной функции $f(F)$ от матрицы F .

Утверждение 4. Функции чувствительности собственных значений $\lambda_{f_{q_k}}$ и сингулярных чисел $\alpha_{f_{q_k}}$ матричной функции $f(F)$ от матрицы F связаны соотношением

$$\alpha_{f_{q_k}} = \text{col} \left\{ \left(\Pi^i_{f_{q_k}} \lambda_f + \Pi^i_f \frac{\partial \lambda_f}{\partial \lambda_i} \lambda_{i_{q_k}} \right)_{|q=q_0}; i = \overline{1, n} \right\} \quad (30)$$

где матрица-строка $\Pi^i_{f_{q_k}}$ определяется с помощью соотношения

$$\begin{aligned} \Pi^i_{f_{q_k}} = & (U_{f_i})^T M \text{diag} \left\{ (M^{-1} V_{f_i})^j; j = \overline{1, n} \right\} + (U_{f_i})^T M_{q_k} \text{diag} \left\{ (M^{-1} V_{f_i})^j; j = \overline{1, n} \right\} - \\ & - (U_{f_i})^T M \text{diag} \left\{ (M^{-1} M_{q_k} M^{-1} V_{f_i})^j; j = \overline{1, n} \right\} + (U_{f_i})^T M \text{diag} \left\{ (M^{-1} V_{f_{i_{q_k}}})^j; j = \overline{1, n} \right\} \end{aligned} \quad (31)$$

а вектор из функций чувствительности собственных значений $\lambda_{f_{q_k}}$ матричной функции $f(F)$ от матрицы F к вариациям k -го элемента q_k вектора параметров q определяется [1] с помощью соотношения

$$\lambda_{f_{q_k}} = \text{col} \left\{ [M^{-1} f_{q_k}(F) M]_{ii}, i = \overline{1, n} \right\}, \quad (32)$$

где $f_{q_k}(F) = M_{q_k} M^{-1} f(F) - f(F) M_{q_k} M^{-1} + M \text{diag} \left\{ \frac{\partial f(\lambda_i)}{\partial \lambda_i} \lambda_{i_{q_k}}, i = \overline{1, n} \right\} M^{-1}$. ■

В выражениях (28), (31) вычисления функций чувствительности собственных векторов и сингулярных базисов соответствующих матриц производится с помощью соотношений [1,3,4]

Утверждение 5. Функции чувствительности $M_{i_{q_k}}$ собственных векторов матрицы F состояния системы (1), $U_{i_{q_k}}$ и $V_{i_{q_k}}$ сингулярных базисов $U(q)$ и $V(q)$ к вариациям k -го элемента q_k вектора параметров q могут быть вычислены с помощью соотношений

$$M_{i_{q_k}} = \sum_{\substack{l=1 \\ l \neq i}}^n \delta_{il}^k M_l \quad (33)$$

$$\delta_{il}^k = \frac{(M^{-1})^l F_{q_k} M_i}{\lambda_i - \lambda_l}; \quad i \neq l; \quad i, l = \overline{1, n}; \quad k = \overline{1, p} \quad (34)$$

$$U_{i_{q_l}} = \sum_{l=1}^v \gamma_{il}^k U_l; \quad V_{i_{q_k}} = \sum_{l=1}^v \beta_{il}^k V_l; \quad k = \overline{1, p}; \quad i = \overline{1, v}; \quad (35)$$

где

$$\gamma_{il}^j = \frac{\alpha_i (U^T)^j \Pi_{q_k} V_i + \alpha_{lk} (U^T)^j \Pi_{q_k} V_l}{\alpha_i^2 - \alpha_l^2}; \quad i \neq l; \quad \gamma_{il}^k = 0; \quad i, l = \overline{1, n}; \quad k = \overline{1, p}, \quad (36)$$

$$\beta_{il}^k = \frac{\alpha_i (U^T)^i \Pi_{q_k} V_l + \alpha_k (U^T)^j \Pi_{q_k} V_i}{\alpha_i^2 - \alpha_l^2}; \quad i \neq l; \quad \beta_{ii} = 0; \quad i, l = \overline{1, n}; \quad k = \overline{1, p} \quad \square \quad (37)$$

Доказательство утверждения 5 приведено в [3].

Пример. Требуется определить связь алгебраических спектров собственных значений и сингулярных чисел для фундаментальной матрицы $f(F) = \exp(Ft)$ системы (1) с матрицей состояния

$$F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -80 & -66 & -15 \end{bmatrix},$$

имеющей спектр собственных значений $\{\lambda_1 = -2; \lambda_2 = -5; \lambda_3 = -8\}$.

Матрица связи Π_f для моментов времени $t=0; 0.13; 1.49$ сек. имеет вид

$$\Pi_f = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \Pi_f = \begin{bmatrix} -8.2633 & 51.0437 & -42.6448 \\ 2.3167 & -2.2109 & 0.8854 \\ 0.0126 & -0.1608 & 0.2919 \end{bmatrix}, \quad \Pi_f = \begin{bmatrix} 10.7081 & -48.8442 & 38.2399 \\ -0.0027 & 4.5574 & -5.4229 \\ 0.0000 & -0.0000 & 0.0234 \end{bmatrix}$$

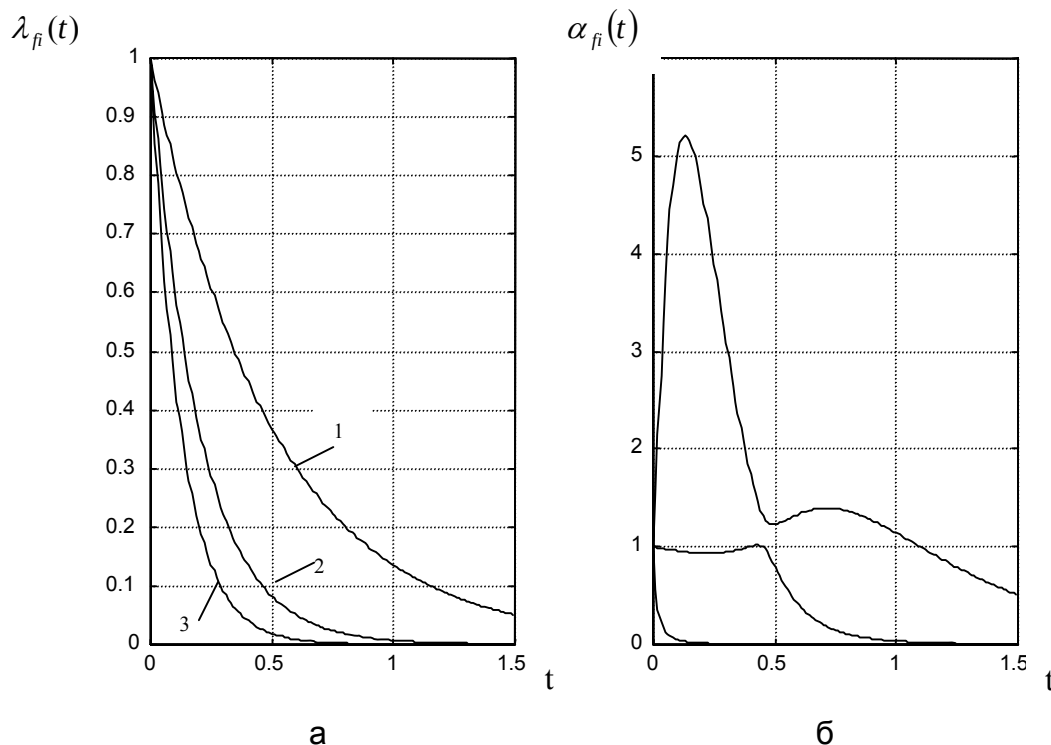


Рис. 1. Кривые: а – собственных чисел, б – сингулярных чисел фундаментальной матрицы $f(F) = \exp(Ft)$

На рис. 1 приведены кривые: а) 1 - $\lambda_{f1}(t) = \exp(-2t)$, 2 - $\lambda_{f2}(t) = \exp(-5t)$, 3 - $\lambda_{f3}(t) = \exp(-8t)$. На рис. 1, б) приведены кривые сингулярных чисел

$\alpha_i(t)$, $i = \overline{1,3}$ фундаментальной матрицы $f(F) = \exp(Ft)$, вычисленные в силу (20).

Заключение

Установленные связи собственных значений и сингулярных чисел матрицы состояния системы (1) и матричной функции $f(F)$ от матрицы F , а также их функций чувствительности позволяют корректно сформулировать требования к допустимым вариациям собственных значений и собственных векторов, которые обеспечивают требуемую стабильность эллипсоидных показателей качества.

Литература

1. Модальные оценки качества процессов в линейных многомерных системах / Т. А. Акунов, С. Алишеров, Р. О. Оморов, А. В. Ушаков; Под ред. А. В. Ушакова. Бишкек: Илим, 1991.
2. Ушаков А.В. Обобщенное модальное управление // Изв. вузов. Приборостроение. 2000. Т.43, №3. С. 8-16.
3. Акунов Т. А., Ушаков А. В. Анализ чувствительности эллипсоидных оценок качества многомерных процессов управления // Изв. вузов. Приборостроение. 1991. Т.34, №8. С. 21-27.
4. Григорьев В.В., Дроздов В.Н., Лаврентьев В.В., Ушаков А.В. Синтез дискретных регуляторов при помощи ЭВМ. Л.: Машиностроение. 1983.
5. Гантмахер Ф.Р. Теория матриц. М.: Наука, 1988.

КОЛЕБАТЕЛЬНЫЕ ПРОЦЕССЫ В ПЛОХО ОБУСЛОВЛЕННЫХ АПЕРИОДИЧЕСКИХ СИСТЕМАХ

А. Акунова, А.В. Ушаков

В работе с использованием аппарата систем сравнения минимальной достаточности, конструируемых в функциональном базисе фундаментальной матрицы системы формулируется критерий, с помощью которого обнаруживается колебательность процессов в плохо обусловленных аperiodических системах. Приводятся примеры.

Введение. Постановка задачи

Традиционно при исследовании процессов в многомерных системах по вектору состояния и выхода в основном учитывается структура собственных значений матрицы состояния системы. При таком подходе в аperiodических непрерывных линейных системах, то есть системах с вещественным спектром собственных значений гурвицевой матрицы состояния простой структуры, невозможно априори ожидать появления процессов, обнаруживающих признаки колебательности, проявляющейся в выбросах на траекториях однородной и неоднородной версий. Однако, если при назначении матрицы состояния системы желаемой структуры собственных значений, которая должна обеспечивать аperiodичность траекторий, не контролировать геометрический спектр ее собственных векторов, то при плохой обусловленности матрицы последних аperiodические свойства могут быть не достигнуты. Этой проблеме посвящена настоящая работа.

Основной результат

Рассматривается непрерывная система

$$\dot{x}(t) = Fx(t) + Gg(t); x(0), y(t) = Cx(t); \quad (1)$$

где $x \in R^n$, $g \in R^m$, $G \in R^{n \times m}$, $F \in R^{n \times n}$, матрица состояния F гурвицева и простой структуры; x - вектор состояния, g - конечномерное экзогенное воздействие ограниченной нормы формируется с помощью автономной системы

$$\dot{z}(t) = \Gamma z(t); z(0) = z_0, g(t) = Pz(t), \quad (2)$$

в котором $z \in R^l$, $\Gamma \in R^{l \times l}$, $P \in R^{m \times l}$.

Утверждение 1. Общее решение системы (1)

$$x(t) = x[x(0), g(t), t] \quad (3)$$

может быть записано в виде

$$x(t) = \Pi(t)\chi(0), \quad (4)$$

где матрица $\Pi(t)$ имеет представление

$$\Pi(t) = [\exp(Ft) \quad T \exp(\Gamma t) - \exp(Ft)T], \quad (5)$$

а вектор $\chi(0)$ задается в форме

$$\chi(0) = [x^T(0), z^T(0)]^T. \quad (6)$$

Матрица T в (5) является решением матричного уравнения Сильвестра

$$T\Gamma - FT = GP. \quad (7)$$

Доказательство утверждения 1 можно найти в работах [1,2]. ■

Утверждение 2. Если система (2) является источником векторного ступенчатого воздействия $g(t) = g_0 = Pz(0)$, то матрица $\Pi(t)$ (5) принимает вид

$$\Pi(t) = [\exp(Ft) \quad (\exp(Ft) - I)F^{-1}GP]. \quad (8)$$

Доказательство утверждения 2 строится на использовании того факта, что при формировании векторного ступенчатого воздействия $g(t) = g_0 = Pz(0)$ минимальная реализация системы (2) характеризуется матричным условием

$$\Gamma = 0. \quad (9)$$

Подстановка нулевой матрицы (9) в уравнение Сильвестра (7) позволяет найти явное решение для матрицы $T = -F^{-1}GP$, а подстановка в (5) дает $\exp(\Gamma t) = I$, что в совокупности приводит к (8). ■

Сконструируем скалярные экспоненциальные системы сравнения (СЭСС) минимальной достаточности [3] в функциональном базисе фундаментальной матрицы $\Phi(t) = \exp(Ft)$ системы (1) для решений $x(t) = x[x(0), g(t) \equiv 0, t]$ однородной (автономной) версии системы (1) и для ее неоднородной версии $x(t) = x[x(0) \equiv 0, g(t) = g_0, t]$.

Утверждение 3. Решение однородной скалярной системы

$$\dot{\zeta}(t) + \eta \zeta(t) = 0; \zeta(0) \geq \beta \|x(0)\| \quad (10)$$

мажорирует решение $x(t) = x[x(0), g(t) \equiv 0, t]$ однородной версии системы (1) в смысле выполнения неравенства

$$\|x(t)\| \leq \zeta(t) = \beta e^{-\eta t} \|x(0)\|, \quad (11)$$

при этом

$$\eta = \min_i \{\operatorname{Re} \lambda_i = \lambda_i; i = \overline{1, n}\}, \beta = C\{M\}, \quad (12)$$

где $C\{M\}$ - число обусловленности матрицы $M = \operatorname{row}\{M_i; i = \overline{1, n}\}$ собственных векторов $M_i: \|M_i\| = 1$ матрицы F состояния системы (1).

Доказательство утверждения 3 строится на непосредственном вычислении нормы $\|x(t)\|$ решения $x(t) = x[x(0), g(t) \equiv 0, t]$, задаваемого выражением (4), в котором следует подставить $\chi(0) = [x(0)^T \ 0^T]^T$ и матрицу $\Pi(t)$ в форме (5). Это вычисление дает цепочку неравенств

$$\begin{aligned} \|x(t)\| &= \|\exp(Ft)x(0)\| \leq \|M\| \|\exp(\Lambda t)\| \|M^{-1}\| \|x(0)\| = \\ &= C\{M\} \|\operatorname{diag}\{e^{\lambda_i t}; i = \overline{1, n}\}\| \|x(0)\| = C\{M\} e^{-\eta t} \|x(0)\|. \end{aligned} \quad \blacksquare \quad (13)$$

Нетрудно видеть, что η - представляет собой степень устойчивости системы (1) с матрицей F простой структуры. Система (10) именуется системой сравнения для решений $x(t) = x[x(0), g(t) \equiv 0, t]$ системы (1).

Доказательство утверждения 3 делает справедливым положения следующего утверждения.

Утверждение 4. Множество непрерывных систем вида (1), обладающих совпадающими степенями устойчивости η и одинаково обусловленными структурами собственных векторов, характеризующихся единым числом обусловленности $C\{M\}$, образует класс эквивалентности в смысле выполнения условия идентичности экспоненциальных покрытий вида (11), задаваемых системой сравнения (10), при равных по норме начальных состояний. ■

Экспоненциальное покрытие (11) с параметрами (12) является критериальным в смысле обнаружения колебательности процессов $\|x(t)\| = \|x[x(0), g(t) \equiv 0, t]\|$. Так, если $C\{M\} = 1$, то матрица F связана с диагональной с помощью ортогональной матрицы M в форме $M\Lambda = FM$ и в силу свойства ортогональной матрицы не менять нормы исходной при умножении на нее, процессы в аperiodической системе (1) носят экспоненциальный характер, начинаясь в точке $\|x(0)\|$.

В случае $C\{M\} > 1$ процессы $\|x(t)\| = \|x[x(0), g(t) \equiv 0, t]\|$, начинаясь в $(\circ) \|x(0)\|$, возрастают со временем, обнаруживая в их ходе выброс, а затем экспоненциально

сходятся. Экспоненциальные покрытия таких процессов с выбросом возможно с помощью (11) только при $C\{M\} > 1$ так, чтобы они начинались в точке $C\{M\}\|x(0)\|$, при этом мажорирующая их оценка $\|\hat{x}(t)\|$ минимальной достаточности в силу (4) определяется максимальным сингулярным числом $\alpha_M\{\Pi(t)\}$ на сфере $\|x(0)\| = x_0$ так, что $\|\hat{x}(t)\| = \alpha_M\{\exp(Ft)\}\|x(0)\|$. Подобное обнаруживается и на траекториях системы (1) $\|x(t)\| = \|x[x(0) \equiv 0, g(t) = g_0, t]\|$.

Утверждение 5. Решение неоднородной скалярной системы

$$\dot{\xi}(t) + \nu\xi(t) = \gamma\nu\|g_0\|; \quad \xi(0) \equiv 0, \quad (14)$$

именуемой неоднородной СЭСС мажорирует решение $\|x(t)\| = \|x[x(0) \equiv 0, g(t) = g_0, t]\|$ неоднородной версии системы в смысле выполнения неравенства

$$\|x(t)\| \leq \xi(t) = \gamma\|g_0\|(1 - e^{-\nu t}), \quad (15)$$

при этом

$$\nu = \max_i \{\operatorname{Re} \lambda_i = \lambda_i; i = \overline{1, n}\} \quad (16)$$

$$\gamma = \frac{1}{\eta} C\{M\}\|G\|. \quad (17)$$

Доказательство утверждения, как и в случае утверждения 3, строится на непосредственном вычислении нормы $\|x(t) = x[x(0) \equiv 0, g(t) = g_0 = Pz(0), t]\|$, задаваемого выражением (4), в котором следует положить $x(0) = [0^T \quad z^T(0)]^T$ и матрицу $\Pi(t)$ в форме (8). Это вычисление дает цепочку неравенств

$$\begin{aligned} \|x(t)\| &= \|(I - \exp(Ft))F^{-1}GPz(0)\| = \|M \operatorname{diag}(\lambda_i^{-1}(1 - e^{\lambda_i t}), i = \overline{1, n})M^{-1}Gg_0\| \leq \\ &\leq \|M\| \|\operatorname{diag}(\lambda_i^{-1}(1 - e^{\lambda_i t}), i = \overline{1, n})\| \|M^{-1}\| \|G\| \|g_0\| \leq C\{M\} \frac{1}{\eta} (1 - e^{\nu t}) \|G\| \|g_0\| \quad \blacksquare \end{aligned} \quad (18)$$

Как и в случае однородной версии системы (1), ее неоднородная версия для процессов $\|x(t)\| = \|x[x(0) \equiv 0, g(t) = g_0, t]\|$ в случае плохо обусловленной структуры собственных векторов, характеризующейся выполнением условия $C\{M\} > 1$, обнаруживает колебательность этих процессов, при этом мажорирующая их оценка $\|\hat{x}(t)\|$ минимальной достаточности в силу (4) определяется максимальным сингулярным числом $\alpha_M\{\Pi(t)\}$ матрицы $\Pi(t)$ вида (8) на сфере $\|x(0)\| = \|[0^T \quad z^T(0)]^T\|$, порождающую в силу $Pz(0) = g_0$ сферу скачкообразного воздействия $\|g(t)\| = \|g_0\|$ так, что

$$\|\hat{x}(t)\| = \alpha_M\{(I - \exp(Ft))F^{-1}G\}\|g_0\|. \quad (19)$$

Установившиеся значения

$$\xi_y(t) = \lim_{t \rightarrow \infty} \xi(t) = \frac{1}{\eta} C\{M\}\|G\|\|g_0\| \quad (20)$$

и

$$\|\hat{x}(t)\|_y = \lim_{t \rightarrow \infty} \|\hat{x}(t)\| = \alpha_M\{F^{-1}G\}\|g_0\| \quad (21)$$

по существу разнятся в $C\{M\}$ раз.

Класс эквивалентности неоднородных версий системы (1) при скачкообразном экзогенном воздействии $g(t) = g_0$ порождается четверкой показателей $\{\eta, \nu, C\{M\}, \|G\|\}$.

Примеры. В качестве иллюстрирующих примеров рассматриваются только однородные версии системы (1) второго порядка, матрицы $F_{jr} (j = \overline{1, p})$, где $(p \times r)$ - мощность выборки эксперимента, которых формируются по правилу

$$F_{jr} = M_j \Lambda_r M_j^{-1}, \quad (22)$$

где $\Lambda_r = \text{diag}\{\lambda_{r1}, \lambda_{r2}\}$, M_j - реализации матриц собственных векторов с задаваемым $C\{M_j\}$ - числом обусловленности.

Рассматриваются две реализации диагональных матриц Λ_r с различными спектрами, но единой степенью устойчивости $\eta = 1$

$$\Lambda_1 = \text{diag}\{\lambda_{11} = -1, \lambda_{12} = -20\} \text{ и } \Lambda_2 = \text{diag}\{\lambda_{21} = -1, \lambda_{22} = -2\}. \quad (23)$$

Матрицы (22) из (23) получены с помощью матриц

$$M_1 = \begin{bmatrix} 1 & 1 \\ 0 & 0,01 \end{bmatrix}; M_2 = \begin{bmatrix} 1 & 0,707 \\ 0 & 0,707 \end{bmatrix}; M_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

характеризующиеся числами обусловленности $C\{M_1\} = 200$; $C\{M_2\} = 2,4$ и $C\{M_3\} = 1$.

Кривые экспоненциальных покрытий $\zeta(t)$ и покрываемых процессов $\|\hat{x}(t)\|$ приведены соответственно на рис.1 ÷ рис.3. В полном соответствии с теорией, при $C\{M\} > 1$ процессы (см. рис.1,2) $\|\hat{x}(t)\|$ обнаруживают выбросы в аperiодических системах, на факт наличия которых экспоненциальные покрытия “реагируют” подъемом в $C\{M\}$ раз начального значения $\zeta(0)$. При $C\{M\} = 1$ (см. рис. 3) выбросы отсутствуют, процесс $\|\hat{x}(t)\|$ аperiодический, при этом $\zeta(t) \equiv \|\hat{x}(t)\|$.

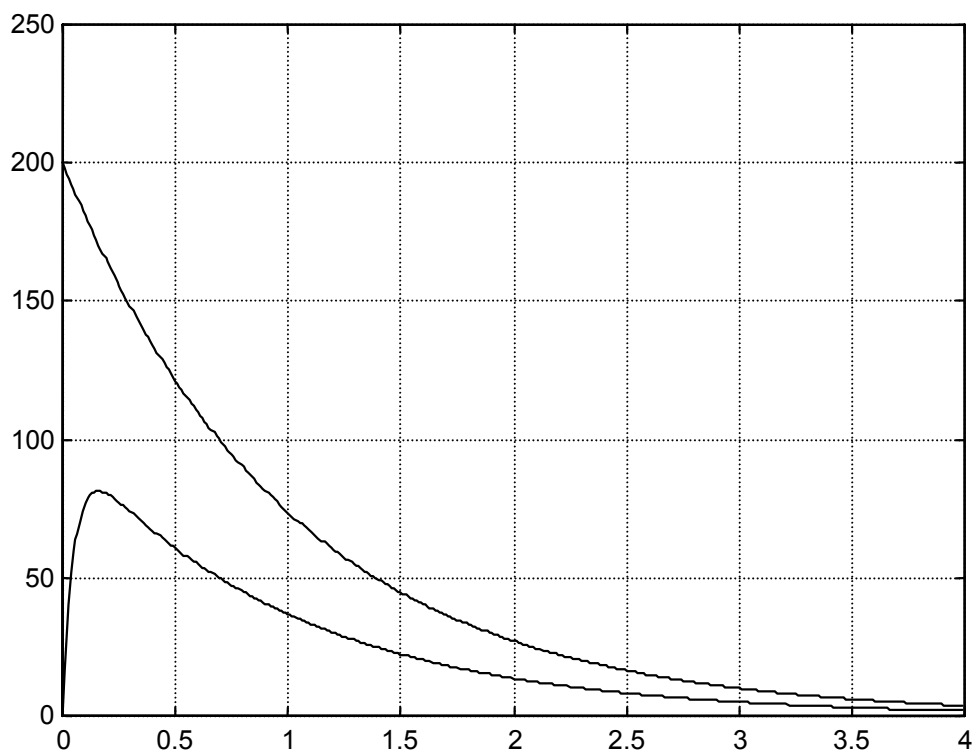


Рис. 1. Кривые $\|x(t)\|$ и мажорирующей экспоненциальной системы сравнения $\zeta(t)$ с параметром $\beta = C\{M_1\} = 200$

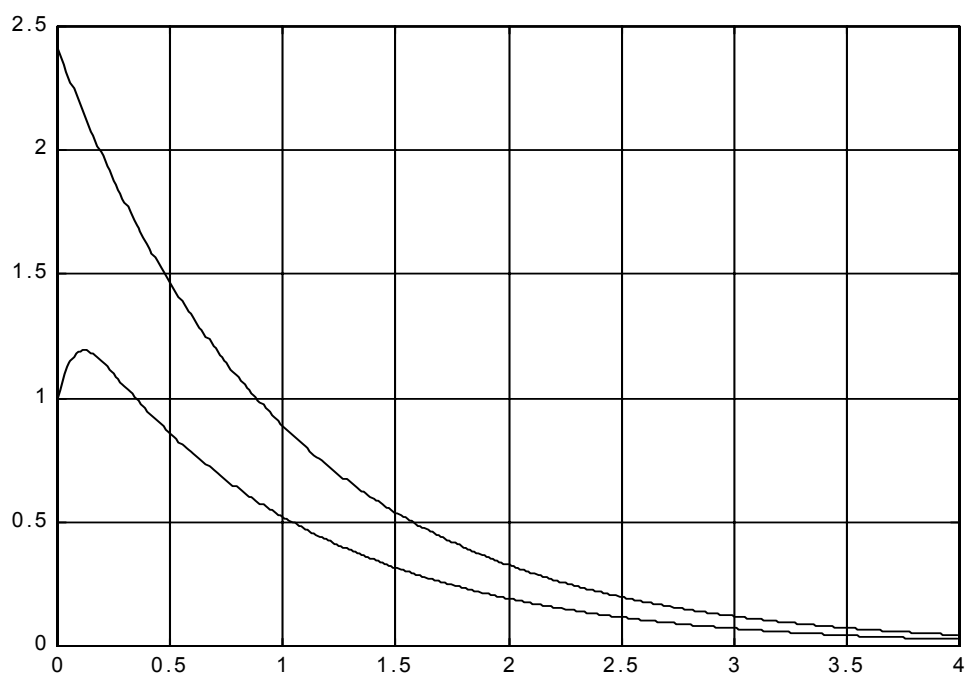


Рис. 2. Кривые $\|x(t)\|$ и мажорирующей экспоненциальной системы сравнения $\zeta(t)$ с параметром $\beta = C\{M_2\} = 2,4$

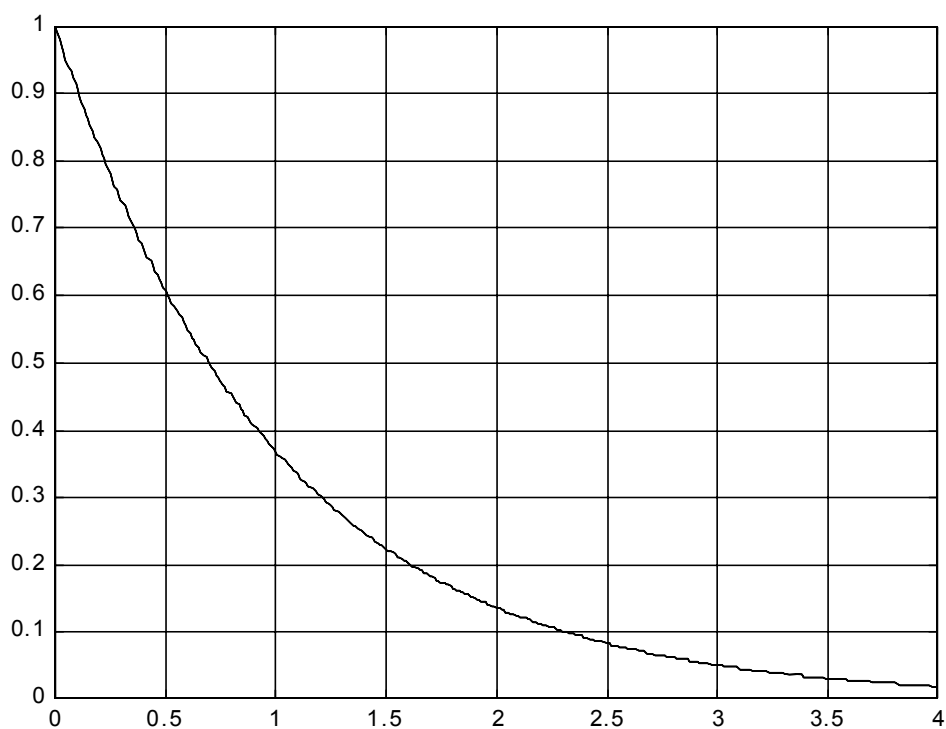


Рис. 3. Кривые $\|x(t)\|$ и мажорирующей экспоненциальной системы сравнения $\zeta(t)$ с параметром $\beta = C\{M_3\} = 1$

Заключение.

При синтезе систем вида (1) с априори желаемой апериодичностью ее процессов необходимо контролировать не только спектр собственных значений матрицы состояния, но и спектр ее собственных векторов. Для этой цели могут быть использованы возможности обобщенного модального управления [4], которая позволяет обеспечить оба желаемых спектра матрицы состояния системы.

Литература

1. Ушаков А.В. Модальные оценки качества процессов управления многомерными системами при гармоническом внешнем воздействии. // Автоматика и телемеханика, 1989. №11.
2. Модальные оценки качества процессов в линейных многомерных системах / Акунов Т.А., Алишеров С., Оморов Р.О., Ушаков А.В.; Под ред. А.В. Ушакова // Препринт. Бишкек: Илим, 1991.
3. Акунова А., Акунов Т.А., Ушаков А.В. Оценка качества решений системы дифференциальных уравнений с помощью систем сравнения минимальной достаточности // Вторая международная конференция “Дифференциальные уравнения и их применения” / Тезисы докладов. Санкт-Петербург, 1998. С.72-73.
4. Ушаков А.В. Обобщенное модальное управление // Изв. вузов. Приборостроение. 2000. Т.43, №3. С. 8-16.

РАЗВИТИЕ МЕТОДОВ РОБАСТНОГО УПРАВЛЕНИЯ В ЗАДАЧАХ АДАПТАЦИИ

А.А. Бобцов, С.А. Холунин

1. Введение

Всплеск популярности теории адаптивных систем имевший место в 70-х годах, постепенно начал угасать к середине 90-х. Темы, связанные с решением задач адаптивного управления перестали широко освещаться на отечественных и международных конференциях, финансирование проектов по этой некогда популярной тематике, постепенно стало уменьшаться. К причинам такого регресса следует отнести, в первую очередь, плохую реализацию на практике имеющихся схем адаптивного управления. Громоздкие, сложные и не всегда помехоустойчивые математические законы управления оказались “не по вкусу” инженерам-практикам. Возникло, как это часто бывает, рассогласование теоретических подходов с практикой. Решение проблемы понимания должно на этот раз исходить именно от математиков и специалистов, занимающихся теорией управления. Первые шаги в этом направлении были сделаны, в том числе, отечественными учеными [1,2], по огрублению алгоритмов адаптации по отношению к внешним неучтенным факторам. Дальнейший уход от алгоритмов адаптации интегрального типа к алгоритмам сигнальной адаптации [1,2], позволил значительно сократить размерность законов управления. Параллельно достижению этих результатов идет развитие методов адаптивного управления по выходу [3,4]. Получен ряд интересных схем и в том числе схем с огрублением для внешних возмущений, но проблема громоздкости и сложности все еще имеет место. К сегодняшнему дню получен ряд оптимальных подходов адаптивного управления по выходу заключающий в себе стратегию робастного управления [5-7].

Предлагаемая работа представляет собой развитие методов робастного управления в задачах адаптации по выходу [5-7]. В статье предлагаются новые схемы управления позволяющие получать менее сложные и громоздкие по размерности алгоритмы.

2. Постановка задачи

Рассмотрим неопределенную систему вида [3-5]

$$\dot{x} = Ax + b\omega(t)^T \theta + bu, \quad (1)$$

$$y = c^T x, \quad (2)$$

где $x = x(t)$ - вектор переменных состояния; y - регулируемая переменная; u - сигнал управления; матрица A - гурвица; $\omega(t)$ - известная функция (регрессор); $\theta \in R^q$ - вектор неизвестных постоянных параметров.

Наряду с моделью (1), (2) также будем рассматривать математическую модель “вход-выход” [3,5]

$$y(t) = \frac{B(p)}{A(p)} [\omega(t)^T \theta + u], \quad (3)$$

где $p = d/dt$ - оператор дифференцирования и полином $A(p)$ - асимптотически устойчивый.

Сформулируем цель управления, как решение задачи синтеза алгоритма, обеспечивающего при любых начальных состояниях объекта выполнение условия

$$\lim_{t \rightarrow \infty} |y(t)| \leq \varepsilon, \quad (4)$$

где ε - любое произвольно малое число.

3. Синтез алгоритма адаптации

Выберем передаточную функцию $W(p)$, удовлетворяющую соотношению

$$W(p) = (p + \alpha)H(p), \quad (5)$$

где α - любая положительная константа. Очевидно, что при таком представлении передаточная функция $W(p)$ является асимптотически устойчивой. Так как

$$H(p) = \frac{1}{p + \alpha} W(p),$$

то модель (3) может быть переписана в виде

$$y = \frac{1}{p + \alpha} [\varpi^T \theta + \bar{u}] + \delta \quad (6)$$

или

$$\dot{y} = -\alpha y + \varpi^T \theta + \bar{u} + \bar{\delta}, \quad (7)$$

где $\delta(t)$ - экспоненциально затухающая функция времени, вызванная ненулевыми начальными условиями; $\bar{\delta} = \dot{\delta} + \alpha\delta$ - экспоненциально затухает; функция $\varpi = W(p)\omega$ и

$$\bar{u} = W(p)u. \quad (8)$$

Введем новую переменную

$$\varphi = \varpi^T \theta + \bar{\delta}, \quad (9)$$

тогда модель (7) примет вид

$$\dot{y} = -\alpha y + \varphi + \bar{u}. \quad (10)$$

Пусть новый закон управления $\bar{u}$ имеет вид

$$\bar{u} = -\hat{\varphi}, \quad (11)$$

где $\hat{\varphi}$ - текущая оценка функции φ . Тогда закон управления будет записан следующим образом

$$u = -W(p)^{-1} \hat{\varphi}. \quad (12)$$

Проблема в синтезе алгоритма (12) - это:

- реализация операции дифференцирования;
- достаточно точная оценка функции φ .

Поэтому требуется выстроить такую схему оценки функции φ , чтобы:

- $(\varphi - \hat{\varphi}) \leq \bar{\varepsilon}$, где $\bar{\varepsilon}$ - малое число, такое что условие (4) будет удовлетворено;
- для нахождения производных от $\hat{\varphi}$ была использована реализуемая вычислительная процедура.

Временно предполагая, что функция φ подлежит измерению, выберем следующий алгоритм оценки

$$\begin{cases} \dot{\xi}_1 = \gamma \bar{\sigma} \xi_2, \\ \dot{\xi}_2 = \gamma \bar{\sigma} \xi_3, \\ \dots \\ \dot{\xi}_m = \gamma \bar{\sigma} (-k_1 \xi_1 - k_2 \xi_2 - \dots - k_m \xi_m + k_1 \varphi), \end{cases} \quad (13)$$

$$\hat{\varphi} = \xi_1, \quad (14)$$

где число m определяет порядок системы (13) и выбирается таким образом, чтобы закон управления (12) был бы реализуем; постоянная $\gamma > 0$; положительная функция $\bar{\sigma}$ является функцией той же скорости роста что и $|\dot{\varphi}|^2$ (т.е. $|\dot{\varphi}|^2 / \bar{\sigma} < \infty$), а все ее $m-1$ производные известны или подлежат измерению; коэффициенты k_i рассчитываются из соображений асимптотической устойчивости модели (13).

Теорема 1. Алгоритм оценки (13), (14) позволяет при увеличении параметра γ максимально приблизить оценку $\hat{\phi}$ к функции ϕ .

Доказательство. Перепишем модель (13), (14) в векторно-матричной форме

$$\dot{\xi} = \gamma \bar{\sigma}(\Gamma \xi + dk_1 \phi), \quad (15)$$

$$\hat{\phi} = h^T \xi, \quad (16)$$

$$\text{где } \Gamma = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -k_1 & -k_2 & -k_3 & \dots & -k_m \end{bmatrix}, \quad d = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \text{ и } h = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Введем в рассмотрение вектор отклонений $\eta = h\phi - \xi$, тогда для его производной получим

$$\dot{\eta} = h\dot{\phi} - \gamma \bar{\sigma}(\Gamma(h\phi - \eta) + dk_1 \phi) = h\dot{\phi} + \gamma \bar{\sigma}\Gamma\eta - \gamma \bar{\sigma}(dk_1 + \Gamma h)\phi. \quad (17)$$

Так как $dk_1 = -\Gamma h$ (проверяется подстановкой), то

$$\dot{\eta} = h\dot{\phi} + \gamma \bar{\sigma}\Gamma\eta,$$

где Γ - гурвицева.

Для доказательства сходимости η в любую заданную область, рассмотрим функцию Ляпунова вида

$$V = \eta^T P \eta, \quad (18)$$

где матрица $P = P^T > 0$, такая что

$$\Gamma^T P + P\Gamma \leq -\lambda P < 0. \quad (19)$$

Дифференцируя (18), получаем

$$\begin{aligned} \dot{V} &= \gamma \bar{\sigma} \eta^T (\Gamma^T P + P\Gamma) \eta + 2\eta^T P h \dot{\phi} \leq -\lambda \gamma \bar{\sigma} \eta^T P \eta + \\ &+ \mu \bar{\sigma} \eta^T P h h^T P \eta + \mu^{-1} \bar{\sigma}^{-1} |\dot{\phi}|^2, \end{aligned} \quad (20)$$

где положительное число μ такое что

$$\dot{V} \leq -\bar{\lambda} \gamma \bar{\sigma} \eta^T P \eta + \mu^{-1} \bar{\sigma}^{-1} |\dot{\phi}|^2, \quad (21)$$

а постоянная $\bar{\lambda} > 0$.

При выводе соотношения (20) было использовано легко проверяемое неравенство

$$(\mu \bar{\sigma})^{-1} (\mu \bar{\sigma} a - b)^2 \geq 0,$$

откуда следует, что

$$2ab \leq \mu \bar{\sigma} a^2 + (\mu \bar{\sigma})^{-1} b^2$$

где $a = \eta^T P h$ и $b = \dot{\phi}$.

Поскольку функция $\bar{\sigma}$ является функцией той же скорости роста что и $|\dot{\phi}|^2$, то из выражения (21) следует, что вектор отклонений $\eta = h\phi - \xi$ ограничен и все его переменные могут быть сведены к любому малому компактному множеству, за счет увеличения параметра γ .

Для доказательства основного положения представленной теоремы, а именно максимального приближения $\hat{\phi}$ к ϕ при увеличении параметра γ , умножим вектор отклонений $\eta = h\phi - \xi$ слева на h^T . Тогда в силу структуры матрицы h , получаем:

$$h^T \eta = \phi - h^T \xi = \phi - \hat{\phi}$$

и при соответствующих γ (достаточно больших) обеспечивается сходимость в любую малую окрестность.

Замечание 1. Следует отметить, что в качестве мажоранты $\bar{\sigma}$ можно использовать функцию $|\dot{\omega}|^2 = \dot{\omega}^T \dot{\omega}$, так как сигнал $\dot{\phi}(t)$ пропорционален $|\dot{\omega}(t)|$, т.е.:

$\frac{|\dot{\phi}(t)|}{|\dot{\omega}(t)|} < \infty$. В тоже время, в законе управления (12) будут использоваться только

измеряемые производные от ω вплоть до m -ой, что в свою очередь, позволяет выбирать мажоранту $\bar{\sigma}$ указанным способом.

Замечание 2. Если функция $\dot{\phi}(t)$ имеет неограниченный рост во времени, то в силу неравенства (21), следует, что при $\dot{\phi}(t) \rightarrow \infty$ переменная $\eta \rightarrow 0$ и, следовательно, $\hat{\phi} \rightarrow \phi$.

Замечание 3. Если регрессор $\omega(t)$ - ограничен, то в качестве мажоранты $\bar{\sigma}$ можно принять любую положительную константу.

Замечание 4. Отметим, что реализация мажоранты $\bar{\sigma}$ с использованием алгоритма приведенного в *замечании 1* для некоторых задач может оказаться громоздким, т.к. предполагает формирование вектора $\dot{\omega}$ каждая компонента которого находится из соотношения $\dot{\omega}_i = pW(p)\omega_i$. Во избежание громоздких процедур расчета мажоранты, целесообразно использовать алгоритм вида:

$$\bar{\sigma} = pW(p)[\omega^T \omega + \beta],$$

где $\beta > 0$.

Теперь построим реализуемую схему алгоритма оценки (13), (14) следующего вида

$$\begin{cases} \dot{\xi}_1 = \gamma \bar{\sigma} \xi_2, \\ \dot{\xi}_2 = \gamma \bar{\sigma} \xi_3, \\ \dots \end{cases} \quad (22)$$

$$\begin{cases} \dot{\varsigma} = \gamma \bar{\sigma} (-k_2 \xi_2 - \dots - k_m \xi_m + k_1 \alpha y) - \gamma \dot{\bar{\sigma}} k_1 y, \\ \xi_m = \varsigma + \gamma \bar{\sigma} k_1 y. \end{cases} \quad (23)$$

Система (22), (23) содержит переменные, которые могут быть измерены или рассчитаны. Возможность применения указанного алгоритма представлена в следующей теореме.

Теорема 2. Алгоритм оценки вида (22), (23) эквивалентен алгоритму (13), (14).

Доказательство. Из уравнения (10) находим

$$\phi = \dot{y} + \alpha y - \bar{u}. \quad (24)$$

Подставляя последнее уравнение в выражение (13), получаем

$$\begin{cases} \dot{\xi}_1 = \gamma \bar{\sigma} \xi_2, \\ \dot{\xi}_2 = \gamma \bar{\sigma} \xi_3, \\ \dots \\ \dot{\xi}_m = \gamma \bar{\sigma} (-k_1 \xi_1 - k_2 \xi_2 - \dots - k_m \xi_m + k_1 (\dot{y} + \alpha y - \bar{u})), \end{cases} \quad (25)$$

учитывая, что

$$\bar{u} = -\hat{\phi},$$

получаем

$$\begin{cases} \dot{\xi}_1 = \gamma \bar{\sigma} \xi_2, \\ \dot{\xi}_2 = \gamma \bar{\sigma} \xi_3, \\ \dots \\ \dot{\xi}_m = \gamma \bar{\sigma} (-k_2 \xi_2 - \dots - k_m \xi_m + k_1 (\dot{y} + \alpha y)). \end{cases} \quad (26)$$

Введем в рассмотрение новую переменную

$$\varsigma = \xi_m - \gamma \bar{\sigma} k_1 y. \quad (27)$$

Тогда дифференцируя (27) для системы уравнений (26), получаем

$$\begin{cases} \dot{\xi}_1 = \gamma \bar{\sigma} \xi_2, \\ \dot{\xi}_2 = \gamma \bar{\sigma} \xi_3, \\ \dots \\ \dot{\varsigma} = \gamma \bar{\sigma} (-k_2 \xi_2 - \dots - k_m \xi_m + k_1 \alpha y) - \gamma \bar{\sigma} k_1 \dot{y}, \end{cases} \quad (28)$$

$$\dot{\xi}_m = \varsigma + \gamma \bar{\sigma} k_1 y. \quad (29)$$

4. Пример

Для иллюстрации работоспособности предложенной в работе схемы управления, рассмотрим числовой пример. Пусть модель (1), (2) имеет вид:

$$\dot{x}_1 = x_2, \quad (30)$$

$$\dot{x}_2 = -x_1 - 2x_2 + 3 \sin 4t + u, \quad (31)$$

$$y = x_1, \quad (32)$$

где вектор неопределенных параметров $\theta = 3$, регрессор $\omega = \sin 4t$, u - искомое управление.

Для объекта (30)-(32) рассмотрим математическую модель математическую модель “вход-выход” вида (3)

$$y(t) = \frac{1}{(p+1)^2} [3 \sin 4t + u]. \quad (33)$$

Выберем передаточную функцию $W(p) = \frac{1}{p+1}$, тогда выражение (33) в соответствии с результатами раздела 3, примет вид (см. уравнение (10)):

$$\dot{y} = -y + \varphi + \bar{u}, \quad (34)$$

где функция $\varphi = \bar{\delta} + W(p)3 \sin 4t$.

Выбирая новый закон управления $\bar{u}$ как

$$\bar{u} = -\hat{\varphi}, \quad (35)$$

получаем для истинного закона управления

$$u = -W(p)^{-1} \hat{\varphi} = -(p+1) \hat{\varphi} = -\hat{\varphi} - \dot{\hat{\varphi}}. \quad (36)$$

Для реализации оценки $\hat{\varphi}$, воспользуемся алгоритмом (22), (23):

$$\begin{cases} \dot{\xi}_1 = \gamma \xi_2, \\ \dot{\varsigma} = \gamma (-2 \xi_2 + y), \end{cases} \quad (37)$$

$$\xi_2 = \varsigma + \gamma y, \quad (38)$$

$$\hat{\varphi} = \xi_1, \quad (39)$$

где коэффициенты $k_2 = 2$, $k_1 = 1$ и в силу ограниченности регрессора $\omega = \sin 4t$ (см. замечание 3), мажоранта $\bar{\sigma} = 1$.

Запишем закон управления в обозначениях алгоритма оценки (37)–(39)

$$u = -\hat{\varphi} - \dot{\hat{\varphi}} = -\xi_1 - \gamma (\varsigma + \gamma y). \quad (40)$$

и проведем компьютерное моделирование. Результаты моделирования для различных значений параметра γ представлены на рис. 1, рис. 2 и демонстрируют уменьшение значения выходной переменной с ростом параметра γ .

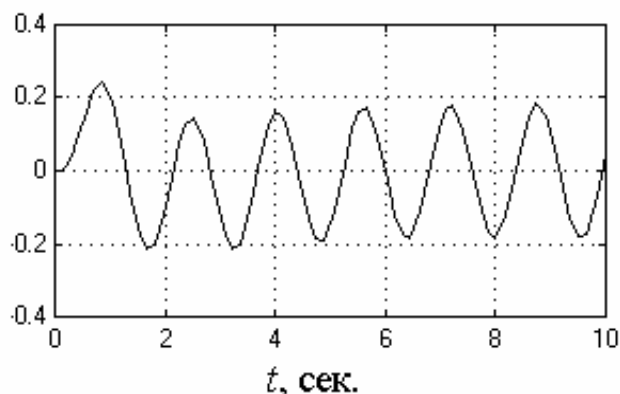


Рис. 1. График переходного процесса по переменной $y(t)$ при $\gamma = 5$

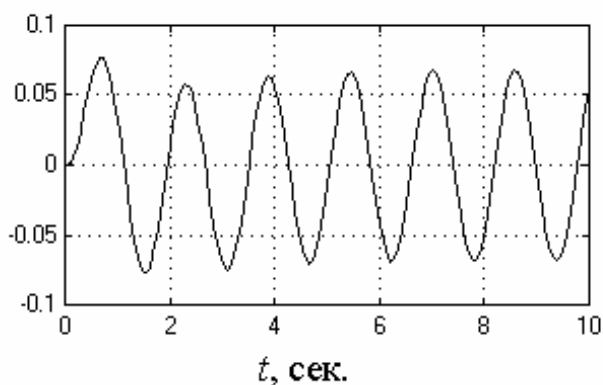


Рис. 2. График переходного процесса по переменной $y(t)$ при $\gamma = 20$.

Заключение

Работа представляет собой развитие методов робастного управления в задачах адаптации по выходу. В статье предлагаются схемы управления позволяющие получать, как менее сложные и громоздкие по размерности алгоритмы и алгоритмы. Структура регулятора является линейной и содержит нестационарный фильтр, параметры которого выбираются из требований предъявляемых к выходной переменной объекта.

Литература

1. Фомин В.Н., Фрадков А.Л., Якубович В.А. Адаптивное управление динамическими объектами. М. Наука, 1981.
2. Фрадков А.Л. Адаптивное управление в сложных системах. М.: Наука, 1990.
3. Никифоров В.О., Фрадков А.Л. Схемы адаптивного управления с расширенной ошибкой. Обзор // Автоматика и телемеханика. 1994. №9.
4. Андриевский Б.Р., Фрадков А.Л., Избранные главы теории автоматического управления с примерами на языке MATLAB. СПб.: Наука, 1999.
5. Мирошник И.В., Никифоров В.О., Фрадков А.Л. Нелинейное и адаптивное управление сложными механическими системами. СПб.: Наука, 2000.
6. Никифоров В.О. Робастное управление линейным объектом по выходу // Автоматика и телемеханика. 1998. № 9.
7. Никифоров В.О. Адаптивное и робастное управление с компенсацией возмущений. Диссертация на соискание ученой степени доктора технических наук. СПб., 2001.

СТАБИЛИЗАЦИЯ ЛИНЕЙНЫХ СТАЦИОНАРНЫХ СИСТЕМ ПО СОСТОЯНИЮ

А.А. Бобцов, М.Б. Гурбашков

1. Введение

Задача стабилизации линейных стационарных систем по состоянию является хорошо изученной. Однако проблема управления в условиях неопределенности параметров продолжает волновать умы специалистов по теории автоматического регулирования и по сей день. В настоящее время получено множество решений, как в классе задач адаптивного [1-4], так и в классе задач робастного управления [4]. Как правило, полученные алгоритмы либо достаточно сложны в реализации [4-5], либо преследуют решение локальной задачи, например, регулирование выхода [4-6], либо математические модели объектов имеют некоторую фиксированную структуру относительно входящих в нее неопределенностей [7].

Результаты предлагаемой статьи не претендуют на универсальность и общность, а лишь расширяют класс подходов управления линейной системой в условиях неопределенности ее параметров. В работе рассматривается линейный стационарный объект, переменные состояния которого подлежат измерению. Предполагая, что ряд несложных предположений выполнен, выбирается закон управления, обеспечивающий асимптотическую устойчивость по всем переменным объекта, но не по переменным его регулятора.

2. Постановка задачи

Рассмотрим линейный стационарный объект управления вида:

$$\dot{x} = Ax + \left(\sum_{i=1}^q \theta_i D_i\right)x + Bu, \quad (1)$$

где $x \in R^n$ - вектор переменных состояния доступный измерениям, θ_i - неизвестный постоянный параметр, A , D_i и B - известные числовые матрицы, u - управление.

Предполагая, что система (1) полностью управляема, требуется выбором управления u обеспечить асимптотическую устойчивость положения равновесия $x = 0$.

3. Синтез алгоритма управления

Для повышения уровня читабельности предлагаемого ниже материала, рассмотрим модель системы (1) для случая $i = 1$. Тогда уравнение (1) примет вид:

$$\dot{x} = Ax + \theta Dx + Bu, \quad (2)$$

где пара $(A + \beta D, B)$ - полностью управляема для любых значений β .

Выберем управление u в следующем виде:

$$u = -BB^T Px, \quad (3)$$

где матрица $P = P^T > 0$ является решением матричного уравнения Риккати вида:

$$A^T P + PA + \hat{\theta}(D^T P + PD) - PBB^T P = -2\alpha P, \quad (4)$$

где $\hat{\theta}$ - настраиваемый параметр, функция $\alpha > 0$ выбирается из следующего условия:

$$\operatorname{Re} \lambda_i \{\alpha I + A + \hat{\theta} D\} > 0. \quad (5)$$

Именно условие (5) обеспечивает наряду с полной управляемостью пары $(A + \beta D, B)$ существование положительно определенного решения уравнения (4) [8].

Для настройки параметра $\hat{\theta}$, воспользуемся следующей вычислительной процедурой:

$$\dot{\hat{\theta}} = \frac{\gamma}{2} x^T (D^T P + PD) x, \quad (6)$$

где коэффициент $\gamma > 0$ влияет на скорость изменения параметра $\hat{\theta}$.

Допущение. Предположим, что существует малый коэффициент $\gamma > 0$, такой что величина $\dot{\hat{\theta}}$ незначительна и для производной по матрице P выполнено условие:

$$\rho P - \dot{P} > 0,$$

где число $\rho > 0$.

Основной результат предлагаемой работы сформулирован в следующей теореме.

Теорема. Пусть пара $(A + \beta D, B)$ - полностью управляема для любых значений β . Пусть *допущение* выполнено, тогда существует малое положительное число γ такое, что закон управления (3) при выполнении условия (5), обеспечивает асимптотическую устойчивость положения равновесия $x = 0$.

Доказательство. Рассмотрим функцию Ляпунова вида:

$$V = x^T P x + \tilde{\theta}^2 / \gamma, \quad (7)$$

где $\tilde{\theta} = \theta - \hat{\theta}$ - параметрическая невязка.

Дифференцируя уравнение (7), получаем:

$$\dot{V} = x^T \dot{P} x + 2\tilde{\theta} \dot{\tilde{\theta}} / \gamma + x^T (A^T P + PA + \theta(D^T P + PD) - 2PBVP) x. \quad (8)$$

Из уравнения (4) находим:

$$A^T P + PA - PBVP = -2\alpha P - \hat{\theta}(D^T P + PD) \quad (9)$$

и, подставляя в (8), имеем следующее неравенство:

$$\dot{V} \leq x^T \dot{P} x + 2\tilde{\theta} \dot{\tilde{\theta}} / \gamma + x^T (-2\alpha P - \hat{\theta}(D^T P + PD)) x + x^T \theta(D^T P + PD) x. \quad (10)$$

Учитывая, что $\tilde{\theta} = \theta - \hat{\theta}$ и $\dot{\tilde{\theta}} = \dot{\theta} - \dot{\hat{\theta}} = -\dot{\hat{\theta}}$ для выражения (10), получаем:

$$\begin{aligned} \dot{V} &\leq x^T \dot{P} x - \tilde{\theta} x^T (D^T P + PD) x + x^T (-2\alpha P + \tilde{\theta}(D^T P + PD)) x = \\ &= x^T \dot{P} x - 2\alpha x^T P x. \end{aligned} \quad (11)$$

Предположим, что коэффициент γ выбран таким, что скорость изменения параметра $\dot{\hat{\theta}}$ незначительна и матрица $\dot{P} = \frac{\partial P}{\partial \hat{\theta}} \dot{\hat{\theta}} + \frac{\partial P}{\partial \alpha} \dot{\alpha}$ мала (см. *допущение*). Тогда, найдется такое α , что для неравенства (11) будет выполнено:

$$\dot{V} \leq -\delta x^T P x \leq 0, \quad (12)$$

где число $\delta > 0$.

Из неравенства (12) следует, что положение равновесия $x = 0$ устойчиво по Ляпунову, вектор x - ограниченный и квадратично интегрируемый, а сигнал $\hat{\theta}$ - ограничен. Ограниченность x и $\hat{\theta}$ обеспечивает ограниченность правой части дифференциального уравнения (2) с законом управления (3). Последнее, гарантирует выполнение следующего условия:

$$\lim_{t \rightarrow \infty} x(t) = 0,$$

что наряду с устойчивостью по Ляпунову положения равновесия $x = 0$, обеспечивает также и его асимптотическую устойчивость.

Замечание 1. Следует отметить, что выбор параметра γ представляет собой отдельную задачу, решение которой не имеет места в рамках предлагаемой работы.

Замечание 2. Выбор параметра α , удовлетворяющего условию (5), может производиться из следующих соображений:

$$\alpha = \alpha_0 + |\hat{\theta}|, \quad (13)$$

где число α_0 выбирается из условия:

$$\operatorname{Re} \lambda_i \{ \alpha_0 I + A + \hat{\theta}_0 D \} > 0. \quad (14)$$

Заключение

Результаты предлагаемой статьи не претендуют на универсальность и общность, а лишь расширяют класс подходов управления линейной системой в условиях неопределенности ее параметров. В работе рассматривается линейный стационарный объект, переменные состояния которого подлежат измерению. Предполагая, что ряд несложных условий выполнен, выбирается закон управления, обеспечивающий асимптотическую устойчивость по части переменных состояния системы, а именно по всем переменным объекта, но не его регулятора.

Литература

1. Фомин В.Н., Фрадков А.Л., Якубович В.А. Адаптивное управление динамическими объектами. М. Наука, 1981.
2. Фрадков А.Л. Адаптивное управление в сложных системах. М.: Наука, 1990.
3. Андриевский Б.Р., Фрадков А.Л., Избранные главы теории автоматического управления с примерами на языке MATLAB. СПб.: Наука, 1999.
4. Мирошник И.В., Никифоров В.О., Фрадков А.Л. Нелинейное и адаптивное управление сложными механическими системами. СПб.: Наука, 2000.
5. Никифоров В.О. Робастное управление линейным объектом по выходу // Автоматика и телемеханика. 1998. № 9.
6. Никифоров В.О., Фрадков А.Л. Схемы адаптивного управления с расширенной ошибкой. Обзор // Автоматика и телемеханика. 1994. №9.
7. Бобцов А.А., Лямин А.В., Сергеев К.А. Синтез закона адаптивного управления для стабилизации не точно заданных нестационарных объектов // Изв. Вузов Приборостроение. - 2001. - №3.
8. Дроздов В.Н., Мирошник И.В., Скорубский В.И. Системы автоматического управления с микроЭВМ. Л.: Машиностроение. Ленингр. отд-ие, 1989.

КРОНЕКЕРОВСКАЯ МАТРИЧНАЯ МОДЕЛЬ ДИНАМИЧЕСКИХ ПРОЦЕССОВ С МОДУЛЯЦИЕЙ

Л.В. Кожевникова, А.В. Ушаков

Решается задача конструирования модели динамических процессов с модуляцией. Для указанных целей используются возможности кронекеровских векторных и матричных структур, получаемую при этом избыточную размерность модели предлагается редуцировать с помощью оператора сужения.

Введение. Постановка задачи

Системы управления и следящие системы с модуляцией составляют заметную часть практики автоматического управления. Модуляторами в таких системах являются: сельсины, поворотные трансформаторы, индуктивные датчики, полудисковые модуляторы лучистой энергии и т.д. [1].

Однако теоретические исследования процессов в системах с модуляцией, иногда именуемых системами на несущей переменного тока [2], в последнее время заметно сократились, причем это произошло на фоне интенсификации исследования динамических систем с использованием возможностей метода пространства состояния (МПС) [3, 4].

Метод пространства состояния характеризуется значительной алгебраизацией общей теории систем, в рамках которой получены хорошие наработки применительно к кронекеровским матричным структурам [5]. Обнаруживается, что возможности этих структур могут быть эффективно использованы применительно к исследованию систем с модуляцией, причем получаемые модельные представления процессов с модуляцией оказываются максимально адаптированными к инструментарию МПС. Проблемам конструирования моделей динамических процессов с модуляцией на основе используемых возможностей кронекеровских векторных и матричных структур, инвариантных относительно размерности вход – выходных отношений посвящается данная работа.

Свойства векторных и матричных кронекеровских структур

Для целей погружения в сформулированную выше проблему приведем определение векторных и матричных кронекеровских структур, а так же те их свойства, которые непосредственно сориентированы на построение матричной модели процессов с модуляцией.

Определение 1 (О.1). Кронекеровским произведением двух векторов X и Y , $X \in R^n$, $Y \in R^m$, называется вектор $X \otimes Y$, составленный из сепаратных произведений $\{X_i Y_j; i = \overline{1, n}; j = \overline{1, m}\}$ их элементов так, что становится справедливым представление

$$X \otimes Y = col\{X_i Y_j; i = \overline{1, n}; j = \overline{1, m}\}, \text{ где } X \otimes Y \in R^{nm} \quad (1)$$

Примечание 1 (П.1). Очевидно, кроме кронекеровского произведения $X \otimes Y$ векторов может быть построено также произведение $Y \otimes X$ векторов, причем, в общем случае эти произведения оказываются не коммутативными так, что

$$X \otimes Y \neq Y \otimes X$$

Определение 2 (О.2). Если размерности векторов X и Y одинаковы, то на их кронекеровском произведении $X \otimes Y$ может быть построено согласованное сужение этого произведения $(X \otimes Y)_S$, задаваемого представлением

$$(X \otimes Y)_S = col\{X_i Y_i; i = \overline{1, n}\} \quad (2)$$

Примечание 2 (П.2). Согласованное сужение кронекеровского векторного произведения $X \otimes Y$ может быть осуществлено с помощью оператора сужения с матрицей S вида

$$S = \text{diag}\{[0_{i-1}^T : 1 : 0_{n-i}^T]; i = \overline{1, n}\} \quad (3)$$

так, что становится справедливой запись

$$(X \otimes Y)_S = S(X \otimes Y) \quad (4)$$

В качестве одного из свойств кронекеровского произведения векторов рассмотрим правило дифференцирования кронекеровских векторных произведений по скалярному параметру, причем в основном сосредоточимся на случае, когда скалярным параметром является время.

Свойство 1 (СВ.1). Дифференцирование векторной кронекеровской структуры в виде их кронекеровского произведения осуществляется по правилам дифференцирования сложной функции, представленной в мультипликативной форме так, что

$$\frac{d}{dt}(X(t) \otimes Y(t)) \stackrel{\Delta}{=} (X(t) \otimes \dot{Y}(t)) = \dot{X}(t) \otimes Y(t) + X(t) \otimes \dot{Y}(t) \quad (5)$$

$$\begin{aligned} \frac{d}{dt}(X(t) \otimes Y(t) \otimes Z(t)) &\stackrel{\Delta}{=} (X(t) \otimes \dot{Y}(t) \otimes Z(t)) = \\ &\dot{X}(t) \otimes Y(t) \otimes Z(t) + X(t) \otimes \dot{Y}(t) \otimes Z(t) + X(t) \otimes Y(t) \otimes \dot{Z}(t) \end{aligned} \quad (6)$$

Определение 3 (О.3). Кронекеровским произведением прямоугольных матриц $A \in R^{n \times m}$, $B \in R^{p \times q}$ называется матрица $(A \otimes B)$ размерности $(np \times mq)$, составленная в силу соотношения

$$A \otimes B = \text{col}\{\text{row}(A_{i,j}B; j = \overline{1, m}); i = \overline{1, n}\} \quad (7)$$

Примечание 3 (П.3): Кронекеровское произведение произвольных прямоугольных матриц не обладает коммутативностью так, что

$$A \otimes B \neq B \otimes A \quad (8)$$

Задача конструирования матричной модели динамических процессов с модуляцией в своей основе использует квадратные матрицы, коими являются матрицы состояния системы, конечномерного источника внешнего воздействия и конечномерного источника модулирующего сигнала, поэтому ниже имеется в виду класс квадратных матриц.

Определение 4 (О.4). Кронекеровской суммой квадратных матриц $A \in R^{n \times n}$ и $B \in R^{m \times m}$ называется матрица $(A \oplus B)$, размерности $(nm \times nm)$, составленная в силу соотношения

$$A \oplus B = A \otimes I_B + I_A \otimes B, \quad (9)$$

где I_A, I_B - единичные матрицы, согласованные по размерности соответственно с матрицами A и B.

Примечание 4 (П.4). Для кронекеровской суммы квадратных матриц A и B, а в общем случае и произвольного числа матриц, существует альтернативное название – преобразование Сильвестра матриц, обозначаемое $Si\{A, B\}$, тогда оказывается справедливой запись

$$A \oplus B \stackrel{\Delta}{=} A \otimes I_B + I_A \otimes B \stackrel{\Delta}{=} Si\{A, B\} \quad (10)$$

Для случая 3-х квадратных матриц A, B, C кронекеровская сумма или их преобразование Сильвестра будет записано в форме

$$Si\{A, B, C\} = A \oplus B \oplus C = A \otimes I_B \otimes I_C + I_A \otimes B \otimes I_C + I_A \otimes I_B \otimes C \quad (11)$$

Отметим тут же, что как и кронекеровское произведение матриц кронекеровская сумма не коммутативна.

Кронекеровские матричные структуры обладают следующими свойствами.

Свойство 2 (СВ.2). Алгебраический спектр собственных значений кронекеровского произведения $A \otimes B$ квадратных матриц $A \in R^{n \times n}$ и $B \in R^{m \times m}$ как матричной функции от матриц обладает тем свойством, что его элементы образованы попарными произведениями собственных значений кронекеровски перемножаемых матриц

$$\sigma\{A \otimes B\} = \{\mu_k : \det(\mu I - A \otimes B) = 0; \mu_k = \lambda_{A_i} \lambda_{B_j}; i = \overline{1, n}; j = \overline{1, m}; k = \overline{1, mn}\} \quad (12)$$

Свойство 3 (СВ.3). Алгебраический спектр собственных значений кронекеровской суммы $A \oplus B$ квадратных матриц $A \in R^{n \times n}$ и $B \in R^{m \times m}$ как матричной функции от матриц обладает тем свойством, что его элементы образованы попарными суммами собственных значений кронекеровски суммируемых матриц

$$\sigma\{A \oplus B\} = \{\nu_l : \det(\nu I - A \oplus B) = 0; \nu_l = \lambda_{A_i} + \lambda_{B_j}; i = \overline{1, n}; j = \overline{1, m}; l = \overline{1, mn}\} \quad (13)$$

В (12) и (13) λ_{A_i} и λ_{B_j} собственные значения матриц A и B .

Сделаем следующее примечание к свойствам (СВ.2) и (СВ.3).

Примечание 5 (П.5). Алгебраические спектры собственных значений кронекеровских произведений $A \otimes B$ и $B \otimes A$ в силу (12) совпадают, аналогичным свойством обладают и спектры кронекеровских сумм $A \oplus B$ и $B \oplus A$.

Свойство 4 (СВ.4). Определитель кронекеровского произведения квадратных матриц удовлетворяет соотношению

$$\det(A \otimes B) = (\det A)^m (\det B)^n, \quad (14)$$

где $A \in R^{n \times n}$ и $B \in R^{m \times m}$.

Свойство 5 (СВ.5). След кронекеровской суммы квадратных матриц удовлетворяет соотношению

$$\text{tr}(A \oplus B) = m \cdot \text{tr} A + n \cdot \text{tr} B, \quad (15)$$

где $A \in R^{n \times n}$ и $B \in R^{m \times m}$.

Свойство 6 (СВ.6). Ранг кронекеровского произведения квадратных матриц удовлетворяет условию

$$\text{rang}(A \otimes B) = \text{rang} A \cdot \text{rang} B, \quad (16)$$

где $A \in R^{n \times n}$ и $B \in R^{m \times m}$.

Для целей решения поставленной задачи полезно напомнить основные свойства кронекеровских матричных произведений произвольных матриц, которые оказываются необходимыми при преобразованиях матричных композиций, содержащих в своем составе эти произведения.

Свойство 7 (СВ.7).

$$(P \otimes Q)(W \otimes V) = PW \otimes QV \quad (17)$$

Свойство 8 (СВ.8).

$$(P + Q) \otimes R = P \otimes R + Q \otimes R \quad (18)$$

$$P \otimes (Q + R) = P \otimes Q + P \otimes R \quad (19)$$

$$P \otimes (Q \otimes R) = (P \otimes Q) \otimes R \quad (20)$$

В (17) – (20) матрицы P, Q, R, W, V имеют произвольные размерности, не противоречащие правилам перемножения и сложения матриц.

Свойство 9 (СВ.9).

$$P \otimes Q = (P \otimes I_Q)(I_P \otimes Q) \quad (21)$$

$$(P_1 \otimes Q_1)(P_2 \otimes Q_2) \dots (P_K \otimes Q_K) = (P_1 P_2 \dots P_K) \otimes (Q_1 Q_2 \dots Q_K) \quad (22)$$

$$(P \otimes Q)^{-1} = P^{-1} \otimes Q^{-1} \quad (23)$$

$$I \otimes (P_1 P_2 \dots P_K) = (I_{P_1} \otimes P_1)(I_{P_2} \otimes P_2) \dots (I_{P_K} \otimes P_K) \quad (24)$$

В выражениях (21) – (24) I_Q, I_P и т.д. – единичные матрицы, по размерности согласованные с соответствующими матрицами Q, P и другими аналогичными.

Свойство 10 (СВ.10). Оператор сужения кронекеровского произведения векторов с матрицей сужения S удовлетворяет соотношению

$$S(PX \otimes QZ) = S(P \otimes Q)(X \otimes Z) \quad (25)$$

Основной результат

Воспользуемся приведенными в предыдущем разделе свойствами векторных и матричных кронекеровских структур для целей построения динамической модели процессов в линейной многомерной непрерывной системе с модуляцией. При построении моделей процессов будем предполагать, что источник внешнего воздействия (ИВВ) является конечномерным и он представим автономной системой; будем предполагать, что модулирующий сигнал также является конечномерным и потому источник модулирующего сигнала (ИМС) также представим автономной системой. Таким образом полное априорное описание задачи приобретает вид

$$\dot{X}(t) = FX(t) + Gv(t); X(0); Y(t) = CX(t) \quad (26)$$

$$\dot{Z}(t) = \Gamma Z(t); Z(0); g(t) = PZ(t) \quad (27)$$

$$Z_M(t) = \Gamma_M Z_M(t); Z_M(0); g_M(t) = P_M Z_M(t) \quad (28)$$

В модели (26) многомерной непрерывной системы X – векторы состояния; v – вектор внешнего воздействия; Y – вектор выхода; $X \in R^n$; $v, Y \in R^m$; F, G, C – матрицы состояния входа и выхода соответственно; $F \in R^{n \times n}$, $C^T, G \in R^{n \times m}$.

В модели (27) источника внешнего воздействия Z и g – вектора состояния и выхода ИВВ соответственно; $Z \in R^l$; $g \in R^m$; Γ, P – матрицы состояния и выхода; $\Gamma \in R^{l \times l}$; $P \in R^{m \times l}$.

В модели (28) источника модулирующего сигнала Z_M и g_M – векторы состояния и выхода ИМС соответственно; $Z_M \in R^K$; $g_M \in R^m$; Γ_M, P_M – матрицы состояния и выхода ИМС; $\Gamma_M \in R^{K \times K}$; $P_M \in R^{m \times K}$.

Процесс формирования модулированного внешнего воздействия $v(t)$ представим в форме

$$v(t) = \text{col}\{g_j(t) \cdot \overline{g_{Mj}(t)}; j = \overline{1, m}\} \quad (29)$$

Нетрудно видеть, что процесс модуляции внешнего воздействия в форме (29) допускает представление его в виде кронекеровского произведения векторов с последующим сужением, что может быть представлено в форме:

$$v(t) = S(g(t) \otimes g_M(t)) \quad (30)$$

Если теперь учесть правила формирования $g(t)$ и $g_M(t)$ представленных в (27) и (28), то (30) в силу свойств кронекеровских произведений матриц может быть записано в форме

$$v(t) = S(g(t) \otimes g_M(t)) = S(PZ(t) \otimes P_M Z_M(t)) = S(P \otimes P_M)(Z(t) \otimes Z_M(t)) \quad (31)$$

Выражение (31) представляет модулированный сигнал $v(t)$ как функцию состояния системы с вектором состояния $Z(t) \otimes Z_M(t)$.

Сформируем динамическую систему, с вектором состояния $Z(t) \otimes Z_M(t)$, опираясь на модели (27) и (28), описывающих процессы по компонентам введенного вектора являющегося кронекеровским произведением векторов, а также свойства матричных кронекеровских структур. В результате получим следующую цепочку равенств

$$\begin{aligned} \frac{d}{dt}(Z(t) \otimes Z_M(t)) &\stackrel{\Delta}{=} (Z(t) \otimes \dot{Z}_M(t)) = \dot{Z}(t) \otimes Z_M(t) + Z(t) \otimes \dot{Z}_M(t) = \\ &= \Gamma Z(t) \otimes Z_M(t) + Z(t) \otimes \Gamma_M Z_M(t) = (\Gamma \otimes I_{\Gamma_M} + I_{\Gamma} \otimes \Gamma_M)(Z(t) \otimes Z_M(t)) = \\ &= (\Gamma \oplus \Gamma_M)(Z(t) \otimes Z_M(t)); \quad Z(0) \otimes Z_M(0) \end{aligned} \quad (32)$$

Для целей дальнейших исследований продолжим процесс построения автономной модели динамической системы с модуляцией, для чего введем в рассмотрение составной вектор состояния

$$\tilde{X} = \begin{bmatrix} X \\ Z \otimes Z_M \end{bmatrix} \quad (33)$$

и сформулируем утверждение.

Утверждение 1 (У.1). Процессы в непрерывной системе (26) с модулированным внешним воздействием (29), компоненты которого задаются (27) и (28), могут быть представлены автономной системой

$$\dot{\tilde{X}}(t) = \tilde{F}\tilde{X}; \quad \tilde{X}(0) \quad (34)$$

$$X(t) = \tilde{C}_X \tilde{X}(t); \quad Y(t) = \tilde{C}_Y \tilde{X}(t), \quad (35)$$

где матричные компоненты соотношений (34), (35) вычисляются с помощью выражений

$$\tilde{F} = \begin{bmatrix} F & \vdots & GS(P \otimes P_M) \\ \dots & \vdots & \dots \\ 0 & \vdots & \Gamma \oplus \Gamma_M \end{bmatrix} \quad (36)$$

$$\tilde{C}_X = [I_X \vdots 0]; \quad \tilde{C}_Y = [C \vdots 0] \quad (37)$$

Доказательство.

Доказательство утверждения строится на покомпонентном формировании производной по времени от вектора (33) с использованием исходной модели (26) многомерной системы, представления (31) процесса формирования внешнего модулированного сигнала, а также соотношения (32).

Приведенное утверждение содержит решение поставленной задачи формирования кронекеровской матричной модели динамических процессов с модуляцией.

Заключение

Нетрудно видеть, что модель вида (34) – (37) оказалась универсальной, т.к. позволяет исследовать процессы как с модуляцией входного воздействия, так и без нее. В последнем случае, в выражении (29) достаточно положить $g_{mj}(t) \equiv 1; \quad j = \overline{1, m}$. Это означает, что источник модулирующего сигнала (27) вырождается в скалярный интегратор с единичным начальным состоянием, нулевой матрицей состояния Γ_M и единичной матрицей выхода P_M .

Литература

1. Сабинин Ю.А. Позиционные и следящие электромеханические системы: Учебное пособие. – СПб.: Энергоатомиздат, Санкт-Петербургское отд-ние, 2001.
2. Куракин К.И., Куракин Л.К. Анализ систем автоматического регулирования на несущей переменного тока. – М.: Машиностроение, 1978.
3. Заде Л., Дезоер Ч. Теория линейных систем. / Пер. с англ. М.: Наука, 1970.
4. Ту Ю. Современная теория управления. / Пер. с англ. М.: Наука, 1971.
5. Ланкастер П. Теория матриц. / Пер. с англ. М.: Наука, 1978.

УПРАВЛЕНИЕ ТРАЕКТОРНЫМ ДВИЖЕНИЕМ АВТОНОМНЫХ РОБОТОВ

И.В. Мирошник, А.Н. Шалаев

Рассматривается траекторная задача управления плоским движением автономных роботов, т.е. задача перемещения робота в рабочем пространстве по предписанной траектории, заданной в аналитическом виде. Основные результаты представлены задачей-ориентированной моделью пространственного движения и соответствующими нелинейными алгоритмами управления.

Введение

Подвижными роботами называются технические системы, приспособленные к самостоятельному перемещению в рабочем пространстве и предназначенные для автоматической транспортировки разнообразных предметов и механизмов [3–4, 6–7]. К данному классу можно отнести рельсовые транспортные средства, безрельсовые платформы и манипуляторы на подвижной основе, автоматические летательные, надводные и подводные аппараты, шагающие механизмы и проч.

Конструкция подвижного робота включает основание (корпус) и двигательную систему, обеспечивающую требуемое направление и скорость перемещения корпуса в рабочем пространстве. Важнейшее место в ряду подвижных роботов занимают *автономные роботы*, движение которых происходит в воздушной или водной среде, т.е. без контакта с опорной поверхностью. Как объект управления автономный робот является многоканальной существенно нелинейной динамической системой. Его математическое описание (*модель движения*) может быть получено с использованием уравнений Лагранжа или Ньютона-Эйлера, в которых силомоментные воздействия производятся двигательной системой. Особенности двигательной системы и взаимодействие с вязкой средой определяют основные отличия моделей движения автономных роботов от моделей твёрдого тела и роботов колесного типа.

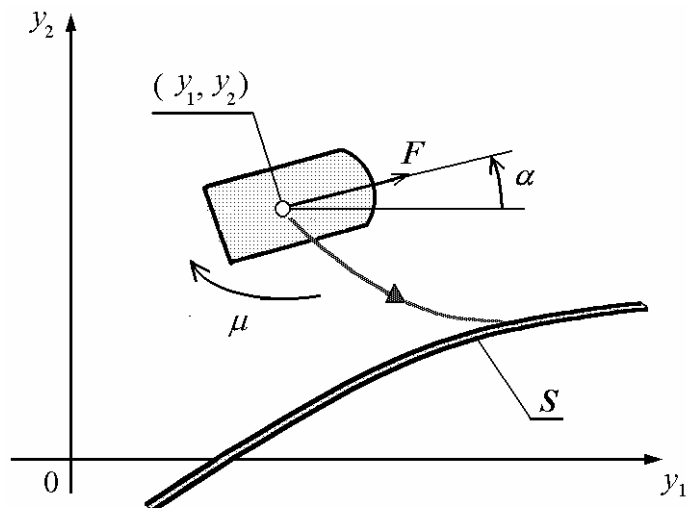


Рис. 1. Автономный робот и траектория движения S

Динамические модели роботов устанавливают связи выходных переменных системы, к которым относятся декартовы координаты платформы y_j и углы её ориентации α_j , и входных (управляющих) переменных, в качестве которых выступают продольная движущая сила F и вращающий момент μ (рис.1). Задача, решаемая системой управления подвижного робота, заключается в создании управляющих воздействий, обеспечивающих перемещение центра масс в рабочем пространстве с учётом сил сопротивления внешней среды и ограничивающих условий,

представленных в виде функциональных соотношений различных переменных системы. В траекторных задачах такие соотношения даются аналитическим описанием заданной траектории движения (трассы), и задача системы управления сводится к парированию отклонений от трассы (стабилизация объекта относительно заданной траектории) и обеспечению желаемого режима продольного перемещения. При этом полагается, что робот оснащён адекватной измерительной системой, позволяющей определить текущее положение и/или значение отклонения от заданной траектории.

В работе рассматривается наиболее распространённая *траекторная задача управления* плоским движением автономных роботов – задача перемещения в рабочем пространстве (плоскости R^2) по предписанной траектории (трассе) S , заданной в аналитическом виде. Используемый подход предусматривает нелинейное преобразование модели робота к системе задачно-ориентированных координат. Это даёт возможность свести сложную многоканальную задачу управления к ряду простых задач компенсации линейных и угловых отклонений, а затем с помощью стандартных приёмов нелинейной стабилизации [1,8] найти адекватные законы управления. Основные результаты представлены задачно-ориентированной моделью пространственного движения и соответствующими нелинейными алгоритмами управления. Результаты являются развитием известных решений задач управления пространственным движением, предложенных в [1, 3, 5].

2. Модель движения и основная задача управления

Положение корпуса робота как твёрдого тела в плоскости $Y=R^2$ характеризуется парой

$$y, \alpha,$$

где $y=(y_1, y_2)$ – вектор декартовых координат центра масс C , α – угловая ориентация. С углом α связана ортогональная матрица (матрица вращения)

$$T(\alpha) = \begin{vmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{vmatrix} \in SO(2), \quad T(0) = I,$$

соответствующая базису с началом в точке C .

Поступательное движение робота описывается моделью

$$\dot{y} = V, \quad m\dot{V} = T^T(\alpha)(F + F_c), \quad (1)$$

а вращательное –

$$\dot{\alpha} = \omega, \quad J\dot{\omega} = \mu + \mu_c \quad (2)$$

где $V \in R^2$ и ω – линейная и угловая скорости движения,

$$F = \begin{vmatrix} f \\ 0 \end{vmatrix}, \quad (3)$$

и $F_c \in R^2$ – заданные в относительной системе координат $Z \in R^2$ управляющие и возмущающие силовые воздействия, μ и μ_c – управляющий и возмущающие моменты, m и J – массо-инерционные характеристики. Отметим, что в относительной системе координат робота Z вектор линейных скоростей $V_z = (V_{z1}, V_{z2})$ находится как

$$V_z = T(\alpha)V. \quad (4)$$

Возмущающие воздействия рассматриваемого класса подвижных роботов определяются выражениями

$$F_c = -K_y(V_z)V_z. \quad (5)$$

$$\mu_c = -k_\alpha(\omega)\omega. \quad (6)$$

где

$$K_y = \begin{vmatrix} k_1|V_{z1}| & 0 \\ 0 & k_2|V_{z2}| \end{vmatrix}, \quad k_\alpha = k_3|\omega|, \quad (7)$$

k_i – коэффициенты вязкого трения.

Уравнения (1)–(6) описывают 3-канальную динамическую систему 6-го порядка с выходами y_1, y_2, α и входами (управлениями) f, μ .

В рассматриваемых здесь траекторных задачах требуемое поведение робота предписывается кривой, заданной в системе координат Y . Будем рассматривать движение относительно гладкого отрезка плоской кривой S , неявное описание которого имеет вид

$$\varphi(y) = 0, \quad (8)$$

а соответствующая локальная координата s (путь) определяется выражением

$$s = \psi(y), \quad (9)$$

где $\varphi(y), \psi(y)$ – гладкие функции. Воспользуемся ортогональными представлениями кривых [1,3], для которых функции φ и ψ таковы, что на линии S матрица Якоби

$$\Phi(y) = \begin{vmatrix} \partial\psi/\partial y \\ \partial\varphi/\partial y \end{vmatrix}$$

ортогональна. Тогда, обозначив $\Phi^*(y) = \Phi(y)|_{y \in S}$, можно записать

$$\Phi^*(y) = T(\alpha^*) \in SO(2),$$

где α^* – угол наклона касательной. Определённая таким образом матрица $T(\alpha^*)$ соответствует подвижному базису (базису Френе), который удовлетворяет следующему уравнению типа Френе [1–2]:

$$\dot{T}(\alpha^*) = \dot{s}\xi T(\alpha^*).$$

Введённые в рассмотрение ортогональные матрицы (и, следовательно, подвижные базисы) подчинены следующим дифференциальным соотношениям:

$$\dot{T}(\alpha) = \omega ET(\alpha), \quad (10)$$

$$\dot{T}(\alpha^*) = \dot{s}\xi(s)ET(\alpha^*), \quad (11)$$

где $E = \begin{vmatrix} 0 & 1 \\ -1 & 0 \end{vmatrix} \in so(2)$, ξ – кривизна. Свойство (10) может быть записано в явной форме

$$\dot{\alpha} = \omega, \quad (12)$$

а уравнение (11) – как

$$\dot{\alpha}^* = \dot{s}\xi. \quad (13)$$

Задача управления траекторным движением автономного робота ставится как задача поддержания голономных соотношений между выходами системы y_i , заданных в форме (8). Она дополняется описанием желаемого режима продольного движения основной точки $s(t)$, обычно устанавливаемого с помощью эталонной переменной $s^*(t)$ или эталонной скорости продольного движения $V_s^* = \dot{s}^*(t)$.

В установившемся режиме относительная ориентация подвижного объекта (по отношению к S) определяется следующим соотношением

$$T(\alpha) = T(\Delta\alpha)T(\alpha^*), \quad (14)$$

где $\Delta\alpha$ – угол разворота, $T(\Delta\alpha) \in SO(2)$, или в скалярной форме

$$\alpha = \Delta\alpha + \alpha^*. \quad (15)$$

Значение $\Delta\alpha$ определяется выражением

$$\Delta\alpha = \arctg\left(\frac{m}{k_2}\xi V^*\right) \quad (16)$$

Введём в рассмотрение ошибки тракторного движения. Нарушение условий (8) характеризуется ортогональным отклонением

$$e = \varphi(y), \quad (17)$$

принимая на множестве S нулевые значения. Текущие нарушения угловых соотношений (15) определяются угловой ошибкой

$$\delta = \alpha - \alpha^* - \Delta\alpha \quad (18)$$

или ортогональной матрицей

$$T(\delta) = T(\alpha)T^T(\alpha^*)T^T(\Delta\alpha). \quad (19)$$

Требуемая ориентация механизма соответствует нулевому значению δ или, соответственно, тождеству $T(\delta) = I$.

Таким образом, задача тракторного управления автономным роботом заключается в определении (в замкнутой форме) входов f и μ , которые обеспечивают:

(а) стабилизацию движения робота относительно кривой S , что подразумевает обнуление вектора пространственных отклонений e ;

(б) поддержание требуемого режима продольного движения механизма $s = s^*(t)$, часто задаваемого с помощью простейшей эталонной модели

$$\dot{s}^* = V_s^* = \text{const}, \quad (20)$$

или обнуления скоростной ошибки

$$\Delta V_s = V_s^* - \dot{s}. \quad (21)$$

3. Синтез алгоритма управления

Для построения системы управления тракторным движением прежде всего получим необходимые скоростные соотношения. Продифференцировав уравнения (9), (17) и (18) по времени и принимая во внимание (1), (14) и (15), найдём, что для достаточно малых отклонений e выполняется

$$\begin{vmatrix} \dot{s} \\ \dot{e} \end{vmatrix} = T(\alpha)\dot{y}, \quad (22)$$

$$\dot{\delta} + \dot{s}\xi = \omega. \quad (23)$$

Теперь требуемая задачно-ориентированная модель пространственного движения системы может быть получена дифференцированием уравнений (22) и (23) по времени. После необходимых подстановок при малых e и δ получаем

$$\begin{vmatrix} \ddot{s} \\ \ddot{e} \end{vmatrix} + \xi\dot{s}E^T \begin{vmatrix} \dot{s} \\ \dot{e} \end{vmatrix} = \frac{1}{m}T^T(\delta + \Delta\alpha)(F + F_c) \quad (24)$$

$$\ddot{\delta} + \xi\ddot{s} + \dot{\xi}\dot{s} = \frac{1}{J}(\mu + \mu_c). \quad (25)$$

Учитывая выражения (5)-(6), последние уравнения можно переписать в виде

$$\begin{vmatrix} \ddot{s} \\ \ddot{e} \end{vmatrix} + \left(\frac{1}{m}T^T(\delta + \Delta\alpha)K_yT(\delta + \Delta\alpha) + \xi\dot{s}E^T\right) \begin{vmatrix} \dot{s} \\ \dot{e} \end{vmatrix} = \frac{1}{m}T^T(\delta + \Delta\alpha) \begin{vmatrix} f \\ 0 \end{vmatrix}, \quad (26)$$

$$\ddot{\delta} + \xi\ddot{s} + \frac{k_3}{J}\dot{\delta} + \left(\frac{k_3}{J}\xi + \dot{\xi}\right)\dot{s} = \frac{1}{J}\mu. \quad (27)$$

Уравнения (26)-(27) полностью описывают поступательное движение механизма и его вращение относительно кривой S . Эти выражения используются для построения алгоритмов управления.

Перепишем уравнение (26) в виде скалярных выражений

$$\ddot{s} - \xi \dot{s} + \frac{1}{m}(k_1 \cos^2(\delta + \Delta\alpha) + k_2 \sin^2(\delta + \Delta\alpha)) \dot{s} + \frac{1}{2m}(k_1 - k_2) \sin(\delta + \Delta\alpha) \cos(\delta + \Delta\alpha) \dot{e} = \frac{1}{m} f \cos(\delta + \Delta\alpha), \quad (28)$$

$$\ddot{e} + \xi \dot{s}^2 + \frac{k_1 - k_2}{2m} \sin(2(\delta + \Delta\alpha)) \dot{s} + \frac{1}{m}(k_1 \sin^2(\delta + \Delta\alpha) + k_2 \cos^2(\delta + \Delta\alpha)) \dot{e} = \frac{1}{m} f \sin(\delta + \Delta\alpha). \quad (29)$$

При малых e, δ выражение (28) принимает вид

$$m\ddot{s} + k_2 \sin^2(\Delta\alpha) \dot{s} = f \cos(\Delta\alpha) \quad (30)$$

где $\Delta\alpha$ определяется выражением (16).

Зададим требуемый режим движения вдоль кривой S с помощью эталонной модели (20). Тогда алгоритм управления продольным движением находится как

$$f = k_s(V^*) \Delta V_s + m \xi V_s^{*2} \sin(\Delta\alpha). \quad (31)$$

Соответствующий выбор коэффициента $k_s > 0$ гарантирует асимптотическое устранение скоростной ошибки $\Delta V_s = V_s^* - \dot{s}$ и требуемое качество системы по переменной $V_s = \dot{s}$.

Полагая $\dot{s} = V_s^*$ и учитывая малость e и δ , перепишем уравнения (29), (27) в виде

$$m\ddot{e} + a(V_s^*) \dot{e} + V_s^* b(V_s^*) \delta = f_e(\xi, V_s^*), \quad (32)$$

$$J\ddot{\delta} + k_3 \dot{\delta} = \mu - \mu_\delta, \quad (33)$$

где $a(V_s^*) = k_1 \sin^2(\delta + \Delta\alpha) + k_2 \cos^2(\delta + \Delta\alpha)$, $b(V_s^*) = (k_1 - k_2) \cos(2\Delta\alpha) - \frac{m}{2} \xi V_s^* \sin(2\Delta\alpha)$,

$$f_e = -\xi V_s^{*2} \cos \Delta\alpha - \frac{1}{2m} V_s^* (k_1 - k_2) \sin(2\Delta\alpha), \quad \mu_\delta = (k_3 \xi + J \dot{\xi}) V_s^*.$$

Алгоритм управления поперечным движением выбирается как

$$\mu = \mu_\delta + \frac{1}{V_s^*} (k_{e1} \dot{e} + k_{e2} e) + k_{\delta 1} \dot{\delta} + k_{\delta 2} \delta, \quad (34)$$

где k_{e1} , k_{e2} , $k_{\delta 1}$ и $k_{\delta 2}$ – коэффициенты обратных связей, которые назначаются из условия получения требуемой асимптотической устойчивости моделей (32)–(33) и желаемых качественных показателей системы по e , что обеспечивает асимптотическое выполнение соотношения (8).

Таким образом, алгоритм управления траекторным движением автономного робота представлен локальными регуляторами (31) и (34). Его реализация предусматривает использование измерений относительного положения и ориентации системы (отклонений e , δ , и ΔV_s).

4. Пример

Рассмотрим задачи управления движением автономного робота по прямолинейной трассе.

Для нормализованного представления прямой запишем

$$\varphi(y_1, y_2) = -\sin \alpha^* y_1 + \cos \alpha^* y_2, \quad \psi(y_1, y_2) = \cos \alpha^* y_1 + \sin \alpha^* y_2$$

и, следовательно

$$\phi(y) \approx T(\alpha^*) = \begin{vmatrix} \cos \alpha^* & \sin \alpha^* \\ -\sin \alpha^* & \cos \alpha^* \end{vmatrix}.$$

Здесь $\dot{\alpha}^* = 0$, $\xi = 0$ и $\Delta\alpha = 0$.

Результаты моделирования системы управления автономным роботом в режиме прямолинейного движения представлены на рис. 2. Здесь: $\alpha^* = \pi/4$, $V^* = 2$, а коэффициенты обратной связи выбраны так: $k_s = -0.5$, $k_{e1} = -0.3$, $k_{e2} = -2$, $k_{\delta 1} = -3$, $k_{\delta 2} = -15$. Траектории движения из различных начальных положений робота и графики переходных процессов $e(t)$ и $V_s(t)$, соответствующие начальному положению $y_1 = 5$; $y_2 = 16$, демонстрируют устойчивость системы и хорошие качественные показатели траекторного движения робота.

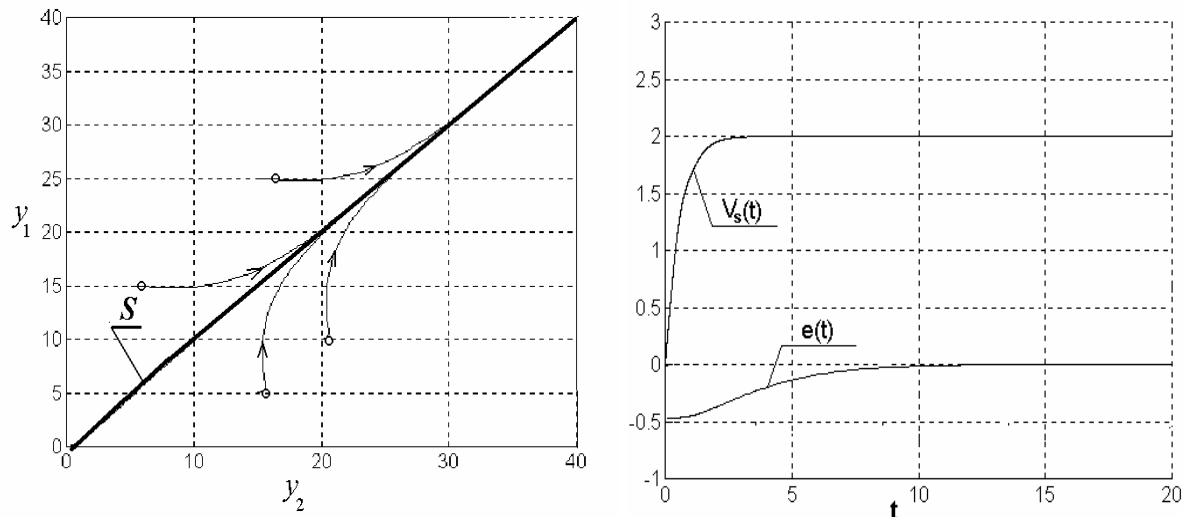


Рис. 2. Движение по прямолинейной трассе

Работа выполнена при поддержке Российского фонда фундаментальных исследований (Грант 99-01-00761) и Комплексной программы 19 Президиума РАН (2001, раздел 1.4).

Литература

1. Мирошник И.В., Фрадков А.Л., Никифоров В.О. Нелинейное и адаптивное управление сложными динамическими системами. СПб.: Наука, 2000.
2. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров. М.: Наука, 1984.
3. Бурдаков С.Ф., Мирошник И.В., Стельмаков Р.Э. Системы управления движением колёсных роботов. СПб: Наука, 2001. 232 с.
4. Canudas de Wit, C., B. Siciliano and G. Bastin (1996). Theory of robot control. Springer-Verlag, London.
5. Miroshnik, I.V. and V.O. Nikiforov (1996). Trajectory motion control and coordination of multilink robots. // Prepr. 13th IFAC World Congress. San-Francisco. Vol.A. PP.361–366.
6. Murray, R.M., I.L. Zexiang and S.S. Sastry (1993) A Mathematical Introduction to Robotic Manipulation. Boca Raton: CRS Press.
7. Dixon, W., D.M. Dawson, E. Zargelloglu, A. Behal. (2001) Nonlinear control of wheeled mobile robot. Springer-Verlag, London.
8. Isidori, A. (1995) Nonlinear control systems. 3rd edition. Springer-Verlag, Berlin.

ИСПОЛЬЗОВАНИЕ СВОЙСТВ МАТРИЦ НАД КОНЕЧНЫМИ ПОЛЯМИ ДЛЯ ОБНАРУЖЕНИЯ НЕУСТОЙЧИВЫХ ЦИКЛОВ И НЕПОДВИЖНЫХ СОСТОЯНИЙ ДВОИЧНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

А. А. Мельников, Е. В. Рукуйжа, А. В. Ушаков

На основе анализа свойств матриц состояния и входа линейной двоичной динамической системы (ЛДДС) над конечным полем GF(2) решается задача обнаружения неустойчивых циклов и неподвижных состояний ЛДДС. Результаты иллюстрируются примерами.

Введение. Постановка задачи.

Рассматривается линейная двоичная динамическая система [1–3], векторно-матричное описание которой задается в формах рекуррентной

$$x(k+1)=Ax(k)+BU(k); x(0) \quad (1)$$

$$y(k)=Cx(k)+HU(k); \quad (2)$$

и суммарной

$$x(k)=A^k x(0)+\sum_{i=0}^{k-1} A^{k-1-i} BU(i)=A^k x(0)+W(k) \cdot V(k) \quad (3)$$

$$y(k)=Cx(k)+HU(k)$$

моделей. В (1) - (3) x – n мерный вектор состояния ЛДДС, U – r мерная входная последовательность, Y – m мерная выходная последовательность, A – $(n \times n)$ матрица состояния, B – $(n \times r)$ матрица входа, C – $(m \times n)$ матрица выхода, H – $(m \times r)$ матрица "вход-выход", V – (k) вектор стратегии управления состоянием ЛДДС, задаваемый на первых k – тактах в форме

$$V(k)=col\left\{U(l); l=\overline{k-1,0}\right\} \quad (4)$$

$W(k)$ – матрица управляемости двоичной системы на первых k тактах, определяемая матричной структурой

$$W(k)=\left[B \mid AB \mid \dots A^{k-2} B \mid A^{k-1} B\right] \quad (5)$$

В дальнейшем используется для оценки свойств пары матриц (A, B) матрица управляемости (5) для случая $k = n$, в силу чего она представляется выражением

$$W=W(k)_{|k=n}=\left[B \mid AB \mid \dots A^{n-2} B \mid A^{n-1} B\right] \quad (6)$$

Ставится задача обнаружения неустойчивых циклов и неподвижных состояний двоичных динамических систем вида (1) - (3) на основе анализа свойств пары матриц (AB) . Анализ опирается на то обстоятельство, что часть свойств матриц над конечными полями идентичны свойствам матриц над бесконечными [4], а другая часть свойств порождена модулярной природой конечных полей.

Свойства матриц состояния и входа над конечными полями

Анализ свойств матрицы состояния проведем опираясь на понятие матричной функции от матрицы с учетом модулярной специфики конечного поля, а анализ свойств матрицы B , в котором в основном ограничимся случаем $r = 1$, будем проводить на предмет установления факта принадлежности матрицы B геометрическому спектру собственных векторов матрицы A , а также установления свойства управляемости пары (A, B) .

Основные результаты изложены в виде системы определений, утверждений и их доказательств.

Определение 1 (О.1) Матричной функцией $f(A)$, представляющей собой $(n \times n)$ матрицу от $(n \times n)$ матрицы A , заданных над конечным полем $GF(2)$ называется матричный ряд, порождаемый рядом от скалярной переменной α вида

$$f(\alpha) = a_0 + a_1\alpha + a_2\alpha^2 + \dots + a_p\alpha^p + \dots \quad (7)$$

путем замены α на A , так что $f(A)$ принимает вид:

$$f(A) = a_0I + a_1A + a_2A^2 + \dots + a_pA^p + \dots \quad (8)$$

В (7), (8) $a_i \in GF(2); i = 1, \overline{p, \dots}$; так, что суммирование и приведение подобных проводится по правилам модулярной арифметики. $\square$

Утверждение 1 (У.1) Матричная функция от матрицы, порожденная модулярным характеристическим полиномом матрицы A

$$D(\lambda) = \det(\lambda I + A) = \lambda^n + a_1\lambda^{n-1} + \dots + a_{n-1}\lambda + a_n \quad (9)$$

обнуляется этой матрицей, так что

$$D(A) = A^n + a_1A^{n-1} + a_2A^{n-2} + \dots + a_{n-1}A + a_nI = 0 \quad \square \quad (10)$$

Доказательство утверждения опирается на положение теоремы Гамильтона–Кэли [4], сформулированное для случая характеристического полинома матриц над бесконечным полем [1]. $\blacksquare$

Определение 2 (О.2) Квадратная $(n \times n)$ матрица A обладает свойством принадлежности показателю μ , где μ – целое положительное, если выполняется условие

$$A^\mu = I \quad \square \quad (11)$$

Утверждение 2 (У.2) Для того чтобы матрица A принадлежала показателю μ достаточно чтобы характеристический полином $D(\lambda)$ этой матрицы входил в разложение двучлена $\lambda^\mu + 1$ в форме

$$\lambda^\mu + 1 = Q(\lambda) \cdot D(\lambda) \quad (12)$$

где $Q(\lambda)$ модулярный многочлен степени $(\mu - n)$ $\square$

Доказательство (У.2) Сконструируем на модулярном многочлене (12) матричную функцию от матрицы

$$A^\mu + I = Q(A) \cdot D(A) = Q(A)(A^n + a_1A^{n-1} + a_2A^{n-2} + \dots) \quad (13)$$

В силу положения утверждения (У.1)

$$D(A) = 0 \quad (14)$$

тогда подстановка (14) в (13) дает

$$A^\mu + I = 0 \text{ или } A^\mu = I \quad \blacksquare$$

Определение 3 (О.3) Матрица A размерности $(n \times n)$ называется нильпотентной с индексом нильпотентности ν , где ν – целое положительное, если выполняется условие

$$A^\nu = 0 \quad \square \quad (15)$$

Утверждение 3 (У.3) Достаточным условием наличия у матрицы A свойства нильпотентности (15) с индексом ν является представимость ее характеристического полинома в виде

$$D(\lambda) = \det(\lambda I + A) = \lambda^n \quad \square \quad (16)$$

Доказательство утверждения строится на свойстве (10) матрицы обнулять свой характеристический полином. $\blacksquare$

Утверждение 4 (У.4) Индекс нильпотентности ν $(n \times n)$ –нильпотентной матрицы A удовлетворяет неравенству

$$2 \leq \nu \leq n \quad (17)$$

Доказательство утверждения строится на канонической форме нильпотентной матрицы. Если каноническая форма матрицы A представляет собой $(n \times n)$ клетку Жордана [4] с нулевыми диагональными элементами, то индекс нильпотентности ν ее равен значению n и является максимальным, что доказывает справедливость правой части неравенства (17). Если отсечением согласованных по размеру левых нулевых столбцов и нижних нулевых строк получается Жорданова клетка размера $\nu < n$, при этом индекс нильпотентности ν такой матрицы устанавливается прямым возведением в степень ν матрицы A , то очевидно, что минимальный размер Жордановой клетки с нулевыми диагональными элементами будет равен двум, что доказывает справедливость левой части неравенства (17).

■

Утверждение 5 (У.5) Если матрица входа B имеет размерность $(n \times 1)$ и при этом она является собственным вектором матрицы состояния A ЛДДС (1), то пространство управляемости ЛДДС имеет размерность равную единице. □

Доказательство утверждения строится на том, что размерность пространства управляемости системы (1) определяется рангом ее матрицы управляемости

$$W_y = [B \mid AB \mid A^2B \mid \dots \mid A^{n-1}B] \quad (18)$$

Воспользуемся свойством собственного вектора матрицы A . Если некоторый вектор ξ является собственным вектором матрицы A , то над конечным полем $GF(2)$ это обстоятельство может быть представлено в форме

$$A\xi = \xi \quad (19)$$

Подставим в (19) вместо ξ матрицу-столбец B , тогда на основе (19) получим матричные равенства

$$\begin{aligned} AB &= B; \\ A^2B &= A(AB) = AB = B; \\ &\vdots \end{aligned} \quad (20)$$

$$A^p B = A^{p-2}(A^2B) = A^{p-2} \cdot B = B$$

Если (20) подставить в (18), то для матрицы управляемости получим

$$W = [B \mid B \mid \dots B] = \text{row} \left\{ W_i = B; i = \overline{1, n} \right\} \quad (21)$$

Матрица управляемости вида (21) обладает тем свойством, что ранг ее равен единице. ■

Утверждение 6 (У.6) Пусть ЛДДС (1) характеризуется передаточной функцией

$$\Phi(d) = \frac{Y(d)}{U(d)} = \frac{M(d)}{\tilde{D}(d)} \quad (22)$$

где $Y(d), U(d)$ – образы соответственно выходной $y(k)$ и входной $u(k)$ последовательностей ЛДДС (1); $M(d), \tilde{D}(d)$ – модулярные многочлены относительно переменной d с коэффициентом из $GF(2)$; тогда характеристический полином матрицы A ЛДДС (1) и знаменатель $\tilde{D}(d)$ передаточной функции (22) этой системы связаны соотношением

$$D(\lambda) = D(d^{-1})|_{d^{-1}=\lambda} \quad (23)$$

причем $D(d^{-1})$ задается соотношением

$$D(d^{-1}) : \tilde{D}(d) = d^n D(d^{-1}) \quad \square \quad (24)$$

Доказательство утверждения строится на применении к векторно–матричному описанию (1) ЛДДС прямого D - преобразования [1,2] при нулевом начальном ее состоянии, тогда с использованием свойств D - преобразования получим

$$d^{-1}X(d)=AX(d)+BU(d); \quad (25)$$

$$Y(d)=CX(d)+HU(d) \quad (26)$$

где $X(d)$ – D образы состояния $x(k)$ входной последовательности $U(k)$ и выходной $Y(k)$ соответственно.

Разрешим соотношение (25),(26) относительно $Y(d)$, исключив из них $X(d)$. Тогда получим

$$Y(d)=\{C(d^{-1}I+A)^{-1}B+H\}U(d) \quad (27)$$

Нетрудно видеть, что (27) представляет собой передаточную функцию ЛДДС, записанную в мультипликативной форме так, что

$$\Phi(d)=C(d^{-1}I+A)^{-1}B+H=\frac{1}{\det(d^{-1}I+A)}C(\Delta(d^{-1}I+A))^T \cdot B+H \quad (28)$$

Из (28) следует

$$\det(d^{-1}I+A)=D(d^{-1}) \text{ откуда следует (23)} \quad \blacksquare$$

Основной результат

Для решения задачи, вынесенной в название статьи, введем и сформулируем дополнительные определения и утверждения.

Определение 4(О.4) ЛДДС (1) называется асимптотически устойчивой при $U(k) \equiv 0$, если существует значение k^* такое, что выполняется соотношение

$$x(k)=x[x(0);U(k) \equiv 0, k]=0 \text{ при } k \geq k^* \text{ и любых } x(0). \quad \square \quad (29)$$

Утверждение 7 (У.7) Если матрица A состояния ЛДДС (1) обладает индексом нильпотентности ν , то эта двоичная динамическая система будет асимптотически устойчива в смысле определения О.4 в форме (29), при этом k^* определяется равенством

$$k^*=\nu \quad \square \quad (30)$$

Доказательство утверждения строится на использовании суммарной модели (3) представления процессов ЛДДС по вектору состояния $x(k)$. Если в (3) положить условие $U(k) \equiv 0$ для любого k , то

$$x(k)=A^k x(0) \quad (31)$$

По условию утверждения матрица A является нильпотентной с индексом нильпотентности ν , в силу чего для $k \geq \nu$, выполняется равенство

$$A^k=0 \quad (32)$$

Соотношения (31)–(32) делают справедливым положение (29), где $k^*=\nu$, а $x(0)$ любое. $\blacksquare$

Доказанное утверждение сформулированное в негативной логике приводит к тому, что если матрица состояния не является нильпотентной, то такая ЛДДС не обладает свойством асимптотической устойчивости, в ней возможны неподвижные состояния, замкнутые неустойчивые циклы и особые режимы поведения в случае, если матрица B оказывается собственным вектором матрицы A.

Определение 5 (О.5) Пусть ЛДДС (1) характеризуется $(n \times n)$ матрицей состояния A, не обладающей свойством нильпотентности, тогда цикл длины T при $U(k) \equiv 0$, сформированный на подмножестве $\tilde{X}$ множеств состояний этой ЛДДС мощностью 2^n , будет называться неустойчивым, если условие

$$x_i(k) = x_i(k+T), \quad (33)$$

выполняется для любого $x_i(k) \in \tilde{X} \quad (i=0, T-1)$ $\square$

Выделим на множестве $\tilde{X}$ удовлетворяющем условию (33) частные случаи T , удовлетворяющие неравенствам

$$1 \leq T \leq 2^n - 1 \quad (34)$$

задаваемые следующими определениями и утверждениями

Определение 6 (О.6) Состояние $x(k)$ называется неподвижным, если оно удовлетворяет условию

$$x(k+1) = x(k) \quad \square \quad (35)$$

Условие (35) может быть реализовано следующими двумя способами, которые определим утверждениями.

Утверждение 8(У.8) ЛДДС (1), с $(n \times n)$ матрицей состояния A , при любой ее реализации всегда имеет в качестве неподвижного состояния нулевое состояние

$$x(k) = 0 \quad \square$$

Доказательство утверждения использует рекуррентную модель ЛДДС (1), в которой положено $U(k) \equiv 0$, так, что она принимает вид

$$x(k+1) = Ax(k) \quad (36)$$

Подстановка в (36) условия $x(k)=0$ дает при любой матрице A , для состояния перехода

$$x(k+1) = 0 \quad (37)$$

что приводит к удовлетворению (35) $\blacksquare$

Утверждение 9 (У.9) ЛДДС (1) с $(n \times n)$ матрицей состояния A произвольной реализации имеет в качестве неподвижного состояния состояние, которое является ее собственным вектором. $\square$

Доказательство утверждения опирается на определение собственного вектора ξ , который с учетом специфики конечного поля $GF(2)$, записывается в форме

$$A\xi = \xi \quad (38)$$

Подстановка $x(k)=\xi$ в (36) в силу (38) приводит к (35). $\blacksquare$

Необходимо сделать следующие примечания.

Примечание 1 (П.1)

Нетрудно видеть, что условию утверждения У.9 удовлетворяет любой вектор ξ , если матрица состояния ЛДДС (1) является единичной. $\square$

Рассмотрим теперь ситуацию, в которой $T = \mu$, где μ - показатель, которому принадлежит матрица A .

Если характеристический полином $D(\lambda) = \det(\lambda I + A)$ является неприводимым, то, как известно [5], показатель μ в этом случае связан с размерностью n соотношением

$$\mu = 2^n - 1 \quad (40)$$

Таким образом, в случае удовлетворяющем (40) все множество мощностью 2^n состояний ЛДДС (1), (36) разбивается на два подмножества, где одно представлено нулевым неподвижным состоянием, а второе образовано μ состояниями, где μ удовлетворяет (40), образующими замкнутый неустойчивый цикл.

Если характеристический полином $D(\lambda)$, является приводимым, то в зависимости от конкретной его реализации μ принимает одно из значений удовлетворяющее неравенству

$$n \leq \mu < 2^n - 1$$

В этом случае 2^n возможных состояний ЛДДС (1), (36) образует сумму подмножеств. Первое из них образуется нулевым неподвижным состоянием. Второе – образуется ρ неустойчивыми циклами, каждое из которых содержит μ состояний. Следующие подмножества образованы неподвижными состояниями соответствующими собственным векторам матрицы A и, наконец, последнее подмножество образовано собственными векторами матрицы A^{χ} составляющими замкнутый цикл из χ состояний, где

$$\chi = 2^n - 1 - \mu - [(\xi_i)] \quad (41)$$

$[(\xi_i)]$ – мощность множества из элементов (ξ_i) (38).

В ключе поставленной проблемы обнаружения неподвижных состояний и замкнутых неустойчивых циклов, рассмотрим теперь поведение ЛДДС (1) для случая ненулевой входной последовательности. При этом выделим ситуацию, когда матрица B размерности $(n \times 1)$ является собственным вектором матрицы A . Для этой ситуации сформулируем следующее утверждение.

Утверждение 10 (У.10) Пусть входная последовательность представляет собой унитарный код, так что

$$U(k) = 1(k) \quad (42)$$

Пусть матрица входа B размерности $(n \times 1)$ является собственным вектором матрицы A так, что выполняется соотношение

$$AB = B \quad (43)$$

Тогда, под действием такой входной последовательности в пространстве состояний ЛДДС (1), возникает вынужденное движение, образующее неустойчивый замкнутый цикл $\{x(k)=0; x(k+1)=B\}$ с периодом $T=2$. $\square$

Доказательство утверждения строится на рекуррентной модели ЛДДС (1) в предположении, что $x(k)=0$ и с учетом условий (42), (43), тогда рекуррентная модель для смежных $k, (k+1), (k+2)$ тактов приводит к результатам

$$x(k+1) = Ax(k) + BU(k) = A0 + B \cdot 1 = B \quad (44)$$

$$x(k+2) = Ax(k+1) + BU(k+1) = AB + B \cdot 1 = B + B = 0 \quad (45)$$

Из (44), (45) становится очевидным справедливость равенства

$$x(k+2) = x(k) \quad \blacksquare \quad (46)$$

Примеры

В качестве примеров, иллюстрирующих основные положения работы, рассмотрим три версии ЛДДС третьего порядка ($n=3$) со следующими реализациями матриц состояния A :

$$A = \text{col}\{[010], [001], [101]\}: D(\lambda) = \det(\lambda I + A) = \lambda^3 + \lambda^2 + 1 \quad (47)$$

$$A = \text{col}\{[010], [001], [111]\}: D(\lambda) = \det(\lambda I + A) = \lambda^3 + \lambda^2 + \lambda + 1 \quad (48)$$

$$A = \text{col}\{[010], [001], [100]\}: D(\lambda) = \det(\lambda I + A) = \lambda^3 + 1 \quad (49)$$

при этом все версии ЛДДС будут иметь матрицу входа B

$$B = [111]^T \quad (50)$$

Анализ информации, представленный (47) показывает, что матрица A (47) обладает неприводимым характеристическим полиномом $D(\lambda)$ поэтому она принадлежит показателю $\mu = 2^n - 1 = 7$. Как следствие, структура состояний ЛДДС с такой матрицей при $U(k) \equiv 0$ представлена двумя подмножествами. Первое подмножество состоит из нулевого состояния, являющегося неподвижным, второе подмножество образует неустойчивый цикл из оставшихся семи состояний. Матрица A

(47) с матрицей В (50) образует полностью управляемую пару, так что матрица управляемости

$$W = [B \mid AB \mid A^2 B] = \text{row} \{ [111]^T, [110]^T, [101]^T \}$$

обладает рангом равным 3, что соответствует выполнению условия управляемости $\text{rang} W = n$. Установленное свойство полной управляемости ЛДДС гарантирует соответствующим выбором входной последовательности перевод системы из любого начального состояния в любое конечное за фиксированное число тактов.

Анализ информации представленный (48) обнаруживает, что матрица А принадлежит показателю $\mu = 4$. Структура пространства ЛДДС для этого случая при $U(k) \equiv 0$ разбивается на 4 подмножества. Первое подмножество представлено неподвижным нулевым состоянием $x = [000]^T$. Второе подмножество представлено состояниями образующими замкнутый неустойчивый цикл с периодом равным 4. Третье подмножество образовано неподвижным состоянием $x = [111]^T$, которое является собственным вектором матрицы А. Четвертое подмножество представлено состояниями, образующими замкнутый неустойчивый цикл с периодом $T = 2$, каждое из которых $x(k) = [010]^T$ и $x(k) = [101]^T$ таково, что для них выполняются соотношения $x(k+2) = A^2 x(k) = x(k)$, является собственным вектором матрицы A^2 .

$$A^2 = \text{col} \{ [001], [111], [100] \} \quad \xi_1 = [010]^T, \xi_2 = [101]^T$$

Матрица А (48) и матрица В (50) образуют не полностью управляемую пару, т.к. матрица В (50) является собственным вектором матрицы А, как следствие при входной последовательности в виде унитарного входа $U(k) = 1(k)$ в структуре состояний ЛДДС возникает неустойчивый вынужденный цикл периода 2, содержащий нулевое состояние $x = [000]^T$ и состояние $x = B = [111]^T$.

Анализ информации представленный соотношением (49) показывает, что матрица А ЛДДС принадлежит показателю $\mu = 3$. Как следствие, при $U(k) \equiv 0$ структура пространства ЛДДС представлена четырьмя подмножествами. Первое подмножество образовано нулевым неподвижным состоянием. Второе и третье подмножества представлены неустойчивыми неподвижными циклами периода $T = \mu = 3$. Четвертое подмножество, представленное неподвижным состоянием $x = [111]^T$, которое является собственным вектором матрицы А. Пара матриц (49), (50) является не полностью управляемой, т.к. В-собственный вектор А. Как следствие, при входной последовательности в виде унитарного кода в поведении ЛДДС обнаруживается неустойчивый вынужденный цикл периода $T = 2$ на состояниях $x = [000]^T$ и $x = [111]^T$.

В заключении необходимо отметить, что полученная на основе анализа свойств матриц А и В в рассматриваемых примерах структура неподвижных состояний и неустойчивых циклов также обнаруживается таблично-графическим способом с использованием таблиц и графов переходов ЛДДС [6].

Литература

1. Гилл А. Линейные последовательностные машины. М.: Наука, 1974.
2. Фараджев Р. Г. Линейные последовательностные машины. М.: Сов. радио, 1975.
3. Бохман Д., Постхофф Х. Двоичные динамические системы. М.: Энергоатомиздат, 1986.
4. Гантмахер Ф. Д. Теория матриц. М.: Наука, 1988.
5. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. / Пер. с англ. М.: Мир, 1976.
6. Алгебраические методы в теории устройств дискретной автоматики и телемеханики // Труды лаборатории телемеханики СПб ГИТМО (ТУ) / под ред. А. В. Ушакова. СПб: СПбГИТМО (ТУ), 2001.

ЧАСТИЧНАЯ УСТОЙЧИВОСТЬ И АТТРАКТИВНОСТЬ НЕТРИВИАЛЬНЫХ МНОЖЕСТВ¹

О.В. Слита, И.В. Мирошник

В работе представлен сравнительный анализ ключевых концепций пространственной динамики (инвариантности и аттрактивности нетривиальных множеств) и частичной устойчивости гладких нелинейных систем. Рассмотрены примеры, показывающие различие свойств аттрактивности и частичной устойчивости.

Введение

В традиционных задачах теории управления представление о желаемом поведении системы обычно связывается с притяжением интегральных кривых к точечным положениям равновесия (точечным аттракторам) и устойчивостью по всем переменным состояниям. В последние годы внимание специалистов привлекают более сложные теоретические и прикладные проблемы, в которых необходимо обеспечить инвариантность и устойчивость системы по отношению к нетривиальным множествам – кривым и многомерным поверхностям [2–4, 7–10]. Подобные задачи тесно связаны с концепцией устойчивости по части переменных состояния x_i или некоторым функциям от x_i , т.е. так называемой *частичной устойчивостью* динамической системы [4, 7–10]. С другой стороны, анализ частичной устойчивости часто ассоциируется с определенными геометрическими концепциями и может быть выполнен с помощью дифференциально-геометрической теории управления.

В статье устанавливаются взаимосвязи понятий аттрактивности многомерных множеств (подмногообразий) и частичной устойчивости нелинейных динамических систем. Также рассмотрены примеры, иллюстрирующие теоретические положения.

1. Притягивающие множества и частичная устойчивость

Рассмотрим гладкую автономную нелинейную динамическую систему, описываемую обыкновенным дифференциальным уравнением вида

$$\dot{x} = f(x), \quad (1)$$

где x – n -мерный вектор состояния системы.

Исследование свойств аттрактивности и частичной устойчивости предполагает анализ поведения системы относительно некоторого v -мерного геометрического объекта Z^* (кривой, поверхности, подмногообразия), все точки которого удовлетворяют уравнению

$$\varphi(x) = 0, \quad (2)$$

где для всех $x \in Z^*$

$$\text{rank } \varphi(x) = n - v,$$

а локальные координаты определяются как

$$z = \psi(x). \quad (3)$$

Специфическое поведение системы (1) в некоторой окрестности множества Z^* связано с понятиями инвариантности и аттрактивности.

Гладкая поверхность Z^* называется *инвариантным подмногообразием* системы (1), если для всех начальных состояний $x_0 \in Z^*$

$$x(t, x_0) \in Z^*,$$

и *аттрактором (притягивающим подмногообразием)*, если выполняется условие равномерной аттрактивности

$$\lim_{t \rightarrow \infty} \text{dist}(x(t), Z^*) = 0 \quad . \quad (4)$$

где $\text{dist}(x, Z^*)$ – расстояние от точки x до множества Z^* .

$$\text{Введем в рассмотрение выходную переменную системы (1)} \\ \xi = \varphi(x), \quad (5)$$

и определим решение системы (1),(5):

$$\xi(t) = \xi(t, x_0) = \varphi(x(t, x_0)).$$

Точка $\xi^* = 0$ называется (частичным) *положением равновесия* системы (1)-(5), когда для всех $x_0 \in Z^*$

$$\xi(t, x_0) = 0, \quad (6)$$

а система (1),(5) в положении равновесия $\xi = 0$ называется *частично асимптотически устойчивой* (по отношению к функции ξ), когда выполняется условие

$$\lim_{t \rightarrow \infty} |\xi(t, x_0)| = 0 \quad (7)$$

(равномерно по x_0).

Понятие частичного положения равновесия связано с инвариантностью системы: $\xi = 0$ является частичным положением равновесия тогда и только тогда, когда для произвольной точки $x_0 \in Z^*$ решение $x(t, x_0)$ системы (1) принадлежит инвариантному множеству Z^* . Однако в общем случае инвариантное множество Z^* частично устойчивой системы не является притягивающим множеством, так как (7) не гарантирует, что соответствующая траектория приближается к Z^* согласно (4), и наоборот. Возникает необходимость изучения взаимосвязи указанных понятий и нахождения условий, при которых данные понятия являются эквивалентными.

2. Условия аттрактивности и частичной устойчивости

Для исследования свойств аттрактивности произвольного множества Z^* и анализа частичной устойчивости системы (1), (5) рассмотрим нелинейное преобразование координат (3)–(5), т.е.

$$\begin{vmatrix} \xi \\ z \end{vmatrix} = \begin{vmatrix} \varphi(x) \\ \psi(x) \end{vmatrix}. \quad (8)$$

Вектор $\xi(t)$ связан с отклонением от множества Z^* , а вектор локальных координат $z(t)$ характеризует продольное движение системы.

Тождество $\xi(t) \equiv 0$ соответствует траекториям системы $x(t, x_0)$, принадлежащим поверхности Z^* (где поведение системы определяется вектором $z(t)$), т.е. является признаком инвариантности подмногообразия. С другой стороны, это же тождество является признаком того, что точка $\xi(t) = 0$ является частичным положением равновесия системы (1), (5). Дифференцируя (8) по времени и подставляя (1), получаем, что для $x_0 \in Z^*$ имеет место тождество

$$\frac{\partial \varphi}{\partial x} f(x) = 0, \quad (9)$$

которое представляет необходимое и достаточное условие как инвариантности подмногообразия Z^* , так и частичного равновесия системы в точке $\xi = 0$.

Введем в рассмотрение открытое множество $\mathcal{E}(Z^*)$ – окрестность поверхности Z^* и матрицу Якоби отображения (8):

$$\left| \begin{matrix} \Phi(x) \\ \Psi(x) \end{matrix} \right| = \left| \begin{matrix} \partial\varphi/\partial x \\ \partial\psi/\partial x \end{matrix} \right|.$$

Будем полагать, что для всех $x \in \mathcal{E}(Z^*)$ матрица Якоби не вырождена, т.е. $\det \left| \begin{matrix} \Phi(x) \\ \Psi(x) \end{matrix} \right| \neq 0$, и существует гладкое обратное отображение

$$x = r(\xi, z). \quad (10)$$

Матрица Якоби преобразования (10) имеет вид

$$[R_\xi(\xi, z) \ R_z(\xi, z)] = \left[\frac{\partial r}{\partial \xi} \ \frac{\partial r}{\partial z} \right].$$

Далее ограничимся анализом случая, когда локальные координаты z_i поверхности Z^* выбраны так, что для всех $x \in Z^*$ выполняется $\Phi \Psi^T = 0$ и, следовательно, $R_\xi^T R_z = 0$ [3].

Рассмотрим поведение системы в достаточно малой окрестности $\mathcal{E}(Z^*)$ и найдем связь переменных $\xi(t)$ и $\text{dist}(x(t), Z^*)$, принимая во внимание, что для произвольной точки $x \in \mathcal{E}(Z^*)$ можно записать

$$\text{dist}(x, Z^*) = |x - x^*|, \quad (11)$$

где $x^* \in Z^*$ – ближайшая к x точка множества Z^* . Определим значение $z^* = \psi(x^*)$, запишем $x^* = r(0, z^*)$ и

$$R_z^T(x - x^*) = 0 \quad (12)$$

Разложим функцию (10) в ряд Тейлора в точке $(0, z^*)$, соответствующей значению $x = x^* \in Z^*$. Пренебрегая остаточным членом ряда, для достаточно малых ξ получим:

$$x \cong r(0, z^*) + \frac{\partial r}{\partial \xi} \xi + \frac{\partial r}{\partial z} (z - z^*) = x^* + [R_\xi^* \ R_z^*] \begin{bmatrix} \xi \\ z - z^* \end{bmatrix}, \quad (13)$$

где $R_\xi^* = R_\xi(0, z)$, $R_z^* = R_z(0, z)$. В силу (12) $z \cong z^*$ и, следовательно,

$$x - x^* \cong R_\xi^*(z) \xi.$$

Тогда

$$|x - x^*|^2 \cong \xi^T Q_\xi(z) \xi = |\xi|_{Q_\xi},$$

где $Q_\xi = R_\xi^{*T} R_\xi^*$ – матрица $n \times n$. Подставляя полученное выражение в (11), получаем

$$\text{dist}(x(t), Z^*) = |\xi|_{Q_\xi}, \quad (14)$$

где

$$|\xi|_{Q_\xi} = (\xi^T Q_\xi(z) \xi)^{1/2}.$$

Таким образом, принимая во внимание, что

$$|\xi| = (\xi^T \xi)^{1/2},$$

заключаем, что основное отличие свойства (4) *аттрактивности* подмногообразия Z^* от свойства (7) *частичной устойчивости* системы определяется матрицей $Q_\xi(z)$. Свойства с очевидностью эквивалентны при $Q_\xi = I$, а также в тех случаях, когда в рассматриваемой области $\mathcal{E}(Z^*)$ матрица $Q_\xi(z)$ ограничена.

Отметим, что матрица $Q_\xi(z)$ является блоком *метрической матрицы* (см. [3]) и характеризует поперечное искажение евклидовой метрики пространства при переходе к

координатам (ξ, z) . Значительное искажение метрики и приводит к несовпадению рассматриваемых здесь свойств нелинейной системы (см. п.3).

3. Изучение частных случаев

Будем исследовать поведение систем второго порядка относительно скалярной выходной переменной $\xi(t)$ и одномерной поверхности (кривой) $Z^* \in R^2$. В общем случае одна и та же кривая может быть описана различными уравнениями (2). Выбор функции $\varphi(x)$, определяющей выходную переменную системы и неявную форму описания Z^* , устанавливает метрические свойства пространства (см. п.2). При этом достигается либо совпадение свойств аттрактивности и частичной устойчивости, либо имеет место лишь одно из указанных свойств.

Далее ограничимся рассмотрением случая, когда множество Z^* является отрезком прямой: $x_1 = x_2$ в области $x_1 > 0, x_2 > 0$.

Рассмотрим линейную систему

$$\begin{aligned}\dot{x}_1 &= \frac{\sqrt{2}}{2} - k \frac{x_1}{2} + k \frac{x_2}{2} \\ \dot{x}_2 &= \frac{\sqrt{2}}{2} + k \frac{x_1}{2} - k \frac{x_2}{2}\end{aligned}\tag{15}$$

с выходной переменной

$$\xi = -\frac{\sqrt{2}}{2}x_1 + \frac{\sqrt{2}}{2}x_2\tag{16}$$

и соответствующее множество $Z^* \in R^2$ (отрезок прямой)

$$-\frac{\sqrt{2}}{2}x_1 + \frac{\sqrt{2}}{2}x_2 = 0$$

с локальной координатой

$$z = \frac{\sqrt{2}}{2}x_1 + \frac{\sqrt{2}}{2}x_2.$$

Нетрудно получить, что преобразование к координатам (ξ, z) не приводит к искажениям метрики пространства (см. рис. 1), и $Q_\xi = 1$. Тогда по формуле (14) находим

$$dist(x, Z^*) = |\xi|,$$

что определяет эквивалентность свойства (4) аттрактивности линии Z^* и свойства (7) частичной устойчивости системы по ξ .

На рис. 1 для случая $k > 0$ представлены интегральные кривые системы в пространстве R^2 , а также переходные процессы $|\xi(t)|$ и $dist(x(t), Z^*)$. При $t \rightarrow \infty$ кривые $x(t)$ приближаются к прямой Z^* , что свидетельствует об аттрактивности Z^* . Переходные процессы $\xi(t)$ и $dist(x(t), Z^*)$ демонстрируют совпадение свойств аттрактивности и (частичной) устойчивости системы по переменной ξ .

Рассмотрим нелинейную систему

$$\begin{aligned}\dot{x}_1 &= \frac{-2k_1x_1x_2^2 - k_2x_1(x_1^2 - x_2^2)}{2(x_1^2 + x_2^2)} \\ \dot{x}_2 &= \frac{-2k_1x_1^2x_2 + k_2x_2(x_1^2 - x_2^2)}{2(x_1^2 + x_2^2)}\end{aligned}\tag{17}$$

с выходной переменной

$$\xi = \frac{x_1^2 - x_2^2}{2}. \quad (18)$$

Здесь множество $Z^* \in R^2$ (отрезок прямой) определяется выражением

$$\frac{x_1^2 - x_2^2}{2} = 0,$$

а вектор локальных координат –

$$z = x_1 x_2.$$

В этом случае преобразование к координатам (ξ, z) вызывает существенное искажение метрики пространства (см. рис.2). Найдем $Q_\xi = \frac{1}{\sqrt{2z}}$ и по формуле (14) –

$$\text{dist}(x, Z^*) = (2z)^{-1/4} |\xi|.$$

Последнее выражение показывает, что в данном случае характер связи свойств аттрактивности Z^* и частичной устойчивости системы по ξ зависит от локальной переменной $z(t)$.

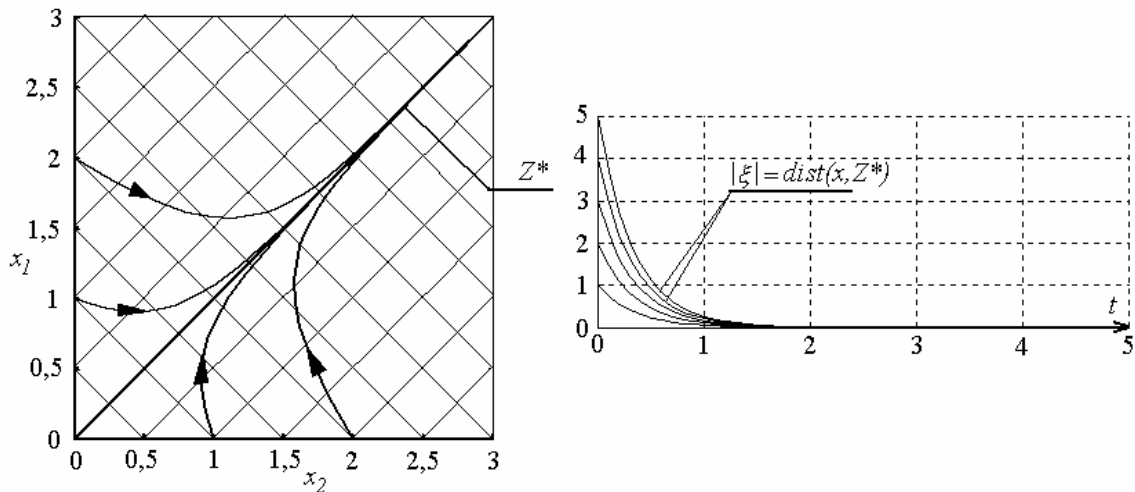


Рис. 1. Траектории и переходные процессы системы (15)–(16)

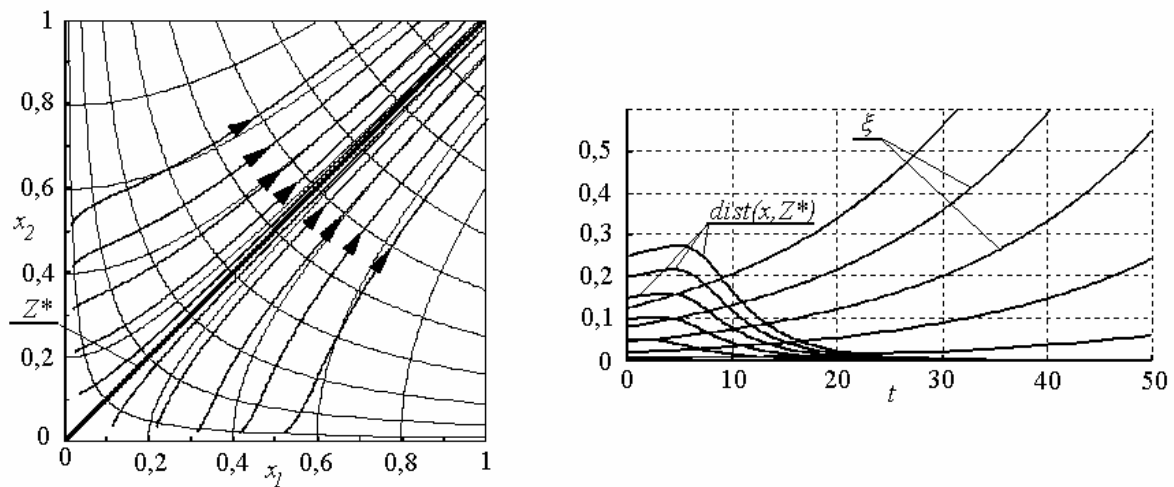


Рис. 2. Траектории и переходные процессы системы (17)–(18)

На рис. 2 для случая $k_1 < k_2 < 0$ представлены интегральные кривые и переходные процессы: $\xi(t)$ и $\text{dist}(x(t), Z^*)$. Переходные процессы $\xi(t)$ показывают, что система является неустойчивой по выходной переменной. В то же время кривые $x(t)$ при $t \rightarrow \infty$ приближаются к прямой Z^* : $\text{dist}(x, Z^*) \rightarrow 0$, что свидетельствует об аттрактивности Z^* .

Рассмотрим нелинейную систему

$$\begin{aligned}\dot{x}_1 &= -\frac{k_1}{2}x_1 + \frac{k_2x_2^2}{4x_1} - \frac{k_2x_1}{4} \\ \dot{x}_2 &= -\frac{k_1}{2}x_2 + \frac{k_2x_1^2}{4x_2} - \frac{k_2x_2}{4}\end{aligned}\quad (19)$$

с выходной переменной

$$\xi = \frac{x_2^2 - x_1^2}{x_1^2 + x_2^2}. \quad (20)$$

Здесь множество $Z^* \in R^2$ (отрезок прямой) определяется выражением

$$\frac{x_2^2 - x_1^2}{x_1^2 + x_2^2} = 0,$$

а вектор локальных координат

$$z = \frac{x_1^2 + x_2^2}{2}.$$

Преобразование к координатам (ξ, z) вызывает существенное искажение метрики пространства (см. рис.3). Находим $Q_\xi = z$ и по формуле (14) получаем

$$\text{dist}(x, Z^*) = z^{1/2} |\xi|,$$

т.е., свойства аттрактивности линии Z^* и частичной устойчивости системы по ξ не эквивалентны, и их связь зависит от локальной переменной $z(t)$.

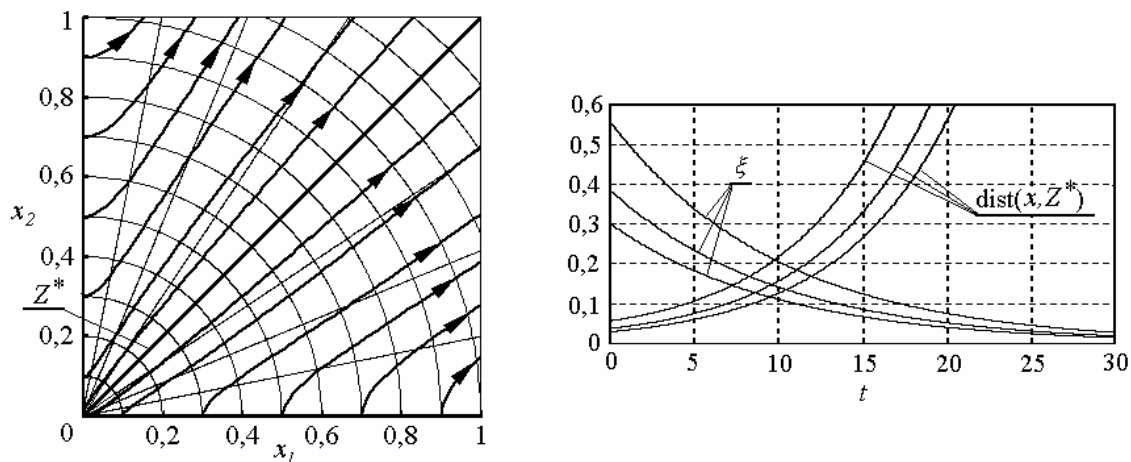


Рис. 3. Траектории и переходные процессы системы (19)–(20)

На рис. 3 для случая $k_1 < 0$, $k_2 > 0$ представлены интегральные кривые и переходные процессы $\xi(t)$ и $\text{dist}(x(t), Z^*)$. Как показывают переходные процессы $\xi(t)$, по выходной переменной система является асимптотически устойчивой: $\xi \rightarrow 0$ при $t \rightarrow \infty$. Тем не менее кривые $x(t)$ при $t \rightarrow \infty$ удаляются от прямой Z^* , что свидетельствует об отсутствии свойства аттрактивности.

Работа выполнена при поддержке Российского фонда фундаментальных исследований (Грант 99-01-00761) и Комплексной программы 19 Президиума РАН (2001, раздел 1.4).

Литература

1. Румянцев В.В., Озиранер А.С. Устойчивость и стабилизация движения по отношению к части переменных. М.: Наука, 1987.
2. Воротников В.И. Задачи и методы исследования устойчивости и стабилизации движения по отношению к части переменных: направления исследования, результаты, особенности // Автоматика и телемеханика. 1993. № 3. С.3-62.
3. Мирошник И.В., Фрадков А.Л., Никифоров В.О. Нелинейное и адаптивное управление сложными динамическими системами. СПб.: Наука, 2000.
4. Мирошник И.В., Королев С.М. Анализ динамики и управление пространственным движением нелинейных динамических систем // АиТ, 2000. № 1. С.29-40.
5. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров. М.: Наука, 1984.
6. Гантмахер Ф.Р. Теория матриц. М.: Наука, 1988.
7. Isidory A. Nonlinear control systems. 3 edition. NY: Springer-Verlag, 1995.
8. Miroshnik I.V. Functional Decomposition and Nonlinear Control of MIMO Systems // 14 IFAC World Congress. Beijing, 1999. V.L, P.199-204.
9. Miroshnik I.V. Attractors and partial stability of nonlinear control systems.// 5th IFAC Symposium on Nonlinear Control Systems (NOLCOS'01). Preprints. Vol 3. St. Petersburg, 2001, pp. 848-853.
10. Vorotnikov V.I. On problems of partial stability and stabilization of nonlinear systems. //5th IFAC Symposium on Nonlinear Control Systems (NOLCOS'01). Preprints. Vol 3. St. Petersburg, 2001, pp. 854-859.

ОЦЕНКА ЗАПАСОВ УСТОЙЧИВОСТИ СИСТЕМ С ИНТЕРВАЛЬНЫМИ ПАРАМЕТРАМИ

С.А. Сударчиков, А.В. Ушаков

В работе рассматриваются системы, спроектированные методами модального управления на заданные качество и требование к относительной интервальности матриц состояния. В качестве дополнительного показателя систем предлагается использовать интервальное значение запаса устойчивости по фазе проектируемых систем, конструируемого на семействе полиномов В.Л.Харитонов. Приводится пример.

Введение. Постановка задачи

Рассматривается многомерная непрерывная система с интервальной матрицей состояния, имеющая векторно-матричное описание

$$\dot{x}(t) = [F]x(t) + Gg(t); Y(t) = Cx(t), \quad (1)$$

которое получено агрегированием объекта управления (ОУ) с интервальной матрицей состояния

$$\dot{x}(t) = [A]x(t) + Bu(t); Y(t) = Cx(t), \quad (2)$$

и регулятора, реализующего закон управления (ЗУ)

$$u(t) = K_g g(t) - Kx(t). \quad (3)$$

В (1) – (3) x, g, y, u – соответственно вектор состояния системы и ОУ, экзогенное воздействие, регулируемый выход, вектор управления; $x \in R^n; g, y \in R^m; u \in R^r$. $[F], [A], G, B, C$ – соответственно интервальные матрицы состояния системы и ОУ, матрицы с фиксированными параметрами входа системы, управления и выхода; $[F], [A] \in R^{n \times n}; G \in R^{n \times m}; B \in R^{n \times r}; C \in R^{m \times n}; K_g$ – матрица прямых связей по экзогенному воздействию $g(t)$; K – матрица прямых связей (ОС) по состоянию; $K_g \in R^{r \times m}, K \in R^{r \times n}$. Матрицы системы (1), (ОУ) (2) и ЗУ (3) связаны соотношением

$$F = A - BK; G = BK_g \quad (4)$$

В дальнейшем используется представление интервальных матриц в виде аддитивной композиции их медианных A_0, F_0 и симметричных интервальных $[\Delta A], [\Delta F]$ частей, так что матрицы $[A]$ и $[F]$ могут быть представлены в виде

$$[A] = A_0 + [\Delta A]; [F] = F_0 + [\Delta F] = A_0 - BK + [\Delta A] \quad (5)$$

Введем в рассмотрение такую характеристику системных матриц, как оценка относительной интервальности δ_{I*} , задав ее для матриц $[A], [F]$ соотношениями

$$\delta_{IA} = \frac{\|[\Delta A]\|}{\|A_0\|}; \quad \delta_{IF} = \frac{\|[\Delta F]\|}{\|F_0\|} = \frac{\|[\Delta A]\|}{\|A_0 - BK\|} \quad (6)$$

Корректность введения понятия оценка относительной интервальности матриц задаваемая с помощью соотношений (6) опирается на то, что симметричные интервальные компоненты $[\Delta A] = [\Delta F]$ исходных интервальных матриц $[A], [F]$, характеризуются одной и той же матричной нормой на всех угловых реализациях их интервальных элементов.

При постановке задачи синтеза закона управления (3) методами модального управления формулируются требования к структуре собственных значений медианного компонента $F_0 = A_0 - BK$, который доставляет системе необходимую динамику и

качество, а также к обеспечению интервальной матрицы состояния необходимой оценки относительной интервальности в форме

$$\delta_{IF} = \frac{\|\Delta F\|}{\|F_0\|} = \frac{\|\Delta A\|}{\|A_0 - BK\|} \leq \delta_{IR}, \quad (7)$$

за счет выбора матрицы обратных связей K .

Матрица прямых связей K_g выбирается из соображения ориентации системы (1) с законом управления (3) относительно экзогенного воздействия $g(t)$ так, чтобы медианная версия системы характеризовалась единичным отношением вход-выход при неподвижном состоянии. Этому условию удовлетворяет равенство

$$\Phi_0(s) = C(sI - F_0)^{-1}BK_g|_{s=0} = -CF_0^{-1}BK_g = I, \quad (8)$$

где $I - (m \times m)$ единичная матрица. Дополним задачу еще одним требованием к показателям системы (1), в качестве которого предлагается использовать значение ее запаса устойчивости по фазе. Этот интервальный показатель также оценим как в абсолютной интервальной постановке, так и в относительной, с оценками $\Delta_{I\Delta\varphi}, \delta_{i\Delta\varphi}$, задаваемых выражениями

$$\Delta_{I\Delta\varphi} = \|\Delta\Delta\varphi\| \text{ и } \delta_{i\Delta\varphi} = \frac{\Delta[\Delta\Delta\varphi]}{\Delta\varphi_0}, \quad (9)$$

где $\Delta\varphi_0$ и $\Delta\Delta\varphi$ связаны соотношением $[\Delta\varphi] = \Delta\varphi_0 + [\Delta\Delta\varphi]$.

Поставленная задача, контроля запаса устойчивости системы (1), с интервальной матрицей состояния $[F]$, решается с привлечением аппарата семейства характеристических полиномов В.Л.Харитонова, на котором конструируются эквивалентные разомкнутые системы.

Основной результат

Введем в рассмотрение понятие *эквивалентная разомкнутая система, сопровождающая некоторый характеристический полином*.

Определение 1 (0.1) Система типа одномерный вход – одномерный выход, полученная путем размыкания единичной отрицательной обратной связи по выходу, с передаточной функцией прямой цепи

$$W(s) = \frac{M(s)}{N(s)}, \quad (10)$$

где $M(s), N(s)$ – полиномы от s с вещественными коэффициентами удовлетворяющие требованиям минимальной фазовости [1] называется эквивалентной разомкнутой системой сопровождающей характеристический полином $D(s)$, если выполняется условие

$$D(s) = M(s) + N(s). \quad (11)$$

В развитие разрабатываемых положений сформулируем следующие утверждения.

Утверждение 1 (У1) Системы с передаточными функциями прямых цепей

$$W_1(s) = \frac{M_1(s)}{N_1(s)} \text{ и } W_2(s) = \frac{M_2(s)}{N_2(s)} \text{ соответственно, замкнутые отрицательной единичной}$$

обратной связью по выходу и представленные в одном и том же базисе, обладают идентичными показателями сходящихся по множеству начальных состояний $\|x(0)\| = x_0$

при отсутствии экзогенного воздействия $g(t)$ процессов $x(t) = x[x(0), g(t) \equiv 0, t]$, если

$$M_1(s) + N_1(s) = M_2(s) + N_2(s) = D(s) \quad \square \quad (12)$$

Доказательство утверждения строится на непосредственном использовании определения (1) с учетом условия (12). ■

С целью дальнейших исследований выделим на классе эквивалентных разомкнутых систем сопровождающий данный характеристический полином системы с фиксированным порядком астатизма при помощи следующего определения.

Определение 2 (О.2) Передаточная функция прямой цепи вида

$$W(s) = \frac{a_{n-(v-1)}s^{v-1} + a_{n-(v-2)}s^{v-2} + \dots + a_{n-1}s + a_n}{s^v(s^{n-v} + a_1s^{n-v-1} + \dots + a_{n-v})} \quad (13)$$

доставляет замкнутой единичной отрицательной обратной связью системе астатизм порядка v и при этом является передаточной функцией эквивалентной разомкнутой системы, сопровождающей характеристический полином

$$D(s) = s^n + a_1s^{n-1} + a_2s^{n-2} + a_3s^{n-3} + \dots + a_{n-1}s + a_n \quad \square \quad (14)$$

Определение 2 позволяет сформулировать следующее утверждение

Утверждение 2 (У.2) На множестве значений порядка астатизма v , передаточные функции вида (13) порождают класс эквивалентных систем в смысле положений утверждения 1. ■

Доказательство утверждения строится на вычислении характеристических полиномов замкнутых систем, образованных замыканием разомкнутых систем с передаточной функцией прямой цепи (13), приводящим для всех порядков астатизма к представлению (14). ■

Применим положения сформулированных определений и утверждений к оценке запасов устойчивости разомкнутых систем, конструируемых на семействе полиномов Харитонова, полученных из (1).

Напомним основные положения теоремы Харитонова В.Л. [2] об устойчивости интервальных систем. Пусть интервальная матрица $[F]$ состояния системы (1) обладает интервальным характеристическим полиномом (ИХП) $[D(\lambda)]$, то есть полиномом с интервальными коэффициентами, который вычисляется в силу соотношения

$$[D(\lambda)] = \det([F] - \lambda I) = [a_0]\lambda^n + [a_1]\lambda^{n-1} + \dots + [a_{n-1}]\lambda + [a_n] \quad (15)$$

где $[a_i] = [\underline{a_i}, \overline{a_i}] ; i = (1, n)$. Тогда ИХП (15) на множестве угловых реализаций интервальных коэффициентов $[a_i]$ будет Гурвицевым, а следовательно будет устойчива система (1) с интервальной матрицей $[F]$, если будут Гурвицевыми следующие четыре полинома с фиксированными параметрами

$$D_1(\lambda) = \underline{a_0}\lambda^n + \underline{a_1}\lambda^{n-1} + \overline{a_2}\lambda^{n-2} + \overline{a_3}\lambda^{n-3} + \dots + \underline{a_{n-1}}\lambda + \underline{a_n}, \quad (16)$$

$$D_2(\lambda) = \underline{a_0}\lambda^n + \overline{a_1}\lambda^{n-1} + \overline{a_2}\lambda^{n-2} + \underline{a_3}\lambda^{n-3} + \dots + \underline{a_{n-1}}\lambda + \overline{a_n}, \quad (17)$$

$$D_3(\lambda) = \overline{a_0}\lambda^n + \underline{a_1}\lambda^{n-1} + \underline{a_2}\lambda^{n-2} + \overline{a_3}\lambda^{n-3} + \dots + \overline{a_{n-1}}\lambda + \underline{a_n}, \quad (18)$$

$$D_4(\lambda) = \overline{a_0}\lambda^n + \overline{a_1}\lambda^{n-1} + \overline{a_2}\lambda^{n-2} + \underline{a_3}\lambda^{n-3} + \dots + \underline{a_{n-1}}\lambda + \overline{a_n}. \quad (19)$$

Нетрудно видеть, что если применить положения определения (1) и (2) к семейству полиномов В.Л.Харитонова, то на этом семействе может быть сконструировано семейство эквивалентных разомкнутых систем (СЭРС), сопровождающих эти полиномы при различных порядках астатизма. Так для астатизма $v=1$ передаточные функции семейства эквивалентных разомкнутых систем примут вид.

$$W_1(s) = \frac{a_n}{s(\underline{a_0}s^{n-1} + \underline{a_1}s^{n-2} + \overline{a_2}s^{n-3} + \overline{a_3}s^{n-4} + \dots + \underline{a_{n-1}})} \quad (20)$$

$$W_2(s) = \frac{\overline{a_n}}{s(\overline{a_0 s^{n-1}} + \overline{a_1 s^{n-2}} + \overline{a_2 s^{n-3}} + \overline{a_3 s^{n-4}} + \dots + \overline{a_{n-1}})} \quad (21)$$

$$W_3(s) = \frac{\overline{a_n}}{s(\overline{a_0 s^{n-1}} + \overline{a_1 s^{n-2}} + \overline{a_2 s^{n-3}} + \overline{a_3 s^{n-4}} + \dots + \overline{a_{n-1}})} \quad (22)$$

$$W_4(s) = \frac{\overline{a_n}}{s(\overline{a_0 s^{n-1}} + \overline{a_1 s^{n-2}} + \overline{a_2 s^{n-3}} + \overline{a_3 s^{n-4}} + \dots + \overline{a_{n-1}})}, \quad (23)$$

для случая $\nu = 2$ это семейство определяется следующими выражениями.

$$W_1(s) = \frac{\overline{a_{n-1} s + a_n}}{s^2(\overline{a_0 s^{n-2}} + \overline{a_1 s^{n-3}} + \overline{a_2 s^{n-4}} + \dots + \overline{a_{n-2}})}, \quad (24)$$

$$W_2(s) = \frac{\overline{a_{n-1} s + a_n}}{s^2(\overline{a_0 s^{n-2}} + \overline{a_1 s^{n-3}} + \overline{a_2 s^{n-4}} + \dots + \overline{a_{n-2}})}, \quad (25)$$

$$W_3(s) = \frac{\overline{a_{n-2} s + a_n}}{s^2(\overline{a_0 s^{n-2}} + \overline{a_1 s^{n-3}} + \overline{a_2 s^{n-4}} + \dots + \overline{a_{n-2}})}, \quad (26)$$

$$W_4(s) = \frac{\overline{a_{n-1} s + a_n}}{s^2(\overline{a_0 s^{n-2}} + \overline{a_1 s^{n-3}} + \overline{a_2 s^{n-4}} + \dots + \overline{a_{n-2}})}. \quad (27)$$

И наконец для $\nu = n$ передаточные функции СЭРС принимают вид.

$$W_i(s) = \frac{V_i(s)}{S^n} (i = 1, n), \quad (28)$$

$$\text{где } V_i(s) = \overline{a_0^{-1} D_i(s)} - s^n \text{ для } i = 1, 2; V_i(s) = \overline{a_0^{-1} D_i(s)} - s^n \text{ для } i = 3, 4. \quad (29)$$

Введение эквивалентных разомкнутых систем, позволяет ввести в рассмотрение такой удобный пользовательский показатель как запас устойчивости по фазе, конструируемый с их помощью. В этой связи оказывается полезным следующее определение.

Определение 3 (О.3) Запас устойчивости по фазе $\Delta\varphi_i$ некоторой системы с единичной отрицательной обратной связью по выходу и передаточной функцией $W_i(s)$ прямой цепи определяется соотношением

$$\Delta\varphi_i = \pi + \arg W_i(j\omega) \Big|_{\omega=\omega_*} \quad (30)$$

где

$$\omega_* = \arg \{ |W(j\omega)| = 1 \}. \quad (31)$$

Применение соотношений (28), (29), (30), (31) к семейству эквивалентных разомкнутых систем, порождаемых полиномами В.Л. Харитонова в форме (20 ÷ 23), (24 ÷ 27), (28 ÷ 29), дают одно и тоже значение запаса устойчивости i -той системы $\Delta\varphi_i$, ($i = 1, 2, 3, 4$). В этой связи предпочтение семейству из числа перечисленных делается исключительно из соображений простоты вычислительного характера. Вычисленные в силу (30, 31), значения запасов устойчивости с помощью передаточных функций эквивалентных разомкнутых систем, сопровождающих полиномы В.Л. Харитонова, порождают интервальную их реализацию вида $[\Delta\varphi] = [\underline{\Delta\varphi}, \overline{\Delta\varphi}]$. Это дает возможность сконструировать оценки (9) интервальности запаса устойчивости по фазе интервальной системы (1).

С пользовательской точки зрения, полученные значения оценок (9) позволяют в достаточно сжатой форме оценить эффект введения в структуру системы регулятора, реализующего закон управления (3) и определить направление модификации [3] его матричных компонентов, для уменьшения значений этих оценок с целью достижения требуемой ее величины.

Пример

В качестве примера рассматривается двухканальная фотоэлектрическая следящая система (ФЭСС), сепаратные каналы которой в медианном исполнении, представляют собой полиномиальные динамические модели второго порядка с распределением мод Батерворта и соответственно характеристическими частотами $\omega_0 = 3c^{-1}$ для канала слежения по азимуту и $\omega_0 = 8c^{-1}$ для канала слежения по углу места. В измерительном устройстве канала слежения ФЭСС могут возникать перекрестные связи, приводящие к интервальной матрице связей в прямой цепи

$$[P_c] = \begin{bmatrix} 1 & q_1 \\ q_1 & 1 \end{bmatrix}, \text{ где } [q_1] = [-0.3; 0.3].$$

Исполнительные приводы системы подвергаются температурному воздействию со стороны окружающей среды, которое проявляется в форме интервального значения коэффициента вязкого трения приводов, приводящего в модельном представлении к мультипликативному члену вида $(1 + [q_2])$ где $[q_2] = [-0.3; 0.3]$. С учетом сказанного интервальная матрица состояния $[F]$ системы примет вид

$$[F] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\nu_{12} & -\nu_{11}(1 + [q_2]) & -\nu_{12}[q_1] & 0 \\ 0 & 0 & 0 & 1 \\ \nu_{22}[q_1] & 0 & -\nu_{22} & -\nu_{21}(1 + [q_2]) \end{bmatrix}.$$

Интервальный характеристический полином матрицы $[F]$ строится путем вычисления характеристических полиномов угловых реализаций матрицы $[F] = F([q])$ в пространстве параметров (q_1, q_2) , мощность множества которых равна четырем. Применение к полученным угловым реализациям характеристических полиномов процедуры интервализации, приводит к ИХП принимающему вид

$$[D(\lambda)] = [1; 1]\lambda^4 + [10.9; 20.2]\lambda^3 + [96.5; 154.1]\lambda^2 + [261.3; 485.3]\lambda + [627.8; 627.8] \quad (32)$$

В соответствии с процедурой В.Л.Харитонова на ИХП (32) строятся четыре полинома В.Л.Харитонова. (16)-(19), которые получают представление

$$D_1(\lambda) = \lambda^4 + 10.9\lambda^3 + 154.1\lambda^2 + 485.3\lambda + 627.8$$

$$D_2(\lambda) = \lambda^4 + 20.9\lambda^3 + 154.1\lambda^2 + 261.3\lambda + 627.8$$

$$D_3(\lambda) = \lambda^4 + 20.9\lambda^3 + 96.5\lambda^2 + 261.3\lambda + 627.8$$

$$D_4(\lambda) = \lambda^4 + 10.9\lambda^3 + 96.5\lambda^2 + 485.3\lambda + 627.8$$

Семейство эквивалентных разомкнутых систем сопровождающих полученные характеристические полиномы характеризуются передаточными функциями (28) – (29), которые принимают вид

$$W_1(s) = \frac{10.9s^3 + 154.1s^2 + 485.3s + 627.8}{s^4}$$

$$W_2(s) = \frac{20.9s^3 + 154.1s^2 + 261.3s + 627.8}{s^4}$$

$$W_3(s) = \frac{20.9s^3 + 96.5s^2 + 261.3s + 627.8}{s^4}$$

$$W_4(s) = \frac{10.9s^3 + 96.5s^2 + 485.3s + 627.8}{s^4}$$

По приведенным передаточным функциям могут быть определены запасы устойчивости по фазе $\Delta\varphi_i (i = \overline{1,4})$

$$\Delta\varphi_1 = 52^0.7, \Delta\varphi_2 = 19^0.24, \Delta\varphi_3 = 12^0.73, \Delta\varphi_4 = 51^0.38.$$

Применение к полученным значениям процедуры интервализации дает интервальные представления

$$[\Delta\varphi] = \Delta\varphi_0 + [\Delta\Delta\varphi] = \Delta\varphi_0 + [\Delta\Delta\varphi, \overline{\Delta\Delta\varphi}] = [12^0.73; 52^0.7] = 32^0.715 + [-19^0.985; 19^0.985]$$

, которые позволяют сконструировать оценки интервальности запаса устойчивости ФЭСС в абсолютной и относительной постановках, принимающие в силу (9) значения

$$\Delta_{I\Delta\varphi} = \|\Delta\Delta\varphi\| = 19^0.985 \quad \delta_{I\Delta\varphi} = \frac{\|\Delta\Delta\varphi\|}{\|\Delta\varphi_0\|} = \frac{19^0.985}{32^0.715} = 0.61.$$

Литература

1. Бесекерский В.А., Попов Е.П., Теория систем автоматического регулирования. - М.: Наука, 1972
2. Харитонов В.Л. Об асимптотической устойчивости положения равновесия семейства дифференциальных уравнений // Дифференциальные уравнения 1978 №11
3. Синтез дискретных регуляторов при помощи ЭВМ / В.В. Григорьев, В.Н. Дроздов, В.В. Лаврентьев, А.В. Ушаков. - Л: Машиностроение, Ленингр. отд-ние, 1983.

ИДЕНТИФИКАЦИЯ ЛИНЕЙНЫХ ОПЕРАЦИЙ

В.В. Хабалов, А.И. Салфетников

Существующие в настоящее время методы дифференцирования сигналов и, в частности, метод конечных элементов обладают определенными недостатками, вытекающими из самого принципа нахождения производной. Так, при наличии помехи в виде белого шума дифференцирование сигнала становится невозможным. Предлагается принципиально другой подход, заключающийся в том, что сначала выбирается модель сигнала, аналитическим путем находится ее производная и далее оцениваются параметры полученной модели. Такой подход исключает существенное влияние помех на искомое значение производной.

Введение

На вход устройства оценивания (рис.В.1) поступают значения процесса, представляющие собой аддитивную смесь полезного сигнала и помехи в виде стационарной случайной последовательности с ограниченным вторым моментом. Требуется произвести операцию дифференцирования процесса, поступающего на вход устройства оценивания. Искомые оценки формируются в режиме поступления данных.

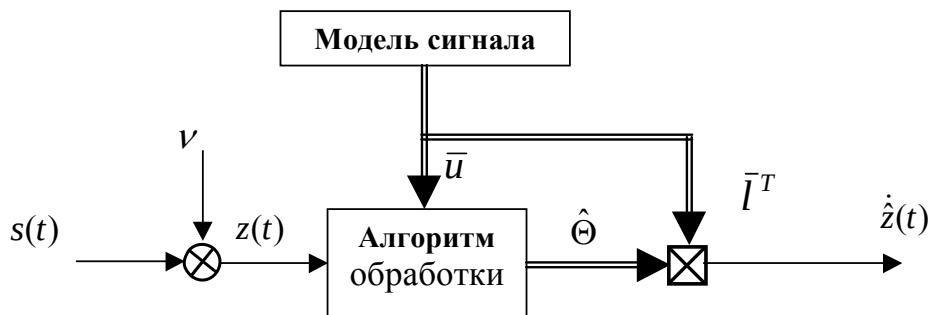


Рис. 1. Схема устройства оценивания

Задача линейного преобразования процессов, представляющих собой аддитивную смесь полезного сигнала и помехи, на первом этапе решается как задача идентификации сигналов по текущим наблюдениям методом наименьших квадратов [1]. При этом используются линейные уравнения наблюдения вида

$$z_j = x_j + v_j = \bar{u}_j^T \Theta + v_j, \quad (B.1)$$

$$x_j = \bar{u}_j^T \Theta,$$

где $z_1, z_2, \dots, z_i$ – последовательность скалярных наблюдений идентифицируемого процесса; Θ – неизвестный, подлежащий оцениванию вектор параметров модели идентифицируемого процесса; $\bar{u}_j^T$ – известные вектора-столбцы, элементами которых являются функции выбранной базисной системы модели идентифицируемого процесса; v_j – последовательность независимых случайных величин, причём

$$Ev_j = 0, Ev_j^2 < \infty, j = 1(1)i. \quad (B.2)$$

Основные положения

Первым этапом решения данной задачи является выбор функций базисной системы модели сигнала, т.е. выбор компонент вектора $\bar{u}_j^T$:

$$\bar{u}_j^T = [f_1(t) \ f_2(t) \dots f_m(t)], \quad (1.1)$$

где $f_i(t)$, $i = 1, 2, \dots, m$ – функции выбранной базисной системы.

Функции $f_i(t)$ выбираются таким образом [2], чтобы полученная модель сигнала

$$x_j = \bar{u}_j^T \Theta = \theta_1 \cdot f_1(t) + \theta_2 \cdot f_2(t) + \dots + \theta_m f_m(t) \quad (1.2)$$

позволяла выполнять простейшие линейные операции над вектором $\bar{u}_j^T$, а также достаточно эффективно решать задачу идентификации наблюдаемого процесса. Наиболее часто для этих целей подходит полиномиальный ряд по степеням t :

$$f_1(t) = 1, f_2(t) = t, f_3(t) = t^2, \dots, f_m(t) = t^{m-1}. \quad (1.3)$$

Далее, в зависимости от условий решаемой задачи, выбирается метод определения вектора параметров θ модели сигнала (1.2). В частности, при отсутствии информации о характеристиках и модели сигнала выбирается классический метод наименьших квадратов [3], метод взвешенных наименьших квадратов или метод наименьших квадратов с учетом старения информации.

Для решения задачи линейного преобразования идентифицированного процесса с целью получения некоторой величины $\hat{y}(t) = \phi[\hat{z}(t)]$, где ϕ определяет требуемое линейное преобразование полученного процесса, введем соотношение:

$$\hat{y}(t) = \phi[\hat{z}(t)] = \bar{l}^T \hat{\Theta}. \quad (1.4)$$

где $\hat{\Theta}$ – вектор текущей оценки параметров модели сигнала, $\bar{l}^T$ – вектор будет определять характер линейного преобразования вектора $\bar{u}_j^T$.

Подробно рассмотрим процедуру получения искомой оценки. Пусть модель оценки идентифицируемого сигнала определяется выражением

$$\hat{z}(t) = \bar{u}^T(t) \cdot \hat{\Theta} = [f_1(t) \ f_2(t) \ \dots \ f_m(t)] \cdot \hat{\Theta}. \quad (1.5)$$

где $\hat{\Theta} = [\hat{\theta}_1 \ \hat{\theta}_2 \ \dots \ \hat{\theta}_m]^T$ – вектор текущей оценки параметров модели сигнала, определяемый методом наименьших квадратов, $\bar{u}^T(t) = [f_1(t) \ f_2(t) \ \dots \ f_m(t)]$ – вектор функций выбранной базисной системы модели сигнала.

Для данного примера в качестве $f_i(t)$, $i = 1, 2, \dots, m$ выберем элементы полиномиального ряда по степеням t :

$$f_1(t) = 1, f_2(t) = t, f_3(t) = t^2, \dots, f_m(t) = t^{m-1}, \quad (1.6)$$

$$\bar{u}^T(t) = [1 \ t \ t^2 \ \dots \ t^{m-1}]. \quad (1.7)$$

Тогда модель оценки идентифицируемого процесса получим в виде

$$\begin{aligned} \hat{z}(t) &= \bar{u}^T(t) \cdot \hat{\Theta} = \theta_1 \cdot f_1(t) + \theta_2 \cdot f_2(t) + \theta_3 \cdot f_3(t) + \dots + \theta_m \cdot f_m(t) = \\ &= \hat{\theta}_1 + \hat{\theta}_2 \cdot t + \hat{\theta}_3 \cdot t^2 + \dots + \hat{\theta}_m \cdot t^{m-1}. \end{aligned} \quad (1.8)$$

Для получения оценки производной полученного сигнала продифференцируем левую и правую части выражения (1.8). Получим:

$$\begin{aligned} \dot{\hat{z}}(t) &= \theta_1 \cdot \dot{f}_1(t) + \theta_2 \cdot \dot{f}_2(t) + \theta_3 \cdot \dot{f}_3(t) + \dots + \theta_m \cdot \dot{f}_m(t) = \\ &= \hat{\theta}_2 + 2\hat{\theta}_3 \cdot t + \dots + (m-1)\hat{\theta}_m \cdot t^{m-2}. \end{aligned} \quad (1.9)$$

Далее возьмем производную по времени от каждого компонента вектора $\bar{u}^T(t)$:

$$\dot{\bar{u}}(t) = [\dot{f}_1(t) \ \dot{f}_2(t) \ \dot{f}_3(t) \ \dots \ \dot{f}_m(t)] = [0 \ 1 \ 2t \ \dots \ (m-1)t^{m-2}]. \quad (1.10)$$

Как видно из выражений (1.9) и (1.10), для получения оценки производной идентифицированного сигнала достаточно проделать такую же операцию (взятие производной) над каждой из компонент вектора базисных функций $\bar{u}^T(t)$ и умножить его на вектор оценки параметров $\hat{\Theta}$ модели сигнала (1.5). Тогда вектор линейного преобразования $\bar{l}^T$ для данной задачи получим в виде

$$\bar{l}^T(t) = \dot{\bar{u}}(t) = [\dot{f}_1(t) \ \dot{f}_2(t) \ \dot{f}_3(t) \ \dots \ \dot{f}_m(t)] = [0 \ 1 \ 2t \ \dots \ (m-1)t^{m-2}], \quad (1.11)$$

и оценка производной полученного сигнала будет определяться из выражения

$$\dot{\hat{z}}(t) = \bar{l}^T(t) \cdot \hat{\Theta}, \quad (1.12)$$

где $\bar{l}^T(t) = \bar{u}(t)$ – вектор линейного идентифицируемого сигнала $\hat{z}(t)$, $\hat{\Theta}$ – вектор текущей оценки параметров модели сигнала (1.5). Из рассмотренного примера следует, что, применяя аналогичные рассуждения и соответствующим образом преобразовывая вектор базисных функций $\bar{u}^T(t)$, можно прийти к решению целого класса задач, определяемых линейным преобразованием вектора $\bar{u}^T(t)$.

В качестве примера выберем модель идентифицируемого сигнала в виде полиномиального ряда по степеням t (1.3). Тогда, применяя преобразования, приведенные выше, для оценки сигнала (B.1), получим вектор линейного преобразования $\bar{l}^T$ в виде:

- 1) для задачи экстраполяции $\bar{l}^T = [1 \quad (t_i + k \cdot \Delta t) \quad (t_i + k \cdot \Delta t)^2 \quad \dots \quad (t_i + k \cdot \Delta t)^{m-1}]$;
- 2) для задачи интерполяции $\bar{l}^T = [1 \quad (t_i - k \cdot \Delta t) \quad (t_i - k \cdot \Delta t)^2 \quad \dots \quad (t_i - k \cdot \Delta t)^{m-1}]$;
- 3) для задачи дифференцирования $\bar{l}^T = [0 \quad 1 \quad 2t_i \quad \dots \quad (m-1)t_i^{m-2}]$;
- 4) для задачи интегрирования $\bar{l}^T = [t \quad \frac{1}{2}t^2 \quad \frac{1}{3}t^3 \quad \dots \quad \frac{1}{m}t^m]$;
- 5) для задачи фильтрации $\bar{l}^T = [1 \quad 0 \quad 0 \quad \dots \quad 0]$.

Для получения экспериментальных данных применим рассмотренные линейные преобразования сигналов к сигналу вида $s(t) = at \cdot e^{-bt} \cdot \sin(ct)$ с параметрами $a = 5$, $b = 0.3$, $c = 2$ в присутствии аддитивной помехи в виде белого шума с нулевым математическим ожиданием и дисперсией $D = 0.5$. В качестве модели сигнала примем элементы полиномиального ряда $u^T = [1 \quad t]$.

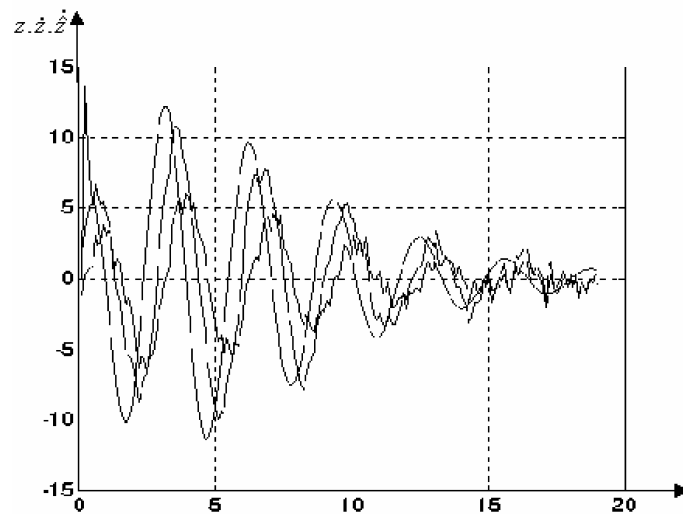


Рис. 1.1. Дифференцирование сигнала методом наименьших квадратов с учетом старения. Длина памяти равна 0.8 с

Из полученного экспериментального материала следует, что линейное преобразование сигналов с использованием метода наименьших квадратов может применяться при решении широкого круга задач теории автоматического управления, где используются алгоритмы обработки сигнала в присутствии помех. Как видно из приведенных выше графиков, алгоритмы эффективно работают в присутствии помех, что особенно важно при решении задачи дифференцирования сигнала.

Литература

1. Алберт А. Регрессия, псевдоинверсия и рекуррентное оценивание. М.: Наука, 1977.
2. Чураков. Е.П. Оптимальные и адаптивные системы. М.: Энергоиздат, 1987.
3. Льюнг. Л. Идентификация систем. М.: Наука, 1991.

ВЕКТОРНАЯ СИСТЕМА УПРАВЛЕНИЯ УСТРОЙСТВОМ РЕКУПЕРАЦИИ НА МИКРОПРОЦЕССОРЕ TMS320F243

А.П. Баев, М.Р. Гончаренко, А.С. Исаков, А.Н. Коровьяков, О.С. Осипцева

В статье представлены результаты разработки и испытаний микропроцессорной системы управления устройством рекуперации энергии в промышленную сеть, входящей в состав системы управления приводами эскалатора метрополитена.

Введение. Постановка задачи

Необходимость разработки описываемой системы возникла при решении поставленной Комитетом по экономике и промышленной политике Администрации Санкт-Петербурга задачи реконструкции систем управления асинхронными двигателями эскалаторов.

Принципы построения и функциональные свойства системы

Современные системы управления асинхронными электроприводами создаются на основе совокупности взаимосвязанных физических принципов, способов организации управления и взаимодействия функциональных элементов системы.

Для повышения экономической эффективности в структуру привода включено устройство рекуперации (УР) энергии. При этом достигается энергосбережение и уменьшение уровня возмущений на сеть со стороны работающего электропривода. Реализация управления УР строится в соответствии с современной концепцией векторного управления электроприводами и устройствами рекуперации. С целью расширения функциональных возможностей система управления реализована на основе микропроцессора и согласующего электронного обрамления. Основным требованием к микропроцессору является его повышенное быстродействие. Поэтому в системе управления УР применен сигнальный процессор TMS320F243, обладающий полным требуемым набором встроенных периферийных устройств АЦП, ШИМ и т.д.

Устройство рекуперации асинхронного электропривода эскалатора должно функционировать в одном из следующих режимов: в режиме синхронизации с сетью, в режиме компенсации реактивной мощности и в автономном режиме. Цели управления и динамические характеристики УР в различных режимах принципиально различны, что при реализации системы управления отражается на структуре программного обеспечения.

Программное обеспечение управляющего микропроцессора построено по модульному принципу и состоит из основной программы и ряда программных модулей, обеспечивающих требуемые функции.

Принцип определения текущего значения фазы сети заключается в том, что трехфазный вектор напряжений сети подается на вход преобразователя координат из неподвижной системы во вращающуюся. Если результирующий вектор трехфазного напряжения совпадет с осью Y вращающейся системы координат, то проекция результирующего вектора напряжения на ось X будет равна нулю, а проекция на ось Y пропорциональна амплитуде фазного напряжения.

Во вращающейся системе координат формируются сигналы задания тока для фазных регуляторов. При этом составляющая тока по оси X задается нулевой, в результате вектор тока в фазах совпадает с вектором фазного напряжения, а реактивная составляющая тока отсутствует.

Контроллер на базе сигнального процессора TMS320F243 выполняет быстрые аппаратные алгоритмы управления инвертором, ввод информации с датчиков, контроль работы, используя развитую систему периферийных устройств и датчиков контролируемых физических величин. Процессор осуществляет алгоритмы

преобразования координат и предварительной фильтрации сигналов, а также согласование работы всех устройств системы.

Связь с управлением верхнего уровня осуществляется через встроенный контроллер CAN интерфейса по последовательному каналу.

Для ввода восьми аналоговых сигналов применены дифференциальные усилители, на которые подается смещение -2.5 В для приведения сигнала к диапазону встроенных АЦП. Гальваническая развязка осуществляется с помощью оптронных пар. Для предотвращения пробоя транзисторов используется вход аппаратного прерывания с высшим приоритетом, по которому аппаратно блокируется выход ШИМ. Дополнительно аппаратно контролируется работа программы на каждом интервале ШИМ с помощью внешнего одновибратора. В случае сбоя программы вырабатывается сигнал немаскируемого прерывания.

При контроле работы устройства применяется программа контроля допустимой разности в вычислении управляющих воздействий на последовательных тактах квантования.

Управление инвертором осуществляется с помощью встроенного в микроконтроллер ШИМ. Он использует таймер общего назначения и имеет двойную буферизацию, что позволяет асинхронно к периоду ШИМ выводить управляющие воздействия.

Результаты лабораторных испытаний

В результате лабораторных испытаний микропроцессорной системы устройством рекуперации установлено, что система в полном объеме выполняет требуемые функции во всех режимах эксплуатации устройства с достижением установленных значений показателей качества. Быстродействие микропроцессора TMS320F243 позволило дополнительно к основным функциям обеспечить в реальном времени диагностику ответственных элементов УР и их защиту в предаварийных ситуациях.

Заключение

Введение в структуру систем управления асинхронными приводами эскалаторов устройств рекуперации с микропроцессорной системой управления повышает экономическую эффективность функционирования приводов и обеспечивает гибкость управления, многофункциональность и надежность.

Устройство рекуперации может использоваться не только совместно с электроприводом, но и как самостоятельное устройство преобразующее постоянное напряжение в трехфазную систему напряжений, а также осуществляющее обратное преобразование трехфазного напряжения в постоянное.

Особо заметим, что при использовании устройства рекуперации по разным назначениям и в различных режимах требуется модифицировать лишь программное обеспечение, выбирая соответствующие программные модули. При этом аппаратные средства изменяются незначительно, что достигается гибкостью устройства рекуперации.

Литература

1. В.Ф. Козаченко. Основные тенденции развития встроенных систем управления двигателями и требования к микроконтроллерам.//CHIP NEWS. –1999.-N1. – с. 2-9
2. В. Ф. Козаченко, С.Я. Грибачев. Новые микроконтроллеры фирмы Texas Instruments TMS32x24x для высокопроизводительных встроенных систем управления электроприводами.//CHIP NEWS. – 1998.-N11-12.-с.2-6

КОНВЕРГЕНЦИЯ ЭЛЕКТРОННЫХ И ПЕЧАТНЫХ СРЕДСТВ ИНФОРМАЦИИ

В.Н. Дроздов, А.Н. Коваленко

Внедрение информационных технологий практически во все направления деятельности человека имеет своей основой следующую цепочку рассуждений:

- одной из основных сущностей человека является принятие решений;
- для принятия решений необходима информация;
- универсальным средством для работы с информацией является компьютер.

Указанная цепочка показывает всеобщность и важность компьютерных систем. К ним необходимо добавить различные системы с компьютеризированными элементами: встроенными микросхемами, электронными матрицами, управлением через набор команд на микротабло и др. Все это также входит является частью современных информационных технологий, важной областью приложения которых являются средства массовой информации (СМИ).

СМИ уже давно принято делить по виду конечной продукции на печатные и электронные. Под последними до 90-х гг. понимались телевидение, радио, справочная телефония. Сейчас в спектр электронных средств вошли сайты и электронные издания в системе Интернет, продукция на компакт-дисках и других магнитных носителях.

В печатных изданиях конечная продукция имеет бумажный или другой твердый носитель запечатываемого материала. Электронным СМИ характеризуются электронным видом продукции в момент ее поставки потребителю.

Если же опираться не на вид информации при ее поставке потребителю, а на способ производства, то деление между бумажными и электронными СМИ в значительной степени размывается. Производство печатной продукции также в подавляющей степени опирается на компьютерные системы и имеет весьма схожие приемы работы с производством электронных средств.

Особенно ярко это видно на примере работы издательств и редакций. Производство печатной продукции от момента переложения замысла автора в компьютерный вид (текст, рисунок и пр.), через компьютерные сети издательско-полиграфических комплексов и до получения сброшюрованного тиража из цифровой печатной машины может не выходить из компьютерной системы. Подчеркнем, что здесь работа с электронной информацией является сутью основного производственного процесса.

О других составляющих отрасли печати. В книгораспространении главным производственным процессом является движение товара. Однако это движение сопровождается мощным информационным потоком, которые охватывает все последовательные стадии от заключения договоров на поставку товара до электронных продаж и уведомления единичных потребителей. Наконец, собственно полиграфия – тиражирование подготовленного продукта – осуществляется на печатных машинах с электронным управлением (в том числе. с расчетом загрузки оборудования) и компьютеризированной системой контроля качества.

Таким образом, все три составляющие отрасли печати (система подготовки издания, полиграфическая стадия, распространение печатной продукции) имеют разное, но весьма насыщенное компьютерное обеспечение.

В целом это и позволяет говорить о схожести печатных и электронных средств информации. Однако сходство оказывается еще глубже.

Фундаментальной особенностью современных информационных технологий является работа с цифровым видом информации. Именно оцифровывание ведет (после изначально различных шагов для текста, изображения, аналоговой информации) к единообразию в передаче, преобразовании, хранении, защите, и, в целом, к упрощению

взаимодействия между собой различных сфер деятельности человека. Считается также, что оцифровывание сокращает путь от разработки научных идей до практического внедрения.

Глобальными объединительными направлениями самых различных информационных технологий (не только в СМИ) являются:

- а) Хранение информации в виде баз данных.
- б) Сетевая работа с информацией (в сетях разных уровней).
- в) Приемы пользовательского интерфейса и организация специализированных рабочих мест.

Общие или схожие проблемы различных информационных технологий на уровне решения конкретных вопросов отражены в следующем списке:

- Создание и использование баз данных;
- Организация сетевой работы;
- Каналы передачи;
- Поиск информации (включая поиск в глобальных сетях);
- Защита информации;
- Форматы и переформатирование;
- Оцифровывание нецифровых данных;
- Передача больших массивов;
- Сжатие информации;
- Перезапись на различные носители;
- Проектирование интерфейса;
- Подготовка текста, таблиц, диаграмм, иллюстраций, звука, анимации;
- Верстка и монтаж;
- Аудио и видеоэффекты;
- Визуальная работа с электронным видом информации;
- Распределенная работа с информацией;
- Дистанционное управление и контроль оборудованием;
- Реклама и презентация работ;
- Юридические аспекты работы с информацией.

Указанный перечень показывает, что цифровое представление информации объективно ведет к конвергенции информационных технологий. Составляющие этого перечня являются объединительными моментами при создании и эксплуатации самых различных компьютерных систем.

Оцифровывание (дигитализация, цифрование) является разновидностью дискретизации и знаменует собой характерное для всего XX века движение от непрерывного описания к дискретному. Цифровой, т.е. дискретный вид изображения и звука в компьютере служит дополнительным аргументом в споре философов о том, является ли непрерывное и бесконечное объективной реальностью, или существует только как способ мышления человека.

Будучи средством профессиональной работы, информационные системы одновременно являются и продуктом профессиональной деятельности их создателей. Отсюда следуют два подхода: 1) пользовательский; 2) подход проектировщика и создателя.

Отметим, что эти подходы не существуют в чистом виде. Специалист по созданию и поддержке информационных систем (часто исторически называемый программистом) всегда что-то знает о предметной области, а пользователь для успешной работы с системой должен иметь определенные знания об ее внутреннем устройстве.

Это порождает базу для подготовки специалистов: они могут быть либо техническими (программистскими) кадрами, ориентированными на быстрое проникновение в конкретную область человеческой деятельности (концепция университетского образования), либо специалистами-предметниками, знающими достаточно много о сути информационных систем и технологий как таковых.

Отраслевое образование ориентируется на второй путь, как более простой при реализации информационных систем на основе стандартизированных элементов и решений. При этом моменты оригинального проектирования информационных систем существуют, но в целом играют подчиненную роль. Такой путь позволяет сочетать

отраслевую подготовку с хорошим компьютерным образованием и поставлять отрасли специалистов по созданию и внедрению разнообразных информационных систем. При этом во многом убирается процедура необходимого и нередко проблемного контакта специалистов-компьютерщиков и специалистов в данной области деятельности.

Уже было показано, что среди различных направлений применения информационных технологий одну из первых ролей играют СМИ. Приведем список элементов современных информационных систем, которые либо появились в СМИ, либо развивались под их сильным влиянием. Это:

Гиперсвязь (гипертекст и др.);	Компьютерная анимация;
Компьютерная верстка;	Мультимедиа;
Компьютерное форматирование;	Интерактивные системы;
Компьютерный дизайн;	Электронные продажи;
Обработка цветowych изображений;	Компьютерные поисковые системы;
Цифровая фотография;	Цифровая печать;
Цифровое видео;	Передача больших массивов информации.
Компьютерная работа со звуком;	

Значительная часть информационных технологий, перечисленных в приведенном списке, развивалась и продолжает свое развитие именно в отрасли печати [1]. Роль печатного дела в совершенствовании информатизации может быть подчеркнута таким примером, как совместный проект ведущих фирм Deutsche Telecom (телекоммуникации) и Heidelberg (производство печатной техники). Проект предусматривает создание глобальной платформы передачи данных, т.е. совокупности программных средств, форматов передачи и, частично, специальной аппаратуры для быстрой и уверенной передачи больших и сверхбольших массивов цифровых данных.

Примером перспективного направления развития информационных технологий в области печати является создание серии форматов представления данных (CIP-3, CIP-4), в которых файл с оригинал-макетом, формирующийся на стадии предпечатной подготовки, содержит в себе управляющие команды для цветоделения, работы печатного оборудования, разрезки, брошюровки и даже упаковки. Технология CIP включает также элементы WorkFlow (управление рабочими потоками), контроль сетевых и системных ресурсов [2].

Показателем передовых информационных рубежей отрасли печати является технология автоматической передачи через Интернет газетных полос и печати их на специальном принтере без участия человека. Этот способ применяется для производства малого тиража центральной газеты в удаленной гостинице, железнодорожной станции или даже дома у единичного клиента. Такие распределенные печатные системы уже предлагают фирмы Lexmark, IBM, HP [2].

Связь отрасли печати с другими звеньями СМИ изображена на Рис.1. Эта схема создана на основе информации из книги Кипхана [3].

Отметим важные моменты схемы на рис.1. Стадия сценарной и допечатной подготовки имеет много общих моментов для всех видов СМИ. Это редакторская, корректорская, дизайнерская работа, контакты с авторами, заказы иллюстраций и звукозаписей, фото и видеозаписи и их обработка и пр. Конечно, уже эта работа учитывает дальнейшую форму СМИ, но общего здесь больше, чем отличий. После подготовки компонент СМИ они превращаются в цифровые данные, к ним добавляются управляющие команды, и наступает окончательное разделение электронной и бумажной линий.

Операция макетирования в электронной стадии включает в себя покадровую, поминутную, позранную и постраничную разработку, системы навигации для электронной печати и ряд других элементов. Все это соответствует стадии изготовления оригинал-макета для данного вида печати. После этого начинается само

производство продукции. Естественно, электронная и печатная линии имеют ряд отличий, которые также отражены на Рис.1.

Распространение СМИ имеет сильное сходство в борьбе за потребителя – это реклама, промоушен, исследование мнения потребителя, его сегментация, быстрота реагирования на потребности потребителя и одновременно умение вести его за собой.



Рис. 1. Конвергенция печатных и электронных СМИ

Важной разновидностью электронных СМИ являются электронные издания. Хотя они имеют своими носителями компьютерные сети, компакт-диски (CD) и иные способы электронного хранения и соответственно помещены в электронную линию на рис.1, их подготовка имеет бросающуюся в глаза общность с подготовкой печатной продукции. Это отражается в самой терминологии: электронные газеты, журналы, книги, каталоги и пр. Производство электронной продукции чаще всего сосредоточено в издательствах, занимающихся одновременно и подготовкой бумажных изданий. Во многом общими являются здесь и проблемы авторства.

Вопрос о долях электронной и печатной продукции исследовался и прогнозировался многими авторами. Источник [3] указывает на их соотношение 30%/70% в 1999 г. и 50%/50% в 2010 г. В этот период бумажные средства будут расти в среднем на 3% ежегодно, а электронные – на 9%. Общий рост всех СМИ за этот период составят примерно 50%. К 2020-2025 годам большинство исследователей предсказывает установление равновесия долей электронной и печатной продукции

примерно в пропорции 2 : 1 (т.е. 67% и 33%). Отметим, что эти данные определены в стоимостном выражении.

Состояние дел, изображенное на рис. 1, а также представленные прогнозы интерпретируются как движение отрасли печати в направлении медиаиндустрии. Последняя будет включать в себя не только технологию переработки информации и собственно СМИ, но и каналы связи, специализированное программное и, в меньшей степени, аппаратное обеспечение, корпоративное отстаивание своих интересов, а также общие научные и учебные центры.

Объединение в мультимедийном продукте текста, цветного изображения, звука и анимации дает так называемый синергетический эффект, который с учетом компьютерных возможностей превосходит возможности такого классического мультимедийного средства как телевидение.

Из этих дополнительных возможностей в первую очередь отметим появление в сети Интернет гиперсвязной (гипертекстовой) структуры и интерактивности. Эти новые элементы уже перешли на CD-издания, на товары, распространяемые по сети, на презентационные материалы и др. Гиперсвязь имеет своим прототипом ссылочный и справочный материал обычных изданий. Интерактивность внутренне связана с чтением книги по выработанной не автором, а читателем схеме, что, например, распространено в справочной литературе. Контекстный переход, основанный не только на элементах текста, но и изображения, звука и даже анимации, создает новые возможности по усвоению материала, скорости поиска, эстетике восприятия. Правда, достижения в эстетике и художественном впечатлении, сравнимые с шедеврами верстки и дизайна бумажных изданий, еще впереди.

В других видах СМИ гиперсвязь и интерактивность только зарождаются. Имеются эксперименты интерактивности на ТВ, есть литературные произведения с несколькими окончаниями сюжета, интерактивны по своей сути передачи с прямым участием зрителей.

В целом и в современных СМИ на первом месте остаются авторская (в т.ч. сценарная), редакторская, дизайнерская (операторская) и режиссерская работа. Технические приемы и возможности должны (что происходит далеко не всегда) быть подчинены замыслу и цели информационного продукта. Это, в частности, должно достигаться типичным для печатного издания тесным контактом создателей формы с создателями сути продукта. Отсюда следует такая черта подготовки специалиста по информационным технологиям в СМИ, как единение технического и гуманитарного обучения, что сближает гуманитарную и естественно-техническую деятельность в целом.

Многое из того, что сказано выше, показывает, что большинство новых идей и новых специальностей берет свое начало именно в традиционной печати.

Поэтому важным для оценки тенденций, в частности, для определения направлений подготовки специалистов, является распределение сегментов печатной продукции. По информации на 1999 год это распределение в денежном выражении показано на рис. 2. Приведенные здесь данные взяты из источника [3].

Отметим, что деловые бумаги на рис. 2 отнесены к информационному сегменту «реклама и предложения».

Важным элементом конвергенции в средствах СМИ является параллельный выпуск электронных и печатных изданий. При этом возможны все варианты: полностью параллельный выпуск с учетом специфики формы издания; выпуск электронного аналога для бумажного издания, а также система «книга плюс CD»; выпуск бумажного сопровождения для электронного издания, например, «CD плюс книга». Общим при этом является работа автора, корреспондента, редактора, корректора, фотокорреспондента и др. Достаточно разной является работа верстальщика, дизайнера, технического изготовителя макета издания.

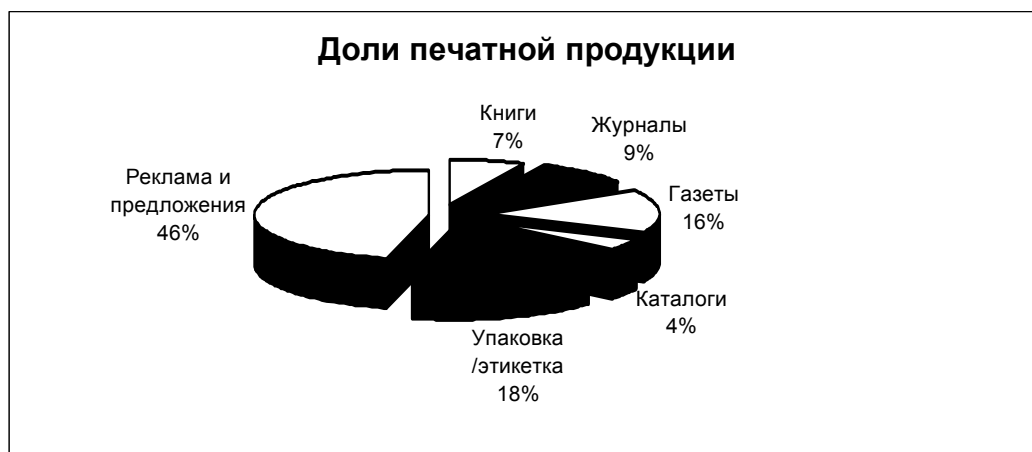


Рис. 2 . Сегменты отрасли печати

Эти рассуждения означают и то, что конвергенция электронных и печатных средств информации имеет свои пределы, и они не сольются полностью. Здесь идет речь не об опасностях для обычной газеты, журнала, книги (они успешно пережили атаки со стороны фотографии, телефона, уличной рекламы, радио, телевидения и др.), а о том, что печатные и электронные средства имеют, по крайней мере, в обозримом будущем, не полностью совпадающие множества «индивидуум плюс его текущие потребности».

На вид электронных, и во вторую очередь, печатных средств сильное влияние оказывает развитие информационных технологий. Расширение пользователей системы Интернет, увеличение объема пейджерной и видеосвязи, мобильные телесистемы, рост качества цифровых музыкальных и мультимедийных записей ставят проблемы приспособления информации, как под технические средства, так и под все более узкие группы потребителей.

Однако возникновение новых приемов и методов собственно работы с информацией в последние годы имеет тенденцию к уменьшению. Можно предположить, что период бурного развития информационных технологий заканчивается. Но это делает еще более насущным осмысление того, что происходило и происходит в операциях переработки и доставки информации до массового потребителя.

Литература

1. Дроздов В.Н., Коваленко А.Н. Известия Вузов, сер. Проблемы полиграфии и издательского дела; №3-4' 2001
2. Газетные технологии. Издание IFRA. Russian edition; октябрь 2001
3. Kipphan H. Handbook of Print Media Technologies and Production Methods. Springer, 2001

СИНТЕЗ ИНФОРМАЦИОННО-ЭНЕРГЕТИЧЕСКИХ СВЯЗЕЙ В СТРУКТУРАХ ИЗОБРЕТЕНИЙ

А.Б. Бушуев

Для получения новых функций технических систем предложен метод синтеза прямых и обратных связей, увеличивающий информационно-энергетическую проводимость систем.

В изобретательской практике для анализа и синтеза структуры технических систем используются функциональные схемы [1]: конструктивные и потоковые. Конструктивная схема отражает элементный состав системы и основные функции, выполняемые элементами. Потокоская схема представляет модель преобразования информационных и энергетических потоков функциональными элементами системы. Эти схемы составляются на основе признаков в формуле изобретения-прототипа и описания его действия.

В теории решения изобретательских задач (ТРИЗ), в частности, в вепольном анализе [2], информационно-энергетическая модель технической системы представляет цепочку веполей или комплексный веполь. Веполь – это простейшая ячейка структуры, состоящая из двух элементов (веществ) и поля, причем поле понимается в широком смысле, как любое действие одного элемента на другой. Поэтому вепольная структура технической системы представляет собой совокупность элементов, связанных причинно-следственными связями.

Задача синтеза нового технического решения ставится следующим образом. Задана структура исходной технической системы в виде одной из рассмотренных выше схем. Система предназначена для выполнения определенной функции. Целью синтеза является получение на базе исходной структуры новой системы, которая должна выполнять новые функции. В этом заключается главное отличие от известного подхода, в частности, в патентном праве, где изобретение и его прототип должны выполнять одну и ту же функцию.

Постановки задачи частично вытекает из закона увеличения степени идеальности технических систем [2]. По этому закону развитие систем идет в направлении увеличения количества и качества функций, выполняемых системой, т.е. одновременно с новой функцией система продолжает выполнять старую функцию. В поставленной задаче старая функция может пропасть, зато появляется множество новых систем, выполняющих новые функции.

Цель достигается изменением структуры исходной технической системы. Определим направления возможного изменения структуры, исходя из закона энергетической проводимости [2] технических систем.

Закон энергетической проводимости определяет необходимое условие принципиальной жизнеспособности технической системы: сквозной проход энергии по всем частям системы. Так как каждая часть системы (элемент) может быть разной физической природы, то поток энергии, проходя через систему, преобразуется по виду. Поскольку одновременно с преобразованием энергии через систему передается информация, а передача информации невозможна без передачи энергии, то содержание закона необходимо уточнить, т.е. говорить об сквозной информационно-энергетической проводимости системы. Информационно-энергетическая проводимость между двумя элементами системы означает их функциональную связь, т.е. при изменении выхода первого элемента согласованно меняется вход второго элемента.

До синтеза исходная структура заведомо признается обладающей неполной информационно-энергетической проводимостью. Повышение проводимости достигается проведением новых прямых и обратных связей, соединяющих информационно-энергетические выходы и входы элементов и всей системы в целом.

Широко известное замыкание системы с выхода на вход по петле отрицательной обратной связи не всегда дает новую функцию. Поэтому необходимо выявить новые информационно-энергетические входы и выходы, не существующие в исходной структуре.

Воспользуемся для этого признаками, используемыми в патентном праве для характеристики объектов изобретений: устройств, способов, веществ [3]. Например, для характеристики устройства, наряду с другими, используются признаки: форма выполнения элемента, в частности геометрическая форма, параметры и другие характеристики элемента, материал, из которого выполнен элемент. Для характеристики способа используется признаки: порядок выполнения действий во времени, режим выполнения действий и другие. Для характеристики вещества применяются признаки, характеризующие физико-химический состав вещества.

Для каждого элемента исходной системы назначаем новые информационно-энергетические входы и выходы в виде признаков, которыми этот элемент может быть описан в формуле изобретения. Далее, в простейшем случае попарно, перебором, соединяем разные выходы и входы разных элементов. При каждом соединении получаем множество новых систем, выполняющих новые функции, причем до синтеза эти функции обычно неизвестны. Таким образом, по способу выбора связей этот метод относится к морфологическому анализу [1], и для бинарных связей оформляется в виде морфологической таблицы, а для многомерных связей - в виде морфологического ящика.

Если структурная модель исходной системы задана в виде цепочки веполей, то для образования новых информационно-энергетических входов и выходов используются вещественно-полевые ресурсы.

После синтеза производится анализ полученных решений, чтобы определить, какие новые функции появились у системы, и возможности их технической реализации.

Недостатки метода заключаются в необходимости анализа большого количества новых решений и отсутствие направленности поиска. Достоинствами метода являются множество новых решений, их оригинальность, неочевидность, что связано с применением устройств, способов, веществ по новому назначению.

Пример

В качестве исходной системы задан асинхронный двухфазный электрический двигатель с короткозамкнутым ротором. В функциональной схеме системы управления двигатель имеет два входа: информационный – обмотка управления и энергетический – обмотка возбуждения; информационно-энергетический выход – вал ротора. Двигатель преобразует электрическое напряжение на обмотке управления в механическую скорость вращения вала ротора (информационный выход) или момент на валу (энергетический выход). Замыкание двигателя отрицательной обратной связью от информационного выхода на информационный вход известно, и дает лишь повышение качества выполнения старой функции.

Поставим задачу – на структуре двигателя синтезировать новую систему, выполняющую новую, пока неизвестную функцию.

Признаками двигателя являются статор, ротор, обмотки, их расположение и связь, форма и материал статора и ротора, число витков, сечение проводов и т.п. Выберем в качестве нового информационного входа форму ротора и свяжем этот вход с уже существующим информационным выходом. Таким образом, в новой системе в зависимости от скорости вращения ротора должна меняться его форма.

В простейшем случае ротор представляет собой алюминиевый стаканчик. При вращении на ротор действует центробежная сила, и тем больше, чем больше скорость

вращения ротора. Чтобы форма ротора могла меняться от действия центробежной силы, он должен быть из материала, обладающего большой подвижностью. Следовательно, изменение формы влечет и изменение материала ротора. Следующим материалом по степени внутренней подвижности после твердого вещества является жидкость, поэтому ротор изготавливаем из жидкости. Чтобы при вращении жидкость не разлеталась, наливаем ее в сосуд, который располагаем вертикально. Сосудом может быть уже существующий стаканчик ротора. Таким образом, двигатель с вертикальной осью приводит во вращение свой ротор, заполненный жидкостью.

При вращении свободная поверхность жидкости образует идеальную параболу с изменяемой кривизной, т.е. прогиб верхнего торца ротора зависит от скорости вращения. Для случая прозрачности жидкости и дна ротора система является линзой, в противном случае - зеркалом. Чтобы при вращении менялась не только форма торца, но и стенки ротора, в стенках вырезаем отверстия, закрытые эластичным материалом. Тогда при изменении скорости вращения будет меняться также и уровень жидкости. Получаем оптическую систему с быстро перестраиваемым фокусным расстоянием. Техническое решение оказалось патентнопригодным и защищено как способ и устройство [4].

Литература

1. Половинкин А.И. Основы инженерного творчества. М.: Машиностроение, 1988.
2. Альтшуллер Г.С. Найти идею. Новосибирск: Наука, Сибирское отделение, 1991.
3. Правила составления, подачи и рассмотрения заявки на выдачу патента на изобретение. М.:ВИНИТИ,1994.
4. Бушуев А.Б. Способ управления оптической системой и управляемая оптическая система. Патент России № 2012911, кл. G02B 26/06, 15.09.94.

ПРОБЛЕМЫ ОБРАЗОВАНИЯ
СПЕЦИАЛИСТА ПО ЗАЩИТЕ ИНФОРМАЦИИ

Л.Г. Осовецкий, В.С. Звонов

Педагогический процесс представляет собой единство трех составных частей: *образование* (передачу упорядоченного знания о том или ином предмете деятельности), *обучение* (процесс, направленный на овладение обучающимся формой деятельности с данным предметом) и *воспитание* (формирующее причастность ученика к системе идеалов, ценностей и интересов, которые порождают цели, критерии и ограничения его формы деятельности). Однако в настоящее время "образование" практически стало синонимом педагогики в целом. В силу специфики деятельности специалиста по защите информации при его подготовке приходится думать о триединстве всех составляющих педагогического процесса. Ведь в руки будущих специалистов вручается могучее оружие, ибо кто владеет информацией, тот владеет миром. И на первый план встает задача формирования личности, способной думать, думать творчески и отвечать за свои действия.

Весь процесс обучения должен высвобождать способность человека творить. Изначально каждый способен к творчеству, но мера творческих возможностей для разных людей различна, а во-вторых, не хотелось бы, чтобы неправильная система воспитания заглушила в студентах творческие задатки. Трудность заключается в том, что специалисту по защите информации предстоит защищать то, что до сих пор не имеет четкого определения.

Имеется много определений понятия информации, от наиболее общего философского – "Информация есть отражение реального мира" [1] до узкого, практического – "Информация есть все сведения, являющееся объектом хранения, передачи и преобразования" [2].

Приведем для сопоставления также некоторые другие определения и характеристики.

1. Информация (Information)- содержание сообщения или сигнала; сведения, рассматриваемые в процессе их передачи или восприятия, позволяющие расширить знания об интересующем объекте [3].
2. Информация - является одной из фундаментальных сущностей окружающего нас мира (акад. Поспелов).
3. Информация - первоначально - сведения, передаваемые одними людьми другим людям устным, письменным или каким -нибудь другим способом (БСЭ).
4. Информация - отраженное разнообразие, то есть нарушение однообразия [1].
5. Информация - является одним из основных универсальных свойств материи [1].
6. Информация - сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления [10].

Под информацией необходимо понимать не сами предметы и процессы, а их отражение или отображение в виде чисел, формул, описаний, чертежей, символов, образов. Сама по себе информация может быть отнесена к области абстрактных категорий, подобных, например, математическим формулам, однако работа с ней всегда связана с использованием каких-нибудь материалов и затратами энергии. Поэтому понятна огромная роль математического образования при формировании специалиста по защите информации, ибо предметом, которым занимается математика,

как раз и является "...изучение понятий, введенных путем абстрагирования от явлений реального мира" [11].

Создание абстрактных понятий является необходимым шагом любого познания. Каждому новому предшествует идея, мысль об этом новом. И только когда идея созреет, она становится базой для практических поисков и применений. Именно этого кафедра БИТ и ждет от общеобразовательных кафедр, и прежде всего от кафедр высшей математики и физики, в деле формирования специалиста по защите информации.

Математика еще не сказала своего решающего слова в формировании языка для описания структуры информационных потоков. Конечно, теория нечетких множеств, теория графов, синергетические подходы, теория катастроф дают определенный базис, но книги по информатике до сих пор носят в основном описательный характер. Возникает парадокс – необходимо создать новый язык для передачи информации по теории информации. Научное изложение на этом новом языке должно быть кратким и исчерпывающим, сохранять полную определенность и не допускать различных толкований. Химия создала свой собственный язык химических формул, способный лаконично и максимально точно передавать особенности химических процессов и даже предвидеть возможные соединения. Галилео Галилей писал: "Философия написана в грандиозной книге – природе, которая открыта для каждого, кто научился понимать язык и знаки, которыми она написана. Написана же она на математическом языке, а знаки ее – математические формулы". Поэтому мы ждем, что математическое образование сформирует творческих людей, способных не только создавать информацию, но и защищать ее.

Специалист по защите информации в своей работе активно опирается на научный стиль мышления, который базируется на следующих принципах.

1. Доказательность, основанная на признании фактов, добытых в опыте, как высшей ценности. "Все, что мы знаем, исходит из опыта и завершается им", - говорил Эйнштейн.
2. Антиавторитарность. Авторитеты не могут служить способом доказательства. Фарадей на этот счет говорил, что "у ученого не должно быть излюбленной гипотезы, он обязан не иметь авторитетов".
3. Здоровый скептицизм по отношению к так называемому очевидному или общепринятому.
4. Критическая самооценка, осторожная осмотрительность в суждениях, антипатия к безапелляционности, к максимализму в суждениях.
5. Антидогматизм, проявляющийся в постоянной готовности к изменению понятий, теорий, методов, исходящий из понимания гибкости понятий, их изменчивости.
6. Понимание преемственности научных знаний, уважительное отношение к ранее добытому в науке. "Если я видел дальше других, то потому, что стоял на плечах гигантов", - говорил Ньютон.
7. Детерминизм как принцип подхода к исследованию, проявляющийся в уверенности в том, что все в мире причинно обусловлено, даже если причины эти пока неизвестны.
8. Понимание неизбежности парадоксального в науке.
9. Принцип системности, проявляющийся в стремлении не просто собрать набор фактов, а выстроить их во взаимосвязанную систему, образующую теорию, к созданию которой всегда тяготеет ученый.
10. Точность и определенность суждений, умение формулировать понятия лаконичным языком, не допускающим двусмысленности и кривотолков.
11. Принцип простоты, проявляющийся в стремлении использовать минимальное число допущений для получения максимального числа следствий.

При этом необходимо отметить, что если для классической науки идеалами научного знания служили простота, линейность, исключение неопределенности, то этого недостаточно для специалиста 21 века, когда основой развития становится нелинейность и непредсказуемость с точки зрения классической науки. Классический, господствующий по сей день подход к управлению сложноорганизованными системами основывается на линейном представлении об их функционировании. Согласно этому представлению результат внешнего управляющего воздействия есть однозначное и линейное, предсказуемое следствие приложенных усилий. Чем больше вкладываешь, тем больше должна быть отдача.

Вместе с бурным развитием статистических теорий (теория игр, теория ошибок, статистическая физика, демография и пр.) происходил переход к вероятностному стилю научного мышления, статистические закономерности перестали рассматриваться как нечто временное и преходящее. Интенсивное развитие системных исследований и кибернетики привело к очередному изменению в стиле научного мышления. Системный подход не отменяет вероятностное видение мира, дополняя его такими важнейшими элементами, как сложность, системность, целенаправленность.

Новое научное мышление должно включать в себя три основные идеи: нелинейность, открытость, диссипативность:

- нелинейность как необычная реакция на внешние воздействия, когда "правильное" воздействие оказывает большее влияние на эволюцию системы, чем воздействие более сильное, но организованное не адекватно ее собственным тенденциям;
- открытость, наличие внешних источников (стоков) как необходимое условие существования неравновесных состояний, в противоположность замкнутой системе, неизбежно стремящейся, в соответствии со вторым началом термодинамики, к однородному равновесному состоянию;
- диссипативность как фактор "естественного отбора", разрушающий все, что не отвечает тенденциям развития.

Формирование нового мышления специалиста по защите информации – главная задача не только кафедры безопасных информационных технологий, но и всего Института точной механики и оптики.

Литература

1. Информатика. Энциклопедический словарь для начинающих / ред. Д.А. Поспелов. М. Педагогика-Пресс, 1994.
2. Шрайберг Я.Л., Гончаров М.В. Справочное руководство по основам информатики и вычислительной техники. М.: Финансы и статистика, 1995.
3. Информатика и культура. Сб. научных трудов. Новосибирск: Наука, СО, 1990.
4. Блюменау Д.И. Информация и информационный сервис. Л.: Наука, 1989.
5. Информационная технология: Вопросы развития и применения. Киев.: Наук.думка, 1988
6. Концепция информатизации образования // Информатика и образование. 1990. № 1.
7. Терминологический словарь по основам информатики и вычислительной техники / ред. А.П. Ершов, Н.М. Шанский.. М.: Просвещение, 1991. 159 с.
8. Заварыкин В.М. и др. Основы информатики и вычислительной техники: Учеб. пособие для студентов пед. ин-тов по физ.-мат. спец. М.: Просвещение, 1989. 207 с.
9. Энциклопедия кибернетики. Киев: Главная редакция украинской советской энциклопедии, 1974.
10. Российская Федерация. Федеральный закон "Об информации, информатизации и защите информации": Принят Государственной Думой 25 января 1995 г. (опубл.: Собрание законодательства Российской Федерации. 1995. № 8).
11. Гнеденко Б.В. Введение в специальность математика. М.: Наука, 1991. 240 с.

НАШИ АВТОРЫ

Акунов Талайбек Абакирович – кандидат технических наук, докторант кафедры систем управления и информатики.

Акунова Акылай – кандидат технических наук, докторант кафедры систем управления и информатики

Алиев Руслан Тауфикович – аспирант кафедры вычислительной техники.

Баев Андрей Петрович – кандидат технических наук, научный сотрудник НИИТМ.

Баженов Антон Георгиевич – студент кафедры компьютерных технологий.

Бобцов Алексей Алексеевич – кандидат технических наук, докторант кафедры систем управления и информатики.

Бондаренко Игорь Борисович – ассистент кафедры проектирования компьютерных систем.

Боярский Кирилл Кириллович – кандидат физ.-мат. наук, доцент кафедры физики, kira@emi.spb.su.

Бушуев Александр Борисович – кандидат технических наук, докторант кафедры систем управления и информатики.

Васильев Владимир Николаевич – доктор технических наук, профессор, ректор.

Виноградов Юрий Николаевич – зам. главного конструктора ОКБ "Электроавтоматика".

Вознюк Юлия Родионовна – аспирант кафедры компьютерных технологий.

Волков Михаил Владимирович – студент кафедры компьютерных технологий.

Воротягин Сергей Николаевич – аспирант кафедры компьютерных технологий

Вурья Константин Евгеньевич – студент кафедры материаловедения..

Гатчин Юрий Арменакович – доктор технических наук, зав. кафедрой проектирования компьютерных систем.

Гончаренко Михаил Робертович – кандидат технических наук, начальник отдела НИИТМ.

Громов Геннадий Юрьевич – старший преподаватель кафедры вычислительной техники.

Гугель Юрий Викторович – зам. директора "Вузтелекомцентра".

Гурбашков Максим Борисович – студент кафедры систем управления и информатики.

Гуров Игорь Петрович – доктор технических наук, профессор, зам. декана факультета компьютерных технологий и управления.

Демин Анатолий Владимирович – доктор технических наук, профессор кафедры информатики и прикладной математики.

Дорохин Денис Алексеевич – аспирант кафедры вычислительной техники.

Дроздов Валентин Нилович – доктор технических наук, профессор, директор СЗИП СПбГУТД.

Захаров Алексей Сергеевич – студент кафедры компьютерных технологий..

Звонов Валерий Степанович – кандидат физ.-мат. наук, доцент кафедры безопасных информационных технологий.

Зыков Анатолий Геннадьевич – доцент кафедры информатики и прикладной математики.

Исаков Алексей Сергеевич – студент кафедры систем управления и информатики.

Казаков Матвей Алексеевич – студент кафедры компьютерных технологий.

Каневский Евгений Александрович – кандидат технических наук, ведущий научный сотрудник СПб ЭМИ, kanev@emi.spb.su.

Кириллов Владимир Васильевич – кандидат технических наук, профессор кафедры вычислительной техники.

Кирчин Юрий Геннадьевич – кандидат технических наук, доцент кафедры компьютерных технологий.

Ключев Аркадий Олегович – кандидат технических наук, доцент кафедры вычислительной техники, kluchev@de.ifmo.ru

Коваленко Александр Николаевич – кандидат физ.-мат. наук, зав. кафедрой СЗИП СПбГУТД.

Ковязин Рустам Раисович – аспирант кафедры вычислительной техники.

Кожевникова Лариса Владиславовна – инженер-программист кафедры систем управления и информатики.

Копорский Николай Сергеевич – соискатель кафедры информатики и прикладной математики.

Корнеев Георгий Александрович – студент кафедры компьютерных технологий.

Коровяков Анатолий Николаевич – кандидат технических наук, доцент кафедры систем управления и информатики.

Коротков Константин Георгиевич – доктор технических наук, профессор кафедры проектирования компьютерных систем.

Крылов Борис Алексеевич – кандидат технических наук, ст. преподаватель кафедры проектирования компьютерных систем..

Лаздин Артур Вячеславович – доцент кафедры информатики и прикладной математики.

Лезин Генрих Валерьевич – кандидат технических наук, ведущий научный сотрудник СПб ЭМИ, lezin@emi.spb.su

Лукичев Александр Николаевич – студент кафедры вычислительной техники, sash@dl.ifmo.ru.

Макаров Николай Сергеевич – студент кафедры компьютерных технологий.

Мельников Андрей Александрович – аспирант кафедры систем управления и информатики.

Мирошник Илья Васильевич – доктор технических наук, профессор кафедры систем управления и информатики.

Муравьева-Витковская Людмила Александровна – ст. преподаватель кафедры вычислительной техники.

Немолочнов Олег Фомич – доктор технических наук, профессор, зав. кафедрой информатики и прикладной математики.

Озерский Александр Маркович – студент кафедры компьютерных технологий.

Окулевич Владимир Васильевич – аспирант кафедры вычислительной техники.

Осиповцева Ольга Святославовна – студентка кафедры систем управления и информатики.

Осовецкий Леонид Георгиевич – доктор технических наук, профессор, зав. кафедрой безопасных информационных технологий..

Павличенко Игорь Петрович – аспирант кафедры вычислительной техники..

Павлушко Ольга Викторовна – студентка кафедры компьютерных технологий.

Парфенов Владимир Глебович – доктор технических наук, профессор, декан факультета.

Платунов Алексей Евгеньевич – кандидат технических наук, доцент кафедры вычислительной техники, platunov@dl.ifmo.ru.

Поляков Владимир Иванович – кандидат технических наук, доцент кафедры вычислительной техники.

Постников Николай Павлович – аспирант кафедры вычислительной техники.

Приходько Олег Анатольевич – кандидат физ.-мат. наук, доцент кафедры материаловедения, mater@spb.runnet.ru

Рогачевский Александр Маркович – зам. главного конструктора ОКБ "Электроавтоматика".

Рубина Нина Алексеевна – аспирант кафедры компьютерных технологий.

Рукуйжа Елена Викторовна – зав. лабораторией кафедры систем управления и информатики.

Салфетников Александр Иванович – аспирант кафедры систем управления и информатики.

Самохвалова Ольга Геннадьевна – аспирант кафедры вычислительной техники, sog_m@hotmail.ru

Скорубский Владимир Иванович – кандидат технических наук, доцент кафедры вычислительной техники.

Слита Ольга Валерьевна – студентка кафедры систем управления и информатики.

Смирнов Дмитрий Сергеевич – студент кафедры компьютерных технологий.

Сорокин Александр Васильевич – аспирант-заочник кафедры компьютерных технологий.

Станкевич Андрей Сергеевич – студент кафедры компьютерных технологий.

Сударчиков Сергей Алексеевич – аспирант-заочник кафедры систем управления и информатики.

Ушаков Анатолий Владимирович – доктор технических наук, профессор кафедры систем управления и информатики.

Хабалов Владимир Викторович – кандидат технических наук, доцент кафедры систем управления и информатики.

Холунин Сергей Анатольевич – студент кафедры систем управления и информатики.

Черемухин Виктор Станиславович – аспирант кафедры вычислительной техники, vit@cs.ifmo.ru.

Чистяков Алексей Григорьевич – аспирант кафедры вычислительной техники, aosp@dl.ifmo.ru

Шалаев Алексей Николаевич – студент кафедры систем управления и информатики.

Юдин Михаил Юрьевич – кандидат технических наук, доцент кафедры проектирования компьютерных систем.

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ	3
Васильев В.Н., Казаков М.А., Корнеев Г.А., Парфенов В.Г., Станкевич А.С. Полуфинальные соревнования чемпионата мира по программированию 2001-2002 гг	3
1. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ	10
Алиев Р.Т. Методы управления трафиком в мультисервисных сетях.....	10
Васильев В.Н., Гугель Ю.В., Гуров И.П. Анализ методов передачи видеоинформации в компьютерных сетях.....	14
Муравьева-Витковская Л.А. Анализ влияния характера информационных потоков на качество функционирования телекоммуникационной сети.....	27
Рубина Н.А., Кирчин Ю.Г. Информационные технологии на основе расширенной иерархии уровней OSI-модели.....	30
Баженев А.Г. Построение систем электронного документооборота на уровне информационной сети предприятия	46
Воротягин С.Н. Анализ нагрузки и классификация приложений в локальных вычислительных сетях.....	52
Боярский К.К., Каневский Е.А., Лезин Г.В. Концептуальные модели в базах знаний	57
Вурья К.Е., Приходько О.А. Разработка базы данных по металлическим материалам для университета	63
Рождественская М.Г., Рождественский И.В. Технология виртуальной реальности в визуализации собраний данных	67
2. ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ПРОГРАММИРОВАНИЕ	71
Ковязин Р.Р., Чистяков А.Г. Использование аппаратных интерпретаторов для управления неоднородными компонентами микропроцессорной системы	71
Платунов А.Е. Архитектурные абстракции в технологии сквозного проектирования встроенных вычислительных систем.....	76
Постников Н.П. Динамические инструментальные компоненты.....	83
Самохвалова О.Г. Выбор оптимального алгоритма планирования при разработке программного обеспечения систем реального времени	88
Окулевич В.В. Исследование операционных сред прикладных задач в информационно-управляющих системах.....	92
Лукичев А.Н., Ключев А.О. Организация тестового программного обеспечения встроенных систем	95
Павличенко И.П. Метод обеспечения замкнутости программной среды путем разграничения прав доступа к файловой системе	100
Громов Г.Ю., Дорохин Д.А., Черемухин В.С. Решение специфических задач аудита базы данных	102

Громов Г.Ю., Дорохин Д.А., Кириллов В.В., Черемухин В.С. Проблема интеграции локальных автоматизированных систем управления в образовательной сфере деятельности.....	105
3. АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ	107
Зыков А.Г., Немолочнов О.Ф., Поляков В.И. Универсальная модель последовательностных схем в САПР	107
Зыков А.Г., Немолочнов О.Ф., Поляков В.И. Построение комплексного покрытия последовательностных схем методом пересечения покрытий систем булевых функций.....	109
Лаздин А.В., Немолочнов О.Ф. Оценка сложности графа функциональной программы.....	112
Лаздин А.В., Немолочнов О.Ф. Метод построения графа функциональной программы для решения задач верификации и тестирования	118
Гончаров Д.А., Скорубский В.И. Расчет размещения схем в логических матрицах	122
Гатчин Ю.А., Бондаренко И.Б. Оптимизация проектных решений в САПР автоматизированных технологических комплексов	127
Юдин М.Ю. Характеристики функционирования гибридной САПР.....	135
Демин А.В., Копорский Н.С., Коршунов А.И. Имитационное моделирование оптических приборов со встроенной системой стабилизации изображения	138
Рогачевский А.М., Виноградов Ю.Н. Принципы разработки и модернизации комплексов бортового оборудования (КБО) перспективных летательных аппаратов	143
4. КОМПЬЮТЕРНЫЙ АНАЛИЗ ИЗОБРАЖЕНИЙ И СИГНАЛОВ.....	148
Беляков А.В., Гуров И.П. Улучшение интерферограмм методом вейвлет-преобразования.....	148
Коротков К.Г., Крылов Б.А. Параметрический анализ полутонных изображений.....	153
Захаров А.С. Восстановление фазы интерференционных полос методом нелинейной двумерной фильтрации Калмана	157
Волков М.В. Восстановление линий экстремумов сложных картин интерференционных полос.....	162
Вознюк Ю.Р. Динамическое оценивание локального направления полос в низкоконтрастных интерферограммах.....	172
Гуров И.П., Смирнов Д.С. Улучшение качества изображений методом Ван Циттерта.....	178
Павлушко О.В. Цифровые фотоаппараты: сравнительные характеристики и применение	183
Макаров Н.С. Оптические волоконные усилители информационных сигналов.....	191
Гуров И.П., Жиганов П.Г., Озерский А.М. Анализ акустических сигналов на основе метода фильтрации Калмана	200
Жаринов И.О. Математико-статистические методы в задаче параметризации описания классов ЭЭГ-сигнала	207

5. УПРАВЛЕНИЕ И ИНФОРМАТИКА В ТЕХНИЧЕСКИХ СИСТЕМАХ	210
Акунов Т.А., Ушаков А.В. Связь алгебраических спектров собственных значений и сингулярных чисел в задаче обеспечения стабильности эллипсоидных показателей качества.....	210
Акунова А., Ушаков А.В. Колебательные процессы в плохо обусловленных апериодических системах.....	217
Бобцов А.А., Холунин С.А. Развитие методов робастного управления в задачах адаптации	223
Бобцов А.А., Гурбашков М.Б. Стабилизация линейных стационарных систем по состоянию	229
Кожевникова Л.В., Ушаков А.В. Кронекеровская матричная модель динамических процессов с модуляцией.....	232
Мирошник И.В., Шалаев А.Н. Управление траекторным движением автономных роботов	237
Мельников А.А., Рукуйжа Е.В., Ушаков А.В. Использование свойств матриц над конечными полями для обнаружения неустойчивых циклов и неподвижных состояний двоичных динамических систем	243
Мирошник И.В., Слита О.В. Частичная устойчивость и аттрактивность нетривиальных множеств	250
Сударчиков С.А., Ушаков А.В. Оценка запасов устойчивости систем с интервальными параметрами	257
Салфетников А.И., Хабалов В.В. Идентификация линейных операций	263
Баев А.П., Гончаренко М.Р., Исаков А.С., Коровьяков А.Н., Осипцева О.С. Векторная система управления устройством рекуперации на микропроцессоре TMS320F243	266
Дроздов В.Н., Коваленко А.А. Конвергенция электронных и печатных форм представления информации	268
Бушуев А.Б. Синтез информационно-энергетических связей в структурах изобретений.....	274
6. БЕЗОПАСНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ	277
Звонов В.С., Осовецкий Л.Г. Проблемы образования специалиста по защите информации.....	277
НАШИ АВТОРЫ.....	280

Научно-технический вестник СПб ГИТМО (ТУ). Выпуск 6.
Информационные, вычислительные и управляющие системы / Главный редактор В.Н. Васильев. СПб: СПб ГИТМО(ТУ), 2002. 285 с.

**НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
СПб ГИТМО (ТУ)
Выпуск 6
ИНФОРМАЦИОННЫЕ, ВЫЧИСЛИТЕЛЬНЫЕ И УПРАВЛЯЮЩИЕ
СИСТЕМЫ**

Главный редактор В.Н. Васильев

Дизайн обложки В.А. Петров
Редакционно-издательский отдел СПб ГИТМО (ТУ)
Зав. РИО Н.Ф. Гусарова
Лицензия ИД № 00408 от 05.11.99.
Подписано в печать 15.11.02.
Заказ 516. Тираж 100 экз.