

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ
РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Межвузовский сборник научных трудов

Рязань 2019

УДК 004

Информационные технологии, межвузовский сборник научных трудов. – Рязань: Рязанский государственный радиотехнический университет имени В.Ф. Уткина, 2019. – 263 с.

ISBN 978-5-6043323-3-7

Публикуются статьи о проблемах использования информационных технологий в науке и образовании.

Сборник рассчитан на научно-педагогических работников вузов и может быть использован студентами и аспирантами инфокоммуникационных и информационных специальностей.

Редакционная коллегия:

Д-р техн. наук, проф. В.П. Корячко (ответственный редактор), д-р техн. наук, проф. Ю.Л. Леохин (Московский технический университет связи и информатики), д-р техн. наук, проф. В.П. Кулагин (МИРЭА – Российский технологический университет), д-р техн. наук, проф. А.Д. Иванников (Институт проблем проектирования в микроэлектронике Российской академии наук), д-р техн. наук, проф. С.В. Скворцов (РГРТУ), д-р техн. наук, проф. А.П. Шибанов (РГРТУ), канд. техн. наук А.Н. Сапрыкин (ответственный секретарь).

Рецензенты:

Кафедра ВПМ Рязанского государственного радиотехнического университета (зав. кафедрой, д-р техн. наук, проф. А.Н. Пылькин), д-р техн. наук, проф. В.А. Минаев (Московский государственный технический университета имени Н.Э. Баумана).

ISBN 978-5-6043323-3-7

©Рязанский государственный
радиотехнический университет
имени В.Ф. Уткина, 2019
©ИП Коняхин А.В. (Book Jet), 2019

СОДЕРЖАНИЕ

Аникеев Д.В., Аникеев С.В., Маркин А.В., Пудова А.В. Образовательные интернет – ресурсы «постреляционные базы данных»	10
Арчаков В.С. Варианты построения фазовращателей	22
Арчаков В.С. Фазовращатель тромбонного типа с диэлектрическими переходами	26
Баландин С.О., Сапрыкин А.Н. Особенности построения решений генетического алгоритма в задаче размещения элементов	29
Баранов В.Д. Тенденции развития современных систем автоматической оптической инспекции (АОИ) печатных узлов	32
Беляков А.В. Метрики качества в задачах ранжирования информации	36
Беляков А.В. Определения сходства изображения на основе глубокого обучения ...	39
Буданова В.В. Использование интерактивных программ тренажеров в обучающем процессе	41
Буробина А.С., Сапрыкин А.Н. Библиотека операторов генетического алгоритма для решения задачи поиска соответствий на двух множествах	44
Васюков В.В. Автоматизация установки предварительного вспенивания полистирола	49

Бу К.М.

Нахождение фундаментальной матрицы в линейной системе управления 53

Бу К.М.

Управляемость «в малом» для линейных неавтономных систем второго порядка..... 55

Бу К.М.

Связь предначальных и начальных условий в задачах управления 58

Диннер И.В.

Способы оптимизации SQL запросов 61

Диннер И.В.

Оптимизация запросов в системах баз данных 64

Дмитриева Т.А., Беляев Д.Ю.

Разработка объектов конфигурации программного обеспечения оформления отчетных документов вуза 67

Дмитриева Т.А., Палагин Д.А.

Разработка классов программного обеспечения, обучающего алгоритмическому языку 71

Дунцев Р.В.

Алгоритм оптимизации пространственной структуры радионавигационных ориентиров по частным производным 77

Емельянов И.С.

Исследование влияния воздействия различных факторов на узлы гибко-жестких печатных плат (ГЖПП) 81

Захаркин Ю.А.

Реализация модуля быстрого преобразования фурье на плис в формате представления чисел с фиксированной точкой 84

Захаркин Н.В., Федоров В.П.

Система оптической передачи информации с помощью линейно-поляризованных электромагнитных волн..... 87

Зверев Е.Г., Васильев Е.В.

Повышение линейности прецизионного амплитудного детектора 90

Зенина М.А.

Структура системы детектирования движущихся объектов в видеопотоке с использованием ПЛИС 95

Зенков Д.В., Десякович М.М.

Универсальный модуль расширения для автоматизации инженерных систем зданий и технологических процессов в промышленности 98

Ибрагимов А.Ю.

Помеха с негауссовским законом распределения 101

Иванова Н.С.

Репликация баз данных в информационных системах 104

Иванова Н.С.

Распределенная обработка данных в информационных системах с использованием Hadoop 107

Ивкина М.С.

Повышение качества классификации 110

Калинин Н.В.

Задача о восьми ферзях и ее применение 113

Калинкина Ю.Н.

Разработка конструкции системы управления приточно-вытяжной вентиляции 115

Крупина Т.М.

Генетический алгоритм для решения задач проектирования ЭС 118

Кузнецов А.И., Коклюков Н.В.

Обзор методов и алгоритмов обработки и анализа изображений 121

Лебедев А.М.

Воздействие излучения звуковой частоты на показания MEMS акселерометров 125

Лебедев И.А.

Проектирование портативного локатора выстрела 128

Лебедева Н.С., Гостин А.М., Сапрыкин А.Н.

Разработка модуля анкетирования образовательного портала РГРТУ 131

Ликучев В.Ю.

Программная архитектура ситемы управления беспилотными летательными аппаратами специального назначения 135

Макарова В.А., Новиков И.С.

Исследование среды обитания аквариумных рыб и беспозвоночных для создания устройства жизнеобеспечения 140

Матюхин А.А.

Использование диаком для проведения диагностики цифровых устройств с ТТЛ логикой 144

Махнач Е.С.

Организация безопасности с использованием технологий «Умный дом» 147

Миронова М.Р.

Особенности работы ультразвуковых измерителей расстояний 150

Михеева М.С.

Жизненный цикл РЭС и способы его увеличения 153

Морозова Ю.С., Скворцов С.В.

Разработка алгоритма многомерной трассировки для решения проектных задач 156

Музюкин И.А., Сапрыкин А.Н.

Использование островной модели генетического алгоритма при компоновке блоков электронно-вычислительных средств 163

Мычко С.И.

Микросервисная архитектура 166

Нгуен А.З., Шибанов А.П.

Система космической связи с резервированием кадров и использованием GERT-сетей 169

Николашин В.А., Якушкин А.Э.

Среда программирования как особенность выбора ПЛК для АСУ ТП 173

Новиков И.С., Макарова В.А.

Программа расчета прочностных параметров несущей конструкции 176

Новиков А.П., Митрошин А.А.

Расширенная матрица компетенций 179

Носенко Д.С.

Разработка и внедрение модуля управления устройства технологической тренировки 183

Носенко Д.С.

Исследование и разработка конструкции модуля обогрева одежды специального назначения 185

Осин И.Г.

Разработка программной системы планирования задач управления группами виртуальных машин в программно-конфигурируемых сетях 188

Пехнов А.С.

Криптографическая библиотека OpenSSL 190

Попов С.В.

Структура модуля системы технического зрения беспилотного летательного аппарата 193

Потапкина Н.С.

Применение ПЛИС при конструировании и проектировании цифровых устройств 196

Приходько Р.А.

О методе формирования микросервисной архитектуры бизнес-приложений 201

Пишеничников В.А., Фоломкин И.И.

Внедрение автоматизации процесса подачи вакуума при литье по газифицируемым моделям 206

Скуров Д.А.

Применение отечественных защищённых ОС в системах контроля специального назначения в условиях импортозамещения 209

Соловьев С.А., Сускин В.В.

Учебно-исследовательская сапр на основе САПР KiCad 213

Струков А.М.

Исследование подходов к параллельной реализации метода каскадов 215

Трухачев В.П.

Разработка и внедрение руководства по качеству калибровки средств измерений 218

Фам А.М., Шибанов А.П.

Комбинированная система моделирования на основе сетей GERT и Петри 221

Филатов К.Ю.

Современные тенденции юзабилити 225

Филимонов В.В., Сускин В.В.

Оценка производительности современных вычислительных систем 227

Филимонов В.В., Сускин В.В.

Методы увеличения вычислительной производительности 231

Фоломкин И.И., Щукарев А.С., Пишеничников А.В.

Моделирование и анализ рисков в управлении проектами 236

Холопов И.В., Васильев Е.В.

Формирование гармонических колебаний с помощью различных алгоритмов на ПЛИС 238

Хоминич Л.Н., Сергеев М.А.

Технологии изготовления резисторов, в том числе наиболее перспективные..... 242

Цветков К.Ю.

Сравнительный анализ сред разработки СУБД..... 246

Чудинин А.В., Сускин В.В.

Структура и принцип работы двигателя внутреннего сгорания на водороде..... 249

Шестакова Т.С.

Исследование и разработка комплекса для обработки электрофизиологической информации 254

Ширенин И.А.

Событийная модель построения графических приложений..... 257

Якушкин А.Э., Николашин В.А.

Модернизация комплекса программно-технических средств установки производства битума..... 260

УДК 004.65

Д.В. АНИКЕЕВ, С.В. АНИКЕЕВ, А.В. МАРКИН, А.В. ПУДОВА
Рязанский государственный радиотехнический университет

ОБРАЗОВАТЕЛЬНЫЕ ИНТЕРНЕТ – РЕСУРСЫ «ПОСТРЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ»

Рассматривается образовательный интернет-ресурс «Постреляционные базы данных», предназначенный для обучения студентов вузов, обучающихся по направлениям инженерного образования, связанным с разработкой, и сопровождением баз данных информационных систем. Курс ориентирован на приобретение знаний и умений в области применения NoSQL СУБД.

Введение

В связи с постоянным ростом объема данных и нагрузки, традиционные, реляционные базы данных (БД) перестают в полной мере справляться с поставленными задачами. Особенно остро снижение быстродействия таких систем управления базами данных (СУБД) ощущается в случае, когда необходимо производить информационный обмен между двумя или большим числом БД распределенных информационных систем. Кроме того во время асинхронного обмена информацией может происходить нарушение целостности хранимых данных. Все это повышает интерес к постреляционным (NoSQL) СУБД и заставляет рассматривать возможность их использования. Они благодаря своей распределенной структуре находятся не в одном, а в нескольких местах, что позволяет, в частности, заменить один дорогой компьютер на несколько дешевых.

В настоящее время наблюдается тенденция к использованию нескольких технологий и систем в рамках одного приложения для решения различных задач. Существуют различные подходы реализации доступа к данным. Они основываются на следующих принципах организации слоя работы с данными при использовании нескольких хранилищ разного типа:

- 1) NoSQL СУБД используется для передачи данных между несколькими реляционными БД;
- 2) NoSQL СУБД заменяет все, кроме одной реляционной СУБД;
- 3) NoSQL СУБД полностью заменяет все СУБД.

Однако, несмотря на это и другие преимущества нового подхода к структуре БД, при создании информационных систем выбор падает на традиционные, а не на постреляционные БД.

Одной из причин является недостаточная осведомленность разработчиков программных средств с технологиями нового подхода, и поэтому они не решаются их использовать в проектах. Поскольку о возможностях NoSQL СУБД разработчики могут узнать только при самостоятельном изучении.

Поэтому появляется необходимость на этапе обучения вузах изучения не только реляционных, но и постреляционных БД.

Разработанный дистанционный учебный курс «Постреляционные базы данных» является электронным ресурсом и размещен на виртуальной кафедре АСУ [1], представляющей собой систему управления обучением в электронной среде. Подключение в рамках рассматриваемого курса внешних ссылок на ресурсы сети Интернет позволяет расширить образовательную среду за счет привлечения дополнительных источников информации.

1. Содержание курса

Курс направлен на изучение документо-ориентированной СУБД MongoDB [2].

Содержание курса предполагает изучение теоретического материала, выполнение лабораторных работ, контрольных заданий, тестов и итоговый контроль полученных знаний и приобретенных практических навыков и умений.

Для работы с дистанционным учебным курсом пользователю необходимо наличие компьютера офисной конфигурации или мобильного телефона с подключением в сеть Интернет (рекомендуется использовать браузер Mozilla Firefox или Google Chrome) и доступом к серверу системы дистанционного обучения вуза, на котором размещается данный курс.

Для просмотра учебного материала и выполнения заданий по темам курса требуется установка соответствующего программного обеспечения:

1) программы для просмотра pdf-файлов: Adobe Reader, Foxit Reader и другие, вместо указанных программ можно использовать браузер;

2) пакета офисных программ (OpenOffice, MS Office);

3) изучаемой СУБД (MongoDB).

Теоретический материал содержит 7 лекций по соответствующим темам. В лекции № 1 изучаются минимальные сведения из систем баз данных, необходимые для понимания всего последующего материала. Здесь излагаются основные понятия и принципы организации постреляционных баз данных на примере учебной базы данных, пред-

ставлена классификация современных БД и СУБД, описаны и проанализированы основные преимущества и недостатки рассматриваемых баз данных.

В лекции № 2 рассматриваются основные принципы работы и особенности использования СУБД MongoDB и графической оболочки Robomongo [3].

В лекции № 3 рассматривается, каким образом осуществляется работа с данными при помощи запросов.

В лекции № 4 рассматривается, каким образом осуществляется манипулирование данными в уже созданной базе данных.

В лекциях № 5, 6 рассматриваются возможности СУБД MongoDB по выборке данных. Здесь изучается построение запросов на выборку к одной и нескольким коллекциям (таблицам) базы данных, а также методы и функции агрегирования данных.

В лекции № 7 рассматриваются основные возможности постреляционных баз данных по репликации и шардингу.

В лекции № 8 рассматриваются вопросы защиты данных. Излагаются общие правила разграничения доступа пользователей к объектам базы данных, осуществление резервного копирования и восстановления базы данных, коллекций и результатов выборки.

Теоретический материал сопровождается большим количеством как простых, так и достаточно сложных примеров запросов на учебной базе данных. Учебная БД представляет собой очень упрощенный пример информационной модели расчетно-аналитической составляющей комплекса «Абонент+» [4]. Указанная модель используется для информационного обеспечения деятельности организаций по оказанию населению жилищно-коммунальных услуг и расчетам за них.

Лабораторный практикум содержит 6 лабораторных работ - одной по каждой теме. При этом осуществлена концепция автоматизированного лабораторного практикума удаленного доступа.

Блок промежуточного контроля предполагает выполнение письменных и тестовых заданий по каждой теме; блок итогового контроля – 1 тест для экзамена.

2. Структура электронного ресурса

В качестве образовательной среды используется система управления курсами электронного обучения Moodle [5]. Представляет собой свободное web-приложение, предоставляющее возможность создавать сайты для онлайн-обучения.

Дистанционный учебный курс «Постреляционные базы данных»

состоит из 5 разделов и 12 модулей. Описание разделов и модулей представлено в таблице 1.

Таблица 1.

Описание модулей дистанционного учебного курса
«Постреляционные базы данных».

№	Название модуля	Содержание	Лекция	Лабораторная работа	Тест
1	<i>Общее</i>	Объявления Новостной форум			
2	<i>Тема 1</i> Раздел 1. Обзор СУБД NoSQL-типа	Введение Лекция по соответствующей теме	Лекция № 1		
3	<i>Тема 2</i> Раздел 2. Документо-ориентированная СУ-РабД MongoDB	Лекция по соответствующей теме Лабораторный практикум тестирование	Лекция № 2	Лабораторная работа № 1	Тест № 1
4	<i>Тема 3</i> Раздел 3. Декларативное программирование в MongoDB	Введение Лекция по соответствующей теме	Лекция № 3		
5	<i>Тема 4</i> Методы модификации данных	Лекция по соответствующей теме Лабораторный практикум, содержащий разобранные задачи и задания для самостоятельной работы Тестирование по соответствующей теме	Лекция № 4	Лабораторная работа № 2	Тест № 2

Продолжение таблицы 1.

6	<i>Тема 5</i> Методы поиска информации	Лекция по соответствующей теме Лабораторный практикум, содержащий разобранные задачи и задания для самостоятельной работы Тестирование по соответствующей теме	Лекция № 5	Лабораторная работа № 3	Тест № 3
7	<i>Тема 6</i> Агрегация	Лекция по соответствующей теме Лабораторный практикум, содержащий разобранные задачи и задания для самостоятельной работы Тестирование по соответствующей теме	Лекция № 6	Лабораторная работа № 4	Тест № 4
8	<i>Тема 7</i> РАЗДЕЛ 4. Распределенные БД MongoDB	Лекция по соответствующей теме Лабораторный практикум, содержащий разобранные задачи и задания для самостоятельной работы Тестирование по соответствующей теме	Лекция № 7	Лабораторная работа № 5	Тест № 5
9	<i>Тема 8</i> РАЗДЕЛ 5. Основы администрирования БД MongoDB	Лекция по соответствующей теме Лабораторный практикум, содержащий разобранные задачи и	Лекция № 8	Лабораторная работа № 6	Тест № 6

Продолжение таблицы 1.

		и задания для самостоятельной работы Тестирование по соответствующей теме			
10	<i>Тема 9</i> Демонстрационное приложение учебной БД «Абонент»	Приложение создано в среде разработки Microsoft Visual Studio на языке C#, с использованием HTML+JavaScript, документо-ориентированной СУБД MongoDB			
11	<i>Тема 10</i> Демонстрационное приложение учебной БД «Абонент»	Приложение создано в среде разработки Microsoft Visual Studio на языке C#, с использованием HTML+JavaScript, In-Memory СУБД Tarantool			
12	<i>Тема 11</i> Демонстрационное приложение учебной БД «Абонент»	Приложение создано в среде эк-пресс-разработки Oracle Aplication Express (APEX) и расположено в облаке Oracle cloud.			

Курс строится в соответствии с семестровым расписанием. На сайте курса размещаются все теоретические материалы, методические указания к лабораторным работам и контрольные задания, ссылки на все необходимые инструментальные средства. Здесь используются практически все ресурсы образовательной среды: общение со студентами в виде обмена сообщениями, форумов и чата; установка сроков

выполнения заданий, формирование учебных групп, тестирование, опросы, анкетирование, информирование о предстоящих событиях, объявления, управление журналом оценок (см. рис. 1).

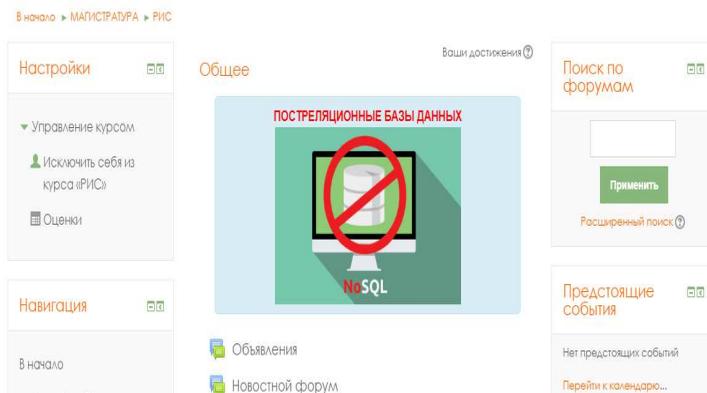


Рис. 1. Первый модуль курса.

Учебно-методические материалы дистанционного курса представлены в виде pdf-файлов, картинок и ссылок на ресурсы Интернет.

Лекции, расположенные в тематических модулях курса, содержат необходимый теоретический материал, который снабжен иллюстрациями, рисунками, разобранными задачами, примерами и контрольными вопросами (см. рис. 2).

Практическая часть курса предназначена для приобретения студентами умений и навыков и заключается в выполнении ряда лабораторных работ.

Тема 1

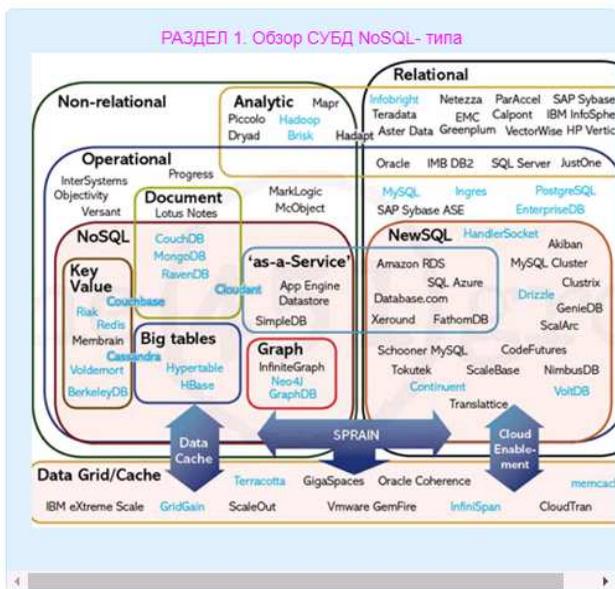


Рис. 2. Первая тема курса.

Выполнение лабораторных работ осуществляется в документо-ориентированной СУБД MongoDB, предоставляющей бесплатную версию с полным функционалом.

Они выполняется по принципу поэтапного усложнения задач использования СУБД от создания базы данных до основ ее администрирования, резервного копирования и сегментирования на нескольких компьютерах (см. рис. 3).

 Лекция 1

Введение

1. Постреляционные базы данных
 - 1.1. Факторы появления постреляционных БД
 - 1.1.1. Рост объема информации
 - 1.1.2. Ограничения в использовании РБД
 - 1.1.3. Новые источники и области применения информационных систем
 - 1.2. Классификация постреляционных баз данных
 - 1.2.1. Хранилища NoSQL типа
 - 1.2.2. Хранилища NewSQL типа
2. Системы управления БД NoSQL типа
 - 2.1. Системы Ключ-значение
 - 2.2. Колонко-ориентированные системы
 - 2.3. Документо-ориентированные системы
 - 2.4. Графовые системы

 Основные виды NoSQL баз данных (видео) □

Рис. 3. Представление лекции на курсе

Лабораторный практикум состоит из следующих работ:

- 1) знакомство с графической оболочкой Robomongo. Создание локальной учебной базы данных;
- 2) модификация данных;
- 3) поиск данных в одной коллекции;
- 4) агрегирование данных;
- 5) создание распределенной учебной базы данных. Реализация репликации и шардинга;
- 6) администрирование БД.

Для защиты лабораторных работ, а также для итогового контроля используется оригинальная информационная система [6]. Указанная информационная система имеет структуру MVC модели построенной в среде программирования Visual Studio и разработана с использованием следующих технологий:

- 1) Angular JS версии 1.6.4 (реализация пользовательского интерфейса);

- 2) jQuery 3.1.1(реализация пользовательского интерфейса);
- 3) FirebirdSQL.Data.FirebirdClient версии 5.9.1 (реализация ядра информационной системы);
- 4) MongoDB.Driver версии 2.6.1 (учебная БД, на которой реализуются запросы);
- 5) TinyMCE версии 4.5.7 (редактор для выполнения запросов);
- 6) Bootstrap версии 3.3.7 (реализация графического интерфейса).

Защита лабораторной работы проводится в форме дистанционного тестирования [6]. При этом в 1-й, 2-й, 5-й и 6-й лабораторных работах контроль знаний ведется с помощью традиционного тестирования. В работах 3-й и 4-й необходимо построить, ввести и выполнить запросы согласно предлагаемым заданиям под контролем информационной системы.

Тесты состоят из заданий разного уровня сложности. Задания на выбор ответа и ввод текстового ответа являются легкими. Запросы подразделены на легкие, средние и сложные. Сложные вопросы отмечены при прохождении теста картинками. Запросы вводятся в форму текстового редактора TinyMCE (см. рис. 4).

4_Агрегирование данных

Задания



Текущее задание

Ввести всю информацию о погашенных заявках с неисправностью "Засорилась водогрейная колонка"

[Содержимое РБД](#) [Описание РБД](#) [Описание ДСБД](#) [Сообщить о некорректности задания](#)

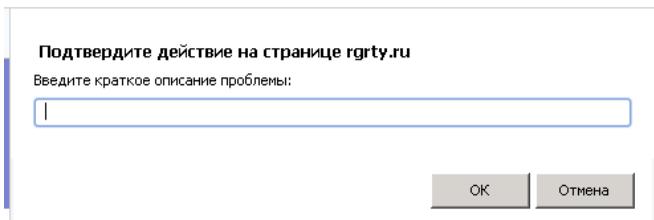
[Посмотреть ТРЗ](#) [Правильная ТРЗ](#)

1. Постройте и введите запрос или ответьте на вопрос.

2. Запросы должны быть в формате `db.getCollection`.

Рис. 4. Форма ввода запроса

Важным аспектом проверки правильности студенческих ответов является программное автоматическое сравнение результатов выполнения эталонного и студенческого запросов к учебной базе данных на реальной СУБД, а также возможность сообщить преподавателю о проблемах, возникших при решении задания (см. рис. 5).



Подтвердите действие на странице rgrtu.ru

Введите краткое описание проблемы:

OK Отмена

Рис. 5. Форма ввода сообщения о проблеме.

Демонстрационные приложения позволяют студентам представить реальное применение и взаимодействие изучаемых технологий.

3. Технология обучения

За несколько дней до проведения лекции ее содержимое в виде pdf - файла выкладывается на страницу курса для предварительного, самостоятельного ознакомления с ней студентов. Лекции проводятся по расписанию в компьютеризированной аудитории, снабженной видеопроектором. Не тратится время на чтение и запись самой лекции студентами, так как она проводится в виде представления и обсуждения особенностей теоретического материала, разбора и выполнения каждым студентом непосредственно на компьютере всех примеров запросов и скриптов, а также ответов на вопросы.

Доступ к выполнению очередной работы лабораторного практикума на сайте курса предоставляется студентам за 5 учебных дней до аудиторного занятия по расписанию. Студенты имеют возможность заранее ознакомиться с лабораторным заданием, порядком выполнения лабораторной работы, требованиями к содержанию отчета, вариантом индивидуального задания. Предлагается по возможности выполнить лабораторную работу в удобное время, оформить отчет и выслать его преподавателю для проверки. Результатом проверки отчета преподавателем могут быть или допуск к его защите или возврат на доработку или исправление. Особенностью лабораторных заданий является необходимость самостоятельного формулирования заданий на выборку или модификацию данных в учебной базе данных, а также на реализацию бизнес-логики; построение соответствующих запросов или программирование хранимых процедур и триггеров; выполнение их на учебной базе данных; доказательство правильности полученных решений.

Заключение

При создании дистанционного курса учитывался основной принцип – максимально полное и наглядное представление учебного

материала, обеспечивающее их самостоятельное изучение в индивидуальном темпе, а также организация эффективного и быстрого доступа обучаемых к необходимой информации. Интерактивные элементы коммуникативного назначения позволяют устанавливать информационное взаимодействие с преподавателем курса, получать необходимые консультации.

После прохождения разработанного дистанционного курса «Постреляционные базы данных» студент приобретет следующие знания и навыки по работе с NoSQL системами управления базами данных, на примере работы с документо-ориентированной СУБД MongoDB:

1) создания и редактирования базы данных, не имеющей табличной структуры (документо-ориентированной);

2) изменения структуры БД не поддерживающей ссылочной целостности и имеющей уровни вложенности;

3) поиска данных в одной коллекции базы данных;

4) выборки информации из нескольких коллекций, не имеющих внешних ключей и связей между коллекциями, и подколлекций/поддокументов (вложенных объектов);

5) по репликации и горизонтальному масштабированию данных при помощи шардинга, как основных преимуществ NoSQL СУБД;

6) администрирования NoSQL БД: создание пользователей, прав пользователя и ролей, резервное копирование и восстановление БД, а также отличие указанных механизмов от используемых в реляционных БД.

Представленный курс апробирован в Рязанском государственном радиотехническом университете по направлению «Информационные системы и технологии» для студентов магистратуры. Может быть использован как для смешанной, так и дистанционной форм обучения. Дистанционное обучение для студентов проводится в рамках основной образовательной программы.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Виртуальная кафедра АСУ РГРТУ [Электронный ресурс] URL: <https://rgrty.ru/> (дата обращения: 27.03.2019).

2. Мартишин С.А., Симонов В.Л., Храпченко М.В. Базы данных. Практическое применение СУБД SQL и NOSQL типа для проектирования информационных систем: учеб. пособие – М.: ИД «Форум»: ИФРА, 2016. – 368 с.

3. Robomongo [Электронный ресурс] URL: <https://robomongo.org> (дата обращения: 30.03.2019).

4. Расчетно-платежный комплекс «Абонент+» [Электронный ресурс] URL: <http://abonentplus.ru/> (дата обращения: 26.03.2019).

5. Moodle [Электронный ресурс] URL: <https://moodle.org> (дата обращения: 09.04.2019).

6. Система дистанционного обучения и контроля знаний SQLTest [Электронный ресурс] URL: <https://rgry.ru/sqltest/> (дата обращения: 23.03.2019).

УДК 621.37

В.С. АРЧАКОВ

Рязанский государственный радиотехнический университет

ВАРИАНТЫ ПОСТРОЕНИЯ ФАЗОВРАЩАТЕЛЕЙ

Рассматриваются принципы построения фазовращателей с различными способами изменения фазы.

Введение. Фазовращатели представляет собой четырехполюсник с регулируемой фазой коэффициента передачи. В устройствах СВЧ используются управляемые фазовращатели различного типа в зависимости от принципа изменения фазы: ферритовые, механические, сегнетоэлектрические и др. Наибольшее распространение в настоящее время получили дискретные с использованием переключающих полупроводниковых диодов. Они обладают высокими точностью, стабильностью установки фазы, быстродействием, могут иметь малые габариты, вес, а также ряд других преимуществ. При проектировании диодных фазовращателей разрабатывается схема и конструкция устройства с переключающими диодами, которые обеспечивают требуемую дискретность фазовой задержки при минимально вносимых потерях. При проектировании в исходных данных указываются требуемые значения фазы коэффициента передачи, точность установки каждого дискрета фазы, допустимый уровень затухания, характеристики и параметры используемых переключающих диодов, диапазон рабочих температур и др.

Фазовращатель СВЧ - устройство, предназначенное для изменения фазы электромагнитных колебаний на выходе линии передачи СВЧ относительно фазы колебаний на её входе, осуществляемого посредством изменения электрической длины этой линии.

Возможные схемы построения фазовращателей. Фазовращатели подразделяются на плавные и ступенчатые. В плавных фазовращателях фаза изменяется плавно в пределах фмин-фмакс. В ступенчатых - фаза имеет несколько фиксированных значений, которые отличаются на $\Delta\varphi$ (для краткости часто называют дискретом фазы). Для реализации n ступеней фаза принимает ряд фиксированных значений. Важнейшими показателями фазовращателей являются: максимально вносимые потери, значение дискрета, число ступеней фазы, стабильность фазы в рабочем диапазоне частот. На начальном этапе проектирования составляется принципиальная схема фазовращателя. Выбор схемы зависит от числа заданных фазовых состояний и от дискрета фазы. Наиболее простым является фазовращатель с двумя фазовыми состояниями, который может быть реализован, используя четвертьволновые отрезки линий, шунтированных на концах коммутационными диодами, которые подключаются через трансформаторы сопротивлений.

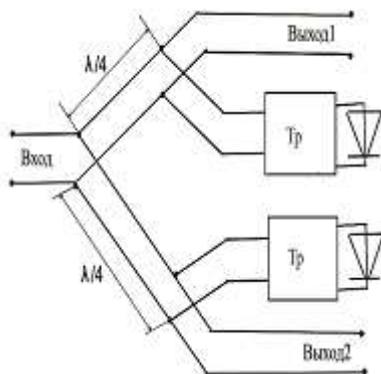


Рис. 1. Схема фазовращателя с двумя фазовыми состояниями.

Если разность фаз в двух состояниях превышает 90° , то потери в таком фазовращателе слишком большие, и тогда используют мостовые схемы. Схему многопозиционного фазовращателя можно составить в виде каскадного соединения нескольких двухпозиционных фазовращателей. Первый каскад представляет собой квадратурный мост и в двух режимах диодов создает фазовые задержки, отличающиеся на 180° . Второй и третий каскады представляют собой двухпозиционные фазовращатели с четвертьволновыми отрезками и предназначены для создания дискретов фазы 90° и 45° .

Наиболее сложными узлами при проектировании фазовращателя являются трансформаторы сопротивлений, поэтому для их проектирования используют численное моделирование на ЭВМ. Наиболее простым является пассивный фазовращатель Шиффмана. Поступающая на вход СВЧ энергия разделяется поровну между двумя плечами, одно из которых представляет собой фазосдвигающий четырехполосник на связанных линиях. Отличительным свойством фазовращателя является постоянство относительного фазового сдвига в 90° в широкой полосе частот. В фазосдвигающем четырехполоснике две связанные линии соединены друг с другом с одной стороны, а входы последовательно подключены к линии передачи данного плеча. Простую конструкцию имеет также ферритовый фазовращатель в печатном исполнении. Основой взаимных и невзаимных фазовращателей такого типа является меандровая линия.

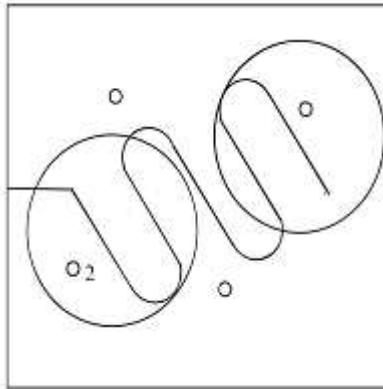


Рис. 2. Меандровая линия взаимных и невзаимных фазовращателей (кривая линия).

Длина одного плеча змейки выбирается равной четверти длины волны в линии передачи, при этом ВЧ-поле имеет область круговой поляризации для построения невзаимного фазовращателя. Расстояние между смежными плечами равно либо меньше толщины подложки, вследствие чего между ними имеет место сильная связь. Через сквозные отверстия в подложке пропускают проводник в виде петли, плоскость которой перпендикулярна плоскости подложки. Через проводник пропускают импульс тока, перемещающийся ферритовую подложку в одно из двух устойчивых состояний.

Во взаимном фазовращателе длина плеча змейки выбирается произвольной, а расстояние между плечами должно быть таким, чтобы связь между ними отсутствовала. В подложке пропущены провода с током, которые намагничивают подложку вдоль или поперек плеча змейки. С изменением направления намагничивания изменяется скорость распространения волны в линии передачи и, таким образом, вносимый фазовый сдвиг. Недостатком меандровых фазовращателей является высокий уровень потерь. Лучшие результаты получаются при использовании щелевых или копланарных линий передачи.

Коммутационные диоды, их назначение и разновидности.

Для переключения мощности, проходящей по СВЧ тракту, используются коммутационные диоды. Они делятся на нерезонансные и резонансные. Эффект переключения достигается за счет изменения сопротивления диода при подаче на него прямого и обратного напряжений. В нерезонансных переключательных устройствах используется свойство, в соответствии с которым при подаче прямого смещения диод обладает малой величиной активного дифференциального сопротивления, а при обратном смещении - значительно большей. Основным требованием, предъявляемым к конструкции такого диода, являются минимальная индуктивность выводов, а также емкость диода и патрона. Емкость диода иногда компенсируют дополнительной индуктивностью. В результате специально принятых мер переключающее устройство будет обладать практически активным сопротивлением в рабочем диапазоне частот.

При подаче управляющего напряжения прямой полярности малое сопротивление перехода шунтирует емкость. В результате образуется параллельный колебательный контур, состоящий из индуктивности и емкости, имеющий на резонансной частоте большое сопротивление. В результате СВЧ сигнал беспрепятственно проходит по тракту. При подаче управляющего сигнала обратной полярности сопротивление перехода резко увеличивается и возникает последовательный резонанс. Как известно, сопротивление последовательного контура очень мало. И, кроме того, он будет дополнительно шунтирован емкостью. Таким образом, изменяя полярность управляющего сигнала на диоде, можно создавать параллельный или последовательный резонанс и коммутировать СВЧ мощность.

Заключение. Анализ схемы построения фазовращателей показал, что пассивные не обеспечивают необходимое число дискретов в заданной полосе частот и имеют большие потери. Наиболее универсальными являются многокаскадные фазовращатели, использующие коммутационные диоды, подключаемые к линии передачи через со-

гласующие устройства. Они могут обеспечить большое число дискретов, однако их проектирование является сложным и требует моделирования с помощью компьютера.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Малорацкий Л.Г. Микро-миниатюризация элементов и устройств СВЧ. М.: Сов. радио, 1976. 172 с.
2. Сигорский В.П., Петренко А.И. Алгоритм анализа электронных схем. М.: Сов. радио, 1976. 97 с.
3. Голубева Н.С., Митрохин В.Н. Основы радиоэлектроники сверхвысоких частот. М., 2008. 368 с.

УДК 621.37

В.С. АРЧАКОВ

Рязанский государственный радиотехнический университет

ФАЗОВРАЩАТЕЛЬ ТРОМБОННОГО ТИПА С ДИЭЛЕКТРИЧЕСКИМИ ПЕРЕХОДАМИ

Описано простое и надежное устройство для согласования подвижного соединения волноводов, уменьшающее паразитное излучение передаваемой СВЧ мощности.

Фазовращатели (фазосдвигатели) находят широкое применение как в измерительной технике, так и при разработке фазированных антенных решёток. В первом случае их работа происходит на низком уровне мощности, во втором случае - на высоком уровне мощности.

В традиционной конструкции фазовращателя при перемещении диэлектрической пластины внутри волновода можно плавно изменять длину волны на участке размещения такой пластины и, тем самым, регулировать электрическую длину линии. Такое устройство хорошо работает на низком уровне мощности. Однако размещение диэлектрической пластины внутри волновода резко снижает его электрическую прочность на высоком уровне мощности. Проблема пробоя в волноводном тракте приобретает особую актуальность в 3-х сантиметровом диапазоне длин волн, когда толщина диэлектрической пластины превышает 10% ширины волновода.

Электрическую длину волновода можно также менять механическим путем. Примером такой конструкции является фазовращатель тромбонного типа.

В статическом состоянии фазовращатель тромбонного типа включает электрический пробой в волноводе. Однако сильные искрения при передаваемой мощности $P > 10$ кВт делают невозможным его применение в установках большой мощности, когда необходимо осуществлять регулировку фазы. Кроме того, в месте контакта волноводов наблюдается также паразитное излучение СВЧ мощности из-за разрыва путей протекания поверхностных токов, направленных вдоль сочленяемых волноводов.

Цель настоящего сообщения - описание фазовращателя со специальным переходом, обеспечивающим хорошее согласование подвижного соединения волноводов различного поперечного сечения, уменьшающим излучение СВЧ мощности и исключаящим искрение в месте контакта волноводов. В качестве примера рассмотрим конструкцию и результаты экспериментальной проверки фазовращателя тромбонного типа.

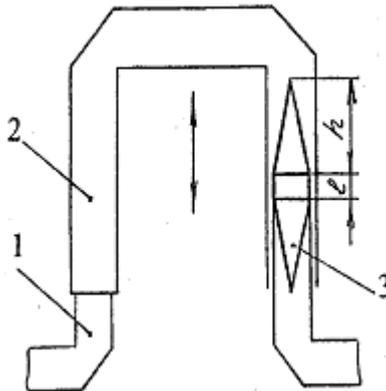


Рис. 1. Схема практического использования диэлектрического перехода, обеспечивающего подвижное соединение волноводов различного поперечного сечения в фазовращателе тромбонного типа: 1) входной и выходной волноводы; 2) и-образное колено; 3) диэлектрический согласующий вкладыш.

Фазовращатель изготовлен из стандартных волноводов трехсантиметрового диапазона длин волн: входной и выходной волноводы - 1 имеют поперечное сечение 23×12 мм², а и-образное колено - 2 имеет сечение 28×12 шЛ.

Вкладыши играют роль диэлектрических антенн, концентрирующих на своих вершинах электрическое поле волны. Они представляют собой две пирамидки, обращенные основаниями друг к другу,

которые соединяются между собой через прямоугольный участок диэлектрика, заполняющего все поперечное сечение волновода.

Для изготовления диэлектрических вкладышей пригодны плотный пенопласт, полиметилметакрилат, полистирол или политетрафторэтилен, то есть материалы, обладающие малой диэлектрической проницаемостью (для уменьшения отражений в волновом тракте) и малыми потерями на сверхвысоких частотах.

Принципиально важными являются результаты проверки предлагаемого устройства на высоком уровне мощности. Эксперимент проводился при импульсной мощности генератора -40 кВт. В качестве нагрузки служили или рупорная антенна, или измеритель СВЧ мощности. При отсутствии вкладышей треск и сильные шумы в тракте фазовращателя значительно влияют на работу магнетронного генератора и не позволяют производить нормальное перемещение волноводов из-за возникающего СВЧ пробоя воздуха. Однако введение согласующих вкладышей позволяет практически исключить пробой в местах сочленения волноводов и на два порядка по сравнению с предыдущим случаем снизить уровень мощности, излучаемой в пространство и подвижного соединения волноводов различного поперечного сечения. При этом величина мощности, поглощенной двумя вкладышами в фазовращателе, не превышает 3 дБ.

Таким образом, предлагаемые диэлектрические переходы обеспечивают надежную работу фазовращателя на высоком уровне мощности. Их применение позволяет получить также хорошее согласование волноводов с различным поперечным сечением и в случае их неподвижного сочленения.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Лебедев И.В. Техника и приборы СВЧ. - М.: Высшая школа, 1970. - 440 с.
2. Протасевич Е.Т. Простые антенны для сверхвысокочастотного зондирования плазмы в вакуумном объеме. Приборы и техника эксперимента. - 1995. - № 6. - С. 90-92.

УДК 004.9

С.О. БАЛАНДИН, А.Н. САПРЫКИН

Рязанский государственный радиотехнический университет

**ОСОБЕННОСТИ ПОСТРОЕНИЯ РЕШЕНИЙ
ГЕНЕТИЧЕСКОГО АЛГОРИТМА В ЗАДАЧЕ РАЗМЕЩЕНИЯ
ЭЛЕМЕНТОВ**

Рассматривается представление решения задачи размещения, дается постановка задачи и предлагается модифицированный генетический алгоритм ее решения, отличающийся ориентацией размещаемых элементов в пространстве.

Размещение конструктивных модулей электронных средств является одной из важных задач конструкторского проектирования. При этом задается множество конструктивных элементов, связанных между собой в соответствии с принципиальной коммутационной схемой узла или блока. Решение задачи размещения состоит в том, чтобы разместить элементы на некотором плоском коммутационном поле так, чтобы выполнялись необходимые ограничения на взаимное расположение элементов, а некоторая целевая функция достигала экстремального значения. Главными критериями задачи является минимальная суммарная длина межсоединений и минимальная площадь области размещенных элементов и межсоединений [1, 2].

Основной целью решения задачи размещения является создание наилучших условий для последующей трассировки. Необходимо найти такой вариант размещения элементов на монтажном пространстве, при котором перекрытие размещенных элементов не допускается, а суммарная длина соединений и площадь, занятая конструкцией, должны быть минимальны.

Для всех размещаемых элементов e_i задаются линейные геометрические размеры (длина a_i и ширина b_i) и список контактов K_i , через которые производится сопряжение текущего элемента с другими элементами схемы. При этом каждый такой контакт имеет координаты (x, y) , которые задаются относительно базовой точки элемента, и номер подключенной к нему цепи v_j . Также каждый элемент в задаче размещения может принимать два положения в пространстве. Пусть, признак ориентации o_i элемента e_i принимает значение «1» при горизонтальном, и «0» – при вертикальном положении.

Математической моделью, описывающей заданную схему, выступает гиперграф $G = (E, V, H)$, где $E = \{e_i \mid i = 1, 2, \dots, n\}$ – множество вершин, соответствующих множеству элементов; $V = \{v_j \mid j = 1, 2, \dots, m\}$ – множество гиперребер, соответствующих множеству цепей; H – инцидентор, заданный на множестве $E \times V$ и определяющий отношение инцидентности элементов и цепей на гиперграфе: $H(e_i, v_j)$ равен «1», если элемент e_i инцидентен цепи v_j , и «0» – в противном случае.

Для того, чтобы найти расстояние между двумя элементами, необходимо сложить расстояния в вертикальной и горизонтальной плоскостях, именно поэтому расстояние между любыми двумя элементами с координатами (x_i, y_i) и (x_j, y_j) определяется по формуле:

$$d_{ij} = |x_i - x_j| + |y_i - y_j|.$$

Оценкой длины цепи v_j будет являться длина минимального связывающего дерева, построенного на множестве вершин $x_j \in X$.

Нужно найти некоторый вариант размещения элементов на монтажном пространстве $r_j = \{(x_1, y_1, o_1), \dots, (x_i, y_i, o_i), \dots, (x_n, y_n, o_n)\}$, где x_i, y_i – координаты базовой точки элемента, а o_i – ориентация элемента e_i . Однако, не каждый найденный вариант размещения будет логически верным, необходимо добавить ограничение, а именно то, что не допускаются перекрытия размещаемых элементов. При этом необходимо добиваться минимизации суммарной длины соединений и площади, занимаемой найденным вариантом размещения элементов.

Задача размещения является многокритериальной, поэтому для оценки качества полученного решения используется аддитивный критерий, в котором отражается сумма нормированных оценок общей длины соединений и площади: $F = w_1 * O(L(r_i)) + w_2 * P(S_{общ}(r_i))$, где r_i – некоторый вариант размещения; w_1 и w_2 – весовые коэффициенты, заданные так, что $\sum w = 1$; $L(r_i)$ – общая (суммарная) длина проводников (межсоединений); $O(L(r_i))$ – нормированная оценка общей длины проводников, приведенная к интервалу $[0, 1]$; $S_{общ}(r_i)$ – площадь размещения, равная площади минимально допустимого прямоугольника, описывающего данный вариант размещения элементов; $P(S_{общ}(r_i))$ – нормированная оценка общей площади размещения, приведенная к интервалу $[0, 1]$.

Таким образом, задача размещения состоит в отыскании минимального значения функции F на множестве допустимых вариантов размещения элементов $R: F(r_i) \xrightarrow{r_i \in R} \min$.

Генетический алгоритм позволяет эффективно учитывать множество входных ограничений, обладая при этом малой временной сложностью [3, 4]. Именно поэтому он является эффективным средством для решения задач размещения элементов большой размерности. Алгоритм возможно применить к задаче размещения, но для этого его необходимо модифицировать, сохранив при этом саму суть.

Решением задачи размещения является совокупность координат размещаемых элементов, а также информации об их ориентации в пространстве. Пытаться закодировать в хромосоме сами координаты элементов в хромосоме нецелесообразно, так как применение генетических операторов по случайному изменению может привести к получению «плохих» решений, в которых отдельные элементы накладываются друг на друга. Для того, чтобы задать размещение элементов, достаточно будет определить порядок следования размещаемых элементов, а уже их непосредственно координаты будут определены при декодировании такой последовательности. Для решения задачи размещения будет использоваться способ «кодирования перестановок», который имеет следующие особенности:

1. линейная временная сложность $O(n)$ декодирования;
2. возможность использовать простые генетические операции с линейной временной сложностью $O(n)$.

Для того чтобы представить некоторый вариант размещения, достаточно использовать две хромосомы – числовую, для представления последовательности элементов размещения, и бинарную (двоичную), для установления признака ориентации элементов в пространстве. Например, для 10 элементов можно получить решение (особь) размещения следующего вида:

$$r_1 = \begin{cases} H_1 & | e_1 e_3 e_5 e_7 e_2 e_4 e_6 e_8 e_9 e_{10} \\ H_2 & | 1 0 0 1 0 1 1 1 1 0 \end{cases}$$

Процесс создания множества возможных решений (популяций) – это циклический процесс, заключающийся в последовательном применении генетических операторов к каждому возможному решению с определенной долей вероятности. Проверка этой вероятности состоит

из генерации случайного числа и сравнения его с пороговым значением вероятности применения каждого генетического оператора.

Предложенная архитектура генетического поиска позволяет эффективно решать задачу размещения элементов большой размерности и получать оптимальные результаты за полиномиальное время.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Курейчик В.М. Математическое обеспечение конструкторского и технологического проектирования с применением САПР: учебник для вузов / В. М. Курейчик. – М.: Радио и связь, 1990. –352 с.

2. Селютин В.А. Автоматизированное проектирование топологии БИС.М.: Радио и связь, 1983. –112 с.

3. Гладков Л.А. Генетические алгоритмы [Текст] / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. – М.: Физматлит, 2010. –317 с.

4. Курейчик В.М. Генетические алгоритмы и их применение. Таганрог: Изд-во ТРТУ, издание второе, дополненное, 2002. –242 с.

УДК 658.562

В.Д. БАРАНОВ

Рязанский государственный радиотехнический университет

ТЕНДЕНЦИИ РАЗВИТИЯ СОВРЕМЕННЫХ СИСТЕМ АВТОМАТИЧЕСКОЙ ОПТИЧЕСКОЙ ИНСПЕКЦИИ (АОИ) ПЕЧАТНЫХ УЗЛОВ

Рассматриваются пути развития и новые решения в современных системах автоматической оптической инспекции (АОИ) как средства контроля качества изготовления электронных устройств.

Большинство производителей электроники при выборе оборудования для изготовления печатных плат руководствуются не только текущими производственными условиями, но и учитывают перспективы дальнейшего развития: увеличение объемов выпускаемых изделий, повышение точности установки компонентов, миниатюризация элементной базы, увеличение плотности монтажа.

Казавшиеся нереализуемыми технологии прошлого сегодня уже выглядят простыми, а зачастую и недостаточными для выпуска конкурентной продукции на рынке. Актуально стоит вопрос о контроле качества изготовления электронных устройств, одним из направлений которого являются системы автоматической оптической инспекции

(АОИ) электронных сборок. Качественные компоненты плюс качественная сборка практически всегда означают хорошее качество печатных узлов. АОИ предоставляет возможность высокопроизводительного и тщательного контроля печатных узлов и обеспечения их качества.

Системы АОИ могут устанавливаться в линию поверхностного монтажа после автомата трафаретной печати и после автоматов установки компонентов, таким образом реализуя метод контроля, направленный на предупреждение возникновения дефектов: недостаток (избыток) пасты или ее смещение на контактных площадках, перемычки, смазывание пасты, отсутствие компонента, неправильная полярность или неправильный номинал компонента, переворот резистора. Система АОИ после печи оплавления уже констатирует наличие тех же дефектов, а также недостаток (избыток) припоя, «компонент на ребре», «эффект могильного камня», приподнятый вывод микросхемы и другие. Таким образом, можно своевременно распознать негативные результаты в технологическом процессе и принимать корректирующие воздействия.

Традиционный подход к инспекции основан на анализе изображений – технологии «золотой платы». Один из недостатков данного подхода – это непостоянство внешнего вида компонента и самой печатной платы, на практике это означает весьма большое количество ложных срабатываний. В последнее время все более востребованными становятся системы 3D-инспекции: двумерных изображений уже становится недостаточно для эффективного выявления дефектов нанесения пасты и монтажа компонентов.

Разработана измерительная технология, в которой получаются точные данные о геометрии изделия в трехмерном пространстве, измеряются размеры компонентов, выводов и других элементов конструкции и сравниваются с допустимыми значениями. Вместо традиционной белой подсветки в виде муаровой сетки используется лазерная технология, которая обладает рядом преимуществ: она позволяет достичь большей скорости выполнения инспекции и проще преодолеть проблему теневого эффекта, особенно остро проявляющую себя при высокой плотности монтажа и наличии на плате высоких компонентов. Скорость работы достигает $65 \text{ см}^2/\text{с}$ при разрешении $14 \times 14 \text{ мкм}$.

В новых системах АОИ применяются два лазерных источника – справа и слева от камеры. При сканировании производится одновременный захват 3D-данных и цветного 2D-изображения, что позволяет получить трехмерное цветное «текстурированное» изображение изделия с высокой точностью и реалистичностью. Таким образом, создание программ занимает значительно меньше времени, а вероятность ошиб-

ки инспекции оказывается намного ниже, чем при традиционном подходе.

3D-технологии для проверки качества сборки электронных узлов становятся общей тенденцией развития. Однако, одна из проблем АОИ – значительное время, затрачиваемое на подготовку и отладку программы проверки для каждого типа изготавливаемого электронного узла. Для ускорения этого процесса разработано программно-аппаратное решение, позволяющее подготовить программу проверки электронного узла быстро и практически без участия оператора. Это решение компании GOPEL, названное One Click Program Preparation, реализуется программным обеспечением PILOT AOI Version 6, в которое включена программа MagicClick, работающая совместно с аппаратным модулем 3D-ViewZ. Данные системы CAD в формате Gerber импортируются, в установку загружается собранный узел, модуль 3D-ViewZ производит сканирование. Программа сама находит и идентифицирует компоненты, определяет их параметры, существенные для оценки качества сборки. Составляется библиотека компонентов. Генерация программы проверки занимает несколько секунд, примерно столько длится само сканирование, и эта программа может применяться уже со второй платы. В качестве образца для сканирования может служить не только «золотая плата», но и обычные серийные изделия. Для измерения высоты компонента применен метод цифровой многочастотной интерференции (многочастотного муара), что позволяет получить точное трехмерное изображение без теневого эффекта. Эти решения кардинально сокращают время подготовки исполнительной программы АОИ, кроме того, с такой системой не требуется высокого уровня подготовки персонала.

В связи со стремительным развитием миниатюризации в электронике одну из ключевых ролей в системах АОИ играют видеокamеры, определяющие и качество получаемого изображения, и скорость инспекции. Чем выше разрешение сенсоров изображения, тем более детально можно контролировать объект. Большая часть систем АОИ оснащена 5-/7-мегапиксельными камерами, однако, существуют и 25-мегапиксельные системы. Высокоскоростная конвейерная система 3D АОИ MV-6e OMNI от компании Mirtek позволяет определить высоту галтели, наклон корпуса, подъем вывода и другие параметры благодаря оснащению 25-мегапиксельной фронтальной камере с телецентрическим объективом. Телецентрическая линза устраняет ошибку, вызванную удаленным расстоянием от компонента, и позволяет получить превосходное изображение без искажений, что снижает число ложных

обнаружений дефектов. Линза камеры обеспечивает разрешение 7,7 мкм.

Используемый в системе интерфейс CoaXPress со скоростью передачи данных 25 Гбит/с позволяет работать с частотой до 120 кадров в секунду, что обеспечивает высокую скорость инспекции. Машина опционально может быть оснащена четырьмя 10-или 18-мегапиксельными камерами бокового обзора. Для 3D-инспекции используется технология многочастного муара: на изделии проецируется несколько сеток с переменным шагом с помощью цифровых проекторов. Это позволяет исключить затененные участки на 3D-модели. Оснащение системы многоуровневой RGB-подсветкой с восьмью вариантами цветовой гаммы дает возможность выявлять трудноразличимые фрагменты изображения. Точность позиционирования сервопривода системы составляет ± 10 мкм.

Таким образом, можно выделить основные тенденции развития современных систем автоматической оптической инспекции печатных узлов: активное применение лазерной технологии, составляющей конкуренцию широко распространенному методу на основе муаровой сетки, рост точности и разрешения, возрастание роли трехмерной инспекции, повышение скорости контроля и упрощение процесса подготовки программ.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Гурылев О.А., Черенькая Л.В. Повышение качества автоматического контроля в сборочном производстве печатных плат // XXII Международной научно-практической конференции. Сборник научных трудов. Санкт-Петербург: Санкт-Петербургский политехнический университет Петра Великого, 2018. с.180-184.

2. Дамико Б., Кожев А. Методы двумерного и трехмерного контроля на одной платформе автоматической оптической инспекции // Технологии в электронной промышленности. Выпуск № 1. 2013. с.52-54.

3. Игнатенкова О.А., Григоров М.С. Концептуальная модель системы оптической автоматической инспекции // VII Международная научно-техническая конференция. Сборник трудов конференции. Белгород: Академия Федеральной службы охраны Российской Федерации, 2018. с.415-417.

4. Кантер А. Контроль нанесения паяльной пасты как инструмент повышения качества выпускаемой продукции // Технологии в электронной промышленности. Выпуск №2. 2014.

5. Кожев А. Новая система контроля MV-6 OMNI Mirtec // Технологии в электронной промышленности. Выпуск №1 (85). 2016. с.48-50.

6. Ниермейер Т., Бойкова Л. Автоматическая оптическая инспекция с применением 3D-технологии // Технологии в электронной промышленности. Выпуск № 6 (74). 2014. с. 60-61.

УДК 004.89

А.В. БЕЛЯКОВ

Национальный исследовательский университет «МИЭТ»

МЕТРИКИ КАЧЕСТВА В ЗАДАЧАХ РАНЖИРОВАНИЯ ИНФОРМАЦИИ

Рассматриваются способы измерения качества результатов в задачах ранжирования документов. Сформулирована решаемая задача и приведены последовательные подходы к её решению.

Задача ранжирования информации возникает в наши дни на каждом шагу и в особенности это проявляется в цифровом пространстве. Выдача поисковых результатов, составление рекомендаций видео и музыки, формирование новостной ленты - данные процессы знакомы сейчас почти каждому, а объём данных, обрабатываемых для этого, растёт день ото дня. В связи с этим ведутся постоянные исследования и разработки в области улучшения алгоритмов ранжирования. Но как объективно оценить, какой метод даёт лучшие результаты наиболее близкие к желаемым? Такой оценкой “близости” являются в данном случае приведённые ниже метрики качества ранжирования.

Для начала сформулируем задачу. Рассмотрим M элементов $r = \{e_i\}_{i=1}^M$. Для элемента множества запросов $u \in U$ результатом работы алгоритма ранжирования будет отображение $r: e \rightarrow R$, ставящее в соответствие каждому элементу e_i его вес $r(e_i) \in [0,1]$, который тем больше, чем больше степень схожести элемента e_i и запроса u . Таким образом после сортировки элементов $r(e_i)$ в порядке убывания получаем перестановку $P: [1, \dots, M] \rightarrow [1, \dots, M]$, которая будет являться входными данными для вычисления значения метрик.

Для оценки качества ранжирования нужно составить эталон $r^{true}(e)$, с которым впоследствии будут сравниваться другие результа-

ты. Обычно для получения такого эталона используют либо исторические данные либо экспертную оценку.

Итак, задача сводится к оценке схожести результатов алгоритма - $r(e)$ и $r^{true}(e)$.

Ниже рассматриваются несколько основных метрик, которые активно используются в настоящем для тестирования работы новых алгоритмов ранжирования.

Самые популярные метрики:

- MAP, MAP@n - Mean average precision;
- NDCG, NDCG@n - Normalized Discounted Cumulative Gain;

Первыми рассмотрим метрики MAP и MAP@n - средней точности нахождения релевантных документов^[1].

Данная метрика применяется в том случае, если $r^{true}(e)$ принимает бинарные значения 0 и 1, т.е. каждый документ полностью либо релевантен либо нет.

Precision at N или точность на N элементах позволяет оценить долю релевантных документов среди первых N элементов ранжированного списка. Допустим, после этапа ранжирования получен отсортированный набор документов $\{r(e)\}_{e \in E}$. Выбрав из них первые N штук, можно посчитать требуемую нам величину по формуле:

$$p@N = \frac{1}{N} \sum_{k=1}^N r^{true}(P'(k)) = \frac{\text{количество_релевантных_документов}}{N}$$

где P' - обратная перестановка, то есть $P'(k)$ - элемент $e \in E$, оказавшийся на k-ом месте после этапа ранжирования.

Average precision at N по сравнению со своим предшественником $p@N$ позволяет учесть место, на котором оказался документ в ранжированном списке. Действительно, гораздо важнее увидеть релевантный документ на 1-ой позиции, чем на N-ой. Формула качества принимает следующий вид:

$$ap@N = \frac{1}{N} \sum_{k=1}^N [r^{true}(P'(k)) * p@k]$$

Mean average precision at N. Данная метрика призвана обобщить результаты для всех возможных запросов $u \in U$ и представляет собой их усреднение. Пусть имеется K возможных запросов. Тогда

$$map@N = \frac{1}{N} \sum_{j=1}^K ap@N_j .$$

Далее рассмотрим метрику $NDCG@n$.

В отличие от предыдущей группы метрик рассмотренные ниже можно использовать и при небинарных значениях релевантности $r^{true}(e)$ [2].

Cumulative gain at N. В простейшем случае суммируем все значения релевантностей документов среди N первых.

$$CG@N = \sum_{k=1}^N r^{true}(P(k))$$

Discounted cumulative gain at N. Модификация позволяет учесть порядок элементов в топе. Делается это домножением релевантности каждого документа на обратный логарифм от номера занятой позиции.

$$DCG@N = \sum_{k=1}^N \frac{2^{r^{true}(P(k))} - 1}{\log_2(k+1)},$$

Использование логарифма в качестве веса можно объяснить тем фактом, что наиболее сильно веса, соответствующие документам, должны отличаться именно в начале списка. Иными словами, гораздо важнее располагается документ на пятой позиции или на десятой, чем если сравнивать 50-ую и 55-ую позиции. Логарифм же позволяет задать именно такой вид зависимости от позиции.

Normalized discounted cumulative gain at N. Нормализовать результаты предыдущей метрики призвана усовершенствованная версия - $NDCG$. Введём такую величину как $IDCG@N$ - максимальное значение метрики $DCG@N$ равно $\sum_{k=1}^K 1/\log_2(k+1)$. Итоговая формула:

$$nDCG@N = \frac{DCG@N}{IDCG@N}.$$

Таким образом, рассмотренные выше метрики, являясь довольно лёгкими для восприятия, позволяют с высокой точностью проанализировать результаты ранжирования документов.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Buckley C., Voorhees E. Evaluating evaluation measure stability. In Proc. of the SIGIR'00, pp. 33-40, 2000.
2. Tie-Yan Liu, Learning to Rank for Information Retrieval. Found. Trends Inf. Retr. 3, 3 (Mar. 2009), 225-331.

УДК 004.89

А.В. БЕЛЯКОВ

Национальный исследовательский университет “МИЭТ”

**ОПРЕДЕЛЕНИЯ СХОДСТВА ИЗОБРАЖЕНИЯ НА ОСНОВЕ
ГЛУБОКОГО ОБУЧЕНИЯ**

Рассматриваются задача определения сходства изображений и применение для её решения нейронных сетей глубокого обучения с использованием набора триплетов - специально подобранных троек тренировочных файлов.

Возможность найти набор похожих изображений по запросу находит множество применений от поиска дубликатов до кластеризации файлов по их содержанию. Например, во всем известных сервисах поиска используются подобные алгоритмы для поиска изображений по образцу.

Сходство изображений можно определять, например, по их принадлежности к той или иной группе. Так две картинки можно считать похожими, если они принадлежат одной категории. Естественно, что каждый файл может принадлежать одновременно к нескольким группам. Однако такой критерий недостаточен как для поиска изображений по образцу, так и для поиска по запросу. Для этих задач нужно уметь выделять незначительные отличия.

Одним из подходов для построения модели схожести изображений может послужить извлечение таких особенностей как фильтры Габора, SIFT^[1] дескрипторы, локальные бинарные паттерны, HOG^[2] и дальнейшее их использование для вычисления степени сходства. Другой путь заключается в использовании глубокого обучения для помощи в идентификации похожих изображений для запроса. Далее более подробно рассматривается модель Deep Ranking^[3], которая обучается определению сходства изображений с помощью набора триплетов.

Триплетами в данном контексте называются специально подобранные группы по три изображения, в каждую из которых входят изображение-запрос, позитивное изображение и негативное изображение. Здесь позитивным названо изображение, которое имеет большую степень сходства с запросом, чем негативное.

С помощью набора триплетов формируются отношения сходства и различия между изображениями. Модели глубокого обучения могут использовать эту детализированную информацию, которая не учитывается в модели разбиения по категориям или в классификационных моделях, для улучшения качества работы.

Так, например, для задач классификации три машины - черная, серая и белая - будут просто машинами. Если же мы используем модель сходства и нам дана на вход в качестве запроса черная машина, то серая машина должна оказаться в ранжированном списке выше белой, но ниже чёрной.

Вся модель Deep Ranking может быть представлена, как функция, ставящая в соответствие каждому изображению точку в евклидовом пространстве. Соответственно два изображения должны быть более похожи, если расстояние между ними в данном пространстве мало. В рассматриваемой модели в качестве расстояния берётся квадрат евклидовой метрики.

Целью каждой задачи обучения является нахождения точки минимума функции потерь. Пусть $f(p)$ - функция, преобразования входного изображения p в точку n -мерного пространства, а $D(f(p_i), (p_j))$ - расстояние между ними в этом пространстве. Относительное сходство внутри тройки p_i, p_i^+, p_i^- (где p_i^+ - позитивное изображение, а p_i^- - негативное) можно выразить следующим образом:

$$D(f(p_i), f(p_i^+)) < D(f(p_i), f(p_i^-)) \quad \forall p_i, p_i^+, p_i^-$$

Таким образом можно составить функцию потерь, которую в дальнейшем нужно будет минимизировать:

$$l(p_i, p_i^+, p_i^-) = \max \{0, g + D(f(p_i), f(p_i^+)) - D(f(p_i), f(p_i^-))\},$$

где g - параметр, регулирующий зазор между схожестью пар (p_i, p_i^+) и (p_i, p_i^-) .

Также необходимо коснуться вопроса формирования самих триплетов. Изначально все тестовые изображения разбиваются на категории, внутри которых каждой паре ставится в соответствие значение $r(p_i, p_j)$ - их сходство. Для каждого запроса p_i из наиболее схожих с ним изображений выбирается экземпляр p_i^+ . Методика же выбора негативных элементов может отличаться. Так в одном подходе в качестве p_i^- может браться изображение из той же категории, но с более низким значением схожести относительно запроса, т.е. $r(p_i, p_i^-) < r(p_i, p_i^-)$. Другой способ предполагает выбор негативного элемента из множества других категорий.

Результаты.

Модель Deep Ranking, спроектированная по данной схеме, показывает отличные результаты (табл. 1) на тестовом наборе ImageNet

ILSVRC-2012, имеющая приблизительно по 1000 изображений в каждой из 1000 категорий.

Ниже приведены результаты сравнения модели с методами на основе извлечения особенностей.

Таблица 1. Результаты тестов.

Metod	Precision	Score-30
Wavelet	62.2%	2735
Color	62.3%	2935
SIFT-like	65.5%	2863
Fisher	67.2%	3064
HOG	68.4%	3099
SPMKtexton1024max	66.5%	3556
L1HashKPCA	76.2%	6356
OASIS	79.2%	6813
Golden Features	80.3%	7165
DeepRanking	85.7%	7004

Как видно из таблицы предложенный метод показывает отличные результаты при решении поставленной задачи.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. David G. Lowe. Object Recognition from Local Scale-Invariant Features. Proc. of the International Conference on Computer Vision, Corfu (Sept. 1999)

2. Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)

3. Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen and Ying Wu. Learning Fine-grained Image Similarity with Deep Ranking. CVPR 2014.

УДК 004.852

В.В. БУДАНОВА

Рязанский государственный радиотехнический университет

ИСПОЛЬЗОВАНИЕ ИНТЕРАКТИВНЫХ ПРОГРАММ ТРЕНАЖЕРОВ В ОБУЧАЮЩЕМ ПРОЦЕССЕ

Рассматриваются способы применения и использования интерактивных программ тренажеров в обучающем процессе.

В современном мире информационные технологии имеют все большее влияние на различные сферы общества. Одной из таких сфер является образование. В условиях постоянного прогресса телекоммуникаций все больше становится заметен процесс развития информационных технологий в сфере образования.

Интерактивные программные тренажеры можно условно разделить на контролирующие и обучающие тестовые программы. Автоматизированные обучающие тренажеры в процессе педагогической деятельности помогают учащимся лучше усваивать материал. Текущее программное обеспечение дает возможность разрабатывать приложения с использованием различных мультимедийных возможностей, таких как изображения, схемы, видео и т.д. В свою очередь контролирующие тренажеры помогают облегчить процесс проверки уровня знаний учащегося, которые он получил в ходе освоения им материала.

Тренажер в общем виде – совокупность, метод симуляции и моделирования, определенные методики, позволяющие человеку принимать наиболее подходящие и быстрые решения. Их использование необходимо, так как они предоставляют возможность получить обучающемуся способности действовать в трудных обстоятельствах, осознавать суть протекающих процессов и их взаимосвязь и умение применять полученные навыки в практической деятельности.

Использование интерактивных программных тренажеров может быть вызвано множеством различных причин: высокой ценой и малым количеством оснащения, которое может быть применено в образовательном процессе, большими расходами на использование реального оборудования, недостаточным количеством времени для обучения персонала работе со сложными техническими комплексами, большими трудозатратами на изменение параметров используемого оборудования, невозможностью модификации текущего программно-технического комплекса.[1]

В настоящее время используется множество терминов, обозначающих автоматизированную обучающую систему, которые, по сути, являются схожими. Наиболее распространенные из них:

- система дистанционного обучения (СДО);
- автоматизированная система дистанционного обучения (АСДО);
- автоматизированная система обучения (АСО);
- обучающая программа;
- обучающая система. [2]

Автоматизированные системы можно разделить по виду связи:

без обратной связи и с обратной связью (см. рис. 1).

Интерактивные автоматизированные тренажеры относятся к замкнутым системам.

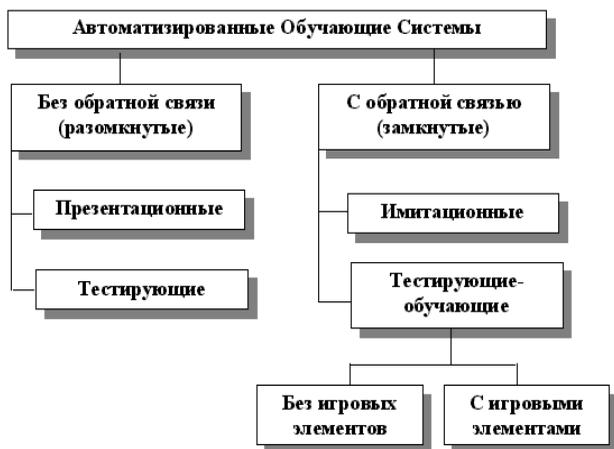


Рис. 1. Классификация обучающих систем.

Интерактивные замкнутые тренажеры могут различаться по сферам применения. Приведем несколько примеров таких тренажеров:

- Имитационные (авиасимуляторы, навигационные тренажеры). Тренажеры, симулирующие для пользователя реальную среду и, имитирующие сложности и тонкости управления реальными системами управления. Например, такие как FlightSimulator от Microsoft, который благодаря реалистичному управлению и физике используется в летных школах.

- Тестирующие-обучающие (иностранные языки, арифметические тренажеры). Наиболее большими функциональными возможностями и высокой эффективностью в учебном процессе обладают системы, где используется обратная связь между учащимся и обучающей системой. [3]

Такие АОС могут быть программами, которые применяют комплексный подход. В зависимости от выбора результата система не только проверяет полученные данные, а также использует игровые ситуации и на основе ответа формирует следующий вариант задания в игровой форме, с помощью гибкого общения с пользователем. Примером таких программ может быть Lingualeo – эффективный сервис для изучения английского языка.

Рассмотренные автоматизированные системы дают возможность выбрать индивидуальный подход к обучению, а также новые варианты и возможности для изучения материала.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Воронин А. А., Горячевская И.В. Автоматизированная система обучения работе с системой контроля версий «GIT» // Радиоэлектроника и информатика, выпуск № 3 (62), 2013. – С.32-35.
2. Мельников А.В., Цытович П.Л. Принципы построения обучающих систем и их классификация, выпуск № 2, 2001. – С.2-4.
3. Лященко Н.И. Анализ моделей компьютерных обучающих систем. Построение подмоделей в компьютерной системе повышения квалификации специалистов // Фундаментальные исследования.– № 10–10, – 2013. – С. 2153-2157.

УДК 004.9

А.С. БУРОБИНА, А.Н. САПРЫКИН

Рязанский государственный радиотехнический университет

БИБЛИОТЕКА ОПЕРАТОРОВ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ РЕШЕНИЯ ЗАДАЧИ ПОИСКА СООТВЕТСТВИЙ НА ДВУХ МНОЖЕСТВАХ

В статье представлена реализация шагов и операторов классического генетического алгоритма и рассмотрено их применение для поиска соответствий на двух множествах.

Генетический алгоритм представляет собой процедуру поиска, базирующуюся на таких механизмах как естественный отбор и наследование. В его основе лежит принцип «выживания» наиболее приспособленной особи в рамках популяции. Генетический алгоритм относится к эвристическим методам поиска.

Поставленную задачу поиска соответствий на двух множествах можно сформулировать следующим образом: пусть даны два множества A и B , элементы которых описаны конечным набором атрибутов, причем каждый атрибут в описании имеет свой ранжированный вес. Ставится задача поиска соответствия между элементами множеств A и B . Дополнительно выдвигается условие о том, что каждому элементу множества A должен соответствовать хотя бы один объект из B .

Для решения поставленной задачи была разработана программа geneticALG.exe, основным элементом которой является библиотека операторов geneticALG.dll. В данной библиотеке реализованы все шаги и необходимые операторы классического генетического алгоритма. Рассмотрим реализацию этапов работы алгоритма. В рамках поставленной задачи популяция состоит из набора генотипов. Генотипы в свою очередь состоят из хромосом. Для моделирования такой структуры в библиотеке разработаны два класса: Chromosome и Genotip.

Класс Chromosome содержит три поля: элемент множества A, список соответствующих ему элементов множества B и значение приспособленности хромосомы, свойства для обработки полей и функцию для расчета приспособленности. Для расчета функции приспособленности используется формула:

$$F_{chrom\ j} = \sum_{i=0}^n W_{pr\ j} \cdot W_{empl\ i} \cdot K_i$$

где

- N – количество элементов, назначенных j -му элементу из A,
- $W_{pr\ j}$ – вес j -го элемента множества B,
- $W_{empl\ i}$ – вес i -го элемента множества A,
- K_i – i -ый множитель

$$K_i = \begin{cases} 1, & \text{если } W_{pr\ j} \leq W_{empl\ i} \\ 0, & \text{в противном случае} \end{cases}$$

Реализация функции расчета приспособленности хромосомы следующая:

```
public void Calculate_fitness_function(int projects_weight,
List<int> employees_weight)
{
    int fitness_function = 0;

    for (int i = 0; i < this.employees.Count; i++)
    {
        if (projects_weight <= employees_weight[employees[i] -
1])
        {
            fitness_function = fitness_function + projects_weight *
employees_weight[employees[i] - 1];
        }
    }

    this.function = fitness_function;
```

}

Класс Genotip хранит список входящих в него хромосом и значение приспособленности, а также свойства для работы с этими полями и функции для получения количества хромосом и расчета оценочной функции генотипа. Функция приспособленности генотипа равна сумме приспособленностей его хромосом.

Операторы и функции генетического алгоритма реализованы в классе geneticALG.

Работа алгоритма начинается с формирования стартовой популяции. Реализация данного этапа выполняется функцией Start_population. На вход подаются два массива, описывающих элементы исходных множеств, ограничение на число объектов В, ставящихся в соответствие элементу из А, а также число генотипов. Данную процедуру удобно реализовать при помощи двух шагов. Чтобы выполнить поставленное условие о необходимости соответствия хотя бы одного элемента множества В объекту множества А на первом шаге случайным образом выбираются элементы из В и ставятся в соответствие случайным объектам А, у которых ещё нет зависимых элементов. На 2 шаге происходит равномерное дораспределение элементов множеств друг с другом, учитывая поставленные ограничения. Результатом данного этапа является текущая стартовая популяция. [1] Цикл для выполнения первого шага формирования популяции выглядит следующим образом:

```

while (pr_chrom.Count < projects.Count)
{
    Project_chromosome chromosome = new Project_chromosome();
    int proj_id = rnd.Next(project_temp.Count);
    chromosome.ID = project_temp.ElementAt(proj_id);
    project_temp.RemoveAt(proj_id);

    int num = rnd.Next(empl_temp.Count);
    chromosome.Add_employees(empl_temp.ElementAt(num));
    empl_temp.RemoveAt(num);

    pr_chrom.Add(chromosome);
}

```

На этапе оценки условия завершения алгоритма проверяется, изменилось ли значение приспособленности лучшего генотипа в популяции в течение k итераций алгоритма, а, также, не достигнут ли задан-

ный предел числа итераций. Оба параметра задаются до начала работы алгоритма.

Если условие завершения алгоритма не выполнилось, то переходят к формированию новой популяции. Для этого сначала выбирают генотипы в родительский пул путем применения метода селекции, а затем формируют потомков из родительских особей путем скрещивания, применением оператора мутации или просто копированием родителей в новую популяцию.

В рассматриваемой библиотеке реализовано три метода селекции: метод рулетки, турнирный метод и ранговый метод.

Селекция методом рулетки выполняется функцией `Selection_roulette`. Эта функция получает на вход массив генотипов и число генотипов в формируемой популяции. Сначала производится расчет величин секторов, основываясь на значениях приспособленности генотипов.

```
for (int i = 0; i < population.Count; i++)
{
    if (i > 0)
    {
        sector.Add(sector.ElementAt(i - 1) + (double) population.ElementAt(i).Fitness_function / (double)function_sum);
    }
    else
    {
        sector.Add((double)population.ElementAt(i).Fitness_function / (double)function_sum);
    }
}
```

Затем производится выбор генотипов в родительский пул по принадлежности к выпавшему «сектору рулетки».

```
for (int i = 0; i < genotip_count; ++i)
{
    double number = rnd.NextDouble();
    int counter = 0;
    foreach (double sect in sector)
    {
        if (number <= sect)
        {
            parent_pool.Add(population.ElementAt(counter));
            break;
        }
    }
}
```

```

    }
    counter++;
  }
}

```

Турнирная селекция представлена функцией `Selection_tournirs`, которая помимо массива генотипов и размера формируемой популяции получает на вход параметр, называемый размер турнира. Этот параметр задает размер группы для сравнения. Группа генотипов формируется случайным образом. В родительский пул попадает элемент с наивысшей оценкой.

```

for (int i = 0; i < genotip_count; ++i)
{
    for (int j = 0; j < group_num; ++j)
    {
        int num = rnd.Next(population.Count);
        group[j] = population.ElementAt(num);
    }

    for (int j = 0; j < group.Length - 1; ++j)
        for (int k = 0; k < group_length - j - 1; ++k)
            {
                if(group[k].Fitness_function < group[k+1].Fitness_function)
                {
                    Genotip temp = group[k];
                    group[k] = group[k + 1];
                    group[k + 1] = temp;
                }
            }

    parent_pool.Add(new Genotip(group[0]));
}

```

Ранговая селекция реализована функцией `Selection_rang`. Функция оперирует массивом генотипов и размером популяции. Ее работа заключается в проставлении рангов в зависимости от значения оценки и выбором генотипов в пул при помощи метода рулетки.

После формирования родительского пула наступает черед использования операторов скрещивания и мутации. Вероятность их применения задается в параметрах алгоритма. Как правило, вероятность скрещивания крайне высока, а вероятность мутации обычно не превышает 5%. Если суммарная вероятность скрещивания и мутации

меньше 100%, то возможно копирование родительской хромосомы в новую популяцию.

Оператор скрещивания реализован функцией Crossengover, которая получает на вход два родительских генотипа и ограничительный параметр. Для поставленной задачи реализовано равномерное скрещивание, позволяющее сохранить исходное конечное множество элементов.

Обычно оператор мутации подразумевает лишь незначительное изменение заданного генотипа. Как правило, его действие состоит в изменении гена с заданной вероятностью. В рамках нашей задачи такая реализация оператора невозможна из-за конечности множества элементов V . Поэтому под мутацией будем понимать формирование нового случайного генотипа, аналогично получению генотипов стартовой популяции.[1]

После окончания работы алгоритма наступает этап выбора наилучшего генотипа. За этот шаг отвечает функция Best_genotip. Данная функция сортирует генотипы в порядке убывания их оценки и в качестве результата работы алгоритма выбирает генотип, стоящий на первой позиции в отсортированном массиве.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Сапрыкин А.Н., Буробина А.С. Разработка операторов классического генетического алгоритма // НИТ-2017. – 2017. – С.163 – 164.

УДК 681.518.3

В.В. ВАСЮКОВ

Рязанский государственный радиотехнический университет

АВТОМАТИЗАЦИЯ УСТАНОВКИ ПРЕДВАРИТЕЛЬНОГО ВСПЕНИВАНИЯ ПОЛИСТИРОЛА

Рассматриваются вопросы автоматизации установки предварительного вспенивания полистирола.

В технологии литья по газифицированным моделям стали применять полистирол. При данном способе отливки модели не удаляются из формы, а газифицируются под действием тепловой энергии металла, заливаемого в форму. Технология позволила повысить точность литья при уменьшении затрат на оборудование и материалы. Для по-

лучения высокого качества материала производится автоматизация установки предварительного вспенивания полистирола.

Модернизация охватывает как конструктивные изменения установки, так и расширение возможностей системы управления.

В процессе разработки автоматизированной системы управления процессом предварительного вспенивания полистирола для изготовления газифицированной модели для литья было проанализировано руководство по эксплуатации и электросхема, подобрано оборудование, разработаны принципиальная и электрическая схемы, написана программа управления на языке релейно-контактных схем (LD) и разработан человеко-машинный интерфейс в виде проекта окон панели оператора.

В ходе работы было рассмотрено несколько вариантов контроллеров и выбран промышленный контроллер DVP-SX2 фирмы DeltaElectronics с модулем расширения (DVP16SP11), обладающий дискретными и аналоговыми входами и выходами, который является основой системы управления рассматриваемой установки.

Сам процесс вспенивания проходит в три этапа при разных скоростях вращения активатора. Для регулировки частоты его вращения выбран преобразователь частоты VFD-C2000 марки DeltaElectronics.

Для ввода-вывода параметров установки, а также для выдачи текстовых сообщений и индикации использована панель оператора DOP-B05S111 указанной фирмы.

Разработан пульт управления установкой, который содержит органы управления. На пульте расположены: панель оператора, переключатель режимов работы установки, аварийный стоп, кнопки управления, элементы индикации, а также поворотный рубильник (см. рис. 1).

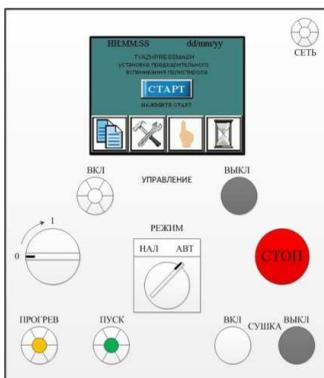


Рис. 1. Пульт управления.

Разработаны принципиальная схема, включающая в себя все элементы установки и связи между ними (см. рис. 2), а также электрическая схема установки (см. рис. 3).

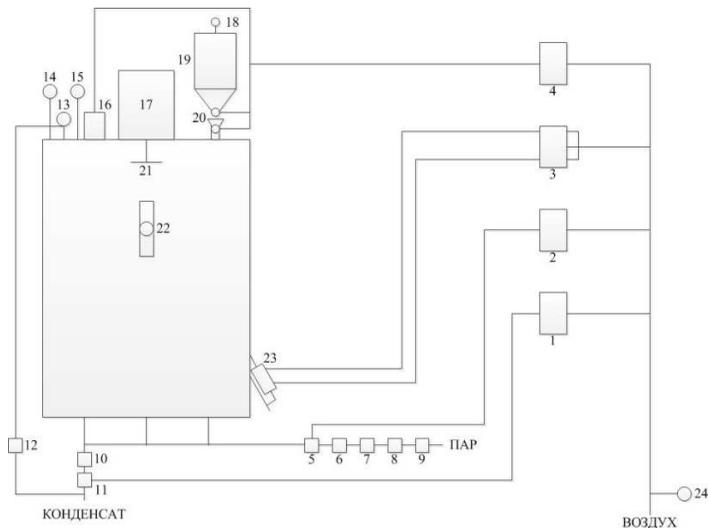


Рис. 2. Схема принципиальная.

Перечень обозначений элементов схемы:

- | | |
|---|---------------------------------|
| 1. Пневмоклапан сброса конденсата; | 13. Манометр электроконтактный; |
| 2. Пневмоклапан подачи пара; | 14. Датчик давления пара; |
| 3. Пневмоклапан открывания люка; | 15. Датчик температуры пара; |
| 4. Пневмоклапан загрузки; | 16. Клапан сброса давления; |
| 5. Отсечной клапан подачи пара; | 17. Мотор-редуктор; |
| 6. Регулировочный кран (задвижка ду25); | 18. Тензодатчик; |
| 7. Обратный клапан; | 19. Бункер весов; |
| 8. Шаровый кран ду25; | 20. Загрузочные клапаны; |
| 9. Регулятор давления; | 21. Активатор; |
| 10. Сетчатый фильтр; | 22. Фотодатчик; |
| 11. Отсечной клапан сброса конденсата; | 23. Механизм открывания люка; |
| 12. Регулировочный кран ду15; | 24. Датчик давления воздуха |

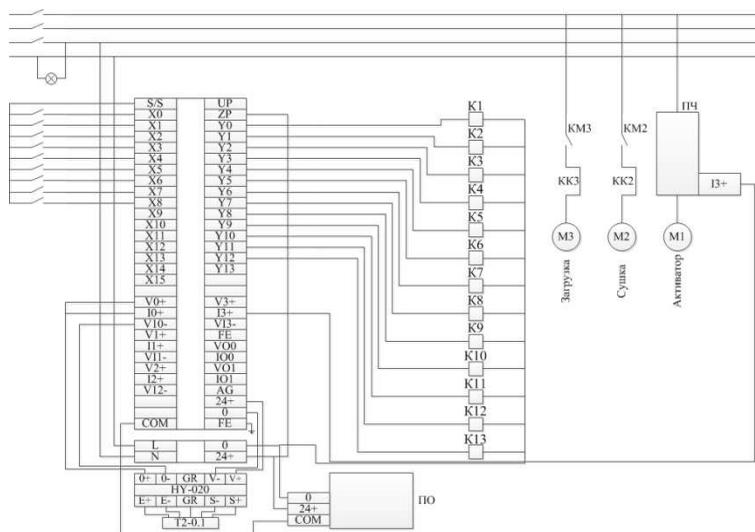


Рис. 3. Схема электрическая.

Перечень буквенных обозначений элементов:

- V – преобразователи, датчики;
- D – интегральная схема, микросборка;
- E – элемент освещения;
- F – защитный автомат, предохранитель;
- G – блок питания, генератор, батарея;
- H – сигнальный элемент, индикатор, светодиод;
- K – контактор, реле, пускатель;
- KK – электротепловое реле;
- L – дроссель, катушка индуктивности;
- M – электродвигатель;
- P – измерительный прибор;
- PE – клемма заземления;
- R – резисторы;
- S – устройства коммутационные в цепях управления и сигнализации;
- SB – выключатель кнопочный;
- T – трансформатор, автотрансформатор;
- V – полупроводниковый прибор;
- X – контактные соединения;
- Y – механические устройства с электромагнитным приводом

Алгоритм написанной программы условно можно разделить на две части. Первая, с неизменной структурой, отображающей работу контроллера в общем случае, представляет собой циклическое выполнение операций чтения входов, выполнения тела программы процессором и запись выходов пока на него подано напряжение питания. Вторая часть - непосредственно тело программы, состоящее из написанных функций и цепочек на языке РКС.

Работа по программированию контроллера DVPSX-2 на языке LD – языке релейно-контактной логики – выполнялась с помощью программного обеспечения фирмы DeltaElectronics для разработки систем автоматизации на основе программируемых логических контроллеров Delta. Программа представляет логические операции как электрическую цепь с замкнутыми и разомкнутыми контактами.

В работе использовалось программное обеспечение фирмы DeltaElectronics для разработки человеко-машинного интерфейса в виде окон панели оператора.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Дельтаэлектроникс www.deltronics.ru/product– (дата обращения: 03.03.2019).
2. Шуляк В.С. Литье по газифицированным моделям. - СПб., Проффессионал, 2007. - 408 с.

УДК 681.511.2

К.М. ВУ

Рязанский государственный радиотехнический университет

НАХОЖДЕНИЕ ФУНДАМЕНТАЛЬНОЙ МАТРИЦЫ В ЛИНЕЙНОЙ СИСТЕМЕ УПРАВЛЕНИЯ

Приведен новый способ построения фундаментальной матрицы без использования Жордановой формы в линейной системе управления и обработки информации.

Рассмотрим систему линейных дифференциальных уравнений в матричной форме:

$$\frac{dx}{dt} = A(t)x, \quad (1)$$

где x – переменный n -мерный вектор фазового состояния, $A(t)$ – матрица размером $n \times n$.

Найдем корни $\lambda_i(t)$ характеристического уравнения $\det(A - \lambda E) = 0$, где E – единичная матрица.

Мы будем искать фундаментальную матрицу в виде [1]:

$$Y = e^{A(t)} = a_0(t)E + a_1(t)A(t) + a_2(t)A^2(t) + \dots + a_{n-1}(t)A^{n-1}(t). \quad (2)$$

Для каждого характеристического числа λ_i , кратность которого равна 1, запишем уравнение:

$$e^{\lambda_i(t)} = a_0(t) + a_1(t)\lambda_i(t) + \dots + a_{n-1}(t)\lambda_i^{n-1}(t). \quad (3)$$

Если кратность какого-либо числа $\lambda_k(t)$ будет больше единицы и равна m , то вместе с уравнением (3) запишем еще $(m-1)$ уравнение, которое получится дифференцированием предыдущего по $\lambda_k(t)$:

$$\begin{aligned} t e^{\lambda_k(t)} &= a_1(t) + a_2(t) 2\lambda_k \dots + a_{n-1}(t)(n-1)\lambda_k^{n-2}(t), \\ t^2 e^{\lambda_k(t)} &= 2a_2 + \dots + a_{n-1}(t)(n-1)(n-2)\lambda_k^{n-3}(t), \end{aligned} \quad (4)$$

...

Из полученной системы найдем коэффициенты $a_0(t), a_1(t), \dots, a_{n-1}(t)$ и сможем построить искомую матрицу (2) для системы (1).

Пример. Найти фундаментальную матрицу следующей системы:

$$x_1(t) = x_1 + 2t x_2,$$

$$x_2(t) = t x_2.$$

Матрица $A(t) = \begin{bmatrix} 1 & 2t \\ 0 & t \end{bmatrix}$, размерность матрицы $n = 2$.

Запишем уравнение $\det(A - \lambda E) = 0$:

$$\begin{vmatrix} 1 - \lambda & 2t \\ 0 & t - \lambda \end{vmatrix} = 0.$$

Отсюда получаем корни: $\lambda_1 = 1$ и $\lambda_2 = t$.

По формулам (3), (4) получаем уравнения для найденных корней с учетом их кратности:

$$\begin{aligned} e^t &= a_0(t) + a_1(t), \\ e^{t^2} &= a_0(t) + a_1(t)t. \end{aligned} \quad (5)$$

Из (5) найдем:

$$a_0(t) = \frac{-(e^{t^2} - te^t)}{t-1},$$

$$a_1(t) = \frac{(e^{t^2} - e^t)}{t-1}.$$

Подставляя $a_0(t)$, $a_1(t)$ в уравнение (2), получим искомую фундаментальную матрицу:

$$Y = a_0(t) E + a_1(t) A(t) = \frac{-(e^{t^2} - te^t)}{t-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{(e^{t^2} - e^t)}{t-1} \begin{bmatrix} 1 & 2t \\ 0 & t \end{bmatrix} =$$

$$= \begin{bmatrix} e^t & \frac{2t(e^{t^2} - e^t)}{t-1} \\ 0 & e^t \end{bmatrix}.$$

В работе предложен новый способ построения фундаментальной матрицы, на основе которого оценки управления и наблюдения объектов в теории управляемости «в малом» осуществляются [2].

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Понтрягин Л.С. Обыкновенные дифференциальные уравнения. – М.:Наука, 1974. – 331 с.
2. Ву, К.М. Проблема наблюдаемости «в малом» для неавтономных линейных систем и ее приложение к анализу движения малого космического аппарата [Текст] / Ву К. М. // Вестник Рязанского государственного радиотехнического университета (РГРТУ). - 2019. - № 1 (выпуск 67). – С. 73-81.

УДК 681.511.2

К.М. ВУ

Рязанский государственный радиотехнический университет

УПРАВЛЯЕМОСТЬ «В МАЛОМ» ДЛЯ ЛИНЕЙНЫХ НЕАВТНОМНЫХ СИСТЕМ ВТОРОГО ПОРЯДКА

Приведен критерий управляемости «в малом» для линейных неавтономных систем второго порядка через классическое свойство управляемости.

Математическая модель плоского управляемого движения объектов представлена и изучается в работе в виде:

$$\dot{x}(t) = A(t)x(t) + v(t)u, \quad (1)$$

где $x(t)$ – 2-мерный вектор фазового состояния, зависящий от времени t ; u – постоянный вектор, заданный координатами u_1, u_2

(коэффициенты усиления управления); $A(t) = \begin{bmatrix} a(t) & b(t) \\ c(t) & d(t) \end{bmatrix}$ – матрицы

соответствующих размерностей с вещественными и непрерывными при $t \in [0, +\infty)$ элементами, где $v(t)$ – кусочно-постоянная функция из области U , называемая допустимым управлением, и удовлетворяющая условию

$$(\forall t) \quad |v(t)| \leq 1. \quad (2)$$

Рассматривается задача об управляемости «в малом» начала координат для системы (1). Определения управляемости «в малом» и неуправляемости в малом приведены в работах [1, 2].

Теорема. Система (1) управляема «в малом» для начала координат на отрезке $[t_0, T]$, тогда и только тогда, когда для любого времени из его отрезка $[t_0, T]$, для координат u_1, u_2 вектора усиления управления $u(u_1, u_2)$ число $k = u_2/u_1$ не является корнем уравнения

$$b(t)z^2 + [a(t) - d(t)]z - c(t) = 0. \quad (3)$$

Доказательство. Необходимость. Пусть система (1) управляема «в малом» для начала координат на отрезке $[t_0, T]$.

Следуя работе [2], система (1) также управляема на отрезке $[t_0, T]$. Следовательно, ранг матрицы управления равен 2

$$\text{rank} [u \ A(t)u] = 2. \quad (4)$$

Из (4) следует

$$\text{rank} \begin{bmatrix} u_1 & a(t)u_1 + b(t)u_2 \\ u_2 & c(t)u_1 + d(t)u_2 \end{bmatrix} = 2.$$

Или

$$c(t)u_1^2 + d(t)u_1u_2 - a(t)u_1u_2 - b(t)u_2^2 \neq 0 \quad \forall t \in [t_0, T]. \quad (5)$$

В данной работе предлагаем, что $u_1 \neq 0$. Разделить (5) на число u_1^2 . Получим

$$c(t) + \frac{d(t) - a(t)}{u_1} u_2 - b(t) \frac{u_2^2}{u_1^2} \neq 0 \quad \forall t \in [t_0, T]. \quad (6)$$

Обозначим число $k = u_2/u_1$ через z . Подставляя k в (6), получим

$$b(t)z^2 + [a(t) - d(t)]z - c(t) \neq 0 \quad \forall t \in [t_0, T].$$

Это означает, что число $k = u_2/u_1$ не является корнем уравнения (3) на отрезке $[t_0, T]$.

Достаточность. Согласно теореме в работе [3], если линейная система управляема на отрезке $[t_0, T]$, то она управляема «в малом» на отрезке $[t_0, T]$. Мы докажем управляемостью на отрезке $[t_0, T]$ системы (1).

Пусть число $k = u_2/u_1$ не является корнем уравнения (3). Необходимо доказать, что система (1) управляема «в малом» на отрезке $[t_0, T]$.

При этом из (5), (6) нетрудно видеть, что

$$\text{rank} \begin{bmatrix} u_1 & a(t)u_1 + b(t)u_2 \\ u_2 & c(t)u_1 + d(t)u_2 \end{bmatrix} = 2 \quad (\forall t) \in [t_0, T].$$

Следовательно

$$\text{rank} [u \ A(t)u] = 2 \quad (\forall t) \in [t_0, T].$$

Это означает, что система (1) управляема $(\forall t) \in [t_0, T]$ или управляема «в малом» на отрезке $[t_0, T]$.

Теорема доказана.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Карасев И.П. О существовании области достижимости // Дифференциальные уравнения. – 1967. – Т.3. – № 12. С. 68-74.
2. Миронова К.В., Корячко В.П. Достижимости цели «в малом» для плоской управляемости движением космического аппарата // Вестник РГРТУ. 2015. № 51. С. 89-95.
3. Ву, К.М. Проблема наблюдаемости «в малом» для неавтономных линейных систем и ее приложение к анализу движения малого космического аппарата [Текст] / Ву К. М. // Вестник Рязанского государственного радиотехнического университета (РГРТУ). – 2019. – № 1 (выпуск 67). – С. 73-81.

УДК 681.511.2

К.М. ВУ

Рязанский государственный радиотехнический университет

**СВЯЗЬ ПРЕДНАЧАЛЬНЫХ И НАЧАЛЬНЫХ УСЛОВИЙ
В ЗАДАЧАХ УПРАВЛЕНИЯ**

Изложен операционный метод решения линейных дифференциальных уравнений при предначальных и начальных условиях, установлена связь между этими условиями.

Пусть техническое устройство реагирует на входное воздействие $x(t)$ и на его выходе образуется пока неизвестный процесс $y(t)$.

$$A(p)y(t) = B(p)x(t), \quad (1)$$

где выражения $A(p)$ и $B(p)$, формально являющиеся соответственно n -го и m -го порядков относительно p , называют линейным дифференцирующими операторами.

Если в дифференциальном уравнении (1) оператор $B(p) \neq const$, т.е. является дифференцирующим, то предначальные и начальные условия не равны друг другу. Физически в момент появления процесса $x(t)$ происходит скачкообразное преобразование предначальных условий в начальные. Установим соотношения позволяющие по предначальным условиям и виду функции $x(t)$ определять начальные и наоборот – по виду функции $x(t)$ и начальным условиям восстанавливать порождающие их предначальные условия [1].

Изображение $Y(s)$ решения дифференциального уравнения при решении уравнения с использованием предначальных и начальных условий должно быть одним и тем же, так как это решение, независимо от способа его поиска, описывает один и тот же объективно существующий физический процесс $y(t)$. Но тогда из работы [1] следует

$$\sum_{i=0}^{n-1} p^i y(0_+) \psi_i(s) - \sum_{k=0}^{m-1} p^k x(0_+) \varphi_k(s) = \sum_{i=0}^{n-1} p^i y(0_-) \psi_i(s). \quad (2)$$

Соотношение (2) выполняется при всех s , если выражения в левой и правой частях этого соотношения, являющиеся коэффициентами в членах с одинаковыми степенями s , будут совпадать. Приравнявая эти выражения, получаем систему уравнений, из которой находят начальные условия в зависимости от предначальных и правых значений процесса $x(t)$ и его производных или наоборот. Запишем эти

уравнения для случая $m = n$ (если $m < n$, соответствующие коэффициенты b_i следует считать равным нулю):

$$\begin{aligned}
 & y(0_+) - b_0 x(0_+) = y(0_-), \\
 & a_1 y(0_+) + p y(0_+) - b_1 x(0_+) - b_0 p x(0_+) = a_1 y(0_-) + p y(0_-), \\
 & a_2 y(0_+) + p y(0_+) + p^2 y(0_+) - b_2 x(0_+) - b_1 p x(0_+) - b_0 p^2 x(0_+) = \\
 & = a_2 y(0_-) + a_1 p y(0_-) + p^2 y(0_-), \\
 & \dots \\
 & a_{n-2} y(0_+) + a_{n-3} p y(0_+) + a_{n-4} p^2 y(0_+) + \dots + p^{n-2} y(0_+) - \quad (3) \\
 & - b_{n-2} x(0_+) - b_{n-3} p x(0_+) - b_{n-4} p^2 x(0_+) - \dots - b_0 p^{n-2} x(0_+) = \\
 & = a_{n-2} y(0_-) + a_{n-3} p y(0_-) + a_{n-4} p^2 y(0_-) + \dots + p^{n-2} y(0_-), \\
 & a_{n-1} y(0_+) + a_{n-2} p y(0_+) + \dots + a_1 p^{n-2} y(0_+) + p^{n-1} y(0_+) - \\
 & - b_{n-1} x(0_+) - b_{n-2} p x(0_+) - \dots - b_1 p^{n-2} x(0_+) - b_0 p^{n-1} x(0_+) = \\
 & = a_{n-3} p^2 y(0_-) + \dots + a_1 p^{n-2} y(0_-) + p^{n-1} y(0_-).
 \end{aligned}$$

Например, для $m = n = 2$ из (3) получаем

$$\begin{aligned}
 & y(0_+) = y(0_-) + b_0 x(0_+), \\
 & p y(0_+) = p y(0_-) + a_1 y(0_-) - a_1 y(0_+) + b_1 x(0_+) + b_0 p x(0_+). \quad (4)
 \end{aligned}$$

Необходимо обратить внимание на следующие. Если $b_0 = b_1 = b_2 = \dots = b_{n-1} = 0$, $b_n \neq 0$, т.е. $B(p) = bn$, и, следовательно, оператор не является дифференцирующим, из (81) следует

$$y(0_+) = y(0_-), p y(0_+) = p y(0_-), \dots, p^{n-1} y(0_+) = p^{n-1} y(0_-).$$

Таким образом, в этом частном случае начальные и предначальные условия совпадают и причин для их противопоставления исчезают. Поэтому при решении дифференциальных уравнений вида $A(p)y(t) = f(t)$ говорят просто о начальных условиях и не конкретизируют их природу. Если же правая часть уравнения имеет вид $B(p)x(t)$, следует совершенно осознанно представлять, при каких условиях решается уравнение при предначальных или начальных. При решении прикладных задач, еще подчеркнем, обычно используются предначальные условия. Если возникает необходимость их по формулам (3) можно представить в начальные условия (например, для решения дифференциального уравнения не операторным методом, а одним из классических, использующих начальные условия) [2]. Из соотношения (3) также видно, что при нулевых предначальных условиях начальные условия в общем случае нулевыми не будут.

Пример. Решить дифференциальное уравнение $y''(t) + 3y'(t) + 2y(t) = 2x(t) + x(t)$ при предначальных условиях $y(0_-) = 1$, $y'(0_-) = 1$ и $x(t) = e^{-3t}$.

Для решения подвергаем обе части уравнения L_- -преобразованию. Для каждого слагаемого получаем

$$L_- \left\{ y''(t) \right\} = s^2 Y(s) - sy(0_-) - y'(0_-),$$

$$L_- \left\{ 3y'(t) \right\} = 3sY(s) - 3y(0_-),$$

$$L_- \left\{ 2y(t) \right\} = 2Y(s),$$

$$L_- \left\{ 2x(t) \right\} = 2sX(s),$$

$$L_- \left\{ x(t) \right\} = X(s) = \frac{1}{s+3}.$$

Следовательно, уравнение в изображениях

$$(s^2 + 3s + 2)Y(s) - (s+3)y(0_-) - y'(0_-) = (2s+1)X(s),$$

и при заданных предначальных условиях и функции $x(t)$ получаем

$Y(s) = \frac{s^2 + 9s + 13}{(s+3)(s^2 + 3s + 2)}$, откуда по формуле Хевисайда

$$y(t) = L^{-1} \left\{ \frac{s^2 + 9s + 13}{(s+3)(s^2 + 3s + 2)} \right\} = -\frac{5}{2}e^{-3t} + e^{-2t} + \frac{5}{2}e^{-t}.$$

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Чураков Е.П. Операторные методы в задачах управления, связи и контроля: Учебное пособие. – Рязань: РРТИ, 1984. – 76 с.
2. Немыцкий В.В., Степанов В.В. качественная теория дифференциальных уравнений. – М.– Л.: Гос. изд. техн.-теоретич. лит., 1947. – 448 с.

УДК 004.65

И.В. ДИННЕР

Рязанский государственный радиотехнический университет

СПОСОБЫ ОПТИМИЗАЦИИ SQL ЗАПРОСОВ*В статье рассмотрены основные способы оптимизации.*

На сегодняшний день почти каждое приложение содержит базу данных. С каждым днем хранение и обработка информации с использованием баз данных увеличивается. На скорость обработки информации влияет очень много факторов, начиная от низкой производительности сети и заканчивая проблемами клиентских ПК. Но также немаловажную роль играет оптимизация запросов к базам данных.

К созданию баз данных разработчику необходимо подходить с учетом большого количества возможных аспектов взаимодействия в процессе работы с ней. Это необходимо для максимально вероятного исключения загрузки сервера. Неумелое создание запросов ведет к понижению производительности системы.

Целью данной статьи является анализ наиболее оптимальных путей выполнения клиентских запросов. Анализируя и перестраивая SQL-запросы, можно снизить время их выполнения в десятки, а иногда и в сотни раз. Итак, после того, как создана структура базы данных, можно проектировать запросы, при помощи которых клиентские приложения будут манипулировать данными на сервере, осуществляя операции выборки, вставки, изменения и удаления данных.

Любую SQL-операцию можно оценить при помощи "коэффициента полезности", он показывает уровень эффективности выполнения операции. Полезность операции прямо пропорциональна баллу данного коэффициента, чем выше балл, тем скорость выполнения SQL-запроса больше [1].

Помимо коэффициента полезности на скорость обработки запроса значительно влияет количество обрабатываемых данных, а так же дополнительные директивы. Коэффициент полезности является удобным способом оптимизации, но не универсальным, в каждой конкретной ситуации необходимо использовать оптимальный способ.

Главным законом при оптимизации запросов выступает закон преобразования. Условие можно изменять как угодно, но результат должен остаться исходным.

Запрос WHERE col1 < col2 AND col2 = col3 AND col1 = 5.

Выполним перестановку:

WHERE 5 < col2 AND col2 = col3 AND col1 = 5.

В итоге результат выполнения запроса одинаковый, а вот продуктивность будет разная, так как применение точного значения значительно влияет на производительность [2].

Оптимизацию запроса также можно проводить, придерживаясь общей идеи создания условий в SQL. Условие всегда должно иметь вид:

<колонка><операция><выражение>.

WHERE col1 - 3 = -col2 должен выглядеть: ...

WHERE col1 = -col2 + 3.

Еще одним основным способом оптимизации запросов выступает оптимизация условных операторов самих запросов SQL:

1. При использовании оператора AND несколько раз необходимо располагать условия в порядке вероятностного возрастания истинности - это позволит повысить производительность за счет отсекаания проверки оставшейся части условия.

2. При использовании оператора OR несколько раз необходимо располагать условия в порядке вероятностного убывания.

3. При расположении одинаковых колонок рядом увеличивается скорость запроса.

4. Необходимо учитывать распределительный закон для сокращения количества операторов. $A \text{ AND } (B \text{ OR } C) = (A \text{ AND } B) \text{ OR } (A \text{ AND } C)$.

5. Для преобразования оператора NOT необходимо использовать правило де Моргана:

$\text{NOT}(A \text{ AND } B) = (\text{NOT } A) \text{ OR } (\text{NOT } B)$

$\text{NOT}(A \text{ OR } B) = (\text{NOT } A) \text{ AND } (\text{NOT } B)$.

6. Операция IN обрабатывает намного быстрее операции OR. BETWEEN обрабатывает намного быстрее операции IN, где применяются последовательные числа необходимо использовать BETWEEN

7. LIKE очень сильно снижает производительность, необходимо использовать ее очень редко.

8. CASE - данная функция повышает производительность запроса, в котором присутствует несколько применений медленной функции в условии.

Следующим способом оптимизации запросов выступает оптимизация сортировки. Сортировка имеет прямо пропорциональную зависимость времени обработки от объема данных. Данная ситуация возникает из-за выделения базой данных в процессе сортировки памяти, учитывая максимальный размер поля участвующего в сортировке. Учитывая данный факт, при установке размерности поля необходимо

указывать минимальное значение, для сокращения выделяемой памяти. Также следует учитывать, что оптимизация сортировки проводится в каждой конкретной ситуации по-разному, оптимального метода в данном случае не существует, необходимо учитывать все параметры сортировки[3].

Оптимизация группировки имеет ряд принципов:

1. При использовании GROUP BY нужно использовать минимальное количество колонок для группировки, избегать лишних условий.

2. WHERE и HAVING не равноценны, скорость их выполнения неодинакова. HAVING используется, когда условие сложно описать при помощи оператора WHERE.

3. DISTINCT обрабатывается быстрее, чем GROUP BY.

4. MIN() и MAX() необходимо использовать по отдельности или используя UNION.

При использовании соединения таблиц (JOINS) очень тяжело определить производительность, так как она зависит от организации таблиц. В данном случае все зависит от архитектуры самой БД и ситуации. Иногда возможно повысить производительность, применяя вложенные циклы в самой программе.

Оптимизация подзапросов является оптимизацией выбора способа осуществления, реализации запросов. Любую задачу с подзапросом можно реализовать при помощи вложенных циклов или соединения таблиц, что дает БД выбрать сам механизм для соединения таблиц. При использовании подзапросов явно применяются вложенные циклы.

В данной работе представлены различные способы оптимизации запросов, приводящие к увеличению производительности SQL-запросов. Но это далеко не все способы. При их применении необходимо учитывать конкретную ситуацию и свойства самой БД. На сегодняшний день каждая база имеет свои встроенные оптимизаторы помогающие провести оптимизацию.

В последнее время процесс оптимизации подзапросов средствами БД значительно улучшился.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Дейт, К. Дж. SQL и реляционная теория. Как грамотно писать код на SQL / К. Дж. Дейт. - М.: Символ-плюс, 2016. - 300 с.

2. Моргунов, Е. П. PostgreSQL. Основы языка SQL: учеб. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова. — СПб.: БХВ-Петербург, 2018. — 336 с.: ил.

3. V2Opt Modular Query Optimizer US Patent 2010/0131490 A1. URL: <https://www.google.com.ar/patents/US20100131490> (дата обращения: 04.04.2019).

УДК 004.65

И.В. ДИННЕР

Рязанский государственный радиотехнический университет

ОПТИМИЗАЦИЯ ЗАПРОСОВ В СИСТЕМАХ БАЗ ДАННЫХ

В статье рассмотрена последовательность выполнения запросов, цель оптимизации

Проблема оптимизации состоит в том, что некоторые системы для достижения определенного уровня производительности требуют оптимизации. Оптимизация позволяет улучшить работу системы, так как одной из сильных сторон реляционного подхода является то, что применение оптимизации к реляционному выражению переводит это выражение на более эффективный семантический уровень.

Точная оптимизация процедур вычисления запросов является вычислительно трудной, и еще больше мешает отсутствие точной статистической информации о базе данных.

Если потребности пользователей ограничены набором стандартных запросов, они могут оптимизироваться вручную путем программирования соответствующих процедур поиска и ограничения пользовательского ввода форматом меню. Однако, если требуется задавать непредвиденные запросы с использованием языка запросов общего назначения, становится необходимой система автоматической оптимизации запросов.

Каждая команда языка манипулирования данными может быть выполнена разными способами. Определение наиболее оптимального плана выполнения запроса называется оптимизацией. Выбором этого плана занимается оптимизатор – специальная компонента СУБД.

Выполнение запроса состоит из последовательности шагов, каждый из которых либо физически извлекает данные из памяти, либо делает подготовительную работу. Последовательность шагов, которую строит оптимизатор, называется планом выполнения.

Обработка запроса, поступившего в систему и представленного на некотором языке запросов, состоит из этапов или фаз, представленных на рис. 1.

На первой фазе запрос, представленный на языке запросов, подвергается лексическому и синтаксическому анализу. При этом вырабатывается его внутреннее представление, отражающее структуру запроса и содержащее информацию, которая характеризует объекты базы данных, упомянутые в запросе (отношения, поля и константы). Информация о хранимых в базе данных объектах выбирается из каталогов базы данных (словаря-справочника данных). Внутреннее представление запроса используется и преобразуется на следующих стадиях обработки запроса.



Рис. 1. Последовательность выполнения запросов в реляционных СУБД

На второй фазе запрос в своем внутреннем представлении подвергается логической оптимизации. При этом могут применяться различные преобразования, "улучшающие" начальное представление запроса. Среди этих преобразований могут быть эквивалентные преобразования, после проведения которых получается внутреннее представление, семантически эквивалентное начальному (например, приведение запроса к некоторой канонической форме).

Третий этап обработки запроса состоит в выборе на основе информации, которой располагает оптимизатор, набора альтернативных процедурных планов выполнения данного запроса в соответствии с его внутренним представлением, полученным на второй фазе. Основой является информация о существующих путях доступа к данным.

На четвертом этапе по внутреннему представлению наиболее оптимального плана выполнения запроса формируется процедурное представление плана.

Наконец, на последнем, пятом этапе обработки запроса происходит его реальное выполнение в соответствии с выполняемым планом запроса. Это либо выполнение соответствующей подпрограммы, либо вызов интерпретатора с передачей ему для интерпретации выполняемого плана.

Исследования оптимизации запросов, представленные в литературе, можно разбить на два класса, которые можно охарактеризовать как восходящий и нисходящий. Исследователи находят общую про-

блему оптимизации запросов очень сложной. Теоретические работы начались с восходящего подхода, изучения особых случаев, таких как оптимальная реализация важных операций и стратегии вычисления для некоторых простых подклассов запросов. В дальнейшем исследователи пытались собрать их этих начальных результатов более крупные строительные блоки.

Потребность в работающих системах инициировала разработку полномасштабных процедур вычисления запросов, что повлияло на общность решений и заставило заниматься оптимизацией запросов в единообразной и эвристической манере. Поскольку часто это не позволяло достичь конкурентоспособной эффективности систем, современной тенденцией представляется нисходящий подход, который обеспечивает возможность включения в общие процедуры большего знания о возможностях оптимизации в частных случаях. В то же время, сами общие алгоритмы усиливаются комбинаторными процедурами минимизации стоимости для выбора между стратегиями.

Применяя общую процедуру вычисления, служащую каркасом для конкретных методов, разработанных при исследовании оптимизации запросов для нисходящего подхода можно выделить следующие шаги:

Шаг 1. Найти внутреннее представление запросов, в которое могут легко отображаться запросы пользователей, оставляющее системе все необходимые степени свободы для оптимизации выполнения запросов.

Шаг 2. Применить логические преобразования к представлению запроса, которые стандартизируют запрос, упрощают запрос, чтобы избежать дублирования усилий, улучшают запрос для упрощения его выполнения и создания возможности применения процедур частных случаев.

Шаг 3. Отобразить преобразованный запросов в возможную последовательность элементарных операций, для которых известна хорошая реализация и соответствующие оценки стоимости. В результате этого шага появляется набор возможных "планов доступа".

Шаг 4. Вычислить общую стоимость каждого плана доступа, выбрать наиболее дешевый план и выполнить его.

Первые два шага этой процедуры являются в большой степени независимыми и поэтому часто могут быть выполнены во время компиляции. Качество шагов 3 и 4, т.е. изобилие генерируемых планов доступа и оптимальность алгоритма выбора сильно зависит от знания значений в базе данных.

Последствия от зависимости от данных являются двойными. Во-первых, если база данных изменчива, то шаги 3 и 4 могут быть выполнены только во время выполнения. Это означает, что возможный выигрыш в эффективности должен соотноситься со стоимостью самой оптимизации. Во-вторых, в метабазе данных (например, расширяемом справочнике данных) должна поддерживаться общая информация о структуре базы данных, равно как и статистическая информация о содержимом базы данных. Как и во многих схожих операционных исследовательских проблемах (например, в управлении запасами) стоимость получения и поддержки этой добавочной информации должна сопоставляться с ее качеством.

Исследования оптимизации запросов продолжают оставаться активной областью. Обещающие направления включают разработку простых, но реалистичных оценок стоимости, оптимизацию запросов с дедуктивными или вычислительными возможностями и одновременную оптимизацию нескольких запросов и обновляющих транзакций.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Дунаев, В. В. Базы данных. Язык SQL для студента / В.В. Дунаев. - М.: БХВ-Петербург, 2017. - 288 с.
2. Маркин, А. В. Построение запросов и программирование на SQL. Учебное пособие / А.В. Маркин. - М.: Диалог-Мифи, 2014. - 384 с

УДК 004.9

Т.А. ДМИТРИЕВА, Д.Ю. БЕЛЯЕВ

Рязанский государственный радиотехнический университет

РАЗРАБОТКА ОБЪЕКТОВ КОНФИГУРАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ОФОРМЛЕНИЯ ОТЧЕТНЫХ ДОКУМЕНТОВ ВУЗА

Рассматривается создание объектов конфигурации на платформе 1С для программного обеспечения, формирующего такой отчетный документ вуза, как рабочая программа дисциплин. Приведен перечень разработанных объектов метаданных и программный код некоторых процедур.

В статье [1] указана актуальность решения проблемы ручного составления различных документов и обоснована необходимость разработки программного обеспечения для составления такого отчетного документа вуза, как рабочая программа дисциплин. Преподаватели

каждый учебный год оформляют рабочую программу, по дисциплинам, которые они преподают. В данный документ входят следующие разделы:

- перечень планируемых результатов обучения;
- объем дисциплины в зачетных единицах с указанием количества;
- академические часов, выделенные на контактную работу обучающихся с преподавателем и на самостоятельную работу обучающихся;
- содержание дисциплины, структурированное по темам (разделам);
- разделы дисциплины и трудоемкость по видам учебных занятий;
- план практических занятий;
- лабораторный практикум;
- курсовое проектирование
- и т.д.

Составление такого документа является очень трудозатратным. Для решения данной проблемы было решено создать конфигурацию на платформе «1С: Предприятие» [1].

Необходимо создать объекты конфигурации для хранения данных, которые требуются для составления отчетного документа, а также объект конфигурации *Документ* для формирования самих рабочих программ.

В результате были созданы следующие объекты конфигурации.

1. **Справочники:** *Дисциплины, Специальности, Сотрудники, Компетенции, Кафедры, Факультеты, Должности.*

2. **Перечисления:** *Квалификация выпускника, Формы обучения.*

3. **Константы:** *Название вуза, Название учреждения, Шапка документа.*

4. **Документ:** *Рабочая программа.*

Дерево объекта метаданных представлено на рис. 1.

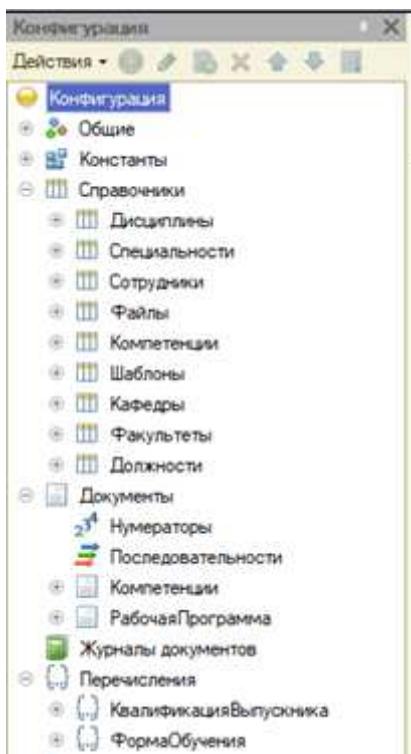


Рис.1. Дерево объектов метаданных.

Составление отчетного документа подразумевает.

1. Создание документа «Рабочая программа».
2. Заполнение реквизитов документов из созданных справочников и перечислений.
3. Формирование печатной формы.

Печатная форма в конфигурации 1С – это готовый шаблон для печати. Печатную форму нельзя редактировать, это является недостатком, так как необходимо реализовать такую возможность, для того чтобы преподаватель смог отредактировать готовый документ, а не менять макет в конфигураторе 1С. Для решения данной задачи можно использовать *MS Word*. Для этого нужно сделать следующее.

1. В документе *MS Word* создать шаблон документа «Рабочая программа», который включает в себя параметры необходимые для заполнения из конфигурации *IC*.

2. В конфигурации *IC* создать справочник *Шаблоны* необходимый для хранения ранее созданного шаблона. Шаблон будет храниться в реквизите справочника *Файл шаблона* с типом данных *Хранилище значения*. Хранилище значений – это тип, позволяющий сохранять прямо в базе различные данные, включая двоичные данные (произвольные файлы).

3. Перевести файл шаблона в двоичные данные и записать их в реквизит с типом *Хранилище значения*.

4. Для заполнения параметров ранее созданного шаблона создать *СОМОбъект* (объект позволяющий устанавливать соединение с внешними приложениями). Пример установки *СОМ*-соединение и вызова процедуры установки параметров показан на рис. 2.

```
Процедура ОткрытьШаблонНаСервере(ИмяФайла)
wordApp = Новый СОМОбъект("Word.Application");

wordApp.Visible = True;
wordApp.Documents.add(ИмяФайла);
wordDoc = wordApp.Application.Documents(1);

УстановкаПараметровWORD(wordApp,wordDoc);

wordDoc.Activate();

КонечПроцедуры
```

Рис. 2. Установка *СОМ*-соединения и вызов процедуры установки параметров.

5. С помощью созданного *СОМ*-объекта заполнить параметры шаблона *MS Word*. Функция заполнения параметров шаблона *MS Word* показана на рис. 3, где *wrдApp* – созданный ранее *СОМ*-объект, *wrдDoc* – *СОМ*-объект шаблона, *ЗаменяемыйПараметр* – параметр шаблона *MS Word*, который необходимо заменить, *ПараметрДляЗамены* – строка для замены параметра шаблона.

```
функция УстановкаПараметра(wrdApp, wrdDoc, Параметр, Замена)
    wrdApp.Selection.Find.Text = Параметр; //строка поиска
    wrdApp.Selection.Find.Replacement.Text = Замена; //строка замены
    wrdApp.Selection.Find.Forward = True;
    wrdApp.Selection.Find.Wrap = 1;
    wrdApp.Selection.Find.Format = False;
    wrdApp.Selection.Find.MatchCase = False;
    wrdApp.Selection.Find.MatchWholeWord = False;
    wrdApp.Selection.Find.MatchWildcards = False;
    wrdApp.Selection.Find.MatchSoundsLike = False;
    wrdApp.Selection.Find.MatchAllWordForms = False;
    wrdApp.Selection.Find.Execute( , , , , , , , , 2);
Конечфункция
```

Рис. 3. Функция заполнения параметров шаблона *MS Word*.

В результате в данной статье был рассмотрен способ создания отчетного документа *MS Word* с заполнением параметров из конфигурации *IC*, с возможностью редактирования. Был использован механизм интеграции *IC* и *MS Word* с помощью *COM*-соединения.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Дмитриева Т.А., Беляев Д.Ю. Разработка программного обеспечения для решения проблемы ручного составления отчетных документов / Математическое и программное обеспечение вычислительных систем. – Рязань: ИП Book Jet, 2019.

УДК 004.422

Т.А. ДМИТРИЕВА, Д.А. ПАЛАГИН

Рязанский государственный радиотехнический университет

РАЗРАБОТКА КЛАССОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ОБУЧАЮЩЕГО АЛГОРИТМИЧЕСКОМУ ЯЗЫКУ

Рассматривается проектирование классов для программного обеспечения, которое будет обучать алгоритмическому языку. Приведены диаграмма разработанных классов и программный код некоторых методов этих классов.

В статье [1] обоснована необходимость разработки интерактивного программного обеспечения обучения алгоритмическому языку программирования. Для разработки данного программного обеспечения необходимо спроектировать классы. Программное обеспечение должно анализировать программу, написанную на алгоритмическом

языке (языке *Pascal*) и автоматически формировать ее схему алгоритма.

При создании схемы алгоритма необходимы объекты для отображения: фигуры и связи (линии). Для этого потребуется два базовых класса *Figure* и *Line* – данные классы хранят базовую информацию для любого наследника. Так же требуется область, на которой будет отображаться схема алгоритма. Для этого можно создать пользовательский класс *DiagramBox*, который, в языке *C#* наследуется от *UserControl*.

Поскольку загруженная программа может быть достаточно большой, нет смысла создавать сразу все объекты фигур. Класс *Diagram* будет хранить в себе только те фигуры и связи, которые необходимо отображать в данный момент. Используя шаблон *Singleton*, исключается возможность создания большого количества объектов класса. Класс *Diagram* используется классом *DiagramBox* для вызова методов отображения у сущностей по средствам обращения к списку объектов.

После того как все классы, связанные с отображением и созданием сущностей созданы, требуется создать классы, необходимые для анализа кода программы.

Язык программирования *Pascal* имеет возможность создания подпрограмм (функций и процедур). Подпрограмма – это фрагмент кода, который имеет свое имя, который создается в случае необходимости выполнять этот код несколько раз. В языке программирования *Pascal*, описывается единожды перед началом основной программы (до ключевого слова *begin*). При составлении схемы алгоритма каждую подпрограмму изображается отдельно. Данная особенность позволяет разбивать схему алгоритма на схему основной программы и отдельные схемы функций и процедур. Для реализации данной возможности необходимо создать два класса:

- класс *DataFuncProc* для хранения кода процедуры или функции, порядковый номер и заголовок (имя);
- класс *CodeProgramm*, в котором хранится весь код программы, список объектов класса *DataFuncProc*, *id* отображаемой функции и метод *CodeSplitting*. Смысл данного метода заключается в следующем: с помощью регулярного выражения, находится заголовок функции или метода; с помощью операторов *begin end* определяется начало и конец функции; создается объект класса *DataFuncProc* с именем процедуры и куском кода. В итоге весь код программы будет расформирован по объектам класса *DataFuncProc*.

В классе *DataFuncProc* хранится уникальный *id* подпрограммы, название метода или функции, и код алгоритма. Данный класс является сущностью и необходим только для хранения информации.

Для анализа текста программы используются регулярные выражения [2], которые предоставляют мощный, гибкий и эффективный способ обработки текста. Комплексная нотация сопоставления шаблонов регулярных выражений позволяет быстро анализировать большие объемы текста для поиска определенных шаблонов символов, проверять текст на соответствие предопределенному шаблону, извлекать, изменять, заменять и удалять текстовые подстроки, а также добавлять извлеченные строки в коллекцию для создания отчета. Для решения рассматриваемой задачи достаточно использовать только проверку строки на вхождение подстроки. В языке программирования *C#* так же реализован метод *IndexOf* для строки, который позволяет найти индекс первого вхождения. Однако, данный инструмент не имеет столь гибких условий поиска, как регулярные выражения.

Как уже было указано ранее, для разбиения всего алгоритма на подпрограммы создается класс *CodeProgramm*. Внутри объекта данного класса хранится код всей программы, веденный пользователем. Основным смыслом класса – разбиение всего алгоритма на подпрограммы. Для этого имеется метод, который по средствам регулярных выражений ищет в каждой строке объявление функции или начало основной программы.

```
public void CodeSplitting()
{
    //регулярное выражение основной программы
    Regex regBegin = new Regex("begin", RegexOptions.IgnoreCase);
    //регулярное выражения функции
    Regex regFunc = new Regex("Function",
        RegexOptions.IgnoreCase);
    //результаты совпадений
    Match match;
    for (; currentStr < code.Length; currentStr++)
    {
        match = regBegin.Match(code[currentStr]);
        if (match.Success)
        {
            CreateProc(currentStr, "Main");
            //метод создания объекта класса DataFuncProc
        }
        match = regFunc.Match(code[currentStr]);
        if (match.Success)
        {
            CreateProc(currentStr, code[currentStr]);
        }
    }
}
```

}

После того, как в коде программы было обнаружено объявление функции или основной программы, в метод *CreateProc* передается номер строки, с которой начинается метод и заголовок. Поскольку основной код программы не имеет заголовка, его можно назвать *Main*.

Метод *CreateProc* ищет операторы *begin* и *end* и определяет уровни вложенности метода. Когда уровень вложенности становится нулевым, это значит, что код метода окончен и можно создавать объект класса *DataFuncProc*. Все созданные объекты хранятся в виде списка.

```
private void CreateProc(int parI, string parName)
{
    List<string> lst = new List<string>();
    int level = 0;
    for (; currentStr < code.Length; currentStr++)
    {
        lst.Add(code[currentStr]);
        if (code[currentStr].IndexOf("begin",
            StringComparison.OrdinalIgnoreCase) > -1)
        {
            level++;
            continue;
        }
        if (code[currentStr].IndexOf("end",
            StringComparison.OrdinalIgnoreCase) > -1)
        {
            level--;
            if (level == 0)
            { break; }
        }
    }
    dataFuncProcs.Add(DataFuncProc.CreateFigure(parName, lst));
}
}
```

После того как весь алгоритм разбит на подпрограммы, можно приступить к анализу конкретной подпрограммы.

Существует 8 базовых блоков для графического представления алгоритма (см. рис. 1).

Базовые блоки.

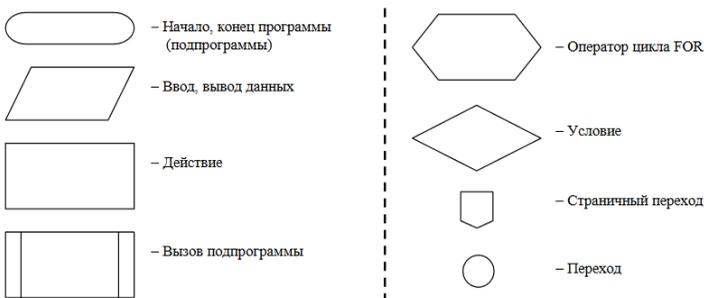


Рис. 1. Базовые блоки схемы алгоритма.

У каждого блока имеется своя индивидуальная структура. Рассмотрим некоторые из них.

1. **Ввод, вывод данных.** Создается при обнаружении операторов *WriteLn*, *Write*, *ReadLn*, *Read*. Уникальная конструкция следующая:

```
<оператор><значение>;
```

Регулярное выражение для оператора записи:

```
".*(read(?:ln).+);"
```

и для оператора чтения:

```
".*(write(?:ln).+);"
```

2. **Оператор цикла с параметром.** Создается при обнаружении конструкции *For* \diamond *to* \diamond *do*. Соответственно, регулярное выражение выглядит следующим образом:

```
for (.+) do
```

3. **Условный оператор.** Создается при обнаружении конструкции *If* \diamond *then*. Регулярное выражение выглядит следующим образом:

```
if(.+)then
```

Подобным образом можно создать регулярное выражение для любого оператора с любым условием. В данном случае приведены примеры, не учитывающие наличие пробелов в начале строки, возможных ошибок конструкции, нестандартных ситуаций.

Используя регулярные выражения для блоков схемы, можно определить базовый блок, который необходимо создать. Ниже приведен пример реализации данного метода *Analyze* класса *Translator*.

```
public static void Analyze(string[] parCode)
{
    foreach (string str in parCode)
    {
        //для каждого регулярного выражения из массива
        for (int i = 0; i < reg.Length; i++)
        {
            Regex regex = new Regex(reg[i],
                RegexOptions.IgnoreCase);
            Match match = regex.Match(str);
            if (match.Success)
            { CreateFigure(i,
                match.Groups[1].Value);
                break;
            }
        }
    }
}
```

В зависимости от номера регулярного выражения, создается определенный объект сущности с названием найденной подстроки входной строки. Все созданные объекты сущностей добавляются в список, для дальнейшей работы с ними.

В итоге получается два списка – список подпрограмм, который хранится в классе *CodeProgramm* и список фигур, который хранится в классе *Diagram*.

Для отображения созданной диаграммы существует пользовательский класс *DiagramBox*, являющийся наследником *UserControl*. Основным методом этого класса является метод *Draw*, который вызывает индивидуальный метод отображения у каждого класса фигуры.

Конечная диаграмма классов выглядит следующим образом (см. рис. 2).

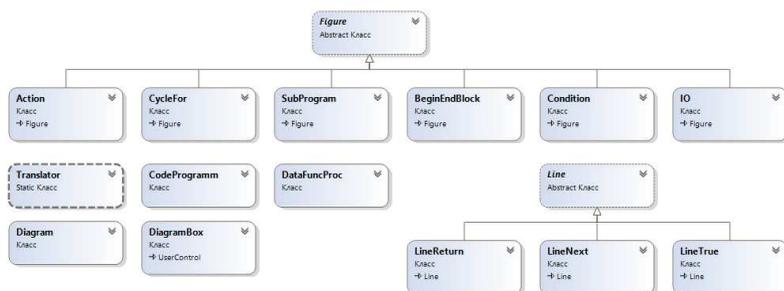


Рис. 2. Диаграмма разработанных классов.

Данные классы позволят реализовать полностью необходимый функционал разрабатываемого программного обеспечения.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Дмитриева Т.А., Палагин Д.А. Разработка интерактивного программного обеспечения обучения алгоритмическому языку / Математическое и программное обеспечение вычислительных систем. – Рязань: ИП Book Jet, 2019.

2. Фридл Дж. Регулярные выражения. – М.: Символ-Плюс, 2008.

УДК 621.396

Р.В. ДУНЦЕВ

Рязанский государственный радиотехнический университет

АЛГОРИТМ ОПТИМИЗАЦИИ ПРОСТРАНСТВЕННОЙ СТРУКТУРЫ РАДИОНАВИГАЦИОННЫХ ОРИЕНТИРОВ ПО ЧАСТНЫМ ПРОИЗВОДНЫМ

Рассматриваются вопросы разработки алгоритма оптимизации на карте местности по частным производным для уменьшения фактора потери точности (PDOP), увеличения точности позиционирования и минимизации СКО.

Задача оптимизации является актуальной в наше время, в основном это касается оптимизации созвездий навигационных спутников, однако использование наземных систем в качестве вспомогательных не утратило актуальность. В данной статье рассматривается разработка алгоритма оптимизации пространственной структуры именно для наземной радионавигации по частным производным. Алгоритм заключается в градиентном перемещении радионавигационных ориентиров (маяков) в сторону уменьшения геометрического фактора потери точности.

Для этого задается матрица направляющихся косинусов для системы из N маяков.

$$\mathbf{H} = \begin{bmatrix} \frac{\Delta x_1}{R_1} & \frac{\Delta y_1}{R_1} \\ \frac{\Delta x_2}{R_2} & \frac{\Delta y_2}{R_2} \\ \vdots & \vdots \\ \frac{\Delta x_n}{R_n} & \frac{\Delta y_n}{R_n} \end{bmatrix} = \begin{bmatrix} \frac{x_1 - x}{R_1} & \frac{y_1 - y}{R_1} \\ \frac{x_2 - x}{R_2} & \frac{y_2 - y}{R_2} \\ \vdots & \vdots \\ \frac{x_n - x}{R_n} & \frac{y_n - y}{R_n} \end{bmatrix},$$

где (x_i, y_i) – координаты маяков, (x, y) – координаты целевой точки,

$R_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} = \sqrt{\Delta x_i^2 + \Delta y_i^2}$ – расстояния от маяков до целевой точки.

Исходными данными для оптимизации являются количество и координаты маяков и целевых точек, в которых будет производиться минимизация геометрического фактора [1].

Геометрический фактор в горизонтальной плоскости имеет следующий вид [2]:

$$PDOP = \sqrt{\delta_x^2 + \delta_y^2}, \quad (1)$$

где δ_x^2 и δ_y^2 – диагональные элементы матрицы Q .

$$Q = (H^T H)^{-1} = \begin{bmatrix} \delta_x^2 & \delta_{xy} \\ \delta_{xy} & \delta_y^2 \end{bmatrix} = \frac{1}{\det(H^T H)} \begin{bmatrix} \sum_{n=1}^N \left(\frac{\Delta x_n^2}{R_n^2} \right) & \sum_{n=1}^N \frac{\Delta x_n \Delta y_n}{R_n^2} \\ \sum_{n=1}^N \frac{\Delta x_n \Delta y_n}{R_n^2} & \sum_{n=1}^N \left(\frac{\Delta y_n^2}{R_n^2} \right) \end{bmatrix} \quad (2)$$

Подставив (1) в (2) получим:

$$PDOP = \frac{\sum_{i=1}^N \frac{\Delta x_n^2 + \Delta y_n^2}{R_n^2}}{\det(\mathbf{H}^T \mathbf{H})} = \frac{N}{\det(\mathbf{H}^T \mathbf{H})} \quad (3)$$

Из (3) следует, что задача минимизации PDOP эквивалентна задаче максимизации $\det(\mathbf{H}^T \mathbf{H})$:

$$\det(\mathbf{H}^T \mathbf{H}) = \sum_{n=1}^N \left(\frac{\Delta x_n^2}{R_n^2} \right) \sum_{n=1}^N \left(\frac{\Delta y_n^2}{R_n^2} \right) - \left(\sum_{n=1}^N \frac{\Delta x_n \Delta y_n}{R_n^2} \right)^2 = \quad (4)$$

$$\sum_{i=1}^{N-1} \sum_{n=i+1}^N \left(\frac{\Delta x_i \Delta y_n - \Delta x_n \Delta y_i}{R_i R_n} \right)^2 = \sum_{i=1}^{N-1} \sum_{n=i+1}^N \frac{D_{in}^2}{R_i R_n}$$

где

$$D = \{D_{in}\} = (\Delta x_i \Delta y_n - \Delta x_n \Delta y_i), \text{ причем } D_{in} = -D_{ni}.$$

Для градиентного алгоритма находим частные производные

по $\frac{d}{d\Delta x_m}(\det(\mathbf{H}^T \mathbf{H}))$ и $\frac{d}{d\Delta y_m}(\det(\mathbf{H}^T \mathbf{H}))$:

$$\frac{d}{d\Delta x_m}(\det(\mathbf{H}^T \mathbf{H})) = \sum_{i=1}^{N-1} \sum_{n=i}^N \frac{d}{d\Delta x_m} \left(\frac{D_{in}^2}{R_i R_n} \right) \quad (5)$$

$$\frac{d}{d\Delta y_m}(\det(\mathbf{H}^T \mathbf{H})) = \sum_{i=1}^{N-1} \sum_{n=i}^N \frac{d}{d\Delta y_m} \left(\frac{D_{in}^2}{R_i R_n} \right)$$

Заметим, что

$$\begin{cases} \frac{d}{d\Delta x_m} \left(\frac{D_{in}^2}{R_i R_n} \right) = 0 \\ \frac{d}{d\Delta x_m} \left(\frac{D_{in}^2}{R_i R_n} \right) = 0 \end{cases} \text{ если } \begin{matrix} m \neq i \\ m \neq n \end{matrix}, \quad (6)$$

Учитывая (6) приводим (5) к следующему виду:

$$\frac{d}{d\Delta x_m} (\det(\mathbf{H}^T \mathbf{H})) = \sum_{n=1}^N \frac{d}{d\Delta x_m} \left(\frac{D_{mn}^2}{R_m R_n} \right), \quad (7)$$

$$\frac{d}{d\Delta y_m} (\det(\mathbf{H}^T \mathbf{H})) = \sum_{n=1}^N \frac{d}{d\Delta y_m} \left(\frac{D_{mn}^2}{R_m R_n} \right).$$

Вид отдельных слагаемых частных производных по $\frac{d}{d\Delta x_m} (\det(\mathbf{H}^T \mathbf{H}))$

и $\frac{d}{d\Delta y_m} (\det(\mathbf{H}^T \mathbf{H}))$ выглядит следующим образом:

$$\frac{d}{d\Delta x_m} \left(\frac{D_{mn}^2}{R_m R_n} \right) = \frac{2D_{mn}}{R_m R_n} \left[\Delta y_n - \frac{\Delta x_m D_{mn}}{R_m} \right], \quad (8)$$

$$\frac{d}{d\Delta y_m} \left(\frac{D_{mn}^2}{R_m R_n} \right) = -\frac{2D_{mn}}{R_m R_n} \left[\Delta x_n + \frac{\Delta y_m D_{mn}}{R_m} \right].$$

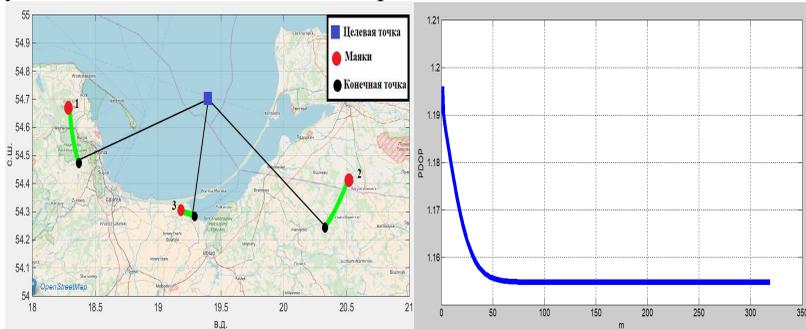
Общий вид градиентного алгоритма включает пошаговый поиск оптимума согласно следующим выражениям:

$$\begin{aligned} x_i^{(t+1)} &= x_i^{(t)} + step \times \frac{d}{d\Delta x_i} \det((\mathbf{H}^T \mathbf{H})) \\ y_i^{(t+1)} &= y_i^{(t)} + step \times \frac{d}{d\Delta y_i} \det((\mathbf{H}^T \mathbf{H})) \end{aligned} \quad (9)$$

где $step$ – шаг градиентного алгоритма, автоматически уменьшаемый каждый раз, когда на очередном шаге PDOP возрастает, что сигнализирует о прохождении оптимума. Условия остановки определяются как уменьшение величины $step$ ниже порогового значения, равного требуемой точности оптимизации.

После чего полученные данные были использованы для построения следующих зависимостей в прикладной среде Matlab. На (см. рис.1,а), показан процесс оптимизации на примере системы из трех маяков и одной целевой точки, а также соответствующий график изменения геометрического фактора потери точности в процессе оптимизации (см. рис.1,б).

Как видно из (см. рис.1,а), маяки тяготеют к прямым пересекающимся в целевой точке, образуя при этом равноугольные направления из целевой точки на маяки, что открывает дополнительные возможности оптимизации путем вращения системы маяков вокруг целевой точки на требуемый угол без потери точности позиционирования. В свою очередь (см. рис.1,б), имеет монотонно убывающий характер, что свидетельствует об уменьшении значения PDOP с 1.196 до 1.155 и увеличении точности позиционирования.



а)

б)

Рис.1. Процесс оптимизации (а), изменение геометрического фактора потери точности в процессе оптимизации (б).

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Дунцев. Р.В. Алгоритм оптимизации геометрии радионавигационных ориентиров на местности. // Новые информационные технологии в научных исследованиях: материалы XXIII Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. Рязанский государственный радиотехнический университет. Том 1. 2018 – С. 135-137.

2. Richard B. Landley. Dilution of Precision // University of New Brunswick, 1999. p. 56.

УДК 621.3.049.75

И.С. ЕМЕЛЬЯНОВ

Рязанский государственный радиотехнический университет

**ИССЛЕДОВАНИЕ ВЛИЯНИЯ ВОЗДЕЙСТВИЯ РАЗЛИЧНЫХ
ФАКТОРОВ НА УЗЛЫ ГИБКО-ЖЕСТКИХ ПЕЧАТНЫХ ПЛАТ
(ГЖПП)**

Рассматриваются основные аспекты и особенности, которые должен принять во внимание конструктор при разработке гибко-жестких печатных плат.

Гибкие печатные платы (ПП) – это системы гибких шлейфов, содержащих одно-, двух и многослойные структуры межсоединений [1]. Конструкция может представлять собой как полностью гибкую, так и комбинацию жестких и гибких частей. По сравнению с жестким монтажом, печатные платы, выполненные из полиимидов, акрилатов, полиэфиров гораздо лучше экономически, так как они обеспечивают:

- более высокую производительность при производстве;
- уменьшение веса и объема изделия;
- увеличение свободы и возможностей для конструктора;
- повышение надежности конечного изделия благодаря простоте безошибочного монтажа.

При необходимости объединить несколько жестких плат обратим наше внимание на гибко-жесткие печатные платы (ГЖПП) состоящие из жестких и гибких элементов и изготавливаемых как единое изделие в производственном цикле. Нам будут интересовать ГЖПП имеющие вид нескольких жестких плат, соединенных гибкими шлейфами. Использование гибко-жестких печатных плат даёт ряд преимуществ в изготовлении, а именно упрощает производство и увеличивает надежность за счет избавления от проводного монтажа, в некоторых случаях может удешевить конечную стоимость изделия.

Рассмотрим основные аспекты, которые должен принять во внимание конструктор при разработке гибких и гибко-жестких печатных плат.

Немаловажная часть при разработке гибких плат – это проектирование контура платы. Радиус во внутренних углах контура платы должен быть не менее 1,6 мм, но при использовании большего радиуса мы сможем значительно повысить надежность и получить лучшую устойчивость к разрыву. Также для повышения устойчивости к разрыву можно добавить специальные материалы во внутренние углы, например, дополнительно закрепить шлейфы, где не нужна их гибкость.

Необходимо предусмотреть, чтобы любой вырез заканчивался отверстием диаметром 1,5 мм или более.

Необходимо чтобы расстояние от края платы до края неметаллизированного отверстия было не менее 0,5 мм. Расстояние между переходом гибкой и жесткой платы до края металлизированного отверстия должно быть не менее 1,9 мм.

У гибко-жестких печатных платы должна быть одинаковая толщина как гибких, так и жестких частей, для обеспечения корректной металлизации отверстий. Разница в толщине жестких частей может привести не только к повышению стоимости изделия и увеличению сроков изготовления, но и к затруднению в реализации проекта в целом.

При разработке ГЖПП, условия к жесткой части приближены к условиям для обычных печатных плат, однако имеется ряд особенностей. При комбинировании материалов гибких и жестких плат может понадобиться специальная конструкция необходимая для выполнения поверхностного монтажа. Во время обработки может добавляться специальный диэлектрический материал для закрытия гибких элементов. После обработки данный материал подлежит удалению с гибких частей платы.

При проектировании гибкой части перед разработчиком встает вопрос использования многослойной структуры, так как стоимость материала гибких плат в 2-3 раза выше, чем для обычных плат. Также при увеличении количества слоев теряется гибкость и увеличивается сложность и стоимость изготовления ПП. Еще одним фактором являются механические требования к печатной плате. Таким образом, может возникнуть потребность в необходимом увеличении толщины защитной пленки вне зависимости от толщины адгезива. Все это ставит разработчика перед выбором многослойной структуры, так как необходимо обойтись минимальным количеством слоев.

Чтобы обеспечить максимальное время жизни и максимальную надежность гибкой части платы проводники должны соответствовать следующим условиям:

- проводники гибкой части должны быть строго перпендикулярно к направлению изгиба и равномерно распределены по его области;
- количество слоев гибкой части должно быть минимальным;
- постоянная ширина;
- не допускается использование металлизированных отверстий;
- максимальная ширина области изгиба.

Многослойные гибкие печатные платы (четыре слоя и более) не обладают большой гибкостью. Чтобы обеспечить более высокую гиб-

кость многослойных плат можно не склеивать определенные части шлейфа. Также иногда применяется техника «книжного переплета» - послойное увеличение длины гибкой части (см. рис. 1).

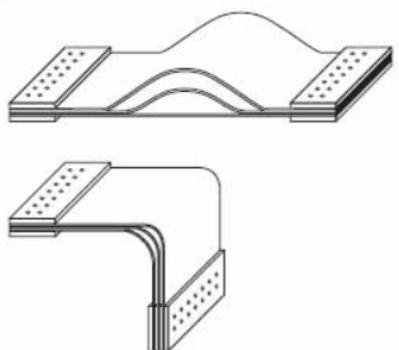


Рис. 1. Конструкция «книжный переплет».

Край жесткой платы, находящийся рядом с гибкой, должен быть скошен фаской или радиусом, либо смягчен нанесением дополнительного утолщения. Контактные площадки не должны подвергаться изгибным нагрузкам.

Так как ГЖПП широко используются в военной и авиакосмической промышленности, в качестве покрытия защищающего узлы ГЖПП можно использовать поли-пара-ксилилен (парилен), так как он обладает низкими показателями влаго- и газопроницаемости, имеет хорошую износостойкость, обладает биологической инертностью, а также стойкостью к воздействию агрессивных химических сред. Главным преимуществом парилена является способность проникать в глубь щелей микронных размеров, полностью обволакивая незащищенные части изделия.

На основе вышесказанного можно сделать вывод, что процесс изготовления гибко-жестких печатных плат значительно отличается от жестких многослойных плат. В первую очередь это использование материалов на полиимидной основе, которые отличаются более высокой температурой прессования и усложнённой процедурой очистки отверстий. Но, не смотря на это, ГЖПП предлагают много жизнеспособных перспективных решений и по мере развития электроники область их использования будет только увеличиваться.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Шейкин М.С. Электроника: наука, технология, бизнес/ М.С. Шейкин // Гибкие печатные платы: теория и практика. – 2014. – №5. – С. 200-207.

УДК 004.383

Ю.А. ЗАХАРКИН

Рязанский государственный радиотехнический университет

**РЕАЛИЗАЦИЯ МОДУЛЯ БЫСТРОГО ПРЕОБРАЗОВАНИЯ
ФУРЬЕ НА ПЛИС В ФОРМАТЕ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ
С ФИКСИРОВАННОЙ ТОЧКОЙ**

Проанализированы аппаратные затраты ресурсов ПЛИС Altera семейства Cyclone-II для реализации модуля 4-точечного быстрого преобразования Фурье в форматах представления дробных чисел по стандарту IEEE 754 и Fixed Point 18.

Преимуществом ПЛИС как аппаратной платформы для реализации алгоритмов цифровой обработки сигналов (ЦОС) в отличие от цифровых сигнальных процессоров является возможность выполнения параллельных вычислений. Например, в задаче обнаружения целей применение ПЛИС оправдано в радиолокационных станциях (РЛС) с импульсно-пачечными сигналами [1]. В данных РЛС временная дискретизация входного потока данных приводит к разбиению развертки дальности на элементарные участки, называемые кольцами дальности, шириной $\delta R = 0,5ct_d$, где $c \approx 3 \cdot 10^8$ м/с – скорость света; t_d – интервал дискретизации. Обработка сигнала в каждом кольце дальности является однородной, что позволяет реализовать на ПЛИС параллельную работу сразу нескольких обнаружителей.

Реализация когерентной обработки радиолокационного сигнала выполняется в цифровом виде. С выхода цифрового согласованного фильтра каждого кольца дальности с периодом повторения импульсов пачки T_n на обнаружитель поступает последовательность N цифровых отсчётов комплексных огибающих $U_i = u_{ci} + ju_{si}$, где $i = 0 \dots N-1$ – номер дискретного отсчёта, j – мнимая единица. Алгоритм обнаружения основан на вычислении дискретного преобразования Фурье (ДПФ) взвешенных отсчётов:

$$X_k = \sum_{i=0}^{N-1} g_i U_i W_N^{ik} \quad (1)$$

где g_i – весовые коэффициенты огибающей пачки, определяемые, как правило, формой диаграммы направленности антенны РЛС; $W_N^{ik} = e^{-j2\pi ik/N}$ – поворачивающий множитель, $k = 0 \dots N-1$ – номер спектрального отсчёта, и сравнении модулей комплексных спектральных отсчетов с порогом. Взвешенные N отсчётов $g_i U_i$ для каждого кольца дальности поочередно поступают в процессор быстрого преоб-

разования Фурье (БПФ). Основная идея БПФ заключается в том, что полное вычисление ДПФ для N -точечной последовательности алгоритм БПФ разделяет на комбинацию 2-точечных ДПФ. Каждое 2-точечное ДПФ реализует базовую операцию, называемую "бабочкой"; при этом минимизируется количество умножений [2]. Как следует из (1), весовые множители W_N^{ik} являются дробными.

Одним из популярных форматов представления чисел с плавающей точкой является стандарт IEEE 754 (далее – FP32) [3]:

$$x = (-1)^S \cdot (m \cdot 2^{-23}) \cdot 2^{exp-128}, \quad (2)$$

где S – бит знака, m – мантисса (23 бита), exp – показатель степени (8 бит).

Поскольку функциональные схемы ПЛИС типа FPGA предполагают наличие уже встроенных аппаратных умножителей 9-разрядных операндов [4], то для реализации алгоритмов ЦОС (в том числе и БПФ) на них целесообразно применять формат, где мантисса содержит не более 9 бит. При этом ресурсы комбинационной логики ПЛИС на реализацию комбинационной схемы "умножитель" не расходуются. К таким форматам можно отнести формат представления числа Fixed Point 18 (далее Fix18) [5], согласно которому

$$x = (-1)^S \cdot (m \cdot 2^{-9}) \cdot 2^{exp-8}, \quad (3)$$

где мантисса m содержит 9 бит, а показатель степени exp – 8 бит;

Недостатком использования Fix18 является меньшая на 4 порядка точность представления чисел: машинный эpsilon для FP32 – $\epsilon_{FP32} = 2^{-23} \approx 1,19 \cdot 10^{-7}$, для Fix18 – $\epsilon_{Fix18} = 2^{-9} \approx 1,9 \cdot 10^{-3}$.

В САПР Quartus II автором реализованы проекты модулей 4-точечного БПФ при представлении чисел как в формате стандарта IEEE 754, так и в формате Fix18. Для формата Fix18 средствами языка описания аппаратуры VHDL разработаны компоненты умножителя и сумматора, для формата IEEE754 – подключены созданные разработчиками Altera модули с помощью инструмента *MegaWizard Plug-In Manager*. Каждая "бабочка" БПФ за один такт производит вычисление сразу для двух отсчетов и требует для своей реализации 4-х умножителей и 6-и сумматоров комплексных чисел.

На рис. 1 и 2 приведены результаты сопоставительного анализа затрат ресурсов кристалла ПЛИС Altera Cyclone II EP2C70F89618 на реализацию 4-точечного БПФ для форматов чисел FP32 и Fix18.

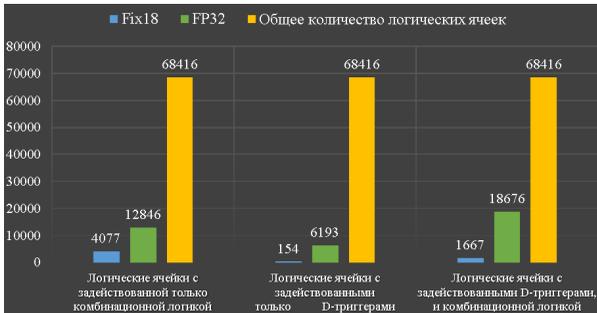


Рис. 1. Затраты ресурсов ПЛИС EPC70F89618 на реализацию 4-хточечного БПФ для форматов представления чисел FP32 и Fix18.



Рис. 2. Затраты 9-разрядных умножителей.

Таким образом, реализация БПФ в формате представления чисел Fix18 требует на 80 % меньше аппаратных ресурсов ПЛИС по сравнению с реализацией БПФ в формате представления чисел FP32.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Бакулев П.А. Радиолокационные системы. – М.: Радиотехника, 2004. – 320 с.
2. Рабинер Л.Р., Шафер Р.В. Теория и применение цифровой обработки сигналов. – М.: Радио и связь, 1981. – 835 с.
3. 754-2008 – IEEE Standard for Floating-Point Arithmetic. Revision of ANSI/IEEE Std 754-1985.
4. Поляков А.К. Языки VHDL и VERILOG в проектировании цифровой аппаратуры. – М.: СОЛОН-Пресс, 2003. – 320 с.
5. Simplified Floating Point for DSP Cyclone 2 and Cyclone 4 version [Электронный ресурс]. Режим доступа: <https://people.ece.cornell.edu/land/courses/ece5760/FloatingPoint/index.html>.

УДК 681.7.068

Н.В. ЗАХАРКИН, В.П. ФЕДОРОВ

Рязанский государственный радиотехнический университет

**СИСТЕМА ОПТИЧЕСКОЙ ПЕРЕДАЧИ ИНФОРМАЦИИ С
ПОМОЩЬЮ ЛИНЕЙНО-ПОЛЯРИЗОВАННЫХ
ЭЛЕКТРОМАГНИТНЫХ ВОЛН**

Рассматривается реализация идеи штрих-кодов в системе оптической передачи с помощью линейно-поляризованных электромагнитных волн.

Главным критерием реализации системы оптической передачи информации является надежная передача данных для правильного ее распознавания. Также к основным преимуществам любых оптических систем будет относиться увеличение скорости передачи сообщения.

Можно представить способ кодирования информации в двумерных штрих-кодах, при котором изображение штрих-кода формируют с помощью цветных штрихов цветовой модели CMYK (С-голубой, М-пурпурный, Y-желтый, К-черный), отличающийся тем, что с целью увеличения количества слов текста, записываемого с помощью одного и того же количества штрихов-модулей кода, и надежности его считывания слова и фразы естественного языка представляют четырехрядными двенадцатеричными числами, а каждую цифру числа представляют одним-двумя штрихами-модулями, причем начало каждой цифровой позиции отмечают черным штрихом. Идея иллюстрирована на рис. 1.

Данная концепция кодирования может быть пригодна в системах оптической передачи. При этом привлекательное на первый взгляд решение использовать по аналогии световые посылки разных цветов не применимо в случаях, когда неодинаковая скорость различных длин волн будет препятствовать повышению скорости передачи.

Между тем известна другая технология придания световым импульсам одной и той же длительности разных состояний. Речь – о такой характеристике волн, как их поляризация. Известен способ передачи информации с помощью линейно-поляризованных электромагнитных волн и система для его осуществления[1].

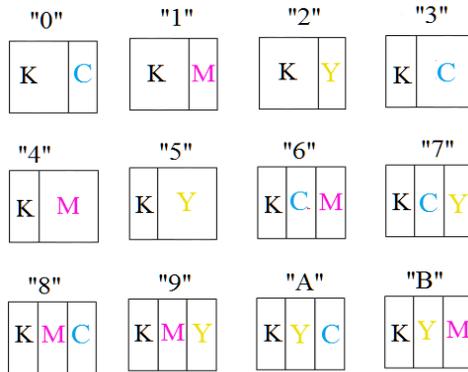


Рис. 1. Двумерный штрих-код цветовой модели СМУК.

В известном способе [1] импульсные сигналы линейно-поляризованы. Вариантом изображения линейной поляризации может быть в дискретных фиксированных положениях E_1, E_2, E_3, E_4 (рис. 2), которые выбираются в диапазоне распознавания. Угловые промежутки между соседними положениями вектора E , могут быть установлены одинаково. Вектору E_1 соответствует значение передаваемой (принимаемой) информации «00», вектору E_2 —«01», вектору E_3 —«10», вектору E_4 —«11» в системе двоичной кодировки.

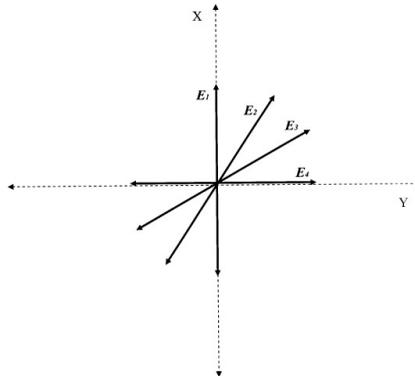


Рис. 2. Положение вектора E напряженности электрического поля.

В случае использования штрих-кода двумерной цветовой модели СМУК вектору E_1 будет соответствовать передаваемая (принимаемая) информация обозначенная цветом «К», вектору E_2 —«Y», вектору E_3 —«М», вектору E_4 —«С». Пример представления цифр двенадцатеричной системы комбинациями поляризованных волн показано на рис.3. Со-

ответствие комбинаций поляризованных волн цифрам 12-ричной системы на рис. 3 дано в том же порядке, что и комбинации штрихов на рис. 1.

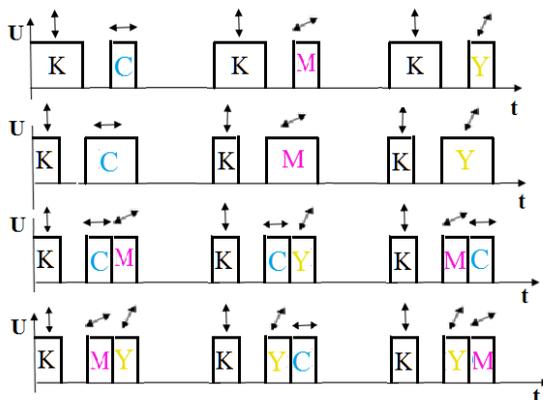


Рис. 3. Представление цифр двенадцатеричной системы комбинациями поляризуемых волн.

Как видно из рис. 3, каждая цифра передается за четыре такта. Между сериями таких тактов, соответствующих двум соседним цифрам, должно быть минимум два такта отсутствия сигнала, что исключит ошибку в восприятии начала цифры. С учетом ещё одной свободной от сигнала позиции между словами передача одного слова, которое представляется четырехразрядным двенадцатеричным числом, осуществляется за 25 тактов.

При традиционной передаче слов последовательностью кодируемых букв тактов на передачу слова потребуется несколько больше. Если среднюю длину слова положить в 5,5 букв, а каждую букву кодировать восемь битами, то тактов должно быть примерно 44. То есть при одной и той же тактовой частоте скорость передачи предлагаемым способом будет примерно в 1,7 раза выше.

Примерная схема реализации данной системы оптической передачи информации представлена на рис. 4. Возможным представляется передача 2-мя вариантами: 1) Использование модулятора, который меняет угол поляризации. Например, можно использовать акустооптический модулятор, который позволяет поворачивать на определенный угол линейную поляризацию [3]. 2) Использование излучателя, который состоит из 4-х лазеров, каждый из которых обладает фиксированным положением поляризации [2], излучает сигнальный импульс. Далее сигнал поступает уже непосредственно в волоконно-оптическую среду. Приемником будет служить устройство, которое не только бу-

дет принимать сигнал, но и обрабатывать его. Для того, чтобы распознать индивидуальную ориентацию импульсного сигнала возможно использование поляриметров [2], чувствительность определения угла поляризации которых очень высока.



Рис. 4. Схема реализации данной системы оптической передачи информации.

Предложенная система оптической передачи информации с помощью линейно-поляризованных электромагнитных волн позволяет увеличить скорость передачи сообщения по сравнению с традиционной передачей, а главное, при передаче комбинациями поляризуемых волн улучшится самосинхронизация, а следовательно, и надежность.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Патент РФ №2359421/ 20.06.2009 – Греш О. Г., Матасов Ф. В. – Способ передачи информации с помощью линейно-поляризованных электромагнитных волн и система для его осуществления.
2. Физика - большой энциклопедический словарь/ Под ред. А. М. Прохорова. - М.: Большая Российская энциклопедия, 1999. - 944 с.; с.576.
3. Антонов С. Н. Акустооптические устройства управления неполяризованным светом и модуляторы поляризации на основе кристалла парателлурита // Журнал технической физики. – 2004. – Том 74, вып. 10.

УДК 621.376.23

Е.Г. ЗВЕРЕВ, Е.В. ВАСИЛЬЕВ

Рязанский государственный радиотехнический университет

ПОВЫШЕНИЕ ЛИНЕЙНОСТИ ПРЕЦИЗИОННОГО АМПЛИТУДНОГО ДЕТЕКТОРА

В статье рассматривается ряд проблемных вопросов повышения линейности и расширения динамического диапазона амплитудного детектора, используемого в радиоэлектронной аппаратуре, формирующий ВЧ колебание с прецизионной точностью.

В настоящее время в аналитической технике с прецизионным управлением, которая имеет в своем составе высоковольтный (ВВ) генератор высокой частоты (ВЧ) с разверткой амплитуды колебания в диапазоне 1 В до 1.5 кВ, необходимо, при постоянной частоте, очень точно поддерживать амплитуду выходного колебания ВЧ генератора, контролируя ее 16-ти разрядным аналого-цифровым преобразователем (АЦП), который является нагрузкой амплитудного детектора (АД). Это определяет высокие требования к линейности разрабатываемого амплитудного детектора. Для управления ВЧ колебанием генератора применяется система автоматического регулирования (САР) амплитуды, в которую входит АД, АЦП, микроконтроллер, реализующий функцию ПИД-регулятора, синтезатор частоты прямого синтеза (DDS).

В соответствии с требованиями к линейности полупроводникового детектора был разработан двухполупериодный полупроводниковый амплитудный детектор со смещением, схемотехническая модель которого представлена на рис. 1:

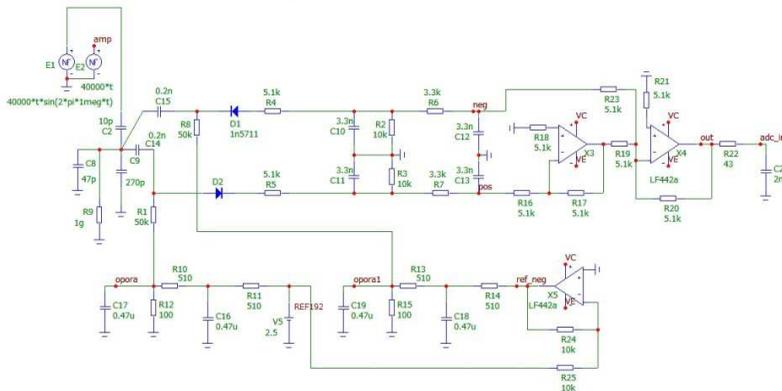


Рис. 1. Двухполупериодная модель полупроводникового детектора со смещением

Принципиальная схема амплитудного детектора построена по двухполупериодной схеме, что необходимо для увеличения его быстродействия. Цепь смещения двухполупериодного детектора обеспечивает высокое качество управления САР в широком диапазоне амплитуд выходного ВЧ колебания генератора, путем смещения детекторной характеристики таким образом, чтобы ее касательная была одинакова на всем рабочем участке детекторной характеристики. Источником напряжения цепи смещения в двухполупериодном полупроводниковом детекторе является прецизионный источник опорного напряжения, ток которого протекает через последовательно соединенный диод и рези-

сторы R4, R8, R1 и R5, общая величина сопротивления которых в сотни раз выше дифференциального сопротивления диода. Благодаря такому схемотехническому решению удастся добиться высокой температурной стабильности цепи смещения полупроводникового амплитудного детектора.

Одним из главных требований, которые предъявляются к амплитудному детектору, является его линейность. Для исследования линейности детекторной характеристики и динамического диапазона двухполупериодного амплитудного детектора была разработана его упрощенная модель, принципиальная схема которой представлена на рис. 2:

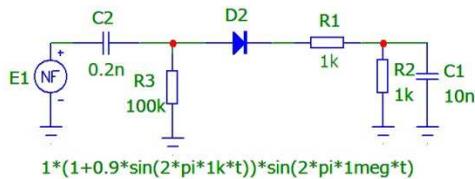


Рис. 2. Принципиальная схема упрощенной модели амплитудного детектора с модулированным ВЧ источником напряжения

Для увеличения линейности детекторной характеристики и динамического диапазона, в цепь упрощенной модели амплитудного детектора, последовательно к полупроводниковому диоду D2, были введены резисторы R3 и R1. Зависимость детекторной характеристики от общей величины сопротивления R резисторов R1 и R3, представлена на рис. 3:

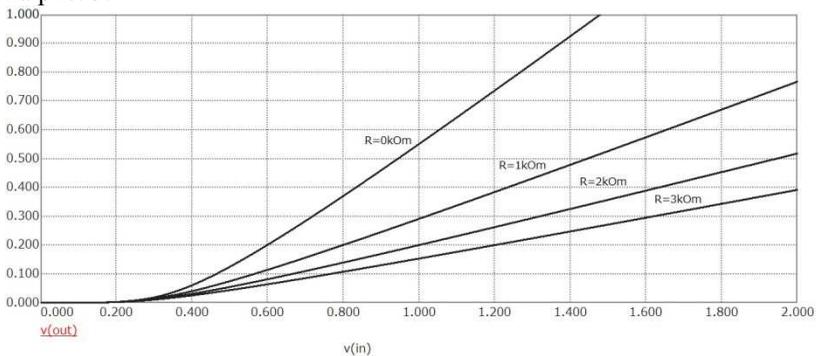


Рис. 3. Зависимость детекторной характеристики от величины линейзирующего резистора

Как видно из приведенных выше графиков, при увеличении величины сопротивления R линейность детекторной характеристики и динамический диапазон амплитудного детектора увеличивается, при этом уменьшается его чувствительность. Однако для разрабатываемого детектора, нет необходимости в высокой чувствительности, так как используемые ВВ колебания на входе амплитудного детектора обеспечивают его работу на всем участке детекторной характеристики.

Для оценки линейности амплитудного детектора можно воспользоваться двумя способами: воздействием низкочастотного гармонического напряжения или воздействием ВЧ напряжения, модулированного по гармоническому закону, на упрощенную модель амплитудного детектора. Линейность полупроводникового амплитудного детектора оценивается по появлению на его выходе гармонических составляющих с частотой модуляции и ее гармоник. Для определения количественной оценки нелинейных искажений используется коэффициент гармоник, который определяется следующим выражением:

$$K_{г} = \frac{\sqrt{U_{m_2}^2 + U_{m_3}^2 + \dots + U_{m_N}^2}}{U_{m_1}}, \text{ где } U_{m_N} - \text{амплитуда } N\text{-ой гармоники.}$$

ники.

В результате исследования линейности полупроводникового амплитудного детектора с кремниевым диодом и диодом с барьером Шоттки (ДБШ), при непосредственном воздействии НЧ гармонического напряжения на упрощенную модель амплитудного детектора, была получена зависимость коэффициента гармоник на его выходе от величины сопротивления R , которая представлена на рис. 4:

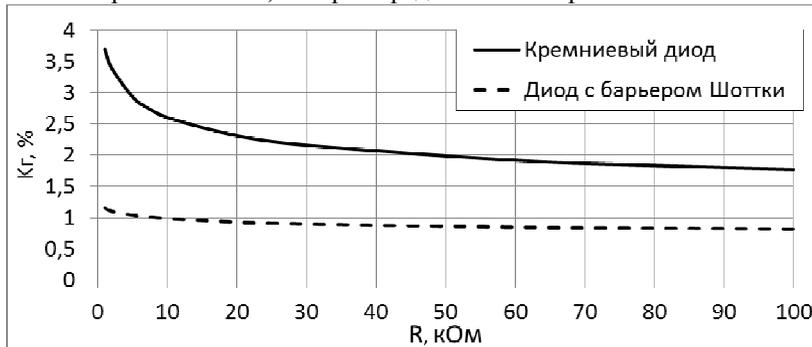


Рис. 4. Зависимости коэффициента гармоник от величины линейризирующих резисторов R в упрощенной модели амплитудного детектора с гармоническим источником напряжения

Как видно из приведенного выше рисунка, при изменении величины линейризирующего резистора на 100 кОм линейность полупроводникового детектора с кремниевым диодом увеличилась на 1.926 % а с ДБШ на 0.34%.

При воздействии на упрощенную модель амплитудного детектора ВЧ колебания с гармонической амплитудной модуляцией, была получена зависимость коэффициента гармоник на его выходе от величины сопротивления R , которая представлена на рисунке 5:

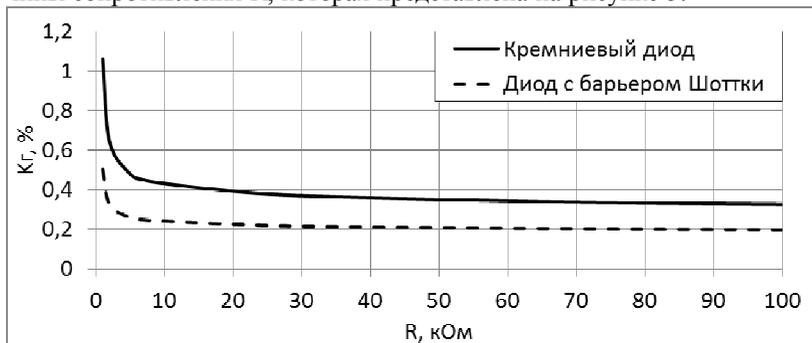


Рис. 5. Зависимости коэффициента гармоник от величины линейризирующих резисторов R , в упрощенной модели амплитудного детектора с модулированным ВЧ источником напряжения

Как видно из приведенных выше графиков, при изменении величины линейризирующего резистора на 100 кОм линейность полупроводникового детектора с кремниевым диодом увеличилась на 0.736 % а с ДБШ на 0.305%. Также стоит отметить, что общая линейность у амплитудного детектора с ДБШ выше по сравнению с кремниевым диодом. По этой причине применение таких диодов в разработанном нами двухполупериодном полупроводниковом детекторе позволяет увеличить динамический диапазон и улучшить линейность детекторной характеристики.

Из всего вышесказанного следует, что обычный диодный детектор это крайне нелинейное устройство с небольшим динамическим диапазоном. Благодаря приведенному выше исследованию, был разработан прецизионный ВЧ детектор с улучшенной линейностью детекторной характеристики, повышенным динамическим диапазоном и быстродействием, что позволяет использовать его в ВВ аналитической аппаратуре в качестве датчика для САР амплитуды.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1.Новиков Ю.Н. Электротехника и электроника. Теория цепей и сигналов, методы анализа: учебное пособие. – СПб.: Питер, 2005. – 384с.:ил.

2.Иванова В.Г., Тяжев А.И. Цифровая обработка сигналов и сигнальные процессоры/Под ред. д.т.н., профессора Тяжева А.И. 2008. – 307с

3.Полупроводниковые приборы и устройства: Учеб. пособие / Д.В.Величко, В.Г. Рубанов. – Белгород: Изд-во БГУ им. В.Г.Шухалова, 2006. – 184с.

УДК 621.3.049.774

М.А. ЗЕНИНА

Рязанский государственный радиотехнический университет

**СТРУКТУРА СИСТЕМЫ ДЕТЕКТИРОВАНИЯ ДВИЖУЩИХСЯ
ОБЪЕКТОВ В ВИДЕОПОТОКЕ С ИСПОЛЬЗОВАНИЕМ ПЛИС**

В статье рассматривается структурная схема системы детектирования движущихся объектов с использованием программируемой логической интегральной схемы (ПЛИС) Altera.

ПЛИС FPGA (field-programmable gate array) Altera представляет собой подготовленный функциональный модуль. Она обладает периферийными интерфейсами такими как VGA, клавиатура и др. Вмонтированные кнопки и семисегментный дешифратор предоставляют дополнительные возможности. Есть возможность подключения дополнительных устройств к ПЛИС по средствам 26 свободных выводов.

В первую очередь для построения системы детектирования на основе ПЛИС необходимо подключение камеры и дополнительного устройства, позволяющего преобразовывать данные с камеры в формат доступный для обработки. В данном случае рационально использование камерного модуля с CMOS матрицей. Такая матрица преобразует энергию падающего на нее света сразу в цифровую информацию. А данные считываются с помощью последовательного интерфейса, что обеспечивает простоту работы с данными. Камера подключается к свободным выводам ПЛИС.

Еще одним устройством, необходимым для полноценной работы системы является устройство для визуального отображения информа-

ции – монитора. Для его подключения на ПЛИС есть специальный разъем.

Основным звеном в рассматриваемой системе детектирования выступает сама ПЛИС и реализованный на ней алгоритм обработки информации [1,2]. Для получения качественного результата необходимо использовать трудоемкие алгоритмы обработки информации. ПЛИС более эффективны в плане быстродействия по сравнению микроконтроллерами, однако временные затраты на подготовку программного продукта больше, в связи с особенностями языков программирования.

При работе с изображениями, приходится иметь дело большими объемами данных, большинство алгоритмов обработки данных требуют многократного доступа к одному и тому же пикселю изображения в различные моменты времени [3]. Чтобы обеспечить систему необходимыми ресурсами памяти, к ПЛИС подключается чип оперативной памяти, который позволяет хранить данные о предыдущем кадре, а также подготавливать данные для передачи на устройство визуального отображения. Подключение дополнительной памяти обычно сопровождается установкой специального контроллера записи-чтения, или обеспечивает эту функцию программно.

Обобщенная структура системы детектирования движения представлена на рис. 1.

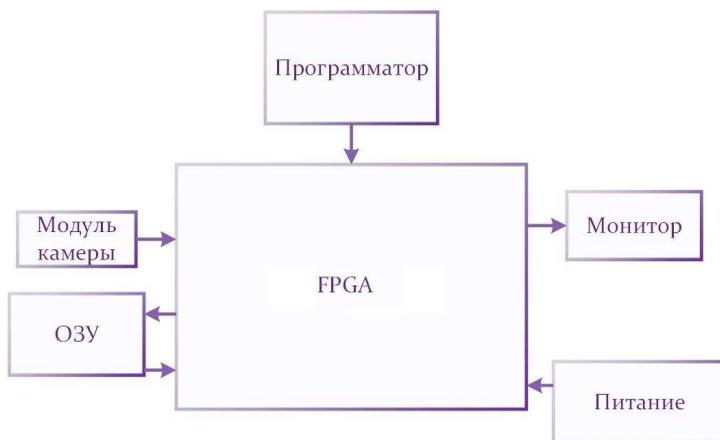


Рис. 1. Обобщенная структура системы детектирования движения.

Данная структура предназначена для исследования основных алгоритмов детектирования движения [4]. Любой алгоритм, как и описа-

ние по управлению периферийными устройствами, записывается в ПЛИС с помощью специального устройства – программатора. С его помощью производится конфигурирование структуры ПЛИС.

В качестве камеры предлагается использовать модуль VGA камеры OV7670 с CMOS матрицей. Данный модуль обеспечивает трансляцию видеопотока в режиме реального времени. Матрица OV7670 обладает интерфейсом передачи данных Serial Camera Control Bus (SCCB) – аналог распространённого I2C, что позволяет обрабатывать поток данных и передавать их на устройство управления (ПЛИС). В основе модуля OV7670 стоит недорогая матрица и сигнальный процессор, способные выдавать изображение с разрешением 640x480 с частотой 30 fps. Также возможна ручная настройка разрешения до 40x30.

Таким образом, предложенная структурная схема системы детектирования движущихся объектов может быть собрана из трех основных элементов: камера, плис, монитор. Однако при этом алгоритм детектирования будет тривиальным, и иметь не высокую степень достоверности. При усложнении алгоритма, необходимо подключение оперативной памяти, что усложняет систему, но при этом делает ее более гибкой.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Гаганов В., Конушин А. Сегментация движущихся объектов в видеопотоке //Компьютерная графика и мультимедиа. Выпуск №3. 2004.
2. Ярославский Л.П. Введение в цифровую обработку изображений. – М.: Мир, 1988.
3. Анисимов Б.В. Распознавание и цифровая обработка изображений – М.: Высш. школа, 1983 – 295с.
4. Зенина М.А. Исследование алгоритмов обработки видео сигнала / Информационные технологии, межвузовский сборник научных трудов. – Рязань: Рязанский государственный радиотехнический университет, 2018 – с.28-31.

УДК 681.518.3

Д.В. ЗЕНКОВ, М.М. ДЕВЯКОВИЧ

Рязанский государственный радиотехнический университет

УНИВЕРСАЛЬНЫЙ МОДУЛЬ РАСШИРЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ ИНЖЕНЕРНЫХ СИСТЕМ ЗДАНИЙ И ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ В ПРОМЫШЛЕННОСТИ

Рассматривается применение современных устройств многоканальной обработки и передачи данных для автоматизации инженерных систем зданий и технологических процессов в промышленности.

Модули расширения (ввода-вывода) – устройства, используемые для увеличения количества каналов поступающей информации для систем удаленного доступа, а также числа каналов сигналов управления и исходящей информации.

Модули ввода/вывода делятся на аналоговые и дискретные. Принцип работы:

- приём, обработка и передача входных сигналов через общую шину в контроллер.
- приём, обработка и передача сигналов общей шины на выходные каналы.

Основные области применения модулей ввода-вывода:

- автоматизация удаленных локальных технологических устройств;
- автоматизация производственных линий, участков, технологических комплексов (сборочных линий, конвейеров, станочных групп);
- системы сбора и передачи информации;
- системы диспетчеризации и управления распределенными производственными комплексами с удаленным вводом-выводом.

Имеется гальваническая изоляция питающей цепи и всех каналов модуля от общей шины.

Модули ввода-вывода работают на одном из протоколов связи: DeviceNet, ModBus-RTU. Обмен данными между модулем и контроллером осуществляется по принципу «Master – Slave». При этом модуль ввода-вывода работает как Slave устройство.

Процессор модуля расширения может также выполнять ряд дополнительных задач, таких как: калибровка и линеаризация показаний датчиков, контроль напряжения питания, диагностика и т. п.

В конце 2018 года компания Segnetics запустила в производство универсальные модули расширения - FMR, являющиеся мощным средством масштабирования подсистемы ввода - вывода.

Внешний вид с указанием и описанием элементов модуля представлен на рис. 1.

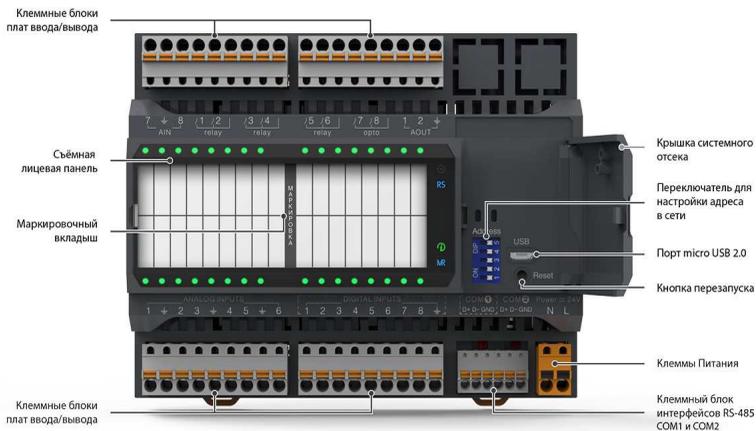


Рис. 1. Внешний вид универсального модуля расширения FMR.

Один модуль расширения поддерживает до 32 каналов. Благодаря возможности каскадного подключения, к одному Master устройству можно подключить до 341 модуля FMR, тем самым объединив до 10912 каналов. Обмен данными между модулями происходит со скоростью до 1 Мбит/с по внутренней высокоскоростной шине MRBus II.

Главное преимущество этой новинки: интерфейс RS 485, с помощью которого происходит подключение к любому контроллеру по протоколу ModBus-RTU.

Настройка, диагностика и отладка модулей производятся с помощью программы FMR Configurator. В режиме настройки не требуется отдельного источника питания — достаточно подключения по USB. Программа позволяет настроить любые параметры устройства.

Важной функцией конфигуратора, помимо настройки, является режим отладки. Программа конфигурирования позволяет производить отладку работы системы, модуля и всех устройств, подключенных к нему по системной шине. Этот режим незаменим, когда требуется проверить работу датчиков, исполнительных устройств на объекте. По

окончании отладки конфигурация системы может быть сохранена в файл.

Модуль оснащен защищенным источником питания с широким входным диапазоном питающих напряжений, устойчивыми к перегрузкам входами, выходами и портами связи. При возникновении аварийных ситуаций, приводящих к потере связи модуля с управляющим контроллером или модулей подключенных друг к другу, для сохранения контроля над состоянием объекта FMR способен переходить в безопасный режим работы.

Дополнительную защиту сетевых данных обеспечит экранированное подключение.

Особенности модуля:

- высокопроизводительное вычислительное ядро на базе 32-битного контроллера Cortex®;
- продуманная компоновка входов/выходов позволяет подобрать оптимальную конфигурацию оборудования для различных систем автоматизации;
- возможность увеличения количества входов/выходов за счет подключения дополнительных модулей расширения;
- изолированные униполярные цифровые входы с функцией счета импульсов;
- конфигурируемые, устойчивые к перегрузкам, аналоговые входы высокого разрешения с поддержкой режимов измерения напряжения, тока, и основных типов температурных датчиков;
- устойчивые к перегрузкам аналоговые выходы 0-10В;
- изолированные дискретные выходы различных типов;
- встроенный, защищенный источник питания с широким входным диапазоном питающих напряжений и возможностью работы от сетевого трансформатора;
- встроенный порт USB позволяет производить конфигурирование модуля и обновление ПО с использованием приложения для ПК.

Лицевая панель модуля оборудована светодиодами, информирующими пользователя о режиме работе входов/выходов и состоянии модуля.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Segnetics. Модуль расширения FMR [Электронный ресурс]. – URL: <http://segnetics.com/fmr>
2. РусАвтоматизация. Модули ввода-вывода [Электронный ресурс]. – URL: <https://rusautomation.ru>

УДК 621.37

А.Ю. ИБРАГИМОВ

Рязанский государственный радиотехнический университет

ПОМЕХА С НЕГАУССОВСКИМ ЗАКОНОМ РАСПРЕДЕЛЕНИЯ

Представлены результаты анализа моделей помех с негауссовским законом распределения, которые применяются для описания статистических помех от морской поверхности.

Помеха, это нежелательный сигнал на входе приёмного устройства РЛС. Бывают пассивные и активные помехи. Пассивная помеха представляет собой отражение сканирующего луча РЛС от стационарных местных предметов, облаков, морских волн, стай птиц и т.п. В силу хаотического перемещения помех в виде «блестящих точек», векторная сумма отражений на приемной антенне РЛС будет иметь случайную фазу и амплитуду, и поэтому пассивную помеху можно представить в виде случайной последовательности [1].

Разные авторы используют различные математические модели для описания статистических характеристик помех от морской поверхности и предлагают методы для оценки их параметров.

Самыми популярными моделями, описывающими статистику огибающей морских помех являются Рэлеевская модель, логарифмически-нормальная модель, модель Вейбулла. При этом наиболее часто для аппроксимации отражений от морской поверхности используют логарифмически-нормальную плотность.

При использовании модели подстилающей поверхности в виде множества совместно независимых отражающих элементов, которые случайно распределены в элементе разрешения, плотность распределения вероятностей амплитуды отраженного сигнала подчиняется закону Рэлея. Это распределение используется для моделирования флуктуаций амплитуд локационного сигнала, который отражен от морской поверхности. Распределение Рэлея:

$$P(E) = \frac{E}{\sigma_c^2} \exp\left[-\frac{E^2}{2\sigma_c^2}\right] \quad (1)$$

где E – огибающая отраженного сигнала; σ_c^2 – среднеквадратическое отклонение отраженного сигнала.

Начальные моменты случайной величины E , имеющей рэлеевское распределение вероятности, определяются выражением [2]:

$$M\{E^r\} = \sigma^r \Gamma\left(1 + \frac{r}{2}\right) \quad (2)$$

где $M\{\cdot\}$ - математическое ожидание; r – номер начального момента огибающей отраженного сигнала; $\Gamma(\cdot)$ – гамма функция.

Логарифмически-нормальное распределение вероятности можно записать в следующем виде [2]:

$$P(E) = \frac{1}{\sqrt{2\pi}\sigma_E E} \exp\left[-\frac{(\ln(E) - m)^2}{2\sigma_E^2}\right] \quad (3)$$

где m – математическое ожидание случайной величины E .

Данное распределение обладает следующим полезным свойством: при возведении случайной величины, распределенной по логарифмически-нормальной плотности, в любую степень полученная случайная величина также имеет логарифмически-нормальное распределение [3].

Для определения параметров распределений для конкретных условий наблюдения морской поверхности необходимо определить параметры распределений через среднюю мощность отражений \tilde{P} и коэффициент вариации мощности K_p . Используя методику приведенную в [4] получим:

$$\sigma_p = \sqrt{\ln(1 + K_p^2)} \quad (4)$$

где $\sigma_p = 2\sigma_E$.

$$m = \frac{\tilde{P}}{\sqrt{\ln(1 + K_p^2)}} \quad (5)$$

$$\sigma_E = \frac{\sigma_p}{2} = \frac{1}{2} \sqrt{\ln(1 + K_p^2)} \quad (6)$$

Средняя мощность отраженного сигнала определяется по известному уравнению радиолокации, которое позволяет учесть и характеристики локатора, и условия наблюдения поверхности моря, и состояние морской поверхности [3]. Выражение для коэффициента вариации для разных условий поверхности моря, ее состояния и параметров локатора приведено в работе [4].

На рис. 1 представлен вид плотности вероятностей огибающей (3) при различных отношениях «помеха/шум»: $\alpha = 0, 10, 20, 30$ дБ; им соответствуют кривые 1 — 4; кривая 5 построена для $P_{ш} = 0$ (внутренний шум не учитывается).

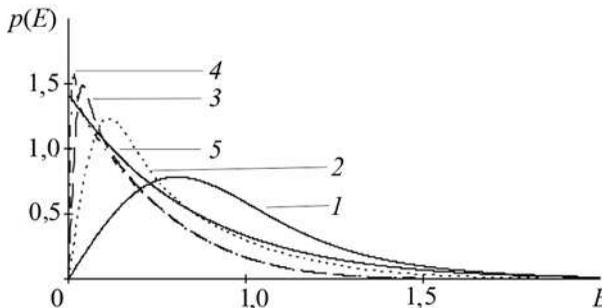


Рис. 1. Плотность вероятностей огибающей (3) при различных отношениях «помеха/шум»

На рис. 2 изображены гистограмма экспериментальных значений огибающей морских отражений (кривая 1) в эксперименте [5] и кривые плотности распределения вероятности огибающей морских помех, основанные логнормальном (кривая 2), рэлеевском (кривая 3).

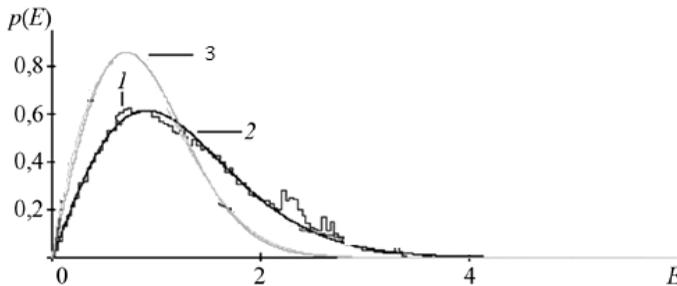


Рис. 2. Гистограмма экспериментальных значений огибающей морских отражений в эксперименте [5].

Расчетные параметры моделей для рис. 2: мощность помехи $P_{\Pi} = 1$ Вт, угол скольжения $\Phi = 1,3^\circ$.

На основании выше сказанного можно сделать вывод, что модель морских помех, основанная на логнормальном распределении, по сравнению с рэлеевским законом распределения наиболее точно описывает структуру морских отражений.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Лещинский В.Ю., Лозовский И.Ф. Метод построения двухканального обнаружителя с постоянным уровнем ложных тревог для помех с нерэлеевским распределением огибающей // Труды XIII Международной научно-технической конференции актуальные проблемы

электронного приборостроения: сб. тр. научн.-технич. конф., - Новосибирск, НГТУ, 2016. Том 12. – С.79-81.

2. Antipov I. Simulation of Sea Clutter Returns.Salisbury: DSTO Electronic and Surveillance Research laboratory, 1998.

3. Исаков В. И., Шепета Д. А. Моделирование локационных сигналов отраженных от кромки земля-море.// Информационно управляющие системы. 2017. № 5 С. 89–94.

4. Тверской Г. Н., Терентьев Г. К., Харченко И. П. Имитаторы эхо-сигналов судовых радиолокационных станций. — Л.: Судостроение, 1973. — 228 с.

5. Li Weibo, Zhang Youzhi, Zhang Guangyi//4th International Conference on Advanced Materials and Information Technology Processing (AMITP 2016).

УДК 004.65

Н.С. ИВАНОВА

Рязанский государственный радиотехнический университет

РЕПЛИКАЦИЯ БАЗ ДАННЫХ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

Рассматривается понятие репликации, ее типы, особенности использования и применение в информационных системах.

Разработка информационных систем и их сопровождение остается одной из самых актуальных и важных функций компьютерных специалистов, наряду с системным администрированием и проектированием систем управления базами данных (СУБД). Широкое внедрение вычислительных машин и электронного варианта использования данных во все сферы промышленности, связи, систем управления и документооборота требует обеспечения качественной обработки, быстрого доступа и надежного хранения данных.

Данные возможности доступны благодаря непрерывно работающим удаленным системам. Так как при промышленной эксплуатации совершенно неприемлемо сталкиваться с потерей или недоступностью данных, то появляется задача повышения отказоустойчивости базы данных.

Один из способов разработки отказоустойчивых баз данных - создание их реплик.

Репликация — одна из техник масштабирования баз данных. Заключается эта техника в том, что данные с одного сервера базы данных регулярно копируются (реплицируются) на один или несколько других (количество таких серверов регулируется параметром, называемым фактором репликации), называемых репликами. Таким образом для информационной системы появляется возможность использовать не один сервер для обработки всех запросов, а несколько, что позволяет распределить нагрузку с одного сервера на несколько.

Репликация может быть синхронной и асинхронной:

- Синхронная репликация называется, когда обновление данных сервера и всех других реплик того же фрагмента происходит в одной и той же транзакции. Логически это означает, что существует лишь одна версия данных. Зачастую синхронная репликация реализуется с помощью триггерных процедур. Недостатком синхронной репликации является то, что она создаёт повышенную нагрузку при выполнении всех транзакций, в которых происходит обновление реплик. Кроме того, могут возникать проблемы, связанные с доступностью данных (например, при блокировках, накладываемых при записи в реплику).

- Асинхронная репликация называется, в случае обновления данных одной реплики, которые распространяется на другие спустя некоторое время, а не в той же транзакции. Таким образом, при асинхронной репликации вводится задержка, или время ожидания, в течение которого отдельные реплики могут быть фактически неидентичными. В большинстве продуктов асинхронная репликация реализуется посредством чтения журнала транзакций или постоянной очереди тех обновлений, которые подлежат распространению. Преимущество асинхронной репликации состоит в том, что дополнительные издержки репликации не связаны с транзакциями обновлений, которое имеет важное значение для функционирования, и позволяет предъявлять высокие требования к производительности. К недостаткам этой схемы относится то, что данные на репликах не всегда идентичны.

Создание и применение реплик в информационных системах решает сразу несколько задач по повышению надежности и производительности:

- Производительность и горизонтальная масштабируемость. Если один сервер не справляется с нагрузкой, создаваемой одновременными операциями чтения и записи в базу данных, то запросы на чтение данных можно перенаправить на реплики, при этом выгода такой системы будет весомей, если на одну операцию записи происходит несколько операций чтения.

- Отказоустойчивость. При выходе из строя одного из серверов, данные будут доступны с других реплик; повышается защита от полной потери данных.

- Резервирование данных. При использовании реплики мы всегда уверены, что есть, «страховочная» база данных, практически идентично повторяющая основную.

- Сложные вычисления. Сложные и медленные запросы к базе данных можно выполнять на отдельной реплике, не замедляя работу всей системы.

- Модернизация оборудования. При необходимости остановки или замены одного или нескольких серверов, не нарушится работа системы в целом.

У данного процесса так же есть ряд недостатков:

- Характерны проблемы, связанные с доступностью данных.

- Необходима одинаковая структура хранения данных в БД.

- Часть времени копии данных не идентичны базовым данным, поэтому пользователи должны учитывать, когда именно были синхронизированы эти данные.

При создании инфраструктуры базы данных на основе метода репликации, не критичен отказ главного сервера, работоспособность системы можно восстановить за несколько минут вручную, или создать контроллер, который будет выполнять эти действия автоматически, сделав одну из реплик главным сервером и заменив настройки остальных реплик таким образом, чтобы они забирали журнал с нового, главного сервера, баз данных.

Метод можно усовершенствовать, прибегнув к созданию отдельного контроллера, который будет фиксировать все запросы к базам данных. И именно он будет отвечать за распределение нагрузки по доступности реплик. В итоге, не будет главного сервера базы данных, а запросы на выборку данных будут распределяться только по тем репликам, у которых данные для запроса актуальны и консистентны.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Каратыгин, С., Тихонов, А.; Долголаптев, В. Базы данных. М.: АБФ, 1995. - 352 с.
2. Дьюба, П. MySQL. М.: Вильямс, 2001. - 236 с.
3. Энсор, Д., Стивенсон, Й. Oracle. Проектирования баз данных. Киев: BHV, 2000. 560 с.
4. Бергер, А.Б. Microsoft SQL Server 2005 Analysis Services.

УДК 004.67

Н.С. ИВАНОВА

Рязанский государственный радиотехнический университет

**РАСПРЕДЕЛЕННАЯ ОБРАБОТКА ДАННЫХ
В ИНФОРМАЦИОННЫХ СИСТЕМАХ С ИСПОЛЬЗОВАНИЕМ
HADOOP**

Рассматривается подход к решению проблемы обработки больших объемов данных, параллельное исполнение вычислений, принципы работы с BigData.

Стремительное развитие информационных технологий и социальных сервисов служит причиной поиска и разработки информационных решений, которые позволяют обрабатывать гигантские объемы входящей информации.

В условиях лавинообразного роста объемов данных на помощь организациям приходят технологии Big Data – они обеспечивают обработку массивов информации из различных источников. Один из основных вариантов эффективной работы с большими данными — построение системы на открытой и расширяемой платформе Hadoop, которую с 2005 г. развивает квалифицированное Open Source сообщество Apache Software Foundation.

Hadoop включает в себя набор утилит, библиотек и фреймворк для разработки, позволяющий создавать производительные системы обработки разнородных данных за счет использования модели распределенных вычислений.

Одной из проблем работы с большими объемами данных является трудность их передачи между серверами для последующей обработки. Также из-за большого числа узлов возможны частые сбои отдельных узлов, поэтому вопрос надежности также является очень важным. Данную проблему можно решить с помощью технологий Hadoop Distributed Filesystem (HDFS) и MapReduce.

HDFS - распределенная файловая система Hadoop. Кластер HDFS состоит из двух типов узлов, работающих по схеме «управляющий - подчиненный»: узел имен (управляющий) NameNode-сервер и несколько узлов данных (подчиненные) DataNode-серверы. HDFS подобна другим традиционным файловым системам: файлы хранятся в виде блоков, существует маппинг между блоками и именами файлов, поддерживается древовидная структура, поддерживается модель доступа к файлам основанная на правах и т. п. Копии блоков (реплики) хранятся на нескольких серверах, по умолчанию - трех.

Вторым важным элементом Hadoop является MapReduce – framework для вычисления распределенных задач. В соответствии с подходом MapReduce обработка данных состоит из двух шагов: Map и Reduce. На шаге Map выполняется предварительная обработка данных, которая осуществляется параллельно на различных узлах кластера. На шаге Reduce происходит сведение предварительно обработанных данных в единый результат.

Таким образом под Big Data понимается не какой-то конкретный объём данных и даже не сами данные, а методы их обработки, которые позволяют распределенно обрабатывать информацию.

Исходя из определения Big Data, можно сформулировать основные принципы работы с такими данными:

1. Горизонтальная масштабируемость. Поскольку данных может быть сколь угодно много – любая система, которая подразумевает обработку больших данных, должна быть расширяемой и иметь возможность увеличивать мощность не за счет добавления аппаратных средств в одну машину, а добавления новых машин в кластер.

2. Отказоустойчивость. Принцип горизонтальной масштабируемости подразумевает, что машин в кластере может быть много. Это означает, что часть этих машин будет гарантированно выходить из строя. Методы работы с большими данными должны учитывать возможность таких сбоев и переживать их без каких-либо значимых последствий.

3. Локальность данных. В больших распределённых системах данные распределены по большому количеству машин. Если данные физически находятся на одном сервере, а обрабатываются на другом – расходы на передачу данных могут превысить расходы на саму обработку. Поэтому одним из важнейших принципов проектирования BigData-решений является принцип локальности данных – по возможности обрабатываем данные на той же машине, на которой их храним.

4. Отправка кода к данным, а не наоборот.

Все современные средства работы с большими данными так или иначе следуют этим принципам. Для того, чтобы им следовать – необходимо придумывать какие-то методы, способы и парадигмы разработки средств разработки данных. Рассмотрим более подробно работу MapReduce.

MapReduce предполагает, что данные организованы в виде некоторых записей. Обработка данных происходит в 3 стадии:

1. Стадия Map. На этой стадии данные предобрабатываются при помощи функции map(), которую определяет пользователь. Работа этой стадии заключается в предобработке и фильтрации данных. Рабо-

та очень похожа на операцию `map` в функциональных языках программирования – пользовательская функция применяется к каждой входной записи.

Функция `map()` примененная к одной входной записи и выдаёт множество пар ключ-значение. Множество – т.е. может выдать только одну запись, может не выдать ничего, а может выдать несколько пар ключ-значение. Что будет находится в ключе и в значении – решать пользователю, но ключ – очень важная вещь, так как данные с одним ключом в будущем попадут в один экземпляр функции `reduce`.

2. Стадия `Shuffle`. Проходит незаметно для пользователя. В этой стадии вывод функции `map` «разбирается по корзинам» – каждая корзина соответствует одному ключу вывода стадии `map`. В дальнейшем эти корзины послужат входом для `reduce`.

3. Стадия `Reduce`. Каждая «корзина» со значениями, сформированная на стадии `shuffle`, попадает на вход функции `reduce()`.

Функция `reduce` задаётся пользователем и вычисляет финальный результат для отдельной «корзины». Множество всех значений, возвращённых функцией `reduce()`, является финальным результатом `MapReduce`-задачи.

Несколько дополнительных фактов про `MapReduce`:

1) Все запуски функции `map` работают независимо и могут работать параллельно, в том числе на разных машинах кластера.

2) Все запуски функции `reduce` работают независимо и могут работать параллельно, в том числе на разных машинах кластера.

3) `Shuffle` внутри себя представляет параллельную сортировку, поэтому также может работать на разных машинах кластера. Пункты 1-3 позволяют выполнить принцип горизонтальной масштабируемости.

4) Функция `map`, как правило, применяется на той же машине, на которой хранятся данные – это позволяет снизить передачу данных по сети (принцип локальности данных).

5) `MapReduce` – это всегда полное сканирование данных, никаких индексов нет. Это означает, что `MapReduce` плохо применим, когда ответ требуется очень быстро.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Уайт Т. `Hadoop`: подробное руководство. СПб. : Питер, 2013. - 672 с.
2. `Hadoop`: Документация [Электронный ресурс] // `Hadoop` : офиц. сайт. URL: <http://hadoop.apache.org/docs/> (дата обращения: 17.03.2016).

3. Лэм Ч. Надоор в действии. М. : ДМК Пресс, 2012. 424 с.

УДК 004.89

М.С. ИВКИНА

Рязанский государственный радиотехнический университет

ПОВЫШЕНИЕ КАЧЕСТВА КЛАССИФИКАЦИИ

Рассматриваются способы повышения качества алгоритмов классификации данных, их реализация и основные проблемы при использовании.

Интеллектуальный анализ данных (Data Mining) – это процесс поддержки принятия решений, целью которого является обнаружение неявных закономерностей в наборах данных [2].

Классификация – это логическая закономерность, позволяющая делать вывод относительно определения характеристик конкретной группы, процесс упорядочения или распределения объектов по классам с целью отражения отношений между ними [3].

Способы повышения качества классификации:

- подбор оптимальных параметров модели;
- балансировка данных.

Для повышения качества классификации с помощью подбора оптимальных значений параметров алгоритма классификации могут быть использованы: алгоритм подбора по сетке, алгоритм роя частиц, байесовская оптимизация и другие алгоритмы нахождения экстремум функций (см. рис. 1).

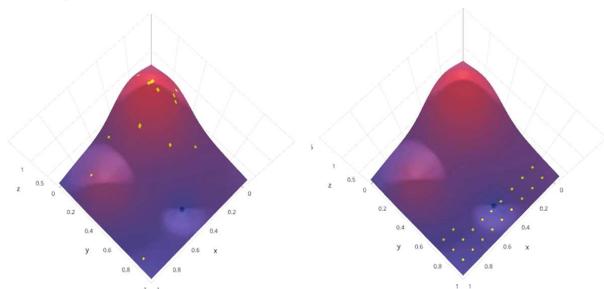


Рис. 1. Пример работы алгоритмов байесовской оптимизации и поиска по сетке.

Повышение качества классификации методом балансировки данных можно реализовывать такими методами как, удаление некоторого количества примеров мажоритарного класса (undersampling) (см. рис. 2) и увеличение количества примеров миноритарного (oversampling) (см. рис. 3) [1].

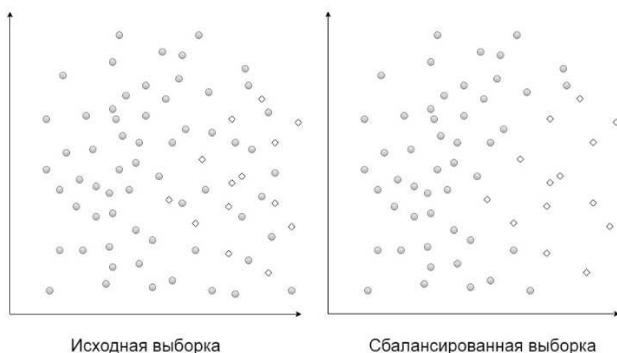


Рис. 2. Удаление мажоритарных примеров.

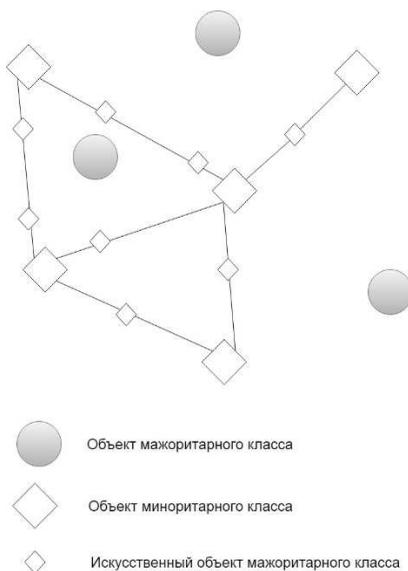


Рис. 3. Искусственно созданные новые примеры миноритарного класса.

Проблемы при подборе параметров

Основными проблемами при нахождении оптимальных значений параметров любого классификатора являются выбор диапазона значений параметра, в котором ищется лучшее значение, и время, которое требуется для нахождения значений.

Все описанные методы строят множество моделей и вычисляют значение показателя качества, которую необходимо максимизировать. Зачастую этот процесс может занимать продолжительное время. На это влияют диапазоны значений, в которых будет искажаться оптимальное значение, то есть, чем больше сочетаний параметров, тем больше времени потребуется на подбор значений. Так же время построения модели зависит от размеров обучающей и тестовой выборок.

Проблемы при балансировки данных

Основной проблемой при балансировке данных является необходимость самой балансировки. Балансировку необходимо делать только в том случае, если объектов одного класса много больше, чем объектов другого класса. Соотношение классов при котором целесообразно проводить балансировку данных 1 к 5.

Так же для большей эффективности можно объединить эти два метода. Каждый из этих способов повышения качества классификации следует применять при необходимости, иначе в лучшем случае качество классификации не изменится, а в худшем упадет.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Ивкина М.С. Решение задачи классификации на основе случайного леса // Прикладные исследования и технологии ART2018: Сборник трудов региональной конференции ART2018. 2018. С. 79-87.
2. Чистяков С. П. Случайные леса: обзор – Труды Карельского научного центра РАН № 1. 2013.
3. Demidova L., Klyueva I., Sokolova Y., Stepanov N., Tyart N. Intellectual Approaches to Improvement of the Classification Decisions Quality on the Base of the SVM Classifier // Procedia Computer Science 2017. P. 222-230.

УДК 004.89

Н.В. КАЛИНИН

Рязанский государственный радиотехнический университет

ЗАДАЧА О ВОСЬМИ ФЕРЗЯХ И ЕЕ ПРИМЕНЕНИЕ

В представленной работе была рассмотрена задача о размещении восьми ферзей на стандартной шахматной доске размером 8 на 8 клеток при условии, что ни один из ферзей не будет находиться под ударом другого.

Такая задача по размещению шахматных фигур была поставлена еще в 19 веке и звучит она следующим образом: необходимо разместить на доске из $n \times n$ клеток n ферзей так, чтобы ни один из них не был под ударом другого. Следует учитывать, что эта фигура бьет все клетки, расположенные по горизонтали, вертикали и диагоналям.

Значимость задачи заключается в том, что проблема этой задачи используется в качестве эталона для искусственного интеллекта, но так из-за простой сложности решения проблемы результаты применения ИИ являются спорными. Разработанные алгоритмы такой задачи можно считать почти универсальными, потому что им потребуются минимальные изменения для решения других задач.

Если фигуры не переходят друг в друга при отражениях и поворотах, а любая другая расстановка получается при помощи этих преобразований из предыдущей, то такое множество расстановок называется основным. Такой основной набор содержит 12 возможных вариантов, которые получаются путем поворота доски и отражений фигур. На рис. 1 представлен один из 92 возможных вариантов решения для стандартной шахматной доски [1].

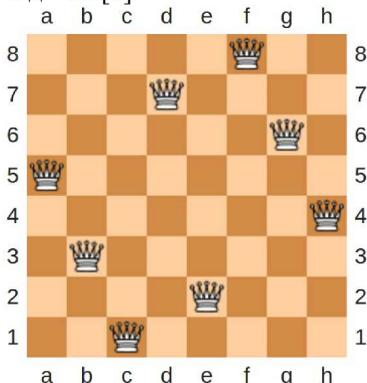


Рис. 1. Одно из решений задачи о 8 ферзях.

Существует несколько способов решения такой задачи:

1. Метод полного перебора. Рассчитаем количество расположений 8 ферзей на стандартной шахматной доске (64 клетки):

$$\frac{64!}{8!(64-8)!} = 4426165368. \text{ На перебор такого количество вариантов ух-}$$

одит множество времени, а если рассмотреть задачу по размещению 1000 фигур, то метод полного перебора становится вовсе не актуальным.

2. Алгоритм, исключаящий позиций, где ферзи находятся на одной горизонтали или вертикали. Это правило может уменьшить число возможных расположений до 16 777 216 (в 263 раза быстрее). А если добавить такое же условие для диагоналей, то число возможных вариантов уменьшится до 40 320.

3. Поиск с возвратом: один ферзь размещается на самую первую горизонталь. Следующий размещается на другую горизонталь так, чтобы он не находился под ударом ранее установленных ферзи. Если такой вариант не был найден, то алгоритм переставляет ранее установленного ферзя.

Задача о восьми ферзях кажется незначительной на первый взгляд, однако ее можно легко адаптировать для решения таких проблем, как:

1. Поиск людей в социальных сетях, которые не знакомы (это поможет составлять более оптимальные списки предложенных друзей).

2. Взлом кода различных защитных систем.

3. Обучение нейросетей.

По словам авторов статьи «Complexity of n-Queens Completion» решение подобных задач можно приравнять к решению задач тысячелетия – это список из семи математических задач, который был сформулирован Математическим институтом Клэя в 2000 году (например, задача равенства классов сложности P и NP)[2]. К сожалению, современные алгоритмы не способны решать задачу о ферзях, если размер доски составляет 1000*1000 клеток и выше, поэтому на сегодняшний день эта задача считается открытой [3].

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Задачи тысячелетия. [Электронный ресурс]. Режим доступа: URL:https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B8_%D1%82%D1%8B%D1%81%D1%8F%D1%87%D0%B5%D0%BB%D0%B5%D1%82%D0%B8%D1%8F
2. Задача о восьми ферзях. [Электронный ресурс] Режим доступа: URL:www.forum.loveoComplexityofnQueensCompletionrigami.info/viewtopic.php?f=21&t=564&start=30
3. Christopher Jefferson, Peter Nightingale. Complexity of n-Queens Completion. 2017 год.

УДК 628.23

Ю.Н. КАЛИНКИНА

Рязанский государственный радиотехнический университет

РАЗРАБОТКА КОНСТРУКЦИИ СИСТЕМЫ УПРАВЛЕНИЯ ПРИТОЧНО-ВЫТЯЖНОЙ ВЕНТИЛЯЦИИ

В статье рассматривается реализация системы управления приточно-вытяжной вентиляцией.

При оборудовании сооружения системами вентиляции решение указанных задач достигается путем подачи в помещения определенных, найденных расчетом, количеств наружного воздуха и удаление негодного для дальнейшего использования воздуха. [1]

Приточно-вытяжная вентиляция является комбинацией из двух видов: приточного и вытяжного типа. Процесс вентиляции состоит таким образом: система забирает из помещения, в котором она установлена, отработанный воздух и осуществляет приток нового.

Вентиляционная система состоит из наборных устройств и ряда специальных функциональных секций таких как: гибкая вставка, воздушная заслонка с приводом, фильтр, нагреватель воздуха канальный (электрический или водяной), рекуператор, шумоглушитель и система автоматики, которая позволяет управлять данной канальной группой.

Для управления системы могут использоваться регуляторы скорости, которые предназначены для плавного регулирования скорости вращения электродвигателя вентилятора; частотные преобразователи, которые позволяют осуществлять регулирование производительности вентилятора, плавный пуск, защиту от перегрузок, задание скорости

вращения вентилятора при помощи аналогового сигнала, изменяя напряжение и силу тока, а также системы автоматики. Они позволяют не только регулировать скорость вращения вентилятора и его производительности, но и осуществлять защиту клапанов и водяного контура нагрева от низких температур, имеется сигнализатор аварии, а также возможность проведения мониторинга работы отдельного механизма канальной группы с помощью датчиков.[2]

Приточно-вытяжная установка, исполненная в наборном виде представлена ниже (см. рис. 1).

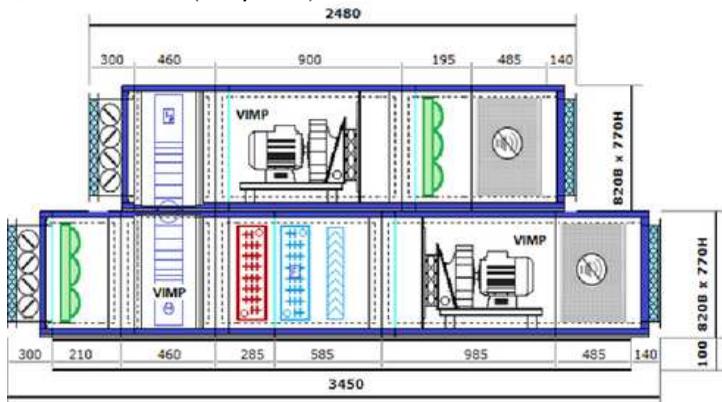


Рис. 1. Приточно-вытяжная установка.

В зависимости от содержания канальной группы выбирается автоматика, которая будет ей управлять. Данный подбор зависит от различных показателей, входящих в ее состав элементов. Исходя из показателей расхода воздуха - давления и мощности, подбирается вентилятор нужного размера, а также фильтр, рекуператор (позволяющий забрать часть тепла у выходящего воздушного потока и насытить им свежий воздух из люка притока), нагреватель воздуха, охладитель воздуха и шумоглушитель. Правильный подбор системы автоматики влияет на производительность установки и правильности выполнения заданных функций.

Для обеспечения работы установки в комплект автоматики входит:

- 1) Шкаф приборов автоматики и управления на базе контроллера Segnetics Pixel;
- 2) Дифференциальный датчик контроля засоренности фильтра;
- 3) Дифференциальный датчик контроля работы вентилятора;
- 4) Канальный датчик температуры воздуха;

- 5) Датчик температуры наружного воздуха;
- 6) Термостат по воздуху;
- 7) Датчик температуры обратной воды. [3]

Системы автоматизации, которые реализованы на базе контроллеров Segnetics имеют возможность подключения с помощью программы MasterScada. Контроллер Segnetics (Pixel) поддерживает работу с сетевыми модулями Ethernet и LON. Программная реализация приточно-вытяжной установки в программе MasterScada представлена ниже (см. рис. 2).



Рис. 2. Программная реализация системы автоматизации.

Таким образом, система автоматики позволяет выполнять все поставленные задачи при проектировании, исходя из правильности подбора оборудования. А также автоматическое регулирование вентиляции уменьшает влияние человеческого фактора и сводит к минимуму возможность ошибок при эксплуатации оборудования.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Стефанов Е.В. Вентиляция и кондиционирование воздуха / Е.В. Стефанов. – Санкт – Петербург: Издательство «Авок Северо-Запад», 2005.–78с.
2. Сигма-Регион. Автоматика [Электронный ресурс], URL: <http://sigma-region.ru/#automat> (дата обращения 30.03.19).
3. Сигма-Регион. Канальное оборудование [Электронный ресурс], URL: <http://sigma-region.ru/#kanal> (дата обращения 30.03.19).

УДК 004.89

Т.М. КРУПИНА

Рязанский государственный радиотехнический университет

**ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ДЛЯ РЕШЕНИЯ ЗАДАЧ
ПРОЕКТИРОВАНИЯ ЭС**

Рассматривается возможность применения генетического алгоритма к решению задачи размещения ЭС.

Основными задачами конструкторского проектирования являются задачи компоновки, размещения и трассировки. Не всегда можно применить математические расчеты для решения задач проектирования ЭС, т.к. они основываются на использовании индивидуальных знаний конструктора. В данной статье рассматривается возможность решения задачи размещения ЭС путем использования генетического алгоритма.

Многие задачи проектирования ЭС сводятся к задачам многокритериальности и оптимизации. Для решения таких задач часто недостаточно использования только технологии ЭС, для которой характерно преобладание символьных методов.

В настоящее время на основе многих исследований формируется направление вычислений с помощью генетических алгоритмов (ГА). Основным достоинством данной методологии являются снисхождение к неполноте исходных данных.

Генетический алгоритм – это алгоритм, используемый при решении задач оптимизации и моделирования случайным подбором, комбинированием и вариации искомых значений с использованием методов, аналогичных естественному отбору в природе. Является разновидностью эволюционных вычислений, с помощью которых решаются оптимизационные задачи средствами естественной эволюции, таких как наследование, мутации, отбор и кроссинговер.

Он реализует «выживание сильнейших» среди рассмотренных структур, формируя и изменяя поисковый алгоритм на основе моделирования эволюции [2].

Основой для возникновения генетических алгоритмов считается модель биологической эволюции и методы случайного поиска [1, 3]. Растрингин считает, что случайный поиск (СП) возник как реализация простейшей модели эволюции, когда случайные мутации моделировались случайными образом из оптимального решения, а отбор «отсеиванием» неудачных вариантов [3].

Так же Растрингин выделяет три особенности ГА [3]:

- каждая новая популяция состоит только из “жизнеспособных” хромосом;
- каждая новая популяция «лучше» (в смысле целевой функции) предыдущей;
- в процессе эволюции последующая популяция зависит только от предыдущей

В классических реализациях генетического алгоритма (ГА) предполагается, что генотип имеет фиксированную длину. Однако существуют вариации ГА, свободные от этого ограничения.

Из полученного множества решений («поколения») с учётом значения «приспособленности» выбираются такие, к которым применяются «генетические операторы» результатом чего является получение новых решений. Для них также вычисляется значение приспособленности, и затем производится отбор («селекция») лучших решений в следующее поколение [2].

Классическая модель ГА была предложена Холландом. Он же предложил схему алгоритма реализации:

1. Сгенерировать исходную популяцию, состоящую из N особей;
2. Оценить приспособленность хромосом в популяции на основе целевой функции $F(x)$;
3. Выполнить операцию селекции;
4. Применить генетические операторы (мутация, скрещивание и тд);
5. Сформировать новую популяцию;
6. Если критерий останова алгоритма не достигнут, перейти к шагу 2, иначе завершить работу [4].

В данной статье рассматривается применение ГА к задаче размещения ЭС. Исходными данными для решения задачи размещения ЭС на монтажном пространстве являются: a, b - габариты монтажного пространства; $\{(a b_1, i), \dots, (a b_i, i), \dots, (a b_n, n)\}$ - множество элементов размещения, характеризующиеся своими габаритными размерами (все размещаемые ЭС аппроксимируются прямоугольниками), a_i и b_i - габаритные ширина и длина i -го ЭС; C - матрица связности элементов размещения.

Необходимо найти такой вариант размещения элементов на монтажном пространстве $Z = \{(x y_{1,1}), \dots, (x y_{i,i}), \dots, (x y_{n,n})\}$, где $x y_{i,i}$ - координаты центра тяжести элемента размещения i , чтобы площадь перекрытия площадей размещенных элементов была равна нулю, а суммарная длина соединений минимальная. Задача размещения ставится как задача оптимизации функции, выражающую норми-

рованную оценку суммы штрафа за перекрытие площадей, размещаемых ЭС и общей длины соединений:

$$F = O * L(\sum z_i) + P * S(\sum z_j)$$

где z_j - вариант размещения, $L_{\sum}(z_j)$ - суммарная длина соединений j -го варианта размещения, $S_{\sum}(z_j)$ - общая площадь размещаемых элементов, $O L(\sum(z_j))$ - оценка общей длины соединений, приведенная к интервалу $[0,1]$. $P S(\sum(z_j))$ - функция штрафа за перекрытие площадей, принимающая значения из интервала $[0,1]$.

Суммарная длина соединений рассчитывается по формуле:

$$L_{\sum} z_j = \sum_{i=1}^n \sum_{k=1}^n d_{ik} * c_{ik}$$

где c_{ik} - число связей между i -м и k -м ЭРЭ (элемент матрицы C), d_{ik} - расстояние между позициями установки ЭС, определяемое по следующей формуле:

$$d_{ik} = \sqrt{((x_i - x_k)^2 + (y_i - y_k)^2)}$$

Нормирование суммарной длины соединений производится вычисляя отношение $L_{\sum}(z_j)$ к L_{\max} :

$$L_z(\sum j) = L_{z_{\sum(j)}} / L_{\max}$$

Общая площадь перекрытия:

$$S_z \sum j = \sum_{i=1}^n \sum_{k=1}^n S_{ki}$$

При разработке генетического алгоритма размещения и его реализации должны быть решены следующие задачи:

1. Выбрать существующую или описать новую модель генетического алгоритма.
2. Разработать целевую функцию;
3. Определить способ представления решения;
4. Выявить начальную популяцию возможных решений;
5. Выбрать операторы изменений, либо разработать новые операторы.
6. Определить правила отбора особей в новую популяцию.
7. Определить способ завершения генетического поиска [2].

Задачи размещения ЭС на монтажном пространстве могут быть решены с помощью генетического поиска. Применение данного метода возможно при решении задач со многими локальными экстремумами и большим пространством поиска, где невозможно применить полный перебор. Адаптация ГА к задачам размещения позволяет получить решения, которые не уступают решениям, полученным при использовании современных САПР ЭС.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Goldberd David E. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company, Inc. 1989, 412p.
2. S.A. Bushin, V.V. Kureychik Different-dimension elements placement genetic algorithm Электронный ресурс: [<https://cyberleninka.ru/article/n/geneticheskiy-algoritm-razmescheniya-raznogabaritnyh-elementov>]
3. Растрингин Л.А. «Статистические методы поиска». М: Наука, 1968.
4. Holland John H., Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology, Control, and Artificial Intelligence. University of Michigan , 1975.

УДК 004.932

А.И. КУЗНЕЦОВ, Н.В. КОКЛЮКОВ

Рязанский государственный радиотехнический университет

ОБЗОР МЕТОДОВ И АЛГОРИТМОВ ОБРАБОТКИ И АНАЛИЗА ИЗОБРАЖЕНИЙ

В данной статье проводится обзор методов FAST, FAST-ER, ORB, которые используются для нахождения особых точек изображения, также будет рассмотрен алгоритм BRIEF, благодаря которому вычисляются дескрипторы особых точек. Описывается область применения, плюсы и минусы данных методов.

Методы поиска особых точек являются важной технологией обработки изображений и широко применяется для медицины, распознавания образов, дистанционного зондирования и компьютерного зрения. Особой точкой изображения называется точка изображения, окрестность которой $o(m)$ отличима от окрестности любой другой точки изображения $o(n)$ в некоторой окрестности особой точки $o_2(m)$. В качестве окрестности точки изображения для большинства алгоритмов берётся прямоугольное окно размером 5×5 пикселей. Процесс определения особых точек достигается путём использования дескриптора и детектора.

Детектором называется метод извлечения особых точек изображения. Детектор обеспечивает инвариантность нахождения одних и тех же особых точек относительно преобразований изображений.

Идентификатор особой точки, выделяющий её из остального множества особых точек называется дескриптором. Он должен обеспечивать неизменяемость нахождения соответствия между особыми точками относительно преобразований изображений [1].

FAST

FAST – один из первых эвристических методов поиска особых точек, который завоевал большую популярность из-за своей вычислительной эффективности. Был предложен в 2005 году [1]. Для принятия решения о том, считать заданную точку C особой или нет, в этом методе рассматривается яркость пикселей на окружности с центром в точке C и радиусом 3:

Сравнивая яркости пикселей окружности с яркостью центра C , получаем для каждого три возможных исхода (светлее, темнее, похоже):

$$\begin{aligned} I_p &> I_c + t \\ I_p &< I_c - t \\ I_c - t &< I_p < I_c + t \end{aligned} \quad (1)$$

Где I – яркость пикселей, t – некоторый заранее фиксированный порог по яркости. Если на кругу есть 12 пикселей подряд, которые темнее, чем центр, или 12 пикселей, которые светлее, тогда точка помечается как особая.

Необходимо проверить примерно девять точек для принятия решения. Для ускорения процесса, авторы предложили сначала проверить только пиксели 1, 5, 9, 13. Если среди них есть более темные или светлые три пикселя, то выполняется проверка по 16 точкам, иначе – точка помечается «не особой». Это сильно сокращает время работы.

FAST-ER

FAST имеет просчёты: около некоторой окрестности может обнаружиться несколько особых точек. Эффективность алгоритма зависит от распределения пикселей и от порядка обработки.

В [2] авторы вводят улучшения для исходного алгоритма: использование машинного обучения для определения особых точек. На одной и той же сцене, рассматриваемой с разных углов, присутствуют особые точки одних и тех же объектов. В данном алгоритме используется кольцо окружности более чем 1 пиксель. Авторы используют алгоритм ID3 для проверки, является ли точка-кандидат особой, с помощью дерева решений. Этот алгоритм оптимизирует порядок, в котором обрабатываются пиксели. На основе этого получается наиболее вычислительно эффективный детектор. Функция стоимости дерева решений выглядит так:

$$\text{cost} = (k_R + R^{-2})(k_N + N^{-2})(k_S + S^{-2}), \quad (2)$$

где R – мера повторяемости, N – количество особых точек, S – число узлов в дереве.

FAST-ER медленнее, чем FAST, но точнее. FAST-ER детектор наилучший в отношении к свойству повторяемости.

Дескриптор BRIEF

После того, как особые точки найдены, вычисляют их дескрипторы – признаки, характеризующие у каждой особой точки её окрестность. BRIEF [3] – быстрый эвристический дескриптор, строится из 256 бинарных сравнений между яркостями пикселей на размытом изображении. Бинарный тест между точками x и y определяется так:

$$\tau(P, x, y) = \begin{cases} 1 : p(x) < p(y) \\ 0 : p(x) \geq p(y) \end{cases} \quad (3)$$

Выбираются точки случайным образом Гауссовским распределением вокруг центрального пиксела. Эта случайная последовательность точек выбирается только один раз и далее остается неизменной. Размер окрестности 31×31 пикселей, а апертура размытия 5.

Полученный бинарный дескриптор быстро вычисляется и сравнивается, устойчив к сменам освещения, перспективному искажению, но неустойчив к вращению в плоскости изображения.

ORB – Oriented FAST and Rotated BRIEF

В алгоритме ORB [4] сделана попытка улучшить производительность при повороте изображения. Было предложено сначала вычислять ориентацию особой точки, а затем проводить бинарные сравнения в соответствии с этой ориентацией. Работает алгоритм так:

1) Обнаруживает особые точки при помощи алгоритма FAST-ER на изображениях из пирамиды уменьшенных изображений и на исходном изображении.

2) Вычисляет для них меру Харриса. Кандидаты с низким значением меры Харриса отбрасываются.

3) Вычисляет угол ориентации особой точки. Для этого, сначала вычисляются моменты яркости для окрестности особой точки:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (4)$$

Где x, y – пиксельные координаты, I – яркость. А затем угол ориентации особой точки:

$$\theta = a \tan 2(m_{01}, m_{10}) \quad (5)$$

Всё это авторы назвали «центроид ориентации». В конечном счёте, получаем для окрестности особой точки некоторое направление.

4) В соответствии с этим углом поворачивает последовательность точек для бинарных сравнений в дескрипторе BRIEF. Новые положения для точек бинарных тестов вычисляются так:

$$\begin{pmatrix} x'_i \\ y'_i \end{pmatrix} = R(\theta) * \begin{pmatrix} x_i \\ y_i \end{pmatrix} \quad (6)$$

5) По полученным точкам вычисляется бинарный дескриптор BRIEF

После того как все особые точки доходят до нулевого угла, случайный выбор бинарных сравнений в дескрипторе перестаёт быть случайным. Это приводит к тому, что некоторые бинарные сравнения оказываются зависимыми между собой и их среднее уже не равно 0.5 (1 – светлее, 0 – темнее, когда выбор был случайным, то в среднем было 0.5) и всё это существенно уменьшает способность дескриптора различать особые точки между собой. Для того что бы решить эту проблему нужно выбрать «правильные» бинарные тесты в процессе обучения. Алгоритм поиска «правильных» тестов такой:

- 1) Вычислить результат всех тестов для всех особых точек
 - 2) Упорядочить всё множество тестов по их дистанции от среднего 0.5
 - 3) Создать список, который будет содержать отобранные «правильные» тесты, список R.
 - 4) Добавить в R первый тест из отсортированного множества
 - 5) Взять следующий тест и сравнивать его со всеми тестами в R.
- Если корреляция больше пороговой, то отбрасываем новый тест, в противном случае – добавляем.
- 6) Повторяем п. 5 пока не наберется нужное число тестов.

Тесты отбираются так, чтобы корреляция между разными тестами была минимальна, а также среднее значение этих тестов было как можно ближе к 0.5, с другой стороны, чтобы корреляция между разными тестами была минимальна.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Edward Rosten and Tom Drummond: "Fusing Points and Lines for High Performance Tracking", Department of Engineering, University of Cambridge, UK, 2006.
2. Edward Rosten, Reid Porter and Tom Drummond: "Faster and better: a machine learning approach to corner detection", Department of Engineering, University of Cambridge, UK, 2008.
3. Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua: "BRIEF: Binary Robust Independent Elementary Features", CVLab, EPFL, Lausanne, Switzerland, 2013.

4. Ethan Rublee Vincent Rabaud Kurt Konolige Gary Bradski: "ORB: an efficient alternative to SIFT or SURF", Willow Garage, Menlo Park, California, 2006.

УДК 623.4.084.2

А.М. ЛЕБЕДЕВ

Рязанский государственный радиотехнический университет

ВОЗДЕЙСТВИЕ ИЗЛУЧЕНИЯ ЗВУКОВОЙ ЧАСТОТЫ НА ПОКАЗАНИЯ MEMS АКСЕЛЕРОМЕТРОВ

Выполнено теоретическое и практическое исследование воздействия акустической помехи на показания MEMS акселерометра ADXL 345 фирмы Analog Devices.

MEMS акселерометры и датчики угловой скорости часто применяются в беспилотных летательных аппаратах (БПЛА) в качестве чувствительного элемента контура автопилота для стабилизации в плоскости горизонта, то для решения задач противодействия БПЛА представляет практический интерес возможность преднамеренного искажения показаний таких датчиков.

Уязвимость MEMS датчиков связана с их микроструктурой. Рассмотрим это на примере сенсора на основе двухстороннего камертонного преобразователя – ДКП (в англоязычной литературе – *double ended tuning fork transducer*, DETF). По сути, ДКП является структурой, чувствительной к силе, преобразующей осевую нагрузку в смещение резонансной частоты. Основой ДКП являются две длинные параллельные балки, прикрепленные к перфорированным блокам с зубцами на концах (см. рис. 1). Меняя нагрузку на балки, т.е. прикладываемую к ним силу F , можно менять резонансную частоту такой структуры. Т-образные конструкции, прикрепленные к центру параллельных балок, предназначены для преобразования механических перемещений в электрический сигнал.

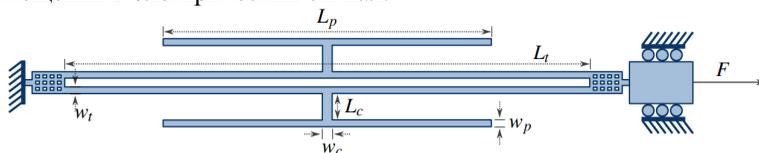


Рис. 1 Структура ДКП.

На рис. 2 представлено устройство преобразователя ускорения в напряжение: позиции 1 и 3 – электроды, с которых снимается выходной сигнал (напряжение V_{out}), позиция 2 – электрод, к которому приложено входное напряжение V_{in} . Указанные электроды являются неподвижными (статорами). Подвижный элемент MEMS акселерометра (ротор) может перемещаться, совершая поступательные движения вдоль оси y .

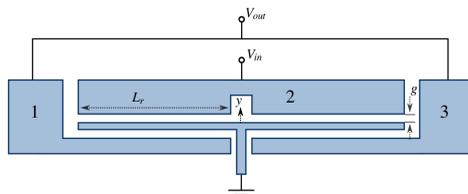


Рис. 2. Пояснение к принципу преобразования ускорения в напряжение.

При смещении ротора емкости дифференциальных конденсаторов, образуемых между ним и статорами, изменяются и через мостовую схему (на рис. 2 не показана) преобразуются в выходное напряжение [1].

ДКП имеет частоту механического резонанса, которая рассчитывается по формуле:

$$\omega_0 = \sqrt{\frac{192EI/L_t^3 - 4L_r t \varepsilon_0 V_b^2 / g^3 + 2.4F/L_t}{0.3429\rho L_t w_t t + m_c}}, \quad (1)$$

где ω_0 – частота собственных колебаний акселерометра, $I = w_t^3 t / 12$, E – модуль Юнга зубца ДКП, I – его момент инерции, ρ – плотность материала ДКП, m_c – масса считывающего электрода, F – осевая сила растяжения на структуре ДКП, V_b – постоянное напряжение на считывающих электродах, L_t , w_t , L_r , g – линейные размеры электродов, приведенные на рис. 1 и 2, t – толщина электродов.

В [2] получено аналитическое выражение для передаточной функции MEMS акселерометра:

$$W(p) = \frac{1}{p^2 + 2\xi\omega_0 p + \omega_0^2}, \quad (2)$$

где ξ – коэффициент затухания. АЧХ звена второго порядка вида (2) имеет подъем на частоте ω_0 . Для большинства популярных MEMS акселерометров потребительского сегмента значение ω_0 лежит в пределах 2..6 кГц [3]. Следовательно, если на MEMS акселерометр оказать принудительное воздействие на его резонансной частоте, то можно

добиться искажения измерения ускорения. Результаты смещения измеренного ускорения при воздействии звуковых волн интенсивностью 120 дБА для некоторых MEMS сенсоров приведены в [3] (см. рис. 3).

Автором проведено экспериментальное исследование уязвимости MEMS акселерометра на примере трехосного сенсора ADXL 345 Analog Devices: средствами языка описания аппаратуры VHDL на отладочной плате Altera DE1 Board реализованы цифровой конечный автомат для обмена данными по интерфейсу SPI [4] и модуль семисегментных индикаторов, позволяющий выводить измеряемое ускорение по каждой из осей чувствительности в долях g.

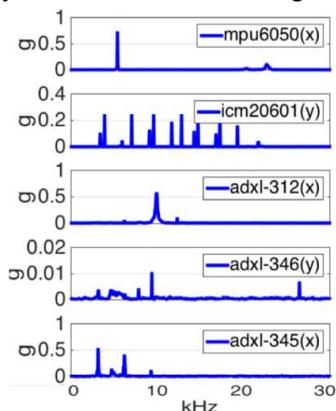


Рис. 3. Ошибка измерения ускорения при действии акустических помех.



Рис. 4. Общий вид макета для исследования.

Результаты проведенного натурного эксперимента показали, что воздействие помехи на частоте 5,6 кГц приводит к ошибке измерения ускорения от 0,02g до 0,46g при уровне помехи от 85 до 120 дБА.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Матвеев В.В., Распопов В.Я. Основы построения бесплатформенных инерциальных навигационных систем / под общ. ред. В.Я. Распопова. – СПб.: ГНЦ РФ ОАО «Концерн «ЦНИИ «Электроприбор», 2009. – 280 с.

2. Павлов Д.В., Лукин К.Г., Петров М.Н. Разработка математической модели MEMS-акселерометра // Вестник новгородского государственного университета. Серия Технические науки. – 2015. – № 8 (91). – С. 22-25.

3. Trippel T., Weisse O., Xu W. at al. WALNUT: Waging Doubt on the Integrity of MEMS Accelerometers with Acoustic Injection Attacks // IEEE European Symp. on Security and Privacy. – 2017. – P. 3-18.

4. Новицкий А. Синхронный последовательный интерфейс SPI в микроконтроллерах «от А до Я» и его реализация на примере ADuC70xx фирмы Analog Devices // Компоненты и технологии. – 2009. – № 3. – С. 53-60.

УДК 631.768

И.А. ЛЕБЕДЕВ

Рязанский государственный радиотехнический университет

ПРОЕКТИРОВАНИЕ ПОРТАТИВНОГО ЛОКАТОРА ВЫСТРЕЛА

Рассматривается внутренняя структура портативного локатора выстрела, описывается принцип его работы. На основе теории распространения баллистической и дульной волны в газообразной среде строится математической модель расчета координат источника звука, рассматривается возможный вариант реализации данного устройства.

На вооружении Российской армии имеются звукометрические комплексы для определения местоположения огневых позиций объектов полевой артиллерии, а для акустической пеленгации выстрелов из стрелкового оружия отсутствуют [1]. В связи с этим возникает необходимость в разработке малогабаритной системы обнаружения стрелка, которое военнослужащий сможет переносить на себе без особых усилий.

Для пояснения работы устройства рассмотрим следующую схему (см. рис. 1).

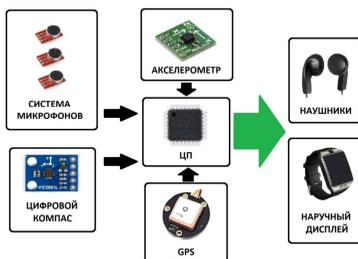


Рис. 1. Конструктивная схема устройства.

Вычислительным ядром является центральный процессор обрабатывающий информацию с подключенных устройств: микрофонов, цифрового компаса, акселерометра и GPS-датчика. В результате математических операций на выходе ЦП появляется информация о положении стрелка и выводится в удобной форме: на наушники и/или на наручный дисплей.

В качестве обрабатываемой информации используется время прихода конусной (баллистической) волны от пули (1) и сферической (дульной) волны (2) от порохового заряда (см. рис. 2).

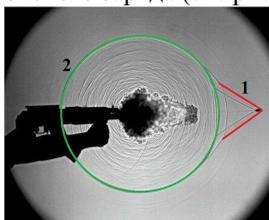


Рис. 2. Баллистическая и дульная волна.

По пиковым значениям громкости звука определяется момент прихода волны на каждый из трех микрофонов t_1 , t_2 , t_3 .

Для получения математической модели перейдем к решению геометрической задачи (см. рис. 3).

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. alternathistory.com/akusticheskie-sistemy-obnaruzheniya-ognysova-slyshit-pulyu/

УДК 004.422

Н.С. ЛЕБЕДЕВА, А.М. ГОСТИН, А.Н. САПРЫКИН
Рязанский государственный радиотехнический университет

РАЗРАБОТКА МОДУЛЯ АНКЕТИРОВАНИЯ ОБРАЗОВАТЕЛЬНОГО ПОРТАЛА РГРТУ

Рассматриваются принципы разработки программного модуля, встраиваемого в Web-сайт образовательного портала РГРТУ и позволяющего проводить онлайн анкетирование как среди обучающихся, так и среди научно-педагогических работников.

Образовательный портал РГРТУ нуждается в разработке качественного и отвечающего требованиям данной системы модуля анкетирования.

Модуль анкетирования необходим для определения качества образовательной деятельности и подготовки обучающихся по программе бакалавриата в рамках систем внутренней и внешней оценки, в которой РГРТУ принимает участие на добровольной основе.

В рамках внутренней системы оценки качества образовательной деятельности и подготовки обучающихся по программе бакалавриата обучающимся предоставляется возможность оценивания условий, содержания, организации и качества образовательного процесса в целом и отдельных дисциплин и практик.

Внешняя оценка качества образовательной деятельности по программе бакалавриата может осуществляться в рамках профессионально-общественной аккредитации, проводимой работодателями, их объединениями, а также уполномоченными ими организациями.

В качестве примера анкетирование, проводимое во время государственной аккредитации образовательной деятельности, включает в себя:

1. Анкетирование обучающихся по вопросам удовлетворенности качеством получаемых образовательных услуг. В анкетировании принимают участие все обучающиеся (не менее 80% от контингента по программе). Если курс составляет от 50 человек и более, эксперт может воспользоваться электронными анкетами. Результаты анкетирования оформляются в виде протокола, подписываются экспертом и пред-

ставителем ОО, ответственным за проведение государственной аккредитации программ образовательной организации (подпись представителя заверяется ОО) и прикладываются к отчету эксперта.

2. Анкетирование научно-педагогических работников (НПР) по вопросам удовлетворенности условиями организации образовательного процесса. В анкетировании принимают участие НПР, которые участвуют в реализации аккредитуемой программы. Результаты анкетирования оформляются протоколом, подписываются экспертом и представителем ОО, ответственным за проведение государственной аккредитации программ образовательной организации (подпись представителя заверяется ОО) и прикладываются к отчету эксперта.

Результаты обеих групп оформляются протоколом.

Ответом на каждый из вопросов является оценка критерия по 5-балльной шкале, где 5 – очень высокий уровень... 1 – крайне низкий уровень критерия.

В свою очередь все вопросы сгруппированы по разделам, по каждой из которых подсчитывается количество баллов. В зависимости от количества вопросов в каждом разделе максимальное количество баллов может меняться.

В протоколе указываются средний балл по каждому вопросу и суммарный средний балл по разделу. А так же подсчитываются проценты результатов анкетирования.

Подсчёт количества баллов можно представить следующей формулой:

$$R_i = \sum_{j=1}^n N_j * C_j$$

Где R_i – сумма баллов для i -того вопроса, N_j – количество людей, поставивших оценку C_j , C_j – оценка, поставленная людьми N_j

Расчёт среднего балла по каждому вопросу можно представить в виде формулы:

$$R_{cp} = \frac{1}{n} * R_i$$

где n – количество людей, участвовавших в данном опросе.

Результаты анкетирования в виде процентов в каждом вопросе и в каждом разделе вычисляются следующим образом:

$$A = \frac{S}{S_{max}} * 100,$$

где S – это фактический средний балл по вопросу/разделу, S_{max} – максимальный средний балл по вопросу/разделу.

Теперь рассмотрим используемые технологии для разработки программного модуля анкетирования следующие:

Языки web-программирования: PHP, JavaScript, HTML, CSS

Субд: MySQL

Web-сервер: Apache

Редакторы исходного кода: PHPStorm

Система ЭИОС Деканат, в которую будет встраиваться разработанный модуль анкетирования, реализована с использованием фреймворка «Pandora», разработанного в ЦНИТ РГРТУ.

В состав этого фреймворка входят шаблонизатор «Twig» и объектно-реляционная модель данных «Eloquent»

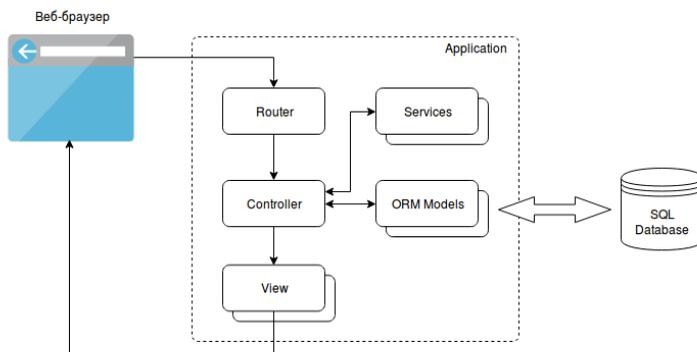


Рис. 1. Общая схема работы фреймворка Pandora.

Пример того, как будет выглядеть данная система представлен на рис. 2.

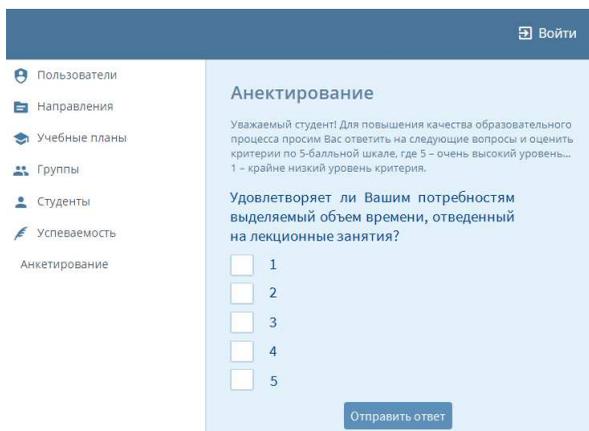


Рис. 2. Вид окна модуля анкетирования.

Для удобства просмотра результатов используются диаграммы с процентным соотношением.

Например: 30% студентов имеют полную удовлетворённость, 60% студентов - частичную удовлетворённость, 10% студентов – частичную неудовлетворённость и 0% студентов – просто неудовлетворённость.



Рис. 3. Пример диаграммы просмотра результатов анкетирования.

Удовлетворённость определяется следующими показателями:

Неудовлетворённость – До 50%

Частичная неудовлетворённость – От 50% до 65%

Частичная удовлетворённость – От 65% до 80%

Полная удовлетворённость – От 80% до 100%

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Кузнецов М.В., Симдянов И.В. Объектно-ориентированное программирование на РНР. – Санкт-Петербург: БХВ-Петербург, 2007.
2. Алла Холматова (Alla Kholmatova) – Design Systems, 2017

УДК 681.51

В.Ю. ЛИКУЧЕВ

Рязанский государственный радиотехнический университет

ПРОГРАММНАЯ АРХИТЕКТУРА СИСТЕМЫ УПРАВЛЕНИЯ БЕСПИЛОТНЫМИ ЛЕТАТЕЛЬНЫМИ АППАРАТАМИ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ

В статье рассматриваются типовые программные архитектуры систем управления БЛА, а также предлагается вариант архитектуры для решения задачи группового функционирования БЛА.

В процессе проектирования беспилотных летательных аппаратов первостепенным является этап синтеза системы управления, который должен осуществляться не только с учетом требований технического задания, но также условий конкретной цели применения. Программная часть, представляющая собой комплекс алгоритмов, каждый из которых реализует то или управляющее звено САУ (либо контур САУ), является как бы функциональной основой всей системы и определяет механизм управления целиком. Разработка программной архитектуры – сложный процесс, где каждая модель многократно испытывается, отлаживается, а только потом внедряется в реальную физическую систему. Укрупненная схема типовой программной архитектуры системы управления БЛА представлена на рис. 1.

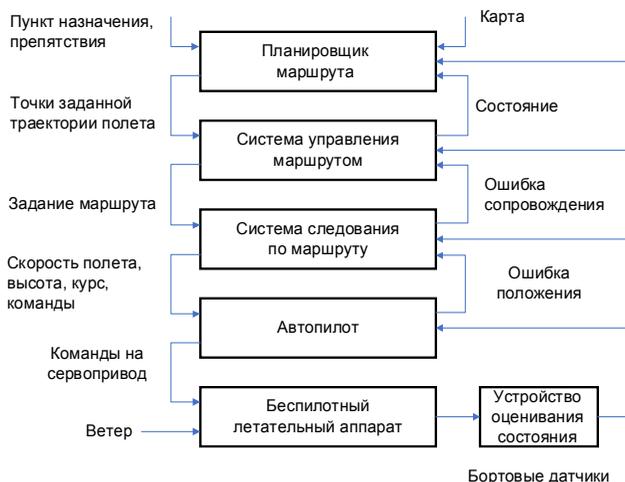


Рис. 1. Программная архитектура системы управления БЛА

Блок, обозначенный как *беспилотный летательный аппарат*, представляет собой физический объект с набором характеристик и несколькими степенями свободы. Его подвижные части отвечают на команды приводов, автоматически корректирующих свое состояние через отрицательную обратную связь (сервоприводы), а также на внешние воздействия среды (порывы ветра и т.д.).

Планировщик генерирует последовательность прямолинейных участков маршрута, либо траекторий, построенных по теореме Дубинса об оптимальных пространственных переходах объекта с фиксированным радиусом поворота.

Система управления маршрутом преобразует последовательность участков маршрута в систему прямолинейных отрезков и дуг окружностей, тем самым адаптируя траекторию полета.

Система следования по маршруту генерирует команды, регулирующие положение корпуса аппарата.

Блок *автопилота*, вырабатывая команды для сервоприводов, поддерживает постоянными значения полетных углов (крена, тангажа, рысканья) скорости полета и высоты.

Все блоки в процессе функционирования руководствуются точными оценками пространственного состояния БЛА, получаемыми путем динамической фильтрации показаний бортовых датчиков – акселерометров, гироскопов, датчиков давления, магнитометров и приемников GPS. В основу механизма фильтрации положен алгоритм Кал-

мана, который оценивает состояние по совокупности текущих измерений и известному предыдущему состоянию фильтра. Таким образом, например, происходит постоянная дооценка вектора координат положения БЛА по данным со спутникового приемника и проинтегрированным угловым ускорениям, снятых с акселерометров.

Для решения многих задач средствами БЛА требуется наличие бортовой оптоэлектронной или инфракрасной камеры. Она используется не только для обеспечения наземных объектов видеоинформацией, но и для целей наведения и управления полетом. Программная архитектура системы управления, использующей данные бортовой видеокамеры представлена на рис. 2.



Рис. 2. Архитектура, включающая видеосистему

Разбирая проблему группового функционирования БЛА в целях противодействия объектам противника, в первую очередь нужно говорить о необходимой степени автономности группы. Радиоэлектронные помехи могут усложнить процесс выделения полезного сигнала с навигационных систем относительного принципа действия (GPS-приемников), а также с наземных пунктов управления, тем самым дестабилизировать работу группы. Аппараты, оснащенные системой управления, которая опирается только на комбинирование инерциального и спутникового методов навигации, способны выполнять лишь

первую часть задачи, то есть полет к месту формируемой группы. На последующих этапах (построение группы, функционирование в составе группы) система управления должна обеспечивать еще более высокую и постоянную во времени точность регулирования пространственных состояний единиц группы (их координат, углов наклона, скорости), и, следовательно, устойчивость всей группы, а для этого необходимо по возможности минимизировать долю внешнего регулирования, или, другими словами, приблизить группу к самоорганизующейся.

Здесь, как средство наведения целесообразно использовать бортовую камеру, а в качестве объектов наведения принять соседние БЛА. Идентифицируя близко-летащие единицы в плоскости изображения, система определяет расстояние до них и генерирует команды для сервоприводов, выполняя «сближение/отдаление» относительно этих единиц группы.

За счет того, что принципы наведения с помощью камеры взаимодействуют с программной архитектурой так же, как в случае использования планировщика маршрута, система в процессе работы может легко менять свою конфигурацию для выполнения разных частей задачи. Так, для целей взлета, движения к месту формирования группы и запланированной посадки система управления БЛА использует заложенный маршрут, регулируя свое положение с помощью инерциальных датчиков и приемника GPS, когда как в процессе маневрирования в составе группы – использует камеру для наведения по положению соседних БЛА. Модульность представленной архитектуры (см. рис. 3) является ее одной из самых привлекательных особенностей.



Рис. 3. Архитектура с возможностью переконфигурирования

Проблема группового функционирования БЛА открывает новые направления в разработке беспилотных авиационных комплексов, которые включают в себя не только сами аппараты, но и все вспомогательное оборудование: датчики, микроконтроллеры, программное обеспечение, компьютеры наземных станций, пользовательские интерфейсы и средства навигации. Система управления БЛА является ключевой составляющей комплекса. Разработка такой системы применительно к специальной задаче группового функционирования аппаратов требует основательного подхода с применением знаний во многих научно-технических областях. Актуальность проблемы только побуждает к проведению дальнейших исследований.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Рэндал У.Биард, Тимоти У. МакЛэйн Малые беспилотные летательные аппараты: теория и практика. Москва: ТЕХНОСФЕРА, 2018. – 312 с.

2. Ликучев В. Ю. Специальная задача для группы беспилотных летательных аппаратов//Информационные технологии, межвузовский сборник научных трудов. – Рязань: Рязанский государственный радиотехнический университет, 2018. – 136с.

УДК 621.3.049.75

В.А. МАКАРОВА, И.С. НОВИКОВ

Рязанский государственный радиотехнический университет

ИССЛЕДОВАНИЕ СРЕДЫ ОБИТАНИЯ АКВАРИУМНЫХ РЫБ И БЕСПОЗВОНОЧНЫХ ДЛЯ СОЗДАНИЯ УСТРОЙСТВА ЖИЗНЕОБЕСПЕЧЕНИЯ

Рассматривается создание конструкции устройства функционирования и жизнеобеспечения для аквариумных рыб и беспозвоночных.

В настоящее время происходит широкое внедрение и развитие информационных технологий во все сферы деятельности человека. Если в помещении есть аквариум, он, несомненно, главный элемент в интерьере. Бесшумный и красивый мир аквариума гармонизирует и делает уютней любое пространство. Зная об этом, владельцы предприятий, заведений зачастую используют этот прием, чтобы привлекать посетителей. Поэтому оборудование для аквариума является, непременно актуальной темой.

Система жизнеобеспечения - комплекс функционально связанных систем и средств, служащих для обеспечения жизнедеятельности, функционирования, а также работоспособности системы устройства. Система жизнеобеспечения поддерживает в замкнутом пространстве заданные физические параметры (давление, температуру, влажность и др.) среды обитания аквариумных рыб и беспозвоночных.

Любую систему жизнеобеспечения характеризуют три основных свойства:

- круг задач, которые призвана решать система жизнеобеспечения;
- производительность системы;
- качество её компонентов.

При разработке конструкции устройства возникает необходимость экономической оценки и обоснования экономической целесообразности проекта. Это обусловлено сильной взаимосвязью технического прогресса и экономики.

Корпус устройства выполнен на единой конструктивно-технологической базе и имеет унифицированный интерфейс. Тем самым снижаются затраты на разработку конструкции.

Использование прогрессивной компонентной базы в схемных решениях и повышение технологичности конструкции приводят к снижению стоимости аппаратуры. Также в конструкции используются дешевые и легкодоступные материалы.

Данное устройство должно осуществлять управление и контроль основных параметров аквариума.

Принципиальная электрическая схема упрощена (см. рис. 1), а число внешних устройств сокращено до минимума: датчика температуры воды, ее подогревателя, воздушного компрессора и раздатчика корма.

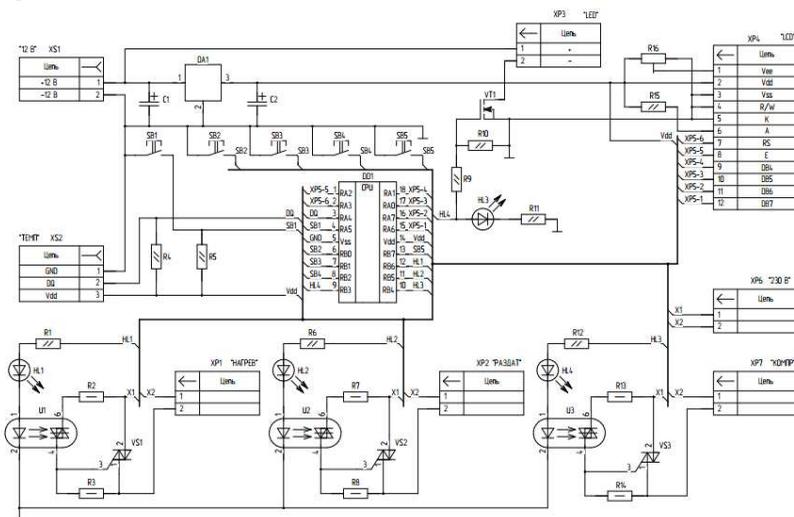


Рис. 1. Схема электрическая принципиальная.

Датчик температуры ВК1 программно настроен на изменение температуры в интервале от 0 до +99°C с шагом 1°C. Этот датчик установлен внутри аквариума.

Программа микроконтроллера отображает измеренную датчиком ВК1 температуру воды аквариума на экране ЖКИ HG1, обновляя показания каждую секунду. Далее программа микроконтроллера сравнивает эту температуру с заданной, введенной в память микроконтроллера. Пока текущая температура меньше заданной, программа поддерживает напряжение высокого логического уровня на выходе RB6 мик-

роконтроллера. В результате через резистор R, светодиод HL1 и излучающий ИК-диод оптрона U1 течет ток. Излучение ИК-диода позволяет фотосимистору оптрона U1 открываться в каждом полупериоде сетевого напряжения, а вместе с ним открывается и симистор VS1, управляющий нагревателем воды в аквариуме.

Как только текущая температура, достигает заданной, программа меняет высокий логический уровень напряжения на выходе RB6 микроконтроллера низким. Ток через излучающий диод оптрона U1 прекращается, поэтому фотосимистор оптрона и мощный симистор больше не открываются. Теперь нагреватель выключен. Температура воды начинает уменьшаться, и как только она становится меньше заданной, высокий логический уровень напряжения на выходе RB6 микроконтроллера вновь включает нагреватель. За счет этого температура воды в аквариуме стабилизируется около заданного значения.

Для микроконтроллера DD1 на плате (см. рис. 2) должна быть установлена панель, куда микроконтроллер вставляют уже запрограммированным.

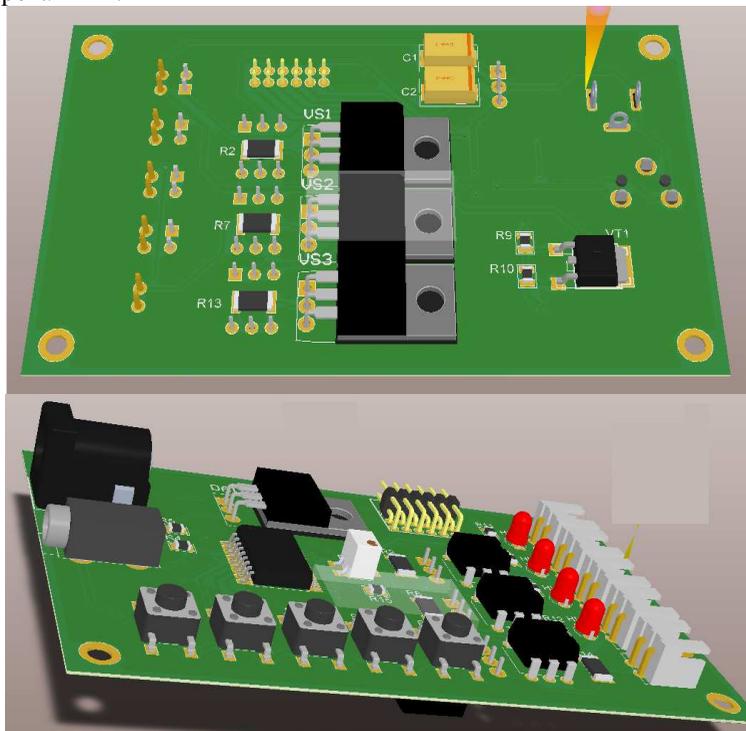


Рис. 2. Печатная плата.

Описанный механизм пригоден не только для управления аквариумом. На его основе можно изготовить, например, симулятор присутствия людей в помещении, систематически включающий и выключающий освещение и бытовые приборы. По сигналу датчика температуры можно автоматически открывать и закрывать теплицы или включать и выключать отопление в доме.

Корпус можно (см. рис. 3) выбрать из номенклатуры покупных изделий, но требуется произвести необходимую доработку. Сделать отверстия для выводов проводов для исполнительных устройств и питания, а также прямоугольное отверстие для индикатора и отверстие для стойки крепления платы.

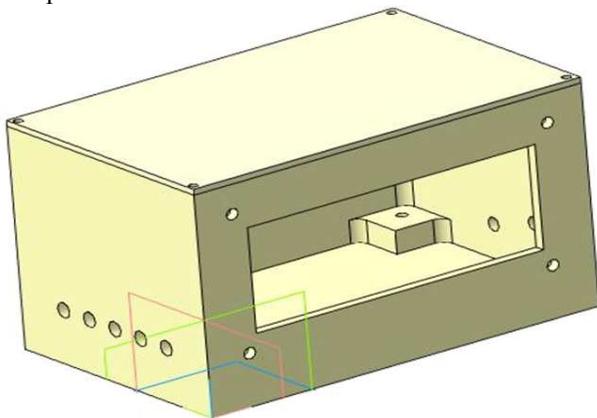


Рис. 3. Корпус устройства.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Импульсные источники питания. Схемотехника и ремонт/ Кашкаров А.П. ДМК Пресс, 2012.
2. Проектирование конструкций радиоэлектронной аппаратуры: учебное пособие для вузов / Е. М. Парфенов, Э.Н. Камышная, В. П. Усачев. - М.: Радио и связь, 1989. - 272с: ил.
3. ОСТ 4ГО.070.014. Детали радиоэлектронной аппаратуры. Общие технические условия.

УДК 621.317

А.А. МАТЮХИН

Рязанский государственный радиотехнический университет

ИСПОЛЬЗОВАНИЕ ДИАКОМ ДЛЯ ПРОВЕДЕНИЯ ДИАГНОСТИКИ ЦИФРОВЫХ УСТРОЙСТВ С ТТЛ ЛОГИКОЙ

Рассматриваются основные моменты использования ДИАКОМ для диагностики устройств.

Комплекс ДИАКОМ был создан и применяется для тестового контроля цифровых устройств. Однако в связи с его недостаточной компактностью и некоторым неудобством в использовании возник вопрос о модернизации данного комплекса.

Система контроля ДИАКОМ позволяет производить контроль сложных сменных цифровых элементов электронной аппаратуры, построенной на ТТЛ логике. Важной особенностью является применение сигнатурного анализа для проверок. В ходе разработки решалась проблема удобства использования и компактности устройства.

Сигнатурный анализ - сопоставление реальной сигнатуры (в конкретной точке устройства), отображенной на дисплее сигнатурного анализатора, с эталонной сигнатурой этой точки сигнатурный анализ основан на преобразовании длинных последовательностей двоичных сигналов в двоичное число, называемое сигнатурой. Измеряемые двоичные последовательности возбуждаются в контрольных точках МПС под действием специальной тестовой программы. Сигнатуры контрольных точек измеряются на заведомо работоспособной системе и указываются на принципиальной схеме МПС подобно тому, как на схемах аналоговых устройств указываются осциллограммы и некоторые параметры аналоговых сигналов.

В качестве управляющего компьютера в диагностическом комплексе ДИАКОМ используется персональный компьютер. Блок контроллера имеет в своем составе: Программно-управляемые устройства для генерации цифровых тестовых сигналов, цифровой осциллограф, программируемые и дополнительные источники питания тестируемого объекта диагностирования (ОД).

Данный комплекс позволяет реализовать в своей работе следующие тестовые воздействия [1]

- исчерпывающие;

- псевдослучайные;
- детерминированные;
- комбинированные.

Структурная схема подключения выглядит следующим образом – управляющий компьютер, через шину данных подключен к модулю диагностики, где содержатся – тест процессор, цифровой осциллограф, логический анализатор, к которому, с помощью коммутирующего устройства подключена плата (ОД).

На жестком диске управляющего компьютера комплекса размещены ремонтные базы данных (РБД) для всех ОД, которые когда-либо тестировались. Для ускорения поиска информации эти базы структурированы.

Тестирование может производиться в двух режимах - «стандартный» и «расширенный».

При выборе «стандартного» пути пользователь подает тест на ОД и щелчком мыши по соответствующей кнопке запускает режим сравнения реакции с эталоном. В случае несовпадения текущей реакции с эталоном, он получает таблицу с перечнем выходных контактов, на которых реакция не совпадает с эталоном. Далее пользователь запускает процесс автоматического анализа реакций ОД для выделения области «подозреваемых микросхем». Используя сигнатурный анализатор в режиме ведомого щупа и, руководствуясь указаниями типа: «Установите щуп в точку ...», пользователь локализует неисправность.

При выборе «расширенного» пути пользователь после съема реакций может не только просмотреть таблицу «Ошибки сравнения...», но и изучить полную временную диаграмму тестирования, как на контактах разъема ОД, так и в его внутренних контрольных точках, к которым подключены клипсы логического анализатора или щупы осциллографа. Установив частоту квантования логического анализатора в несколько раз выше частоты тестирования, можно «растянуть» исследуемую временную диаграмму, рассмотреть фронты сигналов и даже проверить схему на наличие глитчей – одиночных очень коротких импульсов, которые иногда могут возникать в схеме из-за временных рассогласований в цепях сигналов.

Поскольку на жестком диске располагается база РБД со всеми эталонными значениями, это позволяет ускорить процесс локализации неисправности ОД. Для начала тестирования необходимо:

- выбрать «Изделие» и ОД из РБД комплекса;

- загрузить тест и маску;
- щелкнуть по кнопке тест процессор (ТП);
- подключить ОД к коммутирующему устройству;
- установить клипсы логического анализатора (ЛА) в контролируемые точки (при работе с ЛА);
- подать питание на ОД.

После программа произведет тест ОД. Если существует эталон, то можно произвести сравнение, результатом сравнения может быть сообщение «Сравнение прошло успешно!», или на экране ТП появится таблица «Ошибки сравнения...»

№ п/п	Канал	№ Такта
1		
2	1B5	2
3	1B2	2
4	1C9	2
5	1C25	2
6	1A2	
7	1A18	

Рис. 1. Ошибка сравнения.

Программное обеспечение может локализовать неисправность ОД двумя способами:

1. По временной диаграмме и тактам, где ОД не совпадает с эталоном пользователь приступает к локализации неисправности с помощью логического анализатора и осциллографом.

2. Автоматический анализ неисправности ОД. Такой метод позволит найти неисправность с точностью до нескольких корпусов.

Такой подход значительно ускоряет процесс проверки и локализации неисправностей в ОД, что позволяет сократить время на тестирование устройства. Благодаря использованию новой конструктивной базы – сделалась возможным реализовать устройство в более компактном корпусе, что позволяет разместить его в стойке или на столе.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Руководство пользователя Диаком -3М Часть 1. Москва 2001г

2. Руководство пользователя Диаком -3М Часть 3. Москва 2001г

УДК 621.3.049.75

Е.С. МАХНАЧ

Рязанский государственный радиотехнический университет

ОРГАНИЗАЦИЯ БЕЗОПАСНОСТИ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ «УМНЫЙ ДОМ»

Рассматриваются методы реализации охранной системы с использованием технологий «Умный дом».

«Умный дом» – это система автоматизации для управления и контроля инженерными системами современного здания. Исследования последних лет показывают устойчивую тенденцию к увеличению доли стоимости и объёма таких систем в общей стоимости строительных объектов. Помимо обеспечения комфорта и безопасности целью системы «умный дом» является экономия затрачиваемых ресурсов. Система может самостоятельно выключать освещение, обнаружив отсутствие движений, регулировать отопление и кондиционирование, когда хозяина нет дома, обеспечив необходимую температуру к его возвращению. Также система исследует потребление электроэнергии того или иного прибора и в случае неисправности может предупредить аварию, обнаружив поломку.

Организация безопасности является одним из важнейших аспектов реализации системы «умного» дома. Эта технология позволяет управлять всеми охранными системами с помощью одного устройства, позволяет владельцу дистанционно отслеживать статус имущества и принимать участие в предотвращении преступления, уведомляя о несанкционированном доступе. Система может следить за состоянием газовой, водопроводной и противопожарных систем и в случае неисправностей принимать соответствующие меры. Она производит круглосуточный контроль и в случае возникновения внештатной ситуации уведомляет об этом владельца.

На рис. 1 представлена электрическая схема и печатная плата проекта системы безопасности. Он реализован с помощью специальных датчиков для обнаружения несанкционированного проникновения на объект и контролирующего устройства, которое принимает и анализирует сигналы от датчиков.

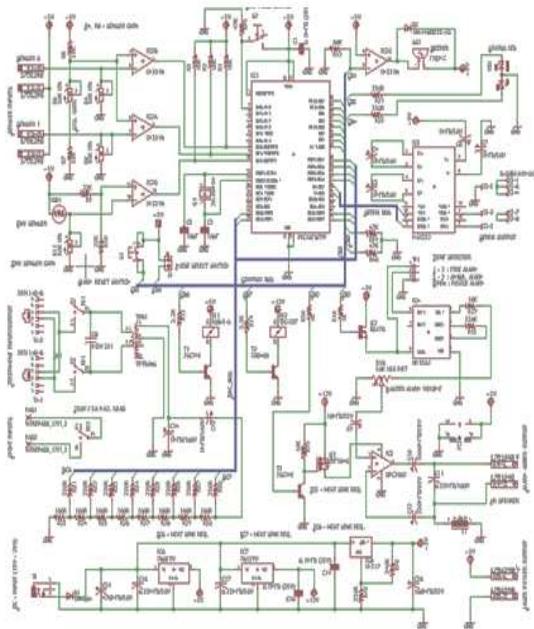


Рис. 1. Электрическая схема проекта охранной системы.

Печатная плата изготовлена из двустороннего фольгированного стеклотекстолита, размерами 100x120, с односторонним монтажом SMD и ТHT компонентов. Центром схемы контролирующего устройства является 8-разрядный микроконтроллер PIC16F877A, имеющий низкую цену и достаточно развитую аналоговую периферию. Гальваническая развязка через трансформатор обеспечивает дистанционный доступ к системе через телефонную линию.

Основными элементами охранной системы являются датчики. Система реагирует на пересечение объектом лучей, организованных приемо-передающими устройствами датчиков периметра охраны. В качестве внешних охранных датчиков используются два фоторезистора, которые реагируют на прерывание светового потока лазерного диода. Интерфейс датчиков образуют компараторы LM339, благодаря которым возможна подстройка чувствительности датчиков для режимов «День» и «Ночь». Внешние датчики и цепи управления подключаются к установленным на печатной плате коннекторам. Кроме того, для предотвращения вскрытия устройства оно имеет встроенный датчик-фоторезистор. Устройство также имеет встроенную систему зву-

кового оповещения и возможность выбора тональности и громкости звучания.

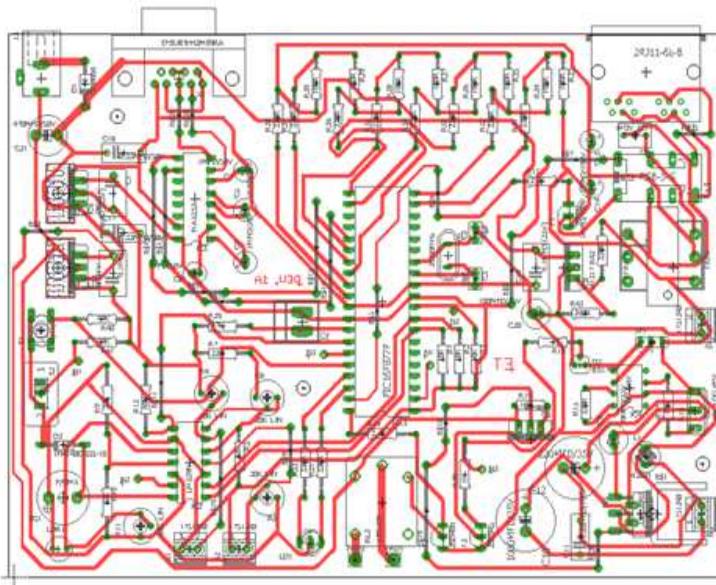


Рис. 2. Изображение печатной платы проекта охранной системы.

Подобные системы – одно из передовых достижений в технике. В России они пока не нашли широкого распространения из-за того, что стоимость иностранных систем значительно превышает экономию от ее применения, а системы отечественного производства практически отсутствуют на рынке.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Мологонцева О.Ф. Проектирование цифровых устройств на однокристалльных микроконтроллерах / В.В. Сташин, А.В. Урусов, О.Ф. Мологонцева. – М.: Энергоатомиздат, 1990. -224с.
2. Харке В.Е. Умный дом. Объединение в сеть бытовой техники и систем коммуникации в жилищном строительстве. / Харке В.Н. – М.: Техносфера, 2006. -292с.

УДК. 681.586.48

М.Р. МИРОНОВА

Рязанский государственный радиотехнический университет

ОСОБЕННОСТИ РАБОТЫ УЛЬТРАЗВУКОВЫХ ИЗМЕРИТЕЛЕЙ РАССТОЯНИЙ

Рассматриваются особенности работы ультразвуковых измерителей расстояний при различных параметрах окружающей среды.

Ультразвуковой измеритель расстояния – это устройство для измерения расстояния бесконтактным способом.

Ультразвуковой измеритель расстояния, как правило, он состоит из двух частей - приемной и передающей. Обе части обычно управляются микроконтроллером, который нужен также для дальнейшей передачи результатов измерения. Измерение расстояния происходит следующим образом: измеритель посылает короткий пучок ультразвуковых волн в направлении нужного объекта, который, отразившись, возвращается обратно. После, электронная схема производит расчет времени между моментом посылки сигнала и моментом приема отраженного сигнала.

Ультразвуковые датчики действуют по принципу эхолокации – расстояние до объекта рассчитывается на основании измерения промежутка времени между моментами посылки и приема звукового импульса и скорости звука в среде.

Формула скорости звука в воздухе при давлении равным примерно 1атм:

$$c(t) = 331 \sqrt{1 + \frac{t}{273}},$$

где t – температура (измеряется в градусах Цельсия).

Исходя из формулы видно, что скорость распространения звуковых волн снижается с понижением температуры воздуха. Ультразвуковые измерители расстояний, предназначенные для работы при низких, высоких или для расширенных диапазонов, как правило, оборудуют автоматической температурной коррекцией.

Важными критериями работы ультразвуковых измерителей расстояния являются точность измерения расстояния, а также дальность распространения звуковой волны или максимальное рабочее расстояние, на которое датчик способен производить измерение. Это расстояние зависит от затухания звука.

Распространение звуковой волны в воздухе происходит за счет продольных колебаний молекул воздуха относительно своего положения и их соприкосновения между собой.

Затухание — это уменьшение амплитуды звукового давления с увеличением расстояния от источника звука (излучателя измерителя).

Если измеряемый объект находится на расстоянии большем, чем максимальное рабочее расстояние, то отправленная и затем отражившаяся обратно волна успевает полностью поглотиться средой и измерение становится невозможным[1].

Исходя из вышесказанного можно сделать вывод о следующих параметрах среды, влияющих на рабочее расстояние измерителя:

1. Температура: с повышением температуры, затухание увеличивается и максимальное расстояние уменьшается.
2. Давление: с ростом давления затухание уменьшается и максимальное расстояние увеличивается.
3. Относительная влажность: на разных частотах, затухание будет максимальным при разной относительной влажности воздуха.
4. Частота ультразвука: чем выше рабочая частота излучения датчиком ультразвуковой волны, тем быстрее затухает сама волна.

Звук поглощается не только средой, но и при взаимодействии с самим объектом. Как только ультразвуковая волна достигает объекта часть энергии волны поглощается объектом, а часть отражается в обратную сторону (см. рис. 1).

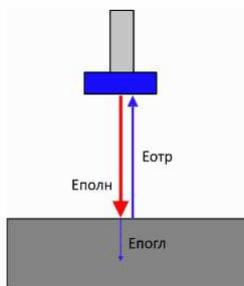


Рис. 1. Поведение звуковой волны при взаимодействии с объектом

Таким образом, чем меньше акустическое сопротивление объекта, тем большая часть энергии $E_{\text{погл}}$ проникает внутрь среды и тем сильнее уменьшается максимальное рабочее расстояние вместе с уменьшением $E_{\text{отр}}$.

Чем больше коэффициент отражения, тем большая часть энергии волны $E_{\text{отр}}$ отразится обратно и тем больше расстояние, на котором может происходить измерение.

Измерение расстояния ультразвуковым измерителем до объектов, имеющих высокое акустическое сопротивление (сталь, стекло, пластик, дерево, жидкости) при нормальных условиях окружающей среды (давление, температура, влажность) возможно на максимальное рабочее расстояние.

Измерение расстояния ультразвуковым измерителем до объектов, имеющих высокое акустическое сопротивление (шерсть, пух, пена, поролон) при нормальных условиях окружающей среды может стать невозможным даже в пределах рабочего диапазона, так как эти объекты имеют высокую степень пористости, большая часть энергии поглощается ими.

Еще одной важной особенностью работы ультразвуковых датчиков является невозможность измерения уровня на близком от датчика расстоянии. Это связано с тем, что излучатель датчика является одновременно и его приемником.

В основе датчика лежит пьезокерамический элемент, который начинает колебаться при подаче на него напряжения. В зависимости от его характеристик, он излучает ультразвуковую волну определенной частоты, после чего переходит в режим приема и ожидает возвращения отраженной от объекта волны.

Однако после прекращения подачи напряжения, в пьезокерамическом элементе наблюдаются затухающие автоколебания, которые невозможно избежать. Если объект находится слишком близко, то отраженная волна вернется к излучателю в тот момент, пока автоколебания еще не прекратились. И как следствие, измерение станет невозможным.

Слепая зона — это расстояние, которое ультразвуковая волна проходит за время полного затухания автоколебаний. Это фиксированная величина, являющаяся характеристикой конкретной модели датчика, не подлежит настройке и не зависит от параметров окружающей среды.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Особенности измерения расстояния / уровня ультразвуковыми датчиками [Электронный ресурс], URL: https://totalkip.ru/articles/osobennosti_izmereniya_rasstoyaniya_urovnya_ultrazvukovymi_datchikami (дата обращения 10.04.19).

УДК 658.5

М.С. МИХЕЕВА

Рязанский государственный радиотехнический университет

ЖИЗНЕННЫЙ ЦИКЛ РЭС И СПОСОБЫ ЕГО УВЕЛИЧЕНИЯ

Рассматривается жизненный цикл радиоэлектронных средств, его содержание, а также влияние отбраковочных испытаний на повышение надежности РЭС.

Жизненный цикл (ЖЦ) РЭС – совокупность периодов проектирования, производства, технической эксплуатации и практического использования, модернизации, а также замены (частичной или полной) по мере технического износа и «морального» старения.

Стадии ЖЦ РЭС представлены на рис. 1.

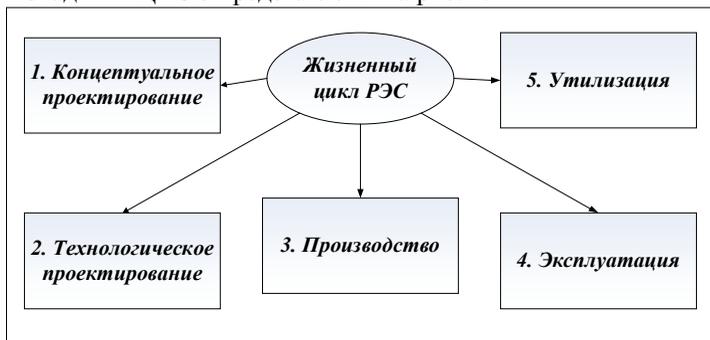


Рис. 1. Стадии жизненного цикла РЭС.

Каждая из этих стадий состоит из отдельных этапов.

1. Концептуальное проектирование.

На данной стадии определяются:

- техническое задание на проектирование концепции;
- необходимость и возможность создания РЭС;
- критерии проектирования РЭС;
- внешний вид РЭС;
- затраты на проектирование, производство, эксплуатацию и утилизацию;
- основные тактико-технические характеристики РЭС.

2. Техническое проектирование.

Данная стадия включает в себя разработку:

- технического предложения;
- эскизного проекта;

- технического проекта;
- рабочего проекта (разработка рабочей конструкторской документации опытного образца, опытной партии).

3. Серийное производство состоит из:

- технологической подготовки производства РЭС;
- изготовления, сборки и настройки РЭС;
- заводских испытаний РЭС;
- комплектации РЭС.

4. На стадии эксплуатации выполняется:

- доставка готовых изделий;
- ввод в эксплуатацию;
- приведение в установленную степень готовности к использованию;
- применение по назначению;
- техническое обслуживание и ремонт;
- хранение;
- транспортировка;
- модернизация (с последующей эксплуатацией);
- снятие с эксплуатации и утилизация.

Графическое представление стадии эксплуатации РЭС после доставки готовых изделий представлено на рис. 2.

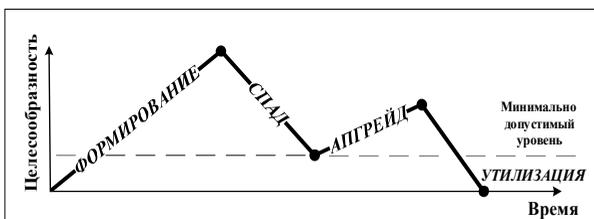


Рис. 2. Графическое представление стадии эксплуатации РЭС

В процессе эксплуатации целесообразность или практическая достижимость целей создания изделия сначала возрастает, проходя период формирования, а затем снижается, проходя периоды спада и апгрейда. В процессе модернизации некоторое время удается поднимать технические характеристики изделия до максимально возможного уровня, но затем оно настолько устаревает, что затраты на ремонт становятся несопоставимыми с получаемым эффектом, поэтому продукция снимается с эксплуатации и утилизируется. Снятие с эксплуатации осуществляют после документального оформления наступления так называемого предельного состояния объекта эксплуатации, когда

дальнейшее использование и ремонт РЭС невозможен или нецелесообразен. В этом случае производится утилизация РЭС.

Надежность РЭС при эксплуатации характеризуется кривой зависимости интенсивности отказов (λ) от времени (t), представленной на рис. 3.

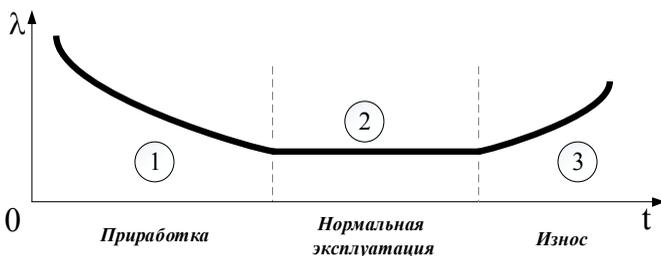


Рис. 3. Кривая зависимости интенсивности отказов от времени.

Рассматривая первый период, можно отметить высокое значение интенсивности отказов, низкое и постоянное значение во втором периоде и возрастающую интенсивность в третьем периоде.

Кривые зависимости интенсивности отказов (λ) от времени (t) изображены на рис. 4.

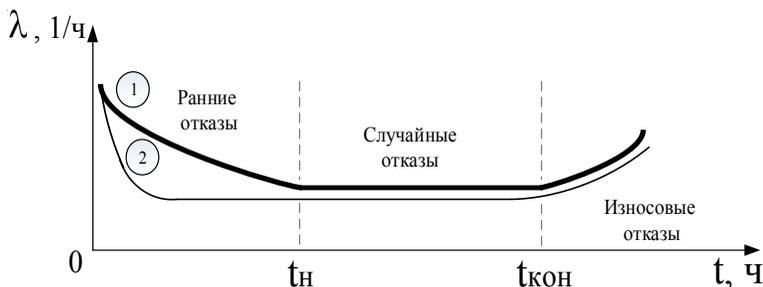


Рис. 4. Влияние отбраковочных испытаний на интенсивность отказов РЭС: (1) без проведения отбраковочных испытаний, (2) с проведением отбраковочных испытаний.

Ранние отказы возникают, как правило, вследствие конструктивных и технологических недостатков. На окончание этого этапа указывает выравнивание кривой интенсивности отказов. Интенсивность отказов в период приработки имеет тенденцию к уменьшению по мере усовершенствования конструкции и технологии.

Принято считать, что случайных отказов РЭС, без причины и следствия приложения некоторой нагрузки, не бывает. Некачественные элементы изделий, оставаясь не выявленными к началу эксплуатации, могут явиться причиной их отказа. Для повышения эффективности отбраковочных испытаний нужно знать, какие нагрузки и как ускоряют появление отказов. Внедрение отбраковочных испытаний значительно повышает средний уровень надежности РЭС.

Известны два направления увеличения надежности РЭС.

- 1) Создание бездефектной технологии.
- 2) Выявление и удаление изделий с отказами из готовых партий.

Создание бездефектной технологии считается более эффективным методом, но это не всегда возможно по различным причинам: экономическим, техническим и т.д.

Распространённым способом повышения качества и надёжности РЭС является проведение отбраковочных испытаний на этапе выходного контроля завода-изготовителя.

В общих технических (ОТУ) и технических условиях (ТУ) обычно указываются состав обязательных отбраковочных испытаний, режимы и последовательность. Как правило, заводы-изготовители РЭС в начальный период серийного производства расширяют состав этих испытаний.

Но по мере повышения управляемости и стабильности технологического процесса (сбор статистики причин отказа изделий на производстве и у потребителей, а также принятие мер по устранению причин преобладающих отказов) завод-изготовитель может сокращать время проведения отдельных испытаний, изменять их режим или вообще отменять отдельные виды отбраковочных испытаний, кроме тех, которые указываются как обязательные в ОТУ и/или ТУ.

УДК. 004.94

Ю.С. МОРОЗОВА, С.В. СКВОРЦОВ

Рязанский государственный радиотехнический университет

РАЗРАБОТКА АЛГОРИТМА МНОГОМЕРНОЙ ТРАССИРОВКИ ДЛЯ РЕШЕНИЯ ПРОЕКТНЫХ ЗАДАЧ

Рассматривается модификации алгоритма слежения за целью (модификации волнового алгоритма) для трассировки в 3D-пространстве.

Трассировка – это прокладка пути в некотором пространстве с препятствиями. Чаще всего трассировка используется для печатных плат (чтобы определять места расположения проводников на печатной плате вручную или с использованием одной из САПР, предназначенной для проектирования печатных плат). Для трассировки печатных плат было разработано множество алгоритмов и их модификации.[1]

Некоторые из этих алгоритмов можно применять не только для трассировки на плоскости, но и для поиска пути в виртуальном пространстве, для проектирования коммуникаций в многоэтажных зданиях, а также для решения других прикладных задач. Например, трассировка в условиях реальной местности актуальна в робототехнике, поскольку функции многих роботов включают перемещение на местности, а если эта местность не имеет форму выпуклой однородной геометрической фигуры, то для поиска пути на ней придётся использовать более сложный алгоритм.

В общем случае поиск пути (патфайтинг) может происходить в условиях любой структуры, которую можно представить, например, графовой моделью. Наиболее известным из применений алгоритмов патфайдинга, не связанных напрямую с местностью, является OSPF (OpenShortestPathFirst) – протокол динамической маршрутизации в сети, использующий для нахождения кратчайшего пути алгоритм Дейкстры.

Таким образом, задача построения пути трассировки в трехмерном рабочем пространстве является актуальной. Рассмотрим известные методы, которые можно использовать для ее решения.

Метод навигационного графа. Данный метод предполагает трассировку пути на графе, где вершины графа – точки трехмерного пространства, ребра графа – отрезки, соединяющие эти точки, с ценой, равной длине отрезка [2]. Задача поиска пути сводится к нахождению ближайших вершин к начальной и конечной точке, а затем к поиску пути на графе между этими вершинами с использованием критерия минимального веса общего пути (см. рис. 1). Метод хорошо подходит для 3D-пространства, но существуют и такие недостатки, как сложность задания навигационного графа, значительное время трассировки в графе с большим числом ребер.

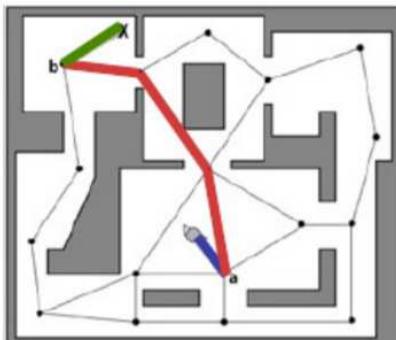


Рис. 1. Пример трассировки с помощью навигационного графа.

Метод сочетания эвристик в 3D-пространстве. Предполагает применение специального алгоритма разрешения проблемной ситуации (обход препятствия) для небольшого числа распространенных случаев. Возможным примером является попытка при столкновении с препятствием найти луч с минимальным отклонением от вектора взгляда персонажа, который бы не пересекался с полигонами 3D-мира. В случае нахождения такого луча, персонаж перемещается по нему, а затем пытается следовать к конечной точке по прямому пути (см. рис. 2). На рисунке серый прямоугольник – препятствие, черный круг – персонаж, стрелка - направление взгляда персонажа, пунктирная линия – неоптимальный путь обхода, черная линия – найденный путь. Основные недостатки этого метода [3]:

- в большом числе ситуаций метод не может найти правильный путь, так как при разрешении проблемной ситуации путь ищется локально;
- в случае столкновения с препятствием сложность разрешения проблемной ситуации может быть очень высокой и напрямую зависеть от уровня детализации пространства поиска.

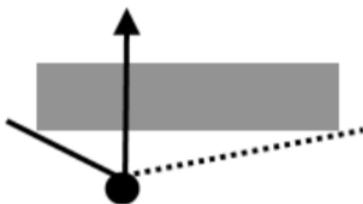


Рис. 2. Метод сочетания эвристик.

Метод navigation mesh [4]. Предполагает задание полигональной модели проходимого трехмерного пространства, в которой дейст-

вует правило, что между любыми двумя вершинами выпуклого полигона есть путь, находящийся внутри этого полигона. Метод хорошо работает как на открытых пространствах, так и в закрытых. В методе navigation mesh имеется возможность выбора использования центров полигонов, центров ребер и вершин полигонов в качестве контрольных точек пути. Центры полигонов обеспечивают разумный набор узлов для графа поиска пути (см. рис. 3).

В приведенном примере зеленая линия - идеальный путь, желтая линия – найденный путь. Кроме того, точки начала и конца добавляются наряду с точкой центра многоугольника, в котором мы находимся. С помощью этого метода трудно учитывать динамические объекты, так как это требует больших вычислительных затрат. Также нужно хранить большое количество полигонов, если трехмерное пространство имеет большое количество объектов сложной формы. Автоматическая генерация navigation mesh значительно затруднена и не может осуществляться в полной мере корректно.

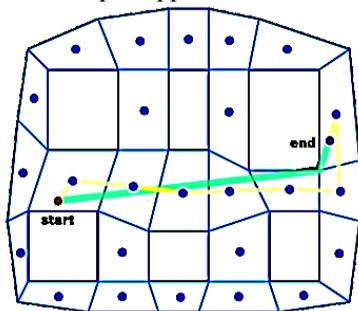


Рис. 3. Пример поиска пути методом navigation mesh с использованием центра полигонов в качестве контрольных точек пути.

Метод трассировки, основанный на разбиении пространства на ячейки одинаковой формы. Для каждой ячейки задано, проходимая она или нет. Объект, для которого идет поиск пути, занимает одну ячейку. По полученной матрице проходимости одним из последовательных алгоритмов трассировки находится путь из одной ячейки в другую. На рис. 4 изображен пример реализации данного метода, где серые круги – точки начала и конца пути, черные клетки – препятствия, окружности – найденный путь. Метод получил большое распространение в 2D-пространстве (в играх-стратегиях и т.д.), однако при применении в 3D-пространстве имеет свои недостатки:

- трехмерное пространство состоит из объектов произвольной формы, которые плохо ложатся на клеточную структуру поиска;

лучевой алгоритм. Волновой алгоритм имеет высокую трудоемкость при большой размерности задачи, а лучевой алгоритм трассировки в большинстве случаев не находит путь. Эти недостатки можно исключить, если выбрать модификацию алгоритма Рабина – алгоритм слежения за целью.

Таким образом, для решения поставленной задачи необходимо модифицировать алгоритм слежения за целью для трехмерного пространства. В этом алгоритме в качестве значений весовой функции $H(i, B)$ на этапе распространения волны используется выражение для 2D-пространства

$$H(i, B) = |x_i - x_B| + |y_i - y_B|,$$

которое определяет кратчайшее расстояние от ячейки i до конечной ячейки B ; x_i, y_i и x_B, y_B – координаты текущей ячейки i и конечной ячейки B соответственно.

Тогда для 3D-пространства формула будет иметь вид:

$$H(i, B) = |x_i - x_B| + |y_i - y_B| + |z_i - z_B|,$$

где x_i, y_i, z_i и x_B, y_B, z_B – координаты текущей ячейки i и конечной ячейки B соответственно.

Волна распространяется от множества ячеек с минимальным весом, начиная с ячейки с большим/меньшим порядковым номером.

Проведение пути с помощью алгоритма слежения за целью основано на использовании метода путевых координат. Здесь распространение волны происходит из тех ячеек очередного фронта, которые ближе расположены к конечной ячейке (запрещенные ячейки (стены) не учитываются).

Последовательность просмотра соседних ячеек будет зависеть от расположения друг относительно друга начальной и конечной ячеек. Пусть A и B – начальная и конечная ячейки пути, их координаты соответственно:

$$A = (1,10,2);$$

$$B = (0,12,1).$$

Тогда из направлений «вправо-влево» (ось X) приоритетным будет «влево» ($1 > 0$), из направлений «вверх-вниз» (ось Y) приоритетным будет «вверх» ($10 < 12$), из направлений «назад-вперед» (ось Z) – «назад» ($2 > 1$).

Теперь определим порядок просмотра соседних ячеек в трех выбранных направлениях. Чем больше разница в одном направлении между координатами начала и конца, тем меньше порядковый номер на-

правления. Если разница двух и более направлений совпадает, то выбираем стандартный порядок: вначале по оси X , потом по оси Y и, наконец, по оси Z .

Для примера, рассмотренного выше, выбираем следующий порядок направлений: «вверх-влево-назад-вниз-вправо-вперед».

Разработанная модификация алгоритма слежения за целью включает следующие шаги.

1. Определить последовательность рассмотрения ячеек очередного фронта описанным выше способом.

2. Центром фронта $j=1$ распространения волны обозначить ячейку, которая задана как начало пути (ячейка A).

3. По формуле $H(i, B) = |x_i - x_B| + |y_i - y_B| + |z_i - z_B|$ определить оценку для каждой ячейки очередного фронта j . Рассмотренной ячейке присвоить порядковый номер и запомнить для нее родителя (ячейку, из которой распространялась волна).

4. Выбрать ячейку с минимальной оценкой. Если таких ячеек несколько, то выбрать ячейку с максимальным (минимальным) порядковым номером. Она становится центром очередного фронта $j=j+1$ распространения волны.

5. Повторить выполнение пунктов 3-4, пока не будет рассмотрена ячейка B (конец пути) на очередном фронте $n+1$ в данном случае является длиной пути от A к B . Порядковый номер, который присвоен последней ячейке, рассмотренной перед ячейкой B , является количеством просмотренных алгоритмом ячеек.

6. Построить путь от конца B к началу A по ячейкам-родителям.

Разработанная модификация алгоритма слежения за целью позволяет автоматизировать трассировку в 3D-пространстве с минимальными вычислительными затратами.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Селютин В. А. Машинное конструирование электронных устройств. М., «Сов. радио», 1977, 384 с.

2. Paul Tozour, "Fixing Pathfinding Once and For All", 2008. <http://www.ai-blog.net/archives/000152.html>.

3. Mika M., Charla C. Simple, "Cheap Pathfinding," AI Game Programming Wisdom, 2002.

4. John C. O'Neill, "Efficient Navigation Mesh Implementation," Journal of Game Development, vol. 1, no. 1, pp. 71-90, 2004.

УДК 004.896

И.А. МУЗЮКИН, А.Н. САПРЫКИН

Рязанский государственный радиотехнический университет

**ИСПОЛЬЗОВАНИЕ ОСТРОВНОЙ МОДЕЛИ ГЕНЕТИЧЕСКОГО
АЛГОРИТМА ПРИ КОМПОНОВКЕ БЛОКОВ
ЭЛЕКТРОННО-ВЫЧИСЛИТЕЛЬНЫХ СРЕДСТВ**

Рассматриваются основные моменты применения островной модели генетического алгоритма для решения задачи компоновки блоков электронно-вычислительных средств.

Задача компоновки конструктивных модулей электронно-вычислительных средств является одной из наиболее важных задач конструкторского проектирования, возникающих на каждом из основных структурных иерархических уровней. Её суть состоит в том, чтобы определить число элементов, которое может быть включено в блок, при этом необходимо, чтобы обеспечивалось их нормальное функционирование. Данная задача носит комбинаторно-логический характер и может выступать в качестве примера многокритериальной оптимизации. Их особенностью является наличие сразу нескольких критериев оценивания и сравнения эффективности возможных альтернатив.

В настоящее время большинство промышленных САПР обходятся без компоновки блоков вовсе или накладывают на нее множество ограничений. Это связано с высокой ресурсоемкостью и сложностью обработки необходимых массивов исходных данных.

Решение задачи компоновки блоков электронно-вычислительных средств основывается на их коммутационной схеме. Для алгоритмизации и формального решения задачи производится переход от коммутационной схемы к графу или гиперграфу [1]. В связи с данной постановкой задачи ее решение сводится к разбиению графа на заданное количество частей, что в свою очередь является NP-полной и NP-сложной задачей. В условиях современного уровня развития электронно-вычислительных средств использование стандартных алгоритмов компоновки является нецелесообразным в силу чрезмерно большого количества критериев оптимизации и количества элементов. Получение оптимального решения возможно только полным перебором различных вариантов разбиения.

Разбиение графа на части, как правило, может производиться несколькими методами. При использовании тривиального метода разбиения с использованием полного перебора находятся все возможные

варианты разбиения, а затем из них отбирается наиболее подходящий. Недостатком данного метода является то, что даже для современных ЭВМ он является чрезмерно ресурсозатратным (рис. 1).

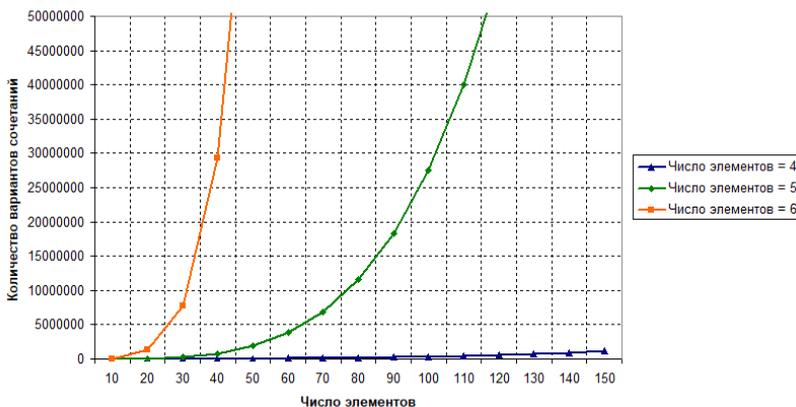


Рис. 1. Примеры количества вариантов сочетаний

На рис. 1 показаны графики зависимости количества вариантов компоновки от заданного количества элементов в блоке. Количество вариантов компоновки находится по следующей формуле:

$$\tilde{C}_n^m = P_{m,n-1} = \frac{(n+m-1)!}{m!(n-1)!}.$$

При увеличении числа элементов или блоков электронно-вычислительного средства, количество возможных вариантов их сочетания резко увеличивается в экспоненциальной форме и на их компоновку с использованием метода полного перебора потребуется предельно большое количество машинного времени. Таким образом, разработка нового метода компоновки блоков электронно-вычислительных средств, нетребовательного к машинным и временным ресурсам, является актуальной задачей.

В данной статье, для решения данной задачи предлагается использование островной модели генетического алгоритма. Он представляют собой один из наиболее эффективных способов решения крупномасштабных многокритериальных задач оптимизации, позволяет эффективно учитывать множество входных ограничений, обладая при этом малой временной сложностью[2,3].

Вся мощь островной модели генетических алгоритмов проявляется при применении распределенных вычислений. Данная модель основана на разбиение популяций на несколько отдельных остров, ка-

ждая из которых независимо друг от друга развивается и обрабатывается. При этом генетические параметры работы должны несколько отличаться друг от друга. Это позволит производить поиск на каждом из таких островов в различных направлениях, а затем через заданное количество итераций должен происходить обмен лучшими особями, для того чтобы скорректировать направление поиска. Обобщенная схема островной модели генетического алгоритма (рис 2.)

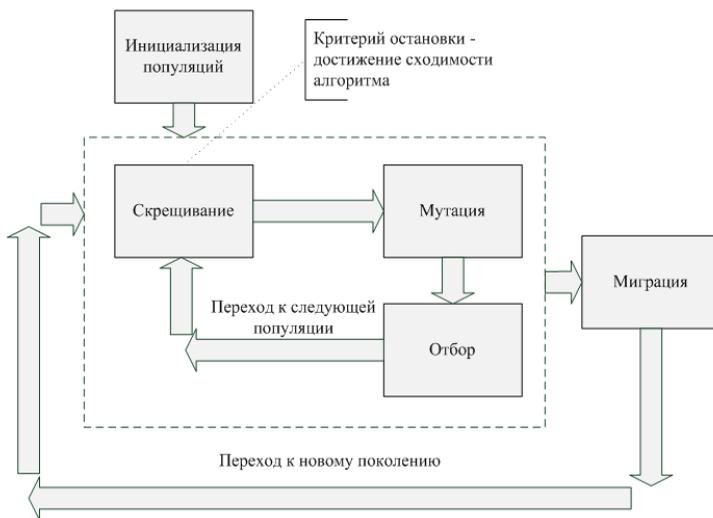


Рис. 2. Обобщенная схема островной модели генетического алгоритма

Все процессы, представленные на рис. 2, подробно описываются в работах [1-3].

Предложенный алгоритм позволяет эффективно решать задачу компоновки блоков электронных средств большой размерности и получать оптимальные результаты за полиномиальное время.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Курейчик В.М. Математическое обеспечение конструкторского и технологического проектирования с применением САПР: учебник для вузов / В. М. Курейчик. – М.: Радио и связь, 1990. – 352 с.
2. Гладков Л.А. Генетические алгоритмы [Текст] / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. –М.: Физматлит, 2010. – 317 с.
3. Курейчик В.М. Генетические алгоритмы и их применение. Таганрог: Изд-во ТРГУ, издание второе, дополненное, 2002. – 242 с.

УДК 004.428

С.И. МЫЧКО

Рязанский государственный радиотехнический университет

МИКРОСЕРВИСНАЯ АРХИТЕКТУРА

Рассматриваются особенности микросервисной архитектуры.

Микросервисная архитектура — вариант сервис-ориентированной архитектуры программного обеспечения, направленный на взаимодействие насколько это возможно небольших, слабо связанных и легко изменяемых модулей — микросервисов, получивший распространение в середине 2010-х годов в связи с развитием практик гибкой разработки и DevOps.

Если в традиционных вариантах сервис-ориентированной архитектуры модули могут быть сами по себе достаточно сложными программными системами, а взаимодействие между ними зачастую полагается на стандартизованные тяжеловесные протоколы (такие, как SOAP, XML-RPC), в микросервисной архитектуре системы выстраиваются из компонентов, выполняющих относительно элементарные функции, и взаимодействующие с использованием экономичных сетевых коммуникационных протоколов (в стиле REST с использованием, например, JSON, Protocol Buffers, Thrift). За счёт повышения granularity модулей архитектура нацелена на уменьшение степени зацепления и увеличение связности, что позволяет проще добавлять и изменять функции в системе в любое время[1].

Важно понимать, что под сервисом понимается целый набор услуг и определенный функционал, который предоставляют потребителю. А микросервисы — это дробление функционала так, чтобы он был доступен другим частям системы. Причем дробление функционала настолько мелкое, что внутри каждого микросервиса реализовано совсем маленькое количество функций.

Микросервисная архитектура приложения родилась из монолитной архитектуры, когда та стала сложной и неудобной в работе. Главное отличие микросервисов от монолита — это использование специализированных более простых программ (модулей) при выполнении сценария приложения. Тогда как в монолитной архитектуре использовались внутривещные взаимодействия. И, что самое удобное, мо-

дули могут находиться на разных серверах и их взаимодействие происходит через сеть по протоколнезависимой технологии.

Микросервисная архитектура имеет ряд преимуществ перед монолитной:

- симметричная архитектура (в монолитных приложениях – иерархическая);
- взаимозаменяемость микросервисов;
- независимость микросервисов друг от друга;
- организация модулей вокруг отдельных функций;
- написание микросервисов с помощью любых программных средств, оптимальных для каждого конкретного модуля.

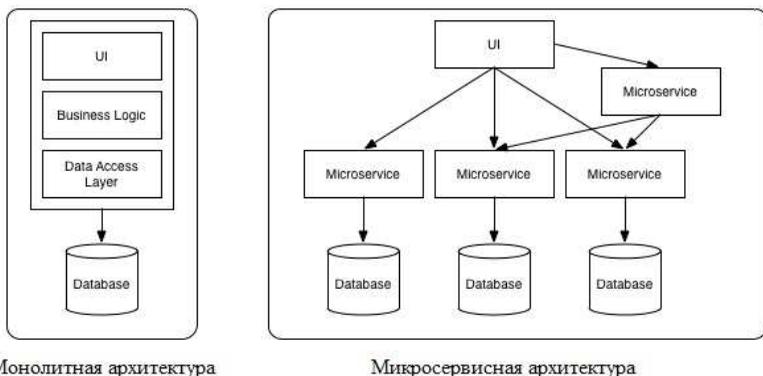


Рис. 1. Различие микросервисной и монолитной архитектуры.

Каждый микросервис – это небольшая монолитная программа, которая выполняет свою функцию. В программный продукт при разработке микросервисной архитектуры можно добавлять любое количество новых микросервисов, расширяя его функциональность. Чтобы добиться подобного в монолитной программе, необходимо вносить изменения в основной продукт.

Монолитная архитектура не справлялась с потребностью бизнеса ускоряться, так как крупные сервисы сложно быстро изменять и адаптировать. Чтобы добиться ускорения, разумным решением стало создание типовых микросервисов, по-разному реализованных в крупных сервисах и выделенных в самостоятельные компоненты. Теперь каждый крупный сервис может пользоваться микросервисами по необхо-

димости. А так как отпала нужда в дублировании типовых функций, это сделало программы более легкими и гибкими.

При микросервисной архитектуре приложения можно обновлять ПО частями, что гораздо проще и безопаснее, чем обновление «монолита». Приложение не выйдет из строя, если произойдет сбой одного или нескольких микросервисов.

О проблеме в использовании монолитных приложений впервые заговорили в 2005 году. Большие цельные программы стали слишком неудобными в разработке, не давали нужной скорости внедрения и функциональности, сложно тестировались и долго вводились в эксплуатацию. Они не позволяли быстро вносить поправки и оперативно реагировать на изменения бизнес-требований. Уже тогда микросервисы назвали новой ступенью развития архитектуры, которая обеспечит приложениям необходимую гибкость и легкость[2].

Микросервисы в современном понимании имеют следующие особенности:

- легкость и простота: каждый модуль выполняет уникальную функцию и имеет небольшой размер;
- конечность, атомарность;
- гибкость: модуль можно легко изменить, добавив в его работу необходимые опции;
- взаимозаменяемость;
- полиморфизм;
- комбинирование микросервисов для реализации разных функций.

Микросервисная архитектура устранила недостатки монолитного ПО, обеспечив изоляцию и минимизацию изменений, ускорение разработки и возможность легко подстраивать ПО под структуру бизнеса. Переход с монолитной архитектуры на микросервисы делает приложение более мобильным и гибким. Компания получает в свое распоряжение программу-конструктор, которая может оперативно подстраиваться под нужды бизнеса.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Микросервисная архитектура [Электронный ресурс] // URL: https://ru.wikipedia.org/wiki/Микросервисная_архитектура.
2. Martin Fowler. Microservices [Электронный ресурс] // URL: <https://martinfowler.com/articles/microservices>.
3. Ньюмен С. Создание микросервисов. — СПб.: Питер, 2016. — 304 с.

УДК 621.396

А.З. НГУЕН, А.П. ШИБАНОВ

Рязанский государственный радиотехнический университет

**СИСТЕМА КОСМИЧЕСКОЙ СВЯЗИ С РЕЗЕРВИРОВАНИЕМ
КАДРОВ И ИСПОЛЬЗОВАНИЕМ GERT-СЕТЕЙ***Рассматриваются вопросы резервирования информационных кадров, передаваемых через геостационарный спутник на основе использования GERT-сетей.*

Система резервирования, в наибольшей степени предназначенная для сетей со спутниковой связью, изображена на рис. 1 и 2.

Последовательности окон, отведенных для передачи данных, предшествует период резервирования, состоящий из m окон резервирования, по одному окну на каждый узел. Пусть v обозначает длину окна резервирования. За единицу измерения времени принята средняя длина пакета данных. Минимальная длина кадра устанавливается так, чтобы превышать время до спутника и обратно 2β . В окнах резервирования текущего кадра производится распределение окон передачи данных следующего кадра.

Каждый резервирующий пакет может выполнять резервирование большого числа данных. Из-за задержки распространения резервирования (250 мс по тракту «земля»-«земля»), выполняемые на одном интервале резервирования, относятся к окнам передачи данных, идущих за следующим интервалом резервирования (см. рис. 2). Предполагаем, что входной поток пуассоновский.

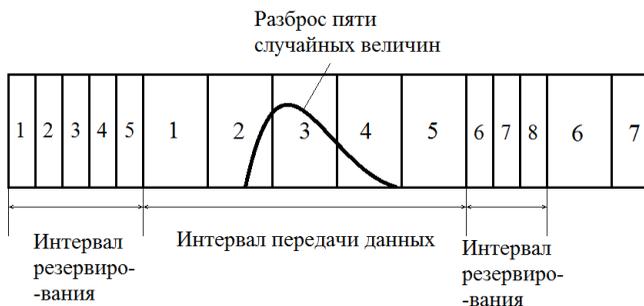


Рис. 1. Интервал передачи данных.

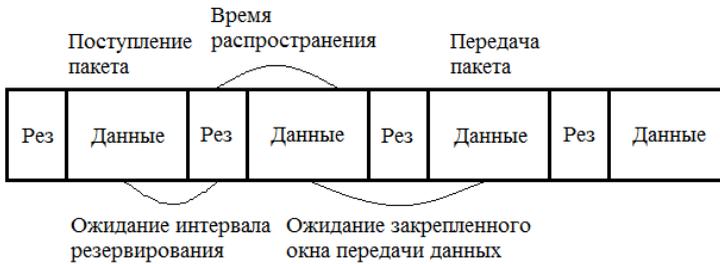


Рис. 2. Пакеты, поступившие в одном кадре, передаются в другом кадре.

Если окна резервирования короткие, то тратится мало времени на источник, которому нечего передавать. Если допустить конфликты в окнах резервирования, то из-за большой задержки по каналу они будут разрешаться очень медленно. При сильной нагрузке кадры становятся очень длинными, и редкие периоды резервирования занимают очень мало времени.

Если $A = mv$, где m – число резервируемых пакетов, а v – время резервирования; величина A умножается на 2, так как пакет перед передачей должен ожидать прохождения двух интервалов резервирования. В [1] показано, что для системы с отсечением справедливо равенство

$$W = \frac{\lambda t_{\text{cp}}^2}{2(1-\lambda)} + \frac{A}{2} + \frac{2A}{1-\lambda}, \quad (1)$$

где W – средняя задержка пакета в очереди системы резервирования; $t_{\text{cp}} = 1/\mu = 1 -$ среднее время передачи пакета, $\lambda = \rho$.

Кроме того, W остается конечным при $\lambda < 1$. Система спутниковой связи хорошо описывается при больших нагрузках (при $\lambda \rightarrow 1$).

Из формулы Литтла следует, что среднее время нахождения N пакетов в системе с постоянными интервалами передачи кадра и учетом сокращения интервала резервирования k за счет композиции случайных величин равно:

$$NT = \frac{1}{\lambda} NWk = \frac{1}{\lambda} Nk \left[\frac{\lambda t_{\text{cp}}^2}{2(1-\lambda)} + \frac{A}{2} + \frac{2A}{1-\lambda} \right], k < 1.$$

Рассмотрим простую GERT-модель резервирования следующих друг за другом пяти пакетов данных.

Для резервирования одного пакета данных, длина которого отражается случайной нормально распределенной величиной с математическим ожиданием 10 и дисперсией 9, на спутнике необходимо выделить интервал $[-0,25; 20,25]$ единиц, несколько больший интервала $[-0; 20]$ (см. рис. 3).

При резервировании следующих без задержек пяти пакетов данных получим плотность распределения вероятностей выхода этой модели, изображенную на рис. 4. Фактически оказался задействован диапазон $[26; 74]$ длиной в 48 ед. То есть выигрыш составляет более 50 %. Такой эффект определяется тем, что событие, заключающееся в том, что выпадет подряд 5 случайных исходов на левой границе интервала, имеет пренебрежимо малую вероятность. То же можно сказать и о правых границах интервалов.

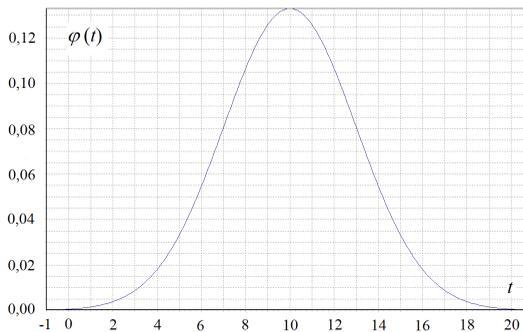


Рис. 3. Плотность распределения вероятностей при резервировании одного пакета данных.

Если в системе космической связи установить программно-аппаратные средства расчета композиции нескольких случайных величин, то можно более экономно произвести резервирование памяти для соответствующих пакетов по сравнению с алгоритмом выделения памяти для всего диапазона каждого пакета. Причем можно использовать, не только случайные величины, имеющие ограниченный диапазон значений, но и определенные на бесконечности.

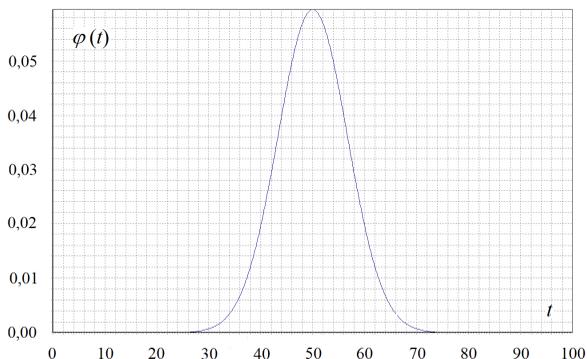


Рис. 4. Плотность распределения композиции 5 случайных величин.

В реальной работе станция «знает» конкретный размер резервируемого на спутнике пакета, и возможно определить требуемое по максимуму пространство памяти для всего множества резервирований. Пусть входной поток от станций разделен на профили трафика и точная длина пакетов каждого профиля заранее не известна. Но колебания длины пакетов в профиле можно охарактеризовать случайной величиной. Полагаем, что только такие данные известны на стадии проектирования системы. Тогда множество выборок случайных наборов пакетов можно описать моделью на рисунке 4. Тогда на основе теоретико-вероятностных соображений можно рассчитать возможную экономию памяти на борту спутника, которую можно использовать для фоновых задач или непосредственно для планирования многих периодов резервирования.

В прототипе кадр резервируется с учетом его минимальных и максимальных значений (кадры 1-5 на рисунке 1). Там же приведена кривая плотности вероятности композиции случайных величин, дающая выигрыш, если при проектировании системы учтены вероятностные предположения, связанные с использованием метода GERT.

Формула (1) справедлива при $A = \text{const}$. Именно, относительно этой величины и берется плотность распределения, определяющая выигрыш по памяти на спутнике.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Д. Берсекас, Р. Галлагер. Сети передачи данных. М.: Мир, 1989. 544 с.

УДК 004.424

В.А. НИКОЛАШИН, А.Э. ЯКУШКИН

Рязанский государственный радиотехнический университет

СРЕДА ПРОГРАММИРОВАНИЯ КАК ОСОБЕННОСТЬ ВЫБОРА ПЛК ДЛЯ АСУ ТП

Рассматривается создание простой программы для ПЛК в различных средах программирования, но на одном и том же языке программирования в целях убеждения в том, что среда программирования играет немаловажную роль в выборе ПЛК для АСУ ТП.

Автоматизированные системы управления в настоящее время присутствуют на большинстве предприятий, в мелкосерийном и крупносерийном производстве, осуществляющие сбор необходимой информации с последующей её обработкой и на основе этой информации выработку необходимых управляющих сигналов технологическим объектом, контроль за технологическими процессами. Человеческий труд полностью или частично заменяют системы автоматизированного управления.

Основой таких систем является программируемый логический контроллер, представляющий собой устройство, имеющее несколько входов и выходов (количество зависит от модели ПЛК), а также внутреннюю память, которая хранит в себе программу (логику). Программой логически связаны входы и выходы контроллера. На входы поступают сигналы (аналоговые или дискретные) с датчиков или с различных устройств. На основе этих сигналов и логики, содержащейся в памяти, контроллер вырабатывает определённые управляющие сигналы на выходах.

Программирование контроллера осуществляется с помощью различных сред программирования, содержащие различные языки программирования логики работы контроллера.

При выборе ПЛК стоит обратить внимание не только на основные характеристики контроллера (количество входов, выходов, питание и т.д.), но и на среду программирования, так как языки программирования могут совпадать только по названию. Убедиться в этом позволяет простой пример программирования переключательной функции. В качестве сред программирования взяты Alpha-Programming и Zelio Soft 2, язык программирования – FBD (функционально-блочные диаграммы), с ним проще всего работать, так как представляет собой

набор логических и функциональных блоков, соединённых между собой и наглядно показывающих логику работы без всякого программного кода.

Переключательная функция позволяет, к примеру, выбирать необходимую строку или состояние в зависимости от входного воздействия, чем-то даже напоминает работу дешифратора состояния. Данная функция используется в большинстве программ, позволяя решать задачи, такие как: выбор режимов работы объекта управления, изменение границ в резервуаре с водой, при достижении которых необходимо включить или отключить насосы для откачки, и тому подобное.

Рассмотрен простой пример реализации функции переключения между тремя диодами.

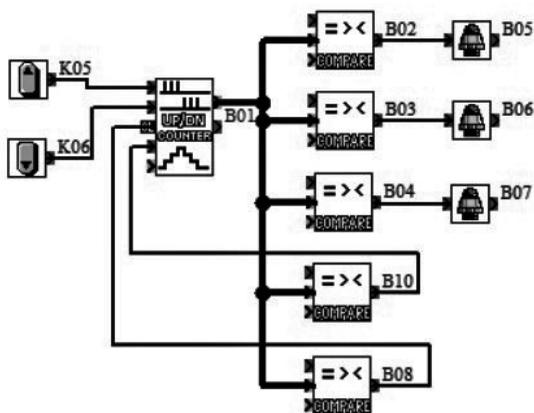


Рис. 1. Схема переключения между тремя диодами в среде программирования Alpha-Programming.

Представлена схема переключения между тремя диодами B05, B06 и B07 (см. рис. 1). Описание блоков: 1) B01 – реверсивный счётчик, имеющий 2 верхних входа для увеличения или уменьшения значения счёта, третий «синий» вход предназначен для сброса значения счёта в 0, четвёртый вход предназначен для присвоения текущему значению счёта значение уставки (значение уставки – максимальное число счёта, выше которого счётчик перестаёт накапливать значение счёта; задаётся значение уставки в контекстном меню блока), нижний аналоговый вход не используется; 2) блоки B02, B03, B04, B10, B08 – блоки компаратора, предназначенного для сравнения значения, поступающего на один из входов со значением уставки, задающейся в контекстном меню блока.

Исходя из текущего значения счёта блока V01 определяется, какой из диодов должен включиться. Допустим, значение счёта 0 будет соответствовать включение в работу диода V05, значение 1 – включение диода V06 и 2 – включение диода V07. Изменять текущее значение счёта позволяют клавиши K05 и K06, соответственно либо уменьшение текущего значения счёта, либо увеличение. Компараторы V02 – V04 сравнивают текущее значение счёта с уставками: у V02 значение уставки, равное 0 (диод V05), у V03 – 1 (диод V06), у V04 – 2 (диод V07). Тем самым обеспечивается выбор включения нужного диода. Компараторы V10 и V08 не позволяют текущему счёту блока V01 выходить за пределы значений 0 – 2. При достижении текущего значения счёта, равного 2, и при дальнейшем нажатии на клавишу K05 счёт должен стать равным 3, но, компаратор V08, если счёт превысит значение 2, сбросит счётчик V01 в 0. Компаратор V10 устроен аналогичным образом, только он сработает при текущем значении счёта меньшим, чем 0, и установит значение счёта равным 2.

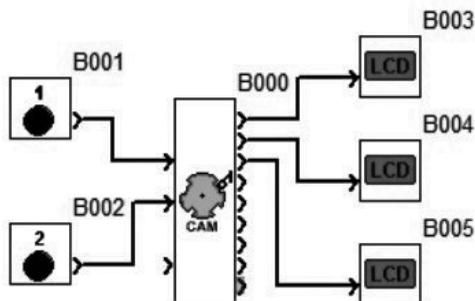


Рис. 2. Схема переключения между тремя диодами в среде программирования Zelio Soft 2.

Представлена схема функции переключения между тремя диодами (см. рис. 2), реализованная в среде программирования Zelio Soft 2, работающая по тому же принципу, что и предыдущая схема (см. рис. 1). Блок V000 выполняет функцию переключения между выходами V003, V004 и V005 (играющие роль диодов), управление переключением осуществляется с помощью клавиш V001 и V002. В контекстном меню блока задаётся логика работы блока, исходя из которого соответствующему по счёту сигналу будет происходить включение соответствующего выхода. Клавишей V001 происходит переключение от блока V003 до блока V005, клавишей V002 происходит переключение в обратном порядке.

Схемы, спроектированные в средах программирования Alpha-Programming и Zelio Soft 2 для решения одной и той же задачи, на одном и том же языке программирования FBD имеют различия. Во втором варианте реализации нет ни компараторов, ни счётчиков, схема приобрела более простой вид в плане количества функциональных блоков, соответственно внутренней памяти ПЛК будет занято меньше.

Здесь можно сколько угодно говорить о преимуществах и недостатках той или иной среды программирования. В данном случае с переключающей функцией естественно лучше использовать для программирования Zelio Soft 2, так как всего необходим один функциональный блок. Но, встречаются и задачи, которые было бы проще решить с помощью Alpha-Programming или любой другой среды программирования, так как некоторых необходимых функциональных блоков в Zelio Soft 2 может и не быть. Естественно, что программирование ПЛК на таких простых задачах не останавливается, скорее всего переключающая функция входит в состав более сложных схем.

Из всего вышесказанного вытекает вывод, что перед проектированием задач АСУ ТП и последующим выбором ПЛК необходимо обращать внимание на среду программирования, так как она играет немаловажную роль в выборе ПЛК.

УДК 004.421

И.С. НОВИКОВ, В.А. МАКАРОВА

Рязанский государственный радиотехнический университет

ПРОГРАММА РАСЧЕТА ПРОЧНОСТНЫХ ПАРАМЕТРОВ НЕСУЩЕЙ КОНСТРУКЦИИ

Рассматривается методика выполнения прочностных расчетов несущей конструкции и программа расчета прочностных параметров печатной платы как частного случая несущей конструкции.

В настоящее время радиоэлектронная аппаратура (РЭА) широко используется в различной технике. Чрезвычайно возросла оснащённость радиоэлектронными устройствами таких объектов, как корабли, самолеты, спутники, и отказ в работе хотя бы одного из устройств может привести к отказу всего объекта. Поэтому требования к надёжности и стабильности РЭА постоянно растут.

В процессе эксплуатации техники на несущие элементы конструкций РЭА, электрорадиоэлементы и узлы действуют различные механические силы.

Аппаратура, устанавливаемая на подвижных объектах, а также стационарная РЭА во время транспортировки подвергается внешним механическим воздействиям: вибрациям или ударам. Источниками этих воздействий могут быть различные двигатели, дорожная тряска и т.д. Поэтому при разработке конструкций РЭА, её составных частей и деталей конструктору необходимо оценивать качество принятых конструктивно-технологических решений для выбора оптимального варианта или определения степени соответствия требованиям технического задания (ТЗ).

Теория сопротивления материалов является основой для оценки статической прочности несущих конструкций РЭА.

Общепринятым подходом для расчета вибрационной и ударной прочности конструкций является приведение динамических задач к статическим. При выполнении оценочных прочностных расчетов следует придерживаться методики, которая включает в себя несколько этапов:

- 1) Выбор расчетных моделей конструкции РЭА и ее элементов;
- 2) Определение нагрузок, испытываемых элементами конструкций;
- 3) Расчет допускаемых значений прочности элементов конструкций;
- 4) Сравнение расчетных показателей прочности с допускаемыми [1].

Как правило, НК – это сложная сборочная единица. Поэтому для расчета её параметров целесообразно заменить её на модель – сочетание деталей упрощенной формы. Одной из наиболее применимых моделей является пластинчатая модель [2].

Пластина – тело призматической формы, высота (толщина) которой мала по сравнению с размерами основания a , b . Крепление пластины жесткое, опертое или свободное. [1].

Рассмотрим печатную плату, которая, в своем роде, является несущей конструкцией для электрорадиоэлементов (ЭРЭ). Для оценки прочности печатной платы её можно представить в виде пластины с распределенной нагрузкой элементов.

В соответствии с общей методикой прочностных расчетов для оценки прочности платы необходимо в первую очередь рассчитать основную частоту собственных колебаний платы [1]:

$$f_0 = \frac{K_M \cdot K_{ЭРЭ} \cdot c \cdot h}{a^2} \cdot 10^2,$$

где K_M – поправочный коэффициент на материал; $K_{ЭРЭ}$ – поправочный коэффициент на массу ЭРЭ; c – частотная постоянная, соответствующая различным вариантам крепления пластины при различных соотношениях сторон a/b (определяется по справочнику); h – толщина пластины, м; a – длина пластины, м.

Для оценки вибропрочности платы используется следующее соотношение:

$$\sigma = \frac{3 \cdot M \cdot g \cdot n_n \cdot \eta \cdot a}{b \cdot h^3} < [\sigma],$$

где σ – механическое напряжение в плате, Н/м²; M – масса установленных на плате ЭРЭ, кг; g – ускорение свободного падения – 9,8 м/с²; n_n – коэффициент вибрационной перегрузки; η – коэффициент динамичности; $[\sigma]$ – допускаемое напряжение, Н/м² [1].

Программа для расчета прочностных параметров печатной платы разрабатывалась в среде Visual Studio 2017 на языке программирования C#. C# – это один из самых оптимальных и универсальных современных языков программирования.

Разработанное приложение представляет собой достаточно простой визуальный интерфейс (см. рис.1 и рис.2) и обладает высокой производительностью. Результатами расчетов является определение собственной частоты колебаний, прочностных характеристик (механических напряжений и деформаций). Программа сравнивает расчетные показатели прочности с допускаемыми параметрами и выдает результат о соответствии или несоответствии техническим требованиям выбранного варианта крепления.

Расчет параметров несущей конструкции

Расчет частот собственных колебаний ПП

<p>Вид платы Односторонняя</p> <p>Материал основания платы Стеклоэпоксид СГЭФ</p>	<p>$a = 0,1 \text{ м}$</p> <p>$b = 0,1 \text{ м}$</p> <p>$h = 1 \cdot 10^{-3} \text{ м}$</p> <p>$M = 0,1 \text{ кг}$</p>	<p>$E_0 = 3,3 \cdot 10^{10} \text{ Н/м}^2$</p> <p>$\rho_0 = 2,47 \cdot 10^3 \text{ кг/м}^3$</p> <p>$\epsilon_0 = 0,279$</p>
---	--	--

Масса платы (кг): 0,025

Поправочный коэффициент на массу ЭРЭ: 0,447

Поправочный коэффициент на материал: 0,707

Частота собственных колебаний платы (Гц): 173,816

Рис.1. Интерфейс приложения.

Расчет параметров несущей конструкции

Расчет механических напряжений в ПП при действии вибрации

$a = 0,1$ м
 $b = 0,1$ м
 $h = 1 \cdot 10^{-3}$ м
 $M = 0,1$ кг
 $n = 10,4$

$\Delta f = 10 - 200$ Гц
 $f_0 = 173,8$ Гц
 $n_n = 4$
 $T = 5400$ с

Материал: Стеклоэпоксид
 $\sigma = 130$
 $\delta = 0,2$

Коэффициент динамичности: 15,708
 Допускаемое напряжение $\cdot 10^6$ (Н/м²): 3,79
 Расчетное напряжение $\cdot 10^6$ (Н/м²): 92,363
 Выбранный вариант крепления не удовлетворяет требованиям ТЗ

Рис. 2. Интерфейс приложения.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Методы оценки статической, вибро- и ударопрочности конструкций РЭА. Методические указания по курсовому и дипломному проектированию студентов специальности «Конструирование и производство радиоаппаратуры» всех форм обучения / Сост.: Ю.Ф. Зиньковский, В. П. Гондюл. – Киев: КПИ, 1985. – 39с.

2. Кольтюков Н. А., Белоусов О. А. Проектирование несущих конструкций радиоэлектронных средств. – Тамбов: Изд-во Тамб. гос. техн. ун-та, 2009. – 84 с.

УДК 378.146

А.П. НОВИКОВ, А.А. МИТРОШИН

Рязанский государственный радиотехнический университет

РАСШИРЕННАЯ МАТРИЦА КОМПЕТЕНЦИЙ

В статье рассматривается использование паспорта компетенций как одного из способов методического сопровождения образовательных программ для организации образовательного процесса, а также возможность его использования в информационной системе обеспечивающей сопровождение процесса создания рабочих программ.

Признанными стандартами качества в сфере высшего образования являются принципы, принятые странами — участницами Болонского процесса на международных конференциях, проводимых с це-

лью унификации стандартов и интеграции национальных систем образования [1].

К Болонскому процессу присоединилась также и Россия, в результате чего возникла необходимость в модернизации российского образования с целью реализации ключевых положений декларации, касающихся принятия системы сопоставимых степеней, введения двух циклов обучения, внедрения европейской системы перевода и накопления баллов (кредитов), и развития и расширения академической мобильности обучающихся [1]. В связи с этим обновленный федеральный государственный образовательный стандарт (ФГОС) стал основываться на компетентностном подходе, сущностной характеристикой которого является обозначение компетентности как цели профессиональной подготовки специалиста в вузе [2].

В статье [3] приводится описание сложностей, возникающих при реализации компетентностного подхода в образовании. В качестве основных сложностей выделяются проблемы связанные с соотношением формируемых компетенций, их формулировок и описаний результатов обучения с отдельными позициями учебного плана, вследствие особенности выработки компетенции (получения знаний, представлений, межличностных, интеллектуальных, инструментальных навыков) в процессе всех форм учебной деятельности обучающегося. А также в связи с тем, что формулировка компетенций в ФГОС носит широкий контекст, на вузовском уровне требуется их уточнение с учетом реализуемого профиля подготовки. По этой причине уделяется пристальное внимание поискам эффективных методик для реализации компетентностного подхода.

В качестве примера одной из методик можно выделить использование паспорта компетенций.

Паспорт компетенции — документ, описывающий реализуемую компетенцию в наглядном структурированном виде посредством декомпозиции компетенции выпускника образовательной программы на планируемые результаты обучения (владения, умения, знания), характеризующие этапы формирования требуемой компетенции в процессе освоения обучающимся образовательной программы [4].

В процессе разработки паспорта компетенции можно выделить два этапа. На первом этапе происходит конкретизация компетенции с целью достижения однозначного понимания, а именно, какие демонстрируемые знания умения являются проявлением данной компетенции, тем самым структура компетенции принимает вид, приведенный на рис. 1.



Рис. 1. Структура уточненной компетенции.

На втором этапе на основе матриц формирования компетенций по дисциплинам учебного плана и уточненных компетенций происходит распределение составных частей компетенции по дисциплинам.

Результатом реализации указанных этапов является документ паспорта компетенций, применение которого позволяет достичь следующих результатов:

- получить наиболее полное описание определенной компетенции с учетом характерных особенностей ее формирования посредством указания индикаторов (результатов обучения) и дескрипторов (показателей успешности достижения результатов обучения);
- сформировать на её основе состав учебных дисциплин (модулей) и практик;
- определить структурно-логическую последовательность изучения дисциплин в процессе реализации образовательной программы;
- разработать согласованные рабочие программы дисциплин (модулей) и практик;
- определить соответствующие образовательные технологии, позволяющие формировать конкретную компетенцию;
- определить оценочные средства и формы контроля для промежуточной и итоговой аттестаций.

В соответствии с порядком организации и осуществления образовательной деятельности по образовательным программам [5], включение паспорта компетенций в состав образовательной программы происходит по решению образовательной организации и является обязательным. А также содержание и структура паспорта компетенций носит рекомендательный характер ввиду несформированных единых требований.

В общем случае паспорт компетенции содержит следующую информацию:

- определение, содержание и основные существенные характеристики компетенции;

- содержательную структуру компонентов компетенции;

- показатели и критерии уровня овладения компетенцией;

- календарный график и траектории формирования компетенции.

В [6] описана информационная система (ИС) для сопровождения процесса создания рабочих программ учебных дисциплин, включающая модули заполнения учебного плана и создания дисциплин. Предполагается, что заполнение учебного плана происходит на основе готового документа, который выступает в качестве исходных данных для модуля создания дисциплин. Такой подход ограничивает функциональные возможности ИС и не позволяет разрешать вышеописанные сложности, возникающие при реализации компетентностного подхода.

Реализация модуля создания паспорта компетенций могло бы позволить получить средство аккумуляции информации о формируемых компетенциях с целью принятия решений о составе учебных дисциплин их последовательность изучения в процессе реализации образовательной программы.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Плаксунова Т.А. Качество российского образования в свете положений Болонской конвенции // Экономический анализ: теория и практика. – 2011. № 25.

2. Пузанова Ж.В., Корнаухова Ю.С. Компетентностный подход в образовательной и управленческой практике: модели компетенций // Вестник Российского университета дружбы народов. Серия: Социология. – 2014. № 1.

3. Елина Е.Г., Ковтун Е.Н., Родионова С.Е. Компетенции и результаты обучения: логика представления в образовательных программах // Высшее образование в России. – 2015. – № 1.

4. Соловова Н.В. Формирование и оценка компетенций: учебное пособие / Соловова Н.В. – Самара: Изд-во «Самарский университет», – 2015.

5. Приказ Министерства образования и науки РФ от 19 декабря 2013 г. № 1367 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры».

6. Митрошин А.А., Новиков А.П. Информационная система для сопровождения процесса создания рабочих программ учебных дисциплин // Новые информационные технологии в научных исследованиях:

материалы XXIII Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. Том 1. Рязанский государственный радиотехнический университет. 2018.

УДК 621.31

Д.С. НОСЕНКО

Рязанский государственный радиотехнический университет

РАЗРАБОТКА И ВНЕДРЕНИЕ МОДУЛЯ УПРАВЛЕНИЯ УСТРОЙСТВА ТЕХНОЛОГИЧЕСКОЙ ТРЕНИРОВКИ

Рассматриваются возможность и актуальность разработки и внедрения модуля управления устройства технологической тренировки.

В настоящее время во всех отраслях промышленности существует несколько проблем. Это проблема качества выпускаемых изделий, вызванная расширением функциональных возможностей и усложнением устройств. Проблема автоматизации контроля качества выпускаемых изделий.

Как известно, отказы возможны даже в хорошо отлаженном производстве. Это связано с различными техническими факторами. По этой причине распространенным способом повышения качества и надежности выпускаемой аппаратуры является тренировка.

В настоящее время довольно остро стоит вопрос о качестве изделий, выпускаемых на предприятиях. Одним из методов отбраковки изделия является тренировка, режимы которой регламентируются ОСТ 4.052.000-86 [1].

Долгое время тренировка проводилась вручную, но зачастую человеческий фактор препятствовал грамотному ее проведению. Изделия низкого качества, ошибочно выпущенные с предприятием, попадали к потребителю, а также устанавливались на военную технику, что сказывалось на обороноспособности нашей страны и имидже предприятия. Для упрощения и повышения эффективности отбраковки разработано схемотехническое решение устройства технологической тренировки, составляющей частью которого является модуль управления, которое обеспечивает установку режимов, согласно специальным требованиям. Модуль управления необходим для автоматизации процесса отбраковки изделий. Это позволит получить более строгие показатели качества, а также сократить время тренировки, что в свою очередь позволит увеличить производительность труда. Как показывает практика,

большинство успешных зарубежных предприятий не допускают выпуска 5% бракованных изделий, что предусмотрено военным стандартом США MIL-STD-883. Приведенные выше факты делают проблему конструирования устройства тренировки, а в частности модуля управления, довольно актуальной.

Устройство технологической тренировки применяется для первичной диагностики специальной аппаратуры. Анализ данных технического задания дает четкое представление о правилах и методах проектирования, возможных трудностях при разработке конструкции и её особенностях [2].

Согласно ГОСТ 15150-69 и техническому заданию модуль управления относится к 3 категории размещения изделия. Модуль предназначен для эксплуатации в закрытых помещениях (объемах) с естественной вентиляцией без искусственно регулируемых климатических условий, где колебания температуры и влажности воздуха и воздействия песка и пыли существенно меньше, чем на открытом воздухе.

Хранение и транспортировка:

Модуль хранится в закрытом или ином помещении с естественной вентиляцией без искусственно регулируемых климатических условий, где колебания температуры и влажности воздуха существенно меньше, чем на открытом воздухе, что относится к 3 классу хранения аппаратуры, температура хранения и транспортировки от -10 до +50 °С, при относительной влажности воздуха 98% при 35°С, прямое воздействие солнечных лучей, дождя исключить.

К устройству технологической тренировки не предъявляется жестких эксплуатационных требований. Модуль должен сохранять работоспособность в диапазоне температур при от +5°С до 35°С, при относительной влажности воздуха до 95% при температуре 35°С, при вибрационных нагрузках от 1 до 120 Гц, при перегрузке 8g при транспортировке.

Время наработки на отказ не менее 20000 часов должно достигаться технологичностью модуля, качеством сборки и качеством элементной базы.

Создание данного устройства в наше время очень актуально, а также экономически выгодно, так как при помощи устройства тренировки, а в частности модуля управления будут получены более строгие показатели качества специальной аппаратуры, что положительным образом отразится на обороноспособности Российской Федерации, а также имидже предприятий [3].

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. ОСТ 4.052.000-86 «Аппаратура радиоэлектронная. Методы проверки».
2. ОСТ 4ГО.091.219-76. Узлы и блоки радиоэлектронной аппаратуры. Методика оценки и нормативы показателей технологичности конструкции.
3. ГОСТ 15150-69. Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды.

УДК 621.31

Д.С. НОСЕНКО

Рязанский государственный радиотехнический университет

**ИССЛЕДОВАНИЕ И РАЗРАБОТКА КОНСТРУКЦИИ МОДУЛЯ
ОБОГРЕВА ОДЕЖДЫ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ**

Рассматриваются возможность и актуальность разработки конструкции модуля обогрева одежды специального назначения.

В настоящее время невозможно представить работу в условиях экстремного холода. В век высоких технологий мы должны позаботиться и о самом человеке, а именно о тех, кто вынужден находится на открытом воздухе в условиях холода. Решение данной проблемы актуально среди тех, кто вынужден производить настройку или обслуживание оборудования на открытом воздухе при отрицательной температуре, кто вынужден долго находиться в не подвижном состоянии или просто работает в условиях экстремального холода, так же актуально среди любителей зимней рыбалки и любителей активного зимнего отдыха.

В ходе исследования было выявлено о ряде преимуществ модулей для обогрева одежды специального назначения на основе инфракрасного излучения. Инфракрасные (ИК) лучи — это электромагнитное излучение, подчиняющееся законам оптики и, следовательно, имеющее ту же природу, что и видимый свет. Они занимают спектральную область между красным видимым светом (длина волны 0,74 мкм) и коротковолновым радиоизлучением (1-2 мм). В свою очередь инфракрасную область спектра условно разделяют на коротковолновую (от 0,74 до 2,5 мкм), средневолновую (2,5 - 50 мкм) и длинновол-

новую (50-2000 мкм). ИК-лучи выделяют все нагретые твёрдые и жидкие тела, при этом длина излучаемой волны зависит от температуры тела – чем она выше, тем короче волны, но выше интенсивность излучения. Следует так же напомнить, что при низких температурах излучение нагретого твёрдого тела почти целиком расположено в инфракрасной области, и такое тело кажется тёмным. При повышении температуры, излучаемые телом волны, смещаются в видимую область спектра, и тело вначале кажется тёмно-красным, затем красным, жёлтым и, наконец, при высоких температурах – белым [1].

Инфракрасные источники тепла обеспечивают:

- Экономии энергии;
- Низкие затраты на монтаж и высокую скорость его проведения;
- Возможность задавать любые температурные режимы и автоматически их обеспечивать.
- Бесшумность, экологичность, влагуустойчивость, пожаробезопасность;
- Высвобождение рабочей зоны от отопительных приборов;
- Организацию локального прогрева желаемых зон;
- Более быстрый прогрев до заданной температуры быстрее, чем при любых других способах;

Изучив всевозможные источники тепла лучше всего изготовить модуль для обогрева одежды специального назначения из углеткани. Обогрев за счет углеткани значительно безопаснее и так же более эффективен своих аналогов и позволяет значительно снизить затраты электроэнергии.

Принцип работы устройства: в конструкции используются инфракрасные источники тепла - специальные карбоносодержащие ткани, преобразующие подаваемое напряжение в инфракрасное излучение. При протекании тока через карбоносодержащую ткань происходит нагревание. Модули греются незначительно, но испускаемые ИК лучи проникают в тело человека на 1 - 2 сантиметра в глубину. Плюсы очевидны: внутренний прогрев тканей и кровотоков улучшает кровоснабжение, что позволяет самые удаленные и замерзшие части тела прогреть изнутри. Безопасное для здоровья человека ИК излучение, с длиной волны равной ИК излучению самого человека. Прогревание происходит равномерно и мягко, при этом температура повышается на безопасную температуру, человек чувствует себя комфортно. Поддерживается на протяжении длительного времени температура обогрева от 30°C до 60°C внутри одежды, чем ближе к телу будут расположены греющие модули, тем выше будет КПД устройства. Больше

тепла будет передаваться телу, и будут меньше потери на обогрев окружающей среды [2].

Конструкция модуля состоит из:

- Блока питания (литиевый аккумулятор 12В);
- Преобразователя напряжения (при питании модулей из сети);
- Блока управления;
- 2-ух модулей.

Существует ряд требований, предъявляемых к модулю:

- Модуль должен обеспечивать требуемый тепловой режим;
- Конструктивная составляющая должна обеспечивать защиту модулей от механический воздействий;
- В конструкции модуля и блока управления должны быть предусмотрены места для прокладки соединений с внешними разъемами;
- Корпус блока управления должен обеспечивать возможность доступа к расположенным в нем элементам для осмотра, ремонта и замены.

Создание данного устройства в наше время очень необходимо, так как работа в условиях экстренного холода стала неотъемлемой частью нашей жизни в условиях нашей полосы и в условиях крайнего севера [3].

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Инфракрасное излучение // Википедия. [2019—2019]. Дата обновления: 19.02.2019. URL: <https://ru.wikipedia.org/?oldid=98212267> (дата обращения: 19.03.2019).

2. Индивидуальная защита от низких температур и обморожений на охоте и рыбалке // Сибирский охотник [2014] URL: <https://www.hunting.ru/articles/view/98115/> (дата обращения: 01.03.2019).

3. Справочник конструктора РЭА: Общие принципы конструирования/Под ред. Р.Г. Варламова. - М.: Советское радио, 1980.

УДК 004.415

И.Г. ОСИН

Рязанский государственный радиотехнический университет

РАЗРАБОТКА ПРОГРАММНОЙ СИСТЕМЫ ПЛАНИРОВАНИЯ ЗАДАЧ УПРАВЛЕНИЯ ГРУППАМИ ВИРТУАЛЬНЫХ МАШИН В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ

В данной статье рассматриваются вопросы актуальности программно-конфигурируемых сетей, их виртуализация и создания системы планирования управления группами виртуальных машин с целью повышения производительности и снижения нагрузки на сеть.

Компьютерные сети как основополагающая инфраструктура — стратегический фактор развития современных ИТ, однако архитектура Сети, основы которой закладывались еще в конце 60-х годов, устарела и уже не всегда способна адекватно и эффективно реагировать на новые потребности. Рост количества и разнообразия мобильных устройств, развитие различных технологий беспроводной связи привели к тому, что сегодня число их пользователей превысило число пользователей сетей с фиксированной связью. Однако рост мощности мобильных терминалов стимулирует увеличение вычислительной емкости приложений, что, в свою очередь, требует увеличения пропускной способности каналов связи — объем мобильного трафика растет в геометрической прогрессии, а виды трафика становятся все более разнообразными.

Одновременно с ростом количественных показателей нагрузки на сети усложнились задачи управления сетями — увеличились их перечень, значимость и критичность, причем на фоне повышения требований к безопасности и надежности. Сети строятся на базе устройств, которые постоянно усложняются, поскольку вынуждены поддерживать все больше распределенных стандартных протоколов (сегодня число активно используемых протоколов и их версий превысило 600), одновременно используя закрытые интерфейсы. В таких условиях провайдеры не могут оперативно вводить новые сервисы, а производители сетевого оборудования не могут быстро модернизировать свои изделия для удовлетворения требований заказчиков. Как следствие, поддержка и управление сложной сетевой инфраструктурой стали искусством, а не инженерией.

Ответом на кризис компьютерных сетей стало появление принципиально нового подхода к их построению — программно-конфигурируемых сетей.

В SDN уровни управления сетью и передачи данных разделяются за счет переноса функций управления (маршрутизаторами, коммутаторами и т. п.) в приложения, работающие на отдельном сервере (контроллере).

Основные идеи SDN:

- разделение процессов передачи и управления данными;
- единый, унифицированный, независимый от поставщика интерфейс между уровнем управления и уровнем передачи данных;
- логически централизованное управление сетью, осуществляемое с помощью контроллера с установленной сетевой операционной системой и реализованными поверх сетевыми приложениями;

• виртуализация физических ресурсов сети.

В архитектуре SDN можно выделить три уровня:

- *инфраструктурный уровень*, предоставляющий набор сетевых устройств (коммутаторов и каналов передачи данных);
- *уровень управления*, включающий в себя сетевую операционную систему, которая обеспечивает приложениям сетевые сервисы и программный интерфейс для управления сетевыми устройствами и сетью;
- *уровень сетевых приложений* для гибкого и эффективного управления сетью.

Одна из идей, активно развиваемая в рамках SDN, — это виртуализация сетей с целью более эффективного использования сетевых ресурсов. Под виртуализацией сети понимается изоляция сетевого трафика — группирование (мультиплексирование) нескольких потоков данных с различными характеристиками в рамках одной логической сети, которая может разделять единую физическую сеть с другими логическими сетями или сетевыми срезами (*network slices*). Каждый такой срез может использовать свою адресацию, свои алгоритмы маршрутизации, управления качеством сервисов и т. д.

Виртуализация сети позволяет: повысить эффективность распределения сетевых ресурсов и сбалансировать нагрузку на них; изолировать потоки разных пользователей и приложений в рамках одной физической сети; администраторам разных срезов использовать свои политики маршрутизации и правила управления потоками данных; проводить эксперименты в сети, используя реальную физическую сетевую инфраструктуру; использовать в каждом срезе только те сервисы, которые необходимы конкретным приложениям.

Для эффективного распределения ресурсов и нагрузки на них необходима программная система планирования задач управления. Виртуальные машины должны быть равномерно и соизмеримо требуемой мощности распределены по узлам сети, которые, в свою очередь, имеют разную как пропускную способность, так и вычислительную мощность. Система планирования поможет увеличить производительность использования программно-конфигурируемых сетей, а также даст возможность дальнейшего развития данной технологии.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. В.П. Корячко, Д.А. Перепелкин «Корпоративные сети: технологии, протоколы, алгоритмы» - М.: Горячая линия-Телеком, 2011. – 216 с.: ил.

УДК 004.422

А.С. ПЕХНОВ

Рязанский государственный радиотехнический университет

КРИПТОГРАФИЧЕСКАЯ БИБЛИОТЕКА OPENSSL

Рассматриваются особенности библиотеки OpenSSL.

Уже довольно давно и остро стоит вопрос информационной безопасности. В век цифровых данных злоумышленники используют широкий инструментарий различных сетевых утилит и устройств для кражи, перехвата, изменения информации в своих целях. Также их целью может быть потеря доступа пользователей к определённому веб-ресурсу и его данным, если речь идёт об атаках типа «отказ в обслуживании». Однако в этой статье речь пойдёт о защищённой передаче данных между клиентом и сервером.

В рамках разработки веб-ориентированной среды автоматизированного анализа докладов научно-технической конференции, остро встаёт вопрос о передаче данных по открытым каналам связи по сети Internet. Передача личных данных пользователей с использованием протокола HTTP (HyperText Transfer Protocol — протокол передачи гипертекста) в таком случае является небезопасной. В случае прослушивания сетевого соединения между клиентом и сервером программами снифферами или в случае атак типа man-in-the-middle данные будут известны злоумышленникам и в итоге перехвачены или изменены. Защиту для данных целей можно обеспечить с помощью расшире-

ния протокола HTTP для поддержки шифрования – HTTPS (HyperText Transfer Protocol Secure).

HTTPS работает на основе закрытых и открытых ключей шифрования. Эти ключи получаются от центра сертификации или генерируются и подписываются самостоятельно (самозаверенные сертификаты). В случае самозаверенного сертификата, он загружается в браузер клиента. Сообщения, приходящие на сервер, расшифровываются с помощью закрытого ключа. В данном случае сертификаты будут выдаваться на каждого пользователя индивидуально с указанием имени и адреса электронной почты.

OpenSSL - это система защиты и сертификации данных, название SSL переводится, как уровень защищённых сокетов (secure socket layer). SSL представляет собой криптографический протокол для безопасной связи. С версии 3.0 SSL был переименован на TLS (transport layer security — безопасность транспортного уровня), однако имя предыдущей версии уже вошло в обиход, так что на текущий момент под SSL наиболее часто понимают TLS.

SSL предназначен для обеспечения безопасной отправки данных. Наряду с этим для аутентификации используются асимметричные алгоритмы шифрования (пара открытый — закрытый ключ), а для сохранения конфиденциальности — симметричные (секретный ключ). Первый тип шифрования требует больше ресурсов, поэтому его комбинация с симметричным алгоритмом помогает сохранить высокую скорость обработки данных.

Когда клиент входит на веб-ресурс, начинается процесс так называемого рукопожатия (handshake), браузер отправляет запрос данных о сертификате сервера, а тот в свою очередь отправляет копию SSL-сертификата с открытым ключом. Далее, браузер проверяет сертификат, название которого должно быть идентичным имени веб-сайта.

Также, производится проверка на срок истечения действия сертификата, и на наличие корневого сертификата, выданного доверенным центром сертификации. В том случае, когда браузер доверяет сертификату, то он генерирует предварительный секрет (pre-master secret) сессии на основе открытого ключа, используя максимально высокий уровень шифрования, который поддерживают обе стороны.

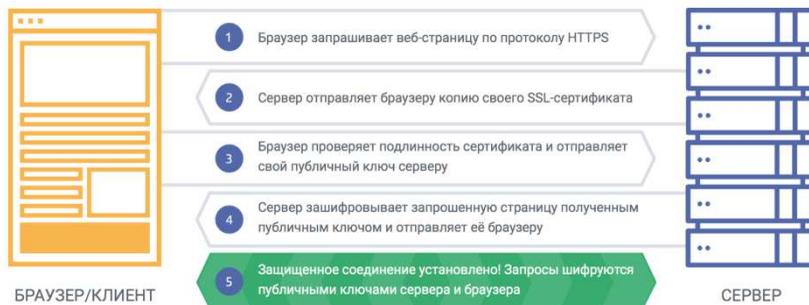


Рис.1. Механизм «рукопожатия» клиента и сервера.

Сервер осуществляет дешифрование предварительного секрета с помощью своего закрытого ключа, затем соглашается продолжить коммуникацию и создать общий секрет (master secret), используя определенный вид шифрования. Теперь обе стороны используют симметричный ключ, его действительность распространяется не более чем на время жизни текущей сессии. Как только она окончится, ключ подлежит уничтожению, а повторные посещения Интернет-ресурса вновь запустят механизм рукопожатия.

OpenSSL используется практически всеми сетевыми серверами для защиты передаваемой информацией[1].

Криптографическую библиотеку OpenSSL предоставляет возможность создавать открытые и закрытые ключи шифрования, для дальнейшей организации с их помощью https соединения клиентов с сервером. К плюсам данной библиотеки я бы отнёс:

- Простота освоения и использования,
- Бесплатность библиотеки и открытость исходного кода,
- Бесплатность использования сертификатов на неограниченный срок, поскольку получение сертификата в Центре сертификации
- Поддержка множества низкоуровневых алгоритмов хеширования, шифрования и электронной подписи,
- Возможность работать с библиотекой на разных платформах
- Реализует большинство популярных криптографических стандартов.

К минусу применения библиотеки OpenSSL можно отнести тот факт, что созданные таким образом сертификаты, скорее всего, придётся подтверждать в браузерах клиента вручную. Поскольку они не подписаны центрами сертификации.

Аналогами OpenSSL являются GnuTLS и LibreSSL.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Использование OpenSSL [Электронный ресурс] // URL: <http://www.opennet.ru/base/sec/openssl.txt.html>
2. «Как это работает»: знакомство с SSL/TLS [Электронный ресурс] // URL: <https://habr.com/ru/company/1cloud/blog/326292/>

УДК 004.383

С.В. ПОПОВ

Рязанский государственный радиотехнический университет

**СТРУКТУРА МОДУЛЯ СИСТЕМЫ ТЕХНИЧЕСКОГО
ЗРЕНИЯ БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА**

В статье рассматривается обобщенная структурная схема системы технического зрения для беспилотного летательного аппарата, а также приводится обоснование применения выбранной структуры.

В процессе формирования искусственного коллектива (ИК), а также при выполнении им некоторых задач, каждая единица ИК, должна иметь возможность исследовать окружение, а при необходимости изучать объекты, удаленные на значительное расстояние [1]. Тогда ИК будет обладать необходимым уровнем осведомленности. В качестве единицы ИК выступает БПЛА (беспилотный летательный аппарат). Учитывая особенности исследуемой структуры, средством детектирования объектов является видеосенсор.

Рассмотрим модуль системы технического зрения (СТЗ) для БПЛА. Для обеспечения выполнения задач, поставленных перед СТЗ ее необходимо оснастить группой видеосенсоров. В общем случае данную систему можно разделить на две подсистемы. Первая – подсистема сферического обзора, состоящая из двух панорамных камер. Вторая – подсистема дальнего детектирования, состоящая из камеры с регулируемым фокусным расстоянием [2].

Такая система позволит получить достоверность, достаточную для формирования роя, а также детектировать и распознавать объекты на расстоянии, необходимом для успешного выполнения поставленной перед БПЛА задачей.

Важным блоком в модуле СТЗ является блок обработки видеосигнала. В рассматриваемой системе в качестве него выступает ПЛИС (программируемая логическая интегральная схема) [3]. ПЛИС имеет высокое быстродействие, что положительно сказывается на времени

обработки информации. Это позволяет использовать более трудоемкие и достоверные алгоритмы обработки видеосигнала, обеспечивая работу в режиме реального времени. Учитывая объем входных данных, а также особенности СТЗ рационально применение двух ПЛИС.

На выход модуля поступает информация об объектах, обнаруженных с помощью двух подсистем. С головной системы приходят специальные сигналы управления режимами обработки данных. Также она осуществляет управление механикой системы дальнего детектирования.

Обобщенная структурная схема модуля СТЗ представлена на рис. 1.

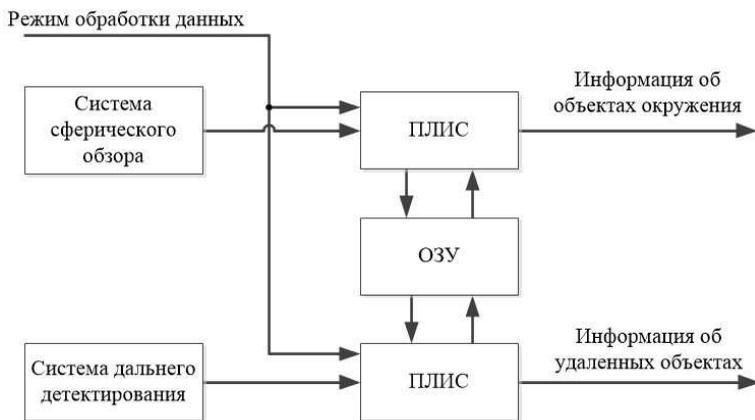


Рис. 1. Обобщенная структурная схема модуля СТЗ.

Фактически каждая ПЛИС выполняет ряд задач, связанных с обработкой видеосигнала, таких как:

- выделение объекта;
- определение его положения относительно БПЛА;
- количественная оценка размеров и параметров каждого объекта;
- классифицирование каждого объекта, присвоение метки;
- формирование специальной структуры данных, для эффективной передачи информации на головное устройство.

В процессе обработки данных, например, при отработке алгоритма детектирования, внутренних ресурсов ПЛИС может оказаться недостаточно. Тогда к ПЛИС подключается внешняя оперативная память, которая обеспечивает хранение больших объемов предобрабо-

танных данных. Такое расширение ресурсов ПЛИС позволяет «помнить» данные о каждом объекте и четко их отличать друг от друга, что повышает качество данных, передаваемых головному устройству.

По предварительной оценке, можно сказать, что при использовании камер полусферического обзора с разрешением FullHD (1920x1080 пикселей), достоверное выделение объекта в зоне окружения может происходить на расстоянии до 80м. На этом расстоянии объект с габаритами 1x1 метр, будет занимать угол поля зрения 1x1 градус, что соответствует 8x8 пикселям на кадре. Используя специальные методы, такие как субпиксельное сканирование[5], можно добиться повышения разрешения изображения.

Камера дальнего детектирования имеет меньший угол обзора и соответственно большее количество пикселей на градус. Данная СТЗ будет установлена на каждом из БПЛА, что позволит качественно оценить обстановку на большом расстоянии, при рациональном распределении ресурсов ИК. Следует отметить, что современные видеосенсорные матрицы позволяют делать изображения с большим разрешением.

Таким образом, предложенная СТЗ для единицы ИК позволяет получить необходимый уровень достоверности об окружении, обеспечивая при этом обработку данных в режиме реального времени. Детектирование удаленных объектов также обеспечивает высокий уровень достоверности.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Ликучев В.Ю. Специальная задача для группы беспилотных летательных аппаратов / Информационные технологии, межвузовский сборник научных трудов. – Рязань: Рязанский государственный радиотехнический университет, 2018 – с.46-50.
2. Козлов В.А., Потапов А.С. Анализ методов выделения движущихся объектов на видеопоследовательностях с шумами. Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. Санкт-Петербург, 2011г. №3 (73), с.39-43.
3. Держановский А.С., Соколов С.М. Обработка зрительных данных в системе технического зрения реального времени с использованием ПЛИС // Препринты ИПМ им. М.В.Келдыша. 2016. № 126. 16 с.
4. Буянов И.А., Исламов В.К. Автономная система ориентирования беспилотного летательного аппарата: состав и схема функционирования в формате 3D. Молодой ученый. Казань, 2017, №50 (184) с. 24-30.

5. С.В. Блажевич, В.Н. Винтаев, Н.Н. Ушакова Синтез космического изображения с улучшенной разрешающей способностью на основе субпиксельного сканирования / Современные проблемы дистанционного зондирования земли из космоса. – М.: изд-во РАН, 10. Т. 7. №2. С. 9-13.

УДК 004.383

Н.С. ПОТАПКИНА

Рязанский государственный радиотехнический университет

ПРИМЕНЕНИЕ ПЛИС ПРИ КОНСТРУИРОВАНИЕ И ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ УСТРОЙСТВ

Рассматривается конструирование и проектирование цифровых устройств с использованием ПЛИС, основные способы описания этих устройств, среда и маршрут проектирования, а также основные особенности ПЛИС. Также рассматривается созданный для удобства работы с устройством корпус.

ПЛИС – это программируемая логическая интегральная схема, позволяющая создавать настраиваемые цифровые электронные схемы. Для проектирования устройств с использованием ПЛИС необходимо иметь печатную плату, содержащую как интегральную схему, так и специальные индикаторы и переключатели для удобства вывода информации. Для загрузки информации на программируемую логическую интегральную схему с других устройств, например, ПК, необходимо наличие соответствующих разъемов на плате.

Пример исполнения подобной печатной платы представлен на рис. 1. Данная плата содержит ПЛИС EP4CE6E22C8N фирмы Altera, имеющую 91 порт, четыре семисегментных индикатора с возможностью отображения разделяющих точек, зуммер, светодиоды, стабилизаторы напряжения, флэш память, индикатор наличия питания, а также различные разъемы, такие как VGA, SMA, USB, JTAG.

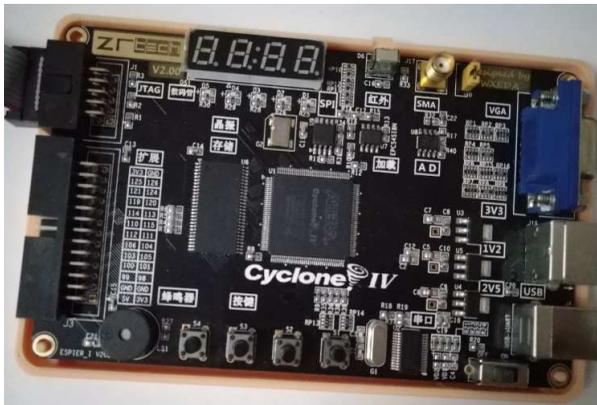


Рис. 1. Пример исполнения печатной платы, содержащей ПЛИС.

Разработки на ПЛИС обусловлены тем, что логика работы задается с помощью программирования, а не определяется при изготовлении, т.е. появляется возможность реализовывать различные устройства посредством одной платы. Также к достоинствам интегральных схем относится работа в режиме реального времени и возможность параллельной обработки команд, чего нельзя сделать на микроконтроллере, наиболее высокое быстродействие. Однако имеются и недостатки, главный из которых заключается в затрачивании большого количества времени на разработку программного алгоритма. Это обусловлено необходимостью вручную назначать сигналы ножкам интегральной схемы, а также проверкой, что в нужный момент именно к заданным ячейкам памяти обращается другой сигнал, который отслеживается или генерируется в данный момент.

Из-за длительности разработки устройств на ПЛИС, с помощью ПО Quartus 2 были созданы такие схемы, как двухтактный синхронный J-K триггера, асинхронный реверсивный счетчик, двоично-десятичный счетчик с синхронной схемой управления, последовательно-параллельный регистр, мультиплексор 4:1 с разрешающим входом, демультиплексор 1:2, унитарный многоступенчатый дешифратор на 32 выхода и многие другие. Все эти схемы созданы в ПО Quartus 2, откомпилированы, а также получены временные диаграммы, отражающие принцип работы схем (см. рис. 2). Теперь каждое из этих устройств представляет собой отдельный блок, который можно импортировать и использовать при других разработках.

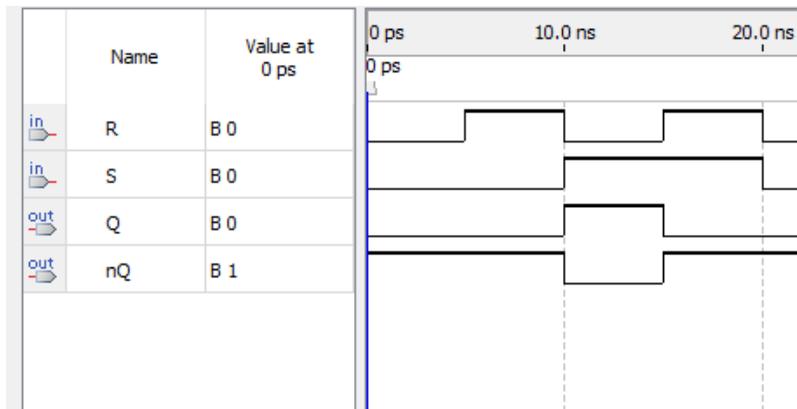


Рис. 2. Временная диаграмма работы схемы одноклапного асинхронного R-S-триггера.

Также все эти принципиальные схемы были описаны с помощью VHDL-кодов (см. рис.3) и теперь могут использоваться в других разработках в среде Quartus и ModelSim посредством подключения соответствующей библиотеки. Созданные схемы и VHDL-коды позволят разработчикам, использующим ПЛИС, тратить намного меньше времени на создание своих устройств, тем самым частично решая основную проблему интегральных схем – длительность разработки программ.

```
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity RS_FF is
port (S,R:in std_logic;
      Q,nQ:out std_logic);
end RS_FF;

architecture behavioral of RS_FF is
begin
PROCESS(S,R)
variable tmp:std_logic;
begin
if(S='0' and R='0') then
tmp:='0';
elsif(S='1' and R='1') then
tmp:='0';
elsif(S='0' and R='1') then
tmp:='0';
else
tmp:='1';
end if;
Q<=tmp;
nQ<=not tmp;
end PROCESS;
end behavioral;
```

Рис. 3. VHDL-код, описывающий работу одноклапного асинхронного R-S-триггера.

Далее полученную схему необходимо загрузить в память ПЛИС с помощью загрузочного кабеля USB Blaster. После этого плата будет работать в соответствии с принципом работы загруженного устройства. При отключении питания или загрузке другого проекта ранее загруженный проект будет стерт из памяти ПЛИС.

Пример прошивки платы схемой одноклапного асинхронного R-S-триггера представлен на рис. 4.

небольшой наклон, чтобы готовую деталь можно было легко извлечь из формы.



Рис. 5. Корпус для платы, представленной на рисунке 1.

УДК 004.428

Р.А. ПРИХОДЬКО

Севастопольский государственный университет

О МЕТОДЕ ФОРМИРОВАНИЯ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ БИЗНЕС-ПРИЛОЖЕНИЙ

Рассмотрены преимущества микросервисной архитектуры клиент – серверных приложений, обозначена актуальность правильного выделения ее компонентов. Предложен метод формирования микросервисной архитектуры.

Микросервисная архитектура (МСА) [1] клиент-серверных бизнес-приложений (БП), становящаяся повседневной практикой в последнее время, не стала неожиданностью для специалистов. Отражаемое в ней стремление к модульности ПО явилось закономерным ответом на нарастающую сложность БП и заняло свое место среди таких однородных технологий, как принцип слабой связности, трехслойная архитектура, паттерны MVC и MVP [2] и т.п. Модульность, достигаемая использованием МСА, решает следующие проблемы [3]:

1. В разработке: громоздкие БП требуют много времени для

анализа при добавлении новой функциональности и при поиске ошибок. Архитектурные приемы, задуманные с целью структурировать БП в рамках монолитной архитектуры (МА), рано или поздно нивелируются. Причиной тому человеческий фактор: высокая скорость разработки и невозможность постоянно держать в голове структуру БП приводят размыванию введенных архитектурных ограничений. Это, в свою очередь, еще больше увеличивает время анализа БП. Кроме того, как показывает практика, интегрированные среды разработки во встроенные в них автоматические анализаторы кода типа ReSharper и т.п. периодически дают сбой в громоздких БП, не справляясь со своими задачами даже при относительно серьезных вычислительных мощностях.

2. При разворачивании: в БП даже при изменениях, затрагивающих небольшую его часть, приходится разворачивать все приложение. Процесс разворачивания довольно длителен, его время умножается на количество стендов. Если речь идет о стендах для разработки и тестирования, время разворачивания пропорционально убыткам из-за простоев команды тестирования и частично разработки. Для стенда конечного потребителя минимизация времени разворачивания критически важна, например, когда нужно быстро исправить обнаруженный дефект. Дополнительным неудобством, имеющим место при разворачивании БП, является необходимость учитывать большое количество сопровождающих вспомогательных процедур, предшествующих разворачиванию или проходящих после него.

При использовании МСА процесс разработки упрощается: микросервисы (МС) полностью отделены друг от друга, что делает невозможным нарушение их границ разработчиками; исходный код каждого из них легче анализируется как человеком, так и средствами ИСР; время и сложность разворачивания значительно уменьшаются.

Как и любое другое средство, МСА эффективна при правильном применении, в противном же случае может явиться источником дополнительных проблем при разработке. При разделении функционала на МС возможны следующие ошибки:

1. Разделяемые модули оставлены неразделенными.
2. Неразделяемые модули разделены.

И если первая ошибка приводит к уже известным разработчикам недостаткам МА, то вторая является более серьезной, так как приводит к тесной зависимости между МС, снижением надежности работы приложения и высоким затратам на сначала создание, а затем исправление проблемы. Поэтому актуальным является вопрос: при

каких условиях рассматриваемый функционал следует выделить в отдельный МС?

Во-первых, он должен является (потенциально) значительным модулем, объемом которого нельзя пренебречь. Выделение слишком мелких модулей не оправдывает затрат на их поддержку и увеличит число СМ до количества, с которым сложно работать.

Далее, что касается структуры БП, разделим МС на два типа: бизнес (БМС) и служебные (СМС). К первым отнесем МС, специфичные для предметной области БП, несущие на себе основную его логику. Ко вторым – инфраструктурные, обеспечивающие вспомогательные, но при этом необходимые функции, такие как взаимодействие с пользователем, аутентификация/авторизация, маршрутизация, тестирование и т.п. Опыт показывает, что в любом БП за редкими исключениями присутствует один и тот же набор СМС:

1. Графический пользовательский интерфейс (ГПИ). При наличии выраженного программного интерфейса ГПИ удобно реализовать в виде независимого приложения, написанного, независимо от используемых на серверной части (СЧ) технологий, на JavaScript платформе.

2. Единый программный интерфейс приложения (ПИП). ПИП в МСА представляет собой абстракцию, позволяющую клиентской части (КЧ) не привязываться к реализации СЧ. Каждый запрос к СЧ ПИП перенаправляет на один или несколько сервисов СЧ. Поскольку ГПИ является общим для всех сервисов СЧ, он выносится в отдельный МС.

3. МС аутентификации/авторизации. Протокол OAuth2 отделяет друг от друга ресурс, к которому требуется доступ и клиентское приложение, нуждающееся в этом доступе. В этом смысле и свой, и сторонние клиенты СЧ рассматриваются одинаково. При этом КЧ текущего приложения обычно не является единственным клиентом БП. В ходе развития к приложению добавляется интеграция со сторонними клиентами, одним из которых может быть приложение - предшественник, на смену которого создается данное БП. Для каждого из сторонних клиентов характерен тот или иной метод аутентификации/авторизации, в результате модуль разрастается. Он независим и изменения в него вносятся не так часто, как в БМС, поэтому его выносят в отдельный МС.

4. Сервис интеграционных (ИТ) и (или) автоматизированных (АТ) тестов. Важность тестов в современной разработке сложно переоценить. АТ, ИТ, модульные тесты все больше приобретают свойства требований, материализованных в коде. Роль требований выдвигает их на первый план по отношению к разработке, что

подтверждается появлением таких методик, как разработка, предваряемая тестированием (РПТ) и разработка, предваряемая поведением [4]. АТ и ИТ дистанцируются от БП и имеют свою кодовую базу, часто на особой платформе.

5. Файловый сервис. Редкие БП обходятся без необходимости работать с документами, такими как договора, отчеты, и т.д. Сущности документов отличаются от других своим сравнительно большим размером, поэтому для их хранения используются специальные механизмы хранения. МС хранения файлов, принимая содержимое файла, возвращает своим клиентам его идентификатор.

Рассмотрим БМС и вопрос целесообразности выделения некоторого функционала в отдельный МС. Модель МС приведена на рис. 1.

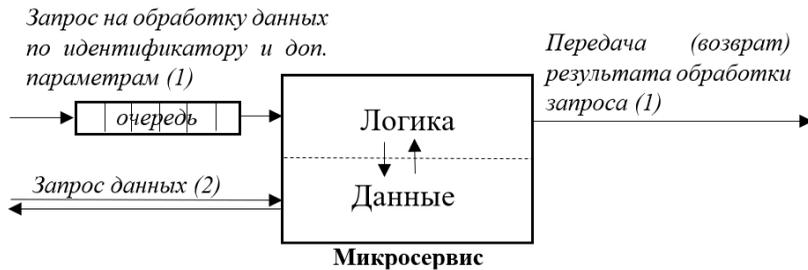


Рис. 1. Модель МС.

МС общего вида содержит некоторые данные и логику обработки этих данных. В МС имеется интерфейс для взаимодействия с остальными частями ПО. МС может принимать асинхронные запросы (1) согласно протоколу AMQP, не требующие немедленного ответа, способные формировать персистентные очереди на обработку. Также МС может принимать и синхронные запросы (2) согласно протоколу HTTP (HTTPS), требующие немедленного ответа. Запрос (1) предназначен для взаимодействия МС между собой. Синхронный запрос (2) используется для получения данных МС из ПИП или ГПИ. Однако, если разбиение на МС выполнено неправильно и имеет место ситуация разделения неразделяемых модулей, запросами (2) МС запрашивают информацию у других МС. Эта ситуация называется цепочкой синхронных запросов (см. рис. 2). Такая цепочка характеризуется высоким временем обработки запроса, низкой надежностью (за счет большого количества вовлеченных МС) и жесткой связностью МС между собой.

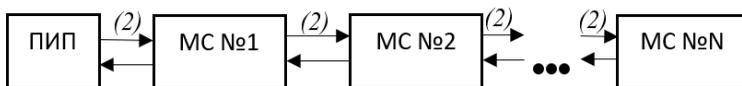


Рис. 2. Цепочка синхронных запросов.

Рассмотренная цепочка является главным индикатором того, что МС разделены напрасно и их следует объединить.

То, что МС не должны запрашивать данные друг друга, нельзя исправить дублированием данных в разных МС, так как возникнет проблема синхронизации этих данных, а ее решение весьма сложно и дорогостояще. Единственный вид информации, которую можно дублировать в других МС – это идентификаторы данных.

Введем следующие требования:

T1. МС может предоставлять свои данные по запросу только их конечному потребителю, т.е. ПИП.

Например, имеются два МС: торгов и организаций. Требуется вывести в таблицу пользовательского интерфейса список торгов и отдельной колонкой показать наименования проводящих эти торги организаций. Последние хранятся в МС организаций, а МС торгов имеет информацию только об их идентификаторах. Запрос из ПИП приходит на МС торгов. Последний при этом не должен запрашивать недостающую информацию в МС организаций. Вместо этого ПИП должен сначала запросить у МС торгов информацию по торгам, а затем, получив с ней идентификаторы организаций, запросить их наименования в МС организаций.

T2. Отсылать свои данные для обработки МС может только сам.

Например, имеются МС торгов, МС организаций. МС организаций содержит банковские счета организаций, необходимые для блокировки денежных средств. Если МС торгов нужно заблокировать средства, он не должен запрашивать банковский счет у МС организаций и блокировать средства. Работает со счетом только его владелец, т.е. МС организаций. Поэтому МС торгов должен послать запрос на МС организаций, а уже последний выполнит блокировку.

На основе введенных требований сформулируем признаки возможности выделения функционала в отдельный БМС.

1. Рассматриваемый функционал является (потенциально) значительным модулем, объемом которого нельзя пренебречь.

2. Модель данных функционала является логически единой, цельной, при этом выделяемой из модели остальных данных

приложения так, что остальные части приложения должны уметь обходиться только идентификаторами этих данных.

3. Логика выделяемого функционала не должна нуждаться в данных других МС.

4. И на первом этапе, и в обозримой перспективе должны соблюдаться введенные выше требования Т1 и Т2.

Рассмотрены преимущества МСА, обозначены возникающие при ее формировании ошибки. Выделена проблема целесообразности выделения некоторого функционала в отдельный МС. Предложен метод решения данной проблемы, позволяющий закладывать правильную МСА еще на этапе проектирования. Дальнейшим направлением исследования планируется разработка системы анализа МСА на наличие ошибок в архитектуре.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Cesar de la Torre, Bill Wagner, Mike Rousos .Net Microservices: Architecture for Containerized .NET Applications

2. Фаулер М. Шаблоны корпоративных приложений. Киев.: Диалектика, 2018. 544 с

3. Lianping Chen Microservices: Architecting for Continuous Delivery and DevOps IEEE International Conference on Software Architecture, 2018, Seattle, USA

4. Приходько Р.А. Анализ «управляемой поведением разработки» программного обеспечения // Информационные системы и технологии. 2018. № 4. С. 75 – 79.

УДК 681.5

В.А. ПШЕНИЧНИКОВ, И.И. ФОЛОМКИН

Рязанский государственный радиотехнический университет

ВНЕДРЕНИЕ АВТОМАТИЗАЦИИ ПРОЦЕССА ПОДАЧИ ВАКУУМА ПРИ ЛИТЬЕ ПО ГАЗИФИЦИРУЕМЫМ МОДЕЛЯМ

Рассматриваются вопросы внедрения автоматизированных процессов при заливке металла в литые по газифицируемым моделям.

Метод литья по газифицируемым моделям (ЛГМ) представляет собой способ получения детали при выжигании (газификации) пенополистирола на этапе заливки металла в форму. Перед заливкой металла осуществляется процесс вакуумирования, который создаёт так называемое разряжение в опоке. Сам процесс состоит из нескольких

составляющих: в первую очередь это вакуумный водоциркулярный насос, трубопровод, специальный пылеуловитель, сепаратор, гребенка, армированный резиновый рукав, вакуумный ресивер и система управления. Сердцем вакуумной системы является вакуумный насос. Вода в насос поступает из сепаратора и создает герметичную прослойку между импеллером и корпусом насоса. Обрато в сепаратор вода возвращается по верхнему трубопроводу в виде капель. Вакуумные опоки вручную подключаются к гребёнке системы через резиновый армированный рукав. Гребенка в стандартной комплектации имеет от 2 до 8 выводов, каждый из которых оснащается ручным регулировочным клапаном – бабочкой. При заливке металлом полистирольная модель выгорает, при этом газы отводятся вакуумной системой и проходят через пылеуловитель мокрого типа. В результате этого продукты деструкции полистирола осаживаются, и очищенный воздух поступает в насос. Предварительная очистка отходящих газов защищает импеллер насоса от загрязнения, а также улучшает экологическую обстановку в цехе. Вакуумный ресивер, поставляемый в комплекте с вакуумной системой, играет роль аккумулятора отрицательного давления, за счет чего при выключенном насосе разряжение в опоке поддерживается в течении 2-6 минут, в зависимости от качества полиэтиленовой пленки. Для поддержания в песке вакуума песчаные опоки герметизируют покрыванием поверхности песка синтетической пленкой, в толщу песка помещают газопроницаемый фильтр, не пропускающий песок и сообщенный трубопроводом с вакуумным-насосом[1].

Включение вакуума в литейный процесс дало следующие положительные результаты:

- резко сократилось загрязнение окружающей среды, уменьшились выбросы вредных газов в атмосферу цеха;
- повысилась точность отливок за счет исключения прилипания к моделям смеси;
- упростилось уплотнение модели из полистирола, вместо силовых нагрузок используют вибрацию сухого песка.

Применение вакуума в форме при литье по газифицируемым моделям позволило отработать режим вакуума в песке, позволяющего откачать из формы все газопродукты газификации пенопластовой модели при замещении ее металлом отливки высокого качества.

Перед тем, как произвести заливку металла в опоку, необходимо произвести ее вакуумирование, при помощи вакуумных насосов, ресиверов, вакуумного трубопровода и гибких к опокам. Все опоки оборудованы присоединениями к вакуумному трубопроводу.

Вакуумирование опоки необходимо для окончательного уплотнения песка в опоке и для удаления газов, образующихся при выгорании моделей их полистриола, температура вакуумирования и вакуумных ресиверов отличается от температуры окружающего воздуха на 20-25 градусов по Цельсию[2].

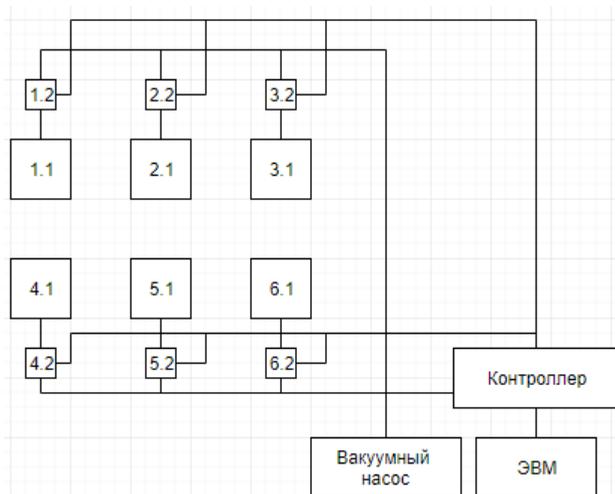


Рис. 1. Схема подачи вакуума.

На рис. 1 представлена структурная схема подачи вакуума в опоки. Показано 6 опок (1.1-6.1), каждая из которых соединена с краном (1.2-6.2) подачи вакуума через датчики, вакуумный насос, который через армированный резиновый рукав создает отрицательное давление в опоке. Контроллер, который через панель оператора отвечает за открытие/закрытие клапана подачи вакуума. ЭВМ (панель оператора), с помощью которой возможно контролировать полный процесс подачи вакуума от начала и до конца, следить за работой вакуума, предупреждать об аварии, ставить на паузу процесс, смотреть сообщения об ошибках системы и следить за настройками процесса вакуумирования.

После внедрения автоматизации процесса подачи вакуума, оператор сможет при помощи сенсорного экрана на панели оператора выбирать и включать подачу вакуума в задействованные опоки, что позволит оптимизировать производство.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Овсянников К. М. Основы расчета вакуумных систем, применяемых в литейном производстве. - М.: Машиностроение, 1971. – 80 с.
2. Дорошенко В. С., Бердыев К. Х. Структура цеха литья по газифицируемым моделям и особенности его проектирования // Металл и литье Украины. – 2010. – №4. – С. 8-16.

УДК 004.451

Д.А. СКУРОВ

Рязанский государственный радиотехнический университет

ПРИМЕНЕНИЕ ОТЕЧЕСТВЕННЫХ ЗАЩИЩЁННЫХ ОС В СИСТЕМАХ КОНТРОЛЯ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ В УСЛОВИЯХ ИМПОРТОЗАМЕЩЕНИЯ

В статье рассматриваются особенности операционной системы специального назначения Astra Linux и её применение в автоматизированных системах контроля радиоэлектронных устройств (РЭУ).

Развитие информационных технологий является одним из ключевых факторов в развитии современного государства. На протяжении многих лет на российском рынке преобладают разработки зарубежных корпораций, в том числе и в сфере оборонной промышленности. Использование зарубежного программного обеспечения (ПО) связано с различными рисками, в том числе утечкой важной информации, направленным сбоем и другими угрозами, которые могут быть использованы в целях ослабления безопасности государства. Поэтому для предприятий ОПК особенно важно иметь ПО отечественных разработчиков, каковым и является операционная система (ОС) Astra Linux.

Разработка данной ОС началась в 2008 году [1]. Её дальнейшее внедрение было обусловлено следованием плану перехода государственных учреждений на свободное ПО [4]. Разработку ведет один из ведущих производителей ПО для государственных структур АО «НПО РусБИТех». В условиях усиливающихся с 2014 года санкций по отношению к РФ, процесс перехода на отечественное ПО получил особую актуальность. В апреле 2015 для обеспечения «устойчивого развития экономики и социальной стабильности» Министерство связи и массовых коммуникаций РФ утвердило План импортозамещения программного обеспечения [5]. Начиная с 2018 года, Минобороны РФ переводит

все военные ПК на Astra Linux и отказывается от применения ОС Microsoft Windows.

Разработка системы ведется на базе ядра Linux. Исходные тексты ядра доступны на сайте разработчика. Существует две ветки разработки ОС:

- Common Edition (Общего назначения);
- Special Edition (Специального назначения).

Издание Common Edition предназначено для среднего и малого бизнеса, образовательных учреждений. В рамках данной статьи рассматривается версия Special Edition, далее кратко представлены основные аспекты данной системы и её использовании при разработке автоматизированных систем контроля.

ОС Astra Linux Special Edition предназначена для построения защищённых автоматизированных систем, обрабатывающих информацию, содержащую сведения, составляющие государственную тайну с грифом не выше «совершенно секретно» [3].

ОС предоставляет пользователям широкие возможности в решении задач, связанных с обработкой информации в условиях сохранения секретности государственной тайны. Для этого ОС оснащена защищённой графической оболочкой и, помимо стандартного пакета офисных программ, включает в себя:

- защищенную СУБД;
- защищенный комплекс программ гипертекстовой обработки данных;
- защищенный комплекс программ электронной почты.

Комплекс средств защиты (КСЗ) предназначен для реализации функций ОС по защите информации от несанкционированного доступа и предоставление пользователю безопасности информации средств управления функционирования КСЗ. КСЗ включает в себя модули подсистемы безопасности, входящие в состав ядра, модули аутентификации и другие компоненты, предоставляющие требуемый уровень безопасности.

Функция идентификации и аутентификации пользователей в Astra Linux основывается на использовании механизма РАМ. Кроме того, в состав операционной системы включены средства поддержки двухфакторной аутентификации.

Стандартная установка ОС включает базовую систему и графический рабочий стол Fly с набором административных и пользовательских графических утилит. Защищённая графическая подсистема в составе ОС функционирует с использованием графического сервера

Хорг. В графическую подсистему по умолчанию встроена мандатная защита.

В ОС используются программные пакеты (далее просто пакеты) в формате .deb. Для управления пакетами в режиме командной строки предназначены наборы команд `dpkg` и `apt`. В графическом режиме управлять пакетами можно с помощью программ `Kpackage` (рабочий стол KDE) и `Synaptic`.

В качестве основной интегрированной среды разработки (IDE) в Astra Linux выступает кроссплатформенная свободная IDE `Qt Creator`.

Из выше описанного следует, что Astra Linux является надёжной и современной ОС, в рамках которой возможно вести разработку программного обеспечения для военных заказчиков. Одним из примеров использования рассматриваемой ОС является разработка программных модулей для автоматизированной системы контроля (АСК).

Развитие радиоэлектроники привело к значительному повышению сложности радиоэлектронных устройств (РЭУ), а также высоким требованиям к качеству изготавливаемых изделий, поэтому так важно иметь средства контроля их разработки.

ААК – это программно-аппаратный комплекс, обеспечивающий выполнение заданного человеком-оператором и программой контроля алгоритма преобразования сигналов и данных для оценки технического состояния ОК и принятия соответствующего решения. В роли ОК выступает заданное техническое устройство [6].

Свои функции ААК осуществляет в составе АСК, которая образуется как совокупность ОК, ААК и человека-оператора на время подготовки и осуществления автоматизированного контроля технического состояния. Обобщенная структура АСК представлена на рис. 1.

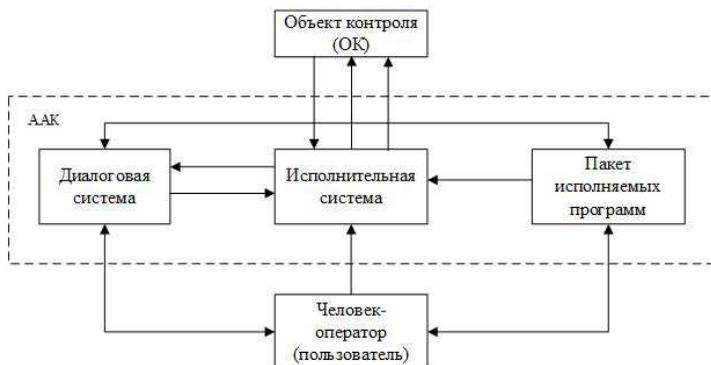


Рис. 1. Обобщенная структура АСК.

В состав ААК входят:

- базовое аппаратного обеспечения (БАО);
- базовое программное обеспечение (БПО);
- базовое методическое обеспечения (БМО).

БПО – это набор программных модулей, совместимых, взаимоконструируемых и настраиваемых на состав аппаратных средств и задачи контроля.

В БПО используются две операционные системы:

- ОС общего назначения;
- базовая операционная система контроля (БОСК).

В качестве базовой ОС общего назначения и выступает Astra Linux SE. Все программные компоненты ААК разрабатываются с использованием языка общего назначения C++. Использование отечественной ОС гарантирует безопасность при выполнении всех этапов создания программных модулей.

В заключение можно сказать, что ОС Astra Linux SE оправдала свой выбор в качестве основной системы для военного заказчика, и удобной системой в повседневном использовании. Она имеет необходимый и достаточный комплекс средств разработки и ПО общего назначения, удобна в использовании и что не мало важно, нетребовательна к аппаратным ресурсам ПК. Совершенствование ОС не прекращается, в настоящее время актуальными являются Astra Linux Special Edition 1.6 (Смоленск) и Astra Linux Common Edition 2.12 (Орёл) [2]

СПИСОК ИСПОЛЪЗЕМЫХ ИСТОЧНИКОВ

1. Astra Linux - Википедия [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/Astra_Linux (дата обращения 15.03.2019).
2. Astra Linux - Универсальная операционная система [Электронный ресурс]. – URL: <https://astralinux.ru/> (дата обращения 15.03.2019).
3. Операционная система специального назначения «Astra Linux Special Edition». Описание применения, Москва, 2010. – 31 с.
4. Распоряжение Правительства Российской Федерации от 17 декабря 2010 г. № 2299-р, Москва.
5. Распоряжение Правительства Российской Федерации от 27 января 2015 г. № 98-р, Москва.
6. Комплекс агрегатных средств автоматизированного контроля (КАСАК-М). Справочные материалы. Общие вопросы построения, Рязань, АО «РКБ «Глобус», 2007. – 260 с.

УДК 004.896

С.А. СОЛОВЬЕВ, В.В. СУСКИН

Рязанский государственный радиотехнический университет

**УЧЕБНО-ИССЛЕДОВАТЕЛЬСКАЯ САПР НА ОСНОВЕ САПР
KICAD**

Рассматривается перспектива алгоритмов и методов решения задач конструкторского проектирования для повышения эффективности использования промышленных САПР.

В известных промышленных САПР печатных плат ЭВА и РЭА используется ограниченный набор алгоритмов и методов решения задач конструкторского проектирования. Разработка новой системы – это длительный и трудоемкий процесс. Многообразие математических методов решения задач конструкторского проектирования, а также необходимость адаптации промышленных САПР к технологическим условиям производства приводят к задаче исследования эффективности использования различных математических методов в промышленных САПР.

Одним из методов решения такой задачи является использование форматов входных и выходных данных отдельных проектных процедур промышленных САПР. Такой подход позволяет использовать все возможности промышленной САПР от использования редактора схем, до объемного моделирования печатной платы и при этом применить новые методы и алгоритмы для проектных процедур и операций. Кроме того, такой подход эффективен в учебных целях, поскольку позволяет охватить весь процесс проектирования печатных плат с помощью промышленных САПР и при этом вносить изменения в математическое и программное обеспечение САПР.

Одним из наиболее важных условий использования промышленной САПР при построении учебно-исследовательской (УИ) САПР является открытость программного обеспечения промышленной САПР - наличие исходных кодов программного обеспечения и форматов входных и выходных данных проектных процедур. Таким требованиям отвечают далеко не все промышленные САПР.

Для использования в УИ САПР была выбрана промышленная САПР KiCad [1], которая свободно распространяется под лицензией GPL [2], что позволяет ее свободно модифицировать, распространять и использовать.

KiCad - это интегрированный, кросс-платформенный (для Linux и Windows) комплекс программ в исходных кодах и свободной лицензией, предназначенный для разработки электронных схем и автоматизированной разводки печатных плат.

Программный комплекс состоит из нескольких основных частей:

Eeschema — редактор схем;

CvPcb — сопоставление УГО компонентов и посадочных мест;

Pcbnew — трассировщик печатных плат;

GerbView — просмотр и редактирование gerber-файлов;

Bitmap2Component — утилита для добавления изображений;

Pcb_calculator — удобный калькулятор;

KiCad — менеджер проекта и всего этого безобразия [3].

В рамках разработки УИ САПР на основе САПР KiCad была разработана визуальная среда проектирования программного обеспечения проектной процедуры размещения (Таблица 1).

Таблица 1

Позволяет:		
1. Выбирать из библиотеки альтернативный программный модуль для размещения элементов на печатной плате.	2. Выбирать маршрут проектирования для проектной процедуры размещения на основе использования библиотеки программных модулей проектных операций.	3. Изменять состав библиотек проектных операций и проектных процедур.

Отличительной особенностью визуальной среды проектирования является построение библиотек по двухуровневой схеме, позволяющей формировать программные модули проектной процедуры размещения на основе выбора программных модулей проектных операций.

Для демонстрации работы визуальной среды проектирования учебно-исследовательской САПР были разработаны несколько программных модулей проектной процедуры размещения и проектных операций.

Программная оболочка визуальной среды проектирования обеспечивают чтение форматов выходных данных редактора схем и получение формата входных данных для проектной процедуры трассировки САПР KiCad и разработана в визуальной среде программирования C++ Builder 6.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. <http://ru.wikipedia.org/wiki/Kicad>
2. <http://www.gnu.org/licenses/gpl.html>
3. <http://www.radiokot.ru/start/soft/KiCad/01/>

УДК 004.382

А.М. СТРУКОВ

Рязанский государственный радиотехнический университет

**ИССЛЕДОВАНИЕ ПОДХОДОВ К ПАРАЛЛЕЛЬНОЙ
РЕАЛИЗАЦИИ МЕТОДА КАСКАДОВ**

Рассматриваются подходы распараллеливания алгоритма синтеза логических схем методом каскадов на центральном и графических процессорах

Метод каскадов основан на разложении Шеннона, согласно которому любая логическая функция, не тождественная нулю, представима в виде дизъюнкции конъюнкций. Он позволяет при исключении k переменных свести реализацию булевой функции от n переменных к реализации функции от $n - k$ переменных. Таким образом, предельное разложение Шеннона булевой функции $f(x_1..x_n)$, не тождественной нулю, имеет вид:

$$f(x_1..x_n) = \bigvee_{\forall (f=1)} \bigwedge_{i=1}^n x_i^{a_i}$$

При исключении переменной x_i из исходной булевой функции $f(x_1..x_n)$ получаем остаточные функции с константой вместо исключенной переменной: единичную $f(x_i = 1)$ и нулевую $f(x_i = 0)$.

В свою очередь, полученную функцию с меньшим числом переменных можно минимизировать путем фиксации оставшихся переменных до тех пор, пока остаточные функции не будут иметь размерность 2 – дальнейшее упрощение в таком случае будет бессмысленно, так как эффективна непосредственная схемная реализация остаточных функций от двух переменных в любом бинарном базисе [1].

Переменные исключаются последовательно. Критерием исключения служит вес частной производной первого порядка от логической функции. Частная производная первого порядка – результат сложения по модулю два единичной и нулевой остаточных функций:

$$\frac{\partial f}{\partial x_i} = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n) \oplus f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$$

Вес частной производной равен числу конституэнт этой производной. Таким образом, синтез схемы можно свести к последовательному вычислению остаточных функций путем исключения переменной из функции, полученной на предыдущем этапе или же заданной в качестве исходной. Наиболее трудоемкой операцией при реализации данного метода является вычисление частной производной.

Результат работы метода каскадов можно представить в виде двоичного дерева, где исключенные переменные являются узлами, а логические константы или остаточные функции от двух переменных – листьями. Пример схемной реализации результата работы алгоритма представлен на рис. 1.

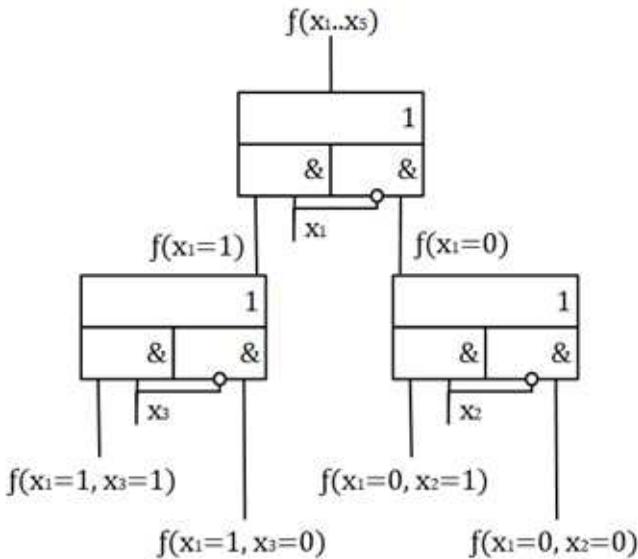


Рис. 1. Двухъярусная каскадная схема в базисе И-ИЛИ-НЕ.

Распараллеливание метода каскадов значительно сократит время синтеза схемы исходной функции. Метод не предполагает конкурентного доступа к ресурсам, так как исходные данные в ходе работы алгоритма не изменяются. Таким образом, функцию исключения перемен-

ной и формирования остаточной функции можно реализовать с использованием рекурсивного подхода.

При распараллеливании средствами ЦПУ для обработки новых запросов на вычисления, возникающих в ходе работы алгоритма можно использовать пул потоков – потоки создаются один раз при старте работы программы, отдельные задачи между ними распределяются средой выполнения оптимальным образом. Еще одним достоинством применения пула является уменьшение накладных расходов, связанных с ручным созданием и завершением потоков для каждого фрагмента кода, допускающего распараллеливание [2].

В данном случае приоритетная обработка тех или иных потоков нецелесообразна из-за того, что бинарное дерево, представляющее схему, является неполным (степень узлов равна 0, 1 или 2). Следовательно, при итеративном подходе к отрисовке полученная схема будет избыточной. Это критично, если учесть, что увеличение количества каскадов носит характер геометрической прогрессии. При вычисленном дереве при помощи обхода в обратном порядке (LRT) можно вычислить ширину каждого яруса, и использовать ее в расчете позиции для каждого элемента схемы.

При распараллеливании средствами графического процессора на реализацию накладываются другие условия, в зависимости от выбранной архитектуры. Например, программы с использованием CUDA выполняются только на ГП NVidia, а в OpenCL нет рекурсивных вызовов. В данном случае оптимальная реализация – итеративный подход.

Остаточные функции, полученные при работе алгоритма на определенном шаге, будут использованы в качестве входных на следующей итерации. Исключенные переменные, константы и тупиковые остаточные функции будут записываться в глобальную память как результат. Основная сложность данного подхода – реализация эффективного доступа к памяти и алгоритма ее выделения между итерациями.

Таким образом, метод каскадов имеет в своей основе алгоритм, позволяющий легко использовать преимущества параллельных вычислений. Организация параллельных вычислений позволит значительно уменьшить время работы алгоритма. При реализации работы алгоритма на графическом процессоре есть возможность параллельного формирования не только остаточных функций, но и непосредственно весов производных, необходимых для исключения переменной. А реализация алгоритма на ЦПУ покажет лучшее время работы при небольшой размерности входной функции из-за отсутствия трудоемких операций с памятью.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Дискретная математика: Учеб. пособие / А. М. Гостин, В. П. Корячко, М. А. Бакулева, А. В. Бакулев - РГРТУ, 2011. - 135 с.
2. Численные методы, параллельные вычисления и информационные технологии: Сборник научных трудов / Под ред. Вл. В. Воеводина и Е. Е. Тыртышников. – М.: Издательство Московского Университета, 2008. – 320 с.

УДК 53.088

В.П. ТРУХАЧЕВ

Рязанский государственный радиотехнический университет

РАЗРАБОТКА И ВНЕДРЕНИЕ РУКОВОДСТВА ПО КАЧЕСТВУ КАЛИБРОВКИ СРЕДСТВ ИЗМЕРЕНИЙ

Рассматриваются возможность и актуальность разработки и внедрения руководства по качеству калибровки средств измерений в метрологической сфере.

В последние десятилетия проблема качества измерений не только не ослабла, но, напротив, значительно обострилась. Это связано с действием ряда объективных причин, обусловленных бурным техническим прогрессом. Одна из причин – непрерывный рост сложности аппаратуры, который значительно опережает рост качества элементной базы, а также совершенствование производства, обеспечение качества и конкурентоспособности продукции, создание современной базы документов по метрологии по всем процессам жизненного цикла продукции и постоянное улучшение действующей системы менеджмента качества: все эти факторы в совокупности сделали актуальным разработку руководства по качеству калибровки средств измерений (далее – «Руководство по качеству»), так как, например, проблема обеспечения высокого качества продукции тесным образом связана с проблемой качества измерений. Между ними явно прослеживается непосредственная связь: там, где качество измерений не соответствует требованиям технологического процесса, невозможно достичь высокого уровня качества продукции [1].

Руководство по качеству организации и выполняемых калибровочных работ - документ, устанавливающий цели, методы и процедуры, позволяющие метрологической службе или калибровочной лаборатории решать задачи, определяемые положением о них [2].

Руководство по качеству описывает систему качества в соответствии с установленной политикой в области качества и целями, а также применяемыми стандартами. Для калибровочных лабораторий таким стандартом прежде всего является национальный стандарт ГОСТ ИСО/МЭК 17025-2009 «Общие требования к компетентности испытательных и калибровочных лабораторий». Несмотря на широкую область применения этого стандарта необходима интерпретация его требований с учетом специфики сферы деятельности лаборатории. Такая специфика может быть изложена в специальных документах. Примерами таких документов являются:

- Руководство по качеству в аналитической химии Руководство ЕВРАХИМ/СИТАК (2003);

- Руководство для судебно-экспертных лабораторий» ИЛАК Руководство-19

- Руководство по аккредитации лабораторий выполняющих микробиологический анализ Руководство ЕВРАХИМ/ЕА (2002);

- Руководство по аккредитации лабораторий органолептического анализа ЕА-4-09.

Не существует единых правил о том, как писать руководство по качеству. Тем не менее, можно выделить несколько общих требований к разработке Руководства по качеству:

1) для описания каждой процедуры следует выделить отдельный раздел Руководства. Каждый раздел рекомендуется начинать на новой странице. Все страницы должны быть пронумерованы. Для удобства рекомендуется ввести колонтитул, в котором можно указать:

- наименование документа;
- наименование лаборатории;
- рег.№ РК;
- номер раздела;
- дату утверждения;
- общее количество страниц;
- номер страницы.

2) объем Руководства не должен быть слишком большим, т. к. это затруднит пользование документом. Следует избегать деталей и ненужных подробностей.

3) для того чтобы сократить объем Руководства некоторые документированные процедуры целесообразно оформлять в виде отдельных документов. (Например, Инструкция «Управление документацией и записями»). Ссылки на эти документы должны быть даны в Руководстве.

4) руководству должен быть присвоен регистрационный номер.

5) руководство утверждается руководителем лабораторией.

6) с Руководством должны быть ознакомлены все сотрудники лаборатории.

7) описание процедуры должно начинаться с формулировки цели внедрения данной процедуры в систему менеджмента лаборатории. Далее должны быть выделены и описаны процессы, являющиеся составляющими описываемой процедуры. Обязательно должны указываться лица, ответственные за выполнение процедуры. В описании должны быть даны ссылки на внутренние документы лаборатории (журналы, папки, протоколы) в которых регистрируются данные, получаемые в ходе выполнения процедуры.

Титульный лист Руководства по качеству должен содержать утверждающую подпись руководителя лаборатории, наименование организации, регистрационный номер, редакцию, дату введения документа. Не следует предусматривать утверждение РК руководством юридического лица, а также органом по аккредитации.

Порядок построения Руководства по качеству носит необязательный характер и зависит от специфики лаборатории. В соответствии с ИСО 10013 "Руководящие указания по разработке руководств по качеству" в его состав рекомендуется включать следующее:

- название, назначение и область применения;
- оглавление;
- вводную часть, включающую сведения об организации и самом руководстве;
- политику и цели организации в области качества;
- описание организационной структуры, ответственности и полномочий;
- описание элементов системы качества и любые ссылки на документированные процедуры системы качества;
- раздел "Определения", если это необходимо;
- путеводитель к руководству по качеству, если необходимо;
- приложения с дополнительной информацией, если необходимо.

Разработка руководства по качеству калибровки средств измерений позволит узнать фактическую погрешность какого-либо устройства, сократить время выполнения работы, снизить затраты на устранение погрешностей и обеспечить качество измеряемых приборов (устройств).

Создание данного документа в наше время очень актуально, а также экономически выгодно, так как мы повысим уровень качества

измерительных приборов, тем самым увеличим прибыль за счет увеличения числа заказчиков и значительно повысим эффективность работы организации в целом [3].

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. ПР 50.2.016-94 ГСИ «Требования к выполнению калибровочных работ».
2. Окрепилов В. В. Основы метрологии: учебное пособие. – СПб.: ГУАП, 2008. – 380 с.
3. Артемьев Б.Г. Метрология и метрологическое обеспечение. – М.: Стандартиформ, 2010. – 568 с.

УДК 004.94

А.М. ФАМ, А.П. ШИБАНОВ

Рязанский государственный радиотехнический университет

КОМБИНИРОВАННАЯ СИСТЕМА МОДЕЛИРОВАНИЯ НА ОСНОВЕ СЕТЕЙ GERT И ПЕТРИ

Рассматриваются вопросы создания комбинированной системы моделирования на основе GERT-сетей и сетей Петри. Исследуются вопросы применения GERT-сетей с положительными и отрицательными случайными величинами для активизации входных и выходных позиций сети Петри.

Сети Петри представляют собой двудольный ориентированный граф, состоящий из дуг и соединяющих их вершин двух типов – позиций и переходов (см. рис. 1).

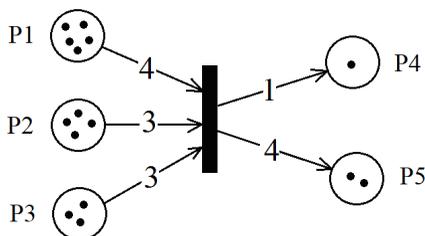


Рис. 1. Сеть Петри с тремя входными, двумя выходными позициями и одним переходом.

Сети Петри – аппарат для моделирования преимущественно асинхронных параллельных процессов [1]. Динамические объекты изображаются метками (маркерами) внутри позиций. Статическим объектам соответствуют вершины сети Петри – входные и выходные позиции и переходы.

Распределение маркеров по позициям называют *маркировкой*. Маркеры могут перемещаться в сети. Каждое изменение маркировки называют событием, причем каждое событие связано с определенным переходом. Каждому условию в сети Петри соответствует определенная позиция. Совершению события соответствует срабатывание перехода, при котором маркеры из входных позиций этого перехода перемещаются в выходные позиции. Последовательность событий образует моделируемый процесс.

Рассмотрим сначала случай, когда переход срабатывает, если для *каждой* из его входных позиций выполняется условие $N_i \geq K_i$, где N_i – число маркеров в i -й входной позиции, K_i – число, характеризующее связь i -й позиции с переходом ($i = \overline{1, n}$ n – число входных позиций); при срабатывании перехода число маркеров в i -й входной позиции уменьшается на K_i , а в j -й выходной позиции увеличивается на M_j , где M_j – число характеризующее связь перехода с j -й выходной позицией. В этом случае входные позиции перехода реализуют логическую функцию И.

Возможен случай, когда входные позиции реализуют логическую функцию ИЛИ. Причем срабатывание перехода происходит тогда, когда срабатывает *один* (первый по времени) из входов. Только для данной входной позиции выполняется условие $N \geq K$, где N – число маркеров во входной позиции, K – число, характеризующее связь позиции с переходом; при срабатывании перехода число маркеров только в сработавшей входной позиции уменьшается на K , а в j -й выходной позиции увеличивается на M_j , где M_j – число характеризующее связь перехода с j -й выходной позицией.

На рисунке 1 показан пример распределения маркеров по позициям перед срабатыванием.

При выполнении логической функции И эту маркировку записывают в виде (5, 4, 3, 1, 2). После срабатывания перехода маркировка становится иной: (1, 1, 0, 2, 6).

При выполнении логической функции ИЛИ после срабатывания одного перехода P1 получается маркировка (1, 4, 3, 2, 6).

Можно вводить ряд дополнительных правил, получая ту или иную разновидность комбинированных сетей Петри-GERT. Можно исследовать не только последовательность событий, но и их привязку ко времени. Это осуществляется приданием переходу веса – продолжительности срабатывания, которую можно определять как выборочные значения случайной выходной случайной величины некоторой GERT-сети. Полученную модель можно назвать комбинированной *стохастической* сетью Петри-GERT. Сети GERT моделируют последовательность выполняемых операций с детерминированным или вероятностным выбором следующей из них. Возможно повторное выполнение операций. В классической теории GERT-сетей для указания продолжительности операций используются основные распределения теории вероятностей: биномиальное, дискретное общего вида, экспоненциальное, гамма, геометрическое, отрицательное биномиальное, нормальное, равномерное, Пуассона. Это множество может быть расширено другими распределениями, для которых известна их характеристическая функция.

Сети GERT могут отражать сложные производственные, информационные и экономические процессы. Их использование в комбинированной системе вместе с сетями Петри для расширения функциональных возможностей моделирования различных процессов не вызывает сомнения.

В стохастических сетях возможно введение вероятностей окончательного срабатывания переходов, для которых выполнены предварительные условия срабатывания.

Рассмотрим возможности комбинированной системы Петри-GERT, когда число маркеров и числовые характеристики входных и выходных позиций заданы случайными величинами. На практике часто возникают ситуации, когда невозможно охарактеризовать эти параметры числами, но имеется информация об их законах распределения. Можно, например, использовать принятие гипотезы о соответствии экспериментальных данных некоторому распределению по критерию χ^2 . Информация о числе меток может быть определена через случайную сумму случайных слагаемых.

При использовании случайных величин надо определить операции входных позиций.

1. Число меток во входной позиции есть случайное число с известным законом распределения.
2. Число меток во входной позиции определяется суммой случайного числа случайных слагаемых.

3. Число, связанное с позицией, есть случайное с известным законом распределения.

4. Число, связанное с позицией, определяется суммой случайного числа случайных слагаемых.

Эти случаи могут комбинироваться в различных сочетаниях друг с другом. Во всех из них присутствует случайное число меток ξ_1 в позиции и связанное с позицией случайное число ξ_2 . Для активизации входа необходимо выполнение условия $\xi_1 - \xi_2 \geq 0$. При срабатывании входа во входной позиции должно остаться число меток, определяемое этой разностью.

Обозначим случайные числа, связанные с выходными позициями перехода, через ξ_3 , а число меток, находящихся в выходной позиции, через ξ_4 . При срабатывании перехода произойдет изменение числа маркеров по правилу $\xi_3 + \xi_4$. Эти метки могут быть, в свою очередь, метками входной позиции другого перехода.

В комбинированной системе моделирования сетей Петри и GERT необходимо выполнять математические действия над законами распределения случайных величин. Для этого удобен аппарат GERT-сетей.

Теория GERT-сетей используется:

- для вычисления разностей случайных величин $\xi_1 - \xi_2$ и анализа возможной активизации некоторой входной позиции;
- для определения выборочного значения выходной случайной величины GERT-сети, определяющей задержку срабатывания перехода;
- для вычисления числа меток в выходных позициях перехода по формуле $\xi_3 + \xi_4$.

Заключение. В следующих публикациях предполагается изложить результаты исследований различных случаев определения величин $\xi_1 - \xi_2 \geq 0$, $\xi_3 + \xi_4$ как сложения и вычитания случайных величин в GERT-сетях.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Норенков И.П. Системы автоматизированного проектирования (САПР). Издательство МГТУ им. Н.Э. Баумана, 2002.

УДК 004.5

К.Ю. ФИЛАТОВ

Рязанский государственный радиотехнический университет

СОВРЕМЕННЫЕ ТЕНДЕНЦИИ ЮЗАБИЛИТИ

Рассматриваются некоторые тенденции в построении интерфейсов современных информационных систем.

В современном мире, если вы хотите позиционировать себя, как сильная компания, способная разрабатывать конкурентоспособный продукт, вам необходимо уделять должное внимание вопросам построения пользовательских интерфейсов. Целевую аудиторию должны привлекать не только особенности вашего дизайна, но также следует обеспечить им идеальное юзабилити или, проще говоря, приятное времяпрепровождение при взаимодействии с сайтом или приложением. К счастью, этот вопрос не нов и многие разработчики охотно делятся своим опытом. В данной статье мы рассмотрим некоторые интересные подходы, связанные с UX-дизайном (User Experience дизайн — многогранная концепция, которая включает множество дисциплин: интерактивный дизайн, информационную архитектуру, визуальный дизайн, взаимодействие между человеком и компьютером), которые однозначно улучшат юзабилити вашего продукта.

Дизайн «на опережение». Данный подход завоевал широкую популярность. Это функциональность, которая основывается на опережение возможных действий пользователя. Более сложные ее представления привели к разработке сравнительно новых шаблонов UX-дизайна. Так опережающий дизайн (Anticipatory Design) уже успешно интегрирован в различные веб-сайты и приложения. Задача ясна — предвидеть поведение пользователей с целью дальнейшего упрощения определенных процессов. На многих интерфейсах при заполнении, к примеру, некоторых данных или анкет, система предлагает вам автоматический ввод вашего адреса, имени и других важных деталей. Тем самым помогает быстро и без хлопот заполнить форму, что существенно ускоряет работу с системой. В настоящее время опережающий дизайн имеет своей целью свести к минимуму потенциальную нагрузку на пользователей за счет принятия решений за них (конечно же, с предварительным запросом на разрешение такого действия).

Wow – эффект. Данный подход предполагает уход от традиционных баннеров и каруселей на главном изображении вашего интерфейса в сторону применения искусно интегрированных красивых

главных изображений (Hero Images) в высоком разрешении. Такие изображения вызывают wow-эффект. Если ваша компания занимается производством тортов, то главное изображение, включающее аппетитное изображение вашей продукции, создаст тот самый wow-эффект. Использование красивого и уместного hero image также способствует удержанию пользователей на вашем веб-сайте или приложении в течение более длительного периода времени. Многие компании размещают на главном изображении видео, где показана, к примеру, рабочая атмосфера офиса: главное изображение демонстрирует, как сотрудники преданы работе и в каких условиях создается продукт компании. Опять же здесь стоит понимать, что это должно быть уместно.

Сторителлинг, ориентированный на пользователя. Дизайн сторителлингом (приём, использующий медиа-потенциал с целью передачи информации и транслирование смыслов посредством рассказывания историй) является одним из наиболее эффективных методов, позволяющих захватить внимание. Если вам нужно завлечь и убедить вашу целевую аудиторию, то подобные короткие рассказы, чтение и просмотр которых занимает совсем немного времени (обычно 2-3 минуты), будут весьма эффективны. Бывают случаи, когда создать историю трудно. Потребуется уделить должное внимание деталям и использовать творческие подходы для написания увлекательной истории о вашем бренде, продукте, услуге или концепции. Обычно применяются следующие методы сторителлинга:

1. Ваши пользователи – герои истории.
2. Применение продвинутых визуальных эффектов.
3. Добавление интерактивных типовых действий.
4. История должна быть целостной – она имеет начало, основную часть и конец.

Убеждающий дизайн. Здесь нужно понять, что убеждающий дизайн (Persuasive Design) должен внушать посетителям сайта, что вы предлагаете хорошую сделку. Для этого необходимо кропотливо проанализировать предметную область и понять психологию потенциальных пользователей, после чего захватить внимание вашей целевой аудитории и сохранить его. Этот тренд UX-дизайна имеет значение для формирования большой клиентской базы. Перечислим основные принципы данного подхода: ясность, визуальная привлекательность, строгая визуальная иерархия, удержание внимания пользователя любой ценой.

Работа над дизайном для переносных устройств. Сегодня практически у каждого человека есть смартфон, поэтому ваш продукт должен иметь возможность воспроизводиться в том числе и на переносных

девайсах. Так же современная тенденция это массовое использованию искусственного интеллекта (AI) и виртуальной реальности (VR), которые являются причиной развития различных интерактивных устройств и новых способов взаимодействия.

Анимация в дизайне. Анимация в дизайне призвана удерживать внимание пользователей на важных для вашей компании моментах во время их взаимодействия с приложением или сайтом. Она может направлять людей дальше по сайту или побуждать выполнять их определенные действия. Например, при использовании всеми известного приложения «ВКонтакте», вы видите анимацию «Иван печатает...». Такая наглядность оказывает замечательный эффект: она удерживает вас в ожидании нового сообщения и помогает управлять временем. Аналогично, если вы интегрируете интерактивную анимацию в свой дизайн, будет меньше шансов, что вашим пользователям станет скучно.

Выше были представлены основные и наиболее интересные тенденции построения пользовательских интерфейсов, удовлетворяющие требованиям юзабилити, которые однозначно сделают ваш продукт более интересным для пользователя.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Принципы и правила юзабилити сайта для мастеров веб-дизайна [Электронный ресурс]. - <https://webformyself.com/principyu-i-pravila-yuzabiliti-sajta-dlya-masterov-veb-dizajna/>

2. 6 трендов UX-дизайна, которым стоит следовать [Электронный ресурс]. - <https://lpgenerator.ru/blog/2016/06/29/6-trendov-ux-dizajna-kotorym-stoit-sledovat-v-2016-godu/>

УДК 004.05

В.В. ФИЛИМОНОВ, В.В. СУСКИН

Рязанский государственный радиотехнический университет

ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ СОВРЕМЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Рассматриваются современные методы оценки производительности, применяющиеся при эксплуатации и разработке ЭВМ, анализируются их достоинства и недостатки, а также возможность использования на высокопроизводительных системах.

В настоящее время требование к производительности вычислительных систем растет очень быстро, перед пользователями абсолютно разных сфер деятельности все чаще возникает потребность в решении сложных задач. Как следствие, производители ЭВМ, стараются удовлетворять потребности клиентов и постоянно совершенствовать свою продукцию. Однако важно понимать, что производительность современных вычислительных систем зависит от множества факторов и даже от предназначения, поэтому задача ее корректной оценки стала сложной, но необходимой как для выбора, так и для производства.

В общем случае, производительность оценивают как временной показатель, показывающий скорость реакции системы на определенную последовательность задач. Стоит отметить, что современные ЭВМ имеют сложную архитектуру и способны выполнять не только математические вычисления, но и графические, следовательно, некорректно оценивать общую производительность системы мощностью одного лишь процессора. Стоит учитывать и скорость обмена данными между компонентами, материнскую плату, видеокарту, оперативную память и т.д.

Для повышения точности оценки производительности необходимы специальные методы, а в некоторых случаях и отдельное тестирование компонентов.

В конце XX века чаще всего использовали метод Гибсона, который был основан только на оценке быстродействия ЭВМ, пренебрегая оценкой качества выполненных задач, что собственно и является его главным недостатком.[1]

Второй метод, Аналитические модели. Он основан на математическом представлении системы или ее компонентов, иногда с помощью него можно быстро и относительно точно оценить производительность системы, однако чаще он сводится к построению простейших моделей не способных полноценно описать исследуемый объект.

Третий метод, Эталонные потоки заданий или Benchmark. Данный метод наиболее распространен на сегодняшний день. Его суть заключается в использовании эталонных наборов программ, причем как специально созданных для тестирования системы, так и программ имеющих практическое применение.[3] Главным плюсом метода является возможность создания такого теста, который будет оценивать не только отдельные компоненты, но и их взаимосвязь между собой. Однако есть и серьезные недостатки:

- 1) Весь набор программ составляется людьми, которые заинтересованы в результатах.

2) Под каждую систему нужен свой уникальный эталонный тест, выявляющий требуемые характеристики.

3) Продолжительность проведения теста охватывающего большую часть системы может занимать десятки часов.

Четвертый метод, Стохастические модели. Он характеризуется последовательностью заданий, состоящих из запросов к ресурсам системы, которые распределяются на основе статистического анализа, из-за чего точность оценки данного метода зачастую ниже 50%. К тому же, метод является системно зависимым, то есть требует доработки под определенную архитектуру, и сам сбор информации вносит дополнительную погрешность.

Пятый метод, Трассы. Для его реализации необходимо сначала собрать данные работающей системы, затем на их основе составить модель последовательности задач, с содержанием этих данных.[2] Аналогично стохастическим моделям метод имеет значительную погрешность в результатах и неприменим для разработки новых вычислительных систем, его использование может быть актуально только для настройки и оптимизации существующих.

При оценке производительности отдельных компонентов стоит учитывать их характеристики и конкретные задачи, которые они решают.

Главной аппаратной составляющей любой современной ЭВМ, отвечающей за скорость вычислительных операций, как известно, является центральный процессор (ЦП). Его основными характеристиками, исходя из которых, можно сделать предварительную оценку производительности являются:

- Тактовая частота, измеряемая в гигагерцах (Гц),
- Количество ядер
- Количество потоков
- Разрядность
- Объем кэш-памяти

Однако при оценке в целостной системе, стоит также учитывать такие показатели как загруженность, время выполнения пользовательских и системных задач, время ожидания отклика и время простоя. [4] Для возможности сравнения ЦП по специальным альтернативным показателям, в современной литературе, используются такие характеристики как:

CPI (Clock cycles Per Instruction) – среднее количество тактов синхронизации на одну команду

MIPS (Microprocessor without Interlocked Pipeline Stages) - миллион команд в секунду.

MFLOPS (Millions of FLoating point OPerations per Second) – миллион операций в секунду над числом с плавающей точкой. На данный момент данная характеристика является одной из самых популярных в компьютерной промышленности, но несмотря на это, рейтинг MFLOPS, не может характеризовать современные вычислительные системы как универсальная единица измерения производительности.

Помимо ЦП, немало важной аппаратной составляющей современной ЭВМ, является видеокарта (GPU), данный компонент отвечает за обработку и воспроизведение графических данных системы. В настоящее время, оценить производительность GPU можно по следующим характеристикам:

- Тактовая частота
- Fillrate (скорость отрисовки пикселей)
- Объем памяти, ее тип и частота
- Ширина шины, влияющая на пропускную способность памяти.

В качестве универсальных средств тестирования и общих единиц измерения производительности видеокарт у персональных компьютеров, часто используют показатели специальных графических Benchmark.

Еще один компонент, требующий отдельного внимания, это оперативная память (ОЗУ), она предназначена для хранения промежуточных данных, которые обрабатывает процессор. Следовательно, от ее производительности будет существенно зависеть общая оценка системы. Наиболее важные характеристики ОЗУ:

- Объем памяти
- Тип памяти
- Частота ОЗУ
- Тайминги (временные задержки между операциями)

В отличие от предыдущих компонентов ОЗУ не тестируется отдельно от других объектов архитектуры вычислительной системы, из-за чего оценка ее производительности происходит при помощи специальных программных средств мониторинга. Впрочем, все остальные компоненты ЭВМ, аналогично оперативной памяти, тестируются в составе целостной системы и имеют собственные характеристики, отражающие общую производительность.

Рассмотренные нами методы оценки вычислительных систем, имеют широкое применение, как для персональных компьютеров, так и для ЭВМ специального назначения. Однако все они сильно связаны с архитектурой компьютера и не способны в полной мере решить проблему для высокопроизводительных и кластерных систем, имеющих сложную компоновку. Таким образом, для получения более точной

характеристики поведения сверхмощных вычислительных систем, требуется разработка уникальной методологии, которая будет учитывать архитектуру и предназначение конкретной ЭВМ.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Смелянский Р. Л. Метод и средств анализа производительности систем с традиционной архитектурой. // М.: Изд-во, 1989.
2. Каляев А.В., Гузик В.Ф. и др. Параллельная обработка информации. // Киев: Наукова думка, 1988.
3. Kaiwalia Dixit. SPECulations. Defending the SPEC Benchmark // Sun Tech Journal, 1991.
4. Оценка производительности вычислительных систем Электронный ресурс. // <http://citforum.ru>.

УДК 004.03

В.В. ФИЛИМОНОВ, В.В. СУСКИН

Рязанский государственный радиотехнический университет

МЕТОДЫ УВЕЛИЧЕНИЯ ВЫЧИСЛИТЕЛЬНОЙ ПРОИЗВОДИТЕЛЬНОСТИ

Рассматриваются перспективы современных архитектур вычислительных систем и анализируются существующие методы повышения производительности. Выявляются достоинства и недостатки используемых способов организации вычислительных систем, а также проблемы их создания, реализация которых позволит наиболее эффективно использовать доступные ресурсы ЭВМ.

Создатели вычислительных машин, с момента их первого появления, стремились к увеличению скорости работы. Разумеется, единого уникального способа решения данной проблемы нет, так как обычно для повышения производительности вычислительных систем используют все возможные методы комплексно. Обработка данных в современных системах осуществляется с помощью центрального процессора (ЦП). Его основной характеристикой, является производительность, следовательно, практически все способы повышения эффективности работы связаны с этой частью системы. Сразу отметим, что под процессором подразумевается аппаратная часть компьютера, которая представляет собой микросхему, установленную в специальный разъем материнской платы.

Большую часть методов повышения производительности вычислительных систем можно классифицировать на интенсивные и экстенсивные, их еще называют качественные и количественные соответственно. Интенсивные методы предполагают повышение быстродействия работы процессора при неизменном количестве, экстенсивные — увеличение количества ЦП с целью суммировать их производительность [1]. Многопроцессорные архитектуры обычно применяются в том случае, когда остальные способы уже использованы и стали не эффективны. Стоит отметить, что увеличение количества процессоров возможно только тогда, когда задача может быть распараллелена.

Проанализировав рейтинг самых быстродействующих ЭВМ мира, можно заметить, что как минимум, первые 500 позиций занимают суперкомпьютеры, использующие кластерную архитектуру, основанную на принципах массового параллелизма (MPP, massively parallel processing) [3]. Для повышения быстродействия таких систем используется два способа:

1) Численное увеличение узлов и их ускорение. Для суперкомпьютеров этот способ является основным, количество узлов в них иногда достигает десятков тысяч, однако такое улучшение сопровождается мегаваттами потребляемого электричества и как следствие большими системами охлаждения.

2) Нарастивание числа вычислительных ядер внутри одного процессора. Реализовать данный способ гораздо сложнее предыдущего, так как здесь происходит увеличение быстродействия системы без изменений типоразмера ЦП. Благодаря этому, метод получил большое распространение не только в суперкомпьютерах, но и в современных домашних системах.

Процесс нарастивания вычислительных ядер осуществляется установкой нескольких одноядерных чипов в один корпус, или созданием специального чипа с несколькими ядрами. В современных процессорах используется комбинация этих методов, благодаря чему получают процессоры с 4 и более ядрами, например Intel Core 2 Quad имеет два двуядерных чипа [4]. Рассмотрим плюсы и минусы такой компоновки.

Достоинства:

- увеличение количества ядер на единицу физического объема системы

- высокоскоростной обмен данными через межъядерные связи и общую кэш-память

Недостатки:

- общие межъядерные связи и кэш-память являются конкурентно разделяемыми ресурсами ядер
- увеличение энергопотребления и температуры нагрева (отвод тепла с микросхемы в компактном корпусе бывает затруднителен)
- более низкая максимальная рабочая частота на ядре, обусловленная отсутствием возможности согласовать характеристики нескольких ядер.

Учитывая все сложности, метод улучшения параметров системы через увеличение числа её компонентов используется уже много лет. В настоящее время наиболее часто встречаются: RAID-массивы жестких дисков, использование 2 и более видеокарт в одной системе, несколько мониторов, двух-трех канальные контроллеры памяти и несколько многоядерных процессоров.

Стоит отметить, что большая часть программ, предназначенных для домашнего использования, плохо распараллеливается на несколько потоков, и не задействует все ядра. Даже в настоящее время, многие ресурсоёмкие приложения практически не ускоряются при числе ядер свыше двух. Следовательно, увеличение производительности самого ядра необходимо, особенно, если оно одно. Как мы уже говорили, эта проблема является самой сложной.

На данный момент, наиболее часто для увеличения частоты используется следующие способы. Физический – заключается в уменьшении размеров проводников и транзисторов на чипе, благодаря этому повышается работоспособность и уменьшается потребление энергии. По нашему мнению, это одна из главных причин, почему при разработке отказываются от ТТЛ и ЭСЛ логики в пользу КМОП, обусловлено это тем, что при использовании ТТЛ и ЭСЛ уменьшение размеров не дает нам таких масштабных преимуществ. Второй способ – архитектурный. Он заключается в увеличении максимальной частоты, посредством разделения операций процессора на отдельные стадии, таким образом, организуя вычислительный конвейер, а в дальнейшем увеличением числа этих стадий. Такой тип организации процессора, с двумя и более конвейерами, называется Суперскалярный.

Помимо увеличения максимальной частоты ЦП, применяют еще несколько технологий.

Первая из них, переименование регистров. Она применяется для обхода архитектурного ограничения (ослабления взаимосвязей команд) с последовательным выполнением инструкций и дает возможность для реализации параллелизма. Изменение наименования общих регистров, приводит к дублированию технических ресурсов, а для записи промежуточных результатов устанавливается соответствие логи-

ческих имен и физических результатов. Процессоры с переименованием регистров фактически имеют более восьми общих регистров, т.е. в процессоре дублируются технические ресурсы. При записи промежуточных результатов устанавливается соответствие логических имен и физических регистров [2].

Использование технологии переименования позволяет осуществить одновременное выполнение нескольких инструкций, которые ссылаются на одно и то же логическое имя регистра, при условии, что отсутствует фактическая зависимость данных.

Вторая технология - продвижение данных. Она используется для оптимизации конвейерных процессоров и позволяет повысить производительность за счет исполнения инструкций раньше готовности операндов. После чего декодированная инструкция с одним операндом помещается в исполнительное устройство, где ожидает второй операнд с другого конвейера.

Третья технология - предсказание переходов. Она предназначена для снижения затрат на ветвление, путем продолжения выборки и декодирования потока инструкций после выборки условного перехода, без проверки условия. Все потоки выборки и декодирований направляются по одной из ветвей.

Четвертая - статические предсказания. Данная технология считается наиболее простой. Она основана на сборе статистики переходов, на базе которой определяется выполнение или не выполнение различных типов переходов.

Пятая - динамическое предсказание. Технология, часто применяющаяся в современных компьютерных системах, основанная на анализе истории вычислительного процесса и фиксации статистики поведения.

Шестая – спекулятивное исполнение (исполнение по предположению). Она основана на выполнении задач до востребования, что позволяет сократить задержку получения результата, однако здесь могут быть ситуации, когда решение не будет использовано и результаты игнорируются.

Седьмая технология - динамическое исполнение. Ее основной принцип заключается в изменении порядка выполнения инструкций, что свойственно RISC-архитектуре, но реализуется и в настоящее время, например для процессоров x86 [5]. Также данная технология характеризуется изменением внутреннего порядка использования данных, однако шинные (внешние) операции ввода, вывода и записи в память выполняются в порядке, который определяет код программы. Стоит отметить, что несовершенство программного кода может ме-

шать полноценному использованию данной технологии, если он сгенерирован без учета возможности смены порядка, наиболее часто это встречается в 16-битных приложениях.

Для достижения наилучшего результата требуется комплекс из современной элементной базы, архитектура построения и эффективно работающее программное обеспечения. Выбор элементной базы влияет на быстродействие системной логики и рассеиваемую мощность, а непосредственно архитектурные варианты построения вычислительной системы определяет оптимальное сочетание производительности и надежности.

Проанализировав все способы, используемые для увеличения вычислительной производительности, мы выявили наиболее важные задачи для работы в этом направлении.

Одной из задач является разработка механизмов определения параллельных участков программ, с целью их независимого выполнения на разных процессорах. Для решения необходимо учитывать методы анализа структур вычислительных задач, методы построения независимых и условно-независимых участков вычислительных задач, методы выявления связности вычислительных операций, а также способы формирования множества параллельных структур. В сочетании таких методов необходимо использовать технологии распределено-параллельной обработки данных, что даст возможность планирования параллельного выполнения и организации слабосвязанных процессов в распределенной вычислительной среде.

Таким образом, чтобы добиться результатов в увеличении вычислительной производительной систем путем организации распределено-параллельной обработки, требуется разработка алгоритма анализа конфигурации вычислительных задач и формирования множества их параллельных структур.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Дорофеев Н. Архитектура IA 64 Электронный ресурс. // iXBT.com.

2. Ким А.К., Волконский В.Ю., Груздов Ф.А. и др. Микропроцессорные вычислительные комплексы с архитектурой «Эльбрус» и их программное обеспечение и др. // Вопросы радиоэлектроники. Сер. ЭВТ. 2009.

3. Michael Feldman. Summit Up and Running at Oak Ridge, Claims First Exascale Application // <https://www.top500.org>

4. Секрет высокой производительности Intel Core 2 Duo: Микроархитектура Core. // Overclockers.ru

5. Таненбаум Э. Архитектура компьютера. 4-е изд. - СПб.: Питер, 2003.

УДК 004.9

И.И. ФОЛОМКИН, А.С. ЩУКАРЕВ, А.В. ПШЕНИЧНИКОВ

Рязанский государственный радиотехнический университет

МОДЕЛИРОВАНИЕ И АНАЛИЗ РИСКОВ В УПРАВЛЕНИИ ПРОЕКТАМИ

Рассматриваются прикладные программы для предприятий, с помощью которых руководители компаний выполняют анализ рисков.

Управление рисками проекта является довольно новым, но актуальным процессом. Этот процесс позволяет снизить финансовые потери от влияния отрицательных факторов. В связи с тем, что исключить риски не представляется возможным, необходимо максимально уменьшить последствия этих факторов. Из этого следует, что одной из главных или самой главной задачей для руководства предприятия является анализ рисков, их оценка и минимизация отрицательных факторов.

Для поставленной задачи возникла потребность в специальном механизме, который будет учитывать риски при принятии решений. Этим механизмом является риск-менеджмент или управление риском – это деятельность руководства предприятия, направленная на минимизацию рисков, которые наносят материальный ущерб и на защиту от нежелательных обстоятельств или последствий. “Управление рисками — комплекс мероприятий, включающих идентификацию, анализ, снижение и мониторинг рисков, с целью снижения отклонения фактических показателей реализации проекта от их запланированных значений” [1].

Управление рисками можно разделить на две подсистемы: управляемую и управляющую. Управляемая система – это объект управления, в качестве которого могут быть представлены экономические отношения между субъектами и рисковые вложения капитала. Субъектом управления выступает руководство предприятия, которое осуществляет деятельность объекта управления путем воздействия на него. Процесс управления способен осуществляться только тогда, когда информация может передаваться из одной подсистемы в другую.

Существует два вида анализа риска: качественный и количественный. В ходе качественного анализа выявляются наиболее значимые риски и уже для них проводится количественный анализ. Количественный анализ дает оценку финансового эффекта от наступления рисков. Среди распространенных методов количественного анализа выделяют: анализ чувствительности, анализ ожидаемой денежной стоимости, моделирование и имитация методом Монте-Карло.

В наше время для наиболее точных и быстрых принятий решений, для анализа рисков, руководители предприятий используют специализированные программы. Наиболее известные из таких программ: @RISK, ConnectWise Identify, Dev Ops Prediction Engine, IQSM - Total Risk Management, Oracle Crystal Ball, ГРИФ 2.0, CRAMM Version 5 и РискМенеджер-Анализ.

@RISK – это программа, выполняющая анализ рисков с помощью метода Монте-Карло. Моделирование процессов в ней позволяет просмотреть большое количество результатов на базе электронной таблицы, а также показывает вероятность каждого результата моделирования. Также @RISK имеет возможность отслеживать и вычислять риски и их вероятности в зависимости от будущих сценариев. Еще важной функцией данной программы является планирование стратегии управления рисками с помощью модуля RISKOptimizer, объединяющий метод Монте-Карло с технологиями поиска решений. Это позволяет оптимизировать любые электронные таблицы с неопределенными значениями. Анализ в @RISK включает в себя три шага: создание модели, процесс моделирования и анализ рисков.

Работа программы. Для начала необходимо неопределенные значения заменить функциями распределения вероятности. Программа имеет шестьдесят пять различных функций распределений, которые предполагают диапазон значений, располагающихся в данной ячейке. Далее нужно выбрать тип диаграммы и нажать кнопку моделирования процесса. Программа внесет значения в модель и сохранит полученный результат. Итоговый результат включает все разнообразие последствий и вероятность их наступления. Полученные данные можно отобразить в диаграммах размаха, разброса, в гистограммах. Также программа позволяет выделить критические факторы с помощью анализа чувствительности и торнадо диаграмм. Эти графики совместимы с Microsoft Excel.

DevOps Prediction Engine – новейшее программное обеспечение для прогнозирования рисков. Данный модуль прогнозирования рисков совмещает машинное обучение с приобретаемым алгоритмом и позво-

ляет разработчикам предотвращать риски до выпуска продукции. Также данная программа способна выявлять слабые места в проектах.

Еще одной программой для анализа и оценки рисков является CRAMM Version 5.0. Данная программа обеспечивает комплексный метод оценки риска с возможностью выполнять три различных вида обзора: CRAMM Express Reviews, BS7799: 2005 Reviews и CRAMM Expert Reviews. CRAMM поддерживает множество дополнительных функций, включая изготовление охранной документации, исследование стандартов и запись информации, необходимой для планирования непрерывности бизнеса. CRAMM содержит: базу данных, охватывающую все аспекты информационной безопасности; процедуры безопасности и другую полезную документацию по безопасности; предварительно определенные оценки риска, охватывающие общие информационные системы и полный набор инструментов управления рисками помощи, для поддержки безопасности проектов.

В условиях современного общества внедрение прикладных программ требуется повсеместно, в том числе и на предприятиях. Для решения необходимых задач, а именно для анализа рисков, и предназначены рассмотренные продукты. Они позволяют облегчить и ускорить данные процессы, смоделировать различные ситуации и их последствия. Это одни из основных программ на рынке, но в действительности их гораздо больше.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. И.И. Мазур, В.Д. Шапиро, Н.Г. Ольдерогге. Управление проектами. 2010.
2. Уланов С. Анализ рисков при управлении инвестиционными проектами // РИСК : ресурсы, информация, снабжение, конкуренция. - 2011. - N 1. - Ч. 2. - С. 645-650.

УДК 004.383

И.В. ХОЛОПОВ, Е.В. ВАСИЛЬЕВ

Рязанский государственный радиотехнический университет

ФОРМИРОВАНИЕ ГАРМОНИЧЕСКИХ КОЛЕБАНИЙ С ПОМОЩЬЮ РАЗЛИЧНЫХ АЛГОРИТМОВ НА ПЛИС

В статье рассматривается возможность осуществления исследования спектральных характеристик высокочастотных гармонических колебаний, сформированных на ПЛИС с использованием различных алгоритмов.

В настоящее время существует возможность формирования высокочастотных (ВЧ) колебаний с использованием цифровых интегральных микросхем, таких как цифроаналоговые преобразователи, цифровые вычислительные синтезаторы, программируемые логические интегральные схемы (ПЛИС). Для формирования гармонического колебания на ПЛИС используются алгоритмы Large ROM, Small ROM, CORDIC, Multiplier-Based.

В том случае, когда цифровое формирование сигнала осуществляется с целью его использования в качестве гетеродинного колебания высокочувствительного радиоприемника или выходного колебания высококачественного генератора в составе измерительного комплекса, к спектральным характеристикам колебания могут предъявляться весьма жесткие требования. В этом случае в процессе проектирования радиоэлектронной аппаратуры, включающей формирователь ВЧ колебания на ПЛИС, необходимо осуществить выбор метода цифрового формирования такого колебания с целью обеспечения наилучшей спектральной чистоты колебания. Моделирование спектральных характеристик гармонических колебаний высокой частоты будем производить в программе Quartus II 9.0, с использованием мегафункции NCO 9.0.

Программный пакет Quartus II 9.0 представляет полную многоплатформенную среду проектирования систем на программируемом кристалле, а также включает в себя все утилиты, необходимые для работы с микросхемами FPGA и CPLD.

Для создания мегафункций используется утилита MegaWizard Plug-in Manager, которая содержит IP-ядро NCO 9.0 [1].

Мегафункция NCO 9.0 позволяет формировать отсчеты сигнала с использованием алгоритмов Large ROM, Small ROM, CORDIC, Multiplier-Based, осуществлять фазовую автоподстройку частоты, формировать сигналы с частотной или фазовой манипуляцией, а также предусмотрена возможность использования фазового дизеринга.

Основная проблема цифровых синусоидальных синтезаторов заключается в конечной точности представления двоичных чисел в цифровой технике, что проявляется в виде наличия паразитных составляющих в спектральном представлении выходного сигнала. Для уменьшения их влияния, после фазового аккумулятора вводится псевдослучайный процесс со специально подобранным спектром. Это повышает общий уровень шума в генераторе, но уменьшает паразитные составляющие, и общий динамический диапазон свободный от паразитных составляющих результирующего сигнала увеличивается.

Интерфейс мегафункции NCO 9.0 (см. рис. 1) позволяет отображать графики временной и частотной области генерируемого сигнала, что даёт возможность в реальном времени, изменяя параметры сигнала или гармонического колебания, оценивать спектральные и временные характеристики.

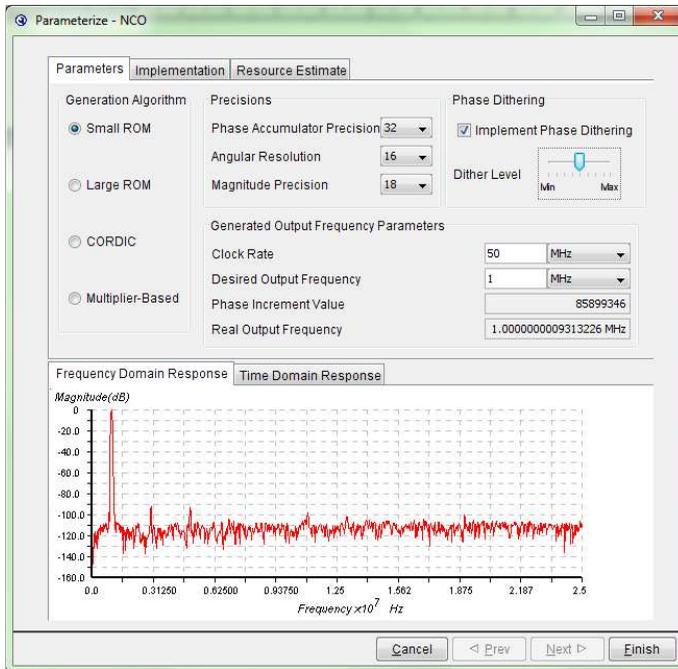


Рис. 1. Интерфейс панели управления формированием ВЧ колебания мегафункции NCO 9.0.

В ходе моделирования будем формировать гармоническое колебание с различной степенью фазового джиринга. Частота колебания 1 МГц, тактовая частота генератора 50 МГц, разрядность фазы 32 бита, разрядность амплитуды выходного сигнала 16 бит.

В качестве критериев качества будем использовать динамический диапазон свободный от паразитных составляющих результирующего сигнала (SFDR) и общий уровень шума (SNR).

После моделирования были построены зависимости исследуемых критериев качества, изображённых на рис. 2 и 3.

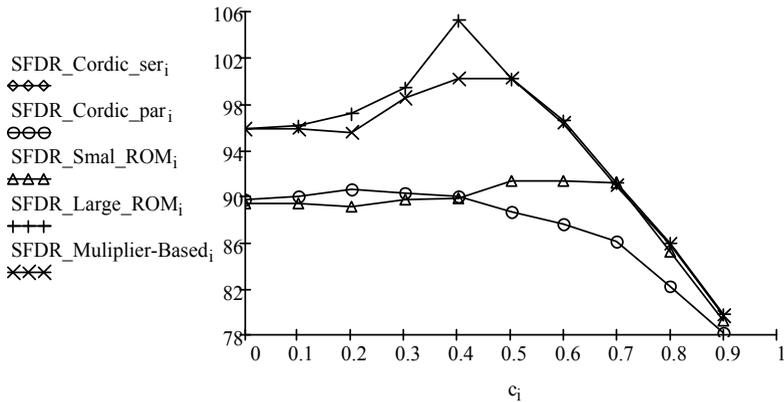


Рис. 2. Зависимости свободного от паразитных составляющих динамического диапазона от степени дизеринга.

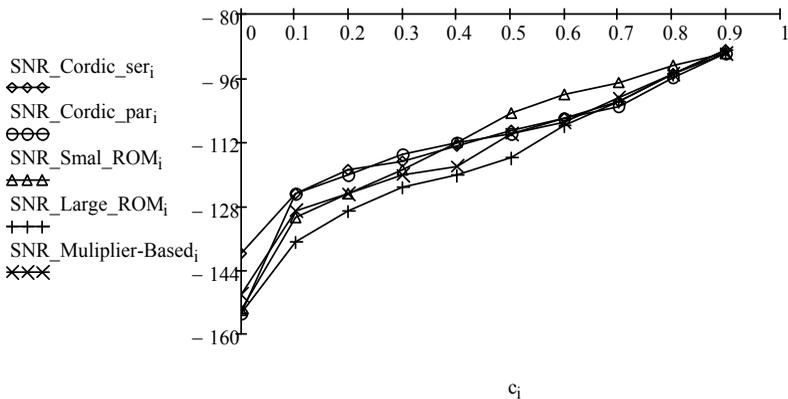


Рис. 3. Зависимости общего уровня шума от степени дизеринга.

Заметим, что самый большой динамический диапазон, свободный от паразитных составляющих, наблюдается у алгоритма Large ROM при степени дизеринга 0.4, и составляет 105 дБ. При этом общий уровень шума составляет -120 дБ. Наихудший результат по SFDR наблюдается у параллельной архитектуры CORDIC и соответствует зна-

чению 90,62 дБ при степени дизеринга 0,2. Общий уровень шума -120 дБ. Таким образом, используя описанный подход к выбору алгоритма и параметров формирования ВЧ колебания на ПЛИС, можно оценить его спектральные характеристики на этапе проектирования радиоэлектронной аппаратуры.

В заключение необходимо отметить, что в данной статье рассматривались спектральные характеристики исследуемых алгоритмов, но не оценивались используемые ресурсы ПЛИС, что в некоторых проектах может быть решающим для выбора алгоритма формирования гармонического колебания.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. NCO IP Core User Guide [Электронный ресурс]: Intel Corporation. 2017.11.06. — Режим доступа: https://www.intel.ru/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_nco.pdf

УДК 621.316.8

Л.Н. ХОМИНИЧ, М.А. СЕРГЕЕВ

Рязанский государственный радиотехнический университет

ТЕХНОЛОГИИ ИЗГОТОВЛЕНИЯ РЕЗИСТОРОВ, В Т.Ч. НАИБОЛЕЕ ПЕРСПЕКТИВНЫЕ

Рассматриваются технологии изготовления резисторов. Проводится анализ и выделяются наиболее перспективные из этих технологий в производстве.

Ни в какой другой отрасли электроники не существует такого многообразия технологий, на базе которых осуществляется изготовление резисторов и резисторных компонентов, как в области резисторостроения – это:

- технология изготовления проволочных резисторов;
- металлофольговая технология;
- тонкопленочная технология;
- технология изготовления резисторов на полимерной основе;
- технология изготовления нелинейных резисторов;
- толстопленочная технология на основе керметных материалов.

Технология изготовления проволочных резисторов. Проволочные резисторы отличаются *высокой стабильностью электрических параметров, повышенной точностью*, но резисторы этого типа имеют *значительные индуктивность и ёмкость, большие габариты и срав-*

нительно дороги. Основным элементом проволочных резисторов – тонкая проволока из сплавов, обладающих высоким удельным сопротивлением, достаточной механической прочностью, пластичностью и термостойкостью. Все элементы конструкций проволочных резисторов выполняются из термостойких материалов, а проводящий элемент защищают от климатических и механических воздействий стеклоэмалью и другими электроизоляционными покрытиями. Основной операцией при изготовлении проволочных резисторов является процесс наматывания проволоки на керамический или пластмассовый каркас. Полупроводниковые резисторы изготавливают из кремния, который обеспечивает высокую рабочую температуру изделия. Исходными заготовками служат кремниевые пластины различных размеров. После промывки и травления на концах пластин создают никелевые контактные площадки. Для этого химическим путем вжигают никель в слой кремния при температуре 780 – 800°С. Затем еще раз покрывают никелем контактные площадки и припаивают выводы.

Вывод: проволочные резисторы дольше других применяются в электротехнике и радиоэлектронике. Однако из-за недостатков, таких как: значительная индуктивность и собственная емкость, большие габариты и высокая стоимость, проволочные резисторы вытесняются другими видами.

Металлофольговая технология. Металлофольговый резистор представляет собой резистивный элемент – тонкую металлическую фольгу 2 – 10 мкм, приклеенную к диэлектрическому основанию. Методом фотолитографии формируется специальная топология резистивного элемента, обеспечивающая получение заданного значения номинального сопротивления. Резистивный элемент герметизируется в металлическом или опрессованном корпусе, для исключения влияния внешнего воздействия на параметры резисторов. Отличительные характеристики особенностью этих резисторов является *возможность сколь угодно точной подгонки номинального сопротивления, которая определяется уровнем метрологического обеспечения.*

У металлофольговых резисторов необходимо отметить ещё *самый низкий уровень токовых шумов по сравнению с другими резисторами, низкое значение термоЭДС по отношению к меди и тактичные постоянные значения коэффициенты напряжения.* Все эти достоинства позволяют на основе данной технологии разрабатывать уникальные резисторы, который невозможно реализовать, используя другие технологии.

Вывод: металлофольговые резисторы в настоящее время являются самыми точными, стабильными и надежными и находят при-

менение в областях с высокой степенью ответственности, повышенными требованиями к качеству, точности или сроку службы, а также в тяжелых климатических условиях — медицинское оборудование, высокопроизводительное аудио оборудование, прецизионные измерительные системы, аэрокосмические или военные приложения.

Тонкопленочная технология. Способ изготовления тонкопленочного резистора включает напыление на диэлектрическую подложку резистивных слоев из разнородных тонкопленочных структур, формирование примыкающих к одной из сторон диэлектрической подложки контактных площадок, формирование методом фотолитографии резистивных элементов из разнородных тонкопленочных материалов, определение величины сопротивления тонкопленочных резистивных элементов, подгонку до требуемой величины сопротивления и ТКС интегрального резистора на основании расчетных соотношений между сопротивлениями разнородных тонкопленочных структур, их ТКС и ТКС интегрального резистора.

К недостаткам упомянутого способа можно отнести недостаточные эксплуатационные характеристики резисторов, а именно надежность, стабильность при высоких удельных мощностях, большое время термообработки и рекомендательный характер температурных режимов, что снижает выход годных резисторов по параметрам ТКС.

Вывод: преимуществом тонкопленочной технологии являются высокая плотность размещения элементов и проводников на подложке, низкие паразитные параметры, низкий уровень электрических шумов и хорошие рабочие характеристики на высоких частотах. Все это позволяет использовать эти резисторы в системах электронной обработки данных, в промышленном электрооборудовании, в бытовой технике, в автомобильной электронике в системах управления двигателя.

Технология изготовления нелинейных резисторов. Способ включает в себя приготовление керамической шихты из смеси оксида цинка с добавками оксидов металлов, прессование заготовок, высокотемпературный обжиг и дополнительную термообработку, отличающийся тем, что с целью увеличения коэффициента нелинейности, срока службы резисторов и повышения их стабильности, перед дополнительной термообработкой определяют температуру фазового перехода стекло – кристалл в керамике, а дополнительную термообработку осуществляют при температуре 300–400°С на воздухе 1–2 ч.

Нелинейные резисторы находят широкое применение в разнообразных современных электронных и электротехнических схемах и

оборудовании. В ряде случаев такой прибор может заменить достаточно сложную электронную схему или существенно ее упростить.

Вывод: технология изготовления нелинейных резисторов дает приборы, которые завоевали широкое признание в связи с их преимуществами перед другими приборами, используемыми для аналогичных целей, такими как высокий уровень температурной чувствительности и уникально высокий уровень импульсной устойчивости в широком диапазоне напряжений, малые габариты, предельная простота устройства, стабильность характеристик во времени, высокая надежность, отсутствие необходимости специального ухода во время эксплуатации, экономичность.

Толсто пленочная технология на основе керметных материалов. В целом толсто пленочная технология состоит из ряда последовательных идентичных циклов. При формировании каждого слоя (резистивного, проводящего, диэлектрического и т.п.) используют соответствующие пасты, которые через сетчатый трафарет наносят на подложку, подвергают сушке и вжиганию. По завершении формирования всех слоёв резисторы и конденсаторы могут проходить подгонку (обычно лазерную) до заданной точности.

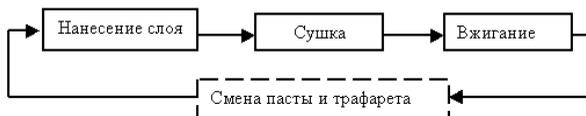


Рис.1. Структурная схема толсто пленочной технологии.

Вывод: в основе толсто пленочной технологии лежит использование дешёвых и высокопроизводительных процессов, требующих небольших одновременных затрат на подготовку производства, что позволяет этой технологии быть экономически целесообразной даже в условиях мелкосерийного производства.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Недорезов В.Г. Резисторы и резисторные компоненты// Электронные компоненты: М., №4, 2005, с. 29-31.
2. Недорезов В.Г. Подстроечные керметные резисторы. Часть 2 // Электронные компоненты: М., №6, 2005, с. 132-134.
3. Недорезов В.Г. Металлофольговые резисторы и наборы резисторов. Часть 4// Электронные компоненты: М., №6, 2005, с.111-114.
4. Технология производства резистора. [Электронный ресурс]. – Электрон. науч. журнал - «Stud.Wiki». 05.12.2010/ режим доступа к

журн.: http://stud.wiki/radio/2c0-a65625a2ac78b5c53b89421316c27_0.html

5. А.Калачев. Десять причин выбрать фольговые резисторы Vishay для вашего проекта. [Электронный ресурс]. - Электрон. журнал – Новости Электроники. №6 / статья 3/ 2011 / режим доступа к журн.: <https://www.compel.ru/lib/ne/2011/6/3-desyat-prichin-vyibrat-folgovyie-rezistoryi-vishay-dlya-vashego-proekta>

6. М.И.Николаевич, С.С.Валерьевич, (ОАО "НПО "ЭРКОН"). Способ изготовления тонкопленочного резистора. [Электронный ресурс]. - Электрон. журнал - ЕДРИД. 10.05.2016/ режим доступа к журн.: <https://edrid.ru/rid/216.015.3cc7.html>

7. Основы толстопленочной технологии. [Электронный ресурс]. - Электрон. журнал - Aksel Templates. 09.04.2008/ режим доступа к журн.: <http://www.nagrevatel.ru/ru/osnovyi-tolstoplenochnoy-tehnologii.html>

УДК 004.65

К.Ю. ЦВЕТКОВ

Рязанский государственный радиотехнический университет

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СРЕД РАЗРАБОТКИ СУБД

Проводится сравнительный анализ некоторых средств проектирования систем управления базами данных. Описываются преимущества использования тех или иных систем в зависимости от сложности проектируемых приложений для работы с базами данных.

Информационные системы представляют собой интеграцию всех статистических показателей, которые характеризуют основную деятельность предприятий, а также деятельность по производству и сбыту продукции. Для хранения и обработки такого массива данных разрабатываются специально заточенные под предметную область базы данных. При проектировании такой системы приходится решать множество вопросов: структура и механизмы работы систем хранения и обработки данных, взаимосвязь данных и их согласованность, и другие. Все эти вопросы имеют большую значимость на этапе проектирования баз данных, ведь от этого зависит дальнейшая корректность их работы.

Рассмотрим общие преимущества внедрения и использования систем управления базами данных при обработке данных. Во-первых, сокращаются объемы данных за счет уменьшения количества опера-

ций, выполняемых человеком. Во-вторых, решение вопроса с избыточностью данных, хранимых в таблицах системы, за счет особенностей работы алгоритмов обработки информации самой средой. В-третьих, информационные системы, которые разрабатываются с помощью современных СУБД, имеют привычный для пользователя интерфейс, что позволяет быстро адаптироваться к работе. В-четвертых, версии систем управления базами данных, разработанные в последние годы, обладают достаточным набором функций по обмену данными с различными технологиями, что делает их более гибкими с точки зрения интеграции. Таким образом, современные СУБД представляют собой единый автоматизированный комплекс, объединяющий различные технические средства обработки информации, в котором нашли применение как техники, так и методологии работы с данными [4].

Перейдем к сравнительному анализу различных современных сред разработки систем управления базами данных. В качестве исследуемых систем возьмем несколько современных разработок, которые получили наибольшую популярность у пользователей: Microsoft Access, Microsoft SQL Server, Visual FoxPro, Oracle.

СУБД Microsoft Access представляет собой систему для создания реляционных приложений, которая включает в себя весь перечень необходимых средств для идентификации, дальнейшей обработки данных и использования их при работе со значительными массивами данных. Благодаря использованию запросов появляется возможность выбора и редактирования хранимых в таблицах данных; также предусмотрены возможности для разработки и эксплуатации форм и отчетов, что позволяет выводить необходимую пользователю информацию на экран, редактировать хранимые данные и добавлять в систему новую информацию [2]. Одним из ключевых преимуществ Access является мастер-строитель, с помощью которого могут легко создаваться различные объекты: таблицы, формы, запросы, отчеты. Для большей автоматизации разрабатываемых систем в среде предусмотрена возможность создания макросов, которые могут самостоятельно выполнять заранее прописанные действия, а для реализации более мощных механизмов самостоятельного выполнения определенной последовательности операций в Access поддерживается язык Visual Basic, дающий еще большее разнообразие разрабатываемых функций.

Совокупность всех средств разработки СУБД позволяет созданным в Microsoft Access приложениям легко взаимодействовать с различными широко известными форматами файлов. Немаловажным достоинством программ Microsoft Access выше седьмой версии является

приспособленность к созданию программных продуктов для работы на рабочих станциях в серверных сетях.

Подведем итоги. СУБД Access – достаточно простая и удобная в работе среда создания информационных систем. Из недостатков стоит отметить невозможность формирования файла исполнения, который включал бы программный код, невысокую скорость работы с большими объемами данных, а также отсутствие отдельного режима для администрирования.

Microsoft SQL Server. Как и Microsoft Access, является системой для работы с реляционными базами, которую разработала корпорация Microsoft. В качестве основного языка для составления запросов используется Transact-SQL [1]. Стоит отметить основные преимущества этой системы: хорошая производительность, мгновенное получение результата обработки данных любого размера, возможность работы в гибридной среде, благодаря которой данные могут частично храниться в облачном хранилище и в случае необходимости могут быть легко скопированы обратно на компьютер. Как отмечалось выше, данная система относится к серверному типу, которая помимо всех преимуществ таких разработок имеет и полноценный набор возможностей для администрирования баз данных.

СУБД Visual FoxPro. Данная разработка отличается наличием собственного языка для программирования с поддержкой возможности генерации исполняемого exe-файла. Что касается недостатков, они такие же, что и у системы Access. Данную систему стоит рекомендовать к использованию в случае необходимости разработки приложений значительной сложности, в том числе, с фиксированным объемом записей, с которыми будет вестись работа.

СУБД Oracle. Представляет собой современную универсальную, полнофункциональную систему для управления реляционными базами данных, которая способна работать в операционных системах различного типа и на компьютерах различной конфигурации [3]. Эта система включает в себя принципы и средства разработки, проектирования и работы с распределенными базами данных, различные языки составления запросов, функции анализа данных, а также мастера и конструкторы форм, отчетов, запросов и многого другого. Отличительными особенностями системы являются значительная сложность и большая стоимость, в связи с чем ее рекомендуют использовать для работы с большими объемами данных больших предприятий.

Подводя итог, хотелось бы отметить, что в результате сравнения систем управления базами данных были описаны их характеристики и даны рекомендации по их использованию в зависимости от требова-

ний к конечной системе. В наше время СУБД занимают ключевое место при автоматизации множества процессов в работе различных организаций, что объясняется многообразием их функциональных возможностей. Среди рассматриваемых систем хотелось бы отметить Microsoft Access, как наиболее простую в работе, но в то же время имеющую достаточный функционал для решения большинства повседневных задач.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Аббакумов А.А., Егунова А.И., Таланов В. М. Базы данных (MS SQL Server). Саранск: Изд-во СВМО, 2015. 66 с.
2. Трофимова И.П., Сосулин Ю.А. Создание баз данных Access. Рязань, РГРТА, 2003.
3. Поляков А.М. Безопасность Oracle глазами аудитора: нападение и защита. - М.: ДМК Пресс, 2010. - 336 с.
4. Коновалов М.В. Обзор и сравнительный анализ промышленных хранилищ данных и баз данных // Молодой ученый. - 2018. - №24. - С. 24-28. Режим доступа: <https://moluch.ru/archive/210/51452/> (дата обращения: 16.03.2019).

УДК 621.433

А.В. ЧУДИНИН, В.В. СУСКИН

Рязанский государственный радиотехнический университет

СТРУКТУРА И ПРИНЦИП РАБОТЫ ДВИГАТЕЛЯ ВНУТРЕННЕГО СГОРАНИЯ НА ВОДОРОДЕ

Проводятся рассмотрение и анализ ДВС, которые могли бы быть использованы для работы на водородном топливе.

Данная работа является предварительным этапом дальнейшего исследования применения водородного топлива в двигателях внутреннего сгорания. Предложенная концепция подразумевает получение водорода на борту транспортного средства и последующее его использование в качестве добавки к основному виду топлива или непосредственно в качестве основного вида топлива.

Итак, двигатель внутреннего сгорания (ДВС) – двигатель, в котором топливо сгорает непосредственно в рабочей камере двигателя. ДВС преобразует тепловую энергию от сгорания топлива в механическую работу.

Виды ДВС:

- поршневые двигатели – камерой сгорания служит цилиндр, возвратно-поступательное движение поршня с помощью КШМ (кривошипно-шатунного механизма) преобразуется во вращение вала;
- газотурбинные двигатели (ГТД) – преобразование энергии осуществляется ротором с клиновидными лопатками;
- роторно-поршневые двигатели (РПД) – преобразование энергии осуществляется за счет вращения рабочими газами ротора специального профиля.

Также существует непосредственно классификация ДВС. Но в данном контексте особую роль играют существующие виды силовых агрегатов. Ведь целью этого раздела работы является выявление типа двигателя, который по своим характеристикам больше всего подойдет для работы на водородном топливе.

Характеристики каждого вида двигателя внутреннего сгорания:**1. Поршневой двигатель.**

Поршневой двигатель – ДВС, в котором тепловая энергия расширяющихся газов, образовавшаяся в результате сгорания топлива в замкнутом объеме, преобразуется в механическую работу поступательного движения поршня за счет расширения рабочего тела в цилиндре, в который вставлен поршень. Поступательное движение поршня преобразуется во вращение коленчатого вала КШМ.

Мощность таких двигателей – от нескольких ватт до 75 МВт. В качестве топлива используются жидкости – бензин, дизельное топливо, спирты и т.д., газы – сжиженный газ, природный газ, водород и т.д., монооксид углерода – вырабатывается в газогенераторе из твердого топлива (угля, торфа, древесины).

Эффективный коэффициент полезного действия (КПД) поршневого ДВС не превышает 60% (обычно 40-42%). Остальная тепловая энергия распределяется между теплом выхлопных газов и нагревом конструкции двигателя.

Основными параметрами ДВС являются: верхняя (ВМТ) и нижняя (НМТ) мертвые точки, радиус кривошипа, ход поршня, такт, объем камеры сгорания, рабочий объем цилиндра, полный объем цилиндра, рабочий объем двигателя для многоцилиндровых двигателей. В данном контексте эти параметры лишь характеризуют конструкцию поршневого ДВС.

Главными же параметрами являются: упомянутый ранее КПД, степень сжатия, надежность, износостойкость. Ведь известно, что ис-

пользование водорода вместо бензина снижает КПД ДВС. Также известно, что использование водородного топлива существенно сокращает рабочий ресурс двигателя. Предположительно, все дело в удельной теплоте сгорания топлива. При сгорании бензина выделяется 42-44 МДж/кг, а при сгорании того же количества водорода – 141 МДж/кг. Вероятно, трехкратная разница этих показателей играет определяющую роль в снижении рабочего ресурса двигателя. Исходя из этого можно сделать вывод о том, что ДВС сможет нормально функционировать на количестве водорода, которое почти в 3 раза меньше, чем количество бензина, необходимое для работы двигателя.

Таким образом, главные показатели, по которым будет определяться пригодность того или иного типа двигателя к работе на водородном топливе, - это КПД двигателя (который зависит от степени сжатия), его надежность и износостойкость. В ходе дальнейшего анализа типов двигателей их показатели будут сравниваться именно с показателями поршневого ДВС.

2. Газотурбинный двигатель.

Газотурбинный двигатель – лопаточная машина, в ступенях которой энергия сжатого и/или нагретого газа преобразуется в механическую работу на валу. Основными элементами конструкции такого двигателя являются ротор и статор. Такие двигатели применяют в промышленности для производства электричества, в авиационной и космической технике и т.д.

Преимущества данного типа двигателей состоят в следующем:

- высокое отношение мощности к весу;
- перемещение в одном направлении с меньшей вибрацией;
- меньшее количество движущихся частей;
- существенно меньший выброс вредных веществ в атмосферу;
- низкие стоимость и потребление смазочного масла;
- низкие требования к качеству топлива – ГТД потребляет любое горючее, которое можно распылить;
- высокая маневренность и диапазон регулирования.

Данный тип двигателя имеет и недостатки:

- стоимость гораздо выше, чем у аналогичных по размерам поршневых ДВС, из-за применяемых материалов;
- при любом режиме работы имеют меньший КПД, чем поршневые двигатели (на номинальной нагрузке – 39);
- резкое снижение КПД на малых нагрузках;
- необходимость использования газа высокого давления, что усложняет конструкцию и требует дополнительного расхода энергии, что сказывается на общей эффективности системы;

- высокие эксплуатационные нагрузки, что объясняет применение высокопрочных материалов;
- более медленный пуск;
- существенное влияние пусков-остановок на ресурс.

Все характеристики даны в сравнении с поршневым ДВС.

Исходя из этого, можно сделать вывод, что недостатки ГТД перекрывают его преимущества. Из основных преимуществ можно отметить простоту конструкции и, как следствие, надежность. Также используемые материалы имеют более высокую прочность, чем материалы поршневого ДВС. Но главные недостатки – высокое потребление топлива и сравнительно низкий КПД – позволяют отдать предпочтение поршневому двигателю.

Тем не менее, человечество уже пыталось не только создать автомобиль с ГТД, но и наладить его серийный выпуск. В этом отличилась компания Chrysler, создав модель A831. Расход топлива был слишком велик (производитель скрывал точные цифры и запрещал их разглашать), а выхлопные газы имели очень высокую температуру, поэтому возникала необходимость внедрения специальных кулеров. Впоследствии, ГТД был использован в танке M1 «Абрамс».

Также следует отметить, что данный тип двигателя был использован в концепте Jaguar C-X75, который был разработан в 2010 году. Автомобиль имел две «микротурбины», работающие на дизельном топливе. Они вращали генератор, питавший четыре электродвигателя. Данное решение открывает новые перспективы использования ГТД, в том числе, для использования водородного топлива и его выработки на борту транспортного средства.

3. Роторно-поршневой двигатель.

Роторно-поршневой двигатель – роторный ДВС, конструкция которого предусматривает применение трёхгранного ротора (поршня), имеющего вид треугольника Рёло, вращающегося внутри цилиндра специального профиля. Конструкция двигателя включает установленный на валу ротор, который жестко соединен с зубчатым колесом. Оно входит в зацепление с неподвижной шестерней – статором. Ротор обкатывается вокруг статора. Герметизация камер сгорания обеспечивается радиальными и торцевыми уплотнительными пластинами. В двигателе отсутствует механизм газораспределения, что делает его значительно проще четырехтактного поршневого, а отсутствие сопряжения между отдельными рабочими камерами обеспечивают компактность и высокую удельную мощность.

К преимуществам данного типа двигателей относятся:

- низкий уровень вибраций;

- высокие динамические характеристики;
- высокая удельная мощность;
- меньшие габаритные размеры;
- меньшее число деталей.

Также РПД имеет и недостатки:

- повышенный износ двигателя – особенность конструкции;
- повышенный износ уплотнителей;
- склонность к перегреву;
- высокий расход топлива на низких оборотах;
- высокая сложность изготовления.

Все характеристики даны в сравнении с поршневым ДВС.

Следует отметить, что данный тип двигателя может работать как на бензине, так и на водороде. Дело в том, что РПД менее чувствителен к детонации. Это является явным преимуществом данного типа. Но ряд недостатков – высокий расход топлива, сложность конструкции, склонность к перегреву, повышенный износ – в конкретном случае позволяет сделать выбор в пользу поршневого ДВС.

Таким образом, из всех перечисленных типов двигателей поршневой ДВС по своему КПД превосходит остальные. Тем не менее, доработка ГТД и РПД может выявить еще большие перспективы для использования в них водорода в качестве топлива, а также открыть новые горизонты применения.

Говоря о ГТД, следует еще раз отметить, что по надежностным характеристикам он превосходит поршневой ДВС. Но решающим показателем все же является КПД двигателя. В данном компоненте, поршневой двигатель имеет преимущество. А важным преимуществом РПД являются простота конструкции и стойкость к детонации топлива. Но решающим снова становится сравнительно низкий КПД.

Также следует отметить, что доработка требуется и поршневому ДВС. Известно, что поршень, совершая возвратно-поступательное движение в цилиндре, сталкивается с сопротивлением боковой стенки цилиндра. Максимальный эффект достигается, когда поршень находится строго между ВМТ и НМТ, то есть, когда коленчатый вал достигает 90 и 270 градусов вращения.

Эта особенность существенно сказывается на КПД двигателя. Чтобы минимизировать это сопротивление, необходимо внести дополнительный элемент в конструкцию КШМ.

Этот элемент будет соединять поршень и шатун. Данная модификация позволит увеличить КПД двигателя примерно на 10%. Подробнее об этом будет описано в дальнейших работах по данной теме.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ховах М.С., Маслов Г.С. Автомобильные двигатели. 2-е изд. «Машиностроение». Москва. 1971.
2. Гагарин Е.И. Развитие конструкций автомобильных двигателей. Изд-во АН СССР. Москва. 1962.
3. Райков И.Я. Испытания двигателей внутреннего сгорания. «Высш. школа». Москва. 1975.

УДК 004.383

Т.С. ШЕСТАКОВА

Рязанский государственный радиотехнический университет

**ИССЛЕДОВАНИЕ И РАЗРАБОТКА КОМПЛЕКСА ДЛЯ
ОБРАБОТКИ ЭЛЕКТРОФИЗИОЛОГИЧЕСКОЙ
ИНФОРМАЦИИ**

В статье рассматривается принцип работы комплекса для обработки электрофизиологической информации

Широкое внедрение цифровой вычислительной техники во все сферы производства и управления привело к возникновению ряда новых научных направлений. Одним из таких направлений является цифровая обработка информации в медицине. Головной мозг человека является сложной и очень уязвимой биологической системой. Лечение нейрозаболеваний одно из самых сложных и длительных. В этом процессе большую помощь врачам оказывают различные контрольные и диагностические приборы. Созданию медицинских приборов нового поколения уделяется большое внимание и в национальных проектах. С этой точки зрения тема моей работы является очень актуальной.

Математическая обработка электрофизиологической информации на ЭВМ широко применяется как при диагностике, так и лечения больных в условиях реанимационного отделения. Уже много лет ведущие фирмы по производству электронной аппаратуры для электрофизиологического исследования выпускают компьютерные системы для получения вызванных потенциалов, спектрального анализа и т. д. Стандартизация методов обработки физиологической информации и выпуск специализированных вычислительных устройств сопровождаются поиском новых, а также совершенствованием известных методик. Всё больше применяется нейромониторинг, интра- и послеоперационный контроль за состоянием тяжёлых больных с помощью ЭВМ.

В состав разрабатываемого вычислительного комплекса входят:

1) система обработки и визуализации информации (микроконтроллер на базе МикроЭВМ, ППЭВМ, печатающее устройство);

2) устройства связи с объектом (магистральный расширитель, одноканальный цифро-аналоговый преобразователь, устройство сопряжения с цифровыми приборами, фотостимулятор, фоностимулятор, Электронным стимулятором, Устройство подачи звуковых щелчков, фотостимулятор для мониторингового наблюдения).

Структурная схема данного комплекса представлена ниже (см. рис.1):

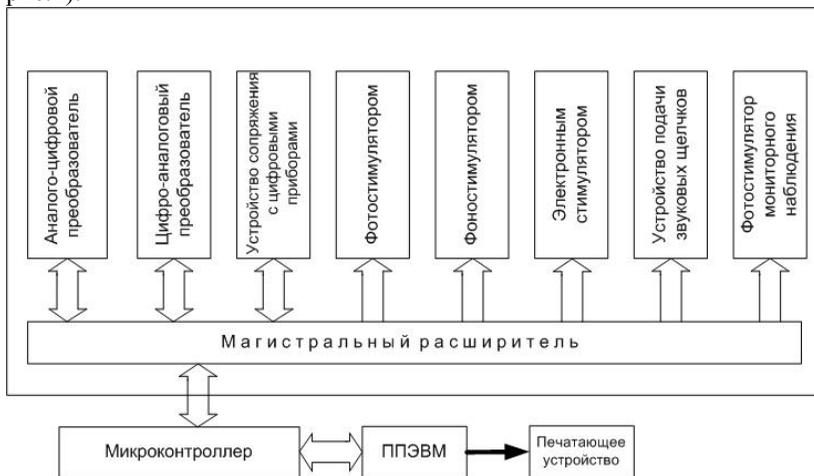


Рис.1. Вычислительный комплекс для обработки электрофизиологической информации.

Рассмотрим подробнее каждое из звеньев, входящих в комплекс.

Микроконтроллер необходим для предварительной обработки и фильтрации входного потока информации, а также согласования с ППЭВМ.

На ППЭВМ производится основная обработка информации. С целью экономии оборудования и снижении стоимости устройства, для отображения оперативной информации о состоянии больного используется стандартный дисплей из состава ППЭВМ.

АЦП (с максимальной частотой опроса 28 кГц) собрана на БИС КП13ПВ1А. Данные преобразования пересылаются в однобайтовом

формате. Суженный диапазон допускаемых сигналов на входе АЦП соответствует диапазону амплитуды выходного сигнала большинства типов усилителей электроэнцефалографов, а время на пересылку результата преобразования сокращается вдвое и во столько же раз экономится объём памяти. Максимальная частота АЦП по одному каналу – 28 кГц. Номер опрашиваемого канала задаётся при обращении к АЦП. Частота опроса регулируется программно. Быстродействие АЦП позволяет регистрировать медленные и быстрые электрофизиологические процессы – стволовые вызванные потенциалы и т. п.

Устройства управления стимуляторами представляют собой дещефратор адреса и преобразователи сигнала ТТЛ-уровня в сигнал требуемой формы и уровня, необходимый для запуска стимулятора.

Фотостимулятор для мониторингового контроля предназначен для подачи с помощью светоизлучательной системы на светодиодах – в виде надеваемых на глаза очков, регулируемых по яркости и длительности вспышек. Устройство подачи звуковых щелчков подобно фотостимулятору для мониторингового контроля.

Структуру вычислительного комплекса можно наращивать и усложнять – подключать (в составе УСО есть плата сопряжения) новые устройства, а также организовывать мультипроцессорную вычислительную систему.

Программное обеспечение комплекса включает в себя программы: анализа биоэлектрической активности мозга, анализа вегетативных процессов, обработки результатов спектрального, корреляционного и статического анализа данных.

Все программы оптимизированы по быстродействию и объёму памяти, записывают исходную и обработанную информацию на жёсткие магнитные диски, т. е. ведётся архив.

Результаты обработки выводятся на дисплей ППЭВМ, печатающее устройство.

СПИСОК ИСПОЗУЕМЫХ ИСТОЧНИКОВ

1. Проектирование конструкций радиоэлектронной аппаратуры: учебное пособие для вузов / Е. М. Парфенов, Э.Н. Камышная, В. П. Усачев. - М.: Радио и связь, 1989. - 272с: ил.
2. Электрофизиологические методы исследования : метод. указания по физиологическим методам исследования / И. Ю. Мышкин; Яросл. гос. унт им. П. Г. Демидова. – Ярославль : ЯрГУ, 2011. – 68 с.
3. Справочник конструктора РЭА: Общие принципы конструирования/Под. Ред. Р. Г. Варламова. - М: Сов. Радио, 1980. - 480с.

УДК 004.4

И.А. ШИРЕНИН

Рязанский государственный радиотехнический университет

СОБЫТИЙНАЯ МОДЕЛЬ ПОСТРОЕНИЯ ГРАФИЧЕСКИХ ПРИЛОЖЕНИЙ

Рассматривается подход к проектированию приложений, основанный на публикации и обработке событий, а также преимуществ применения такого подхода. Рассмотрение происходит на примере создания приложения для расчета периметра треугольника по трем сторонам.

Один из самых распространенных подходов к написанию графических приложений основан на генерации и обработке событий. Суть такого подхода очень проста – одни компоненты являются источниками событий, а другие - слушателями.

Алгоритм работы модели:

- Зарегистрировать слушателя событий в источнике событий.
- Сгенерировать событие источником.
- Найти подходящий обработчик на сгенерированное событие.
- Вызвать обработчик.

Преимущества такого подхода:

- Весь код, предназначенный для вычислений, находится в отдельных компонентах.
- Приложение легко расширяется путем добавления новых источников и обработчиков событий.
- Разделена логика процесса выполнения приложения и управления этим процессом [1].
- Возможность создания цепочек событий, когда слушатель событий А является источником событий В.

Напишем пример простого приложения, которое будет вычислять периметр треугольника по трем сторонам.

Сперва необходимо создать интерфейс, реализовав который можно будет стать слушателем событий.

```
public interface CalculatingResultListener {  
    void onError(ResultEvent event);  
    void onSuccess(ResultEvent event);  
}
```

Здесь *ResultEvent* – класс, который по своей сути является

оберткой над описанием успеха/неуспеха операции и результатом вычисления.

Далее определим, кто будет являться слушателем событий, а кто источником. Класс формы должен отображать информацию о вычислении пользователю, то есть реагировать на результат вычисления. Класс, содержащий логику вычисления, должен будет оповестить форму о том, что введенные данные корректны/некорректны и выдать результат вычисления в случае корректности данных.

В нашем случае слушателем будет являться класс формы, а источником – класс с логикой вычислений.

Теперь напомним класс, который будет являться источником событий и выполнять расчет периметра.

```
public class Core {
    private List<CalculatingResultListener> listeners;

    public Core() {
        this.listeners = new ArrayList<>();
    }

    public void calculate(double a, double b, double c) {
        /* расчет
           ВЫЗОВ publishSuccessEvent/publishErrorEvent
        */
    }

    public void addListener(CalculatingResultListener
listener) {
        this.listeners.add(listener);
    }

    private void publishSuccessEvent(final ResultEvent
event) {
        listeners.forEach(e -> e.onSuccess(event));
    }

    private void publishErrorEvent(final ResultEvent
event) {
        listeners.forEach(e -> e.onError(event));
    }
}
```

Теперь мы можем зарегистрировать слушателя, передав его в метод `addListener`. После вызова `calculate`, все слушатели бу-

дуг оповещены о результатах вычисления.

Остается только создать класс формы, реализовав интерфейс *CalculatingResultListener*, и зарегистрировать его в качестве слушателя в компоненте с вычислениями. Самый простой способ сделать это – передать в метод *addListener* ссылку на самого себя (*this*).

```
public class Form implements CalculatingResultListener {
    public Form() {
        core = new Core();
        core.addListener(this);
        perimeterButton.addMouseListener(new
MouseListener() {
            @Override
            public void mouseClicked(MouseEvent e) {
                core.calculate(a, b, c);
            }
        });
    }

    public void onError(ResultEvent event) {
        this.resultField.setText(event.getDescription());
    }

    public void onSuccess(ResultEvent event) {
        this.resultField.setText(event.getDescription()
            + event.getResult());
    }
}
```

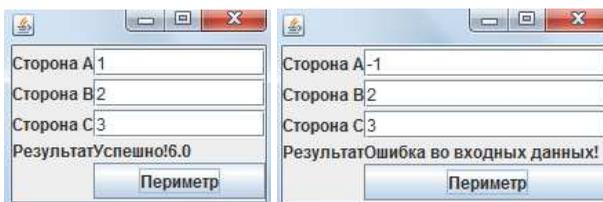


Рис. 1. Результат выполнения.

Заключение

В данной статье был описан такой подход к программированию, как применение событийной модели. Такой подход обладает большой гибкостью, легкостью расширения функционала и простотой реализации. Он позволяет нужным слушателям быстро реагировать на изменившееся состояние системы.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Событийная модель построения приложения [Электронный ресурс], URL: <https://foxford.ru/wiki/informatika/sobytiynaya-model-postroeniya-prilozheniya> (дата обращения 16.03.19).

УДК 681.518.3

А.Э. ЯКУШКИН, В.А. НИКОЛАШИН

Рязанский государственный радиотехнический университет

МОДЕРНИЗАЦИЯ КОМПЛЕКСА ПРОГРАММНО-ТЕХНИЧЕСКИХ СРЕДСТВ УСТАНОВКИ ПРОИЗВОДСТВА БИТУМА

Рассматриваются вопросы по модернизации комплекса программно-технических средств установки производства битума.

Битумная установка предназначена для получения нефтяных битумов.

Битумы нефтяные – остаточные продукты переработки нефти, имеющие твердую или вязкую консистенцию и состоящие из углеводородов и гетерогенных (кислородных, сернистых, азотистых, металл-содержащих) соединений.

Основные свойства нефтяных битумов, определяющие их пригодность к применению в народном хозяйстве: малое изменение пластичности при изменении температуры; высокие вязкость и цементирующая способность; стабильность и долговечность; теплостойкость (высокая температура размягчения), обеспечивающая сохранение необходимой прочности сооружений и изделий летом; упругость, благодаря которой сохраняется достаточная пластичность и эластичность при низких температурах.

На битумной установке получают окисленные нефтяные битумы: строительные марок БН-70/30, БН-90/10, кровельный марки БНК-40/180 и дорожные вязкие марок БНД-90/130, БНД-60/90.

Основными параметрами, влияющими на процесс окисления, являются температура окисления, количество воздуха, подаваемое на окисление, давление и уровень в колонне.[1]

Для подачи воздуха используются блок компрессоров в составе воздушных компрессоров ВК-1 и ВК-2. Проект перевода блока компрессоров на электронные РСУ и ПАЗ (см. рис. 1) включает в себя ра-

боты по программно-техническому перевооружению системы управления.

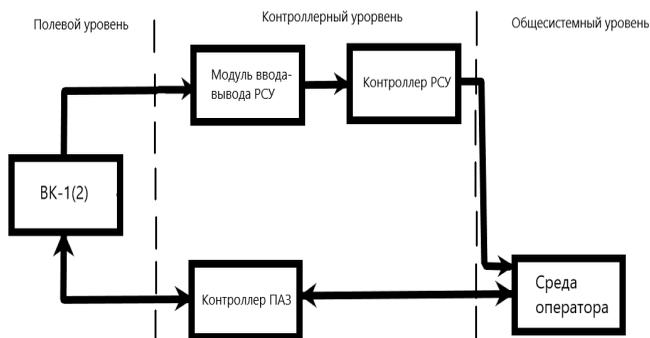


Рис. 1. Структурная схема управления блоком компрессоров.

Функционально Система делится на две подсистемы:

PCY (DCS, Distributed Control System - распределенная система управления) подсистема контроля и управления технологическими процессами.

К основным функциям PCY относятся:

1) Автоматический контроль состояния технологического процесса, сигнализация при выходе технологических показателей за установленные границы.

2) Автоматическая обработка, регистрация и хранение поступающей в Систему информации.

ПАЗ (ESD, Emergency Shutdown System - система аварийных отключений) подсистема противоаварийной автоматической защиты (ПАЗ) интегрированная с PCY на контроллерном и программном уровне.

К основным функциям системы ПАЗ относятся:

1) Автоматический сбор аналоговой и дискретной информации от датчиков технологических параметров и дискретных параметров состояния исполнительных механизмов.

2) Контроль входной информации на достоверность; логическую обработку и анализ входной информации в зависимости от стадии циклического процесса.

3) Автоматическую выдачу сигналов двухпозиционного управления на исполнительные механизмы, согласно заранее сконфигурированным алгоритмам управления ПАЗ.

Кроме функционального деления на подсистемы РСУ и ПАЗ, Система аппаратно делится на три уровня:

1) Полевой уровень – уровень возникновения информации. На этом уровне размещаются КИП и исполнительные устройства. Далее в данном документе не рассматривается.

2) Контроллерный уровень - уровень автоматического контроля и регулирования, включающий также систему защиты и блокировки. Строится на базе микропроцессорных контроллеров.

3) Общесистемный уровень - уровень управляющей сети и операторского интерфейса. Включает в себя автоматизированные рабочие станции (АРМ), сетевое оборудование, принтер, прочее вспомогательное оборудование. На этом уровне обеспечивается передача и представление информации.

Реализация данного проекта позволит:

- 1) повысить безопасность ведения технологического процесса и общую надежность элементов системы управления;
- 2) упростить контроль технологических параметров операторским персоналом;
- 3) сократить время реагирования на аварийную и предаварийную ситуацию.

Кроме того внедрение ПАЗ на контроллерном уровне повысит скорость выдачи воздействий на объект защиты, что позволит избежать критических последствий.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Утвержденный 31.01.2011г. постоянный технологический регламент битумной установки 19/5 , индекс ТР2.041.033.11;
2. Технические требования на РСУ и ПАЗ установки производства битумов 19/5.
3. ГОСТ 21.408-2013 « Межгосударственный стандарт. Система проектной документации для строительства. Правила выполнения рабочей документации автоматизации технологических процессов.
4. Нестеров А.Л. Проектирование АСУТП: методическое пособие / А.Л. Нестеров – Книга 1, изд. ДЕАН, 2010 – 551с.

Информационные технологии

Межвузовский сборник научных трудов

Компьютерная верстка: А.Н. Сапрыкин, Н.С. Лебедева
И.А. Музюкин, Ф.В. Машков

Подписано в печать 18.09.19.
Формат бумаги 60x84/16. Бумага офсетная.
Печать струйная. Усл. печ. л. 16,5.
Тираж 100 экз. Заказ № 2512.

Издательство ИП Коняхин А.В. (Book Jet)

Отпечатано в типографии Book Jet
390005, г. Рязань, ул. Пушкина, д.18
Сайт: <http://bookjet.ru>

ISBN 978-5-6043323-3-7



9 785604 332337