

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

СЕМИДЕСЯТАЯ
МЕЖДУНАРОДНАЯ СТУДЕНЧЕСКАЯ НАУЧНАЯ
КОНФЕРЕНЦИЯ ГУАП

Часть 2

ТЕХНИЧЕСКИЕ НАУКИ

Сборник докладов
17–21 апреля 2017 г.



УДК 001
ББК 72
С30

С30 Семидесятая международная студенческая научная конференция ГУАП: сб. докл.:
В 4 ч. Ч. 2. Технические науки / СПб.: ГУАП., 2017. 352 с.: ил.
ISBN 978-5-8088-1208-6 (Ч. 2)
ISBN 978-5-8088-1206-2

Представлены доклады Семидесятой международной студенческой научной конференции
ГУАП, прошедшей в апреле 2017 г.

Научные работы студентов ГУАП посвящены актуальным проблемам авиационного и аэро-
космического приборостроения, проблемам развития новых информационных технологий, радио-
техники, электроники и связи, современным проблемам экономики, философии и права.

Под общей редакцией
доктора юридических наук, профессора
В. М. Боера

Редакционная коллегия:

А. Р. Бестугин, доктор технических наук, доцент,
А. С. Будагов, доктор экономических наук, доцент,
А. Д. Жуков, канд. технических наук, доцент,
К. В. Лосев, доктор экономических наук, доцент,
И. А. Павлов, кандидат военных наук, доцент,
Е. Г. Семенова, доктор технических наук, профессор,
М. Б. Сергеев, доктор технических наук, профессор,
В. А. Фетисов, доктор технических наук, профессор,
Н. А. Чернова, кандидат экономических наук,
В. Ф. Шишлаков, доктор технических наук, профессор

Ответственный секретарь редакционной коллегии *В. Н. Соколова*

ISBN 978-5-8088-1208-6 (Ч. 2)
ISBN 978-5-8088-1206-2

© Санкт-Петербургский государственный
университет аэрокосмического
приборостроения, 2017

УДК 519.673

Б. К. Акопян, А. В. Городецкая

студенты кафедры проблемно-ориентированных вычислительных комплексов

О. О. Жаринов

канд. техн. наук, доцент – научный руководитель

ИССЛЕДОВАНИЕ МЕТОДОВ ОБРАБОТКИ СИГНАЛА В ПРИЕМНИКЕ СИСТЕМЫ ЦИФРОВОЙ СВЯЗИ С ПЕРЕДАЧЕЙ ПО ОПТИЧЕСКОМУ КАНАЛУ

Тесты разработанного макетного прототипа системы цифровой связи с передачей по оптическому каналу [1] показали, что основным недостатком системы является чувствительность к общему уровню внешней освещенности, что сопровождается искажением формы импульсов; это приводит к ошибкам приема сигнала. Вследствие этого для решения задачи обеспечения максимально возможной достоверности принимаемой информации было решено обратиться к методам обработки сигналов на стороне приемного устройства. Задачей является настройка решающего правила таким образом, чтобы оно существенно уменьшало значение средней вероятности ошибки на бит.

Однопороговая схема сравнения

Однопороговая схема сравнения является простейшим способом обработки. Принцип работы этой схемы заключается в том, что среднее значение напряжения i -го бита U_i сравнивается с некоторым пороговым значением L :

$$U_i \geq L \rightarrow "1", U_i < L \rightarrow "0". \quad (1)$$

U_i определяется как среднее арифметическое значений отсчетов напряжения:

$$U_i = \frac{1}{n} \sum_{j=1}^n y_j, \quad (2)$$

где n – число отсчетов на бит, y_j – значение напряжения j -го отсчета бита.

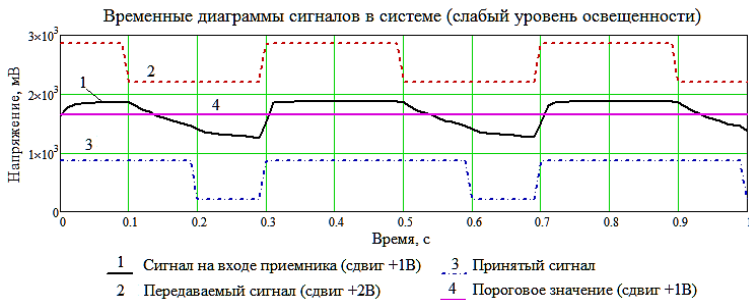


Рис. 1. Результат обработки по однопороговой схеме в режиме слабой внешней освещенности

Перед отправкой сообщения с фоторезистора считываются значения настройки напряжения, соответствующие включенному ($U_{вкл}$) и выключенному ($U_{выкл}$) состоянию светодиода. Разность между

значениями настройки целесообразно принять в качестве параметра, характеризующего уровень освещенности: чем сильнее освещенность, тем меньше значение параметра. Особое внимание уделяется определению порогового значения решающего правила, которое может быть выбрано различными способами с целью компенсации внешней засветки. В макетном прототипе за пороговое значение была принята сумма среднего арифметического значения напряжения сигнала настройки и поправочного слагаемого, пропорционального параметру, характеризующему уровень внешней освещенности:

$$L = \frac{U_{\text{вкл}} + U_{\text{вькп}}}{2} + \frac{U_{\text{вкл}} - U_{\text{вькп}}}{6}. \quad (3)$$

Результаты обработки представлены на рис. 1. По результатам экспериментов средняя вероятность ошибки составила 0,015, что является недостаточно хорошим результатом, поскольку для приемника вероятность ошибки на бит должна составлять от 10^{-3} до 10^{-5} [2].

Усовершенствование решающего правила

Для компенсации затягивания фронтов импульса целесообразно дополнить решающее правило обработкой дополнительного параметра, который бы охарактеризовал форму сигнала. Предлагаемый нами подход заключается в определении для сигнала, соответствующего каждому биту, коэффициента наклона регрессионной прямой. Наиболее распространенным методом аппроксимации за счет вычислительной простоты является метод наименьших квадратов.

Регрессионная прямая для i -го бита имеет вид:

$$y = K_i x + b_i. \quad (3)$$

Коэффициенты этой прямой a и b определяются как:

$$[K_i, b_i] = \arg \min \sum_{j=1}^n (K_i x_j + b_i - y_j)^2, \quad (4)$$

где x_j – момент времени, соответствующий j -му отсчету. Определяющими параметрами дополненного решающего правила являются величина и знак коэффициента наклона K_i .

Алгоритм обработки, как и в случае с однопороговой схемой, разделяется на два этапа: настройку пороговых значений и непосредственно обработку сигнала по решающему правилу. На рис. 2 представлена блок-схема алгоритма решающего правила.

Пусть $K_{\text{сп}}$ – пороговое значение коэффициента регрессии, тогда если абсолютное значение коэффициента регрессии бита K_i превышает $K_{\text{сп}}$, то исследуется знак коэффициента регрессии бита:

$$K_i > 0 \rightarrow "1", K_i < 0 \rightarrow "0". \quad (5)$$

В противном случае (при отсутствии эффекта затягивания) значение бита определяется по однопороговой схеме.

Настройка осуществляется следующим образом: до передачи сообщения отправляется заранее известная тестовая последовательность. Целесообразно принять в качестве тестовой последовательности с чередованием 0 и 1, т. е. в таком тестовом сигнале можно проследить тенденции изменения формы импульсов. Тестовая последовательность обрабатывается по однопороговой схеме и проверяется на наличие ошибок. Если бит был принят ошибочно, то определяется его коэффициент регрессии. Коэффициенты регрессии записываются в подмассив, по которому определяется пороговое значения коэффициента регрессионной прямой: в качестве порогового значения переключения $K_{\text{сп}}$ целесообразно принять минимальное по модулю значение коэффициента регрессии ошибочно принятого бита. Следует заметить, что переключение решающего правила при таком выборе порогового значения осуществляется только тогда, когда однопороговая схема неизбежно приводит к ошибкам.

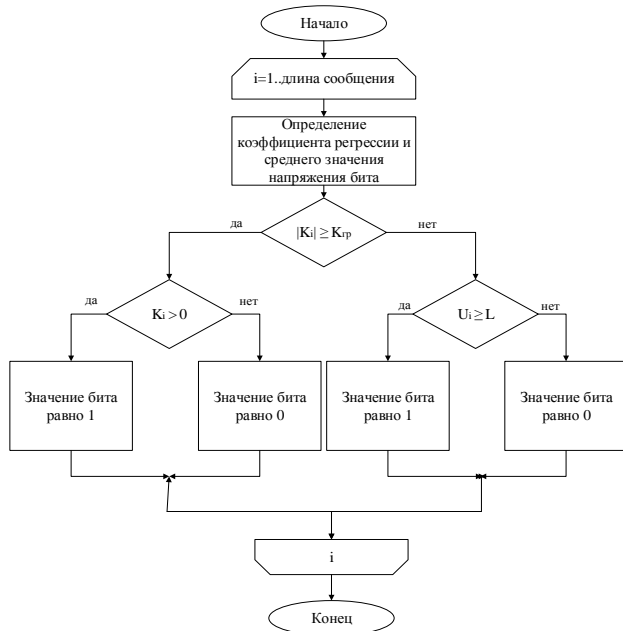


Рис. 2. Блок-схема алгоритма решающего правила

На рис. 3 представлены результаты обработки. На временных диаграммах видно, что импульсы, несмотря на искаженную форму, обработаны верно за счет переключения правила.

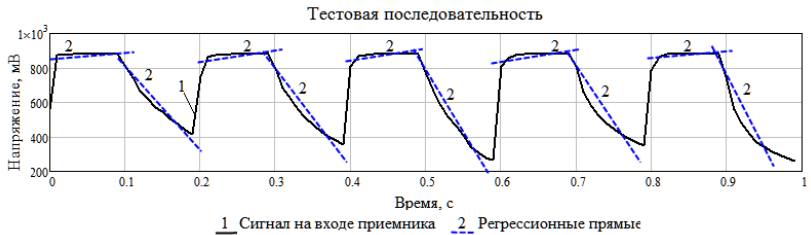


Рис. 3. Временная диаграмма сигнала тестовой последовательности

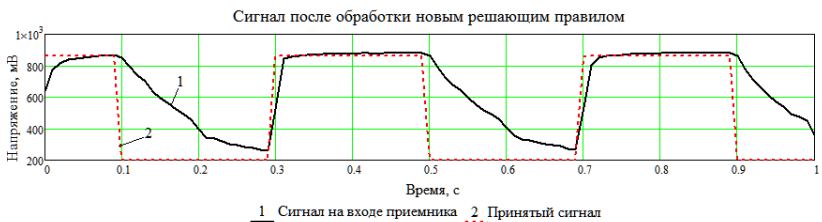


Рис. 4. Временные диаграммы принимаемого и обработанного сигналов в режиме слабой внешней освещенности

Результаты

В результате экспериментов было установлено, что средняя вероятность ошибки на бит уменьшилась с 0,015 при однопороговой схеме сравнения до $6 \cdot 10^{-4}$ при использовании усовершенствованного решающего правила. Таким образом, усовершенствованное решающее правило привело к улучшению точности обработки принятого сигнала. Полученные рекомендации необходимо учитывать при разработке макетного прототипа системы связи с передачей по оптическому каналу на расстояние.

Библиографический список

1. *Акопян Б. К., Городецкая А. В.* Разработка макетного прототипа систем цифровой связи с передачей по оптическому каналу // Семидесятая международная студенческая научная конференция ГУАП: Сб. докл.: В 3 ч. Ч. 1. Технические науки / СПб.: ГУАП., 2017. (в наст. сборнике).
2. *Скляр Б.* Цифровая связь: Теоретические основы и практическое применение. 2-е изд., испр./ пер.с англ. Е. Г. Грозы. М.: Издательский дом «Вильямс», 2003. 1104 с.
3. *Мироновский Л. А., Слаев В. А.* Алгоритмы оценивания результата трех измерений. СПб.: Проффессионал, 2010. 192 с.

УДК 621.372

Б. К. Акопян, А. В. Городецкая

студенты кафедры проблемно-ориентированных вычислительных комплексов

О. О. Жаринов

канд. техн. наук, доцент – научный руководитель

РАЗРАБОТКА МАКЕТНОГО ПРОТОТИПА СИСТЕМЫ ЦИФРОВОЙ СВЯЗИ С ПЕРЕДАЧЕЙ ПО ОПТИЧЕСКОМУ КАНАЛУ

Теория передачи дискретных сообщений является наиболее разработанной частью общей теории связи [1]. Ее основной задачей является поиск таких методов передачи, приема и обработки сигналов, которые смогли бы обеспечить малые потери информации, содержащейся в сообщении, высокую скорость передачи и максимально возможную достоверность информации, которая характеризуется средней вероятностью ошибки на бит [2]. Наиболее перспективными являются системы передачи информации, принцип работы которых основан на цифровых методах передачи. К таким относится система цифровой связи с передачей по оптическому каналу.

Принцип работы

Принцип работы системы цифровой связи с передачей по оптическому каналу заключается в осуществлении передачи посредством изменения характеристик оптического излучения. На рис. 1 представлена функциональная схема системы цифровой связи с передачей по оптическому каналу.

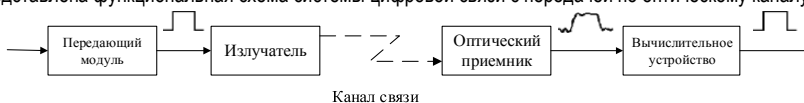


Рис. 1. Функциональная схема системы цифровой связи с передачей по оптическому каналу

Информация поступает в передающий модуль, в котором кодируется различными помехоустойчивыми кодами, трансформируется системой передатчика в излучение и передается в атмосферу. На принимающей стороне оптическая система фокусирует сигнал на фотоприемнике, который преобразует оптический пучок в электрический сигнал. Далее принятый сигнал обрабатывается вычислительным устройством, в основу работы которого положено некоторое решающее правило, и преобразуется в сигналы выходного интерфейса.

Преимущества беспроводной оптической связи в том, что оптический канал невосприимчив к радиопомехам и минимизирует время задержки передачи по сравнению с кабельными линиями. Основными недостатками подобной системы являются чувствительность к уровню внешней освещенности и следующее из этого искажение формы импульсов, что может приводить к ошибочному приему сигнала. Разработанный прототип предназначен для исследования основных проблем, возникающих в системах связи с передачей по оптическому каналу.

Аппаратная реализация системы

Оптический канал системы реализован на оптопаре: в роли излучателя выступает светодиод, в роли фотоприемника – фоторезистор (рис. 2). Поскольку прототип не предназначен для передачи на расстояние, то передающий модуль и вычислительное устройство объединены в один приемопередающий модуль. В качестве основы для приемопередающего модуля макетного прототипа использована плата с программируемым микроконтроллером Arduino Uno – самая популярная платформа любительской и образовательной электроники. Достоинствами данной платформы являются ее доступность и простота практического использования. Приемопередающий модуль реализован на одной

плате, поскольку данный прототип использовался исключительно для исследования основных проблем, возникающих при передаче по оптическому каналу.

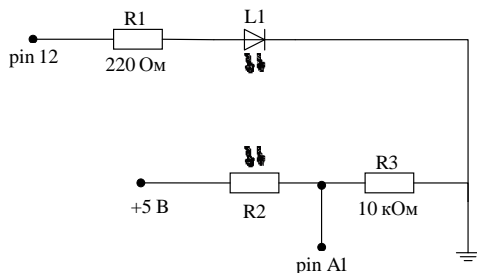


Рис. 2. Электрическая схема оптического канала

Внешний вид макетного прототипа представлен на рис. 3.

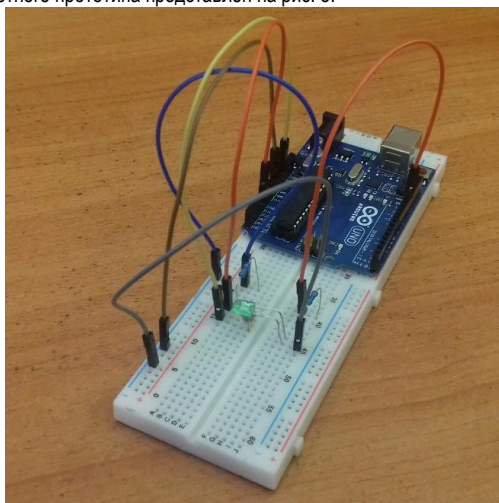


Рис. 3. Внешний вид макетного прототипа

Алгоритм работы приемопередающего модуля

Программная реализация приемо-передающего модуля осуществляется согласно блок-схеме, представленной на рис. 4. Алгоритм обработки подразделяется на два этапа: настройку пороговых значений и непосредственно обработку сигнала по решающему правилу. Перед отправкой сообщения с фоторезистора считываются значения настройки напряжения, соответствующие включенному ($U_{вкл}$) и выключенному ($U_{выкл}$) состоянию светодиода. Разность между значениями настройки целесообразно принять в качестве параметра, характеризующего уровень освещенности: чем сильнее освещенность, тем меньше значение параметра.

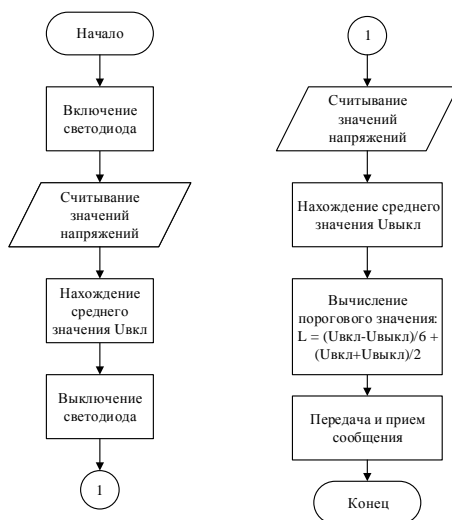


Рис. 4. Блок-схема алгоритма работы приемопередающего модуля

Далее осуществляется передача последовательности импульсов напряжения, которая соответствует исходному бинарному коду, при этом на каждый бит сообщения приходится некоторое количество отсчетов. Импульсы вызывают оптическое излучение на светодиоде, что приводит к изменению сопротивления фоторезистора. Микроконтроллер считывает и обрабатывает полученные значения напряжения в соответствии с предварительно полученными данными и осуществляет обработку сообщения.

Способ обработки основан на однопороговой схеме сравнения. Принцип работы этой схемы заключается в том, что среднее значение отсчетов напряжения i -го бита U_i сравнивается с некоторым пороговым значением L :

$$U_i \geq L \rightarrow "1", U_i < L \rightarrow "0". \quad (1)$$

Выбор порогового значения оказывает большое влияние на показатели качества связи, поэтому его определению уделяется особое внимание. В макетном прототипе за пороговое значение была принята сумма среднего арифметического значения напряжения сигнала настройки и поправочного слагаемого, введенного с целью компенсации внешней засветки. При разработке были учтены уровни слабой, средней и сильной внешней освещенности. Поправочное слагаемое пропорционально параметру, характеризующему уровень внешней освещенности: при слабом уровне внешней освещенности, когда значение параметра велико, порог будет завышаться, что должно скомпенсировать затягивание фронтов импульса (рис. 5); для сильного же уровня освещенности вклад поправочного слагаемого будет незначителен. Пороговое значение L рассчитывается по формуле:

$$L = \frac{U_{\text{вкл}} + U_{\text{вькл}}}{2} + \frac{U_{\text{вкл}} - U_{\text{вькл}}}{6}. \quad (2)$$

Результаты

Тесты показали, что в условиях сильной внешней освещенности значения напряжения, соответствующие нулю или единице бинарного кода, отличаются незначительно – это следует из того, что из-за сильного освещения фоторезистор становится менее чувствительным к изменению освещенно-

сти, вносимым светодиодом передатчика. С другой стороны, в случае слабой внешней освещенности наблюдается затягивание фронтов импульса. Это можно объяснить тем, что при снижении уровня внешней освещенности возрастает сопротивление фоторезистора, что приводит к росту зависящей от него постоянной времени заряда/разряда сопутствующих паразитных емкостей. По результатам тестов устройства средняя вероятность ошибки на бит составила 0,015, что является недостаточно хорошим результатом.

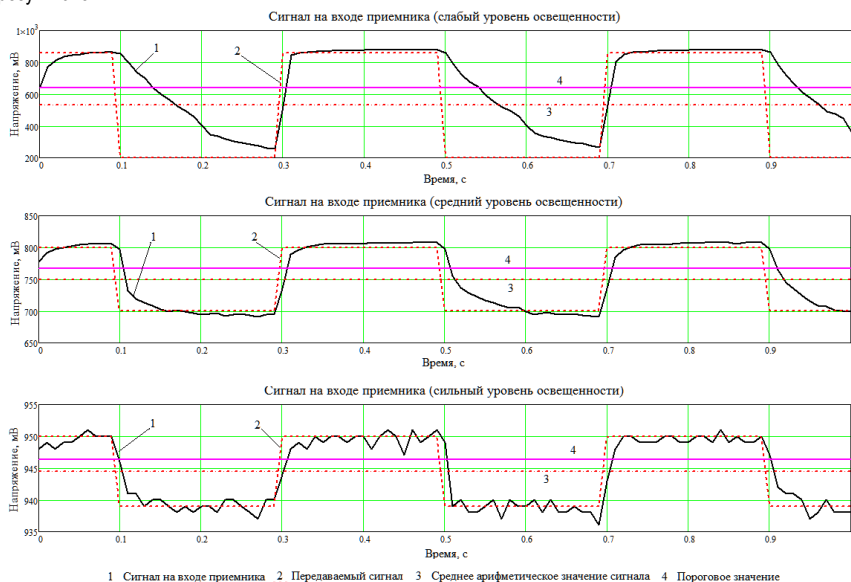


Рис. 5. Временные диаграммы сигналов на приемнике при слабом, среднем и сильном уровне внешней освещенности

Таким образом, исследования выявили ряд проблем, требующих дальнейшего поиска решений и доработки программы для улучшения работоспособности системы.

Библиографический список

1. Финк Л. М. Теория передачи дискретных сообщений. 2-е изд., перераб. и доп. / М.: Советское радио, 1970. 728 с.
2. Скляр Б. Цифровая связь: Теоретические основы и практическое применение/ 2-е изд., испр. / пер.с англ. Е. Г. Грозы. М.: Издательский дом «Вильямс», 2003. 1104 с.

УДК 004.42

О.Д. Александров

студент кафедры компьютерных технологий и программной инженерии.

В.Ю. Гамов

канд. в.наук, доцент – научный руководитель

РАЗРАБОТКА И ИНТЕГРАЦИЯ КОМПЬЮТЕРНОЙ МОДЕЛЯ ДЛЯ ПРОГНОЗИРОВАНИЯ ВЕРОЯТНОСТИ ИСХОДА ШАХМАТНОЙ ПАРТИИ

Основная цель работы – разработать программное средство, которое на основе имеющихся статистических данных делает предположение о исходе предстоящего события. Исходную цель можно представить, как ряд задач решение, которых позволит достичь поставленной цели, а именно:

- сбор данных;
- анализ данных;
- разработка тестовой модели и выбор оптимальной модели или совокупности моделей;
- доработка программного обеспечения.

Задача сбора данных подразумевает под собой сбор данных, которые будет использоваться на дальнейших этапах исследования. Для этого необходимо:

- найти источники данных;
- обнаруженные данные необходимо собрать и представить в формате удобном для исследования.

Так как невозможно, в разумные сроки проанализировать все существующие статистические данные, на анализируемые данные наложен ряд ограничений:

- только шахматисты из федерации FIDE;
- только данные за последние 10 лет;
- только шахматные партии, в которых известны имена игроков;
- только шахматные партии, в которых рейтинг обоих игроков больше 0;
- только шахматные партии, в которых известно наименование турнира;
- только шахматные партии, о которых существуют записи о ходах сделанных каждым из игроков в партии;
- только данные из полученные из fide.com и kingbase-chess.net;
- только те данные, которые распространяются в формате png.

Решение задачи анализа данных, подразумевает под собой обнаружение закономерностей и взаимосвязей в данных, а также выявления дополнительной информации о шахматной партии.

В ходе решения задач сбора и анализа данных было сформированы следующие факторы, которые имеют следующее представление(информация представлена в виде sql запроса для Postgresql).

```

1: CREATE TABLE ChessGames (
1: id SERIAL,
2: -- название турнира
3: event TEXT,
4: -- сайт, в рамках которого проходила партия.
5: -- если событие происходило оффлайн поле отсутствует
6: site TEXT,
7: date TIMESTAMP,
8: round TEXT,
9: white BIGINT REFERENCES Players(id),
10: black BIGINT REFERENCES Players(id),
11: result INT,
12: -- elo рейтинг шахматиста
13: whiteElo INT,
14: blackElo INT,
15: eco TEXT,
16: -- последовательность ходов каждого из шахматистов
17: game TEXT,
18: -- количество ходов, которое заняла шахматная партия
19: movesCount INT,
20: -- поля отражают количество грубых ошибок допущенных шахматистами.
21: whiteBlunder INT,
22: blackBlunder INT,
23: -- в среднем потерянно сотых пешки
24: whiteAcpl INT,
25: blackAcpl INT,
26: -- количество ошибок
27: whiteMistake INT,
28: blackMistake INT,
29: -- количество сомнительных ходов
30: whiteInaccuracy INT,
31: blackInaccuracy INT,
32: -- все поля далее принимают значение 0 или 1
33: standartGame INT2 DEFAULT 0,
34: rapidGame INT2 DEFAULT 0,
35: blitzGame INT2 DEFAULT 0,
36: isPlayOff INT2 DEFAULT 0,
37: isFinal INT2 DEFAULT 0,
38: isPrelim INT2 DEFAULT 0,
39: -- u12 – юноши до 12 лет, u14 юноши до 14 лет и так далее
40: isU12 INT2 DEFAULT 0, isU14 INT2 DEFAULT 0, isU16 INT2 DEFAULT 0
41: );
42:
43:

```

Исходный код 1 – sql запрос для формирования таблицы с анализируемыми в ходе работы данными.

Решение задачи разработка тестовой модели подразумевает следующее: полученные данные прогоняются через множество существующих алгоритмов машинного обучения. Среди которых выби-

раются несколько показавших наиболее высокую точность и затем выполняется оптимизация выбранных моделей с целью получения более точных результатов.

Для получения первых результатов были реализовано обучение с помощью следующих алгоритмов:

- gradient boosting;
- нейронные сети;
- random forest.

Лучший результат показал нейронные сети при тестирование на обучающей выборке точность прогнозов составила 61%, а на тестовой 49%.

Доработка программного обеспечения подразумевает под собой анализ полученных результатов и в случае необходимости реализации программного обеспечения на более низкоуровневом языке программирования, с целью ускорения работы полученной модели.

В настоящий момент в ходе выполнения работы была проанализирована предметная область, выделены основные ее понятия и сформулирована основная задача исследования, которая была представлена как ряд подзадач, решения которых позволит достигнуть цели исследования. Помимо этого были собраны и проанализированы данные и на основе этого сформировано представление данных.

Дальнейшее направление исследования связано с улучшением разработанного программного обеспечения, поиск дополнительных факторов и обучение с использованием других алгоритмов.

УДК 629.735.33

Д. А. Афонькин

студент кафедры компьютерных технологий и программной инженерии

Д. В. Богданов

канд. техн. наук, доцент – научный руководитель

МОДЕЛИРОВАНИЕ ЕДИНОГО ИНФОРМАЦИОННОГО ПРОСТРАНСТВА ПРИ ПОМОЩИ СЕТЕЙ ПЕТРИ

В области управления воздушным движением существует множество процессов от планирования до контроля за воздушным движением, многие выполняются в ручном и полуавтоматическом режиме: процессы формирования и корректировки планов, получения информации о воздушных объектах, обнаружение нарушений и аварийных ситуаций. В связи с этим, для повышения эффективности и безопасности управления необходимо комплексное решение автоматизации, которым и стало единое информационное пространство воздушных объектов [1].

Информационная система представляет из себя комплекс информационных систем, взаимодействующих по единым правилам в рамках единого информационного пространства. Структурно оно представляется распределенной сетью с асинхронной и параллельной обработкой данных.

Любая комплексная система требует исследований структуры, динамики работы, различных параметров для повышения эффективности: производительности, надежности, безопасности. Для этих целей используют различные модели.

Для анализа статической структуры и динамики применяются разные способы. Перечислим только некоторые методы для анализа динамики: дифференциальные уравнения, исчисления процессов, Сети Петри, конечные автоматы. Так как система носит асинхронный, параллельный и распределенный характер, то больше всего подходит Сети Петри, так как метод позволяет описать информационную систему с нужными параметрами и имеет простую и понятную структуру, визуально выражающую структуру самой системы.

Теория Сетей Петри начала свой путь в диссертации д-ра Петри в 1962 г. и по настоящий день служит самым распространенным инструментом для исследования дискретных информационных систем. Анализ может дать важную информацию о структуре и поведении моделируемой динамической системы, что способствует выработке предложений по улучшению структуры и функционального поведения. Область применения данного метода – исключительно моделирование. Предполагается, что модель описана на математическом языке и представляет наиболее характерные особенности изучаемого объекта и явления, которые важны для анализа [2].

Структура единого информационного пространства представляет собой многоуровневую коммуникационную сеть, которая включает:

- 1) Информационные ресурсы, включающие данные, сведения и знания, хранящиеся в базе данных;
- 2) Организационные структуры, обеспечивающие функционирование и развитие единого информационного пространства;
- 3) Средства информационного взаимодействия граждан и организаций, обеспечивающие им доступ к информационным ресурсам на основе соответствующих информационных технологий, включающие программно-технические средства и организационно-нормативные документы.

Взаимодействие происходит при помощи внутренних информационных платформ конкретных организаций, которые в рамках системы являются независимым компонентами, взаимодействующими по единым правилам. Каждая компонента имеет состояние, выражающее необходимую информацию для описания ее действий и работает параллельно и асинхронно.

Система должна обладать свойством надежности. Под надежностью в работе понимается устойчивость инфокоммуникационной системы к внешним воздействиям и внутренним возмущениям, а

также как гарантия получения имеющейся в системе информации в полном и неискаженном виде, только заданными адресатами и в удовлетворительные сроки.

Создание модели для всей системы слишком трудоемко, поэтому будет использоваться принцип декомпозиции и моделирование небольшой ее части – кластера серверов – внутренних платформ организаций с распределителем. В качестве модели будет использоваться дискретная стохастическая временная цветная Сеть Петри. На схеме 1 представлена условная анализируемая часть системы.

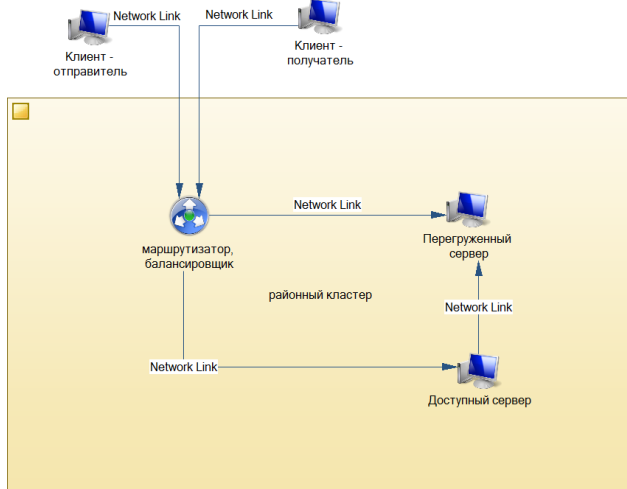


Схема 1

В терминах математической модели, сервера и клиенты – асинхронные и независимые компоненты системы, где появление на них данных – соответствующие события в Сети Петри. Условия и функции перехода информации между серверами – переходы Сети Петри. Математическая модель представляет собой набор:

$$C_i = \{P_i, T_i, W_i, \Omega_i, M_i\},$$

где P_i – множество позиций сети; T_i – множество вероятностных переходов сети, M_i – начальная маркировка сети; W_i – функция весов дуг между весами и переходами и наоборот, $\Omega_i : T_i \rightarrow (0; 1)$ – функция условных вероятностей переходов. Графическое представление Сети Петри представлено на схеме 2.

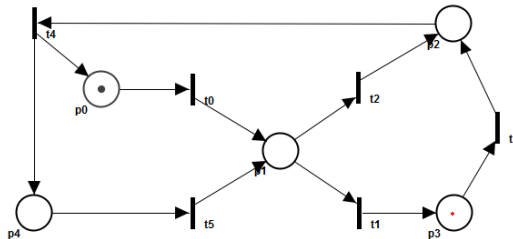


Схема 2

Модель имеет множество мест $P_i = \{p_0, p_1, p_2, p_3, p_4\}$ и переходов $T_i = \{t_0, t_1, t_2, t_3, t_4, t_5, t_6\}$. Функция условных вероятностей переходов определяется так: $\Omega_i(t_0) = pt_1$, $\Omega_i(t_1) = \Omega_i(t_2) = pt_2$, $\Omega_i(t_3) = \delta$, $\Omega_i(t_3) = pt_3$, $\Omega_i(t_4) = pt_4$, $\Omega_i(t_5) = pt_5$.

События интерпретируется следующим образом: r_0 – запрос данных, r_4 – Запрос на публикацию данных, r_1 – поиск нужного сервера и перенаправление запроса, r_2 – поиск и выдача нужной информации, r_3 – получения запроса на хранение или выдачу.

Переходы и их условные вероятности имеют следующую интерпретацию когда:

- 1) пользователь сделал запрос на получение информации, он посылается на узел маршрутизации и балансировки (переход t_0 с условной вероятностью pt_0);
- 2) пользователь сделал запрос на публикацию информации, он посылается на узел маршрутизации и балансировки (переход t_5 с условной вероятностью pt_4);
- 3) найден подходящий сервер, запрос отправляется к нему (переход t_1 и t_2 с вероятностью pt_1);
- 4) появился маркер сбоя, то данные реплицируются на доступный сервер (переход t_3 с вероятностью pt_3);
- 5) данные готовы к отправке, они отправляются обратно запрашивающему клиенту (переход t_4 с вероятностью pt_2).

В результате была получена математическая и графическая модель в виде Сети Петри для решения задачи построения надежной системы, а именно небольшой ее части, состоящей из двух клиентов, серверов и узла маршрутизации и балансировки. Это позволяет выполнять анализ достижения необходимых маркировок при помощи различных методов анализа.

Библиографический список

1. О Единой системе организации воздушного движения Российской Федерации: постановление от 28 августа 2015 г. N 901.
2. Тарасюк И. В. Стохастические сети Петри // Системная информатика: Выпуск 9: Формальные методы и модели информатики: Новосибирск: Изд-во СО РАН., 2004. С. 136-191.

УДК 339.144

М.А. Бабкова

магистрант кафедры компьютерных технологий и программной инженерии

М.В. Фаттахова

канд. физ.-мат. наук, доцент – научный руководитель

ИССЛЕДОВАНИЕ ЗАДАЧИ И МЕТОДОВ РЕШЕНИЯ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ ЗАПАСАМИ С УЧЁТОМ СЛУЧАЙНЫХ ВАРИАЦИЙ СПРОСА

Целью статьи является исследование и применение моделей и методов оптимального управления запасами.

С проблемой управления запасами сталкивается любой человек, постоянно решая, сколько и чего ему закупить для дальнейшего использования, будь то продукты для ужина или покупка жилья поближе к озеру или лесу, принятие решения в этих ситуациях зависит от множества факторов и ограничений, которые человек накладывает на поставленную цель для её достижения.

Относительно фирм по продаже ежедневно менеджер по запасам пытается разрешить задачу оптимального управления запасами для эффективной работы предприятия. Многие решают эту задачу на основе многолетнего опыта и статистических данных прошлого, некоторые прибегают к созданию математических моделей и методов для принятия решений и прогнозирования дальнейших действий. В связи с поставленной целью требуется решить следующие задачи:

- Изучить модели и методы оптимального управления запасами.
- Произвести первичную обработку реальных статистических данных для дальнейшего анализа.
- Применить методы оптимального управления запасами к обработанным статистическим данным.

Существует несколько моделей и методов оптимального управления запасами, которые требуются для создания математической постановки задачи и её решения [1].

Задачи по управлению запасами разделяются на задачи в условиях полной определенности и в условиях неопределённости и риска. Полная определённость предполагает задание всех величин математической модели. Для моделей в условиях полной определённости существуют множество методов нахождения интересующих переменных задачи, таких как:

Переменные издержки – издержки хранения.

Каждая область бизнеса характеризуется своей требуемой нормой доходности. Капитал, вложенный в этот бизнес, в среднем (по стране, региону, городу) должен давать определенный процент дохода ежегодно. Капитал, вложенный в запасы, такого процента не дает. Следовательно, неполученный процент – это издержка хранения.

Предельные издержки хранения всегда будут пропорциональны количеству хранимых единиц запаса и времени хранения.

Постоянные издержки – издержки по запуску новой партии продукции (производство) или затраты на формирование и оформление заказа – (торговля) Эти издержки не зависят от величины предполагаемой партии продукции (заказа).

В торговле их чаще всего связывают с оплатой труда менеджеров, «ведущих» этот заказ, с возможными затратами на сопровождение заказа сотрудником фирмы (контроль погрузки – разгрузки, ускорение прохождения оформления документов на таможне и т.п.), с офисными расходами при оформлении и размещении заявки поставщику на новый заказ. В производстве этим постоянным издержкам соответствуют затраты на переналадку оборудования для выпуска данной партии продукции.

Модель экономического размера заказа

Заключается в контроле количества запасов на складе, для своевременного оформления следующего заказа, чтобы избежать нехватки или избытка товара на складе.

Оптимальная частота заказа для группы товаров.

Основывается на группировке товаров по поставщикам для эффективного и экономичного управления запасами, так как в производстве может быть тысячи наименований товаров и каждый расходуется по-разному, что приводит к необходимости их группировать.

Модель производства оптимальной партии продукции.

Применяется для решения задач, в которых требуется произвести на конвейере детали для дальнейшего их использования, что приводит к вопросам произвести детали на весь период работы или производить детали по мере их уменьшения на складе, оба решения приводят к разным издержкам хранения, которые требуется минимизировать.

Условия неопределённости и риска предполагают, что спрос является случайным, что усложняет решение задачи об оптимальном управлении запасами. Но в реальной жизни спрос всегда случайный, поэтому для предприятий, перед которыми стоит задача оптимального управления запасами, необходимо создавать модели и, исходя из этого применять методы решения, такие как:

Модель фиксированного периода между заказами.

Заключается в расчёте безопасного резерва запасов для снижения риска дефицита или избытка. Расчёты зависят от закона распределения спроса и разделяются на модели фиксированного размера заказа и фиксированного периода между заказами.

Модель фиксированного размера заказа, согласно которой при постоянном (в среднем) спросе один и тот же заказ Q (объем заказа, количество единиц) делается в момент, когда уровень запаса падает до значения ROP (точка перезаказа (reorder point)). В случае если никаких случайных вариаций спроса нет, то $ROP=d \cdot L$, где d - ежедневный спрос, а L - время выполнения заявки на пополнение запаса поставщиком. При наличии случайных вариаций спроса, к среднему значению спроса за время ожидания поставки ($d \cdot L$) прибавляется безопасный резерв $SS=z \cdot \alpha \cdot s$, который обеспечивает снижение риска дефицита с 50% (если оставленный запас равен среднему спросу $d \cdot L$) до α . При этом моменты времени, когда делается заказ на восполнение запаса, перестают быть строго периодичными: поскольку спрос случаен, уровень ROP достигается в одном периоде раньше, а в другом – позже.

Такая модель удобна, если фирма торгует одним или небольшим числом товаров, каждый из которых заказывается у поставщика отдельно. На реальном оптовом складе нередко находятся несколько тысяч (а иногда и десятки тысяч) наименований различных товаров. При этом количество поставщиков, обычно гораздо меньше, так что у каждого поставщика фирма заказывает несколько различных товаров (а иногда несколько десятков и даже сотен наименований). В этом случае товары для заказа объединяются в группу и определяется оптимальная частота заказа группы товаров, минимизирующая издержки управления запасами. В этой модели моменты времени, когда делается заказ, фиксированы и строго периодичны, а размер заказа меняется в зависимости от того, каким был спрос в предыдущий период, и каким он прогнозируется в следующем периоде. В данном методе необходимо учитывать особенности периодов заказа и длительность поставки, для того чтобы получить корректные результаты.

Однопериодная модель заказа.

Модель применяется в ситуации, когда приобретаемый запас должен быть распродан в течение ограниченного промежутка времени (скоропортящиеся продукты, модная сезонная одежда и пр.). Если товар не продан по нормальной цене в этот промежуток времени (в сезон), он обязательно реализуется по сниженным ценам на внесезонной распродаже.

Модель определяет оптимальный размер заказа, максимизирующий прибыль продавца в условиях случайного спроса, когда неизбежны либо потери от распродажи излишков, либо упущена выгода при возникновении дефицита товара. Пусть прогнозируемый средний спрос на данный товар на сезон составляет d , а стандартное отклонение спроса s . Пусть нормальная цена при продаже товара в сезон составляет p , при себестоимости c , а цена единицы товара на распродаже $руцен < c$. Тогда потери от распродажи 1 единицы избытка товара составит $сизб = c - руцен$, а потери от дефицита в 1 единицу товара оценим как упущенную прибыль от несостоявшейся продажи этой единицы товара $сдеф = p - c$.

При оценке оптимального размера запаса, максимизирующего прибыль, экономисты используют подход, известный как маржинальный анализ. Согласно этому подходу максимум прибыли (или ми-

нимум упущенных возможностей, что равнозначно, если под упущенными возможностями понимать на равных основаниях и прямые потери и незаработанную прибыль) получится, если ожидаемые потери от 1 единицы дефицита равны ожидаемым потерям от 1 единицы избытка. Термин «ожидаемые» означает среднее значение потерь при многократном повторении заказа (т.е. потери за много сезонов подряд, или во многих магазинах в данном сезоне).

Данные были представлены в виде таблицы, в которой был спрос на товар за 13 недель с названием коллекций. Первичная обработка заключалась в том, чтобы распределить все коллекции на три сегмента: 1 - дорогой, 2 - дешевле, и mast sale - товар который необходимо продавать каждый день. После разделения товара по сегментам, был применён метод с фиксированным периодом между заказами.

Анализ полученных данных после применения метода:

Проведём анализ для самой продаваемой коллекции для каждого сегмента. Самой продаваемой в 1-й категории является коллекция molly, и для неё количество заказов в год = 1,49 раз. Безопасный резерв 991 единицы.

Точка перезаказа 1081 единицы.

Среднее время продажи безопасного резерва равна 64 дня.

Самая продаваемая коллекция во 2 сегменте это princessa silvia:

Количество заказов в год = 1,13 раз. Безопасный резерв 749 единиц.

Точка перезаказа 816 единиц.

Среднее время продажи безопасного резерва равна 35 дней.

Самая продаваемая коллекция сегменте mast sale это kamilla:

Количество заказов в год = 0,59 раз. Безопасный резерв 394 единицы.

Точка перезаказа 429 единиц.

Среднее время продажи безопасного резерва равна 33 дня.

В дальнейшем планируется разработка программного продукта, который будет реализовывать методы для предоставленных исходных данных.

Библиографический список

1. *Зайцев М.Г., Варюхин С.Е.* Методы оптимизации и принятия решений / М.: Изд-во «ДЕЛО», 2008. 664 с.

УДК 004.912

М. В. Бабушкин, А. Ю. Курова

студенты кафедры компьютерных технологий и программной инженерии

М. Д. Поляк

ст. преподаватель – научный руководитель

МЕРЫ СРАВНЕНИЯ ТЕКСТОВ

Одним из популярных на сегодняшний день алгоритмов анализа текста является *TF-IDF* (*term frequency – inverse document frequency*). Это простой и удобный способ оценить важность термина для какого-либо документа относительно всех остальных документов. Если слово встречается в одном документе часто, при этом встречаясь редко во всех остальных документах – это слово имеет большую значимость для того самого документа. Частоты таких терминов собираются в вектора. Угол между векторами, описывающими разные документы, показывает нам коэффициент схожести рассматриваемых текстов [1].

Слова, не несущие основной смысл, например предлоги или междометия – получают очень низкий вес TF-IDF, потому что часто встречаются во всех документах.

Сама метрика TF-IDF состоит из двух составляющих:

TF (*term frequency* – частота слова) – отношение числа вхождений некоторого слова к общему числу слов документа.

$$TF(t, d) = \frac{n_t}{\sum_k n_k}$$

Таким образом, оценивается важность слова t в пределах отдельного документа, где n_t есть число вхождений слова t в документ, а в знаменателе – общее число слов в данном документе.

IDF (*inverse document frequency* – обратная частота документа) – инверсия частоты, с которой некоторое слово встречается в документах коллекции. Учёт IDF уменьшает вес широкоупотребительных слов. Для каждого уникального слова в пределах конкретной коллекции документов существует только одно значение IDF.

$$IDF(t, D) = \log \frac{|D|}{|\{d_i \in D \mid t \in d_i\}|},$$

где $|D|$ – число документов в корпусе, $|\{d_i \in D \mid t \in d_i\}|$ – число документов из коллекции, в которых встречается t (когда $n_t \neq 0$).

Алгоритм TF-IDF принимает во внимание данные, основанные на соотношении частоты употребления ключевой фразы или слова в тексте.

TF работает только с текстом, анализирует статью на частоту ключевых слов в пределах одной страницы сайта. Например, в тексте в 100 знаков ключом является слово «спорт», TF берет во внимание общее количество знаков текста и частоту использования слова (ключа). В итоге, если в статье 3 раза присутствует слово «спорт», мы получим частоту $TF \ 3/100 = 0,03$.

IDF снижает значение часто используемых ключевых слов или фраз. Для примера опять берем текст в 100 знаков с ключевым словом «спорт». Вычисление происходит с учетом количества документов с тем же ключом. Так, всего 10 000 000 страниц из них 1000 страниц с ключом «спорт». Значит, показатель $IDF \ 1000/10000000 = 0,0001$.

В итоге вычитаем полученные значения по формуле и получаем действительный коэффициент схожести текстов.

На базе этого алгоритма строятся усовершенствованные метрики сравнения, например LDA (Latent Dirichlet allocation, латентное размещение Дирихле).

LDA является вероятностной моделью, которая извлекает тематическую структуру в большой коллекции документов. Его отличие в том, что алгоритм распознает не только частотность слов, но и определяет их тематику по именам собственным, специфическим терминам.

Модель предполагает, что каждая тема представляет собой распределение слов в словаре, и каждый документ (описанный в одном и том же словаре) является распределением небольшого подмножества этих тем. Это статистический способ сказать, что каждая тема (например, космос) имеет несколько характерных слов (звезда, планета и т.д.), и каждый документ посвящен лишь нескольким из этих тем.

Ключевым преимуществом этого метода является то, что мы допускаем вариативность терминов, поскольку документ представлен на более высоком концептуальном (тематическом) уровне, а не на уровне исходного слова.

Оцениваем лучшую тему для каждого слова, видимого в каждом документе по формуле

$$p(Z_{d,n}=k) \propto (C_k^d + \alpha) \times \frac{C_k^w + \beta}{C_k + V\beta},$$

где w – слово, указанное в n -й позиции документа d , C_k^d – количество раз, когда тема k появилась в документе d , C_k^w – это количество раз, когда слово w было оценено с темой k во всем контексте и C_k – это количество раз, когда тема k появлялась в контексте.

В рамках проведенного исследования было рассмотрены несколько текстов, и проведено их сравнить с помощью алгоритмов TF-IDF и LDA.

Используя код парсера, были проанализированы несколько статей с разных информационных источников (Time и BBC) на различные и похожие темы. Для Time алгоритм парсинга на языке Python выглядит следующим образом [2]:

```
r = requests.get(url)
from bs4 import BeautifulSoup
soup = BeautifulSoup(r.text)
film_list = soup.find('article', {'class': 'row'})
from lxml import html
tree = html.fromstring(r.text)
film_list_lxml = tree.xpath('//p[@class = «column small-12 medium-10
medium-offset-1 large-offset-2 text size-1x-large line-height-large_10M0Ygc4»')
```

В свою очередь, для BBC алгоритм парсера принимает вид:

```
r = requests.get(url)
soup = BeautifulSoup(r.text)
bbc = soup.find('div', {'property': 'articleBody'})
bbc_p = bbc.find_all('p')
```

Среди проанализированных статей были: «Trump 'compromising' claims: How and why did we get here?», «Ex-M16 officer Christopher Steele in hiding after Trump dossier» на разные темы с одного источника и скорее всего написанные одним автором, но так как одна статья является продолжением другой, алгоритм TF-IDF определил коэффициент схожести, как 0,64. Это достаточно большое значение, т.е. несмотря на то, что тексты написаны разными авторами и не имеют одинакового заголовка, алгоритм все равно нашел схожесть тем.

Если взять другую пару текстов «The Story Behind TIME's Original 9/11» и «The 9/11 terrorist attacks», их заголовки довольно схожи. Но так как эти тексты взяты с разных источников и написаны разными авторами в разный промежуток времени и несхожими стилями, то коэффициент схожести получился всего 0,28. В одном описывались ужасные события 11 сентября, и датирован он 2001 годом, во втором же были просто воспоминания фотографа о том ужасном дне, написанные спустя несколько лет после трагедии. Код алгоритма TF-IDF имеет следующий вид:

```
stemmer = nltk.stem.porter.PorterStemmer()
remove_punctuation_map = dict((ord(char), None) for char in string.punctuation)
```

```
def stem_tokens(tokens):
```

```
return [stemmer.stem(item) for item in tokens]

def normalize(text):
    return

stem_tokens(nltk.word_tokenize(text.lower().translate(remove_punctuation_map)))
#приводим все буквы к нижнему регистру и убираем стоп слова
vectorizer = TfidfVectorizer(tokenizer=normalize, stop_words='english')

def cosine_sim(text1, text2):
    tfidf = vectorizer.fit_transform([text1, text2])
    #перемножение матрицы на обратную
    return ((tfidf * tfidf.T).A)[0,1]
```

Такой способ сравнения используется на новостных сайтах и в любых сферах, где нужно быстро работать с информацией, ведь в наше время ее очень много, и поэтому необходимы способы в ней ориентироваться. Данный метод активно используется поисковыми сервисами и позволяет обобщить статьи на схожие темы.

Однако данный алгоритм имеет свои недостатки, которые можно нивелировать с помощью алгоритма LDA, а так же увеличить точность определения схожести текстов за счет анализа имен собственных (NER, Named Entity Recognition).

Библиографический список

1. TF-IDF [Электронный ресурс]: Википедия – свободная энциклопедия. URL: <https://ru.wikipedia.org/wiki/TF-IDF> (Дата обращения: 20.03.2017).
2. *Марк Луц, Дэвид Ашер. Изучаем Python / М. Луц; Издательство O'REILLY, 2011.*

УДК 629.735.33

О.Л. Балацкая

магистрант кафедры компьютерных технологий и программной инженерии

Д.В. Богданов

канд. техн. наук, доцент – научный руководитель

ИССЛЕДОВАНИЕ ВОПРОСОВ ПРИМЕНЕНИЯ МЕТОДОВ СТАТИСТИЧЕСКОЙ ОБРАБОТКИ ДЛЯ КОНТРОЛЯ ПОЛЕТОВ ВОЗДУШНЫХ ОБЪЕКТОВ

Возрастающая интенсивность полетов воздушных судов и повышение требований по безопасности полетов требует применения современных методов и инструментов в Единой системе организации воздушного движения, предназначенной для организации рационального использования воздушного пространства РФ.

Организация безопасного, экономичного и регулярного воздушного движения подразумевает установление структуры воздушного пространства, планирование и координирование использования воздушного пространства между пользователями воздушного пространства, обеспечение разрешительного порядка использования воздушного пространства, организацию и обслуживание (управление) воздушным движением, контроль за соблюдением Федеральных правил использования воздушного пространства РФ.

Организация полетов и воздушного движения включает планирование, состоящее в предварительном распределении заявленных воздушных судов по месту и времени в воздушном пространстве и управление ходом выполнения планов, состоящее в контроле состояния и местоположения воздушных судов от взлета до посадки и выдаче экипажам команд по изменению местоположения воздушных судов. Данные действия реализуются оперативными органами Единой системы организации воздушного движения, в частности, – диспетчерами.

Большие объемы информации о воздушной обстановке, о движении воздушных судов, данные от средств наблюдения и навигации делают актуальными использование статистических методов обработки такого рода информации.

Деятельность диспетчера управления воздушным движением основана на построении и анализе мысленного образа, складывающейся в его секторе воздушной обстановки и тенденций ее развития. Для создания такой модели ему необходимо предоставить полноценную исходную информацию, фактическую и прогнозируемую, о местоположении контролируемых воздушных судов, о курсе и скорости движения, об их плановых траекториях, о метеоусловиях.

Методы статистической обработки – это математические приемы, формулы, способы количественных расчетов, с помощью которых показатели, получаемые в ходе эксперимента, можно обобщать, приводить в систему, выявляя скрытые в них закономерности [1].

Как известно, все методы математико-статистического анализа условно делятся на первичные и вторичные.

Первичные методы статистической обработки – это такие методы, с помощью которых можно получить показатели, непосредственно отражающие результаты проводимых в эксперименте измерений.

К первичным методам относят, например, определение выборочной средней величины, выборочной дисперсии, выборочной моды и выборочной медианы.

Вторичными называют методы статистической обработки, с помощью которых на базе первичных данных выявляют скрытые в них статистические закономерности.

В число вторичных методов обычно включают корреляционный и регрессионный анализы, методы сравнения первичных статистик у двух или нескольких выборок.

Одной из основных составляющих любого анализа данных является описательная статистика. Ее главной задачей является предоставление сжатой и концентрированной характеристики изучаемого явления в числовом и графическом виде.

При изучении той или иной генеральной совокупности нам неизвестен либо закон ее распределения, либо параметры распределения. В подобных случаях в математической статистике выдвигается некоторое предположение относительно свойств генеральной совокупности. Такое предположение носит название статистической гипотезы.

Наиболее часто употребляемыми критериями для проверки гипотезы о законе распределения являются критерий Пирсона (критерий χ^2), критерий Романовского, Колмогорова, Ястремского, Стьюдента, Фишера и другие.

В качестве примера применения методов статистической обработки, а именно методов первичной обработки, можно привести результаты анализа.

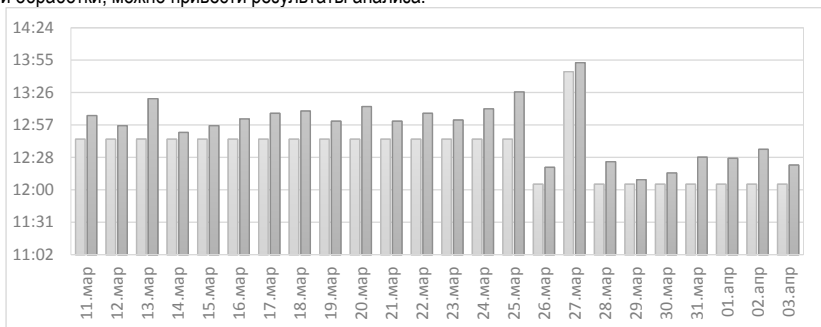


Рис. 1. Сравнение планируемого времени вылета рейса с реальным

Можно заметить, что ни один из рейсов не вылетел в точно установленное время (рис.1).

Как видно из графика (рис. 2), самое минимальное время, на которое задерживался рейс, составляло 4 минуты, самое максимальное – 1 час 48 минут.

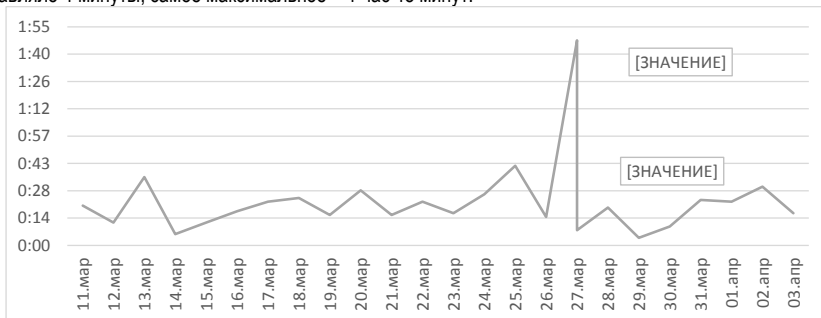


Рис. 2. Время задержки вылета рейса

Применяя методы статистической обработки, был рассчитан доверительный интервал, который показывает, что средняя задержка вылета должна находиться в промежутке от 23 до 35 минут.

Для того, чтобы проверить соответствие выборки нормальному закону распределения, были применены критерий Пирсона, критерий Романовского и приближенный критерий, который использует выборочные статистики, а именно асимметрию и эксцесс.

1) Расчетное значение критерия Пирсона намного больше критического ($20 > 0,711$), следовательно, данные выборки, характеризующие задержку вылета рейса, не подчиняются нормальному распределению.

2) Расчетное значение критерия Романовского больше 3 ($6,71 > 3$). Это говорит о том, что расхождение между эмпирическим и теоретическим распределением довольно существенно, поэтому данные выборки также не подчиняются нормальному распределению.

3) Приближенный критерий. Так как расчетные значения асимметрии и эксцесса больше их среднеквадратических отклонений ($3,99 > 0,33$; $21,44 > 0,09$), то делаем вывод, что данные выборки, характеризующие задержку вылета рейса, не подчиняются нормальному закону распределения.

В ходе данного исследования были изучены и проанализированы первичные данные, в частности – планируемое и реальное время отправления рейса воздушного судна. К этим данным были применены первичные методы статистической обработки и описательная статистика. Для проверки согласованности эмпирического распределения с теоретическим нормальным, были применены 3 критерия, ни один из которых не подтвердил близость выборочной совокупности к нормальному распределению.

Библиографический список

1. Губин В.И. Статистические методы обработки экспериментальных данных: учеб. пособие / Тюмень: ТюмГНГУ, 2007. 202 с.

УДК 004.514

В. С. Батунов

студент кафедры проблемно-ориентированных вычислительных комплексов

Е. А. Бакин

канд. техн. наук, доцент – научный руководитель

РАСШИРЕНИЕ ФУНКЦИОНАЛЬНОСТИ БИБЛИОТЕКИ ПОСТРОИТЕЛЯ ГРАФИКОВ В СРЕДЕ PYTHON

Назначение научной графики – наглядное изображение объектов исследований, графическая обработка результатов расчетов, проведение вычислительных экспериментов с наглядным представлением их результатов [1].

Одним из основных средств, для такого моделирования и построения графиков, является пакет Matlab. Язык, инструментарий и встроенные математические функции в него позволяют исследовать различные подходы и получать решение быстрее, чем с использованием электронных таблиц или традиционных языков программирования, таких как C/C++ или Java [2]. Также, благодаря встроенному в Matlab редактору, можно построить и оформить график не написав не единой строчки кода. Но данный пакет, как и множество других (Wolfram, AnyLogic), распространяется как проприетарное программное обеспечение.

Аналогом, свободно распространяемым и не уступающим по функционалу, является связка языка программирования Python и Matplotlib, а конкретнее – модуль PyPlot. Данный модуль позволяет строить сложные графики, ничем не уступающие Matlab, а Python во многом похож на язык программирования Matlab.

Но при использовании данной связки возможности по оформлению графика при помощи графического интерфейса ограничены лишь изменением легенды и свойств линий графика, таких как цвет линии, ее толщина.

При этом все функции, содержащиеся в Matlab (нанесение текстовых подписей на график, геометрических фигур, добавление различных указателей и т.д.), присутствуют в PyPlot, но все они доступны строго из кода.

Для понимания, насколько это может быть неудобно, в PyPlot имеется 11 различных способов нарисовать на графике «стрелку», и у каждой из них свои особенности. Некоторые из них построены как полигоны на поле графика и деформируются при масштабировании. А при изменении свойств стили указателя стрелки пользователь столкнется с проблемой отсутствия единого интерфейса.

Основываясь на наблюдениях, был сформулирован следующий план на разработку:

- копирование, удаление, вставка графиков;
- изменение основных параметров графиков (толщина линии, тип линии, цвет, маркер, размер маркера);
- добавление и настройка легенды графика;
- добавление и изменение подписи осей;
- реализовать «зум» графика с помощью колеса прокрутки;
- возможность добавления графических элементов «стрелка», «прямоугольник», «текст», «окружность»;
- копирование, удаление, вставка всех графических элементов;
- изменение основных параметров графических элементов и их расположения (перемещение);
- сохранение графика в различных форматах (png, svg, tif);
- редактирование точек графиков;
- создание единого интерфейса задания стиля указателей.

Как можно заметить, цель работы – сделать библиотеку более дружелюбной к пользователю с помощью GUI, так же была написана универсальная функция для изменения свойств и стиля указателя стрелки.

Из множества библиотек для построения GUI была выбрана Qt (PyQt), удобный и мощный инструмент, наиболее популярный при решении задач проектирования графического интерфейса с использованием языка Python, а благодаря наличию готового решения для работы PyQt с этой библиотекой процесс разработки был значительно облегчен.

PyQt распространяется по лицензии GPL, с ее использованием написана оболочка KDE и такие программы как VLC, Autodesk Maya, Skype. Так же, данная библиотека является кросс-платформенной, что позволит использовать проекты с использованием PyQt на всех современных платформах. А в сети интернет можно с легкостью найти примеры реализации различного функционала с ее использованием.

При проектировании графического интерфейса с использованием PyQt принято располагать все элементы на слоях, которые автоматически меняют размеры относительно окна, сохраняя общую геометрию приложения. На один из таких слоев помещается специальный объект из библиотеки PyQt, в котором выводятся все графики. В результате при разработке область для графиков является обычным объектом библиотеки PyQt.

Как и для всех приложений с использованием PyQt, для разработанного решения применимы те же самые особенности. Так, перед вызовом окна необходимо создать объект приложения (экземпляр QApplication).

После создания объекта приложения, пользователю будет доступен вывод окна с графиками, для этого необходимо вызвать команду из разработанной библиотеки «plot(x,y,kwags)», где x, y – точки по осям x, y соответственно; kwags – список свойств линии (поддерживается синтаксис языка Matlab, например «r-» сделает линию красной и пунктирной).

Команда «plot()» возвращает экземпляр объекта класса «QWidget» содержащий графики. Для нанесения дополнительных линий, пользователю необходимо вызывать команду «plot» для объекта окна. Так же доступны основные контейнеры библиотеки «pyplot», такие, как figure, axes, canvas по указателям в одноименных переменных.

Пример разработанной программы указан в листинге 1, где plotGUI название модуля разработанного решения. Результатом выполнения данного кода будет вывод двух окон с графиками. В первом график будет сплошной линией синего цвета, во втором – красной пунктирной. На рис. 1 изображен график, оформленный при помощи разработанного решения.

Листинг 1. Код программы с использованием разработанного решения

```
import sys
import random
import plotGUI
from PyQt4 import QtGui

app = QtGui.QApplication(sys.argv)

# вывод первого окна. График синяя сплошная линия
x = [i-5 for i in range(10)]
y = [round(random.random()*100)/100 for i in range(10)]
win1 = plotGUI.plot(x,y,'b-')
win1.show()

# вывод второго окна. График красная пунктирная линия
x = [i-5 for i in range(10)]
y = [round(random.random()*100)/100 for i in range(10)]
win2 = plotGUI.plot(x,y,'r-')
win2.show()
```

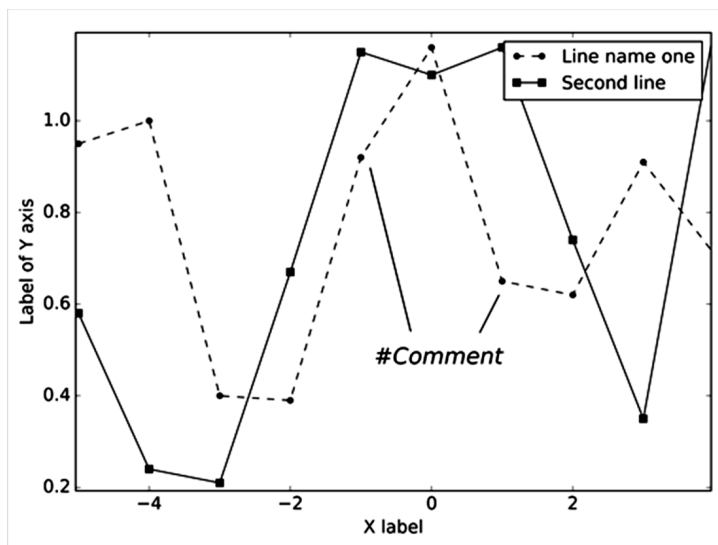


Рис. 1. Пример оформленного графика

Дальнейшим шагом в развитии модуля будет добавление поддержки функции subplot и исправление бага с масштабированием, когда после увеличения сбиваются пропорции видимой области, добавление функционала горячих клавиш.

Библиографический список

1. Пантюхин П. Я., Быков А. В., Репинская А. В. Компьютерная графика. М.: Инфра-М, 2007. 88 с.
2. Кривилев А. В. Основы компьютерной математики с использованием системы MATLAB. М.: Изд-во Лекс-Книга, 2005. 496 с.
3. Matplotlibtutorial, <https://www.labri.fr/perso/nrougier/teaching/matplotlib/>
4. GettingStartedwithPyQt, <https://wiki.python.org/moin/PyQt/Tutorials>
5. PyQt Class Reference, <http://pyqt.sourceforge.net/Docs/PyQt4/classes.html>

УДК 004.514.64, 004.514.62, 004.582.

Т.А. Белозуб

студентка кафедры компьютерных технологий и программной инженерии

С.В. Щёкин

канд. техн. наук, доцент – научный руководитель

АДАПТАЦИЯ ИНТЕРФЕЙСА СРЕДЫ BLENDER 3D ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ РАБОТЫ С ЭЛЕМЕНТАМИ ГРАФИЧЕСКОЙ СРЕДЫ

На примере среды Blender 3D было рассмотрено выполнение основных принципов, соблюдаемых для создания удобного для пользователя интерфейса. Согласно методологии по созданию дизайна пользовательского интерфейса был проанализирован и дополнен интерфейс 3D-редактора Blender.

Blender является бесплатным 3D-редактором с открытым исходным кодом и с превосходным функционалом, которым может посоревноваться со своими платными аналогами для 3D-моделирования и анимации. Но он не особо популярен среди дизайнеров из-за того, что его интерфейс отпугивает большую часть аудитории [1]. Создание качественного интерфейса требует значительно большего, чем просто соблюдение некоторых инструкций. Оно предполагает реализацию принципа «интересы пользователя превыше всего» и соответствующую методологию разработки всего программного продукта [2].

Постановка проблемы: интерфейс Blender'a особенно неудобен для новичков, он сильно замедляет процесс обучения в данной среде.

Актуальность данной работы заключается в том, что усовершенствованный интерфейс Blender'a расширит аудиторию пользователей, ведь далеко не у всех 3D-дизайнеров, особенно начинающих, есть достаточно средств на программы вроде 3DS Max или Maya, а Blender является бесплатным ресурсом, и по функционалу ни чем не уступает широко используемым платным своим аналогам.

Целью работы является изучение структуры исходного кода и адаптация интерфейса среды Blender 3D версии 2.78 для повышения эффективности работы с этим 3D-редактором.

К каждой функции в Blender'e привязана своя комбинация клавиш, и поэтому он сложен в изучении. Вместе с этим, именно горячие клавиши в Blender'e помогают пользователю быстро делать проекты.

Для того чтобы повысить эффективность работы с элементами графической среды Blender 3D, следует выделить следующие задачи:

- найти и проанализировать отзывы пользователей о Blender'e;
- выявить часто используемые горячие клавиши на основе уроков для начинающих;
- сформировать план по изменению интерфейса согласно правилам по созданию дизайна пользовательского интерфейса;
- скорректировать исходный код среды Blender 3D 2.78.

На форумах [3,4] были собраны отзывы о программе Blender. Пользователи озадачены тем, как использовать Blender, и если анализировать вообще всю критику пользователей касательно интерфейса программы, то их можно разделить на несколько групп: жалобы, касающиеся настроек пользователя, жалобы, касающиеся обучающего материала и жалобы, касающиеся недостатка отображения функционала в виде кнопок на панелях интерфейса.

Перечислим, исходя из курсов для начинающих [5,6], некоторые часто используемые сочетания клавиш: Shift-A – вызов меню для добавления объектов, Shift-S – масштабировать, R – вращать, Alt-E – выдавливать, CTRL-J – соединить объекты, CTRL-Z – отменить, CTRL-Y – повторить. Согласно правилам по созданию пользовательского интерфейса [2] был сформирован список изменений интерфейса. В рамках данной статьи внесение в пакет Blender обучающего материала рассматриваться не будет.

1) *Объектный режим*


```

layout.operator(«ed.undo»)
layout.operator(«ed.redo»)
if mode_string == 'OBJECT':
    layout.operator(«object.join»)
elif mode_string == 'EDIT_MESH':
    layout.operator(«mesh.extrude_faces_move», text=«Extrude»)
    
```

Все эти функции уже были реализованы в среде Blender, их оставалось только вывести на экран. Подобным образом можно расширять круг отображаемых функций для пользователя, если это будет нужно. Одно из правил по созданию дизайна интерфейса гласит, что хороший интерфейс – это неперегруженный интерфейс. Так что теперь, когда панель с нужными пользователю инструментами готова и перенесена в верхнюю часть окна, бросаясь этим в глаза пользователю, наша работа подошла к завершению.

В итоге получили следующий результат:

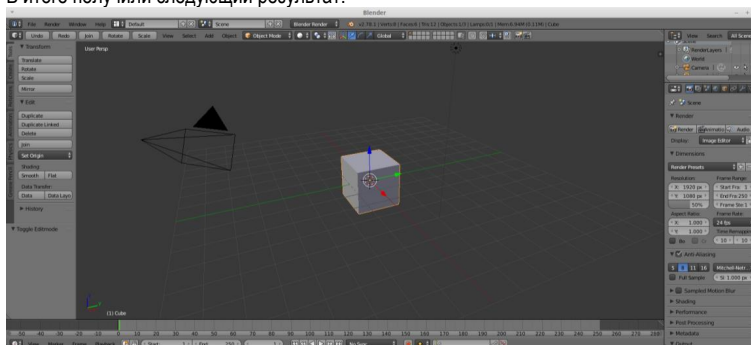


Рис. 4. Конечный результат

В завершении данной статьи хочется отметить, что интерфейс Blender'a можно еще доработать, внося в него обучающий материал, примеры готовых моделей, но на данном этапе мы пришли к поставленной цели, интерфейс Blender'a стал адаптирован как для пользователей, открывающих его впервые, так и для пользователей, которые им пользовались уже продолжительное время. Важным было внести изменения таким образом, чтобы интерфейс все также оставался понятным для тех, кто его помнит прежним.

Библиографический список

1. Официальный сайт Blender. [Электронный ресурс]. – Режим доступа: <http://blender.org> (дата обращения 28.03.2017)
2. Принципы разработки пользовательского интерфейса. [Электронный ресурс]. – Режим доступа: <http://www.cmsmagazine.ru/library/items/usability/principles-of-user-interface-design/> (дата обращения 28.03.2017)
3. Blender – программа для создания 3D моделей и игр – отзывы [Электронный ресурс]. – Режим доступа: http://otzovik.com/reviews/blender-programma_dlya_sozdaniya_3d_modeley_i_igr/ (дата обращения 29.03.2017)
4. Отзывы о программе Blender. [Электронный ресурс]. – Режим доступа: <http://www.softportal.com/response-4443-blender-1.html> (дата обращения 29.03.2017)
5. 3D-моделирование в Blender: курс для начинающих. [Электронный ресурс]. – Режим доступа: <http://younglinux.info/blender.php> (дата обращения 30.03.2017)
6. Видеоуроки для начинающих. [Электронный ресурс]. – Режим доступа: <http://4create.ru/training/104-uroki-blender-3d-rus.html> (дата обращения 30.03.2017)

УДК 681.518.2

В.В. Берёза

бакалавр кафедры проблемно-ориентированных вычислительных комплексов

Н.Н. Григорьева

ст. преподаватель – научный руководитель

ИНФОРМАЦИОННАЯ СИСТЕМА ПО СБОРУ ДАННЫХ О СОСТАВЕ И ТЕХНИЧЕСКОМ СОСТОЯНИИ ПЕРЕДАЮЩИХ РАДИОТЕХНИЧЕСКИХ ОБЪЕКТОВ

В настоящее время системы сбора данных от удаленных источников в центр их обработки встречаются почти во всех областях человеческой деятельности. В данной работе рассматривается проблема сбора и передачи информации о составе, техническом состоянии и соответствии требованиям санитарных норм передающих радиотехнических объектов (ПРТО), расположенных в различных регионах.

В состав региональных комплексов ПРТО могут входить: базовые станции сетей сотовой связи (БС), радио- и телевизионных станции, земные станции спутниковой связи (ЗССС), радиолокационные станции (РЛС). Пример структуры системы сбора информации из регионов приведен на рис. 1.

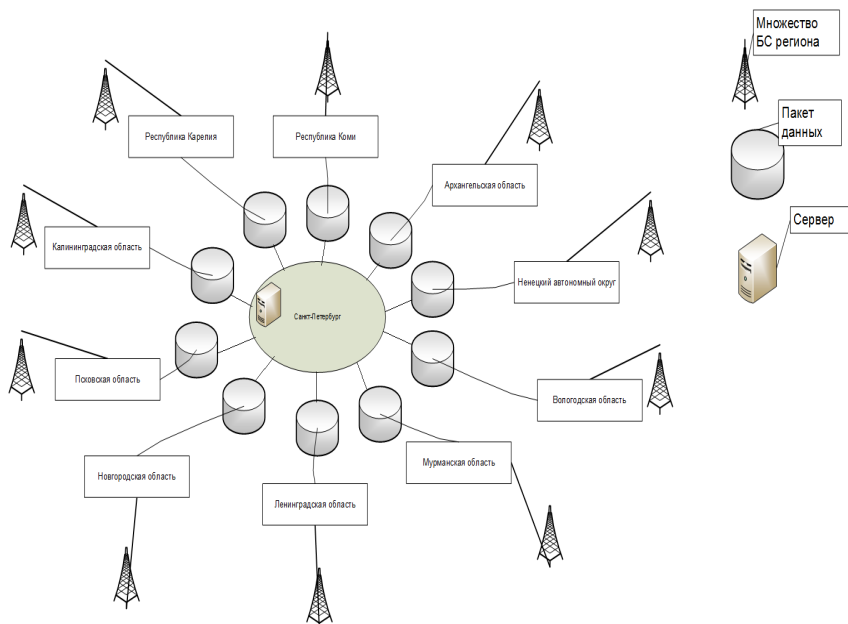


Рис. 1. Схема передачи данных

В регионах собирается для передачи в центр информация различного характера. Собираются общие сведения о ПРТО, начиная от наименования и типа оператора, описания используемого оборудования и места его размещения. Обязательно собирается информация о ПРТО, по которым проведе-

на модернизация и осуществлён их ввод в эксплуатацию. Регистрируются результаты измерений параметров ПРТО, количество протоколов по контрольным измерениям и по обнаруженным нарушениям, в частности – о превышениях предельно допустимых уровней измеряемых параметров (ПДУ) [4].

Такие измеряемые показатели технического состояния ПРТО, как напряженность электрического поля E и плотность потока энергии ППЭ, измеряются вручную по каждому радиотехническому объекту специально привлекаемыми специалистами. Проверяется соответствие этих показателей основным нормативным документам, регламентирующим допустимые уровни воздействия электромагнитного излучения [1, 2, 3].

В качестве примера формы представления информации об измеренных параметрах ПРТО можно рассмотреть табл. 1.

Таблица 1

Результаты измерений параметров ПРТО

Место измерения	Высота над уровнем земли, м	Расстояние До АМС ЭМИ, м	Полоса частот		Результаты измерений интенсивности ЭМИ				
			Fмин МГц	Fмакс МГц	E В/м	E пду В/м	ППЭ мкВт/см ²	Погрешность мкВт/см ²	ППЭ пду мкВт/см ²

Разрабатываемая информационная система создаётся для расширения функций уже работающей системы, поэтому разработка должна вестись в MS Access. Для организации запросов будет использоваться SQL.

Информация из регионов в Центр должна поступать из всех регионов раз в неделю пакетами, экспортируемыми в определённом формате. Формат данных будет уточняться заказчиком. В соответствии с требованиями к системе и структурой собираемых данных были определены основные таблицы разрабатываемой базы данных: измерения, информация о ПРТО, организации и сотрудники, проводящие измерения. Укрупнённая схема базы представлена на рис. 2.

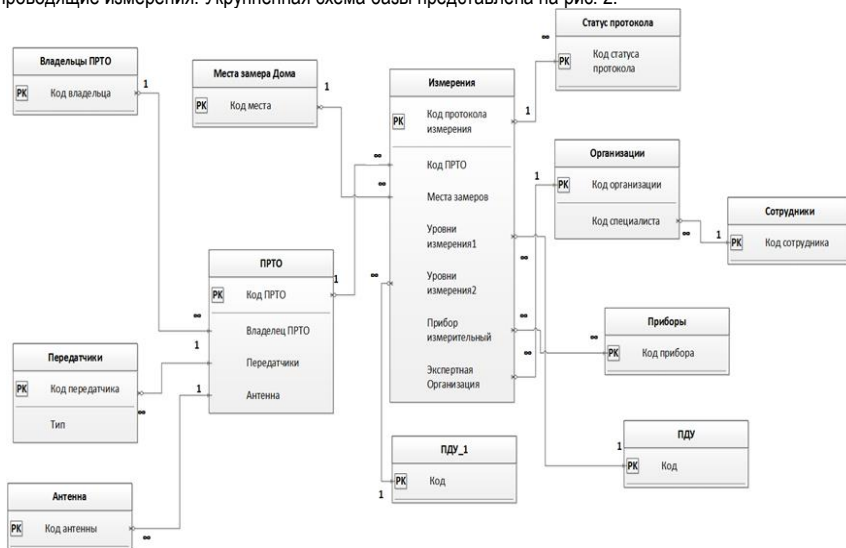


Рис. 2. Укрупнённая схема БД

Основной целью проектируемой информационной системы является создание отчётов и протоколов. В ходе детальной проработки предметной области было выявлено две глобальные проблемы:

- в данный момент регионы высылают в центр отчёты, выполненные в различных формах.
- форма протоколов, оформляемых различными проверяющими организациями (центр гигиены, радиочастотный центр и др.), отличается как по организациям, так и по регионам.

Следовательно, при разработке информационной системы в первую очередь должны быть решены следующие задачи:

1) сформировать общую форму отчётов по сбору, предоставлению и отправке информации из регионов в центр;

2) упорядочить варианты оформления протоколов различных проверяющих организаций и создать возможность добавления в стандартизованные формы комментариев от экспертов.

Таким образом, в данной работе проведён анализ передачи данных внутри организации, составлена структурная схема базы данных. Планируется создание форм отчётов и протоколов с возможностью комментирования и редактирования [5].

Библиографический список

1. ГОСТ 12.1.006-84 «ССБТ. Электромагнитные поля радиочастот. Допустимые уровни»
2. СанПиН 2.2.4/2.1.8.055-96 «Электромагнитные излучения радиочастотного диапазона».
3. МУК 4.3.1167-02 – Определение плотности потока энергии электромагнитного поля в местах размещения радиосредств, работающих в диапазоне частот 300 МГц – 300 ГГц.
4. Агеева А. А. Исследование электромагнитной обстановки от передающих объектов в г. Владивостоке с использованием геоинформационных систем // Известия ЮФУ. Технические науки. – № 9 (122). 2011.
5. Михеева В.Д., Харитонова И.А. Microsoft Access 2002. – СПб.: БХВ – Петербург, 2007. – 1040 с.

УДК 004.658.2

В. В. Боженко

студентка кафедры проблемно-ориентированных вычислительных комплексов

Г. С. Евсеев

канд. техн. наук, доцент – научный руководитель

ОЦЕНКА ТРУДОЕМКОСТИ СОЗДАНИЯ OLAP-КУБОВ В MSSQLСЕРВЕРЕ

Информационные системы предприятия обычно включают в себя системы для поддержки принятия решений, которые располагают средствами предоставления пользователю агрегатных данных для всевозможных выборок из первоначального набора в удобном для анализа виде. Эти агрегатные сведения образуют многомерный набор данных, который называется куб. Куб состоит из измерений (например, «дата, товар, покупатель, поставщик») и мер. При этом каждому набору значений измерений соответствует ячейка, в которой размещены числовые показатели, связанные с этим набором. Таким образом, между объектами бизнес-процесса и их числовыми характеристиками будет установлена однозначная связь. На рис. 1 представлено наглядное изображение куба данных.

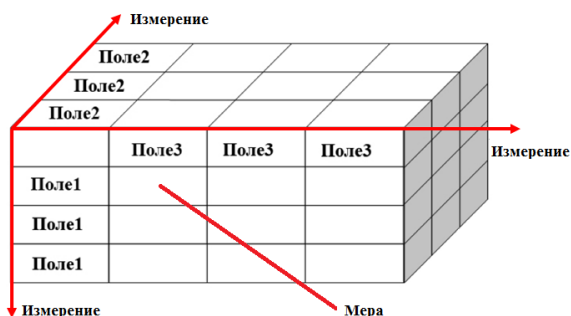


Рис. 1. Визуализация куба данных

OLAP (OnlineAnalyticalProcessing) –технология оперативной аналитической обработки данных, которая использует методы и средства для сбора, хранения и анализа многомерных данных в целях поддержки процессов принятия решений.

Главная идея OLAP заключается в создании многомерных кубов, которые будут доступны для запросов пользователей. Термин OLAP неделимо связан с термином хранилище данных, так как исходные данные для построения OLAP-кубов обычно содержатся в хранилищах данных. Назначение хранилищ данных – это обеспечение пользователей информацией для статистического анализа и принятия управленческих решений. Этапы создания OLAP-системы представлены на рис. 2. Из обычной реляционной базы для оперативной обработки транзакций (OLTP), которая обычно ведется на предприятии, создается хранилище данных, а оно в свою очередь является источником для многомерной базы данных, которая хранится на OLAP-сервере и используется для создания кубов и анализа данных, хранящихся в этой базе.

Структура базы данных для хранилища обычно разрабатывается таким образом, чтобы максимально облегчить анализ информации, ведь это основная функция хранилища [1]. При этом классическая схема хранилища данных значительно отличается от структуры базы данных для оперативной обработки транзакций. Обычно данные копируются в хранилище из OLTP-систем согласно расписанию. Теоретически, процесс пополнения хранилища – это только добавление новой информации за какой-то период времени, без изменения уже имеющейся в хранилище информации. Чаще всего хранилище

данных частично денормализовано и может допускать избыточность данных, однако это делает возможным повышение скорости выполнения запросов. При этом ошибки в хранилищах данных не предполагаются, так как могут привести к неправильной работе аналитических алгоритмов, кроме этого в хранилище данные содержатся за большие периоды времени.



Рис. 2. Этапы создания OLAP-системы

Хранилище данных состоит из таблицы фактов и таблицы измерений. Таблица фактов – это главная часть хранилища данных [1]. Принято, что она содержит информацию об объектах или событиях, которые будут подвержены анализу. Таблица фактов имеет составной первичный ключ, состоящий из первичных ключей всех таблиц измерений. По этой причине первичный ключ таблицы фактов состоит из нескольких внешних ключей [2]. Помимо этого, таблица фактов включает в себя одно или несколько числовых полей, на основании которых потом будут получены агрегатные значения для анализа информации.

В таблицах измерений обычно находятся редко изменяемые данные, которые не требуют постоянных обновлений. Таблицы измерений также содержат минимум одно описательное поле и целочисленное ключевое поле для того чтобы однозначно определить член измерения. Кроме того, каждая таблица измерений должна находиться в отношении «один ко многим» с таблицей фактов [2]. Заметим, что скорость роста таблиц измерений должна быть небольшой по сравнению с ростом таблицы фактов.

Важным требованием для хранилища данных является то, что оно должно иметь структуру «звезды» или «снежинки». Для «звезды» характерно, что каждое измерение куба содержится в одной таблице, а для «снежинки» – хотя бы одно измерение содержится в нескольких связанных таблицах (например, «год, квартал, месяц»).

Хранилище часто рассматривается как промежуточная часть, фундамент для многомерных баз данных. Именно из хранилища создается многомерная база, из которой затем получаются OLAP-кубы. Стоит отметить, что для визуального представления данных, хранящихся в кубе, используются обычные двумерные таблицы, имеющие сложные иерархические заголовки строк и столбцов. Такие таблицы называются сводными. В горизонтальной оси таблицы (заголовки столбцов) показано одно измерение, в вертикальной (заголовки строк) – другое, а в ячейках таблицы указаны значения мер. При этом набор мер в сущности рассматривается как одно из измерений: либо выбирается для показа одна мера (и тогда можно разместить в заголовках строк и столбцов два измерения), либо показываются несколько мер (и тогда одну из осей таблицы займут названия мер, а другую – значения единственного «неразрезанного» измерения).

Наиболее значимыми критериями при создании кубов являются режим хранения данных и уровень их агрегирования. В многомерных базах хранятся агрегатные данные различной подробности, например, выручка за день, месяц, год. Целью хранения этих данных является сокращение времени выполнения запросов для анализа и прогнозов. При создании многомерной базы вычисляются некоторые агрегатные данные, однако зачастую аналитикам интересны суммарные данные, поэтому хранение всевозможных вариантов агрегатных данных в большинстве случаев не обосновано. Проблема в том, что при добавлении новых измерений объем данных куба растет экспоненциально. От количества измерений куба и членов измерений на различных уровнях иерархий этих измерений зависит степень

роста объема агрегатных данных. Например, для куба с тремя измерениями необходимо создать три сводные таблицы, а для четырех измерений количество таких таблиц возрастет до шести.

Данные, из которых создается куб, могут храниться в реляционных или в многомерных структурах. В MSSQL сервере используются три способа для хранения данных:

1. MOLAP (Multidimensional OLAP) – исходные и агрегатные данные хранятся в многомерной базе данных. Этот режим предполагает больше места, чем остальные, но предоставляет максимальную скорость при обработке запросов.

2. ROLAP (Relational OLAP) – исходные данные находятся в реляционной базе данных, где они изначально содержались. Агрегатные данные также находятся в той же базе данных, только перемещаются в специально организованные для их хранения служебные таблицы.

3. HOLAP (Hybrid OLAP) – агрегатные данные хранятся в многомерной базе данных, а исходные данные не перемещаются из исходной реляционной базы данных, то есть остаются там, где были изначально.

Для аналитики в MSSQLServer используются различные службы. SQL ServerDatabaseEngine – высокопроизводительная СУБД, без которой невозможно создание хранилища данных, может хранить большие объемы данных, необходимых для построения хранилища. SQL Server Integration Services – среда для извлечения, преобразования и загрузки данных, которая помогает заполнить хранилище и обеспечить его синхронизацию с данными из обычной реляционной базы данных. SQL Server Analysis Services играет ключевую роль для аналитики, так как обеспечивает возможность построения OLAP-решений, вытекающих из анализа OLAP-кубов.

В заключение отметим, что OLAP-сервер имеет удобные средства доступа, просмотра и анализа данных для аналитика. Пользователь получает простую, понятную модель данных, организованную в виде многомерных кубов.

Основное назначение OLAP –реализация систем, направленных на динамический анализ данных, тенденций и прогнозов.

OLAP-кубы являются обязательной составляющей программ анализа финансового и экономического состояния предприятий для результативного управления крупным и средним бизнесом.

Библиографический список

1. *Грофф Д., Вайнберг П.* SQL: полное руководство: пер. с англ. М.: Вильямс, 2015. 960 с.
2. *Петкович, Д.* MicrosoftSQLServer 2012. Руководство для начинающих / пер. с англ. СПб.: БХВ-Петербург, 2013. 816 с.

УДК 004

А. М. Бочарова

студент кафедры компьютерных технологий и программной инженерии

Д. В. Богданов

канд. техн. наук, доцент – научный руководитель

ИСПОЛЬЗОВАНИЕ И ПОЛУЧЕНИЕ ГЕОПРОСТРАНСТВЕННЫХ ДАННЫХ В СИСТЕМАХ АЭРОНАВИГАЦИОННОЙ ИНФОРМАЦИИ

При переходе к современной концепции управление аэронавигационной информацией выходит на передний план и становятся все более острыми проблемы качества и вероятность искажения геопространственных данных на этапах их получения, обработки и передачи, а также наличие большого количества разнородных форматов и соответственно процедур конвертации. В статье рассмотрены методы решения возникающих проблем путем использования стандартизированного формата для обмена и хранения геопространственных данных.

Геопространственными характеристиками обладают точки, используемые при построении траектории при полете по зональной навигации, а также объекты, расположенные в зоне аэродрома, радионавигационные объекты воздушных трасс. Такие объекты, как навигационные средства, препятствие, взлетно-посадочная полоса / рулежная дорожка / линия наведения / место стоянки, аэропорт, вертодром, маршрут, воздушное пространство и другие, обладающие геопространственными характеристиками и представляющими интерес с точки зрения аэронавигационной информации. Зональная навигация – это метод навигации, позволяющий воздушным судам выполнять полёт по любой желаемой траектории в пределах зоны действия радиомаячных навигационных средств или в пределах, определяемых возможностями автономных средств.[1]

Для получения необходимых геопространственных данных об этих объектах в данный момент используются различные методы, к которым относятся:

- обычная геодезическая съемка;
- аэрофотосъемка;
- спутниковая съемка и зондирование;
- лазерная локация.

Данные методы различны по способу получения данных, точности, разрешающей способности и результирующему формату представления данных.

Рассмотрим типичную цепочку прохождения геопространственной составляющей аэронавигационных данных от момента их появления до их использования потребителями.

Первым этапом жизненного цикла геопространственных данных является их появление в памяти комплекса приборов, при помощи которого осуществляется получение данных. После окончания работ данные из этого комплекса импортируются при помощи прилагающегося ПО, поставляемого производителем, на внешних носителях в виде файлов различных форматов. После этого полученные данные сводятся в специальные отчеты для отправки их на экспертизу. В отчетах содержится информация об условиях, времени проведения съемки, используемом оборудовании. Сами данные сводятся в таблицы, прикладываемые к отчету.

Далее, если данные удовлетворяют необходимым требованиям, они могут быть предоставлены заказчику в виде «сырых» данных, содержащихся в файле, либо, в наиболее распространенном случае, в виде текстовых таблиц, содержащихся в отчете. Данные таблицы лично составляются оператором вручную. Дальнейший способ использования этих данных, способ их распространения, конвертация в другой формат зависят от самого заказчика, однако широкое распространение имеет передача в бумажном виде. При необходимости, и если данные удовлетворяют требованиям по качеству, эти данные могут быть опубликованы в сборниках AIP. Из этих сборников данные попадают к промежуточным компаниям-интеграторам. На этом этапе может проводиться ввод этих данных в какую-либо ин-

формационную тему, базу данных вручную при помощи соответствующих интерфейсов. Далее данные передаются конечным потребителям: бортовые системы, системы проектирования процедур, информационные системы и т.д. [2].

Какие же проблемы можно выделить в данной цепочке? Первой проблемой является как минимум один, а в худшем случае и несколько переносов данных вручную оператором. Каждое появление такого элемента в цепочке движения данных отрицательно сказывается на качестве и увеличивает вероятность искажения за счёт человеческого фактора. Еще одной ключевой проблемой является наличие в цепочке большого количества разнородных форматов и соответственно процедур конвертации. Разные форматы могут основываться на разных технологических базах (иногда уже морально устаревших), ориентированных на разную обработку данных и различаться по своему составу. Из этого следует, что даже при автоматизированной конвертации, помимо вероятности возникновения ошибки (пусть и относительно небольшой по сравнению с ручной конвертацией), существует еще и вероятность потери качественных характеристик, а именно полноты и разрешающей способности.

При переходе к современной концепции управления аэронавигационной информацией данные проблемы выходят на передний план и становятся всё более острыми, так как имеют непосредственное влияние на параметры качества данных.

Ключом к решению возникающих проблем может послужить использование стандартизированного формата для обмена и хранения геопространственных данных. В качестве такого формата предлагается использовать GML – язык географической разметки.

На данный момент формат проходит фазу активного внедрения, с каждым днем его поддерживает всё больше систем геопространственного отображения и обмена, что открывает широкие перспективы для интеграции с ними.

GML – является грамматической системой xml, написанной с помощью xml схемы для моделирования, передачи и хранения географической информации. GML обеспечивает множество видов объектов для того, чтобы описать географию включая особенности, координатные справочные системы, геометрию, топологию, время, единицы измерения и обобщенные значения [3].

Рассмотрим подробнее, в чём же причина такого успеха и преимущества GML:

- открытость и кроссплатформенность. GML разрабатывался при участии всех сторон рынка обмена геопространственной информацией (поставщики, потребители, разработчики геоинформационных систем). Основан на XML, который поддерживается на всех современных платформах;
- детальная проработанность. Разработка стандарта длилась в течение нескольких лет, учитывались все основные существующие технологии и тенденции. GML позволяет описать геометрию объектов различной сложности;
- унифицированность. Является стандартом ISO;
- интегрируемость. Структура формата позволяет легкую интеграцию с приложениями и различными форматами обмена, например, AIXM;
- гибкость. Реализует механизм «профилей», позволяющий использовать в конкретном приложении, которое реализует GML, только необходимую её часть.

В GML представлен набор так называемых примитивов – базовых элементов геометрии, на основе которых потом можно описать более сложную геометрию. Примитивы определены для 1-мерных объектов (точка); 2-мерных (геометрия на плоскости) и 3-мерных объектов (геометрия в пространстве). Помимо этого, в формате определены механизмы их комбинирования для создания сложных объектов. Использование профилей позволяет гибко настраивать и комбинировать использование примитивов GML.

Использование подобной модели позволяет накладывать метеорологические данные на карты с аэронавигационными векторными объектами и генерировать географически точные 3D-изображения.

Еще одной привлекательной особенностью GML является то, что он не ориентирован на какую-либо конкретную систему координат. Система координат по умолчанию также не ориентирована, поэтому при любом обмене информацией система координат указывается явно. Это позволяет использовать как международные системы координат, такие, например, как WGS, так и отечественную ПЗ-90.02, которая используется для обеспечения работы системы ГЛОНАСС. Хранение и передача координат

возможна только в одной системе координат, однако процедуры преобразования между основными системами известны, их реализация не является проблемой. Так, формула преобразования координат из wgs-84 в пз-90.02 проведена в ГОСТе.

Библиографический список

1. *Воев В. И., Липин А. В., Сарайский Ю. Н.* Занальная навигация. СПб., 2004. 127 с.
2. <http://www.studfiles.ru/preview/4672088/> (дата обращения: 15.05.2017)
3. Научная сессия ТУСУР–2010: Материалы докладов Всероссийской научно-технической конференции студентов, аспирантов и молодых ученых, Томск, 4–7 мая 2010 г. – Томск: В-Спектр, 2010. Ч. 2. – 374 с

УДК 004.04

С.В. Бубликов

студент кафедры компьютерных технологий и программной инженерии

П.А. Степанов

старший преподаватель – научный руководитель

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИЙ BIG DATA ПРИ ВЫПОЛНЕНИИ РЕТРОСПЕКТИВНОГО АНАЛИЗА ВАЛЮТНОГО РЫНКА

Любой из видов рынков, что фондовый, что валютный, что рынок драгоценных металлов обеспечивают перераспределение денежного капитала в те сектора экономики, где капитал принесет наибольшую выгоду [1]. Исходя из этого, основными инструментами движения экономики являются финансовые рынки и трейдинг.

Для эффективной торговли требуются инструменты анализа. Существующие инструменты анализа имеют большой процент неточности, так как данных становится все больше как структурированных, так и неструктурированных. Но наиболее важной задачей, для каждого участника рынка является именно точность прогнозов дальнейшего движения торговых инструментов.

Помочь в анализе могут BigData технологии. Цель Big Data заключается в анализе очень больших объемов данных, для обеспечения более точной и понятной человеку информации. Наиболее точно Big Data можно охарактеризовать как технологию и методы производительной обработки очень больших объемов данных, в том числе неструктурированных, в распределенных информационных системах, обеспечивающих организацию качественно новой полезной информации [2].

С помощью технологий BigData можно создать систему для анализа поведения трейдеров, использующую данные, которые берутся с форумов и комментариев, и разделяет их на две группы, называемые медвежьим и бычьим движениями, то есть о понижении и повышении цены соответственно). Например, комментарий к новости о повышении ставки содержанием «Неплохо» будет отнесен к бычьему движению, а сообщение на форуме «Мне кажется фунт будет в течение недели падать» будет воспринят как сигнал о понижении цены фунта по отношению к доллару.

Реализация разделения комментариев на два класса таких, как положительные и отрицательные, осуществляется при использовании метода опорных векторов, так как использование данного метода дает наиболее точные результаты по классификации, чем другие машинные методы. [3] Идея метода заключается в том, что даны точки на плоскости из обучающей выборки, разбитые на два класса, а сам метод опорных векторов найдет наиболее лучшую гиперплоскость, разделяющую эти два класса. Для реализации простой классификации с двумя параметрами используется линейная функция, а соответственно гиперплоскостью тогда будет изображена прямая. Лучшей гиперплоскостью для метода опорных векторов является та, что расположена максимально равноудаленно от векторов обоих классов. Далее, все новые точки (не из обучающей выборки) автоматически классифицируются таким образом, что точка, которая находится выше прямой, попадает в первый класс, а точка ниже прямой попадает во второй класс. Метод опорных векторов является наиболее известным алгоритмом машинного обучения и технологий Big Data [3].

На рис. 1 показана классификация данных методом опорных векторов. Прямая, по условиям метода, проходит так, чтобы расстояние между прямой и векторами обоих классов обучающей выборки было минимальным.

Такая система для обработки данных технического и фундаментального анализа, может собирать информацию с новостей, аналитических данных, индикаторов, свечного и графического анализа. Обработка новостей и аналитики обрабатывает данные структурированные и неструктурированные. Неструктурированные данные обрабатывается по принципу анализа поведения трейдеров. Задаются флаги и по ним фильтруется информация. Например, при выходе новости о том, что в Великобритании прошел референдум о выходе из Евросоюза, то система при обработке данных сообщит о понижении

стоимости фунта по отношению к доллару. Так же, например, если индикатор показывает возможное повышение стоимости какой-либо валюты или индекса, то система учтет эти показания при анализе валюты в целом. Индикаторы делятся на несколько групп: основные, «агрессивные» и консервативные.

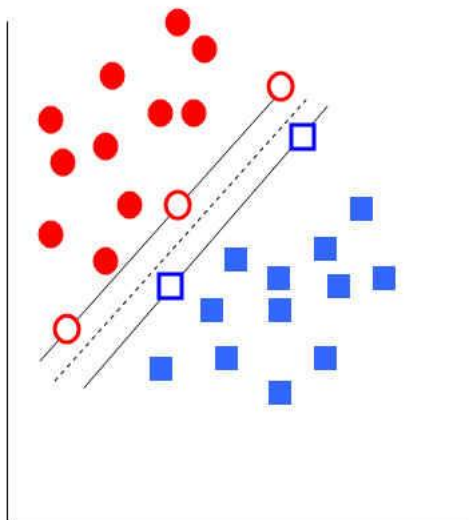


Рис. 1. Метод опорных векторов

Свечной и графический анализы работают по такому же принципу, что и новостной. Храня все вышеперечисленные данные, можно точно сказать, какое значение было у индикатора «Stochastic», например, в 12:45 3 марта 2016 года, соответственно можно представить картину дальнейшего движения валюты, используя точные данные по всем индикаторам, не зная, как в дальнейшем двигалась валюта.

Для определения направления открытия сделки используется алгоритм MapReduce[2]. На Map-шаге рабочие узлы получают задание на предварительную обработку данных. На Reduce-шаге рабочие ноды осуществляют свертку результатов обработки после Map-шага. После выполнения этого шага головной узел получает итоговый результат. На входе используются данные, такие как человеческий фактор, то есть мнения людей, данные с индикаторов и графических паттернов, период времени анализа. На выходе выводятся сгруппированные данные, разбитые по категориям и дальнейшее поведение актива.

На рис. 2 изображена схема работы системы. Входными данными являются наименование инструмента, в данном примере это стоимость фунта стерлингов по отношению к американскому доллару, и значение временного периода инструмента, например 1 час. После выполнения алгоритма MapReduce, который использует данные анализа мнения трейдеров, индикаторов и новостей, выходными данными становится заключение о дальнейшем движении инструмента.

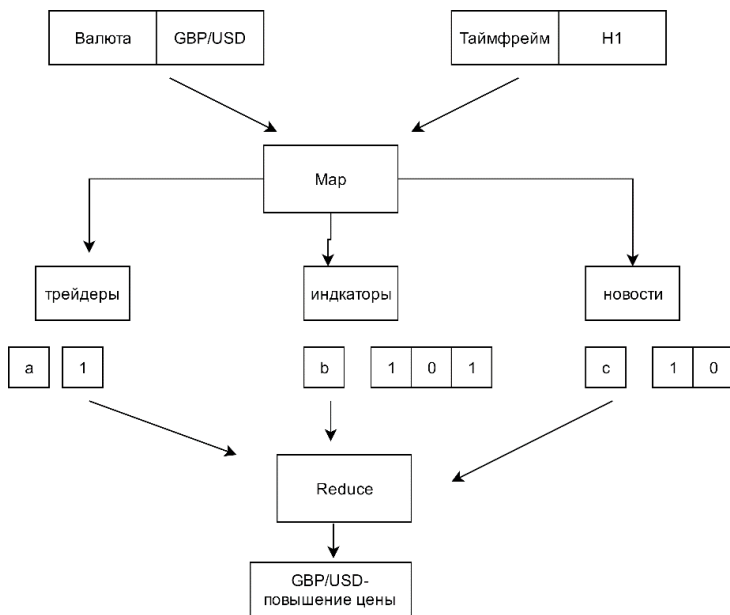


Рис. 2. Пример работы системы

Данный процесс можно автоматизировать. Используя данные, полученные на выходе, программа, основываясь на истории и заложенных алгоритмах действий, при той или иной ситуации будет автоматически осуществлять торговые операции, что существенно облегчит работу трейдера.

На первом этапе работ будет реализован функционал, связанный с анализом индикаторов и свечных моделей при помощи технологий Big Data

Заключение

Система обработки данных с валютного рынка включает в себя постоянно растущие данные, полученные от трейдеров и новостных источников, а количество индикаторов, свечных и графических паттернов насчитывает более 500. Система имеет возможность обработки данных на разных периодах времени, начиная с 15 минут, заканчивая дневными графиками.

В итоге, с методами обработки больших данных получается система, которая определяет направление валюты и индексов с точностью до 60-75%.

Библиографический список

1. Рубен Лл. Работа мировых рынков. Управление финансовой инфраструктурой / М.: Альпина Паблишер, 2016. 536 с.
2. Силен Д., Мейсман А., Али М. Основы Data Science и Big Data. СПб.: Питер, 2017. 336 с.
3. Национальный Открытый Университет Интуит. URL: <http://www.intuit.ru/studies/courses/6>
4. Вьюгин В. Математические основы машинного обучения и прогнозирования. М.: МЦМНО, 2014. 304 с.

УДК 004.9

Д.В. Валяев, А.А. Герцев, О.Н. Максимова, А.С. Топчий

магистранты кафедры вычислительных систем и сетей

А.В. Никитин

канд. техн. наук, доцент – научный руководитель

РАЗРАБОТКА ПРОГРАММНОЙ СРЕДЫ ДЛЯ РЕАЛИЗАЦИИ ПРОЕКТОВ С ИСПОЛЬЗОВАНИЕМ ШЛЕМОВ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ HTC VIVE И OCULUS RIFT

Разработка приложений, поддерживающих возможность работы со шлемами виртуальной реальности, особенно такими, как Oculus Rift и HTC Vive, является на данный момент очень перспективным направлением, однако, материалов на эту тему очень мало, особенно на русском языке.

Следовательно, перед нами встала задача создания подробных русскоязычных руководств по настройке и использованию данных шлемов, а также подключению их к проектам на Unity 3D. Выбор движка в данном случае был обусловлен его крайней востребованностью, широким распространением и наличием соответствующих SDK. Кроме того, Unity относительно прост в изучении, поэтому хорошо подходит для студенческих и учебных проектов [1].

Но, прежде чем перейти непосредственно к проблемам, связанным с реализацией программных проектов с использованием данных шлемов, необходимо кратко охарактеризовать само оборудование.

Oculus Rift – очки виртуальной реальности, созданные компанией Oculus VR. CV1 («Consumer version 1») – «Пользовательская версия 1», то есть последняя из существующих на данный момент и лучшая по характеристикам версия Oculus. Данный шлем поставляется в комплекте с сенсором, позволяющим отслеживать его положения и осуществлять навигацию в виртуальном пространстве, а также геймпадом Xbox One [2][3].

HTC Vive – шлем виртуальной реальности совместной разработки Valve Corporation и HTC – является частью программы Steam VR. В комплекте с данным шлемом идут две базовые, то есть отслеживающие, станции и два контроллера. Таким образом, для данного шлема становятся доступны как сидячий режим, так и навигация в пределах виртуальной комнаты, а также взаимодействие с объектами при помощи контроллеров [4][5].

Для подключения Oculus Rift к проекту на Unity мы использовали два различных метода подключения: при помощи включения поддержки устройств виртуальной реальности в самом Unity [6] и с использованием Oculus Utilities [7]. Первый способ намного проще, так как требует выбрать в настройках проекта в Unity пункт Virtual Reality Supported. После этого достаточно добавить в сцену стандартный префаб FPSController, представляющий собой персонаж с видом от первого лица. При запуске игры при подключенном шлеме изображение начинает передаваться в шлем, при этом становится доступно управление с клавиатуры (можно перемещаться по сцене, прыгать) и отслеживание положения головы (пользователь может осмотреться по сторонам). Однако, в данном случае движения игрового персонажа получаются слишком резкими, что может стать причиной возникновения тошноты и головокружения у игрока.

Второй вариант предполагает использование Utilities for Unity от Oculus. Данный пакет включает в себя интерфейс для доступа к различным устройствам виртуальной реальности, специально спроектированную для VR камеру и контроллер персонажа, расширенные функции рендеринга, средства отладки и многое другое. Подключив его, остаётся добавить в сцену контроллер персонажа OVRPlayerController. Для него реализованы система плавных движений, коллайдер, система меню со стереорендерингом текста и прицела. Этот контроллер гораздо лучше подходит для использования с устройством виртуальной реальности, чем стандартный контроллер персонажа от первого лица, поскольку его движения медленнее и плавнее, а также он более естественно реагирует на повороты головы.

Для подключения к проекту шлема HTC Vive использовался SteamVR Plugin [8]. Он представляет собой интерфейс для работы со шлемами виртуальной реальности и предоставляет доступ к отслеживаемым контроллерам, сопровождению, рендерингу моделей для отслеживаемых устройств и синхронизации частоты обновления физики с системой рендера.

Компоновщик SteamVR позволяет просматривать контент в виртуальной реальности, используя игровой режим Unity, оставляя обычное окно игры в качестве сопутствующего экрана на главном мониторе.

После установки данного плагина в проект необходимо добавить в сцену объект CameraRig, который отвечает за контроль шлема. Однако, в нашем случае его возможностей оказалось недостаточно. Для того, чтобы контроллеры Vive смогли взаимодействовать с окружением и в сцене стала возможна телепортация, оказалось необходимо внести некоторые изменения в сцену (а именно – пометить некоторые поверхности как доступные для телепортации) и скрипты на C#, реализующие данный функционал при помощи Steam VR API [9].

Итогом нашей работы стала подготовка подробных инструкций по настройке и использованию описанных выше шлемов, а также по подключению их к проектам на Unity. Они оформлены в виде лабораторного практикума, предназначенного для магистрантов первого года обучения направления подготовки «Информатика и вычислительная техника». Кроме того, были созданы условия для создания студенческих проектов и магистерских диссертаций с использованием данных шлемов.

Библиографический список

1. Unity – Game Engine. – Unity, 2017 [Электронный ресурс]. URL: <https://unity3d.com/ru/> (дата обращения: 01.03.2017)
2. Oculus Rift | Oculus – Oculus VR, LLC, 2017 [Электронный ресурс]. URL: <https://www.oculus.com/rift/> (дата обращения: 01.03.2017)
3. Изучение шлема виртуальной реальности Oculus Rift CV1. Инструкция по настройке и основам работы» / Д.В. Валяев, А.А. Герцев, Д.В. Казаков, О.Н. Максимова, А.С. Топчий // Лабораторный практикум «Системы виртуальной и дополненной реальности», часть 1, ГУАП, 2017 [Электронный ресурс]. URL: <http://fs.guap.ru/labvr/OculusRift.pdf> (дата обращения 13.05.2017)
4. VIVE™ Россия – HTC, 2017 [Электронный ресурс]. URL: <https://www.vive.com/ru/> (дата обращения: 01.03.2017)
5. Изучение шлема виртуальной реальности HTC Vive. Инструкция по настройке и основам работы» / Д.В. Валяев, А.А. Герцев, Д.В. Казаков, О.Н. Максимова, А.С. Топчий // Лабораторный практикум «Системы виртуальной и дополненной реальности», часть 1, ГУАП, 2017 [Электронный ресурс]. URL: <http://fs.guap.ru/labvr/HTCVive.pdf> (дата обращения 01.03.2017)
6. Unity User Manual (5.5). Virtual Reality – Unity, 2017 [Электронный ресурс]. URL: <https://docs.unity3d.com/Manual/VirtualReality.html> (дата обращения: 01.03.2017)
7. Oculus Utilities for Unity – 2016, Oculus VR, LLC [Электронный ресурс]. URL: <https://developer3.oculus.com/documentation/game-engines/latest/concepts/unity-utilities-overview/> (дата обращения: 01.03.2017)
8. SteamVR Plugin – Asset Store, Unity Technologies – 2017 [Электронный ресурс]. URL: <https://www.assetstore.unity3d.com/en/#//content/32647> (дата обращения: 01.03.2017)
9. SteamVR – Valve Developer Community, Valve – 2017 [Электронный ресурс]. URL: <https://developer.valvesoftware.com/wiki/SteamVR> (дата обращения: 17.01.2017)

УДК 519.687.2
ГРНТИ50.41.00

В. С. Васильева

студентка

А. Г. Степанов

научный руководитель

МОДЕЛИРОВАНИЕ РАБОТЫ ДИСПЕТЧЕРА ЗАДАЧ НА ЯЗЫКЕ VBA

Сгенерированы наборы тестовых заданий с известной реакцией программы диспетчера задач. Созданы предпосылки автоматизации процесса тестирования разрабатываемого программного средства. Изложен метод подготовки к реализации концепций гибкой методики неполного жизненного цикла.

Ключевые слова: язык VBA, диспетчер задач, тестирование, диспетчеризация процессов, дескриптор задачи, дисциплина SRT.

Известные формализованные подходы к разработке программного обеспечения базируются на использование моделей жизненного цикла [1]. В то же время в работе [2] отмечаются недостатки применения таких моделей и предлагаются так называемые гибкие методики разработки. В этом случае «программный продукт выпускается короткими итерациями, каждая из которых заканчивается определенной сборочной версией, которая поступает в эксплуатацию, а также используется заказчиком для уточнения требований и предварительного тестирования» [2, с. 66]. В то же время разработка тестов представляет собой самостоятельную задачу, требует повышенного внимания от разработчика и больших трудозатрат. Возникает противоречие, связанное с необходимостью распределения средств и сил разработчика между разработкой программного обеспечения, тестов и собственно тестирования с целью сокращения времени на выполнение очередной итерации. Оно может быть преодолено за счет создания программной модели разрабатываемого средства на другой доступной платформе с целью скорейшей выдачи программистам наборов тестовых заданий. В настоящей работе описывается такая технология на примере разработки макета диспетчера задач для операционной системы.

В общем смысле диспетчер задач – компьютерная программа (утилита) для вывода на экран списка запущенных процессов и потребляемых ими ресурсов (в частности статус, процессорное время и потребляемая оперативная память) [3]. В качестве дополнительных функций диспетчер задач может предложить возможность завершить один из процессов или присвоить ему другой приоритет. В некоторых операционных системах эта же самая программа позволяет наблюдать загруженность процессора, оперативной памяти, сетевых подключений и других ресурсов [4].

Диспетчеризация процессов (задач) [3] – это определение очередности получения процессора или процессоров задачами, находящимися в состоянии готовности с целью их выполнения. Реализация диспетчеризации представляет собой перевод в состояние выполнения. Диспетчеризация может выполняться многократно. Это происходит в силу того, что в каждый такт процессорного времени на одном процессоре могут выполняться команды только одной задачи. Диспетчеризация предполагает создание и модификацию очереди готовых к выполнению задач. Элементами такой очереди являются дескрипторы задач.

Дескриптор задачи [3] – это специальная информационная структура, в которой хранятся характеристики задачи, необходимые для целей управления со стороны операционной системы. Информация о задаче, которая хранится в дескрипторе, разделяется на несколько групп. Различается информация:

- по идентификации задачи;
- о ресурсах, которые необходимы задаче для её выполнения и о ресурсах, которые используются в настоящее время;

- о текущем состоянии задачи;
- о родственных связях задачи;
- информация, необходимая для целей планирования и диспетчеризации.

Будем рассматривать дисциплину диспетчеризации [3] как некоторое основное правило, реализующее очередность предоставления (выделения) процессора (процессорного времени) готовым к выполнению задачам. В рассматриваемом случае выбрана дисциплина SRT [3]. SRT (shortest remaining time) – раньше процессор получает та задача, которая имеет меньше всего времени для своего завершения. Это время определяется как разность между заказанным временем обслуживания и тем процессорным временем, которая задача уже получила. Эта дисциплина является «невытесняющей» (рис. 1), то есть в ней нет прерываний задач, а также дискриминационной по отношению к «длинным» процессам. Для подготовки тестов разрабатываемого программного изделия использовалась встроенная среда разработки VBA в Excel. Был создан общий вид диспетчера, рассчитанный на 6 задач, а результаты его работы использовались для создания тестовых заданий.



Рис. 1. Дисциплины диспетчеризации

Алгоритм программы моделирования представлен на рис. 2.

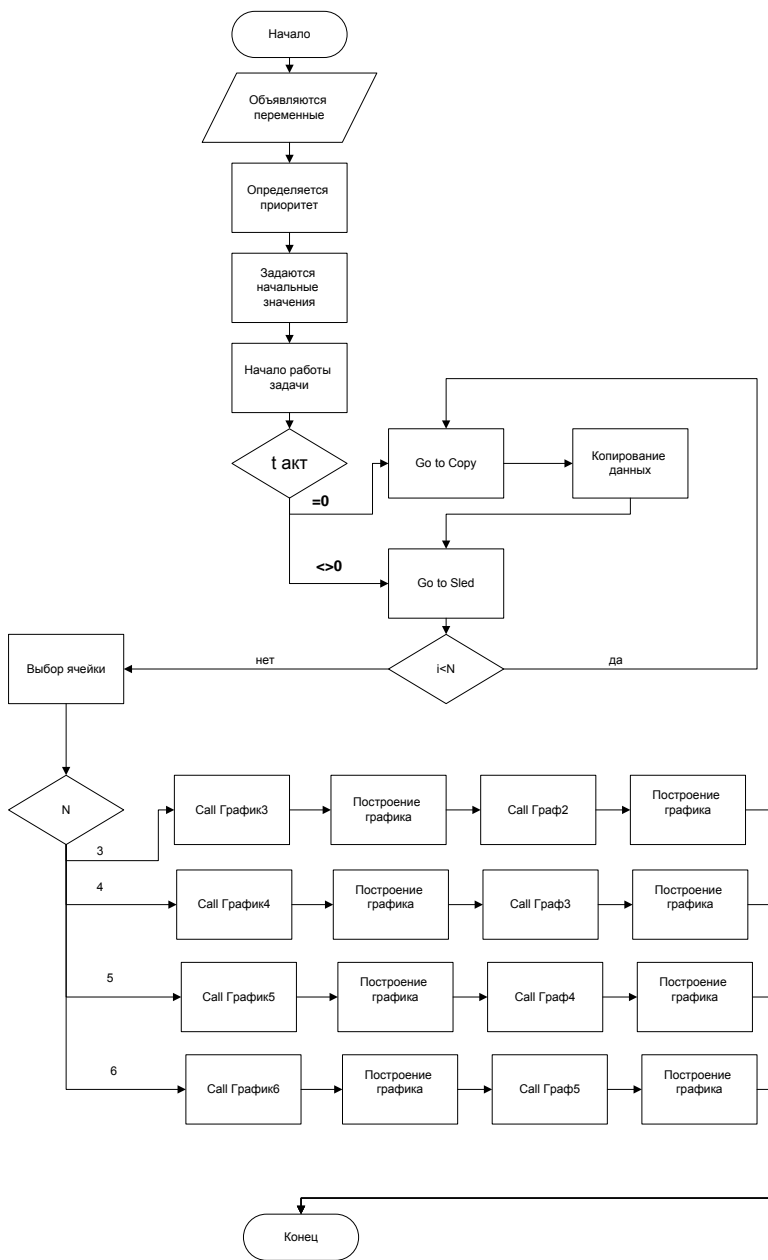


Рис. 2. Алгоритм программы

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ

Примеры результатов работы модели диспетчера задач, основанной на выбранной дисциплине диспетчеризации, представлены в табл. 1.

Таблица 1

Состояние программы до и после запуска диспетчера

До запуска программы					После запуска программы				
Имя	t акт.	t cпу	t обл.		Имя	t акт.	t cпу	t обл.	
A	0	4	4		A	0	4	4	
B	2	4	4		C	2	4	3	
C	2	4	3		B	2	4	4	
D	3	4	3		E	3	4	2	
E	3	4	2		D	3	4	3	
F	4	4	3		F	4	4	3	

Имя	начало	конец	t cпу		Имя	начало	конец	t cпу	
	0	0		1 1	A	0	4	4	1 1
	0	0		1 1	C	4	8	4	1 1
	0	0		1 1	B	8	12	4	1 1
	0	0		1 1		12	12		1 1
	0	0		1 1		12	12		1 1
	0	0		1 1		12	12		1 1

Ожидание						
Имя	t акт.	конец	конец	конец	конец	конец
0		0				
0		0				
0		0	0			
0		0	0	0		
0		0	0	0	0	
0		0	0	0	0	0

Ожидание						
Имя	t акт.	конец	конец	конец	конец	конец
A	0	0				
C	2	4				
B	2	4	8			
0		4	8	12		
0		4	8	12	12	
0		4	8	12	12	12

До запуска программы

t cпу

До запуска программы

t rot

Созданная модель дает возможность генерировать наборы тестовых заданий с известной реакцией программы диспетчера, что, как следствие, позволит автоматизировать процесс тестирования

итерации разрабатываемого программного средства. Применение изложенного метода ко всему набору задач, входящему в разрабатываемый пакет, позволит, в конечном итоге, на практике реализовать концепцию гибкой методики неполного жизненного цикла [2].

Библиографический список

1. *Голосовский М. С.* Информационно-логическая модель процесса разработки программного обеспечения // Программные системы и вычислительные методы. 2015. № 1. С. 59-68.
2. *Добрынин А. С., Койнов Р. С., Кулаков С. М.* Модель неполного жизненного цикла программного обеспечения. Вестник Астраханского государственного технического университета. Сер. управление, вычислительная техника и информатика. 2015. № 2. С. 65-70.
3. *Востриков А. А., Кучин Н.В.* Основы организации операционных систем: учеб. пособие. СПб.: ГУАП, 2011.
4. *Мартышкин А. И.* Исследование диспетчеров задач многопроцессорных систем на моделях массового обслуживания // XXI век: итоги прошлого и проблемы настоящего плюс. № 5(09), 2012. С. 139-145.
5. *Колисниченко Д.Н.* Самоучитель работы на компьютере. М., СПб.: АСТ, Прайм-Еврознак, 2008.

УДК 621.391:004.934.2

А. Н. Величко

магистрантка кафедры компьютерных технологий и программной инженерии

Б. В. Соколов

д-р техн. наук, профессор – научный руководитель

А. А. Карпов

д-р техн. наук, доцент – научный со-руководитель

В. Ю. Будков

канд. техн. наук, доцент – научный консультант

КРАТКИЙ ОБЗОР МЕТОДОВ, ПРИМЕНЯЕМЫХ ПРИ ПАРАЛИНГВИСТИЧЕСКОМ АНАЛИЗЕ РЕЧИ

Компьютерная паралингвистика изучает такие невербальные аспекты человеческой коммуникации и речи как: естественные эмоции, интонации, особенности произношения, параметры голоса диктора, истинность речевых высказываний и т.д. Актуальной задачей является автоматическое выявление истинности или ложности сообщений. Многие современные исследования посвящены разработке математического и программного обеспечения для автоматизированных систем распознавания лжи в речи человека.

На настоящий момент большинство исследований на тему определения лжи рассматривают только визуальное ее проявление, а именно: мимику, жесты, биометрические параметры. Все эти явления можно распознать при исследованиях с использованием полиграфа. Однако, несмотря на популярность данного метода, он не является оптимальным, поскольку требует особых условий для работы с аппаратом, которые распространяются как на место исследований, так и на испытуемого. Именно по этой причине появился интерес к бесконтактным методам, исследующим речевую деятельность и невербальные сигналы.

Современные системы паралингвистического анализа речи используют пространства признаков огромного размера (низкоуровневые описатели – Low Level Descriptors, LLD) для интегрального описания целых фраз, а не отдельных слогов и фонем. Эти наборы могут включать в себя частоту основного тона (ЧОТ), форманты (резонансные частоты голосового тракта), мел-частотные кепстральные коэффициенты (Mel-Frequency Cepstral Coefficients, MFCC), модулированный спектр сигнала, коэффициенты перцептивного линейного предсказания (Relative Spectral Transform – Perceptual Linear Prediction, RASTA-PLP), энергетические признаки сигнала и их вариативность (джиттер и шиммер – jitter, shimmer) и т.д. MFCC и RASTA-PLP-признаки довольно давно применяются в области автоматического распознавания речи и привнесены в распознавание паралингвистических явлений из нее, также как и частотные признаки речи LSF (Line Spectral Frequency) [1]. Также используется высокоуровневый набор элементов речи, включающий в себя четыре группы: гласные, фонемы, псевдослоги и паузы.

Наиболее распространенными методами моделирования и классификации паралингвистических явлений являются: искусственные нейронные сети (Artificial Neural Networks, ANN), векторы Фишера (Fisher Vectors, FV), смеси гауссовых распределений (Gaussian Mixture Model, GMM), регрессия опорных векторов (Support Vector Regression, SVR), скрытые марковские модели (Hidden Markov Models, HMM), модель экстремального машинного обучения (Extreme Learning Machines, ELM), метод частичных наименьших квадратов (Partial Least Squares, PLS), последовательная минимальная оптимизация (Sequential Minimal Optimization, SMO), метод k ближайших соседей (k-nearest neighbors algorithm, k-NN) случайный лес (Random Forest), бэггинг (Bagging, Bootstrap Aggregating), деревья принятия решений (Decision Trees), среднеквадратическое отклонение (Standard Deviation), шенноновский критерий минимума требуемой избыточности (МИ).

Искусственная нейронная сеть – это математическая модель упрощенной биологической нейронной сети мозга. Является частным случаем методов распознавания образов, дискриминантного анализа, методов кластеризации и т.п. На данный момент существует большое количество типов ис-

кусственных нейронных сетей, однако в сфере распознавания речи чаще используются глубокие нейронные сети (Deep Neural Networks, DNN), свёрточные нейронные сети (Convolutional Neural Networks, CNN) и рекуррентные нейронные сети (Recurrent Neural Networks, RNN).

Скрытая марковская модель является статистической моделью, которая имитирует работу процесса, схожего с марковским процессом с неизвестными параметрами, где стоит задача предсказания неизвестных параметров на основе наблюдаемых. Марковский процесс подразумевает процесс, эволюция которого после любого заданного значения временного параметра t не зависит от эволюции, которая предшествовала t , при условии, что значение процесса в данный момент фиксировано.

Векторы Фишера – это метод кодирования низкоуровневых описателей. Данный метод популярен в сфере компьютерного зрения, особенно в крупномасштабном поиске изображений (Large-Scale Image Retrieval). Преимущество векторов Фишера в том, что они требуют намного меньше компонентов в модели смеси гауссовых распределений, чем модель «мешка слов» (Bag-Of-Words) и не нуждаются в обучении на очень большом корпусе. Он является частным и улучшенным случаем ядерной функции Фишера [2].

Смеси гауссовых распределений являются статистической моделью для представления нормально распределенных субпопуляций внутри общей популяции. Смеси распределений позволяют строить или приближать сложные вероятностные распределения, также с их помощью можно решать задачу разбиения заданной выборки объектов или ситуаций на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались (кластеризация данных). Задача кластеризации относится к широкому классу задач обучения без учителя.

Регрессия опорных векторов – это задача регрессии, решаемая методом опорных векторов. В регрессионной машине опорных векторов задачей является оценка функциональной зависимости зависимой переменной y на множестве независимых переменных x . Машина опорных векторов решает задачи классификации и регрессии с помощью построения нелинейной плоскости, разделяющей решения. Метод основан на концепции гиперплоскостей, которые определяют границы гиперповерхностей (гиперповерхность является обобщением трехмерной поверхности для случая евклидова пространства произвольной размерности). Разделяющая гиперплоскость – это гиперплоскость, которая отделяет группу объектов, имеющих различную классовую принадлежность. Гиперплоскость – это подпространство с размерностью, на единицу меньшей, чем объемлющее пространство. Гиперплоскость делит пространство соответствующей размерности на два полупространства, все точки каждого из которых определяются неравенствами [3].

Модель экстремального машинного обучения является искусственной нейронной сетью прямого распространения (Feed Forward Neural Network, FF or FFNN) с одним скрытым слоем, на котором веса и смещения нейронов инициализируются случайным образом. Используется для задач классификации или регрессии. Здесь под сетью прямого распространения имеется ввиду искусственная нейронная сеть, в которой информация движется только в одном направлении – вперед, от входа к выходу.

Метод частичных наименьших квадратов является методом для предсказания, альтернативным обычной регрессии наименьших квадратов. Объединяет в себе свойства метода главных компонент и множественной регрессии. В начале метод выделяет набор скрытых факторов, которые описывают как можно больше ковариации между зависимыми и независимыми переменными. На шаге регрессии предсказываются значения зависимых переменных с использованием декомпозиции независимых переменных [4].

Последовательная минимальная оптимизация – является одним из алгоритмов для решения задачи квадратичного программирования (через поиск двух множителей Лагранжа). Заключается в том, что выделяются пары переменных α для пары базовых векторов, наиболее близко примыкающих к границе классов и лежащих по разные стороны от границы. Для пары переменных с ограничениями решение задачи квадратичного программирования строится аналитически в явной форме. После того, как пройден шаг оптимизации одна (или обе) переменная обращается в ноль, а соответствующие вектора исключаются из списка кандидатов. Алгоритм завершается, когда становится невозможным нахождение пары векторов, оптимизация параметров которых может улучшить решение [5].

Метод к ближайших соседей является простейшим метрическим классификатором. Суть метода в том, что объект присваивается тому классу, которому принадлежат ближайшие к нему объекты (т.е. соседи) обучающей выборки. Однако, алгоритм принимает во внимание не только количество определенных классов, но и удаленность от нового значения.

Вышеописанные методы имеют разные достоинства и недостатки, но являются популярными в данной области. В рамках международной конференции INTERSPEECH Computational Paralinguistic Challenge (ComParE) с 2009 года ежегодно проходят соревнования, посвященные различным направлениям исследований в области компьютерной паралингвистики. В 2016 году на этих соревнованиях впервые появились такие темы, как: распознавание лжи в речи, распознавание степени искренности человека, идентификация родного языка диктора по его англоязычной речи. Всем участникам предоставлялась возможность использовать собственные алгоритмы и наборы признаков помимо тех, что были представлены организаторами. Для обучения и тестирования алгоритмов предоставлялись аудиозаписи по темам соревнований. В соревнованиях по детекции лжи в речи призовые места заняли работы [6, 7, 8] которые при начальной планке качества распознавания UAR=68.3% (Unweighted Average Recall – среднее значение полноты), установленной организаторами соревнований, показали результаты 74,9%, 69,3%, 68,9% соответственно. Из этих результатов можно сделать вывод, что наиболее качественные результаты показывают системы, использующие следующие методы: низкоуровневые и высокоуровневые описатели, просодические и эмоциональные признаки, смеси гауссовых распределений, векторы Фишера, метод экстремального машинного обучения, метод частичных наименьших квадратов, машина опорных векторов, метод к ближайших соседей, последовательная минимальная оптимизация.

Библиографический список

1. Карпов А. А., Кайа Х., Салах А. А. Актуальные задачи и достижения систем паралингвистического анализа речи // Научно-технический вестник информационных технологий, механики и оптики. 2016. Т. 16. № 4. С. 581–592. doi:10.17586/2226-1494-2016-16-4-581-592.
2. Sanchez J., Perronnin F., Mensink T., Verbeek J. Image Classification with the Fisher Vector: Theory and Practice. [Research Report] RR-8209, INRIA. 2013. <hal-00779493v3>.
3. Воронцов К. В. Лекции по методу опорных векторов. 21 декабря 2007 г., URL: <http://www.ccas.ru/voron/download/SVM.pdf>.
4. IBM Knowledge Center, Partial Least Squares Regression, URL: https://www.ibm.com/support/knowledgecenter/en/SSLVMB_22.0.0/com.ibm.spss.statistics.help/spss/base/idh_idd_pls_variables.htm#idh_idd_pls_variables.
5. Терехов С. А. Технологические аспекты обучения нейросетевых машин // Научная сессия МИФИ-2006. VIII Всероссийская научно-техническая конференция «Нейроинформатика-2006»: лекции по нейроинформатике. – М.: МИФИ, 2006. С. 13-72.
6. Montacé C., Caraty M.-J. Prosodic Cues and Answer Type Detection for the Deception Sub-Challenge // In Proc. INTERSPEECH-2016. San Francisco, USA. 2016. pp. 2016-2020
7. Herms R. Prediction of Deception and Sincerity from Speech using Automatic Phone Recognition-based Features. In Proc. INTERSPEECH-2016. San Francisco, USA, 2016. pp. 2036-2040.
8. Amiriparian S., Pohjalainen J., Marchi E., Pugachevskiy S., Schuller B. Is deception emotional? An emotion-driven predictive approach // In Proc. INTERSPEECH-2016. San Francisco, USA. 2016. pp. 2011-2015.

УДК 519.876.5

А.Ф. Виноградов, А.С. Клеченов

студенты кафедры проблемно-ориентированных вычислительных комплексов

Е.П. Виноградова

старший преподаватель – научный руководитель

МОДЕЛИРОВАНИЕ ЦИФРОВЫХ МЕТОДОВ ОБРАБОТКИ ЗВУКА В СРЕДЕ MATHCAD

Значительную роль в развитии цифровой обработки звуковых сигналов играет математическое моделирование.

В настоящее время существует целый ряд готовых программных изделий, позволяющих реализовать те или иные алгоритмы обработки звука. Этот путь, безусловно, является наиболее простым. Однако, он не позволяет раскрыть математическую сущность алгоритмов, лежащих в основе преобразования сигналов. Поэтому кажется наиболее привлекательной идея реализации математических моделей цифровой обработки звука с помощью средств компьютерной математики и моделирования. Такие возможности предоставляет, в частности, пакет MathCad [2].

Выбор средства реализации математических моделей алгоритмов обработки звуковых сигналов обоснован, в первую очередь, наглядностью и простотой работы со звуковыми сэмплами. Так, для загрузки звукового сигнала в формате *.wav матрицу используется оператор READWAV, а для записи результатов обработки – WRITEWAV [2]. В работе используются одноканальные звуковые файлы (моно). Однако, полученные результаты могут быть распространены и на стереодорожки. Использование среды MathCad позволяет также вывести результаты обработки на экран в виде графиков или диаграмм. Это дает возможность сопоставить звуковые файлы с изображениями, им соответствующими. В частности, простой однонаправленный звук во временной области представляет собой синусоидальное колебание с частотой, соответствующей высоте звука. Чем выше частота колебания, тем выше соответствующий звук.

На рис. 1 изображен фрагмент программы, позволяющий проиллюстрировать то, как «звучит синусоида».

$$j := 0..199999$$

$$S1_j := \cos(0.03 \cdot j) \cdot (2 \times 10^4) \quad S2_j := \cos(0.05 \cdot j) \cdot (2 \times 10^4)$$

$$\text{WRITEWAV}(\text{"C:\s1.wav"}, 4 \times 10^4, 16) := S1$$

$$\text{WRITEWAV}(\text{"C:\s2.wav"}, 4 \times 10^4, 16) := S2$$

$$S3 = \frac{S1 + S2}{2}$$

$$\text{WRITEWAV}(\text{"C:\s1-2.wav"}, 4 \times 10^4, 16) := S3$$

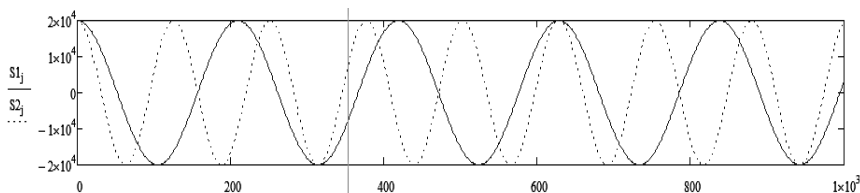


Рис. 1. Звучащая синусоида

Достаточно большое количество алгоритмов обработки звуковых сигналов связано с понятием «модуляция». В частности, эффект модуляции используется при моделировании характерного звучания отдельных музыкальных инструментов. Использование амплитудной модуляции приводит к возникновению эффекта «тремоло», характерного для такого инструмента, как электроорган.

На рис. 2 приводится структурная схема устройства для реализации данного эффекта и приводится математическое описание алгоритма.

$$y(n) = [1 + m \cdot m(n)] \cdot x(n)$$

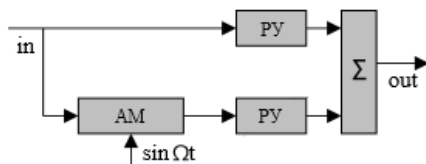


Рис. 2. Структурная схема для реализации эффекта «тремоло»

Программа, иллюстрирующая моделирование эффекта, приводится на рис. 3.

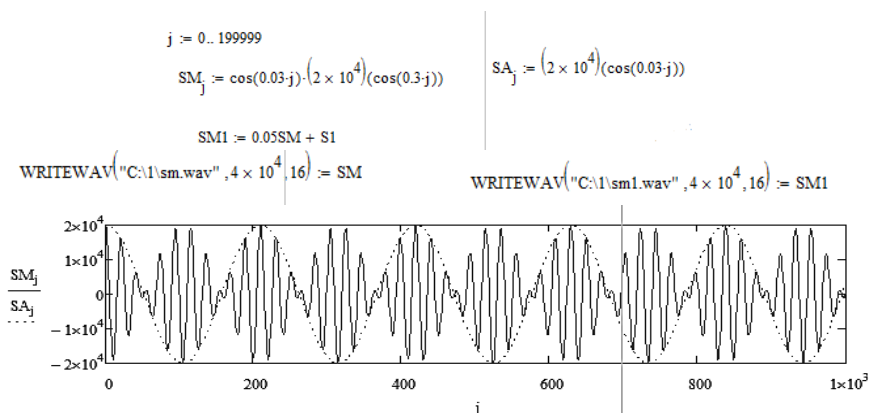


Рис. 3. Моделирование эффекта «тремоло»

Фазовая и частотная модуляция также довольно часто используется для реализации различных звуковых эффектов. В частности, для создания тембра звучания деревянных духовых инструментов (гобой) применяется эффект «вibrато», основанный на эффекте фазовой модуляции [1].

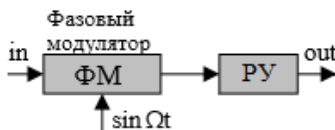


Рис. 4. Структурная схема для реализации эффекта «вibrато»

В дискретно-временной области модулированное колебание представляет собой выражение:

$$x(n) = A_m \cdot \sin(2 \cdot \pi \cdot F \cdot n + I \cdot \sin(2 \cdot \pi \cdot F_{md} \cdot n))$$

Ниже приводится реализация эффекта в среде MathCad.

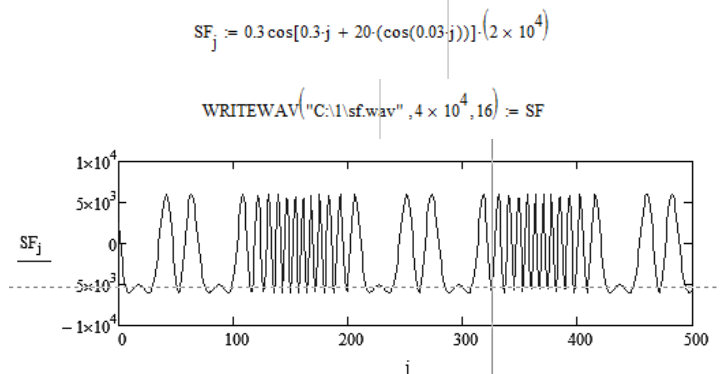


Рис. 5. Моделирование эффекта «вибрато»

Если объединить вместе амплитудную и фазовую модуляции, то можно получить эффект, близкий к эффекту «вау-вау».

$$SLOW_j := 0.3 \cos \left[0.3 \cdot \left(j - 150 \cdot \frac{j}{199999} \right) + 20 \cdot (\cos(0.03 \cdot j)) \right] \cdot (2 \times 10^4)$$

$$\text{WRITEWAV}("C:\1\sf1.wav", 4 \times 10^4, 16) := SLOW$$

Рис. 6. Моделирование эффекта «вау-вау»

Кроме приведенных выше простейших эффектов для придания звуковым дорожкам естественного эффекта нередко применяется такой эффект, как эффект «эхо». Для реализации подобного эффекта обычно используются линии задержки. Структура устройств, реализующих подобный эффект, приведена на рис. 10 [1].

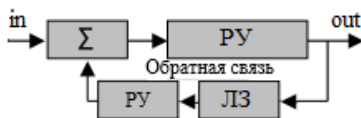


Рис. 7. Структурная схема для реализации эффекта «эхо»

Применение одной или нескольких линий задержки в совокупности с обратными связями позволяет микшировать исходный сигнал с его ослабленной и смещенной во времени копией, что создает иллюзию эха. Обратная связь создает бесконечное число повторов звука, следующих с интервалом, равным времени задержки. Однако в реальности за счет эффекта затухания звука, число воспринимаемых ухом повторений обычно невелико. Поэтому при моделировании работы подобного устройства, для иллюстрации можно ограничиться тремя повторениями.

На рис. 8 приводится программная реализация эффекта «эхо» в среде MathCad.

$$k := 0..199999$$

$$SEH1_k := \begin{cases} S1_k & \text{if } k \leq 20000 \\ \left[S1_k \cdot e^{-0.00005 \cdot (k-20000)} \right] & \text{otherwise} \end{cases}$$

$$k := 0..199999$$

$$SEH1_k := \begin{cases} SEH_k & \text{if } k \leq 50000 \\ SEH_{k-50000} \cdot 0.3 + SEH_k & \text{if } 50000 < k \leq 100000 \\ \left(SEH_{k-100000} \cdot 0.1 + SEH_{k-50000} \cdot 0.3 + SEH_k \right) & \text{otherwise} \end{cases}$$

$$\text{WRITEWAV}\left("C:\1\seh1.wav", 4 \times 10^4, 16\right) := SEH1$$

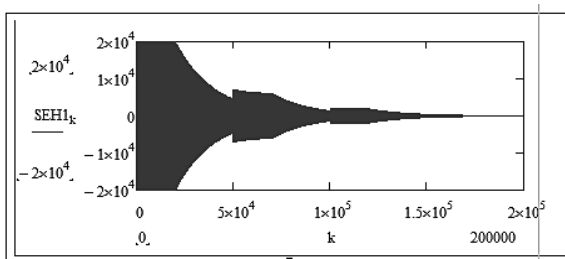


Рис. 8. Моделирование эффекта «эхо»

Выводы:

Возможность обработки сигнала в среде MathCad имеет большие преимущества над имеющимися известными методами. Во-первых, самым важным является возможность самому создавать любой эффект, который поддается математическому описанию. Во-вторых, важным фактором является то, что есть возможность настраивать тот или иной эффект до мельчайших деталей. При обработке звукового сигнала мы, помимо всего прочего, получаем также возможность визуально оценить полученный звук на предмет наличия, например, искажений, которые могли возникнуть в ходе записи. Единственным существенным минусом является то, что обрабатываемый файл должен быть с расширением «.wav», что на данный момент сильно ограничивает его функциональность, так как это расширение является не самым популярным в музыкальной среде (явно уступая, например, «.mp3»). Тем не менее, программа обладает огромным потенциалом, а при должном уровне знаний может являться незаменимым помощником в сфере обработки звуковых сигналов.

Библиографический список

1. Вологдин Э.И. Методы и алгоритмы обработки звуковых сигналов. СПб.: КОРОНА, 2012. 96 с.
2. Макаров Е.Г. Инженерные расчеты в Mathcad. СПб.: Питер, 2015. 448 с.

УДК 629.735.33

Л.В. Гашева

магистрант кафедры компьютерных технологий и программной инженерии

В.И. Сеньченков

д-р техн. наук, профессор – научный руководитель

АЛГОРИТМИЧЕСКАЯ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ РЕГРЕССИОННОГО АНАЛИЗА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

Методы параметрического статистического анализа широко используются как в различных областях современной науки, так и на производстве. Они находят своё практическое применения в таких видах деятельности, как медицина, экономика, сельское хозяйство, животноводство, и др.

Одним из основных методов параметрической статистики является регрессионный анализ, позволяющий в условиях стохастической зависимости исследуемой величины от неслучайных или случайных переменных определять данную зависимость.

В настоящее время он реализован в сложных математических пакетах, таких как MATLAB, STATISTICA, MATHCAD, EXCEL и др. Но данные средства являются тяжело доступными, дорогостоящими. Кроме того, в них заложено большое разнообразие функционала, что усложняет данные программные средства и требует значительного времени для усвоения работы с ними.

Также в интернете существует множество сервисов, помогающих проводить регрессионный анализ онлайн, но в них он реализован лишь частично, чаще всего такие сервисы позволяют провести только однофакторный регрессионный анализ.

Таким образом, необходим программный продукт, в котором был бы реализован весь функционал регрессионного анализа, имелся понятный интерфейс и который был бы прост в установке и сопровождении.

Регрессионный анализ является одним из наиболее применимых методов параметрической статистики. Его сущность состоит в замене стохастической зависимости между переменными \hat{y} и $\hat{x}_j, j = \overline{1, k}$ некоторой детерминированной зависимостью f , достаточно хорошо аппроксимирующей основные свойства исходной зависимости. Иначе говоря, в процессе регрессионного анализа устанавливается зависимость между некоторой характеристикой случайной величины \hat{y} и независимыми переменными $\hat{x}_j, j = \overline{1, k}$.

Существует две модели регрессионного анализа:

– модель, в которой зависимая переменная – случайная величина, а независимые переменные являются неслучайными, точно заданными:

$$\hat{y} = \hat{f}(x_1, x_2, \dots, x_m) + \hat{\varepsilon}, \quad (1)$$

где \hat{y} – зависимая переменная, $x_j, j = \overline{1, k}$ – независимые переменные, $\hat{\varepsilon}$ – ошибка при наблюдении;

– модель, в которой как зависимые, так и независимые переменные являются случайными величинами:

$$\hat{y} = \hat{f}(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m) + \hat{\varepsilon}, \quad (2)$$

где \hat{y} – зависимая переменная, $x_j, j = \overline{1, k}$ – независимые переменные, $\hat{\varepsilon}$ – ошибка при наблюдении.

Можно выделить основные этапы регрессионного анализа:

– выбор класса функций, в рамках которого определяется взаимосвязь между \hat{y} и $\hat{x}_j, j = \overline{1, k}$.

Класс функций определяется пользователем, экспертом в предметной области, данный выбор не входит в вычислительные задачи алгоритма;

- формирование множества информативных факторов, которое может являться избыточным;
- вычисление оценок параметров, определенных в рамках выбранного класса функций;
- проверка адекватности полученной зависимости экспериментальным данным (генеральной совокупности);
- проверка значимости факторов. На данном этапе устраняется избыточность факторов, выбранных на втором этапе.

Рассмотрим метод построения уравнений зависимости между переменными \hat{y} и $\hat{x}_j, j = \overline{1, k}$. Если случайная величина подчинена нормальному закону распределения, то уравнение регрессии в общем виде можно записать следующим образом:

$$y = f(x_j; a_0, a_1, \dots, a_k), \quad (3)$$

где y – зависимая переменная, $x_j, j = \overline{1, k}$ – независимые переменные, a_0, a_1, \dots, a_k – коэффициенты уравнения. В данном случае для определения коэффициентов уравнения (3) используется метод наименьших квадратов (МНК). Он основывается на выполнении экстремального условия, согласно которому сумма квадратов отклонений экспериментальных значений зависимой переменной от теоретических должна быть минимально возможной:

$$V^* = \min_{a_j \in \mathbb{R}} \left\{ \sum_{i=1}^n (y_i - f(x_i; a_0, a_1, \dots, a_k))^2 \right\}, j = 0, 1, \dots, k. \quad (4)$$

Геометрический смысл условия (4) представлен на рис. 1. Известные данные зависимой и независимой переменной, полученные в результате проведения эксперимента отмечаются точками. Каждая точка представляет собой пару значений y_i и x_i . Если длину отрезков между отмеченными точками и значениями функций обозначить за e_i , то согласно МНК необходимо подобрать функцию $y=f(x; a_0, a_1, \dots, a_k)$ таким образом, чтобы сумма квадратов отклонений e_i экспериментальных данных от значений полученной функции в точках x_i была минимальна.

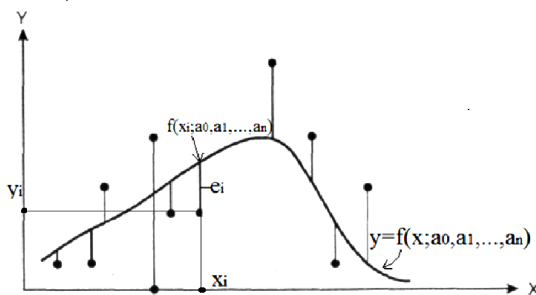


Рис. 1. Геометрический смысл МНК и МНМ

Тогда экстремальное условие (4) можно записать в следующем виде:

$$\sum_{i=1}^n e_i^2 \rightarrow \min. \quad (5)$$

Если же случайная величина подчиняется закону распределения Лапласа, то для определения коэффициентов уравнения регрессии применяется метод наименьших модулей (МНМ), согласно которому сумма модулей отклонений экспериментальных значений зависимой переменной от теоретических должна быть минимальной. Данное условие имеет вид

$$R^* = \min_{a_j \in \mathbb{R}} \left\{ \sum_{i=1}^n |y_i - \varphi(x_i; a_0, a_1, \dots, a_k)| \right\}, j = 0, 1, \dots, k. \quad (6)$$

Умножим слева обе части уравнения (15) на обратную матрицу $(F_{[k+1, n]}^T F_{[n, k+1]})^{-1}$:

$$(F_{[k+1, n]}^T F_{[n, k+1]})^{-1} (F_{[k+1, n]}^T F_{[n, k+1]}) A_{<k+1>} = (F_{[k+1, n]}^T F_{[n, k+1]})^{-1} F_{[k+1, n]}^T Y_{<n>}. \quad (16)$$

Если учесть, что в правой части уравнения (16) образуется единичная матрица, получаем формулу для вычисления векторов коэффициентов уравнения:

$$A_{<k+1>} = (F_{[k+1, n]}^T F_{[n, k+1]})^{-1} F_{[k+1, n]}^T Y_{<n>}. \quad (17)$$

На данный момент в ходе выполнения научной работы реализован однофакторный линейный и полиномиальный регрессионный анализ.

Модель линейной функции:

$$y = \tilde{b}_1 x + \tilde{b}_0 + \varepsilon, \quad (18)$$

Формулы для расчетов коэффициентов уравнения (18), записанные в скалярной форме:

$$\tilde{b}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad (19)$$

$$\tilde{b}_0 = \bar{y} - \tilde{b}_1 \bar{x},$$

где $\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$, $\bar{y} = \frac{\sum_{i=1}^n y_i}{n}$

Модель полинома k-ой степени:

$$y = \sum_{i=0}^k \tilde{b}_i x^{k-i} + \varepsilon, \quad (20)$$

Формулы для расчетов коэффициентов уравнения (20), записанные в матричной форме:

$$B_{<k+1>} = (F_{[k+1, n]}^T F_{[n, k+1]})^{-1} F_{[k+1, n]}^T Y_{<n>}, \quad (21)$$

где

$$B_{<k+1>} = (\tilde{b}_0, \tilde{b}_1, \dots, \tilde{b}_k)^T;$$

$$F = \begin{pmatrix} \mathbf{x}_1^k & \mathbf{x}_1^{k-1} & \dots & \mathbf{1} \\ \mathbf{x}_2^k & \mathbf{x}_2^{k-1} & \dots & \mathbf{1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_n^k & \mathbf{x}_n^{k-1} & \dots & \mathbf{1} \end{pmatrix}.$$

Стоит отметить, что за основу при реализации алгоритма нахождения линейной зависимости можно также использовать матричную форму записи (21), т.к. из общего выражения (20) можно получить выражение для полинома первой степени, что является частным случаем выражения. Но такая реализация будет требовать больше вычислительных ресурсов, как временных, так и физических.

Таким образом, имея некоторую выборку взаимосвязанных данных, как технических, так и экономических или медицинских, можно построить уравнения, с некоторой степенью адекватности описывающие зависимости между входными переменными и откликом. Это позволяет использовать разрабатываемый программный продукт в любых областях деятельности.

Библиографический список

1. Дрейнер Н., Смит Г. Прикладной регрессионный анализ. 3-е изд. / пер. с англ. – М.-СПб. – Киев: Диалектика, 2016. – 912 с.
2. Сеньченков В.И. Статистические методы обработки экспериментальных данных. СПб.: ГУАП, 2006. – 244 с.
3. Гмурман В.Е. Теория вероятностей и математическая статистика. М.: Высшая школа, 2004.

УДК 004.032.26

Г. В. Гетманенко

магистрант кафедры вычислительных систем и сетей

Н. В. Соловьев

канд. техн. наук, доцент – научный руководитель

ПРИМЕНЕНИЕ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧАХ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ

Нейронные сети – один из подходов к машинному обучению распознаванию образов, основы которого зародились в 1943 году, еще до появления термина «искусственный интеллект». Нейронная сеть представляет собой достаточно грубую модель мозга, состоящую из множества слоев связанных между собой искусственных нейронов. Искусственный нейрон содержит сумматор и блок активации, на который подается сумма взвешенных значений входов нейрона.

Классические нейронные сети имеют несколько полносвязных слоев (рис. 1). На входной слой $X_1 \dots X_n$ подается вектор входных значений, а выходной слой вырабатывает вектор $Y_1 \dots Y_m$, при правильном обучении сети соответствующий распознанному образу. В последнее время для обучения многослойных полносвязных нейронных сетей наиболее часто применяется метод обратного распространения ошибки. При большом количестве слоев нейронов и большой размерности вектора входных значений обучение сети значительно замедляется, из-за больших затрат вычислительных ресурсов. Это характерно для обучения сети распознаванию растровых изображений, при котором элементы входного вектора представляют собой яркости пикселей изображения. Большая изменчивость изображений объектов, относящихся к одному распознаваемому классу также существенно увеличивает время обучения и снижает достоверность классификации.

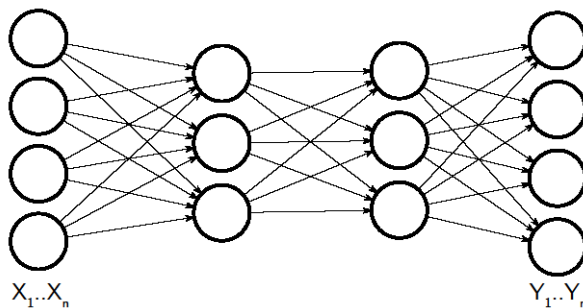


Рис. 1. Пример классической нейронной сети с двумя скрытыми полносвязными слоями

Сверточным нейронным сетям в меньшей степени присущи эти недостатки. Сверточная нейронная сеть (англ. convolutional neural network, CNN) – архитектура нейронных сетей, основанная на функции свертки [1]. CNN (рис. 2) представляет собой чередующиеся сверточные слои (англ. convolutional layers), выполняющие свертку предыдущего слоя (рис. 3) и субдискретизирующих слоев (англ. subsampling layers, слой подвыборки), уменьшающих размерность предыдущего слоя (рис. 4). Также в сети присутствуют полносвязные слои. После последнего слоя сверточной нейронной сети могут быть один или несколько полносвязных слоев.

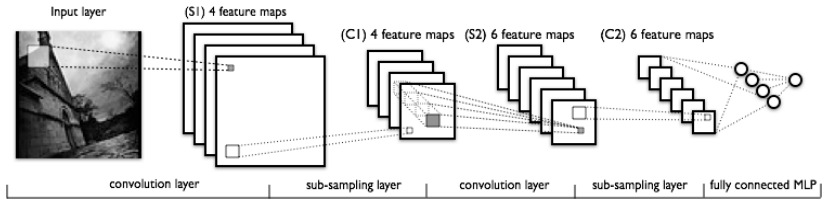
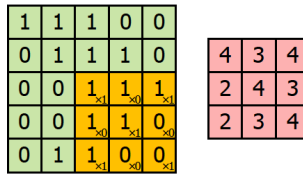


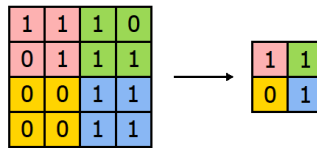
Рис. 2. Топология сверточной нейронной сети



Изображение Свертка

Умножение на коэффициенты ядра свертки изображено красным $\times 1$, $\times 0$.

Рис. 3. Свертка (convolution)



Изображение Выборка

Выборка максимального с фильтром 2×2 и сдвигом 2

Рис. 4. Субдискретизация (subsampling)

Ключевым моментом в понимании сверточных нейронных сетей является понятие так называемых «разделяемых весов», т.е. часть нейронов некоторого рассматриваемого слоя нейронной сети может использовать одни и те же весовые коэффициенты [2]. Нейроны, использующие одни и те же веса, объединяются в карты признаков (англ. feature maps), а каждый нейрон карты признаков связан с частью нейронов предыдущего слоя.

Сверточные нейронные сети имеют следующие преимущества:

- меньшее количество вычислений,
- относительная устойчивость к смещению и масштабу объектов на обрабатываемом изображении,
- учитывается двумерная структура изображений.

Сверточные нейронные сети имеют множество параметров, которые можно изменять при инициализации сети:

- количество слоев,
- размерность ядра свертки для каждого из слоев,
- количество ядер свертки,
- шаг сдвига маски слоев субдискретизации,

- степень уменьшения ими размерности,
- функции по уменьшению размерности (выборка максимума, среднего),
- передаточная функция нейронов,
- наличие и параметры входной полносвязной нейросети на выходе сверточной.

Все эти параметры существенно влияют на результат. Существует несколько выверенных и прекрасно работающих конфигураций, но математического объяснения и рекомендаций, по которым нужно строить сеть для новой задачи, на данный момент не существует.

Задача классификации изображений – это приём начального изображения и вывод его класса (кошка, собака и т.д.) или группы вероятных классов, которая лучше всего характеризует изображение. Обучение нейронной сети методом обратного распространения ошибки (англ. backpropagation) делится на четыре этапа: прямое распространение, вычисление функцию ошибки (англ. loss function, функция потери), обратное распространение и обновление веса.

Этапы повторяются до тех пор, пока не будет достигнута достаточная точность в распознавании. При прямом распространении входное изображение проходит через все слои нейронной сети. После этого вычисляется ошибка распознавания. Функция ошибки может быть разной. Как правило, используется среднеквадратическая ошибка:

$$E = \frac{1}{K} \sum_{i=1}^K (target_i - output_i)^2,$$

где K – количество классов, $target$ – ожидаемое значение; $output$ – значение на выходе сети.

Метод обратного распространения ошибки аналогичен методу градиентного спуска для минимизации функций. Задача обучения сводится к минимизации функции ошибки, то есть необходимо откорректировать веса так, чтобы ошибка была минимальной. Существует несколько методов для корректировки весов [3]. Они имеют схожую идею, но дают несколько различные результаты. Методы корректировки весов:

- Stochastic Gradient Descent,
- SGD with momentum,
- SGD with Nesterov momentum,
- AdaGrad, AdaDelta, AdaMax,
- RMSProp,
- RProp,
- NAG.

Немаловажную роль играет такой параметр как скорость обучения (рис. 5). При низком значении параметра сеть будет обучаться довольно долго, а при достаточно высоких сеть вовсе не будет обучаться.

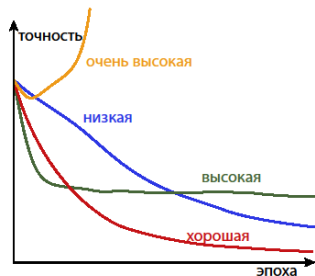


Рис. 5. Скорость обучения

В процессе обучения может возникать такое явление, как переобучение [4]. Переобучение – это излишне точное соответствие нейронной сети конкретному набору обучающих примеров, при котором сеть теряет способность к обобщению. Для борьбы с переобучением применяются методы регуляризации.

Сверточным нейронным сетям нашли применение в информационных технологиях, метеорологии, медицине. Однако относительно точные рекомендации для построения таких сетей отсутствуют. Таким образом стоит задача исследования влияния начальных параметров сверточной нейронной сети на качество и скорость распознавания на различных обучающих множествах.

Библиографический список

1. Глубокое обучение для новичков: распознаем изображения с помощью сверточных нейронных сетей // Блог компании Wunder Fund // Хабрахабр URL: <https://habrahabr.ru/company/wunderfund/blog/314872/> // Дата публикации: 14.11.2016.
2. Deconvolutional Neural Network // Хабрахабр URL: <https://habrahabr.ru/company/nordavind/> // Дата публикации: 24.03.2015.
3. Optimization for Deep Networks // Ishan Misra // URL: http://www.cs.cmu.edu/~imisra/data/Optimization_2015_11_11.pdf
4. Preventing Overfitting in Neural Networks // Nikhil Buduma // URL: <http://www.kdnuggets.com/2015/04/preventing-overfitting-neural-networks.html>

УДК 004.771

Д.В. Горелик

студент кафедры вычислительных систем и сетей

А.В. Гордеев

д-р техн. наук, профессор – научный руководитель

ИСПОЛЬЗОВАНИЕ КОНТЕЙНЕРОВ ДЛЯ ПОСТРОЕНИЯ СИСТЕМЫ РАБОТЫ С УДАЛЕННЫМИ ВИРТУАЛЬНЫМИ МАШИНАМИ

В процессе работы над созданием интерактивного учебного модуля по дисциплине «Корпоративные сети со службой каталога» возник ряд задач по организации эффективной и надежной работы одновременно нескольких пользователей со своими виртуальными машинами.

Виртуализация – это технология запуска одной операционной системы поверх другой [2]. В 2001 году компания VMware выпустила серверный продукт для виртуализации на основе гипервизора, привлечший внимание корпоративных заказчиков. Практически в то же самое время компания Parallels представила решение для контейнерной виртуализации Virtuozzo, завоевавшее признание у пользователей.

Гипервизор работает таким образом: операционная система хоста эмулирует аппаратное обеспечение, поверх которого уже запускаются гостевые операционные системы. Это означает, что взаимосвязь между гостевой и хостовой операционными системами следует правилу: все, что делает аппаратная часть, должно быть доступно гостевой операционной системе со стороны хостовой [2].

Контейнеры, напротив – это виртуализация на уровне операционной системы, то есть каждая гостевая операционная система использует то же самое ядро, что и хостовая. Это дает контейнерам большое преимущество: они меньше и компактнее гипервизорных гостевых сред, поскольку у них с хостом на много больше общего.

Главным отличием между гипервизорной и контейнерной виртуализациями является среда, в которой работает система. Если запуск осуществляется в виде контейнера, то приложение попадает в ту же операционную среду, в которой работают и остальные приложения, с которыми происходит взаимодействие. Если запуск осуществляется в виде виртуальной машины на сервере, то любые запросы, совершенные в её сторону со стороны контактирующих приложений, проходят длинный путь из одной операционной среды в другую и обратно.

Переход на использование программных контейнеров и виртуальных машин призван решить главную задачу: выделить программный объект из базовой среды, чтобы вести его обработку в особых, оптимальных для него условиях. Но если контейнеры позволяют отделить объект от системной программной оболочки, работающей на сервере, то виртуальная машина служит для совершенно иных целей: она подменяет аппаратные возможности их виртуальными абстракциями, чтобы сделать процесс обработки независимым от реальных физических особенностей используемого оборудования.

Такое разделение продолжалось почти 12 лет: гипервизорной виртуализации не удавалось завоевать сколько-нибудь значимый кусок рынка, а контейнеры не могли проникнуть в корпоративный сегмент. Переломным стал 2013 год, когда новому разработчику Docker удалось привлечь внимание представителей бизнеса к преимуществам контейнерной технологии виртуализации [2].

Docker – это открытая платформа для разработки, доставки и эксплуатации приложений [1]. В своем ядре он позволяет запускать практически любое приложение, безопасно изолированное в контейнере. Безопасная изоляция позволяет запускать на одном хосте много контейнеров одновременно. Легковесная природа контейнера, который запускается без дополнительной нагрузки гипервизора, позволяет экономить аппаратные ресурсы.

В процессе изучения этих технологий виртуализации и программных продуктов, позволяющих их использовать были определены два пути, по которым можно работать с виртуальными машинами на сервере:

1. Использование непосредственно VirtualBox в операционной системе сервера. При использовании такого подхода возникает необходимость в решении вопросов: создания системы аутентификации и авторизации для пользователей, чтобы они не имели доступ к виртуальным машинам друг друга. А также необходима система, которая позволяла бы при нештатных ситуациях отключения одного из серверов, продолжить работу на другом без потери данных.

2. Использование Docker для обеспечения каждого пользователя контейнером. А уже после этого запускать Virtual-Box непосредственно в контейнере. В отличие от первого способа, это позволит сразу изолировать пользователей друг от друга, так как контейнеры являются полностью изолированными окружениями. А также при непредвиденной работе позволяет восстановить работу контейнера с момента, на котором произошел сбой, не потеряв при этом никаких данных.

Для определения – в каком направлении двигаться, необходимо провести небольшое исследование, которое на практике покажет достоинства и недостатки этих двух подходов и выявит оптимальный для создаваемой системы. Поэтому мною разработан следующий алгоритм исследования:

1. Так как в моем распоряжении имеется два абсолютно идентичных сервера, то на первом сервере была установлена программа Oracle Virtual-Box, а на втором – программное обеспечение Docker.

2. Для мониторинга нагрузки на аппаратную часть серверов был написан специальный скрипт, который с момента его запуска записывает в отдельный файл время, нагрузку на CPU в % и занятость оперативной памяти в %.

3. Для осуществления испытаний был избран следующий метод: так как к наиболее ресурсозатратным действием в Virtual-Box можно отнести установку операционной системы на виртуальную машину, поскольку при этом интенсивно загружаются и центральный процессор виртуальной машины, и его внешние устройства, то был создан специальный образ устанавливаемой операционной системы. В качестве устанавливаемой системы была выбрана Windows 7, и была подготовлена процедура автоматической установки без какого-либо дополнительного вмешательства пользователя.

4. На первом сервере была создана виртуальная машина, запущен скрипт документирования нагрузки, и затем запущена виртуальная машина, а в ней осуществлялась автоматизированная установка операционной системы. Полученные результаты были обработаны, и на основе полученных данных построен график зависимости нагрузки от времени (рис. 1).

5. На втором сервере был запущен контейнер с аналогичной серверу операционной системой вместе с программой Virtual-Box. Далее в программе VirtualBox создана виртуальная машина, запущен скрипт документирования нагрузки в основной операционной системе, и затем запущена виртуальная машина в контейнере. И тоже была запущена установка операционной системы. Полученные результаты были обработаны, и на основе полученных данных построен второй график зависимости нагрузки от времени (рис. 2).

Для более полной оценки было проведено исследование, которое заключается в увеличении количества виртуальных машин и контейнеров с VirtualBox. Алгоритм испытания аналогичен предыдущему, за исключением:

1) Количество виртуальных машин на сервере увеличено до 3, они полностью аналогичны друг другу и на них одновременно проводится установка операционной системы Windows 7;

2) Количество контейнеров увеличено до 3. В каждом из них одновременно запущена своя Virtual-Box с одной виртуальной машиной.

Следует обратить внимание, что на обоих рисунках левая шкала отображает загрузку центрального процессора, а правая – использование оперативной памяти.

По полученным графикам можно сделать следующие выводы:

1. Виртуальные машины в стандартном гипервизорном окружении работают быстрее, но потребляют больше аппаратных ресурсов. У одной виртуальной машины на установку операционной системы уходит порядка 550 секунд, в то время как у контейнера с VirtualBox уходит приблизительно 700 секунд.

2. На всем протяжении работы, контейнер потребляет в общей сложности ресурсов меньше, чем виртуальная машина, но работает дольше.

3. Если увеличить количество виртуальных машин до 3 раз, то количество потребляемых аппаратных ресурсов увеличивается в 2,5 раза. При этом время установки ОС на трех машинах одновременно удлинилось, но всего на 100 секунд, по сравнению с одной виртуальной машиной.

4. При одновременной работе трех контейнеров с VirtualBox общее потребление ресурсов так же возросло по сравнению с работой одного контейнера, а вот время работы увеличилось существенно – на 560 секунд.



Рис. 1. Нагрузка процессора и оперативной памяти на одном контейнере/виртуальной машине

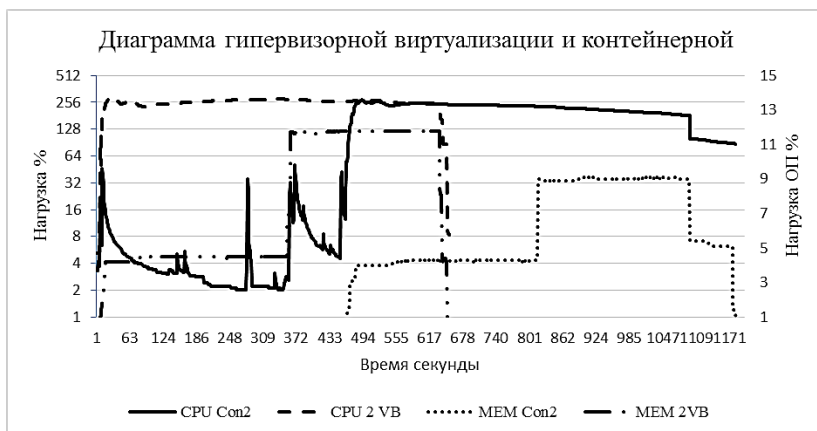


Рис. 2. Нагрузка процессора и оперативной памяти на трех контейнерах/виртуальных машинах

По результатам проделанных исследования можно прийти к следующему выводу: для построения системы работы с удаленными виртуальными машинами оптимальным решением можно считать использование виртуальных машин без контейнеров. Исследование показало, что потребление аппаратных ресурсов увеличивается практически одинаково с увеличением единиц виртуальных машин и контейнеров. Время выполнения действий так же растет, но с разной зависимостью. В случае виртуальных машин время растет на 1/5, а в случае контейнеров на 5/7. Для нашей системы этот критерий является очень важным. Он перечеркивает все основные плюсы контейнеров описанных выше. Для

полного подтверждения этих выводов требуется провести еще одно испытание, в котором количество виртуальных машин и контейнеров будет увеличено до десяти. Это поможет вывести наиболее точную зависимость по потреблению ресурсов и по времени затрачиваемому на работу.

Библиографический список

1. Jessie Frazelle's: Docker Containers on the Desktop (<https://blog.jessfraz.com/post/docker-containers-on-the-desktop/>) (дата обращения 7 апреля 2017).
2. *Маркелов А.* OpenStack. Практическое знакомство с облачной операционной системой издание «ДМК-Пресс» (2016).

УДК 004.031.42 65.011.56

Е.Ю. Гурьянов

Г.В. Аракчеев

студенты кафедры компьютерных технологий и программной инженерии

Т.М. Максимова

канд. техн. наук, доцент – научный руководитель

СОЗДАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОДДЕРЖКИ ОПЕРАЦИОННЫХ ПРОЦЕССОВ МАЛОГО БИЗНЕСА

В современных условиях важным аспектом функционирования малого бизнеса стало наличие программного обеспечения, которое позволит собирать и перерабатывать информацию, необходимую для принятия обоснованных управленческих решений. Передача информации о деятельности каждой точки малого бизнеса на высший уровень управления и взаимный обмен информацией между администраторами и «работчими» осуществляются на базе современной электронно-вычислительной техники.

В нашем случае, система разрабатывается для сети кофеен. Без использования ПО операционные процессы производились следующим образом:

- бариста вручную вели учет продаж в бумажном журнале;
- все поступления ингредиентов также записывались в отдельный журнал вручную;
- администратор вынужден лично объезжать все кофейни для контроля остатков ингредиентов;
- для сбора статистики продаж необходимо вручную анализировать огромное количество журналов учета.

Для оптимизации бизнеса, необходимо автоматизировать процессы, которые приносят прибыль и позволяют не выполнять лишних действий, как для бариста, так и для администратора.

Задачи.

Создание клиент-серверного приложения для автоматизации операционных процессов малого бизнеса. Целевое приложение предназначено для:

- ведения товарной матрицы целевой сети кофеен, включающей категории продуктов, их перечень и цену;
- ведения перечня закупаемых ингредиентов для приготовления продуктов;
- ведения перечней состава продуктов с указанием количества ингредиентов для их приготовления;
- ведения складских остатков ингредиентов для каждой из кофеен сети;
- ведения продаж продуктов по всей сети и для каждой из кофеен сети в отдельности;
- статистический модуль расчета операционных параметров как всей сети в целом, так и для каждой кофейни отдельно;

• создания и полуавтоматического поддержания системы мотивации бариста;

• организации оперативного пользовательского интерфейса для бариста;

• ведение базы данных гостей;

Целями разработки и внедрения системы являются:

- повышение прозрачности операционных процессов сети кофеен, их систематизация;
- повышение мотивированности персонала и эффективности его работы;
- увеличения степени общего контроля над сетью кофеен;
- увеличение прибылей от работы сети кофеен;
- повышение лояльности гостей к бренду целевой сети;
- минимизация временных затрат на внесение операционных данных и анализ метрик эффективности за счет создания простых в использовании пользовательских интерфейсов.

Для реализации клиент-серверного приложения были использованы такие технологии как – html/css, javascript, php, mysql, в частности, был использован php-фреймворк Webasyst и js-библиотека jQuery.

Соответственно MVC-архитектуре фреймворка система делится на три составных части – модель, представление и контроллер. Модель реагирует на команды контроллера и предоставляет контроллеру и представлению данные и методы работы с ними.

Представление получает данные от модели и выводит их пользователю. Контроллер обеспечивает связь между пользователем и системой, координирует и направляет данные в системе, обеспечивая её функционирование [1]. MVC архитектура позволяет уменьшить сложность приложения, код получается более структурированным и понятным. Это решение позволяет облегчить поддержку, тестирование и повторное использование решений, что актуально в случае нашей системы, потому что малый бизнес стремится к развитию и в системе постепенно появляются новые модули.

Модульность и гибкость – именно это и стало ключевыми критериями выбора данных технологий для построения системы, поэтому в выборе отпал пакет 1С. Также такие технологии как Spring Framework и ASP.NET были отброшены ввиду избыточности их компонентов, так как наша цель – легковесное решение.

Структурно приложение выглядит следующим образом (рис.1).

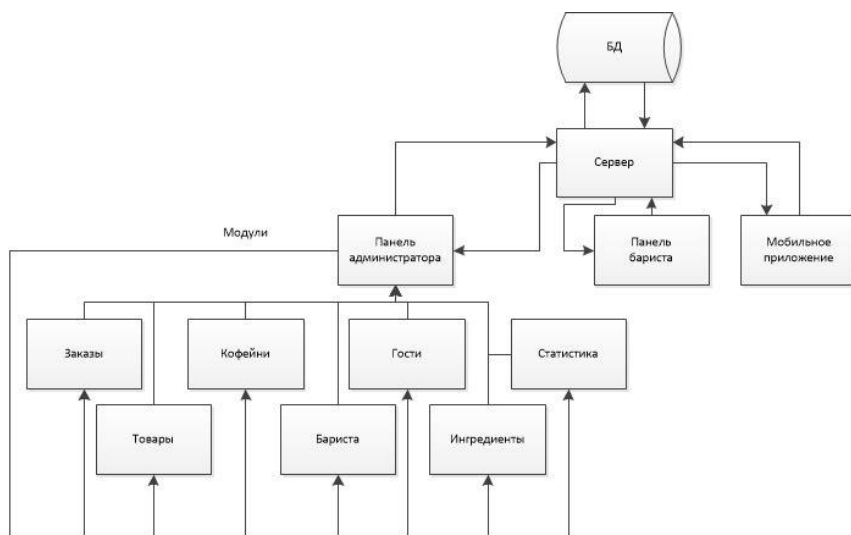


Рис. 1. Общая структура приложения

Как мы видим на схеме, система стоит из различных модулей. Панель администратора подразумевает под собой систему для внесения новых товаров, кофеен, бариста и гостей в систему. Также в данной панели будут выводиться данные о заказах, состоянии кофеен и возможность просмотра статистики.

Статистический модуль отражает состояние определенных явлений, по которым можно следить за их динамикой, ростом или спадом за выбранный промежуток времени и с нужным уровнем детализации. Эти данные позволят собрать информацию о направлениях развития бизнеса и повышении его доходности.

Панель бариста позволяет выполнять необходимый заказ, вносить гостей в систему и вносить данные о поставках.

Мобильное приложение

Опираясь на сказанное выше можно утверждать, что административное приложение покрывает большинство задач по управлению. Но не стоит забывать о гостях, ведь они являются движущей силой, которая способствует развитию бизнеса. Сейчас почти у каждой торговой сети есть мобильное приложение/мобильная версия сайта, предназначенная для гостей. Основная их задача – удержать старых гостей и привести новых, путем акций, скидок, отзывов и т.д. Сначала нам предстояло решить, что именно мы хотим получить на выходе, сайт или приложение. Давайте рассмотрим основные достоинства и недостатки

- Информативность: приложение содержит только самую необходимую информацию для гостя, такую как счета карт, список кафе и их обзорные страницы, информация о скидках и акциях. Если пользователю понадобится дополнительная информация, ему придется звонить по контактным телефонам или обращаться в кафе напрямую к персоналу, что абсолютно неудобно. В свою очередь сайт содержит всю необходимую пользователю информацию, вплоть до лицензий, но в то же время может содержать и множество избыточной для гостя информации, которая, скорее всего, ему не пригодится.
- Интерфейс: платформа позволяет разработчику унифицировать интерфейс, путем предоставления большого количества стандартных элементов. Приложения, сделанные таким образом, выглядят красиво, удобно и привычно для пользователя. В свою очередь, создание элементов интерфейса сайта полностью лежит на верстальщике, что дает большую свободу в реализации.
- Быстродействие: приложение сильнее привязано к платформе, следовательно, такие задачи, как отрисовка элементов и обработка событий будут выполняться значительно быстрее, чем на сайте.

Платформой для разработки был выбран AndroidAPILevel 25, минимальная версия – 4.0.3.

Для реализации серверной части был выбран php-фреймворк webasyst. Базой для серверной части послужило приложение, описанное выше. Под нужды мобильной части были созданы REST сервисы, нужным образом настроен роутинг и созданы отдельные таблицы в базе данных. Данные от сервера клиенту передаются в формате JSON, что обеспечивает универсальность и простоту в создании/обработке. На ранней стадии рассматривался вариант с передачей данных в XML виде, но этот вариант был отклонен ввиду несколько большей сложности создания XML на стороне сервера.

Сервисы, реализованные на серверной стороне, отвечают главным требованиям REST:

- модель клиент-сервер;
- отсутствие состояния;
- единообразие.

REST подход позволяет упростить написание API для взаимодействия с сервером и легко масштабировать структуру сервисов [2]. Это очень важно в нашем случае, так как некоторые вещи могут быть упущены из виду на стадии написания технического задания. Для взаимодействия с REST-сервисами мы используем библиотеку AsyncHttpClient, позволяющую легко делать асинхронные запросы.

Структурно мобильное приложение делится на следующие блоки (рис. 2).

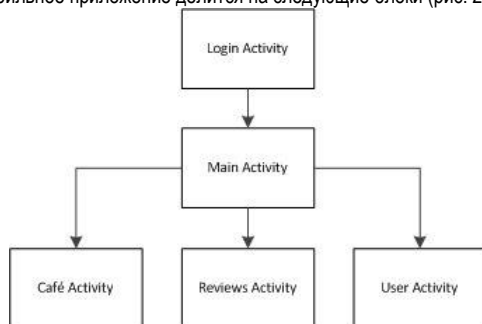


Рис. 2. Структура мобильного приложения

- Login Activity: предназначено для безопасного логина/регистрации пользователя.
- Café Activity: предназначено для просмотра информации о кофейнях сети. Представляет собой список кофеен с адресами, при клике на элемент которого открывается новое окно с подробностями.
 - Reviews Activity: предназначено для отправления отзывов. Отзыв может быть как обо всей сети, так и о конкретном кафе или конкретном бариста.
 - User Activity: предназначено для просмотра информации о госте. Можно узнать количество скидочных баллов, просмотреть акции и скидочные программы, в которых он участвует.
 - Menu Activity: предназначено для просмотра меню всей сети или конкретной кофейни.

Таким образом на текущем этапе разработки мы получили каркас приложения, который можно расширять дополнительными модулями. В дальнейшие планы входит верстка дизайна и запуск приложения в Play Маркет.

В данной работе мы рассмотрели систему для поддержки операционных процессов сети кофеен. Данная система, позволяет упростить хранение, сбор и обработку информации о деятельности всей сети и каждой кофейни в отдельности. Кроме того, обеспечена гибкость, позволяющая системе развиваться функционально, что необходимо для дальнейшего развития сети. Данные нововведения позволят улучшить контроль за деятельностью сети для оптимизации её работы и повышения прибыли.

Библиографический список:

1. Википедия // Свободная энциклопедия, [электронный ресурс] URL: <https://ru.wikipedia.org/wiki/Model-View-Controller>
2. Википедия // Свободная энциклопедия, [электронный ресурс] URL: <https://ru.wikipedia.org/wiki/REST>

УДК 004.921, 004.514, 004.946

М.М. Егоров

студент кафедры компьютерных технологий и программной инженерии

С.В. Щёкин

канд. техн. наук, доцент – научный руководитель

СРАВНИТЕЛЬНЫЙ АНАЛИЗ И ОЦЕНКА СРЕДЫ BLENDER 3D ВЕРСИЙ 2.49 И 2.78

На сегодняшний день трёхмерное представление информации для человека является наиболее понятным и привычным. В связи с этим растёт количество средств для 3D моделирования и методов для создания объёмных объектов. Так же распространяются и Virtual reality technologies, что даёт еще больший полёт фантазии для разработчиков.

Blender 3D – это свободный, профессиональный пакет для создания трёхмерной компьютерной графики, включающий в себя средства моделирования, анимации, рендеринга, постобработки и монтажа видео со звуком, компоновки с помощью «узлов» (Node Compositing), а также служит для создания интерактивных игр. В настоящее время пользуется наибольшей популярностью среди бесплатных 3D-редакторов в связи с его быстрым и стабильным развитием, которому способствует профессиональная команда разработчиков. Blender так же является Open source проектом и это является большим плюсом. Любой пользователь может самостоятельно устранить, при наличии соответствующих навыков, недоработки и внести надстройки для собственного удобства [1].

Для анализа я выбрал 2 версии Blender 3D: 2.49(a) и 2.78(a). Версия 2.49 – это последняя версия до глобального изменения и перекодировки среды. В табл. 1 приведены данные по кол-ву кода версий 2.49 и 2.78

Таблица 1

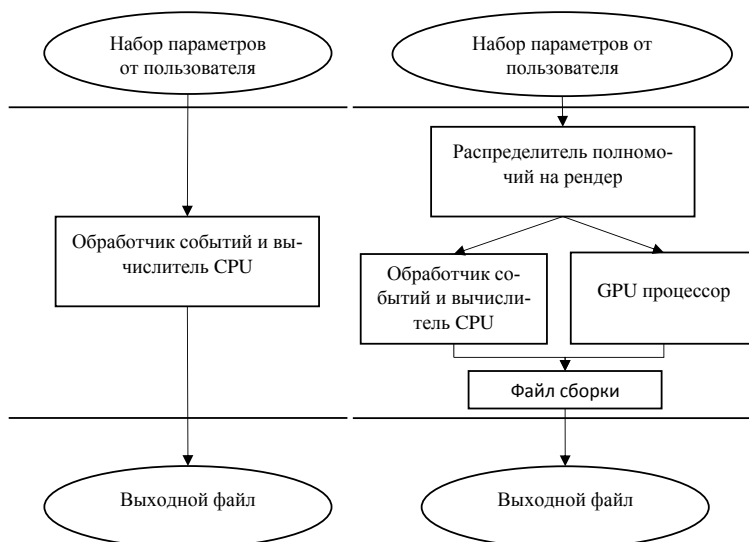
Версия	Дата	OpenCL	GLSL	CUDA	Assembler
2.49	13.06.2009	0	1567	0	107309
2.78	28.02.2017	389	3154	178	0

Сразу бросается в глаза полное исчезновение и кода Assembler и появление OpenCL вместе с CUDA. Это обусловлено несколькими факторами:

- интерфейс версии 2.49 устарел и не мог удовлетворять всем требованиям. Необходим полный редизайн;
- на момент разработки последующих версий необходимо было встраивание нового API Python;
- переработка обработчика событий, инструментов и данных, монтирование нового движка Cycles.

Сравнение стоит начать с самого трудоёмкого в затратном процессе в данной среде, а именно – визуализации. Для версии 2.49 нет никакого выбора в аппаратном подспорье при визуализации, так как используется всего 1 движок Blender [2]. В версии 2.78 присутствует второй движок Cycles. На мой взгляд – это одно из наибольших различий. Движок Blender использует только лишь ваш CPU, в то время как Cycles способен использовать CPU при поддержке GPU. На текущий момент очень активно развивается ряд видеокарт Nvidia и AMD, в которых постоянно растёт объём памяти, скорость обработки, Sli-режим и многое другое, тем не менее способ визуализации по-средством GPU имеет ограничения. Для карт Nvidia CUDA максимальное количество индивидуальных текстур ограничено 88 целочисленными текстурами (PNG, JPEG и так далее) и 5 текстурами с плавающей запятой (OpenEXR, 16-битный TIFF и прочие) на картах серии GTX 4xx/5xx. Более поздние карты не имеют такого ограничения [1].

Таким образом, процессы рендеринга для 2.49 и 2.78 выглядят следующим образом:



Из всего вышесказанного можно сделать следующие выводы:

- Рендеринг в версии 2.78 производится на порядок быстрее, чем в версии 2.49.
- Возможность подстраиваться под постоянно растущие запросы, по средствам подключения дополнительного аппаратного обеспечения.
- Унификация процессов.

Теперь мы можем провести общую оценку версий 2.49 и 2.78. Для оценки программного продукта будет использоваться «Методика сравнения технических критериев» [3]:

1. Сформируем таблицу критериев оценки, являющихся наиболее важными для технической составляющей. Таблица критериев оценки должна быть достаточно полной, но в то же время не должна быть излишне перегруженной во избежание излишнего «шума» в показателях. Все данные представлены в табл. 2.

2. Для каждого критерия необходимо определить методику оценки выполнения критерия, таким образом, чтобы «обезразмерить» исходные показатели. Данная процедура обычно называется шкалированием. Для этого результат оценки переводится в один из двух вариантов: равен 0 в случае невыполнения критерия и равен установленному значению в случае, если критерий оценки выполнен. Значение устанавливается самостоятельно в зависимости от количества необходимых выполнений требований.

3. Для каждого критерия выставляются коэффициенты значимости критерия для оценки продукта. Коэффициенты распределяются на отрезке от 0 до 1.

4. Весомость критерия необходимо задавать опираясь на собственные знания подкреплённые данными.

5. Производится расчет аддитивной суммы интегральной оценки для каждого сравниваемого продукта по следующей формуле:

$$Ozn = \sum_{i=1}^n (Zi * Ki) ,$$

где O_{zp} – интегральная оценка заявленного продукта, n – количество критериев сравнения заявленных продуктов согласно таблице критериев, Z_i – значение выполнения критерия (1 – выполнен, 0 – не выполнен), K_i – коэффициент значимости критерия сравнения (от 0 до 1).

Таблица 2

Наименование критерия	Методика оценки критерия	Коэффициент значимости критерия
Наличие API и его описания для интеграции в другие программные продукты	Критерий считается выполненным, если имеется описание API-продукта и методов его интеграции.	1
Документационное обеспечение продукта	Критерий считается выполненным, если содержится достаточно полная эксплуатационная и пользовательская документация.	1
Наличие GUI-интерфейса пользователя (без дополнительных надстроек)	Критерий считается выполненным, если в продукт по умолчанию встроен GUI-интерфейс пользователя.	1
Наличие встроенной системы помощи в интерфейсе пользователя	Критерий считается выполненным, если имеется встроенная система помощи в GUI-интерфейсе пользователя.	0.25
Полнота предоставляемого продукта	Критерий считается выполненным, если при установке не требуется производить дополнительной установки стороннего ПО для нормального функционирования.	0.5
Функциональность	Критерий считается выполненным, если возможно достижения функциональности, заявленной в документации.	1

Выполнив все расчёты, выясним, что итоговая оценка версии 2.49 = 4, а версии 2.78 = 4.75. В версии 2.49 в пункте 4 и 5 критерии не выполняются и следовательно имеют 0 баллов.

После проведённого выше анализа возможно сделать следующий вывод: 3D-технологии не стоят на месте, а такие проекты как Blender 3D один из лучших примеров Open source проектов в данной области, помимо всего этого мало кто занимается ретроспективным анализом, несмотря на его, порой, явную необходимость. Ретроспективный анализ крайне полезен и необходим, так как он позволяет провести черту между необходимостью и желанием проводить обновления. Вы сможете наблюдать реакцию пользователей и отслеживать когда лучше производить обновления, а также как часто это необходимо делать.

Библиографический список

1. Blender 2.78 Manual [Электронный ресурс]. – Режим доступа: <https://docs.blender.org/>
2. Michel Anders Blender 2.49 Scripting / Michel Anders – Packt Publishing, 2014, 182 с.
3. Методика сравнения программных продуктов [Электронный ресурс]. – Режим доступа: http://www.itsec.ru/articles2/control/o_metodike_sravneniya_programmnih_produktov/

УДК 519.246.85

Г. И. Емельянов

студент кафедры компьютерных технологий и программной инженерии

М. Д. Поляк

ст. преподаватель – научный руководитель

МЕТОДЫ И ПОДХОДЫ К РАСПОЗНАВАНИЮ ДОРОЖНЫХ ЗНАКОВ

Целью работы является исследование методов распознавания дорожных знаков с помощью сверточных нейронных сетей.

Бурное развитие вычислительных технологий компьютерного зрения привело к возможности создания приложений, ориентированных на массовый потребительский класс. Поскольку компьютеры позволяют использовать всё больше и больше мощностей, стало возможным использование информационных технологий для распознавания ситуаций на дороге в реальном времени. Использование методов компьютерного зрения позволяет повысить безопасность дорожного движения. Так, например, многие модели автомобилей уже оснащены механизмами распознавания как дорожных знаков, так и других объектов дорожного движения:

- светофоры;
- пешеходы;
- другие автомобили;
- разметка;
- и др.

Человеку, знакомому с достижениями в области компьютерного зрения, проблема обнаружения и распознавания дорожных знаков может показаться довольно просто решаемой. Современные алгоритмы обнаружения и классификации действительно могут очень качественно обнаруживать и классифицировать дорожные знаки на изображениях. Тем не менее, поскольку исследование приблизилось к коммерческим приложениям, угол рассмотрения проблем изменился: для систем помощи водителю уже стало неэффективно обнаруживать и распознавать дорожные знаки на одном изображении – необходимо максимально надежно обнаруживать их на сотнях тысяч видеок кадров с почти нулевой вероятностью ошибки, часто с использованием низкокачественных датчиков, доступных массовому потребителю.

В задачах распознавания дорожных знаков выделяют три метода:

- метод, основанный на цветовой палитре кадра (англ. Color-based detection method);
- метод, основанный на форме классифицируемого объекта (англ. Shape-based detection method);
- сверточные нейронные сети (англ. Convolutional Neural Networks).

Преимущество использования метода, основанного на цветовой палитре кадра, очевидно – на изображении можно найти область, содержащую интересующие нас цвета, используя пороговую или более продвинутые методы сегментации изображения [1]. Затем полученные участки изображения или немедленно обозначаются как дорожные знаки или передаются на следующий этап, как области, представляющие интерес (теоретическое местоположение дорожного знака). Основной недостаток такого подхода состоит в том, что цвет имеет тенденцию быть ненадежным: в зависимости от времени суток, погодных условий, освещения и других факторов изображение может варьироваться.

Существует несколько подходов к обнаружению дорожных знаков на основе формы классифицируемого объекта [2]. Наиболее распространенным из них является преобразование Хафа. Также популярны подходы, основанные на определении угла, или подходы, основанные на простом соответствии шаблону. Недостатком данного метода является также зависимость от условий съемки, а именно: расположения объекта и угла съемки.

Особенностью сверточных нейронных сетей (СНС) является наличие так называемых сверточных слоев, которые выполняют операцию свертки [3]. Выражение «свернуть изображение» означает,

что необходимо пространственно пробежать по изображению и вычислить скалярные произведения. Каждый сверточный слой передаёт в новый слой так называемые карты признаков, которые и являются результатами скалярного произведения. Эти карты признаков поступают на вход субдискретизирующих слоёв, задача которых состоит в уменьшении размерности карт признаков. Для уменьшения размерности часто используют одну из двух функций: max pooling, average pooling. Первая функция выбирает максимальное значение уменьшаемого участка карты признаков. Вторая функция выбирает уже среднее значение. Таким образом находится признак, наличие которого является наиболее ключевым фактором отнесения классифицируемого изображения к одному из классов. Задача выбора правильной очередности слоёв и вызывает наибольшие затруднения при разработке архитектур СНС.

В данной работе использовались 6 классов дорожных знаков, в каждом из которых находится примерно по 2000 изображений (за исключением одного класса – в нем находится всего 300 изображений).

Этот набор данных разбит на тренировочную выборку и валидационную, задача которой состоит в предотвращении переобучения сети.

Тренировочная выборка содержит 8325 изображений. Валидационная – 2775. Тестовая выборка, на которой будет проверяться результат обучения сети, содержит 3660 изображений. На рис. 1 приведен пример изображений.



Рис. 1. Пример изображений

В рамках данного исследования были построены две архитектуры СНС. Первая из них была спроектирована Яном Лекуном во время одного из соревнований по распознаванию дорожных знаков. Эта сеть содержит 3 сверточных слоя и один полносвязный. Речь идет только о трех слоях, потому часто сверточный и субдискретизирующий слои объединяют в один (особенно часто это можно встретить в англоязычной литературе). Полная схема архитектуры сети приведена на рис. 2.

Вторая разработанная архитектура называется LeCun+ и отличается от первой тем, что результат каждого снижения размерности передается на полносвязный слой, что увеличивает количество признаков, позволяющих отнести изображение к одному из классов. Схема архитектуры LeCun+ приведена на рис. 3.

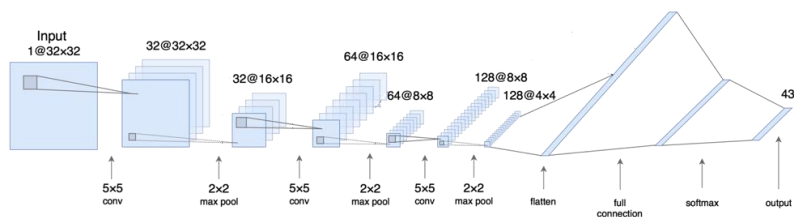


Рис. 2. Архитектура LeCun

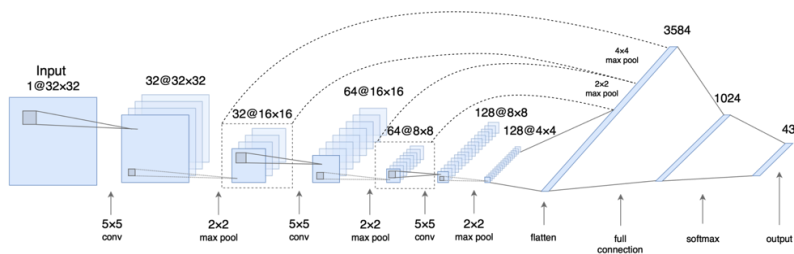


Рис. 3. Архитектура LeCun+.

Результаты, показанные сетями с архитектурами LeCun и LeCun+ после обучения, приведены в таблице.

Результаты обучения

	LeCun	LeCun+
Продолжительность обучения	01:44:10	01:48:01
Функция потерь для обучающей выборки	0.00110227	0.00088339
Доля правильных ответов для обучающей выборки	100,00 %	100,00 %
Функция потерь для валидационной выборки	0.00886583	0.00659086
Доля правильных ответов для валидационной выборки	99,75 %	99,82 %
Функция потерь для тестовой выборки	0.01890047	0.01761010
Доля правильных ответов для тестовой выборки	99,40 %	99,54 %

Как видно из результатов, сеть LeCun+ показала лучшие результаты, нежели сеть LeCun. Следовательно, можно сделать вывод, что формирование большего признакового пространства позволяет повысить качество классификации дорожных знаков.

Библиографический список

1. Gevers T. Color-based object recognition/Smeulders, A.W.M. – 1999. с. 453-464.
2. Heitz G. Shape-Based Object Localization for Descriptive Classification/ Elidan, G, Packer, B, Koller, D. – 2009. с.23
3. Nielsen, M. : Neural Networks and Deep Learning. 2016. URL: <http://neuralnetworksanddeeplearning.com/>

УДК 629.735.33

И. Б. Еремеев

студент кафедры проблемно-ориентированных вычислительных комплексов

О. О. Жаринов

канд. техн. наук, доцент – научный руководитель

ОБНАРУЖЕНИЕ СИГНАЛА ПО АМПЛИТУДНОМУ СПЕКТРУ ПРИ ПОМОЩИ НЕЙРОННОЙ СЕТИ

Задача обнаружения сигнала часто встречается в радиолокации и других областях радиотехники. Часто бывает необходимо знать, есть ли полезная составляющая в принимаемом сигнале [2]. Нейронная сеть позволяет автоматизировать процесс обнаружения сигнала.

Нейронная сеть представляет собой адаптивный фильтр, преобразующий входной сигнал в выходной [1]. Она обладает способностью давать близкие по значению выходные сигналы для схожих по определенным признакам входных. Это позволяет применять нейронные сети для решения задач фильтрации и классификации сигналов, построения систем автоматического регулирования, обнаружения полезного сигнала на фоне помех и шумов.

Основной задачей является создание такой сети, при подаче на вход которой сигнала с полезной составляющей, на выходе будет логическая единица, а при подаче шума – логический ноль. Чтобы сеть начала работать как надо, требуется ее обучить на выборке сигналов с полезной составляющей и выборке шумов.

Задачами работы являются: построение нейронной сети, работающей в режиме обнаружения сигнала, подбор структуры и параметров обучения, исследование работы сети, работающей в режиме обнаружения. Для моделирования использовались ClojureScript и Python с пакетами Anaconda и Theano.

Принципы работы нейронных сетей

Структура нейронных сетей

Нейросети возникли из попыток воспроизвести способность биологических нервных систем обучаться и исправлять ошибки, тем самым имитируя низкоуровневую структуру мозга [1]. Структурной единицей любой нейросети является нейрон (рис. 1).



Рис. 1. Модель искусственного нейрона

Искусственный нейрон на самом примитивном уровне имитирует свойства биологического нейрона. На его вход поступают сигналы со всех нейронов предыдущего слоя, умножаются на соответствующие им веса синапсов, затем в нейроне эти произведения суммируются, и полученное значение проходит через функцию активации, образуя выходной сигнал нейрона.

Функция активации ограничивает амплитуду выходного сигнала нейрона, возвращая ее в определенный диапазон. Обычно нормированный диапазон лежит в интервале $[0, 1]$ или $[-1, 1]$. Ниже приведены примеры функций активации (рис. 2).

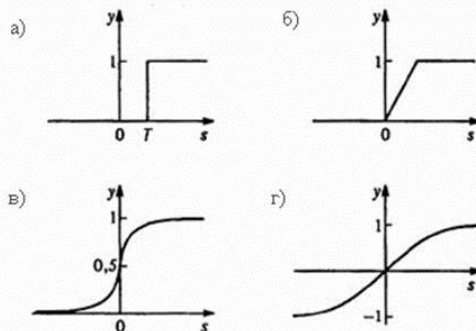


Рис. 2. Виды активационных функций: функция единичного скачка (а); кусочно – линейная функция (б); сигмоидальная функция (в); гиперболический тангенс (г)

Каждый нейрон в сети соединен синапсами со всеми нейронами около лежащего слоя. Изначально все синапсы имеют случайный вес.

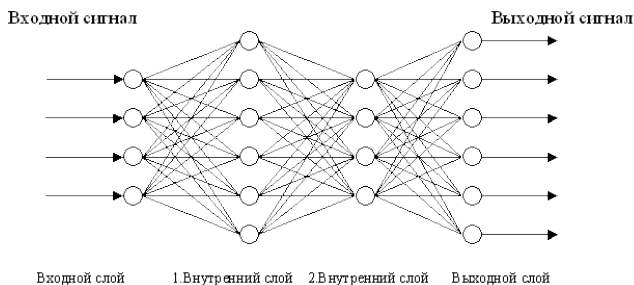


Рис. 3. Базовая структура нейронной сети

Обучение сети

Обучение заключается в настройке весов синапсов таким образом, чтобы выходной сигнал минимально отличался от желаемого. Для обучения сети создается тренировочный набор, каждый элемент которого представляет собой пару: входной вектор и соответствующий ему желаемый результат на выходе. Обучение сети на одной паре из тренировочного набора называют итерацией, а обучение сети на всем наборе – эпохой.

В сеть подается входной вектор, после чего выходной сигнал сравнивается с желаемым, вычисляется ошибка сети. Существует несколько способов расчета ошибки. В данной работе применяется один из самых распространенных, MSE или метод наименьших квадратов.

Для нахождения нужного значения веса синапса используется метод градиентного спуска. Градиентный спуск представляет собой способ нахождения локального минимума или максимума функции с помощью движения вдоль градиента. В качестве функции выступает зависимость ошибки сети от веса синапса. Задача в том, чтобы найти для каждого синапса глобальный минимум такой функции. В этом случае ошибка сети будет минимальна. Большую проблему представляют локальные минимумы, для преодоления которых используется такой гиперпараметр как момент (momentum). Момент позво-

лит преодолеть локальный минимум, если его значение окажется достаточным, но при выборе чересчур большого момента возможно «перескочить» глобальный минимум. Другим гиперпараметром является скорость обучения. При ее увеличении становится возможным более быстрое достижение глобального минимума, но при этом падает точность его определения.

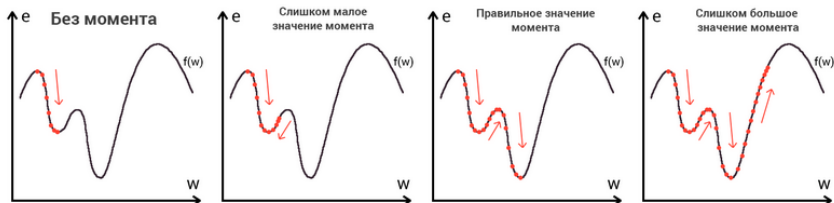


Рис. 4. Метод градиентного спуска с применением момента

Таким образом значения весов будут меняться в ту сторону, которая даст наилучший результат.

Применение сети для обнаружения сигнала

Для решения задачи обнаружения была сконструирована сеть с одним скрытым слоем и одним выходным нейроном, так как для решения текущей задачи не требуется больше одного скрытого слоя, а в качестве выходного сигнала достаточно одного числа, лежащего в диапазоне от 0 до 1. В качестве входного вектора используется амплитудный спектр сигнала, нормированный по максимальному значению, что было сделано для минимизации разнообразия входных данных при обучении сети.

Для тренировочного сета было сгенерировано множество сигналов разной частоты и с разной амплитудой, после чего к каждому сигналу был добавлен белый шум, СКО которого позволяло с большой уверенностью зрительно выделить наличие сигнала в амплитудном спектре. После на созданном сете проводилось обучение.

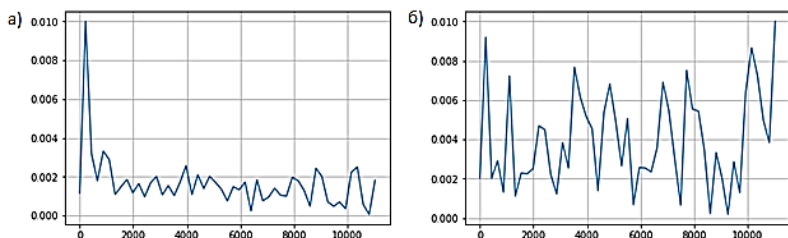


Рис. 5. Амплитудный спектр сигнала с полезной составляющей (а); амплитудный спектр шума (б)

Анализ сигналов обученной нейросетью

Чтобы оценить качество работы сети в режиме обнаружения, в нее подавалась выборка из пяти тысяч сигналов с определенным отношением амплитуды полезного сигнала к среднеквадратическому значению шума. После чего выбиралось решающее правило, проводящее сравнение выходного значения сети с пороговым значением, отвечающее на вопрос наличия сигнала. Так как сеть выдает результаты в диапазоне от 0 до 1, то и пороговое значение выбирается из этого диапазона. Таким образом можно вычислить вероятность правильного обнаружения, рассчитывающуюся как отношение количества обнаруженных сигналов к объему выборки (рис. 6). Изменяя отношение амплитуды полезного сигнала к девиации шума, возможно получить зависимость вероятности правильного обнаружения от соотношения амплитуда – шум. Результаты представлены на рис. 6, а. Заметно, что при соотношении ам-

плитуда – шум выше единицы, сеть почти что безошибочно обнаруживает сигнал. Но при уменьшении данного соотношения обнаружение становится все более затруднительным.

Другой немаловажной характеристикой является вероятность ложного обнаружения. Для ее расчета в сеть подается большое количество выборок шума (в данной работе 10 000). После чего высчитывается отношение элементов, принятых за сигнал, к общему числу шумов в выборке. Изменяя для данной выборки пороговое значение, возможно получить зависимость вероятности ложного обнаружения от порогового значения. Результат представлен на рис. 6,б.

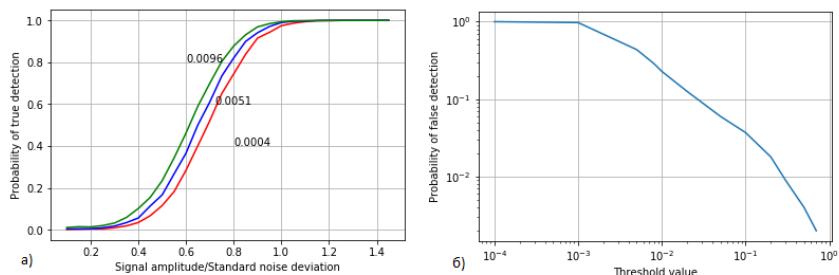


Рис. 6. Зависимость вероятности правильного обнаружения от соотношения амплитуда – шум для разных вероятностей ложного обнаружения (а); зависимость вероятности ложного обнаружения от порогового значения (б)

Имея эти характеристики, можно построить зависимость вероятности правильного обнаружения от вероятности ложного обнаружения (рабочая характеристика обнаружения).

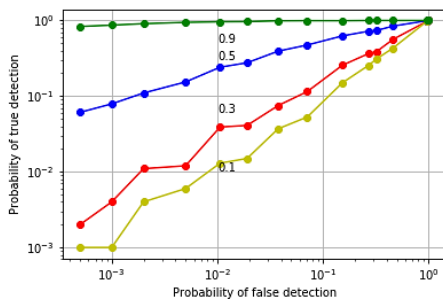


Рис. 7. Семейство рабочих характеристик сети в режиме обнаружения для различных соотношении амплитуда – шум

Заключение

Была сконструирована и обучена нейронная сеть, способная работать в режиме обнаружения сигнала на фоне шума. К достоинствам разработанного метода можно отнести адаптацию под любой уровень шума, а также простоту и скорость обучения сети. Из недостатков текущей версии стоит отметить реализацию в частотной области, из-за чего обнаружение в реальном времени происходит с задержкой, так как для преобразований Фурье требуется выборка. Свойства нейронных сетей позволяют производить обнаружение в реальном времени, что планируется реализовать в новых версиях проекта.

Библиографический список

1. Хайкин С. Нейронные сети: полный курс, 2-е изд. Издательский дом «Вильямс», 2008. 1103 с.
2. Воробьев С. Н. Эффективное обнаружение детерминированных сигналов. 2003. 139 с.

УДК 004.032.26

А. Ю. Еремченко

магистрант кафедры компьютерных технологий и программной инженерии

М. Ю. Охтилев

д-р техн. наук, профессор – научный руководитель

ПЕРЕХОД К МОДЕЛЯМ С ПЛАВАЮЩИМИ ВЕСОВЫМИ КОЭФФИЦИЕНТАМИ ДЛЯ АНАЛИЗА И ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ

В настоящее время для изучения свойств сложных систем и прогнозирования их поведения, в том числе и при экспериментальных исследованиях, является актуальным подход, основанный на анализе данных, которые получены в ходе работы системы. Обычно значения такого набора данных называют наблюдаемыми значениями.

Наблюдаемый набор данных – это последовательность значений некоторой переменной (или переменных), регистрируемых через некоторые равные промежутки времени (т.е. временной ряд). Исходя из этого можно сформулировать следующую постановку задачи.

Имеется набор данных:

t	t_1	t_2	t_3	...	t_n
x	x_1	x_2	x_3	...	x_n

Построить модель для прогноза значений $[x_n; x_{n+k}]$ на отрезке $[t_n; t_{n+k}]$ такую, что бы значение отклонения от зафиксированных значений $[x_1; x_{n-m}]$ на отрезке $[t_1; t_{n-m}]$ было минимальным.

Прежде чем приступить к построению моделей, имеющиеся данные нужно перевести в формат пригодный для анализа. Этот этап называется предварительной подготовкой данных. Сформулировать задачу для данного этапа можно следующим образом. Пусть поведение имеющихся зафиксированных данных $[x_i; x_j]$ описывается функцией $y=f(x)$, такой что:

x	x_1	x_2	x_3	...	x_{i-1}
y	$y_1 = x_2$	$y_2 = x_3$	$y_3 = x_4$...	$y_{i-1} = x_i$

Существует множество классических математических методов для анализа данных. Если рассматривать такие методы как модели, то становится возможным переход от математических моделей к нейронным. Например, для метода наименьших квадратов имеется возможность перейти к простейшему перцептрон (с одним скрытым слоем). Математическая запись метода наименьших квадратов выглядит следующим образом:

$$\sum_i e_i^2 = \sum_i (y_i - f_i(x))^2 \rightarrow \min_x$$

А модель, которая реализует тот же самый метод наименьших квадратов, но представляет собой простейший перцептрон, изображена рис. 1. Важно понимать, что с формальной точки зрения то, что отображено на рисунке и то, что записано в виде формулы, – это одно и тоже.

Из всего множества независимых переменных поочередно на вход перцептрона подается по одному элементу. Значение данного элемента (x_i) умножается на веса (w_i), суммируется между собой и прибавляется b .

Перцептрон – это простейший (с одним скрытым слоем) искусственный нейрон [1]. Согласно современной терминологии, перцептрон можно классифицировать как искусственную нейронную сеть, если он удовлетворяет следующим требованиям:

- состоит из одного скрытого слоя;
- является сетью с прямым распространением сигнала;

– использует пороговую передаточную функцию.

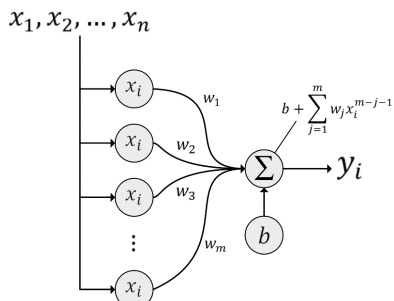


Рис. 1. МНК, представленный в виде простейшего персептрона

Простые нейронные сети с прямой связью склонны к переобучению. Попытаться решить эту проблему можно при помощи Байесовского вывода. В научной работе [2] было показано (но не доказано), что введение неопределенности может улучшить прогнозируемую точность в регрессионных моделях.

Все веса в нейронной сети должны быть описаны каким-либо распределением (рис. 2). Таким образом появляется возможность вместо обучения одной сети, обучить ансамбль сетей, где каждая сеть имеет свои веса, полученные из общего заданного распределения вероятностей. Для конфигурации значений весов можно использовать численные методы, например метод Монте-Карло.

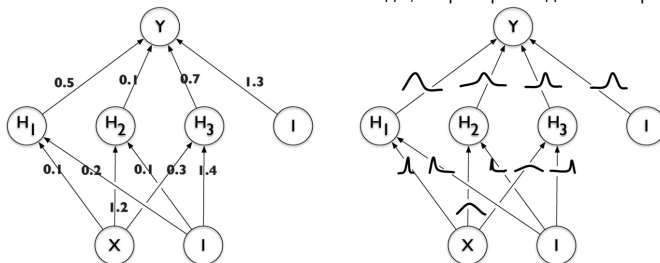


Рис. 2. Схемы нейронных сетей

Слева: каждый вес имеет фиксированное значение, что обеспечивается классическим методом обратного распространения ошибки. Справа: каждому весу назначается распределение, как это предусмотрено Байесовским методом обратного распространения ошибки.

В общем случае точный байесовский вывод о весах нейронной сети трудно поддается определению, поскольку, число параметров очень велико и функциональная форма нейронной сети не поддается точной интеграции. Для решения этой проблемы можно воспользоваться вариационной аппроксимацией для точного байесовского обновления. Или изменить модель нейронной сети на элементарный персептрон.

Неопределенность в скрытых слоях нейронной сети позволяет утверждать наличие неопределенности в отношении конкретного наблюдения, неопределенность в весах дополняет собой уже имеющиеся неопределенность. Это позволяет утверждать о наличии неопределенности относительно того, какая нейронная сеть является подходящей, что приводит к регуляризации весов и усреднению модели.

Самым главным условием такой сети является то, что при маленьком объеме выборки такой подход только ухудшит показатели работы сети, однако, по мере увеличения количества данных неопределенность может уменьшаться (а может и нет), позволяя решениям, принимаемым сетью, становиться более детерминированными, поскольку среда лучше понимается.

Согласно определению переобучение – это эффект, заключающийся в том, что модель показывает хорошие результаты при обучении, но плохие при непосредственном использовании [3]. Это происходит из-за того, что на основе такой модели принимается решение о нахождении «идеального» набора параметров. У данного подхода есть очень важное преимущество перед классическим подходом – данная модель не подвержена переобучению. Это достигается за счет того, что при данном подходе найти «идеальный» набор данных невозможно.

Алгоритмически реализовать такую модель можно следующим образом:

1. Инициализировать одномерный массив независимых переменных x .
2. Инициализировать одномерный массив значений функции $y=f(x)$.
3. Найти максимальный элемент в массиве y .
4. Инициализировать степень искомого полинома p .
5. Цикл со счетчиком i от 2 до p (функция подготовки независимых переменных для дальнейшей обработки):
 - a. формируется временный массив, где каждый элемент является элементом массива x возведенным в степень i ;
 - b. полученный временный массив добавляется в конец массива x ;
 6. цикл со счетчиком i от 1 до p (формируется массив максимальных значений):
 - a. i элемент массива равен максимальному значению, найденному на 3 шаге, возведенному в степень i ;
 7. Новые значения для матрицы x являются результатом поэлементного деления матрицы x на массив максимальных значений;
 8. Инициализация переменной b значением равным -1;
 9. Инициализация массива коэффициентов искомого полинома p -й степени, происходит по формуле: $w=\text{rand}-0.5$, где rand – функция получения случайного числа в диапазоне $[0;1]$ подверженного равномерному распределению;
 10. Цикл от 1 до общего числа итераций, которые необходимы для нахождения w (значение по умолчанию 100000):
 - a. переменная h является результатом скалярного произведения x и w плюс b ;
 - b. вычисляется новое значение

$$w = w - 0.1 \cdot \frac{1}{n} \sum_{i=1}^n ((h - y) x),$$

- c. вычисляется новое значение

$$b = b - 0.1 \cdot \frac{1}{n} \sum_{i=1}^n (h - y),$$

где n – размерность массива, полученного в результате выполнения операций, находящихся под знаком суммы.

В качестве вывода можно отметить, что переход к моделям с плавающими весовыми коэффициентами может значительно повысить точность результатов относительно моделей с фиксированными весовыми коэффициентами. Также такие модели более чувствительны к динамике изменения изучаемой системы. И нельзя не отметить изначальную невосприимчивость к переобучению.

Библиографический список

1. *Rosenblatt F.* Principles of Neurodynamics / Washington, DC: Spartan Books, 1962.
2. *Weight Uncertainty in Neural Networks / Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, Daan Wierstra.* Proceedings of the 32nd International Conference on Machine Learning (ICML 2015) – Google DeepMind, May 21, 2015
3. *Hastie T., Tibshirani R., Friedman J.* The Elements of Statistical Learning, 2nd edition / Stanford: Springer, 2009. 533 с.

УДК 004

Ерохин Р.А.

студент группы Z6440М.

М.Б. Игнатьев

д-р техн. наук, профессор – научный руководитель

ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА АДАПТИВНОГО ШАГАЮЩЕГО РОБОТА ДЛЯ ОБСЛЕДОВАНИЯ ГАЗОПРОВОДА

В статье рассматривается возможный вариант информационно-вычислительной системы робота, предназначенного для обследования газопровода. Конструкция и принцип действия адаптивного шагающего робота описаны в публикациях [1], [2] и патенте № 2571242 [3] (автор профессор, доктор технических наук, М.Б. Игнатьев и др.). Для обследования газопроводов применяются различные методы и устройства. Так, например известен дефектоскоп-снаряд, с установленными на нем различными датчиками, движущийся под воздействием давления перекачиваемого продукта. Недостатком данного устройства является то, что для проведения качественного обследования трубы необходимо снижение скорости перекачки газа, что приводит к задержке его поставки. Имеются конструкции роботов колесного типа, недостатком которых является невозможность прохождения сильно-наклонных и вертикальных участков.

Предлагаемый вариант адаптивного шагающего робота, наряду с традиционными средствами диагностики, позволит повысить качество и частоту обследования газопроводов. При этом не требуется перекрывать или ограничивать подачу газа в трубе, что является неоспоримым преимуществом. Робот является полностью автономным за счет применения оригинальной ветровой турбины, обеспечивающей получение необходимой электроэнергии потоком газа. Слово адаптивный, в названии системы, подразумевает возможность робота приспосабливаться к различным деформациям и изгибам, подстраиваться под диаметр трубы в заданных пределах. Это достигается за счет оригинальной конструкции ног робота, которые поочередно втягиваются или вытягиваются до соприкосновения со стенками трубы. Для связи с оператором может быть использован стандартный модуль Wi-Fi, а в качестве волновода сама труба газопровода. По предложению автора на роботе может быть установлено до шести web-камер с полупроводниковыми светодиодами подсветки, что обеспечит полный круговой осмотр трубы. Также вместо камер могут быть установлены ультразвуковые генераторы и приемники. Таким образом, оснащенный робот может передвигаться в теле трубы на многие километры и производить обследование в автономном режиме. В случае застревания или поломки робота необходимо предусмотреть возможность его эвакуации, например с помощью другого робота, используя систему стыковки или захвата. Также в случае ввода в эксплуатацию, необходимо предусмотреть систему помещения робота в тело трубы и его извлечения.

Таким образом, поставлена задача разработки информационно-вычислительной системы, которая обеспечит необходимую функциональность, управление перемещением робота, координацию диагностических систем и системы электропитания, связь с оператором. Эксплуатация газопроводов производится в различных климатических условиях, от -60°C на Крайнем Севере и до $+40^{\circ}\text{C}$ в южных районах, причем температура на выходе компрессорной станции может достигать и 60°C . Таким образом, к элементной базе создаваемой системы предъявляются достаточно жесткие требования. Здесь возможно применение элементов зарубежного производства, исполнения (industrial) или отечественного (специального назначения) с необходимыми параметрами температурного режима работы. На рис. 1 [2] представлена блок-схема системы управления роботом.

Блок энергетике представляет собой совокупность источника энергии, формирователей необходимых напряжений для питания исполнительных узлов и управляющих блоков, возможно, систему резервного питания, а также схему управления и диагностики самого блока. Автор предлагает в качестве источника энергии системы использовать стандартный ветровой генератор, например фирмы

«Siemens». Для формирования необходимых напряжений питания можно использовать линейные или импульсные стабилизаторы, как зарубежных, так и отечественных производителей. Например, отечественной промышленностью выпускаются интегральные стабилизаторы напряжения серии 1264 (прототип LT1083 фирмы «Linear Technology»), рассчитанные на диапазон выходных напряжений от 1,25 В до 12 В и обеспечивающие ток нагрузки до 7 А с диапазоном рабочей температуры от -60 до +125°С. Согласно предложенной конструкции робота вся электроника размещается в герметичной полости, сделанной из металла, поэтому сам корпус может быть использован как радиатор для отвода выделяемого тепла электронных компонентов.

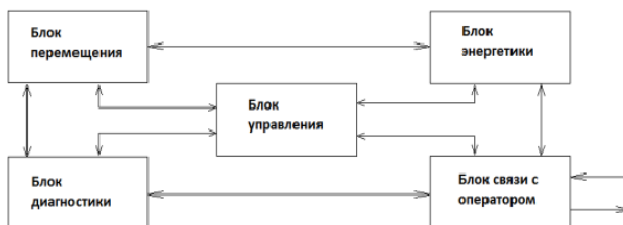


Рис. 1

Сердцем робота является блок управления, обеспечивающий координацию работы всех узлов системы, управление движением, включением и переключением датчиков и исполнительных механизмов, обеспечение управления приемом и передачей информации, обработку и интерпретацию принимаемых команд от оператора и т.д. Блок управления может быть организован на базе микропроцессора (микроконтроллера) с набором необходимых периферийных элементов (преобразователей уровня, согласования, формирователей, тактовых генераторов и т.п.). В настоящее время в России ряд предприятий выпускают современные микроконтроллеры, например фирма «Миландр» предлагает серию (1986ВЕхх) 32-разрядных микроконтроллеров, построенных на ядре ARM Cortex-M3 с тактовой частотой до 144 МГц. Диапазон рабочих температур от -60 до +125°С. На платформе микроконтроллеров ARM можно проектировать встраиваемые узлы с применением операционных систем реального времени (ОСРВ или RTOS), например семейства QNX. Подобные системы компактны, не требовательны к аппаратным ресурсам. Позволяют создавать гибкие, настраиваемые устройства, менять свойства и задачи даже в процессе эксплуатации. В такой конфигурации часть функций блока диагностики может быть возложена именно на такой микроконтроллер (как вариант исполнения). А именно, функции определения возможных дефектов трубы газопровода по получаемым данным видео или фотоизображений, а также на основе данных ультразвуковых датчиков. Такая система позволит производить диагностику без постоянного участия оператора, т.е. в полуавтоматическом режиме, до обнаружения возможного дефекта. В случае обнаружения, робот останавливается, делает запрос на пульт оператору, который визуально или с помощью дополнительных средств диагностики, а также с применением экспертных систем определяет характер и степень повреждения или его отсутствия. Возможный алгоритм и аппаратно-программную реализацию данной функции постараюсь рассмотреть в следующих статьях, а здесь остановимся на системе управления. На рис. 2 представлен возможный вариант структурной блок-схемы с применением микроконтроллера на ядре ARM серии 1986ВЕхх. Микроконтроллер (МК) под управлением встроенной операционной системы координирует работу всех модулей робота. В качестве шин управления и передачи данных между микроконтроллером и внешними блоками могут быть использованы стандартные, встроенные в МК, например I²C, CAN, UART, USB и др.

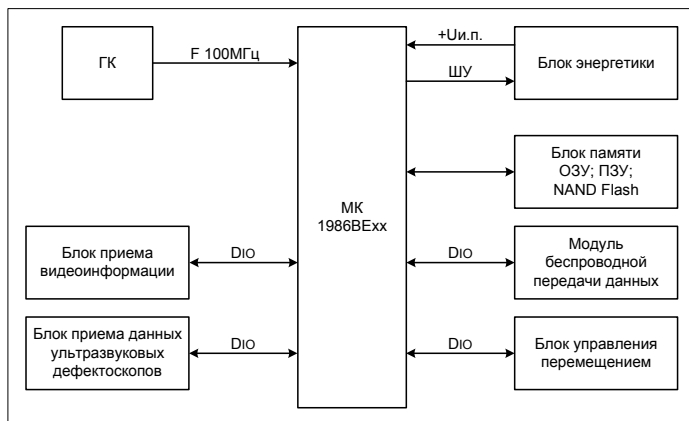


Рис. 2

МК оснащен различными модулями АЦП, ЦАП, компараторами, температурными датчиками, сторожевыми таймерами, таймерами-счетчиками с функциями ШИМ и регистрации событий, достаточным количеством пользовательских линий ввода-вывода, что позволяет строить полноценные системы с нужной функциональностью. Имеются встроенные в МК интерфейсы SWD и JTAG, что позволяет оперативно, в составе аппаратуры производить отладку и менять конфигурацию, обеспечивая гибкость настройки и возможность адаптации системы под конкретные условия эксплуатации и задачи.

Блок энергетик обеспечивает необходимые напряжения питания и связан с МК шиной управления. Блок ГК (генератор кварцевый), предназначен для формирования тактовых импульсов (F МГц), необходимых для работы системы. Отечественной промышленностью выпускается ряд кварцевых генераторов, обеспечивающих стабильность параметров в диапазоне температур от -60 до +85°С, например ГК-108-П. Блок внешней памяти предназначен для хранения программ и обрабатываемых данных и может работать с аппаратной поддержкой режима DMA.

Блок перемещения включает в себя устройство управления исполнительными механизмами, обеспечивающими передвижение робота по заданной программе. Алгоритм передвижения представляет собой циклический процесс с возможностью внешнего управления командами, например стоп, вперед, назад. Данную систему проще всего реализовать в виде конечного автомата, с возможностью управления внешними сигналами. Для создания подобного автомата в аппаратной реализации наиболее целесообразно использовать ПЛИС (программируемая логическая интегральная схема), на базе которой достаточно просто формируются необходимые функции, интерфейсы обмена данными, управляющие автоматы и т.п. Отечественной промышленностью освоен выпуск ряда микросхем ПЛИС, способных работать при температурах от -60 до +125°С, например серия 5576ХСxxx. Типовая логическая емкость таких микросхем порядка 200 000 вентилей, примерно 10 000 логических элементов, что вполне достаточно для реализации поставленной задачи. В качестве среды разработки (конфигурирования) данных ПЛИС можно использовать MAX+PLUS II или Quartus II, т.е. типовые, известные САПР. Применение ПЛИС позволяет создавать устройства, с возможностью многократного изменения конфигурации в составе аппаратуры, что значительно расширяет возможности для дальнейшей модернизации и упрощает процесс отладки. На рис. 3 представлен возможный вариант структурной блок-схемы управления перемещением робота.

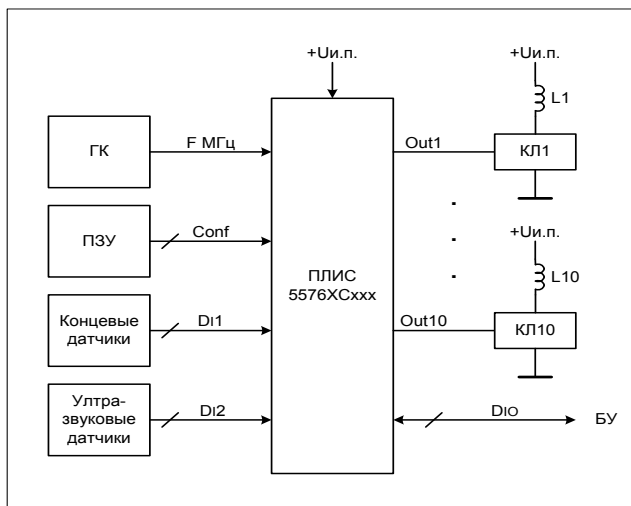


Рис. 3 Блок управления перемещением

На кристалле ПЛИС размещается вся логика автомата управления, выработки управляющих сигналов на исполнительные механизмы, обработки входных данных с концевых и ультразвуковых датчиков (D1, D2), а также интерфейса обмена данными и получения инструкций от микроконтроллера блока управления. Для связи с микроконтроллером может быть применена стандартная шина обмена данными (DIO), например I²C, UART, CAN и др., с применением системы прерывания. Данные шины достаточно просто реализуются на ПЛИС, причем существуют уже готовые так называемые IP-блоки данных интерфейсов как со свободной, так и платной лицензией. Блок ПЗУ – постоянное запоминающее устройство предназначено для хранения конфигурационных данных, загружаемых в ПЛИС. Выходы ПЛИС Out1..Out10, подаются на входы соответствующих силовых ключей КЛ1..КЛ10, обеспечивающих переключение подачи напряжения на катушки L1..L10 (соленоиды) исполнительных механизмов. В качестве ключей могут быть использованы биполярные или полевые транзисторы, либо твердотельные реле, рассчитанные на необходимый ток коммутации. Также на кристалле ПЛИС может быть размещен блок, выполняющий функции или часть их, определения возможных дефектов полости трубы по получаемым видеоданным или данных ультразвуковых дефектоскопов (на схеме не показано).

Блок диагностики выполняет функцию обнаружения всевозможных дефектов и нарушений полости трубы газопровода, посредством установленных на работе различных датчиков, в качестве которых могут быть использованы ультразвуковые датчики для дефектоскопии или видеокamеры (веб-камеры со светодиодной подсветкой). По предложению автора, для обеспечения кругового обзора достаточно 6 камер. Для организации обработки и передачи изображений с 6 камер можно применить метод мультиплексирования или параллельной обработки изображений всех камер с выводом одной, выбранной оператором или алгоритмом программы. Как упоминалось выше, функции обработки изображений и распознавания возможных дефектов может быть возложена на микроконтроллер блока управления или с применением отдельного чипа, а также на кристалле ПЛИС блока перемещения. На данный момент вопрос не решенный, зависит от конкретной реализации с применением определенной элементной базы и выработанной стратегии разработки системы. Для обнаружения больших препятствий и поворотов в трубе можно использовать ультразвуковой датчик расстояния, установленный на торце корпуса робота. Например, модуль HC-SR04 использует акустическое излучение для определения расстояния до объекта, обеспечивает высокую точность и стабильность измерений в диапазоне от 2 до 400 см, и углом измерений в 30°. Оснащенный таким датчиком робот сможет автоматически адап-

тироваться к ситуации, без вмешательства оператора и в зависимости от программы останавливаться, менять направление движения, посылать сообщение и т.п.

В качестве блока связи может быть использован стандартный модуль Wi-Fi, обеспечивающий беспроводную передачу информации с датчиков робота на пульт и прием управляющих команд и инструкций от оператора. Модуль должен обеспечить достаточную пропускную способность передачи видеоинформации в реальном времени, что позволит производить качественное обследование полости трубы во время движения робота. Так, например, для использования в промышленности фирма «Siemens Automation & Drives» предлагает Wi-Fi-решения для своих контроллеров SIMATIC в соответствии со стандартом IEEE802.11g в свободном диапазоне 2,4 ГГц и максимальной скоростью передачи 54 Мбит/с, что при первом приближении вполне достаточно для данной системы.

Создание информационно-вычислительной системы адаптивного шагающего робота, обеспечивающей необходимую функциональность, задача не простая, но интересная. В процессе разработки потребуется решить ряд технических вопросов, связанных с обеспечением электропитания нужной мощности, необходимости использования элементной базы с рабочим диапазоном температуры от -60 до +60°С, при ограниченности размеров и формы печатных плат устройства, обеспечение связи на большие расстояния. И возможно другие, еще не известные на данный момент, вопросы, которые наверняка возникнут.

Библиографический список

1. *Игнатьев М.Б.* Кибернетическая картина мира. Сложные киберфизические системы. СПб., 2014.
2. *Игнатьев М.Б.* Просто кибернетика. Научно-популярное издание. СПб.: »Страта», 2016. 248 с.
3. *Игнатьев М.Б., Липинский Я.А., Жаринов О.О., Ненашев В.А., Макин П.И., Герасимов Г.М.* Автономный адаптивно шагающий робот для диагностики газопроводов. Патент № 2571242.
4. Интернет-ресурсы. (Википедия и др.).

УДК 0010828(960)2326464

А.С. Журавлев

студент кафедры проблемно-ориентированных вычислительных комплексов

Н.Н. Григорьева

старший преподаватель – научный руководитель

РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ИНТЕРФЕЙСА БОРТОВОГО ИНДИКАТОРА

В структуре бортового радиоэлектронного оборудования пилотируемого летательного аппарата системы индикации занимают значительное место. Индикационное оборудование на борту летательных аппаратов является неотъемлемым инструментом для отображения информации, необходимой для пилотирования. В связи с увеличением диапазона функциональных задач для бортовых индикаторов и, как следствие, увеличением объема приборного оборудования, устанавливаемого в кабине, необходимо уменьшать габаритные размеры без потери функциональности. Одним из путей решения данной задачи является отказ от модулей «дисплей + механические кнопки» в пользу сенсорных дисплеев, однако последние в настоящее время плохо воспринимаются операторами. Для разрешения данного конфликта был спроектирован стенд, предназначенный для тестирования сенсорного индикатора.

Бортовые индикаторы, размещаемые в кабине экипажа, можно охарактеризовать рядом параметров – светотехнических, геометрических, массогабаритных и других. К основным характеристикам можно отнести:

- допустимые углы обзора;
- состав и форму пикселя;
- яркость изображения, равномерность яркости по площади экрана, возможности по регулировке яркости;
- контрастность изображения;
- цветовые возможности индикатора;
- коэффициенты отражения экрана;
- частоту регенерации изображения;
- геометрические искажения, точность воспроизведения изображения;
- используемое электропитание, потребляемую мощность;
- массу и габариты;
- наработку на отказ.

Традиционно информация от бортовых индикаторов представляется в знаково-графическом виде. Однако выяснилось, что даже если по стрелочному прибору можно определять точное значение параметра на шкале, зачастую оператор воспринимает прибор как графический образ - для него не менее важны направление и скорость движения стрелки, а также цвет, форма и размер индикаторов и элементов. В настоящее время преобладающее большинство бортовых индикаторов встраиваются в дисплейные комплексы, имеющие механические кнопки по краям. Для уменьшения габаритных размеров можно частично (в перспективе полностью) отказаться от механических кнопок. Для этих целей можно применить сенсорные экраны.

Необходимо разработать стенд, демонстрирующий интерфейс бортового оборудования по отображению информации на основе сенсорной панели.

В качестве решения данной задачи и целях экономии места, было решено использовать сенсорный экран, так как механические кнопки заняли бы примерно 1/3 занимаемой площади, к тому же он имеет хорошую подсветку в условиях низкой освещенности.

Сенсорные экраны имеют следующие варианты технологического исполнения:

- 1) резистивные (4-проводные, 5-проводные, 8-проводные);

- 2) емкостные;
- 3) на поверхностных акустических волнах(ПАВ или SAW);
- 4) инфракрасные.

Резистивные экраны – наиболее популярный и отработанный в технологическом плане вид экранов. За долгие годы развития, резистивные экраны получили высокую надежность и низкую стоимость. К тому же, подразумевая специфику использования данного дисплея, резистивный экран более предпочтителен из-за того, что оператор может нажимать на сам экран непосредственно в экипировке(перчатке). Наибольшую популярность имеют 4- и 5-проводные конструкции. В данной работе используется четырехпроводной экран, принцип работы которого показан на рис. 1.

Резистивный сенсорный экран состоит из гибкой пластиковой мембраны и стеклянной панели, на которые нанесено резистивное покрытие. Пространство между стеклом и мембраной заполнено микроизоляторами, которые равномерно распределены по активной области экрана и надёжно изолируют проводящие поверхности. При нажатии на экран, панель и мембрана замыкаются, и контроллер с помощью АЦП регистрирует изменение сопротивления, а далее преобразует его в координаты прикосновения (X и Y).

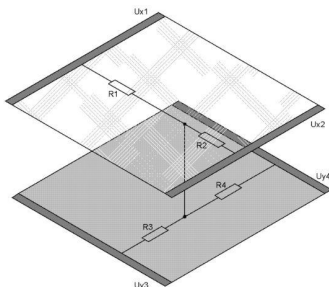


Рис. 1. Конструкция резистивного четырехпроводного сенсорного экрана

Структурная схема макета, демонстрирующая интерфейс рабочего кадра бортового индикатора, представлен на рис. 2

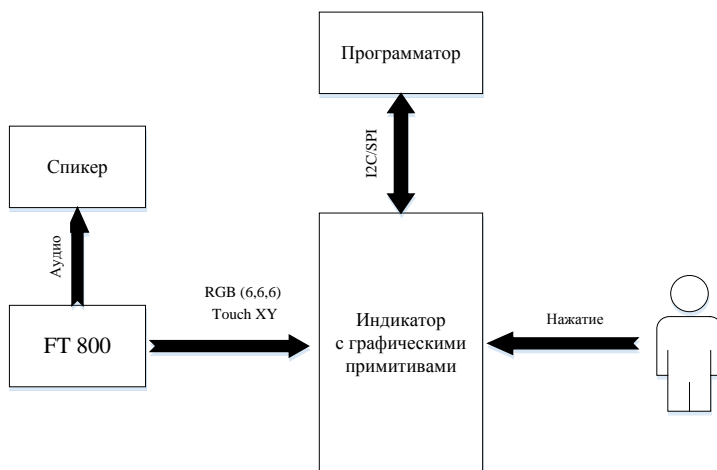


Рис. 2. Структурная схема макета

Для разработки стенда используется отладочная плата VM800C50A-D компании MicroChip. Данный модуль изображен на рис. 3, он имеет встроенный видеопроцессор и 3.5-дюймовый ЖК-экран.



Рис. 3. Модуль разработки

В качестве «связи» с ПК, в модуле используется кабель SPI. Источник питания должен выдавать +5В и иметь разъем microUSB.

В качестве проверки работоспособности видеопроцессора к воспроизведению динамических изображений была построена синусоида путем генерирования точек по заданному алгоритму.

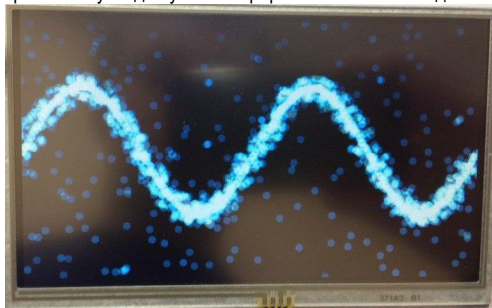


Рис. 4. Синусоида на ЖК экране

Рабочий кадр экрана изображен на рис. 5, он был выполнен с использованием аддитивной цветовой модели RGB. Выбор основных цветов обусловлен особенностями физиологии восприятия цвета сетчаткой человеческого глаза, а также стандартами в авионики.

В результате проделанной работы была разработана программа, имитирующая работу интерфейса бортового индикатора на резистивном сенсорном экране.



Рис. 5. Пример рабочего кадра дисплея

Рабочий кадр обеспечивает наглядную передачу информации об уровне полета и координатах местоположения, а также о текущей скорости летательного аппарата.

Разработанный макет имеет конструктивные плюсы, основными из которых являются: отказ от дополнительных механических кнопок и как следствие отказ от дополнительной подсветки для кнопок. Малые габаритные размеры являются также огромным плюсом, а интуитивный графический вид создает простоту в освоении. Разработанный макет может послужить для решения целесообразности использования сенсорных индикаторов в авиации.

Библиографический список:

1. Алиев Т.М., Вигдоров Д.И., Кривошеев В.П. Системы отображения информации. – М.: Высшая школа, 1988. 223 с.
2. Смирнов В.М. Устройства отображения информации: учеб. пособие. – СПб.: ГУАП, 2007. – 91 с.
3. Жаринов И.О., Жаринов О.О. Бортовые средства отображения информации на плоских жидкокристаллических панелях: учеб. пособие. – СПб.: ГУАП, 2005. – 144с.
4. Долеушин С. Начинаем работать с графическим контроллером FT880 FTDI. Компоненты и технологии №8. 2014.

УДК 004.054

М.С. Иванова, А.В. Сергеев

студенты кафедры проблемно-ориентированных вычислительных комплексов

Г.С. Евсеев

канд. техн. наук, доцент – научный руководитель

ОЦЕНКА ПОКАЗАТЕЛЕЙ КАЧЕСТВА ОБСЛУЖИВАНИЯ ПОЛЬЗОВАТЕЛЕЙ ВИДЕОКОНТЕНТА

Постановка задачи

Оценка качества видеосервиса требует учета некоторых субъективных критериев [1]. Например, разрешение видео, прерывания в процессе воспроизведения, время до начала воспроизведения и др. Методика оценки значений качества (vMOS) была предложена в 2009 году компанией Huawei. В результате исследования была предложена формула (1) для комплексной оценки качества видео:

$$vMOS = (sQ - 1) \left(\frac{\alpha(s_I - 1) + \beta(s_V - 1)}{4(\alpha + \beta)} \right) + 1, \quad (1)$$

где sQ – оценка качества видео; s_I – оценка времени начальной буферизации; s_V – оценка процента простоя системы; α и β – весовые параметры восприятия интерактивности и вида, которые определяются посредством опроса потребителей.

Все эти величины задаются таблично. В данном исследовании была использована следующая таблица с рекомендуемыми оценками (табл. 1).

Таблица 1

Оценка	Критерии качества		
	sQ	s_I	s_V
5	Разрешение видео / Битрейт [Мб/с] 1080p / 4.5	Время начальной загрузки (с) <=0.1	Процент простоя системы 0
4	720p / 2.5	1	5%
3	360p / 0.75	3	10%
<3	<360 (240) / <0.5	>5	>15%

Коэффициенты α и β являются постоянными величинами и равны 0.71 и 0.77 соответственно. В нашем исследовании все оценки находятся в диапазоне от 3 до 5, так как при значении ниже трех оценка считается неудовлетворительной. Исследование проводилось для просмотра видео на мобильном телефоне.

Здесь рассматривается модель системы, включающая канал передачи видеоблоков и плеер с входным буфером (рис. 1).

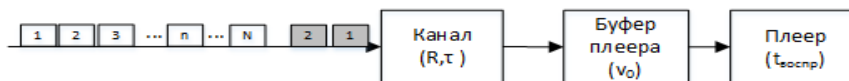


Рис. 1. Моделируемая система

В канал поступает поток видеоблоков, в буфер плеера блоки попадают с некоторой случайной задержкой τ , которая вызвана либо колебаниями качества канала, либо параллельным обслуживанием других потребителей видеоконтента [2-4]. Считается, что в канале установлена скорость передачи R

(Mb/s). Из канала блоки попадают в буфер плеера, который при достижении требуемого объема начальной загрузки v_0 начинает передавать блоки в плеер для воспроизведения. Время воспроизведения одного блока $t_{\text{воспр}}$ полагалось равным одной секунде.

В процессе работы системы буфер может опустеть из-за задержек блоков в канале. В этом случае буфер вновь заполняется до величины v_0 , после чего воспроизведение продолжается.

Задачи исследования

- разработать программу моделирования, позволяющую при заданных значениях скорости, объема начальной загрузки и характеристик задержки оценивать значение $vMOS$;
- исследовать случай, когда задержка распределена экспоненциально с параметром λ , т.е. среднее значение задержки.

Результатом исследования является получение зависимости минимально необходимой скорости передачи в канале от средней задержки в канале, когда требуемое значение $vMOS=3,4,5$. При этом, по значению v_0 следует провести оптимизацию. Это можно записать следующим образом.

$$R_{\min} = \left(\frac{1}{\lambda}, vMOS \right) = \min_{v_0} R \left(\frac{1}{\lambda}, vMOS, v_0 \right) \quad (2)$$

Результаты исследования

На рис. 2 представлена зависимость R_{\min} от средней задержки в канале.

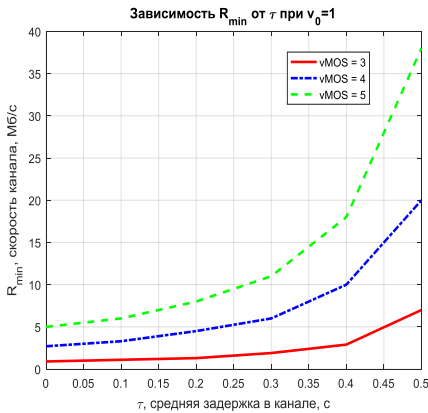


Рис. 2. Зависимость скорости от средней задержки в канале при $v_0 = 1$

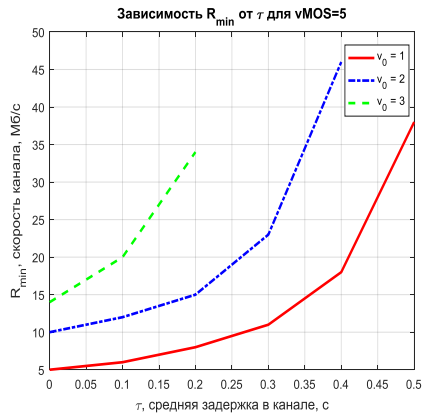


Рис. 3. Зависимость скорости от средней задержки в канале для $vMOS=5$

Для получения данной зависимости объем начальной загрузки (v_0) был выбран равным единице. Заметим, что для увеличения $vMOS$ необходимо увеличивать выделяемую абоненту скорость в канале. При достижении определенной средней задержки в канале получить требуемую оценку $vMOS$ не представляется возможным в интервале приемлемых скоростей [до 50 Мб].

Выбор минимального объема буфера для результата, представленного на рис. 2, является оптимальным (см. рис. 3). Это связано с тем, что при заданных табличных значениях (см. табл. 1) оценки s_I и s_V быстро ухудшаются, то есть, заданы слишком жесткие требования к параметрам качества.

После получения таких результатов в табл. 1 были произведены замены значений, т.е. требования к параметрам были «смягчены» (см. табл. 2).

Таблица 2

Оценка	Смягченные требования к параметрам качества			
	sI		sV	
	Было	Стало	Было	Стало
	Время начальной загрузки (с)		Процент простоя системы	
5	<=0.1	<=1	0	0
4	1	10	5%	15%
3	3	30	10%	30%
<3	>5	>50	>15%	>45%

После «смягчения» требований к параметрам были получены следующие результаты (см. рис. 4).

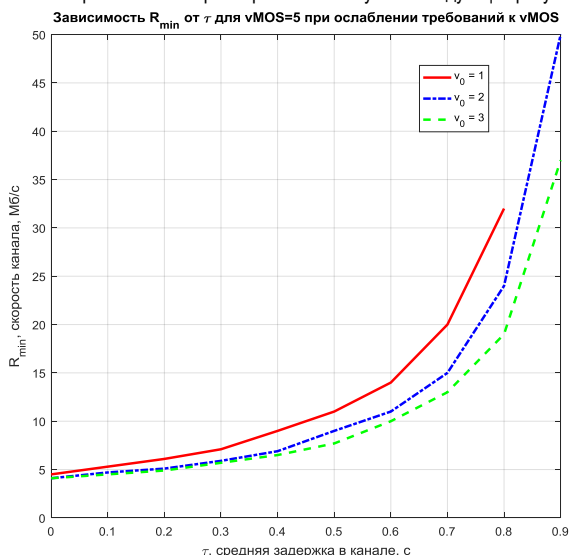


Рис. 4. Зависимость скорости от средней задержки в канале для vMOS=5 при ослабленных требованиях к качеству

Эксперимент проводился для vMOS=5. Как видно, в этом случае минимизация v_0 не является оптимальным решением. Более того, если и дальше продолжать «смягчать» требования, то оптимальное значение v_0 будет увеличиваться. Это связано с тем, что задержки в канале при передаче будут больше, следовательно, в буфере должно будет накопиться большее количество видеоблоков для обеспечения непрерывного воспроизведения.

Выводы

В ходе исследования комплексной оценки качества видео получены следующие результаты.

1. Получены оценки требуемых скоростей передачи, которые необходимы для достижения заданного значения vMOS при известных задержках в канале.
2. Высокие задержки в канале препятствуют достижению желаемого значения vMOS.
3. Оптимальный объем буфера для рекомендуемых значений критериев качества равен единице.

4. При смягчении требований на критерии качества оптимальное значение v_0 будет увеличиваться.

Библиографический список

1. ITU-R Recommendation BT.1788, «Methodology for the subjective assessment of video quality in multimedia applications,» 2007.
2. «Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update», 2016–2021, February 2016.
3. De Pessemier, T., et al. «Quantifying the Influence of Rebuffering Interruptions on the User's Quality of Experience During Mobile Video Watching» IEEE Trans. on Broadcasting, vol. 59, no. 1, March 2013.
4. ITU-T P.800.1, «Mean Opinion Score (MOS) terminology,» July. 2006.

УДК 004.658.2

А. Н. Игошин

студент кафедры проблемно-ориентированных вычислительных комплексов

Г. С. Евсеев

канд. техн. наук, доцент, – научный руководитель

ИСПОЛЬЗОВАНИЕ СТАТИСТИК ДЛЯ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ MSSQL СЕРВЕРА

Статистика для оптимизации запросов – это объекты, содержащие статистические сведения о распределении значений в одном или нескольких столбцах таблицы или индексируемого представления [1]. Оптимизатор запросов использует статистические сведения для оценки количества элементов, то есть числа строк в результатах запроса. Такая оценка количества элементов позволяет оптимизатору запросов создавать оптимальный план запроса. Например, оптимизатор запросов может использовать оценку количества элементов, чтобы выбрать оператор `indexseek` вместо оператора `indexscan`, который потребляет больше ресурсов, и благодаря этому повысить производительность запроса.

Статистика может создаваться различными способами. Автоматически статистика формируется для каждого индекса, который создается в таблице (рис.1). Если включить «Автоматическое создание статистики базы данных», то SQL сервер автоматически создает статистику для неиндексированных столбцов, которые используются в запросах (рис. 2).



Рис. 1. Автоматически созданная статистика для индекса

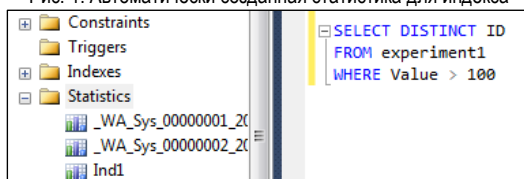


Рис. 2. Автоматически созданная статистика

В некоторых случаях план запроса можно улучшить, создав вручную дополнительную статистику с помощью инструкции `CREATE STATISTICS` (рис. 3).

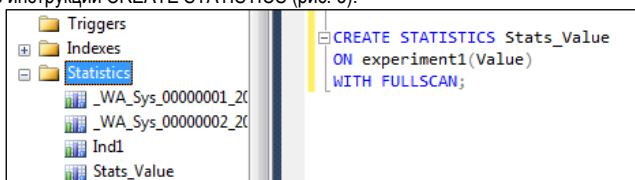


Рис. 3. Статистика, созданная пользователем вручную

Эта дополнительная статистика может фиксировать зависимости, которые не учитывает оптимизатор запросов при создании статистики для индексов или отдельных столбцов [1]. И с помощью 100

статистик, созданных пользователем, можно повысить скорость выполнения запросов. Хорошо отфильтрованная статистика может улучшить план выполнения запроса по сравнению с автоматически созданной статистикой по всей таблице.

Для того чтобы статистику поддерживать, ее необходимо обновлять. Существует два способа обновления статистики: автоматический и ручной. Если включен параметр `AUTO_UPDATE_STATISTICS`, то оптимизатор запросов определяет, когда статистика может оказаться устаревшей, и обновляет ее, если она используется в запросе. Статистика становится устаревшей, после того как операции вставки, обновления, удаления изменяют распределение данных в таблице или представлении. Оптимизатор запросов определяет, когда статистика может стать устаревшей, подсчитывая операции изменения данных с момента последнего обновления статистики и сравнивая количество изменений с каким-то пороговым значением. Пороговое значение основано на количестве строк в таблице. В некоторых случаях можно улучшить план запроса и тем самым повысить скорость выполнения запроса, обновляя статистику чаще, чем она обновляется при включенном параметре `AUTO_UPDATE_STATISTICS`. Для этого необходимо обновлять статистику самостоятельно инструкцией `UPDATE STATISTICS`[2].

Обновление статистики гарантирует, что запросы будут компилироваться с актуальной статистикой [1]. Однако обновление статистики вызывает перекомпиляцию запросов, поэтому нельзя обновлять статистику слишком часто, нужно найти баланс между повышением производительности с помощью улучшения планов запросов и потерей времени на перекомпиляцию запросов. Для различных приложений такой баланс может отличаться и зависит от потребностей того или иного предприятия.

Для того, чтобы отобразить статистику, необходимо выполнить инструкцию `DBCC SHOW_STATISTICS`, которая отображает статистику оптимизации текущего запроса для таблицы. Результат выполнения такой инструкции представлен на рис.4.

	RANGE_HI_KEY	RANGE_ROWS	EQ_ROWS	DISTINCT_RANGE_ROWS	AVG_RANGE_ROWS
1	0	0	321	0	1
2	10	946	726	2	473
3	17	573	556	1	573
4	20	0	778	0	1
5	27	590	546	1	590
6	30	0	753	0	1
7	37	563	561	1	563
8	40	0	773	0	1
9	47	555	555	1	555
10	50	0	719	0	1

Рис. 4. Отображение статистики в MSSQL сервер

На основе статистики может быть построена гистограмма. Гистограмма измеряет частоту появления каждого различающегося значения в наборе данных. Оптимизатор запросов вычисляет гистограмму для значений столбца в первом ключевом столбце объекта статистики, выбирая значения столбцов путем статистической выборки строк или при помощи полного просмотра всех строк в таблице или представлении. Если гистограмма создается на основе выбранного набора строк, то сохраняемые итоговые значения количества строк и количества различающихся значений являются приближенными и не всегда выражаются целыми числами.

Чтобы создать гистограмму, оптимизатор запросов сортирует значения полей, попавших в заданный интервал (высота столбца гистограммы), максимальное количество интервалов в гистограмме в MSSQL сервере равно 200, после чего запоминает границы интервалов и высоту столбцов.

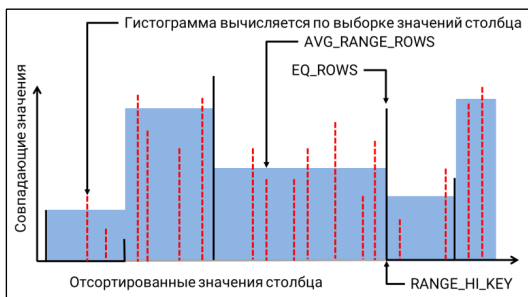


Рис. 5. Гистограмма, построенная с использованием статистики

Наименьшим из отсортированных значений столбца является верхнее граничное значение первого отрезка гистограммы. На рис.5 изображена гистограмма с шестью шагами. Первый отрезок – это область слева от первого верхнего граничного значения. В каждом отрезке гистограммы полужирной линией обозначено верхнее граничное значение столбца для отрезка гистограммы (RANGE_HI_KEY) и количество его вхождений (EQ_ROWS). EQ_ROWS – предполагаемое количество строк, значение столбцов которых равно верхней границе отрезка гистограммы. Закрашенная область слева от RANGE_HI_KEY обозначает диапазон значений столбца и среднее количество вхождений каждого значения столбца (AVG_RANGE_ROWS). В первом отрезке гистограммы значение AVG_RANGE_ROWS всегда равно 0. Пунктирные линии обозначают выбранные значения, которые используются для оценки общего числа различающихся значений (DISTINCT_RANGE_ROWS) и общего числа значений в диапазоне (RANGE_ROWS). Оптимизатор запросов использует RANGE_ROWS и DISTINCT_RANGE_ROWS для вычисления AVG_RANGE_ROWS и не хранит выбранные значения.

Использование статистики для повышения производительности сервера имеет следующие преимущества: уменьшение времени выполнения запросов, упрощение администрирования базы данных, пользовательские статистики позволяют расширить функциональность информационной системы.

Библиографический список

1. *Электронный ресурс* URL: <https://msdn.microsoft.com>[Дата посещения: 20.05.2017].
2. *Бен-Ган И.* MicrosoftSQLServer 2012 / пер. с англ. М.: Эксмо, 2015. 401 с.

УДК 681.5

М.С. Исаков

магистр кафедры вычислительных систем и сетей

В. Н. Елизаров

канд. техн. наук, доцент – научный руководитель

ОБЗОР ИНСТРУМЕНТАЛЬНОЙ ГРАФИЧЕСКОЙ СРЕДЫ ISAGRAF ДЛЯ ПРОГРАММИРОВАНИЯ ПЛК

ISaGRAF – мощная программная технология для разработки встраиваемых приложений основывается на концепции открытой автоматизации и обеспечивает существенное сокращение времени создания изделий и ускоряет их выход на рынок[1]. В данной статье будет рассмотрен ISaGRAF 6, так как он является наиболее актуальным продуктом в текущее время.

Технология ISaGRAF была разработана во Франции компанией CJ International Inc в 1990 году. С 2007 года права принадлежат компании Rockwell Automation.

Rockwell Automation – ведущий американский производитель ПЛК, средств промышленной автоматизации электрооборудования и программного обеспечения (ПО). ПЛК таких производителей, как Siemens, Kooyo, AB, Mitsubishi, в основном ориентированы на языки программирования стандарта МЭК 61131-3. Среда разработки ISaGRAF включает дополнительно функциональные блоки в соответствии со стандартом МЭК 61499 [2].

ISaGRAF поддерживает большое количество производителей ПЛК. В основном поддержку данной технологии оказывают малоизвестные фирмы. Очень популярна технология у отечественных производителей. Список ПЛК российской разработки: Колибри-К1, ПЛК3000, Segnetics SMH2Gi, Symbol-100, Эни-750, Агава и другие [2]. Самые известные контроллеры, поддерживающие ISaGRAF, – это контроллеры I-7000 и I-8000 фирмы ICP DAS.

ISaGRAF – инструментальная графическая среда разработки прикладных программ для программируемых логических контроллеров (ПЛК) на языках стандарта МЭК 61131-3 и МЭК 61499, позволяющая создавать локальные или распределенные системы управления.

Основа технологии ISaGRAF – среда разработки приложений ISaGRAF Workbench и адаптируемая под различные программно-аппаратные платформы исполнительная система ISaGRAF Runtime (Target) [2].

ISaGRAF Workbench основан на открытой технологии плагинов[5]. Плагин – независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования её возможностей [6]. Каждый компонент в ISaGRAF 6 Workbench был разработан с использованием новой технологии автоматизации ISaGRAF, называемой Automation Collaborative Platform (ACP). Как одна из составляющих ACP, среда ISaGRAF 6.0 Workbench основана на открытой технологии подключаемых модулей. В свою очередь, каждый компонент в ISaGRAF 6 Workbench разрабатывается по технологии Единой платформы автоматизации ACP. Следовательно, функциональность ISaGRAF 6 Workbench, расширяющая возможности предыдущих версий, реализуется с помощью плагинов.

Каждый пользователь может создать настраиваемый Workbench, выбрав только те подключаемые модули, которые лучше всего подходят для его применения. Каждый оригинальный производитель ПЛК имеет возможность создавать свои собственные уникальные пакеты для удовлетворения потребностей в своих сегментах рынка. Однако обратим внимание на следующий важный для рынка и многочисленных пользователей факт: в ISaGRAF 6 обеспечивается поддержка работы с исполнительными системами (Target) для ISaGRAF 3 & 5.

В ISaGRAF 6 реализована поддержка нового графического языка SAMA (Scientific Apparatus Makers-Manufactures Association). Язык SAMA представляет собой специальный вид функциональных диаграмм управления, широко применяемых, например, в области энергоснабжения. Эти диаграммы используются для описания и документирования стратегий управления объектами, позволяют легко

представлять такие простые вычислительные функции, как сумматор, верхний/нижний ограничитель и блоки PID-регулирования, строить расширенную функцию управления. Язык SAMA реализован на базе FBD.

В соответствии со стандартом МЭК 61499 ISaGRAF устанавливает обобщенную архитектуру функциональных блоков и предоставляет руководство для их применения в распределенных системах промышленной автоматизации. В таких системах ПО распределено между несколькими физическими устройствами (ПЛК) и несколькими функциональными блоками (ФБ) [3]. Каждый ФБ имеет поток событий (Event flow) и поток данных (Data flow) Соединение ПЛК происходит с помощью коммуникационной сети. Контроллеры же непосредственно руководят работой автоматизированного процесса. Общая диаграмма системы стандарта МЭК 61499 представлена на рис. 1.

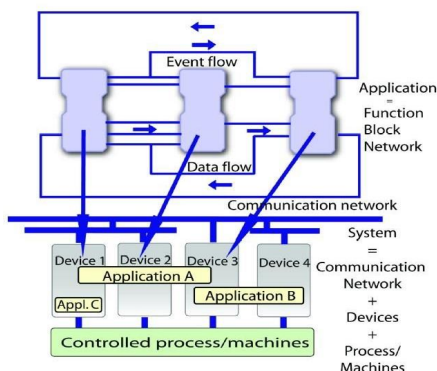


Рис. 1. Общая диаграмма системы в стандарте МЭК61499

Каждый ФБ имеет вход для потока событий и потока данных и соответственные выходы. Принятое событие инициализирует какой-то алгоритм внутри блока, в результате чего могут вырабатываться другие события. Архитектура ФБ представлена на рис. 2.

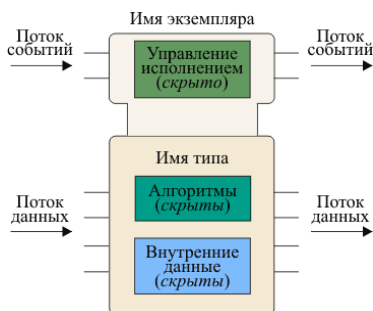


Рис. 2. Архитектура ФБ МЭК 61499

Различают базовые (рис. 3) и композиционные (рис. 4) функциональные блоки МЭК61499. Для определения базового функционального блока надо задать следующие элементы: входные и выходные переменные, входные и выходные события, диаграмму управления выполнением и собственно алгоритм блока.



Рис. 3. Пример базового функционального блока МЭК 61499

Композиционные функциональные блоки – это набор базовых функциональных блоков.

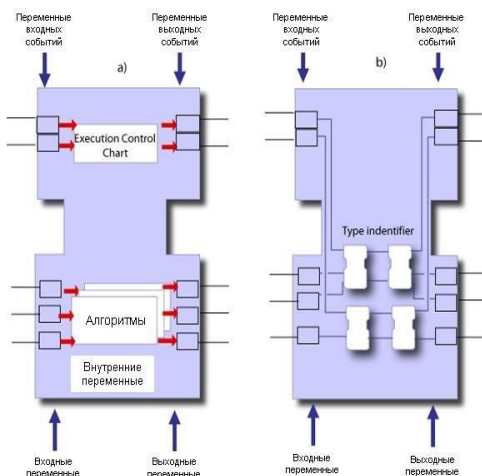


Рис. 4. Пример композиционного функционального блока МЭК 61499

В стандарте определены 18 типов функциональных блоков, некоторые из которых приведены на рис. 5. ISaGRAF поддерживает все типы функциональных блоков МЭК 61499. Рассмотрим несколько блоков. Блок расщепления событий представляет собой ФБ, в котором на входе имеется один вход для события и два для выхода, то есть выполнения этого блока вызывает выполнение двух других блоков. ФБ выбора между двух событий осуществляет выбор одного из двух входящих событий. Для выбора конкретного события используется логический сигнал, с помощью которого и происходит выбор. Такой блок является мультиплексором для событий. ФБ задержки событий необходим, для того чтобы задержать событие. Например, это может быть использовано для синхронизации двух параллельных событий, но одно будет приходить на вход быстрее, что может повлечь за собой неправильное выполнение всего ФБ.

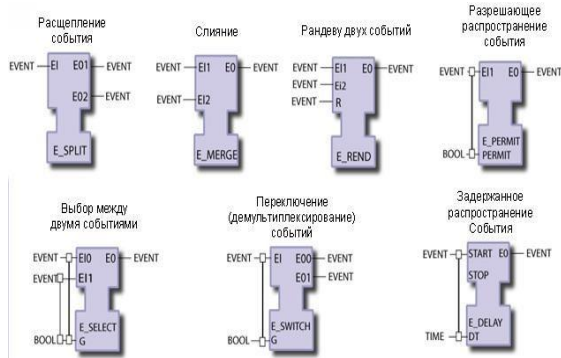


Рис. 5. Фрагмент библиотеки базовых функциональных блоков МЭК 61499

После исследования системы стало понятно, что ISaGRAF можно применять в самых обширных сферах жизнедеятельности. К настоящему времени было реализовано достаточное количество проектов, основанных на этой технологии. В России была создана умная теплица, которая осуществляет автоматизированное управление. А во Франции созданы системы управления высокоскоростным поездом. Кроме этих проектов реализованы и многие другие [4].

ISaGRAF зарекомендовала себя, как универсальная система для автоматизации. Она поддерживает все традиционные технологии автоматизации, а также новые и перспективные. Одним из главных плюсов – это система распространения. ISaGRAF Workbench доступен бесплатно и полностью локализован на русский язык, а это значит, что программировать может научиться любой желающий. Платными являются модули контроллеров, которые приобретаются вместе с ПЛК. Это позволяет изучать ISaGRAF 6 в вузах.

Библиографический список

1. Программирование контроллеров с помощью ISaGRAF [Электронный ресурс]. – Режим доступа <https://colibri.fiord.com/31-programmnoe-obespechenie/271-programmirovanie-kontrollerov/>, свободный.
2. Ризо А.Е., Золотарев С.В. Отечественные производители контроллеров выбирают технологию программирования ISaGRAF 6: функциональность, расширяемость и безопасность, Автоматизация и ИТ в энергетике, № 4, 2016 г.
3. В. Н. Дубинин, В. В. Вяткин. Модели функциональных блоков IEC 61499, их проверка и трансформации в проектировании распределенных систем управления. – Пенза : Издательство ПГУ, 2012. –351 с.
4. ISaGRAF [Электронный ресурс]. – Режим доступа: <http://isagraf.ru>, свободный.
5. IEC 61131 and IEC 61499 Software [Электронный ресурс]. – Режим доступа: <http://www.isagraf.com/index.htm>, свободный.
6. Плагин [Электронный ресурс]. – Режим доступа: <http://ru.wikipedia.org/wiki/Плагин>, свободный.

УДК 629.735.33

Р.В. Ишкинин

студент кафедры вычислительных систем и сетей

Т.Н. Соловьёва

канд. техн. наук, доцент – научный руководитель

ТЕСТОВОЕ ДИАГНОСТИРОВАНИЕ ЛИНЕЙНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

В технических системах, таких как военные средства автоматизации, системы управления авиационными, космическими и подводными комплексами, системы контроля и управления АЭС и т.д., не малую роль играет техническая диагностика динамических систем. Главной задачей диагностирования является определение технического состояния системы, а также корректного выполнения возложенных на систему функций [1].

В настоящее время существует большое количество методов контроля и диагностики динамических систем. Методы диагностирования можно классифицировать по различным признакам. Существует классификация по режимам диагностирования, при которой методы диагностирования делятся на функциональное и тестовые (рис. 1). В первом случае объект диагностирования находится в рабочем состоянии, и на вход подаются соответствующие рабочие сигналы. При тестовом диагностировании на вход объекта подаются специальные тестовые сигналы. В обоих случаях регистрируются выходные сигналы, по которым в дальнейшем делается вывод об исправности объекта [3].



Рис. 1. Классификация методов по режимам диагностирования

Основным недостатком тестового диагностирования является необходимость снятия объекта с эксплуатации, невозможность проведения диагностирования в режиме реального времени. Однако в отличие от функционального диагностирования при тестовом диагностировании имеется возможность подачи любых входных сигналов, следовательно, выявления таких дефектов, которые на ранней стадии развития могут не проявляться в рабочих режимах.

Целью доклада является анализ чувствительности двух методов тестового диагностирования динамических систем. Исследование производится на примере модели сервопривода третьего порядка [2] (рис. 2), который имеет один вход u и два выхода: y_1 и y_2 .

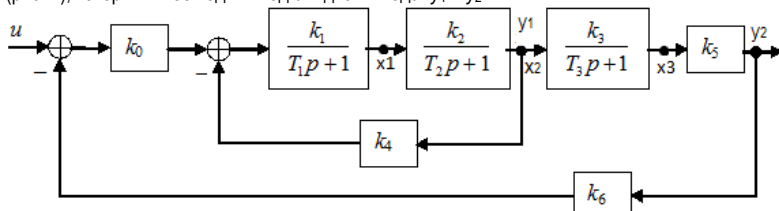


Рис. 2. Схема модели сервопривода третьего порядка

Ниже приведены значения параметров сервопривода:

$$k_0 = 2; k_1 = 4; k_2 = 8,8; k_3 = 3,2; k_4 = 0,02; k_5 = k_6 = 0,016;$$

$$T_1 = 0,5; T_2 = 0,44; T_3 = 0,2.$$

Описание сервопривода в пространстве состояния имеет вид $X' = AX + bu$, $y = cX$, где:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, A = \begin{bmatrix} -\frac{1}{T_1} & -\frac{k_1 k_4}{T_1} & -\frac{k_0 k_1 k_5 k_6}{T_1} \\ \frac{k_2}{T_2} & -\frac{1}{T_2} & 0 \\ 0 & \frac{k_3}{T_3} & -\frac{1}{T_3} \end{bmatrix}, c = \begin{bmatrix} 0 & 0 & k_5 \\ 0 & 1 & 0 \end{bmatrix}, b = \begin{bmatrix} \frac{k_0 k_1}{T_1} \\ 0 \\ 0 \end{bmatrix}.$$

Переменными состояниями являются выходные сигналы отдельных блоков сервопривода рис. 2.

Передаточные функции системы от входного сигнала до выходов y_1 и y_2 имеют вид:

$$Q_1(p) = \frac{k_0 \cdot k_1 \cdot k_2 \cdot (T_3 \cdot p + 1)}{\left(T_1 \cdot p + T_2 \cdot p + T_3 \cdot p + k_1 k_2 k_4 + T_1 \cdot T_2 \cdot p^2 + T_1 \cdot T_3 \cdot p^2 + T_2 \cdot T_3 \cdot p^2 + T_1 \cdot T_2 \cdot T_3 \cdot p^3 + T_3 \cdot k_1 \cdot k_2 \cdot k_4 \cdot p + k_0 \cdot k_1 \cdot k_2 \cdot k_3 \cdot k_5 \cdot k_6 + 1 \right)},$$

$$Q_2(p) = \frac{k_0 \cdot k_1 \cdot k_2 \cdot k_3 \cdot k_5}{\left(T_1 \cdot p + T_2 \cdot p + T_3 \cdot p + k_1 k_2 k_4 + T_1 \cdot T_2 \cdot p^2 + T_1 \cdot T_3 \cdot p^2 + T_2 \cdot T_3 \cdot p^2 + T_1 \cdot T_2 \cdot T_3 \cdot p^3 + T_3 \cdot k_1 \cdot k_2 \cdot k_4 \cdot p + k_0 \cdot k_1 \cdot k_2 \cdot k_3 \cdot k_5 \cdot k_6 + 1 \right)}.$$

Подставим значения сервопривода и получим:

$$Q_1(p) = \frac{u(p)}{y_1(p)} = \frac{320(p+5)}{p^3 + 9,273p^2 + 29,11p + 40,04},$$

$$Q_2(p) = \frac{u(p)}{y_2(p)} = \frac{81,92}{p^3 + 9,273p^2 + 29,11p + 40,04}.$$

Рассмотрим тестовое диагностирование динамических систем методом нулевого режима. Нулевой режим линейной динамической системы – это режим, при котором на систему подается сигнал, отличный от нуля, а выходной сигнал системы при этом равен нулю. Такой эффект достигается за счет подачи сигнала на частоте передаточного нуля и установки специальных начальных условий.

Передаточный нуль системы – это значение частоты $p = \alpha$, при котором передаточная функция системы $Q(p)$ становится равной нулю. Передаточные нули системы являются корнями полинома числителя передаточной функции.

Таким образом, для обеспечения нулевого режима входной сигнал будет иметь вид:

$$u(t) = e^{\alpha t}.$$

Реакция системы на внешнее воздействие состоит из трех составляющих: собственное движение системы, вызванное скачком входного сигнала в начальный момент времени; свободное движение системы из начальных условий; вынужденная составляющая, определяемая входным сигналом.

Подачей входного сигнала на частоте передаточного нуля можно скомпенсировать третью из составляющих выходного сигнала. Для компенсации первой и второй составляющих выходного сигнала системы необходимо установить начальные условия, определяемые формулой

$$X_0 = (\alpha E - A)^{-1} b,$$

где E – единичная матрица.

Для проведения контроля необходимо установить, рассчитанные начальные условия и подать на объект соответствующий входной сигнал. В случае если объект исправен, выходной сигнал будет равен нулю с заданной точностью, иначе – объект неисправен. Для сервопривода (рис. 2) передаточный нуль будет иметь только передаточная функция от входа до выхода y_1 . Диагностирование мето-

дом нулевого режима по сигналу y_2 невозможно. Единственный передаточный нуль сервопривода равен $\alpha = T_3 \cdot 10^{-1} = 5$. Начальные условия для обеспечения нулевого режима сервопривода равны

$$X_0 = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{k_5 k_6} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 3906.3 \end{bmatrix}.$$

Результаты моделирования нулевого режима для исправной и неисправной системы приведены на рис. 3 и рис. 4.



Рис. 3. Моделирование нулевого режима для исправной системы

Для расчета входного сигнала и начальных условий для проведения диагностирования по нулевому режиму написана программа в среде matlab:

```
A = [-1/T1 -(k1*k4)/T1 -(k0*k1*k5*k6)/T1; k2/T2 -1/T2 0; 0 k3/T3 -1/T3];
b = [(k0*k1)/T1;0;0];
c = [0 1 k5];
d = 0;
sys = ss(A,b,c,d);
n = length(A);
Transferring_zero = zero(tf(sys));
X0 = inv(Transferring_zero *eye(n)-A)*b;
R = real(X0)
legend('входной сигнал','выходной сигнал');
m=abs(max(y));
; Z = imag(X0);
r = real(Transferring_zero); z = imag(Transferring_zero);
t=0:0.001:2*n;
u1 = (exp((r*t))).*cos(z*t);
u2 = (exp((r*t))).*sin(z*t);
y = lsim(sys,u1,t,X0);
plot(t,u1,t,y);grid;
fprintf('максимальное выхдное значение по модулю : %d', m);
```

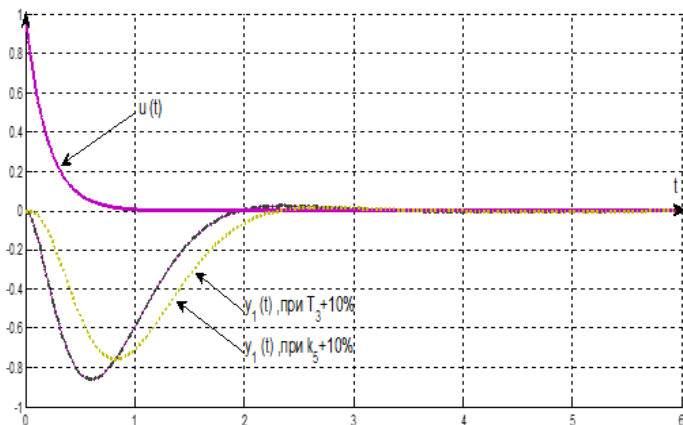


Рис. 4. Моделирование нулевого режима для неисправной системы с поочередным изменением значения каждого параметра на 10%

Для анализа чувствительности нулевого режима к отклонениям параметров сервопривода будем определять максимальное отклонение выходного сигнала от нуля в исправном и неисправном случаях. Для добавления в объект неисправностей будем поочередно увеличивать отклонение на 10%, а затем уменьшать на 10% значения параметров сервопривода. Результаты моделирования приведены в табл. 1.

Таблица 1

Анализ чувствительности метода нулевого режима

Параметры сервопривода	Максимальное отклонение выходного сигнала от нуля		
	при увеличении на 10%	при уменьшении на 10%	для исправного объекта
k_0	$1.97 \cdot 10^{-5}$	$1.61 \cdot 10^{-5}$	$1.7 \cdot 10^{-5}$
K_1	$1.94 \cdot 10^{-5}$	$1.63 \cdot 10^{-5}$	
K_2	$1.79 \cdot 10^{-5}$	$1.63 \cdot 10^{-5}$	
K_3	$1.77 \cdot 10^{-5}$	$1.79 \cdot 10^{-5}$	
K_4	$1.77 \cdot 10^{-5}$	$1.81 \cdot 10^{-5}$	
K_5	$8.59 \cdot 10^{-1}$	$8.6 \cdot 10^{-1}$	
K_6	$8.59 \cdot 10^{-1}$	$8.6 \cdot 10^{-1}$	
T_1	$1.72 \cdot 10^{-5}$	$1.87 \cdot 10^{-5}$	
T_2	$1.73 \cdot 10^{-5}$	$1.86 \cdot 10^{-5}$	
T_3	$7.56 \cdot 10^{-1}$	$7.8 \cdot 10^{-1}$	

Из табл. 1 и рис. 4 видно, что метод нулевого режима позволяет контролировать только те параметры, которые учитываются при расчете входных сигналов и начальных условий, и не чувствителен к другим параметрам системы.

Нулевой режим удобен при контроле системы. Для контроля достаточно просто отслеживать ее выходной сигнал, который должен быть нулевым при исправной системе и отличаться от нуля в случае отклонения хотя бы одного из параметров системы. Схожим преимуществом обладает метод Шрайбера [1].

Диагностирование по методу Шрайбера заключается в том, что на систему n -го порядка подается комплементарный тестовый сигнал в виде $n+1$ прямоугольных импульсов. Первый импульс выводит систему из нулевого состояния, а $n+1$ -й вновь возвращает ее в нуль. Таким образом, тестовый сигнал имеет вид

$$u(t) = \alpha_0 \eta(t) + \alpha_1 \eta(t+h) + \dots + \alpha_n \eta(t+nh), \quad (6)$$

где $\eta(t) = 1(t) - 1(t+h)$ – импульс конечной длительности h ; α_i – коэффициент комплементарного сигнала, который необходимо рассчитать; n – порядок системы.

Для нахождения тестового сигнала необходимо рассчитать характеристический полином матрицы $\Phi = e^{Ah}$. Коэффициенты этого полинома будут являться коэффициентами комплементарного сигнала. Для сервопривода (см. рис. 2) коэффициенты комплементарного сигнала (рис. 6–8) равны: $\alpha_0 = 1$, $\alpha_1 = 0.067862$, $\alpha_2 = 0.015106$, $\alpha_3 = -9.3954 \cdot 10^{-5}$, при $h=1$.

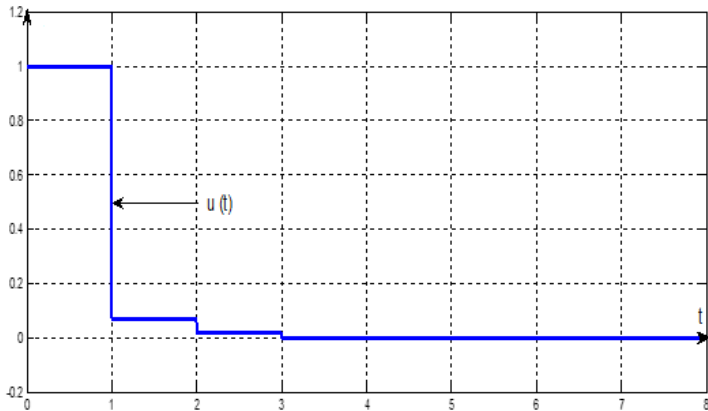


Рис. 6. Моделирование входного сигнала для метода Шрайбера

Таким образом, для того чтобы провести контроль заданной системы необходимо подать на ее вход сформированный комплементарный сигнал. Если объект полностью исправен, то по окончании входного воздействия система перейдет в нулевое состояние.

Для анализа чувствительности также как и в методе нулевого режима будем по очереди изменять параметры системы. Время моделирования будет разделено пополам, на первом этапе будет задаваться входной сигнал, равный тестовому сигналу Шрайбера, на втором этапе будет подаваться нулевой входной сигнал. Чувствительность будет определяться отношением максимального по модулю значения выходного сигнала на втором этапе моделирования, к максимальному по модулю значению выходного сигнала на первом этапе моделирования.

Для автоматизации анализа чувствительности была написана программа в среде matlab :

```
sys = ss(A, b, c, d);
dlin_imp = 1;
EXPM_A = expm(A*dlin_imp);
HP_EXPA = poly(EXPM_A);
Form = length(HP_EXPA);
```

```

t = 0:0.0001:2*Form*dlin_imp;
u=0;
for Hp=1:Form
    u = u+(HP_EXPA(Hp)*(stepfun(t,(Hp-1)*dlin_imp)-stepfun(t,(Hp-1)*dlin_imp+dlin_imp)));
end
y = lsim(sys,u,t);
plot(t,u);grid;figure;
plot(t,y);grid;figure;
plot(t,u,t,y);grid;
legend('входной сигнал','выходной сигнал');
kon = length(t);
nach = length(0:0.0001:Form*dlin_imp);
mid = kon-nach;
k1 = max(abs(y(1:mid)));
k2 = max(abs(y(nach:kon)));
Kof = k2/k1;
fprintf('максимальное выходное значение по модулю : %d', Kof);

```

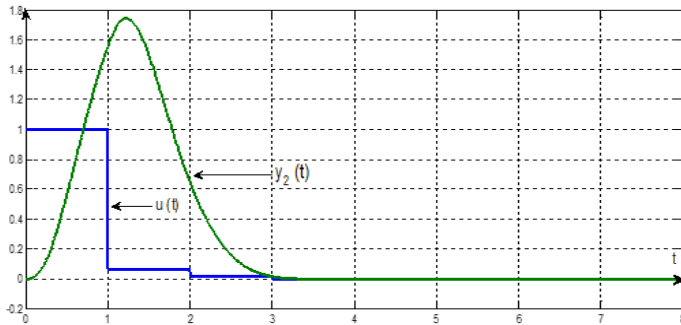


Рис. 7. Моделирование метода Шрайбердля исправной системы

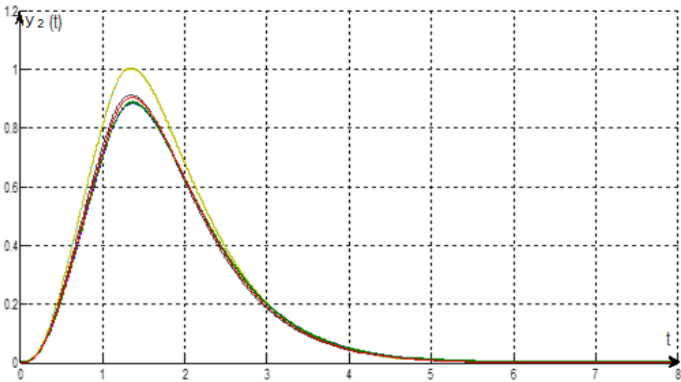


Рис. 8. Моделирование метода Шрайбера для неисправной системы с поочередным изменением каждого значения параметра на 10%

Из рис. 8 и табл. 2 видно, что при отклонении каждого из параметров выходной сигнал на втором этапе моделирования(при $4 < t < 8$) отличен от нуля.

Таблица 2

параметров сервопривода	Анализ чувствительности метода Шрайбера		
	Максимальное отклонение выходного сигнала от нуля		для исправного объекта
	при увеличении на 10%	при уменьшении на 10%	
k0	$4.67 \cdot 10^{-2}$	$4.67 \cdot 10^{-2}$	$2.00007 \cdot 10^{-15}$
K1	$4.67 \cdot 10^{-2}$	$4.67 \cdot 10^{-2}$	
K2	$4.67 \cdot 10^{-2}$	$4.673 \cdot 10^{-2}$	
K3	$4.67 \cdot 10^{-2}$	$4.67 \cdot 10^{-2}$	
K4	$4.67 \cdot 10^{-2}$	$4.67 \cdot 10^{-2}$	
K5	$4.67 \cdot 10^{-2}$	$4.67 \cdot 10^{-2}$	
K6	$4.67 \cdot 10^{-2}$	$4.67 \cdot 10^{-2}$	
T1	$5.63 \cdot 10^{-2}$	$3.71 \cdot 10^{-2}$	
T2	$5.46 \cdot 10^{-2}$	$3.86 \cdot 10^{-2}$	
T3	$4.86 \cdot 10^{-2}$	$4.36 \cdot 10^{-2}$	

Из табл. 2 видно, что максимальной чувствительностью метод Шрайбера обладает к увеличению параметров T1, T2, T3.

В ходе работы были исследованы два метода диагностирования: метод нулевого режима и метод Шрайбера. Исследование проводилось на примере модели сервопривода третьего порядка, в пакете matlab. Для каждого метода были найдены тестовые входные сигналы для реализации диагностирования, а также был произведен анализ чувствительности методов к отклонению параметров сервопривода. Для данного объекта рекомендуется использовать метод Шрайбера, так как он в отличие от метода нулевого режима, обладает высокой чувствительностью к отклонению всех параметров объекта.

Библиографический список

1. Бритов Г.С., Мироновский Л.А. Расчет тестового режима линейных систем управления // Приборы и системы. Управление, контроль, диагностика. 2006. №11. С. 44-49.
2. Мироновский Л.А. Моделирование линейных систем: учеб. пособие. СПб.: ГУАП, 2009. 244 с.
3. Мироновский Л.А. Функциональное диагностирование динамических систем: научное издание. СПб.: ГУАП, 1998. 256 с.

УДК 004.413.5

Е. А. Капранова, В. А. Харланов

студенты кафедры вычислительных систем и сетей

М. Б. Игнатьев

д-р техн. наук, профессор;

А. В. Аксенов

ст. преподаватель – научные руководители

ПРОГРАМНОЕ ОБЕСПЕЧЕНИЕ УМНОГО АВТОМОБИЛЯ

Умный автомобиль – это автомобиль с установленными подсистемами для обеспечения безопасности на дороге. Умные автомобили могут быть автономными (самоуправляемыми) и полуавтономными; обладающими продвинутой бортовой системой навигации; экологически чистые электромобили и автомобили с альтернативными системами питания. Все чаще под умными автомобилями подразумеваются именно автомобили, которым не нужен водитель, или те, которые могут упростить вождение. Но для того, чтобы автомобиль смог сделать движение на дороге более безопасной и помочь водителю управлять транспортным средством, он должен иметь специальное программное обеспечение (ПО). Поэтому главной нашей задачей было разобраться, имея какие ресурсы, умный автомобиль сможет повысить безопасность на дороге. Мы преследовали основную цель: проанализировать, с помощью каких систем автомобиля можно предотвратить аварии на дорогах и снизить смертность, комфортно передвигаться по городу, а также показать возможность управлять автомобилем людям с ограниченными способностями.

Данная статья – обзор подсистем умных автомобилей. Далее речь пойдет о следующих подсистемах:

- адаптивный круиз-контроль, который предназначен для автоматического управления скоростью движения автомобиля;
- система обнаружения пешеходов, предназначенная для предотвращения столкновения с людьми;
- парковочные системы, облегчающие процесс парковки;
- система помощи движению по полосе, которая помогает водителю придерживаться выбранной полосы.

Адаптивный круиз-контроль (Adaptive Cruise Control, ACC) предназначен для автоматического управления скоростью движения автомобиля. Он является дальнейшим развитием системы круиз-контроля, которая поддерживает заданную постоянную скорость движения [1]. Система адаптивного круиз-контроля включает датчик расстояния, блок управления и исполнительные устройства. Датчик расстояния служит для измерения скорости и расстояния до впереди идущего автомобиля. В качестве датчика расстояния используются радары или лидары. Датчик расстояния устанавливается на переднем бампере или решетке радиатора автомобиля. Радиус действия датчика составляет порядка 150 м. В последних разработках адаптивного-круиз-контроля используется датчики расстояния короткого и длинного диапазонов. Датчик короткого диапазона обеспечивает замедление автомобиля до полной остановки. Датчик длинного диапазона – до 30 км/ч. Это расширяет функциональные возможности системы, позволяя использовать автомобиль при движении с малой скоростью на небольшой дистанции (например, при движении в «пробках»). С целью повышения безопасности автомобиля отдельные конструкции адаптивного круиз-контроля могут быть дополнены системами превентивной безопасности, экстренного торможения, GPS-навигации. Адаптивный круиз-контроль служит технической основой разрабатываемых систем автоматического управления автомобилем. Электронный блок управления принимает сигналы от датчиков расстояния, а также входную информацию от других систем, с помощью которых определяется:

- скорость и дистанция до впереди идущего автомобиля;
- скорость управляемого автомобиля;
- угол поворота рулевого колеса;

- боковое ускорение;
- радиус кривой.

Система обнаружения пешеходов предназначена для предотвращения столкновения с пешеходами. Система распознает людей возле автомобиля, автоматически замедляет автомобиль, снижает силу удара и даже избегает столкновения. Применение системы позволяет на 20% сократить смертность пешеходов при дорожно-транспортном происшествии и на 30% снизить риск тяжелых травм. В системе обнаружения пешеходов реализованы следующие взаимосвязанные функции:

- обнаружение пешеходов;
- предупреждение об опасности столкновения;
- автоматическое торможение.

Для обнаружения пешеходов используется видекамера и радар, которые эффективно работают на расстоянии до 40 м. Если пешеход обнаружен видекамерой и результат подтвержден радаром, система отслеживает движение пешехода, прогнозирует его дальнейшее перемещение и оценивает вероятность столкновения с автомобилем (рис. 1). Результаты обнаружения выводятся на экран мультимедийной системы. Система также реагирует на транспортные средства, которые стоят на месте или движутся в попутном направлении. Если системы установила, что при текущем характере движения автомобиля столкновение с пешеходом неизбежно, посылается звуковое предупреждение водителю. Далее система оценивает реакцию водителя на предупреждение – изменение характера движения автомобиля (торможение, изменение направления движения). Если реакция отсутствует, система обнаружения пешеходов автоматически доводит автомобиль до остановки. В этом качестве система обнаружения пешеходов является производной системы автоматического экстренного торможения.

Парковочная система (парктроник) является вспомогательной системой активной безопасности автомобиля, облегчающей процесс парковки автомобиля. Наибольшая эффективность от применения парковочной системы реализуется при движении автомобиля задним ходом, в темное время суток, при сильной тонировке стекол, а также в стесненных условиях (парковка, гараж и др.). Парковочные системы можно условно разделить на две большие группы – пассивные и активные. Пассивные парковочные системы представляют только необходимую для парковки информацию, при этом управление автомобилем осуществляется водителем. Активные парковочные системы обеспечивают парковку автомобиля в автоматическом или автоматизированном (автоматически выполняются отдельные функции) режиме. В качестве датчиков парковки используются ультразвуковые датчики. Обычно устанавливается 4-8 датчиков парковки, из которых 4 задних датчика и, при необходимости, 2-4 передних датчика. Датчики устанавливаются, как правило, в переднем и заднем бампере автомобиля. Они посылают сигнал ультразвуковой частоты (порядка 40 кГц) и принимают его отражение от препятствия. Чем меньше время возвращения сигнала, тем ближе находится препятствие. Эффективная работа датчика парковки осуществляется на расстоянии 0,25-1,8 м от препятствия. Электрические сигналы от датчиков поступают в электронный блок управления. В зависимости от величины сигналов электронный блок формирует информацию для устройства индикации. Устройство индикации (индикаторное устройство) служит для отображения информации о приближении к препятствию и предупреждения водителя об опасности (рис. 2). В устройствах применяются следующие виды индикации: звуковая; световая; цифровая; оптическая.

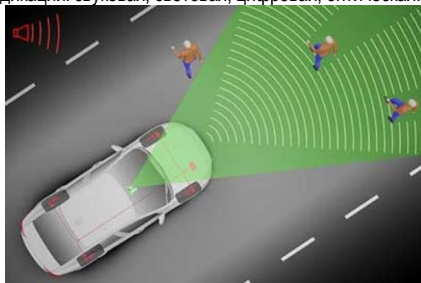


Рис. 1. Принцип обнаружения пешеходов

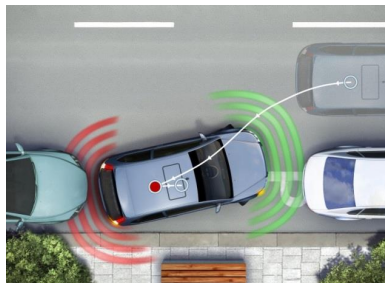


Рис. 2. Принцип работы парковочной системы

С целью улучшения заднего обзора и облегчения движения и парковки задним ходом в автомобилях может устанавливаться камера заднего вида. В настоящее время это одна из востребованных опций, предлагаемых при покупке автомобиля. Видеокамера снимает происходящее за автомобилем и передает на информационный дисплей. Помимо этого, на информационный дисплей может выводиться рекомендуемое направление движения. Камера заднего вида является одним из элементов системы кругового обзора. Включение камеры производится при включении передачи заднего хода. По своей сути, камера заднего вида является разновидностью пассивной парковочной системы [2].

Система помощи движению по полосе (система удержания полосы движения) помогает водителю придерживаться выбранной полосы движения и тем самым, предотвращать аварийные ситуации. Система эффективна при движении по автомагистралям и обустроенным федеральным дорогам, т.е. там, где имеется качественная дорожная разметка. Различают два вида систем помощи движения по полосе: пассивные и активные. Пассивная система предупреждает водителя об отклонении от выбранной полосы движения. Активная система наряду с предупреждением производит корректировку траектории движения. Система помощи движению по полосе является электронной системой и включает клавишу управления, видеокамеру, блок управления и исполнительные механизмы. С помощью клавиши управления производится включение системы. Видеокамера производит запись изображения на определенном расстоянии от автомобиля и его оцифровку. В системе используется монохромная камера, которая распознает линии разметки как резкое изменение градации серого. Камера объединена с блоком управления. Объединенный блок располагается на лобовом стекле за зеркалом заднего вида. Исполнительными устройствами системы помощи движения по полосе являются контрольная лампа, звуковой сигнал, вибромотор на рулевом колесе, нагревательный элемент лобового стекла, электродвигатель электромеханического усилителя руля. Корректировка траектории движения осуществляется принудительным подруливанием системы рулевого управления с помощью электромеханического усилителя руля подтормаживанием колес с одной стороны автомобиля. Во время работы активной системы помощи движения по полосе реализуются следующие основные функции:

- распознавание траектории полосы движения;
- визуальное информирование о работе системы;
- корректировка траектории движения;
- предупреждение водителя.

Обстановка перед автомобилем проецируется на светочувствительную матрицу камеры и преобразуется в черно-белое изображение, которое анализируется электронным блоком управления [3].

Все эти системы призваны сократить высокий уровень смертности на дорогах, предотвратить страшные аварии и избежать ошибок водителей, которые были сделаны по невнимательности. Так же умный автомобиль дает шанс людям с ограниченными возможностями комфортно передвигаться по городу.

Библиографический список

1. *Суслинников А. А.* Системы современного автомобиля, <http://systemsauto.ru/active/acc.html>
2. Nick Pitchik. Volkswagen Technical Site, http://vwts.ru/pps/pps_620_syst_adapt_kruiz_control_acc_rus.pdf
3. *Игнатьев М.Б.* Кибернетическая картина мира: учеб. пособие. – СПб.: ГУАП, 2010. – 416 с

УДК 004.655.3

А. С. Каратаев

адрес кафедры вычислительных систем и программирования

А. В. Фомин

канд. техн. наук, доцент – научный руководитель

РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ МОБИЛЬНОЙ БАЗЫ ДАННЫХ С ПОМОЩЬЮ ТЕХНОЛОГИИ REALM

Сегодня на мобильные устройства возлагается все больше и больше различных задач, а использование высокоскоростных каналов мобильной связи для обмена данными между устройствами уже стало нормой. Большой объем передаваемой и хранимой информации стал причиной появления специализированных мобильных хранилищ данных, причем объем этих хранилищ может колебаться от нескольких мегабайт с метаданной до десятков гигабайт медиа контента. Обычно, мобильные операционные системы поддерживают сразу несколько видов хранилищ, отличных по своим принципам работы и назначению, но в большинстве случаев разработчикам мобильных приложений приходится иметь дело с мобильными системами управления базами данных, рассказать об одной из таких систем и стало целью моего доклада.

Realm это кросс-платформенная мобильная система управления базами данных на мобильных устройствах. Данная технология предполагает использование одних и тех же файлов данных для все поддерживаемых мобильных операционных систем. Realm создавался на замену SQLite и что уже де-факто, успел стать стандартом в индустрии разработки мобильных приложений. Технология обладает высокой производительностью и масштабируемостью, поэтому является хорошим выбором для мобильных проектов, работающих с большим количеством данных, так как превосходит по производительности выполнения базовых операций большинство своих ближайших конкурентов, таких как SQLite, CoreData, FMDB[3]. На текущий момент технология имеет отличный рейтинг на GitHub, лицензируется по Apache License 2.0 и распространяется сразу для нескольких языков, включающих Swift, Objective-C и Java.

Технология Realm обладает простым, лаконичным синтаксисом и богатыми функциональными возможностями, что довольно выгодно отличает ее от подобных конкурирующих технологий, например таких как CoreData. основополагающей концепцией Realm является легковесный контейнер объектов с одноименным названием. Как и в обычных базах данных, данные, находящиеся в Realm, могут быть запрошены и отфильтрованы, но в отличие от традиционных СУБД объекты в Realm являются интерактивными, что позволяет отслеживать изменения данных в хранилище, не отправляя запрос повторно [1]. Вы всегда будете работать с «живыми» объектами, т.е. если один поток или процесс модифицировал объект, другие потоки или процессы могут быть непосредственно оповещены, объект всегда остается синхронизированным со своим хранилищем[2].

Структура хранилища в Realm определяется схемой базы данных, состоящей из таблиц, поля которых могут содержать множественные типы данных, в том числе типы определяемые пользовательскими моделями данных, таким образом структура базы данных в Realm является объектно-ориентированным графом. Что также немаловажно для мобильного разработчика, Realm распространяется с помощью систем управления зависимостями, таких как CocoaPods или Carthage, что избавляет от необходимости постоянно следить за обновлениями установленного в приложение пакета. Так же, Realm имеет браузер для чтения и редактирования файлов базы данных, что очень сильно упрощает разработчику отладку приложения на ранних этапах разработки. Для визуального проектирования схемы базы данных можно воспользоваться специальным плагином для среды разработки XCode.

Модели в Realm представлены обычными классами со своими свойствами. Для того чтобы определить пользовательскую модель данных, необходимо унаследовать Object, который является базовым классом для всех объектов моделей в Realm, и определить необходимые поля данных со-

гласно спроектированной схеме вашей базы данных. Realm проверяет все модели при запуске вашего приложения, если какая-то из моделей вашей схемы окажется нерабочей, сработает предупреждение библиотеки Realm, которое прервет дальнейшее выполнение вашего приложения, поэтому важно, чтобы все поля моделей имели значения по умолчанию или были опциональными [2].

В заключение стоит отметить что в своей дипломной работе я использовал технологию Realm для проектирования и разработки базы данных приложения для изучения иностранных языков. Особенностью разработанной БД является функция дедупликации данных, поддерживаемая на уровне физической схемы базы данных. Данная схема БД была портирована в рабочий проект из прототипа, где использовалась технология Core Data. Благодаря мощному функционалу, простому и лаконичному синтаксису, технология Realm позволила минимизировать время внедрения базы данных в проект.

Библиографический список

1. *Томас Коннолли, Каролин Бегг*. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Издательство Вильямс, 2017. 1440 с.
2. Realm Swift 2.7.0 [электронный ресурс] URL: <https://realm.io/docs/swift/latest/> (15.03/2017)
3. Введение в Realm [электронный ресурс] URL: <https://news.realm.io/news/introducing-realm/> (15.03/2017)

УДК 004.65

Г.В. Катамадзе

магистрант кафедры проблемно-ориентированных вычислительных комплексов

Г.С. Евсеев

канд. техн. наук, доцент – научный руководитель

ОРГАНИЗАЦИЯ УПРАВЛЕНИЯ ИНДЕКСАМИ В MSSQLSERVER

Зачастую при разработке информационных систем, основой которого является ядро «База данных», перед разработчиком возникают задачи оптимизации. Одна из таких задач – максимизация производительности *SQL-сервера*. Одним из важнейших путей для достижения этой цели является использование индексов. Индекс ускоряет процесс запроса, предоставляя быстрый доступ к строкам данных в таблице, аналогично тому, как указатель в книге помогает быстро найти необходимую информацию.

Индексы создаются для столбцов, они представляют собой путь для быстрого поиска интересующих нас данных в этих столбцах. Например, если создавать индекс, а затем послать запрос на *SQL-сервер* на поиск строки, то сначала сервер будет искать значение индекса, а затем по этому индексу найдет строку с данными. Без использования индексов запрос будет выполнять полное сканирование (просмотр) всех строк таблицы, что будет являться ресурсозатратной операцией и скажется на производительности *SQL-сервера* [2].

Индексы состоят из набора страниц, которые представлены в виде дерева (сбалансированное дерево). При формировании запроса по индексированному столбцу, подсистема запроса начинает поиск с вершины иерархии корневого уровня или узла (*Rootlevel* или *RootNode*), постепенно двигаясь к конечным узлам нижнего уровня (*LeafNodes* или *Leaflevel*) через промежуточные узлы. Каждый промежуточный узел (уровень) может содержать детальную информацию о данных. Нижний уровень содержит либо указатель на строку с данными, либо сами данные, в зависимости от типа индекса: кластеризованный индекс; некластеризованный индекс. Оба этих индекса по своей структуре представляют B-дерево [3].

Кластеризованный индекс (рис. 1) хранит реальные строки данных в узлах индекса. Это значит, что строка, связанная со значением ключевого поля, будет храниться в самом индексе. Одним из преимуществ использования кластеризованного индекса является то, что все значения отсортированы либо по возрастанию, либо по убыванию.

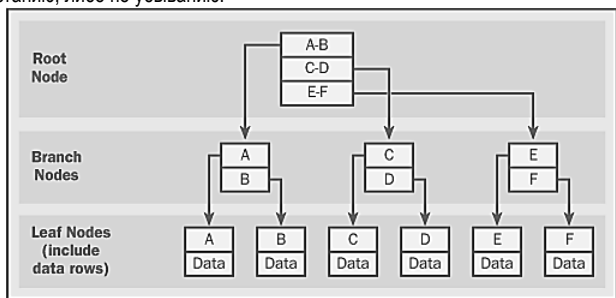


Рис. 1. Кластеризованный индекс в виде B-дерева

Некластеризованный индекс в отличие от кластеризованного содержит только ключевые столбцы, где определен этот индекс и также содержит указатели на строки с реальными данными. Узлы в структуре некластеризованного индекса могут содержать информацию о местоположении строк

данных. Например, если в таблице присутствует только некластеризованный индекс, то на узлы этого индекса хранят идентификаторы строк (*RowID*) (рис. 2).

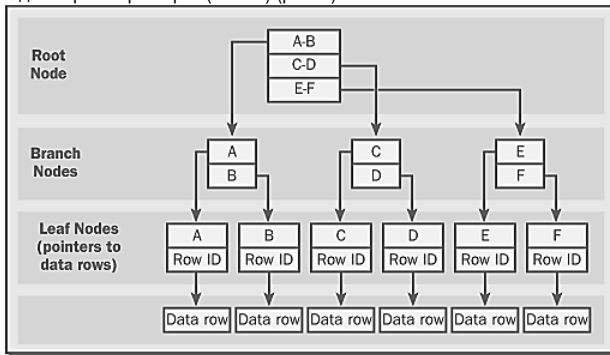


Рис. 2. Некластеризованный индекс по таблице, не имеющей кластеризованного индекса в виде В-дерева

Каждый *RowID* строки указывает реальную строку данных в таблице. Таким образом, обеспечивается быстрый доступ к этим данным, с указанием точного местоположения.

Тестовый пример:

Исходная таблица состоит из трех полей: «*ID*» – порядковый номер записи в таблице, «*Name*» – текстовое поле, «*Age*» – числовое поле.

Исходная таблица заполнялась случайно выбранными данными в процедуре (рис. 3).

Проведем несколько экспериментов для разного объема таблицы:

- запрос на выборку «*SELECT*» для таблицы объемом в (128 тыс., 256 тыс., 512 тыс., 768 тыс. записей) по одному индексированному полю, двум, трем;
- запрос на добавление «*INSERT*» для таблицы объемом в (128 тыс., 256 тыс., 512 тыс., 768 тыс. записей) по одному индексированному полю, двум, трем.

```

use test
GO
CREATE PROCEDURE ranf (@i int)
AS
  Declare @j int;
  Declare @Age int;
  Declare @case nvarchar(50);
  set @j=0;
  while (@j<@i) begin
    set @Age = round ((20+RAND()*(60-20)),0);
    set @case= round ((1+RAND()*(10-1)),0);
    set @case =
    case
      when (@case=1) then 'Иван'
      when (@case=2) then 'Петр'
      when (@case=3) then 'Ольга'
      when (@case=4) then 'Ирина'
      when (@case=5) then 'Анна'
      when (@case=6) then 'Михаил'
      when (@case=7) then 'Федор'
      when (@case=8) then 'Инга'
      when (@case=9) then 'Матвей'
      when (@case=10) then 'Сергей'
      else 'Геннадий'
    end;
    INSERT INTO dbo.Userg values (@case,@age)
    set @j=@j+1;
  end;
GO

```

Рис. 3. Пример процедуры заполнения таблицы случайными значениями

Каждый запрос посылался на сервер десять раз для каждого объема таблицы. Для каждого эксперимента засекалось время выполнения запроса. Из десяти значений высчитывалось арифметическое среднее, результаты которого представлены на графике (рис. 4, 5).

Запрос на выборку «SELECT»

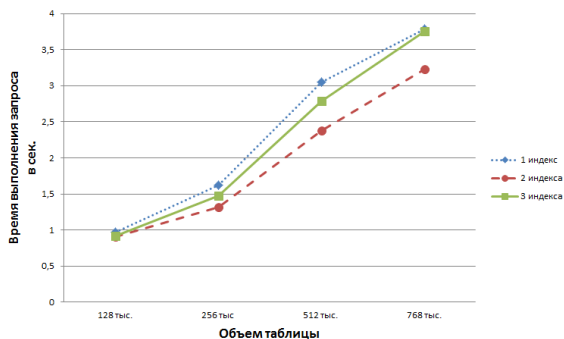


Рис. 4. График зависимости времени выполнения запроса типа «SELECT» от объема таблицы

Запрос на добавление «INSERT»

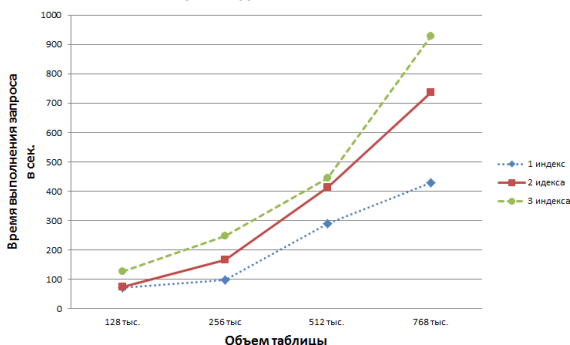


Рис. 5. График зависимости времени выполнения запроса типа «INSERT» к таблице

В заключение можно отметить, что эффективность индексов будет зависеть от того, как «аккуратно» они будут спроектированы. Поскольку большое количество индексов могут занимать значительное пространство на диске. Также необходимо учитывать, как часто данные в таблицах будут подвергаться модификации (обновлению, удалению, вставки), так как когда изменяются данные, индексный массив тоже подвергается модификации, что может снизить производительность системы. В таких таблицах нужно использовать, как можно меньше индексов. Не стоит индексировать небольшие таблицы, порой намного эффективнее выполнить сканирование таблицы, так как дополнительные нагрузки на сервер, возникающие при поддержке индекса, сводят на «нет» преимущества их использования.

Библиографический список:

1. Бондарь А.Г. Microsoft SQL Server 2014. СПб.: БХВ-Петербург, 2015. 592 с.
2. 14 вопросов про индексы в SQLServer [Электронный ресурс], — Режим доступа: <https://habrahabr.ru/post/247373/>, свободный. — Загл. с экрана.
3. В-дерево [Электронный ресурс], — Режим доступа: <https://ru.wikipedia.org/wiki/B-%D0%B4%D0%B5%D1%80%D0%B5%D0%B2%D0%BE>, свободный. — Загл. с экрана.

УДК

Е. К. Кернер

магистрант кафедры компьютерных технологий и программной инженерии

В. В. Бураков

профессор, д-р техн. наук, доцент – научный руководитель

ПРОГРАММНАЯ ОПТИМИЗАЦИЯ КОДА НА ОСНОВЕ МЕТРИК

Производство программных средств дошло до такого уровня, при котором появилась необходимость использовать инженерные методы для:

- Оценки результата проектирования на всех этапах жизненного цикла.
- Отслеживания количественной оценки показателей качества.
- Оценки рисков.
- Возможность применения существующих компонентов для понижения цены производства нового программного продукта.

Повышение качества является ключевой задачей программирования, для решения которой были разработаны методы формирования требований к качеству, методы выбора и улучшения моделей измерения количественных показателей качества, методы измерения метрик показателей качества на каждом этапе жизненного цикла программного продукта [1].

Качество программного обеспечения – предмет стандартизации. Стандарт ISO/IEC 12207 определил не только основные процессы жизненного цикла разработки программного средства, но и организационные и дополнительные процессы, которые регламентируют инженерию, планирования и управления качеством программного обеспечения [2].

Введем понятие метрики и оптимизации:

Определение 1. Метрики кода – это показатели, с помощью которых можно количественно оценить качество исходного кода программного средства. Благодаря метрикам можно лучше разобраться с кодом, определить состояние проекта на данный момент времени, определить дефектные/узкие места, определить порядок задач для рефакторинга [3].

Определение 2. Оптимизация – это различные преобразования кода, выполняемые для повышения его характеристик и эффективности. К целям оптимизации можно отнести [4]:

- Повышение сопровождаемости.
- Уменьшение объема кода (длины кода).
- Уменьшение объема использования оперативной памяти.
- Уменьшение количества операций ввода вывода.
- Уменьшение размера программ.
- Ускорение времени выполнения функций программы.

Основное из требований, которые обычно предъявляется к методам оптимизации – программа после выполнения оптимизации над ней должна иметь тот же выходной результат и побочные эффекты при том же наборе входных данных, что и неоптимизированная версия программы. При этом, данное требование можно исключить, если выигрыш оптимизированного программного средства быть сочтен более важным, чем последствия от изменения его поведения.

На текущий момент в открытом доступе не существует систем, которые могут выполнять оптимизацию кода в целях улучшения значений заданных метрик, однако есть отдельное программное обеспечение, используемое для:

- расчета метрик;
- выполнений оптимизации кода и рефакторинга;

Для решения задачи оптимизации кода и расчета метрик необходимо выполнять анализ кода. В работе будут использоваться существующие методы анализа: статистический и динамический.

Определение 3. Динамический анализ кода – представляет собой метод анализа программного кода, осуществляемый при непосредственном исполнении программы [5].

При помощи динамического анализа кода могут быть вычислены следующие метрики:

— Программные ошибки – такие как деление на ноль, разыменование нулевого указателя, утечки памяти;

— Наличие в программе уязвимостей.

— Используемые ресурсы – время выполнения программы в целом или ее отдельных модулей, количество внешних запросов (например, к базе данных), количество используемой оперативной памяти и других ресурсов;

Динамический анализ находит применение в тех областях, где главным критерием является время отклика программы, надежность, или потребляемые ею ресурсы. В таких областях любая допущенная ошибка может оказаться критической.

Определение 4. Статический анализ кода – представляет собой метод анализа программного кода, осуществляемый без реального выполнения программы [6].

Задачи, которые решаются статическими анализаторами можно разделить на 3 группы:

— Выявление ошибок в программном коде.

— Советы по стилю кода. Статические анализаторы осуществляет проверку соответствия программного кода определенным стандартам оформления.

— Подсчет метрик. Метрика программного обеспечения – это мера, позволяющая получить численное значение некоторого свойства программного обеспечения или его спецификаций. Существует большое количество разнообразных метрик, которые можно подсчитать, используя те ли иные инструменты.

Рассмотрим возможные преобразования над кодом [7]:

— *иницирование переменных – стремится к присваиванию значений при объявлении.*

— уменьшение числа переменных;

— оптимизация типов данных;

— *удаление излишних операторов присваивания;*

— *отождествление переменных;*

— *удаление тождественных операторов;*

— *устранение невыполняемого/недостижимого кода;*

— *сокращение ввода-вывода;*

— *извлечение функций;*

— *сокращение функций;*

— *подмена арифметических операций;*

— улучшение работы с циклами;

В качестве результата исследовательской работы предполагается получение программного средства на вход которого поступает:

— входной оптимизируемый текст;

— множество допустимых преобразований;

— множество проверяемых метрик;

— настройки выходного результата;

На выходе получается результат преобразований с целью улучшения заданных метрик, полученный в желаемой форме, например, в множестве советов о преобразованиях или как автоматический оптимизированный код.

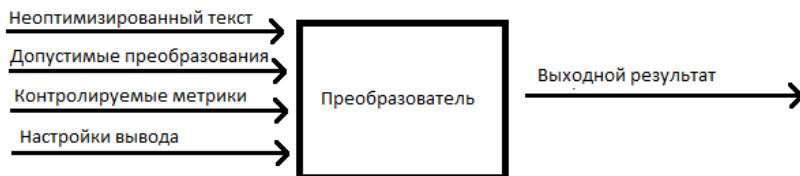


Рис. 1. Общая схема работы программы.

Описанная структура позволит пользователю гибко настраивать оптимизацию кода. Таким образом можно будет улучшать значения метрик, путем использования шаблонных преобразований кода, что должно сократить затраты на оптимизацию и рефакторинг кода.

Библиографический список

1. НОУ «ИНТУИТ». Лекция 11: Модели качества и надежности в программной инженерии [Электронный ресурс] // <http://www.intuit.ru>: [сайт]. [2009]. URL: <http://www.intuit.ru/studies/courses/2190/237/lecture> (дата обращения: 3.апрель.2017).
2. SC 7 System and Software Engineering. ISO/IEC 12207:2008 Systems and software engineering — Software life cycle processes // <http://disi.unal.edu.co>. 2008. URL: http://disi.unal.edu.co/dacursci/sistemasycomputacion/docs/SystemsEng/ISO_IEC_12270_2008.pdf (дата обращения: 03.апрель.2017).
3. Понятие метрики. Направления применения метрик. Метрические шкалы. Метрики сложности. Метрики стилистики. [Электронный ресурс] // Метрики: [сайт]. URL: <http://www.metric-narod.ru/page1.htm> (дата обращения: 03.апрель.2017).
4. Терехов А.А., Вояковская Н., Бульчев Д., Москаль А. Разработка компиляторов на платформе.NET. - СПб.: Кафедра системного программирования СПбГУ, 2001. // ict.edu. 2001. URL: <http://www.ict.edu.ru/ft/005128/ch11.pdf> (дата обращения: 3.апрель.2017).
5. Ball T. The Concept of Dynamic Analysis // ACM SIGSOFT Software Engineering Notes, No. 24, Ноябрь 1999. pp. 216-234.
6. Марков А.С., Фадин А.А. Статический сигнатурный анализ безопасности программ // Программная инженерия и информационная безопасность, 2013.
7. Касьянов В.Н. Оптимизирующие преобразования программ. М.: Наука, 1988. 336 pp.

УДК 004.451

Д. Ю. Кунцевич

магистрант кафедры компьютерных технологий и программной инженерии

В. В. Бураков

профессор, д-р техн. наук, доцент – научный руководитель

УПРАВЛЕНИЕ СТРУКТУРНЫМИ СВОЙСТВАМИ ПРОГРАММ

Помимо того, что программы должны работать правильно, они должны работать быстро. Недостаточно использовать эффективные алгоритмы и структуры данных. Нужно писать такой код, который компилятор легко оптимизирует и транслирует в быстрый исполняемый код.

Основной задачей исследования является исследование возможности программной оптимизации кода при помощи метрик.

Метрика ПО – мера, позволяющая получить численное значение некоторого свойства программного продукта или его спецификаций.

Метрики кода разделяются на категории и могут оценивать совершенно различные аспекты программной системы: сложность и структурированность программного кода, связность компонентов, и другие [3].

Метрики сложности программ принято разделять на три основные группы [4]:

- метрики размера программ;
- метрики сложности;
- объектно-ориентированные метрики.

Метрики первой группы базируются на определении количественных характеристик, связанных с размером программы, и отличаются относительной простотой [1].

Такая метрика может высчитываться по формуле

$$F_i = \text{SIGN}(N_{\text{комм.}i} / N_i - 0,1),$$

где $N_{\text{комм.}i}$ – количество комментариев в i -м фрагменте, N_i – общее количество строк кода в i -м фрагменте. Суть метрики проста: код разбивается на n -равные куски и для каждого из них определяется F_i [6].

Метрики второй группы базируются на анализе управляющего графа программы [2].

Для объектно-ориентированных метрик, помимо таких очевидных, как общее число классов, методов, атрибутов, средних показателей числа методов и атрибутов на класс, применяются гораздо более комплексные метрики, с помощью которых можно судить не только об объеме исходного кода проекта, но и о его сложности, качестве, соответствии основным принципам объектно-ориентированной парадигмы и т.д.

Также мерой качества программного обеспечения служит связанность модулей программы.

Кодовая оптимизация – это преобразования кода за счет различных методов с целью улучшения различных атрибутов и характеристик [7]. Основными целями кодовой оптимизации являются следующие составляющие:

- ускорение компиляции кода;
- ускорение работы самой программы;
- уменьшение объема кода;
- уменьшение потребления системных ресурсов.

Существуют несколько типов оптимизации:

– реерhole-оптимизация. Её суть состоит в том, что идет рассмотрение «инструкций», чтобы сделать вывод: можно ли сделать с ними трансформацию для кодовой оптимизации;

– внутрипроцедурная оптимизация – это глобальная оптимизация, которая выполняется целиком и только в рамках одной единицы трансляции: функции.

– локальная оптимизация охватывает рассмотрение информационный базовый блог, строго за один шаг. В самих базовых блоках нет переходов к потокам управления, что значит следующее: данная оптимизация экономит время и оперативную память, но теряет информацию к следующему шагу оптимизации;

– межпроцедурная оптимизация. С помощью данного метода оптимизации можно анализировать сразу весь код, без определенного поиска нужных блоковых частей. Встраивание копии тела функций, с помощью данного метода, позволяет сэкономить системные ресурсы, которые связаны именно с вызовом функций.

Рассмотрим основные приемы оптимизации программ:

– уменьшение числа переменных. Временные переменные, которые отработали на участке программы (управляющие переменные цикла, например) продолжают занимать память и далее, на протяжении всей программы. Ее нужно удалить;

– преобразование выражений. Предварительное упрощение может привести к исключению многих арифметических операций;

– работа над арифметическими операциями. Арифметические операции выполняются с разной скоростью и иногда целесообразно заменить одну операцию на другую (например, x^3 заменить на $x + x + x$) [4];

– уменьшение цикломатической сложности. Чистка цикла уменьшает время выполнения цикла (а, следовательно, и самой программы) путем удаления из тела частей арифметических выражений, не зависящих от управляющей переменной цикла. Так же существует возможность объединения циклов. Такая процедура производит соединение нескольких циклов с одинаковыми заголовками в один. Иногда бывает полезным и вовсе отказаться от использования циклов.

Существует также определенный тип метрик, связанных со скоростью работы приложений и их производительностью. Они так и называются – метрики производительности.

Основные метрики, используемые при анализе производительности[5]:

– Потребление ресурсов центрального процессора. Метрика, показывающая сколько времени из заданного определенного интервала было потрачено процессором на вычисления для выбранного процесса. В современных системах важным фактором является способность процесса работать в нескольких потоках для того, чтобы процессор мог производить вычисления параллельно. Анализ истории потребления ресурсов процессора может объяснять влияние на общую производительность системы потоков, обрабатываемых данных, конфигурации приложения и операционной системы, мультипоточности вычислений и других факторов.

– Потребление оперативной памяти. Метрика, показывающая количество памяти, использованной приложением. Исползованная память может делиться на три категории:

1. Virtual

Это объём виртуального адресного пространства, используемого процессором. Он не обязательно подразумевает использование соответствующего дискового пространства или оперативной памяти.

2. Private

Это объём адресного пространства, занятого процессом и не разделяемого с другими процессами.

3. Working Set

Это набор страниц памяти, недавно использованных процессом. В том случае, когда свободной памяти достаточно, страницы остаются в наборе, даже если они не используются. Когда остается мало свободной памяти, использованные страницы удаляются. На время, требующееся для очистки памяти, доступ процесса к страницам выделенной памяти может быть заблокирован, что может повлиять на конечное время обработки этим процессом данных[2].

– Потребление сетевых ресурсов. Эта метрика не связана непосредственно с производительностью приложения, однако её показатели могут указывать на пределы производительности системы в целом.

Оптимизация в основном фокусируется на одиночном или повторном времени выполнения, использовании памяти, дискового пространства, пропускной способности или некотором другом ресурсе. Это обычно требует компромиссов – один параметр оптимизируется за счёт других, например увеличение размера программного кэша чего-либо улучшает производительность времени выполнения, но также увеличивает потребление памяти[8].

В заключение нужно отметить, что для оптимизации требуется найти критическую часть кода, которая является основным потребителем необходимого ресурса. Утечка ресурсов (памяти, дескрипторов и т. д.) также может привести к падению скорости выполнения программы.

Библиографический список

1. *Звездин С.* Метрики как средство управления качеством, <https://www.osp.ru/os/2009/08/10748698>
2. *Ройс У.* Управление проектами по созданию программного обеспечения / ЛОРИ, 424 с.
3. PVS-Studio: Первые наши практические исследования в области расчета метрик, <https://www.viva64.com/ru/b/0014/>
4. Метрики: Понятие метрики. Направления применения метрик. Метрические шкалы. Метрики сложности, <http://www.met-fix.narod.ru/>
5. *Черноножкин С.К.* Методы и инструменты метрической поддержки разработки качественных программ, автореферат, Новосибирск, 1998.
6. PVS-Studio: Метрики кода программного обеспечения, <https://www.viva64.com/ru/a/0045/>
7. *Ричард Г.* Оптимизация ПО. Сборник рецептов / СПб.: Издательский дом Питер, 2010. – 348 с.
8. *Касперски К.* Техника оптимизации программ. Эффективное использование памяти / СПб.: БХВ-Петербург, 2003. – 464 с.

УДК 378.096 и 007.51

А.И. Легкоступов

магистр кафедры компьютерных технологий и программной инженерии (№43)

А.А. Ключарев

канд. техн. наук, доцент – научный руководитель

АВТОМАТИЗАЦИЯ СОСТАВЛЕНИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ ВУЗА В СООТВЕТСВИИ С ФГОС

В рамках вуза ГУАП необходимо автоматизировать процесс создания рабочих программ дисциплин (РПД) в соответствии с предъявляемыми государственными стандартами. Система автоматизации составления РПД должна быть модернизируемой для упрощенного перехода к новым стандартам.

Процесс создания новой дисциплины для преподавания в ВУЗе является сложно формулируемой задачей с множеством условий. Подобная сложность вызвана большим спектром направлений, представленных для обучения. Создание гуманитарной дисциплины существенно отличается от технической и т.п. Федеральные государственные стандарты в высшем профессиональном образовании (ФГОС ВО) формулируют основные требования к составлению дисциплин различного рода, разделенных по направлениям. Согласно федеральному закону «Об образовании в Российской Федерации» для каждого направления должна быть разработана образовательная программа (ОП), которая будет учитывать: структуру учебного процесса, количество зачетных единиц, наборы компетенций и др. Таким образом, создание дисциплины ограничивается множеством федеральных документов с огромным количеством требований, которые в течении времени изменяются в соответствии с различными взглядами на образовательный процесс законодателей. Для преподавателей, желающих составить новую дисциплину или изменить текущие, составление РПД представляет существенное затруднение. В большинстве случаев РПД по различным предметам составляются одними и теми же уполномоченными сотрудниками кафедр, которые соблюдают необходимые требования и положения. Ценность таких документов существенно снижается из-за некомпетентности сотрудников в структуре и материале дисциплины. Так происходит в связи с необходимостью поддержания отчетности для контролирующих органов. Основной идеей разработки РПД является возможность понять, что представляет из себя дисциплина, какое место она занимает в ОП и какую ценность для слушателя несёт. При подходе «РПД для отчётности» данный документ не сможет точно регламентировать, что же будет прочитано слушателю. Это является существенным недостатком текущего подхода.

В настоящее время вводятся множество понятий дистанционного подхода. Это может быть личный кабинет (ЛК) для доступа к материалам, это может быть полнофункциональная система дистанционного обучения (СДО) с интерактивными лекциями и заочными экзаменами, это может быть база выполненных работ во время обучения, для контроля за процессом обучения в учебном заведении и т.д. В нашем ВУЗе ГУАП разрабатываются системы ЛК и СДО, что в дальнейшем позволит привлекать большое количество удаленных студентов и иметь при этом прозрачную систему получения дипломов и степеней. При наличии системы ЛК и доступа к ней любого сотрудника и студента облегчается возможность коммуникации участников учебного процесса, а также распространение информации в образовательной среде ВУЗа в сжатые сроки, что позволяет поддерживать её актуальности. Поэтому создание сервиса автоматизированного формирования РПД в ЛК станет шагом к полному переходу от бумажного документооборота к электронному.

При проектировании сервиса появились вопросы, связанные с унификацией РПД для различных типов дисциплин. За эталонную модель РПД, вобравшую в себя все требования законов, приказов и иных документов, прошедшую проверку несколькими аккредитационными инспекциями, была выбрана существующая в ВУЗе. Для формирования электронных форм заполнения РПД необходимо проанализировать структуру документа, выявить основные сущности, получить взаимосвязи разделов и привести к компактному и понятному виду.

Проведя анализ эталонной модели РПД были выявлены несколько типов данных: формальные данные дисциплины, списки или справочники и уникальные данные дисциплины.

Формальные данные представляются собой перечень условий, содержащихся в ОП, при составлении РПД для конкретной дисциплины сервис автоматически запросит их и сформирует на основании этих данных дальнейшие условия для заполнения. Виды формальных данных: наименование ВУЗа, наименование дисциплины, код дисциплины, форма обучения, данные заведующего кафедрой, реализующей дисциплину, данные разработчика, ответственного за дисциплину, данные ответственного за направление подготовки, данные ответственного за ОП, ссылка на ФГОС актуальной версии, перечень компетенций, обеспечиваемых дисциплиной, распределение часов и зачетных единиц по видам аудиторных занятий, самостоятельной работы студента (СРС) и промежуточному контролю.

Списки или справочники – это строго типизированные и индексированные данные системы. Типы списков: библиотечные фонды учебной литературы, фонд электронных образовательных ресурсов, фонд Интернет-ресурсов, фонд информационно-справочных систем, фонд материально-технического обеспечения, фонд лицензионного и свободно распространяемого программного обеспечения, варианты форм и процедур текущего контроля успеваемости и промежуточной аттестации. Все списки фильтруются на основании формальных данных. При исследовании списков было выявлено отсутствие некоторых из них в университете ГУАП. В данный момент формируются справочники аудиторного и материального фондов.

Уникальные данные о содержании дисциплины – это данные, формируемые разработчиком рабочей программы. Типы данных: наименование разделов (модулей) дисциплины с указанием формируемых компетенций указанных в ОП, содержание разделов (модулей дисциплины), содержание СРС, перечень тем практических занятий с указанием трудоемкости в часах ссылкой на методические материалы, перечень тем лабораторных занятий с указанием трудоемкости в часах и ссылкой на методические материалы, перечень тем курсового проектирования, перечень дополнительной литературы, фонд оценочных средств и дополнительная информация, вносимая по усмотрению разработчика. Весь уникальный контент РПД распределяется на два вида: разделы дисциплины с привязкой практических и лабораторных работ, а также указанием реализуемых компетенций разделом; общая информация в виде используемой не библиотечной литературы, формата оценки знаний с указанием вопросов для контроля и дополнительной информации для студентов.

Рассмотрев полученный объём данных была произведена структуризация процесса создания РПД, результат представлен на рис. 1. На схеме изображена концепция сервиса, формирующего РПД поэтапно на основании заполнения разделов. Данная концепция выбрана из-за особенностей системы в которой реализуется сервис.

В сети Интернет довольно часто случаются сбои, поэтому для минимизации потерь информации формирование РПД происходит некоторыми логическими блоками (разделами). После составления очередного раздела РПД преподаватель может прервать процесс создания и возобновить его в любое удобное для него время. Каждый раздел проверяется на соответствие шаблону. Шаблоны формируются в соответствии с формальными данными из ОП. Процесс контроля будет проверять несколько параметров: дата последней корректировки РПД, соответствие суммарной трудоемкости по видам занятий учебному плану, учет всех заданных компетенций разделами (модулями) дисциплины, соответствие основной литературы и используемых электронных ресурсов требованиям по сроку первого издания (10 лет), обеспеченность дисциплины материально-техническими и учебно-методическими ресурсами, обеспеченность практических и лабораторных занятий, курсового проектирования методическими материалами и наличие описанного в РПД фонда оценочных средств. В конце процесса формируется документ РПД, который можно представить к согласованию на кафедре. На рис. 2 представлена работа контролирующего лица с проектируемым сервисом.

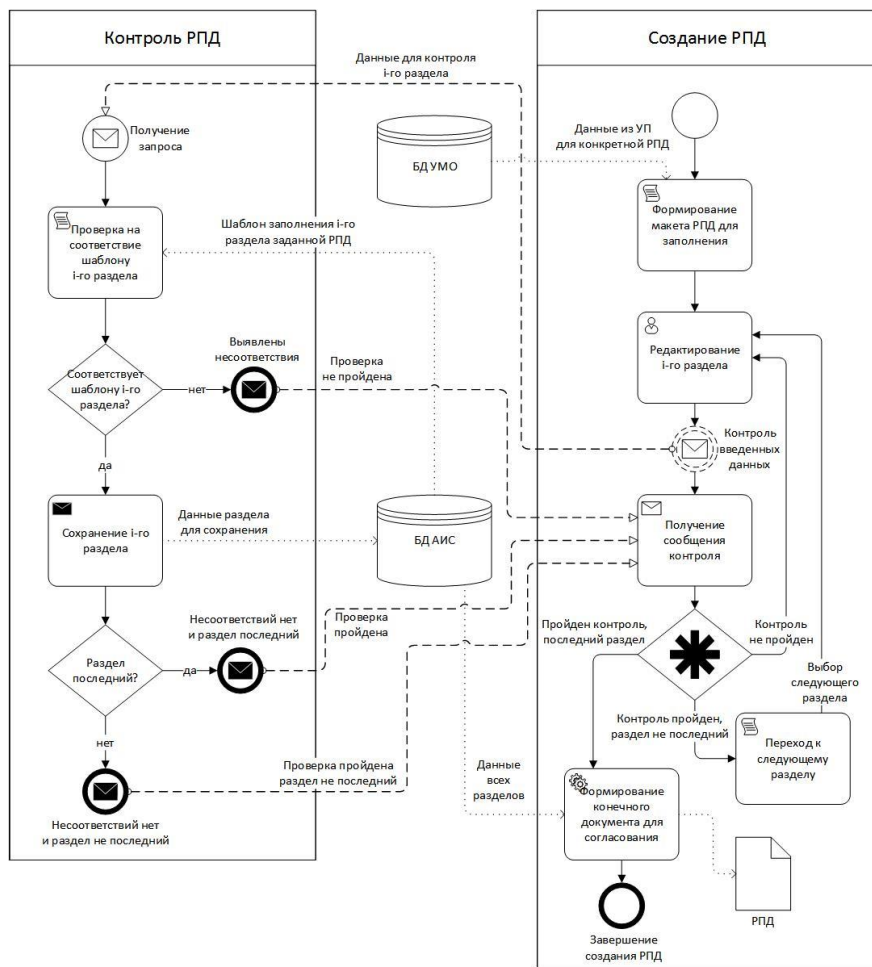


Рис. 1. Схема взаимодействия процессов при формировании РПД

Процесс согласования не имеет чёткой формулировки. Какие именно требования предъявляются к РПД на различных кафедрах является внутренней информацией кафедры, поэтому при проектировании этапа согласования было принято решение не вносить процесс кафедрального обсуждения в систему, а оставить его бизнес-правилом. Большое количество неформализованных данных в РПД влечёт необходимость проверки заинтересованным лицом (ответственным за направление). Если РПД удовлетворяет внутренним требованиям кафедры, то РПД отправляется на согласование. После согласования, в случае успеха будет готова конечная версия РПД (согласованная), соответствующая стандартам ФГОС и требованиям разрабатывающей кафедры.

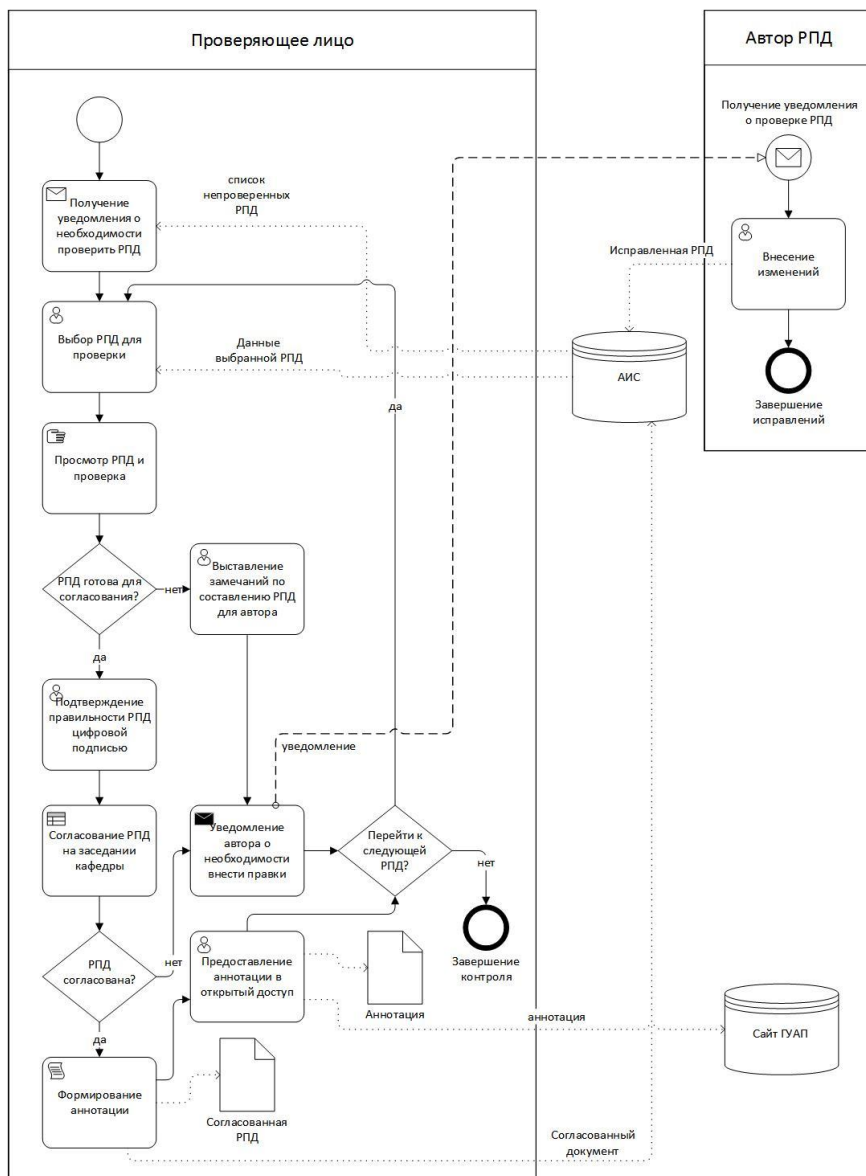


Рис. 2. Процесс согласования содержания РПД

Стандарты требуют, чтобы РПД или её краткую аннотацию ВУЗ предоставлял в открытый доступ на своём сайт в сети Интернет. В следствии того, что РПД является интеллектуальной собствен-

ностью автора и уникальна для данного предмета, то преподаватели не желают обнародовать собственный труд, поэтому имеется возможность предоставлять в открытый доступ обзорную версию РПД (аннотацию). Сейчас аннотация формируется отдельным файлом и требует знаний того, что необходимо указать в ней. При реализации сервиса формирование аннотации будет автоматизировано. При желании автор сможет дополнять автоматически сформированную аннотацию.

Автоматизированный процесс создания РПД позволит ускорить разработку новых дисциплин, уменьшит число ошибок при реализации и позволит производить контроль разработки РПД на уровне ВУЗа. Мониторинг системы составления РПД по кафедрам будет состоять из оперативного контроля укомплектованности ОП согласованными РПД, покрытие компетенций ОП дисциплинами и контроль знаний студентов по изученным дисциплинам в соответствии с указанными параметрами компетенций. На основании данных из РПД в дальнейшем, в системе СДО можно будет облегчить создание интерактивных курсов для дистанционного обучения.

УДК

И. В. Львова

магистрант кафедры компьютерных технологий и программной инженерии

П. А. Степанов

старший преподаватель – «научный руководитель»

АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ ШАБЛОНОВ ПРОЕКТОВ НА ОСНОВЕ ФОРМАЛЬНЫХ ТРЕБОВАНИЙ

В современном мире требования к программному обеспечению становятся все более и более сложными, так как растут пользовательские ожидания к новым продуктам и увеличивается конкуренция на рынке прикладных программ. Современные тенденции в разработке программного обеспечения заключаются в том, что приложения уходят в Интернет и становятся доступны миллионам пользователей через протокол http или мобильные клиенты. Появляются облачные технологии, позволяющие разворачивать приложения любой сложности и рассчитанные на различную нагрузку, а также представлять приложение в виде сервиса, предоставляемого подписчикам [1]. На смену распространённым в прошлом монолитным архитектурам пришли микросервисы, которые позволяют развертывать и модернизировать каждую свою компоненту отдельно.

В зависимости от решаемых задач и расчетной нагрузки архитектура программного продукта может быть существенно различной, а также может отличаться способ развертывания приложения, его обслуживания и администрирования. Стоимость внесения изменений и расширение возможностей приложений влияют на стоимость владения данным продуктом. При создании новых приложений возникает вопрос какие технологии и программное обеспечение использовать. На сегодняшний день затруднено расширение некоторых существующих проектов из-за достижения пиковых значений характеристик используемых технологий или тесной связи всех модулей системы, т.е. когда изменение в одном классе или функции может требовать серьезных изменений в остальной части программы. Разрабатываемая в рамках магистерской диссертации экспертная система способна по конкретным параметрам, например, количеству запросов в секунду, одновременно работающих пользователей и т.д., будет рекомендовать использовать ту или иную архитектуру проекта с использованием наиболее подходящего дополнительного программного обеспечения. Таким образом, возможно будет избежать ошибок по несовместимости возможностей технологий и требований к конечному разрабатываемому продукту и уменьшить затраты и время на обслуживание и модернизацию программного комплекса.

Введем определение архитектуры и связанных с ней терминов.

Определение 1. Архитектура – набор системных фундаментальных концепции или свойств элементов системы для ее проектирования и модернизации, а также принципов взаимоотношений с окружающей средой [2].

Определение 2. Окружающая среда – системный контекст, определяющий условия и обстоятельства всех воздействий на систему, и включает в себя производственные, технологические, деловые, операционные, организационные, политические, экономические, правовые, регулятивные, экологические и социальные факторы [2].

В рамках работы рассматриваются следующие архитектуры:

– Desktopное приложение – содержит все ресурсы на локальном компьютере и не рассчитаны на одновременную работу пользователей [3].

– Клиент-серверное приложение – состоит из ядра системы, которым является сервер баз данных, представляющий собой приложение для управления данными: выполнение запросов, хранение и резервное копирование данных, отслеживание ссылочной целостности, проверку прав и привилегий пользователей, ведение журнала транзакций; и клиентской части, отправляющей запросы по работе с данными [4].

– Web-приложение – подмножество клиент-серверных приложений. Их особенность заключается в том, что Web-приложение развертывается и выполняется на сервере и клиент при этом получает только результаты его работы, которая основывается на получении запросов от пользователя (клиента), их обработке и выдачи результата. Передача запросов и результатов их обработки происходит через сеть, обычно через протокол http и более высокоуровневые протоколы, построенные поверх него (JSON, REST, SOAP и другие).

– Приложение как сервис (SaaS) – модель программного обеспечения, при которой поставщик разрабатывает и управляет веб-приложением, предоставляя доступ к нему через интернет подписчикам. Основное преимущество модели для потребителя состоит в отсутствии необходимости установки, обновления и поддержки программного обеспечения, а с точки зрения разработчика, SaaS позволяет упростить техподдержку, а также эффективно бороться с нелегальным использованием программного обеспечения, поскольку само программное обеспечение не попадает к конечным заказчикам [5]. В основе модели обычно лежит PaaS.

– Облачное приложение (PaaS) – модель программного обеспечения, в которой вычислительные ресурсы предоставляются по запросу, на которых заказчик имеет возможность развернуть и запустить произвольное программное обеспечение, включающее в себя операционные системы и приложения. В рамках данной модели заказчик не управляет и не контролирует лежащую в основе физическую инфраструктуру, но имеет контроль над операционными системами и развернутыми приложениями [6].

– Архитектура EDA – способ написания приложений, в котором все сообщения доставляются асинхронно через очередь. В этом случае потребители сообщений имеют возможность всегда обрабатывать количество запросов не более своей расчетной нагрузки, что делает нагрузку на приложение более равномерной. Однако в пиковые моменты работы приложения оно может сильно проигрывать в скорости, а также возможно переполнение очереди сообщений.

– Сервис-ориентированная архитектура (SOA) – один из вариантов реализации архитектуры EDA, предназначенный для создания информационной сети предприятия, использующей принципы ориентации на сервисы для достижения тесной связи между бизнесом и поддерживающими его информационными системами, таким образом данная архитектура улучшает взаимосвязь между предприятием и бизнесом, позволяет создавать интегрированные приложения, основанные на существующих сервисах, создает гибкие бизнес-процессы **Указан недопустимый источник.**

– Load balancer – технология, позволяющая распределять нагрузку между сервисами, способными обработать запрос **Указан недопустимый источник.** Одним из наиболее известных способов балансировки является DNS Round Robin – технология, при которой сервер DNS возвращает каждый раз различный адрес для домена, перечисляя адреса из некоторого набора серверов, которые могут обработать запрос.

Сравнительный анализ по допустимой нагрузке приведен в табл. 1.

Таблица 1

Сравнительный анализ по количеству обрабатываемых запросов

Архитектура	Количество запросов в секунду			
	Десятки	Сотни	тысячи	миллионы
Десктопное приложение				
Клиент-серверное приложение	+	+		
Web-приложение	+	+		
SaaS	+	+	+	+
PaaS			+	+
EDA			+	+
SOA	+	+	+	
Load balancer			+	+

Разрабатываемая экспертная система предназначена для определения наиболее подходящей архитектуры программного обеспечения по набору нефункциональных требований, таких как количе-

ство запросов от клиента, количество одновременных пользователей, скорость обработки запроса, возможность передавать данные через Интернет.

На текущий момент известны программные средства для управления требованиями, которые позволяют анализировать связанность требований с разрабатываемыми модулями системы. К сожалению, программных продуктов, которые подбирают архитектуру системы по списку требований на сегодняшний день не известно.

В качестве основы для анализа такой системы используется стандартизированная форма спецификации требований, разработанная по ГОСТ 19.201 – 78. Далее требования анализируются в соответствии с разработанной математической моделью для подбора лучшей архитектуры. В спорных ситуациях выбор конкретной архитектуры из предложенных предоставляется эксперту, анализируется и добавляется решение в базу данных системы. Далее генерируется заложенный в систему набор артефактов, например, модели базы данных, модели базовых классов и др.

Таким образом, разрабатываемая система сможет рекомендовать подходящую архитектуру и генерировать базовые объекты для каждого конкретного варианта, что позволит сократить количество ошибок, связанных с человеческим фактором при разработке программных продуктов, а так же предложить архитекторы разные, возможно, неочевидные модели системы в рамках заданных ограничений.

Библиографический список

1. Microsoft Developer Network. Обзор проектирования архитектуры [Электронный ресурс] // Microsoft Developer Network: [сайт]. [2017]. URL: <https://msdn.microsoft.com/ru-ru/hh144976.aspx> (дата обращения: 02.04.2017).
2. INTERNATIONAL STANDARD ISO/IEC/IEEE 42010. Systems and software engineering — Architecture description. New York: Institute of Electrical and Electronics Engineers, Inc., 2011.
3. Молчанов Ю. Системное программное обеспечение. 3rd ed. СПб: ПИТЕР, 2010. 400 с pp.
4. Елманова. Перенос приложений C++ Builder в архитектуру клиент/сервер // CITForum. 2015. URL: http://citforum.ru/programming/application/builder_cs3.shtml (дата обращения: 02.04.2017).
5. Software as a service [Электронный ресурс] // TADVISER: [сайт]. [2016]. URL: http://www.tadviser.ru/index.php/SaaS_%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BA%D0%B0%D0%BA_%D1%83%D1%81%D0%BB%D1%83%D0%B3%D0%B0 (дата обращения: 08.04.2017).
6. ОБЛАЧНЫЕ СЕРВИСЫ ИЛИ ЧТО ТАКОЕ IAAS? ОТЛИЧИЕ ОТ SAAS И PAAS [Электронный ресурс] // IT-Grad: [сайт]. [2014]. URL: http://www.it-grad.ru/tsentr_kompetentsii/blog/33/ (дата обращения: 08.04.2017).
7. Обзор терминологии SOA: Часть 1. Сервис, архитектура, управление и бизнес-термины [Электронный ресурс] // IBM DeveloperWorks: [сайт]. [2008]. URL: <https://www.ibm.com/developerworks/ru/library/ws-soa-term1/> (дата обращения: 02.04.2017).
8. Протасов С.С. Б.С.М.Т.А.Г. Математическая модель и метод построения сервиса // ИССЛЕДОВАНО В РОССИИ. 2004. No. 204.

УДК 004.42

И.В. Маковечик

студент кафедры компьютерных технологий и программной инженерии.

В.Ю. Гамов

канд. в. наук, доцент – научный руководитель

РАЗРАБОТКА КОМПЬЮТЕРНОЙ МОДЕЛИ ДЛЯ РАСЧЕТА ВЕРОЯТНОСТИ ИСХОДА СПОРТИВНОГО СОБЫТИЯ.

Каждый день в мире происходят встречи спортсменов, и многих людей интересует вопрос, как именно закончится данное состязание. Особый интерес вызывают теннисные матчи, так как встречи спортсменов происходят достаточно часто, и сам спорт хорошо популяризирован.

Основная цель данной работы разработать компьютерную модель способную с некоторой точностью рассчитывать вероятность исхода теннисного матча или предсказать его исход.

Для достижения поставленной цели необходимо решить ряд задач:

- описать основные факторы, влияющие на исход теннисного матча;
- ввести основные критерии, которые накладывают ограничения на описанные факторы;
- произвести анализ различных алгоритмов и математических моделей;
- выявить более точный способ прогнозирования теннисного матча;
- реализация программного обеспечения на основе выбранного способа.

Существует множество теорий и написано немало статей людьми, разбирающимися в спорте о том, на какие показатели необходимо обратить внимание для успешного прогнозирования исхода матча. После анализа теннисных матчей и прочтения различных статей спортивных прогнозистов по оцениванию теннисных встреч выявлены следующие факторы, влияющие на исход спортивного события:

- место в турнирной таблице игрока;
- мировой рейтинг спортсмена;
- средний показатель побед игрока в текущем сезоне;
- средний показатель побед при 3 сетах;
- средний показатель побед при 5 сетах;
- процент заработанных очков при своих подачах;
- процент заработанных очков при чужих подачах;
- возраст игрока;
- процент побед при личных встречах теннисистов;
- покрытие корта;
- стиль игры спортсмена.

Для упрощения модели, а также для того, чтобы избежать обработки больших объемов данных, необходимо ввести ограничения:

- Турнир большого шлема WTA и ATP – Выбран один из самых больших теннисных турниров, 3 и 5 сетовые матчи.
- Мужчины – исключили женщин в пользу упрощения математической модели.
- В расчет не берутся тайм-брейки.
- Ограничений по возрасту нет.
- Считаем, что игрок не пропускал никаких турниров до данного.
- Все игроки здоровы, подготовлены и умеют играть в теннис.

Информация по игрокам и по их встречам находится в открытом доступе на множестве ресурсов, один из таких сайт <http://atpworldtour.com/>. Был произведен сбор всей необходимой информации, а также ее преобразование в формат удобный для использования в моделях, представленный в табл. 1. В данной таблице представлена часть характеристик 1 и 2 игрока встречи.

Таблица 1

Формат данных для анализа.

	avgacesp ermatch_ 1	avgacesp ermatch_ 2	avgdbfault spermatch _1	avgdbfault spermatch _2	co ur t	firstSer vePct_ 1	firstSer vePct_ 2	firstServePoi ntsWonPct_ 1	firstServePoi ntsWonPct_ 2
1	18,5	6,3	2,8	3,4	1	66,8	63,7	84,1	72,9
2	18,5	1,7	2,8	3,1	1	66,8	60,9	84,1	66,8

Для сбора информации об игроках и построения моделей использован язык программирования Python с подключением библиотеки машинного обучения sklearn. В данной библиотеке уже реализовано большинство математических моделей, что позволяет быстро и легко их использовать, тем самым выявить наиболее подходящую модель для решения поставленной задачи. На текущем этапе были выбраны и проанализированы следующие алгоритмы и математические модели:

- линейная регрессия(Linear Regression);
- алгоритм «Случайный лес»(Random Forest Regressor) – алгоритм машинного обучения, придуманный Лео Брейманом и Адель Катлер для решения задач классификации, регрессии, кластеризации, поиска аномалий, селекции признаков и т.д.;
- алгоритм k ближайших соседей(K Neighbors Regressor) – метрический алгоритм для автоматической классификации объектов;
- метод опорных векторов – является одной из наиболее популярных методологий обучения по прецедентам, предложенной В. Н. Вапником и известной в англоязычной литературе под названием SVM (Support Vector Machine);
- логистическая регрессия (Logistic Regression) – это статистическая модель, используемая для предсказания вероятности возникновения некоторого события путём подгонки данных к логистической кривой.

Для каждой модели из списка использовалась одна обучающая выборка и одна тестовая выборка. После обучения модели производилось предсказание результатов матча по обучающей выборке, а затем по тестовой, после чего был найден коэффициент детерминации для предсказаний, результаты представлены на рис. 1, а числовые значения в табл. 2.

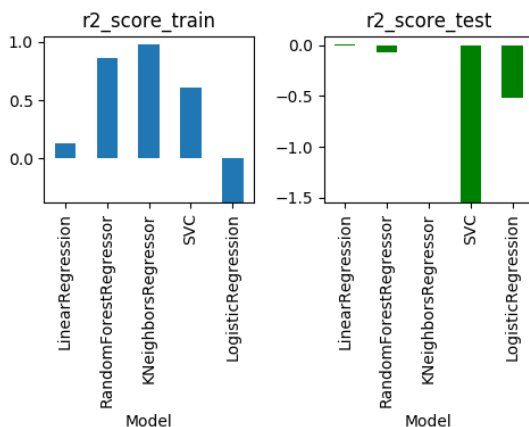


Рис. 1. Гистограмма коэффициентов детерминации моделей

Таблица 2

Числовые значения коэффициентов детерминации моделей.

	Linear Regression	Random Forest Regression	K Neighbors Regression	SVC	Logistic Regression
train	0.132655	0.860214	0.975293	0.610101	-0.372769
test	0.003230	-0.070804	-0.007060	-1.543527	-0.513821

Из рис. 1 и табл. 2 видно, что прогнозы алгоритмов по обучающей выборке выполнены с небольшим количеством ошибок, однако прогнозы по тестовой выборке дало большое число неточностей. Следует предположить, что сам расчет коэффициента может быть неверен, так как по данным полученным после предсказания не проводилось дополнительное преобразование, что и могло привести к ошибочному расчету коэффициента, и как результат неверное оценивание ошибочности модели.

Так как «Метод опорных векторов» и «Логистическая регрессия» лежат в основе алгоритмов классификации, можно по-другому оценить точность данных методов. Для таких алгоритмов создана специальная метрика для оценки точности. Как видно из рис. 2 точность метода опорных векторов по обучающей выборке составила больше 80%, а логистической регрессии чуть больше 60%. В свою очередь SVC алгоритм сдал свои позиции на тестовой выборке и понизил точность до 40%, когда логистическая регрессия дала предсказание с аналогичной точностью.

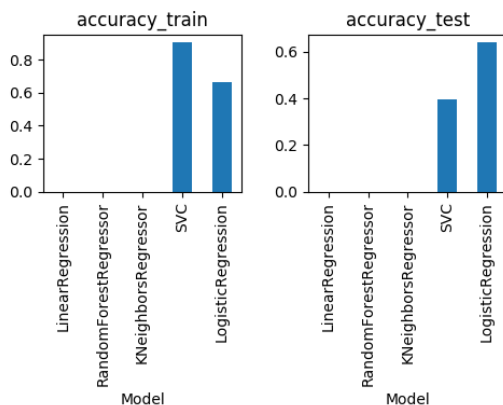


Рис. 2. Оценка точности алгоритмов

Анализ представленных алгоритмов производился с настройками по умолчанию. По данным полученным в процессе выполнения работы можно сделать вывод, что лучший результат прогнозирования показали алгоритмы на основе «Метода опорных векторов» и «Логистической регрессии». Однако данные алгоритмы требуют доработки для получения более высокой точности предсказания, чего можно добиться подбором подходящих параметров настройки алгоритмов.

В настоящий момент в ходе выполнения работы была проанализирована предметная область, выявлены факторы, влияющие на исход спортивного состязания, произведен подбор и анализ алгоритмов машинного обучения. С помощью данных алгоритмов были предсказаны результаты ряда теннисных матчей с точностью до 60%, что позволяет считать цель текущего этапа исследования достигнутой.

Дальнейшее развитие работы связано с разработкой программного обеспечения, способного в автоматическом режиме получать всю необходимую информацию об игроках, а также о предстоящих играх. Необходимо произвести улучшение существующих моделей, с целью повышения их точности, а также подбор и анализ новых.

УДК 651.011.56

А.Ю. Миронов

магистрант кафедры компьютерных технологий и программной инженерии

М.В. Фаттахова

канд. физ.-мат. наук, доцент – научный руководитель

МОДЕЛЬ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ АДМИНИСТРАТИВНЫМ ДЕЛОПРОИЗВОДСТВОМ

Ежегодно на территории региона должностные лица 105 органов исполнительной власти составляют несколько миллионов протоколов об административных правонарушениях. Расследованием и рассмотрением по подведомственности дел о них уполномочены заниматься, кроме областных, районных и мировых судей, 68 правоохранительных и исполнительных органов государственной власти РФ и субъекта РФ, которые после вручения правонарушителю постановления по делу об административном правонарушении обеспечивают также исполнение административных наказаний, информирование о привлечении к административной ответственности.

Административное делопроизводство, опирающееся на подшивание рукописных или напечатанных документов в дело об административном правонарушении, остается крайне трудоемким и неэффективным, провоцирующим на нарушение разумного срока осуществления административного процесса. Подведомственные стадии производства по делам о административных правонарушениях реализуются и учитываются органами по исполнению административного законодательства обособленно, подручными методами и средствами. Передача дел по подведомственности и взаимодействие между ними производится нарочным или по почте. Управление административным процессом на системном уровне отсутствует.

В результате, часть административных протоколов длительное время или навсегда остается без решения, административные наказания не исполняются в полном объеме. Только от не собираемости более 40% административных штрафов федеральный и региональные бюджеты теряют десятки процентов доходной части или сотни миллионов рублей в регионе за год. Региональные законы об административных правонарушениях фактически бездействуют из-за отсутствия механизма реализации.

Таким образом, назрели объективные необходимость и условия в организации на уровне субъекта РФ многоэтапного производства по делам об административных правонарушениях в общем пространстве автоматизированных процедур, осуществляемых в разумные сроки под оптимальным управлением регионального участника административного процесса, в подведомственности которого находится текущая административная процедура [1].

В условиях дестабилизирующего влияния случайных объективных и субъективных действий и бездействий участников производства и иных факторов окружающей среды (далее – среды) X , управлением административным делопроизводством является организация уполномоченным органом исполнительной власти (далее – субъект) такого подведомственного воздействия U на административный процесс (далее – объект) Y , в результате которого в разумный срок удовлетворяются нормативные требования административного законодательства к субъекту, определяющие должностные обязанности субъекта:

$$A = (\alpha_1, \dots, \alpha_j, \dots, \alpha_k), \quad (1)$$

где α_j – состояние i -й должностной обязанности субъекта, которая выражается неотрицательным числом, характеризующим насущность и актуальность исполнения этой обязанности в административном процессе.

В режиме управления субъект исполняет должностную обязанность так, чтобы минимизировать насущность совокупности обязанностей, критичных для реализации цикла производства по делам об административных правонарушениях, т. е. решает задачу многокритериальной оптимизации:

$$\alpha_i(X, Y, U) \rightarrow \min_{r \in R} (i = \overline{1, k}), \quad (2)$$

где R – ресурсы r субъекта. Эта зависимость отражает неизвестную, но существующую связь исполнения критичных обязанностей α_i с состоянием среды X , объектом Y и поведением U субъекта.

Способ решения задачи (2) определения оптимального поведения U_X^* субъекта, которое минимизирует совокупность критичных обязанностей A , представляет собой алгоритм управления, определяющий эффективность функционирования объекта Y в среде X :

$$U_X^* = \varphi(A_t, X, Y), \quad (3)$$

где φ – алгоритм, позволяющий синтезировать управление по состоянию среды X , объекту Y и критичных обязанностей A_t субъекта, которые изменяются не только под влиянием среды или объекта, но и в связи с жизнедеятельностью субъекта во времени t . Алгоритм позволяет пошагово улучшать управление $U_{N+1} = \varphi(U_N, A_t, X, Y)$ в смысле уменьшения критичного уровня совокупности должностных обязанностей субъекта $A_t(X, Y, U_N) < A_t(X, Y, U_{N+1})$.

В режиме автоматизированного управления осознанное исполнение должностных обязанностей A_t , критичных по времени для реализации цикла административного делопроизводства, достигается декомпозицией алгоритма управления на 2 этапа $A_t \xrightarrow{1} Z^* \xrightarrow{2} U^*$:

1 этап – формирование цели управления – когда субъект интуитивно, экспертным путем, определяет цель управления

$$Z^* = \varphi_1(X, Y, A_t), \quad (4)$$

где φ_1 – алгоритм синтеза цели Z^* по состоянию среды X , объекту Y и критичных по времени исполнения должностных обязанностей A_t ;

2 этап – синтез управления – когда управляющее устройство УУ автоматизированной системы управления СУ алгоритмически, формальным образом, вырабатывает для достижения объектом цели Z^* оптимальную команду управления

$$U_X^* = \varphi_2(Z^*, X, Y), \quad (5)$$

где φ_2 – алгоритм управления, который синтезирует по состоянию среды X и объекту Y команду управления U_X^* , оптимальную для достижения объектом цели Z^* .

Процесс автоматизированного управления заключается в итерационном исполнении в отношении находящихся в производстве дел об административных правонарушениях цикла из 4 фаз управленческой деятельности с использованием АСОУ АД:

- 1) получение от субъекта целеуказания на управление (Z^*);
- 2) получение собранных и накопленных в АСОУ АД данных о результатах управления, характеризующих текущее состояние среды и объекта (X, Y);
- 3) анализ полученной информации и выработка решения на управление (U_X^*);
- 4) реализация управленческого решения путем задействования субъектом в автоматизированном режиме должностных обязанностей и ресурсов административного делопроизводства, нацеленных на активизацию процедур объекта, которые достигли критичного по времени исполнения уровня.

Синтез модели управления состоит из последовательности нижеследующих этапов (рисунок) [2].

На очередной итерации происходит возвращение к предыдущему этапу с целью адаптации под текущие условия.

В части формулирования целей управления, при изменении во времени состояния среды и динамики функционирования объекта, минимизация насущности служебных обязанностей, нормативные сроки исполнения которых истекают, достижения субъектами в условиях стремления к оптимальным характеристикам эффективности административного делопроизводства [3]:

- разумному сроку административного процесса за счет сокращения длительности административных процедур;
- полноте проведения процессуально предусмотренных административных процедур;
- наивысшей собираемости административных штрафов.



Рис. Синтез модели управления

Опыт административной практики в части болевых точек административного делопроизводства указывает на целесообразность установления для модели автоматизированного управления таких экстремальных целей, как:

- 1) минимизация длительности административного производства, когда

$$Z_1 = \sum_{i=1}^M \sum_{j=1}^K t_{ij} \rightarrow \min, \quad (6)$$

где t_{ij} – длительность j -й административной процедуры по i -му делу об административном правонарушении;

- 2) максимальная завершенность административного производства, когда

$$Z_2 = \sum_{i=1}^M \sum_{j=1}^{K'} N_{ij} \rightarrow \max, \quad (7)$$

где N_{ij} – отметка о завершении j -й административной процедуры по i -му делу об административном правонарушении (завершена при $N_{ij} = 1$, не завершена при $N_{ij} = 0$);

- 3) максимальная собираемость административных штрафов, когда

$$Z_3 = \sum_{i=1}^m \sum_{g=1}^n s_{ig} \rightarrow \max, \quad (8)$$

где s_{ig} – сумма оплаты g -й квитанцией по i -му административному штрафу.

Аддитивная свертка (6)-(8) приводит к единому обобщенному критерию оптимизации управления:

$$Z = \sum_{i=1}^M \sum_{j=1}^K [(1 - N_{ij}) * t_{ij}] \rightarrow \min, \quad (9)$$

где N_{ij} – отметка о завершении j -й административной процедуры по i -му делу об административном правонарушении (завершена при $N_{ij} = 1$, не завершена при $N_{ij} = 0$); t_{ij} – длительность j -й административной процедуры по i -му делу об административном правонарушении.

Под объект управления надлежит выделить часть среды, состояние которой субъект может изменить, эффективно исполняя свои должностные обязанности. Тогда для АСОУ АД объектом управления выступает совокупность M производств типовой последовательности из K административных процедур по каждому i -му делу об административном правонарушении.

Объект должен иметь наименьший объем. Это необходимо с целью сокращения трудоемкости его изучения при синтезе модели. При этом существенным ограничением выступает достижимость обобщенной цели управления Z^* в рамках выделенного для этого ресурса R . Это означает, что для любого состояния среды X должно быть найдено управление $U^* \in R$, с помощью которого можно добиться обобщенной цели Z^* , соответствующей множеству поставленных целей $\{Z_k^*\}$.

Таким образом, в качестве объекта управления следует рассмотреть сумму просрочки процессуальных сроков производства активных (находящихся в стадии реализации) административных процедур по каждому i -му делу об административном правонарушении.

Модель F , связывающая входы X и U с выходом Y , определяется структурой объекта и его параметрами $C = \{C_1, \dots, C_k\}$. На этапе структурного синтеза модели определяется структура, т. е. мо-

дель объекта с точностью до значений ее параметров C . Этот этап включает определение внешней структуры модели, декомпозицию модели, определение внутренней структуры элементов модели.

Синтез внешней структуры сводится к содержательному определению входов X и U , а также выхода Y без учета внутренней структуры объекта. В качестве внешнего воздействия среды X выбранный объект управления учитывает текущую дату t . Управляющее воздействие U принуждает завершить активную процедуру или предоставляет объекту дополнительное время θ_j , нормативно определенное КоАП РФ, на исполнение j -ой административной процедуры, на которое уменьшает просрочку Y .

Декомпозиция модели предназначена для того, чтобы, воспользовавшись априорными сведениями о структуре объекта, упростить синтез структуры модели. В результате декомпозиции производства по делу об административном правонарушении получена типовая последовательность административных процедур с временными нормативами их исполнения.

Синтез структуры модели сводится к определению вида оператора F модели объекта с точностью до параметров C :

$$Y = F(X, Y, C), \quad (10)$$

где F – оператор преобразования структуры объекта, характеристики которой внесены в переменные параметры C . Определение внутренней структуры административного процесса приводит (10) к параметризованной модели производства по делам об административных правонарушениях:

$$Y_i = \sum_{j=1}^K [C_{ij} * (t - T_{ij} - \theta_j * p_{ij} - \theta_j * u_{ij})], \quad (11)$$

где Y_i – текущая просрочка исполнения активных административных процедур по i -му делу об административном правонарушении; K – количество административных процедур в типовом процессе; C_{ij} – признак активности (нахождение в стадии реализации) j -й административной процедуры по i -му делу об административном правонарушении; t – текущая дата; T_{ij} – дата начала реализации j -й административной процедуры по i -му делу; θ_j – период продления в сутках j -й административной процедуры, нормативно определенный КоАП РФ; θ_j – период продления в сутках j -й административной процедуры, нормативно определенный КоАП РФ; p_{ij} – количество повторов (предыдущих продлений) j -й административной процедуры по i -му делу об административном правонарушении; u_{ij} – команда на продление или завершение j -й административной процедуры по i -му делу об административном правонарушении.

Идентификация модели управления связана с определением числовых значений параметров $C = \{C_1, \dots, C_k\}$ в режиме нормального функционирования объекта: $C = \{K, C_{ij}, T_{ij}, \theta_j, p_{ij}\}$

В соответствии с декомпозицией производства по делу об административном правонарушении типовой административный процесс насчитывает 16 административных процедур с учетом возможных передач по последованности, возвратов, этапов администрирования доходов от административных штрафов, отсрочек (рассрочек) исполнения административного наказания и исполнительного производства ФССП, т.е. $K = 16$.

В ходе накопления сведений об административном процессе в АСОУ АД ведется таблица состояния административных процедур по делам об административных правонарушениях. В ней на момент последнего сохранения в базе данных реквизитов по i -му делу об административном правонарушении обновляются признак состояния C'_{ij} , время начала реализации T'_{ij} , время окончания реализации T'_{ij} , количество повторов p_{ij} для каждой j -й административной процедуры.

В зависимости от текущих квалификации административного правонарушения, назначенных основного и дополнительного наказаний, а также типа субъекта управления по i -му делу об административном правонарушении, признак состояния C'_{ij} принимает целые значения от -2 до 2:

$C'_{ij} = -2$, если j -я процедура не применяется в i -м процессе;

$C'_{ij} = -1$, если j -я процедура не обязательна в i -м процессе и ее реализация не начата;

$C'_{ij} = 0$, если j -я процедура обязательна в i -м процессе, но ее реализация не начата;

$C'_{ij} = +1$, если в i -м процессе реализация j -й процедуры начата, но не окончена;

$C'_{ij} = +2$, если в i -м процессе окончена реализация j -й процедуры.

Тогда на момент синтеза управления признак активности (нахождение на исполнении) j -й административной процедуры по i -му делу об административном правонарушении принимает значения

$$C_{ij} = \begin{cases} 1, & \text{если } C'_{ij} = +1 \\ 0, & \text{если } C'_{ij} \neq +1 \end{cases} \quad (12)$$

Дата начала реализации j -й административной процедуры по i -му делу об административном правонарушении T_{ij} определяется по времени окончания предыдущей обязательной административной процедуры $T'_{i(j-1)}$: $T_{ij} = T'_{i(j-1)}$

Период θ_j продления j -й административной процедуры устанавливается ведомственными нормативными документами по административной практике, в пределах временных ограничений административного процесса, определенных КоАП РФ.

Количество повторов p_{ij} увеличивают, начиная с 0, на 1 при каждом повторе (продлении) в ходе j -й административной процедуры по i -му делу об административном правонарушении.

При синтезе управления принимается решение о том, каково должно быть управление U , чтобы достигнуть обобщенной цели управления Z^* в объекте. Это решение опирается на имеющуюся модель объекта F , заданную цель Z^* , информацию о состоянии среды X и выделенный ресурс управления R , который представляет собой ограничения, накладываемые на управление U в связи со спецификой объекта и возможностями АСОУ АД.

Исходя из параметризованной модели (11), достижение обобщенной цели Z^* осуществимо выбором управления $\{u_{ij}^* \in U^*\}$, которое нацелено на своевременное завершение или обоснованное продление активных процедур по каждому делу и соответствует решению экстремальной задачи:

$$\sum_{i=1}^M \sum_{j=1}^K [C_{ij} * \theta_j * u_{ij}] \rightarrow \max \rightarrow u_{ij}^* \in \{U^*\} \quad (13)$$

с ограничениями

$$\sum_{j=1}^K [C_{ij} * (t - T_{ij} - \theta_j * p_{ij})] \geq \sum_{j=1}^K [C_{ij} * \theta_j * u_{ij}], \quad (14)$$

где M – количество дел об административных правонарушениях, находящихся в производстве; K – количество административных процедур в типовом административном процессе по делу; C_{ij} – признак активности (нахождение в стадии реализации) j -й административной процедуры по i -му делу об административном правонарушении; t – текущая дата; T_{ij} – дата начала реализации j -й административной процедуры по i -му делу об административном правонарушении; θ_j – период продления в сутках j -ой административной процедуры, нормативно определенный КоАП РФ; p_{ij} – количество повторов (предыдущих продлений) j -ой административной процедуры по i -му делу об административном правонарушении; u_{ij} – команда на продление или завершение j -ой административной процедуры по i -му делу об административном правонарушении:

$$u_{ij} = \begin{cases} 1, & \text{если надо дать команду} \\ 0, & \text{если не нужна команда} \end{cases}$$

Необходимость коррекции (адаптации) управления, сопряженное с периодическим обращением к предыдущему этапу синтеза модели управления, вызывается изменением состояния среды X , а также параметров объекта управления $C = \{C_1, \dots, C_k\}$.

При сохранении характеристик объекта управления $C = \{K, C_{ij}, T_{ij}, \theta_j, p_{ij}\}$ неизменными с предыдущей итерации управления, возвращаются к решению экстремальной задачи синтеза управления (стрелка а), в результате решения которой определяется оптимальное управление $u_{ij}^* \in \{U^*\}$ на текущий момент времени t .

Однако в ходе производства вместе с течением времени t обычно изменяется количество M и состав неоконченных дел об административных правонарушениях, состояние активных процедур по ним.

Поэтому при функционировании АСОУ АД во времени t коррекция управления связана с подстройкой (стрелка б) параметров модели $C = \{K, C_{ij}, T_{ij}, \theta_j, p_{ij}\}$. Такого рода коррекцию называют адаптацией модели, а управление – адаптивным управлением.

Особенностью математической модели автоматизированного оптимального управления адми-

нистративным делопроизводством является тот факт, что искомое оптимальное управление $\{u_{ij}^* \in U^*\}$ принимает дискретные целочисленные значения 0 или 1. Это ограничение придает задаче нелинейный характер, который сопряжен с тем, что минимальное значение целевой функции решаемой задачи больше значения целевой функции соответствующей задачи линейного программирования. Несмотря на конечное множество допустимых решений, явный перебор для нахождения экстремума чрезвычайно затратен и практически не применим для задачи оптимизации с большой размерностью. Однако в связи с тем, что нелинейность порождается лишь условием целочисленности результата, решение можно представить в виде математической оптимизации линейных моделей, на переменные которых последовательно накладывается ограничение целочисленности. Поэтому математическая модель оптимального управления административным делопроизводством реализуется за счет решения полностью целочисленной задачи линейного программирования методами бивалентного программирования, как частного случая целочисленного (дискретного) программирования.

Наибольшее распространение среди комбинаторных методов целочисленного программирования получил метод ветвей и границ, в котором осуществляется последовательное разбиение множества допустимых решений. В нем на каждой итерации строятся 2 задачи, в основе которых лежит «ослабление» задачи предыдущей итерации. При этом, происходит деление пространства решений исходной задачи и отбрасывание области, не содержащей допустимого целочисленного решения [4].

Реализация предложенной модели оптимального управления предоставит руководителю каждого территориального подразделения органа по исполнению административного законодательства механизм для выработки необходимых управленческих решений в части продления или завершения административных процедур по находящимся в производстве делам об административных правонарушениях, чем обеспечит осуществление административного делопроизводства в разумный срок.

Библиографический список

1. *Миронов А. Ю.* Автоматизированная система управления производством по делам об административных правонарушениях на региональном уровне // Шестьдесят девятая международная студенческая научная конференция ГУАП: Сб. докл.: В 2 ч. Ч. I. Технические науки / СПб.: ГУАП, 2016. С.314-318.
2. *Меньков А.В., Острейковский В.А.* Теоретические основы автоматизированного управления: учебник для вузов. М: Изд-во Оникс, 2005. 640 с.
3. *Дугенец А.С., Масленников М.Я.* Сроки рассмотрения дел об административных правонарушениях: правовое значение и условия соблюдения // Научный портал МВД России. М.: ВНИИ МВД России, 2009. С.13-21.
4. *Михайлов В.С.* Теория управления. Киев.: Выща шк. Головное изд-во, 1988. 312 с.

УДК 004.04.09

Н.О. Мухортов

студент кафедры проблемно-ориентированных вычислительных комплексов

В.С. Павлов

д-р техн. наук, профессор – научный руководитель

ОБОСНОВАНИЕ ПРОГРАММНОГО ШАБЛОНА ИНФОРМАЦИОННО-КОММЕРЧЕСКИХ ВЕБ-СИСТЕМ МАЛОГО БИЗНЕСА

Представители малого бизнеса видят в обращении к веб-ресурсам дополнительную возможность для популяризации и развития своего бизнеса, либо для создания его «с нуля». Наибольшей популярностью, безусловно, пользуются площадки для продажи товаров – интернет-магазины.

Говоря о информационно-коммерческих веб-системах малого бизнеса, подразумевают различные веб-системы, в частности интернет-магазины, созданные на основе шаблонно-ориентированного подхода проектирования программных систем. Система управления веб-контентом (ContentManagementSystem: CMS) – функциональный инструмент, сильно упрощающий процесс разработки веб-системы и публикации информации на нем. Оформление и предлагаемая информация на веб-странице не зависят от процесса создания контента. Это происходит из-за того, что веб-страницы и навигация в самой веб-системе создается динамически из информации, содержащейся в базе данных [1]. Из всех веб-систем, зарегистрированных в сети интернет, чуть более 30% используют CMS. Далее представлен график, показывающий динамику роста доли веб-систем на CMS, начиная с сентября 2015 года по сентябрь 2016 года. В плане роста числа веб-систем, использующих CMS, тенденция является положительной потому, что эти инструменты действительно облегчают весь процесс работы [2].

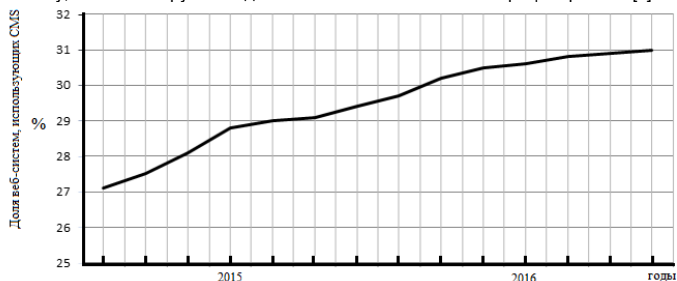


Рис. 1. Динамика использования CMS в веб-системах

Говоря о типизированной разработке веб-ресурсов на основе CMS, стоит отметить, что в ней присутствуют как положительные, так и отрицательные стороны [3].

К положительным сторонам можно отнести:

- возможность самостоятельного наполнения веб-системы;
- возможность распределенного администрирования ресурса;
- структурированность информации;
- быстрое создание ресурса «с нуля»;
- удобство использования;
- отсутствие проблем с поддержкой системы из-за ее популярности.

Стоит также учесть и отрицательные стороны использования подобных систем:

- неоптимизированность в внутренней структуры, большое количество копий страниц;
- непонимание работы подобных систем без должного образования;

- громоздкость системы, лишний функционал;
- низкий уровень безопасности из-за однотипности структуры;
- зависимость от обновлений и расширений, отсутствие обратной поддержки;
- требовательность к ресурсам, низкий уровень загрузки контента;
- удорожание разработки веб-системы за счет более сложного программирования;
- последующие доработки веб-системы также сопряжены с правкой CMS;
- плата за лицензию (в случае платных CMS).

Среди них можно выделить проприетарные программные решения и решения с открытым исходным кодом. В табл. 1 и 2 представлена информация (в период с сентября 2015 года по сентябрь 2016 года) о количестве проектов, созданных на популярных CMS и процентном соотношении их использования. Проприетарные CMS создаются для получения прибыли от продажи лицензии и технической поддержки.

Таблица 1

Статистика проприетарных CMS

Наименование	Проекты (количество)	Процентиспользования
1-С Битрикс	17 759	77.80%
UNI.CMS	4 210	8.78%
NetCat	2 679	6.35%

Стоит отметить, что с явным преимуществом лидирует 1-С Битрикс, по праву занимая первую строчку [4]. В пользу этой CMS говорит развитая система поддержки и глобальное сообщество пользователей. При этом она имеет и свои недостатки. Один из этих недостатков состоит в том, что система обширна и для ее эффективного использования существуют специализированные курсы с соответствующей сертификацией специалистов.

Свободно распространяемые CMS – отличаются открытостью исходного кода, что делает их привлекательными для доработки и публикации улучшений.

Таблица 2

Статистика свободнораспространяемых CMS

Наименование	Проекты (количество)	Процентиспользования
WordPress	7 560	32.59%
Joomla	7 430	30.53%
Drupal	6 350	29.86%

Из табл. 2 видно, что процент указанных CMS примерно одинаков. В табл. 3–5 приведены более подробные характеристики.

Так как разрабатываемый проект будет являться проектом с открытым исходным кодом и распространяться бесплатно, целесообразно более подробно рассмотреть частные случаи наиболее популярных свободно распространяемых CMS.

Таблица 3

Анализ CMS WordPress

Плюсы	Минусы
Удобная панель администратора	Большая нагрузка на сервер
Поддержка различными сообществами	Низкая скорость загрузки страниц
Большая возможность для расширения	Плохой отклик поисковых систем
Возможность вносить изменения в код	Низкий уровень защиты информации
Бесплатность системы	Нелогичная структура БД
	Некачественные темы и дополнения
	Сложности разработки (верстка, программирование)

Таблица 4

Анализ CMS Joomla	
Плюсы	Минусы
Удобная панель администратора	Дублирование страниц и изображений
Мультиязычность	Большинство расширений платные
Большая возможность для расширения	Часто подвергается взлому
Возможность вносить изменения в код	Большая нагрузка на сервер, долгая скорость загрузки
	Плохой отклик поисковых систем

Таблица 5

Анализ CMS Drupal	
Плюсы	Минусы
Приятный интерфейс админ панели	Большая нагрузка на сервер
Удобное разделение на категории	Низкая скорость загрузки страниц
Возможность для расширения	Запутанная структура шаблона
Полностью бесплатна	Низкий уровень защиты информации
Частое обновление и устранение ошибок	Невозможность установки модулей без навыков программирования
	Проблемы с локализацией
	Плохой отклик поисковых систем

Из приведенной выше информации видно, что наиболее популярные CMS [5], [6], [7] имеют много схожие недостатки в виде их громоздкости, низкой безопасности, большой нагрузки на сервер, низкой скорости загрузки страниц, лишнего функционала, отсутствия оптимизированности, некачественного представления желаемого функционала (дублирование страниц, отсутствие изображений), плохого отклика поисковых систем. Все это в значительной степени влияет на работу, реализованного на основе CMS интернет-магазина и впечатление о нем.

Анализируя табл. 3-5, можно вывести определенную закономерность:

1) CMS реализуют однотипную систему защиты, а значит, и соответствующие типовые ошибки. Использование типовых программных решений для реализации CMS часто приводит к взлому и риску потери контроля над разработанной веб-системой, а также утрате личных данных пользователей.

2) Макеты веб-систем для шаблонов CMS имеют сложную, а также различную структуру – это значит, что без необходимых знаний в программировании невозможно создать индивидуальный в дизайнерском и функциональном отношении магазин.

3) Использование модулей и расширений для CMS подразумевает наличие ошибок, что может ухудшить функциональные возможности реализуемой веб-системы, а также заметно понизить целый ряд показателей, таких как: нарушение целостности отображаемой страницы, увеличение нагрузки на сервер, увеличение времени загрузки страницы, а также сделать невозможным трансляцию как отдельных страниц, так и веб-системы в целом.

Уместно сказать, что без достаточных знаний в программировании невозможно создать качественный и функциональный проект даже на основе расширений CMS, так как разработчику не избежать вмешательства в создаваемый программный код.

Исходя из этого, следует, что одной из основных задач разрабатываемого программного шаблона веб-систем малого бизнеса является устранение существующих недостатков, рассмотренных CMS.

Эти недостатки можно разделить на три основные подкатегории, которые представлены на рис. 2 [8].



Рис. 2. Схема выявленных недостатков

В разрабатываемом программном шаблоне веб-системы малого бизнеса предполагается устранение этих недостатков путем разработки не типизированной системы защиты информации на клиентской и серверной сторонах.

Создание логичной структуры программного шаблона веб-систем малого бизнеса для представления информации позволит снизить нагрузку на сервер и гораздо быстрее загружать страницы с созданным контентом.

В плане работы с системой предполагается позволить пользователю вносить изменения в реализуемый проект и, создавая представление информации, основываясь на внесенных данных в базу данных (БД) [9], не имея при этом дополнительных знаний относительно веб-разработки.

За счет предельно понятной структуры кода разрабатываемый шаблон будет отлично индексироваться поисковыми системами.

В дополнение к ранее перечисленным возможностям CMS будет иметь логичную структуру БД, что позволит иметь максимально понятную структуру хранения информации, к которой можно будет обращаться из других приложений[10].

Вследствие этого целью работы является обоснование разработки программного шаблона информационно-коммерческих веб-систем малого бизнеса для дальнейшей эксплуатации в сети интернет.

Главными задачами разрабатываемого программного шаблона веб-системы малого бизнеса являются:

- разработка шаблона веб-приложения;
- разработка шаблона БД;
- реализация обработки заказа и проверки достоверности;
- реализация просмотра и непосредственной продажи товаров в приложении;
- реализация шаблона корзины и удобного оформления заказа;
- реализация шаблона административной панели, с необходимым уровнем защиты.

Данный программный шаблон веб-системы малого бизнеса должен позволять управлять веб-системой в максимально удобной форме. В результате должен получиться эффективный и прогрессивный программный шаблон веб-системы малого бизнеса, с помощью которого возможно будет осу-

ществлять комфортное управления всеми товарами, заказами, пользователями, оплатой, а также статистикой.

Реализация основных требований при разработке программного шаблона веб-системы малого бизнеса позволит при внедрении:

- получать непосредственное построение интернет-магазина из БД;
- обеспечить доступ покупателя к полному каталогу продаваемой продукции;
- также должно присутствовать разделение по категориям товаров;
- следить за экономическими операциями с возможностью получения данных и статистики;
- осуществлять регистрацию и авторизацию пользователей;
- должна быть возможность добавлять новые товары в административной панели изменять и удалять их;
- получать информацию о пользователях и заказах из административной панели;
- должна быть возможность оформить заказ прямо в веб-системе;
- должна присутствовать возможность добавлять новые товары в корзину, удалять их;
- обеспечение защиты информации;
- создание английской версии веб-системы;
- возможность полноценного тестирования шаблона;
- обеспечение совместимости с различными системами;
- обеспечить быструю загрузку страниц и минимальную нагрузку на сервер;
- предупреждение негативных сценариев.

Разрабатываемый программный шаблон веб-системы малого бизнеса должен предоставлять для пользователя удобный просмотр товара и оформления заказа, а также возможность для изменения интернет-магазина через административную панель. На рис. 3 представлен программный шаблон веб-системы малого бизнеса с заполненной БД и активными функциональными элементами.

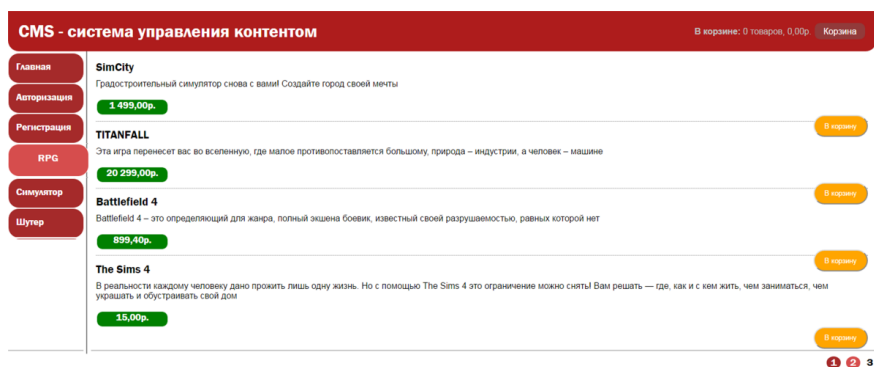


Рис. 3. CMS с заполненной БД и активными элементами навигации

Вывод:

На основе анализа использования CMS для создания веб-систем малого бизнеса были выявлены их недостатки, для устранения которых была обоснована разработка программного шаблона веб-систем малого бизнеса, а также представлен пример реализации, устраняющий проанализированные недостатки.

Библиографический список:

1. *Christianson C., Cochran J.* ASP. Net Content Management System Development. СПб.: БХВ 2011. 25 с.

2. Рейтинг Рунета. URL: <http://www.ratingruneta.ru/raiting-cms>
3. Портал для веб-мастеров. URL: <https://www.wwwmaster.ru/60-plyusy-i-minusy-besplatnyh-cms>
4. 1С-Битрикс. Система управления интернет-проектами. URL: <http://www.1c-bitrix.ru/>
5. Русский WordPress. URL: <https://ru.wordpress.org/>
6. Все о Joomla. URL: <http://joomla.ru/>
7. Drupal в рунете. URL: <http://www.drupal.ru/>
8. *Гамма Э., Хелм Р., Джонсон Р., Влссидес Дж.* Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: ПИТЕР. 2008. 150 с.
9. *Прохоренок Н.* АНТМЛ, javascript, PHP и MySQL. Джентльменский набор Web-мастера. СПб.: БХВ. 2011. 205 с.
10. *Ташков П.А.* Веб-мастеринг: HTML, CSS, javascript, PHP, CMS, AJAX, раскрутка. СПб.: БХВ. 2010. 125 с.

УДК 004.623

Д. В. Никулин

студент кафедры 43 компьютерной математики и программирования

А. В. Фомин

канд. техн. наук, доцент – научный руководитель

ПРИМЕНЕНИЕ REST API В МОБИЛЬНОМ ПРИЛОЖЕНИИ ДЛЯ ИЗУЧЕНИЯ ИНОСТРАННЫХ ЯЗЫКОВ

Современные информационные системы часто строятся с клиентом в виде мобильного приложения, так как это удобно для пользователей. Серверная часть системы должна быть эффективной и учитывать требования, предъявляемые со стороны клиентской части. Для реализации обмена данными сервер должен предоставлять API, которое позволит обращаться к различным функциям сервера по сети. Существует множество различных подходов к обмену данными между клиентом и сервером, обладающих различными свойствами, преимуществами и недостатками. В настоящее время одним из наиболее популярных подходов для реализации API как для мобильных, так и для Web-приложений является REST.

REST – это архитектурный стиль взаимодействия клиента и сервера. Аббревиатура расшифровывается как Representational State Transfer, то есть «передача состояния представления». Это означает, что сервер не хранит состояние клиента и предоставляет клиенту информацию о состоянии запрошенного ресурса [1]. REST подразумевает использование протокола HTTP для обмена данными между клиентом и сервером и использует различные методы HTTP-запросов. Как правило, используются четыре HTTP-метода для реализации действий CRUD (англ. Create, Read, Update, Delete – создание, чтение, обновление и удаление ресурсов). REST-архитектура накладывает определённые ограничения на приложение. Чтобы сервис считался соответствующим REST-архитектуре (RESTful), обязательно необходимо соблюдение пяти условий:

1. Архитектура клиент-сервер. Предполагает разделение ответственности между клиентом и сервером.
2. Отсутствие состояния. Это условие подразумевает, что сервер не хранит состояние клиента между запросами. Вся необходимая серверу информация передается в теле запроса.
3. Кэширование. Некоторые типы запросов могут быть кэшируемыми, то есть ответ сервера может быть сохранён клиентом или промежуточным узлом. Сервер должен явно указывать, какие запросы можно кэшировать и на какой срок. Для управления кэшированием используются заголовки ответа Expires (устанавливает дату, до которой кэшированный ответ можно считать актуальным) и Cache-Control.
4. Единообразие интерфейса. К интерфейсу применяются следующие ограничения:
 - Идентификация ресурсов в URI.
 - Манипуляции над ресурсами через представления. Когда пользователь имеет представление о ресурсе, он имеет достаточно информации для изменения или удаления ресурса на сервере.
 - Самодокументируемые сообщения. По информации, передаваемой в теле запроса, должно быть понятно, как этот запрос можно обработать.
5. Многоуровневость системы. Клиент не может определить, общается ли он с сервером напрямую, или с некоторым промежуточным узлом. Это позволяет повысить гибкость и масштабируемость системы за счёт возможности балансировки нагрузки и кэширования на промежуточных узлах.

Также, есть одно необязательное свойство REST-архитектуры – «код по требованию», что означает возможность передачи сервером дополнений к клиентской логике. Примером может служить передача JavaScript-сценариев для браузера [1]. Для передачи данных, как правило, используются такие форматы, как JSON и XML с соответствующими заголовками Content-Type [3]. Для обозначения статуса операции используются стандартные статус-коды протокола HTTP.

Применение REST при разработке серверной части мобильного приложения

REST-подход был выбран, так как он относительно прост в использовании и работает с протоколом HTTP, для поддержки которого есть множество библиотек. В качестве формата передачи данных был выбран JSON по нескольким причинам:

- Формат более лаконичен по сравнению с XML, данные, передаваемые в формате JSON будут занимать меньший объём.
- Формат хорошо читаем человеком, что означает более простую отладку работы приложения.
- Сервер реализован на Node.js, то есть написан на языке JavaScript, на котором удобно взаимодействовать с данными в формате JSON. В языке есть встроенные средства для работы с JSON (методы `JSON.parse` и `JSON.stringify`) [2].
- Хорошая поддержка формата на клиентской стороне (платформа iOS и язык Swift).

В виде JSON передаётся как тело запроса, так и тело ответа сервера. Для всех запросов предусмотрен общий формат ответа. Сервер отвечает в формате JSON, с соответствующим заголовком `Content-Type`, равным `«application/json»`. В зависимости от успеха запроса, возможны два варианта ответа:

- Успешный запрос. Ответ всегда содержит поле `success`, равное `true` и имеет код ответа HTTP, начинающийся с 2 (как правило, 200). Опционально может содержать поле `data` с возвращаемыми данными, если они есть. Поле «данные ответа» в описании конкретных методов описывает содержимое поля `data`. Пример ответа на успешный запрос `POST /login`:

```
{
  «success»: true,
  «data»: {
    «token»: «vtob16Z0iuabEae/5djJkkpeBe0v2TGP+0zh3lWjhwY=«
  }
}
```

- Запрос, завершившийся ошибкой. Всегда содержит поле `success`, равное `false` и имеет код ответа HTTP, начинающийся с 4 или 5. Обязательно имеет поле `error` с полями `code` и `message`, означающими код ошибки и её словесное описание соответственно. Пример ответа на запрос `POST /login` с некорректными логином/паролем:

```
{
  «success»: false,
  «error»: {
    «code»: 1,
    «message»: «User with provided login/password pair does not exist»
  }
}
```

Все запросы, кроме `/login` и `/register`, могут в качестве `query`-параметра принимать токен авторизации. Для некоторых методов авторизация опциональна, так как приложение подразумевает необязательную регистрацию.

Для защиты конфиденциальных пользовательских данных, таких как адреса e-mail и пароли, используется шифрованный протокол HTTPS с версией транспортного протокола TLS не ниже 1.1.

Для примера ниже описаны запросы для работы с некоторыми сущностями приложения:

- `User` – описывает пользователя системы. Имеет связь 1:N к `UserType`, то есть у пользователя задается тип – системный, администратор или обычный пользователь приложения.

Запросы для работы с пользователями:

- `POST /register` – регистрация пользователей. В теле запроса передаются e-mail пользователя и пароль, которые он указывает в форме регистрации. Стоит заметить, что сущность пользователя,

создаваемая таким запросом, будет иметь тип обычного пользователя. Пользователя с правами администратора можно создать только при помощи скрипта при наличии доступа к БД.

В ответе не присылается дополнительных данных, помимо статуса операции. Если пользователь с таким e-mail уже есть, то код ответа будет 409 «Conflict», то есть конфликт с уже существующим ресурсом.

- `POST /login` – вход пользователя. В теле запроса передаются e-mail пользователя и пароль, указанные при регистрации. Если пара логин/пароль совпадает с указанной при регистрации, то сервер возвращает код 200 «OK» и токен – закодированную в формате base64url последовательность длиной 32 байта, используемую для проверки подлинности в остальных запросах. В случае, если данные не совпали, присылается код ответа 401 Unauthorized.

- `GET /admin/users` – получение всех пользователей. Доступно только администраторам. При этом хэш пароля не присылается.

- `PUT /admin/users/<id>` – обновление пользователя. Обновить можно тип пользователя, флаг подтверждения и статус наличия платного аккаунта. Доступен только администраторам.

- `Category` – категория слов. Имеет связи N:M с `Language` и `Word`, то есть в категорию входит набор слов и у категории есть названия в разных языках (`CategoryValue`). Создание новых категорий доступно администраторам и пользователям, купившим платный аккаунт. При этом категория, созданная администратором, будет видна всем и в качестве автора будет иметь системного пользователя, а категория, созданная пользователем, будет видна только ему и в качестве автора будет прописан пользователь.

Запросы для работы с категориями:

- `GET /categories` – получение списка категорий без вложенных слов.

- `POST /categories` – создание новой категории. Для использования этого метода обязательна авторизация.

- `PUT /categories/<id>` – обновление данных о категории `<id>`. При обновлении нельзя изменить пользователя категории, т. е. поле `userId` просто игнорируется. Доступно только создателю категории или администратору. В случае отсутствия прав доступа возвращается код ответа 403 Forbidden.

- `GET /categories/<id>/words` – получение слов, входящих в категорию.

- `POST /categories/<id>/words` – добавление слова в категорию.

- `DELETE /categories/<id>/words/<wordId>` – удаление слова из категории. Доступно только создателю категории или администратору. В случае отсутствия прав доступа возвращается код ответа 403 «Forbidden».

- `DELETE /categories/<id>` – удаление категории. Доступно только создателю категории или администратору. В случае отсутствия прав доступа возвращается код ответа 403 «Forbidden».

В результате, с применением архитектурного стиля REST был создан сервер для информационной системы, который предоставляет универсальный API как для мобильного клиента, так и для веб-панели администратора, которая обращается к API при помощи AJAX-запросов. Применение REST позволило достичь переносимости и простоты серверной части.

Библиографический список

1. REST in Practice. Hypermedia and Systems Architecture / by Jim Webber, Savas Parastatidis, and Ian Robinson – O'Reilly Media, 2010

2. Спецификация ECMAScript 5.1 [электронный ресурс] URL: <http://www.ecma-international.org/ecma-262/5.1/>

3. RESTful Web Services. Web services for the real world / by Leonard Richardson, Sam Ruby – O'Reilly Media, 2007

УДК:004.052.44

Д.В. Оганян

студент группы 4512М

Е.А. Бакин

научный руководитель

СРАВНЕНИЕ АЛГОРИТМОВ ДЕКОДИРОВАНИЯ КОДОВ-ПРОИЗВЕДЕНИЙ

В наш век информационных технологий немислимо существование без повседневного использования электронных устройств. Абсолютное большинство из них – цифровые устройства. При передаче цифровых данных через канал существует возможность возникновения ошибки в последовательности. В связи с этим цифровую последовательность обычно кодируют, чтобы получатель смог узнать, произошла ли при передаче ошибка, или нет, а также, в некоторых случаях, исправить ее. Такое кодирование называют канальным кодированием, или помехоустойчивым.

Что же такое коды-произведения? При обычном кодировании в качестве сообщения выбирается блок определенной длины. К данному блоку добавляются еще несколько проверочных символов, которые определяют избыточность кода. По этим проверочным символам и определяется наличие, и позиционирование ошибочного бита. В случае кодов-произведений, за передаваемый блок берется матрица, каждая строка которой является сообщением. Данная матрица кодируется как по строкам, так и по столбцам. В кодах-произведениях увеличивается избыточность, но также увеличивается надежность. Для анализа алгоритмов декодирования кодов-произведений в данной работе используется код Хэмминга, который позволяет идентифицировать и исправить одну ошибку. Пример кода-произведения представлен на рис. 1. В данном примере черной рамкой выделен информативный блок передаваемой матрицы, а красной – избыточный блок.

$$r = \begin{bmatrix} m_{1,1} & m_{1,2} & \dots & m_{1,k} & c_{1,n-k} & c_{1,n-k+1} & \dots & c_{1,n} \\ m_{2,1} & m_{2,2} & \dots & m_{2,k} & c_{2,n-k} & c_{2,n-k+1} & \dots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{k,1} & m_{k,2} & \dots & m_{k,k} & c_{k,n-k} & c_{k,n-k+1} & \dots & c_{k,n} \\ c_{n-k,1} & c_{n-k,2} & \dots & c_{n-k,k} & c_{n-k,n-k} & c_{n-k,n-k+1} & \dots & c_{n-k,n} \\ c_{n-k+1,1} & c_{n-k+1,2} & \dots & c_{n-k+1,k} & c_{n-k+1,n-k} & c_{n-k+1,n-k+1} & \dots & c_{n-k+1,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \dots & c_{n,k} & c_{n,n-k} & c_{n,n-k+1} & \dots & c_{n,n} \end{bmatrix}$$

Рис. 1. Пример кода-произведения

Существуют несколько алгоритмов декодирования. Основным является алгоритм синдромного декодирования. Данный алгоритм обычно применяется в ДСК-канале, так как работает только с бинарными сообщениями. Алгоритм заключается в следующем. В начале определяются синдромы строк принятой матрицы. Если при передаче в декодируемом векторе не произошло ошибки, синдром будет равен нулю. В противном случае, синдром отличен от нуля. По определенной заранее таблице синдромов определяется и исправляется бит, в котором возникла ошибка. Такие же действия необходимо произвести повторно, только теперь в качестве векторов выступают не строки, а столбцы принятой матрицы. Важно отметить, что алгоритм декодирования является итерационным, т.е. при повторном декодировании результат улучшается. Блок-схема алгоритма синдромного декодирования представлена на рис. 2,а.

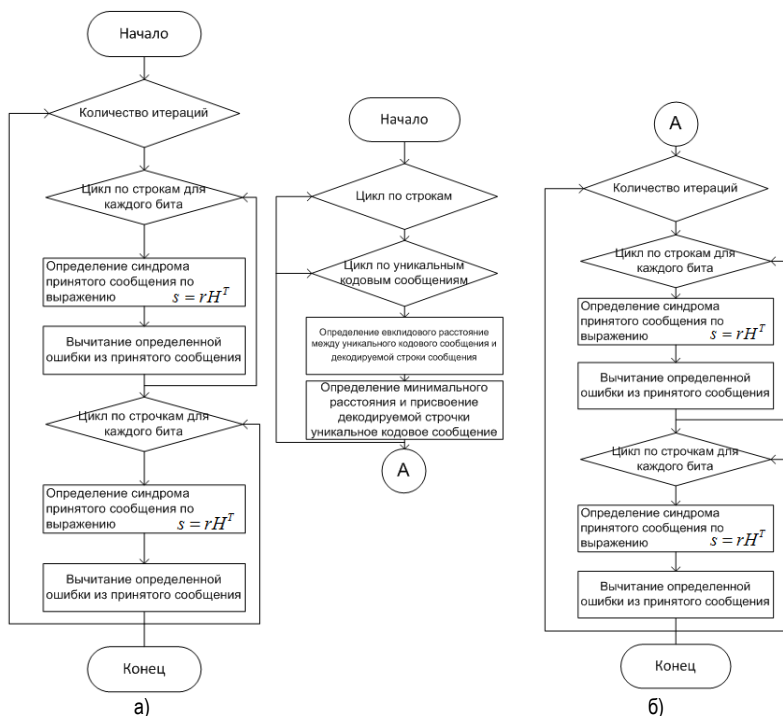
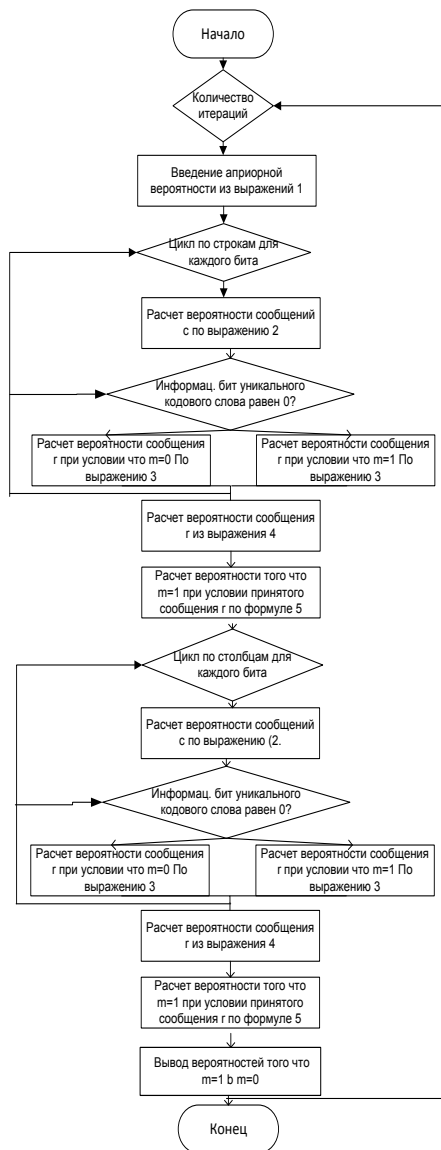


Рис. 2. Блок-схема: синдромного алгоритма декодирования (а), мягкого алгоритма декодирования (б)

Второй алгоритм – алгоритм мягкого декодирования применяется в АБГШ каналах, и состоит из 2-х частей. В первой части, из числа всех уникальных кодовых сообщений выбирается наиболее похожее с точки зрения евклидова расстояния. После этого принятая матрица принимает бинарный вид, и к данной матрице применяется синдромный алгоритм декодирования описанный ранее. Блок-схема мягкого алгоритма декодирования представлена на рис. 2.б.

Третьим является алгоритм турбо-декодирования (рис. 3). В результате алгоритма турбо- декодирования определяется не значение бита, а апостериорная вероятность значения 0 или 1 в конкретном информативном символе. Эта вероятность определяется согласно формулам и алгоритму представленным на рис. 3. Так же необходимо отметить, что как и синдромное декодирование, алгоритм турбо-декодирования является итерационным. При каждой следующей итерации, априорная вероятность заменяется на апостериорную вероятность. В теле алгоритма приводятся ссылки на выражения представленные справа.

По результатам моделирования замечено, что синдромное декодирование в ДСК канале, показывает лучший результат, чем турбо-декодирование, выигрыш составляет 10 раз. Также замечено, что качество декодирования в обоих методах улучшается при проведении нескольких итераций. При синдромном декодировании вторая итерация значительно увеличивает количество правильно декодированных символов, а третья не дает преимуществ. Вторая итерация при турбо-декодировании также увеличивает количество правильно декодированных бит, но не так значительно, а при третьей итерации, результат не улучшается.



$$p_{i,j}^a = P\{m_{i,j}=1\} = \frac{1}{2} \quad 1)$$

$$q_{i,j}^a = P\{m_{i,j}=0\} = \frac{1}{2} \quad 1)$$

$$P\{c_i\} = \prod_{i=1}^k P\{m_i = c_i\} \quad 2)$$

$$P\{r_i | m_{i,j}=1\} = \sum_{c_i=1}^k \prod_{k=c_i} P^{I\{r_i[k]=c_i\}} (1-P)^{I\{r_i[k] \neq c_i\}} P\{c_i\}$$

$$P\{r_i | m_{i,j}=0\} = \sum_{c_i=0}^k \prod_{k=c_i} P^{I\{r_i[k]=c_i\}} (1-P)^{I\{r_i[k] \neq c_i\}} P\{c_i\}$$

$$P\{r_i | m_{i,j}=1\} = \sum_{c_i=1}^k \prod_{i=1}^k \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(r_i-c_i)^2}{2\sigma^2}} \quad 3)$$

$$P\{r_i | m_{i,j}=0\} = \sum_{c_i=0}^k \prod_{i=1}^k \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(r_i-c_i)^2}{2\sigma^2}}$$

$$P\{r_i\} = P\{r_i | m_{i,j}=1\}P\{m_{i,j}=1\} + P\{r_i | m_{i,j}=0\}P\{m_{i,j}=0\} \quad 4)$$

$$P\{m_{i,j}=1 | r_i\} = \frac{P\{r_i | m_{i,j}=1\}P\{m_{i,j}=1\}}{P\{r_i\}} \quad 5)$$

Рис. 3. Блок-схема и формулы для алгоритма турбо-декодирования

Результаты моделирования в канале ДСК представлены на рис. 4.

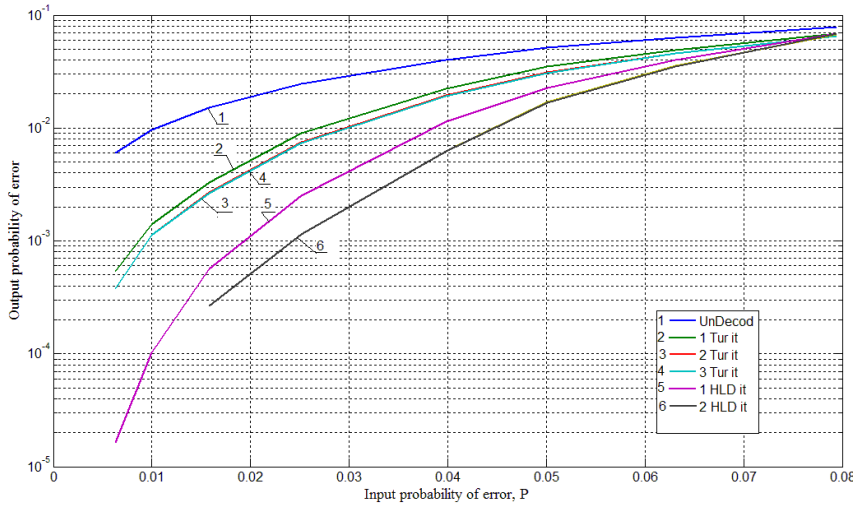


Рис. 4. Результаты моделирования в ДСК канале

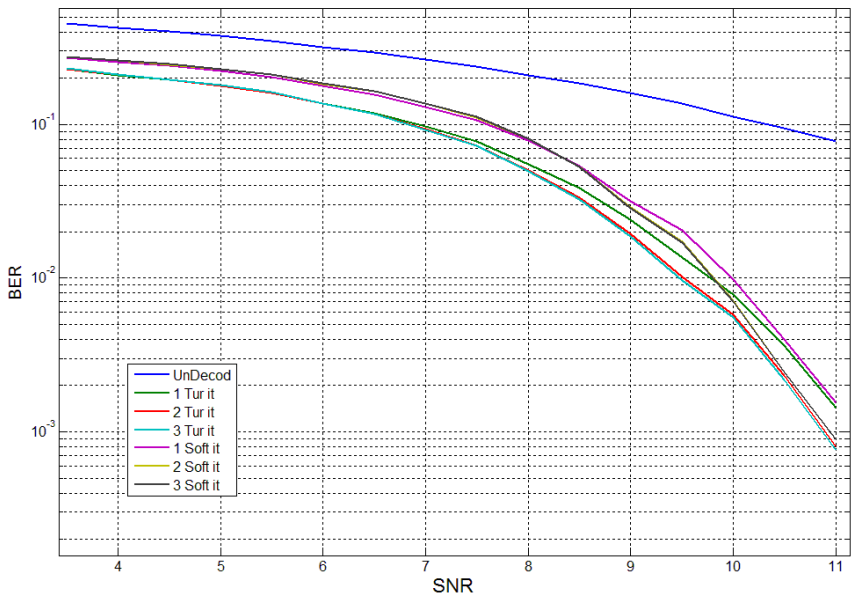


Рис. 5. Результаты моделирования в АБГШ канале

Так же замечено, что в канале АБГШ, после значения отношения сигнал-шум 8 дБ наилучший результат показывает мягкое декодирование. Замечено, что каждая итерация значительно улучшает оба метода при малых значениях интенсивности шума. Результаты моделирования в канале АБГШ представлены на рис. 5.

Библиографический список

1. Бернад Скляр. Цифровая связь. 2003.
2. Зюко А.Г., Кловский Д.Д. Теория электрической связи. 1999.
3. Васильев К.К., Глушков В.А., Дормидонтов А.В., Нестеренко А.Г. Теория электрических связи: учеб. пособие. 2008.

УДК 732

Павлова М.В.

студентка

М.Б. Игнатъев

д-р техн. наук, профессор – научный руководитель

АНАЛИЗ СТРУКТУРЫ ДЕЯТЕЛЬНОСТИ ПО ОБСЛУЖИВАНИЮ ИНВАЛИДОВ

Аннотация. Инвалиды как часть современного общества имеют большие проблемы с получением информации о принимаемых в их отношении решениях, о культурных событиях и решениях государственных органов власти. Необходимым признаком демократического общества всегда является открытость информирования населения о деятельности органов власти, и здесь большую роль призваны играть библиотеки. Современное информационное общество определяется как ориентированное на интересы людей, открытое для всех и направленное на формирование инновационной модели развития, высокотехнологичного общества, в котором каждый гражданин имеет возможность создавать и накапливать информацию и знания, иметь к ним свободный доступ, пользоваться и обмениваться ими, чтобы дать возможность каждому человеку в полной мере реализовать свой потенциал для обеспечения личного и общественного развития и повышения качества жизни, а одним из инструментов развития информационного общества является электронное управление. Среди задач, внедрение которых законодательно задекларировано, – необходимость долговременного хранения электронных документов в государственных архивах, музеях, библиотеках, поддержания их в актуализированном состоянии и предоставления доступа к ним. Таким образом, библиотеки должны стать активными участниками процесса внедрения системы электронного управления.

Ключевые слова: информационные системы, инвалиды, хранение документов, библиотечная система, ограниченные возможности, удаленное управление.

Реализация подобного проекта для граждан с ограниченными физическими возможностями создает перспективу привлечения интерактивного общения с органами государственной власти и органами местного самоуправления. В этом контексте неотложным требованием сегодняшнего дня является разработка принципиально новых компьютерных технологий доступа к информационным ресурсам, нового программного обеспечения, обустройства в библиотеках современных компьютеризированных рабочих мест, ориентированных именно на такую категорию пользователей. Болезненным этот вопрос является для лиц с недостатками зрения, библиотечное обслуживание которых набирает актуальность, ведь потеря человеком возможности воспринимать зрительную информацию справедливо считается одной из самых тяжелых форм потери здоровья. Отметим, что в стране насчитывается 250 тысяч человек с недостатками зрения, которые нуждаются в специализированных компьютерных средствах и использовании специфических технологий получения информации. В целом в мире, по данным Всемирной организации здравоохранения, по состоянию на июнь 2017 года насчитывается 45 миллионов полностью слепых, 90% из которых проживают в развитых странах. Ежегодно эта цифра увеличивается на 2 млн человек и наблюдается четкая тенденция к ее росту [1, 2, 3].

К решению проблемы предоставления доступа к информации пользователям с особыми потребностями сегодня активно приобщаются библиотеки любых уровней и подчинений как посредники в общении власти и граждан. Но приходится констатировать, что лишь незначительное количество из них могут удовлетворить информационные потребности таких читателей. Для пользователей этой категории, в соответствии с анализом, представленным на основе исследования, проведенного Всероссийской библиотечной ассоциацией, лишь 26 компьютеров, среди которых 17 находятся в областных библиотеках, 9 в городских и районных [4]. Причем географически эти библиотеки

расположены только в ЦФО. Всероссийское общество слепых удерживает 79 библиотек для незрячих по всей Украине, фонды которых не пополнялись новинками, напечатанными шрифтом Брайля на протяжении двух последних лет.

Библиотеки различного подчинения, при соответствующей организационной работе, должны стать центрами в системе социокультурной реабилитации инвалидов по зрению. Основная функция библиотек, которые причастны к обслуживанию этой категории пользователей, заключается в предоставлении доступа к информации с помощью современных технических средств информационных технологий. Сегодня к этому процессу активно подключается ряд библиотек и, в частности РГБ. В ней с помощью шведских партнеров из Академии Свефе (г. Хапаранда) оборудованы компьютеризированные рабочие места для пользователей с нарушениями зрения с подключением к принтеру, осуществляющего печать шрифтом Брайля и так называемого брайлевского дисплея и DAISY-плеера.

К исследованию этого вопроса обращалось немало библиотекаев. Роль публичных библиотек в процессе общения с органами государственной власти с использованием современных технологий, в частности Интернет, с учетом задач, стоящих сегодня перед библиотеками, определение проблем и путей их решения, возрастание роли публичных библиотек в государственных, демократических процессах, содействие осведомленности граждан, обеспечения органов государственной власти информацией, способствующей принятию управленческих решений рассматривались в трудах Л. Ковальчук [5], В. Пашковой [6] и др.

Определенный фундамент научному изучению проблемы библиотечного обслуживания инвалидов по зрению и дефектологического библиотековедения заложено в труде А. Шапошникова [7]. Продолжили научные поиски в этом направлении российские ученые – это труды С. Ванишина, Г. Диянской, Д. Жаркова, Л. Сучковой. Впоследствии в исследовании А. Литвина [8] сделана попытка обобщить проблемы организации библиотечно-информационного обеспечения пользователей с недостатками зрения, однако в нем лишь незначительное внимание уделяется новейшим технологиям в этой отрасли.

В предлагаемом исследовании мы остановимся на особенностях библиотечного обслуживания читателей с недостатками зрения книгами, которые «разговаривают».

Книга «разговаривает», которая уже давно служит для людей с инвалидностью по зрению основным средством получения информации, способствует развитию личности и стала незаменимой опорой образования, самообразования, информационно обогащает досуг. Эта технология прошла значительный путь становления в условиях развития и совершенствования технических средств производства и воспроизводства, смены носителей и постоянного роста спроса на продукцию, что «разговаривает», особенно в условиях катастрофического дефицита изданий, напечатанных шрифтом Брайля (до 2009-го на год выходило в свет только 15 названий книг для незрячих читателей, сегодня издание таких книг вообще не финансируется государством).

Первым прототипом книги, которая «разговаривает», можно считать «озвученные» в начале XX века слова песни «У Мэри был барашек», опубликованные в 1830 году американской писательницей Сарой Хейл. Знаменитый Т. Эдисон впервые записал это произведение на восковой валик фонографа, создал и первый аудионоситель. Этот носитель информации полностью отвечал своему назначению. Он мог автономно ее хранить, воспроизводить в нужный момент независимо от человека, когда ее озвучивают, и давал возможность тиражировать эту информацию в большом количестве по приемлемой цене. Конечно, книги, которые «разговаривают», давно вышли за рамки только реабилитационных, познавательных и развлекательных средств. Получив широкое распространение, они стали доступны не только инвалидам по зрению. Сегодня доля рынка книг, которые «разговаривают», на западе составляет более 20% от общего количества привычных для нас бумажных книг, и эта цифра неуклонно растет. Создание и использование библиотечных информационных ресурсов стало повседневной нормой, и современный пользователь испытывает все большую потребность в новейших библиотечных услугах, мобильном и комфортном информационном обслуживании с использованием новейших компьютеризированных технических средств, в беспрепятственном доступе к любой релевантной информации.

В начале XXI века важным технологическим шагом для библиотек стал переход на принципиально новую цифровую информационную технологию записи на микрочипах и компакт-дисках, которые получили название – цифровая книга, которая «разговаривает» (DigitalTalkingBooks – дальше DTB). Именно этот информационный контент должен в первую очередь накапливаться в фондах библиотек, что становятся причастными к обслуживанию пользователей с особыми потребностями. Аналогичная задача возникает и перед РГБ.

На очереди – решение такого вопроса, как оборудование библиотек специальными техническими средствами, в частности проигрывателями DTB с большим количеством опций. Самыми простыми являются мобильные устройства с возможностью воспроизведения аудиофайлов, более сложные имеют речевые синтезаторы. Сегодня разрабатываются современные поколения проигрывателей, которые могут поддерживать визуальное и звуковое сопровождение, синтез речевых сигналов и вывод информации с помощью азбуки Брайля на соответствующий дисплей. Цифровые книги, которые «разговаривают» (DTB), охватывая как аудио-, так и текстовые данные, являются неоценимым мультимедийным информационным ресурсом для людей с нарушениями зрения. Спецификации DTB для лиц с полной или частичной потерей зрения, людей с ограниченной дееспособностью или другими физическими недостатками регламентируются международным стандартом ANSI/NISO (AmericanNationalStandardsInstitute/NationalInformationStandardsOrganization) и определяются как необходимость обеспечения пользователей специальным информационным контентом с помощью определенной системы поддержания его воспроизводства на базе различных сервисов и их организации. Этот стандарт определяет спецификации файлов DTB и их взаимосвязи и спецификации для DTB-устройств воспроизведения в двух сферах: работа проигрывателя в соответствии с требованиями файла и поведение проигрывателя в ситуациях, определенных требованиями пользователя.

Стандарт ANSI/NISO разработан для унифицирования имеющихся различных стандартов и спецификаций, в том числе созданных Консорциумом Всемирной паутины (World Wide Web Consortium, W3C) и Форумом свободных электронных книг (Open Book Forum).

Технологии цифровых книг, которые «разговаривают», создаются с целью обеспечения доступа людей с физическими ограничениями до информационного контента, подавая информацию в форматах, отличающихся от обычного печатного текста. Такие форматы обеспечивают представление информации с помощью голосового аудиозаписи, азбуки Брайля, речевого синтезатора, широкоформатной печати, специальных проигрывателей с возможностью обратного возвращения и перезагрузки. Таким образом, цифровая книга, которая «разговаривает» (DTB), – это набор электронных файлов соответствующей тематики для представления информации с помощью альтернативных средств ее отображения и воспроизведения, а именно синтезированных речевых текстов, рельефно-точечного алфавита Брайля, визуальных изображений и, в частности, широкоформатной печати. Файлы, сформированные и поданные в DTB согласно указанному стандарту, создают широкий спектр возможностей, среди которых быстрая, гибкая навигация; маркировка и использование закладок; поиск по ключевым словам; поиск слов по звуковым запросам; контроль пользователей отобранного по запросу контента (примечания, номера страниц). Такие способы представления информации решают немало проблем для читателей с проблемами зрения. Контент DTB подается с помощью аудио сигналов, комбинации аудио, текста и изображений или только с помощью текста.

Управление цифровым доступом обеспечивается с помощью DAISY (Digital Accessible Information System), то есть открытого международного стандарта для мультимедийного доступа, разработанного DAISY-Консорциумом вместе с работниками библиотек для лиц с физическими недостатками и изъянами зрения в сотрудничестве с передовыми учеными и пользователями со всего мира. Ядро технологии DAISY – это синхронизация текста, графики и аудиозаписей на базе рекомендаций W3C в соответствии с потребностями людей, которым нужен особый доступ к информации. Самая современная спецификация DAISY известна как ANSI/NISO Z39.86-2005 [9].

На сегодня в Консорциум входят 14 равноправных стран-участников, в которых реализуются DTB в формате DAISY, а также почти 60 стран-участников, которые не имеют права голоса.

Начальная версия DAISY создана в Шведской национальной библиотеке Брайля и базируется на цифровой системе представления аудиоинформации. Книга которая «разговаривает», фактически была электронной версией издания, но записанная на электронном носителе (кассете). Тогда обязательной ее составляющей была таблица, что вмещала содержание книги, и указатель, который предоставлял возможность выполнять навигацию на страницах книги, что «разговаривает». Основные характеристики – качество звука, продолжительность воспроизведения, удобство в использовании и совместимость с аналогичными книгами международных библиотек – соответствовали параметрам книг, созданных профессиональными издателями аудиопродукции.

Одним из ключевых моментов в процессе разработки и усовершенствования технологии DAISY было участие самих пользователей. Педро Зурита, Генеральный секретарь Всемирного союза слепых, в декабре 1995 года присоединился к подготовительному совещанию по созданию DAISY-Консорциума. Норма Тоукедо из Уругвая, Йаап из Нидерландов, Ева Бьорк и Кики Нордстрим из Швеции и другие исследователи активно способствовали развитию DAISY, представляя интересы Всемирного общества слепых [10]. Отдел библиотек для слепых IFLA в рамках конференции этой международной организации, что в 1996 году проходила в Токио, провел международный симпозиум, посвященный технологиям книг, которые «разговаривают», где были обсуждены проблемы внедрения цифровых технологий создания таких книг и предоставление услуг провайдеров соответствии с интересами пользователей. По резолюции симпозиума – ни одна из цифровых аудиотехнологий на том этапе не соответствовала требованиям непосредственных пользователей.

На основании соглашения DAISY-Консорциума, утвержденной в Цюрихе в апреле 1997 г., в Сигтун – недалеко от аэропорта Орланда в Стокгольме, было проведено международное техническое совещание, на котором всесторонне рассматривался стандарт DTB. Как результат, для DAISY-Консорциума было предложено проанализировать веб-технологии и стандарты, как HTML (Hyper Text Markup Language) и SMIL (Synchronized Multimedia Integration Language).

Впоследствии это способствовало созданию DAISY-Консорциумом уникального бесплатного программного инструмента, который поддерживает все языки в DAISY-формате и предоставляет пользователям возможность навигации в аудио- или видеоконтенте, представленном в структурированном виде. В сентябре 1998 г. технические спецификации книг, которые «разговаривают», утверждены как версия DAISY 2.0 и базируются на рекомендациях совещания в Сигтуне. На саммите Организации Объединенных Наций (World Summit on the Information Society, WSIS) с участием представителей DAISY-Консорциума, IFLA/LBS, IFLA/LSDP и общественных организаций, объединяющих людей с нарушениями зрения, рассмотрены вопросы развития информационного общества, в частности разработка технологий для людей с особыми потребностями. Секция WSIS по проблемам инвалидов успешно дополнила документы саммита концепцией универсальных разработок [11], одной из которых является технология DAISY, которая удовлетворяет потребности каждого, в том числе и людей с ограниченными возможностями. Используя технологию DAISY, библиотеки и библиотекари имеют возможность расширять и внедрять на практике достижения в области информационных технологий, которые дают слепым людям возможность «видеть» достаточные объемы информации, «идти в ногу» с развитием общества, достичь результатов, которые под силу здоровым людям. Синхронизированное мультимедиа – это ключевая информационная технология для библиотечного обслуживания пользователей с особыми потребностями в ближайшем будущем. Структура DAISY книги определяется ее жанром. Так, для популярных романов и других книг с простой линейной структурой, которые редко используются в образовательных целях и обычно читаются последовательно от начала до конца, вполне достаточным является наличие аудиозаписи с возможностью перехода от главы до главы. Цифровые аудиозаписи отдельных научных, учебных и научно-популярных произведений будут значительно удобнее для чтения, если пользователь будет иметь возможность «передвигаться» по структурным подразделениям (например, по частям, главам, разделам, подразделам) и при этом будет иметь возможность обойтись без просмотра полного текстового файла. Книга в формате DAISY может состоять из аудио, текста и изображений или их комбинации. Все книги этого формата используют общий набор файловых типов, хотя некоторые файлы не обязательны. Практически все типы файлов базируются на основе XML. Чтобы книга в

DAISY-формате была удобной для использования, она должна содержать такие типы файлов: пакетный, текстового контента, изображения, аудио-файл, файл синхронизации, файл навигационного контроля, ресурсный файл, файл презентации стиля, файл преобразований [12].

Нами были проанализированы основные классы программных средств, методы и информационные технологии, которые упрощают создание компьютерного контента для обучения людей с недостатками зрения; обеспечивают оптимизацию, классификацию по иерархии и способы хранения данных.

Библиографический список

1. Fujiyoshi M., Fujiyoshi A. (2006) A New Audio Testing System for the Newly Blind and the Learning Disabled to Take the National Center Test for University Admissions. //Miesenberger K., Klaus J., Zagler W.L., Karshmer A.I. (eds) Computers Helping People with Special Needs. ICCHP 2006. Lecture Notes in Computer Science, vol 4061. Springer, Berlin, Heidelberg
2. Brzozza P. (2008) Multimedia MathReader for Daisy Books. In: Miesenberger K., Klaus J., Zagler W., Karshmer A. (eds) Computers Helping People with Special Needs. ICCHP 2008. Lecture Notes in Computer Science, vol 5105. Springer, Berlin, Heidelberg
3. Kahlisch T. (2002) DZB – A German Library for the Blind Goes Digital. In: Miesenberger K., Klaus J., Zagler W. (eds) Computers Helping People with Special Needs. ICCHP 2002. Lecture Notes in Computer Science, vol 2398. Springer, Berlin, Heidelberg
4. Tazawa N., Totihara S., Iwahana Y., Imai A., Seiyama N., Takagi T. (2010) Rapid Listening of DAISY Digital Talking Books by Speech-Rate Conversion Technology for People with Visual Impairments. In: Miesenberger K., Klaus J., Zagler W., Karshmer A. (eds) Computers Helping People with Special Needs. ICCHP 2010. Lecture Notes in Computer Science, vol 6179. Springer, Berlin, Heidelberg
5. Kerscher G. (2006) The Essential Role of Libraries Serving Persons Who Are Blind and Print Disabled in the Information Age. In: Miesenberger K., Klaus J., Zagler W.L., Karshmer A.I. (eds) Computers Helping People with Special Needs. ICCHP 2006. Lecture Notes in Computer Science, vol 4061. Springer, Berlin, Heidelberg
6. Yamaguchi K., Suzuki M. (2012) Accessible Authoring Tool for DAISY Ranging from Mathematics to Others. In: Miesenberger K., Karshmer A., Penaz P., Zagler W. (eds) Computers Helping People with Special Needs. ICCHP 2012. Lecture Notes in Computer Science, vol 7382. Springer, Berlin, Heidelberg
7. Ballieu Christensen L. (2010) Exploiting the Daisy Format for Automated, Single-Source Braille Publishing. In: Miesenberger K., Klaus J., Zagler W., Karshmer A. (eds) Computers Helping People with Special Needs. ICCHP 2010. Lecture Notes in Computer Science, vol 6179. Springer, Berlin, Heidelberg
8. Gardner J.A. et al. (2010) DAISY Kindergarten Books and Rocket Science Journals. In: Miesenberger K., Klaus J., Zagler W., Karshmer A. (eds) Computers Helping People with Special Needs. ICCHP 2010. Lecture Notes in Computer Science, vol 6179. Springer, Berlin, Heidelberg
9. Minatani K. (2012) Development of a DAISY Player That Utilizes a Braille Display for Document Structure Presentation and Navigation. In: Miesenberger K., Karshmer A., Penaz P., Zagler W. (eds) Computers Helping People with Special Needs. ICCHP 2012. Lecture Notes in Computer Science, vol 7382. Springer, Berlin, Heidelberg
10. Kerscher G. (2002) Structured Access to Documents, Digital Talking Books, and Beyond: The DAISY Consortium. In: Miesenberger K., Klaus J., Zagler W. (eds) Computers Helping People with Special Needs. ICCHP 2002. Lecture Notes in Computer Science, vol 2398. Springer, Berlin, Heidelberg
11. Yamaguchi K., Suzuki M. (2010) On Necessity of a New Method to Read Out Math Contents Properly in DAISY. In: Miesenberger K., Klaus J., Zagler W., Karshmer A. (eds) Computers Helping People with Special Needs. ICCHP 2010. Lecture Notes in Computer Science, vol 6180. Springer, Berlin, Heidelberg
12. Brzozza P., Spinczyk D. (2006) Multimedia Browser for Internet Online Daisy Books. In: Miesenberger K., Klaus J., Zagler W.L., Karshmer A.I. (eds) Computers Helping People with Special Needs. ICCHP 2006. Lecture Notes in Computer Science, vol 4061. Springer, Berlin, Heidelberg

УДК 004.94

И.С. Панова

студентка кафедры компьютерных технологий и программной инженерии

В.Ю. Гамов

канд. воен. наук, доцент – научный руководитель

МОДЕЛЬ ПРОГНОЗИРОВАНИЯ КОЛИЧЕСТВА ПУСКОВ РАКЕТ-НОСИТЕЛЕЙ ЛЕГКОГО КЛАССА

В настоящее время, с постоянным развитием космической отрасли возникает необходимость проведения испытаний ракет – носителей различного назначения. В связи с этим процесс запусков космических аппаратов зависит не только от заранее запланированных Правительством и Министерством обороны пусков, но и от ряда случайных факторов. Поэтому необходимо проанализировать и получить оценки случайного процесса запусков космических аппаратов для прогнозирования оптимальных структур с целью обеспечения испытаний ракетно-космической техники. Предполагается, что данные о запусках ракет – носителей представлены в виде значений временного ряда. Временной ряд – это множество наблюдений, генерируемых последовательно во времени [1]. Таким образом, модель прогнозирования должна адекватно описывать исследуемый процесс запусков космических аппаратов и являться основой для получения его будущих значений.

В качестве модели прогнозирования выбрана модель ARIMA (Autoregressive Integrated Moving Average, АРПСС) – модель авторегрессии и проинтегрированного скользящего среднего, предложенная Боксом и Дженкинсом в 1976 году. ARIMA – процессы – это класс стохастических процессов, используемых для анализа временных рядов. Модель включает в себя три типа параметров: параметры авторегрессии (p), порядок разности (интегрирования) (d) и параметры скользящего среднего (q). В обозначениях Бокса и Дженкинса модель записывается как ARIMA (p, d, q). Данный класс моделей используется при моделировании нестационарных временных рядов. Общий вид модели для нестационарного временного ряда Y_t имеет вид [2,3]:

$$\Delta^d Y_t = c + \sum_{i=1}^p \alpha_i \Delta^d Y_{t-i} + \sum_{j=1}^q b_j \varepsilon_{t-j} + \xi,$$

где c – константа, α_i – параметры авторегрессии, b_j – параметры скользящего среднего, Δ^d – оператор разности временного ряда порядка d , ξ – случайное воздействие.

Схематичное описание модели можно представить в следующем виде на рис. 1.

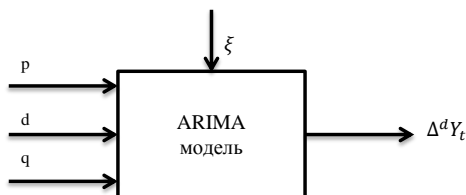


Рис. 1. Общая схема ARIMA – модели

Объектами данной схемы являются: p, d, q – входные переменные; $\Delta^d Y_t$ – выходная переменная; ξ – случайное воздействие.

Прогнозирование исследуемого временного ряда с помощью ARIMA – модели включает наличие следующих этапов:

1. Преобразование ряда к стационарному. Применение данной модели предполагает обязательную стационарность исследуемого ряда. Ряд является стационарным, если он имеет постоянные во времени математическое ожидание, дисперсию и автокорреляцию с удалёнными сезонной, циклической и трендовой составляющими. Для перехода к стационарному ряду традиционно применяют оператор взятия последовательных разностей, тем самым на данном этапе определяется значение параметра d (порядка разности).

2. Идентификация пробной модели. Основными инструментами идентификации ARIMA модели являются автокорреляционная (АКФ) и частная автокорреляционная (ЧАКФ) функции. Необходимо исследовать характер поведения этих функций и выдвинуть гипотезы о количестве параметров p (порядка авторегрессии) и параметров q (порядка скользящего среднего). По виду АКФ можно определить какое количество параметров q присутствует в модели, а из ЧАКФ следует определение параметров p .

3. Оценка параметров пробной модели. После осуществления идентификации модели необходимо оценить её параметры. Для этих целей используется метод максимального правдоподобия (ММП).

4. Исследование адекватности модели. Для проверки пробной модели на адекватность анализируется ряд остатков. У адекватной модели остатки, в основном, малые, распределены случайно и представляют собой «белый шум», т.е. их выборочные АКФ не должны отличаться от нуля. Для проверки гипотезы о том, что наблюдаемые данные являются реализацией «белого шума», используется Q -статистика Льюнга-Бокса.

5. Построение прогноза на основе выбранной модели. Полученные оценки параметров используются на последнем этапе для вычисления нового значения ряда и построения доверительного интервала для прогноза. Если заданная модель на предыдущем этапе оказалась неадекватна, то продолжается выбор новой улучшенной модели, но уже с другими параметрами.

В заключении необходимо отметить, что данный класс моделей позволяет получать точные прогнозы, опираясь только на информацию о прошлых наблюдениях. В большинстве научных публикаций и работ по прогнозированию среди различных статистических методов применяется именно класс моделей ARIMA, как наиболее надёжный и обоснованный алгоритм. К достоинствам ARIMA – моделей при прогнозировании можно отнести: получение точных прогнозов на краткосрочный период (напр., год), а так же описание широкого спектра характеристик временного ряда. Из недостатков можно отметить, что количество наблюдений для составления прогноза должно быть не менее пятидесяти и при добавлении новых данных в ряд наблюдений требуется перестройка, а иногда выбор абсолютно новой модели.

Библиографический список

1. Бокс Дж., Дженкинс Г. Анализ временных рядов. Прогноз и управление / пер. с англ. В.Ф. Писаренко. М.: Мир, 1974. 406 с.
2. Гамов В.Ю. Применение принципов прогнозирования к процессам обоснования ОШС ФВЗ КА/ Мирный.: 1 ГИК МО РФ, 2001.124 с.
3. Гамов В.Ю. Теоретические основы прогнозирования оптимальных организационно – штатных структур, обладающих гибкостью, формирований войск запуска космических аппаратов: учебник. типография в/ч 13991, Мирный.: 1 ГИК МО РФ, 2011. 296 с.

УДК 681.32

К. А. Парахневич

студент кафедры проблемно-ориентированных вычислительных комплексов

О. О. Жаринов

канд. техн. наук, доцент – научный руководитель

ПРИНЦИП ОТКРЫТОЙ АРХИТЕКТУРЫ В ОСНОВЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ МИКРОКОНТРОЛЛЕРОВ

Многие сталкиваются с необходимостью изучения программируемых микросхем, но не всегда удаётся самостоятельно получить нужную информацию или необходимые навыки быстро и в полной мере. Проблема заключается в том, что большая часть информации по микроконтроллерам и другим элементам электронной схемы зачастую не переведена на русский язык. Это касается не только литературы, но и программного обеспечения: интерфейс профессиональных программ по работе с микроконтроллерами зачастую сложен даже для опытных инженеров. Для решения перечисленных проблем было разработано программное обеспечение [1]. В настоящее время функциональность программы дополнительно расширена.

При разработке программы была поставлена задача создания гибкого инструмента, позволяющего изучать уже имеющиеся примеры по работе с микроконтроллерами и разрабатывать новые. В основу программы был положен принцип открытой архитектуры, по которому дорабатывать программу может любой желающий без изменения программного кода самого приложения. Как результат, для добавления новых программ или микроконтроллеров в базу данных достаточно создать новую папку в базе данных и заполнить её файлами по аналогии с уже имеющимися примерами.

Схема взаимодействия разрабатываемой программы с программным обеспечением сторонних разработчиков и аппаратными средствами представлена на рис. 1.

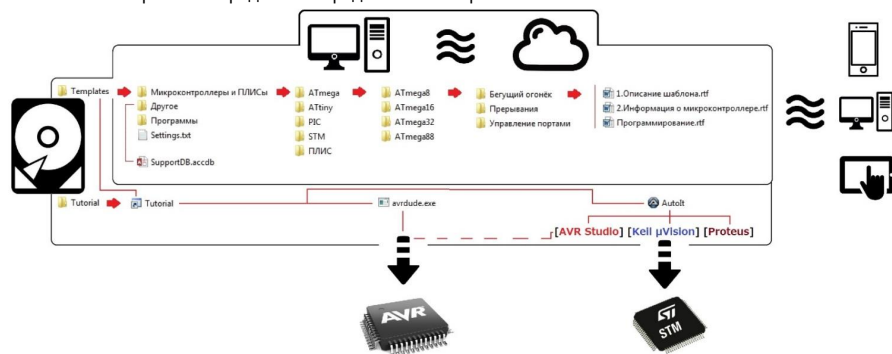


Рис. 1. Общая схема программы

Файловая база программы, которая хранится в облаке, содержит шаблоны – примеры по работе с элементами электронных схем, включающие текстовое описание, изображения, расположение выводов и программный код, который сразу же можно прошить в подключенное устройство. Есть возможность прошить собственный код. Текстовое содержимое примеров можно редактировать в MicrosoftWord. Разработанная программа, названная Tutorial, позволяет прошивать различные микроконтроллеры, заимствуя методы программы avrdude и других программ.

Структура файловой базы данных устроена следующим образом: папка Templates содержит три основные папки: Микроконтроллеры и ПЛИС, содержащие примеры по работе с различными устрой-

ствами, Программы, содержащие примеры по работе с программным обеспечением других разработчиков, и «Другое» – папку, используемую для внутренних нужд программы. Каждая папка состоит из подпапок. На основе структуры этих папок формируется содержимое полей со списками на форме выбора Шаблона, как показано на следующем рис. 2.

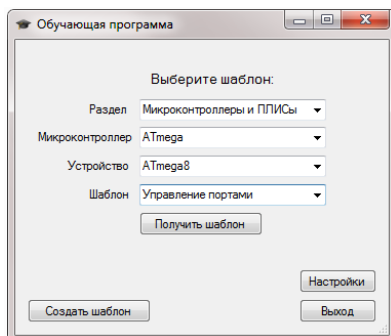


Рис. 2. Форма выбора шаблона

После выбора шаблона для изучения пользователь переходит на основную форму. Основная форма с описанием шаблона формирует свои вкладки на основе файлов Word, находящихся в конечной папке конкретного шаблона. Эта форма представлена на рис. 3.

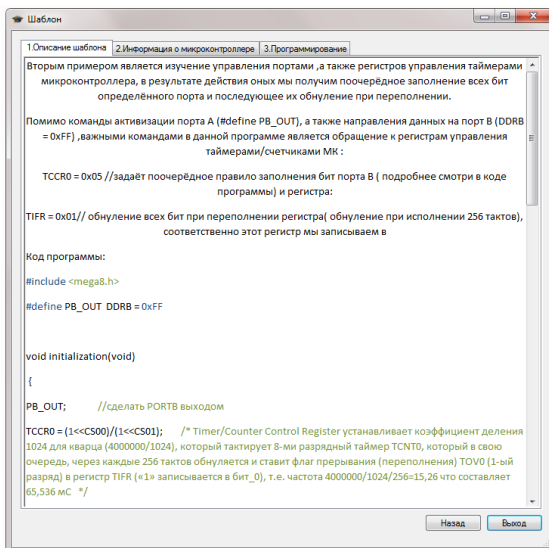


Рис. 3. Форма описания шаблона

Чем удобна такая система?

- Во-первых, шаблоны можно изучить не только через программу, но и через любое приложение, поддерживающее форматы Microsoft Word на любой операционной системе, будь то стационарный компьютер, ноутбук или мобильное устройство.

- Во-вторых, простота и интуитивность такой структуры позволяет пользователю добавлять новые виды микроконтроллеров и шаблоны к ним по аналогии с уже существующими примерами, без необходимости заполнения сложных программных интерфейсов.

- В-третьих, при добавлении новых программ и скриптов, их достаточно сохранить в соответствующую папку – программа сама распознает исполняющие файлы и создаст соответствующую кнопку для их запуска на вкладке программирования. В свою очередь скрипты могут быть написаны в любом приложении, удобном пользователю, для их распознавания программой достаточно лишь поменять текстовый файл в шаблоне по аналогии с файлами других шаблонов. Эти скрипты позволяют воспроизводить действия на экране в профессиональных программах, показывая пользователю порядок выполнения той или иной операции, выдавая пояснения.

- В-четвёртых, возможность доработки существующей структуры и создания новой: порядок и содержимое вкладок на форме описания шаблона зависит от нумерации и содержания файлов Microsoft Word в базе данных. Примеры могут создаваться как на основе уже имеющихся шаблонов, так и формироваться по собственному принципу.

Подводя итог, программа способна совместно работать с любыми другими программами, существует возможность добавлять шаблоны по работе с любыми элементами электронной схемы. Именно принцип открытой архитектуры, уникальный для программного обеспечения по программированию микроконтроллеров, лежащий в основе разработанной программы, позволяет сделать процесс обучения максимально гибким и полным для любого пользователя.

Библиографический список

1. *Парахневич К.А.* Разработка программной части обучающего комплекса для формирования навыков работы с программируемыми элементами электронных схем // Шестьдесят девятая международная студенческая научная конференция ГУАП. СПб.: ГУАП, 2016. С.86.

УДК51–74

Д. В. Печенин, М. Р. Гусманова

магистранты кафедры вычислительных систем и сетей

Л.А. Мироновский

д-р техн. наук, профессор – научный руководитель

ИССЛЕДОВАНИЕ АВТОМОРФНЫХ ПЕРЕДАТОЧНЫХ ФУНКЦИЙ

Передаточные функции широко применяются в теории автоматического управления. В данной работе рассматривается специальная разновидность таких функций – автоморфные передаточные функции (АПФ). Дается определение АПФ, решается задача поиска дробно-линейного преобразования (ДЛП) для заданной передаточной функции, а также обратная задача поиска передаточной функции по заданному ДЛП.

Рассмотрим дробно-рациональную передаточную функцию порядка n :

$$Q(p) = \frac{b_n p^n + b_{n-1} p^{n-1} + \dots + b_1 p + b_0}{a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0},$$

где b_i, a_i – постоянные коэффициенты; p – оператор Лапласа.

Будем называть передачную функцию $Q(p)$ автоморфной, если она после замены $p \rightarrow \varphi(p)$, где $\varphi(p)$ – дробно-линейное преобразование вида

$$\varphi(p) = \frac{Ap + B}{Cp + D}, \quad (1)$$

переходит сама в себя [1, 2].

Рассмотрим пример. Пусть имеется передачная функция второго порядка

$$Q_1(p) = \frac{1p^2 + 2p + 3}{4p^2 + 5p + 6}. \quad (2)$$

Выполним замену следующую замену переменных:

$$p \rightarrow \frac{6p + 3}{-3p - 6}. \quad (3)$$

В результате получаем передачную функцию

$$Q_2(p) = \frac{1p^2 + 2p + 3}{4p^2 + 5p + 6},$$

совпадающую с исходной. Следовательно, передачная функция (2) автоморфна.

Прямая задача. Перейдем к задаче поиска коэффициентов ДЛП для заданной передачной функции 2-го порядка. Запишем исходную передачную функцию в виде:

$$Q(p) = \frac{b_2 p^2 + b_1 p + b_0}{a_2 p^2 + a_1 p + a_0}, \quad (4)$$

где $b_2, b_1, b_0, a_2, a_1, a_0$ – известные коэффициенты.

Выполняя замену переменной (1) $p \rightarrow \varphi(p)$, получаем преобразованную передачную функцию:

$$Q_1(p) = \frac{p^2 (b_2 A^2 + b_1 AC + b_0 C^2) + p (2b_2 AB + b_1 AD + b_1 BC + 2b_0 CD) + (b_2 B^2 + b_1 BD + b_0 D^2)}{p^2 (a_2 A^2 + a_1 AC + a_0 C^2) + p (2a_2 AB + a_1 AD + a_1 BC + 2a_0 CD) + (a_2 B^2 + a_1 BD + a_0 D^2)}.$$

Приравняв коэффициенты при одинаковых степенях p в исходной и преобразованной передачных функциях, получаем систему:

$$\left\{ \begin{array}{l} b_2 = b_2A^2 + b_1AC + b_0C^2, \\ b_1 = 2b_2AB + b_1AD + b_1BC + 2b_0CD, \\ b_0 = b_2B^2 + b_1BD + b_0D^2, \\ a_2 = a_2A^2 + a_1AC + a_0C^2, \\ a_1 = 2a_2AB + a_1AD + a_1BC + 2a_0CD, \\ a_0 = a_2B^2 + a_1BD + a_0D^2. \end{array} \right. \quad (5)$$

Это переопределенная система из 6 уравнений с 4 неизвестными A, B, C, D. Одно из ее решений, полученное с помощью символического пакета MAPLE, имеет вид:

$$\left\{ \begin{array}{l} A = -b_2a_0 + a_2b_0, \\ B = -b_1a_0 + a_1b_0, \\ C = b_2a_1 - a_2b_1, \\ D = b_2a_0 - a_2b_0. \end{array} \right. \quad (6)$$

Тем самым установлена связь коэффициентов ДЛП с коэффициентами исходной передаточной функции (4)

Подставляя коэффициенты (6) в ДЛП (1), получаем решение поставленной задачи

$$\varphi(p) = \frac{(-a_0b_2 + a_2b_0)p - a_0b_1 + a_1b_0}{(a_1b_2 - a_2b_1)p + a_0b_2 - a_2b_0}.$$

Альтернативой описанному выше методу является «минорный» способ поиска коэффициентов ДЛП. Рассмотрим его.

Для начала представим передаточную функцию(4) в виде

$$Q(p) = \begin{bmatrix} b_2 & b_1 & b_0 \\ a_2 & a_1 & a_0 \end{bmatrix} \begin{bmatrix} p^2 \\ p \\ 1 \end{bmatrix} = \mathbf{S}\mathbf{P},$$

где \mathbf{S} – матрица коэффициентов передаточной функции, а \mathbf{P} – вектор-столбец степеней p [3]. Выпишем все миноры второго порядка матрицы \mathbf{S} :

$$\mathbf{M}_1 = \begin{vmatrix} b_2 & b_1 \\ a_2 & a_1 \end{vmatrix}, \mathbf{M}_2 = \begin{vmatrix} b_2 & b_0 \\ a_2 & a_0 \end{vmatrix}, \mathbf{M}_3 = \begin{vmatrix} b_1 & b_0 \\ a_1 & a_0 \end{vmatrix}.$$

Несложно доказать, что коэффициенты ДЛП могут быть следующим образом выражены через указанные миноры:

$$A = -D = \begin{vmatrix} b_2 & b_0 \\ a_2 & a_0 \end{vmatrix} = -b_2a_0 + a_2b_0,$$

$$B = -\begin{vmatrix} b_1 & b_0 \\ a_1 & a_0 \end{vmatrix} = -b_1a_0 + a_1b_0,$$

$$C = \begin{vmatrix} b_2 & b_1 \\ a_2 & a_1 \end{vmatrix} = b_2a_1 - a_2b_1.$$

В качестве примера найдём ДЛП для передаточной функции (2):

$$A = -D = -b_2a_0 + a_2b_0 = -1 \cdot 6 + 4 \cdot 3 = 6,$$

$$B = -b_1a_0 + a_1b_0 = -2 \cdot 6 + 5 \cdot 3 = 3,$$

$$C = b_2a_1 - a_2b_1 = 1 \cdot 5 - 4 \cdot 2 = -3.$$

Подставив ДЛП с заданными коэффициентами в передаточную функцию (2) получаем

$$Q_3(p) = \frac{1p^2 + 2p + 3}{4p^2 + 5p + 6},$$

Что совпадает с исходной функцией (2).

Обратная задача. Перейдем к задаче поиска коэффициентов $b_2, b_1, b_0, a_2, a_1, a_0$ передаточной функции (4) по заданным коэффициентам A, B, C , ДДЛП (1).

Для решения этой задачи запишем систему уравнений (5) в матричном виде

$$\begin{bmatrix} A^2 & AC & C^2 & 0 & 0 & 0 \\ 2AB & AD + BC & 2CD & 0 & 0 & 0 \\ B^2 & BD & D^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & A^2 & AC & C^2 \\ 0 & 0 & 0 & 2AB & AD + BC & 2CD \\ 0 & 0 & 0 & B^2 & BD & D^2 \end{bmatrix} \begin{bmatrix} b_2 \\ b_1 \\ b_0 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} b_2 \\ b_1 \\ b_0 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}. \quad (7)$$

Введём следующие обозначения:

$$\mathbf{R} = \begin{bmatrix} A^2 & AC & C^2 \\ 2AB & AD + BC & 2CD \\ B^2 & BD & D^2 \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix},$$

где \mathbf{R} – матрица, составленная из коэффициентов ДЛП, \mathbf{a} и \mathbf{b} – вектор-столбцы, составленные из коэффициентов знаменателя и числителя передаточной функции [3].

Тогда можно записать систему уравнений (7) более компактно:

$$\begin{bmatrix} \mathbf{R} & 0 \\ 0 & \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{a} \end{bmatrix}. \quad (8)$$

Систему уравнений (8) можно разбить на две независимые подсистемы:

$$\begin{aligned} \mathbf{R}\mathbf{a} &= k\mathbf{a}, \\ \mathbf{R}\mathbf{b} &= k\mathbf{b}, \end{aligned}$$

где k – некоторый коэффициент [4].

Мы получили две классические задачи о собственных векторах. Тем самым обратная задача оказалась сведена к стандартной алгебраической проблеме собственных значений.

В ходе исследования были получены следующие результаты:

- разработан алгоритм для поиска коэффициентов ДЛП по заданным коэффициентам передаточной функции;
- разработан алгоритм для поиска передаточной функции по коэффициентам ДРЛП.

В дальнейшем планируется исследовать связь между собственными числами и собственными векторами матриц \mathbf{R} и матрицы, составленной из коэффициентов ДЛП.

Библиографический список

1. Мировский Л.А. Инварианты математических моделей: текст лекций. СПб.: ЛИАП, 1991. 32 с.
2. Мировский Л.А. Инварианты математических моделей: учеб. пособие. СПб.: ГУАП, 1993. 116 с.
3. Беллман Р. Введение в теорию матриц / пер. с англ В. Я. Катковник. СПб.: Наука, 1969. 367 с.
4. Карчевский Е.М., Карчевский М.М. Лекции по линейной алгебре и аналитической геометрии: учеб. пособие. Казань: Казанский университет, 2012. 303 с.

УДК 004.622

Д. В. Печенин

магистрант кафедры вычислительных систем и сетей

Л.А. Мироновский

д-р техн. наук, профессор – научный руководитель

ИССЛЕДОВАНИЕ СТРИП – МЕТОДА ПРЕОБРАЗОВАНИЯ ИЗОБРАЖЕНИЙ

На сегодняшний день, в связи с ежегодным увеличением количества передаваемой человеком информации, всё острее становится вопрос повышения точности передачи данных по каналу связи. Одним из самых популярных видов информации являются изображения. Для повышения точности их передачи можно использовать стрип-метод преобразования изображений [1]. В данной работе рассматриваются такие аспекты этого метода, как поиск инвариантных изображений для заданных матриц преобразования и поиск матриц преобразования, для которых заданное изображение будет инвариантным.

Суть стрип-метода преобразования изображений заключается в «разрезании» заданного изображения на фрагменты и последующее получение линейных комбинаций этих фрагментов. Если в ходе передачи по каналу связи возникнет импульсная помеха, приводящая к потере одного или нескольких фрагментов, то после обратного преобразования мы получим исходное изображение, но худшего качества [2].

Дадим более строгое определение стрип-преобразования. Пусть X – матрица изображения, A и B – матрицы преобразования, тогда прямое преобразование имеет вид

$$AXB = Z,$$

где Z – изображение, поступающее в канал связи. На приёмной стороне получаем изображение Z' , подвергшееся воздействию некой помехи. Применяя к нему обратное стрип-преобразование, получаем восстановленное изображение [3]:

$$A^{-1}Z'B^{-1} = X'.$$

Структурная схема стрип-преобразования представлена на рис. 1.

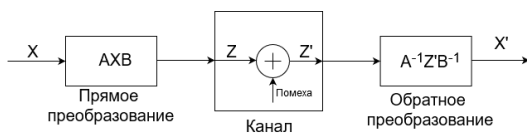


Рис 1. Стрип-метод преобразования изображений

Рассмотрим пример. На рис. 2,а представлено исходное изображение, разделённое на 6 равных частей, к которому мы применили стрип-преобразование и отправили по каналу связи, где произошла импульсная ошибка, уничтожившая 1/6 часть изображения (рис. 2,б). Получив «испорченное» изображение, применяем к нему обратное стрип-преобразование и видим (рис. 2,в), что в результате получили исходное изображение, но его качество заметно ниже.

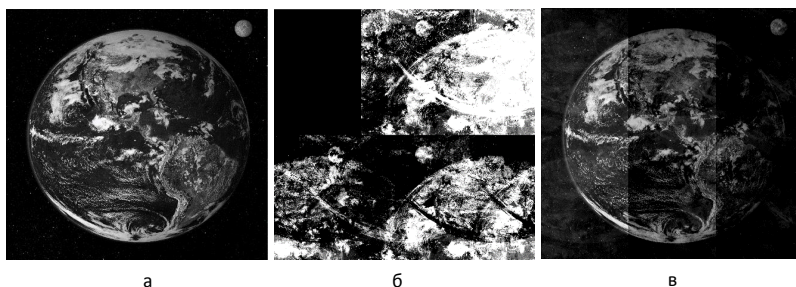


Рис. 2. Стрип-метод преобразования изображений

При этом возникает вопрос: «Существуют ли изображения, которые после применения к ним стрип-преобразования, не изменяются?». Для ответа на него выделим две подзадачи:

1. Поиск матриц X , инвариантных по отношению к преобразованию с помощью заданных матриц A и B ;

2. Поиск матриц A и B по заданной инвариантной матрице X .

Начнём с первой подзадачи. Пусть матрицы A , B и X имеют следующий вид:

$$A = \begin{bmatrix} -1 & 2 & 2 \\ 2 & -1 & 2 \\ 2 & 2 & -1 \end{bmatrix}, B = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, X = \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \end{bmatrix}.$$

Условие инвариантности матрицы X относительно стрип-преобразования, можно записать в форме

$$BXA - kX = 0, \tag{1}$$

где k – некий коэффициент.

Соотношение (1) представляет собой систему линейных алгебраических уравнений относительно переменных x_1, \dots, x_6 . Матрицу этой системы находим, вычисляя матрицу Якоби левой части соотношения (1):

$$J = \begin{bmatrix} -k-1 & 2 & 2 & -1 & 2 & 2 \\ -1 & 2 & 2 & 1-k & -2 & -2 \\ 2 & -k-1 & 2 & 2 & -1 & 2 \\ 2 & -1 & 2 & -2 & 1-k & -2 \\ 2 & 2 & 1 & 2 & 2 & -2 \\ 2 & 2 & -1 & -2 & -2 & 1-k \end{bmatrix}.$$

Чтобы полученная система имела нетривиальное решение, определитель матрицы Якоби должна быть равен нулю:

$$\det|J| = -k^6 + 54k^4 - 972k^2 + 5832 = 0.$$

Это уравнение имеет два вещественных корня 4.24 и -4.24. Подставим $k=4.24$ в систему (1) и найдём ранг матрицы Якоби:

$$\begin{cases} 2x_2 - x_1 + 2x_3 - x_4 + 2x_5 - 4.24x_1 = 0, \\ 2x_2 - x_1 + 2x_3 + x_4 - 2x_5 - 4.24x_4 = 0, \\ 2x_1 - x_2 + 2x_3 + 2x_4 - x_5 + 2x_6 - 4.24x_2 = 0, \\ 2x_1 - x_2 + 2x_3 + 2x_4 + x_5 - 2x_6 - 4.24x_5 = 0, \\ 2x_1 + 2x_2 - x_3 + 2x_4 + 2x_5 - x_6 - 4.24x_3 = 0, \\ 2x_1 + 2x_2 - x_3 - 2x_4 - 2x_5 + x_6 - 4.24x_6 = 0, \end{cases} \quad (2)$$

$$\text{rank}|J_1| = 3.$$

Так как ранг матрицы Якоби равен 3, можно сделать вывод, что количество линейно независимых переменных равно 3, т.е. мы три переменные можем взять произвольно. В частности, при $x_4 = 0$, $x_5 = 0$, $x_6 = 1$ система (2) примет следующий вид:

$$\begin{cases} 2x_2 - x_1 + 2x_3 - 4.24x_1 + 2 = 0, \\ 2x_2 - x_1 + 2x_3 - 2 = 0, \\ 2x_1 - x_2 + 2x_3 - 4.24x_2 + 2 = 0, \\ 2x_1 - x_2 + 2x_3 - 2 = 0, \\ 2x_1 + 2x_2 - x_3 - 4.24x_3 - 1 = 0, \\ 2x_1 + 2x_2 - x_3 + 1 = 0. \end{cases} \quad (3)$$

Решая систему уравнений (3), находим, что $x_1 = 0.9428$, $x_2 = 0.9428$, $x_3 = 0.5286$. Соответствующая матрица Химеет вид:

$$X = \begin{bmatrix} 0.9428 & 0.9428 & 0.5286 \\ & 0 & 0 \\ & & 1 \end{bmatrix}.$$

Убедимся, что данная матрица **X** действительно инвариантна относительно матриц **A** и **B** (рис. 3).

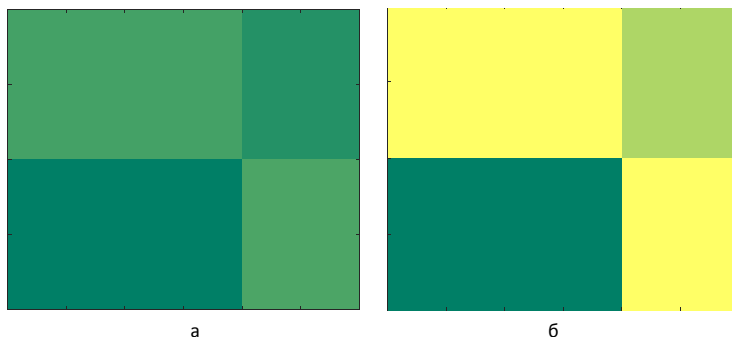


Рис. 3.

Мы видим, что рисунок до преобразования (рис. 3,а) и после преобразования (рис. 3,б) визуально очень похожи. Далее, взглянув на численные значения яркости «кусочков» изображения, мы можем убедиться в том, что рисунок действительно не изменился, а только стал в 4.24 раза ярче[4].

Перейдем к рассмотрению второй подзадачи – поиску матриц **A** и **B** по заданной инвариантной матрице **X**. При ее решении введём следующие ограничения:

1. **A = B**;
2. $k = 1$.

С учётом данных ограничений уравнение (1) примет вид:

$$\mathbf{AXA} = \mathbf{X}.$$

Здесь матрица **X** задана, матрица **A** подлежит определению, т.е. это квадратное матричное уравнение.

Известно, что его решение имеет вид [5]:

$$\mathbf{A} = \mathbf{X} \left((\mathbf{XX})^{\frac{1}{2}} \right)^{-1}.$$

Рассмотрим пример. Пусть матрица **X** имеет вид:

$$\mathbf{X} = \begin{bmatrix} 2 & 10 & 2 \\ 2 & 2 & 2 \\ 10 & 10 & 2 \end{bmatrix}.$$

Найдём матрицу преобразования **A**:

$$\mathbf{A} = \mathbf{X} \left((\mathbf{XX})^{\frac{1}{2}} \right)^{-1} = \begin{bmatrix} -0.33 & 1.14 & 0.38 \\ 0.38 & -0.33 & 0.22 \\ 1.14 & 1.94 & -0.33 \end{bmatrix}.$$

На рис. 4 представлены исходная матрица **X** и матрица **X** после прямого стрип-преобразования, и результат обратного стрип-преобразования.



Рис. 4.

Невооруженным глазом видно, что все три рисунка идентичны. Это подтверждает правильность нашего решения – заданная матрица **X** действительно инварианта относительно стрип-преобразования для найденной матрицы **A** [6,7].

В ходе данного исследования были получены следующие результаты:

1. Получена программная реализация стрип-метода преобразования изображений в пакете Matlab;
2. Получена программа для поиска матрицы **X**, инвариантной относительно заданных матриц преобразования **A** и **B**;
3. Получена программа для поиска матриц **A** и **B**, для которых заданная матрица **X** будет инвариантной.

В дальнейшем планируется продолжить исследования в данной области и более детально рассмотреть задачу поиска матриц **A** и **B**, для которых заданная матрица **X** будет инвариантной и найти решение, которое позволит избавиться от накладываемых нами ограничений [3,7].

Библиографический список

1. *Мироновский Л. А., Слаев В. А.* Двустороннее помехоустойчивое стрип-преобразование и его корневые изображения // Измерительная техника. 2012. №10. С.6-10.
2. *Mironovsky L. A., Slaev V. A.* Strip-method for image and signal transformation // Berlin: De Gruyter. 2011.
3. *Мироновский Л. А., Слаев В. А.* Стрип-метод преобразования изображений и сигналов. СПб.: Политехника. 2006.
4. *Востриков А. А., Сергеев М. Б., Литвинов М. Ю.* Маскирование цифровой визуальной информации: термин и основные определения // Информационно-управляющие системы. 2015. №5. С.116-123.
5. *Shurbet G. L., Lewis T. O.* Quadratic matrix equations // The Ohio journal of Science. 1974. №5.
6. *Востриков А. А.* О матрицах Адамара–Мерсенна и маскировании изображений // Информационные технологии. 2013. №11. С.37-39.
7. *Востриков А. А., Мишура О. В., Сергеев А. М., Чернышев С.А.* О выборе матриц для процедур маскирования и демаскирования изображений // Фундаментальные исследования. 2015. №2-24. С.5335-5339.

УДК 004.415.25:004.051

А. И. Позняков

магистрант кафедры вычислительных систем и сетей

А. В. Гордеев

д-р техн. наук, профессор – научный руководитель

АДАПТИВНАЯ СИСТЕМА РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В МОБИЛЬНОМ ПРИЛОЖЕНИИ

При разработке мобильного приложения, позволяющего работать одновременно с несколькими интернет-сервисами, возникла необходимость в реализации системы, позволяющей распределять ресурсы, выделяемые приложению.

Приложение позволяет проверять наличие уведомлений во всех интегрированных интернет-сервисах и оповещать пользователя. Однако при большом количестве одновременно подключенных интернет-сервисов приложение будет потреблять много ресурсов устройства, что может привести к быстрому расходу заряда батареи, интернет трафика или вызывать затруднения при работе с устройством.

В мобильных операционных системах имеются стандартные механизмы, позволяющие ограничивать ресурсы, потребляемые приложениями. Например, режимы «Doze» и «AppStandby» в операционной системе Android [1]. Такие механизмы позволяют полностью блокировать доступ определенного приложения к определенным ресурсам, а не часть приложения, тем самым ограничивая ресурсы, выделяемые всем интегрированным интернет-сервисам.

В большинстве случаев одни интернет-сервисы в приложении будут использоваться чаще других, поэтому было бы разумнее ограничивать ресурсы лишь для тех интернет-сервисов, которые реже используются. Например, если пользователь обращается к интернет-сервису «А» по нескольку раз в час, а к интернет-сервису «Б» лишь один раз в сутки, то для сервиса «Б» следует проверять наличие уведомлений с более низкой интенсивностью.

Исходя из этого, можно составить основные требования к разрабатываемой системе распределения ресурсов, выделяемых интегрированным интернет-сервисам для проведения процедуры проверки наличия уведомлений. Для определения предпочтений пользователя система должна уметь проводить мониторинг локальных действий пользователя при его работе с приложением [2]. Далее на основе мониторинга вычислять параметры работы системы уведомлений. Также система должна учитывать особенности устройства и операционной системы, так как каждое устройство и операционная система могут потреблять разное количество ресурсов при проведении процедуры проверки уведомлений. Система должна вести статистику потребляемых каждым интернет-сервисом ресурсов при проведении процедуры проверки для повышения точности вычислений параметров.

Внесение изменений в параметры должно происходить автоматически при определенных условиях, либо через установленные промежуточные времена.

В соответствии с этими требованиями составлена модель, структурно-функциональная схема которой изображена на рис. 1.

Согласно модели в мобильном приложении есть система уведомлений, оповещающая пользователя о появлении в интегрированных интернет-сервисах новых сообщений, отзывов, обновлений и прочих уведомлений. За процедуру проверки наличия уведомлений отвечает служба, которая работает в фоновом режиме.

Служба состоит из базы параметров, исполнительного модуля и системы адаптации. Исполнительный модуль содержит методы для проверки уведомлений для каждого подключенного интернет-сервиса и использует их при получении управляющих команд от системы адаптации в соответствии со значениями в базе параметров. База параметров содержит правила проверки уведомлений для каждого подключенного интернет-сервиса. Например, приоритет сервиса и максимальная задержка между проверками уведомлений.

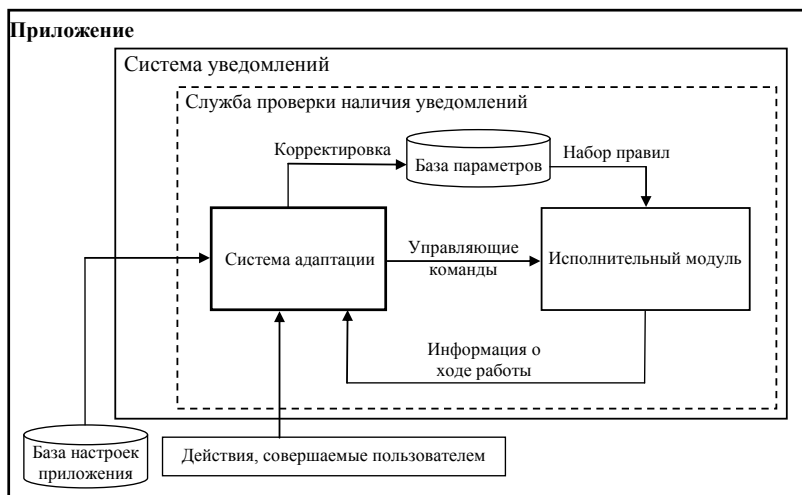


Рис. 1. Структурно-функциональная схема модели системы

Значения в базе параметров зависят от действий, совершаемых пользователем, настроек приложения и информации о ходе работы исполнительного модуля.

События и действия, приводящие к изменению параметров, бывают активными, то есть требующими немедленной обработки и вычисления новых параметров. К таким относятся действия пользователя, такие как: подключение сервиса, удаление сервиса, изменение настроек приложения и другие. Со стороны исполнительного модуля к активным относятся события, возникающие при сбое одного из интернет-сервисов. Также системой обрабатываются события, не оказывающие высокого влияния на значения параметров работы службы и не требующие немедленного вычисления новых параметров. Такие события называются пассивными. Система производит статистику по таким событиям и, через определенные промежутки времени, принимает решение о вычислении новых параметров. К таким данным могут относиться, например, время использования интернет-сервиса, количество использованных интернет-сервисами ресурсов при проведении процедуры проверки, частота появления уведомлений и другие.

Значения новых параметров вычисляются системой адаптации, структурно-функциональная схема которой изображена на рис. 2. Система адаптации состоит из модуля анализа, модуля статистики, модуля регулирования и базы статистики. Модуль статистики отвечает за обработку пассивных действий пользователя и исполнительного модуля. При поступлении информации о совершенных таких действиях модуль статистики считывает имеющиеся отношения к ним значения из базы статистики, вычисляет новые значения и записывает обратно в базу статистики.

Значения в базе статистики используются при вычислении параметров работы исполнительного модуля. Для каждого интернет-сервиса записываются два экземпляра наборов значений. В первом хранятся параметры статистики, по которым были вычислены актуальные параметры работы исполнительного модуля. Во втором находятся значения, обновляемые модулем статистики при совершении пользователем или исполнительным модулем пассивных действий.

Значения в базе статистики отражают информацию о том, как используется каждый интернет-сервис. Например, количество запусков интернет-сервиса и количество полученных уведомлений. Также в базе хранятся значения потребляемых каждым интернет-сервисом ресурсов при проверке наличия уведомлений. Например, среднее время отработки и средний потребляемый трафик.



Рис. 2. Структурно-функциональная схема системы адаптации

Запуск процедуры адаптации производит модуль анализа. В модуле присутствует таймер, после срабатывания которого, происходит сравнение значений базы статистики и, при определенном отклонении, производится запуск алгоритма адаптации.

При поступлении на модуль анализа информации о активных действиях пользователя или исполнительного модуля алгоритм адаптации запускается мгновенно.

Изменение параметров работы исполнительного модуля выполняет модуль регулирования. Он вычисляет новые параметры работы службы и отправляет команды исполнительному модулю.

Алгоритм адаптации исполнительного модуля к новым параметрам следующий:

- 1) производится остановка таймера в модуле анализа;
- 2) модуль регулирования отправляет команду остановки исполняющему модулю;
- 3) производится замена значений активных параметров в базе статистики новыми, вычисленными модулем статистики во время работы службы;
- 4) модуль регулирования производит вычисление новых параметров работы службы, используя настройки приложения и значения активных параметров в базе статистики;
- 5) замена значений в базе параметров вычисленными модулем регулирования;
- 6) модуль регулирования отправляет команду запуска исполняющему модулю;
- 7) производится запуск таймера в модуле анализа.

После этого служба проверки наличия уведомлений продолжает работу по новым параметрам.

В заключение нужно отметить, что основным преимуществом предложенной модели является повышение точности вычислений параметров со временем работы приложения. Количество ресурсов, потребляемых каждым интернет-сервисом при проверке наличия уведомлений может зависеть не только от действий пользователя и особенностей устройства, но и изменений, вносимых разработчиками интернет-сервисов. Система, реализованная по предложенной модели, сможет автоматически подстроиться к новым условиям без необходимости внесения изменений в приложение.

Библиографический список

1. Google Inc.:Optimizing for Doze and App Standby, URL: <https://developer.android.com/training/monitoring-device-state/doze-standby.html>
2. Wikipedia: Система контроля действий пользователя, URL: https://ru.wikipedia.org/wiki/Система_контроля_действий_пользователя

УДК 004.41

А.А. Розов

магистрант кафедры компьютерных технологий и программной инженерии

В. В. Бураков

проф., д-р техн. наук, доцент – научный руководитель

ИМИТАЦИЯ АВИАЦИОННОГО ОБОРУДОВАНИЯ

При Разработке радиолокационного оборудования остро стоит задача проверки его работоспособности. В целях проведения испытательных работ и обеспечения всесторонней работоспособности изделия создаётся программный комплекс направленный на имитацию оборудования установленного на самолёте.

Сам программный комплекс разделён на две части:

- Утилита, отвечающая за работу каналов связи и передач информации по ним.
- Разрабатываемое программное обеспечения имитирующее бортовое оборудование

Обмен по каналам каналом связи осуществляется по средствам утилиты фирмы «Элкус». Так же с утилитой поставляется база команд и библиотека для работы с драйвером. Утилита получает на вход команды в сокращённом виде ищет их расшифровку с базе и передаёт на соответствующий канал связи. Если основной канал отказывает совершается вторая попытка передачи по резервной линии. Если и она отказывает, то передаётся признак ошибки.

Информация передаётся словами, в виде цепочек сообщений. В начале каждой цепочки идёт команда передать векторное слово. В каждой цепочке есть признак отвечающий за необходимость передачи ответного сообщений.

Полные версии слов хранятся в утилите «Элмус», мы же оперируем только обращения к базе команд и признаками о выполнении дополнительных действий. Так же можем устанавливать последовательность следования слов в цепочке. При необходимости изменения команды можно задать переход на другую команды в базе команд и поведения локатора изменится.

Ответные сообщения преобразуются непосредственно в реализуемом программном комплексе. Основной информацией, получаемой из ответных сообщений являются координаты по которым стариться графическое изображение, и признак типа объекта.

С точки зрения пользователя программа осуществляет имитацию управления работы органами управления из состава ОПРНК.

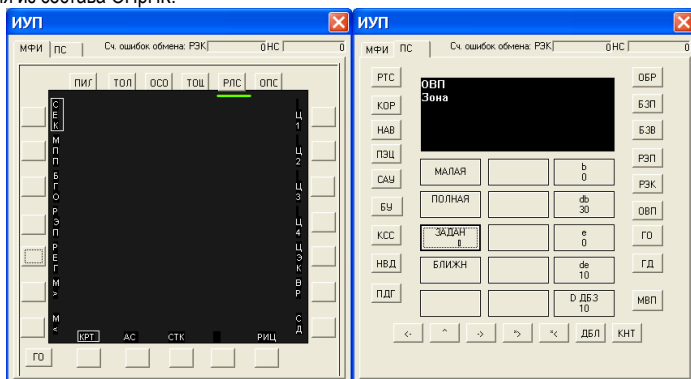


Рис. Вид экрана МФИ и ПС

Имитатор позволяет тестировать разрабатываемое СПО и проводить отладочно-диагностические операции в той же мере, в которой это может предоставить реальная аппаратура при условии прогона программ СПО в реальных условиях.

Имитируемое ИУП содержит две страницы, на которых отображаются многофункциональный индикатор (МФИ) и пульт системы (ПС). Для переключения страниц установить курсор манипулятора типа «мышь» на вкладке с надписью МФИ или ПС и щёлкнуть левой клавишей «мыши».

Для нажатия кнопки на имитируемых органах управления надо установить курсор манипулятора типа «мышь» на изображение кнопки и щёлкнуть левой клавишей «мыши». Включение кнопки производится в момент нажатия и отображается пунктирной линией по контуру изображения кнопки. Наличие рамки белого цвета вокруг наименования кнопок означает, что они включены.

На рисунке продемонстрирован вид экранов имитируемого оборудования.

Библиографический список

1. Руководство программиста по работе с «Элкус» для Microsoft Windows xp/7 ЗАО «электронная компания Элкус».
2. ГОСТ 26765.52-87
3. Бьернер. Страуструп язык программирования С++ М.: Издательство БИНОМ, 2012.

УДК 629.735.33

А. В. Савин

студент кафедры «Компьютерных технологий и программной инженерии»,

П. А. Степанов

ст. преподаватель – научный руководитель

МОБИЛЬНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА СЕКЦИИ ЕДИНОБОРСТВ

В мире есть люди, активно занимающиеся в школах восточных единоборств. В процессе занятий в таких школах ученику не всегда удается проводить какие-либо записи или заметки, чтобы можно было запомнить какой-либо прием и воспроизвести его в другой раз. Также ученикам не всегда выпадает возможность ходить на занятия регулярно, чтобы не утратить знания, мысли и навыки, получаемые в зале. Хотя мастера пишут книги, дают интервью в различных организациях, на телеканалах, интернет сайтах, найти нужную информацию по конкретному приему или движению бывает весьма затруднительно.

Существует также категория людей, которая не знает ничего о конкретной школе единоборств и не представляет себе, что это такое. Бывает, что найдется человек, который хочет заняться каким-либо стилем восточных единоборств, но не может, так как территориально школа находится далеко от его места проживания.

Есть и такая категория людей, которым занятия в школе восточных единоборств будет не только интересно, но и полезно. Так, в последнее время все больше детей с врожденным ДЦП начинают заниматься в китайской оздоровительной школе УШУ, чтобы улучшить свою координацию движения и укрепить свой организм [1].

Для решения этих проблем было решено написать в качестве бакалаврской выпускной квалификационной работы мобильное приложение для операционной системы Android, которое позволит любому пользователю прочитать и изучить прием или движение, изучаемые в школе Кунг Фу Вин Чун.

Некоторые приложения под ОС Android для Вин Чун уже написаны. В приложении «Wing Chun Technique» демонстрируются видео уроки мастеров. Приложение «Wing Chun – Zhen kungfu» демонстрирует трехмерную нарисованную модель, которая демонстрирует приемы и движения в зависимости от нажатой кнопки на экране. Школа Вин Чун, по своей природе, основана на четких проработанных с точки зрения математики и физики движениях, и каждому ученику следует их соблюдать [2]. Поэтому, в разрабатываемом приложении предполагается использовать информацию, основанную на этих понятиях, а соблюдать их или нет уже становится делом пользователя.

Операционная система Android на сегодняшний день является одной из самых распространенных среди пользователей. На момент 2013-2015 года, 53 % пользователей смартфонов имели ОС Android, а 51% всех скаченных приложений использовались также в ОС Android [3]. Приложения для этой операционной системы можно бесплатно скачать через Play Market – системное приложение, встроенное в каждую ОС Android и позволяющее находить и скачивать приложения по всему интернету.

Главными возможностями в использовании разрабатываемого приложения являются: выборка по тегу и/или по поисковой строки необходимой секции, отображение на экране текстовой информации и изображений при соответствующем нажатии кнопки, загрузка и кеширование изображений с сервера по запросу клиента, а также авторизация и возможность добавлять страницы в избранное, чтобы потом можно было найти данные с использованием этого тега.

Android разработка включает в себя написание Layout – внешнего вида приложения, с которым будет взаимодействовать пользователь, а также Activity – своего рода программной реализации Layout, где пользователь может ввести свой исходный код и обработчики событий. Приложение можно настроить так, что при возникновении какого-либо события, внешний вид менялся с одного Layout на другой [4].

В разрабатываемом приложении главный Layout, то есть тот, в котором пользователь будет вводить поисковые значения, имеет элементы SearchView, ListView и Spinner.

SearchView представляет собой поисковик, в который можно ввести текст. Этот элемент будет осуществлять поиск информационной секции по заголовкам.

Информационная секция включает в себя конкретное название приема или движения, подробную информацию о нем, а также имеет набор тегов и изображений, которые к нему относятся.

Spinner является выпадающим списком. В разрабатываемом приложении он играет роль списка тегов.

Теги позволяют найти прием или движение в зависимости от конкретной области, которая необходима пользователю. Так, включив тег «Понятия», пользователю будут выданы «Понятие центральной линии» и «Понятие внешних и внутренних ворот», а секции, не относящиеся к этому тегу, будут скрыты.

ListView выводит список информационных секций. При нажатии на одну из них, откроется новый экран, где будет представлена текстовая информация описания секции. В верхней части экрана находятся несколько кнопок, нажимая на которые пользователь может возвращаться назад, добавлять страницу в избранное или смотреть изображения. В последнем случае так же откроется новое окно и в нем будут представлены картинки, взятые из кеша приложения, либо загружены с сервера, если в кеше их не оказалось.

Также при первом запуске приложения, пользователю предложат авторизоваться или зарегистрироваться. Без регистрации, пользователь не сможет эксплуатировать приложение.

Загрузка с сервера осуществляется через стороннюю Java-библиотеку Picasso, которая позволяет быстро и без проблем не только загрузить изображение с URL, но и сразу добавить его в кеш или взять оттуда [5]. Тем не менее, Java Android имеет встроенную библиотеку для получения данных через HTTP – запрос – HttpURLConnection.

HttpURLConnection используется для авторизации или регистрации клиента. В качестве ответа, Apache-сервер обрабатывает POST-запрос клиента, и высылает ему ответ в виде строки «истина» или «ложь». На стороне сервера данные о пользователе берутся из MySQL Database.

Информационные секции и принадлежащие к ним теги хранятся в XML-файле.

XML-файл состоит из одного главного узла и двух вложенных. В первом из них описаны теги, их разновидности и названия. Во втором узле описаны секции. В каждой из них так же располагаются узлы с принадлежащими к секции тегами, узлы с текстовой информацией, а также узлы с названиями принадлежащих к секции изображений.

Пользовательские данные располагаются в другом XML-файле. Он состоит из одного корневого узла и некоторого количества вложенных узлов. В качестве атрибута вложенных узлов указан логин пользователя, а в содержимом описаны страницы, добавленные в избранное.

Все эти файлы располагаются во внутренней директории приложения, куда имеет право обращаться только приложение или пользователь с root-правами, но тенденции развития Android-смартфонов сводятся к серьезному ограничению в плане получения root-прав.

Вывод: разрабатываемое приложение будет выгодно отличаться от аналогов, так как оно позволит пользователю быстро найти информацию секцию по тегу и/или с использованием поисковой строки, прочитать и изучить необходимую для него информацию, вывести на экран прикрепленные к информационной секции изображения, добавить информацию секцию в избранное, авторизоваться в системе.

Библиографический список

1. VIVILAND: Полезны ли занятия УШУ для детей, <http://vikiland.ru/articles/useful-if-the-classes-in-wushu-for-children/>
2. Дудукчан И., Федоренко А. ВИН ЧУН КУНГ ФУ, Энциклопедия. Книга 1, базовые формы: Красноутинск, Бумажная типография, 1995. С. 86.
3. Planetiphon: статистика: ios vs Android, , <https://www.planetiphone.ru/news/028/statistika-ios-protiv-android.html>
4. Android developers: Макеты | Android developers , <https://developer.android.com/guide/topics/ui/declaring-layout.html?hl=ru>
5. Picasso: A powerful image downloading and caching library for Android, <http://square.github.io/picasso/>

УДК 62-52

Семенов А.А.

студент

Игнатьев М.Б.

д-р техн. наук, профессор – научный руководитель

СИСТЕМА АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ АВТОМОБИЛЕМ-НАРУШИТЕЛЕМ

В настоящее время идет активная разработка автомобилей с автоматическим управлением, также уже сейчас активно работают и развиваются автоматические системы контроля правил дорожного движения (ПДД), такие как камеры фото и видео фиксации. При объединении этих разработок нарушения ПДД будет возможно не только фиксировать, но и предотвращать автоматически.

Современный автомобиль сейчас оснащается такими блоками и устройствами, как системы активной безопасности:

- антиблокировочная система тормозов;
- антипробуксовочная система;
- система курсовой устойчивости;
- система распределения тормозных усилий;
- система экстренного торможения;
- система обнаружения пешеходов;
- электронная блокировка дифференциала.

К вспомогательным системам активной безопасности относятся:

- парковочная система,
- система кругового обзора,
- адаптивный круиз-контроль,
- система аварийного рулевого управления,
- система помощи движению по полосе,
- система помощи при перестроении,
- система ночного видения,
- система распознавания дорожных знаков,
- система контроля усталости водителя,
- система помощи при спуске,
- система помощи при подъёме.

Объединение вышеперечисленных систем в скором времени приведет к полностью беспилотному серийному автомобилю.

Но нельзя не отметить, что все равно главным остается человек, а человек – это не программа, и порой способен нарушать правила.

На данный момент одним из эффективных способов борьбы с нарушением ПДД являются камеры фото и видеofиксации, которые способны на данный момент фиксировать такие нарушения как:

- превышение скорости;
- движение по встречной полосе;
- выезд на полосу для общественного транспорта;
- движение по обочине;
- движение по трамвайным путям (если это запрещено);
- проезд грузового транспорта в зоны, где его движение ограничено или запрещено;
- проезд на запрещающий сигнал светофора (в том числе на железнодорожных переездах);
- выезд на полосу реверсивного движения при запрещающем сигнале специального светофо-

ра;

– выезд на перекресток в случае образования затора за ним;
– пересечение стоп-линии при запрещающем сигнале светофора;
– нарушение требований разметки (в том числе поворот из не предназначенного для этого ряда);

- нарушение требований запрещающих и предписывающих знаков;
- нарушения правил парковки;
- контроль платной парковки;
- неиспользование ремней безопасности;
- разговор по мобильному телефону без устройства hands-free;
- нарушения правил использования световых приборов.

Но они только фиксируют, и наказание (чаще всего в виде штрафа) произойдет не сразу, а через определенное время. А ведь нарушая, водитель может попасть в ДТП, что может привести к человеческим жертвам. Так почему бы не сделать так, чтобы нарушение предотвращалось?

Пример. Автомобиль едет с превышением скорости, опасно перестраивается, или проезжает на запрещающий сигнал светофора. Камера фиксирует нарушение и посылает сигнал тревоги диспетчеру а также сигнал о переходе на автоматическое управление, установленное в автомобиле нарушителе. Автомобиль-нарушитель, получив сигнал, должен перейти на автоматический контроль, то есть в зависимости от ситуации сбросить скорость, определив положение свое и автомобилей на трассе, при помощи системы автоматической парковки припарковать себя на обочине, заглушить мотор, включить аварийную сигнализацию и при необходимости заблокировать водителя в автомобиле и вызвать полицию.

Для реализации такого проекта необходимо разработать:

- модуль управления системами автомобиля, который позволит в автоматическом режиме перехватить управление и без участия водителя предотвратить нарушение ПДД или даже ДТП. Модуль, который даст указания машине сбросить скорость, выйти из потока движения, припарковаться также возможно глушение двигателя, запираение машины и вызов полиции;
- модуль распознавания нарушения непосредственно для камеры фото- и видеofиксации, который передаст сигнал о нарушении, а также сигнал на автомобиль о перехвате управления.

Также необходимо определиться с типом сигнала:

- радиосигнал,
- wi-fi-сигнал,
- передача сигнала по интернету,
- GSM-сигнал.

У каждого типа есть свои преимущества и недостатки.

Также возникают вопросы об идентификации автомобиля – система должна посылать краткий запрос, и, получив ответ от автомобиля, передавать сигнал о перехвате.

Для реализации проекта по автоматическому управлению автомобилем нарушителя необходимо еще решить ряд программных и аппаратных задач по разработке модулей обнаружения, фиксации и управления нарушений ПДД, но актуальность данной разработки не вызывает сомнений.

Библиографический список

1. Гудков В.А., Комаров Ю.Я., Рябчинский А.И., Федотов В.Н. Безопасность транспортных средств (автомобили): учеб. пособие для вузов. 2010. 431 с.
2. Ломакин В. В., Покровский Ю. Ю., Степанов И. С., Гоманчук О. Г. Безопасность автотранспортных средств: учебник для вузов. / под общ. ред. В.В. Ломакина. – М: МГУТ «МАМИ», 2011. – 299 с.
3. сайт systemsauto.ru
4. сайт habrahabr.ru

УДК 004.353.254

А. И. Соловьев, Е. А. Токмакова

студенты кафедры вычислительных систем и сетей

А. В. Аксенов

старший преподаватель – научный руководитель

ЦИФРОВОЙ ЧЕТЫРЁХСЕГМЕНТНЫЙ ИНДИКАТОР ПОКАЗАНИЙ ДАТЧИКОВ «ВАРВАРА»

Электроника в настоящее время используется во многих сферах жизни. Она универсальна и часто имеет интуитивный интерфейс, облегчающий взаимодействие пользователя с ней. Современные дисплеи используют чрезвычайно малые по размерам пиксели и обладают реалистичной цветопередачей. Однако для обычной повседневной техники по-прежнему продолжают использоваться ЖК-индикаторы и сегментные светодиодные панели. Разрядные дисплеи используются как в дешёвой электронике (калькуляторы, весы, часы), так и в более дорогой бытовой технике (стиральные машины, микроволновые печи, холодильники).

В данной статье предлагается новый способ отображения численной информации – четырёх-сегментный цифровой индикатор (рис. 1).

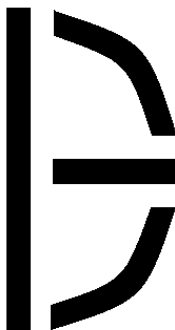


Рис. 1. Четырёхсегментный индикатор «ВАРВАРА»

Для того чтобы нагляднее продемонстрировать данный индикатор, необходимо сравнить его с уже существующими разрядными индикаторами. Наиболее популярным является семисегментный индикатор, который способен отображать арабские цифры и некоторые буквы:

- В латинице: заглавные *A, B, C, E, F, G, H, I, J, L, N, O, P, S, U, Y, Z*, строчные *a, b, c, d, e, g, h, i, n, ñ, o, q, r, t, u*.

- В кириллице: *А, Б, В, Г, г, д, Е, е, и, й, Н, О, о, П, п, Р, С, с, У, Ч, Ы* (два разряда), *Ь, Э/З*. [1]

Существует также девятисегментный индикатор, который можно рассматривать как вариацию семисегментного. Он имеет, кроме семи сегментов, как у семисегментного, еще дополнительные вертикальные или диагональные сегменты для иного, более удобного, отображения арабских цифр. [2] Девятисегментные индикаторы, несмотря на своё относительное удобство, не прижились в электронике, уступив место более простому семисегментному индикатору. Стоит упомянуть и о других многосегментных индикаторах (четырёх-, шестнадцатисегментных и др.), использующихся для отображения арабских цифр, а также всех букв латиницы и кириллицы.

Преимущество же индикатора, предлагаемого в данной статье, состоит в его максимальной простоте, которая достигается за счёт уменьшения числа сегментов в два раза, по сравнению с семи-сегментным индикатором, при сохранении возможности отображения всех десяти арабских цифр. Для отображения информации новый индикатор требует в два раза меньше подключений, чем семисегментный, который, по сути, является восьмисегментным, имея помимо семи сегментов ещё и десятичную точку.

Существует также бинарный индикатор, способный отображать все десятичные цифры с помощью четырёх сегментов-точек. Однако, основанный на вычислениях, он выполняет лишь декоративную функцию. Поэтому в разработке индикатора «ВАРВАРА» был сделан упор на визуальную схожесть отображаемых аналогов арабских цифр с реальными арабскими цифрами. Четырёхсегментный индикатор способен принимать только шестнадцать различных состояний, поэтому необходимо было отыскать такое сочетание сегментов, чтобы оно могло образовывать десять состояний, визуально схожих с десятью арабскими цифрами. Было рассмотрено несколько вариантов формы индикатора, в результате анализа которых была выделена та, что наиболее точно и понятно отображает большинство арабских цифр.

Многоразрядные индикаторы в микроэлектронике часто подключаются по динамическому принципу: выводы одноимённых сегментов всех разрядов соединены вместе. Чтобы выводить информацию на такой индикатор, управляющая микросхема должна циклически подавать ток на общие выводы всех разрядов, в то время как на выводы сегментов ток подаётся в зависимости от того, зажжён ли данный сегмент в данном разряде. Таким образом, чтобы получить десятиразрядный экран микрокалькулятора с использованием семисегментных индикаторов, нужны всего восемнадцать выводов (8 анодов и 10 катодов) – а не 81 [1]. Четырёхсегментный индикатор «ВАРВАРА» также можно подключить по динамическому принципу (рис. 2).

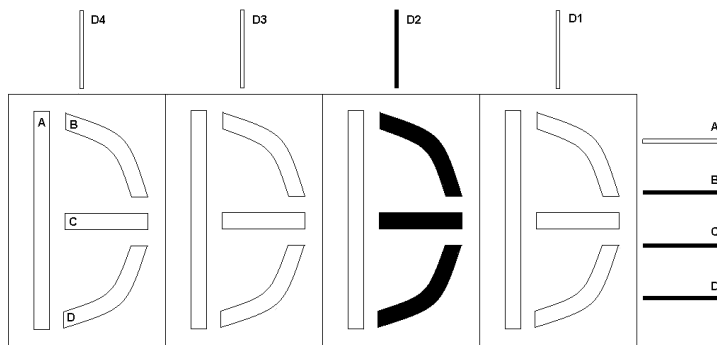


Рис. 2. Динамический принцип подключения индикатора «ВАРВАРА»

Достоинства индикатора «ВАРВАРА»:

1. Меньшее количество сегментов при возможности отображения всех десятичных цифр (рис. 3).
2. Более простой в плане подключения. Это достоинство может быть полезно, например, начинающим радиолюбителям.
3. Имеет «иммунитет» к горизонтальному отражению и поворотам, что связано с его несимметричностью.

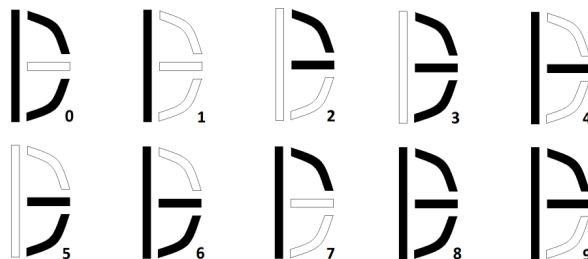


Рис. 3. Эквиваленты арабских цифр, отображаемые индикатором «ВАРВАРА»

Что касается недостатков данного индикатора, стоило бы остановиться на следующих:

1. Порой отходящий от привычного представления внешний вид цифр, а следовательно, невозможность всеобщего внедрения.
2. Не такое явное преимущество над семисегментными и другими многосегментными индикаторами, в связи с их возможностью работать по динамическому принципу.
3. Невозможность распознавания отображаемой цифры при выходе из строя хотя бы одного из четырёх сегментов. Однако стоит отметить, что и семисегментный индикатор в этом плане не идеален. Цифру «8», например, не отличить от нуля при выходе из строя центрального сегмента. А цифру «6» от цифры «5» при нерабочем левом нижнем сегменте. «0» же может казаться, например, цифрой «9», если не работает левый нижний сегмент.

Учитывая все преимущества и недостатки данного индикатора, предлагаются следующие способы его применения:

1. В технической сфере для различных приборов, обслуживающий персонал которых возможно обучить в короткий срок.
2. В декоративных целях, например, для отображения времени (рис. 4) и даты в электронных часах.
3. На газовых счётчиках, счётчиках воды и электричества, которые порой располагаются в труднодоступных местах, в перевернутом состоянии.

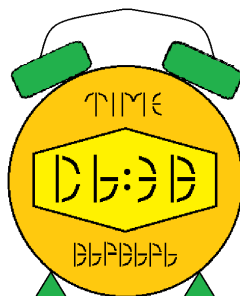


Рис. 4. Цифровые часы с индикатором «ВАРВАРА»

Библиографический список

1. Батушев В. А. Электронные приборы: учебник для вузов. – 2-е, перераб. и доп. – М.: Высшая школа, 1980. – С. 302-303. – 383 с.
2. Электроника 4-71Б, URL: http://www.leningrad.su/museum/show_calc.php?n=12, 2012-12-09

УДК 004.04.09

Ю.А. Страбыкина

Студентка кафедры проблемно-ориентированных вычислительных комплексов

В.С. Павлов

д-р техн. наук, профессор – научный руководитель

ПРОГРАММНОЕ УПРАВЛЕНИЕ ПРОИЗВОДИТЕЛЬНОСТЬЮ И НАДЕЖНОСТЬЮ СИСТЕМ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА

Репозиторий Alfresco [1] и требования к аппаратным средствам веб-клиента являются переменными, и в значительной степени зависят от количества одновременно работающих пользователей, которые получают доступ к системе. Для обеспечения виртуализации одним и распространенных программных обеспечений является VMware Workstation [2].

Цель работы состоит в исследовании условий, при которых достигается наибольшая производительность ECM-системы Alfresco в случае ее реализации на виртуальной машине (ВМ) VMware Workstation и обеспечивается надлежащий уровень надежности холодного резервного копирования и восстановления данной ВМ при больших объемах обрабатываемых данных.

Рекомендуются следующие меры для получения наилучшей производительности:

- избегать 32-битные системы. Тест показал значительный прирост производительности при использовании 64-разрядного оборудования;
- использовать систему с тактовой частотой выше 2.5 ГГц;
- зарезервировать достаточный RAM (объем оперативной памяти) для операционной системы, после требуемой памяти для ВМ VMware;
- хранить индексы поиска на локальном диске, а не на сетевом ресурсе.

Размер репозитория Alfresco определяет объем требуемого дискового пространства. По умолчанию содержимое в Alfresco хранится непосредственно на диске, поэтому, например, для хранения 1000 документов 1 MB потребуется 1000 MB дискового пространства. Также нужно убедиться, что достаточно пространства для временных файлов и версий, так как каждая версия файла хранится на диске в виде отдельной копии этого файла.

Производительность чтения/записи содержимого исключительно зависит от скорости сети и скорости дискового массива. Дополнительные расходы Alfresco для чтения контента очень низкие, как содержимое передается непосредственно с дисков в выходной поток. Дополнительные расходы считывания также низкие, но в зависимости от параметров индексации могут быть некоторые дополнительные расходы, например, содержимое индексируется или метаданные, извлеченные из содержимого в каждом файле [3].

Alfresco хорошо совместим с виртуализацией, однако ожидается значительное снижение производительности. При настройке машины необходимо выделить в два раза больше ресурсов для заданного числа пользователей, если эти ресурсы являются виртуальными. Паравиртуализация, или виртуализированные доступы к собственным объемам узла не потребуют много ресурсов.

Хранилище кэша вместе с начальными издержками ВМ и базовой системой памяти Alfresco требует установить по умолчанию примерно 1024 MB. Это означает, что можно запустить хранилище Alfresco и веб-клиента с большим количеством пользователей, получающих доступ к системе с одним сервером центрального процессора компьютера (CPU) и только 1024 MB оперативной памяти предназначено Alfresco в VMware, тем не менее нужно добавить дополнительную память, так как пользовательская база растет.

Процессоры добавляются в зависимости от сложности задач, которые ждут выполнять пользователи и сколько одновременных пользователей доступа к клиенту.

Одновременными пользователями считаются пользователи с постоянным доступом к системе через веб-*Alfresco*. Обратим внимание, что N одновременных пользователей считается эквивалентным $10N$ случайных пользователей.

Предлагаемые настройки памяти совместно с CPU на сервере:

1. Для 50 одновременно или до 500 случайных пользователей: 1,5 GB оперативной памяти VM, CPU сервера типа 2x (или 1xDual-ядро).
2. Для 100 одновременно работающих пользователей или до 1000 случайных пользователей: 1,5 GB оперативной памяти VM, CPU сервера типа 4x (или 2xDual-ядро).
3. Для 200 одновременно работающих пользователей или до 2000 случайных пользователей: 2,5 GB оперативной памяти VM, CPU сервера типа 8x (или 4xDual-ядро).

Alfresco Enterprise 2.1 и *MySQL 5* на *Windows 2000* с *Tomcat* был развернут на сервере. Примерно 10000 документов и 10 экземпляров пользователей были импортированы в систему.

Клиент – пользователь, с небольшой паузой между запросами (3-10 секунд максимум) и непрерывным круглосуточным доступом. Обычными пользователями считаются пользователи, иногда получающие доступ к системе через веб-клиент *Alfresco* с большим временем между запросами (например, случайный доступ к документам в течение рабочего дня). Максимальный размер памяти VM можно рассчитать по формуле: $V_1 + V_0 + V_S L < 2 \text{ GB}$, где V_1 – объем управляемых данных; V_0 – объем исходных данных; V_S – размер стека потока; L – число потоков.

Из приведенной формулы следует, что максимальный размер памяти не может превышать 2 GB на 32-битных операционных системах (ОС) x86 для *Windows* или *Linux*. Это означает, что даже если устанавливается 4 GB оперативной памяти в сервер, один экземпляр VM *Java* не может выйти за пределы 2 GB на 32-битной машине сервера.

Обратим внимание, что 64-разрядная версия имеет гораздо большее поле адресации. Рекомендуется использование 64-битной ОС с большой аппаратной памятью (> 2 GB) используется для развертывания > 250 одновременных или > 2500 случайных пользователей.

Клонирование VM предполагает выполнение последовательности следующих действий.

1. Выключить копируемую VM и на ее вкладке в основном окне *VMware* перейти по ссылке «*Clone this virtual Machine*» («Клонировать эту VM»). Это приведет к запуску мастера клонирования.

2. Полуавтоматическая конфигурация предложит выбрать, какое именно состояние копируемой VM считать «эталонным». Для новой VM, для которой еще не создавались снимки состояния, таковым является текущее состояние, и потому в окне мастера будет доступен лишь переключатель «*From current state*» («Из текущего состояния»).

3. Далее выбирается способ клонирования.

«*Create a linked clone*» («Создать связанный клон») – в этом случае в действительности создается не клон, а «зеркальное отражение» исходной VM, поскольку содержимое виртуальных дисков и некоторые другие данные не копируются, на них лишь создаются ссылки, соответственно, каждое изменение в исходной VM ведет к идентичным изменениям клона (такой вариант удобен тем, что не требует значительного пространства на жестком диске хост-компьютера; вместе с тем связанный клон может работать только на том хост-компьютере, где расположена исходная VM);

«*Create a full clone*» («Создать полный клон») – в этом случае дублируются все файлы исходной VM и помещаются в указанную пользователем папку на жестком диске хост-компьютера.

В завершение, вне зависимости от выбранного метода клонирования потребуется указать имя новой VM и ее расположение. Достоинством операции клонирования в *VMware* является то, что каждый клон получает собственный MAC-адрес и уникальный идентификатор (*Universal Unique Identifier, UUID*). Это позволяет объединять клоны в сеть. Нельзя клонировать VM, созданные в предыдущих версиях *VMware Workstation*.

В результате проделанной работы было выявлено, что для наилучшей производительности требуется следующее.

- использование 64-разрядного оборудования;
- убедиться в наличии достаточного объема дискового пространства. Исходя из того, что для хранения 1000 документов 1 MB требуется 1 GB дискового пространства;

- зарезервировать объем оперативной памяти более 1024МВ, так как пользовательские базы растут. Как приводилось ранее, для 200 одновременно работающих пользователей или до 2000 случайных пользователей ориентировочно потребуется 2,5 GB оперативной памяти виртуальной машины.

При восстановлении данных после холодного резервного копирования в критическом объеме 100000 документов, импортированных в развернутую в указанных условиях ЕСМ-систему Alfresco, были полностью восстановлены. В связи с этим можно сказать, что холодное резервное копирование всей виртуальной машины является надежным и производительным.

Библиографический список

1. Интернет-портал LiveBusiness.Doc. url:<http://www.doc-online.ru/software/alfresco>
2. Официальный сайт VMware. url: <https://www.vmware.com/ru/products/workstation.html>
3. Официальный сайт Alfresco.url:<https://community.alfresco.com/community/ecm>

УДК 004.04.09

Ю.А. Страбыкина

Студентка кафедры проблемно-ориентированных вычислительных комплексов

В.С. Павлов

д-р техн. наук, профессор – научный руководитель

СПЕЦИФИКА ОРГАНИЗАЦИИ ХРАНЕНИЯ ДАННЫХ В СИСТЕМАХ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА УРОВНЯ ПРЕДПРИЯТИЯ

Система электронного документооборота (СЭД) – это общее понятие систем, автоматизирующих организационно-распорядительный документооборот на предприятии (регистрация входящих, исходящих, организационно-распорядительных документов). Такие системы ведут учет документов и их исполнение. Информационной единицей СЭД является документ, у которого есть заданный набор реквизитов [1].

В соответствии с особенностями делопроизводства существует 2 вида информационных данных на любом предприятии: структурированные данные и неструктурированные данные. Структурированные это «считываемые» данные, которые представлены в структурированном виде (таблицы, записи) и могут достаточно просто восприниматься машинным образом.

Неструктурированные данные (или контент) это «читаемые» данные, такие как тексты, таблицы, чертежи и графики, презентации, документы в формате PDF, видео- и аудиофайлы, сообщения электронной почты и даже СУБД, расположенные на локальных и сетевых дисках, на сайтах в интернете. Воспринимаются такие данные преимущественно самим человеком [2].

Неструктурированные данные принято называть корпоративным контентом (Enterprise Content), система управления корпоративным контентом (Enterprise Content Management) называется ЕСМ. В России два близких понятия СЭД и ЕСМ иногда отождествляются в силу пересечения функций. Цель ЕСМ-систем заключается в создании и поддержке единого информационного пространства организации, непротиворечивости информации в разных документах, обеспечении получения каждым пользователем всей необходимой информации в удобном виде в соответствии с правами доступа. Функции ЕСМ-системы представлены в виде схемы на рис. 1.



Рис. 1. Функции ЕСМ-системы

Одним из факторов, влияющих на выбор СЭД, является способ распространения и лицензирования программного обеспечения (свободное или проприетарное), совокупная стоимость его владения

ем, возможность доработки под конкретные нужды. Так, в ряде случаев более эффективным для проведения научных вычислений, экспериментов и разработок будет использование свободного программного обеспечения (СПО).



Рис. 2. Виды программного обеспечения

Промышленное внедрение научно-технических разработок с использованием проприетарного ПО на крупных предприятиях влечет зависимость от данных продуктов и их дорогостоящую закупку для каждого требуемого компьютера. В большинстве случаев реально будет использоваться мизерная доля функциональности купленных программных пакетов. Для решения многих задач качество СПО лишь не значительно хуже, а в ряде случаев и лучше коммерческих продуктов. Развитие и повышение качества СПО обеспечивается именно благодаря идее активного сотрудничества, открытого обмена опытом, научными и программными наработками различными университетами, коммерческими и некоммерческими организациями по всему миру с целью достижения общей выгоды. Обычно стоимость владения свободным ПО оказывается значительно ниже стоимости владения проприетарным продуктом, а потребитель получает доработанный под его конкретные потребности программный продукт, наиболее полно удовлетворяющий его требованиям, и оперативную техническую поддержку в отличие от корпоративных проприетарных продуктов [6].

Одним из недооцененных свободных программных продуктов является ECMAlfresco. Существуют версии продукта – AlfrescoCommunity и AlfrescoEnterprise. Технические возможности версий не отличаются, но сборки Community распространяются абсолютно свободно и без какой-либо поддержки, а Enterprise-версия ориентирована на корпоративное использование и приобретается вместе с официальной поддержкой от разработчиков [7].

Говоря об Alfresco, стоит отметить, что это в первую очередь платформа, а не «коробочный» продукт. Основное преимущество Alfresco – простое расширение и работа по модели OpenSource с открытым программным кодом для быстрого создания на ее базе своих решений и отсутствие лицензионных отчислений за владение системой. Другая ее особенность – полная кросс-платформенность – серверная и клиентская части могут работать под управлением различных операционных систем, таких как: MS Windows, Linux, Mac OS . Кроме того, Alfresco ECM обладает русскоязычным интерфейсом, а ее алгоритмы создания маршрутов документов и вся организационная структура максимально адаптирована под «русское» восприятие.

Целью работы является анализ достоинств и недостатков платформы Alfresco,ее архитектуры и возможностей. Для иллюстрации принципиально важной возможности обеспечения сохранности данных в Alfresco будет рассмотрен пример реализации резервного копирования и восстановления с использованием сервера баз данных в операционной системе Linux.

Всю платформу Alfresco можно разделить на несколько основных компонентов. DocumentManagement (DM) – система управления документами (унифицированное хранилище данных, поддержка бизнес-процессов). RecordsManagement (RM) – архив данных, построенный на основе репозитория данных Alfresco. WebContentManagement – система управления веб-контентом (совместная работа над материалами, веб-публикации). Share – система совместной работы, позволяющая создать подобие социальной сети внутри организации [7].

Основным преимуществом открытых ЕСМ является отсутствие зависимости от одного поставщика. Закрытые системы жестко связаны с другими продуктами того же разработчика, такими как СУБД, офисный пакет, почтовые сервера. Замена любого компонента закрытой системы вызывает проблемы интеграции и требует проведения большого объема работ. Открытые ЕСМ основаны на открытых стандартах, что обеспечивает простую интеграцию со сторонними продуктами [5].

Из экономических аспектов внедрения основным преимуществом открытых ЕСМ является минимальная цена «входа». Это обосновано тем, что нет необходимости в покупке лицензий перед началом внедрения системы, оплачивается только поддержка и обновления.

Другой немаловажной особенностью системы с точки зрения разработчика является наличие большого числа доступных API (RESTful API [8], Java API [9], JavaScript API [10]), с помощью которых легко получить доступ к репозиторию, задачам, системе управления сайтами и т.д. Более того, Alfresco дает возможность писать пользовательские веб-скрипты, которые расширяют функциональность стандартных. К примеру, можно легко написать скрипт, позволяющий загружать файлы в репозиторий и запускать бизнес-процессы автоматически. Наличие открытого API дает возможность интегрировать систему в инфраструктуру предприятия с минимальными затратами на написание кода[12].

Работать с программой можно через любой веб-браузер. Также возможна интеграция с клиентами: MS Office, SAP, IBM Lotus, Kofax, GoogleDocs, Zimbra. Можно подсоединить архив Alfresco к системе как внешний сетевой диск (SMB, FTP).

Архитектура Alfresco может быть представлена в виде трех уровней (рис. 3).

- первый уровень отвечает за физическое размещение данных репозитория на сервере;
- второй уровень – сервер управления контентом. Он реализован на Java и поддается масштабированию при работе под большими нагрузками. Сервер управления контентом состоит из непосредственно репозитория контента и набора сервисов, через которые пользователь может получать доступ к контенту;
- третий уровень – клиенты, подключенные к серверу управления контентом через различные протоколы и интерфейсы API[11].

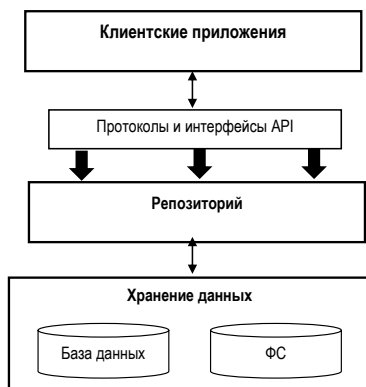


Рис. 3. Архитектура Alfresco

Таким образом, Alfresco полностью отвечает за реализацию первых двух уровней и позволяет разработчикам подключать свои приложения на третьем уровне.

Также Alfresco поддерживает возможность кластеризации как для задач повышения отказоустойчивости, так и для задач повышения производительности. Благодаря модульности системы все ключевые компоненты Alfresco могут быть разнесены по разным физическим серверам и дублированы. При обработке больших объемов данных используются следующие компоненты системы:

- внешние сервера хранения данных (SAN, NAS, CAS);

- СУБД в режиме кластера (Oracle, DB2, PostgreSQL);
- масштабируемые серверы приложений (Jboss, WebSphere), поверх которых работает распределенный сервер Alfresco;
- сервер индексирования контента SOLR может быть вынесен на специализированный физический сервер и также может быть масштабирован при необходимости [11].

Как уже было отмечено ранее, система работает с неструктурированными данными, для хранения которых активно используется бесплатная СУБД PostgreSQL в версии продукта AlfrescoCommunity.

PostgreSQL, как и любая другая СУБД, имеет богатые возможности по резервному копированию как кластера баз данных (БД), так и отдельных баз, но основным механизмом управления при этом является командная строка. Несмотря на то, что утилита PgAdmin позволяет выполнять основные задачи через графический интерфейс, работа с PostgreSQL выполнялась через командную строку, что позволяет уверенно чувствовать себя в любой ситуации и открывает широкие возможности по автоматизации.

Для повышения надежности хранения информации обеспечим автоматическое резервное копирование в соответствии с заданными условиями на сервере СУБД, где расположены базы Alfresco. Прежде всего настроим авторизацию для СУБД. Так как основные операции будут производиться из скрипта, то имеет смысл разрешить локальный доступ к СУБД без авторизации. Учитывая, что доступ к серверу БД имеет ограниченный круг лиц и расположен он в периметре сети, безопасность пострадает слабо.

Откроем файл `pg_hba.conf`, он находится в `/var/lib/postgresql/data` и приведем к следующему виду строку: `localalltrust`.

Перезапустим СУБД: `service postgresql restart`.

Для создания резервной копии воспользуемся утилитой `pg_dump`, которая позволяет создать дамп [13] для указанной БД. Создание дампа происходит без блокирования таблиц и представляет снимок БД на момент выполнения команды, таким образом создаются дампы во время работы пользователей.

Синтаксис `pg_dump` предельно прост, нам нужно указать имя базы и расположение и название файла дампа. Просмотреть список баз можно командой:

```
psql -U postgres -l
```

Кроме списка баз вывод содержит ряд полезной информации, например о кодировке базы, данная информация пригодится нам при восстановлении БД на другом сервере.

Теперь, уточнив наименование баз на сервере создается резервная копия базы Alfresco:

```
pg_dump -U postgres alfresco > ~/alfresco.pgsql.backup.
```

Результатом выполнения команды будет файл дампа в домашней директории. Расширение файла может быть любым, но лучше указывать таким образом, чтобы по нему было понятно назначение данного файла `pgsql.backup`, это может быть важно, если поиском дампов будут заниматься коллеги.

При необходимости можем создать сжатый дамп:

```
pg_dump -Upostgresalfresco | gzip> ~/alfresco.pgsql.gz.
```

Сжатие позволяет уменьшить размер дампов примерно вдвое, поэтому следует его использовать при передаче резервных копий по сети интернет или при ограниченном размере хранилища.

Перед восстановлением дампа для надежности можно создать целевую БД (при ее отсутствии). Для создания базы выполним:

```
createdb - T template0 alfresco2.
```

Теперь перенесем полученный дамп в только что созданную базу **alfresco2**:

```
psql - U postgres alfresco2 < ~/alfresco2.pgsql.backup.
```

Например, файл дампа находится в **С:\backup\домашней директории** соответственно.

Разработанный скрипт для Linux инициализируется командой

```
touch /etc/pgsql-backup,
```

а его программный код приводится ниже.

```
#!/bin/sh
#Зададим переменные
DATE=$(date +%Y%m%d)
FTP=ftp.domain.local
```

```

FTPU=«user»
FTPP=«password»
#Резервноекопирование
cd /root/backup
#Базаalfresco
pg_dump -Upostgresalfresco | gzip>$DATE-alfresco.pgsql.gz
ftp -n$FTP<<END
quoteUSER$FTPU
quotePASS$FTPP
bin
passive
put$DATE-alfresco.pgsql.gz
quit
END
#очистка
rm-f$DATE-alfresco.pgsql.gz
Сохраним с скрипт и дадим права на выполнение:
chmod +x /etc/pgsql-backup
Также не забудем создать каталог /root/backup
mkdir /root/backup

```

В результате проделанной работы установлено, что платформа Alfresco является единственной полнофункциональной Open-Source ECM-системой (с открытым программным кодом). Наряду с этим, в платформе Alfresco присутствует гибридное «облачное» расширение, обеспечивающее синхронизацию контента. Из недостатков выявлен малый объем общедоступной прикладной информации о данной системе, что затрудняет адаптацию Alfresco к документообороту крупных компаний.

С целью иллюстрации эффективности и надежности резервного копирования и восстановления больших объемов данных в Alfresco создан скрипт, который формирует резервные копии базы данных PostgreSQL и размещает их на FTP-сервере.

Библиографический список

1. РБК. Системы электронного документооборота/ECM-системы. url:http://marketing.rbc.ru/reviews/it-business/chapter_2_2.shtml
2. Русскоязычный портал сообщества. url: <http://www.ossportal.ru/technologies/alfresco/blogs/738>
3. Информационно-аналитический портал DOCFLOW. Корпоративные хранилища документов. url: http://www.docflow.ru/news/analytics/detail.php?ID=21126&sphrase_id=1443792
4. PCWEEK(Документооборот/ECM). Мировой ECM-рынок: без революционных перемен. url:<http://www.pcweek.ru/ecm/article/detail.php?ID=180728>
5. ECM портал. Мировой рынок ECM. url:<http://www.ecm-portal.ru/practice/expertise/336/>
6. Электронная библиотека БГУ. Свободное программное обеспечение. url:<http://elib.bsu.by/>
7. Официальный сайт Alfresco (англ.)url: <http://www.alfresco.com>
8. RESTful API (англ.) url: http://wiki.alfresco.com/wiki/Repository_RESTful_API_Reference
9. Java API (англ.) url: http://wiki.alfresco.com/wiki/Java_Foundation_API
10. JavaScript API. url:http://wiki.alfresco.com/wiki/JavaScript_API
11. *Caruana D., Newton J., Farman M.* Professional Alfresco: Practical Solutions for Enterprise Content Management, Published by Packt Publishing Ltd., 2010.569p.
12. Potts J. Alfresco Developer Guide, Published by Packt Publishing Ltd.2008.533p.
13. Что это такое. url:<https://chto-eto-takoe.ru/dump>

УДК 004.052.2

И.В. Токаревский

магистрант кафедры вычислительных систем и сетей

А.А. Востриков

канд. техн. наук, доцент – научный руководитель

ПРОГРАММНО-АППАРАТНЫЕ РЕАЛИЗАЦИИ АЛГОРИТМОВ МАСКИРОВАНИЯ ДАННЫХ ВО ВСТРАИВАЕМЫХ МИКРОПРОЦЕССОРНЫХ СИСТЕМАХ

В современном мире автоматизированным встраиваемым микропроцессорным системам стали доверять управление критическими процессами, такими как мониторинг движения транспортных средств, видеонаблюдение с беспилотных летательных аппаратов, так и расчет гипотез по дальности движения баллистических ракет. Последствия несанкционированных воздействий и сбоев в таких системах могут носить весьма серьезный, а зачастую и катастрофический характер. Поэтому на настоящий момент актуальными задачами прикладного направления, связанного с трансляцией цифрового изображения, является защита информации от несанкционированного доступа и обнаружение фактов подмены и искажения передаваемых данных.

Как правило, при разработке типичных System-on-a-Chip (SoC) систем встраиваемого класса, разработчики не ставят цель защитить информацию от несанкционированного доступа при передаче информации по каналам связи внутри устройства. В данной же работе стоит цель провести обзор возможных решений программно-аппаратных реализаций алгоритмов маскирования во встраиваемых системах.

Одним из способов процедуры прямого и обратного преобразования цифровой визуальной информации на основе криптографических методов является криптографическое маскирование и демаскирование. Ввиду того что изображение, как вид информации, обладает свойством избыточности, разработчики методов криптографического маскирования стараются разрушить в маскированном изображении низкочастотные составляющие. Например, изображения, содержащие высококонтрастные объекты, имеют высокую избыточность, что влечет за собой сложность в процессе преобразования цифровой визуальной информации к шумоподобному виду. Кроме того, в условиях длительной защиты от несанкционированного доступа, приходится сокращать вычислительную емкость криптографических методов.

Наибольшее практическое значение среди программных шифров имеют симметричные блочные шифры, сочетающие как высокую скорость преобразования информации, так и возможность обеспечения независимого шифрования отдельных блоков данных, поэтому разработка скоростных блочных шифров является важной задачей прикладной криптографии.

Все современные криптосистемы построены по принципу Керкгоффса, т.е. секретность зашифрованных сообщений определяется секретностью ключа. Другими словами, сам алгоритм шифрования известен криптоаналитику, но он не в состоянии расшифровать сообщение, если не располагает соответствующим ключом. Очевидно, что стойкость таких шифров определяется размером используемого в них ключа.

Так, например, в виде операций замены и перестановки с некоторым раундовым ключом K , аналогичных ГОСТ 28147-89, достигается надежная защита от основных типов атак. А использование схемы сетей Фейстеля с не менее чем восемь раундами шифрования обеспечивает свойством разрушения структуры видеoinформации, тем самым гарантируя надежное маскирование передаваемой информации. Авторами статьи [1] предложено использовать любой симметричный алгоритм блочного преобразования типа ГОСТ, как функцию от двух переменных $F(K_0, i)$, где K_0 – начальное значение параметра, i – номер передаваемого сообщения. Таким образом, в качестве функции $F(*, *)$ выбрано арифметическое сложение 256 битных чисел с игнорированием старшего разряда.

Явным преимуществом метода криптографического маскирования является относительно более высокая устойчивость к атакам по сравнению с другими видами маскирования, но недостаток метода в том, что требуется более высокая вычислительная ресурсоемкость.

Другим способом маскирования цифровой визуальной информации является использование матричной арифметики. Такой подход к защите данных вызван прикладными особенностями ряда микросхем, таких как интегральные схемы цифровых сигнальных процессоров и программируемой логики в качестве вычислительной основы встраиваемых систем. Интегральные схемы таких классов обладают так называемыми DSP (Digital signal processor) – модулями, ускоряющими операции скалярного произведения векторов и операции с плавающей запятой.

В работах [2,3] предлагается строить блочные криптографические шифры на основе обратных матриц. В основу алгоритмов положен модифицированный асимметричный протокол Диффи-Хеллмана. Суть модификации сводится к замене больших простых чисел алгоритма гарантированно невырожденными n -полными двоичными матрицами высокого порядка. Такие матрицы обладают следующим свойством: количество различных элементов в общем случае крайне мало, поэтому при вычислении матриц малое количество возможных вариантов множителя позволяет заменить операцию умножения на табличную выборку, за счет чего сокращаются необходимые аппаратные ресурсы.

Рассмотрим краткое описание протокола обмена ключами шифрования между абонентами A и B .

1. Предварительно абоненты договариваются о выборе некоторой n -полной матрицы M порядка n , которой соответствует обратная матрица M^{-1} . В результате перестановки n их строк и столбцов образуются матрицы M и M^{-1} .

2. Абонент A вырабатывает случайный показатель x , вычисляет матрицу $A=M^{-1x}$ и посылает ее пользователю B .

3. Абонент B вырабатывает случайный показатель y , вычисляет матрицу $B=My$ и посылает ее пользователю A .

4. Абоненты A и B возводят полученные матрицы в свои показатели, формируя на стороне Алисы матрицу шифрования MK , а на стороне Боба – матрицу расшифрования MK .

Такой подход к маскированию информации существенно сокращает вычислительную ресурсоемкость методов шифрования. Далее рассмотрим, какую аппаратную платформу для практической реализации предпочтительнее выбрать.

При выборе устройства для обработки и маскирования цифровых изображений разработчики рассматривают ряд параметров:

- производительность системы;
- энергопотребление;
- число компонентов и форм-фактор системы;
- предполагаемые области применения и возможность модернизации;
- экономические параметры, такие как разовые расходы на проектирование и внедрение в производство, стоимость используемых материалов, время вывода на рынок и риски проекта.

В таблице представлены основные DSP-устройства компании Texas Instruments и FPGA фирмы Intel (Altera) в различных ценовых категориях и мощностях (ММАС). ММАС (millions of multiply – accumulate operations per second) – величина, показывающая, сколько миллионов операций умножения с накоплением над 32 – битными числами с фиксированной точкой или числами с плавающей точкой с одинарной точностью выполняет устройство в секунду.

У обоих типов устройств – DSP и FPGA – есть сферы, где их применение является оптимальным решением. Например, FPGA – наилучший выбор для маскирования информации в сетевых приложениях, поддерживающих трафик на скоростях порядка гигабит в секунду, а шифрование с помощью DSP имеет преимущество в видео приложениях, таких как системы наблюдения. Вместе с тем существуют и задачи, для решения которых пригодны оба типа устройств.

Но все же важнейшим фактором, при котором DSP обходит оппонента является наличие полного дистрибутива Linux на базе процессоров ARM. Linux позволяет разработчикам использовать суще-

ствующее встроенное ПО с открытым исходным кодом и быстро интегрировать в ПО библиотеки третьих производителей.

Таблица

Свойства DSP и FPGA в различных ценовых категориях.

Семейства ПЛИС	Стоимость, доллары	ММАС	Семейства процессоров	Стоимость, доллары	ММАС
Cyclone II Cyclone III	10–30	270–660	DSP с фиксированной точкой C5000	<10	25–300
Cyclone II Cyclone III	30–100	380–1900	Цифровые медиапроцессоры DaVinci	10–30	50–1200
Cyclone II Cyclone III Stratix II	100–300	1000– 4500	DSP с фиксированной точкой C6000	30–100	50–1440
Cyclone II Cyclone III Stratix III	300–1000	1000– 8300	DaVinci DSP с фиксированной точкой C6000	100–300	266–2000
Stratix II Stratix III	1000–3000	3000– 11000	DSP с фиксированной точкой C6000	300–330	2000– 2400
Stratix II Stratix III	3000– 10000	4000– 11000			

Таким образом, рассмотрев методы маскирования информации на основе криптографических методов и с использованием матриц, как базиса ортогональных преобразований, более перспективный, на наш взгляд, является второй метод, ввиду того, что помимо процедуры шифрования такой метод позволяет выполнять традиционные для методов сжатия изображений усекающие действия в отношении наименее информативной части спектра и устранять избыточность в оставшемся объеме информации. С точки зрения аппаратной части, реализация алгоритмов маскирования на базе встраиваемых микропроцессорных систем более перспективна, так как на разработку готовой системы требуется меньше ресурсов, в отличие от программируемых логических интегральных схем.

Библиографический список

1. Беззатеев С. В., Литвинов М. Ю., Трояновский Б. К., Филатов Г. П. Выбор алгоритма преобразования, обеспечивающего изменение структуры изображений // Информационно-управляющие системы. 2006. № 6(25). С. 2–6.
2. Белецкий А.Я., Белецкий А.А., Стеценко Д.А. Матричные алгоритмы криптографической защиты информации и обмена ключами шифрования // Информационные технологии в образовании. 2010. № 7. С. 156-165.
3. Ерш И. Л., Сергеев М. Б. Скоростное шифрование разнородных сообщений // Вопросы передачи и защиты информации: сб. ст. СПб.: СПб ГУАП, 2006. С. 133-155.

УДК 004.451.9

И.В. Токаревский

магистрант кафедры вычислительных систем и сетей

А.А. Востриков

канд. техн. наук, доцент – научный руководитель

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ОС LINUX В МИКРОПРОЦЕССОРНЫХ СИСТЕМАХ ВСТРАИВАЕМОГО КЛАССА

Термин встраиваемая система используется, когда речь идет о небольшой ЭВМ, интегрированной в устройство, которым она управляет. Как правило такие системы имеют собственную операционную систему, включающую в себя комплекс программного обеспечения, предназначенный для управления ресурсами и организации взаимодействия с пользователем. Такой подход к реализации проектов удобен, когда устройство обладает большим количеством сложных интерфейсов ввода-вывода или выполняет сложные функции.

Существует большое количество различных встраиваемых операционных систем, предназначенных для управления специализированными устройствами и, как следствие, способные работать в условиях ограниченных ресурсов. Характерными особенностями встраиваемых ОС является модульная структура, компактность, производительность и масштабируемость. Именно такими характеристиками обладает встраиваемая ОС Linux.

Большой набор возможностей Linux проистекает из модели открытых исходных кодов, поощряющей совместное владение кодом [1]. Другими словами, инженеры, зачастую работающие в разных компаниях, могут объединить усилия для создания ядра ОС и поддержания ее актуальности. Огромная кодовая база дает возможность поддерживать и обслуживать самые разные устройства – от мощнейших суперкомпьютеров до смартфонов.

Таким образом, у проектировщика появляется большой набор возможных решений задачи, но вместе с ним и проблема выбора оптимального решения.

Далее попытаемся разобраться, как сконструировать встраиваемую систему с использованием ОС Linux из свободных программ с открытым кодом для создания надежного и эффективного продукта.

Итак, начнем с общего обзора встраиваемых Linux-систем, и определения причин, почему они так популярны.

Простой ответ дает закон Мура, который говорит о том, что плотность компонентов на кристалле микросхемы удваивается примерно каждые два года [2]. Сердцем большинства встраиваемых устройств является интегральная микросхема, содержащая одно или несколько процессорных ядер, интерфейсы с оперативной памятью и разнообразные периферийные устройства. Всё это называется «Системой на кристалле» (SoC – System on Chip), и сложность таких систем растет в соответствии с законом Мура. Как правило, для типичных SoC-систем прилагается справочное техническое руководство размером порядка тысячи страниц, к тому же устройства должны выполнять различные сложные функции. Для решения и управления всеми задачами нужна полноценная операционная система.

Ниже перечислено несколько причин для выбора на эту роль Linux.

- Исходный код Linux открыт, так что в него можно внести необходимые изменения.
- Linux портирована на многие процессорные архитектуры, в том числе применяемые в SoC-системах, таких как: ARM, MIPS, x86 и другие.
- В Linux уже имеется необходимая функциональность: поддержка USB, Wi-Fi, запоминающих устройств, присутствует планировщик задач, хороший сетевой стек и т. д.
- Linux всегда актуальна и поддерживает все современные протоколы и оборудование, ввиду того что существует активное сообщество. Так, например, новая версия ядра выходит раз в 10-12 недель, причем авторами кода являются порядка 1000 разработчиков.

В силу этих причин, создание сложных систем на основе Linux – отличный выбор. Они обладают достаточными возможностями и гибкостью для эффективной работы с оборудованием. Однако существует необходимость прикладывать значительные усилия для освоения того, как устроена встраиваемая система, и продолжать учиться по мере развития экосистемы Linux.

Говоря о проекте для встраиваемых систем, часто речь идет о четырех составляющих элементах системы, представленных в таблице.

Набор инструментов	Состоит из компилятора и прочих инструментальных средств, необходимых для создания программы, управляющей целевым устройством.
Начальный загрузчик	Необходим для загрузки и инициализации ядра.
Ядро	Является сердцем системы, осуществляет взаимодействие с оборудованием и управляет всеми ресурсами.
Корневая файловая система	Содержит библиотеки и программы, исполняемые после завершения инициализации ядра.

Первым элементом встраиваемой Linux-системы является набор инструментов. Он является отправной точкой проекта. Получить набор инструментов просто – необходимо скачать и установить пакет. Он предназначен для компиляции исходного кода в исполняемый файл, который можно выполнить на целевом устройстве. Прежде всего, набор инструментов нужен, чтобы собрать остальные три элемента встраиваемой Linux-системы: начальный загрузчик, ядро и корневую файловую систему.

Существует два типа наборов инструментов:

- Платформенный;
- Перекрестный.

Платформенный работает в системе того же типа, что и система, для которой генерируются программы [3]. Перекрестный работает в системе, отличной от целевой платформы, т.е. позволяет, например, вести разработку на быстром ПК, а загружать и тестировать сгенерированный код на встраиваемом устройстве. В большинстве случаев встраиваемые Linux-системы разрабатываются с помощью перекрестных наборов инструментов. Однако с другой стороны приходится кросс-компилировать все библиотеки и инструменты, необходимые для целевой платформы.

Затем возникает вопрос о выборе библиотеки C, так как программный интерфейс обращения к ОС Unix определен на языке C и закреплён в стандарте POSIX. Библиотека C является шлюзом к ядру Linux из программ. Существует несколько библиотек C на выбор: glibc, eglibc, uClibc, musl libc.

Таким образом, располагая набором инструментов, можно собрать другие компоненты встраиваемой системы, например, начальный загрузчик, который запускает процесс загрузки программного обеспечения.

Начальный загрузчик запускает устройство и загружает ядро операционной системы. Когда выполняются первые строки кода начального загрузчика, основные интерфейсы платы не доступны, поэтому недоступна как внешняя память, управляемая контроллерами NAND или MMC памяти, так и оперативная память, ввиду того что контроллер динамической памяти (DRAM) не инициализирован. Обычно в самом начале работает только одно процессорное ядро и статическая память на кристалле. Поэтому загрузка системы состоит из нескольких этапов.

В отсутствие надежной памяти код, исполняемый сразу после сброса или включения питания, должен храниться на кристалле SoC-системы (Код ПЗУ). Коду в ПЗУ доступна небольшая часть статической памяти с произвольной выборкой (SRAM), в которую можно загрузить небольшой фрагмент кода из нескольких предопределённых мест. Например, микросхемы OMAP компании Texas Instruments (TI) будут пытаться загрузить код из нескольких первых страниц флэш-памяти типа NAND, или из флэш-памяти, подключенной по интерфейсу SPI, или первых секторов MMC-устройства, или код в ПЗУ пытается читать поток байтов из сети Ethernet, из порта USB или UART. Если размера SRAM-памяти не хватает для размещения всего начального загрузчика, например, загрузчика U-Boot, приходится создавать

промежуточный загрузчик, который называется вторичным загрузчиком программ, или SPL (Secondary program loader).

Вторичный загрузчик (SPL) настраивает контроллер памяти и другие необходимые части системы, в преддверии загрузки третичного загрузчика, в динамическую оперативную память. Он может прочитать программу с некоторого запоминающего устройства, причем код может находиться, начиная с предопределенного смещения от начала флэш-памяти, или в известном файле, например, `u-boot.bin`, загрузчика TI `x-loader`.

В конце этого этапа в DRAM-памяти находится третичный загрузчик (TPL – third stage program loader), и SPL может перейти на его начало.

На данном этапе начинает работать полный начальный загрузчик (TPL), например, U-Boot. Как правило, для него существует интерфейс командной строки, который позволяет выполнять загрузку нового загрузочного образа и образа ядра во флэш-память, считывание и загрузку ядра.

В конце третьего этапа ядро находится в памяти, готовое к работе.

Ядро отвечает за управление ресурсами и организацию интерфейса с оборудованием, а значит, влияет практически на все аспекты программной системы. Ядро Linux можно комбинировать с программами GNU, работающими в пользовательском адресном пространстве, создавая дистрибутивы для работы на настольных компьютерах или на серверах (GNU/Linux) или с небольшим комплектом пользовательских утилит Busybox для создания компактной встраиваемой системы.

У ядра есть три основные задачи: управление ресурсами, интерфейс с оборудованием и предоставление API для пользовательских приложений.

Четвертый и последний элемент встраиваемой системы – корневая файловая система. Основная задача – создать минимальную корневую файловую систему, достаточную, чтобы получить приглашение оболочки.

Таким образом, рассмотрев четыре элемента встраиваемой Linux-системы, которые встречаются во всех проектах, можно сделать вывод о том, что Linux обладает достаточными возможностями и гибкостью для эффективной работы с оборудованием.

Библиографический список

1. *Karl F.* Producing Open Source Software: How to Run a Successful Free Software Project / O'Reilly Media.:1005 Gravenstein Highway North, Sebastopol, 2006. 250 p.
2. *Майоров С. А., Кириллов В. В., Приблуда А. А.* Введение в микро-ЭВМ. Л.: Машиностроение, Ленингр. отд-ние, 1988, 121 с.
3. *Симмондс К.* Встраиваемые системы на основе Linux / пер. с англ. А.А. Слинкина. М.: ДМК Пресс, 2017. 360 с.

УДК 004.004.9

А.С. Толубаева

магистрант кафедры проблемно-ориентированных вычислительных комплексов

Е. А. Бакин

канд. техн. наук, доцент – научный руководитель

ПОДОПТИМАЛЬНЫЙ АЛГОРИТМ ВЫРАВНИВАНИЯ СТРОКОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Практическая ценность неточного сопоставления последовательностей обоснована природой молекулярных данных, а именно наличием в них активных мутационных процессов, а также задачами поиска образца в генетических базах данных, задачами определения функциональных, структурных или эволюционных взаимосвязей организмов посредством сравнения их первичных структур, проблемой наличия ошибок в молекулярных данных [1]. Недостатком имеющихся классических оптимальных алгоритмов строкового выравнивания является большая вычислительная и пространственная сложность.

В данной работе будет рассмотрен эвристический алгоритм глобального выравнивания двух нуклеотидных последовательностей с пониженной сложностью. Эффективность предложенного метода оценивалась путем сравнения с оптимальным алгоритмом Нидлмана-Вунша посредством моделирования.

Предложенный эвристический алгоритм основан на алгоритме глобального парного выравнивания последовательностей Нидлмана-Вунша [2]. Алгоритм Нидлмана-Вунша использует подход восходящего динамического программирования и систему оценки, включающую матрицу весов и штраф за разрыв, для вычисления оценки схожести последовательностей и нахождения оптимального выравнивания. Такое выравнивание содержит совпадающие и несовпадающие символы и разрывы в двух последовательностях, расположенные таким образом, чтобы число совпадений между идентичными или связанными символами было максимально возможным.

Рекуррентное соотношение для данного алгоритма может быть записано следующим образом:

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + S(A_i, B_j), \\ F(i-1, j) + d, \\ F(i, j-1) + d, \end{cases} \quad (1)$$

$$i = \overline{0, n}, \quad j = \overline{0, m},$$

где $F(i, j)$ – взвешенное редакционное расстояние, S – матрица весов, d – штраф за разрыв (вставка, удаление), n, m – длины последовательностей A, B, A_i, B_j – текущие элементы последовательностей A и B соответственно.

Данный алгоритм является оптимальным, т.е. позволяет найти такое выравнивание, при котором значения оценки схожести последовательностей на каждом шаге максимальны. Однако вычислительная и пространственная сложность этого алгоритма может быть оценена как $O(nm)$ [1]. Такая сложность алгоритма часто становится критической при больших n, m , что характерно для геномных данных.

Одним из подходов к решению данной проблемы может служить усечение диапазона поиска выравнивания последовательностей на основании статистических данных и заданных параметров (размер k -меров, требуемая доля совпадений подстрок p).

Применение алгоритма глобального выравнивания двух последовательностей обосновано в том случае, когда последовательности имеют приблизительно одинаковую длину и есть основания полагать, что они гомологичны по всей длине [3]. Оценка регионов, в которых последовательности схожи, до начала процедуры выравнивания поможет избежать необоснованных вычислительных затрат. С этой целью строится так называемая точечная матрица D (dot-matrix), отражающая регионы близкого

сходства двух последовательностей [4]. Элемент точечной матрицы $D(i, j)$ содержит «точку» (логическую единицу), если подстрока длины k , начинающаяся в позиции i строки A , идентична подстроке длины k , начинающейся в позиции j строки B :

$$D(i, j) = \begin{cases} 1, & \text{если } A(i : i + k) = B(j : j + k), \\ 0, & \text{иначе,} \end{cases}$$

$$i = \overline{1, n}, \quad j = \overline{1, m}, \quad (2)$$

где k – размер искомого подстрок (k -меров), n, m – длины последовательностей A, B .

В построении точечной матрицы использован алгоритм быстрого поиска подстрок Кнута-Морриса-Практа (КМП), вычислительная сложность которого оценивается как $O(n + m)$ [5].

Следующим шагом в решении задачи является определение границ области поиска выравнивания по полученному облаку точек. Одним из простых подходов определения границ области поиска выравнивания может служить задача глобального выравнивания с k различиями, позволяющая уменьшить границу используемых при решении времени и памяти с $O(nm)$ до $O(km)$. Однако подход отличается симметричностью интервала относительно главной диагонали, что приводит к увеличению диапазона поиска выравнивания в случае наличия больших геномных перестроек (нелокальных вставок или удалений) в одной из последовательностей, либо к невозможности найти оптимальное выравнивание в заданных границах (при $|n - m| > k$).

В случаях, когда точечная матрица носит выраженный асимметричный характер, можно оценить какие из диагоналей матрицы содержат наибольшее число совпадений k -меров. Определение диагоналей, между которыми достигается требуемая доля совпадений (параметр алгоритма p), позволяет более точно задать границы области поиска выравнивания.

Пусть $G(h_1, h_2)$ – область поиска выравнивания, заключенная между диагоналями (h_1, h_2) точечной матрицы. Тогда искомые границы области поиска выравнивания могут быть записаны следующим образом:

$$h_1^*, h_2^* = \operatorname{argmin}_{h_1, h_2} \left\{ |h_1 - h_2| : \frac{1}{K} \sum I\{(i, j) \in G(h_1, h_2)\} \geq p \right\}, \quad (3)$$

где (i, j) – координаты k -меров, K – количество совпадений подстрок, $p = 0,9$.

Тогда область поиска выравнивания задается как

$$S = G(h_1^*, h_2^*). \quad (4)$$

Ниже приведен алгоритм поиска выравнивания последовательностей.

Шаг 1. Разбиение одной из строк на подстроки длины k , k -меры.

Шаг 2. Поиск уникальных k -меров на точное совпадение во второй последовательности по алгоритму КМП. Совпадения помечаются точками в матрице dot-matrix. Характер размещения точек в матрице зависит от параметра k – размера искомого подстрок (параметр алгоритма). Если строки имеют значительное сходство, облако точек располагается в окрестностях главной диагонали матрицы, в противном случае они признаются несхожими.

Шаг 3. Разбиение облака точек по диагоналям, которым они принадлежат. Расчет доли совпадающих участков двух последовательностей, лежащих на каждой из диагоналей.

Шаг 4. Определение границ области поиска выравнивания путем нахождения двух диагоналей, в границы которых попадает 90% (параметр алгоритма p) совпадающих участков.

Шаг 5. Поиск подоптимального выравнивания с использованием алгоритма Нидлмана-Вунша в пределах найденных диагоналей (h_1^*, h_2^*) .

Для целей демонстрации работы алгоритма сгенерируем исходную строку над алфавитом $\{A, T, G, C\}$ заданной длины, вторую строку получим путем внесения мутаций (вставки, удаления, замены сегментов исходной строки). Моделирование выполнялось с использованием высокоуровневого языка технических расчетов MATLAB.

На рис. 1 приведен пример зависимости точечной матрицы для строк длины 1000 от параметра k – размера искомого подстрок (при $k = 4$ матрица содержит случайные совпадения подстрок, 204

«шум»). Цветом на рис. 1 выделена область поиска выравнивания, ограниченная двумя диагоналями h_1^* и h_2^* .

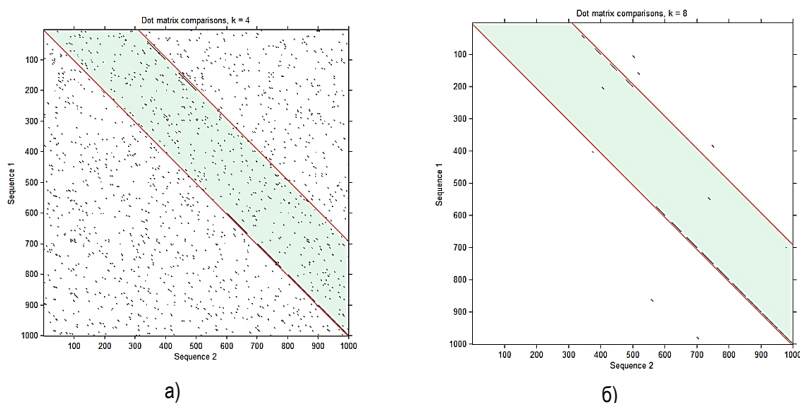


Рис. 1. Зависимость точечной матрицы для строк длины 1000 от параметра k : а) $k = 4$; б) $k = 8$

Рис. 2-3 отражают результат работы алгоритма на строках длины 2000, $k=8$, требуемая доля совпадений строкв пределах области поиска выравнивания – 90%.

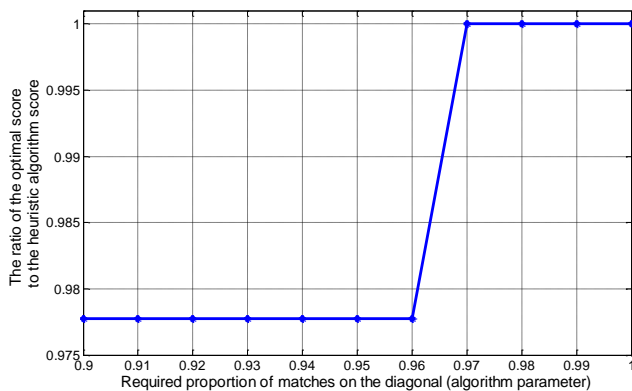


Рис. 2. Отношение оценок схожести, полученных оптимальным алгоритмом Нидлмана-Вунша и эвристическим алгоритмом соответственно

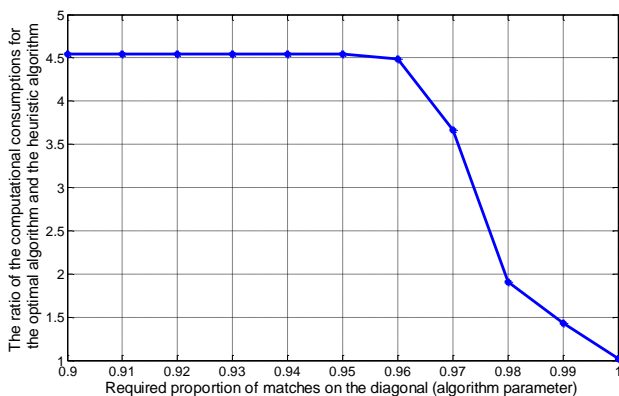


Рис. 3. Отношение требуемых вычислительных затрат для оптимального алгоритма Нидлмана-Вунша и эвристического алгоритма соответственно

Рассмотренный эвристический алгоритм позволяет сократить вычислительные затраты путем определения границ области поиска выравнивания последовательностей. Эффективность эвристического алгоритма зависит от характера входных данных (чем ближе строки в терминах редакционного расстояния, тем уже область поиска), от выбора параметра k – размера подстрок, для которых осуществляется поиск на точное совпадение по алгоритму КМП и строится точечная матрица, от выбора параметра $\text{алгоритмар} = \text{доли точек в точечной матрице, лежащих в пределах области поиска выравнивания}$ ($p = 0.9$). Перспективы дальнейшего развития подхода, применяющего усечение области поиска, связаны с выбором метода, позволяющего задавать нелинейные границы области поиска.

Библиографический список

1. *Гасфилд Д.* Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология / пер. с англ. И.В. Романовского. СПб.: Невский Диалект; БХВ-Петербург, 2003. 654 с.
2. *Needleman S. B., Wunsch C.D.* A general method applicable to the search for similarities in the amino acid sequence of two proteins // *Journal Molecular Biology*. 1970. – 48, P. 443-453.
3. *Mount D. W.* Bioinformatics: Sequence and Genome Analysis / D. W. Mount. 2nd ed. NY: Cold Spring Harbor Laboratory Press, 2004. 692 p.
4. *Gibbs A.J., McIntyre G.A.* The diagram, a method for comparing sequences. Its use with amino acid and nucleotide sequences // *Eur. J. Biochem*. 1970. -16, P. 1-11.
5. *Кормен Т.* и др. Алгоритмы: построение и анализ / 3-е изд.: пер. с англ. М.: ООО «И. Д. Вильямс», 2013. 1328 с.

УДК 004.052.2

М. А. Топорков

студент кафедры вычислительных систем и сетей

Н. А. Балонин

д-р техн. наук, профессор – научный руководитель.

ИЗУЧЕНИЕ ОРНАМЕНТАЛЬНОЙ МАТРИЦЫ $N=NT^T$ В ЗАДАЧАХ МАКСИМУМА ДЕТЕРМИНАНТА

В работе рассматриваются ортогональные и близкие к ортогональным матрицы с максимальным модулем детерминанта на множестве матриц с элементами, не большими по значению 1 [1, 2]. Отметим, что у квадратных матриц из ортогональности столбцов следует ортогональность строк (и наоборот) [3].

В статьях по целочисленным ортогональным или экстремальным по детерминанту матрицам символом J принято обозначать блок из 1, символом $I = \text{diag}(1, 1, \dots, 1)$ – единичную матрицу. У циклических матриц все строки образованы сдвигом верхней строки вправо, с подстановкой вытесняемых элементов слева. Помимо циклических и бициклических форм блоков и матриц, существует негациклическая форма [4, 5], с подстановкой вытесняемых элементов слева с инверсией их знака.

Определение 1. Матрица Адамара H – квадратная матрица порядка n с элементами $\{1, -1\}$ и ортогональными столбцами $H^T H = nI$, где I – единичная матрица.

Ввиду жестких ограничений на значения элементов, порядок матриц Адамара может быть равен только 1, 2 или $4t$, $t=1, 2, 3, \dots$ (целое число). Кроме того, известно, что определитель этих матриц максимален по абсолютной величине на классе матриц с элементами, не большими 1 по абсолютному значению [6, 7].

В рассматриваемой задаче ищутся две последовательности a, b с элементами 1 и -1 , определяющие первые строки циклических матриц A, B (моноциклов). Матрицы входят в состав бицикла

$$T = \begin{pmatrix} A & B \\ B^T & -A^T \end{pmatrix}, \text{ все элементы которого } 1 \text{ или } -1.$$

Определение 2. Матрица T называется *бициклом Томаса* в том случае, если взятая с нормализованной каймой $[T] = \begin{pmatrix} -1 & e^T \\ e & T \end{pmatrix}$, все элементы вектора e составляют 1, она имеет максимум детерминанта на множестве матриц такой структуры [8].

Бициклы Томаса с каймой из 1 нечетных порядков $3=2+1, 7=6+1, 11=10+1, \dots$ состоят из двух моноциклов и каймы из 1, за исключением первого элемента, равного -1 . Такая структура в сравнении ее с моноциклическим видом приведена на рис. 1.

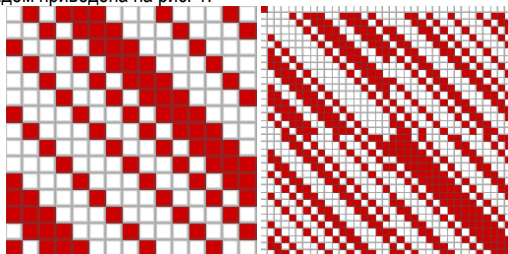


Рис. 1. Моноцикл и бицикл Томаса с каймой [T]

1. Бицикл Томаса и орнаментальная матрица

Оригинальные исследования ориентированы на поиск бициклов без каймы T^* четного порядка n и с элементами $\{0, 1\}$. Блоки A и B состоят из циклически смещаемых первых строк a и b каждого блока (орнамент «косой дождь»).

Матрица T^* без каймы порядка n с элементами 0 и 1, вместо 1 и -1 , отличается детерминантом от связанного с ней бицикла Томаса с каймой нечетного порядка $n+1$ ровно в 2^n раз. Это обстоятельство нередко используется – ищут именно ее.

В отчетах Томаса [8], соответственно, указываются две строки бицикла, бинарные элементы обозначают символами «+» и «-». Например, для порядка 34 имеем решение +-----+---- и -----+----+-. Мы используем несколько иную визуализацию матриц в виде портретов, на которых отрицательный элемент изображается цветным или темным квадратом, а положительный – белым.

Определение 3. Бицикл $N=T^*T$ будем называть *орнаментальной матрицей*, поскольку нас будет интересовать его узор.

На порядках, кратных 4, максимальные по детерминанту матрицы N (бициклы и любые иные орнаментальные структуры) ортогональны. Первая строка матрицы $N^*N=nl$ (последовательность) всегда имеет вид $(n, 0, 0, \dots, 0)$, задача ее определения тривиальна. Для матриц максимума детерминанта и локального максимума детерминанта [1, 2] это не так, инварианты симметрии матрицы N (также бицикла, если T бицикл) представляют собой первостепенный объект изучения [9].

2. Свойства орнаментальной матрицы

Бицикл Томаса T без каймы ищется на порядках $4t+2$ – он максимален по детерминанту на классе бициклических структур только с каймой. Поэтому сам по себе он не ортогонален, нулю равны только внедиагональные блоки бицикла. Изучение первой строки блочно-циклической матрицы N показало, что она похожа на строку единичной матрицы: на первом месте стоит n , ненулевые элементы диагональных блоков заменяется значением -2 . Среди элементов, равных -2 , заметны *альтернансы*: это элементы того же значения, но с противоположным знаком. Это определяет «полосатый» вид матриц N , которую назовем орнаментальной (рис. 2).

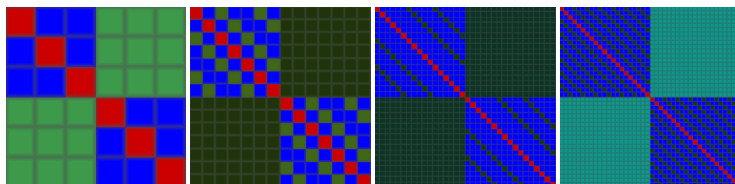


Рис. 2. Орнаментальные матрицы порядков 6, 14, 34, 42

Диагональные блоки равны A^*A+B^*B по способу вычисления T^*T . Как видно, можно вычислять только его, что сокращает вычисления вдвое. При численном моделировании орнаментальных матриц выяснились следующие ее свойства.

На порядках с 6 по 10 в матрице N , помимо n и 0, встречаются только элементы -2 , полосок на портрете орнаментальной матрицы порядка 6 (рис. 2) нет.

На порядках с 14 по 34 встречается две 2 (парный альтернанс). На порядках с 42 по 98 встречается шесть элементов со значением 2, это инвариант матриц Томаса в широком диапазоне ее стартовых порядков. Выше порядка 100 встречается 12 полосок.

3. Алгоритм поиска бицикла и результаты

Для поиска бициклов Томаса привлекаются специализированные программные комплексы и математические сети [10, 11], используемые для изучения свойств бициклических матриц [12, 13]. Алгоритм поиска бицикла Томаса состоит в следующем: генератором случайных чисел составляется таблица по 100 000 вариантов последовательностей a, b для каждой из сторон матрицы. При перекрестном сравнении избираются только те последовательности, при которых равенство Aa и $-$

Vb удовлетворяет определенному уравнению, где A и B – сами матрицы, a и b – их первые строки, рассматриваемые в произведениях, для простоты, как соответствующие им по размеру векторы. Если результат согласуется с условием, матрицы являются ортогональными.

Для ортогональных матриц сумма Aa и $-Bb$ порождает первый столбец единичной матрицы вида, т.е. последовательность 1, 0, 0, ... 0. Программная реализация алгоритма ищет не только эти, но и приближенные варианты, а именно варианты с не совсем идеальной диагональю бицикла $N=TT$.

Детерминант матрицы Томаса с каймой связан с детерминантом N , более того, бицикл N с желаемыми элементами 2, -2 и l на диагонали можно искать независимо. Вместо бицикла N мы ищем вложенный в него блок $A^T A + B^T B$ вдвое меньшего порядка. Тем самым, задача определения орнаментальной матрицы и значения оптимального детерминанта матрицы Томаса заметно проще нахождения ее блоков A , B .

Перебор элементов не двух блоков A , B , а только субблока бицикла N , дает нам заметное ускорение и надежность вычислений, позволившее установить выше описанные свойства – как количество альтернансов зависит от порядка матрицы.

Перебирая задаваемое количество полос альтернанса в N , ищем решение с максимумом детерминанта. Описанный подход позволяет проверить результаты Томаса не находя A и B . Повысить качество найденных матриц – их симметрию, а также улучшить доказательность материала. Максимальный порядок, который поддается вычислению вплоть до блоков искомой матрицы на типовой компьютере равен 86. Не имея таких вычислительных мощностей для нахождения матриц более высокого порядка, мы имеем, тем не менее, эффективный метод проверки через поиск орнаментальной матрицы. Установлено, что для порядков выше 100 результаты работы [8] могут быть теоретически улучшены, так как расчет показывает лучший детерминант.

Благодаря описанному методу, найдено множество редких симметричных матриц, бициклов Томаса, не указанных в его работе, порядков 6, 10, 14, 18, 22, 26, идущих через 4. Начиная с 26 порядка, исключая порядок 42 (заметим, что размер плеча $42/2=21$ неприводимой к симметрии матрицы – не простое число и не степень простого числа), симметричные матрицы идут с шагом 8: 26, 34, нет 42, 50, 58, 66. На порядках 74 и 82 симметричных матриц не найдено.

Предположение. Максимальный порядок симметричных бициклов Томаса равен 68.

Программы поиска на языке интерпретатора PHP оказались неэффективными даже в серверном исполнении, быстрее работают программы на javascript, для которого верхняя граница находимых порядков достигает 70. Программа, компилируемая на языке Pascal использует таблицу a , b -последовательностей до миллиона записей и поднимает уровень находимых порядков до 86. Орнаментальная матрица используется в ней как критерий остановки: если достигнут нужный орнамент, то поиск завершен. Ее роль велика в формировании четкого критерия завершения поиска.

Найденная матрица порядка 70 с максимальным среди бициклических структур детерминантом и орнаментальная для нее матрица представлены на рис. 3.

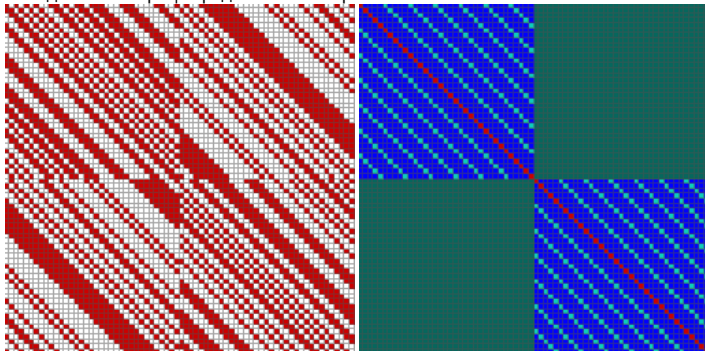


Рис. 3. Бицикл Томаса порядка 70 и орнамент $N=TT$

С помощью комплекса на javascript результат искался и проверялся несколько дней подряд. Это оказался бицикл с каймой с детерминантом $3,017 \times 10^{65}$, отвечающей матрице из 0 и 1 с детерминантом $2,555 \times 10^{44}$. Соревновательный результат Томаса был меньше в 1.61 раз, экстремальный результат получен им позднее с привлечением большего времени или более высокопроизводительной техники. Порядок 70 уникален тем, что все полоски идут на равном расстоянии. То же самое касается порядка 42, на котором происходит некоторое перестроение. Это важные для исследования порядки, следующий порядок такого типа спрогнозирован на 136, но он до сих пор не найден.

Работа выполнена при поддержке Минобрнауки РФ при проведении научно-исследовательской работы в рамках проектной части государственного задания в сфере научной деятельности по заданию № 2.2200.2017/ПЧ.

Библиографический список

1. Балонин Н. А., Сергеев М. Б. Матрицы локального максимума детерминанта // Информационно-управляющие системы. 2014. № 1(68). С. 2–15.
2. Balonin N. A., Seberry, Jennifer. Remarks on extremal and maximum determinant matrices with moduli of real entries ≤ 1 // Informatsionno-upravliaiushchie sistemy, 2014, № 5 (71), P. 2–4.
3. Балонин Н. А., Сергеев М. Б. Матрицы Мерсенна и Адамара // Информационно-управляющие системы. 2016. № 1(80). С. 2–15.
4. Balonin N. A., Djocovich D. Z. Negaperiodic Golay pairs and Hadamard matrices. // Informatsionno-upravliaiushchie sistemy, 2015, № 5, P. 2–17. doi:10.15217/issn1684-8853.2015.5.2
5. Балонин Ю. Н., Егорова И. С., Сергеев А. М. Негациклические матрицы и фильтры Мерсенна // Вестник компьютерных и информационных технологий. 2016. № 11 (149). С. 20–24.
6. Hadamard J. Resolution d'une question relative aux determinants. Bulletin des Sciences Mathematiques. 1893. Vol. 17. pp. 240–246.
7. Shalom E. La conjecture de Hadamard (I) – Images des Mathématiques, CNRS. 2012. <http://images.math.cnrs.fr/La-conjecture-de-Hadamard-I.html> (accessed: 15.03.2017).
8. Rokicki T. New records for maximal determinants, based on pairs of circulant matrices <http://tomas.rokicki.com/newrec.html> (accessed: 15.03.2017).
9. Балонин Н. А., Джоквич Д. Ж. Симметрия двуциклических матриц Адамара и периодические пары Голя // Информационно-управляющие системы. 2015. № 3.
10. Балонин Ю.Н., Сергеев М.Б. Алгоритм и программа поиска и исследования М-матриц // Научно-технический вестник информационных технологий, механики и оптики. 2013. № 3 (85). С. 82–86.
11. Балонин Н.А., Марлей В.Е., Сергеев М.Б. Новые возможности математической сети для коллективных исследований и моделирования в Интернете // Информационно-управляющие системы. 2014. № 3 (70). С. 40–46.
12. Балонин Н.А., Сергеев М.Б. Двуциклическая М-матрица 22-го порядка // Информационно-управляющие системы. 2014. № 2 (69). С. 109–111.
13. Балонин Н.А., Сергеев М.Б. Взвешенная конференц-матрица, обобщающая матрицу Белевича на 22-м порядке // Информационно-управляющие системы. 2013. № 5 (66). С. 97–98.

УДК 621.397.43:004.032

Д.А. Тропин, Н.А. Уваров

магистры кафедры информатика и вычислительная техника

А.В. Гордеев

д-р техн. наук, профессор – научный руководитель

СИСТЕМА ОХРАННОГО ВИДЕОНАБЛЮДЕНИЯ FUSIONCAM

Современные системы охранного видеонаблюдения – это комплекс взаимодействующего между собой оборудования, выполняющий фиксирование, записывание, анализирование и хранение информации, поступающей от видеокамер, различных устройств контроля доступа, охранных датчиков, и соответственно, в таких системах предусмотрено «принятие решения» по защите объекта. Системы устанавливаются для охраны периметра вокруг жилого объекта, охраны парковок и торговых площадей, для регистрации ДТП, охраны общественного порядка, слежения за непредвиденными ситуациями на производстве и т.п. Охрана путем внедрения камер является бурно развивающимся направлением. Существуют множество вариантов построения систем охранного видеонаблюдения, основным элементом, которых являются, конечно же, камеры. На рынке существует огромный выбор различных типов камер. В зависимости от задачи, которая должна решать система, выбираются то или иное дополнительное оборудование, например, если у Вас установлены аналоговые камеры, то чаще всего к такой системе устанавливается видеоманитофоны с выходами на мониторы для слежения оператором за объектами. С аналоговыми камерами очень много проблем, одна из самых важных связана с масштабируемостью системы. Так, если Вы, например, захотите заменить устаревшие или вышедшие камеры на более современные, необходимо заменять видеорегистратор, а зачастую на рынке просто не оказывается камер устаревшего типа. Системы видеонаблюдения бывают проводными и беспроводными. Сейчас особое внимание уделяется беспроводным, так как это бы решило проблему с прокладкой кабеля, например на большие расстояния, на таких объектах, как заповедник, устранила бы ряд проблем связанных с монтажными работами и экзотичностью объекта. В таких системах широко распространено использование усиливающих сигнал антенн. Передача сигнала осуществляется по общедоступной технологии Wi-Fi. Но с этой технологией тоже пока все не просто, так не решена проблема с безопасностью и шифрованием передаваемых данных, а так же улучшением качества сигнала, например, как сделать так, чтобы сигнал, проходящий через препятствие, будь то стена или дерево, не терял своих свойств, и не было потери важной информации. Сегодня любой желающий может собственноручно установить и наладить свою систему наблюдения, руководствуясь примерами из интернета, или помощью продавцов из технических магазинов. Существуют даже уже собранные готовые системы, которые требуют лишь, чтобы их распаковали и установили.

Предлагаемая к разработке система видеонаблюдения «FusionCam» – это уникальная система, не имеющая аналогов на рынке, за счет интеграции всех программно-аппаратных составляющих в одном устройстве. Идея реализации заключается в том, что в одном устройстве объединяются сервер, камера, накопитель, средства мониторинга. Дело в том, что развертывание современных систем видеонаблюдения требуют решения многих проблем, связанных с большими затратами ресурсов и времени. Например, разработка проекта, обязательное наличие специалиста, некое количество инженеров и техников, установки и настройки необходимого оборудования, программного сопровождения, решения проблем с питанием. Все эти задачи способна решить FusionCam, так как есть возможность сконфигурировать систему под любые нужды заказчика. Преимущество системы в существенном сокращении времени установки и настройки, так как есть возможность заранее спроектировать и собрать систему, а затем приехать к заказчику и подключить ее. Есть аппаратная поддержка как PoE, так и питания от стандартного адаптера 48 В. Интегрировано подключение к сети интернет по технологии Wi-Fi и через кабель. Такая система занимает очень малое пространство, ее можно интегрировать так, что она будет являться частью интерьера, заметно повышая приоритеты скрытности и безопасности. Устраня-

ется проблема порчи стен и потолков из-за отсутствия развернутых монтажных работ. Сокращается необходимость в дополнительных специалистах «в поле» так, как FusionCam легко установить в нужных местах и настроить, ведь сборка осуществляется до приезда на место установки. Устраняется проблема с масштабируемостью. Отказоустойчивость реализуется за счет технологии хранения данных аналогичных технологии RAID-массив. Благодаря технологии сжатия и одновременной доступности данных на всех камерах, минимизируется нагрузка трафиком сетевых устройств. Для просмотра видеоинформации достаточно подключится к web-интерфейсу, что дает возможность системе быть кроссплатформенной.

В зависимости от типа объекта и решаемых задач, заказчику предлагаются разные пути построения системы охранного видеонаблюдения. Основными задачами являются видеофиксация объекта, видеодетекция движения, обработка видеоизображения, хранение видеоматериалов, структурирование видеоматериалов, пересылка данных в компетентные органы безопасности, удаленный доступ к данным.

Аналоговые системы были первыми на рынке видеонаблюдения, но сейчас их можно смело отнести к прошедшему этапу. Такие системы обычно состоят из оптических приборов с ПЗС матрицей, образующей видеосигнал из светового потока. Запись видеоматериалов производится на видеомангитонфон с выводами на монитор, что в свою очередь, затрудняет просмотр конкретного события так, как запись ведется непрерывно. Значительным минусом системы является проблемы расширения ее возможностей, неспособность поддерживать более одного видеоканала, и необходимость в постоянном обслуживании, например смена кассеты, чистки и замены видеоголовок в магнитофоне [2].

Далее появились гибридные и комбинированные системы. В них используется цифровая запись, есть возможность взаимодействия аналоговых и сетевых камер. Запись с камер фиксируется с помощью более современных видеорегистраторов, а в гибридной системе у видеорегистраторов, есть интерфейс, для соединения с компьютерной сетью. Наличие «интеллекта» в видеорегистраторах, заметно облегчает процесс видеонаблюдения. В свою очередь, в гибридных системах для просмотра видео используется специальное программное обеспечение[1].

Сетевые системы широко распространены на рынке из-за использования IP-камер, которые имеют свой собственный IP-адрес и встроенное программное обеспечение. Это позволяет им функционировать как автономные сетевые устройства. Они обладают высоким качеством картинки. Такие системы ведут запись на жесткий диск, обеспечивают быстрый поиск нужных данных в архиве, доступ к записям, используя интернет, а так же возможность просмотра данных на отдельных камерах. Питание осуществляется по технологии PoE. Быстро развиваются системы, в которых данные пишутся локально и на отдельные виртуальные сервера. Главной особенностью такой системы является то, что можно удаленно просмотреть видео с камер, отдельно или сразу со всех. Аренда выделенного сервера обходится значительно дешевле аренды сопоставимого по мощности физического сервера. Простая масштабируемость – как замена отдельной камеры, при выходе из строя, так и добавление мощности виртуальному серверу. Но система имеет ряд минусов, например, общая дисковая система и канал связи. Так же, при потере интернет соединения, данные записаться не будут.

В системе FusionCam предусмотрены и учтены аспекты, связанные с потерей данных, проблемы с установкой виртуального сервера, используются особые методы хранения и безопасности данных. Устанавливать такую систему просто. Она не требует существенных затрат на дополнительное оборудование и программное обеспечение. Так, просмотр видео в реальном времени, может осуществляться с любого смартфона или планшета, имеющие доступ в интернет. Также есть возможность просматривать данные на обычных домашних телевизорах, благодаря ориентированной на такие устройства программной системе автоматического управления и контроля. Видеоданные обрабатываются и сжимаются непосредственно на устройстве. Данные дублируются на внешний сервер, но при необходимости доступны и на внутреннем хранилище. Дублирование позволяет снизить затраты ресурсов связанных с мощностью устройства и его габаритов. Нет необходимости докупать и устанавливать дорогостоящие видеорегистраторы, способные справляться с огромным количеством подключаемых камер. Устройства в системе FusionCam способны принимать, обрабатывать и пересылать данные на любые устройства вывода видеоинформации. Такая система существенно экономит занимаемое

пространство. Отсутствует необходимость выделения отдельного помещения с особыми климатическими условиями, использование дополнительной мебели, блоков питания, розеток, кондиционеров. Решаются проблемы из-за огромного количества проводов, которые мешают иногда добраться до нужного устройства, если оно вышло из строя. FusionCam полностью мобильна, достаточно отключить кабель и открутить пару болтов. Устройства имеют удобную конструкцию с регулируемым держателем. Система является универсальной, у нее имеется возможность конфигурирования «как хочешь», что заметно повышает ее конкурентоспособность на рынке существующих систем видеонаблюдения. Из-за взаимозаменяемости, простоте установки элементов устройства ее можно подстраивать под любые задачи, предварительно собрав ее на месте, после того, как будет доступна информация об объекте и пожеланий заказчика. Все необходимые элементы способные решать поставленные задачи, сконфигурированы и собраны в одном устройстве. Расширять систему, а также менять возможности системы можно до бесконечности.

Как и было описано выше система FusionCam будет универсальной из-за взаимозаменяемости своих составляющих. Нами предполагается использовать главным элементом устройства системную плату со встроенными интерфейсами, интегрированным процессором и видеокартой. Плата обеспечивает соединение всех встраиваемых элементов в одно устройство. Интерфейсы SATAIII для подключения 3-х дисковых накопителей SSD. Жесткие диски с помощью программного обеспечения объединяются по технологии аналогичной технологии RAID-массив, что обеспечивает возможность восстановить данные при выходе из строя одного из дисков. Нами так же предполагается использовать процессор Intel Core i7 с поддержкой DDR4. Процессор способен загрузить любую топовую видеокарту. Видеокарта на плате устанавливается в разъем PCI-E 3.0. Такой набор комплектующих способен давать достаточно ресурсной мощности для обработки видеоданных с высоким качеством. Программируемое асинхронное устройство предварительной подкачки данных с отдельными каналами считывания, позволяет скрыть задержки от доступа к памяти и полнее использовать АЛУ. Интерфейс для подключения внешнего питания на 48 В. Сетевой интерфейс поддерживает гигабитный режим работы, с поддержкой PoE. Встроенная сетевая плата. Вся конструкция будет оснащена особой системой охлаждения по технологии складного радиатора. Такие радиаторы разрабатывает Nasa для космических спутников. Радиатор складывается и раскладывается по принципу оригами, он способен контролировать уровень теплоотдачи, регулируя глубину складок: чем она выше, тем больше тепла будет поглощать прибор. Мостом для подключения разных типов видеокамер нами будет использоваться мезонинная плата, имеющая аналоговые и дискретные входы/выходы, она так же расширяет память и увеличивает количество коммуникационных портов. В зависимости от типовых возможностей камеры выбирается мезонинная плата с нужными характеристиками. Записи с камер можно хранить, как на встроенном накопителе, так и на внешней системе хранения данных с поддержкой интерфейса iSCSI, подключающаяся по технологии Ethernet. Через интерфейс Usb или PCI подключаются различные типы датчиков – инфракрасные, датчики движения, вскрытия и т.п. Встроенный Wi-fi-чип.

Для успешной работы всех устройств системы FusionCam необходима разработка и установка следующего программного обеспечения. Программное обеспечение для хранения и обработки видеоданных (сервер). ПО для возможности хранения информации для работы отказоустойчивости. Программный рейд-массив на одной камере. Встроенный web-интерфейс для отображения и настройки всех web-камер. ПО для поиска новых видеокамер внутри сети. ПО для работы внешнего сетевого хранения данных (СХД). Приложение с настраиваемыми функциями управления движением камеры, включением записи, распознаванием лиц, номеров машин и т.п. ПО автоматического добавления камер. Операционная система (Linux). Поддержка технологии WakeonLan (удаленно включение камеры). ПО сжатия трафика. ПО настройки сетевого подключения. Программная поддержка интеграции с облаком. Так же для большей безопасности на жестких дисках есть отдельно зашифрованный каталог, в который пишутся данные локально при падении сети. Связь между удаленным сервером и устройствами FusionCam настраивается по протоколу IPv6, что обеспечивает высокую безопасность транслируемых данных, высокую скорость, особенно хорошо для режима «реального времени».

На данный момент существует множество проблем, мешающих реализации проекта FusionCam, начиная от ценовой политики и заканчивая технологическими возможностями. Рассмотрим

проблемы реализации. Из-за сложности создания отдельных деталей, состоящих из редких сплавов и металлов, повышается цена комплектующих. Отдельные детали имеются только в виде прототипов или находятся на начальном этапе развития. Примером прототипа можно рассматривать Оригами-радиатор с памятью формы, разрабатываемый NASA. Устройство способно регулировать уровень теплоотдачи по принципу оригами, регулируя глубину складок: чем выше температура, тем больше тепла будет поглощать прибор. Основным материалом радиатора служит оксид ванадия, который при температуре в 67°С переходит из полупроводникового в металлическое состояние, что повышает его теплоотдачу. На данный момент оригами-радиатор является прототипом, поэтому все его характеристики засекречены, и скорее всего для использования его в системе FusionCam придётся разрабатывать радиатор самим, используя только идею NASA.

В большинстве SSD-дисков используется NAND, или, как ее еще называют, flash-память. Подобные чипы применяются в обычных флеш-накопителях, только в SSD-дисках они надежнее и имеют большую скорость обработки информации благодаря наличию контроллера. Этот вид памяти является энергонезависимым и не потребует дополнительного питания для работы. Также этот вид памяти имеет большую скорость записи и чтения данных благодаря использованию чипов, применяемых при создании ОЗУ. Он требует постоянного подключения к электричеству, поэтому в SSD-дисках, использующих RAM, часто устанавливают дополнительные аккумуляторы на случай внезапного отключения питания. В сравнении с традиционным жесткими дисками (HDD), SSD-диски имеют меньший размер и вес, но в несколько раз (6–7) большую стоимость за гигабайт, и значительно меньшую износостойкость (ресурс записи). SSD-диски имеют низкую помехозащищенность операций чтения из ячеек памяти и наличие сбойных ячеек, что приводит к необходимости использования в контроллерах более сложных кодов исправления ошибок.

Производители стремятся к повышению качества получаемого видеоизображения, однако непревзойдённое качество «несжатого» изображения не может окупить высокую его стоимость. Большое количество лимитирующих факторов, таких как максимальный предел в разрешающей способности системы, равный 2 Мп и выше (разрешение 4к), максимальная дальность передачи сигнала, повышенные требования к производительности записывающих устройств и отсутствие единой стратегии развития у разработчиков формата, существенно повышает стоимость устройства и снижает качество универсальности устройства.

Большие компании имеют патент на производство нужных для системы FusionCam деталей, из-за этого нет возможности покупать более дешёвые комплектующие у более мелких фирм, что значительно повышает цену на производство системы FusionCam. Большие корпорации не дают выйти на рынок мелким компаниям по производству техники, которые имеют аналоги их продуктов, возможно более качественных. Также в нынешнее время частному лицу сложно найти инвесторов для своих идей, что также не помогает в развитии нужных технологий. Одна из основных проблем FusionCam – это размер устройства. На данный момент производство некоторых комплектующих FusionCam невозможно из-за их маленького размера. Существуют аналоги таких комплектующих, но они имеют размер значительно больше, чем нужно для разработанной системы FusionCam.

Выпуску системы FusionCamна рынок мешает ряд проблем связанных со сложностью производства деталей подходящего размера, либо затраты на создание таких комплектующих не окупаются. Однако за счет интеграции всех программно-аппаратных составляющих в одном устройстве, разных способов питания, малого размера устройства, повышенной отказоустойчивости, способности справляться с обработкой высококачественного видео большого разрешения, легкости масштабируемости и своей универсальности, система FusionCamстанет серьезным конкурентом существующим системам охранного видеонаблюдения.

Библиографический список

1. Фисенко Т., Гончаров К. IP система видеонаблюдения. СофтПресс, 2011. 20 с.
2. Кашкаров А. Системы видеонаблюдения: практикум / Изд-во: Феникс, 2010. 152 с.

УДК 339.182

И.Г. Хянина

магистрант кафедры компьютерных технологий и программной инженерии

М.В. Фаттахова

канд. физ.-мат. наук, доцент – научный руководитель

ИССЛЕДОВАНИЕ МОДЕЛЕЙ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ ЦЕПЯМИ ПОСТАВОК

Настоящий период развития рыночной экономики обозначен возрастанием интереса к логистике и управлению цепями поставок. Внешняя среда, в которой осуществляются логистические операции, постоянно претерпевает изменения под воздействием развития рынка и условий конкуренции. Для того чтобы своевременно и адекватно реагировать на эти изменения, любой компании нужна методология систематического планирования и проектирования логистической системы, позволяющая учитывать сложившиеся обстоятельства и оценивать возможные альтернативы ее развития. В проектной логистике активно развивается направление – проектирование цепи поставок.

Итак, цепь поставок – это совокупность потоков и соответствующих им кооперационных и координационных процессов между различными участниками цепи создания стоимости для удовлетворения требований потребителей в товарах и услугах [1].

Актуальность выбранной темы заключается в том, что управление цепями поставок становится эффективным инструментом повышения производительности компаний. Стремительное развитие рынка, ужесточение конкуренции, требование улучшения качества сервиса клиентов ставят перед компаниями новые задачи. Чтобы сохранить конкурентоспособность и усилить свои преимущества, современному предприятию необходимо оптимизировать все процессы создания стоимости – от поставки сырья до сервисного обслуживания конечного потребителя. Для решения этих задач руководство компаний и обращается к управлению цепями поставок.

В данной работе исследуется один из наиболее универсальных и часто встречающихся видов цепей поставок – многоуровневые цепи поставок с древовидной распределительной структурой (пример такой цепи изображен на рис. 1).

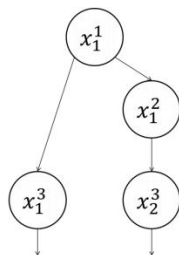


Рис. 1. Цепь поставок

Рассмотрим древовидный граф с множеством вершин (узлов) и множеством рёбер. Обозначать узлы (вершины) будем как x_i^j , где верхний индекс соответствует номеру уровня, на котором расположена эта вершина, а нижний индекс – порядковому номеру этой вершины.

Пусть каждая вершина цепи поставок состоит из конечной совокупности элементов. Эта совокупность элементов содержательно является группой конкурирующих фирм, производящих и потребляющих однородный продукт, а так же имеющих разные затраты на производство.

Возможны два подхода к решению задачи координации такой цепи. Первый подход – децентрализованное решение, получающееся в том случае, если все участники цепи поставок действуют

независимо друг от друга. И второй подход – оптимизация общей прибыли цепи в кооперативной игре, так называемое централизованное решение [2].

Рассмотрим первый подход. Опишем процедуру принятия решения в децентрализованной модели:

1. Корневой узел определяет цену, по которой он продаёт товар узлам своего сектора.
2. Вершины второго уровня цепи поставок, получая информацию от корневого узла, назначают цену товара вершинам своего сектора. Далее процедура повторяется до узлов предпоследнего уровня включительно.
3. Концевые вершины на основе цен, полученных от своих поставщиков, и функций спроса определяют объёмы выпуска товара на рынок.
4. Происходит процедура распределения объёмов между фирмами в каждой из вершин конечного уровня.
5. Информация об объёмах поступает на все верх лежащие уровни и внутри каждого узла происходит процедура распределения объёмов между фирмами.
6. Подсчёт прибыли каждого из участников цепи поставок.

Составим математическую модель. Каждая фирма из определённого узла имеет свои переменные управления – объём выпуска товара фирмы, цену и затраты на производство. Целевой функцией является функция прибыли, заданная следующим образом:

$$\pi_{ijk} = q_{ijk} (p_{ij} - p_{rn} - v_{ijk}), \quad \forall i = \overline{1, l}; j = \overline{1, m}; k = \overline{1, n_j},$$

где π_{ijk} – функция прибыли; q_{ijk} – переменный объём выпуска товара; p_{ij} – цена, по которой фирмы продают единицу производимого товара; p_{rn} – цена поставщика данного узла; v_{ijk} – затраты на производство.

В случае децентрализованного решения каждая фирма хочет максимизировать свою собственную прибыль так, чтобы затраты на производство, объём выпуска продукции и цена за единицу товара были не отрицательными.

При таком подходе каждая фирма из цепи поставок максимизирует собственную прибыль:

$$\pi_{ijk} \rightarrow \max, \quad \forall i = \overline{1, l}; j = \overline{1, m}; k = \overline{1, n_j}.$$

При ограничениях: $v_{ijk} \geq 0$; $q_{ijk} \geq 0$; $p_{ij} \geq 0, \forall i = \overline{1, l}; j = \overline{1, m}; k = \overline{1, n_j}$.

В результате решения общей модели получились следующие формулы:

1. Общий объём выпуска: $Q_{ij} = \sum_{k=1}^{n_j} q_{ijk}, \forall i = \overline{1, l}; j = \overline{1, m}; k = \overline{1, n_j}$.

2. Цена на товар: $p_{ij} = a_{ij} - b_{ij} Q_{ij}, \forall i = \overline{1, l}; j = \overline{1, m}; k = \overline{1, n_j}$.

3. Прибыль: $\pi_{ijk} = q_{ijk} (p_{ij} - p_{rn} - v_{ijk}), \forall i = \overline{1, l}; j = \overline{1, m}; k = \overline{1, n_j}$.

Этот метод был применён к реальным данным, результат – оптимальное решение, представленное в виде таблицы – вы можете увидеть на слайде.

Таблица 1

Значения переменных и прибыли

Объём выпуска	Цена	Прибыль участников
$q_{111} \approx 7923$	$p_{11} \approx 2648$	$\pi_{111} \approx 9092365$
$q_{112} \approx 7888$		$\pi_{112} \approx 9013310$
$q_{211} \approx 9375$	$p_{21} \approx 4613$	$\pi_{211} \approx 118664742$
$q_{311} \approx 1606$		$p_{31} \approx 3391$
$q_{312} \approx 1614$	$\pi_{312} \approx 651173$	
$q_{313} \approx 1622$	$\pi_{313} \approx 657644$	
$q_{314} \approx 1594$	$\pi_{314} \approx 635134$	
$q_{321} \approx 4699$	$p_{32} \approx 5156$	$\pi_{321} \approx 1987132$
$q_{322} \approx 4677$		$\pi_{322} \approx 1968381$

Таким образом, мною была изучена децентрализованная модель оптимального управления цепями поставок, эта модель проверена на реальных данных, оптимальное решение представлено в виде таблицы.

Библиографический список

1. *Бочкарев А. А.* Планирование и моделирование цепи поставок: учеб.-практ. пособие / М.: Издательство «Альфа-Пресс», 2008. 192 с.
2. *Bashinskaya A., Koroleva M., Zenkevich N.* «Coordinating contracts in cooperative supply networks»// Contributions to Game Theory and Management. Collected papers presented on the X International Conference «Game Theory and Management» / Editors Leon A. Petrosjan, Nikolay A. Zenkevich. – SPb, Graduate School of Management, SPbU. – 2016. – Vol IX.

УДК 004.896

И. А. Цветков

магистрант кафедры компьютерных технологий и программной инженерии

М. Ю. Охтилев

д-р. техн. наук, профессор – научный руководитель

БАЗА ЗНАНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ МОНИТОРИНГЕ СОСТОЯНИЙ КОСМИЧЕСКИХ СРЕДСТВ

В настоящее время, в связи с активным развитием техники, возрастает сложность эксплуатируемых технических объектов. К их числу можно отнести объекты, широко используемые в атомной энергетике, гидроэнергетике, в производственных и транспортных системах, а также в космической отрасли, где используются различные космические аппараты, ракеты космического назначения и другие средства, применяемые в целях освоения космоса. При эксплуатации этих объектов неизбежно возникает вопрос оценки их технического состояния. Задача контроля состояния технических объектов сводится к получению информации от объекта с помощью специальных каналов передачи данных, его обработкой, анализом, прогнозированием возможных штатных и нештатных ситуаций и принятием решения по результатам проведенных мероприятий [1, 2, 3].

В традиционной схеме проведения мониторинга, представленной на рис. 1, информация, поступающая в ситуационные центры, оценивается вручную по показаниям огромного количества датчиков. Согласно этой схеме, персонал номеров расчетов получает измеряемую информацию от объекта анализа, передают ее начальникам расчета, а те, в свою очередь, лицу, принимающему решение для окончательного принятия решения. При таком способе мониторинга не исключены ошибки операторов и несвоевременное принятие решений [3, 4, 5, 6].

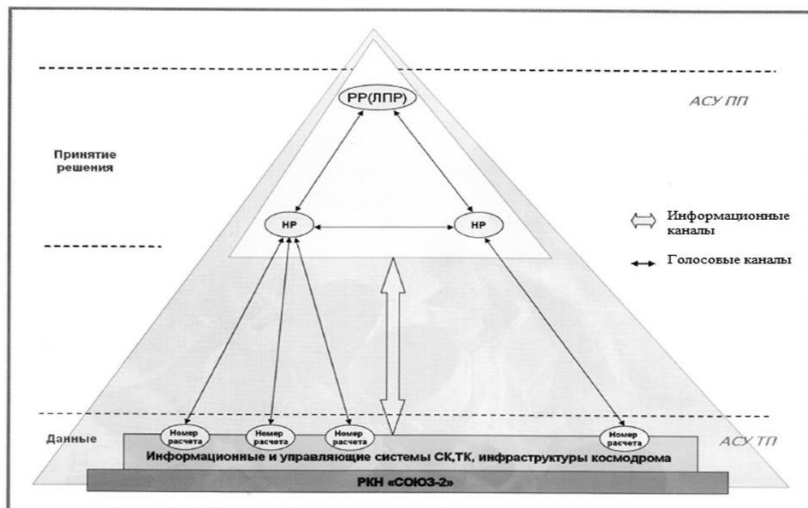


Рис. 1. Традиционная схема мониторинга состояний космических средств

Для решения этой проблемы существуют различные направления, одним из которых являются экспертные системы, основанные на базе знаний (БЗ). В БЗ экспертных систем описывается логика

функционировании объекта анализа. На основе данных из БЗ формируются схемы анализа – алгоритмы обработки информации. Получаемая от объекта контроля информация обрабатывается схемами анализа, и состояние объекта выводится на экранную форму с помощью различной индикации [3, 4, 5].

Одним из примеров таких систем является комплекс программного обеспечения автоматизированной системы анализа (КПО АСА). Разработка базы знаний в КПО АСА происходит по следующему алгоритму [7]:

- создание классов однотипных объектов, описывающих логику функционирования объекта анализа;
- создание базы знаний, детализация и создание иерархической структуры объекта анализа в соответствии с его структурой;
- создание экземпляров объектов определенного класса или объектов БЗ непосредственно описывающих логику анализа;
- задание исходных данных для объектов, привязка соответствующих атрибутов диагностики.

Окно редактора БЗ, который позволяет выполнять все описанные операции, представлено на рис. 2.

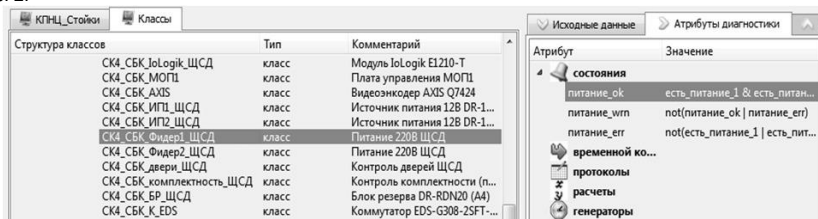


Рис. 2. Окно редактора базы знаний об объектах анализа

Целью моей работы является доработка и адаптация базы знаний комплекса программного обеспечения автоматизированной системы анализа состояний сложных технических объектов.

По результатам проделанной работы, среди наиболее значимых изменений, можно выделить следующие:

1. Реализация множественного наследования с использованием типа «интерфейс», разработка которого повлекла за собой множественные изменения по всему программному проекту, в том числе изменения архитектуры самого приложения, архитектуры формата хранения баз знаний и др.

Интерфейс – это новый тип данных, который отличается от типа «класс» тем, что в нем указывается только перечень атрибутов диагностики и исходных данных без указания конкретных значений или выражений.

Второе ключевое отличие интерфейса от класса заключается в том, что создавая новый элемент можно произвести наследование от нескольких интерфейсов. В этом случае все атрибуты диагностики и исходные данные будут объединены.

2. Реализация системы поддержки старых версий БЗ, задача разработки которой вытекает из разработки системы множественного наследования. Поддержка версий необходима при переходе на новые версии формата хранения баз знаний. В случае перехода все базы знаний, разработанные на старой версии приложения, должны корректно читаться и видоизменяться с учетом требований новой версии приложения.

3. Усовершенствование процесса разработки БЗ в редакторе БЗ:

- предоставление пользователю инструментов для более быстрой и удобной разработки БЗ, таких как отмена и возврат действий, поиск по дереву классов и состояний, быстрое добавление состояний из дерева и прочие. Данные возможности позволяют сделать процесс разработки БЗ более интерактивным, предостеречь пользователя от возможных ошибок, которые он может совершить во время разработки.
- более тщательный контроль над действиями пользователя с целью недопущения ошибок и сбоев

- более полная и информативная система оповещения пользователей о процессе разработки и последствиях совершенных им манипуляций.

Также в результате проделанной работы было произведено большое количество менее значимых изменений, таких как изменения существующего функционала, исправление некоторых исключительных ситуаций, возникавших во время разработки, доработка архитектуры приложения и доработка архитектуры БЗ приложения.

Подводя итоги проделанной работы можно сказать, что в рамках данной работы было выполнено:

- Обзор и анализ подходов к построению интеллектуальных систем поддержки принятия решений
- Обзор и анализ теоретических основ построения экспертных систем
- Обзор и анализ существующих моделей представления знаний
- Разработка и внедрение как глобальных, так и менее значительных функций редактора БЗ
- Изменение и адаптация существующих возможности редактора

Библиографический список

1. *Майданович О.В., Охтилев М.Ю., Соколов Б.В., Юсупов Р.М.* Комплексная автоматизация мониторинга состояний космических средств на основе интеллектуальной информационной технологии // Приложение к журналу Информационные технологии. 2011. № 10. С. 2-32.
2. *Зеленцов В.А, Ковалев А.П., Охтилев М.Ю., Соколов Б.В., Юсупов Р.М.* Методология создания и применения интеллектуальных информационных технологий наземно-космического мониторинга сложных объектов // Труды СПИИРАН. № 5. С. 7-81. 2013.
3. *Охтилев М.Ю., Соколов Б.В., Юсупов Р.М.* Интеллектуальные технологии мониторинга и управления структурной динамикой сложных технических объектов. М.: Наука, 2006. 410 с.
4. *Ахметов Р.Н., Васильев И.Е., Капитонов В.А., Охтилев М.Ю., Соколов Б.В.* Концепция создания и применения перспективной АСУ подготовкой и пуском ракеты космического назначения Союз-2: новые подходы к интеграции, интеллектуализации, управлению // Журнал Авиакосмическое приборостроение. 2015. № 4. С. 3-54.
5. *Майданович О.В., Каргин В.А., Охтилев М.Ю., Соколов Б.В.* Теория и практика построения автоматизированных систем мониторинга технического состояния космических средств: монография, Санкт-Петербург: Военно-космическая академия имени А.Ф. Можайского, 2011. 219 с.
7. *Комплекс программного обеспечения автоматизированной системы анализа при проведении испытаний. Руководство оператора.* 2013. 149 с.

УДК 519.246.85

Д. А. Чуев

студент кафедры компьютерных технологий и программной инженерии

В. Ю. Гамов

канд. воен. наук, доцент – научный руководитель

РАЗРАБОТКА МОДЕЛИ ПРОГНОЗИРОВАНИЯ КОЛИЧЕСТВА ПУСКОВ РАКЕТ-НОСИТЕЛЕЙ СРЕДНЕГО КЛАССА

Целью работы является прогнозирование максимально возможного количества пусков ракет-носителей среднего класса с космодрома «Плесецк». Данное исследование необходимо для того, чтобы узнать возможности космодрома, предсказать сколько запусков сможет сделать космодром исходя из ресурсов. Спрогнозировав количество запусков, на эксплуатирующих организациях космодрома специалисты, размещенные на штатных должностях в структурах формирования запусков космических аппаратов, будут знать, сколько ракет-носителей среднего класса они могут запускать.

Разработка модели прогнозирования производилась с помощью программы STATISTICA, которая предназначена для статистического анализа и обработки данных. Данная программа нашла свое применение в космонавтике, медицине, экономике и т.д. Она имеет в наличии более десяти тысяч вычислительных процедур, среди которых имеется большой выбор самых современных инструментов для моделирования и прогнозирования, включая возможность автоматического выбора оптимальной модели.

Основной целью анализа временного ряда является определение природы ряда и прогнозирование, заключающееся в построении модели с помощью которой можно предсказывать последующие значения временного ряда основываясь на имеющихся измеренных данных [5].

Основными стадиями анализа временного ряда являются [1]:

- 1) графическое отображение и описание поведения временного ряда;
- 2) выделение и удаление зависящих от времени закономерных составляющих временного ряда, а также низкочастотных и/или высокочастотных составляющих процесса;
- 3) построение модели для описания случайной составляющей и проверка ее адекватности;
- 4) прогнозирование последующих значений временного ряда, основываясь на разработанной модели.

За пятидесятилетний период выявлено, что процесс запусков космических аппаратов зависит не только от заранее запланированных Правительством и Министерством обороны космических пусков, а еще и от случайных факторов (например, таких как внеплановое прекращение функционирования космических аппаратов, переоснащение орбитальной группировки перспективными космическими аппаратами, наращивание военной угрозы и т.д.) [4]. Исходя из этого, появилась потребность в оценке способности комплектования космическими аппаратами ракет-носителей среднего класса специалистами, размещенными на штатных должностях в структурах формирования запусков космических аппаратов, и прогнозировании их оптимальных структур, для предотвращения рискованных ситуаций, связанных с отсутствием возможности комплектования ракет-носителей среднего класса космическими аппаратами, из-за чего может последовать срыв боевой задачи по процессу подготовки и запуска космических аппаратов.

На рис. 1 построен график всех пусков ракет-носителей среднего класса, проведенных на космодроме «Плесецк» с 1966 года по 2016 год, который является временным рядом.

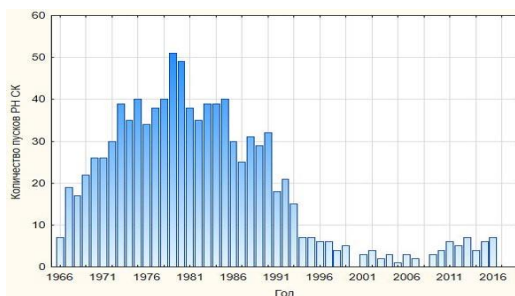


Рис. 1. График всех пусков ракет-носителей среднего класса

Из построенного временного ряда видно, что ряд характеризуется убывающей тенденцией и отсутствием сезонности. Также стоит учесть, что на промежутке 1992-1994 годов происходило воздействие.

На рис. 2 построены гистограммы с шагом 3 и 5, где на оси абсцисс откладывалось количество пусков РН СК, а по оси ординат количество повторений этого количества пусков с 1966 года по 2016 год.

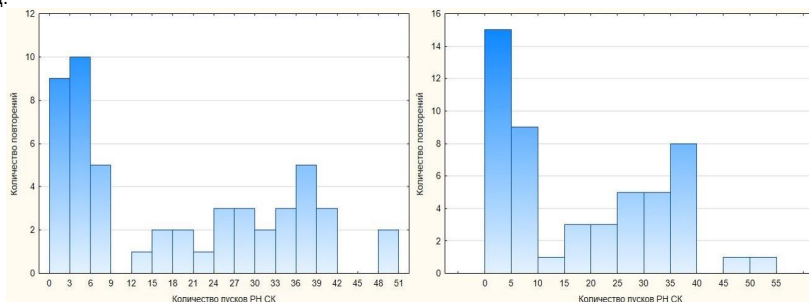


Рис. 2. Гистограммы с шагом 3 и 5 соответственно

На гистограммах выявлена «двугорбость» распределения количества повторений запусков. Наибольшие пики гистограмм находятся вблизи 0-6 и 35-40 запусков. Из этого следует, что временной ряд представляет случайный процесс, на который было проведено воздействие, в результате чего у процесса изменился параметр положения (т.е. количество повторений) или математическое ожидание, а также параметр масштаба (т.е. мера рассеивания количества повторений) или дисперсия [2]. Следовательно, временной ряд рассматривается как два случайных процесса. Шаги 3 и 5 показали, что были выбраны все изменения, которые происходили на протяжении пятидесяти лет.

Для построения модели прогнозирования временной ряд приводится к стационарному виду, после чего, производится моделирование с использованием класса моделей типа АРПСС (авторегрессионное проинтегрированное скользящее среднее).

Методология АРПСС является очень популярной и практика показала ее гибкость и мощность. Модель АРПСС включает три типа параметров: параметры авторегрессии (p), порядок разности (d) и параметры скользящего среднего (q). К примеру, модель АРПСС (1, 0, 1) содержит 1 параметр авторегрессии и 1 параметр скользящего среднего, который вычисляется для ряда после взятия разности с лагом 0.

При моделировании значения параметров АРПСС не превышало 2, т.к. дальнейшее увеличение параметров АРПСС приводит к снижению точности описываемого временного ряда, что значительно скажется на результатах прогнозирования.

Произведя сравнение разработанных моделей, была выбрана адекватная модель прогнозирования, которую удалось получить взятием разности с лагом единица от исходного временного ряда и выполнив моделирование с использованием модели АРГСС (0, 1, 1). Для анализа остатков полученной модели на рис. 3 представлены автокорреляционная функция и нормальный график распределения остатков.

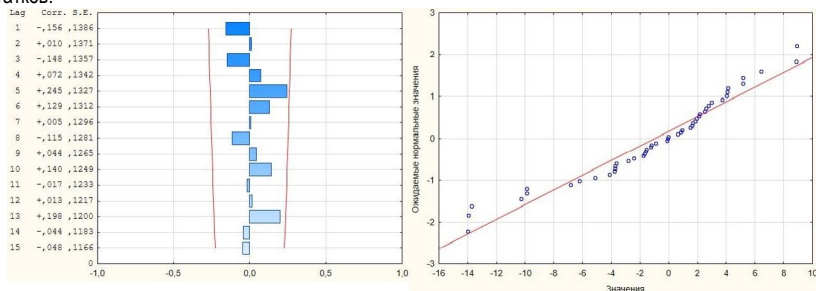


Рис. 3. Автокорреляционная функция и нормальный график распределения остатков

Проанализировав полученные графики видно, что на автокорреляционной функции и на нормальном графике распределения остатков остатки распределены равномерно [3]. Из этого следует, что полученная модель прогнозирования подходит для построения прогноза на следующий год.

В результате моделирования получены прогнозные значения в диапазоне от -10 до 11. Т.к. отрицательных значений количества пусков быть не может, возьмем диапазон значений от 0 до 11 и выполним прогноз количества пусков ракет-носителей среднего класса на 2017 год, который представлен на рис. 4.

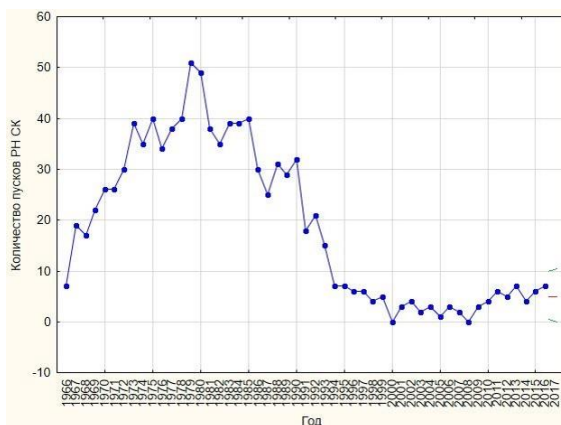


Рис. 4. Прогноз количества пусков ракет-носителей среднего класса на 2017 год

В результате прогноза, можно сказать, что количество пусков ракет-носителей среднего класса в 2017 году будет от 0 до 11, но вероятнее всего пусков будет 5.

Для проверки достоверности прогноза взят временной ряд за период от 1966 года по 1995, 2000, 2005, 2010, 2015 года и выполнен прогноз на следующий год. В ходе выполнения прогноза количества пусков ракет-носителей среднего класса на предыдущих годах, все прогнозы оказались успешными. Из этого следует, что разработанная модель прогнозирования корректна и прогноз на 2017 год можно считать достоверным.

В результате проделанной работы, можно сказать, что разработанная модель подобрана правильно и прогноз на 2017 год достоверный. С космодрома «Плесецк» в 2017 году максимум возможно будет выполнено до 11 пусков ракет-носителей среднего класса, но скорее всего пусков будет 5.

Созданная методика построения и прогнозирования количества пусков ракет-носителей среднего класса позволит выработать предложения и рекомендации командирам и штабам формирования запусков космических аппаратов.

Библиографический список

4. Айвазян, С. А., Мхитарян В. С. Прикладная статистика и основы эконометрики: учебник для вузов. – М.: ЮНИТИ, 1998. – 1022 с.
5. Гамов В. Ю. Автоматизированные системы научных исследований: учеб. пособие. – СПб.: ГУАП, 2015. – 96 с.
6. Гамов В. Ю. Компьютерное моделирование: Методические указания к выполнению контрольной работы: учеб. пособие. – СПб.: ГУАП, 2014. – 39 с.
7. Гамов В. Ю. Теоретические основы прогнозирования оптимальных структур, обладающих гибкостью, формирований войск запуска космических аппаратов // Горный информационно-аналитический бюллетень (научно-технический журнал). 2009. Т. 2. № 12. С. 115-125.
8. Колемаев, В. А., Староверов О. В., Турундаевский В. Б. Теория вероятностей и математическая статистика: учеб. пособие для экон. спец. вузов / под ред. В. А. Колемаева. – М.: Высш. шк., 1991. – 400 с.

УДК 629.735.33

И.О. Шальнее

магистрант кафедры компьютерных технологий и программной инженерии

С.В. Щекин

канд. техн. наук, доцент – научный руководитель

ИССЛЕДОВАНИЕ ПРОТОКОЛА ОБЪЕКТНОГО ВЗАИМОДЕЙСТВИЯ ВИРТУАЛЬНЫХ СИСТЕМ

Идея протокола заключается в управлении программы на стороне клиента сервером. Если разработчик захотел изменить пользовательский интерфейс или бизнес логику, то он, изменяя программу не должен обременять пользователя обновлением, клиент лишь генерирует пользовательский интерфейс в соответствии с измененной логикой. В исследовании и реализации этого протокола и состоит задача. Объектное взаимодействие подразумевает управление удаленными объектами. В процессе работы программы, работающей на этой технологии, будет формироваться скрипт, который в дальнейшем будет передан на клиентскую сторону. На клиентской стороне этот скрипт будет обработан в виртуальной стек-ориентированной машине, сделав то, что было написано в программе. Скрипт представляет собой псевдо-ассемблерный язык, состоящих из инструкций и присущих ему аргументов.

В терминальных системах мы привыкли, что именно сервер выполняет те или иные задачи удаленно. В этой же модели именно клиентская часть выполняет эту роль сервера. Сервер говорит клиенту что надо сделать, а клиент это выполняет, и сервер не должен объяснять, как это сделать. Подобно как человек делает заказ в ресторане, а исполнитель это делает, не задавая вопросов как это сделать. Таким образом:

- реализовывать программу можно исходя из того, что присутствует в «меню» клиента;
- вся бизнес-логика находится на сервере;
- сервер не должен знать методы, которыми оперирует клиент.

Объект одного типа представляется как совокупность объектов серверной и клиентской части. Например, при создании объекта типа EServerWidget на серверной части создается класс EClientWidget на клиентской. На стороне клиентского объекта находится вся функциональная часть отрисовки объекта, а также события взаимодействия с пользователем, в то время как серверный объект занимается созданием скрипта и обработкой запросов клиента.

```
EServerWidget::EServerWidget(Connection *connection) :
EServerObject(connection)
{
    PacketAssembler packet(mConnection);
    packet.addCommand(DECL);
    mConnection->declVariable(this);
    packet.addCommand(this->getId());
    packet.addCommand(E_Widget);
}
```

Рис. 1. Листинг объекта EServerWidget

На рис. 1 приведен псевдокод конструктора объекта EServerWidget. В нем формируется скрипт команды объявления объекта. DECL это инструкция ассемблера, который принимает на вход ID и тип создаваемого объекта. В дальнейшем этот скрипт будет отправлен на сторону клиента, где классом VClientProcessor будет выполнена эта инструкция (рис. 2). Все объекты наследованы от класса EClientObject, поэтому была создана фабрика классов, которая по идентификатору типа объекта создаст нужный экземпляр класса и поместит его в хэш, в котором находятся все созданные объекты (Диаграмма классов приведена на рис. 3).

```

void VClientProcessor::execCommand()
{
    ...
    case DECL:
    {
        int TempId=mProgram.at(++mIP).toInt();
        EClientObject *Object=EObjectFactory::createObject(mProgram.at(++mIP));
        mVariables.insert(TempId, Object);
        mIP++;
        break;
    }
}
    
```

Рис. 2. Листинг метода объекта VClientProcessor

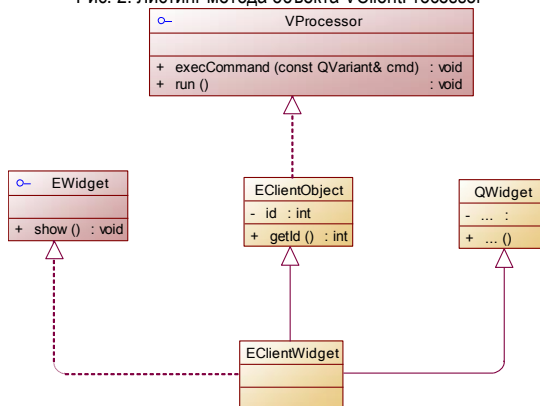


Рис. 3. Иерархия классов

VProcessor это класс, который разбирает скрипты. Аргументы в стек-ориентированных виртуальных машинах находятся на вершине стека. То есть инструкция суммирования SUM возьмет два аргумента из вершунки и положит результат туда же.

```

void EServerWidget::showWidget()
{
    PacketAssembler packet(mConnection);
    packet.addCommand(PUSH);
    packet.addCommand(WIDGET_SHOW);
    packet.addCommand(PULL);
    packet.addCommand(this->getId());
    packet.addCommand(EXEC_COMMAND);
}

void VClientProcessor::execCommand()
{
    ...
    case DECL:
    {
        int TempId=mProgram.at(++mIP).toInt();
        EClientObject *Object=EObjectFactory::createObject(mProgram.at(++mIP));
        mVariables.insert(TempId, Object);
        mIP++;
        break;
    }
}
    
```

Рис. 4. Листинг объекта EServerWidget и метода объекта VClientProcessor

Во время вызова метода какого-нибудь объекта на стороне сервера также формируется скрипт. На рис. 4 приведен пример вызова удаленной процедуры объекта EClientWidget. Здесь также формируется скрипт, который впоследствии отправится на сторону клиента. За вызов удаленной процедуры отвечает инструкция EXEC_COMMAND. В качестве аргументов передается id объекта и идентификатор вызываемой функции. Все объекты клиентские объекты наследуют интерфейс VProcessor с виртуальными функциями virtual void execCommand(const QVariant& cmd)=0; и virtual void run()=0; (Диаграмма классов приведена на рис. 3). Таким образом, когда VClientProcessor встретит инструкцию

EXEC_COMMAND, он вызовет перегруженную функцию нужного объекта, id которого передан в качестве первого аргумента.

Программист, разрабатываемый программу на этой технологии, должен быть максимально абстрагирован от реализации и не должен быть загружен дополнительными знаниями о тонкостях создания сетевого приложения. Он должен писать сетевое приложение как самую простую desktop программу. То есть создав объект, написав «Button submit=new Button();», он создаст кнопку на клиентской части.

В заключение хотелось бы отметить, что технология будет кроссплатформенной, так как разрабатывается на основе библиотеки Qt, но основным ее преимуществом является то, что она позволяет объединить множество приложений в одно единственное, что соответствует нынешним тенденциям рынка. Ведь вся бизнес-логика находится на сервере, основной графический функционал на клиенте и это, во-первых, дает возможность не нагружать цифровое устройство пользователя большим объемом информации, а, во-вторых, позволяет расширять систему различными приложениями, магазинами, социальными сетями и всем, что придет в голову.

УДК 629.735.33

В. А. Шведчиков

магистрант кафедры компьютерных технологий и программной инженерии

М. В. Фаттахова

канд. физ.-мат. наук, доцент – научный руководитель

ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА С ИСПОЛЬЗОВАНИЕМ СЕРВИСА GOOGLE ДЛЯ ЗАДАЧ CVRP

Транспортная логистика немыслима без активного использования информационных технологий. Трудно себе представить формирование и организацию работы цепей доставки товаров без интенсивного оперативного обмена информацией между участниками транспортного процесса, без возможностей быстрого реагирования на потребности рынка транспортных услуг.

Все более важным становится обеспечение непрерывности управляемых процессов в узловых точках, где осуществляется прохождение грузов между сетями различных транспортных агентов и тем самым там, где осуществляется прохождение информации между различными сетями.

Существующий на данный момент в компании «ЭнергоЛогистик» порядок учета, структурирования и хранения информации о заказчиках, «ручное» выстраивание логистики при выполнении заявок снижают конкурентоспособность фирмы из-за больших временных затрат на решение этих задач. В связи с тем, что обработкой контактной информации и реквизитов компаний-партнеров занимаются люди, то стоит взять во внимание и человеческий фактор. Кроме человеческого фактора стоит отметить то, что вся база клиентов ведется на бумажных носителях, что соответственно осложняет процедуру поиска, редактирования и хранения документов, задача распределения транспортных средств, выполняющих заявки потребителей, решается «по наитию»: весь цикл решения транспортной задачи занимает до 3 часов.

Задачи маршрутизации транспорта (ЗМТ) представляют собой огромный класс задач, включающий в себя подклассы, каждый из которых предназначен для более полного представления деталей действительности. Переход от ЗМТ общего вида к конкретному подклассу осуществляется с помощью включения дополнительных ограничений в условие задачи.

К условиям, формирующим разновидности ЗМТ, относятся следующие ограничения: ограничения по грузоподъемности автомобиля, ограничения на количество клиентов в каждом маршруте, условие существования нескольких общих вершин (нескольких складов), а также многие другие ограничения и их возможные комбинации. Каждое дополнительное ограничение влияет на постановку задачи и приводит ее в соответствие с требованиями и условиями работы конкретной компании. В рамках настоящей работы исследуется задача маршрутизации транспорта с ограничением на количество транспортных средств, находящихся в распоряжении компании, с учетом их грузоподъемности (CVRP).

Транспортная компания «ООО ЭнергоЛогистик» осуществляет доставку топлива с базы хранения (депо) по объектам в Санкт-Петербурге и Ленинградской области. Пример составленной матрицы расстояний между клиентами была получена с помощью онлайн сервиса Google и представлена в табл. 1.

Таблица 1

Матрица расстояний между клиентами компании (м)

	№	0	1	2	3	4	5
Депо	0	0	15173	15873	16369	10655	6729
Дорога на Турухтанские острова, 17	1	15173	0	13842	22478	16054	8918
Васильевский о., Бол. проспект	2	15873	13842	0	3824	6600	6918
Спецмост, Уральская ул., 33	3	16369	22478	3824	0	7134	16545
Ул. Оренбургская 44	4	10655	16054	6600	7134	0	7791
Ул. Заозерная	5	6729	8918	6918	16545	7791	0

Потребности каждого клиента указаны в табл. 2.

В распоряжении компании находятся 4 транспортных средства, информация о которых содержится в табл. 3.

Таблица 2

Объемы заказанного груза городами-клиентами

Объекты	0	1	2	3	4	5	Всего
Заказы (л)	0	5000	1000	4000	4500	1000	15500

Таблица 3

Грузоподъемности транспортных средств компании

Транспортное средство	Грузоподъемность (л)
Транспортное средство №1	5000
Транспортное средство №2	30000
Транспортное средство №3	29670
Транспортное средство №4	29480
Общая грузоподъемность	94150

Целью компании является оптимизация способа доставки заказанного груза каждому клиенту с помощью эффективного подхода к загрузке транспортных средств и формированию маршрутов их следования, приводящего к сокращению транспортных издержек.

Для удобства восприятия данной задачи перейдем от названий адресов-клиентов к их номерам, причем нумерацию проведем согласно следованию адресов в алфавитном порядке, за нулевую вершину примем морской порт.

Как видно по матрице расстояний, рассматриваемая задача является симметричной, т.е. выполняется равенство длины пути из точки А в точку Б и наоборот. Из соотношения величин суммарного заказанного груза и общей грузоподъемности транспортных средств можно сделать вывод о том, что поставленная задача не сбалансирована. Ввиду того, что грузоподъемность больше общего объема груза ($94150 \text{ л} > 15500 \text{ л}$), предназначенного для перевозки, используемые автомобили будут загружены не полностью. Для того чтобы получить наилучшее значение загрузки автомобилей и минимальное расстояние между клиентами, был применен генетический алгоритм.

Генетический алгоритм представляет собой способ решения задач оптимизации и моделирования путем имитации механизмов и процессов, наблюдаемых в ходе биологической эволюции. Главным «механизмом» генетического алгоритма является естественный отбор, который приводит некоторую популяцию к состоянию, наиболее приспособленному к выживанию.

Математическая модель генетического алгоритма для задач CVRP имеет следующий вид.

Вспомогательные переменные:

n_k – количество целевых вершин в маршруте транспортного средства, $k = \overline{1, m}$;

g_i – потребность товара -го клиента;

q – грузоподъемность транспортного средства, $k = \overline{1, m}$;

m – количество автомобилей;

k – номер автомобиля;

C_{ij} – расстояние между каждым клиентом;

x_{ijk} – поедет ли k -й автомобиль от клиента i к клиенту j ;

Целевая функция – суммарное расстояние, пройденное всеми транспортными средствами:

$$F = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^n c_{ij} x_{ijk} \quad (1)$$

Задача ЗМТ состоит в минимизации суммарного расстояния.

Вспомогательные ограничения:

$$x_{ijk} = \begin{cases} 1 - \text{Поедет ли автомобиль } k \text{ от клиента } i \text{ к клиенту } j; \\ 0 - \text{Иначе} \end{cases};$$

$$y_{ijk} = \begin{cases} 1 - \text{Доставил ли товар, автомобиль } k \text{ клиенту } j; \\ 0 - \text{Иначе} \end{cases}.$$

Ограничения:

- количество товара, требуемого покупателем, не превышает общую вместимость машины:

$$\sum_{i=1}^n g_i y_{ik} \leq q, k = 1, 2, \dots, m \quad (2)$$

- доставка каждому клиенту должна осуществляться только одним автомобилем m :

$$\sum_{k=1}^m y_{ik} = \begin{cases} m & i = 0 \\ 1 & i = 1, 2, \dots, n \end{cases} \quad (3)$$

- условие, что автомобиль m доставил груз и может двигаться к следующей вершине:

$$\sum_{j=0}^n x_{ijk} = y_{ik} \quad i, j = 1, 2, \dots, n; \quad k = 1, 2, \dots, m \quad (4)$$

Таблица 4

Маршруты следования транспортных средств, найденные с помощью генетического алгоритма

Транспортное средство	Грузоподъемность, кг	Маршрут	Загруженность, кг	Пройденное расстояние, м
Транспортное средство №1	30000	0-5-0	5000	30346
Транспортное средство №2	30000	0-2-5-4-3-0	10500	22624
Транспортное средство №3	-	-	-	-
Транспортное средство №4	-	-	-	-
Всего	60000	-	1500	52970

Полученное итоговое значение 52970 м представляет собой среднее из всех найденных результатов и является близким к оптимальному. Построение решения задачи маршрутизации транспортного потока в соответствии с генетическим алгоритмом происходит за 1,5-2 минуты, в то время как на составление маршрутов вручную уходит 30-40 минут.

Исходя из всего вышеизложенного, можно выделить основные особенности применения метаэвристического метода к решению задач маршрутизации.

Во-первых, метод осуществляет сокращение общего времени вычислений за счет использования настраиваемых управляющих параметров, что делает более гибкой систему принятия решений.

Во-вторых, качество получаемых решений напрямую зависит от начальных решений и пределов изменения контрольных параметров, выбор которых связан с проведением некоторых предварительных вычислений. Также к специфическим особенностям метаэвристик следует отнести тот факт, что в связи с наличием во многих из них процедур, основанных на случайном выборе каких-либо обрабатываемых элементов (например, скрещивание или мутация в генетическом алгоритме), каждый новый запуск соответствующего алгоритма при неизменных исходных данных может возвращать новое результирующее значение.

В ходе выполнения работы была спроектирована и разработана автоматизированная информационная система, обеспечивающая:

- оптимизацию транспортного потока компании на основе применения генетического алгоритма совместно с онлайн сервисом Google Maps;
- оперативный сбор, структурирование и хранения информации о клиентах компании и их заявках в электронном виде.

Библиографический список

1. *Zhou D.S., Li L., and Liu W. K.* A solution to the problem of logistics distribution algorithm, Jiamusi University (Natural Science), том. 3, Март. 2006, 417-419 с.
2. *Робинсон С., Корнес О., Глинн Д.* С# для профессионалов; М.: Лори, 2005. – 396 с.
3. *Taillard E.* Parallel interative search method for vehicle routing problems. Networks, т. 23. 1993. 661-673 с.

УДК 004.65

М.С. Шуин

студент кафедры проблемно-ориентированных вычислительных комплексов

Г.С. Евсеев

канд. техн. наук, доцент – научный руководитель

УПРАВЛЕНИЕ ЭСКАЛАЦИЕЙ БЛОКИРОВОК В MSSQL СЕРВЕРЕ

В данной статье будет рассмотрен такой процесс в MSSQL-сервере как эскалация блокировок. Эскалация блокировок это процесс преобразования нескольких более мелких блокировок в одну более крупную [1]. В MSSQL-сервере предусмотрено несколько уровней блокировок.

Блокировки на уровне базы данных. Такие блокировки автоматически накладываются на любую базу, к которой подключен пользователь. Основным их назначением является запрет некоторых действий, например удаление данных из баз, к которым подключены пользователи.

Блокировки на уровне объекта (например: таблицы). Такие блокировки накладываются на таблицы или индексы как при выполнении обычных запросов, так и при выполнении служебных операций с этими объектами.

Блокировки на уровне экстенда – такие блокировки можно очень редко встретить в базе, в основном они применяются при создании новых таблиц или увеличении размеров файлов базы данных.

Блокировки на уровне страницы – такие блокировки используются довольно часто, они блокируют только часть таблицы, страницу размером 8 кбайт, такой тип блокировок может применяться не только для страниц данных, но и для страниц индексов.

Блокировки на уровне записи/ключа – такие блокировки накладываются на отдельные записи [2].

В данном исследовании были использованы только три типа блокировок: блокировка таблицы, блокировка страницы и блокировка поля.

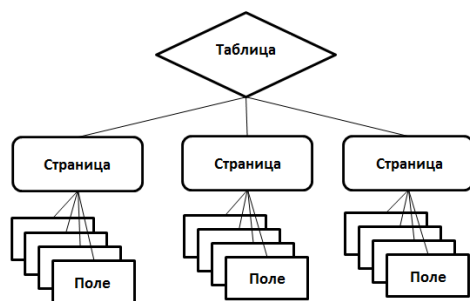


Рис. 1. Иерархия блокировок

Когда же происходит укрупнение блокировок? Существует две причины для этого. Во-первых, в том случае, когда запрос пытается применить к одному объекту более пяти тысяч блокировок на уровне записи. Второй случай связан с объемом памяти, затрачиваемой на выполнение запроса. На каждую блокировку сервер отводит 96 байт. Объем памяти, необходимый для записи блокировок, изначально равен 0; как только размер этой области начинает превышать 40% от общего объема оперативной памяти, происходит эскалация блокировок [3].

Цель данного исследования состоит в том, чтобы проверить, каким образом эскалация блокировок влияет на производительность сервера. Для этого был проведён следующий опыт, была создана таблица и заполнена данными, пример процедуры заполнения представлен на рис. 2.

```
CREATE PROCEDURE rand3 (@a INT)
AS
DECLARE @i int;
set @i = 0;
while (@i<@a) BEGIN
insert into t (v,filler) values (1 , 'aaa');
set @i = @i+1;
end;
GO
```

Рис. 2. Пример процедуры для заполнения таблицы

После заполнения таблицы, были составлены запросы на обновление данных с различными типами блокировок (*ROWLOCK*, *PAGLOCK*, *TABLOCK*) и последовательно отправлены к серверу. Для измерения времени выполнения каждого запроса была использована функция *datediff*. После получения данных о времени выполнения запросов с каждой из блокировок количество записей в таблице удваивалось таким образом, что объем таблицы был увеличен с пятисот до шестидесяти четырёх тысяч записей.

```
update dbo.t with (tablock) set v = 2 where v = 1 and filler = 'vvv' option (maxdop 1);
```

Рис. 3. Пример запроса, отправляемого к серверу

В ходе проведения данного эксперимента были получены следующие результаты, приведенные в таблице.

Результаты выполнения запросов

Число записей	<i>ROWLOCK</i>	<i>PAGLOCK</i>	<i>TABLOCK</i>
500	242	135	13
1000	302	176	15
2000	426	170	15
4000	647	216	24
8000	1192	300	76
16000	2319	371	229
32000	6360	951	406
64000	6509	1259	683

На основании полученных данных был построен график.

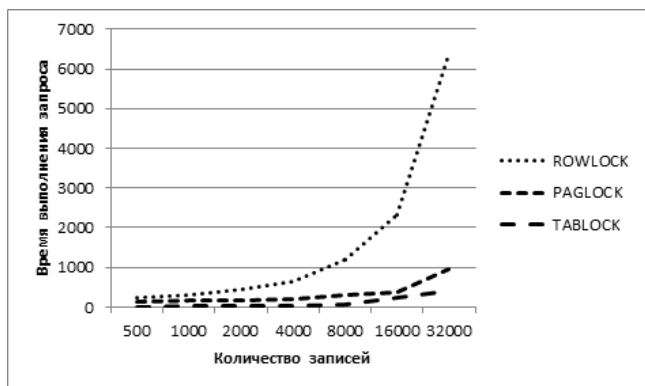


Рис. 3. График зависимости времени выполнения запроса от объема таблицы

По полученному графику можно сделать вывод, что, используя блокировки на уровне строки, мы увеличиваем время выполнения запроса за счет того, что каждый раз при модификации данных происходит запись информации в журнал блокировок о каждой из задействованных в запросе строк, что в свою очередь и увеличивает время выполнения запроса. В рамках развития данной темы предлагается рассмотреть параллельное выполнения двух и более запросов. Также представляет интерес анализ объема затрачиваемых аппаратных ресурсов в зависимости от изменения типа блокировки.

Библиографический список

1. Бондарь А.Г. Microsoft SQL Server 2014 / Отдельное издание.– СПб.: БХВ-Петербург, 2015.– 592 с.
2. Эскалация блокировок [Электронный ресурс], — Режим доступа: <http://www.gilev.ru/escalation/>, свободный. – Загл. с экрана.
3. Delaney K. SQL Server Concurrency: Locking, Blocking and Row Versioning / P.: Simple Talk, 2012. – 192 с.

УДК 004

М.П. Агеев, С.Ю. Гуков, А.В. Зайцева, Н.В. Миневцев

студенты кафедры информационно-сетевых технологий

О.И. Красильникова

канд. техн. наук, доцент – научный руководитель

ИДЕНТИФИКАЦИЯ ЛИЧНОСТИ ЧЕЛОВЕКА В БУДУЩЕМ

Введение

На протяжении всего времени своего существования человек хотел каким-то образом систематизировать определение личности. Мы сейчас живем в переходный период, то есть мы имеем технологии для идентификации человека, но не имеем единого решения. В этой статье как раз и пойдет речь о едином методе, но сперва мы кратко расскажем о основных существующих технологиях.

Обзор технологий

Идентификация по отпечаткам пальцев

Идентификация по отпечаткам пальцев – самая распространенная, надежная и эффективная биометрическая технология. Отпечатки всех пальцев у каждого человека уникальны по рисунку папиллярных линий и различаются даже у близнецов. Отпечатки пальцев не изменяются в течение всей жизни человека. Для получения сведений об отпечатках пальцев применяются специализированные сканеры. Известны три основных типа сканеров отпечатков пальцев:

- емкостные,
- прокатные,
- оптические.

Наиболее совершенную технологию идентификации по отпечаткам пальцев реализуют *оптические сканеры*. Они несколько дороже сканеров других типов, но лишены их многочисленных недостатков, долговечны и потому более практичны, удобны и просты в использовании.

Идентификация геометрии лица.

Идентификация геометрии лица – вторая по степени распространенности и популярности биометрическая технология. Эта технология применяется в качестве вспомогательной по отношению к другим биометрическим методам. К преимуществам геометрии лица, прежде всего, относятся бесконтактный способ получения сведений и широкий выбор источников этих сведений (фотографии, видеоряд, данные видеонаблюдения). При идентификации биометрическая система автоматически выделяет и обрабатывает сведения, характеризующие отдельные участки и особенности лица: контуры носа, губ, бровей, расстояние между ними и т.п. На основе этих сведений формируются цифровые модели идентификаторов, которые затем сравниваются между собой. К сожалению, технологии идентификации по лицу весьма чувствительны к внешним условиям и изменениям внешности. Это приводит к тому, что данные технологии характеризуются самым низким процентом успешного распознавания пользователей и самым высоким процентом ложных срабатываний.

Идентификация с помощью подкожного чипа

В начале 21 века компания ADS презентовала технологию «Цифровой ангел». Сутью изобретения был подкожный чип для человека, вживление которого осуществлялось с помощью специального шприца. При размере устройства с рисовое зернышко, функционал устройства был достаточен для решения поставленной задачи. Задача же ставилась следующая – определить местоположение человека. Чип является радиочастотным устройством, которое в автоматическом режиме отслеживается GPS-системами, и, соответственно, его носитель находится под наблюдением невидимого глаза в течение 24 часов. Стоит уделить особое внимание такой интересной возможности подкожного чипа, как способность питаться от энергии мышц. Он буквально «срастается» с мышечной тканью и его удаление является чрезвычайно проблематичным.

Идентификация радужной оболочки глаза

Данная технология биометрической аутентификации личности использует уникальность признаков и особенностей радужной оболочки человеческого глаза. Радужная оболочка образовывается ещё до рождения человека, и не меняется на протяжении всей жизни. Рис. радужки очень сложен, это позволяет отобразить порядка 200 точек, с помощью которых обеспечивается высокая степень надежности аутентификации. Для сравнения, лучшие системы идентификации по отпечаткам пальцев используют 60-70 точек.

Технология распознавания радужной оболочки глаза была разработана для того, чтобы свести на нет навязчивость сканирования сетчатки глаза, при котором используются инфракрасные лучи или яркий свет. Ученые также провели ряд исследований, которые показали, что сетчатка глаза человека может меняться со временем, в то время как радужная оболочка глаза остается неизменной. И самое главное, что невозможно найти два абсолютно идентичных рисунка радужной оболочки глаза, даже у близнецов. Для получения индивидуальной записи о радужной оболочке глаза черно-белая камера делает 30 записей в секунду. Одна из записей затем оцифровывается и сохраняется в базе данных зарегистрированных пользователей. Затем полученное изображение радужки преобразуется в упрощенную форму, записывается и хранится для последующего сравнения. Очки и контактные линзы, даже цветные, не воздействуют на качество

Новый метод идентификации человека

Каждая из описанных ранее технологий по-своему уникальна, но мы ставили перед собой задачу – идентифицировать человека в различных условиях. Исходя из нашей задачи было принято решение объединить две технологии, а именно идентификацию с помощью отпечатка пальца и чипа.

Идентификация с помощью отпечатка пальца

В нынешнем мире слишком много стало документов: паспорт гражданина, заграничный паспорт, документы на машину, на недвижимость и так далее. Мы предлагаем создать единую базу, в которой будет храниться вся информация о человеке. А чтобы проверить информацию нужно будет просто приложить палец к считывающему устройству, а уже устройство выдаст информацию о человеке.

Рассмотрим на примере. Вы двигаетесь на автомобиле, Вас останавливает сотрудник полиции, чтобы проверить документы. Сейчас Вам необходимо предоставить водительское удостоверение, документы на машину, страховку. Наша система требует просто приложить палец к устройству. После этого у инспектора будет вся информация о человеке. Таких примеров можно привести множество, но суть будет одна. Почему именно идентификация по отпечатку пальца? На данный момент никому не получится повторить рисунок отпечатка. Каждый человек отличается от другого именно отпечатком. Вероятность одинаковых отпечатков стремится к нулю. Сами устройства идентификации достаточно дешевые в производстве. В любом смартфоне от 100 долларов уже есть данный датчик, так что проблема внедрения, с точки зрения финансовой стороны, не составит труда.

Идентификация с помощью чипа

Для абсолютной идентификации человека недостаточно только отпечатка. На массовых мероприятиях, где необходимо узнать, кто находится в толпе, по отпечатку пальца это сделать практически невозможно.

Практически у любого мобильного телефона есть GPS-трекер, который позволяет определить геопозицию девайса. С человеком такой метод будет неэффективен, так как такой трекер любой злоумышленник может просто выкинуть. Мы предлагаем следующее: внедрить подкожный чип в тело человека.

Как будет работать определение геопозиции? Спутниковая система GPSи ГЛОНАСС уже давно трудятся на благо человека. Как раз эти спутниковые системы и будут определять местоположение человека.

Почему именно так мы предлагаем определить местоположение человека? Данные спутниковые системы уже показали себя с положительной стороны, а внедряемый чип в тело человека будет практически невозможно достать.

Заключение

Внедрение данного метода идентификации человека возможен примерно через 50-60 лет. Это связано с тем, что внедрение чипа в тело человека возможно только в детстве. Главным требованием этого метода будет мировое внедрение, все государства нашей планеты должны поддержать этот метод. Конечно, возникнут противники нашей идеи, они будут апеллировать правом человека на свободу. Но отсюда вытекает и главное преимущество, а именно снижение преступности во всем мире. Если гос. органы смогут идентифицировать человека, то многие преступники не смогут совершать свои деяния. Жизнь человека станет гораздо удобней, так как не будет необходимости носить с собой документы, стоять в очередях за очередной справкой и так далее.

Библиографический список

1. Google: способы идентификации личности, URL: <http://www.idexpert.ru/technology/119/> (Дата обращения: 25.04.17)
2. Google: цифровой ангелчип, URL: <http://www.membrana.ru/particle/2360> (Дата обращения: 26.04.17)
3. Google: способы идентификации личности человека, URL: <http://www.mirprognozov.ru/prognosis/society/sposobyi-identifikatsii-lichnosti-cheloveka/> (Дата обращения: 26.04.17)

УДК 004.932

А.А. Афанасьев

студент кафедры инфокоммуникационных систем.

М.Р. Гильмутдинов

канд. техн. наук, доцент – научный руководитель.

ИССЛЕДОВАНИЕ МЕТОДОВ СЖАТИЯ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТЕЙ С МАСШТАБИРОВАНИЕМ БИТОВЫХ ПОТОКОВ

Введение

Большинство современных систем характеризуются разным уровнем соединения и особенностями принимающих устройств. Неоднородное качество связи является результатом совместного использования ресурсов этих сетей, которые взаимодействуют с изменяющимся во времени потребностями в пропускной способности данных у различного количества пользователей. Различные разрешения устройств пользователей, а также качество уровня соединения между пользователем и сервером требует наименьших затрат при сжатии потокового видео контента. Сжатие видео с масштабированием битовых потоков (*Scalable Video Coding*) является одним из основных способов для уменьшения нагрузки в современных системах. Данный режим реализован в стандарте *H.264* в расширении *H.264/SVC*^{[1][2][3][6]} наряду с кодированием независимых одиночных потоков *H.264/AVC*^{[3][8][9]}.

Цель *H.264/SVC* состоит в том, чтобы обеспечить кодирование высококачественного видео, которое будет содержать один или несколько потоков битов. При декодировании части данных потоков битов можно получить исходный контент с разрешением и частотой кадров, которые удовлетворяют характеристикам пользователя. Качество декодированной видеопоследовательности должно быть аналогичным тому, что достигается в *H.264/AVC*. *SVC*, и успешно применяется в следующих элементах ИТ-индустрии^{[1][2]}:

- потоковая передача,
- системы IP-TV,
- конференц-связь,
- видеонаблюдение,
- широковещательная передача.

Данная статья посвящена сравнительному анализу эффективности методов сжатия *H.264/AVC* и *H.264/SVC* применительно к передаче в системах IP multicast и unicast.

Режимы масштабирования *H.264/SVC*

Принцип масштабируемого видеокодера заключается в том, чтобы разделить обычное однопоточное видео в многопоточное, состоящее из нескольких компонентов, называемых слоями. Многоуровневая структура любого масштабируемого видеоконтента может быть определена как комбинация из базового слоя и нескольких дополнительных слоев улучшения. Базовый слой соответствует самому низкому качеству видео, в то время как слои улучшения позволяют уточнить вышеупомянутый базовый слой.^[1]

Существуют следующие режимы масштабирования битовых потоков:

- временное масштабирование,
- пространственное масштабирование,
- масштабирование по качеству.

Пространственное и временное масштабирование используется в случаях, когда слои масштабируемого битового потока являются исходным содержимым, но с уменьшенным размером изображения (пространственным разрешением) или частотой кадров (временным разрешением) соответственно. Благодаря масштабируемости по качеству полученный подпоток обеспечивает такое же простран-

ственно-временное разрешение, как и полный поток битов, но с более низкой точностью, где под точностью понимают отношение сигнал-шум (*SNR*). Масштабирование по качеству также часто называют масштабированием точности или *SNR*-масштабированием. Различные типы масштабируемости можно комбинировать, чтобы можно было поддерживать множество с различными разрешениями и скоростями передачи в одном масштабируемом битовом потоке. [3]

Временная масштабируемость может быть эффективной при использовании концепции иерархических В-кадров. Кадры улучшенного слоя кодируются как В-кадры основываясь на предшествующем и последующем кадрах. Причем идентификатор временного слоя у опорных кадров ниже, чем у предсказываемого. Каждый набор временных слоев $\{T_0...T_k\}$ может быть декодирован независимо от всех уровней с использованием идентификатора временного слоя $T > k$.

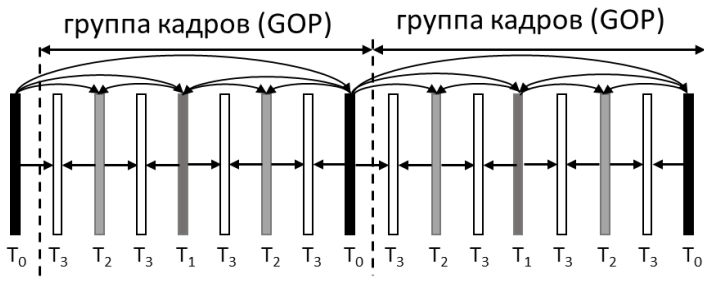


Рис. 1. Иерархическая структура предсказания в режиме временного масштабирования

В каждом пространственном слое используется предсказание с компенсацией движения и интра-предсказание, как для одноуровневого кодирования. Однако для улучшения эффективности кодирования по сравнению с одновременной передачей различных пространственных разрешений через *simulcasting* присутствует алгоритм межслоевого предсказания (*inter-layer prediction*). Чтобы ограничить требования к памяти и сложности декодера, в режиме *H.264/SVC* установлен один и тот же порядок кодирования для всех поддерживаемых пространственных слоев. Кадры с различными пространственными разрешениями для данного момента времени образуют блок доступа и должны передаваться последовательно в возрастающем порядке их соответствующих идентификаторов зависимости *D*. Как показано на рисунке ниже, кадры нижнего уровня не должны присутствовать во всех блоках доступа, что позволяет объединить временную и пространственную масштабируемость.

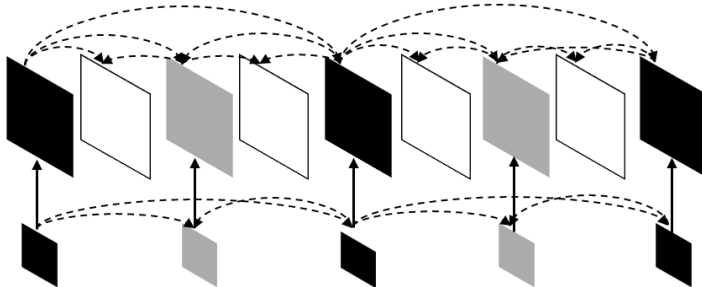


Рис. 2. Структура видеопоследовательности с межслойным предсказанием пространственного масштабирования

Оценка эффективности методов сжатия на базе режимов *H.264/AVC* и *H.264/SVC*

Использование режима *H.264/SVC* является одним из главных способов повышения степени сжатия в современных системах в режиме передачи *multicast* [2, 3, 5]. На сервере кодируется некая видеопоследовательность с её максимальным разрешением и частотой кадров. Абонентские устройства, которые не нуждаются, в частности, в *HD* качестве данной видеопоследовательности могут декодировать только часть потока с необходимым разрешением.

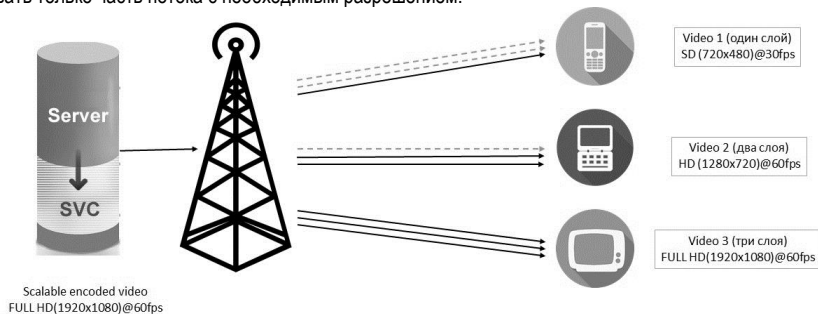


Рис. 3. Система передачи multicast закодированного видеопотока в режиме *H.264/SVC*

Для анализа преимущества метода *SVC* в режиме *multicast* [5] передачи был выбран видеокодер *JSVM (Joint Scalable Video Model)* [1], созданный и разрабатываемый *JVT (Joint Video Team)*. Программное обеспечение позволяет произвести моделирование, учитывая подбираемые настройки, которые повлияют на эффективность кодирования как в режиме *H.264/SVC*, так и в режиме *H.264/AVC*. В качестве исходных видеопоследовательностей были выбраны «Foreman», «Football», «Mobile» с разрешением *CIF (352x288)* и частотой кадров в 30 Гц.

Модель, относительно которой сравнивается преимущество одного метода над другим, представляет 3 набора различных устройств, каждый из которых поддерживает различное максимальное разрешение и частоту кадров у видеопоследовательности. Всем наборам передается в режиме *multicast* закодированная видеопоследовательность. Пусть у этих наборов следующие характеристики:

- 176x144 (15 Гц),
- 176x144 (30 Гц),
- 352x288 (30 Гц).

В случае *H.264/AVC* каждая видеопоследовательность с выше описанными характеристиками кодируется независимо и передается в режиме *unicast* [5]. В случае *H.264/SVC* закодированная видеопоследовательность имеет 3 слоя. Слои соответствуют тем же характеристикам, однако закодированы в единый битовый поток. Эффективность кодирования оценивается с помощью пикового отношения сигнал-шум (*PSNR*) от *bitrate*.

Суммарное значение *bitrate* для каждого из двух режимов представлено ниже. Небольшая разница в *PSNR* меньше 1 дБ показывает, что сильных потерь в качестве видеопоследовательности после её декодирования на приемной стороне не выявлено, однако в случае режима *H.264/AVC* суммарный *bitrate* гораздо выше, чем в случае *H.264/SVC* [9]. При использовании разных видеопоследовательностей разница в *bitrate* существенно изменяется. Это зависит от количества движения в видеопоследовательности. В среднем *bitrate* закодированной видеопоследовательности в режиме *H.264/SVC* ниже примерно в два раза случая с использованием *H.264/AVC*. Выигрыш для фильмов с малым движением *bitrate* выше также, как и *PSNR*.

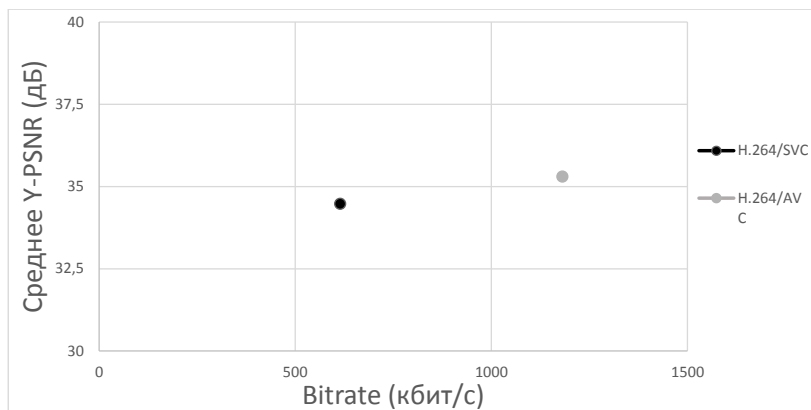


Рис. 4. Сравнение $PSNR$ от суммарного $bitrate$ в модели из трех наборов различных устройств на примере «Foreman»

$Bitrate$ в режиме $H.264/SVC$ выше, если сравнивать его с отдельно закодированной видеопоследовательностью в режиме $H.264/AVC$ максимального разрешения и частоты кадров. Однако немасштабируемый режим проигрывает в режиме *multicast*^[5] передачи с различными видеопоследовательностями.

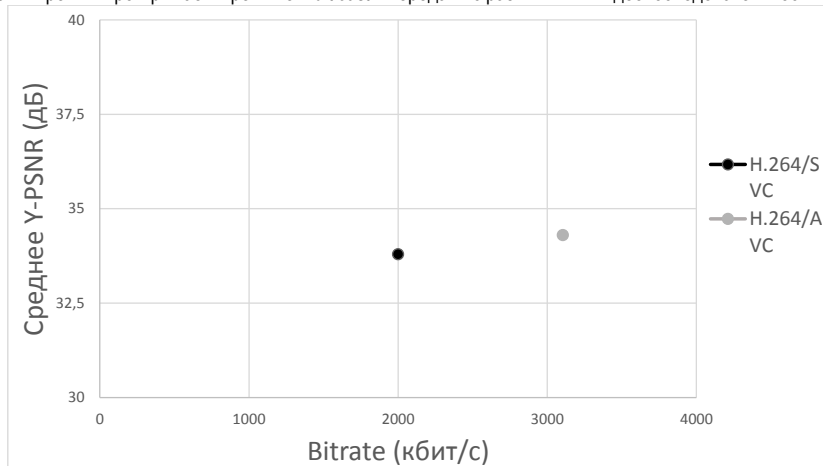


Рис. 5. Сравнение $PSNR$ от суммарного $bitrate$ в модели из трех наборов различных устройств на примере «Football»

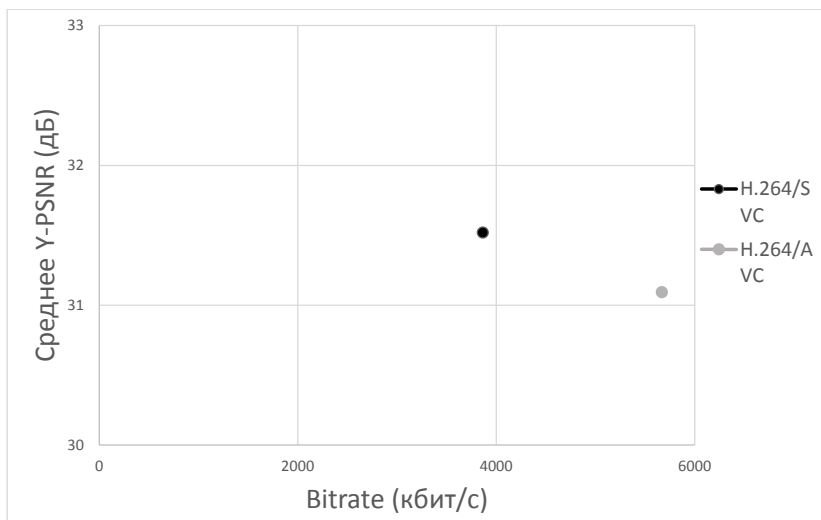


Рис. 6 – Сравнение PSNR от суммарного *bitrate* в модели из трех наборов различных устройств на примере «*Mobile*»

В заключение следует сказать, что кодирование видеопоследовательностей с масштабированием битовых потоков одно из решений проблем, созданных характеристиками современных систем передачи видео. Режим выигрывает у такого типа передачи как *unicast*^[5], но проигрывает, если количество разнообразных устройств существенно мало. Выше отображенные результаты количественно подтверждают эффективность применения SVC при групповом вещании.

Библиографический список

1. Schwarz H., Marpe D., Wiegand T. Overview of the Scalable Video Coding. //IEEE Transactions on circuits and systems for video technology, vol. 17, no. 9. – September, 2007. – P. 1103–1120,
2. Choupani R., Wong S., Tolun M. Scalable Video Coding: A Technical Report,
3. Wiegand T., Sullivan G. J., Bjontegaard G., Luthra A. Overview of the H.264/AVC Video Coding Standard. //IEEE Transactions on circuits and systems for video technology, vol. 13, no. 7. – July, 2003. – P. 560–576,
4. Гук И. Особенности сжатия видеоданных по рекомендации H.264 // Компоненты и технологии № 2. – 2006,
5. Unicast, broadcast, multicast, Ethernet MAC Address – <http://www.erg.abdn.ac.uk/users/Gorry/course/intro-pages/uni-b-mcast.html> – [Online; accessed 30-May-2017],
6. Unanue I., Urteaga I., Husemann R. A Tutorial on H.264/SVC Scalable Video Coding and its Tradeoff between Quality, Coding Efficiency and Performance. // Recent Advances on Video Coding. – June, 2011. – P. 1–28,
7. Doria T.H. The Great Codec Debate The Myths, Realities, and Considerations You Need to Know Before Making Your Next Videoconferencing Purchase. – 2013,
8. Richardson I.E. The H.264 Advanced Video Compression Standard. – 2010,
9. Wiegand T., Sullivan G. J. The H.264/AVC Video Coding Standard. // IEEE Signal Processing Magazine. – March, 2007.

УДК 004.056.53

А.П. Бандуровский

студент кафедры технологий защиты информации

К.А. Жиданов

ассистент – научный руководитель

ПРОБЛЕМЫ БЕЗОПАСНОСТИ СКУД

Система контроля и управления доступом (СКУД) – совокупность программно–аппаратных технических средств безопасности, имеющих целью ограничение и регистрацию входа–выхода объектов (людей, транспорта) на заданной территории через «точки прохода»: двери, ворота, КПП.

Основная задача – управление доступом на заданную территорию (кого пускать, в какое время и на какую территорию), включая также:

- ограничение доступа на заданную территорию
- идентификация лица, имеющего доступ на заданную территорию

Как правило, включает в себя контроллер, считыватель и замок. Также необходим блок питания [1].

Автономный контроллер – это «мозг» системы: именно контроллер определяет, пропустить или нет владельца идентификатора в дверь, поскольку хранит коды идентификаторов со списком прав доступа каждого из них в собственной энергонезависимой памяти. Когда человек предъявляет (подносит к считывающему устройству) идентификатор, считанный из него код сравнивается с хранящимся в базе, на основании чего принимается решение об открытии двери.

Сетевой контроллер объединяется в единую систему с другими контроллерами и компьютером для возможности централизованного контроля и управления. В таком случае решение о предоставлении доступа может приниматься как контроллером, так и программным обеспечением головного компьютера.

В случаях, когда необходимо обеспечить работу контроллера при авариях электросети, блок контроллера обеспечивается собственным аккумулятором, либо внешним блоком резервного питания. Время работы от аккумулятора может составлять от нескольких часов до нескольких суток.

Z–5R – автономный контроллер, обладающий всеми необходимыми функциями для создания системы контроля доступа на одну или несколько дверей. Используется в СКУД. Он предназначен для хранения базы данных разрешенных ключей и управления доступом к двери. Открывает замок для занесенных в базу ключей [2].

Контроллер рассчитан на подключение считывателя карт «по входу» и кнопки открывания двери «по выходу». Возможно подключение до 6 считывателей (параллельно) без возможности их различения.

Предусмотрено подключение датчика двери для корректного управления замком.

Считыватель «MATRIX– II Iron Logic» используется в системах контроля доступа в качестве считывателя бесконтактных карт PROXIMITY стандарта EM–Marine и HID и преобразования кода считанной карты в код Dallas Touch Memory и Wiegand 26. Считыватель позволяет в системах СКУД одновременно использование карточек двух стандартов EM–Marine и HID [3].

Ход исследования:

В ходе работы было выяснено, что контроллер доступа Z5R Iron Logic поддерживает подключение считывателя в системе СКУД по двум протоколам: 1Wire и Wiegand 26.

1–Wire – разработанная компанией Dallas Semiconductor система автоматизации. Это информационная сеть, использующая для осуществления цифровой связи одну линию данных и один возвратный (или земляной) провод. Ограничение максимальной длины однопроводной линии без специальных дополнительных вспомогательных устройств (повторителей) регламентировано на уровне 300м. Основой архитектуры сетей 1–Wire, является топология общей шины, когда каждое из устройств

подключено непосредственно к единой магистрали, без каскадных соединений или ветвлений. При этом используется структура сети с одним ведущим или мастером и многочисленными ведомыми.

Wiegand – простой проводной интерфейс связи между устройством чтения карточки (считывателем) и контроллером, предназначенный для передачи уникального кода карты.

Двоичный 26–битный формат карт, который характеризуется специфическим расположением двоичных данных (Wiegand 26 bit).

Схема подключения СКУД по протоколу 1–Wire приведена на рис. 1 [2]

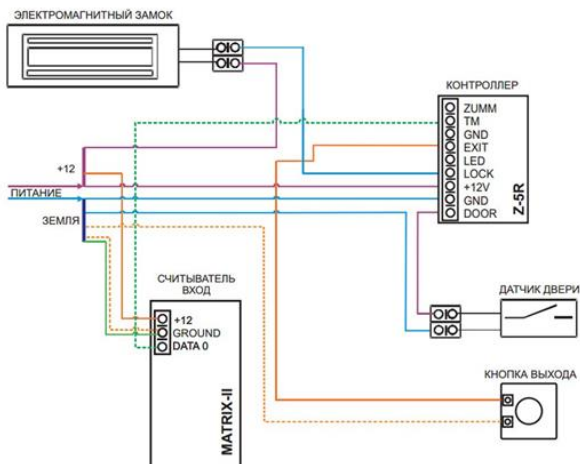


Рис. 1. Схема подключения СКУД по протоколу 1–Wire

Для подключения по Wiegand достаточно подключить порт DATA1 считывателя MATRIX-II к порту EXIT контроллера Z-5R.

Схема подключения контроллера по протоколу Wiegand (рис. 2) [2]:

Wiegand 26

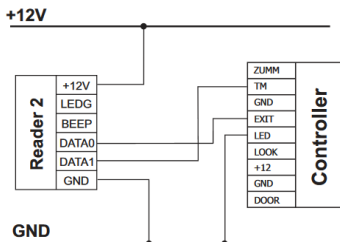


Рис. 2. Схема подключения контроллера по протоколу Wiegand

При передаче данных по протоколу Wiegand, используется 2 провода, подключенных в порты DATA0 и DATA1 на считывателе и в ТМ и EXIT на контроллере. Данные о считанной карте передаются со считывателя на контроллер по этим проводам согласно протоколу.

Кнопка выхода, хоть и является опциональной, имеет свой фиксированный и работающий в любом режиме контакт. Подключается кнопка следующим образом (рис. 3) [2]

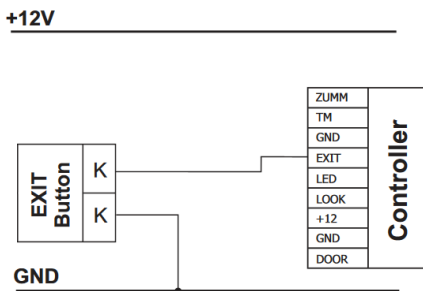


Рис. 3. Кнопка выхода

Из схемы видно, что кнопка является ключом между портами EXIT и GND. При нажатии кнопки, ключ замыкает контакт, что приводит к открытию двери.

Суть обнаруженной уязвимости заключается в принципе работы кнопки и портах, которые используются. Чтобы открыть дверь, имитируя нажатие кнопки, необходимо кратковременно замкнуть Порты EXIT и GND контроллера.

При подключении считывателя по протоколу Wiegand, как уже упоминалось, один из проводов DATA идет на порт EXIT контроллера. А так как земля в схеме общая, для открытия двери достаточно получить доступ к контактам считывателя и при помощи любого проводника кратковременно замкнуть необходимые контакты. Необходимая DATA определяется опытным путём, так как замыкание ТМ и GND безопасно и не приводит к выходу системы из строя в случае выбора другого порта DATA. Таким образом, получив доступ к считывателю, подключенному по протоколу Wiegand, можно открыть дверь, лишь замкнув необходимые контакты.

Библиографический список

1. Теоретическая информация о СКУД [Электронный ресурс] URL: http://pro-sec.ru/systems_access.html Дата обращения: 20.02.2017
2. Инструкции к контроллеру Z-5R [Электронный ресурс] URL: [https://ironlogic.ru/il.nsf/file/ru_317s_z-5r.pdf/\\$FILE/317s_z-5r.pdf](https://ironlogic.ru/il.nsf/file/ru_317s_z-5r.pdf/$FILE/317s_z-5r.pdf) Дата обращения: 03.03.2017
3. Инструкции к считывателю MATRIX 2 [Электронный ресурс] URL: [https://ironlogic.ru/il.nsf/file/ru_matrix_ii.pdf/\\$FILE/matrix_ii.pdf](https://ironlogic.ru/il.nsf/file/ru_matrix_ii.pdf/$FILE/matrix_ii.pdf) Дата обращения: 15.03.2017

УДК 004.514

Д. Д. Богданов
Т. В. Кучерук
А. А. Матвеева
К. Д. Семенов

студенты кафедры информационно-сетевых технологий
Красильникова О. И.
канд. техн. наук, доцент – научный руководитель

СОВРЕМЕННЫЕ ТЕНДЕНЦИИ В ВЕБ-ДИЗАЙНЕ

Информации в сети становится всё больше, взаимодействие пользователей с сайтами всё короче, а их требования к страницам в интернете всё выше. Приёмы и инструменты веб-дизайна, которые казались новыми и интересными ещё год назад, сегодня уже не актуальны. В связи с этим веб-сообщество постоянно работает над усовершенствованием пользовательского опыта, опираясь на технологические возможности, аналитику и пользовательские отзывы [6, 7].

Изменениям подвергаются даже фундаментальные вещи. В данной статье рассмотрим самые значимые изменения в веб-дизайне в настоящее время.

Сплит-дизайн. Разделение экрана на две части – сплит-дизайн, особенно заметный тренд в веб-дизайне. Сплит-дизайн представляет собой расположение двух вертикальных панелей рядом друг с другом и следует правилу: один экран, два сообщения. Каждая панель содержит отдельный элемент: фото, текст или иллюстрацию. Этот оригинальный дизайн сайта несет в себе следующие преимущества: акцентирование внимания на чем-то определенном; игра на контрастах; нешаблонный формат представления информации.

Такой тип оформления идеально подходит для навигации на большом экране или планшете. Для мобильных устройств используется вертикальное расположение панелей друг за другом.

Сплит-дизайн – это функциональный и адаптивный способ создания привлекательного оформления сайта. Но при практической реализации необходимо учесть такие моменты как количество свободного пространства для внедрения дизайна и отношение целевой аудитории к подобному роду дизайна. Данный дизайн – простой способ донести информацию [2, 10].

Градиенты и яркие цвета. Яркие цветовые палитры – тенденция 2017 года. С плоским дизайном в веб пришли новые возможности цветового оформления сайта, но при работе с данным стилем необходимо не допускать обезличивания сайта. Для решения данной проблемы используются яркие цвета и градиентные решения.

Данный тренд используется как для фона, так и для создания переходов между цветами для дальнейшего наложения их на иллюстрации. Яркие цвета позволяют выделять элементы на странице, что придает странице динамики и глубины [10].

Модульная сетка. Модульная сетка представляет собой набор невидимых направляющих, вдоль которых располагаются элементы веб-страницы. Это облегчает размещение данных на веб-сайте, обеспечивает визуальную связь между отдельными блоками и сохраняет преемственность дизайна при переходе от одной страницы к другой.

Как правило, классическая модульная сетка, разделяет данные по колонкам, каждый блок страницы отделен от других с помощью пустого пространства и рамок [1, 8].

В 2016 году появилось много разнообразных способов расположения элементов на странице сайта. Теперь у страницы нет деления на несколько фиксированных колонок: фотографии и текст могут располагаться хаотично, даже могут наезжать друг на друга, тем самым составляя необычные композиции [9]. Пример нестандартной сетки приведен на рис. 1.

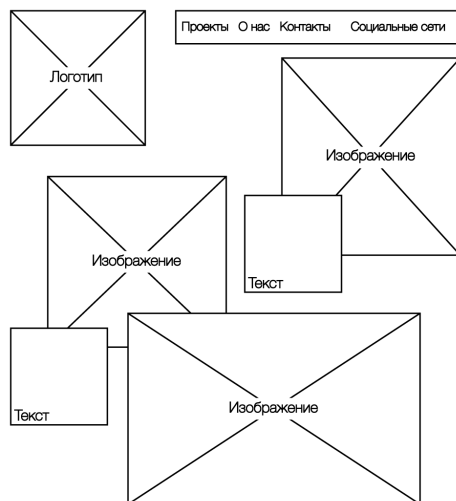


Рис. 1. Нестандартная модульная сетка

Основными достоинствами нестандартной модульной сетки являются возможность создания уникальной эстетики сайта, простота восприятия информации, новые горизонты для творчества. При грамотном использовании данных преимуществ происходит увеличение удержания пользовательского внимания к сайту.

Существенным недостатком является то, что при чрезмерном использовании возможностей данной сетки есть вероятность добиться обратного эффекта: неразумное использование блоков и излишняя информация приведет к осложнению восприятия [3].

Шрифт. В 2016 году стали повсеместно использоваться большие надписи и шрифты, которые являются ключевыми элементами веб-дизайна. Данная тенденция будет стремительно набирать обороты. На это влияет распространение минимализма на сайтах, удаление базовых составляющих и оформление надписей, как самостоятельных графических элементов.

Шрифты стали крупнее из-за того, что в последнее время все больше и больше людей используют мобильные устройства с небольшими экранами. Именно данная особенность стала вытеснять другие элементы дизайна.

До недавнего времени в интернете использовали простые, неогротескные шрифты, такие как Helvetica, Roboto, Lato или Open Sans. Шрифты с засечками же практически не использовались [2,4].

За последние два года ситуация начала меняться: получили распространение различные виды шрифтов; появилось больше контрастов – сочетаний шрифтов с засечками и без них; отказались от мягкого, гармоничного сочетания шрифтов для более выраженного контраста. Экспрессивные комбинации усиливались высоким контрастом размеров текстов. Крупные и декоративные шрифты сочетали с простыми геометрическими, также как геометрические заголовки становились в пару со шрифтами с засечками в тексте контента.

Тексты анимируются, разбиваются на отдельные буквы, в текстах размещаются различные эффекты, изображения и видео[4].

Анимация. Анимированные элементы давно играют ключевую роль в веб-интерфейсе, и эта тенденция продолжится в будущем. Доступ к инструментам для разработки привлекательных анимаций, делает эти элементы более заметными и более изощренными. Создание анимации упрощается с каждым днем, поэтому при разработке анимированных элементов необходимо уделять внимание це-

лям и потребностям пользователя, чтобы достигнуть эффект, при котором анимация будет выполнять значимые задачи. Формат может быть любым: WebGL CSS, GIFs, SVG или видео.

Видеоконтента стало больше за последний год, и люди стали более требовательны к высокому качеству видео. Видеоролик на сайте может быть коротким и поставлен на автоповтор, показывая продукт или масштабный кинематографический проект, который будет держать зрителя, вовлеченным в историю. Особенный интерес представляет синемаграфия – изображение с незначительными элементами движения.

Смысл синемаграфии заключается в том, что на статичной фотографии выделяется некоторая второстепенная деталь образа, будь то волосы, льющаяся вода или какое-либо движение и эта часть изображения анимируется при помощи локальной анимации. В течение короткого времени данные эффекты стали очень популярны.

Полученный результат напоминает обычные GIF изображение, но при этом совсем другого качества и эстетики. Данное направление – это уже не фотография, но и не видео, а что-то среднее между ними. Сама же технология создания такого изображения на самом деле значительно сложнее, чем создание простого анимированного изображения GIF.

Для создания синемаграфии самое главное при съемке на видео запечатлеть именно тот момент, который будет являться первостепенным при дальнейшей обработке изображения. При помощи любого видеоредактора осуществляется обработка полученного видеозображения и вырезание из него той части, которую мы в дальнейшем будем анимировать. Из вырезанной части видеозображения делают анимированную GIF-картинку и накладывают ее на статическую фотографию. Требуется опыт владения фотографической техникой и специализированными программами по обработке изображений.

Сама идея создания синемаграфий однозначно перспективна и будет непременно иметь большой успех [5].

Навигация. В настоящее время не обязательно размещать горизонтальную навигацию в шапке веб-сайта или использовать гамбургер-меню (иконка с тремя горизонтальными полосами, при нажатии на которую появляется меню сайта). Дизайн перестает делиться на мобильный и полноэкранный. В текущем году осуществляется отказ от общепринятых правил оформления и наблюдается множество экспериментов с размещением и формой меню.

Нестандартное меню навигации позволяет использовать пространство в новом виде. Новые вариации могут быть представлены в виде интуитивной прокрутки вниз и в сторону закрепленного вертикального блока навигации, а также других скрытых выезжающих элементов в зависимости от макета сайта. Такое меню позволяет вместить полностью все пункты на одном экране, что делает навигацию по сайту детальной и полезной для пользователя.

Обязательно следует протестировать эффективность нового меню на практике иначе можно сломать всю интуитивность дизайна и усложнить интерфейс [9, 10].

При создании сайта нет мелочей, необходимо уделять внимание от функциональности и быстрой реакции до внешнего вида.

Необходимость из года в год оставаться актуальным и востребованным, среди веб-сообщества, побуждает уделять большое внимание тенденциям и новым разработкам в веб дизайне. Каждый год появляется все больше и больше возможностей персонализации интерфейса. Это позволяет оставаться востребованным во все времена.

Стоит отметить, что хоть на данный момент изменению подвергаются даже базовые элементы дизайна, радикально новые направления появляются довольно редко. Частично тенденции прошлого остаются актуальными и сегодня.

Библиографический список

1. *Мюллер-Брокманн Й.* Модульные системы в графическом дизайне: пособие для графических дизайнеров, типографов и оформителей выставок. М.: Изд-во Студии Артемия Лебедева, 2014. – 184 с.

2. Уэйншенк, С. 100 главных принципов дизайна: Как удержать внимание / СПб.: Питер, 2011. – 290 с.
3. Голден Кришна. Хороший интерфейс – невидимый интерфейс. СПб.: Питер, 2016. – 256 с.
4. Тренды веб-дизайна 2017 года, URL: http://deadsign.ru/design/web_design_trends_2017
5. Синемаграфия – «ожившие» фотографии, URL: <https://allfreefoto.ru/1064-sinemagrafiya.html>
6. 6 трендов веб-дизайна в 2017 году, URL: <http://www.cossa.ru/152/145136>
7. 18 web design trends for 2017: Responsive web design tool, CMS, and hosting platform, URL: <https://webflow.com/blog/18-web-design-trends-for-2017>
8. Модульная сетка, URL: <http://htmlbook.ru/content/modulnaya-setka>
9. Тренды веб-дизайна в 2017 году: анимация, боты и нестандартная сетка, URL: <https://test.ru/2016/12/28/web-design-trends-in-2017>
10. Тенденции и тренды веб-дизайна сайтов 2017, URL: <http://design-mania.ru/web-design/trends2017>

УДК 004.056.53

Н.А. Голос

студент кафедры технологий защиты информации

С.В. Беззатеев

д-р техн. наук, доцент – научный руководитель

АНАЛИЗ МЕТОДОВ ДВУХФАКТОРНОЙ АУТЕНТИФИКАЦИИ

В данной статье рассмотрены задачи, решаемые аутентификацией, ее области применения, популярные способы однофакторной и двухфакторной аутентификации, анализ их уязвимостей и методов защиты от них.

Аутентификация – это проверка подлинности объекта. С помощью нее можно установить, является ли объект тем, за кого себя выдает. Чтобы доказать свою подлинность, объект должен предоставить так называемый фактор аутентификации. Есть три основных фактора аутентификации:

- Пользователь знает некоторую секретную информацию, например, пароль
- Пользователь имеет определенный предмет с уникальными характеристиками, например, RFID метку или мобильный телефон
- Уникальные биологические, психологические и иные характеристики пользователя, являющегося непосредственно его частью.

Для того чтобы пройти аутентификацию, пользователь должен обладать одним из перечисленных факторов. Составим таблицу преимуществ и недостатков каждого отдельного фактора аутентификации (табл. 1).

Таблица 1

Плюсы и минусы факторов аутентификации

Фактор аутентификации	Секретная информация	Уникальное устройство	Биологические и иные х-ки пользователя
Преимущества	1. Прост в реализации 2. Не может быть изменен без желания пользователя (сломаться или испортиться)	1. Не нужно запоминать секретную информацию	1. Данный фактор не передать другому объекту 2. Не нужно запоминать и невозможно потерять
Недостатки	1. Возможно подобрать пароль (из-за пользователей, которые пренебрегают надежностью пароля) 2. Можно перехватить пароль при вводе, передаче по каналу 3. Пароль можно забыть	1. Устройство может быть похищено/потеряно 2. Возможно изготовление копии устройства 3. Может сломаться	1. Недостаточно точный (есть вероятность аутентификации ложно пользователя или отказа в аутентификации достоверному объекту) 2. Можно подделать или снять в другом месте биометрию пользователя 3. Сложен для внедрения 4. Биометрические показатели объекта могут изменяться (в результате травм, болезни и т.д.)

Благодаря простоте реализации аутентификация с помощью логина/пароля получила наибольшее распространение, особенно в Интернете. Однако, как видно из рис. 1, данный фактор имеет ряд существенных недостатков, поэтому, для повышения безопасности систем была реализова-

на двухфакторная аутентификация, которая требует от объекта, соответственно, предоставить не один, а два фактора аутентификации.

Основными способами двухфакторной аутентификации являются:

- 1) аутентификация с помощью логина/пароля и смс-токена,
- 2) аутентификация с помощью логина/пароля и email-токена,
- 3) аутентификация с помощью логина/пароля и аппаратного токена,
- 4) аутентификация с помощью логина/пароля и программного токена.

Первый способ работает следующим образом:

Пользователь вводит логин и пароль, после проверки их на корректность, сервис посылает текстовое сообщение, содержащее специальный код или пароль, так называемый OTP (one time password), на телефон пользователя, который он должен ввести на странице входа для завершения аутентификации.

Рассмотрим уязвимости данного способа аутентификации:

- Так как логин/пароль и смс код вводятся пользователем с одного устройства, то злоумышленник, установив вредоносную программу, которая считывает данные с клавиатуры, так называемый клавиатурный шпион, на устройство пользователя, может записать все необходимые данные для аутентификации.

- Аутентификация с помощью логина/пароля и смс-токена уязвима к атаке «Человек посередине». Для ее реализации злоумышленнику достаточно сделать веб-страницу, идентичную по внешнему виду настоящей странице. Далее, если пользователь, который попадает на ложную страницу, введет там свой логин и пароль, то ложная страница передаст их на настоящую, затем попросит пользователя ввести смс-код и опять же передаст его на настоящую страницу. В итоге злоумышленник пройдет аутентификацию, а настоящий пользователь нет.

- Следующая уязвимость касается сим-карт, точнее тех, которые используют шифрование DES. На 2013 год таких карт было примерно 750 млн [3], и называется она «Sim card cloning». Суть ее заключается в том, что злоумышленник делает клон сим карты пользователя и может получить смс код аутентификации.

Аутентификация с помощью логина/пароля и email токена работает следующим образом: пользователь вводит логин и пароль, если они корректны, то сервис отправляет OTP на email, который пользователь должен ввести для завершения аутентификации.

Данный способ дешевле реализовать, чем аутентификацию через смс или аппаратный токен, но он считается менее надежным. Рассмотрим уязвимости данного способа.

Так же как и аутентификация с помощью логина/пароля и смс токена, данный способ аутентификации уязвим к атаке через клавиатурный шпион и атаке «человек посередине». Но, из-за того, что OTP приходит не через смс, а на email данный способ аутентификации обладает следующими уязвимостями:

- доступ к почте злоумышленник может получить через троянскую программу;
- зачастую пользователи используют слабый пароль, поэтому есть возможность его подобрать по словарю наиболее популярных паролей;
- доступ к почте можно получить, путем ответа на секретный вопрос, который, как показали исследования, еще менее надежен чем пароль [1].

Аутентификация с помощью логина/пароля и аппаратного токена реализуется следующим образом. Токен содержит некоторые секретные сведения, например, пароль, который передается при аутентификации, или же он генерирует пароли каждый определенный интервал времени, который пользователь должен ввести. Токен может подключаться к устройству, с которого проводится аутентификация через Bluetooth или USB, или же быть беспроводным с дисплеем для отображения данных аутентификации, такой метод используют токены модели RSA SecurID SID700. Основная уязвимость аппаратного токена заключается в том, что он может быть утерян, украден или клонирован. Если же система позволяет аутентифицироваться через сеть, например, Интернет, то так же возможна атака «Человек посередине».

Дополнительным минусом данного способа является то, что он является более дорогим в реализации по сравнению с остальными методами двухфакторной аутентификации.

Рассмотрим аутентификацию с помощью логина/пароля и программного токена. Чтобы пользователю не приходилось носить с собой дополнительное устройство, как в предыдущем способе, на телефон, компьютер, планшет или аналогичное устройство пользователя устанавливают специальное приложение, которым можно воспользоваться при необходимости аутентификации. После проверки логина и пароля на корректность, нужно будет ввести OTP, который генерирует данное приложение (приложение Google Authenticator) в специальном отдельном поле. В связи с тем, что токен хранится не на отдельном устройстве, у данного способа аутентификации есть следующие уязвимости и недостатки:

- Так как токен хранится на устройстве, которое используют не только для аутентификации, но, например, для хранения файлов, выхода в Интернет и т.д., есть возможность установить на него вирус, который сможет извлечь секретную информацию из программного токена. Зачастую вредоносный код встраивается в обычные приложения, например для прослушивания музыки или игры.

- Аппаратный токен представляет собой приложение, следовательно, оно может быть установлено злоумышленником, а затем с помощью реверс-инжиниринга злоумышленник может выяснить как приложение генерирует OTP.

- Устройство с аппаратным токеном также может быть потеряно или украдено

Как видно, двухфакторная аутентификация не гарантирует полную безопасность пользователя. Чтобы повысить ее безопасность нужно выполнять следующие рекомендации:

1. Пароль, используемый при аутентификации, должен соответствовать определенным правилам, таких правил достаточно много, можно взять в качестве примера требования к паролю Стенфордского университета [5], они зависят от длины пароля:

- 8-11 символов: пароль должен содержать буквы верхнего и нижнего регистра, цифры и специальные символы,

- 12-15 символов: пароль должен содержать буквы верхнего и нижнего регистра и цифры,

- 16-19 символов: пароль должен содержать буквы верхнего и нижнего регистра,

- 20+ символов: нет требований к паролю,

- пароль не должен совпадать с логином,

- он не должен быть одним словом, которое есть в словаре английского и любого другого словаря.

Как видно, чем длиннее пароль, тем он безопаснее, однако, есть проблема запоминания пароля. Для этого рекомендуют составлять пароль следующим образом: взять несколько случайных слов и соединить их в один пароль, например, «eagle orange key shoe» выходит пароль длиной в 21 символ, включая пробелы, и он достаточно прост для запоминания.

2. На устройствах, которые используются для аутентификации, должен стоять антивирус, чтобы обнаруживать и устранять вредоносные программы, например трояны и клавиатурные шпионы.

В заключение, рассмотрим перспективные способы двухфакторной аутентификации. В рассмотренных ранее способах, оба фактора аутентификации вводятся с одного устройства, а это значит, что при заражении устройства, которое используется для ввода факторов аутентификации, злоумышленник получит необходимые данные для аутентификации, поэтому следует вводить один фактор с одного устройства, а второй – с другого. Такой способ реализован в протоколе U2F (universal 2-factor) аутентификации. Он работает следующим образом, сначала, пользователь вводит в специальную форму логин и пароль, если они корректны, то на специальное устройство пользователя отправляется запрос, если пользователь, в простейшем случае, нажмет на нем кнопку, то аутентификация пройдет успешно. Данный протокол позволяет бороться аутентификации с помощью логина/пароля и аппаратного токена с подменой токена, фишингом, атакой «человек посередине». Соответственно, для этого используется:

- добавление URL источника к ответу устройства;

- встроенный счетчик, где каждая подпись и регистрация увеличивает его состояние;

- использование «client data», более подробно описана в документации [6].

Так же похожий способ реализован в двухфакторной аутентификации в приложениях Google и называется Google Prompt. Работает он следующим образом: после проверки корректности логина и пароля, на специальное приложение на смартфоне пользователя приходит уведомление, на которое пользователь должен нажать, чтобы пройти аутентификацию. Однако, данный способ менее надежен чем U2F, так как смартфон используется не только для аутентификации, но и для других целей, а это значит, что на него можно установить вредоносное ПО и подтвердить аутентификацию вместо пользователя.

Библиографический список

1. It's no secret/ Measuring the security and reliability of authentication via 'secret' questions / Stuart Schechter, A. J. Bernheim Brush, Serge Egelman. – P. 1-16
2. Security analysis of mobile two-factor authentication schemes / Aleksandra Dmitrienko. – 2014. – P. 1-161
3. Sim Card Cloning Hack affect 750 millions users around the world. / Mohit Kumar. – 2013. – P. 1
4. Hacking soft tokens advanced reverse engineering on android / Bernhard Mueller. – 2016. – P. 1-68
5. Требования к паролю Стенфордского университета [Электронный ресурс]. URL: <https://uit.stanford.edu/service/accounts/passwords>
6. Universal 2nd Factor (U2F) Overview / Sampath Srinivas, Dirk Balfanz, Eric Tiffany. – 2014. – P. 1-24

УДК 629.735.33

В. А. Егорлицкий

магистрант кафедры безопасности информационных систем

Е. М. Линский

канд. техн. наук, доцент – научный руководитель

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ПРОТОКОЛА КЭШИРОВАНИЯ PACKET-RE

В настоящее время нужды пользователей растут быстрее, чем скорость линий связи. Поэтому провайдеры стремятся к экономии трафика и сохранению пропускной способности. Также разные пользователи часто загружают одни и те же данные из сети Интернет, просматривая новостную ленту или популярный контент (видеозаписи, изображения и т.д.), поэтому сохранение пропускной способности может быть достигнуто за счёт использования кэширования. В данном докладе рассматриваются популярные протоколы кэширования, исследуется эффективность протокола кэширования Packet-RE и предлагается новый подход к кэшированию трафика.

Существуют различные протоколы кэширования, наиболее популярные – классическое HTTP-кэширование [1] и префиксное веб-кэширование. Классическое HTTP-кэширование основано на идентификации данных по единственному указателю ресурса (англ. «URL» – Uniform Resource Locator). Если пользователь получает доступ к ресурсу с некоторым URL (например, <http://example.suai.ru/1.jpeg>), кэш сохраняет полученный ресурс и запоминает соответствующий ему URL. При следующем обращении к данному ресурсу, кэш сравнит URL запрашиваемого ресурса с имеющимися, и, в случае совпадения, вернёт запрашиваемые данные (без повторной загрузки их из сети Интернет) – такая ситуация называется «попаданием» (англ. «hit») в кэш. Обратная ситуация, когда запрашиваемый элемент не найден в кэш, называется промахом (англ. «miss»). Классическое HTTP-кэширование имеет существенный недостаток, связанный с идентификацией данных. Если пользователь попытается получить из сети один и тот же ресурс с разными URL (например, одно и то же изображение с разных серверов), то «попадания» в кэш не произойдёт. Этот недостаток можно избежать, используя префиксное веб-кэширование. В префиксном кэшировании идентификация данных производится по их содержанию, а именно по так называемому «префиксу», т.е. для того, чтобы определить наличие данных в кэш, производится сравнение части получаемых данных с уже имеющимися в кэш. При таком подходе, большую роль играет размер «префикса». Иногда возникает ситуация, когда разные ресурсы совпадают по началу файла. Таким образом, короткий «префикс» может привести к тому, что два различных ресурса будут считаться одинаковыми с точки зрения кэш. Такие ситуации являются главным недостатком префиксного веб-кэширования и называются «ложными срабатываниями» (англ. «Falsepositives»).

Наиболее современным подходом к кэшированию является Packet-RE [2], который может использоваться в сотовых сетях для экономии пропускной способности транзитной сети. Packet-RE функционирует на сетевом уровне и структурно представляет из себя два промежуточных кэширующих устройства с идентичным состоянием. Основной объект кэширования – это данные пакета без заголовков протоколов канального, сетевого и транспортного уровней. Кэшированию подвергается только поток по направлению из внешней сети к пользователю (т.е. передача популярного трафика в эту сторону наиболее вероятна). Пакет, попадая на верхний кэш (непосредственно связанный с сетью Интернет), анализируется на предмет наличия в нем. Если пакет новый, то его достаточно добавить в оба кэширующих устройства. При следующем «попадании» достаточно лишь передать идентификатор (например, хэш данных пакета) между устройствами. В случае, если его длина меньше длины исходных данных пакета, по каналу связи между двумя кэшами проходит меньше трафика. Данное кэширование не зависит от URL ресурса и идентифицирует данные по всей их длине, а значит устраняет недостатки классического HTTP-кэширования и префиксного веб-кэширования. Однако и у данного подхода имеется недостаток, также связанный с идентификацией данных. Так как Packet-RE использует идентификацию по данным отдельных пакетов, то такой подход не будет корректно работать в случае,

если один и тот же ресурс будет по-разному разделён на пакеты. Такое поведение может наблюдаться при использовании различных операционных систем (ОС). Был проведён ряд экспериментов, которые включали в себя загрузку одинаковых ресурсов из сети Интернет на различных ОС (Windows и Linux). Одновременно выполнялся анализ трафика сети с использованием программы Wireshark [3]. В результате, были получены различные последовательности пакетов одного и того же ресурса (изображения с удалённого сервера). Из графика на рис.1 видно, что на ОС Linux, полученный ресурс представляет собой всего 38 TCP-сегментов, в то время как на ОС Windows тот же ресурс представлен в виде 73-х сегментов. Также можно заметить, что пакеты имеют различную длину, что, очевидно, значительно снижает число «попаданий» в кэш.

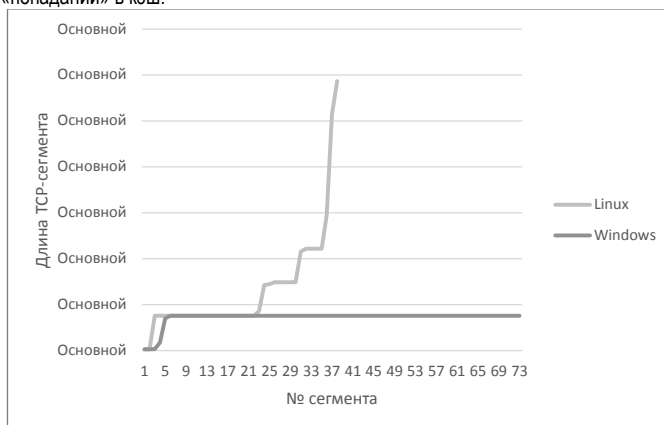


Рис. 1. Различное разделение на пакеты одного и того же ресурса

Существует альтернатива протоколу Packet-RE, именуемая TCP-RE. Основная идея и отличие данного протокола от Packet-RE заключается в том, что кэширование применяется не отдельно для каждого пакета, а для ресурса в целом. Как правило, ресурс – это отдельный TCP-поток, который можно идентифицировать по таким параметрам IP-пакета, как порт отправителя, порт получателя, IP-адрес отправителя и IP-адрес получателя. Казалось бы, кэширование можно производить, сохраняя весь поток и возвращая его данные при «попадании» в кэш. Однако, такая концепция TCP-RE кэширования предполагает наличие всех сегментов TCP потока, что может вызвать дополнительные задержки в работе сети. Поэтому предлагается несколько иная стратегия кэширования TCP-RE: кэш представляет собой структуру (набор) потоков, и при получении нового пакета определяется его принадлежность потоку, происходит сохранение в кэш; при повторной попытке получить ресурс из сети данные каждого получаемого пакета могут быть найдены среди имеющихся потоков, и, в случае «попадания» в кэш, необходимо лишь вернуть номер потока с искомыми данными, смещение от начала данных потока и длину сегмента (см. рис.2). Таким образом, TCP-RE кэширование не зависит от того, как ресурс был разделён на сегменты, т.к. поиск данных будет происходить по всем имеющимся потокам, т.е. данный подход устраняет основной недостаток Packet-RE кэширования, а значит и недостатки HTTP и префиксного кэширования также. Более того, между двумя кэширующими устройствами будет передаваться фиксированное число байт (номер потока, смещение от начала и длина сегмента).

В дальнейшем, чтобы достоверно убедиться в том, что использование предложенного подхода эффективнее Packet-RE, необходимо для обоих подходов рассчитать процентные соотношения числа «попаданий» от общего числа сегментов, используя результаты анализа трафика, полученные в ходе экспериментов.

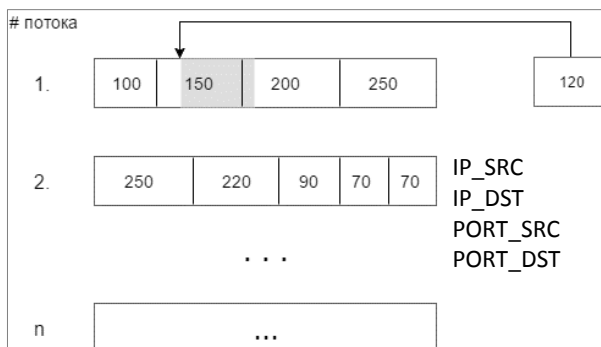


Рис. 2. Концепция поиска сегмента в TCP-RE кэш

Библиографический список

1. Squid. The most popular HTTP cache, URL: <http://www.squid-cache.org/Intro/>
2. Comparison of Caching Strategies in Modern Cellular Backhaul Networks/KyoungSoo Park KAIST, Daejeon, South Korea, 2013. 9 с.
3. Wireshark: About, URL: <https://www.wireshark.org/>

УДК 004.89

А. А. Заусалин

студент кафедры безопасности информационных систем

А. И. Веселов

канд. техн. наук, доцент – научный руководитель

ПРОГРАММНЫЙ МОДУЛЬ ДЛЯ ИДЕНТИФИКАЦИИ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ РАСПОЗНАВАНИЯ ЛИЦ

В современном мире часто ставятся задачи идентификации персоны по изображению. Такие задачи актуальны для систем контроля доступа в защищенных объектах, для поиска и слежки за подозреваемым для предотвращения преступления или восстановления хронологии уже случившегося события. Задачу распознавания лиц на изображении можно решить несколькими способами.

1. Обучить персонал или службу охраны узнавать всех сотрудников в лицо. Такой способ даст хороший результат при маленьком числе сотрудников и малой интенсивности потока входящих людей. Но в таком случае придется обучать персонал и доплачивать ему за дополнительную работу.

2. Попиксельно сравнивать изображения. Данный метод, хоть и имеет малую вычислительную сложность, но не обладает необходимой точностью, так как при распознавании необходимо, чтобы лицо на изображении, на котором необходимо идентифицировать персону, было максимально схоже с эталонным изображением.

3. Обучить компьютер находить и распознавать лица на кадрах, захваченных с камер видеонаблюдения. Данный класс методов, как правило, основан на машинном обучении [1] и является наиболее распространенным в настоящее время.

Как правило, при распознавании лиц используются методы, работающие только с одним изображением видеоряда. В данной работе будет введена в рассмотрение теоретическая модель, позволяющая учесть несколько изображений. С использованием данной модели будет экспериментально определено оптимальное число кадров, используемых при распознавании с применением простого мажорирующего правила.

Здесь и далее будем предполагать, что система идентификации включает три основных модуля:

- модуль трекинга – выполняет отслеживание за детектированными лицами в видеоряде;
- модуль детектора – выполняет поиск лица на изображении;
- модуль распознавания – выполняет распознавание ранее детектированного лица.

При создании модели учтем следующие ситуации, которые могут возникать при распознавании лиц.

Первым рассматриваемым случаем будет являться ошибка детектора лиц, то есть в какой-то момент времени может произойти ложное срабатывание детектора и в последовательности окажется изображение, на котором нет лица.

Вторым случаем рассматривается ошибка трекинга объектов. Это означает, что алгоритм трекинга при отслеживании объекта ошибочно указал область, в которой нет отслеживаемого объекта. В таком случае на ошибочном кадре может быть детектировано лицо другого человека.

Также, стоит учесть ошибки классификации при распознавании лиц. Для упрощения модели введем допущение, что в базе присутствует только два человека. В таком случае, ошибка классификатора включает в себя следующие ситуации:

- классификатор вынес решение о том, что на изображении присутствует человек с меткого класса 1, хотя на изображении был человек с меткой класса 0 и наоборот;
- классификатор отнес неизвестного человека к 0-му классу или к 1-му;
- классификатор не узнал человека из базы;
- классификатор вынес решение по изображению без лица.

На основе вышеперечисленных ситуаций можно построить следующую модель системы распознавания на базе скрытого Марковского процесса (рис. 1).

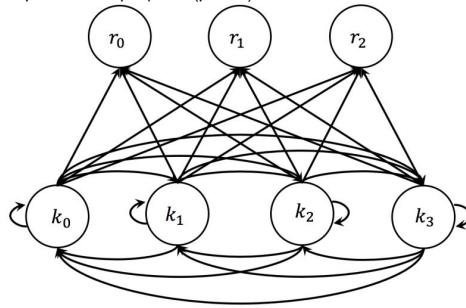


Рис. 1. Схема модели.

В данной модели используются случайные величины $K \in \{k_0, k_1, k_2, k_3\}$ и $R \in \{r_0, r_1, r_2\}$, где k_0 – лицо человека с меткой класса 0, k_1 – лицо человека с меткой класса 1, k_2 – лицо человека, которого нет в базе, k_3 – не лицо (ошибка детектора), r_0 – решение классификатора: 0 класс, r_1 – решение классификатора: 1 класс и r_2 – решение классификатора: неизвестно. Случайный процесс для описанной модели, развернутый во времени для последовательности кадров с номерами $1, 2, \dots, T$ представлен на рис. 2.

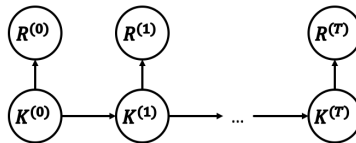


Рис. 2. Упрощённая схема модели

Решающее правило для распознавания лиц с использованием данной модели можно сформулировать как:

$$c^* = \operatorname{argmax}_c \sum_{i=1}^T I\{R^i = r_c\},$$

Для данной модели вероятность ошибки распознавания для фиксированной длины последовательности кадров T можно посчитать как:

$$P_{err}(T) = \Pr\left\{\sum_{i=1}^T I\{R^i \neq r_0\} > \frac{T}{3} \mid r_0\right\}.$$

Таким образом, можно сформулировать следующую оптимизационную задачу для определения оптимального числа кадров:

$$\begin{cases} \min T \\ P_{err}(T) < P_{target} \end{cases},$$

где P_{target} – ограничение на вероятность ошибки распознавания.

Для поиска решения представленной оптимизационной задачи было выполнено имитационное моделирование. Для моделирования поведения предложенной модели были взяты характеристики те-

стирования детектора лиц из библиотеки Dlib [3] и результаты тестирования трекера Tracking-Learning-Detection (TLD) из статьи, опубликованной авторами метода [4].

Имитационное моделирование учитывало следующие параметры:

- набор вероятностей для величин r_0, r_1, r_2, r_3 ;
- ошибку второго рода для детектора Dlib;
- ошибку второго рода для трекера TLD.

В процессе имитационного моделирования для каждой последовательности длины T было проведено 100000 экспериментов. Результатом имитационного моделирования являлась вероятность правильного распознавания первой персоны. Моделирование проходило до того момента, пока вероятность ошибки не стабилизируется или пока длина последовательности кадров не превысит 100. Результаты имитационного моделирования представлены на рис. 3. По данному графику видно, что вероятность ошибки при T равной единице, то есть принятии решения на основе только одного кадра, была равна приблизительно 0.026, а при увеличении периода T ошибка уменьшалась, пока при $T = 15$ не стала приблизительно равна нулю.

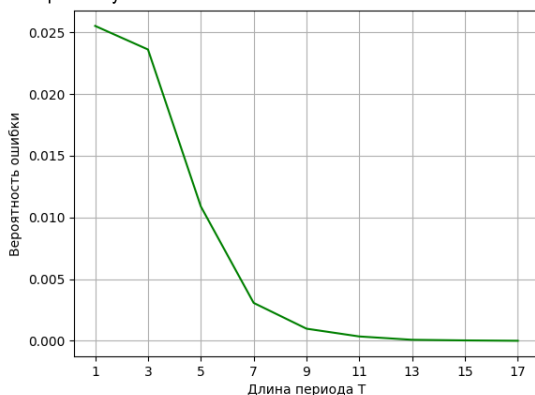


Рис. 3. Результаты имитационного моделирования

В данной работе была предложена и построена модель для распознавания лиц на основе последовательности кадров. Получены результаты тестирования алгоритма. Проведено имитационное моделирование поведения предложенной модели и выявлена длина последовательности кадров, при которой ошибка стремится к нулю.

Можно указать следующие направления будущих исследований:

- анализ различных методов принятия решений для нейронной сети ResNet;
- тестирование модели с использованиями других видов трекинга.

Библиографический список

1. Математические методы обучения по прецедентам [Электронный ресурс]. URL: <http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf>
2. He K. Deep Residual Learning for Image Recognition / K. He, X. Zhang, S. Ren and J. Sun // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016, pp. 770-778.
3. Открытая библиотека для языков C++ и Python [Электронный ресурс]. URL: dlib.net
4. Tracking-Learning-Detection Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas [Electronic resource]. http://kahlan.eps.surrey.ac.uk/featurespace/tld/Publications/2011_tpami

УДК 004.415.25

В.И. Ильин

студент кафедры технологий защиты информации

К.А. Жиданов

старший преподаватель – научный руководитель

АНАЛИЗ ТЕХНОЛОГИЙ ДЛЯ ПОСТРОЕНИЯ АРХИТЕКТУРЫ API ВЕБ-СЕРВИСОВ

API (application programming interface) – это интерфейс прикладного программирования. Он служит в качестве связующего звена между различными программными продуктами или между различными частями одной системы и облегчает их взаимодействие, подобно тому, как пользовательский интерфейс облегчает взаимодействие с программами. API могут варьироваться от низкоуровневых инструментов, которые позволяют приложениям Windows и OS X использовать аспекты операционной системы, до тех, которые обеспечивают работу мобильных приложений. В данной статье рассматриваются API, которые позволяют взаимодействовать с веб-приложениями.

Области использования API заключаются в построении систем с использованием функций других систем, автоматизации использования функций систем, а также использование API при разработке мобильных приложений. Примером использования API может стать разработка мобильного приложения, которое может выступать как альтернатива существующему. Главным условием возможности разработки такого приложения является наличие открытого API у исходного приложения, которым будет пользоваться новое. В этом случае новое приложение сможет выполнять те функции, которые возможно выполнить при помощи API, но кроме этого сможет добавит новые функции, доступные только внутри нового приложения. Другим примером использования API может послужить система мониторинга и автоматизации определенных действий. Такая система может анализировать данные полученные от внешнего API веб-сервиса и определенным образом реагировать на них.

При выборе технологий при помощи которых будет построено API веб-сервиса следует учитывать многие факторы, влияющие на внутреннюю архитектуру API, такие как удобство его использования и количество времени, потраченное на разработку. Неправильный выбор технологий для разработки может привести к повышенным издержкам в тоже время, полученный результат может быть аналогичен результату, построенному при помощи других технологий и в более краткие сроки. Целью данного исследования является анализ и сравнение существующих технологий для построения архитектуры API веб-сервисов.

Существует две основных технологии, которые можно использовать при построении API. В зависимости от выбора между ними будет зависеть архитектура API, способы обращения к API, формат возвращаемых данных и время обработки запроса, поэтому решение переключиться с использования одной технологии на другую будет означать полное написание API заново. Именно поэтому важно в самом начале правильно определить технологии применимые в конкретном случае.

Первым подходом к построению API является использование протокола доступа к объектам SOAP который определяет механизм передачи структурированной информации в формате XML по протоколам HTTP, SMTP, TCP, UDP и др. [1]. Для описания SOAP веб-сервиса используется язык описания веб-сервисов WSDL. При помощи него составляется файл, описывающий в формате XML все функции, их входные и выходные параметры и их типы.

Другим способом построения API является архитектурный подход REST который использует протокол HTTP для передачи данных в различных форматах (XML, JSON, YAML и др.). Веб-сервис представляется как набор ресурсов каждый из которых имеет свой уникальный URI (Uniform Resource Identifier). Например, если обратиться к веб-сервису по URI <http://www.example.com/api/users/5>, то веб-сервис вернет информацию о пользователе, у которого идентификатор равен пяти. Для выполнения операций с ресурсами используются стандартные методы HTTP.

При выборе одной из технологий появляются преимущества и недостатки присущие системе. В первую очередь с API должно быть удобно работать. Это экономит время разработчиков, которые бу-

дут использовать его. Для этого API должен быть хорошо документирован, а формат возвращаемых данных должен быть удобен для использования в конкретных системах. В таблице 1 приведено сравнение двух технологий по ключевым для API критериям.

Таблица 1

Сравнение технологий для разработки API

Характеристика\технология	SOAP	REST
Концепция	Объекты и методы	Ресурсы
Формат данных	Строгое использование XML	Использование YAML, XML, JSON или любых других
Типизация параметров	Строгая типизация	Отсутствие строгих требований
Создание документации	Получение документации из файла описания сервиса	Составление документации вручную
Удобство использования при помощи Javascript (AJAX)	Не удобен, сложная реализация, излишняя логика построения запросов на стороне клиента	Удобен

Технологии используют разные подходы обращения к API – при использовании SOAP мы обращаемся к определенным методам и передаем им объекты, а при использовании REST мы используем понятие уникальных ресурсов. Интуитивно понятнее назначение и построение URI при использовании REST, в то время разобраться в файле описания веб-сервиса займет гораздо больше времени.

SOAP жестко привязан к использованию формата данных XML [2], в то время как REST может работать с любыми форматами данных, в том числе и с XML, что дает большую гибкость в использовании. Использовать XML удобно не во всех системах поэтому при использовании REST по этому критерию мы получаем преимущество.

В SOAP используется строгая типизация входных и выходных параметров, которая отражена в файле описания сервиса, в то время как при использовании REST нет строгих требований, из-за чего потенциально могут возникнуть ошибки при разработке API.

По содержанию файла описания веб-сервиса можно автоматически сгенерировать документацию, а при использовании REST разработчикам придется писать документацию вручную. В этом случае SOAP не имеет преимуществ над REST потому что для того, чтобы написать файл описания веб-сервиса, разработчикам нужно потратить столько же времени, как и для написания документации соответствующего уровня для REST API.

Важным критерием является удобство использования API при помощи технологии AJAX (Asynchronous JavaScript And XML) которая позволяет сделать запрос и получить ответ от сервера без перезагрузки веб-страницы. Она используется на клиентской части веб-приложений и в случае использования на сервере SOAP на клиентской стороне возникает сложность построения запросов и излишняя логика которой можно избежать при использовании REST.

Кроме этого, веб-сервисы, основанные на SOAP, имеют стандартную поддержку взаимодействия с таким протоколом как WS-Security, который обеспечивает безопасность передаваемых сообщений от источника сообщения к его получателю (end-to-end) [3]. В REST нет способа обеспечения безопасности, взаимодействие построено на протоколе HTTP (HTTPS), который в свою очередь обеспечивает безопасность только между промежуточными узлами связи (point-to-point) [3]. Поэтому использование SOAP в системах где предъявляются высокие требования к безопасности информации полностью оправдано.

Одним из важнейших критериев при выборе технологии является время ответа сервера на запрос. Для определения оптимального решения по данному критерию было разработано API с использованием обеих технологий. Функции, предоставляемые API, были одинаковыми и предполагали базовые операции работы с сущностями базы данных – создание, чтение, редактирование и удаление. Для тестирования был использован тестовый сервер с частотой процессора 4 ГГц, оперативной памятью 8 Гб, в качестве операционной системы использовалась Windows 10, в качестве веб-сервера использовался IIS10, база данных

использовалась MSSQL2016, серверная часть приложения было построена при помощи фреймворка ASP.NET Web API 2 и WCF. В таблице 2 приведены практические результаты тестирования API.

Таблица 2

Сравнение технологий для разработки API

Метод API/Технология	SOAP		REST	
	Задержка получения ответа, мс	Размер пакета, байт	Задержка получения ответа, мс	Размер пакета, байт
Создание конференции	372.7	833	99.7	269
Получение информации о конференции	339.4	612	96.4	165
Добавление участника	337.8	680	107.3	210
Удаление участника	323.4	674	104.1	181
Получение участников конференции	340	730	100.8	201

Было выявлено, что время ответа сервера и размер SOAP пакета значительно больше чем для выполнения тех же операций при помощи REST веб-сервиса. Кроме этого объем информации, посылаемой при общении с SOAP веб-сервисом может вызвать проблемы с производительностью поскольку при формировании сообщения протокол добавляет дополнительные заголовки и части сообщения для того, чтобы сообщение было сформировано в соответствии со стандартом, устанавливаемым SOAP. Это может быть крайне критично в системах где ограничен объем пересылаемой информации.

Таким образом в рамках данного исследования были рассмотрены две технологии при использовании которых можно построить API веб-сервиса – протокол SOAP и подход REST. Сделать выбор между ними предстоит перед началом разработки поскольку перейти с использования одной технологии на другую и перенести существующий функционал нельзя. Выбор технологии зависит от требований, предъявляемых к API. Прежде всего SOAP подойдет в системах, где необходимо обеспечить высокую безопасность обмена данными. Для этого стоит использовать надстройку над протоколом SOAP WS-Security. Однако его нельзя использовать в случаях ограниченного трафика, поскольку для взаимодействия посредством SOAP в сообщения включены дополнительные заголовки и части сообщения необходимые для соответствия шаблона сообщения стандарту. Как показали практические результаты тестирования, размер пакета при использовании SOAP ощутимо больше чем при использовании REST, поэтому в условиях ограниченного трафика лучше использовать REST. Использование REST оправдано при необходимости делать вызовы к API из клиентской части приложения при помощи технологии AJAX, потому что для этого не потребуется нагружать клиентскую часть приложения излишней логикой формирования запросов. Помимо этого, мы выяснили, что веб-сервис разработанный при помощи REST обрабатывает запрос быстрее, чем SOAP веб-сервис, что делает REST более предпочтительным вариантом при выборе технологии для построения архитектуры высоконагруженного веб-сервиса.

Библиографический список

1. Aaron Skonnard: Understanding WS-Security [Электронный ресурс] – march, 2003 – Режим доступа: <https://msdn.microsoft.com/en-us/library/ms995800.aspx>, свободный, [обращение 01.05.2017].
2. Leidago Noabeb: Anatomy of a Web Service: XML, SOAP and WSDL for Platform-independent Data Exchange [Электронный ресурс] – june, 2010 – Режим доступа: http://www.webreference.com/authoring/web_service/2.html, свободный, [обращение 04.05.2017].
3. Scott Seely: Understanding WS-Security [Электронный ресурс] – october, 2002 – Режим доступа: <https://msdn.microsoft.com/en-us/library/ms977327.aspx>, свободный, [обращение 09.05.2017].

УДК 519.633

А. В. Ким, А.Ю. Бобина

студенты кафедры информационно-сетевых технологий

А. Д. Жуков

канд. техн. наук, доцент – научный руководитель

ИССЛЕДОВАНИЕ ВЛИЯНИЯ АРХИТЕКТУРЫ НЕЙРОННОЙ СЕТИ НА ТОЧНОСТЬ АППРОКСИМАЦИИ АНАЛИТИЧЕСКОЙ ФУНКЦИИ

В окружающем нас мире все взаимосвязано, поэтому одной из наиболее часто встречающихся задач является установление характера зависимости между различными величинами, в чем и состоит идея теории аппроксимации функций. Ее основная задача – нахождение функции близкой (т.е. аппроксимирующей) в некотором нормированном пространстве к исходной функции [1].

Также в настоящее время наблюдается взрыв интереса к нейронным сетям, которые успешно применяются в самых различных областях – бизнесе, медицине, технике, геологии и т.д. Нейронные сети – мощная математическая модель, представляющая собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов), позволяющая воспроизводить чрезвычайно сложные зависимости[2].

Из вышесказанного следует, что для решения задачи аппроксимации функции можно использовать многослойную нейронную сеть. При проектировании сети главной и актуальной по сей день проблемой является выбор ее архитектуры. В общем случае она определяется сложностью и характером решаемой задачи. Таким образом, в зависимости от особенностей задачи выбираются параметры сети, основным из которых являются количество слоев и количество элементов (нейронов) в каждом слое.

Для исследования данных характеристик с помощью нейропакета NeuralNetworkToolbox (NNT) системы математического моделирования Matlab® была создана однонаправленная нейронная сеть, которая была обучена аппроксимировать исходную функцию синуса в условиях воздействия шума.

Количество слоев нейронной сети определяется сложностью решаемой задачи, количеством данных для обучения, требуемым количеством входов и выходов сети и ресурсами компьютера. Задача аппроксимации синусоиды является линейной и узконаправленной, поэтому в сети используются два слоя нейронов, что является достаточным условием для корректной работы данной сети.

Определим число нейронов в каждом слое. На вход сети подаем идеальный (незашумленный) вектор p . Его компоненты расположены на отрезке $[-1;1]$ с шагом 0,05 (выбранным экспериментально), то есть размерность входного вектора $n = 41$ [3].

Выходной вектор t выбираем таким образом, чтобы его компоненты соответствовали векторам входа. В нейропакете NeuralNetworkToolbox системы Matlab задание входного и выходного вектора записывается так:

$$p = [-1:0,05:1] \text{ входной вектор}$$

$$t = \sin(2\pi \cdot p) \text{ выходной (целевой) вектор}$$

Число нейронов в скрытом слое определим по формуле

$$\frac{mN}{1 + \log_2 N} \leq L_{\omega} \leq m \left(\frac{N}{m} + 1 \right) (n + m + 1) + m, \quad (1)$$

где n – число входных нейронов ($n = 1$), m – число выходных нейронов ($m = 1$), N – число элементов обучающей выборки ($N = 41$), L_{ω} – число синаптических весов.

Подставив соответствующие значения, получим:

$$6 \leq L_{\omega} \leq 127.$$

Оценив необходимое число весов, рассчитаем число нейронов в скрытом слое:

$$L = \frac{L_0}{n+m} \Rightarrow 3 \leq L \leq 62. \quad (2)$$

При выборе точного значения вышеупомянутого параметра была исследована зависимость между количеством нейронов в скрытом слое и эпохами обучения сети с заданным значением ошибки, которая представлена на рис. 1. Она отображает тот факт, что с увеличением данного параметра число эпох уменьшается.



Рис. 1. Зависимость эпох обучения сети для заданной ошибки от количества нейронов в скрытом слое

Однако после определенного значения числа нейронов (около 40) сеть теряет способность к обобщению: график аппроксимированной функции начинает вести себя непредсказуемо.

Экспериментально установлено, что для данной задачи лучшие результаты даёт двухслойная однонаправленная сеть, содержащая один нейрон во входном слое, 32 нейрона в скрытом слое и один нейрон в выходном слое. Результат аппроксимации синусоиды нейронной сетью с заданными характеристиками представлен на рис. 2.

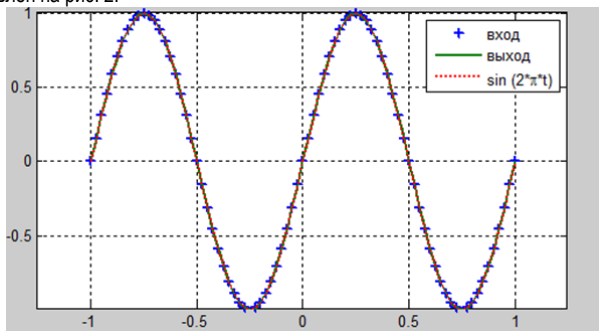


Рис. 2. Аппроксимация функции синуса с наилучшими характеристиками сети без воздействия шума

После исследования параметров сети и выявления их наилучшей комбинации, была создана и обучена нейронная сеть, не чувствительная к шуму. Шум – это случайные искажения, вносимые в значения аппроксимируемой функции в точках её определения. Заметим, что способность сети аппроксимировать исходную функцию при наличии шума, особенно важна для практики.

Чтобы добиться правильной аппроксимации зашумленной функции сеть была обучена с применением одного идеального и одного зашумленного векторов входа. Второй вектор имеет различные значения шума, распределенные по нормальному закону. Целевой вектор состоит из двух векторов:

$p_1 = [p \ p + \text{randn}(\text{size}(p)) \cdot u]$ – входной вектор, состоящий из идеального и зашумленного векторов;

$t_1 = [\sin(2\pi \cdot p) \ \sin(2\pi \cdot p)]$ – целевой вектор, состоящий из двух векторов.

Проверка функционирования системы производится следующим образом. Шум с диапазоном значений от 0 до 0,13 с шагом 0,01, распределенным по нормальному закону, добавляется ко входному вектору сети. Затем вычисляется ее выходной вектор сети и определяется его средний квадрат ошибки. Для того чтобы определить эффективность функционирования сети, построим соответствующий график зависимости среднего квадрата ошибки от уровня входного шума, который представлен на рисунке 3.

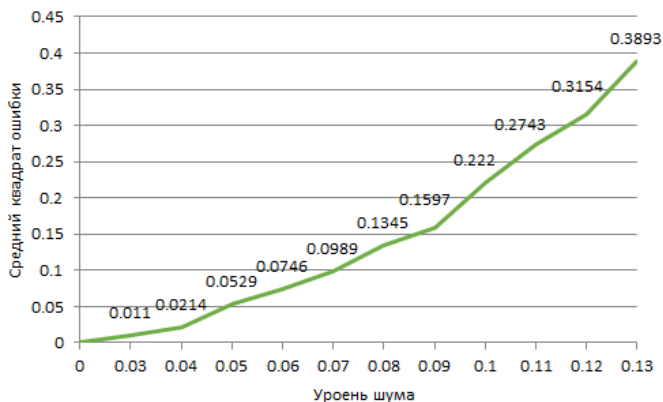


Рис. 3. Зависимость среднего квадрата ошибки аппроксимации от входного значения шума

Данный график показывает, что чем больше уровень входного шума, тем выше значение среднего квадрата ошибки. Однако на участке уровня зашумленности [0.03; 0.085] нейросеть демонстрирует меньшую чувствительность к помехам, чем на промежутке [0.086; 0.13].

На рис. 4 представлен результат аппроксимации функции синуса с достаточно высоким значением входным воздействием шума, равным 0.085.

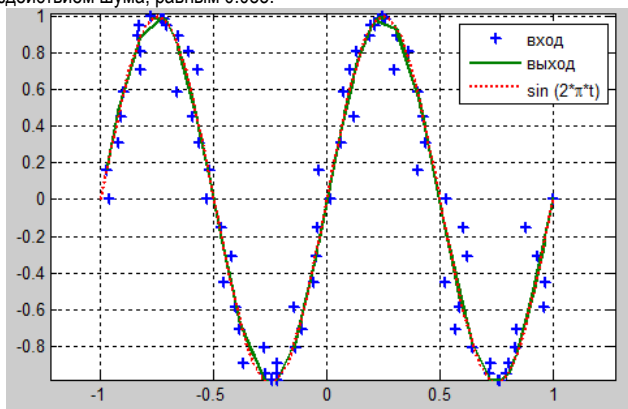


Рис. 4. Аппроксимация функции синуса с воздействием шума

Видно, что построенная аппроксимационная функция расположена вблизи изначально заданной функции, хотя и не точно повторяет ее.

Результаты тестирования показали, что данная нейронная сеть, обученная при наличии шума, способна обеспечить высокую степень аппроксимации функции, как и нейронная сеть, обученная при отсутствии шума.

Таким образом, данный метод создания сети и выбранная архитектура являются эффективными для аппроксимации функций в условиях наличия входных помех.

Библиографический список

1. StudFiles Файловый архив студентов [Электронный ресурс] – Режим доступа: <http://www.studfiles.ru/preview/5514386/page:46/>
2. *Филатова Т.В.* Применение нейронных сетей для аппроксимации данных // Вестник Томского государственного университета. 2004. № 284. С. 121-125
3. *Виноградова Ю.В., Ляхов А.Ф.* Практикум. – Нижний Новгород: Нижегородский госуниверситет, Аппроксимация функции нейронной сетью. 2009. 34 с.

УДК 629.735.33

С.И. Ковалева

студентка кафедры информационно-сетевых технологий

Т. В. Семененко

ст. преподаватель – научный руководитель

ЭТАПЫ СОЗДАНИЯ СИСТЕМЫ КЛАССА CRM

В современном мире необходимость автоматизации бизнес – процессов стала привычным явлением. Уже становится сложно представить себе складской или бухгалтерский учет без применения специализированного программного обеспечения, торговые представители используют специальные приложения для оформления и отправки заказа в офис прямо с планшета или смартфона, достаточно большая часть заказов приходит с сайта уже в виде готовых документов. Но при этом взаимоотношения с клиентами очень часто ведутся без внедрения автоматизации и достаточного внимания к учету.

Что происходит, если работа отдела продаж ведется без системы учета? Каждый менеджер по продажам работает так, как считает нужным, ведет фиксацию звонков, других видов взаимодействия с клиентами по собственному усмотрению.

Входящие звонки или заявки с сайта от новых заказчиков также не фиксируются, зачастую даже сложно понять, кто из менеджеров занимается входящей заявкой. В результате реальный учет ведется только на уровне оплаченных заказов и отгрузки товара. В случае увольнения или болезни сотрудника, все его неоконченные переговоры и необработанные контакты компания может потерять, что также крайне нежелательно для эффективной работы отдела продаж. Выход из этой ситуации – автоматизация и стандартизация управления отношений с клиентами, т.е. создание CRM-системы.

CRM-система (Customer Relationship Management) – информационные технологии, позволяющие формализовать и автоматизировать различные аспекты взаимодействия с клиентами подразделений маркетинга, продаж и сервисного сопровождения на основе автоматических/автоматизированных процессов (в том числе сбытовых) и единого «информационного пространства» организации [1].

CRM-система может включать:

- фронтальную часть, обеспечивающую обслуживание клиентов на точках продаж с автономной, распределенной или централизованной обработкой информации;
- операционную часть, обеспечивающую авторизацию операций и оперативную отчетность;
- хранилище данных;
- аналитическую подсистему;
- распределенную систему поддержки продаж: реплики данных на точках продаж или смарт-карты.

Рассмотрим этапы проектирования CRM:

1. Предпроектное исследование.
2. Проектирование, создание технического задания.
3. Разработка.
4. Внедрение.
5. Аудит проекта.

Остановимся на первых трех этапах, связанных с проектированием и разработкой системы, и изучим их подробнее:

Первый этап необходим для оценки потребностей компании. Цель этапа – детально проанализировать текущее состояние бизнеса в области управления взаимоотношениями с клиентами, выявить проблемные участки и понять желаемое состояние бизнеса. Этап предполагает серию интервью, проводимых членами проектной команды с руководителями и сотрудниками различных департаментов компании-заказчика. На этапе предпроектного исследования выясняются следующие вопросы:

- организационная структура компании;
- продукты (услуги) компании;
- существующие классификации и характеристики клиентов;
- портреты клиентов;
- каналы сбыта продукции;
- инструменты маркетинга;
- основные бизнес-процессы отделов (департаментов);
- используемые аналитические инструменты, отчеты.

В ходе предпроектного исследования выявляются проблемные ситуации. Формируется список целей и задач, описывающих желаемую ситуацию. Цели внедрения необходимы для оценки результатов проекта внедрения CRM-системы по его завершению и на этапе аудита проекта.

Результатом предпроектного исследования является отчет, который описывает, что необходимо реализовать в системе – «Что нужно сделать?». На этом этапе не нужно пытаться построить детальный дизайн системы и детально прописывать способы реализации. Это необходимо делать на следующем этапе – при подготовке «Технического задания», которое отвечает на вопрос «Как нужно делать то, что описано в отчете о предпроекте?».

На основе информации, собранной в рамках первого этапа, создается план проекта. В дальнейшем план будет уточняться по мере продвижения от одного этапа проекта к следующему.

В рамках второго этапа проводится проектирование внедряемой CRM-системы.

Результатом этапа проектирования является «Техническое задание», который включает в себя способы реализации требований компании-заказчика к внедряемой CRM-системе, описанных в «Отчете о предпроектном исследовании». Оно включает в себя все требования к системе. Выбирается методология разработки, наиболее популярная сейчас – SCRUM – гибкая методология разработки, основной концепцией которой является постоянная проверка хода работ и постоянно выяснение того, что хочет заказчик, и соответствует ли его ожиданиям выполнение проекта [2].

«Техническое задание» должно быть понятно как постановщику задания, так и программисту, и не должно содержать многозначностей.

После написания «Технического задания» можно переходить к третьему этапу – разработке.

Цели третьего этапа – разработка, настройка и тестирование программы согласно «Техническому заданию». Параллельно с настройкой программы готовится документация по эксплуатации системы (инструкции пользователей). По мере готовности модулей системы консультанты должны проводить согласование и утверждение разработанных модулей с заказчиком.

Этап разработки состоит из следующих шагов.

- Встраивание.

CRM-система может быть встроена в существующую систему управленческого учета на базе различных продуктов, например, «1С». Между CRM-системой и «1С: Бухгалтерией» разрабатывается механизм обмена информацией: синхронизация справочников «Контрагенты», «Номенклатура», документов «Счет».

- Программирование.

На данном шаге программист выполняет разработку функционала и реализует CRM – систему. Работы выполняются в соответствии со способами реализации, изложенными в «Техническом задании».

- Настройка.

Цель шага – подготовить систему для работы пользователей. На данном этапе настраиваются основные параметры учета, заполняются справочники (создается необходимая аналитика), проводится настройка интерфейсов, определяются роли (наборы прав доступа для групп пользователей), регистрируются пользователи системы, назначаются интерфейсы и права доступа для пользователей.

Значения данных для настройки должны быть описаны в «Отчете о предпроекте» и «Техническом задании».

- Разработка эксплуатационной документации.

Цель данного шага – разработать для всех категорий пользователей понятные инструкции по работе с CRM-системой.

- Тестирование.

Цель шага – проверить работоспособность всех реализованных модулей и блоков CRM-системы. Примеры задач для тестирования описываются и согласовываются на этапе проектирования.

Выполненные настройки должны соответствовать требованиям «Технического задания». В ходе выполнения тестирования выявляется список несоответствий и согласовывается порядок их устранения.

На этом шаге разработка заканчивается.

Результатом выполнения всех трех этапов является готовая CRM-система и подготовленная эксплуатационная документация. Можно переходить к внедрению и аудиту проекта.

Цель достигнута – создание системы класса CRM завершено. Точнее, завершена первая итерация системы, которая может дорабатываться под новые нужды пользователей.

Библиографический список

1. *Осипов Л. А., Яковлев С.А.* Введение в системную инженерию. СПб.: ГУАП, 2016. 130 с.
2. *Сазерленд Джефф.* Scrum. Революционный метод управления проектами / пер. с англ. М. Гескиной. М.: «Манн, Иванов и Фербер», 2017. 272 с.

УДК 004.057

В. Ю. Коваленко

студент кафедры технологий защиты информации

К. А. Жиданов

научный руководитель

SQL-ИНЪЕКЦИИ И МЕТОДЫ ЗАЩИТЫ ОТ НИХ

База данных (БД) – именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области [1].

Развитие технологии «клиент-сервер» привело к тому, что БД стали практически обязательным компонентом современных информационных систем, и, как следствие, появилось большое количество серверов баз данных, часто подвергающихся атакам злоумышленников [2].

SQL (Structured Query Language) – стандартный язык запросов для работы с реляционными БД. Язык SQL был разработан в 70-х годах компанией IBM. В дальнейшем, получив широкое распространение, стал стандартом для языков манипулирования данными в реляционных системах управления базами данных (СУБД) [1]. SQL-инъекция – встраивание вредоносного кода в запросы к базе данных [3]. Уязвимостям типа SQL-инъекция подвержены любые клиент-серверные и сервис-ориентированные приложения, работающие с базами данных. Однако зачастую SQL-инъекции рассматривают относительно WEB-приложений. Согласно данным открытого исследования «Positive Technologies» за 2016 г., SQL-инъекции являются самым распространенным видом атак на WEB-приложения, и составляют 84% всех атак. В данной работе анализируется принцип действия SQL-инъекций и организация программной защиты WEB-приложений.

В большинстве случаев такая атака подразумевает ввод собственных SQL-выражений через поля форм или URL-строку в целях изменения логики конечного запроса к СУБД.

Данный подход нацелен на недостаточную обработку входных параметров, полученных из пользовательского ввода на стадии конструирования SQL-запроса к базе данных, отсутствие фильтрации, уязвимости в базе данных сервера и условные ошибки. В целях большей эффективности SQL-инъекция может комбинироваться с другими типами атак. Например, с атакой типа XSS и использовать пользовательские COOKIE для встраивания SQL.

Также существует множество специализированного программного обеспечения для автоматизированного поиска и эксплуатации найденных уязвимостей, которое особо опасно для проектов с открытым исходным кодом. Например, SQLMAP, поддерживающий эксплуатацию всех основных уязвимостей [4].

Следствием такой атаки может стать как получение дополнительных привилегий или несанкционированного доступа к конфиденциальной информации других пользователей, так и выполнение произвольных команд операционной системы на атакуемом сервере. Это в свою очередь приведет к полной компрометации приложения.

Базовой защитой от SQL-инъекций является жесткая фильтрация вводимых пользователем строковых и числовых параметров.

Однако данных мер будет недостаточно при использовании злоумышленником дополнительных техник, таких как:

1. Инъекция нульбайта;
2. Смешанное содержимое;
3. Встроенные комментарии;
4. Раздробленные запросы (chunked);
5. Переполнение буфера;
6. HTTP Parameter Pollution;
7. URL encoding (hex);

8. Разделение или дублирование ключевых слов;
9. Сброс сессионной COOKIE;
10. Header injection;

А также их различные комбинации. Вследствие чего такая защита будет являться неполной и поставит под угрозу безопасность WEB-приложения.

Для целостной защиты необходимо обеспечить:

1. Фильтрацию и усечение строковых параметров;
2. Использование параметризованных SQL-запросов;
3. Использование функций блокировки и хранимых процедур;
4. Создание менее привилегированного пользователя;
5. Контроль сообщений об ошибках;
6. Применение регулярных выражений;
7. Ограничения максимального значения;
8. Скрытие структуры БД;
9. Использование White List.

Данные меры следует использовать комплексно, чтобы избежать возможных угроз. Все запросы, использующие внешние данные, требуется защитить от SQL-инъекций. Также следует учитывать, что внешние данные могут быть переданы не только в качестве GET-параметров, но и методом POST, взятые из COOKIE, со сторонних сайтов или из базы данных, в которую пользователь имел возможность занести информацию.

Все числовые параметры следует явно преобразовывать в числовой вид с помощью функций.

Сложность SQL-запроса не будет определять его устойчивость. Особенно это относится к открытым проектам, исходный код которых является публичным. Т.к. уязвимости в них также можно найти автоматизировано.

Рекомендуется проводить тестирование приложения на возможность проведения SQL-инъекций, например, тестирование «черный ящик», а также использовать специальные утилиты для их автоматизированного поиска:

1. Статические кодовые контроллеры (JDBC-Checkers);
2. Статический и динамический анализ (AMNESIA, SQLGuard, SQLcheck);
3. Анализ логических потоков (WebSSARY);
4. Инкапсуляция SQL-запросов (SQLDOM, Safe Query Object).

Библиографический список

1. Карлова Т.С. Базы данных: модели, разработка, реализация. СПб.: Питер, 2001. С. 66, С. 20.
2. Информационные и телекоммуникационные технологии: образование, наука, практика: Труды Международной научно-практической конференции. – 2012. – Т. 2. – С. 245-248.
3. Егоров М. Выявление и эксплуатация SQL-инъекций в приложениях. // Эшелон. 2011. № 2.
4. Журнал «Хакер», [Электронный ресурс] URL: <https://haker.ru/2011/12/06/57950>.

УДК 004.056.5

А. В. Кондратов

студент кафедры технологий защиты информации.

С. В. Беззатеев

д-р техн. наук, профессор – научный руководитель

ПРОБЛЕМЫ УТЕЧКИ ДАННЫХ ПРИ ПЕРЕДАЧЕ ПО ЭЛЕКТРОМАГНИТНЫМ КАНАЛАМ

В 1996 году американский ученный Пол Кочер опубликовал ряд статей о целом классе атак – атаки по побочным каналам. Хотя истории известны более ранние случаи атак по побочным каналам, но всемирную известность данный класс атак получил, именно, благодаря Кочеру [1].

Атаки по побочным каналам – класс атак, направленный на недочеты в практической реализации криптосистем. До 1996 года было приятно считать криптосистему, как «черный ящик» к которому нет доступа и на него невозможно никакое механическое воздействие, поэтому всё внимание было приковано, непосредственно, к криптоалгоритмам, как математическим объектам. В следствии чего, реализации криптосистем уделялось не так много времени. Но после ряда публикаций Пола Кочера, криптосистемы стали проектировать таким образом, что бы они могли противостоять атакам по побочным каналам [1].

Атаки по побочным каналам реализуются во время мощных атак против криптографических реализаций, и производятся они ради примитивов, протоколов, модулей и устройств файловых систем [2]. Данные атаки представляют серьезную угрозу для безопасности криптографических модулей. Данный вид атак является наиболее эффективным средством получения информации, чем классический криптоанализ. В результате нужно оценить криптографические реализации с точки зрения сопротивления таким атакам и рассмотреть введение различных контрмер [3].

Атаки по побочным каналам можно классифицировать:

По методу доступа к системе как:

Агрессивные атаки используются при получения прямого доступа к внутренним компонентам криптографических модулей и устройств. Полуагрессивные атаки используются для доступа к устройству без нарушений пассивирующего слоя или без прямого контакта с криптографическим модулем.

Неагрессивные атаки используются без какого-либо воздействия на криптосистему, собирая только внешне доступную информацию о работе системы [1].

По контролю над вычислительным процессом как:

Пассивные – при атаке, злоумышленник получает необходимую информацию без заметного влияния на работу системы.

Активные – при атаке, злоумышленник производит воздействие на систему, в результате чего, может изменить поведение данной системы [1].

По методу применяемому в процессе анализа:

Простые – методы отслеживания зависимостей между полученной информацией и процессами происходящими в криптосистеме во время ее получения.

Разностные – статистические методы отслеживания зависимостей между входными данными и полученной, в ходе атаки, информации [1].

После рассмотрения классификаций атак по побочным каналам, можно перейти к рассмотрению типов атак на электромагнитные каналы. Наиболее известный тип атак на электромагнитные каналы – атака по электромагнитному излучению. Работа любой электронной криптосистемы вызывает электромагнитное излучение [4]. Злоумышленник может перехватить данное излучение, проанализировав его, он может получить информацию о криптосистеме в целом, а так же заполнить информацию о секретном ключе.

Электромагнитное излучение возникает во время работы модулей криптосистемы, таких как – модуль управления, модуль ввода-вывода или в других модулях. Данное излучение может быть пред-

намеренным или непреднамеренным. Каждый элемент проводящий ток в системе, не только производит собственное излучение, но так же влияет на излучения других элементов из-за связи или расположения схемы [5]. Из этих многочисленных излучений наибольшую пользу несут те, которые вызваны операциями обработки данных. Чаще всего прямое излучение на высоких частотах, поступающие от системы, является одной из главных целей для агрессивных активных простых атак. Но данные атаки легко обнаруживаются и современные системы способны скрывать источник сигнала или же отфильтровывать его. При непреднамеренных излучениях, которые возникают в системе при непосредственной близости проводников тока, что приводит к связи между электромагнитными полями [5]. Такие сигналы являются целью для полугрессивных пассивных разностных атак, атаки на эти сигналы требуют высокого качества оборудования для сбора данных, так же их почти невозможно отследить, в следствии чего они являются наиболее опасными для атак на электромагнитные каналы.

Рассмотрим атаку на электромагнитные каналы, на примере, который был предоставлен исследовательским центром «IBM Watson». Атака на прямое электромагнитное излучение. Используется считыватель смарт-карты, ручной зонд (считывающий диапазон частот до 500МГц) для считывания прямого излучения и металлический лист соразмерный считывателю смарт-карты. Металлический лист располагается позади считывателя, к коаксиальному кабелю прикрепляется зонд. Металлический лист усиливает излучение, испускаемое внутри считывателя, в то время как зонд считывает полученное электромагнитное излучение. Далее прикладываются смарт-карты, которые есть в базе считывателя, после чего зонд с полученной информацией удаляется вместе с металлическим листом. Таким образом были получены статистические данные, о том, как обрабатывается информация, полученная от смарт-карты. На низких частотах был уловлен сигнал от чипа смарт-карты. Эти данные, которые были получены с помощью зонда, дали возможность ученым изучить алгоритм обработки информации, что позволило, взломать считыватель смарт-карты [5]. Для того, что бы защитить систему, от подобной атаки следует внутри считывателя, установить генератор шума, который при работе системы выделял дополнительный шум, что бы злоумышленник не мог получить прямое излучение или же получил неправильные данные о работе системы.

Для того, что бы уберечь информацию от злоумышленника, требуется не только эффективный шифрующий алгоритм, а так же нужно построение криптосистемы для передачи данных, которая будет в свою очередь уберечь информацию от атак по побочным каналам». Что бы уберечь криптосистему от атак на электромагнитные каналы, нужно : правильно спроектировать схему модулей, так что бы проводники тока не оказывались близко друг к другу, так же требуется, что бы внутри системы были установлены блокираторы, которые в свою очередь при нарушении целостности системы отключали ее, так же необходим фильтр для сглаживания электромагнитных излучений или шумовой генератор. Все эти меры, не гарантируют полную безопасность криптосистемы, но повышают ее криптостойкость при атаках по побочным каналам.

Библиографический список

1. Zhou Y. B., Feng D. G. Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing //ACR Cryptology ePrint Archive. – 2005. – Т. 2005. – С. 388.
2. Жуков А. Е. Криптоанализ по побочным каналам (side channel attacks) //Защита информации. Инсайд. – 2010. – №. 5. – С. 28-33.
3. Онуфриева Т. А., Шиманова К. В. Об особенностях анализа защищенности информационной системы, реализованной на ПЛИС // сб. ст. Междунар. науч.-практ. конф. «Инновационное развитие современной науки».Уфа. Т. 1. 2015. – С. 70-73.
4. Carlier V. et al. Electromagnetic side channels of an FPGA implementation of AES //CRYPTOLOGY EPRINT ARCHIVE, REPORT 2004/145. – 2004.С. 2-7.
5. Agrawal D. et al. The EM side-channel (s): attacks and assessment methodologies. 2002. 2012. С. 4-30.

УДК 004

Т.А. Минаева

студент кафедры технологий защиты информации

Н.В. Волошина

канд. техн. наук, доцент – научный руководитель

АНАЛИЗ RS-МЕТОДА СТЕГАНОАНАЛИЗА ДЛЯ BMP-ИЗОБРАЖЕНИЙ

Оцифровка данных привела к революции в области связи, что сделало обмен данными очень быстрым и удобным. Поскольку Интернет, используемый в качестве среды для обмена данными, является открытым источником, таким образом проблемы безопасности, такие как модификация и перехват данных, становятся актуальными. Были разработаны несколько подходов для безопасного обмена данными в Интернете. Криптография является одним из таких методов. Суть этого метода в том, что секретная информация преобразуется в неизвестную форму, таким образом, что только пользователь, у которого есть секретный ключ, может преобразовать ее обратно в исходную информацию. Но в некоторых ситуациях мало превратить секретную информацию в неизвестную форму, необходимо полностью скрыть факт передачи секретной информации по каналу связи. Стеганография – один из методов, который выполняет данное требование. Одним из способов организации скрытого канала связи является внедрение информационных сообщений в цифровые объекты, свободно распространяемые по сети. В качестве таких объектов выступают цифровые изображения, аудио- и видеофайлы и т.д. [1].

Одновременно с появлением стеганографии появились методы стеганоанализа, позволяющие выявить скрываемую информацию в цифровых объектах (контейнерах). На сегодняшний день самым распространенным, но наименее стойким к обнаружениям методом стеганографического сокрытия информации в изображениях, является метод замены наименее значимых бит (НЗБ, LSB). Идея метода заключается в замене от одного до четырех битов в байтах цветового представления точек изображения битами скрываемого сообщения. Такой метод применяется к растровым изображениям не прошедшим компрессию, например, изображения формата BMP.

В статье [2] представлена следующая классификация стеганоаналитических методов:

- 1) в зависимости от количества информации, доступной аналитику: направленные, универсальные;
- 2) по критерию цели атаки: статические, динамические, вспомогательные;
- 3) в зависимости от объекта поиска в контейнерах: визуальные, сигнатурные, статистические.

В данной статье будет рассмотрен один из статистических методов – RS-метод, впервые опубликованный в 2001 г. коллективом ученых под руководством Дж. Фридрих. Сокращения в названии расшифровывается как Regular-Singular, что с перевода означает «регулярно-сингулярный».

Для оценки эффективности RS-метода было реализовано приложение на языке программирования C++, осуществляющее анализ BMP-изображения на наличие скрытой информации. Скрываемая информация, представляющая собой псевдослучайный набор нулей и единиц, заносилось в изображение последовательно в наименее значимые биты. Для тестирования были взяты восемь различных изображений размером 256×256 пикселей.

Перед тем как будет рассмотрен статистический метод, необходимо оценить уровень шума, создаваемый встраиванием в НЗБ BMP-изображения, которое было использовано при реализации и оценке RS-методом стеганоанализа.

Для того чтобы найти уровень искажения изображения (PSNR) необходимо вычислить абсолютную оценку – среднеквадратичное отклонение реального сигнала, описывающего изображение, от полезного:

$$RMSE = \sqrt{\frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |I(i, j) - K(i, j)|^2}{mn}},$$

где $RMSE$ – среднеквадратичное отклонение реального сигнала, описывающего изображение, от полезного; $I(i, j)$ – значение элемента заполненного контейнера; $K(i, j)$ – значение элемента «пустого» контейнера; $m \times n$ – количество элементов в контейнере.




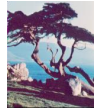


Используя среднеквадратичное отклонение можно вычислить относительную оценку – пиковое отношение полезного сигнала к шуму PSNR, измеряемое в децибелах:

$$PSNR = 20 \log_{10} \left(\frac{A_{\max}}{RMSE} \right),$$

где $PSNR$ – пиковое отношение полезного сигнала к шуму; A_{\max} – максимальное значение сигнала (максимальное значение, принимаемое пикселем изображения, для полноцветного BMP-изображения максимальное значение будет 255); $RMSE$ – среднеквадратичное отклонение реального сигнала, описывающего изображение, от полезного.

Таблица 1

Сравнение PSNR (дБ) тестовых изображений при разном проценте встраивания информации

Процент встраивания	Тестовое изображение					
						
	ball.bmp	plane.bmp	pepper.bmp	tree.bmp	clock.bmp	lena.bmp
5%	45,1346	44,5542	44,9678	44,7167	44,958	44,7229
10%	41,9956	41,568	41,745	41,7157	41,8224	41,7722
30%	37,1384	36,9255	36,9522	37,025	37,1936	37,0573
50%	34,8804	34,8255	34,8034	34,853	35,3228	34,8518
70%	33,3201	33,4147	33,3389	33,391	34,0096	33,3443
100%	31,894	31,8562	31,8348	31,7595	32,2365	31,7726

В табл. 1 приведены значения PSNR при разном проценте встраивания информации в тестовые BMP-изображения путем замены наименее значимых бит (четыре последних бита в цветовых компонентах каждого пикселя изображения). Как видно из таблицы, после встраивания значения PSNR не превышают 50 дБ. Так как типичные значения PSNR для сжатия изображений лежат в пределах 30-40 дБ, то даже при полном заполнении изображения оно не опускается ниже 30 дБ. Так как PSNR это отношение полезного сигнала к шуму, то, чем выше значение PSNR, тем меньше шума создает встраивание.

Суть RS-метода заключается в том, что все изображение разбивается на группы по n значений цветовых компонентов пикселей $G(x_1, x_2, \dots, x_n)$, находящихся рядом по горизонтали, где n четное. Для каждой группы определяется функция регулярности $f(G)$, рассчитанная как сумма перепадов смежных значений, где значения представляют собой целые числа от 0 до 255.

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|,$$

где $f(x_1, x_2, \dots, x_n)$ – функция регулярности; x_i – значение элемента группы; n – количество элементов в группе.

Далее определяются две функции флиппинга $F(x)$: F_1 , представляющая собой инверсию младшего бита значения; F_{-1} , представляющую собой инверсию с переносом в старший бит.

$$F_1 = \begin{cases} x+1, & 0 \leq x \leq 254 \\ 0, & x = 255 \end{cases} \quad F_{-1} = \begin{cases} x-1, & 1 \leq x \leq 255 \\ 255, & x = 0 \end{cases}$$

При применении флиппинга к каждому значению группы получаются две преобразованные группы значений, после чего все группы делятся на классы следующим образом:

- регулярные группы: $G \cup R \cup f(F(G)) > f(G)$;
- сингулярные группы: $G \cup S \cup f(F(G)) < f(G)$;
- неиспользуемые группы: $G \cup U \cup f(F(G)) = f(G)$.

В дальнейшем исследуется соотношение между группами в изображении. Для этого определяется количество групп, попавших в тот или иной класс: R_1, S_1, U_1 и R_{-1}, S_{-1}, U_{-1} . Индексы у классов обозначают применение F_1 и F_{-1} функций флиппинга для получения распределения.

Метод основывается на статистическом предположении, что для естественного изображения, которое не подвергалось встраиванию информации, характерно следующее: $R_1 \cong R_{-1}$ и $S_1 \cong S_{-1}$. Предполагается, что применение функции флиппинга F_{-1} даст тоже распределение, что и функция F_1 на изображении, значение компонентов пикселей которого сдвинуты на единицу. Для изображения без встраивания соотношение между группами не должно существенно меняться, а значительное расхождение свидетельствует о применении НЗБ-встраивания.

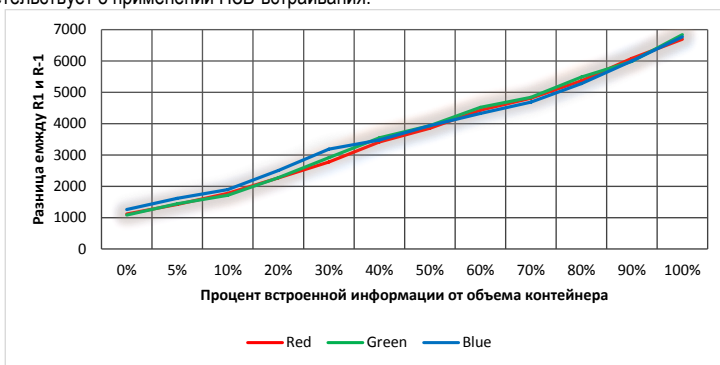


Рис. 1. График зависимости разницы между классами регулярной группы от процента встраиваемой информации

Для анализа было взято изображение размером 256×256 . Так как взято 24-битное BMP-изображение, то считались группы по каждому компоненту отдельно. На графике представлена разница регулярного класса между R_1 и R_{-1} , зависящая от процента встраиваемой информации. Видно, что при 100%-м заполнении изображения информацией разница между R_1 и R_{-1} составляет около 7000, учитывая, что количество групп было 32768. Это подтверждает, что в изображение была внесена информация с помощью замены НЗБ. Но при малом проценте встраивания информации очень сложно предположить

RS-метод реагирует на естественный шум в изображении, поэтому очень сложно предположить объем встраиваемой информации в контейнер. При низком проценте встраивания информации в изобра-

жение невозможно предположить о наличии в изображении стегановставок, при высоком проценте встраивания – это существенно заметно.

Данный метод показал хорошие результаты от среднего процента встраивания информации в изображениях различных типов: пейзажей, портретов, натюрмортов, черно-белых изображений и т.д. Путем создания искусственного шума в изображении, RS-метод был проверен на возможность ложного срабатывания, но вызвать его не удалось. Так как с помощью этого метода невозможно определить встроена ли информация при маленьком проценте встраивания, то целесообразно использовать метод в комплексе с другими методами стеганоанализа как статистическими, так и методами из других категорий.

Библиографический список

1. P. Mohamed Fathimal, P. Arockia Jansi Rani, «K Out of N Secret Sharing Scheme with Steganography and Authentication», in Computational Intelligence, Cyber Security and Computational Models, Springer, 2016, pp. 413–425.
2. Генне О.В. Основные положения стеганографии // Защита информации. Конфидент. 2000. №3. С.20-25.
3. Грибунин В.Г. Цифровая стеганография. СПб.: СОЛОН-Пресс, 2002. 280 с.
4. Аграновский А.В., Балакин А.В., Грибунин В.Г., Сапожников С.А. Стеганография, цифровые водяные знаки и стеганоанализ: Монография. М.: Вузовская книга, 2009. 220 с.
5. Fridrich J., Du R., Meng L. Steganalysis of LSB Encoding in Color Images, ICME 2000, New York City, July 31–August 2, New York.
6. Лапшенков Е.М. Незатонная оценка уровня шума цифрового изображения на основе гармонического анализа//Компьютерная оптика. 2012. Т. 36. №3. С. 439-447.

УДК 004.056

И. А. Миронова

студент кафедры технологий защиты информации

С. В. Беззатеев

д-р техн. наук, профессор – научный руководитель

TOUCH ID/ДАКТИЛОСКОПИЧЕСКИЕ ДАТЧИКИ

Использование биометрических технологий в информационных технологиях началось довольно давно. Уже в 1998 году Siemens создали прототип мобильных телефонов, обеспечивающих защиту аппарата сканированием пальца пользователя [3]. Но только в настоящее время биометрические механизмы аутентификации в смартфонах стали популярными. В 2013 году компания Apple выпустила iPhone 5s, в котором появился дактилоскопический сканер. После этого большинство мобильных телефонов, таких как iPhone 5s / 6/ 6+ / SE / 7, Samsung Galaxy S5 / S6, HTC One Max, Huawei Honor 7, Meizu MX4 Pro и другие, теперь выпускаются со встроенным датчиком для разблокировки телефона. Предсказывается, что 50% смартфонов, проданных к 2019 году, будут иметь встроенный датчик отпечатков пальцев [1].

Защита устройства является важной функцией и может повысить вероятность того, что пользователь не оставит устройство незащищенным. Touch ID позволяет разблокировать телефон одним касанием пальца, что делает этот процесс гораздо быстрее и проще, чем ввод пароля, поэтому пользователь скорее всего будет использовать защиту своего смартфона.

С введением Apple Pay распознавание отпечатков пальцев на мобильных устройствах используется не только для разблокировки устройства, но и для безопасного мобильного платежа и других транзакций. Соответственно, вопрос о безопасности биометрической защиты стоит остро: может ли пользователь быть уверен в том, что система не будет взломана или что при потере устройства, его данными не смогут воспользоваться?

На официальном сайте Apple следующее описание дактилоскопического датчика. Кнопка выполнена из кристалла и выступает в роли линзы, фокусирующей на пальце. Стальное кольцо, окружающее кнопку, обнаруживает палец и отправляет датчику Touch ID команду на начало чтения отпечатка. В датчике используется емкостная сенсорная панель, позволяющая получить изображение высокого качества для мелких сегментов субэпидермальных слоев кожи. Детали отпечатка пальца распределяются по трем основным категориям – дуга, кольцо или завиток. Отмечаются детали гребешков, которые не видны человеческому глазу, и даже распознаются мелкие отклонения в направлениях гребешков, создаваемые порами и структурой краев. Затем математическое представление отпечатка пальца шифруется, причем ключи шифрования, используя эллиптические кривые, вычисляются во время загрузки устройства на основе уникального аппаратного ключа и кода блокировки, который вводит пользователь [7].

В среде сенсора отпечатков пальцев используются компоненты аппаратной безопасности для надежного хранения биометрического шаблона и поддержки использования безопасной среды исполнения (*Trusted Execution Environment, TEE*) для верификации отпечатков. В чипе Apple A7 используется сопроцессор «Secure Enclave», который используется для хранения всех данных отпечатка пальца, полученных от Touch ID [6].

Система «Secure Enclave» разработана для защиты данных паролей и отпечатков. Данные об отпечатках шифруются и защищаются ключом, доступным только для «Secure Enclave». Они используются исключительно системой и только для проверки совпадения отпечатка с сохраненными данными. Система отделена от остальной части процессора и операционной системы iOS, соответственно, эти данные не попадают в iCloud и не передаются на сервер компании. При этом система время от времени удаляет данные отпечатков даже из оперативной памяти устройства, вынуждая пользователя авторизоваться с помощью кода блокировки, который и возобновляет работу датчика.

Все выше перечисленное делает Touch ID на iPhone надежной системой. Даже, несмотря на то, что отпечатки пальцев не хашируются, но из-за хранения в «Secure Enclave» данные не покидают устройство и не хранятся в незашифрованном виде.

Какие же проблемы безопасности возникают при использовании Touch ID? Для начала, проблемы спуфинга и перебора – это уязвимости, которые влияют на все биометрические устройства. Злоумышленник, украв смартфон жертвы, все-таки может получить доступ к данным.

Суть спуфинга (spoofing – подмена) заключается в том, что поддельный отпечаток может быть получен снятием отпечатка пальца жертвы с предмета, который она трогала, и скопирован с помощью специальных материалов [5].

Например, через несколько дней после релиза iPhone 5S Chaos Computer Club – немецкое общество хакеров взломали сенсор, сняв подлинный отпечаток владельца с поверхности стакана, создали ложный отпечаток, напечатав его с очень высоким разрешением.

В 2016 году руководитель Vkansee – компания, которая также занимается созданием дактилоскопических датчиков, демонстрирует разблокировку iPhone, при помощи слепка отпечатка пальца, сделанного из детского пластилина Play-Doh. Для создания слепка Vkansee использовали специальный стоматологический материал. После того, как слепок владельца был снят и затвердел, к вогнутому вовнутрь слепку приложили пластилин и получили дактилоскопический рисунок. Таким образом, получили имитацию пальца для разблокировки аппарата [2].

Проблема перебора или метод «грубой силы» можно разделить на две группы: физический и цифровой.

Физический метод «грубой силы» использует уязвимость сенсора к шумам. Взломщик может использовать мигающий свет или лазер, чтобы перезагрузить сенсор и, затем, получить доступ.

Цифровой метод делится также на две группы. Во-первых, это может происходить аналогично взлому пароля перебором. Так как биометрия должна сравнивать шаблон человека с множеством других шаблонов, то это позволяет злоумышленнику просто отправить неограниченное количество данных на сенсор, пока он не получит правильную комбинацию. Во-вторых, это может быть выполнено путем анализа энергопотребления устройства, когда он использует свой биометрический датчик. Данная слабость позволяет злоумышленнику потенциально обнаружить программный код и даже попытаться разорвать шифрование с помощью собранной информации [5].

Также технический документ по безопасности Apple, оценивая безопасность Touch ID, отмечает, что телефон будет более защищенным по гораздо более сложному паролю, который будет вводиться реже (потому что Touch ID используется на его месте) [7]. Это важное наблюдение, хотя в настоящее время iOS все еще допускает относительно небезопасные 4-значные коды доступа. До тех пор, пока программное обеспечение не начнет требовать от пользователей сложные буквенно-цифровые коды, простые коды доступа все еще будут самым слабым звеном, а дактилоскопический датчик может быть относительно безопасным.

Биометрия является мощным механизмом аутентификации, но ее текущая реализация на мобильных устройствах является неполной и небезопасной. Она в настоящее время больше восхваляется за её удобство по сравнению с ручным вводом пароля, а не из-за её повышенной гарантии безопасности. Инженерные усилия производителей смартфонов создали отличную основу для безопасной обработки цифровых биометрических данных. Теперь те же усилия должны быть направлены на программное обеспечение. Как вариант, для усиления безопасности должна быть сделана двухфакторная аутентификация [4]. В итоге, хотя такой способ использования дактилоскопического датчика не лишен недостатков, биометрическая аутентификация имеет потенциал для развития, потому что это быстрый и естественный способ для людей пользоваться персональным идентификатором на основе сканирования пальца.

Библиографический список

1. Cao K., Jain A. K. Hacking mobile phones using 2D Printed Fingerprints. – MSU Technical report, MSU-CSE-16-2, 2016.
2. Chacksfield M. Watch Touch ID get fooled with some Play-Doh // Techradar.com, 25.02.2016.

[Электронный ресурс] – Режим доступа. – URL: <http://www.techradar.com/news/phone-and-communications/watch-touch-id-get-fooled-with-some-play-doh-1315801> (дата обращения: 29.03.2017)

3. Goode A. Bring your own finger—how mobile is bringing biometrics to consumers //Biometric Technology Today. – 2014. – Т. 2014. – №. 5. – С. 5-9.

4. King A. Biometric Security in the Mobile Age. – 2016.

5. Tipton S. J. et al. iOS security and privacy: Authentication methods, permissions, and potential pitfalls with touch id //International Journal of Computer and Information Technology. – 2014. – Т. 3. – №. 03.

6. Афонин О. Сканер отпечатка пальца: безопасность и обход защиты // Хакер.ру, 24.10.2016 [Электронный ресурс] – Режим доступа. – URL: <https://хакер.ру/2016/10/24/fingerprint-scanners/> (дата обращения 28.03.2017)

7. Сведения о защите Touch ID на iPhone и iPad // Support.apple.com, 10.11.2015 [Электронный ресурс] – Режим доступа. – URL: <https://support.apple.com/ru-ru/HT204587> (дата обращения 28.03.2017)

УДК 004.056.5

С. В. Нестерук

студент кафедры технологий защиты информации

С. В. Беззатеев

д-р техн. наук, доцент – научный руководитель

АНАЛИЗ МЕТОДОВ ЛИНГВИСТИЧЕСКОЙ СТЕГАНОГРАФИИ

Стеганография – это наука и искусство сокрытия информации путём внедрения одного сообщения в другое (контейнер). Стеганография в отличие от криптографии не шифрует информацию, а скрывает сам факт её передачи [1].

Ограничение криптографии состоит в том, что третья сторона всегда знает о факте передачи данных из-за неразборчивости текста. Стеганография в свою очередь прячет секретное сообщение в контейнере, не вызывающим подозрения [2].

Цель данной работы заключается в описании методов лингвистической стеганографии, сравнении их характеристик и оценке возможности их применения на практике.

Выделяют такие характеристики стегосистемы как:

1. Полезная нагрузка – отношение скрытой информации к размеру контейнера.
2. Взломоустойчивость – способность системы противостоять изменениям в контейнере.
3. Естественность – степень неотличимости контейнера от обычного текста [1].

Компромисс между полезной нагрузкой и естественностью является фундаментальным, так как любая попытка спрятать больше информации изменяя исходный текст увеличивает шанс появления ошибок и аномалий [3].

На сегодняшний день в качестве контейнера чаще всего используются различные медиафайлы. Такие стегосистемы основаны на изменении значения наименьшего значащего бита так, чтобы из-за психофизической избыточности разница была незаметной для человека. Но полезная нагрузка этих стегосистем относительно мала. Также следует отметить, что их неудобно использовать на практике. Во-первых, сложно придумать предлог для обмена большим количеством такого контента. Во-вторых, медиафайлы необходимо создавать самостоятельно, поскольку изменения в файлах, взятых из Интернета, просто обнаружить, проведя поиск по похожим файлам. Гораздо лучшие показатели полезной нагрузки можно получить, если в качестве контейнера использовать обычный текст [4].

Можно выделить такие подходы к формированию текстового контейнера как:

1. Методы, основанные на формате.
2. Случайная и статистическая генерация.
3. Лингвистическая стеганография.

В стеганографии, основанной на формате, изменяется физическое представление текста. Например, добавляются лишние пробелы или знаки заменяются на визуально похожие. Полученный таким образом стеготекст легко отличить от естественного текста как примитивными программными средствами, так и человеком [1].

К тому же, если доступен оригинальный текст, обычное сравнение сразу обнаружит факт изменения файла. Поэтому имеет смысл не брать готовый контейнер, а генерировать новый [2].

Для формирования контейнера можно использовать статистические данные о длине слов, частоте появления букв и их сочетаниях в некотором естественном языке. Такой стеготекст может пройти статистические тесты, но уязвим перед анализом человеком.

Это стимулирует развитие лингвистической стеганографии, которая может решать задачу по формированию контейнеров, представляющих собой естественный текст, у которого не только правильная грамматика, но и логичный контекст. В таком тексте очень сложно распознать стеготекст поскольку понимание контекста – сложная задача для компьютера. [1]

Лингвистическую стеганографию можно разделить на уровни по возрастанию естественности, и соответственно сложности обнаружения: синтаксический, лексический и онтологический. Примером стегосистемы синтаксического уровня может быть свободная от контекста грамматика. Например, стегосистема NICETEXT. При таком подходе контейнер используется как синтаксический шаблон. За основу берётся любой текст. Чтобы получить из него шаблон, вместо каждого слова выписывается часть речи, которой оно соответствует. После чего шаблон заполняется словами из заранее составленного словаря в зависимости от скрываемого сообщения. Такой стеготекст выглядит синтаксически верным, но при проверке на лексическом уровне, он будет неестественным. К лексическому уровню стоит отнести стегосистемы, основанные на методах замены синонимов. Они практически не изменяют смысл текста-контейнера, поэтому обнаружить их сложнее. На онтологическом уровне можно изменять, например, порядок слов в предложении или добавлять вводные конструкции. В таком случае смысл текста остаётся абсолютно нетронутым, но полезная нагрузка системы ощутимо падает. [5]

К тому же работа с текстом на онтологическом уровне слишком сложна. Поэтому мы рассмотрим подробнее лексические методы, а именно методы, основанные на замене некоторых слов синонимами той же части речи.

Для обмена секретной информацией два собеседника могут использовать транслируемый в Интернет канал, общаясь при этом не друг с другом, а с кем угодно в чате. Для такого обмена информацией собеседники должны заранее знать к какому каналу присоединиться и знать имена друг друга. По протоколу Деффи-Хелмана они обмениваются длинным секретным ключом K . Используя ключ K , словам в заранее составленном словаре ставится в соответствие битовая комбинация. Далее в ходе сокрытия сообщения, в зависимости от значения этой комбинации, слова в контейнере заменяются одним из синонимов или остаются нетронутыми. [1]

Очевидными недостатками такого подхода являются неустойчивость перед статистическим анализом и возможность появления грамматических ошибок при замене слов. Также важно учесть, что многие синонимы можно употреблять только в определённом контексте. Чтобы следить за корректностью употребления синонимов, можно использовать лингвистические корпуса. С помощью корпуса оценивается относительная частота появления заменяющего синонима в данном контексте и анализируется частота использования слов в естественном языке. Также можно использовать словари, показывающие изменение частей речи для уточнения окончаний у заменяемых слов. [5]

Ещё одна проблема заключается в том, что у слова может быть много значений, а следовательно, в разных наборах синонимов одному слову будут поставлены в соответствие разные битовые строки. Одним из решений данной проблемы может быть метод раскраски вершин.

Для того чтобы воспользоваться этим методом, нужно представить множество всех слов в виде ненаправленного графа, в котором в вершинах находятся слова, а синонимы с близкими значениями соединены рёбрами. Далее вершины графа маркируются так, чтобы никакие две смежные вершины не имели одинаковую маркировку (цвет). Наименьшее количество цветов, необходимых для раскраски такого графа G , будем называть хроматическим числом $X(G) = k$. Тогда граф будет называться k – хроматическим. Таким образом, получится граф синонимов, в котором не будет синонимов, маркированных одинаковым цветом. Теперь можно однозначно идентифицировать слово в наборе синонимов, заменив цвета на соответствующие им битовые комбинации

Как отмечено в работе [3] подавляющее большинство наборов синонимов содержит не более 8 слов. А для того, чтобы сделать наборы более равномерными, диапазон возможных значений хроматического числа k в рассматриваемой системе установлен от 1 до 4. Соответственно, максимальное количество слов в наборе равно 4, а для кодирования одного слова необходимо 2 бита.

Стегосистема, в которой слова кодируются таким образом, имеет очевидную избыточность поскольку существуют наборы, состоящие из меньшего количества синонимов. Для устранения этой избыточности необходимо сокращать кодовое слово так, чтобы в меньших наборах использовались только старшие значащие биты. Отметим, что в граф синонимов не включаются наборы, состоящие из единственного слова, а также синонимические выражения.

Таким образом, лингвистическая стеганография является перспективным для изучения направлением в сфере защиты информации. Преимущества описанных методов заключаются в возможности 282

сокрытия самого факта тайной переписки, используя открытые каналы. Наиболее оптимальными с точки зрения соотношения полезной нагрузки и естественности являются методы, основанные на замене синонимов. Но, чтобы не понижать естественность, а значит и скрытность текста, для их правильного использования необходимо использовать лингвистические корпуса.

Библиографический список

1. *Wyseur B., Wouters K., Preneel B.* Lexical natural language steganography systems with human interaction// ECIW 2007, Heverlee, Belgium, 2007
2. *Agarwal M.* Text staganographic approaches: a comparsion// International Journal of Network Security and Its Applications (IJNSA), № 1, 2013.
3. *Chang C., Clark S.* Practical Linguistic Steganography using Contextual Synonym Substitution and Vertex Colour Coding// Cambridge: University of Cambridge, 2013.
4. *Нестерук С. В., Беззатеев С. В.* Обзор подходов лингвистической стеганографии// 69 международная студенческая конференция ГУАП, Сб. докл.: ч. 1 Технические науки, СПб.: ГУАП, 20016, С. 393-396
5. *Nanhe A. M., Kunjir M. P., Sakdeo S. V.* Improved synonym approach to linguistic steganography design and proof-of-concept implementation// 2008

УДК 004.725.2

С. В. Нестерук, А. В. Шишко

студенты кафедры технологий защиты информации

С. В. Беззатеев

д-р техн. наук, доцент – научный руководитель

АСПЕКТЫ БЕЗОПАСНОСТИ БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЕЙ

Интернет активно развивается, и сегодня мы видим, что, пройдя путь от локальных сетей, он начинает охватывать и физический мир, поэтому всё большее распространения получают сенсорные сети.

Беспроводная сенсорная сеть представляет собой самоорганизующуюся систему, состоящую из связанных радиоканалом маломощных узлов, которые могут выполнять роль пассивных датчиков для сбора данных или роль исполнительных устройств. В системе, построенной таким образом, устройства должны общаться между собой и реагировать на изменения в окружающей среде так, чтобы выполнялась поставленная задача. [1]

Разрабатываемая нами система должна решать задачу экомониторинга: собирать и обрабатывать значения параметров окружающей среды на участке, где она установлена. Такие системы начинают использовать в точном земледелии для прогнозирования урожайности, изучения влияния различных факторов на развитие растений и улучшение качества продукции на основе проведённых исследований [2]. В общем случае, системы, взаимодействующие с окружающей средой, отвечают концепции Интернета вещей.

Задачами данной работы является краткий аналитический обзор основных аспектов безопасности сенсорных сетей и выбор эффективного алгоритма взаимной аутентификации датчиков в сети.

Системы, основанные на концепции Интернета вещей, предоставляют широкие возможности для автоматизации и контроля процессов в физическом мире. По этой причине они уже активно внедряются на производствах и даже в повседневной жизни. Но с расширением возможностей появляется множество угроз безопасности, кроме того, разработку таких систем усложняет отсутствие единого стандарта безопасности для систем Интернета вещей [3].

Системы Интернета вещей сложные и многоуровневые. В разных моделях определяют от 4 до 7 уровней. Верхние уровни аналогичны тем, к которым мы привыкли в стандартных приложениях: они решают задачи транспортировки, хранения, обработки и отображения данных. На каждом из этих уровней существуют свои угрозы безопасности [4], но способы борьбы с ними известны и стандартизированы.

В разрабатываемой нами системе можно выделить 4 уровня: уровень устройства, шлюз, сервер и уровень приложения. Поскольку мы выбрали тип архитектуры «тонкий клиент», основным уровнем является сервер. На сервере информация обрабатывается и хранится в реляционной базе данных. Непосредственный сбор данных осуществляется на нижнем уровне, уровне устройства. В данной системе этот уровень представлен сетью датчиков. Датчики самостоятельно решают задачу маршрутизации собранных данных и передают данные на шлюз, который необходим для пересылки данных на сервер. Последний уровень – это клиентское приложение, которое необходимо для отображения информации и взаимодействия с пользователем.

При разработке информационной системы необходимо обеспечить безопасное хранение и передачу данных на каждом уровне. Отдельного внимания заслуживает вопрос обеспечения безопасности на уровне устройства, поскольку на него может быть направлен основной вектор атак и методы борьбы с ними не стандартизированы. Для работы сенсорной сети на этом уровне необходимо обеспечить безопасную работу датчиков во время всего жизненного цикла системы, т. е. должна происходить инициализация датчика в сети, его стабильная работа и удаление датчика из сети [5].

Для шифрования сообщений между датчиками и их взаимной аутентификации в простейшем случае можно предусматривать общий ключ на все устройства во время их сборки. Но в таком случае, получив доступ к одному устройству, злоумышленник сможет перехватывать сообщения, переда-

ваемые между любыми датчиками и кроме того внедрять свой датчик в сенсорную сеть. Чтобы этого избежать, для каждой пары датчиков можно использовать свой ключ. Такой ключ невозможно установить на устройства заранее, поскольку неизвестно, как будут расставлены устройства и, соответственно, какие из них окажутся в зоне действия друг друга и образуют пары. Также предустановка парных ключей исключает возможность масштабирования сети после начальной инициализации. Таким образом, появляется необходимость выработки парных ключей в момент инициализации датчиков.

Рассмотрим следующий протокол. Перед началом работы на все устройства сенсорной сети устанавливаем мастер-ключ MK . Также каждый датчик i должен иметь свой уникальный для данной сети идентификатор ID_i . На этапе инициализации сети датчики обмениваются идентификаторами с датчиками в радиусе своего действия. Будем считать, что на этом этапе не может производиться атака на систему. На основе информации об идентификаторах соседних датчиков и мастер ключе каждый датчик вычисляет парные ключи для взаимной аутентификации с соседними датчиками. Для датчиков i и j ($i < j$) с идентификаторами ID_i и ID_j ($ID_i < ID_j$) ключи будут вычислены следующим образом:

$$K_{ij} = K_{ji} = H\left(ID_i \| ID_j \| MK\right)$$

Так, для получения парного ключа, необходимо сначала произвести конкатенацию идентификаторов и мастер-ключа, а после посчитать хеш-функцию от полученной строки. Кроме этого, необходимо вычислить вспомогательный ключ для каждого датчика, который понадобится в дальнейшем для масштабирования сети:

$$K_{ii} = H\left(ID_i \| MK\right)$$

После создания ключей мастер-ключ удаляется из памяти всех устройств.

При необходимости добавления нового датчика он должен опросить идентификаторы соседних датчиков и вычислить на их основе парный ключ для каждого соседа

$$K_{new,j} = H\left(ID_i \| MK\right) = K_{ij}$$

Этот ключ будет использоваться для взаимной аутентификации [5]. Если в рассмотренной системе нелегально переместить датчик так, что поменяются все окружающие его датчики сети, т. е. совершить физическую атаку, он не сможет пройти процедуру аутентификации, поскольку рядом не будет датчиков, с которыми у него есть общие парные ключи.

Таким образом, рассмотренный алгоритм взаимной аутентификации позволяет решить задачи начальной инициализации сети, безопасного масштабирования сети, а также обеспечивает защиту от несанкционированного перемещения датчика. Важно отметить, что данный алгоритм не требует большого количества операций, что особенно важно при использовании на устройствах с малым энергоресурсом.

Библиографический список

1. Рагозин Д. В. Моделирование синхронизированных сенсорных сетей // Проблемы программирования. 2008. №2-3. С. 721-729.
2. Нестерук С. В., Беззатеев С. В. Разработка и внедрение системы экомониторинга на предприятия агропромышленного комплекса (региональный аспект)// Технологическая перспектива в рамках Евразийского пространства: новые рынки и точки экономического роста: Сб. докл./ СПб.: Изд-во «Астерион», 2016. С. 388.
3. Безмальный В. IoT угрозы безопасности: [Электронный ресурс]. URL: <https://www.pcweek.ru/security/article/detail.php?ID=178122>. (Дата обращения: 15.02.2017)
4. Сабанов А. Г. Некоторые проблемы обеспечения безопасности Интернета вещей // Защита информации. Инсайд. 2016. №4. С. 1-5.
5. Jang J., Kwon T., Song J. A time-based key management protocol for wireless sensor networks// Proceedings of ISPEC, 2007, LNCS 4464, pp. 314-328.

УДК 004.056.5

А.С. Ровкова

Магистр кафедры информационных технологий и защиты информации

А. В. Афанасьева

научный руководитель

РАЗРАБОТКА МЕТОДА ИЗВЛЕЧЕНИЯ СТЕГАНОГРАФИЧЕСКИХ ДАННЫХ ИЗ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ С МЯГКИМ ВЫХОДОМ

В статье описан «мягкий» метод извлечения ЦВЗ из видеопоследовательности, а так же приведены результаты моделирования данного метода для оценки его эффективности. Проведено сравнение «мягкого» и «жесткого» метода извлечения битов ЦВЗ в сочетании с антикоалиционными кодами Тардоша по двум критериям: процент ложного обвинения и среднее число найденных участников коалиции.

Ключевые слова: Цифровой водянок знак, коды Тардоша, коалиционная атака.

В современном мире все актуальней становится проблема незаконного копирования информации, представленной в цифровом виде, и защиты авторского права лицензионных копий цифрового контента. Наиболее часто используемым решением этих проблем является внедрение индивидуальных цифровых водяных знаков (ЦВЗ). Идея метода состоит в том, что в каждую лицензионную копию цифрового контента внедряется индивидуальная метка, по которой можно однозначно определить владельца данной копии контента. Но использование индивидуальных цифровых знаков привело к появлению нового метода атаки на цифровый контент, целью которого является незаконное распространение контента – коалиционная атака. Данная атака проводится следующим образом: несколько законных владельцев копий контента объединяются в коалицию. Далее они путем сравнения своих копий цифрового продукта выявляют позиции битов индивидуальных ЦВЗ и создают новую копию контента, в которой будет уже новая индивидуальная метка, по которой невозможно определить владельца копии.

Для борьбы с коалиционными атаками были разработаны антикоалиционные коды. Целью их использования является поиск участников коалиции по полученной из пиратской копии цифровой метке.

«Мягкий» метод извлечения. Исторически сложилось так, что разработка методов внедрения и извлечения битов ЦВЗ и методов генерации антикоалиционных меток с функцией поиска участников коалиции велась раздельно и независимо друг от друга. А в связи с этим и рассматривались различные модели атак: при разработке методов внедрения и извлечения метки рассматривались атаки внешения шума путем модификаций контента (перезапись файла, наложение фильтров, перекодирование в другой формат). При разработке методов генерации меток, основной рассматриваемой атакой, являлась атака коалиции. В результате оказалось, что антикоалиционные свойства меток ухудшались при применении совместной атаки модификацией контента и коалиционной, такие атаки стали приводить к ложным обвинениям. Один из способов решения данной проблемы был предложен в работе [1]. В данной статье в качестве решения проблемы ложных обвинений было предложено согласовать функцию поиска участников коалиции с каналом путем замены «жесткого» метода извлечения «мягким».

Под «мягким» методом извлечения понимается возможность оценки надежности извлекаемого символа. Под надежностью символа понимается вероятность того, что значение извлеченного бита равно значению внедренного бита. Впоследствии, эта надежность бита должна быть использована в функции поиска участников. В данной статье будет предложен способ реализации «мягкого» метода извлечения ЦВЗ и модификация функции обвинения для работы с надежностями.

Генерация и внедрение метки. В качестве антикоалиционных меток будет использован вероятностный код Тардоша, как наилучший известный на данный момент. Метод построения кода и доказательство оптимальности его параметров были описаны в статье [2]. Для реализации «мягкого» мето-

да извлечения ЦВЗ необходимо модифицировать обе процедуры: вставки и извлечения. При внедрении ЦВЗ в данной работе предлагается использовать известный метод расширения спектра, ранее использовавшийся в «слепых» стеганографических алгоритмах. При этом для вставки каждого символа внедряемой метки будет использоваться одна из ортогональных последовательностей, количество используемых последовательностей равно размеру алфавита внедряемой метки. В качестве ортогональных последовательностей в данной статье будут использоваться последовательности Уолша-Адамара, состоящие из 0 и 1. Внедрение метки производится в коэффициенты ДКП, в зависимости от внедряемого бита, будет выбираться используемая ортогональная последовательность. Функция внедрения представлена в формуле 1.

$$DCT[current] = DCT[current] + \alpha * H[version][current], \quad (1)$$

где $DCT[current]$ – коэффициент ДКП; α – коэффициент внедрения; H – ортогональная последовательность; $version$ – номер выбранной ортогональной последовательности; $current$ – номер бита ортогональной последовательности.

В результате один бит метки, закодированный одной последовательностью, будет внедрен в один блок ДКП. Выходом этапа внедрения помимо блока с меткой, также являются метаданные, необходимые для последующего извлечения ЦВЗ, а именно номер видеокадра и координаты блока, в который был внедрен бит. На рис. 1 представлена схема внедрения бита ЦВЗ.

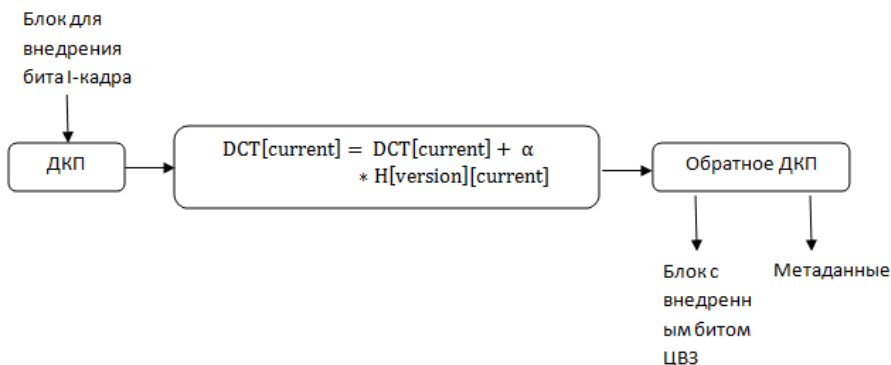


Рис. 1. Схема внедрения бита ЦВЗ

Таким образом, внедряются все биты ЦВЗ

Извлечение метки. В данной статье используется не «слепой» метод извлечения, следовательно, для извлечения битов ЦВЗ понадобится исходное видео без внедренной метки и метаданные, полученные на этапе внедрения.

На вход подаются два блока видеокадра: один с внедренным битом, второй из исходного видео. Над двумя блоками производится ДКП и далее считается разность между соответствующими коэффициентами ДКП двух видеоблоков. Таким образом, получена последовательность, состоящая из разностей коэффициентов ДКП. На следующем этапе считаются корреляции между полученной последовательностью и двумя используемыми ортогональными последовательностями (формула 2), соответствующими битам метки. На значение внедренного бита определяется по максимуму корреляции

$$correlate[version] = \sum_{i=0}^{size} \Delta DCT[i] H[version][i]; \quad (2)$$

$$H_{used} = \max_{version} (|correlate[version]|) \quad (3)$$

В качестве надежности извлеченного бита используется значение корреляции, отнормированное к единице

$$p_{y[i]} = \frac{\text{correlate}[H_{used}]}{\alpha * \sum_{i=0}^{\text{size}} H[H_{used}][i]} \quad (4)$$

Поиск участников коалиции. Оригинальная функция подсчета виновности для кодов Тардоша представлена в работах [2] и [3]. Значение функции виновности рассчитывается для всех кодовых слов, и решение о виновности для каждого пользователя принимается по этому значению для его слова. Данная формула была модифицирована для использования дополнительной информации о надежности символов:

$$\text{Accusation}[user] = \sum_{i=0}^{m-1} p_{y[i]} * U(y[i], \text{code}[user][i], p[i])$$

где m – длина кода Тардоша, $y[i]$ – извлеченный бит последовательности, $p_{y[i]}$ – надежность символа $y[i]$, $\text{code}[user][i]$ – бит кодовой последовательности, принадлежащей пользователю $user$.

Такое решение должно привести нас к желаемому результату в связи с тем, что коды Тардоша являются вероятностными кодами и благодаря корреляционной природе функции обвинения.

В данной работе были рассмотрены три варианта определения виновных пользователей:

1. По порогу: виновными считаются все пользователи, у которых функция виновности оказалась больше фиксированного порога. Для «жесткого» метода извлечения, при котором извлекается только бит цифрового водяного знака, формула для расчета порога была предложена в [2]. Данная формула была модифицирована для «мягкого» метода извлечения пем внесения в качестве поправочного коэффициента вероятности достоверного извлечения:

$$Z_{\text{soft}} = Z_{\text{hard}} * (1 - p_{\text{err}})$$

2. По максимуму функции обвинения: виновным признается участник с максимальным значением функции обвинения. Данный способ был предложен в статье [1]. При таком способе находится только один участник коалиции.

3. По изменению дисперсии. Все значения функции обвинения сортируются по убыванию. После чего вычисляется исправленная дисперсия для подмножества значений функции, каждый раз увеличивая выборку на 1 в сторону уменьшения до тех пор, пока не будет достигнут максимальный размер коалиции для кода. Виновными считаются все пользователи, для которых изменение дисперсии максимально.

Результаты моделирования. В данной работе было проведено моделирование описанных методов. Для моделирования коалиционной атаки случайным образом выбиралось 5 пользователей, и на основе их индивидуальных ЦВЗ генерировалась новая последовательность, которая не принадлежала ни одному пользователю. Именно эта последовательность внедрялось в видеофайл. В качестве атаки модификации контента была использована атака пережатия файла с повышенным коэффициентом квантования. Было проведено по 100 экспериментов на 10 различных роликах видео. Сводная таблица результатов экспериментов приведена в табл. 1.

Анализ результатов. По результатам анализа полученных данных были сделаны следующие выводы:

1) При использовании «мягкого» метода извлечения значительно снижается процент ложных обвинений пользователей для любого метода определения границы обвинения.

2) При определении границы обвинения по порогу среднее значение вероятности ложного обвинения ниже, чем при определении границы по изменению дисперсии. Наименьшее значение вероятности ошибки ложного обвинения достигается при использовании метода определения границы по максимуму обвинительной функции, но у данного метода наименьшее число найденных участников коалиции.

Таблица 1

Результаты экспериментов

N видео	Пороговое решение				Макс U		Макс дисперсии			
	Ложные обвинения		Найденные участники		Ложные обвинения		Ложные обвинения		Найденные участники	
	Hard	Soft	Hard	Soft	Hard	Soft	Hard	Soft	Hard	Soft
1	75	13	4,04	3,9	0	0	4	3	2,7	3,07
2	87	6	3,61	3,5	2	0	16	3	2,5	2,94
3	87	7,7	3,62	4,1	2,88	0	22,1	1	2,5	3,1
4	95	6	3,38	3,3	4	0	23	8	2,3	3,07
5	89	8	3,34	2,4	4	0	26	10	2,3	2,84
6	100	0	2,68	1,1	11	0	52	15	2	2,81
7	66	5	4,08	3,3	1	0	14	11	2,8	2,93
8	68	3	4,1	3,7	0	0	14	2	2,8	3,21
9	91	2	3,23	2,1	3	0	31	14	2,2	2,74
10	87	6	3,58	2,6	5	0	23	9	2,5	2,8

При использовании метода определения границ обвинения по порогу среднее число найденных участников коалиции выше, чем у остальных методов. Наименьшее число найденных участников коалиции принадлежит методу определения границы по максимуму функции обвинения. В табл. 2 представлено среднее количество найденных участников коалиции для всех трех методов.

Таблица 2

Среднее количество найденных участников коалиции для всех трех методов

Участники реализации	По порогу	По максимуму функции обвинения	По изменению дисперсии
Среднее число найденных участников коалиции	3,02	1	2,95

Библиографический список

1. Skoric B., Katzenbeisser S., Schaathu H. G., Celik M. U. Tardos Fingerprinting Codes in the Combined Digit Model // IEEE Transactions on Information Forensics and Security. 2009. V. 6. Is. 3.
2. Tardos G. Optimal Probabilistic fingerprint Codes // STOC. 2003.
3. Shahid Z., Chaumont M., Puech W. Spread spectrum-based watermarking for Tardos code-based fingerprinting for H.264/AVC video // IEEE International Conference on Image Processing. 2010.

УДК 004.942

А. В. Сермягина

студент кафедры безопасности информационных систем

Т. М. Татарникова

д-р техн. наук, профессор – научный руководитель

ПРИМЕНЕНИЕ МЕТОДОВ СИСТЕМНОГО АНАЛИЗА В ЗАДАЧЕ ИДЕНТИФИКАЦИИ

Предложено решение задачи идентификации веществ по спектру с применением методов системного анализа.

Ключевые слова: идентификация, спектр вещества, кластерный анализ, метод главных компонент, факторный анализ.

В настоящее время имеется достаточное количество методов обнаружения и идентификации веществ, одни из них находятся в стадии разработки, а другие уже реализованы в серийно выпускаемых приборах и системах. Многие приборы для обнаружения и идентификации веществ используют электрохимические методы, такие как газохроматографический, спектроскопию ионной подвижности, масс-спектрометрию, метод цветowych химических реакций, а также методы, представляющие собой объединение выше названных методов или же их модификацию. Их серьезным недостатком является длительное время анализа исследуемого вещества. На практике время, необходимое для проведения качественного анализа занимает до 30 секунд, что не всегда приемлемо, особенно в случае массового досмотра. Таким образом, в этой области стоит важная задача уменьшения времени и увеличения скорости работы алгоритмов обнаружения и идентификации.

В работе предлагается решать задачу идентификации методами системного анализа, применение которых позволяет сократить размерность пространства признаков и соответственно уменьшить время идентификации вещества. Суть предлагаемого решения заключается в следующем.

Для выполнения идентификации вещества необходимо получить спектр этого вещества, иметь базу данных известных веществ и разработать программу детектора, выполняющую классификацию вещества по его спектру.

Таким образом, работа выполнялась в два этапа:

- создание базы данных,
- разработка программы детектора.

Для создания базы данных спектров используется метод кластерного анализа – метод k-средних. В результате получаем разбиение известных веществ на кластеры и значения их центров. На рис. 1 представлены графики образцов, сформированных в кластер.

Создание базы данных основывается на полученной ранее базе спектров и использовании метода главных компонент, который позволяет рассчитать матрицу компонентного отображения, с помощью которой можно легко вычислить значения главных компонент для нового образца. Данная матрица рассчитывается единожды и становится частью создаваемой базы данных.

Используя найденные разбиение на кластеры и их центры, можно построить базу данных при помощи правила стандартного отклонения.

Для идентификации вещества необходима созданная база данных и спектр идентифицируемого вещества. Тогда умножаем полученный спектр на матрицу компонентного отображения, получив в результате главные компоненты вещества. И сравниваем первую главную компоненту с интервалами в базе.

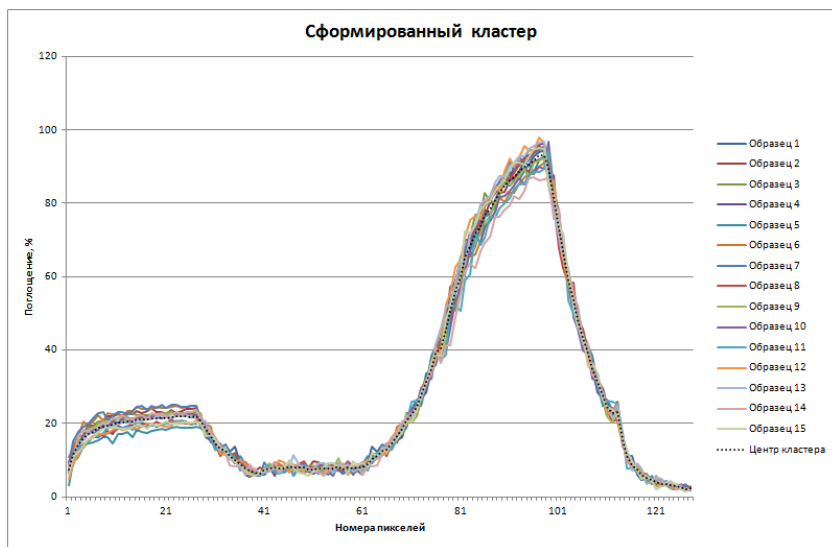


Рис. 1. Кластер, сформированный в процессе кластерного анализа

На рис. 2 и 3 представлен алгоритм работы программы.



Рис. 2. Алгоритм формирования базы данных веществ

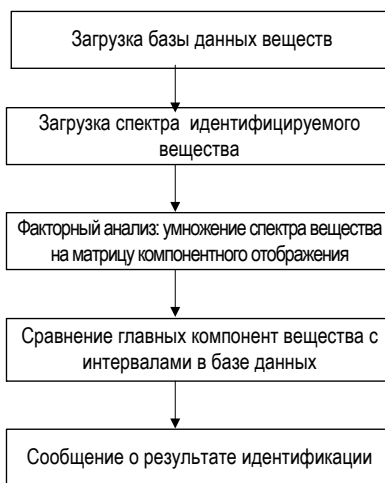


Рис. 3. Алгоритм идентификации

Сравнительные результаты скорости идентификации с помощью предложенного алгоритма и программной части прибора «Данник-4» представлены в табл. 1.

Таблица 1

Сравнение скорости работы алгоритмов

Объект сравнения	Скорость работы (секунды)
Разработанный алгоритм	$(15-20) \cdot 10^{-2}$
Прибор «Данник-4»	$(23-30) \cdot 10^{-2}$

Использование методов кластерного анализа позволили проверить полноту и достаточность исходных данных для решения задачи идентификации, а также выявить некорректные данные. Использование метода главных компонент обеспечивает высокую точность и малое время, необходимые для идентификации вещества.

Библиографический список

1. *Васильев Н.С.* Методы и алгоритмы идентификации веществ по сильно зашумленным спектрам: автореф. дис. ... канд. техн. наук: 05.13.18/ Московский государственный технический университет им. Н. Э. Баумана. М., 2015. 34 с.
2. *Васильев Н.С., Морозов А.Н.* Алгоритм идентификации веществ по набору спектров вторичного излучения//Инженерный журнал: наука и инновации. 2013. Вып. 8. URL: <http://engjournal.ru/catalog/fundamentals/physics/1109.html>.
3. *Ким Дж.-О., Мьюллер Ч.У., Клекка У.Р.* и др. Факторный, дискриминантный и кластерный анализ: Пер. с англ./ под ред. И.С. Енюкова. М.: Финансы и статистика, 1989. 215 с.
4. *Такахаси Син.* Занимательная статистика. Факторный анализ. – М.: ДМК Пресс, 2015. 146 с.

УДК 629.735.33

Ю.В. Соколова, А.В. Дроздов

студенты кафедры информационно-сетевых технологий

С.И. Зиятдинов

д-р техн. наук, профессор – научный руководитель

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ СИНТЕЗА ЛИНЕЙНЫХ СИСТЕМ
НА ОСНОВЕ ИНВАРИАНТНОЙ ИМПУЛЬСНОЙ И ПЕРЕХОДНОЙ ХАРАКТЕРИСТИК**

В статье дан сравнительный анализ динамических характеристик линейных систем синтезированных на импульсной и переходной характеристиках. В качестве примера рассмотрен фильтр нижних частот первого порядка.

Существующие методы синтеза дискретных фильтров делятся на синтез во временной и частотных областях. Методы синтеза во временной области включают в себя синтез на основе инвариантной импульсной характеристики и инвариантной переходной характеристики.

Для примера рассмотрим фильтр нижних частот первого порядка. Передаточная функция которого в непрерывном варианте [1]

$$W(j\omega) = \frac{1}{1 + j\omega\tau}, \quad (1)$$

где τ – постоянная времени фильтра, которая определяет скорость реакции фильтра на скачкообразное изменение входного сигнала.

Импульсная характеристика такого фильтра находится из выражения

$$h_H(t) = \frac{1}{\tau} e^{-\frac{t}{\tau}}, \quad (2)$$

где $\omega_{cp} = \frac{1}{\tau}$ – частота среза.

Вид импульсной характеристики данного непрерывного фильтра нижних частот первого порядка изображен на рис. 1.

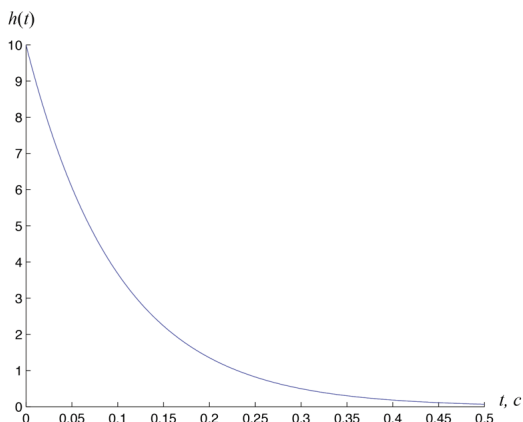


Рис. 1. Импульсная характеристика непрерывного фильтра нижних частот первого порядка

Переходная характеристика фильтра (рис. 2) определяется следующим соотношением

$$g(t) = 1 - e^{-\frac{t}{\tau}}, \quad (3)$$

где постоянная времени $\tau = RC$.

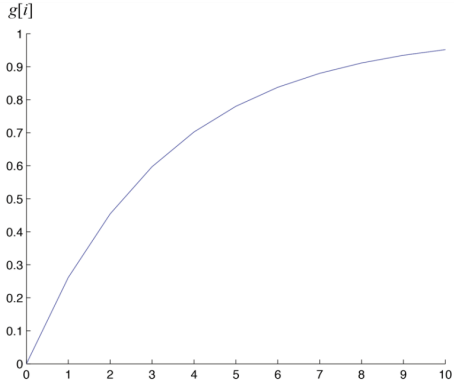


Рис. 2. Переходная характеристика непрерывного фильтра

При синтезе на основе инвариантно-переходной характеристики необходимо разработать дискретный фильтр, переходная характеристика которого максимально точно совпадала бы с переходной характеристикой непрерывного фильтра.

При этом в общем виде при заданной импульсной характеристике непрерывного фильтра $h_n(t)$, выходной сигнал $y(t)$ при входном сигнале $x(t)$ находится с использованием интеграла Дюамеля [2]:

$$y(t) = \int_0^t x(t-\tau)h_n(\tau)dt. \quad (4)$$

Заменяя интеграл суммой, получаем алгоритм работы дискретной линейной системы

$$y[k] = T \sum_{i=0}^k x[k-i]g[i], \quad (5)$$

где ω_{cp} – частота среза фильтра, которая определяется выражением; T – период дискретизации.

Если воспользоваться формулой (5), то выходной сигнал фильтра при единичном ступенчатом входном воздействии имеет вид, представленный на рис. 3. На этом же рисунке сплошной линией показана переходная характеристика непрерывного фильтра.

Чтобы обеспечить совпадение динамических характеристик дискретного и непрерывного фильтра в интеграле Дюамеля импульсную характеристику непрерывного фильтра представим в виде приращения переходной характеристики следующим образом:

$$h(t) = \frac{dg(t)}{dt} \approx \frac{g(t) - g(t-T)}{T} = \frac{\Delta g[i]}{T}. \quad (6)$$

После подстановки данного выражения в выражение (5), получим следующее выражения для вычисления выходного сигнала дискретного фильтра с использованием метода инвариантной переходной характеристики:

$$U_{вых}(n) = T \sum_{i=0}^{\infty} U_{вх}[n-i] \frac{\Delta g[i]}{T} = \sum_{i=0}^{\infty} U_{вх}[n-i] \Delta g[i]. \quad (1.7)$$

В этом случае переходная характеристика фильтра будет показана на рис. 4. Здесь же пунктирной линией представлена переходная характеристика непрерывного фильтра

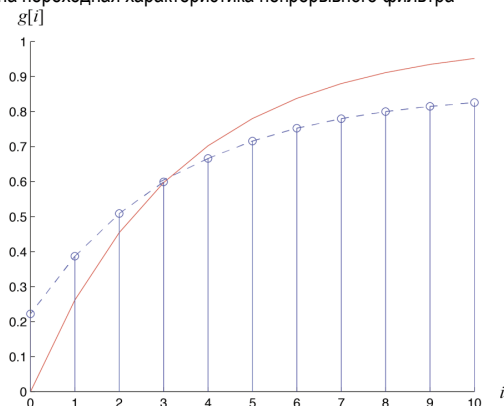


Рис. 3. Выходной сигнал фильтра при единичном ступенчатом входном воздействии $g_d[i]$

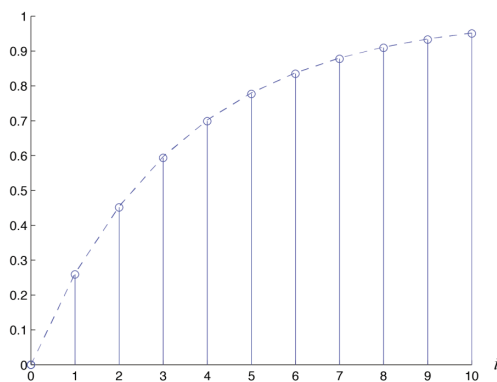


Рис. 4. Переходная характеристика фильтра

Вывод: синтез дискретных фильтров на основе метода инвариантной импульсной характеристики не обеспечивает точное совпадение динамических характеристик дискретных и непрерывных фильтров. Однако, синтез дискретных фильтров на основе метода инвариантной переходной характеристики дает полное совпадение динамических характеристик дискретных и непрерывных линейных систем. Чтобы на практике использовать интеграл Дюамеля необходимо в нем использовать не отсчеты импульсной характеристики, а отсчеты приращения переходной характеристики.

Библиографический список

1. Воробьев С.Н. Цифровая обработка сигналов: учебник для студ. учреждений высш. проф. образования. – М.: Издательский центр «Академия», 2013. – 320 с.
2. Гоноровский И.С. Радиотехнические цепи и сигналы. – М.: Радио и связь, 1986. 512 с.

УДК 004.942

А. А. Татарникова

студент кафедры безопасности информационных систем

А. М. Тюрликов

д-р техн. наук – научный руководитель

СИСТЕМА УДАЛЕННОГО МОНИТОРИНГА ПАЦИЕНТОВ С ЗАБОЛЕВАНИЯМИ ОПОРНО-ДВИГАТЕЛЬНОГО АППАРАТА

В данной работе исследуется возможность разработки алгоритмов обнаружения патологий опорно-двигательного аппарата с применением акселерометров мобильных устройств и создания на их основе системы удаленного мониторинга без использования дополнительных измерительных приборов.

Ключевые слова: акселерометры в мобильных устройствах, дискретное преобразование Фурье, заболевания опорно-двигательного аппарата.

Настоящая работа построена следующим образом. Рассмотрены современные решения в области удаленного мониторинга пациентов и используемые в ортопедии и травматологии системы диагностики заболеваний опорно-двигательного аппарата. На их основе предложен способ обнаружения патологий опорно-двигательного аппарата с помощью акселерометров мобильных устройств. Приводится описание проведенных испытаний, которые проводились для того, чтобы проверить выдвинутое предположение, полученных результатов экспериментов и выводы.

В здравоохранении первостепенную важность для большинства людей имеет легкий доступ к медицинским услугам, а также возможность самостоятельно следить за своим здоровьем. Из-за этого медицинские учреждения во всем мире постепенно переходят к ориентированной на пациента инструментальной модели здравоохранения. Такой переход осуществляется и в нашей стране и связан с электронным здравоохранением (общепринятый термин – eHealth). Во многих медицинских учреждениях есть услуга дистанционной записи к врачу и электронные медицинские карты, а в некоторых уже введены такие услуги и технологии как виртуальное здравоохранение (телемедицина) и Интернет медицинских устройств (Internet of Health Things, IoT). Также относительно недавно появилась новая ветвь электронного здравоохранения – мобильное здравоохранение (mHealth).

Мобильное здравоохранение – это термин, обозначающий использование мобильных устройств и беспроводных технологий в целях медицинской помощи, а также обеспечения здорового образа жизни человека.

Сегодня mHealth – это два крупных направления, которые развиваются параллельно, с разной скоростью, оказывая при этом взаимное влияние друг на друга. Первое – это системы и устройства, предназначенные для контроля за соблюдением здорового образа жизни, второе – это технологии, устройства, приложения и услуги для лечения и ухода за пациентами.

Второе направление специалисты в отрасли медицины и здравоохранения считают наиболее перспективным, так как подобные устройства позволяют снизить частоту посещений медицинских учреждений и обеспечить более персонализируемый и настраиваемый план лечения.

Примерами решений в данном направлении являются такие приложения как: «Wireless Blood Pressure Monitor» компании «Withings» и «Scout» компании «Scanadu», «Qardiocore wearable ECG/EKG» компании «Qardio». Однако большинство из этих приложений позволяют измерять только простые показатели (такие как давление или частота сердечных сокращений) и при этом для их использования требуются дополнительные измерительные приборы.

При всем развитии современных технологий в медицине, состояние больных с травмами и заболеваниями опорно-двигательного аппарата до сих пор определяется только на основании данных клинического и рентгенологического исследований. На данный момент распространены в применении

такие устройства как рентген, компьютерная томография, магнитно-резонансная томография, компьютерная оптическая топография [1–3] и различные приборы, в состав которых входят акселерометры [4–6].

Вместе с тем стоит отметить, что функциональные нарушения ОДА являются распространенной патологией, особенно среди детского населения страны. Современными исследованиями установлено, что 75% школьников в возрасте от 9 до 14 лет имеют различные нарушения опорно-двигательного аппарата, из них 54% – нарушения осанки, 43% – плоскостопие, 34% разноразмерность нижних конечностей. Без проведения реабилитационных мероприятий в периоды интенсивного роста у подростков количество функциональных нарушений ОДА нарастает [1].

Многочисленные исследования, посвященные состоянию ОДА у детей и подростков, свидетельствуют о том, что с годами эта проблема не теряет своей актуальности. А функциональные нарушения ОДА у детей и подростков создают предпосылки к развитию дегенеративных изменений в межпозвоноковых суставах, формированию юношеского остеохондроза, появлению грыж дискового аппарата позвоночника, дистрофических изменений тазобедренных и коленных суставов, что приводит к инвалидизации во взрослом периоде жизни [1].

Из всего выше перечисленного можно сделать вывод, что на сегодняшний день диагностика заболеваний ОДА до сих пор является одной из актуальных проблем в области медицины, особенно для детей и подростков, так как именно в этот период жизни человека происходит активный рост, и, как следствие, увеличивается риск появления нарушений ОДА. Для решения данной проблемы требуются более современные методы диагностирования, которые можно было бы применять удаленно. А как видно, сейчас в области удаленного мониторинга в целом наиболее эффективны и используются решения, требующие дополнительных измерительных приборов. В области ортопедии и травматологии пока используются традиционные дорожные и громоздкие системы диагностики заболеваний ОДА, которые невозможно применять без присутствия пациента в медицинском учреждении.

Данную проблему в настоящей работе предлагается решить, используя современные мобильные телефоны, которые снабжены множеством различных сенсоров, таких как акселерометры, гироскопы, датчики давления, света и т.д. Акселерометры в мобильных устройствах позволяют измерять ускорения сотового телефона, но при этом они пока используются для достаточно примитивных задач, например, для позиционирования экрана при повороте устройства. Однако, если зафиксировать мобильный телефон на теле человека, можно измерять ускорения, происходящие во время его движения. Эти данные можно использовать для обнаружения возможных заболеваний опорно-двигательного аппарата человека.

Используя идею, представленную в работе [4], предлагается следующий подход к построению системы: при возникновении заболевания опорно-двигательного аппарата в частотном спектре возможны изменения, например, появление большого количества высокочастотных составляющих. По походке человека с помощью мобильного приложения, которое будет фиксировать показания акселерометра в мобильном телефоне, можно обнаруживать патологии опорно-двигательного аппарата и отправлять соответствующие данные лечащему врачу.

Стоит отметить, что использование современных технологий в медицине приводит к учащению случаев нарушения защиты персональных данных пациентов, особенно по мере того, как система здравоохранения оцифровывает свои данные. В связи с этим в медицинских учреждениях требуется принимать срочные меры по обеспечению безопасности идентификационных данных пациентов. Из всего выше перечисленного можно сделать вывод, задача разработки защищенной системы удаленного мониторинга пациентов, в том числе с заболеваниями опорно-двигательного аппарата, становится наиболее актуальной.

Чтобы проверить выдвинутую гипотезу, был проведен ряд экспериментов, которые проходили следующим образом.

Мобильный телефон фиксировался на теле человека с правой стороны на пояс в вертикальном положении. Для этого использовался специальный чехол для телефона, который можно закрепить на ремне. После чего испытуемый в течение 10-15 секунд ходил по прямой своей обычной походкой.

За это время происходила запись данных с трех осей акселерометра. Отсчеты брались с частотой 100 Гц (каждые 10 мс).

Далее данные, полученные с акселерометров мобильного устройства, обрабатывались дискретным преобразованием Фурье с предварительной обрезкой участков в массиве данных, где испытуемый стоял на месте, после чего полученные спектры подвергались анализу на качественном уровне.

Всего в эксперименте участвовало 34 человека. Среди них без заболеваний опорно-двигательного аппарата 14 человек, со сколиозом 14 человек, с двухсторонним коксартрозом 2 человека, со смешанными заболеваниями опорно-двигательного аппарата 4 человека.

В данном исследовании использовался мобильный телефон HTC Desire 300 на платформе Android 4.1 со встроенным трехосевым акселерометром «BMA250» от компании «Bosch», полное описание характеристик и свойств которого представлено в [7].

Несмотря на описанные свойства данного акселерометра в [7], далеко не все из них можно использовать в настоящем устройстве из-за особенностей API операционной системы самого мобильного телефона.

В результате экспериментов были выявлены следующие типовые особенности в спектрах.

1. Максимальные амплитуды колебаний по всем осям акселерометра у всех испытуемых без заболеваний опорно-двигательного аппарата находятся на одних и тех же частотах.

2. У всех испытуемых без заболеваний опорно-двигательного аппарата наиболее сильные колебания присущи по направлению вверх-вниз (вертикальная ось движения).

3. Максимальные значения амплитуд колебаний трех осей у всех испытуемых со сколиозом смещены по частотам.

4. У всех испытуемых со сколиозом наиболее сильные колебания происходят по направлению вверх-вниз или вперед-назад (вертикальная или сагиттальная оси движения).

5. Также было выявлено, что с возрастом у человека появляются более сильные высокочастотные составляющие в спектре вне зависимости от наличия заболеваний опорно-двигательного аппарата.

Проведенный ряд экспериментов подтвердил возможность разработки алгоритмов, обнаруживающих патологии опорно-двигательного аппарата с помощью акселерометров мобильных устройств, и как следствие, создания системы удаленного мониторинга пациентов.

На основе полученных результатов были разработаны следующие алгоритмы обработки данных мониторинга с мобильного устройства.

1. Алгоритм определения точного участка движения человека в массивах данных, полученных с акселерометров мобильных устройств.

2. Алгоритм определения состояния опорно-двигательного аппарата пациента по спектру.

3. Алгоритм преобразования полученных спектров для выявления высокочастотных составляющих.

Также автором настоящей работы были разработаны мобильное приложение для пациентов, которое снимает показания с акселерометров, и web-интерфейс для просмотра полученных данных лечащим врачом с удаленного сервера (рисунок).

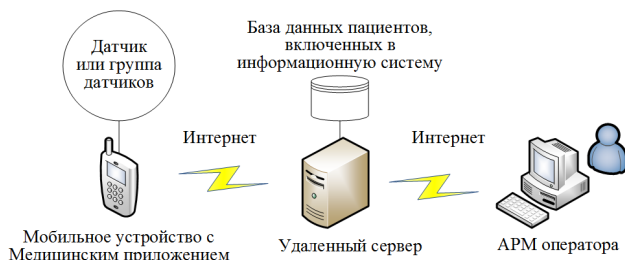


Рисунок. Структурная схема системы удаленного мониторинга

В данной работе проведен обзор существующих решений в области удаленного мониторинга пациентов и диагностики заболеваний опорно-двигательного аппарата и ряд экспериментов, результаты которых подтвердили возможность разработать алгоритмы, обнаруживающие патологии опорно-двигательного аппарата с помощью акселерометров мобильных устройств. Разработаны соответствующие алгоритмы и программные модули, являющиеся основными узлами в предлагаемой системе удаленного мониторинга пациентов.

Библиографический список:

1. *Гайдук А.А.* Диагностика и медицинская реабилитация функциональных нарушений опорно-двигательного аппарата при переломах таза у детей и подростков: Автореф. дис. ... д-ра мед. наук: 06.03.2014 / Санкт-Петербургский гос. педиатрический мед. университет. СПб., 2014. 38 с.
2. *Агеева Л., Гайдук А., Малкин Р., Сотникова Е.* Рентгенография и компьютерная оптическая топография в диагностике функциональных нарушений позвоночника и таза у детей и подростков// Современные проблемы науки и образования. 2016. №3. С. 1–13.
3. *Брагина С., Куликова О., Москалев В., Синьхао Л.* Применение компьютерной оптической топографии для оценки восстановления статических нарушений после переломов// Экология человека. 2015. №7. С. 45–49.
4. *Жиляев А.А.* Биомеханические и электрофизиологические критерии оценки функционального состояния опорно-двигательного аппарата нижних конечностей: Автореф. дис. ... д-ра техн. наук: 01.02.08/ Центр. научно-исследовательский ин-т травматологии и ортопедии им. Н. Н. Приорова. М., 2003. 34 с.
5. Пат. 35964 РФ, МПК А61В 5/11. Система диагностики заболеваний опорно-двигательного аппарата.
6. *Блескин Е., Львов С., Маслов Л., Смирнов Д., Шапин В.* Применение вибрационных неразрушающих методов диагностики в ортопедии// Российский журнал биомеханики. 2006. Т.10, №1. С. 15–29.
7. Web: <http://www1.futureelectronics.com>.

УДК 004

С.С. Ткачев

студент кафедры технологий защиты информации

С.В. Беззатеев

д-р техн. наук, доцент – научный руководитель

ОБЗОР УЯЗВИМОСТЕЙ SIM-КАРТ СТАНДАРТА GSM

GSM-сеть является одной из крупнейших сетей на Земле. На конец 2015 года количество подключений по стандарту GSM явилось самым большим по сравнению с остальными технологиями мобильной связи[1]. Хотя по прогнозам количество подключений по данному стандарту будет стремительно уменьшаться со временем, необходимо помнить, что новые технологии поддерживают обратную совместимость, таким образом, данный вид связи всё ещё остаётся актуальным. Большинство пользователей подключены к сети 24 часа в день, но не многие знают, как обеспечивается безопасность сети GSM, как она работает и насколько она хороша. Данная статья посвящается описанию механизмов защиты SIM-карт стандарта GSM и анализу их уязвимостей.

SIM-карта – это смарт-карта, используемая в мобильных устройствах, она имеет собственную ОС, память EEPROM, процессор, ПЗУ, ОЗУ и т. д. На карте располагаются следующие контакты (рис. 1): Vcc – питание; Reset – контакт управления картой; Clock – тактовая частота; GRD – «земля»; Vpp – напряжение программирования, которое используется при записи служебной информации; I/O – линия последовательного интерфейса ввода/вывода; Optional Pad for USB – контакты для обмена информацией с мобильным терминалом в режиме USB, обеспечивающем более высокую скорость передачи информации, чем через обычный I/O интерфейс SIM.

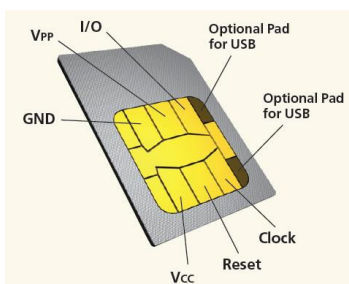


Рис. 1. Схема SIM-карты

На SIM-картах есть файловая система (рис. 2), так же как и на настольных компьютерах. Файлы бывают 2х типов: DF (Dedicated file – аналог папки) и EF (Elementary file – аналог обычного файла). Корневой DF файл называется MF (Master File).

Объем памяти, которую можно использовать для хранения информации в SIM, бывает разный и постепенно растет по мере развития технологий. На SIM-карте хранится информация следующего вида:

- 1) IMSI – International Mobile Subscriber Identity – последовательность цифр, которая не только идентифицирует конкретную SIM и ее владельца, но и указывает, какой оператор из какой страны ее выпустил;
 - 2) Ki – секретный ключ, который обеспечивает возможность для сети убедиться, что это действительно легитимная SIM-карта, а не мошенник, использующий чужой IMSI для получения доступа к услугам;
- 300

- 3) файлы для хранения другой служебной информации;
- 4) пользовательская информация – телефонная книга, принятые SMS-сообщения.

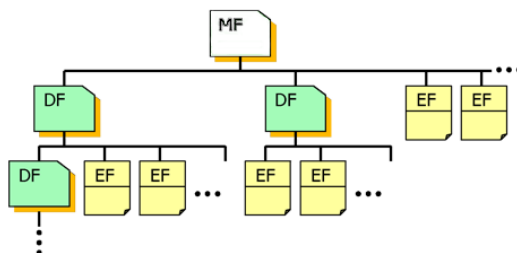


Рис. 2. Структура файловой системы SIM-карты

Питание и сигнал тактовой частоты для SIM-карты предоставляет телефон (или другой мобильный терминал), а также он полностью отвечает и за обмен информацией между устройствами – телефон всегда выступает в роли ведущего, а карта всегда является ведомым. Телефон посылает в SIM команды/запросы, а SIM только отвечает на них, сопровождая ответы информацией о статусе ответа.

Как происходит регистрация абонента в сети? На рис. 3 схематично представлены следующие шаги:

- 1) Телефон (Мобильная станция) абонента подключается к сети оператора.
- 2) Для подтверждения своей подлинности телефон посылает специальный идентификационный код, называемый TMSI (Temporary Mobile Subscriber Identity).
- 3) Центр Аутентификации (ЦА) генерирует 128-битное случайное число RAND и посылает его на Мобильную Станцию (МС).
- 4) МС зашифровывает полученное число RAND, используя свой секретный ключ K_i и алгоритм аутентификации A3.
- 5) МС отправляет обратно на ЦА сигнал отклика SRES (signed response) – первые 32 бита из последовательности, полученной на предыдущем шаге.
- 6) ЦА продлевает ту же операцию и получает 32 битную последовательность XRES (expected response).
- 7) После чего ЦА сравнивает SRES и XRES. В случае если оба значения равны, телефон считается аутентифицированным.
- 8) МС и ЦА вычисляют сессионный ключ шифрования, используя секретный ключ K_i и алгоритм формирования ключа A8 $K_c = A8_{k_i}(RAND)$

При этом на практике алгоритм аутентификации A3 и алгоритм формирования ключа A8 объединяются в один алгоритм, называемый COMP128 (он имеет множество модификаций COMP128v1, COMP128v2, COMP128v3, COMP128v4)[2]. COMP128 представляет собой хэш-функцию, на входе которая принимает 128-битную последовательность и на выходе возвращает 96-битную. Таким образом: SRES – это первые 32 бита от COMP128 ($K_i || RAND$), а K_c – последние 64 бита COMP128 ($K_i || RAND$), где $||$ – знак конкатенации[3].

Безопасность алгоритма COMP128v1 основывалась на предположении о неизвестности алгоритма для атакующего, но спецификации алгоритма каким-то образом частично попали в сеть и в 1998 году недостающие части были дополнены группой исследователей состоящей из Marc Briceno, Ian Goldberg и David Wagner. При этом были обнаружены следующие особенности:

- K_i – это 64-разрядное число и его последние 10 бит всегда равняются нулю;
- алгоритм состоит из нескольких раундов шифрования с подстановками;
- за счёт статических таблиц постановок последующие элементы зависят от предыдущих.

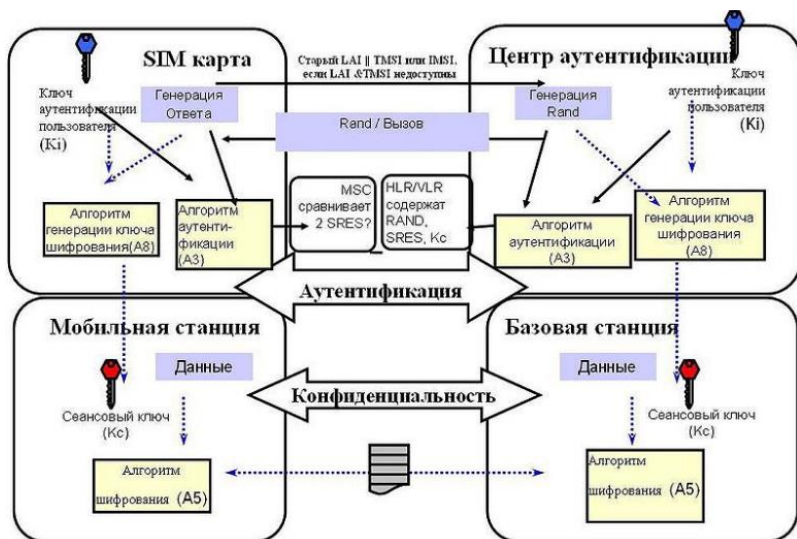


Рис. 3. Схема аутентификации SIM-карты в сети оператора

После того как стала известна структура алгоритма шифрования, а также указанные особенности, для извлечения секретного ключа SIM-карты, всё той же группой исследователей была предложена атака, основанная на парадоксе дней рождения, который используется в криптографии для поиска коллизий хэш-функций. А так как весь GSM трафик шифруется только с помощью ключа K_i и незашифрованного числа RAND, то злоумышленник, извлекая ключ, может прослушивать все разговоры и SMS-сообщения жертвы.

SIM-карта имеет ограниченное количество регистраций в сети оператора. Оно достаточно велико, чтобы не мешать обычному пользователю. Но атакующий, чтобы узнать секретный ключ, много раз посылает своё число RAND карте, чтобы получать различные значения отклика SRES, таким образом, прослеживая изменения и вычисляя секретный ключ. Благодаря описанным выше особенностям алгоритма шифрования, подбор ключа удавалось выполнить эффективно, исключая полный перебор, и, не расходуя полное количество попыток регистрации SIM-карты.

Осуществить атаку злоумышленник способен как физическим путём, используя SIM-ридер, так и удалённо, в этом случае злоумышленник имитирует легитимную сеть GSM и многократно использует процедуру аутентификации для извлечения K_i .

В связи с неустойчивостью алгоритма COMP128v1 к атакам на извлечение секретного ключа, а также как следствие к клонированию SIM-карт[4], операторам сотовой связи рекомендуется использовать алгоритмы COMP128v2 и выше.

Кроме SIM-карт, сама GSM-сеть обладает следующими уязвимостями:

- Атака «человек посередине». Из уже представленного алгоритма аутентификации абонента в сети видно, что МС никак не проверяет достоверность оператора сети. Эта проблема была известна при разработке протокола, однако не предполагалось, что злоумышленник сможет воспользоваться дорогостоящим специализированным оборудованием. Однако по прошествии многих лет такое оборудование стало доступным, подключив своё оборудование к сети оператора, злоумышленник сможет прослушивать трафик.

- Подмена номера. Данная уязвимость – расширение возможностей предыдущей, поскольку внутри сети оператора между станциями также не имеется подтверждений подлинности, следовательно

но, если станция злоумышленника подменит номер и перешлёт дальше запрос на другую станцию оператора, это воспримется за истину и таким образом осуществится подмена номера;

– Прослушивание переговоров сотрудниками оператора связи. Поскольку сотрудники имеют доступ к используемому оборудованию, они могут прослушивать передаваемый трафик, что, конечно же, недопустимо, но угроза такого вида имеет место[5].

В заключении отмечается, что на данный момент операторы сотовой связи практически или вовсе не используют SIM-карты на основе COMP128v1, переходя на более поздние его версии, которые не были взломаны. Современные SIM-карты, начиная с тех, что поддерживают технологию 3G входят в класс смарт-карт, называемых Java card, команды на которых написаны на сильно упрощённом языке Java[6]. Такой подход открывает возможности для поиска новых уязвимостей[7].

Библиографический список

1. Patrik Cerwall. Ericsson Mobility Report : On the pulse of the networked society [Электронный ресурс] / Anette Lundvall, Stephen Carson, Anette Lundvall – June, 2016 – Режим доступа: <https://www.ericsson.com/res/docs/2016/ericsson-mobility-report-2016.pdf>, свободный, [обращение 10.04.2017].
2. Максименко В.В., Афанасьев В.В., Волков Н.В. Защита информации в сетях сотовой подвижной связи / Под ред. д.т.н., проф. О.Б. Макаревича. – М.: Горячая линия–Телеком, 2007. – 360 с.
3. Хабрахабр [Электронный ресурс] : Безопасность GSM сетей: шифрование данных – Электрон. дан. – habrahabr.ru, 2013 – Режим доступа: <https://habrahabr.ru/post/186838/>, свободный, [обращение 10.04.2017].
4. Nuril Anwar. Forensic SIM Card Cloning Using Authentication Algorithm // Int. J. of Electronics and Information Engineering / Imam Riadi, Ahmad Luthfi. – 2016 г.
5. Marcin Olawski. Security in the GSM network [Электронный ресурс] – 2011, Режим доступа: https://ipsec.pl/files/ipsec/security_in_the_gsm_network.pdf, свободный, [обращение 10.04.2017].
6. Peter Edsbäcker. SIM cards for cellular networks : An introduction to SIM card application development // Computer Engineering C, 15 ECTS [Электронный ресурс] – 12 июня, 2011 – Режим доступа: <http://www.diva-portal.org/smash/get/diva2:423013/fulltext02>, свободный, [обращение 10.04.2017].
7. Karsten Nohl. Mobile Network Attack Evolution [Электронный ресурс] – May, 2014 – Режим доступа: <https://www.slideshare.net/phdays/140521karsten-nohl-phdmobileattackevolution-35161629>, свободный, [обращение 10.04.2017].

УДК 621.396

Д. В. Топошило

студент кафедры технологий защиты информации

С. В. Беззатеев

д-р техн. наук, профессор – научный руководитель

ТЕХНОЛОГИИ МОБИЛЬНОЙ СВЯЗИ 5G

С момента своего появления сети мобильной связи непрерывно эволюционировали. Возможности, которые предоставляют мобильные технологии, уже давно вышли за рамки голосовых услуг: мы получили новые способы общения и обмена данными. Возникло множество новых типов устройств и услуг – от смартфонов и планшетов до интеллектуальных электросчетчиков и бытовых приборов. Неудивительно, что их развитие и распространение привело к экспоненциальному росту трафика в сетях по всему миру и появлению новых и самых разнообразных требований к этим сетям в зависимости от типа устройства[1,2]. Например, таких как: высокая пропускная способность, поддержка одновременного подключения множества устройств, малое время отклика, надежность, низкая стоимость развертывания, эксплуатации и обслуживания сети, а также низкий расход батареи устройств.

В связи с этим, можно сделать вывод, что подход «одна технология для всех» не может быть эффективным для удовлетворения растущего числа потребностей со стороны бизнеса, общества и отдельных пользователей. Следовательно, требуется разработка новых, более совершенных и эффективных решений беспроводного доступа для решения поставленных задач. Усовершенствованные версии существующих технологий будут дополнены новыми разработками, более целостными на конкретные сценарии и способы применения. В перспективе такого развития к 2020 году появится то, что мы называем 5G, то есть набор органически интегрированных технологий радиодоступа.

Спецификация технологии беспроводной связи 5G всё ещё проходит окончательное утверждение, и должна быть готова к концу 2018 года. На сегодняшний день можно говорить лишь о потенциальных технологиях, применяемых в стандарте[3]:

1. Массивные MIMO. Для обработки сверхвысокого объема пользовательского трафика и достижения скорости передачи данных в несколько Гбит/с в конкретных сценариях предвидится сверхплотное развертывание сетей, базовые станции в которых будут использовать очень широкую полосу пропускания в верхних диапазонах частот с помощью новых технологий радиодоступа. Сверхплотные сети будут состоять из маломощных базовых станций, устанавливаемых с гораздо более высокой плотностью по сравнению с базовыми станциями, существующими сегодня. Эта, так называемая, технология MIMO, успешно применяемая в сетях четвертого поколения, найдет применение и в сетях 5G. При этом если в настоящее время в сетях используется MIMO 2x2 и 4x4, то в будущем число антенн должно увеличиться[4].

2. Проксимальная связь. В ситуациях, когда пользователи находятся близко друг от друга, и в особенности, когда информация специфична для конкретного места использования (безопасность движения, охрана общественного порядка), имеет смысл организация обмена данными непосредственно между устройствами по протоколу «устройство-устройство» (D2D), нежели посредством инфраструктуры сети. В рамках стандартов LTE уже делаются первые шаги к интеграции D2D в сетевые коммуникационные технологии. Преимуществом такой технологии является возможность переноса передачи данных в нелицензируемую часть спектра, что позволит дополнительно разгрузить сеть.

3. Мультитехнологичность. В промышленной связи, например, при обеспечении безопасности движения, современные радиосети не всегда могут удовлетворить требования к надежности. Надежность в значительной степени зависит от архитектуры и конфигурации сети, а также достаточного количества ресурсов для обработки пиковых нагрузок. В отношении инфраструктурного оборудования нового поколения необходимо будет дифференцировать различные виды трафика на уровне сети и в первую очередь обрабатывать критически важные. Значительная сложность заключается в сочетании сверхмалого времени задержки и высокой надежности. Для этого следует изменить соотношение между конструкцией управляюще-

го канала, кодированием, адаптивной модуляцией и управлением радиоресурсами. Там, где необходимо крайне малое время задержки, например, одна миллисекунда или ниже, нужно будет внедрить новую технологию, позволяющую обеспечить более короткие временные интервалы передачи данных[5].

4. Поддержка M2M-устройств. Для взаимодействия огромного количества маломощных M2M-устройств, разрабатываются альтернативные технологии полностью интегрируемые в сотовые. Это позволит обеспечить устройства с крайне высокими ограничениями по энергопотреблению возможностью непрерывного доступа. К примеру, капиллярные сети, использующие альтернативные технологии, будут соединяться с остальным миром посредством сотовых сетей.

5. Использование направленных антенн. Эффективным способом снижения уровня помех может быть использование направленных секторных антенн с узкими диаграммами направленности. В секторе такой направленной антенны сигнал излучается преимущественно в одну сторону (в зависимости от количества абонентов и их распределения по соте), а уровень излучения в противоположном направлении сокращается до минимума. Деление сот на секторы позволяет чаще применять частоты в сотах повторно.

6. Обеспечение энергоэффективности. Сокращение радиуса сот в плотных сетях, а также интеллектуальные возможности «спящего» режима базовых станций, минимизация сигнального трафика при обнаружении сети и синхронизации значительно уменьшат энергопотребление в сетях 5G[6].

Вышеперечисленные технологии потенциально составляют основу уже реализованных работ, которые проходят тестирование.

Первые тесты технологии 5G проведены в России в июне 2016 г. оператором связи МегаФон совместно с Huawei. В сентябре МТС при тестировании на канале связи с частотой 4,65-4,85 ГГц была достигнута скорость передачи данных 4,5 Гбит/с при полосе 200 МГц[7].

22 сентября 2016 г. Российский оператор сотовой связи «МегаФон», на бизнес-саммите в Нижнем Новгороде, запустили мобильный 5G-интернет. В ходе испытаний была достигнута скорость передачи данных 4,94 гигабита в секунду – через построенную сеть передавался панорамный ролик в разрешении 8K Ultra HD (7680×4320 точек).

Таким образом, из всего вышесказанного можно сделать вывод, что сети 5G, появление которых на рынке ожидается к 2020 г., будут ориентированы на существенное улучшение характеристик, в том числе качества обслуживания.

Поскольку принципы управления качеством обслуживания пользователей при переходе от 4G к 5G будут сохранены, основные усилия разработчиков 5G должны быть сосредоточены на виртуализации сетевых функций, которые позволят максимально эффективно использовать сети существующих и новых стандартов связи. Еще одним направлением развития сетей 5G станут алгоритмы классификации трафика, это обеспечит поддержку вызовов рынка по динамике изменения спроса на услуги и потребности абонентов.

Библиографический список

1. Бахур В. Издание о высоких технологиях. Коммерческие перспективы использования частот для сетей 5G // snews.ru, 17.03.2017 [Электронный ресурс] – Режим доступа. – URL: http://www.snnews.ru/news/line/2017-03-17_json_partners_consulting_kommercheskie_perspektivy.
2. Ericsson, 2013. Networked society essentials // ericsson.com, [Электронный ресурс] – Режим доступа. – URL: <http://www.ericsson.com/res/docs/2013/networked-society-essentials-booklet.pdf>.
3. Tudzarov A., Janevski T. Functional Architecture for 5G Mobile Networks: International Journal of Advanced Science and Technology Vol. 32, July, 2014.
4. Dumbre, N., Patwa, M. 5G wireless technologies – Still 4G auction not over, but time to start talking 5G: International Journal of Science, Engineering and Technology Research (IJSETR) Volume 2, Issue 2, February 2013.
5. Patil S., Patil V. A Review on 5G Technology: International Journal of Engineering and Innovative Technology (IJEIT) Volume 1, Issue 1, January 2012.
6. Rao N., Aasha S. A. 5G technologies: Journal of Global Research in Computer Science Volume 2, No. 7, July 2014.
7. МТС протестировал технологию 5G со скоростью 4,5 Гбит/с // ria.ru, 15.09.2016 [Электронный ресурс] – Режим доступа. – URL: <https://ria.ru/economy/20160915/1477056049.html>.

УДК 537.226.3

Е. В. Чекаль

студентка кафедры информационно-сетевых технологий

А. Д. Жуков

канд. техн. наук, доцент – научный руководитель

ИЗМЕРЕНИЕ ОТНОСИТЕЛЬНОЙ ДИЭЛЕКТРИЧЕСКОЙ ПРОНИЦАЕМОСТИ И ЗАВИСИМОСТИ ТАНГЕНСА УГЛА ДИЭЛЕКТРИЧЕСКИХ ПОТЕРЬ ОТ ТЕМПЕРАТУРЫ ТВЕРДЫХ И ЖИДКИХ ДИЭЛЕКТРИКОВ

В последнее время углеродосодержащие материалы широко применяются в радиотехнике сверхвысоких частот. Они применяются в качестве материалов, обеспечивающих радиотехнические характеристики радиоэлектронных устройств, для изготовления рефлекторов антенн, применяются в качестве защитных укрытий антенн и радиоэлектронной аппаратуры от мощного электромагнитного излучения, используются для маскирования объектов и т.д., и в качестве конструкционных материалов [1].

Поэтому остается актуальной проблема разработки и создания новых методов и средств исследования и измерения диэлектрической проницаемости и тангенса угла диэлектрических потерь трансформаторного масла, полиэтилена, полистирола [2]. Они могут применяться в судовой, авиационной и космической промышленности в качестве защитных укрытий от сверхмощных направленных электромагнитных полей или от воздействия направленных потоков заряженных частиц [3].

Основными характеристиками, определяющими электрические свойства диэлектриков при низких температурах, являются тангенс угла диэлектрических потерь $tg(\delta)$, диэлектрическая проницаемость E_r и электрическая постоянная E_0 [4].

Диэлектрическая проницаемость E_r – это отношение абсолютной диэлектрической проницаемости электроизоляционных материалов E_a к электрической постоянной E_0 :

$$E_r = E_a / E_0. \quad (1)$$

Значение относительной диэлектрической проницаемости электроизоляционных материалов можно вычислить, сравнив емкости двух конденсаторов, одинаковых по форме и геометрическим размерам с различными диэлектриками:

$$E_r = C_x / C_0, \quad (2)$$

где C_x – емкость конденсатора, между обкладками которого находится испытуемый диэлектрик; C_0 – емкость конденсатора при тех же геометрических размерах, но между обкладками которого имеется вакуум.

Диэлектрическая проницаемость E_r не является константой и зависит от температуры, частоты переменного тока, напряженности электрического поля, структуры диэлектриков, что нужно учитывать при ее измерении [5].

Для измерения относительной диэлектрической проницаемости и тангенса угла диэлектрических потерь используют мост переменного тока МД-16 (малогабаритный), принципиальная схема которого показана на рис. 1, где Тр – испытательный трансформатор; C_x – испытываемый объект; C_n – образцовый конденсатор высокого напряжения; Г – вибрационный гальванометр; R_1 – магазин сопротивлений; R_2 – постоянный резистор; C_1 – магазин емкостей, Э – экран, Р – разрядник.

Одним плечом моста МД-16 является испытываемый объект C_x (плечо АС), другим – образцовый конденсатор C_n (плечо ВС). В плечо AD включен магазин сопротивлений R_1 . Плечо BD состоит из постоянного резистора R_2 , равного 3184 Ом, и магазина емкостей C_1 , включенных параллельно друг другу. Между точками А и В включен вибрационный гальванометр G, являющийся индикатором равно-

Для выполнения измерений необходимо собрать схему соединений. Соединения осуществляются с помощью мыши перетаскиванием контактов слева направо. После того, как схема собрана, следует включить источник напряжения и измерительный мост, выбрать напряжение и диэлектрик, для которого будет происходить измерение. Далее задаём значение R_1 и R_2 таким образом, чтобы установилась минимальная чувствительность моста, т.е. чтобы эллипс помещался на экране-индикаторе. После уравновешивания моста нажимаем кнопку «Вычисление зависимости», с помощью которой выводятся значения C_x , R_x , $tg(\delta)$, гальванометра, диэлектрической проницаемости и отображается индикатор равновесия вибрационного гальванометра. Виртуальный лабораторный стенд после сборки и вычисления зависимости представлен на рис. 3.

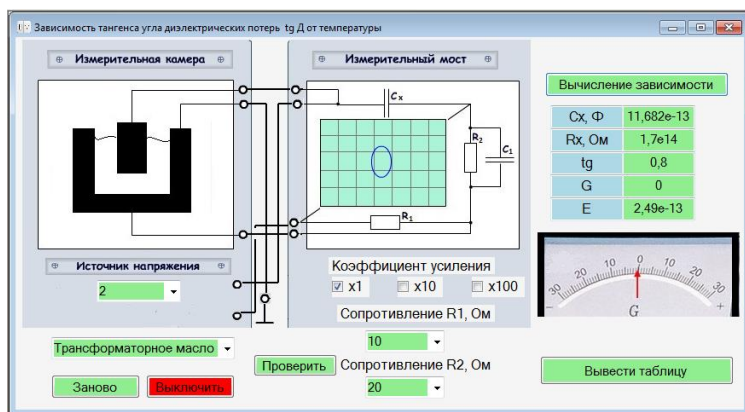


Рис. 3. Виртуальный лабораторный стенд после сборки и вычисления зависимости

Для каждого диэлектрика выводится таблица зависимости тангенса угла диэлектрических потерь от температуры, по которой строится график зависимости.

Справочный график зависимости $tg(\delta)$ трансформаторного масла от температуры представлен на рис. 4.

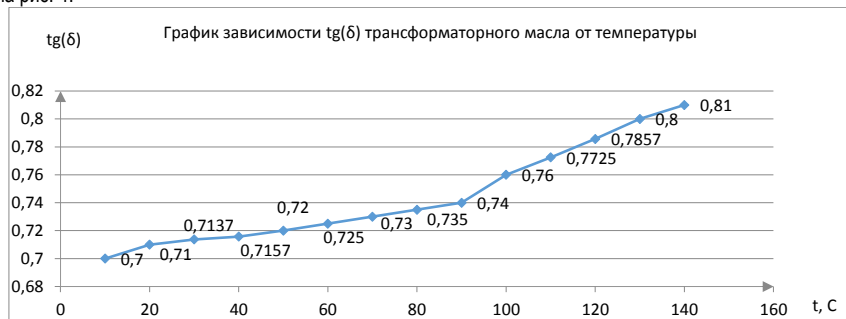


Рис. 4. Справочный график зависимости $tg(\delta)$ трансформаторного масла от температуры

В программе строится график зависимости $tg(\delta)$ трансформаторного масла от промежуточных значений температур, который представлен на рис. 5.

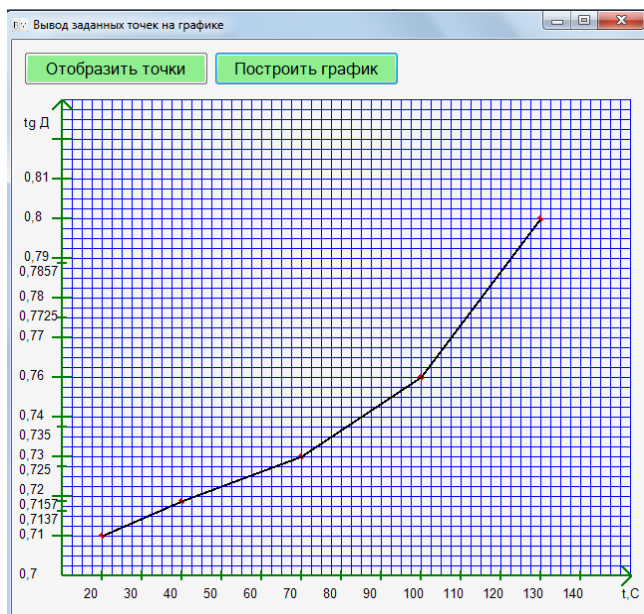


Рис. 5. График зависимости $tg(\delta)$ трансформаторного масла от промежуточных значений температур

Для полиэтилена и для пропилена таким же образом строятся графики зависимости $tg(\delta)$ от температуры. Таким образом, виртуальный лабораторный стенд заменяет работу моста МД-16.

Библиографический список

1. 1001 Реферат [Электронный ресурс] – Режим доступа: <http://www.1001referat.ru/nauka-i-tehnika/12232-dielektricheskaya-pronitsaemost-i-metodika-ee-izmereniya>
2. Научная библиотека диссертаций и авторефератов disserCat [Электронный ресурс]// Методы и средства измерения комплексной диэлектрической проницаемости импедансных материалов – Режим доступа: <http://www.dissercat.com/content/metody-i-sredstva-izmereniya-kompleksnoi-dielektricheskoi-pronitsaemosti-impedansnykh-materiy#ixzz2yJ2RaeX5>
3. Богородицкий Н.П. и др. Электротехнические материалы. Учебник для вузов 7 – е издание. Ленинград: Энергоатомиздат, 2015. 304 с.
4. Серебряков А. С. Электротехническое материаловедение. Электроизоляционные материалы. Питер: Маршрут, 2014. 275 с.
5. Васильев Н.П. Лабораторные работы по электроматериаловедению – 6-е изд. – М.: «Высшая школа», 2013. 93с.
6. Шишкин А.В. Исследование диэлектрической проницаемости диэлектрических потерь в твердых диэлектриках – Новосибирск: кафедра «Автоматизированные электротехнологические установки», 2009. 59с.

УДК 004.93'1

Шарапова Д.А.

студент кафедры Инфокоммуникационных систем

М. Р. Гильмутдинов

Канд. техн. наук, доцент – научный руководитель

АНАЛИЗ АЛГОРИТМА СОПРОВОЖДЕНИЯ ОБЪЕКТОВ TRACKING-LEARNING-DETECTION

Главная особенность, необходимая машинам для того, чтобы они могли «видеть», понимать и реагировать на окружение, – это их способность обнаруживать и отслеживать объекты интереса [1]. Предположим, что у нас есть видеопоток, взятый с камеры, содержащий различные объекты, входящие и выходящие из поля зрения камеры. Нам дана ограниченная область, определяющая объект нашего интереса в кадре, наша задача – автоматически определять эту область или оповещать, что объект невидим в каждом последующем кадре.

Тогда, можно дать следующее определение:

Видео трекинг – это процесс оценки положения одного или более объектов с течением времени по данному видеоряду [1].

Задача *видеотрекинга* в [1] формулируется следующим образом:

Изначально имеется некая последовательность видеок кадров. На первом кадре фиксируются регионы одного или нескольких объектов. И на последующих видеок кадрах оживается положение этих объектов.

Оскую модель процесса видеотрекинга [1] (рис. 1):

Пусть $I = \{I_k : k \in \mathbb{N}\}$. – множество кадров в видеопоследовательности, такое что $I_k \in E_I$ – кадр в момент времени k , определенный в E_I . – пространство всех возможных изображений. Каждый кадр имеет следующие параметры: высота W и ширина H . Интенсивность пикселя $I_{x,y}$, где x, y – положение пикселя может принимать значение от 0 до 255. Отслеживание одного объекта может быть сформулировано как оценка временного ряда $x = \{x_k : k \in \mathbb{N}\}$. по множеству дискретных моментов времени с индексом k , основываясь на информации из I . Векторы $x_k \in E_S$ – состояния объекта, а E_S – пространство состояний. Временной ряд x – траектория объекта в E_S . I_k может быть отображен на E_O – пространство особенностей-признаков, которое выделяет информацию, соответствующую проблеме отслеживания. Наблюдение объекта закодировано в $z_k \in E_O$, где E_O имеет меньшую размерность в сравнении с пространством оригинального изображения, E_I .

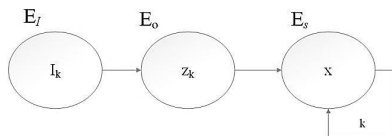


Рис. 1. Поток информации между пространствами: E_I – пространство возможных изображений, E_O – пространство признаков (наблюдения) объекта, E_S – пространство состояний объекта 1

1. Типовой алгоритм видео-трекинга.

Для начала стоит определить основные шаги в сопровождении объекта (см. рис. 2). Сначала на вход поступает видеопоследовательность. На первом кадре определяется область объекта интереса. Затем из кадра извлекается информация об объекте и происходит обучение, благодаря которому детектор в дальнейшей работе сможет обнаружить объект и определить объект. После чего запускается

рекурсивный трекинг, в процессе которого извлекается следующий кадр, из него извлекаются интересные нас признаки. Затем детектор и классификатор определяют объект, после чего трекер пытается найти изменение в положении объекта.

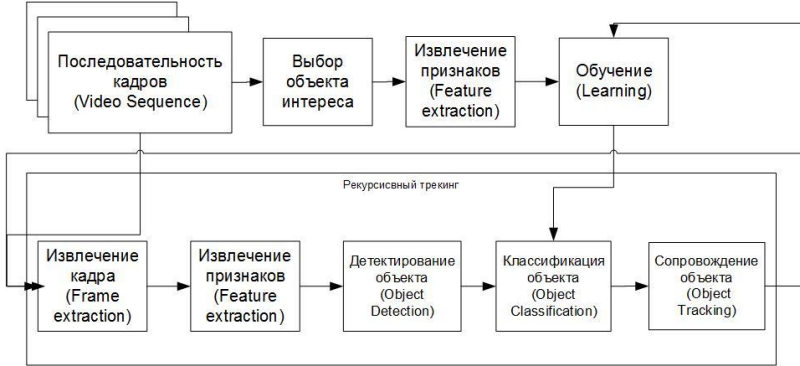


Рис. 2. Типовой алгоритм сопровождения объекта

2. Алгоритм TLD.

В данной работе был рассмотрен алгоритм TLD, также известный как Predator. Это достаточно простой и быстрый онлайн алгоритм для трекинга неизвестных объектов в видеопотоках. Объект интереса определяется ограничительной рамкой в первом кадре, затем одновременно отслеживает объект, обучается на его внешнем виде и обнаруживает объект всякий раз, когда он появляется в видео [2]. Была введена структура TLD (Tracking-Learning-Detection), которая разбивает долгосрочный трекинг на 3 подзадачи: трекинг, обучение, детектирование (рис. 3). Каждая из подзадач имеет свои компоненты, которые работают одновременно. Трекер отслеживает объект из кадра в кадр. Детектор локализует все появления, которые были обнаружены и корректирует трекер, если необходимо. Обучение оценивает ошибки детектора и обновляет их, чтобы избежать ошибок в будущем.

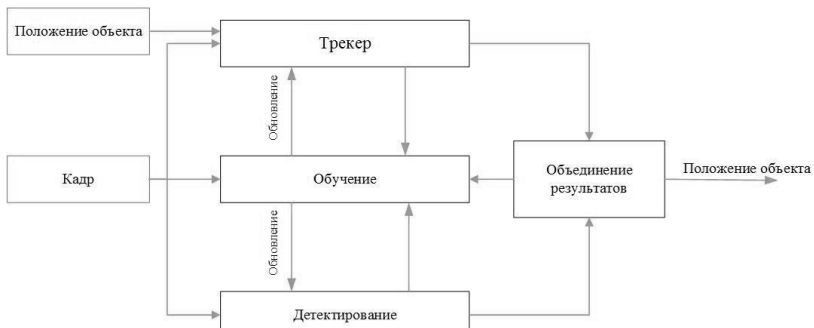


Рис. 3. Упрощенная блок-схема алгоритма TLD 2

Также была введена новая обучающая парадигма (модель) – P-N Learning (P-N обучение). Детектор оценивает каждый кадр в видео. Его ответы анализируются двумя «экспертами»: P определяет, пропущенные обнаружения и N – ложные обнаружения. Ошибки прибавляются к тренирующему набору, и детектор заново обучается, чтобы избежать эти ошибки в будущем.

В процессе анализа выяснилось, что алгоритм очень чувствителен к начальным данным, а именно к ограничительной рамке объекта, в область которой может входить как сам объект, так и окружающие его детали или задний план.

3. Оценка эффективности:

Качество трека зависит от поставленной задачи и от содержания видео ряда. Чтобы избежать субъективности визуальной оценки, требуется автоматическая процедура – объективная оценка. Для исследования были взяты следующие классификационные критерии: True Positive, False Positive, False Negative, True Negative, Recall, Precision, F-score. Также в качестве независимой оценки была взята оценка качества отслеживания одной цели (Single-target scores) [1]:

Это простая мера ошибки между оценкой состояний и настоящим (Ground Truth) положением \tilde{x} – евклидово расстояние:

$$d(x, \tilde{x}) = \sqrt{\sum_{i=1}^D (x_i - \tilde{x}_i)^2} \quad (1)$$

где D – размер пространства состояний E_s . Чем меньше этот параметр, тем лучше, т.к. центры двух ограничительных рамок находятся близко друг к другу, а значит отклонение в отслеживании объекта минимально. В исследовании находилось расстояние между ограничительными рамками объекта, полученных в результате трекинга, и рамок истинного положения объекта (Ground Truth), затем находили среднее значение по всем расстояниям для каждой последовательности.

В рамках исследований были взяты последовательности кадров с разными сценариями, такими как: движение камеры, частичные и полные преграждения, изменение позы, изменение освещенности, изменение масштаба, наличие похожих объектов. Оценив работу алгоритма на подобных последовательностях, можно будет сказать, насколько хорошо алгоритм справляется с основными проблемами видео трекинга. В процессе анализа алгоритма выяснилось, что он очень чувствителен к начальным данным, в частности к ограничительной области объекта. Для исследования влияния выбора ограничительной рамки на эффективность работы алгоритма были проделаны эксперименты с изменением размера начальной рамки объекта. Наилучшие результаты представлены в табл. 1.

Таблица 1

Оценка эффективности алгоритма TLD
(наилучшие результаты, полученные в ходе экспериментов по изменению начальных данных)

Последовательность	frames	prec	rec	fscore	Dist	TP	TN	FN	FP
David	761	1.00	1.00	1.00	5.31	761	0	0	0
Jumping	313	1.00	0.99	1.00	5.64	310	0	3	0
Pedestrian1	140	1.00	0.91	0.95	6.37	127	0	13	0
Pedestrian2	338	1.00	0.97	0.99	4.92	259	72	7	0
Pedestrian3	184	1.00	1.00	1.00	1.99	156	28	0	0
Car	945	0.96	1.00	0.98	4.94	856	46	4	39
Motocross	2665	0.96	0.72	0.82	5.88	1015	1208	397	45
Volkswagen	8576	1.00	0.95	0.97	7.35	4872	3428	269	7
Carchase	9928	0.91	0.68	0.78	7.75	6423	1215	2237	53
Panda	3000	0.87	0.86	0.87	10.24	2703	262	27	8

Так как авторы не рассматривали критерий расстояния, сравнение проводилось по трем классификационным критериям: Recall, Precision, F-score (табл. 2). Как видно из сравнения, достигнуть тех же результатов, что и авторы, не удалось. Однако в некоторых последовательностях получены результаты лучше, что говорит о том, что авторы, возможно, не учли влияние ограничительной рамки объекта на эффективность работы алгоритма.

Таблица 2

Сравнение результатов, полученных в ходе экспериментов по изменению начальных данных, и результатов приведенных авторами алгоритма TLD

Последовательность	Исх.код			статья		
	<i>prec</i>	<i>rec</i>	<i>fscore</i>	<i>prec</i>	<i>rec</i>	<i>fscore</i>
David	1.00	1.00	1.00	1.00	1.00	1.00
Jumping	1.00	0.99	1.00	1.00	1.00	1.00
Pedestrian1	1.00	0.91	0.95	1.00	1.00	1.00
Pedestrian2	1.00	0.97	0.99	0.89	0.92	0.91
Pedestrian3	1.00	1.00	1.00	0.99	1.00	0.99
Car	0.96	1.00	0.98	0.92	0.97	0.94
Motocross	0.96	0.72	0.82	0.89	0.77	0.83
Volkswagen	1.00	0.95	0.97	0.80	0.96	0.87
Carchase	0.91	0.68	0.78	0.86	0.70	0.77
Panda	0.87	0.86	0.87	0.58	0.63	0.60

Из полученных результатов, можно сделать вывод о том, что ограничительная рамка должна иметь размер достаточно информативный для алгоритма и не затрагивать детали заднего плана, тогда можно вынести максимум эффективности от работы алгоритма.

Библиографический список

1. *Cavallaro E. Maggio*, Video Tracking: theory and practice, John Wiley & Sons, 2011, p.263;
2. *Kalal Z., Mikolajczyk K., Matas J.* Tracking-Learning-Detection, IEEE transactions on pattern analysis and machine intelligence, vol. 6, no. 1, January 2010, pp 1409-1422;
3. *Kalal Z., Mikolajczyk K., Matas J.* Forward-Backward Error: Automatic Detection of Tracking Failures, International Conference on Pattern Recognition, 23-26, August, 2010, pp 2756-2759;
4. *David J. Fleet, Yair Weiss*, Optical Flow Estimation, Handbook of Mathematical Models in Computer Vision, pp 237-257, January 2006;
5. *Jean-Yves Bouget*, Pyramidal Implementation of the Lucas Kanade Feature Tracker, Intel Corporation Microprocessor Research Labs, 2000;
6. *Viola P., Jones M.* Rapid Object Detection using a Boosted Cascade of Simple Features, Computer Vision and Pattern Recognition, IEEE Computer Society Conference on In 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, 15 April 2001, pp. 511-518;
7. *Ozuysal M., Fua P., Lepetit V.* Fast Keypoint Recognition in Ten Lines of Code, IEEE Conference on Computing Vision and Pattern Recognition, 2007;
8. *Everingham M., Van Gool L., Williams C., Winn J., Zisserman A.* The pascal visual object classes (VOC) challenge, International Journal of Computer Vision, June 2010;
9. *Zdenek Kalal*. Official source code for TLD, <https://github.com/zk00006/OpenTLD> ;
10. Tracking-Learning-Detection, Dr. Zdenek Kalal <http://personal.ee.surrey.ac.uk/Personal/Z.Kalal/index.html>.

УДК 629.7

Н. А. Галыга, Д. А. Черных

студенты учебного военного центра

П. М. Безняков

канд. техн. наук – научный руководитель

ИССЛЕДОВАНИЕ ЧИСЛЕННЫХ АЛГОРИТМОВ СИСТЕМЫ НАВЕДЕНИЯ БАЛЛИСТИЧЕСКОГО ЛЕТАТЕЛЬНОГО АППАРАТА ПО ПРОГНОЗУ ПАССИВНОГО УЧАСТКА ТРАЕКТОРИИ

Введение

Комплексный алгоритм наведения на цель баллистического летательного аппарата, реализующий на борту метод наведения по требуемой скорости включает в свой состав алгоритм расчета параметров движения ракеты на прогнозируемый момент окончания пассивного участка траектории. Этот алгоритм осуществляет численное интегрирование дифференциальных уравнений движения аппарата.

Исходя из специфики движения межконтинентальных баллистических летательных аппаратов, большая часть их траектории представляет собой пассивный участок, где движение осуществляется только под действие гравитационных сил Земли. В этом случае алгоритм расчета параметров движения аппарата на прогнозируемый момент окончания пассивного участка траектории вносит наибольшую методическую погрешность [1].

Цель работы – продемонстрировать подход к исследованию влияния численных алгоритмов интегрирования на качество работы алгоритма расчета параметров движения ракеты на прогнозируемый момент окончания пассивного участка траектории.

Постановка задачи

Примем следующие допущения:

- 1) идеальность работы системы стабилизации баллистического аппарата;
- 2) отсутствие вращения Земли вокруг своей оси;
- 3) сферичность формы Земли с радиусом $R_3 = 6371$ км;
- 4) полет аппарата на пассивном участке траектории является невозмущенным.

Качество работы алгоритма расчета параметров движения на прогнозируемый момент окончания пассивного участка траектории может быть охарактеризовано абсолютной погрешностью вычисления сферической дальности полета баллистического аппарата:

$$\eta = |L_{ан} - L_{ч}|, \quad (1)$$

где $L_{ан}$ – истинное значение сферической дальности; $L_{ч}$ – рассчитанное с помощью бортового алгоритма значение сферической дальности.

Задача нахождения сферической дальности состоит в:

- 1) решении системы дифференциальных уравнений (2) с начальными данными (3):

$$\begin{cases} \frac{d\vec{v}(t)}{dt} = \vec{g}(\vec{r}(t)); \\ \frac{d\vec{r}(t)}{dt} = \vec{v}(t); \end{cases} \quad (2)$$

$$\begin{aligned} \vec{v}(t_0) &= \vec{v}_0; \\ \vec{r}(t_0) &= \vec{r}_0. \end{aligned} \quad (3)$$

где $\vec{v}(t)$ – скорость центра масс полета аппарата в момент времени t ;

$\vec{r}(t)$ – координаты центра масс аппарата в момент времени t ;

\vec{g} – ускорение свободного падения в точке $\vec{r}(t)$ пространства;

\vec{v}_0 – скорость центра масс аппарата в момент отключения двигателей последней ступени;

\vec{r}_0 – координаты центра масс аппарата в момент отключения двигателей последней ступени.

2) решении уравнения (4) относительно $t_{\text{Ц}}$:

$$r(t_{\text{Ц}}) = r_{\text{Ц}}. \quad (4)$$

3) расчете сферической дальности

$$L = R_3 \arccos\left(\frac{\vec{r}_0 \vec{r}_{\text{Ц}}}{|\vec{r}_0| |\vec{r}_{\text{Ц}}|}\right), \quad (5)$$

где $\vec{r}_{\text{Ц}} = \vec{r}(t_{\text{Ц}})$ – координаты центра масс аппарата в точке окончания пассивной траектории.

За истинное значение сферической дальности $L_{\text{ан}}$ можно принять значение полученное на основе решения системы дифференциальных уравнений (2)–(3) аналитическим методом.

За рассчитанное значение дальности $L_{\text{ц}}$ алгоритмом наведения можно принять значение, полученное на основе решения системы дифференциальных уравнений методом, реализованным в бортовых алгоритмах при тех же начальных данных.

Выберем в качестве начальных данных алгоритма значения, величины которых позволяют практически выполнить полет баллистического летательного аппарата на межконтинентальные дальности:

$h_0 = 250$ км – высота над поверхностью Земли в начале пассивного участка траектории (высота бросания);

$v_0 = 7.1$ км/с – скорость в начале пассивного участка траектории (скорость бросания);

θ_0 – оптимальный по дальности угол бросания (требует нахождения);

$h_{\text{Ц}} = 100$ км – высота над поверхностью Земли в конце пассивного участка траектории.

Решим задачу нахождения истинного значения сферической дальности $L_{\text{ан}}$, для использования в оценке численных алгоритмов наведения по формуле (1).

Решение задачи

Решение задачи нахождения сферической дальности $L_{\text{ан}}$ на основе решения системы (2) с начальными данными (3) может быть получено аналитическим способом в рамках эллиптической теории движения баллистических летательных аппаратов [2]. В этом случае предполагается, что Земля обладает геометрической и механической (в смысле распределения масс) симметрией.

Общее решение системы уравнений невозмущенного движения тела (2) в центральном поле притяжения приводит к трем основным интегралам, из которых следуют основные выводы для анализа невозмущенного движения баллистического летательного аппарата [3].

Выражение

$$v^2 - 2 \frac{\gamma M_3}{r} = h \quad (6)$$

имеет название «интеграл энергии» или «интеграл живых сил». Здесь h – скалярная постоянная интегрирования.

Выражение

$$\vec{r} \times \vec{v} = \vec{\sigma} \quad (7)$$

имеет название «интеграл площадей». Здесь $\vec{\sigma}$ – векторная постоянная интегрирования.

Выражение

$$\vec{v} \times \vec{\sigma} - \gamma M_3 \frac{\vec{r}}{r} = \vec{\lambda} \quad (8)$$

называется «интегралом Лапласа», в котором $\vec{\lambda}$ – векторная постоянная – «вектор Лапласа».

Все постоянные интегрирования могут быть найдены из начальных данных (3).

Из интегралов (6), (7), (8) путем некоторых преобразований можно получить уравнение эллиптической орбиты в полярной системе координат [3], где полюс системы расположен в геометрическом центре Земли, а полярная ось совпадает по направлению с вектором Лапласа.

Уравнение эллиптической орбиты примет следующий вид:

$$r = \frac{\rho}{1 + e \cos \vartheta}, \quad (9)$$

где $\rho = \frac{\sigma^2}{\gamma M_3}$ – фокальный параметр, характеризующий размер орбиты; $e = \frac{\lambda}{\gamma M_3}$ – астрономический эксцентриситет, характеризующий форму орбиты; ϑ – полярный угол положения центра масс объекта на эллиптической орбите.

Задача нахождения сферической дальности сводится к нахождению разности между полярными углами ϑ_0 и ϑ_{II} точек начала и окончания пассивного участка траектории соответственно.

Выделив из формулы (9) полярный угол можно записать выражение для определения сферической дальности для произвольного угла бросания θ :

$$L(\theta) = R_3 \left[-\arccos \left(\frac{\rho(\theta) - (R_3 + h_{II})}{e(\theta)(R_3 + h_{II})} \right) - \arccos \left(\frac{\rho(\theta) - (R_3 + h_0)}{e(\theta)(R_3 + h_0)} \right) + 2\pi \right]. \quad (10)$$

Значения сферических дальностей, рассчитанных в соответствии с начальными данными для различных углов бросания, представлены на графике (рис. 1).

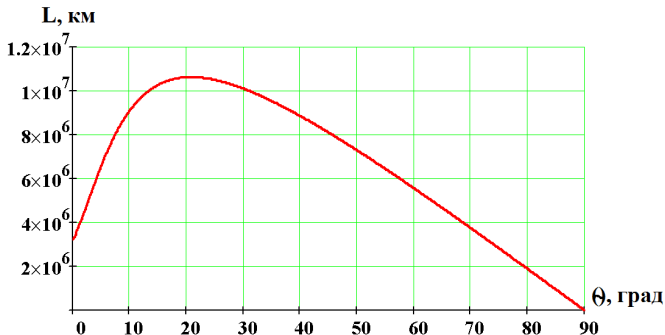


Рис. 1. Дальность полета баллистического летательного аппарата

Оптимальный угол бросания находится путем нахождения экстремума функции $L(\theta)$:

$$\theta_0 = \max_{\theta \in [0, 90] \text{ град.}} \{L(\theta)\}. \quad (11)$$

В результате решения задачи (11) получим

$$\theta_0 = 20.82 \text{ град}$$

или

$$\theta_0 = 0.363 \text{ рад.}$$

Таким образом, за истинное значение сферической дальности $L_{ан.}$ примем сферическую дальность, рассчитанную по формуле (11) с учетом начальных данных и найденного оптимального угла бросания:

$$L_{ан.} = 10622025 \text{ м.}$$

Выводы

Качество работы алгоритма расчета параметров движения ракеты на прогнозируемый момент окончания пассивного участка траектории может быть охарактеризовано абсолютной погрешностью в виде выражения (12):

$$\eta = |10622025 - L_{ч.}|. \quad (12)$$

Библиографический список

1. Разоренов Г. Н., Титов Ю. Ф., Бахрамов Э. А. Системы управления летательными аппаратами: учебник для вузов. М.: Машиностроение, 2003. – 586 с.
2. Агафонов А. С., Зимин Г. В., Сырцев А. И. и др. Основы теории полета баллистических ракет и космических аппаратов. Калинин: ВКА ПВО, 1972. – 428 с.
3. Гуков В. В., Чернов В. В. Баллистический полет летательного аппарата. М.: МАИ, 1989. – 58 с.

УДК 681.5: 53.08

П. И. Гусейнов, А. А. Канцер

студенты Института военного образования

В. В. Котович

канд. техн. наук – научный руководитель

ИНТЕРФЕЙС ЛАБОРАТОРНОЙ УСТАНОВКИ ДЛЯ ПОВЕРКИ ГИРОПРИБОРОВ

Разработанный интерфейс предназначен для обмена данными между ПК и автоматизированной установкой КПА-5А [1, 2], разработанной на базе неавтоматизированной штатной установки КПА-5. Установка КПА-5А служит для поверки гироскопов. Разработанный интерфейс позволяет одновременно управлять установкой и её 3D моделью, созданной в редакторе SolidWorks. Для разработки этого интерфейса использовались программы LabVIEW 2009 и SolidWorks 2012 [3]. Кроме перечисленных программ в работе использовались дополнительные модули, устанавливаемые отдельно.

Для LabVIEW 2009 использовались модули:

– 2009.v9.0.Simulation.Interface.Toolkit-TBE – модуль обеспечивает прямую интеграцию пакета Simulink компании MathWorks и LabVIEW;

– 2009.v9.0.Real.Time.Module-TBE – модуль для графической разработки приложений сбора данных и управления, работающих в режиме жесткого реального времени;

– 2009.v9.0.SoftMotion.Module-TBE – модуль обрабатывает графические приложения, отвечающие за управление движением;

Для SolidWorks 2012 использовались модули:

– SW Simulation – модуль для расчета на прочность конструкций (деталей и сборок) в упругой зоне;

– SW Motion – модуль для комплексного динамического и кинематического анализов механизмов.

Установка КПА-5А с закреплённым на ней гироскопом представлена на рис. 1. В правой части рисунка размещается компьютер, с помощью которого осуществляется управление установкой. На мониторе компьютера отображается интерфейс управления, выполненный в LabVIEW, и 3D-модель установки, выполненная в SolidWorks.



Рис. 1. Установка КПА-5А

В установке КПА-5А изменение положения рабочей поверхности в трёх плоскостях с закреплённым на ней гиросприбором, осуществляется с помощью трёх шаговых двигателей. Измерение угла отклонения рабочей поверхности осуществляется с помощью 3-степенного датчика-акселерометра, обеспечивающего точность измерения углов до 4 угл. мин. 3D-модель установки выполнена в упрощённом виде, отображает лишь подвижные части установки и не содержит моделей платы управления, трёх шаговых двигателей и рабочей поверхности с расположенным на ней гиросприбором. 3D-модель установки изображена на рис. 2.

Упрощённая модель позволяет использовать гораздо меньше ресурсов ПК, тем самым положительно влияя на его быстродействие, сохраняя при этом наглядное изображение установки.

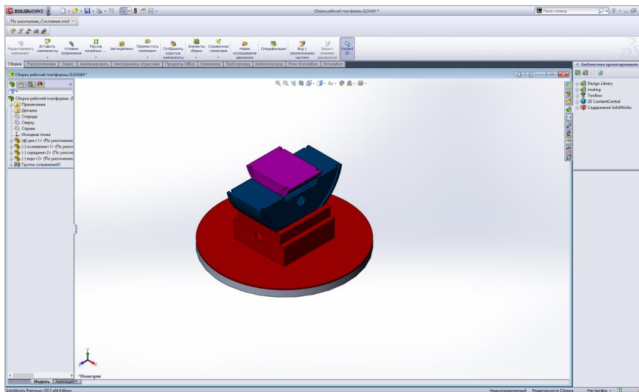


Рис. 2. 3D-модель установки

На рис. 3 изображены: дерево проекта (левая часть рисунка) и сам интерфейс управления установкой (правая часть рисунка), реализованные в программе LabView 2009. Дерево проекта содержит виртуальный прибор управления установкой и саму 3D-модель установки, которая включает в себя оси всех двигателей установки. В интерфейсной части (справа) находятся следующие кнопки (сверху вниз): выбор COM-порта для микроконтроллера (зеленая область), ввод команды управления и вывод отправленного значения (розовая область), индикаторы отправления команды на соответствующий двигатель (голубая область).

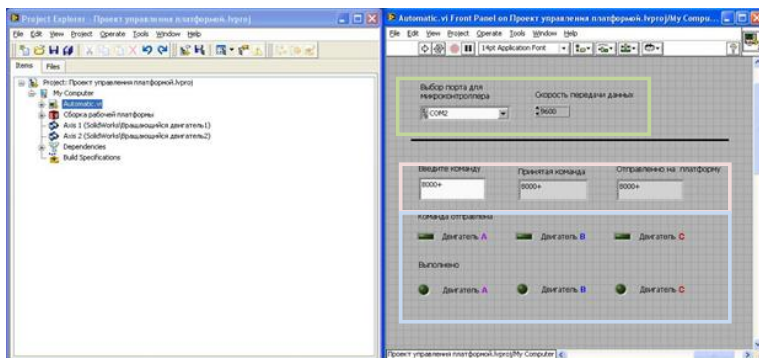


Рис. 3. Дерево проекта в LabView (слева) и рабочий интерфейс (справа)

Формат команды имеет следующий вид: «A010+».

Первый символ: А, В или С – выбор двигателя;

Со второго по четвертый символ – количество градусов, на которые должен повернуться двигатель;

Пятый символ – знак направления вращения («+» – по часовой или «-» – против часовой стрелки).

В графическом коде программы LabVIEW [4] (в окне блок-диаграмм) используются соответствующие математические модули для обработки команды, которую должны одинаково распознать установка КПА-5 и 3D-модель в SolidWorks. Первая часть графического кода приведена на рис. 4.

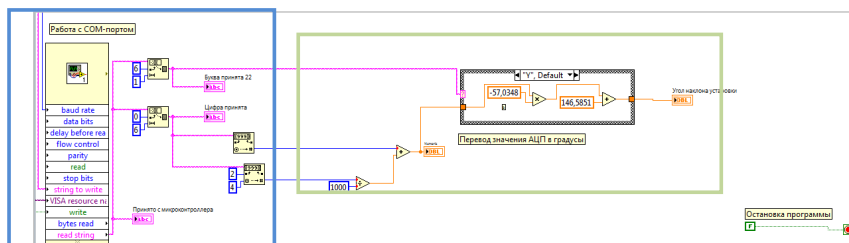


Рис. 4. Графический код в окне блок-диаграммы

В голубой области расположены элементы, отвечающие за связь программы LabVIEW с микроконтроллером. Слева расположен SubVII (файл LabVIEW, подключенный как отдельная программа), инкапсулирующий настройки com-порта, настройки обработки информации. Правее расположены элементы, извлекающие определенные символы из принимаемой строки с микроконтроллера. В зелёной зоне расположен модуль, переводящий значения АЦП акселерометра установки в значения градусов совершенного.

Вторая часть графического кода приведена на рис. 5.

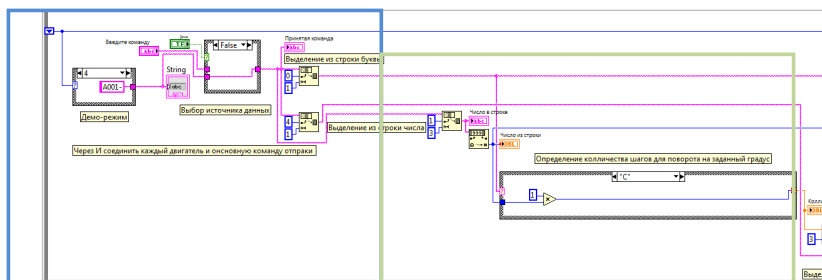


Рис. 5. Графический код в окне блок-диаграммы

В голубой области расположен переключатель источника данных, отправляемых на установку КПА-5 и 3D-модель. Если переключатель (элемент True/False) расположен в положении TRUE, то элемент switch принимает данные, вводимые пользователем, если он в положении FALSE, то элемент switch принимает данные, генерируемые LabVIEW. В зелёной области происходит умножение введенного числа в градусах на количество шагов для передачи этого числа шаговым двигателями.

Третья часть графического кода приведена на рис. 6.

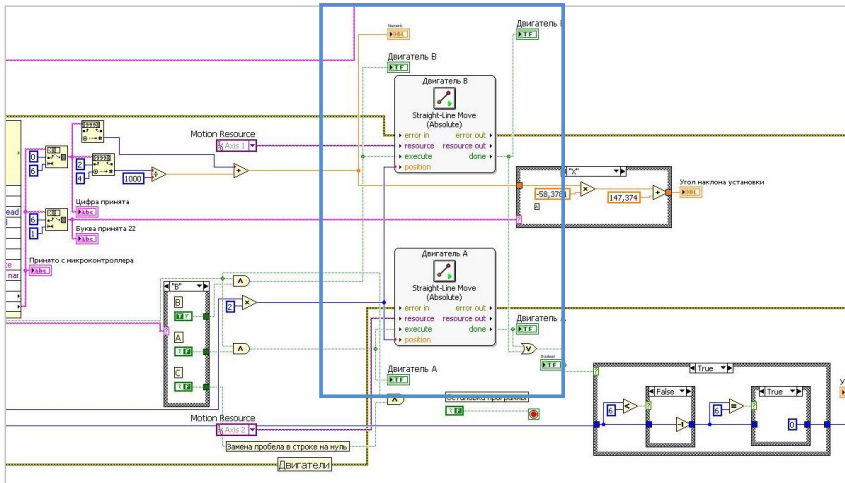


Рис. 6. Графический код в окне блок-диаграммы

В синей области расположены модули Straight-LineMove, отвечающие за связь между LabVIEW и SolidWorks.

Итоговый результат с запущенным интерфейсом и 3D-моделью показан на рис. 7.

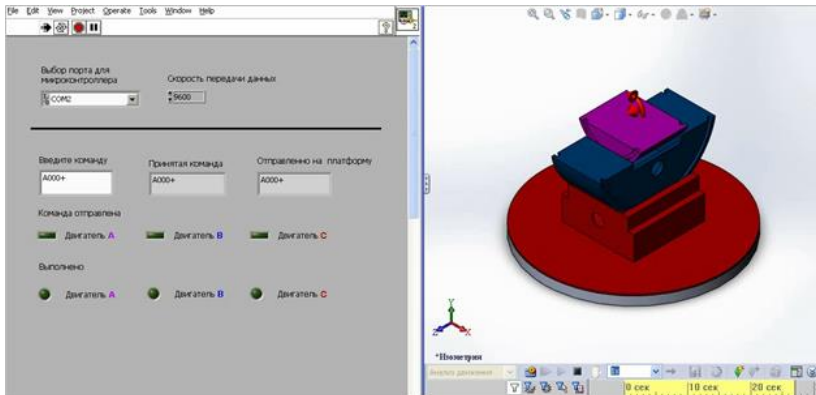


Рис. 7. Слева – интерфейс для управления установкой (LabVIEW), справа – 3D-модель сборки (SolidWorks)

Разработанный интерфейс позволил упростить операции управления оператором установкой, уменьшить вероятность ошибки его работы за счет создания наглядной 3D-модели установки КПА-5.

Простота управления установкой и 3D-моделью была достигнута за счёт создания эргономичного интерфейса панели управления в LabVIEW. Этот подход позволяет разработать интерфейс практически к любому прибору или установке, управление которых может быть реализовано на ПК.

Библиографический список

1. *Смолий А. А.* Автоматизированная установка для проверки работоспособности гироскопических приборов летательных аппаратов / Шестьдесят четвертая студенческая научно-техническая конференция: сб. докл. Ч. 2. Гуманитарные науки. СПб.: ГУАП, 2011.
2. *Цветков А. М., Плотников М. А.* Автоматизированная лабораторная установка по проверке гироскопических приборов / Шестьдесят пятая студенческая научно-техническая конференция: сб. докл. Ч. 2. Гуманитарные науки. СПб.: ГУАП, 2012.
3. *Алямовский А.* Инженерные расчёты SolidWorks. ДМК-Пресс, СПб., 2010.
4. *Трэвис Дж., Кринг Дж.* LabVIEW для всех. Четвёртое издание. – ДМК, М., 2011.

УДК 65.015.2

Р. Р. Колобеков

студент Института военного образования

П. М. Безняков

канд. техн. наук – научный руководитель

ИСПОЛЬЗОВАНИЕ МЕТОДА КРИТИЧЕСКОГО ПУТИ В ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССАХ ЭКСПЛУАТАЦИИ СПЕЦИАЛЬНОГО ВООРУЖЕНИЯ

Введение

Эксплуатация вооружения включает, совокупность различных процессов, направленных на поддержание техники в готовности к применению. Для реализации эксплуатационных процессов с высоким качеством и в ограниченные сроки необходимо их планировать. Важнейшим условием успешного выполнения задач эксплуатации вооружения является четкое, правильное и заблаговременное планирование. Отсюда вытекает необходимость научно обоснованных методов сетевого планирования процессов эксплуатации [3].

Наиболее известными являются метод критического пути (Critical Path Method – CPM) и система планирования и руководства программой разработок (Program Evaluation and Review Technique – PERT). В этих методах проекты рассматриваются как совокупность некоторых взаимосвязанных процессов (видов деятельности, этапов или фаз выполнения проекта). В статье рассмотрены теоретические основы метода критического пути и пример его использования [2].

Постановка задачи

В качестве исходных данных метода является список работ технологического процесса с перечнем всех работ, их последовательности и продолжительности [1]. Данный список оформляется в виде таблицы работ.

Исходные данные формируются в результате выполнения следующих действий:

1. Формулировка состояния исходного события технологического процесса.
2. Формулировка конечной цели комплекса работ.
3. Составление таблицы работ.

Формализация задачи

Формализация задачи сетевого планирования заключается в:

- 1) построении сетевой модели технологического процесса (сетевой графика);
- 2) упорядочивании нумерации событий сетевой графика.

Математической базой метода сетевого планирования и управления служит теория графов (рис. 1). В основе метода сетевого планирования и управления лежит сетевая модель эксплуатационного процесса, с помощью которой отображается объем предстоящих работ, последовательность их выполнения, а также логическая взаимосвязь работ.

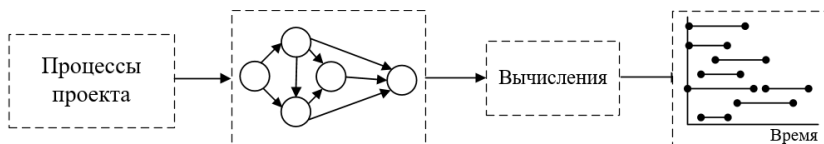


Рис. 1. Основные этапы выполнения метода CPM

Решение задачи

С целью выявления критического пути и резервов времени для работ, расположенных на других путях, производится расчет сетевого графика и определение его временных параметров. Основные параметры событий сетевого графика технологических процессов эксплуатации вооружения вычисляются по сетевому графику. Для визуализации результатов расчетов полученные данные отображаются в вершинах графа.

На сетевом графике вычисляется [1]:

- 1) ранний срок наступления события t_p для исходного события срок наступления равно 0;
- 2) поздний срок поступления события t_n ;
- 3) ранний срок начала работы t_{pn} ;
- 4) ранний срок окончания работы t_{po} ;
- 5) полный резерв времени работы R_n ;
- 6) свободный резерв времени работы R_c .

Пример расчета указанных величин будет рассмотрен ниже.

Пример расчета технологического процесса

В качестве примера рассмотрим методику построения сетевого графика технологического процесса проведения проверки систем агрегата военной техники.

Таблица 1

Составление таблицы работ

Код работы	Наименование работы	Какой работе предшествует	Продолжительность, мин
1	Подача регламентной аппаратуры на агрегат	2,3	10
2	Подготовка регламентной аппаратуры	4	40
3	Сборка схемы для проведения проверки	4	30
4	Проверка систем агрегата	5,6	150
5	Свертывание регламентной аппаратуры	7	30
6	Приведение систем в исходное положение	7	40
7	Убытие	–	10

Построение сетевой модели необходимо начинать с изображения исходного события. Данную таблицу необходимо представить в виде сетевого графика. После проверки сетевой модели на отсутствие ошибок необходимо провести нумерацию событий. Нумерация событий проводится методом «вычеркивания дуг» (рис. 2).

Расчет параметров сетевого графика и его анализ проводится в следующей последовательности:

1. Расчет ранних сроков наступления событий по формуле (1)

$$t_p(2) = \max\{t_p(1) + t(1,2)\} = \max\{0 + 10\} = 10.$$

2. Расчет поздних сроков наступления событий по формуле (2)

$$t_n(6) = \min\{t_n(7) - t(6,7)\} = \min\{240 - 0\} = 240.$$

3. Ранний срок начала работы по формуле (3)

$$t_{pn}(7,8) = t_p(8) = 250.$$

4. Ранний срок окончания работы по формуле (4)

$$t_{po}(2,4) = t_p(2) + t(2,4) = 10 + 40 = 50.$$

5. Полный резерв времени по формуле (5)

$$R_n(5,6) = t_n(6) - t_p(5) - t(5,6) = 240 - 200 - 40 = 0.$$

6. Свободный резерв времени по формуле (6)

$$R_c(3,4) = t_p(4) - t_p(3) - t(3,4) = 50 - 40 - 10 = 10.$$

После расчета параметров сетевого графика получаем заполненный граф (рис. 2).

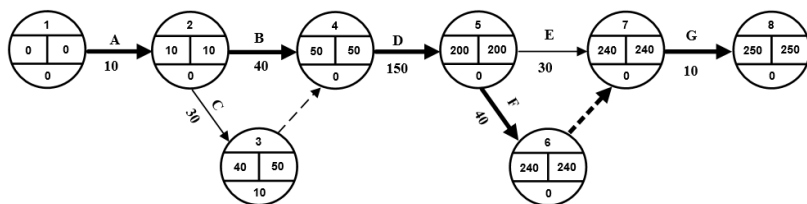


Рис. 2. Сетевой график технологического процесса

Критический путь для удобства на графе выделяется жирными стрелочками, в верхнем секторе написан номер события, в левом секторе – ранний срок наступления события, в правом секторе – поздний срок наступления события, в нижнем секторе – резерв времени события. Для удобства проверки и дальнейших расчетов данные сведены в табл. 2.

Таблица 2

Результаты расчета сетевого графика

Мероприятия	Символ	Время, мин	Раннее начало, мин	Позднее окончание, мин	Полный резерв, мин	Свободный резерв, мин
1–2(крит.)	A	10	0	10	0	0
2–3	C	30	10	50	10	0
2–4(крит.)	B	40	10	50	0	0
3–4		0	40	50	10	10
4–5(крит.)	D	150	50	200	0	0
5–6(крит.)	F	40	200	240	0	0
5–7	E	30	200	240	10	10
6–7(крит.)		0	240	240	0	0
7–8(крит.)	G	10	240	250	0	0

В табл. 2 в первой колонке помечены мероприятия, как критические (крит.), так и некритические. Также в таблицу заносятся значения, вычисленные на графе, и записывается расчет полного и свободного резервов времени.

Решение задач с использованием специализированного программного обеспечения

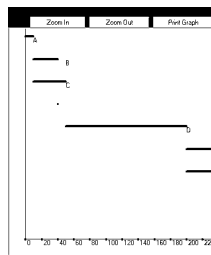
Данный метод прост в реализации, но при наличии большего количества процессов и работ расчеты существенно усложняются. Поэтому вычисления параметров сетевого графика на практике

проводятся с помощью специального программного обеспечения (рис. 3). Одним из таких программных продуктов является программа TORA [2].

Рассмотрим порядок работы с программой на примере. В окне «Inputgrid» заносим в таблицу исходные данные задачи планирования (см. рис. 3,а).

Row	From Node	To Node	Activity Symbol	Duration
1	1	2	A	10.0
2	2	3	B	30.0
3	2	4	C	40.0
4	3	4	A	0.0
5	4	5	D	150.0
6	5	6	E	40.0
7	5	7	F	30.0
8	6	7	G	20.0
9	7	8	G	10.0

Step	Node	Earliest Time	Step	Node	L.P.
1	1	10.00	8	8	
2	2	10.00	10	7	
3	3	40.00	11	6	
4	4	50.00	12	5	
5	5	200.00	13	4	
6	6	240.00	14	3	
7	7	240.00	15	2	
8	8	16.00	16	1	



а) Таблица с исходными данными в программном представлении

б) Результаты расчета сетевого графика

в) Отображение затрат времени на каждую работу в виде графа

Рис. 3. Интерфейс работы программы

На выходе, получаем таблицу с результатами и расчетами значений для построения временно-го графика (рис. 3,б). Временной график отображается в отдельном окне (рис. 3,в). Красным цветом отображается критический путь (A, C, D, E, G), который нельзя смещать во времени, синем цветом выделяются работы, которые можно смещать в заданных промежутках времени (рис. 3,в). Сокращение времени выполнения критических работ может быть достигнуто за счет повышения мастерства исполнителей работ и механизации ручных операций.

Заключение

Основными достоинствами сетевого метода планирования и управления являются:

1. Простая математическая основа и программная реализация.
2. Не требуется существенной переработки модели при возникновении изменений в ходе выполнения работы.
3. Имеется возможность осуществлять в ходе выполнения работ прогнозирование хода выполнения оставшихся работ.
4. Позволяет руководителю в процессе управления сосредоточить основное внимание на главных участках – на критических работах.

А главным недостатком является то, что время выполнения каждого процесса работы является детерминированным, что не согласуется с реальностью. На самом деле время выполнения работ является случайной величиной и должно характеризоваться плотностью распределения, что не учитывает рассмотренный метод.

Библиографический список

1. Чурочкин А. П., Морсаков Ю. С. Построение сетевых моделей технологических эксплуатационных процессов и расчет их параметров с использованием: уч.-метод. материал. М.: Военная академия имени Ф.Э. Дзержинского, 1989. 35 с.
2. Таха Х.А. Введение в исследование операций: научно-популярное издание. СПб.: Вильямс, 2005. 901 с.
3. Бездомов В.Л., Карякин А.Г. Эксплуатация специального вооружения: конспект лекций. Пермь, 1999. 121 с.

УДК 681.5: 53.08

В. В. Кульбик

студент Института военного образования

А. А. Бакланов

научный руководитель

РАЗРАБОТКА ИЭТР ДЛЯ ПОДГОТОВКИ АВИАЦИОННЫХ СПЕЦИАЛИСТОВ

С развитием высоких технологий в аэрокосмической области появились возможности создавать более сложные изделия. В связи с этим повысился объем знаний, необходимый для эксплуатации современной авиационной техники. При этом объём и сроки подготовки высококвалифицированных инженеров остались неизменны. Решение данной проблемы требует привлечения современных методов обучения: использование современных виртуальных тренажеров, электронных учебников, обучающих компьютерных программ и т.д.

В учебном военном центре при ГУАП ведется работа по созданию интерактивного электронного технического руководства (ИЭТР). Использование ИЭТР в учебном процессе позволит будущим авиационным специалистам повысить эффективность своего обучения и освоения материала.

Таким образом, целью данной работы является создание ИЭТР для воздушного судна. Для её реализации были поставлены следующие задачи:

- анализ и синтез этапов проектирования ИЭТР;
- изучение и выбор технической документации на воздушное судно;
- создание трехмерной модели с использованием систем автоматизированного проектирования (САПР);

- разработка ИЭТР на изучаемое воздушное судно.

В качестве объекта для создания ИЭТР выбран самолет Ан-26.

Последовательность создания ИЭТР представлена на рис. 1.

Она включает в себя этапы:

- подбора документации, САПР, редактора ИЭТР;
- создания 3D-модели ЛА;
- формирования ИЭТР;
- тестирования и совершенствования ИЭТР.

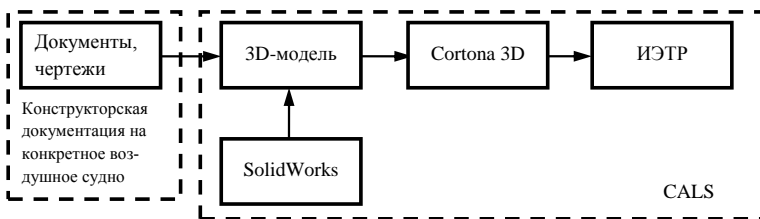


Рис. 1. Последовательность реализации ИЭТР

Исходя из поставленных задач и особенностей реализации ИЭТР, составлен перечень требований для современных САПР:

- создавать 3D-модели;
- возможность конвертировать созданные 3D-модели;
- взаимодействовать с Excel, Word и другими приложениями Windows;

– наличие технической литературы.

Для создания 3D-модели применяется программа SolidWorks, которая позволяет выполнять операции со сборками.

Этап создания 3D-модели воздушного судна (ВС) является наиболее трудоёмким процессом. Это обуславливается сложностью реализации объемных элементов и их количеством. Качество проработки каркасной группы воздушного судна, при моделировании, влияет на информативность конечной сборочной модели ВС. На рис. 2,а представлена часть модели; Ан-26 – а, продемонстрировано качество проработки сборочной модели Ан-26 – б.

На этапе формирования ИЭТР созданная 3D-модель самолета Ан-26 импортируется в Cortona3D RapidAuthor наряду с текстовой документацией и 2D моделями (схемами) ЛА, где производится комбинирование анимации трехмерной модели и текста.

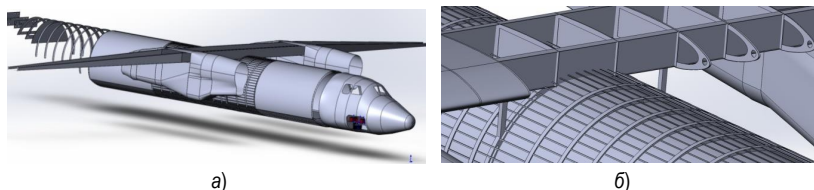


Рис. 2. Сборочная модель Ан-26: а) в профиле; б) расширенном виде

Cortona3d RapidAuthor – комплексный программный продукт, позволяющий интегрировать информацию об объекте в текстовом, графическом, звуковом формате и является основой создаваемого ИЭТР. Интерфейс RapidAuthor позволяет создавать как обучающие программы, так и тестирующие. В состав RapidAuthor входят программы: RapidManual, RapidCatalog, RapidGenerator.

RapidManual – программа, позволяющая создавать трехмерные интерактивные технические руководства по ремонту оборудования, техническому обслуживанию, сборке/разборке оборудования;

RapidGenerator – программа конвертации 3D-изображений из САПР программы.

Критериями выбора данного редактора стали: удобный интерфейс, наличие обучающей литературы, широкие возможности экспорта/импорта из САПР программы; присутствие русского языка в интерфейсе, простой процесс моделирования.

Созданное таким образом интерактивное 3D/2D-руководство реализует 3D-анимацию и текстовое описание процедур, которые синхронизированы между собой. Когда воспроизводится шаг анимации, выделяется его описание и наоборот. Кроме этого ИЭТР позволяет реализовать манипуляцию трехмерной модели для изучения с разных точек обзора.

Руководство обеспечивает возможность быстрого перехода к нужным шагам процедуры воздушного судна, при этом отдельные шаги процедур можно повторить многократно для качественного усвоения материала.

Результатом работы является создание и реализация в учебном процессе ИЭТР самолета Ан-26 для обучения будущих авиационных специалистов.

Разработка ИЭТР ведется в соответствии с требованиями и государственными стандартами:

– Р50.1.029-2001 "Информационные технологии поддержки жизненного цикла изделия. Интерактивные электронные технические руководства. Общие требования к содержанию, стилю и оформлению"

– ГОСТ 2.052-2006 "Единая система конструкторской документации. Электронная модель изделия. Общие положения".

– ГОСТ 2.053-2006 "Единая система конструкторской документации. Электронная структура изделия. Общие положения".

Библиографический список

1. Ан-26. Техническое описание. кн. 1. Летно-технические характеристики.
2. Ан-26. Техническое описание. кн. 2. Конструкция самолета.
3. *Гребеньков О.А.* Конструкция самолетов.
4. *Житомирский Г.И.* Конструкция самолетов.
5. <http://www.cortona3d.com/ru>
6. *Зиновьев Д.* Основы проектирования в SolidWorks 2016.

УДК 004.048

И. М. Левицкий

студент военной кафедры

А. В. Старостин

п/п-к, канд. техн. наук – научный руководитель

В. В. Грибов

п/п-к, канд. техн. наук – научный руководитель

АНАЛИЗ ИНТЕЛЛЕКТУАЛИЗАЦИИ УПРАВЛЕНИЯ ОБОРОНОЙ РОССИЙСКОЙ ФЕДЕРАЦИИ. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ

Анализ современных военных конфликтов показал, что высокий уровень информационного обеспечения боевых действий войск становится определяющим фактором достижения стратегического и оперативно-тактического превосходства над противником, следовательно, успешность ведения боевых действий зависит от использования программно-аппаратных комплексов, которые решают задачи своевременного комплексного информационного обеспечения боевых действий, реализуя функцию интеллектуальной поддержки принятия решений.

Решение задач обеспечения обороноспособности страны на требуемом уровне предполагает создание военных технических систем нового поколения, разрабатывающихся с учетом концепций Единого информационного пространства (ЕИП) Вооруженных сил и сетецентрической войны.

«Единое информационное пространство представляет собой специальным образом упорядоченную и взаимосвязанную совокупность информационных, вычислительных и телекоммуникационных ресурсов, организованных и функционирующих во времени и пространстве с целью повышения качества управления Вооруженными Силами и оружием в мирное и военное время» [1]. Соответственно под концепцией сетецентрической войны подразумевается способ ведения боевых действий, при котором все участники (органы управления, военная техника, живая сила) объединены единым информационным пространством.

Вооруженные силы, объединенные ЕИП, получают возможность качественно нового обмена информацией. В результате общая ситуационная осведомленность такова, что позволяет обеспечивать необходимые сотрудничество и самосинхронизацию, повышает устойчивость и скорость передачи команд, что, в свою очередь, резко повышает эффективность выполнения боевой задачи и ведения боевых действий в целом.

Одним из самых главных этапов организации ЕИП является создание центрального узла, обладающего огромными вычислительными мощностями и возможностями, который и будет формировать ЕИП. Для ВС РФ таким узлом является Национальный центр управления обороной Российской Федерации (НЦУО), заступивший на боевое дежурство в декабре 2014 года. Национальный центр управления обороной Российской Федерации (НЦУО) предназначен для обеспечения централизованного боевого управления Вооруженными силами Российской Федерации, обеспечения управления повседневной деятельностью Военно-воздушных сил и Военно-Морского Флота, сбора, обобщения и анализа информации по военно-политической обстановке в мире, на стратегических направлениях и по общественно-политической обстановке в Российской Федерации в мирное и военное время [4].

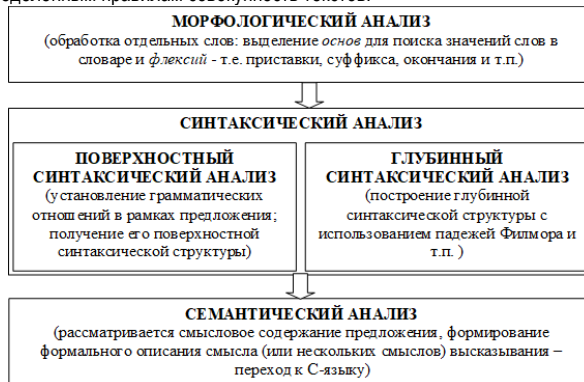
Основу НЦУО составляют три центра управления: по специальным вопросам, боевого управления и управления повседневной деятельностью. Главным же из них является Центр боевого управления, на него направлена большая часть мощностей НЦУО, суперкомпьютер НЦУО имеет производительность в 16 петафлопс¹ и объем памяти в 236 петабайт, тем самым являясь самым мощным военным суперкомпьютером на планете [5]. В секунду суперкомпьютер способен обрабатывать объем ин-

¹ Флопс – внесистемная единица, используемая для измерения производительности компьютеров, показывающая, сколько операций с плавающей запятой в секунду выполняет данная вычислительная система.

формации равный 50 библиотекам имени Ленина [5]. Центр боевого управления осуществляет постоянный мониторинг военно-политической обстановки в мире, что позволяет прогнозировать развитие угроз для Российской Федерации или её союзников. И самое главное, обеспечивает управление применением ВС, а также войск и воинских формирований, не входящих в структуру Минобороны.

Информация, поступающая в НЦУО из различных источников, представляется в форме: аудио, видео, фото и текстов. Возникает необходимость преобразования этих данных в единую формализованную структуру, удобную для хранения и дальнейшего использования с целью анализа и извлечения знаний. Такой структурой можно считать представление данных в виде специальных текстовых файлов, с охранением структур и смысловых отношений информационными блоками, которые формируют базу знаний и базу данных системы поддержки принятия решений.

Обработка полученных структур должна осуществляться путем выполнения основных этапов анализа текстовых форм, проводимых человеком: морфологический анализ, синтаксический анализ и семантический анализ (рисунок). Если не брать во внимание специфические задачи, такие как анализ эмоциональной окрашенности, фонетический анализ и т.д., то с точки зрения задач анализа текста, можно выделить синтаксический и семантический анализы. Синтаксический – это анализ линейной последовательности слов с целью построения дерева зависимостей. Цель – анализ структуры предложения и отношения его компонентов. Семантический - анализ зависимости слова или фразы от общего контекста. Разрешение проблем полисемии, синонимии и т.д. Основой является подобранная и обработанная по определённым правилам совокупность текстов.



Этапы обработки текста

Предметную область анализа (множество структурированных и неструктурированных текстовых данных, полученных из различных источников T) можно представить в следующем виде:

$$T = \{L_1, L_2, \dots, L_i\};$$

$$L_i = (E, R),$$

где L_i – анализируемый текст; $E = \{e_1, e_2, \dots, e_n\}$ – множество объектов анализа; $R = \{r_1, r_2, \dots, r_n\}$ – множество отношений между ними. Процесс анализа текста, производимый человеком можно представить следующим выражением:

$$C = S_m(I_h(V_m(L))),$$

где L – высказывание на естественном языке; V_m – формальный лингвистический анализ; I_h – семантическая интерпретация оператора; C – формальное представление высказывания; S_m – формальная процедура окончательной обработки.

Формальное описание области анализа, формируемое при компьютерном анализе, должно остаться таким же, как и при анализе человеком, т.е. должно выполняться следующее условие:

$$I_h(L_f) \approx I_m(L_f),$$

где I_h – семантическая интерпретация оператора; L_f – результат формального лингвистического анализа; I_m – семантическая интерпретация компьютера.

Следовательно, компьютерный анализ текста можно описать как

$$C = S_m(I_m(V_m(L))) \gg [2].$$

Компьютерный анализ текста предполагает использование систем, построенных на основе методов искусственного интеллекта. Лучшими в этой области являются СИИ основанные на нейрокибернетических методах Data Mining², т.е. СИИ на основе искусственных нейронных сетей.

Анализ СИИ старого поколения показал, что они имеют «жесткую» структуру, работают в рамках заданного алгоритма, правил и созданной структуры, данные особенности предопределили выбор использования нейросетевой технологии. В процессе функционирования одна часть нейронной сети может рассматривать состояние другой и изменять топологию другой части нейронной сети. Это связано со способностью нейрона динамически изменять выполняемую функцию и соединения с другими нейронами.

Нейронные сети – самообучающиеся системы, они учатся на примерах. Когнитолог (инженер по знаниям) формирует исходные данные путем разработки алгоритмов функционирования и правил обучения НС и предоставляет сети право найти любой способ достижения цели, до тех пор, пока не будет достигнут требуемый результат. Таким образом, сеть начнет понимать, чего от нее хотят, и пытается найти наилучший способ достижения этой цели. Именно поэтому нейросетевые СИИ «умнее» своих аналогов основанных на базах знаний, и являются на данный момент лучшим выбором для создания интеллектуальных систем для анализа данных представленных различными структурами или формами.

Выбор исходных данных для решения задач систем поддержки принятия решения зависит от формирования релевантного запроса. В результате от точности выбранных данных зависят результаты, представляемые лицу, принимающему решение для окончательного выбора. Предложенный подход анализа формализованных данных в виде специальных текстовых форм позволяет описать все возможные связи объектов и обеспечить оперативный доступ к ним.

Технологию семантического анализа с помощью нейронных сетей недавно внедрил Яндекс в свою поисковую систему. Понимая смысл поискового запроса, а не выполняя лишь поиск по ключевым словам в запросе, можно значительно увеличить полноту и точность выдаваемой информации. Рассмотрим сравнение результатов (табл. 1) поисковой выдачи нейронной модели и модели, основанной на соответствии слов запроса и текста (BM25). В качестве примера возьмем запрос [келлская книга] и посмотрим, какое значение принимают факторы на разных заголовках [3].

Таблица 1

Заголовок страницы	BM25	Нейронная модель
Келлская книга википедия	0.91	0.92
Ученые исследуют келлскую книгу вокруг света	0.88	0.85
Bookofkellswikipedia	0	0.81
Ирландские иллюстрированные евангелия viiiivv	0	0.58
Икеа гипермаркеты товаров для дома и офиса ikea	0	0.09

² Data Mining – методы обнаружения в данных ранее неизвестных, нетривиальных, практически полезных и допустимых интерпретации знаний, необходимых для принятия решений.

«Все факторы в Яндексе нормируются в интервал [0;1]. Вполне ожидаемо, что BM25 имеет высокие значения для заголовков, которые содержат слова запроса. И вполне предсказуемо, что этот фактор получает нулевое значение на заголовках, не имеющих общих слов с запросом. Теперь обратите внимание на то, как ведет себя нейронная модель. Она одинаково хорошо распознаёт связь запроса как с русскоязычным заголовком релевантной страницы из Википедии, так и с заголовком статьи на английском языке! Кроме того, кажется, что модель «увидела» связь запроса с заголовком, в котором не упоминается келлская книга, но есть близкое по смыслу словосочетание («ирландские евангелия»). Значение же модели для нерелевантного заголовка существенно ниже» [3].

Теперь изменим слова запроса [евангелие из келлса], оставим его смысл прежним и посмотрим результат (табл. 2) [3].

Таблица 2

Заголовок страницы	BM25	Нейронная модель
Келлская книга википедия	0	0.85
Ученые исследуют келлскую книгу вокруг света	0	0.78
Bookofkellswikipedia	0	0.71
Ирландские иллюстрированные евангелия viiiivv	0.33	0.84
Икеа гипермаркеты товаров для дома и офиса ikea	0	0.10

«Для BM25 переформулировка запроса превратилась в настоящую катастрофу – фактор стал нулевым на релевантных заголовках. А нейронная модель демонстрирует отличную устойчивость к переформулировке: релевантные заголовки по-прежнему имеют высокое значение фактора, а нерелевантный заголовок – низкое. Кажется, что именно такое поведение мы и ожидали от штуки, которая претендует на способность «понимать» семантику текста» [3].

Таким образом, можно сделать вывод, что в настоящее время к информационному оснащению вооруженных сил и управлению ими, со стороны любого государства предъявляются все более высокие требования, которые ужесточаются и расширяются с каждым годом. Создание ЕИП ВС это не просто «следование моде», это эволюция АСУ ВС. Единое информационное пространство увеличивает качество информации, ее полноту, точность и своевременность. Это происходит за счёт постоянного обмена информацией внутри ЕИП и многослойным наложением информации. А интеллектуальные системы по обработке и анализу данных поступающих в ЕИП позволяют человеку принимать качественные решения в сложной обстановке (лимит времени, анализ большого количества данных и т. д.) и тем самым увеличивают качество и скорость в управлении войсками.

Библиографический список

1. Копытко В. К., Шентура В. Н. Проблемы построения единого информационного пространства Вооруженных сил Российской Федерации и возможные пути их решения // АВН РФ, 2012.
2. Старостин А. В. Анализ текстов на ограниченном естественном языке с целью пополнения базы знаний // Труды VIII Международной научно-практической конференции молодых ученых, студентов и аспирантов "Анализ и прогнозирование систем управления". – СПб.: СЗТУ, 2007, ч. II. – С.141–144.
3. Сафронов А. Искусственный интеллект в поиске. Как Яндекс научился применять нейронные сети, чтобы искать по смыслу, а не по словам // Habrahabr. 2016.
4. URL: http://structure.mil.ru/structure/ministry_of_defence/details.htm?id=11206@egOrganization
5. URL: https://ru.wikipedia.org/wiki/Суперкомпьютер_Национального_центра_управления_обороной

УДК 681.5: 53.08

Д. В. Оганян, М. Г. Олейник

студенты Института военного образования

В. В. Котович

канд. техн. наук – научный руководитель

РАЗРАБОТКА ЛАБОРАТОРНОЙ УСТАНОВКИ ДЛЯ ПОВЕРКИ ГИРОПРИБОРОВ

Установка КПА-5А, предназначена для автоматизированной поверки гироскопических устройств. Данная установка разработана на базе существующей установки КПА-5, которая позволяет выставлять углы в трёх плоскостях с точностью 6 угл. мин в ручном режиме. Автоматизация данной установки позволит исключить ошибку вносимую оператором, а также увеличить точность установки до 4 угл. мин. за счет использования датчика на основе акселерометра. В данной установке используются следующие элементы:

- акселерометр ADXL203;
- аналого-цифровой преобразователь (АЦП) AD7705;
- гальваноразвязка REF5050;
- микроконтроллер (МК) Atmega32;
- bluetooth модуль HC-05;
- USB-UART преобразователь FT232RL;
- стабилизатор ADUM1401;

Конструктивно автоматизированная установка отличается от установки КПА-5 наличием платы с датчиком – акселерометром, платы управления, трёх шаговых двигателей (ШД), задающих положение рабочей поверхности установки в пространстве (в трёх степенях), и bluetooth модуля, расположенно на плате управления (рис.1).

Плата с датчиком служит для преобразования значений углов наклона подвижной части установки (тангажа и крена) в напряжение с помощью 2-степенного датчика-акселерометра, и отправки преобразованных значений на плату управления по интерфейсу SPI.

Плата управления служит для управления работой трёх ШД по команде от компьютера и для обработки данных от платы с датчиком. Она принимает значения напряжения от платы с датчиком, преобразовывает их в градусы и передаёт данные на компьютер по интерфейсу UART. По этому же интерфейсу плата управления получает команды на перемещение рабочей поверхности установки.

Плата с драйверами служит для усиления выходного сигнала микроконтроллера, до уровня напряжения необходимого для работы трёх ШД.

Bluetooth модуль позволяет дистанционно управлять установкой КПА-5А.

ШД управляются сигналами, поступающими с микроконтроллера. Угол, на который должна наклониться рабочая поверхность установки, определяется командой, поступающая на микроконтроллер с компьютера.

Блок-схема данной установки представлена на рис. 1. Электрическая схема платы с датчиком-акселерометром представлена на рис. 2.Схема (рис. 2) выполнена в редакторе Prioteus и использовалась для моделирования работы датчика. Обозначения, используемые на рис. 2 совпадают с перечисленными ранее элементами.

Схема платы управления с размещёнными на ней микроконтроллером, драйверами управления шаговыми двигателями, USB-UART преобразователем и соответствующим “обвесом” представлена на рис. 3. Схема условно разбита на блоки:

- блок А – разъем USB для подключения платы управления к компьютеру;
- блок Б – Bluetoothмодуль для дистанционного управления установкой КПА-5А;

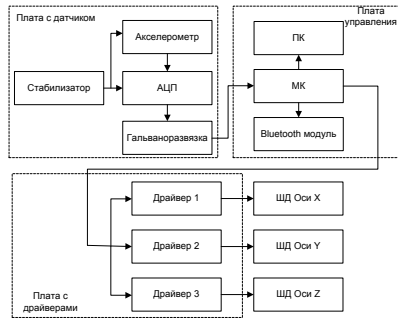


Рис. 1. Блок-схема установки КПА-5А

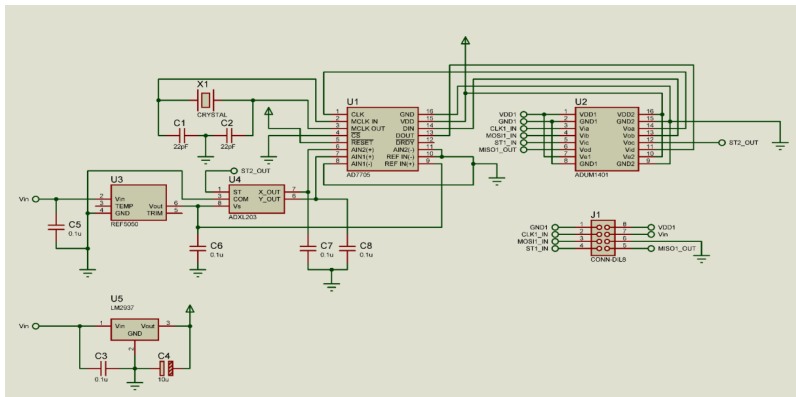


Рис. 2. Схема платы с датчиком

- блок В – преобразователь интерфейса USBв интерфейс UART, с помощью которого осуществляется связь микроконтроллера с компьютером;
- блок Г – микроконтроллер, позволяющий преобразовывать значения напряжения, поступающие с АЦП, преобразовывать их, отправлять на ПК илиBluetoothмодуль, а также управлять ШД с помощью команд, поступающих с ПК илиBluetoothмодуля;
- блок Д – разъем для подключения платы с датчиком;
- блок Е – плата с драйверами управления ШД;
- блок Ж – разъем для подключения ШД.

Алгоритм работы программы микроконтроллера (рис. 4) следующий:

- 1) каждые пол секунды микроконтроллер 50 раз опрашивает 16-ти разрядный АЦП по двум каналам, и получает преобразованные значения напряжения, соответствующие углам тангажа и крена. После заполнения буфера памяти результатами опросов микроконтроллер усредняет полученные значения;
- 2) усредненный результат измерения выходного напряжения акселерометра преобразуется в ASCII-код и передается по интерфейсу UARTна компьютер для дальнейшей обработки;

3) в момент времени, когда с компьютера поступают данные управления ШД, включается подпрограмма прерывания, которая реализует алгоритмы управления работой шаговых двигателей.

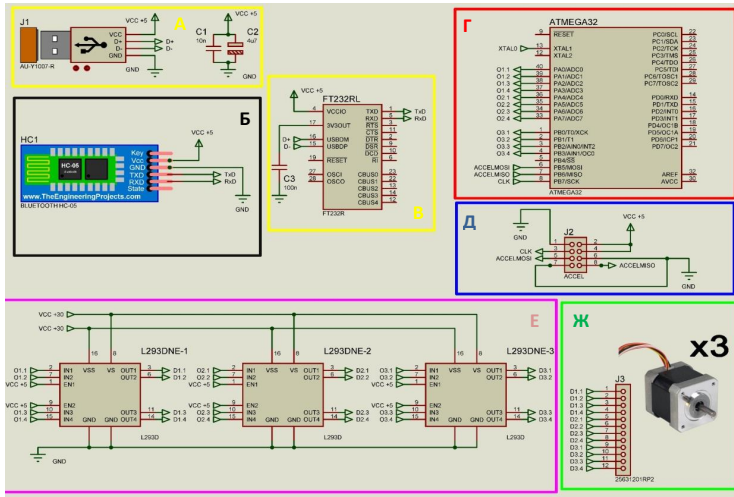


Рис. 3. Схема платы микроконтроллера, платы с драйверами и подключенными к ним ШД

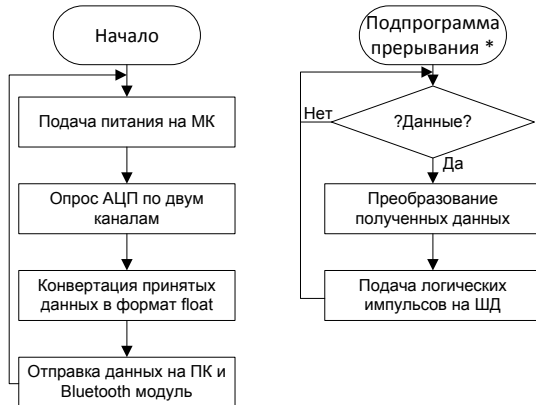


Рис. 4. Алгоритм работы программы микроконтроллера

Дополнительно, плата управления содержит bluetooth модуль, который подключается с помощью тумблера. Данный модуль отправляет выходные данные с МК на подключенное к нему устройство, например мобильный телефон или планшет. С помощью тумблера задаётся режим управления установкой, либо управление осуществляется от компьютера, либо по интерфейсу bluetooth от смартфона.

В результате проектирования был разработан опытный образец установки для автоматизированной проверки гироскопических устройств (рисунок 5). Установка позволяет автоматизировать процесс проверки и обработки её результатов. Дополнительный модуль Bluetooth позволяет производить проверку дистанционно.

плата с датчиком
шаговые двигатели
плата управления
плата с драйверами



Рис. 5. Изображение опытного образца установки

Библиографический список

1. Смолий А. А. Автоматизированная установка для проверки работоспособности гироскопических приборов летательных аппаратов / Шестьдесят четвертая студенческая научно-техническая конференция ГУАП: Сб. докл.: Ч. 2. Гуманитарные науки. СПб., 2011.
2. Using an Accelerometer for Inclination Sensing AN-1057 / Christopher J. Fisher // Application Note / Analog Devices, 2010.

УДК 681.5: 53.08

Д. В. Оганян

студент Института военного образования

Д. В. Бондаренко

канд. техн. наук – научный руководитель

ИССЛЕДОВАНИЕ МЕТОДА КОМПЕНСАЦИИ ОШИБКИ АКСЕЛЕРОМЕТРА В УСТАНОВКЕ КПА-5А

Целью данной работы является уменьшение погрешности установки для проверки акселерометров. Установка позволяет сравнивать показания поверяемого акселерометра и эталонного акселерометра. Одной из основных причин возникновения погрешности установки являются вибрации различной частоты, возникающие от внешних источников (движения человека, автомобиля и т.д.) или от поверяемых гиросприборов, которые конструктивно содержат быстровращающиеся гироскопы. Методы устранения воздействия внешних источников на подобные установки известны и широко применяются на практике. Задача устранения вибрации от поверяемых гиросприборов, размещённых на самой установке, требует окончательного решения. Для уменьшения ошибки выходного сигнала эталонного акселерометра, размещённого на установке, предлагается использовать второй акселерометр аналогичной конструкции, функционально связанный с первым, но размещённый на неподвижной части установки. Функциональная схема реализации предлагаемого метода компенсации ошибки представлена на рис. 1.

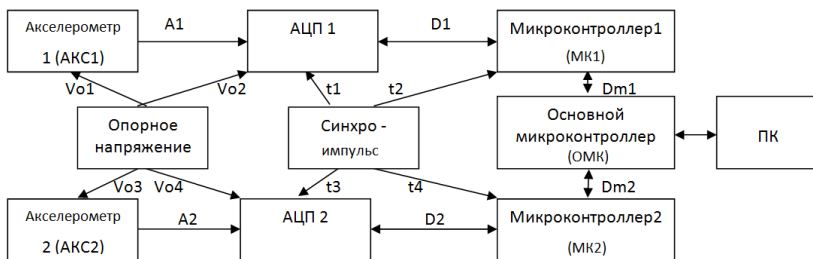


Рис. 1. Функциональная схема компенсационного метода измерения

Суть метода заключается в следующем:

Используются два идентичных акселерометра. Первый акселерометр (далее АКС1) устанавливается на неподвижную часть установки и предназначен для измерения сигнала шума (A1). Второй акселерометр (далее АКС2) закреплён на подвижной части установки (платформе с закреплённым на ней гиросприбором) и предназначен для измерения положения платформы в пространстве. АКС2 измеряет угол наклона платформы и, одновременно, шум (A2) от вибрации гиросприбора. Неподвижная и подвижная части установок условно изображены на рис. 2. Вибрации от поверяемых гиросприборов, размещённых на самой установке, то есть шум, воздействуют на всю установку в целом, следовательно, параметры шума, снимаемые с АКС1 и АКС2 идентичны. Чтобы избавиться от ошибки, который измеряется двумя акселерометрами, нужно вычитать из выходного сигнала, снимаемого с АКС2, выходной сигнал, снимаемый АКС1.

На рис. 2 подвижная часть установки (платформа) обозначена плоскостью $T_1 T_2 T_3$, а неподвижная – плоскостью $T_31 T_32 T_33$. Акселерометр АКС2 закреплён на плоскости $T_1 T_2 T_3$, а АКС1 закреплён на плоскости $T_31 T_32 T_33$.

Блок-схема (алгоритм) данного метода представлена на рис. 3.

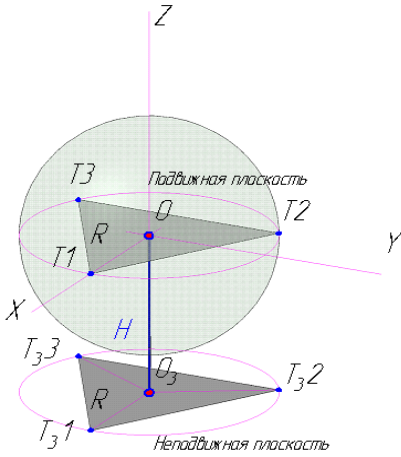


Рис. 2. Схематичное изображение установки

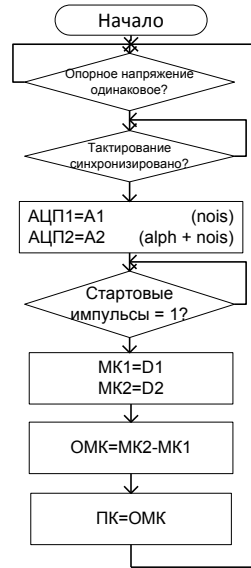


Рис. 3. Блок-схема алгоритма компенсации

Для обработки аналоговых выходных сигналов акселерометров необходимо преобразовать их в цифровой вид. Для этого используется АЦП1 (значение выходного сигнала на рис. 1 обозначено D1) и АЦП2 (значение выходного сигнала на рис. 1 обозначено D2).

Основная идея данного метода заключается в получении откликов от двух акселерометров одновременно. Для этого необходимо параллельное снятие показаний с двух АЦП. Так как микроконтроллеры не могут выполнять два действия одновременно, необходимо использовать три микроконтроллера. Два микроконтроллера (МК1 и МК2) синхронно считывают показания от акселерометров (D1 и D2), а третий микроконтроллер (ОМК) синхронизирует их работу, производит математические расчеты и отправляет полученные данные на компьютер (ПК).

В ходе выполнения данной работы были написаны программы для микроконтроллеров. Их работоспособность была промоделирована в среде Proteus. Схема установки в среде Proteus изображена на рис. 4.

Поскольку в библиотеке элементов Proteus отсутствуют акселерометры, вместо них при моделировании используются генераторы сигналов, выделенные красным и синим цветом на рис. 4. Генератор 1 (далее Г1), подсоединенный через батарею В2 к АЦП1 (на схеме обозначен U3), имитирует шумовой сигнал А1. Подключение батареи необходимо для поднятия среднего сигнала генератора до 2,5 В. Минус данного генератора заземлен, а выводы АМ и ФМ подсоединены к генераторам синусоидального сигнала А и F. Данные генераторы вносят амплитудную и фазовую модуляции выходному сигналу Г1. Для имитации сигнала А2 используется идентичный генератор (далее Г2), подключенный через батарею В3 к АЦП2 (на схеме – U6), выходной сигнал которого также модулируется с помощью генераторов синусоидальных сигналов. Разница генераторов Г1 и Г2 заключается в том, что минус Г2 подключен не к земле, а к генератору синусоидального сигнала В. Генератор В имитирует полезный сигнал, вызванный углом наклона подвижной части установки.

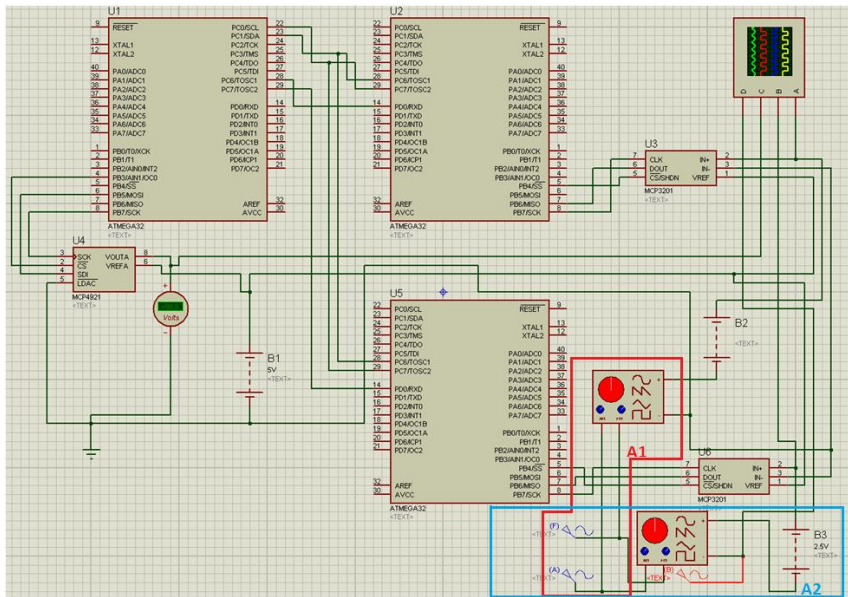


Рис. 4. Промоделированная схема установки

В данной схеме не используется синхронизирующий тактовый генератор, так как в среде моделирования Proteus тактирование цифровых устройств происходит одновременно, если обратное не указано в настройках.

Результирующее значение сигнала с микроконтроллер (ОМК) поступает на цифроаналоговый преобразователь ЦАП, обозначенный на схеме *U4*. Входные сигналы, поступающие на АЦП (*U3* и *U6* соответственно) и выходные сигналы (с ЦАП *U4* и с генератора *B*) поступают на осциллограф. В результате моделирования были получены графики, представленные на рис. 5.

На рис. 5 желтым цветом обозначен сигнал *A1*, синим цветом сигнал *A2*, красным цветом обозначен выходной сигнал с ЦАП, зеленым цветом – выходной сигнал с генератора *B*. Как видно из рис. 5 микроконтроллеры работают корректно. В результате вычитания входных сигналов АЦП, выходной сигнал ЦАП практически идентичен сигналу генератора *B*. Небольшая задержка и пилообразность выходного сигнала ЦАП вызвана временем преобразования данных, передачи данных и вычитанием данных, полученных с двух АЦП.

После успешного моделирования в среде Proteus был изготовлен опытный образец устройства. Для реализации синхронизирующего генератора использовался четвертый микроконтроллер, задача которого заключалась в подаче тактовых сигналов *t1*, *t2*, *t3* и *t4* на частоте 1 МГц. Для обеспечения одинакового опорного напряжения акселерометров и АЦП (*Uo1*, *Uo2*, *Uo3* и *Uo4*) использовался стабилизатор напряжения, который выравнивает входное напряжение до уровня 5В.

Затем были проведены испытания. В ходе выполнения испытаний обнаружила проблема, которая заключалась в том, что разность ошибки была не постоянной. В ПК можно было наблюдать за характерным инерционным явлением, которое заключалось в том, что при резком перемещении скрепленных акселерометров, выходной сигнал первого акселерометра превышал выходной сигнал второго акселерометра, а после остановки выходной сигнал второго акселерометра превышал выходной сигнал первого акселерометра. Возможно, причиной наблюдаемого явления являлась несинхронность измерения либо АЦП, либо акселерометров. Для проверки этого предположения была изменена схема

и подключен только один акселерометр к двум АЦП. В результате испытаний убедились, что разность показаний АЦП стала стабильной и колебалась в районе 0.0005 В. при перемещении акселерометра. На основании эксперимента был сделан вывод о том, что АЦП работают корректно, измерения напряжения происходят синхронно и с одинаковой точностью. Следовательно, погрешность измерения вызвана несинхронной реакцией акселерометров на входные воздействия.

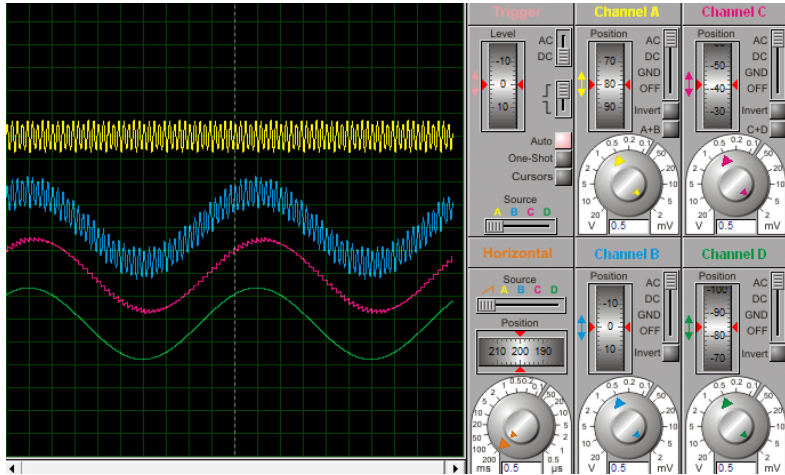


Рис. 5. Результаты моделирования

Из проведенных исследований можно сделать вывод: используемые МЭМС акселерометры не удовлетворяют требуемой точности, выходное напряжение у обоих акселерометров меняется неравномерно и не одновременно, таким образом, попытка компенсировать ошибку предлагаемым методом при данных параметрах акселерометров, невозможна.

Библиографический список

1. Статья. МЭМС, акселерометры, гироскопы и геомагнитные датчики. www.rlocman.ru/revive/article.html.

УДК 681.5: 53.08

Н. В. Чердинцев

студент Института военного образования, 442 уч. взвод.

В. В. Котович

канд. техн. наук – научный руководитель

ПРИМЕНЕНИЕ МЕТОДОВ ПЕРИФЕРИЙНОГО СКАНИРОВАНИЯ ПРИ ДИАГНОСТИРОВАНИИ СХЕМ

За последние десятилетия электронные устройства стали занимать всё более значимое место в жизни современного человека. Их конструкции значительно усложнились, появилось множество функциональных цифровых элементов, которые размещаются на схемах вместе с аналоговыми элементами. С появлением новых элементов, разрабатываются новые и совершенствуются известные методы и средства контроля и диагностирования электронных устройств и их компонентов на всех этапах их жизненного цикла. В связи с этим, ещё на этапе проектирования разработчику электронного устройства необходимо предусмотреть возможности контроля работоспособности и самоконтроля устройства на различных стадиях этапа проектирования, производства и эксплуатации данного устройства.

В процессе изготовления образца электронного устройства различными методами и средствами контролируются: корректность трассировки печатных плат, в том числе отсутствие коротких замыканий и обрывов; номенклатура и исправность элементной базы; правильность и надёжность монтажа; функционирование отдельных элементов, узлов, модулей и устройства в целом. Весь этот комплекс контролируемых мероприятий позволяет выявить дефекты компонентов, ошибки разработки электрической схемы, ошибки трассировки печатной платы, дефекты изготовления платы, дефекты монтажа, ошибки в программах, "прошитых" в микроконтроллерах, ПЛИС, DSP и т.д. Эффективное выявление дефектов существенно сокращает время создания нового устройства, но требует постоянного совершенствования методов контроля и диагностирования.

Простейшая методика контроля исправности и поиска дефектов создаваемого электронного устройства предусматривает следующие действия. После изготовления печатной платы осуществляется проверка её цепей питания на предмет наличия коротких замыканий, целостность функциональных линий между элементами (связи между ними), а также проверка посадочных мест для микросхем и других элементов платы. После монтажа элементов схемы, её проверка осуществляется повторно. В случае отсутствия дефектов после проведения повторной проверки, разработчик осуществляет проверку платы при подаче входного напряжения. При этом он измеряет электрические параметры схемы (напряжение, сопротивление, силу тока) и сравнивает полученные значения с параметрами, установленными разработчиком. В результате выполнения вышеперечисленных работ выявляется большая часть дефектов устройства и в первую очередь дефекты аналоговых элементов схемы. Однако этот подход к контролю не позволяет выявлять большую часть дефектов цифровых элементов схемы, таких как микроконтроллер, процессор или ПЛИС.

Для выявления скрытых дефектов перечисленных цифровых элементов схемы существует метод граничного сканирования или JTAG сканирование.

Название метода JTAG (Joint Test Action Group) произошло от названия группы разработчиков стандарта IEEE 1149. Официальное название стандарта Standard Test Access Port and Boundary Scan Architecture. Интерфейс предназначен для подключения сложных цифровых микросхем или устройств уровня печатной платы к стандартной аппаратуре тестирования и отладки.

Практически все цифровые микросхемы оснащаются этим интерфейсом для:

- выходного контроля микросхем при производстве;
- тестирования собранных печатных плат;
- прошивки микросхем с памятью;

– отладочных работ при проектировании аппаратуры и программного обеспечения.

Метод тестирования, реализованный в стандарте IEEE 1149, называется Boundary Scan (граничное сканирование). Стандарт предусматривает: наличие в микросхеме функциональных блоков, входы которых можно отсоединить от остальной схемы; возможность подачи на логическое ядро схемы заданных тестирующих комбинаций сигналов и оценки состояния выходов каждого функционального блока. Весь процесс управления сканированием осуществляется специальными командами, передаваемыми по интерфейсу JTAG, при этом никакого физического вмешательства в работу микросхемы не требуется. Для данного метода граничного сканирования разработан специальный язык управления данным процессом – Boundary Scan Description Language (BSDL).

Стандарт предусматривает возможность подключения большого количества устройств (микросхем) через один физический порт (разъем).

Порт тестирования (TAP – Test Access Port) представляет собой четыре или пять выделенных выводов микросхемы: TCK, TMS, TDI, TDO и TRST (опционально), пример JTAG- сканирования показан на рисунке 1.

Функциональное назначение этих линий:

- **TDI** (test data input – «вход тестовых данных») – вход последовательных данных периферийного сканирования. Команды и данные вводятся в микросхему с этого вывода по переднему фронту сигнала TCK;
- **TDO** (test data output – «выход тестовых данных») – выход последовательных данных. Команды и данные выводятся из микросхемы с этого вывода по заднему фронту сигнала TCK;
- **TCK** (test clock – «тестовое тактирование») – тактирует работу встроенного автомата управления периферийным сканированием;
- **TMS** (test mode select – «выбор режима тестирования») – обеспечивает переход схемы в/из режима тестирования и переключение между разными режимами тестирования;
- **TRST** – для инициализации порта тестирования, что необязательно, так как инициализация возможна путем подачи определённой последовательности сигналов на вход TMS.

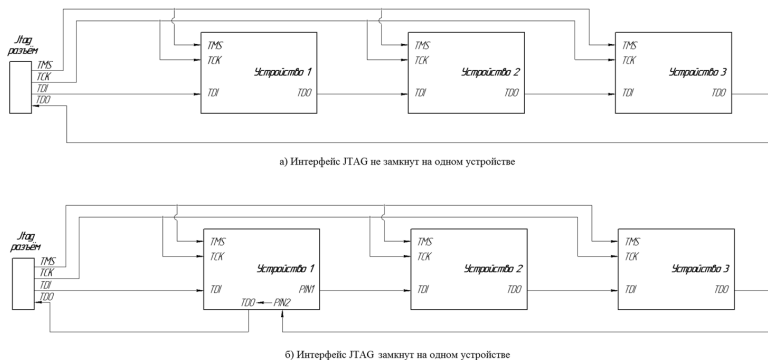


Рис. 1. Соединение компонентов схемы для проведения JTAG- сканирования

Работа средств обеспечения интерфейса JTAG подчиняется сигналам автомата управления, встроенного в микросхему. Состояния автомата определяются сигналами TDI и TMS порта тестирования. Определённое сочетание сигналов TMS и TCK обеспечивает ввод команды для автомата и её исполнение.

Если на плате установлено несколько устройств, поддерживающих JTAG, они могут быть объединены в общую цепочку (рис. 2).

Механизм граничного сканирования заключен в реализации сдвигового регистра (Boundary Scan), подключенном между выводами микросхемы и ядром, и мультиплексоре который подключает в нужный момент этот регистр. Каждому конкретному выводу соответствуют так называемые «ячейки» (рис. 3).

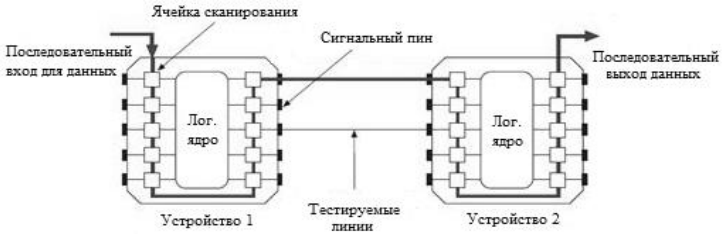


Рис. 2. Тестирование связывающих линий

В состав ячейки входит один триггер регистра граничного сканирования и мультиплексор для выбора данных. Ячейки могут быть разных типов в зависимости от выводов микросхемы и команд интерфейса. Производители не придерживаются строго стандарта, поэтому существует множество модификаций.

Как видно из рис. 3, каждая ячейка способна не только проводить тестирование путём сдвига входной последовательности от предыдущей ячейки к следующей, но и выводить данные на выходы микросхемы. Это позволяет разорвать тестирование ячеек на одном устройстве и перенести его на следующее устройство.

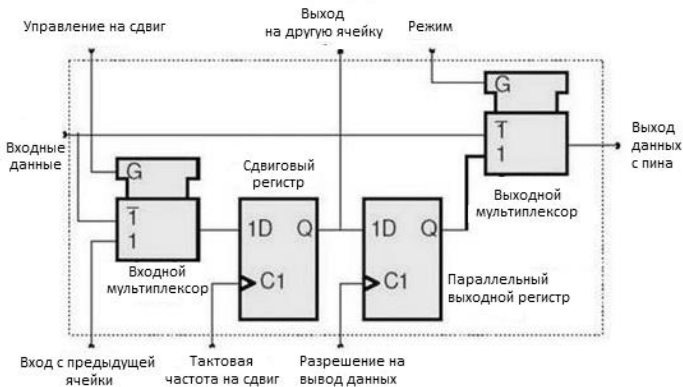


Рис. 3. Состав ячейки тестирования

Необходимость использования описанного метода граничного сканирования возникла в ходе разработки платы, предназначенной для транзита данных с преобразованием интерфейса. В процессе отладки была выявлена неисправность, связанная с некорректной передачей данных. Обычными методами диагностировать неисправность не удалось. В результате принято решение выполнить диагностику схемы методом граничного сканирования.

Проверка осуществлялась с помощью программы JTAG LIVE, в которой были проведены следующие тесты граничного сканирования:

Считывание IDустройства.

Этот тест обращается к устройству для считывания его ID. Это позволяет проверить целостность ядра микросхемы и сохранённых в нём параметров.

Тестирование ячеек выводов микросхемы.

Этот тест записывает инструкцию в управляющий регистр ядра микросхемы, после чего ядро замыкает все ячейки микросхемы следующим образом: TDI–ячейки выводов– TDO. Это позволяет убедиться в целостности ячеек выводов.

Тестирование ячеек выводов нескольких микросхем.

В отличие от предыдущего, данный тест позволяет не только проверить ячейки одной микросхемы последовательно, но и поставить точку разрыва, в которой обрывается связь ячейка – ячейка и данные выводятся на выбранный вывод микросхемы. Это позволяет соединить несколько микросхем в единый тест.

Выполнялось граничное сканирование платы, в состав которой входит ПЛИС фирмы-производителя Altera. С официальных сайтов были скачаны bsd1 – файлы, необходимые для проведения сканирования.

Для выполнения сканирования был создан проект, в который поочерёдно были добавлены bsd1 – файлы ПЛИС Cyclone III. После создания проекта был выбран путь тестирования (рис. 4).

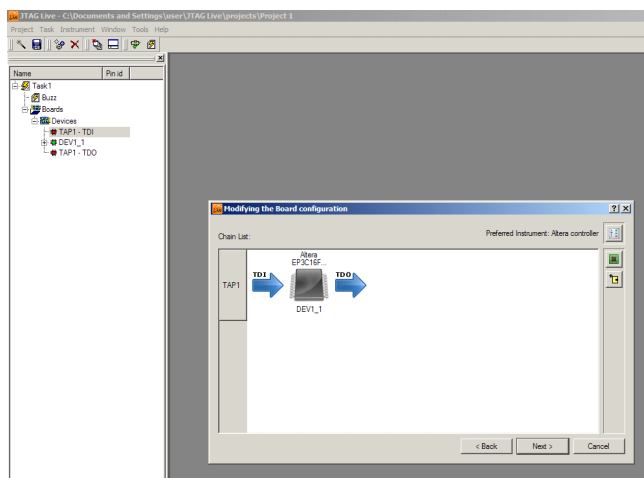


Рис. 4. Путь тестирования ПЛИС по интерфейсу JTAG

После определения пути сканирования были проведены тест ID (IDENT Test) и тест ячеек выводов микросхем (CAPTURE Test), завершённые успешно (рис. 5)

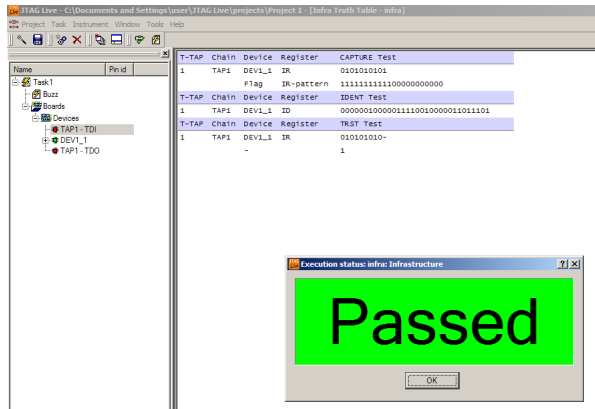


Рис. 5. Результаты теста ID и теста ячеек выводов

По результатам первого теста установлено, что при попытке загрузки любого BSDL-файла, не соответствующего ПЛИС Cyclone III, тестирование не выполнялось. С BSDL-файлом для Cyclone III тест был завершён успешно. По результатам второго теста установлено, что при конфигурации ядра ПЛИС на проведение теста ячеек выводов в соответствии с BSDL-файлом для Cyclone III, тест был завершён успешно.

Применение механизма граничного сканирования сократило время поиска неисправности минимум в два раза. В дальнейшем рекомендуется при проектировании электронных устройств на основе цифровых элементов разработчику учитывать возможность проведения всех тестов граничного сканирования.

Библиографический список

1. Записки тестового инженера, Остек, 121467, 2015.
2. www.studopedia.ru
3. www.pcbtech.ru
4. www.mirznanii.com

СОДЕРЖАНИЕ

Вычислительные системы и программирование

Акопян Б. К., Городецкая А. В. ИССЛЕДОВАНИЕ МЕТОДОВ ОБРАБОТКИ СИГНАЛА В ПРИЕМНИКЕ СИСТЕМЫ ЦИФРОВОЙ СВЯЗИ С ПЕРЕДАЧЕЙ ПО ОПТИЧЕСКОМУ КАНАЛУ.....	3
Акопян Б. К., Городецкая А. В. РАЗРАБОТКА МАКЕТНОГО ПРОТОТИПА СИСТЕМЫ ЦИФРОВОЙ СВЯЗИ С ПЕРЕДАЧЕЙ ПО ОПТИЧЕСКОМУ КАНАЛУ.....	7
Александров О. Д. РАЗРАБОТКА И ИНТЕГРАЦИЯ КОМПЬЮТЕРНОЙ МОДЕЛЯ ДЛЯ ПРОГНОЗИРОВАНИЯ ВЕРОЯТНОСТИ ИСХОДА ШАХМАТНОЙ ПАРТИИ.....	11
Афонькин Д. А. МОДЕЛИРОВАНИЕ ЕДИНОГО ИНФОРМАЦИОННОГО ПРОСТРАНСТВА ПРИ ПОМОЩИ СЕТЕЙ ПЕТРИ.....	14
Бабкова М. А., Фаттахова М. В. ИССЛЕДОВАНИЕ ЗАДАЧИ И МЕТОДОВ РЕШЕНИЯ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ ЗАПАСАМИ С УЧЁТОМ СЛУЧАЙНЫХ ВАРИАЦИЙ СПРОСА.....	17
Бабушкин М. В., Курова А. Ю. МЕРЫ СРАВНЕНИЯ ТЕКСТОВ.....	20
Балацкая О. Л. ИССЛЕДОВАНИЕ ВОПРОСОВ ПРИМЕНЕНИЯ МЕТОДОВ СТАТИСТИЧЕСКОЙ ОБРАБОТКИ ДЛЯ КОНТРОЛЯ ПОЛЕТОВ ВОЗДУШНЫХ ОБЪЕКТОВ.....	23
Батунов В. С. РАСШИРЕНИЕ ФУНКЦИОНАЛЬНОСТИ БИБЛИОТЕКИ ПОСТРОИТЕЛЯ ГРАФИКОВ В СРЕДЕ PUNTON.....	26
Белозуб Т. А. АДАПТАЦИЯ ИНТЕРФЕЙСА СРЕДЫ BLENDER 3D ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ РАБОТЫ С ЭЛЕМЕНТАМИ ГРАФИЧЕСКОЙ СРЕДЫ.....	29
Берёза В. В. ИНФОРМАЦИОННАЯ СИСТЕМА ПО СБОРУ ДАННЫХ О СОСТАВЕ И ТЕХНИЧЕСКОМ СОСТОЯНИИ ПЕРЕДАЮЩИХ РАДИОТЕХНИЧЕСКИХ ОБЪЕКТОВ.....	32
Боженко В. В. ОЦЕНКА ТРУДОЕМКОСТИ СОЗДАНИЯ OLAP-КУБОВ В MSSQLСЕРВЕРЕ ..	35
Бочарова А. М. ИСПОЛЬЗОВАНИЕ И ПОЛУЧЕНИЕ ГЕОПРОСТРАНСТВЕННЫХ ДАННЫХ В СИСТЕМАХ АЭРОНАВИГАЦИОННОЙ ИНФОРМАЦИИ.....	38
Бубликов С. В. ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИЙ BIG DATA ПРИ ВЫПОЛНЕНИИ РЕТРОСПЕКТИВНОГО АНАЛИЗА ВАЛЮТНОГО РЫНКА.....	41
Валяев Д. В., Герцев А.А., Максимова О.Н., Топчий А.С. РАЗРАБОТКА ПРОГРАММНОЙ СРЕДЫ ДЛЯ РЕАЛИЗАЦИИ ПРОЕКТОВ С ИСПОЛЬЗОВАНИЕМ ШЛЕМОВ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ HTC VIVE И OCULUS RIFT.....	44
Васильева В. С. МОДЕЛИРОВАНИЕ РАБОТЫ ДИСПЕТЧЕРА ЗАДАЧ НА ЯЗЫКЕ VBA.....	46
Величко А. Н. КРАТКИЙ ОБЗОР МЕТОДОВ, ПРИМЕНЯЕМЫХ ПРИ ПАРАЛИНГВИСТИЧЕСКОМ АНАЛИЗЕ РЕЧИ.....	51
Виноградов А. Ф., Клеченов А.С. МОДЕЛИРОВАНИЕ ЦИФРОВЫХ МЕТОДОВ ОБРАБОТКИ ЗВУКА В СРЕДЕ MATHCAD.....	54
Гашева Л. В. АЛГОРИТМИЧЕСКАЯ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ РЕГРЕССИОННОГО АНАЛИЗА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ.....	58
Гетманенко Г. В. ПРИМЕНЕНИЕ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧАХ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ.....	62
Горелик Д. В. ИСПОЛЬЗОВАНИЕ КОНТЕЙНЕРОВ ДЛЯ ПОСТРОЕНИЯ СИСТЕМЫ РАБОТЫ С УДАЛЕННЫМИ ВИРТУАЛЬНЫМИ МАШИНАМИ.....	66
Гурьянов Е. Ю., Аракчеев Г. В. СОЗДАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОДДЕРЖКИ ОПЕРАЦИОННЫХ ПРОЦЕССОВ МАЛОГО БИЗНЕСА.....	70

Егоров М. М. СРАВНИТЕЛЬНЫЙ АНАЛИЗ И ОЦЕНКА СРЕДЫ BLENDER 3D ВЕРСИЙ 2.49 И 2.78.....	74
Емельянов Г. И. МЕТОДЫ И ПОДХОДЫ К РАСПОЗНАВАНИЮ ДОРОЖНЫХ ЗНАКОВ.....	77
Еремеев И. Б. ОБНАРУЖЕНИЕ СИГНАЛА ПО АМПЛИТУДНОМУ СПЕКТРУ ПРИ ПОМОЩИ НЕЙРОННОЙ СЕТИ.....	80
Еремченко А. Ю. ПЕРЕХОД К МОДЕЛЯМ С ПЛАВАЮЩИМИ ВЕСОВЫМИ КОЭФФИЦИЕНТАМИ ДЛЯ АНАЛИЗА И ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ.....	84
Ерохин Р. А. ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА АДАПТИВНОГО ШАГАЮЩЕГО РОБОТА ДЛЯ ОБСЛЕДОВАНИЯ ГАЗОПРОВОДА.....	87
Журавлев А. С. РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ИНТЕРФЕЙСА БОРТОВОГО ИНДИКАТОРА.....	92
Иванова М. С., Сергеев А. В. ОЦЕНКА ПОКАЗАТЕЛЕЙ КАЧЕСТВА ОБСЛУЖИВАНИЯ ПОЛЬЗОВАТЕЛЕЙ ВИДЕОКОНТЕНТА.....	96
Игошин А. Н. ИСПОЛЬЗОВАНИЕ СТАТИСТИК ДЛЯ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ MSSQL СЕРВЕРА.....	100
Исаков М. С. ОБЗОР ИНСТРУМЕНТАЛЬНОЙ ГРАФИЧЕСКОЙ СРЕДЫ ISAGRAF ДЛЯ ПРОГРАММИРОВАНИЯ ПЛК.....	103
Ишкинин Р. В. ТЕСТОВОЕ ДИАГНОСТИРОВАНИЕ ЛИНЕЙНЫХ ДИНАМИЧЕСКИХ СИСТЕМ.....	107
Капранова Е. А., Харланов В. А. ПРОГРАМНОЕ ОБЕСПЕЧЕНИЕ УМНОГО АВТОМОБИЛЯ.....	114
Каратаев А. С. РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ МОБИЛЬНОЙ БАЗЫ ДАННЫХ С ПОМОЩЬЮ ТЕХНОЛОГИИ REALM.....	117
Катамадзе Г. В. ОРГАНИЗАЦИЯ УПРАВЛЕНИЯ ИНДЕКСАМИ В MSSQLSERVER.....	119
Кернер Е. К. ПРОГРАММНАЯ ОПТИМИЗАЦИЯ КОДА НА ОСНОВЕ МЕТРИК.....	122
Кунцевич Д. Ю. УПРАВЛЕНИЕ СТРУКТУРНЫМИ СВОЙСТВАМИ ПРОГРАММ.....	125
Легкоступов А. И. АВТОМАТИЗАЦИЯ СОСТАВЛЕНИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ ВУЗА В СООТВЕТСВИИ С ФГОС.....	128
Львова И. В. АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ ШАБЛОНОВ ПРОЕКТОВ НА ОСНОВЕ ФОРМАЛЬНЫХ ТРЕБОВАНИЙ.....	133
Маковчик И. В. РАЗРАБОТКА КОМПЬЮТЕРНОЙ МОДЕЛИ ДЛЯ РАСЧЕТА ВЕРОЯТНОСТИ ИСХОДА СПОРТИВНОГО СОБЫТИЯ.....	136
Мионов А. Ю. МОДЕЛЬ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ АДМИНИСТРАТИВНЫМ ДЕЛОПРОИЗВОДСТВОМ.....	139
Мухортов Н. О. ОБОСНОВАНИЕ ПРОГРАММНОГО ШАБЛОНА ИНФОРМАЦИОННО- КОММЕРЧЕСКИХ ВЕБ-СИСТЕМ МАЛОГО БИЗНЕСА.....	145
Никулин Д. В. ПРИМЕНЕНИЕ REST API В МОБИЛЬНОМ ПРИЛОЖЕНИИ ДЛЯ ИЗУЧЕНИЯ ИНОСТРАННЫХ ЯЗЫКОВ.....	151
Оганян Д. В. СРАВНЕНИЕ АЛГОРИТМОВ ДЕКОДИРОВАНИЯ КОДОВ-ПРОИЗВЕДЕНИЙ....	154
Павлова М. В. АНАЛИЗ СТРУКТУРЫ ДЕЯТЕЛЬНОСТИ ПО ОБСЛУЖИВАНИЮ ИНВАЛИДОВ.....	159
Панова И. С. МОДЕЛЬ ПРОГНОЗИРОВАНИЯ КОЛИЧЕСТВА ПУСКОВ РАКЕТ- НОСИТЕЛЕЙ ЛЕГКОГО КЛАССА.....	164

Парахневич К. А. ПРИНЦИП ОТКРЫТОЙ АРХИТЕКТУРЫ В ОСНОВЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ МИКРОКОНТРОЛЛЕРОВ.....	166
Печенин Д. В., Гусманова М. Р. ИССЛЕДОВАНИЕ АВТОМОРФНЫХ ПЕРЕДАТОЧНЫХ ФУНКЦИЙ.....	169
Печенин Д. В. ИССЛЕДОВАНИЕ СТРИП – МЕТОДА ПРЕОБРАЗОВАНИЯ ИЗОБРАЖЕНИЙ.....	172
Позняков А. И. АДАПТИВНАЯ СИСТЕМА РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В МОБИЛЬНОМ ПРИЛОЖЕНИИ.....	177
Розов А. А. ИМИТАЦИЯ АВИАЦИОННОГО ОБОРУДОВАНИЯ.....	180
Савин А. В. МОБИЛЬНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА СЕКЦИИ ЕДИНОБОРСТВ.....	182
Семенов А. А. СИСТЕМА АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ АВТОМОБИЛЕМ-НАРУШИТЕЛЕМ.....	184
Соловьев А. И., Токмакова Е. А. ЦИФРОВОЙ ЧЕТЫРЬХСЕКМЕНТНЫЙ ИНДИКАТОР ПОКАЗАНИЙ ДАТЧИКОВ «ВАРВАРА».....	186
Страбыкина Ю. А. ПРОГРАММНОЕ УПРАВЛЕНИЕ ПРОИЗВОДИТЕЛЬНОСТЬЮ И НАДЕЖНОСТЬЮ СИСТЕМ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА.....	189
Страбыкина Ю. А. СПЕЦИФИКА ОРГАНИЗАЦИИ ХРАНЕНИЯ ДАННЫХ В СИСТЕМАХ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА УРОВНЯ ПРЕДПРИЯТИЯ.....	192
Токаревский И. В. ПРОГРАММНО-АППАРАТНЫЕ РЕАЛИЗАЦИИ АЛГОРИТМОВ МАСКИРОВАНИЯ ДАННЫХ ВО ВСТРАИВАЕМЫХ МИКРОПРОЦЕССОРНЫХ СИСТЕМАХ..	197
Токаревский И. В. ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ОС LINUX В МИКРОПРОЦЕССОРНЫХ СИСТЕМАХ ВСТРАИВАЕМОГО КЛАССА.....	200
Толубаева А. С. ПОДОПТИМАЛЬНЫЙ АЛГОРИТМ ВЫРАВНИВАНИЯ СТРОКОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ.....	203
Топорков М. А. ИЗУЧЕНИЕ ОРНАМЕНТАЛЬНОЙ МАТРИЦЫ $N \times n \times n$ В ЗАДАЧАХ МАКСИМУМА ДЕТЕРМИНАНТА.....	207
Тропин Д. А., Уваров Н. А. СИСТЕМА ОХРАННОГО ВИДЕОНАБЛЮДЕНИЯ FUSIONCAM.....	211
Хянина И. Г. ИССЛЕДОВАНИЕ МОДЕЛЕЙ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ ЦЕПЯМИ ПОСТАВОК.....	215
Цветков И. А. БАЗА ЗНАНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ МОНИТОРИНГЕ СОСТОЯНИЙ КОСМИЧЕСКИХ СРЕДСТВ.....	218
Чуев Д. А. РАЗРАБОТКА МОДЕЛИ ПРОГНОЗИРОВАНИЯ КОЛИЧЕСТВА ПУСКОВ РАКЕТ-НОСИТЕЛЕЙ СРЕДНЕГО КЛАССА.....	221
Шальнев И. О. ИССЛЕДОВАНИЕ ПРОТОКОЛА ОБЪЕКТНОГО ВЗАИМОДЕЙСТВИЯ ВИРТУАЛЬНЫХ СИСТЕМ.....	225
Шведчиков В. А. ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА С ИСПОЛЬЗОВАНИЕМ СЕРВИСА GOOGLE ДЛЯ ЗАДАЧ CVRP.....	228
Шуин М. С. УПРАВЛЕНИЕ ЭСКАЛАЦИЕЙ БЛОКИРОВОК В MSSQL СЕРВЕРЕ.....	232
Информационные системы и защита информации	
Агеев М. П., Гуков С.Ю., Зайцева А. В., Миневцев Н. В. ИДЕНТИФИКАЦИЯ ЛИЧНОСТИ ЧЕЛОВЕКА В БУДУЩЕМ.....	235
Афанасьев А. А. ИССЛЕДОВАНИЕ МЕТОДОВ СЖАТИЯ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТЕЙ С МАСШТАБИРОВАНИЕМ БИТОВЫХ ПОТОКОВ.....	238

Бандуровский А. П. ПРОБЛЕМЫ БЕЗОПАСНОСТИ СКУД	243
Богданов Д. Д., Кучерук Т. В., Матвеева А. А., Семенов К. Д. СОВРЕМЕННЫЕ ТЕНДЕНЦИИ В ВЕБ-ДИЗАЙНЕ	246
Голос Н. А. АНАЛИЗ МЕТОДОВ ДВУХФАКТОРНОЙ АУТЕНТИФИКАЦИИ	250
Егорлицкий В. А. ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ПРОТОКОЛА КЭШИРОВАНИЯ PACKET-RE	254
Заусалин А. А. ПРОГРАММНЫЙ МОДУЛЬ ДЛЯ ИДЕНТИФИКАЦИИ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ РАСПОЗНАВАНИЯ ЛИЦ	257
Ильин В. И. АНАЛИЗ ТЕХНОЛОГИЙ ДЛЯ ПОСТРОЕНИЯ АРХИТЕКТУРЫ API ВЕБ-СЕРВИСОВ	260
Ким А. В., Бобина А. Ю. ИССЛЕДОВАНИЕ ВЛИЯНИЯ АРХИТЕКТУРЫ НЕЙРОННОЙ СЕТИ НА ТОЧНОСТЬ АППРОКСИМАЦИИ АНАЛИТИЧЕСКОЙ ФУНКЦИИ	263
Ковалева С. И. ЭТАПЫ СОЗДАНИЯ СИСТЕМЫ КЛАССА CRM	267
Коваленко В. Ю. SQL-ИНЪЕКЦИИ И МЕТОДЫ ЗАЩИТЫ ОТ НИХ	270
Кондратов А. В. ПРОБЛЕМЫ УТЕЧКИ ДАННЫХ ПРИ ПЕРЕДАЧЕ ПО ЭЛЕКТРОМАГНИТНЫМ КАНАЛАМ	272
Минаева Т. А. АНАЛИЗ RS-МЕТОДА СТЕГАНОАНАЛИЗА ДЛЯ ВМР-ИЗОБРАЖЕНИЙ	274
Миронова И. А. TOUCH ID/ДАКТИЛОСКОПИЧЕСКИЕ ДАТЧИКИ	278
Нестерук С. В. АНАЛИЗ МЕТОДОВ ЛИНГВИСТИЧЕСКОЙ СТЕГАНОГРАФИИ	281
Нестерук С. В., Шишко А. В. АСПЕКТЫ БЕЗОПАСНОСТИ БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЕЙ	284
Ровкова А. С. РАЗРАБОТКА МЕТОДА ИЗВЛЕЧЕНИЯ СТЕГАНОГРАФИЧЕСКИХ ДАННЫХ ИЗ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ С МЯГКИМ ВЫХОДОМ	286
Сермягина А. В. ПРИМЕНЕНИЕ МЕТОДОВ СИСТЕМНОГО АНАЛИЗА В ЗАДАЧЕ ИДЕНТИФИКАЦИИ	290
Соколова Ю. В., Дроздов А. В. СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ СИНТЕЗА ЛИНЕЙНЫХ СИСТЕМ НА ОСНОВЕ ИНВАРИАНТНОЙ ИМПУЛЬСНОЙ И ПЕРЕХОДНОЙ ХАРАКТЕРИСТИК	293
Татарникова А. А. СИСТЕМА УДАЛЕННОГО МОНИТОРИНГА ПАЦИЕНТОВ С ЗАБОЛЕВАНИЯМИ ОПОРНО-ДВИГАТЕЛЬНОГО АППАРАТА	296
Ткачев С. С. ОБЗОР УЯЗВИМОСТЕЙ SIM-КАРТ СТАНДАРТА GSM	300
Толопило Д. В. ТЕХНОЛОГИИ МОБИЛЬНОЙ СВЯЗИ 5G	304
Чекаль Е. В. ИЗМЕРЕНИЕ ОТНОСИТЕЛЬНОЙ ДИЭЛЕКТРИЧЕСКОЙ ПРОНИЦАЕМОСТИ И ЗАВИСИМОСТИ ТАНГЕНСА УГЛА ДИЭЛЕКТРИЧЕСКИХ ПОТЕРЬ ОТ ТЕМПЕРАТУРЫ ТВЕРДЫХ И ЖИДКИХ ДИЭЛЕКТРИКОВ	306
Шарапова Д. А. АНАЛИЗ АЛГОРИТМА СОПРОВОЖДЕНИЯ ОБЪЕКТОВ TRACKING-LEARNING-DETECTION	310
Военное образование	
Галыга Н. А., Черных Д. А. ИССЛЕДОВАНИЕ ЧИСЛЕННЫХ АЛГОРИТМОВ СИСТЕМЫ НАВЕДЕНИЯ БАЛЛИСТИЧЕСКОГО ЛЕТАТЕЛЬНОГО АППАРАТА ПО ПРОГНОЗУ ПАССИВНОГО УЧАСТКА ТРАЕКТОРИИ	314
Гусейнов П. И., Канцер А. А. ИНТЕРФЕЙС ЛАБОРАТОРНОЙ УСТАНОВКИ ДЛЯ ПОВЕРКИ ГИРОПРИБОРОВ	318

Колобеков Р. Р. ИСПОЛЬЗОВАНИЕ МЕТОДА КРИТИЧЕСКОГО ПУТИ В ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССАХ ЭКСПЛУАТАЦИИ СПЕЦИАЛЬНОГО ВООРУЖЕНИЯ..	323
Кульбик В. В.	327
РАЗРАБОТКА ИЭТР ДЛЯ ПОДГОТОВКИ АВИАЦИОННЫХ СПЕЦИАЛИСТОВ	327
Левицкий И. М. АНАЛИЗ ИНТЕЛЛЕКТУАЛИЗАЦИИ УПРАВЛЕНИЯ ОБОРОНОЙ РОССИЙСКОЙ ФЕДЕРАЦИИ. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ.....	330
Оганян Д. В., Олейник М. Г. РАЗРАБОТКА ЛАБОРАТОРНОЙ УСТАНОВКИ ДЛЯ ПОВЕРКИ ГИРОПРИБОРОВ.....	334
Оганян Д. В. ИССЛЕДОВАНИЕ МЕТОДА КОМПЕНСАЦИИ ОШИБКИ АКСЕЛЕРОМЕТРА В УСТАНОВКЕ КПА-5А.....	338
Чердинцев Н. В. ПРИМЕНЕНИЕ МЕТОДОВ ПЕРИФЕРИЙНОГО СКАНИРОВАНИЯ ПРИ ДИАГНОСТИРОВАНИИ СХЕМ.....	342

Научное издание

СЕМИДЕСЯТАЯ
МЕЖДУНАРОДНАЯ СТУДЕНЧЕСКАЯ
НАУЧНАЯ КОНФЕРЕНЦИЯ ГУАП

Часть 2

ТЕХНИЧЕСКИЕ НАУКИ

Сборник докладов
17–21 апреля 2017 г.

Ответственный за выпуск:
А. В. Бобович

Материалы публикуются в авторской редакции
Подготовка к печати *А. Н. Колешко*

Подписано к печати 19.09.17. Формат 60x84 1/16.
Усл. печ. л. 20,3. Уч.-изд. л. 21,2. Тираж 150 экз. Заказ № 376.

Редакционно-издательский центр ГУАП
190000, Санкт-Петербург, ул. Б. Морская, д. 67