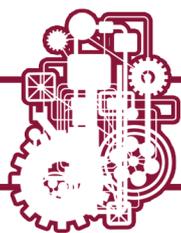


МОЛОДОЙ
УЧЁНЫЙ



VIII Международная научная конференция

ТЕХНИЧЕСКИЕ НАУКИ В РОССИИ И ЗА РУБЕЖОМ



Краснодар

Главный редактор: *И. Г. Ахметов*

Редакционная коллегия:

М. Н. Ахметова, Ю. В. Иванова, А. В. Каленский, В. А. Куташов, К. С. Лактионов, Н. М. Сараева, Т. К. Абдрасилов, О. А. Авдеюк, О. Т. Айдаров, Т. И. Алиева, В. В. Ахметова, В. С. Брезгин, О. Е. Данилов, А. В. Дёмин, К. В. Дядюн, К. В. Желнова, Т. П. Жуйкова, Х. О. Жураев, М. А. Игнатова, Р. М. Исаков, И. Б. Кайгородов, К. К. Калдыбай, А. А. Кенесов, В. В. Коварда, М. Г. Комогорцев, А. В. Котляров, А. Н. Кошербаева, В. М. Кузьмина, К. И. Курпаяниди, С. А. Кучерявенко, Е. В. Лескова, И. А. Макеева, Е. В. Матвиенко, Т. В. Матроскина, М. С. Матусевич, У. А. Мусаева, М. О. Насимов, Б. Ж. Паридинова, Г. Б. Прончев, А. М. Семахин, А. Э. Сенцов, Н. С. Сенюшкин, Е. И. Титова, И. Г. Ткаченко, М. С. Федорова С. Ф. Фозилов, А. С. Яхина, С. Н. Ячинова

Международный редакционный совет:

З. Г. Айрян (Армения), П. Л. Арошидзе (Грузия), З. В. Атаев (Россия), К. М. Ахмеденов (Казахстан), Б. Б. Бидова (Россия), В. В. Борисов (Украина), Г. Ц. Велковска (Болгария), Т. Гайич (Сербия), А. Данатаров (Туркменистан), А. М. Данилов (Россия), А. А. Демидов (Россия), З. Р. Досманбетова (Казахстан), А. М. Ешиев (Кыргызстан), С. П. Жолдошев (Кыргызстан), Н. С. Игисинов (Казахстан), Исаков Р. М. (Казахстан), К. Б. Кадыров (Узбекистан), И. Б. Кайгородов (Бразилия), А. В. Каленский (Россия), О. А. Козырева (Россия), Е. П. Колпак (Россия), А. Н. Кошербаева (Казахстан), К. И. Курпаяниди (Узбекистан), В. А. Куташов (Россия), Кыят Э. Л. (Турция), Лю Цзюань (Китай), Л. В. Малес (Украина), М. А. Нагервадзе (Грузия), Ф. А. Нурмамедли (Азербайджан), Н. Я. Прокопьев (Россия), М. А. Прокофьева (Казахстан), Р. Ю. Рахматуллин (Россия), М. Б. Ребезов (Россия), Ю. Г. Сорока (Украина), Г. Н. Узаков (Узбекистан), М. С. Федорова Н. Х. Хоналиев (Таджикистан), А. Хоссейни (Иран), А. К. Шарипов (Казахстан), З. Н. Шуклина (Россия)

Т38 **Технические науки в России и за рубежом** : VIII Междунар. науч. конф. (г. Краснодар, июнь 2019 г.) / [под ред. И. Г. Ахметова и др.]. — Краснодар : Новация, 2019. — iv, 34 с.
ISBN 978-5-907133-87-7

В сборнике представлены материалы VIII Международной научной конференции «Технические науки в России и за рубежом». Предназначен для научных работников, преподавателей, аспирантов и студентов медицинских специальностей, а также для широкого круга читателей.

УДК 62(01)
ББК 30

СОДЕРЖАНИЕ

ИНФОРМАТИКА И КИБЕРНЕТИКА

Гусаров А. В., Хачикова Н. А.Программные и аппаратные генераторы двоичных последовательностей
в информационных системах 1**Денисенко А. А.**

Параллельное программирование 3

Еременко Л. Е.Особенности проектирования базы данных для информационной системы
«Учет заказов по выполнению отделочных работ» 7**Пудовкин В. В.**

Инфографика как метод визуального анализа данных 12

Якуничева Е. Н., Горшкова А. А.

Инструменты проектирования виртуальных помощников 14

ЭЛЕКТРОНИКА, РАДИОТЕХНИКА И СВЯЗЬ

Айтулина А. М., Ивель В. П., Герасимова Ю. В., Калиаскаров Н. Б.

Структурные особенности построения прецизионной системы управления ветрогенератором 17

Калиаскаров Н. Б., Ивель В. П., Разинкин В. П., Герасимова Ю. В., Несипова С. С.Разработка устройства беспроводной системы для мониторинга состояния трещин и стыков зданий
и мостовых сооружений с использованием двухпроцессорных Wi-Fi-передатчиков 19

АВТОМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Гусаров А. В., Кошляков П. С.

Исследование методов настройки ПИД-регулятора для систем с малыми постоянными времени 22

Гусаров А. В., Медведев Е. Ю.

Реализация методов настройки ПИД-регулятора для малоинерционных систем 23

Добржинская Т. Ю., Рогова О. С., Алентьева Е. Р.

Исследование современных методов тестирования и диагностики компьютерных систем 27

МЕТАЛЛУРГИЯ

Маткаримов С. Т., Сафаров А. Х.

Задачи и методы обеднения шлаков на ОАО «Алмалыкский ГМК» 29

СТРОИТЕЛЬСТВО

Наседкин В. Э.

Современное состояние жилищно-коммунального комплекса в РФ и странах ЕС 31

ИНФОРМАТИКА И КИБЕРНЕТИКА

Программные и аппаратные генераторы двоичных последовательностей в информационных системах

Гусаров Александр Вячеславович, кандидат технических наук, доцент;

Хачикова Наталия Алексеевна, студент магистратуры

Рыбинский государственный авиационный технический университет имени П. А. Соловьёва (Ярославская обл.)

Случайные и псевдослучайные двоичные последовательности находят всё более широкое применение в системах сбора и обработки информации. Генерация таких последовательностей требует применения математических моделей как в процессе генерации, так и в процессе оценки качества случайных и псевдослучайных двоичных последовательностей. Темой данной работы является исследование процесса генерации последовательностей псевдослучайных двоичных последовательностей и проверка их качества.

Ключевые слова: псевдослучайные двоичные последовательности, случайные двоичные последовательности, качество ключей, симметричная криптосистема, энтропия.

В современных компьютерных системах сбора и обработки информации используются случайные и псевдослучайные двоичные последовательности различной длины. Для их генерации используются аппаратные и программные генераторы двоичных последовательностей. Важнейшим требованием, предъявляемым к таким последовательностям, является случайный характер следования нулевых и единичных битов внутри последовательности.

Случайные и псевдослучайные двоичные последовательности используются в качестве исходных данных для получения ключей, синхропосылок, стартовых векторов и других случайных параметров в системах защиты информации [1, 2], системах связи и навигации, при тестировании интегральных микросхем высокой и сверхвысокой степени интеграции, а также в других областях науки и техники.

Программные генераторы двоичных последовательностей выполняют генерацию двоичной последовательности по какому-либо алгоритму, поэтому случайность такой последовательности всегда под вопросом. Двоичные последовательности, генерируемые таким способом, следует обязательно проверять на случайность, т. к. фактически такие последовательности являются псевдослучайными из-за наличия фиксированного периода повторяемости и их вычисления по алгоритму. Вследствие этого программные генераторы двоичных последовательностей фактически являются генераторами псевдослучайных чисел (ГПСЧ). Несмотря на свои недостатки, ГПСЧ имеют одно неоспоримое достоинство. Они позволяют генерировать повторяющиеся псевдослучайные двоичные последовательности, если начальные условия для генерации не меняются. Это бывает необходимо, на-

пример, в процессе генерации гаммы шифра на приемном и передающем устройстве в системах засекреченного обмена информацией [1].

Основной проблемой при генерации последовательностей длиной более 64 бит является то, что большинство известных генераторов псевдослучайных последовательностей формируют результат длиной не более 64 бит, поэтому более длинные последовательности должны формироваться путем «склеивания» из более коротких. Еще одним вариантом формирования длинных последовательностей является использование хэш-функции, например функции по ГОСТ Р 34.11–94 или по ГОСТ Р 34.11–2012.

Для генерации действительно случайных двоичных последовательностей применяются аппаратные генераторы на основе датчиков случайных чисел. В качестве датчиков случайных чисел могут использоваться, например, полупроводниковые стабилитроны или специальные полупроводниковые генераторы шума. Однако не всегда есть возможность использовать такие датчики, так как они обычно интегрируются в специализированные печатные модули, устанавливаемые в слот материнской платы компьютера. В связи с этим возникает задача создания программного генератора, позволяющего реализовать двоичную последовательность с характеристиками, близкими к характеристикам случайных последовательностей. В его основу должна быть положена математическая модель, позволяющая реализовать программный генератор двоичных последовательностей, близких к случайным. Двоичные последовательности, генерируемые таким способом, следует обязательно проверять на случайность при помощи известных критериев.

Для математической модели должна быть обеспечена энтропия как минимум одного из параметров, иначе теряется «случайность» результата. В ситуации, когда математическая модель реализуется на персональном компьютере, источником энтропии могут быть либо пользователь, либо генераторы внешних событий, подключаемые к сети или внешним устройствам ПК. Проанализируем эти источники энтропии.

Достоинствами пользователя (человека) как источника энтропии являются:

- широкие возможности в выборе способов формирования энтропии;
- огромный разброс параметров человеческого организма, увеличивающий энтропию;
- нет необходимости в дополнительных устройствах, подключении к сети, зависимости от внешних событий. Человек использует стандартные устройства ввода информации (например, клавиатуру, мышь).

К недостаткам пользователя как источника энтропии можно отнести:

- возможность влияния на энтропию путем сознательного или подсознательного упорядочивания своих действий, что может привести к снижению энтропии;
- необходимость в сложных алгоритмах обработки входных данных, которые служат источником энтропии;
- необходимость присутствия человека во время генерации двоичных последовательностей, что ограничивает гибкость процесса генерации, «привязывает» его к определенным периодам времени;
- сравнительно низкую скорость генерации двоичных последовательностей вследствие ее связи со скоростью реакции человека на анализируемые события.

Достоинствами генераторов внешних событий как источника энтропии являются:

- возможность автоматизации процесса генерации двоичных последовательностей;
- высокая скорость генерации двоичных последовательностей.

К недостаткам генераторов внешних событий как источника энтропии можно отнести:

- высокую стоимость реализации генераторов внешних событий;
- генераторы внешних событий должны, наряду с основными функциями, обеспечивать энтропию, что снижает их производительность при выполнении основных функций;
- для подключения генератора задействуется один из внешних разъемов компьютера либо канал коммутатора, что накладывает ограничения на номенклатуру периферийных и сетевых устройств.

Литература:

1. Романец, Ю. В., Тимофеев П. А., Шаньгин В. Ф. Защита информации в компьютерных системах и сетях / Под ред. В. Ф. Шаньгина. — 2-е изд., перераб. и доп. — М.: Радио и связь, 2001. 376 с.: ил.

Таким образом, в качестве источника энтропии целесообразно использовать человека, т. к. чаще всего высокая скорость генерации двоичных последовательностей не требуется, и последовательности генерируются заранее. В то же время необходимость задействовать дополнительные аппаратные средства и каналы передачи данных является существенным недостатком при использовании генераторов внешних событий.

Для оценки случайности сгенерированных ключевых последовательностей используется методика, предложенная в [3] и состоящая из трех этапов.

На первом этапе необходимо установить, значимо или нет отличается относительная частота появления нулей (единиц) в данной выборке от теоретического значения вероятности появления единицы (нуля), равной 0,5. Другими словами, нужно проверить, является ли количество нулей и единиц в выборке примерно равным. Если нет, то значимо ли отклонение в ту или иную сторону. В качестве уровня значимости здесь и в дальнейшем достаточно выбрать $\alpha = 0,05$. Этот этап, конечно, не дает ответа на вопрос о случайности выборки. В случае значимого отклонения относительной частоты от 0,5, можно утверждать, что выборка не является случайной. Положительным для дальнейшего исследования является незначимое отклонение.

На втором этапе используется критерий пиков (критерий поворотных точек), на основе которого можно проверить, не слишком ли часто (или редко) в последовательности возникают изолированные нули или единицы. Этот этап позволяет исключить из рассмотрения выборки, имеющие вид «гребенки» или наоборот, содержащие только наборы групп нулей и единиц.

На третьем этапе проверки будет использоваться критерий серий. При использовании этого критерия отсекаются выборки, содержащие большое количество слишком длинных (или слишком коротких) наборов последовательных нулей и единиц.

Разработана программа, реализующая данный метод [4]. Проводятся исследования, направленные на оценку влияния параметров программного ГСПЧ и способа формирования 256-битовой ключевой последовательности на случайность получаемых ключей. Для этого разработан программный комплекс, состоящий из программ генерации ключей, программ оценки качества ключей и интерфейса пользователя, позволяющего задать исходные данные и сформировать результат на экране дисплея компьютера и в файле на диске [4]. Проводятся сравнительные исследования качества ключей, полученных программными генераторами псевдослучайных последовательностей, а также разрабатываются математические модели, позволяющие реализовать источники энтропии.

2. ГОСТ 28147–89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. Введ. 01.07.1990. Переиздание. М.: ИПК Издательство стандартов, 1996. — 26 с.
3. Гусаров, А. В., Гусарова Н. И. Об одном способе оценки параметров криптографических ключей // Информационные системы и технологии, 2013, № 1 с. 124–128.
4. Гусаров, А. В., Жолобов А. М. Исследование качества генерации псевдослучайных чисел в техническом вузе // «Технические науки: теория и практика»: Материалы IV Междунар. науч. конф. (г. Казань, ноябрь 2018 г.). — Казань: Молодой ученый, 2018. — с. 6–7.

Параллельное программирование

Денисенко Алексей Александрович, разработчик программного обеспечения
ИП Денисенко Алексей Александрович (г. Киев, Украина)

Рассмотрены ключевые понятия параллельных компьютерных систем и популярные технологии параллельного программирования.

Ключевые слова: *поток, связность, вектор, синхронизация, сопроцессор, мультипроцессор, мультикомпьютер, распределенные вычислительные системы, многозадачность, поток управления, взаимоблокировка, состояние гонок, критическая секция, функция ожидания, событие, мьютекс, семафор.*

Параллельное программирование по праву считается наиболее перспективным и востребованным направлением в области разработки программного обеспечения. Это связано с известным противоречием: возрастающая сложность промышленных, технических и научных задач, решаемых с помощью компьютерных систем, предъявляет к последним требования производительности, которые не могут быть выполнены, ввиду ограничений, накладываемых законами физики (предельная скорость света, теплоотдача, принцип неопределенности). Однако, такие задачи решать нужно. Поэтому единственным выходом из этой ситуации является использование идеи параллелизма вычислений (хотя и на эту идею накладывается ограничение в виде математического закона Амдала). Теоретически параллелизм можно вводить на любом уровне абстракции компьютерной системы. На каком бы уровне он не был введен, там будет существовать коммутация, а, следовательно, управление и программирование. Именно многоуровневость является основным фактором разнообразия средств и технологий параллельного программирования. Чтобы не запутаться в этом разнообразии укажем на два полезных общих понятия: классификация и связность компьютерных систем. В свете этих понятий рассмотренные ниже технологии параллельного программирования выстроятся в четкую логическую структуру.

Классификация компьютерных систем. Разнообразие способов организации параллельных вычислительных систем побудило научный мир к решению задачи классификации. Самая ранняя и наиболее известная классификация М. Флинна (M. Flynn) предложена в 1966 году. Ключевое понятие здесь — поток. На основе числа потоков команд и потоков данных Флинн делит все системы на четыре класса архитектур:

— SISD (Single Instruction stream/Single Data stream) — одиночный поток команд и одиночный поток данных. К этому классу относятся машины классического, фон-неймановского типа.

— SIMD (Single Instruction stream/Multiple Data stream) — одиночный поток команд и множественный поток данных. Здесь один поток команд манипулирует множественными (векторными) данными. В качестве известного примера можно привести технологию SSE компании Intel.

— MISD (Multiple Instruction stream/Single Data stream) — множественный поток команд и одиночный поток данных. К этому классу относят отказоустойчивые системы.

— MIMD (Multiple Instruction stream/Single Data stream) — множественный поток команд и множественный поток данных. К этому классу относятся мультипроцессорные системы.

На рисунке 1 показано схематическое представление рассмотренной классификации, взятое из работы [1]. Здесь приняты следующие условные обозначения: УУ — устройство управления, ПД — память данных, ПР — процессорные элементы. Данная классификация является самой распространенной. С её помощью можно определить базовые принципы работы компьютерной системы. Однако, она не является единственной. В работе [1] рассмотрено множество других классификаций.

Итак, любая архитектура компьютерной системы может быть отнесена к одному из четырех вышеперечисленных классов. Любые вычислительные элементы этих систем, как и сами системы в целом, могут быть интегрированы. Исследование вопроса степени интеграции приводит к рассмотрению понятия связности.

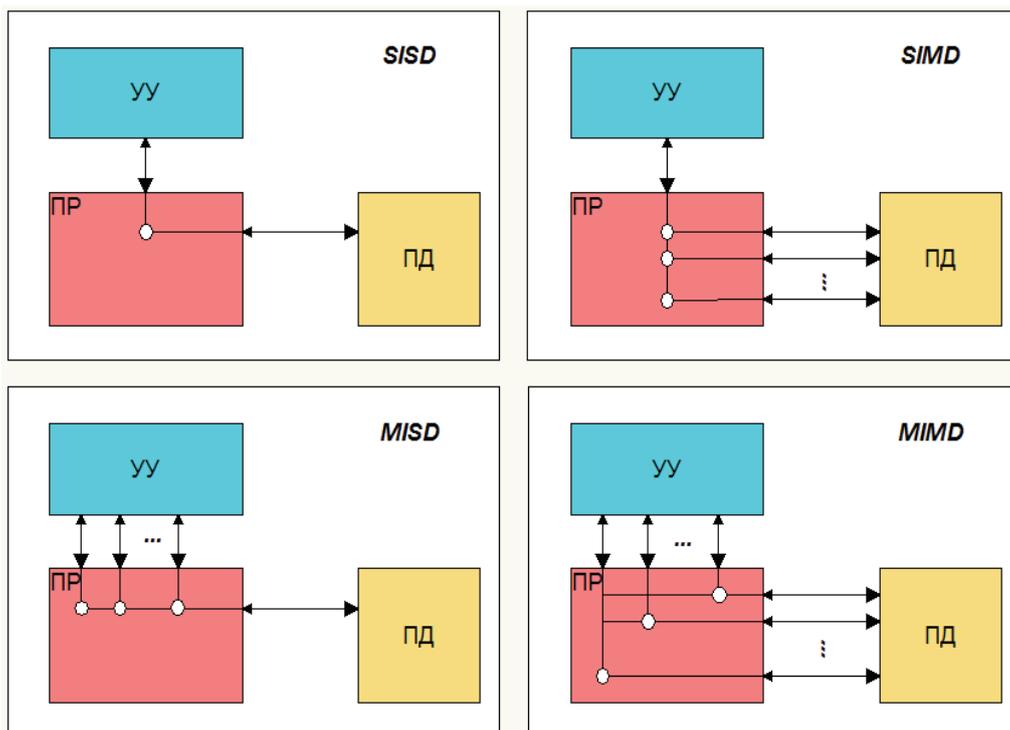


Рис. 1. Классификация компьютерных архитектур М. Флинна

Связность параллельных компьютерных элементов и систем. По характеру интеграции компьютерные системы делятся на два больших класса: сильно связанные (tightly coupled) и слабо связанные (loosely coupled). Эти два класса плавно переходят в друг друга. Предельный случай сильно связанных систем — это внутрипроцессорный параллелизм. Предельный случай слабо связанных — распределенные вычислительные системы (grid computing). Рисунок 2 иллюстрирует связность систем на шкале связности [4]. Здесь под цифрами соответственно обозначены:

1. *Внутрипроцессорный параллелизм.* К этому классу относятся такие технологии: параллелизм на уровне команд процессора (суперскалярность), процессоры с сверхдлинным командным словом (VLIW, Very Long Instruction Word), внутрипроцессорная многопоточность (on-chip multithreading). Как правило программист не имеет доступ на этот уровень. Система работает автоматически.

2. *Сопроцессоры.* Представляют собой дополнительные процессоры, предназначенные для обработки специализированных данных: ввода-вывода, чисел с плавающей точкой, графических, сетевых, криптографических и др. Как и всякий процессор сопроцессор является программируемым. Пример реализации — технологии MMX, SSE.

3. *Мультипроцессоры.* К этому классу относятся системы, в которых процессоры используют общую физическую память. Самыми популярными технологиями параллельного программирования здесь являются: поточное программирование (потоки ОС), OpenMP (Open Multi-Processing), Intel TBB (Threading Building Blocks), Boost.

4. *Мультикомпьютеры.* Здесь процессоры используют собственную локальную память. Ввиду этого взаимодействие между ними организовано посредством коммуникационной сети. К мультикомпьютерам относятся такие известные системы как IBM BlueGene/L, Red Storm, кластер Google. Программирование систем этого класса осуществляется посредством таких технологий как MPI (Message-Passing Interface), PVM (Parallel Virtual Machine), Linda, Express, OpenCL (Open Computing Language), mpC, Fortran-DVM, C-DVM, MapReduce, Hadoop.

5. *Распределенные вычислительные системы.* Полноценных технологий программирования этого уровня пока не существует. Поэтому разработчики пользуются уже существующими технологиями уровня мультикомпьютинга, а для коммуникации применяют сетевые технологии. В настоящее время организацией Global Grid Forum создан стандарт OGSA (Open Grid Services Architecture), который определяет предоставляемые системами услуги и протоколы доступа к ним.

Рассмотрим перечисленные выше технологии программирования в порядке связности систем.

Программирование сопроцессора — SSE. SSE — это SIMD расширение процессоров Intel. Большим плюсом при использовании инструкций SSE является то, что не нужно при этом решать главную задачу параллельного программирования — задачу синхронизации. Программирование, при этом, ничем не отличается от последовательного. Просто вместо скалярных операндов здесь используются векторные.

Поддержка SSE реализована во многих современных компиляторах. На рисунке 3 показаны примеры исполь-

of execution). В виртуальном адресном пространстве одного процесса может действовать несколько потоков. Если эти потоки являются взаимодействующими, то для

организации их работы нужно решать задачу синхронизации [3]. Основные функции для работы с потоками показаны на рисунке 4.

```
function CreateThread(
    lpThreadAttributes: Pointer;           // атрибуты защиты
    dwStackSize: DWORD;                   // размер стека потока в байтах
    lpStartAddress: TFNThreadStartRoutine; // адрес функции
    lpParameter: Pointer;                  // адрес параметра
    dwCreationFlags: DWORD;                // флаги создания потока
    var lpThreadId: DWORD                  // идентификатор потока
): THandle;

procedure ExitThread(
    dwExitCode: DWORD //код завершения потока
); stdcall;

function SuspendThread(
    hThread: THandle //дескриптор потока
): DWORD; stdcall;

function ResumeThread(
    hThread: THandle //дескриптор потока
): DWORD; stdcall;

procedure Sleep(
    dwMilliseconds: DWORD //миллисекунды
); stdcall;
```

Рис. 4. Основные функции для работы с потоками

Всего в библиотеке WinAPI определено 49 функций для работы с потоками.

Итак, в чем состоит задача синхронизации работы взаимодействующих потоков? Она состоит в задаче корректного доступа потоков к общему, критическому ресурсу. Если она надлежащим образом не решена, то потоки могут зависнуть. Возникнет их взаимоблокировка (тупиковая ситуация), либо состояние гонок. Для решения этой задачи в операционных системах реализованы следующие примитивы синхронизации:

- Критические секции (*critical section*)
- Функции ожидания (*wait functions*)
- События (*event*)
- Мьютекс (*mutex*)

— Семафор (*semaphore*)

Примитивы синхронизации, описанные разными стандартами и реализованные в разных операционных системах, имеют схожий интерфейс.

Многопоточное программирование мультипроцессоров. **OpenMP**. Стандарт OpenMP реализован в языках программирования C/C++ и Fortran и работает как в Unix, так и в Windows системах. Технология представляет собой набор особых директив компилятора, процедур и переменных окружения. Процесс программирования сводится (в упрощенной форме) к вставке директив в места где код может быть распараллелен [1]. Сказанное иллюстрируется рисунком 5.

```
int main()
{
    int i;
    getchar();
    getchar();
    getchar();
    #pragma omp parallel
    {
        #pragma omp for
        for (i = 0; i < 100000; i++)
            printf("%d ", i);
    }
    getchar();
    return 0;
}
```

← последовательная секция

← параллельная секция

← последовательная секция

Рис. 5. Применение технологии OpenMP

На этом рисунке код в параллельной секции создаст потоки ОС (количеством равным количеству ядер процес-

сора), которые совместно выполняют все итерации этого цикла.

Заключение. Мы рассмотрели основные популярные технологии параллельного программирования, указали на ключевые понятия, которые позволяют комбинировать эти технологии и находить общие архитектурные решения. Однако, тема статьи настолько обширна, что достойное раскрытие её возможно в рамках толстых томов

монографий. Так мы, не коснулись важного вопроса в этой области — вопроса разработки параллельных алгоритмов. Так же мы опустили вопрос разработки и дискретизации математической модели. Тем не менее, рассмотренные технологии и понятия достаточны для введения в этот раздел программирования.

Литература:

1. Воеводин, В. В., Воеводин Вл. В. Параллельные вычисления. — СПб.: БХВ-Петербург, 2002. — 600 с.
2. Качко, Е. Г. Параллельное программирование: Учебное пособие. — Харьков: Форт, 2011. — 528 с.
3. Лупин, С. А., Посыпкин М. А. Технологии параллельного программирования. — М.: ФОРУМ, 2011. — 208 с.
4. Таненбаум, Э. Архитектура компьютера. — СПб.: Питер, 2009. — 844 с.

Особенности проектирования базы данных для информационной системы «Учет заказов по выполнению отделочных работ»

Еременко Лариса Евгеньевна, преподаватель
Рязанский технологический колледж

В настоящее время работу любой организации трудно представить без базы данных и автоматизированной информационной системы, ведь с помощью автоматизации процессов организации и различных услуг можно добиться получения быстрых и точных результатов путем обработки информации. Организация, предоставляющая услуги по выполнению отделочных работ — не исключение. Работникам данной организации приходится выполнять большое количество операций: поддержка прейскуранта цен на услуги в актуальном состоянии, оформление заказов, расчет стоимости заказа, прием оплаты за услуги, отслеживание выполнения и оплаты заказов и т. д. Поэтому имеет смысл автоматизировать если не все, то хотя бы большую часть этих операций. Поэтому разработка базы данных для учета оплаты услуг в организации по выполнению отделочных работ является достаточно актуальной и нужной задачей. Она сможет поднять на новый уровень процессы управления за счет использования информационной технологии для предоставления услуг отделки, расчетов её стоимости, а также совершенствования системы работы с клиентами и ведения базы данных клиентов.

1. Анализ предметной области

Некоторая организация предлагает широкий спектр отделочных работ различных категорий: устройство полов, отделка потолков, стен, плиточные работы и т. д. Выполнение работ осуществляется в пределах города. Отделочные работы выполняются по предварительным заказам клиентов.

Процесс приема заказов выполняется следующим образом.

В организацию обращается клиент, желающий заказать услуги по отделочным работам. При этом он может

ознакомиться с прайс-листом всех предоставляемых работ или воспользоваться помощью менеджера по поиску сведений о конкретной работе.

Если клиент собирается заказать какие-либо работы, то сведения о нем заносятся в базу данных. Затем менеджер приступает к оформлению заказа. В заказе указываются данные клиента, дата составления заказа, перечень отделочных работ, количество по каждой работе, рассчитывается стоимость каждой работы, общая стоимость работ. В стоимость работ не включена стоимость расходных материалов, которые клиент обязан предоставить сам. Менеджер проставляет примерный срок исполнения заказа. В заказе так же проставляется номер свободной ремонтной бригады (в соответствии с профилем работ), которая будет работать с заказом. По согласованию с клиентом указывается дата начала выполнения работ, которая может быть изменена (опять же по согласованию с клиентом).

При оформлении заказа клиент вносит предоплату, которая должна составлять не менее 40% от стоимости заказа. Затем в двух экземплярах печатается бланк заказа: один — передается клиенту, второй — остается в организации и в дальнейшем будет передан ремонтной бригаде. После выполнения работ менеджер проставляет в заказ фактическую дату завершения ремонта. После приема ремонта клиентом, он оплачивает остальную стоимость оказанных услуг, а менеджер проставляет в заказ дату полной оплаты.

Кроме анализа выполнения заказов менеджер может выбирать данные об оплате услуг: денежных средствах, полученным по заказам от клиентов за некоторый временной интервал; отслеживать оплаченные и неоплаченные заказы. Есть возможность определить данные обо всех заказах конкретного клиента.

2. Определение сущностей и связей между ними

Определить сущности достаточно легко на основе анализа предметной области. Но не стоит забывать и о будущем интерфейсе разрабатываемой системы. Очевидно, основным действием при заполнении базы данных будет

формирование заказа на выполнение отделочных работ. Поэтому, следует представить, как будет выглядеть соответствующее окно. Вид будущей формы оформления заказа представлен на рис. 1.

Рис. 1. Форма оформления заказа

Компоненты формы определены на основе функций, которые должны выполняться информационной системой.

Анализируя предметную область и представляя внешний вид основного диалогового окна ввода данных, на первом этапе проектирования можем выделить следующие сущности.

— работы;

— заказ;
— бригада;
— мастер;
— клиент.

Представим информационную модель задачи в виде диаграммы «сущность — связь» (рис. 2).



Рис. 2. Диаграмма «сущность — связь»

Таблица 1

Описание связей между объектами (сущностями)

Связь	Тип связи	Пояснение
Работы — Заказ	«многие-ко-многим»	одна работа может присутствовать в нескольких заказах, а в одном заказе может быть указано несколько работ
Клиент — Заказ	«один-ко-многим»	один клиент может участвовать в нескольких заказах, но один заказ оформляется только на одного клиента
Бригада — Заказ	«один-ко-многим»	одна бригада может работать по нескольким заказам, но один заказ обрабатывается только одной бригадой
Бригада — Мастера	«один-ко-многим»	к одной бригаде относится несколько мастеров, но один мастер относится только к одной бригаде

Описание связей между объектами представлено в таблице 1 [2, с. 12].

3. Построение реляционной модели

Поскольку в концептуальной модели отсутствует связь «один-к-одному», то в реляционной модели данных будет минимум пять таблиц.

Связь «многие-ко-многим» будет преобразована за счет введения дополнительной сущности СОСТАВ, ко-

торая должна содержать первичные ключи сущностей ЗАКАЗ и РАБОТЫ (они будут внешними ключами), а также атрибуты, имеющие отношение к отдельной заказанной работе (например, количество, стоимость выполнения одной работы в указанном количестве) — атрибуты связи.

Таким образом, получим эквивалентный вариант диаграммы сущность-связь (см. рис. 3).



Рис. 3. Диаграмма «сущность – связь» после удаления связи «многие-ко-многим»

4. Задание первичных ключей, определение атрибутов сущностей

Для каждой сущности определим атрибуты, которые будут храниться в базе данных (см. табл. 2).

Таблица 2
Атрибуты и первичные ключи сущностей

Сущность	Первичный ключ	Атрибуты
Заказ	код_заказа	код_заказа код_клиента код_бригады дата_заказа срок_исполнения сумма дата_начала дата_испол_ориент дата_испол_факт адрес_исполнения предоплата доплата_по_исполн дата_полной_оплаты

Состав		код_работы код_заказа кол_во цена_работы
Мастер	код_мастера, код_профиля	код_мастера код_бригады ФИО_мастера код_профиля профиль
Клиент	код_клиента	код_клиента ФИО_клиента тел_клиента адрес_клиента
Сущность	Первичный ключ	Атрибуты
Работа	код_работы, код_категории, код_ед_измерения	код_работы код_ед_измерения ед_измерения код_категории категория наименование цена_ед
Бригада	код_бригады	код_бригады занятость

5. Приведение модели базы данных к первой нормальной форме

Отношение находится в первой нормальной форме [2, с. 22], если для него выполняются условия:

- отсутствуют повторяющиеся атрибуты;
- каждый атрибут неделим.

Согласно требованиям первой нормальной формы, необходимо преобразовать атрибуты «ФИО_клиента», «адрес_клиента» в сущности «Клиент», «адрес_исполнения» в сущности «Заказ», «ФИО_мастера» в сущности «Мастер» так, чтобы получить неделимые атрибуты. В дальнейшем это позволит реализовать такие функции, например, как поиск информации о клиентах или мастерах по фамилии, поиск заказов по адресу и

6. Приведение модели базы данных ко второй нормальной форме

Отношение находится во второй нормальной форме [2, с. 23], если оно удовлетворяет следующим требованиям:

- выполняются условия первой нормальной формы;
- все поля записи функционально полно зависят от первичного ключа.

В сущности «Работы» атрибуты «ед_измерения», «категория» зависят только от частей составного первичного ключа «код_ед_измерения» и «код_категории» соответственно. Поэтому отношение «Работа» не находится во второй нормальной форме и его следует преобразовать, выделив из него отдельные сущности «Категория» и «Ед_измерения». Атрибуты «код_ед_измерения» и «код_категории» в сущности «Работы» станут внешними ключами.

В сущности «Мастер» атрибут «профиль» зависит только от частей составного первичного ключа: «код_про-

филя». Поэтому отношение «Мастер» не находится во второй нормальной форме и его следует преобразовать, выделив из него отдельную сущность «Профиль». Атрибут «код_профиля» в сущности «Мастер» станет внешним ключом.

7. Приведение модели базы данных к третьей нормальной форме

Отношение находится в третьей нормальной форме [2, с. 23], если оно удовлетворяет следующим требованиям:

- должны выполняться условия второй нормальной формы;
- внутри каждой сущности должны отсутствовать транзитивные связи.

В сущности «Заказ» атрибут «дата_исполн_ориент» зависит от атрибутов «дата_начала» и «срок_исполнения» этой же сущности. Поэтому удалим данный атрибут из сущности, создав на форме поле, вычисляемое по формуле:

$$\text{Дата_исполн_ориент} = \text{дата_начала} + \text{срок_исполнения}, \quad (1)$$

где «дата_начала», «срок_исполнения» — поля сущности «Заказ».

В сущности «Заказ» атрибут «доплата_по_исполн» зависит от атрибутов «сумма» и «предоплата» этой же сущности. Поэтому удалим вычисляемый атрибут из сущности, создав на форме поле, вычисляемое по формуле:

$$\text{Доплата_по_исполн} = \text{сумма} - \text{предоплата}, \quad (2)$$

где «сумма» — поле сущности «Заказ».

В сущности «Состав» атрибут «сумма_работы» зависит от атрибута «кол_во» этой же сущности. Поэтому

удалим данный атрибут из сущности, создав на форме поле, вычисляемое по формуле:

$$\text{Сумма_работы} = \text{цена_ед} * \text{кол_во}, \quad (3)$$

где «цена_ед» — поле сущности «Работа»;

«кол_во» — поле сущности «Состав».

Приведем диаграмму взаимосвязей между атрибутами сущностей в третьей нормальной форме (рис. 4).

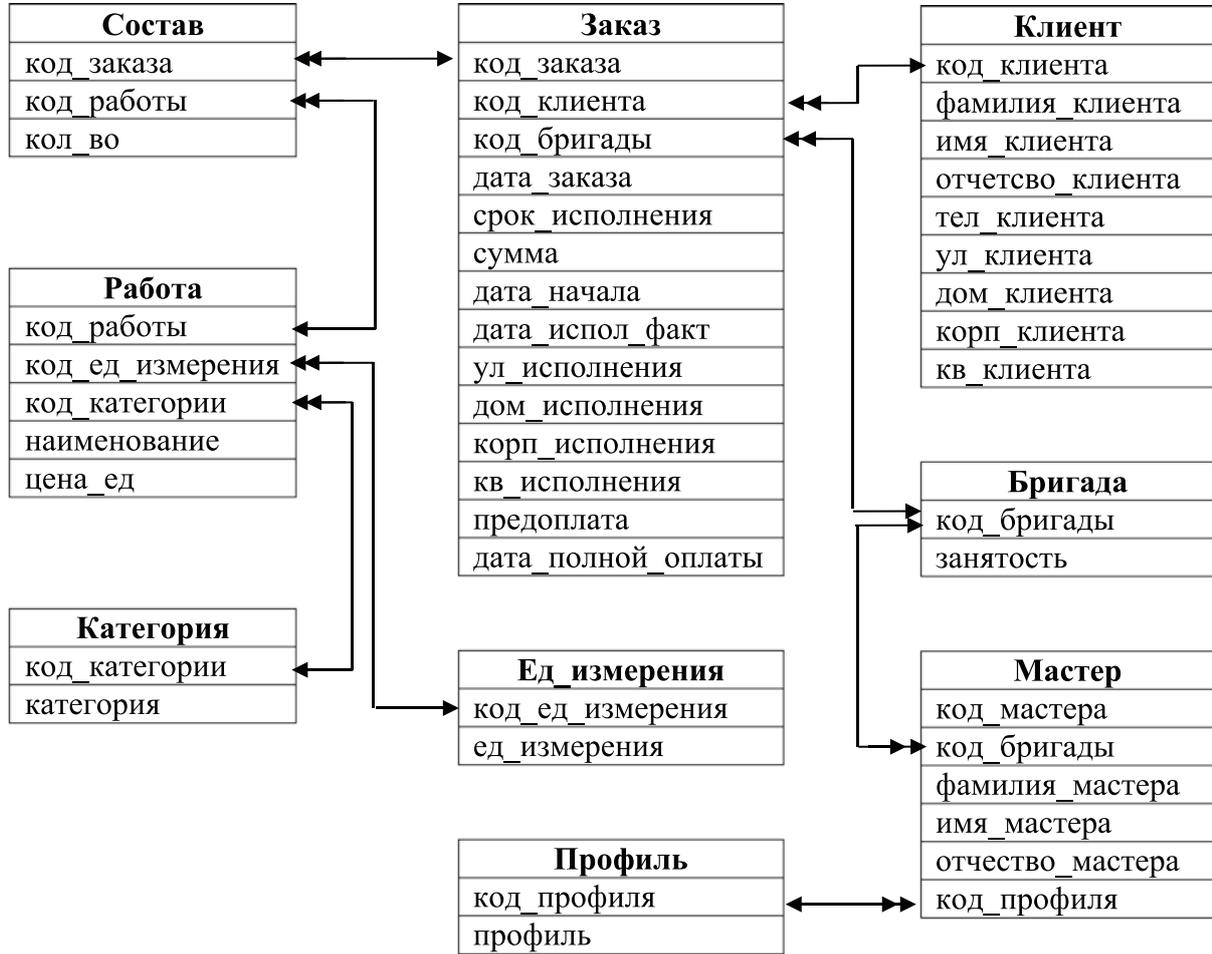


Рис. 4. Диаграмма взаимосвязей между атрибутами сущностей в третьей нормальной форме

Теперь получена модель базы данных, которую можно реализовать средствами выбранной СУБД.

Литература:

1. Хомоненко, А., Цыганков В., Мальцев М. Базы данных: Учебник для высших учебных заведений. — М.: КО-РОНА-принт, 2011.
2. Кузин, А. В. Базы данных: учеб. пособие для студ. высш. учеб. заведений / А. В. Кузин, С. В. Левонисова. — 5-е изд., испр. — М.: Издательский центр «Академия», 2012.

Инфографика как метод визуального анализа данных

Пудовкин Валентин Викторович, студент магистратуры

Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнёва (г. Красноярск)

Количество информации, потребляемой человеком, постоянно увеличивается, а её усвояемость находится на низком уровне.

Современному человеку для работы приходится работать с большими массивами данных, которые имеют свойство быстро меняться ввиду высокой динамики и слабой предсказуемости внешней, среды.

Поэтому на сегодняшний день требуется решение по структуризации и визуальной интерпретации данных для повышения эффективности работы с информацией.

Существуют различные методы визуализации данных в различных сферах деятельности такие как, текстовое оформление, графики, изображения. Они, в свою очередь, привязаны к профессиональным обществам и имеют специфические направленности. А во время обмена данными с представителями других профессий, возникают сложности в понимании процессов, демонстрируемых при помощи привычных для специалистов одной области средств. К примеру, некоторые данные могут содержать в себе терминологию и графические элементы, которые понятны только для определенных областей.

Стандартными способами визуального представления информации является текст, графические элементы, цветное сопровождение. Но иногда данные средства в том виде, в котором они существуют, не всегда отвечают поставленным задачам, представленным данных в той или иной среде. Таким образом, необходимо более четкое изображение данных — их формулировка, для эффективности усвоения информации, которое, как правило, представлено в виде текста. Одним из способов, отвечающим такой задаче, является инфографика.

Инфографика — это графический способ подачи информации, данных и знаний, целью которого является быстро и четко преподнести сложную информацию [1].

Помимо организации данных, различных по своему объёму, инфографике присуща способность при помощи имеющихся инструментов отразить динамику объектов во временных и пространственных интервалах, а также показать существующие тенденции протекающих процессов.

Инфографика может быть в виде таблиц, графиков, карты, алгоритма [2]. Кроме того, инфографику можно разделить по подаче информации: сравнение, где важно подчеркнуть разницу исходных и результативных данных, сценарий, где описывается процесс или действие, описывающееся во времени, коллаж, в которой необходимо показать структуру и типы данных. с помощью фрагментов отдельных изображений.

Исходя из возможностей использования различных графических элементов и цвета, важно гармонично их располагать и точно устанавливать взаимосвязь, чтобы инфографика совмещала в себе простоту и содержала в себе изначальную смысловую нагрузку, заложенную автором.

К примеру, существует задача, сформулированная Джоном Стерманом и Линдой Бут Суини. Есть емкость, в которую вливается и одновременно выливается вода [3]. Из условий задачи, следует, изначальное количество воды равно 100 л, а потоки воды имеют разную скорость и разную динамику, представленную на рисунке 1, где по оси Y отложены данные по скорости потоков (в литрах в минуту), а по оси X — время в минутах. Входящий поток воды, изображенный сплошной линией, изменяется неравномерно — линейно убывает с пикового значения в 100 л в минуту до нуля в период с начала до четвертой минуты, далее линейно возрастает с нуля до значения 100 л в минуту с 4-й по 12-ю минуту и затем цикл снова повторяется. В то время, как исходящий поток стабилен на всем отрезке времени и составляет 50 л в минуту. Вопрос: Как меняется количество воды в ванночке в течение времени?

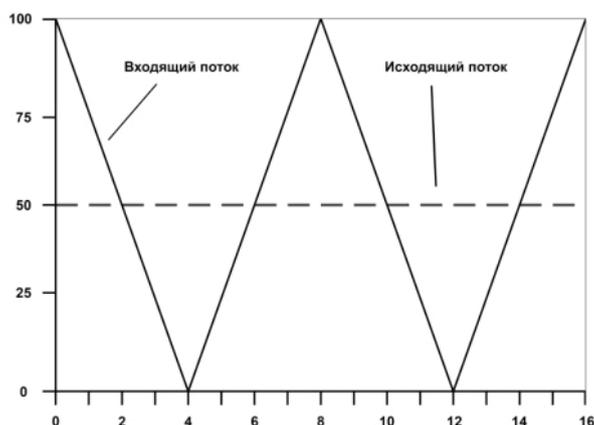


Рис. 1. Соотношение входящих и исходящих потоков воды

Данная задача решалась с помощью инфографики. На рис. 1 представлен график, в котором ясно изображены две оси: ось Y, отображающая скорость потоков воды (литры в минуту) и ось X, на которой отмечено время (в минутах).

Помимо этого, на графике четко представлена графическая визуализация входящего потока и исходящего потока воды. Уже исходя из условий задачи, можно понять, что в качестве непрерывной линии изображен входящий поток, который имеет неравномерную скорость и динамику, в то время как исходящий поток представлен пунктирной, параллельной оси X линией, отображающей стабильность потока. Исходя из вопроса в данной задаче, а именно «Как меняется количество воды в ванночке в течение времени?», сложно визуально увидеть количественное изменение воды в емкости. По результатам эксперимента, в котором и предлагалось изобразить изменения воды во время движения входящего и исходящего

потоков, только 36% студентов, которым было предложено решить данную задачу, смогли приблизительно найти наиболее корректное решение для изображения графика.

Данную задачу графически можно решить с помощью введения дополнительного графического элемента, а именно — цвета [4]. Цветовое сопровождение на графике имеет цель решить главную проблему в понимании задач такого типа, а именно — процесса аккумуляции (когда баланс входящих и исходящих потоков влияет на количество определенного вещества: будь то входящие и потраченные калории, и накопление жиров или ванночка как в описываемом примере). За всей представляемой на графике информацией теряется понимание общей динамики исследуемого вещества, что может негативно сказаться при принятии решения, основываясь на имеющихся данных. На рис. 2 каждому введенному цвету соответствует некий колеблющийся объем вещества.

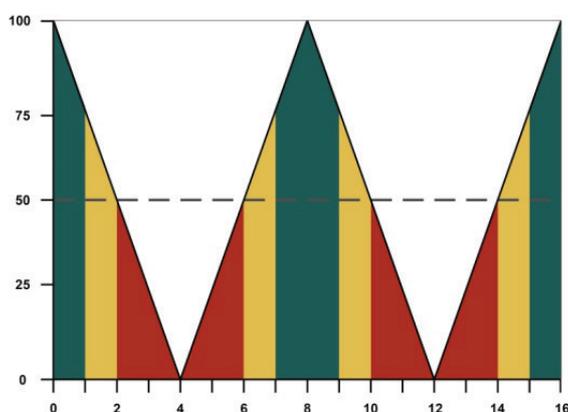


Рис. 2. График с цветовым сопровождением, показывающим уровень воды

Условно: зеленый — 75–100 литров, желтый — 50–75 литров и красный 0–50 литров. Теперь отслеживая динамику входящего и исходящего потока легче отслеживать и уровень воды в емкости, что существенно облегчает помимо понимания еще и время на чтение графически представленной информации.

Ещё одним вариантом изображения уровня воды на графике с помощью цвета можно решить градиентом. На рис. 3 был выбран голубой цвет для градиента, так как этот цвет ассоциативно связан с водой.

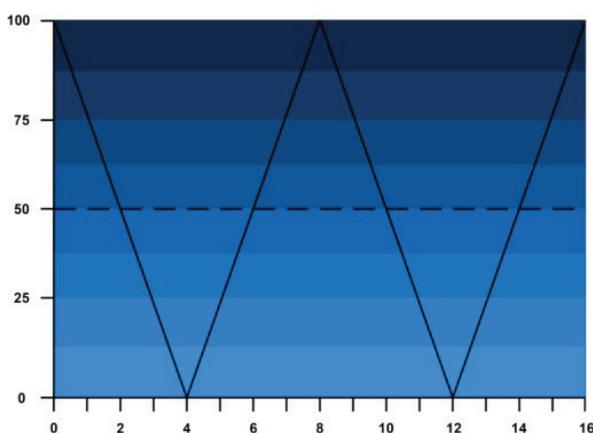


Рис. 3. График с цветовым градиентом

Каждый сегмент обозначает определённый уровень воды, где более тёмным цветом показаны уровни 75–100 л, а светлым — 0–25 л. Таким образом, помимо направляющих линий входящего и исходящего потоков, цвет также помогает решить задачу изображения количества воды в емкости.

Множество ресурсов содержат в себе большие объёмы информации, которые представлены и в виде текста, и в

сопровожаемой, либо самостоятельной инфографике. Таким образом, для эффективности усвоения информации необходимо более четкое изображение данных. Грамотная подача и оформление текста или других графических элементов в виде схем и графиков даёт возможность быстрого прочтения и понимания представленной информации.

Литература:

1. Newsom, D., Haynes, J. Public Relations Writing: Form & Style. — Cengage Learning, 2010. — 448 p.
2. Создание инфографики // Web Focus. URL: <https://webfocus.by/design/sozdanie-infografiki> (дата обращения: 20.04.2019).
3. Каталевский, Д. Ю. Основы имитационного моделирования и системного анализа в управлении: учебное пособие. — 2-е изд. перераб. и доп. — М.: Издательский дом «Дело», 2015. — 496 с.
4. Советы о выборе палитры для инфографики // Look At Me. URL: <http://www.lookatme.ru/mag/live/experience-news/218241-palettes> (дата обращения: 20.04.2019).

Инструменты проектирования виртуальных помощников

Якуничева Елена Николаевна, кандидат технических наук, доцент;

Горшкова Александра Алексеевна, студент магистратуры

Санкт-Петербургский государственный университет промышленных технологий и дизайна

В статье представлен сравнительный анализ инструментов проектирования виртуальных помощников с точки зрения используемых технологий, наличие дружелюбного интерфейса, возможности выбора канала распространения; наличия дополнительных информационных материалов и справочников.

Ключевые слова: виртуальный помощник, чат-бот, искусственный интеллект, машинное обучение.

Развитие IT-инструментов для проектирования виртуальных помощников, в частности чат-ботов, имеет очень положительную тенденцию. Данное направление становится все популярнее как среди владельцев бизнеса, которым чат-боты помогают решать ряд основных вопросов компании, касающихся консультирования, поддержания контактов и увеличения продаж, так и среди обычных пользователей, вдохновленных романами и кинематографом, дающими человечеству идею о том, что искусственный интеллект может обладать разнообразными функциональными возможностями и быть полезным человеку.

Авторами был проведен сравнительный анализ инструментов для проектирования виртуальных помощников. Перед проведением анализа был выдвинут список требований к рассматриваемым сервисам:

- наличие технологии NLU (Natural Language Understanding — понимание естественного языка);
- возможность анализа диалогов;
- наличие статистики;
- возможность создавать базы знаний в платформе;
- наличие открытого API для интеграции с другими платформами;

- наличие машинного обучения;
- понимание русского языка;
- наличие дружелюбного интерфейса;
- возможность выбора канала распространения;
- наличие бесплатной технической поддержки;
- наличие дополнительных информационных материалов, вебинаров, справочников.

Были рассмотрены наиболее популярные среди разработчиков инструменты проектирования виртуальных помощников, в том числе и нейронные сети. Нейронная сеть представляет собой не конструктор с удобным интерфейсом, а математическую модель в аппаратном воплощении, но несмотря на это она предоставляет возможность проводить глубокое машинное обучение и является конкурентоспособным инструментом.

В таблице 1 приведена краткая информация по следующим параметрам рассматриваемых сервисов:

- наименование;
- тип лицензии;
- преимущества;
- недостатки;
- информационные источники;
- каналы распространения.

Таблица 1

Обзор инструментов проектирования виртуальных помощников

Наименование, страна разработки	Тип лицензии	Преимущества	Недостатки	Информационные источники	Каналы распространения
Aimylogic (Россия)	Условно бесплатная (от 990 рублей в месяц)	Сбор и анализ диалога, машинное обучение, технология NLU, база знаний, статистика, дружелюбный интерфейс, голосовое ассистирование, взаимодействие с другими сервисами, шаблоны	Платные функции, нельзя подгружать словари	Техническая поддержка, вебинары от компании, справочники	ВКонтакте, Facebook, Telegram, Алиса, Google Assistant, виджет на сайт
Dialogflow (США)	Условно бесплатная (стоимость зависит от задач проекта)	Аналитика диалогов, машинное обучение, поддерживает 20 языков, взаимодействие с другими сервисами, чат-бот имеет собственную память, шаблоны	Платные функции, новая политика, поддержка на английском языке, минимальное количество полезной информации на русском языке	Техническая поддержка, справочники на английском языке	Google Assistant, Amazon Alexa, Facebook Messenger, Telegram, Viber, Twitter
ManyChat (США)	Бесплатный пробный период, далее от 10\$ в месяц	Чат-бот сообщает разработчику об ошибках, разработчики сервиса могут самостоятельно создать чат-бота за оплату, аналитика и статистика	Чат-бот не знает русского языка, в бесплатной версии ограничен функционал, подходит только для Facebook Messenger	Бесплатный видеокурс на английском языке, техническая поддержка	Facebook Messenger
Bot Framework (США)	Оплата только за необходимые функции	Шаблоны, чат-бот может распознавать фотографии и лица людей, давать советы, определять настроение пользователя, переводить текст	Техническая поддержка от 29\$, знание C#, Node. JS, подходит для коммерческих решений, привязан к продуктам Microsoft	Техническая поддержка, учебник	Facebook Messenger, Skype, Telegram, Slack, Microsoft Teams
Amazon Lex (США)	Платная (от 2,38\$ в месяц)	Можно интегрировать с «умным домом», технология NLU, статистика и анализ диалогов, шаблоны, дружелюбный интерфейс, техническая поддержка на русском языке, возможно создать голосового помощника	Обязательная интеграция с AWS, ограниченное число каналов	Бесплатные вебинары, учебник, техническая поддержка	Facebook, Slack, Kik
ChatFuel (США)	Условно бесплатная, платная от 15\$	Простая настройка, разработчик может вмешаться в диалог, шаблоны, платная помощь при разработке при необходимости, возможность встроить в диалог карту, картинку, видео, анализ диалогов, статистика	Бесплатный тариф включает только 100 пользователей, качественная техническая поддержка доступна только в тарифе «Premium» по 300\$, подходит только для Facebook	Платная техническая поддержка от 300\$	Facebook
Нейронные сети	Бесплатная	Глубокое машинное обучение, решение глобальных проблем, возможность обучить практически всем навыкам	Необходимость в виртуальном окружении, наличие серверов и специализированного ПО, знания языков программирования, ориентирован на крупные проекты	Справочники, учебники, статьи в Сети Интернет	Любой канал

Для создания небольших по функционалу чат-ботов, которые будут заданным образом реагировать на фразы пользователя, специальных навыков программирования в большинстве случаев не требуется (Aimylogic, ManyChat, ChatFuel). Но для разработки более интеллектуального продукта нужны навыки программирования в C#, Python, PHP, навыки работы с Node. JS, знания из высшей математики (Aimylogic, Bot Framework, Amazon Lex, нейронные сети).

Опираясь на выдвинутые требования и проведенный анализ был сделан вывод, что наиболее удобным инструментом для создания виртуального помощника по подбору необходимого пользователю контента является Aimylogic (Российская компания «Just AI»). Данный конструктор содержит модуль NLU, который позволяет чат-боту понимать смысл речи пользователя, не используя программирование во время создания проекта. Имеется возможность создать чат-бота на русском и английском языках, а так же подключить его к разным каналам: ВКонтакте и Facebook (социальные сети), Telegram мессенджер, Алиса. Яндекс и Google Assistant (виртуальные помощники) и чат-виджет для сайта. С помощью данной платформы можно создавать базы знаний, которые помогут чат-боту понимать суть вопроса и смысл реплики от пользователей, следовательно, таким образом открываются возможности вводить машинное обучение.

Основная функциональная рабочая область — вкладка «Сценарий», в которой содержится алгоритм проведения диалога между пользователем и чат-ботом. Для реализации алгоритма существует функциональные «Блоки»:

— «Ответ бота» содержит возможность внести реплику и изображение, которые будет присылать чат-бот пользователю при общении. Этот блок является основной

функциональной частью, позволяющий виртуальному помощнику общаться с человеком;

— «Ввод пользователя». Данный блок содержит в себе несколько функциональных возможностей для пользователя: ввести текст, число, номер телефона или написать командную фразу.

Эти блоки позволяют строить осмысленное общение с ботом и предоставляют возможность боту запоминать необходимую информацию не путая данные. Преимуществом такого подхода выступает то, что пользователь не знает о том, что введенная им информация имеет разный тип данных для чат-бота. Пользователь видит диалог с ботом, как обычное общение с человеком, следовательно, у него повышается уровень доверия к виртуальному помощнику.

— «Действия» — функциональный блок для более продвинутых пользователей, но имеющих минимальные навыки программирования и знания в области веб-приложений. С его помощью создаются логические условия, передаются данные по HTTP-запросу и осуществляются переходы между экранами сценария.

Конструктор Aimylogic подходит как для коммерческих, так для не коммерческих проектов, обладает интуитивно понятным интерфейсом и практически безграничными возможностями для проектирования. Минусом платформы является расширенный платный тариф и ограниченное количество пользователей в месяц.

Заключение

Проведенное исследование демонстрирует, что в зависимости от тематики, направленности, масштаба, цели проекта и навыков разработчика можно подобрать наиболее подходящий сервис, способный решать поставленные задачи.

ЭЛЕКТРОНИКА, РАДИОТЕХНИКА И СВЯЗЬ

Структурные особенности построения прецизионной системы управления ветрогенератором

Айтулина Айжан Мингожановна, докторант;

Ивель Виктор Петрович, профессор;

Герасимова Юлия Викторовна, доцент;

Калиаскаров Нурбол Балтабаевич, докторант

Северо-Казахстанский государственный университет имени М. Козыбаева (г. Петропавловск)

Современные ветроэнергетические установки или ветротурбины — это комплекс взаимосвязанного оборудования и сооружений, которые преобразуют кинетическую энергию ветра в механическую энергию вращающегося ветроколеса, а затем в электрическую энергию.

Ветрогенераторы используются для зарядки аккумуляторных батарей может упоминаться как ветер зарядное устройство. Результат более чем тысячелетнего развития ветроэнергетики и современной инженерии, современных ветровых турбин изготавливаются в широком диапазоне вертикальных и горизонтальных осей типов. Общая структура ветротурбины обычно включает следующие блоки [1]:

- ветроколесо, как правило, состоящее из трех лопастей;
- управляемый редуктор;
- ветрогенератор;
- система управления включая датчики измерения скорости ветра, давления и др.

При проектировании ветроустановок надо учитывать следующие их особенности:

- для обеспечения максимальной эффективности работы ветроколеса следует изменять частоту его вращения при изменении скорости ветра, сохраняя постоянным коэффициент быстроходности, в то же время для максимально эффективной работы ветрогенератора необходима практически постоянная частота вращения;

- механические системы управления частотой вращения ветроколеса достаточно сложны и дороги. Гораздо эффективнее и дешевле управлять частотой его вращения, изменяя электрическую нагрузку ветрогенератора;

- оптимальная частота вращения ветроколеса тем меньше, чем больше его радиус, поэтому только очень малые ветроколеса (радиусом не более 2 м) удаётся соединить с генератором напрямую. При больших размерах ветроколеса приходится использовать мультипликаторы, удорожающие ветроустановку и её обслуживание. Альтернативой им могут стать многополюсные ветрогенераторы, работающие при меньших частотах вращения;

- в конструкции ветроустановки предусматривается, как правило, возможность отключения генератора от ветроколеса и вращения его от химического или механического аккумулятора энергии, поэтому систему управления ветрогенератором не связывают с работой ветроколеса. При отсутствии такой связи даже при мягком соединении генератора с ветроколесом необходимы специальные демпфирующие устройства для исключения механических ударов, перегрузок и бросков напряжений на выходе ветрогенератора [2].

Проведенные ранее обширные исследования систем управления ветровых турбин постоянной скорости показали, что значительное увеличение эффективности может быть получено при применении соответствующего контроллера (регулятора) в рамках стратегии нелинейного управления. Программа разработки подобных систем регулирования ветровых турбин с регулируемой скоростью вращения ветрогенератора в настоящее время является предметом интенсивных исследований. Поэтому данная тема является актуальной [3]. Задачей данного исследования является разработка прецизионной системы управления ветрогенератором. При решении поставленных задач использованы математический аппарат теории автоматического управления, методы математического моделирования. На рисунке 1 представлена структурная схема системы управления ветроэнергетической установки.

На рисунке 1 приняты следующие обозначения: ПДЗ — пропорционально-дифференцирующие звенья; ЭД_{вк} — электродвигатель для поворота лопастей ветроколеса; БОМ — блок определения максимального значения мощности ветроколеса; Д_{ωр} — датчик угловой скорости ротора; ИЗ — интегрирующее звено; Д_{ωк} — датчик угловой скорости ветроколеса; ВК — ветроколесо (блок умножения); БД — блок дискретизации; РЕД — редуктор (блок умножения); Г — электрогенератор; p — оператор Лапласа; Ст — стабилизатор напряжения; $\omega_{звк}$ — заданная угловая скорость ветроколеса; $\omega_{зр}$ — заданная угловая скорость редуктора или электрогенератора; $\omega_{вк}$ — угловая скорость ветроколеса; $P_{\max}(v_b)$ — блок определения максимального значения мощности ветроколеса; v_b — скорость

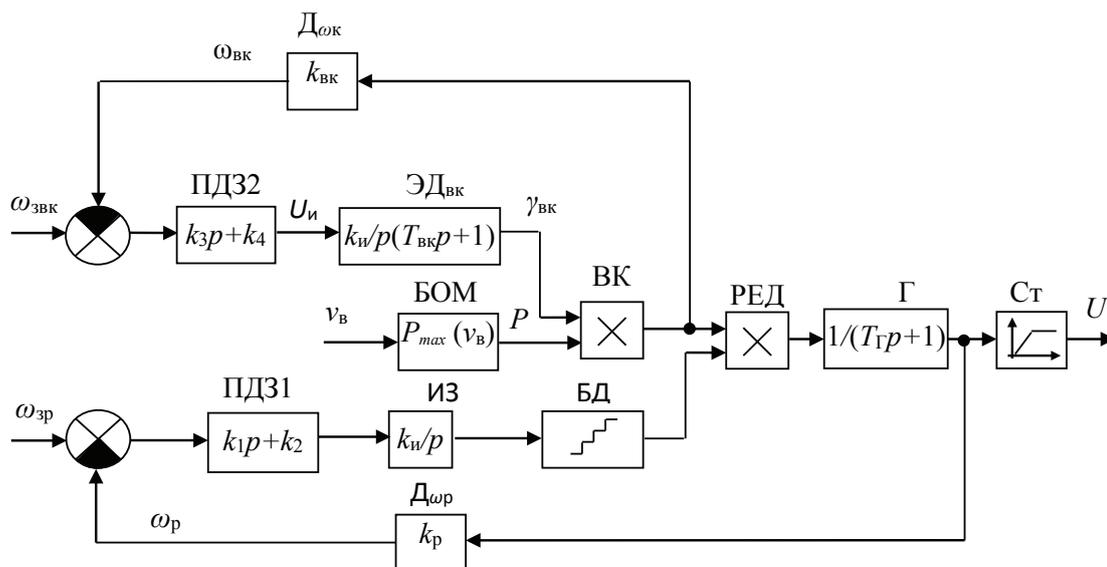


Рис. 1. Структурная схема системы управления ветрогенератором

ветра; P — мощность ветра; ω_p — угловая скорость редуктора; $\gamma_{вк}$ — угол поворота ветроколеса, U — напряжение на выходе генератора.

Устройство управления ветрогенератором включает два контура: контур управления угловой скоростью ротора ветрогенератора и вспомогательный контур стабилизации скорости ветроколеса. Основной контур включает звенья: ПДЗ1 — звено, предназначенное для форсирования процесса управления, ИЗ — блок необходимый для придания переходному процессу апериодических свойств, Д_р — цифровой датчик измерения угловой скорости ротора ветрогенератора, БД — блок, моделирующий дискретный сигнал для управления редуктором, РЕД — редуктор, представляющий собой исполнительный механизм управляющий ротором и Г — звено, моделирующее работу электрогенератора. Вспомогательный контур необходим для регулирования угла поворота лопастей ветроколеса в зависимости от текущей скорости ветра. Контур включает звенья: ПДЗ2 — звено,

предназначенное для форсирования процесса управления по вспомогательному контуру, ЭД_{вк} — звено, моделирующее работу электродвигателя для поворота лопастей ветроколеса, $P_{max}(v_в)$ — блок, моделирующий внешнее воздействие как преобразователь скорости ветра в его мощность, ВК — блок умножения, на который поступают сигналы от блоков ЭД_{вк} и $P_{max}(v_в)$ и который является моделью ветроколеса. Скорость вращения ветроколеса зависит от двух параметров: угла поворота ветроколеса и скорости ветра и должна поддерживаться на заданном уровне. Передаточная функция электрогенератора служит для получения как угловой скорости ротора ω_p , так и выходного напряжения электрогенератора U . Блок С_т выполняет роль стабилизатора выходного напряжения.

Разработка прецизионной системы управления ветроэнергетической установка требует комплексного подхода с учетом множества факторов, важнейшие из которых — вид генератора, режим работы ветрогенератора, система регулирования угла поворота лопастей.

Литература:

1. Шингисов, Б. Т. Обоснование кинематических и конструктивных параметров электропривода генератора ветроустановки // Алматы 2015 г.
2. <http://www.mensh.ru/articles/osobennosti-vetrogeneratorov>
3. Босов, Е. А. Построение системы управления и диагностики генераторами ветровой электростанции // Автореферат на магистерскую диссертацию 2006 г.

Разработка устройства беспроводной системы для мониторинга состояния трещин и стыков зданий и мостовых сооружений с использованием двухпроцессорных Wi-Fi-передатчиков

Калиаскаров Нурбол Балтабаевич, докторант;

Ивель Виктор Петрович, профессор

Северо-Казахстанский государственный университет имени М. Козыбаева (г. Петропавловск, Казахстан)

Разинкин Владимир Павлович, профессор

Новосибирский государственный технический университет

Герасимова Юлия Викторовна, доцент

Северо-Казахстанский государственный университет имени М. Козыбаева (г. Петропавловск, Казахстан)

Несипова Салтанат Саматовна, преподаватель

Карагандинский государственный технический университет (Казахстан)

Современные средства автоматизированного контроля и измерений позволяют создавать системы удаленного мониторинга типовых мостовых сооружений. Это способствует широкому распространению этой технологии. Разработка систем мониторинга конкретных объектов сводится к решению задачи обеспечения заданного уровня безопасности при установленном ограничении на затраты путем подбора конфигурации системы. В рамках системного подхода необходимо определить: что следует понимать под безопасностью сооружения в контексте использования средств контроля, и каким образом можно оценить увеличение безопасности сооружения от внедрения системы мониторинга. В этом случае систематизированное проектирование систем мониторинга мостовых сооружений позволит анализировать конфигурацию системы при наличии общих ограничений на стоимость и уровень безопасности [1].

Мониторинг состояния моста (мониторинг моста) — систематическое наблюдение за работой моста в эксплуатационных условиях в течение заданного существенного промежутка времени с применением специальных технических средств, размещаемых на конструкциях моста. При мониторинге выполняется экспериментальная оценка количественных параметров (измерение) и качественных признаков, характеризующих техническое состояние моста, к 3 которым относятся геометрические параметры; напряженно-деформированное состояние; температура элементов сооружения; динамические характеристики; дефекты; нагрузки и воздействия, атмосферные и др. условия эксплуатации; жесткостные, прочностные и прочие свойства конструкций и материалов. Оцениваться могут как действующие значения параметров, так и их изменение в процессе мониторинга. Мониторинг состояния моста делится на различные виды, такие как: контрольный, исследовательский, сравнительный, непрерывный, периодический [2]. Все они отличаются друг от друга видами поступающей информации, временем их обработки и дальнейшего анализа и количеством объектов исследования.

Основной особенностью мониторинга является то, что он позволяет регулировать процесс содержания, который позволит сохранить мостовые сооружения и любые здания. При этом необходимо изменить идеологию содержания мостовых сооружений и любых зданий, исходя из следующей концепции: «пока сооружение не начало разрушаться, необходимо своевременно его защитить от повреждений» [3]. При разработке системы непрерывного мониторинга состояния мостового сооружения и зданий нужно создать такую систему, с помощью которой можно получать информацию о состоянии мостового сооружения и любых зданий в реальном режиме времени. Для разработки этой системы необходимо:

- выбрать регистрируемые факторы и аппаратные средства сбора данных;
- разработать и привести алгоритмы преобразования регистрируемых данных к виду, пригодному для контроля и анализа;
- сохранять данные и предоставлять их по запросу;
- выбирать средства и формы предоставления данных для пользователя [4].

Мотивацией для разработки устройства мониторинга технического состояния мостов и сооружений является то, что в результате сравнительного анализа существующих систем мониторинга выявлены некоторые недостатки, такие как:

- невозможность мониторинга трещины или стыка в здании, если контролируемая трещина находится в труднодоступном месте или на большой высоте;
- технические сложности при контроле большого количества трещин в здании или сооружении;
- применение в некоторых системах в качестве датчика измерения трещин датчика реостатного типа, выходной сигнал которого имеет ступенчатый характер, приводит к появлению дополнительной погрешности в выходном сигнале, а наличие контактных электромеханических частей в датчике снижает срок эксплуатации данного устройства;

— отсутствие контроля за динамикой изменения размеров трещины снижает информативность данной системы и качество прогноза о состоянии здания.

В данной работе представлены результаты разработки беспроводного устройства мониторинга состояния трещин и стыков зданий и сооружений лишенные отмеченных недостатков. Задачей данного исследования является разработка устройства мониторинга состояния трещин и стыков зданий и сооружений, основанного на использовании дополнительных блоков, позволяющих уменьшить потребление электроэнергии, увеличить срок эксплуатации устройства и повысить точность, информативность и качество прогноза о состоянии сооружений и зданий.

Решение задачи улучшения качественных характеристик и снижения электропотребления устройством мониторинга базируется на использовании метода построения автономных беспроводных систем с минимальным расходом электроэнергии, триангуляционном методе измерения расстояния до объекта, основанного на изменении угла падения луча лазера в зависимости от расстояния до объекта, применении схемотехнических методов построения интегральных электронных модулей. На рисунке 1 представлена структурная схема беспроводного устройства мониторинга состояния трещин и стыков зданий и сооружений.

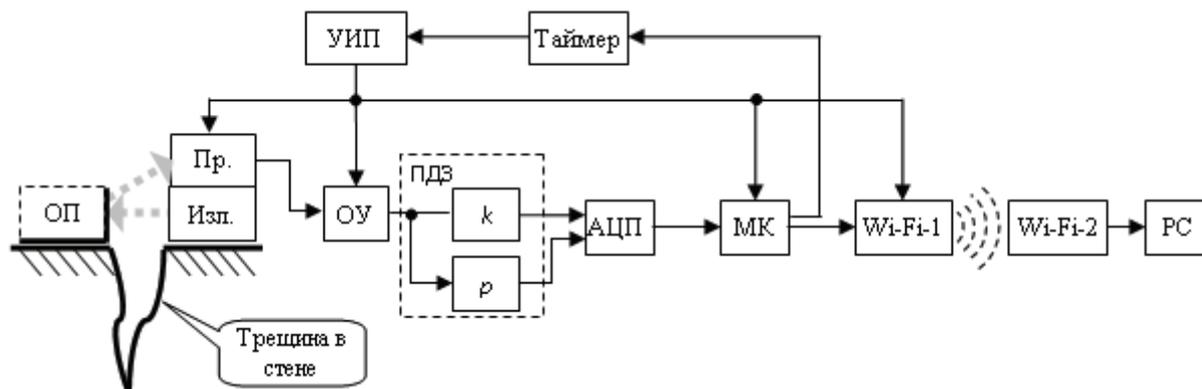


Рис. 1. Структура беспроводного устройства мониторинга состояния трещин и стыков зданий и сооружений

Общая структура устройства мониторинга (рисунок 1) включает в себя оптический датчик, состоящий из отражающей панели (ОП), излучателя (Изл) и приемника (Пр), операционный усилитель (ОУ), аналого-цифровой преобразователь (АЦП), пропорционально-дифференцирующее звено (ПДЗ), микроконтроллер (МК), Wi-Fi передатчик (Wi-Fi-1), Wi-Fi приемник (Wi-Fi-2), персональный компьютер (PC), управляемый источник питания (УИП) и таймер.

Оптический датчик передает аналоговый сигнал пропорциональный размеру трещины на усилитель, выход которого подключен к пропорционально-дифференцирующему звену. Сигнал с ПД-звена поступает на АЦП. Затем оцифрованный сигнал с АЦП передается на микроконтроллер. В микроконтроллере сигнал подвергается обработке в соответствии с заданным алгоритмом и далее по Wi-Fi-интерфейсу, который включает передатчик и приемник, поступает на компьютер, где происходит окончательная обработка информационного сигнала. Кроме информационного сигнала микроконтроллер формирует сигнал включающий таймер, функция которого заключается в периодическом отключении источника питания от всех активных блоков устройства мониторинга с целью экономии электроэнергии.

Устройство работает следующим образом. При расширении контролируемой трещины увеличивается расстояние между отражающей панелью и приемником, так как, входящие в состав оптического датчика измерения трещин

приемник и излучатель расположены с одной стороны трещины, а отражающая панель — с другой. В результате с приемника снимается сигнал пропорциональный ширине контролируемой трещины.

В качестве датчика измерения трещин используется лазерный триангуляционный датчик класса РФ602Х, принцип работы которого основан на измерении угла между прямым лазерным лучом и отраженным от объекта.

Сигнал с приемника оптического датчика передается на высокоточный 16-разрядный АЦП ADS 1115 с низким энергопотреблением (150 мкА в рабочем режиме) и встроенным программируемым. Оцифрованный информационный сигнал (в структуре алгоритма сигнал обозначен как X1) по интерфейсу I2C поступает на Wi-Fi-передатчик, в качестве которого применяется Wi-Fi-модуль WeMos d1 mini pro. В состав модуля входит 32-битный микроконтроллер ESP8266EX. Достоинством предложенного Wi-Fi-модуля является его простота программирования. Для программирования модулей с микроконтроллером ESP8266EX рекомендуется использовать среду программирования Arduino Ide. Таким образом, модули ADS 1115 и WeMos d1 mini pro фактически выполняют функции всей цепочки, представленной на структурной схеме (рисунок 1) — ОУ ПДЗ АЦП МК Wi-Fi-1.

Отличительной особенностью данного устройства является наличие информации о скорости расширения из-

меряемой трещины, что дает наиболее полную картину о степени разрушительной активности в стенах контролируемого здания и позволяет оперативно реагировать на возникающие угрозы, а организация беспроводной передачи сигнала и разработанный алгоритм, включающий ограничение времени передачи полезного сигнала, позволяют значительно экономить расход электроэнергии и использовать автономный источник питания.

Кроме того, использование в качестве датчика измерения трещин оптического микрометра значительно повышает точность измерений, поскольку выходной сигнал датчика в этом случае имеет непрерывную форму, что позволяет дифференцировать этот сигнал. А отсутствие в датчике контактных электромеханических элементов, которые в процессе измерения могут подвергаться быстрому износу, повышает надежность всего устройства.

Литература:

1. Тулеушова, Р., Наурызбаев М.К. Оценка эффективности системы мониторинга мостового сооружения// Интернет-журнал «НАУКОВЕДЕНИЕ» Том 7, № 2 (2015) <http://naukovedenie.ru/PDF/04TVN215.pdf> (доступ свободный). Загл. с экрана. Яз. рус., англ. DOI: 10.15862/04TVN215.
2. Руководство по проведению мониторинга состояния эксплуатируемых мостовых сооружений. ОДМ 218.4.002—2008. Федеральное дорожное агентство (Росавтодор). Москва 2008.
3. Бильченко, А.В. Концепция сохранения и развития мостовых сооружений в г. Харькове до 2012 г. [Текст] / А.В. Бильченко, А.Г. Кислов, Е.А. Бадаева. // — Харьков, 2008. — 39 с.
4. Овчинников, И.Г., Овчинников И.И., Нигаматова О.И., Михалдыкин Е.С. Прочностной мониторинг мостовых сооружений и особенности его применения. Часть 2. Непрерывный мониторинг состояния мостовых сооружений / Овчинников И.Г., Овчинников И.И., Нигаматова О.И., Михалдыкин Е.С. // Транспортные сооружения, 2014. — Том 1, № 2. — с. 1—37.

АВТОМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Исследование методов настройки ПИД-регулятора для систем с малыми постоянными времени

Гусаров Александр Вячеславович, кандидат технических наук, доцент;

Кошляков Павел Сергеевич, студент магистратуры

Рыбинский государственный авиационный технический университет имени П. А. Соловьёва (Ярославской обл.)

В работе описывается исследование системы регулирования скорости вращения асинхронного двигателя. Данная тема является актуальной в связи с тем, что в настоящее время настройка параметров системы регулирования скорости вращения асинхронного двигателя является сложной задачей, требующей больших затрат времени на подбор коэффициентов.

Ключевые слова: ПИД-регулятор, асинхронный двигатель, автоподбор коэффициентов.

ПИД-регуляторы находят широкое применение в современных системах автоматического регулирования, таких как управление термосистемами и системами позиционирования. Использование ПИД-регуляторов помогает уменьшить энергетические потери на настройку системы и обеспечивают более быстрый выход на требуемые параметры.

Для корректной работы регулятора необходимо правильно подобрать его коэффициенты. От правильности подбора коэффициентов будет зависеть точность и скорость работы системы. Довольно часто эмпирический подбор коэффициентов не приносит желаемых результатов и занимает много времени. В связи с этим возникает необходимость в использовании математических методов настройки регулятора. Существует несколько таких методов, но в данной работе будут рассмотрены самые распространённые: метод Зиглера-Никольса, метод *CHR* и метод Коэна-Куна.

Зиглер и Никольс предложили два метода настройки ПИДрегуляторов [1]. Один из них основан на параметрах отклика объекта на единичный скачок, второй — на частотных характеристиках объекта управления.

В отличие от Зиглера и Никольса, которые использовали в качестве критерия качества настройки декремент затухания, равный 4, Chien, Hrones и Reswick (*CHR*) [2] использовали критерий максимальной скорости нарастания при отсутствии перерегулирования или при наличии не более чем 20процентного перерегулирования. Такой критерий позволяет получить больший запас устойчивости, чем в методе ЗиглераНикольса.

Метод *CHR* даёт две разные системы параметров регулятора. Одна из них получена при наблюдении отклика на изменение уставки (таблица 1.2), вторая — при наблюдении отклика на внешние возмущения (таблица 1.3). Какую систему параметров выбирать, зависит от того, что важнее для конкретного регулятора: качество регулирования при изменении уставки или ослабление внешних воздействий. Если же важно и то и другое, то необходимо использовать регуляторы с двумя степенями свободы.

Метод *CHR* использует аппроксимацию объекта моделью первого порядка с задержкой.

Правила настройки Коэна-Куна хорошо работают практически во всех саморегулирующихся процессах. Эти правила первоначально были разработаны для быстрого реагирования, но это приводит к колебаниям с большим перерегулированием. При небольшой модификации правил настройки Коэна-Куна контуры управления могут реагировать быстро, но они гораздо менее подвержены колебаниям [3].

Рассматриваемые методы, в основном, используются для систем с большой инерцией. Они хорошо работают с системами, в которых время регулирования занимает продолжительное время (единицы, десятки секунд и более). В связи с этим возникает необходимость проверить актуальность рассматриваемых методов для малоинерционной системы с малыми постоянными времени.

Для исследования методов настройки ПИД-регуляторов был разработан стенд, включающий в себя асинхронный двигатель, который вращает вентилятор с заслонкой на выходном патрубке вентилятора, датчик положения заслонки и имитатор изменения давления на заслонку, регулируемую поток воздуха от вентилятора. Реализация автоматического регулирования мощностью потока воздуха выполнена на основе ПИД-регулятора.

ПИД-регулятор формирует управляющий сигнал, являющийся суммой трёх слагаемых, первое из которых пропорционально разности входного сигнала и сигнала обратной связи (сигнал рассогласования), второе — интеграл сигнала рассогласования, третье — производная сигнала рассогласования.

В разработанном лабораторном стенде существует возможность использования двух вариантов реализаций ПИД-регулирования: рекуррентная и дискретная. Используя данные, полученные в ходе проведения эксперимента, были выбраны основные методы для расчёта коэффициентов регулятора.

При использовании программной реализации метода автоматического регулирования часто переходят к рекуррентной реализации регулятора. Особенностью рекуррентной реализации является использования ошибки рассогласования за текущий шаг и за два предыдущих:

$$U(n) = U(n-1) + K_p(\varepsilon(n) - \varepsilon(n-1)) + K_i\varepsilon(n) + K_d(\varepsilon(n) - 2\varepsilon(n-1) + \varepsilon(n-2)), \quad (1)$$

где ε — ошибка рассогласования; K_p , K_i , K_d — пропорциональный, интегральный и дифференциальный коэффициенты.

Ошибка рассогласования рассчитывается по формуле

$$\varepsilon = y_i - y_{\text{зад}}, \quad (2)$$

где y_i — выходное значение регулятора; $y_{\text{зад}}$ — заданное входное значение (уставка).

В дискретной реализации метода расчёта выходного сигнала используется формула

$$U(n) = K_p\varepsilon + K_i \sum_{i=0}^n \varepsilon_i + K_d(\varepsilon_i - \varepsilon_{i-1}), \quad (3)$$

где ε — ошибка рассогласования; K_p , K_i , K_d — пропорциональный, интегральный и дифференциальный коэффициенты.

Ошибка рассогласования рассчитывается по формуле (2).

В ходе проведённых экспериментов было определено, что метод Коэна-Куна не подходит для малоинерционных систем. С коэффициентами, полученными в результате расчёта методом Коэна-Куна, система ведёт себя нестабильно.

Методы *CHR* и Зиглера-Никольса являются актуальными для малоинерционных систем. Наиболее хорошо в малоинерционных системах реализуется метод *CHR*.

Литература:

1. Ziegler, J. G., Nichols N. B. Optimum settings for automatic controllers // Trans. ASME. 1942. Vol. 64. P. 759768.
2. Chien, K. L., Hrones J. A., Reswick J. B. On automatic control of generalized passive systems // Trans. ASME. 1952. Vol. 74. P. 175185.
3. Метод Коэна-Куна. [Электронный ресурс]. URL: <https://www.dataforth.com/tuning-control-loops-for-fast-response.aspx> (дата обращения 20.04.1019).
4. ПИД-регулятор. [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/ПИД-регулятор> (дата обращения 20.04.1019).
5. Назначение регуляторов. [Электронный ресурс]. URL:
6. <https://elektronchic.ru/avtomatika/pid-regulyator-polnoe-opisanie-primeneniye.html> (дата обращения 20.04.1019).

Реализация методов настройки ПИД-регулятора для малоинерционных систем

Гусаров Александр Вячеславович, кандидат технических наук, доцент;

Медведев Евгений Юрьевич, студент магистратуры

Рыбинский государственный авиационный технический университет имени П. А. Соловьёва (Ярославская обл.)

В работе описывается реализация методов регулирования скорости вращения асинхронного двигателя. Данная тема является актуальной в связи с тем, что в настоящее время настройка параметров системы регулирования скорости вращения асинхронного двигателя является сложной задачей, требующей больших затрат времени на подбор коэффициентов.

Ключевые слова: регулирование скорости вращения, ПИД-регулятор, асинхронный двигатель, автоподбор коэффициентов, алгоритмы, программная реализация.

Регулирование скорости вращения мощного асинхронного двигателя при переменной нагрузке является сложной задачей. Для этого чаще всего используют ПИД-регуляторы. Использование ПИД-регуляторов помогает уменьшить энергетические потери на настройку системы и обеспечивает более быстрый выход на требуемые параметры.

Для эффективной работы регулятора в процессе работы необходимо правильно подбирать его коэффициенты и изменять их при изменении внешних условий. В связи с этим возникает необходимость в разработке программного обеспечения для реализации математических методов настройки регулятора.

В реализованном методе расчёта выходного сигнала $U[n]$ через промежутки времени, равные периоду дискретизации T , в моменты времени $n \cdot T$ используется формула

$$U[n] = K_p \cdot \varepsilon_n + K_i \cdot \sum_{n=0}^N \varepsilon_n + K_d \cdot (\varepsilon_n - \varepsilon_{n-1}), \quad (1)$$

где n — номер текущего периода дискретизации;

ε_n и ε_{n-1} — текущая и предыдущая ошибка рассогласования соответственно;

K_p, K_i, K_d — пропорциональный, интегральный и дифференциальный коэффициенты ПИД-регулятора.

Ошибка рассогласования ε_n рассчитывается по формуле

$$\varepsilon_n = y_n - y_{\text{зад}}, \quad (2)$$

где y_n — выходное значение регулятора; $y_{\text{зад}}$ — заданное входное значение (уставка).

В процессе реализации регуляторов часто возникает ситуация, когда нет возможности точно поддерживать одинаковый период дискретизации при программном управлении. В этом случае сильно ухудшается качество управления, так как в процессе расчета выходного сигнала не учитывается неравномерность периода дискретизации. Это приводит к появлению такой проблемы, как неверное вычисление значения выходного сигнала на текущем шаге. В этом случае следует выполнить переход к описанию коэффициентов регулятора через постоянные времени интегрирования T_i и дифференцирования T_d :

$$K_i = \frac{T}{T_i}, \quad (3)$$

$$K_d = \frac{T_d}{T}, \quad (4)$$

где T — период дискретизации.

Окончательно формула для расчета выходного сигнала цифрового регулятора $U[n]$ примет вид:

$$U[n] = K_p \cdot \left(\varepsilon_n + \frac{T_n}{T_i} \cdot \sum_{n=0}^N \varepsilon_n + \frac{T_d}{T_n} \cdot (\varepsilon_n - \varepsilon_{n-1}) \right), \quad (5)$$

где n — номер текущего периода дискретизации;

T_n — период дискретизации на текущем шаге;

t_n, t_{n-1} — время запуска функции регулятора на текущем и предыдущем шаге;

T_i — постоянная времени интегрирования;

T_d — постоянная времени дифференцирования.

K_p — пропорциональный коэффициент ПИД-регулятора.

Период дискретизации на текущем шаге определяется по формуле

$$T_n = t_n - t_{n-1}. \quad (6)$$

Данный переход полностью устраняет обозначенную выше проблему ухудшения качества регулирования, так как в нем учитывается время, прошедшее с момента предыдущего расчета выходного сигнала.

В целях адаптации классического алгоритма расчета коэффициентов к решаемой задаче был разработан и реализован модифицированный программный ПИД-регулятор. Алгоритм функции расчёта коэффициентов для классического ПИД-регулятора представлен на рис. 1, а, для метода модифицированного ПИД-регулятора — на рис. 1, б.

Ниже приведен текст подпрограммы для реализации регулятора.

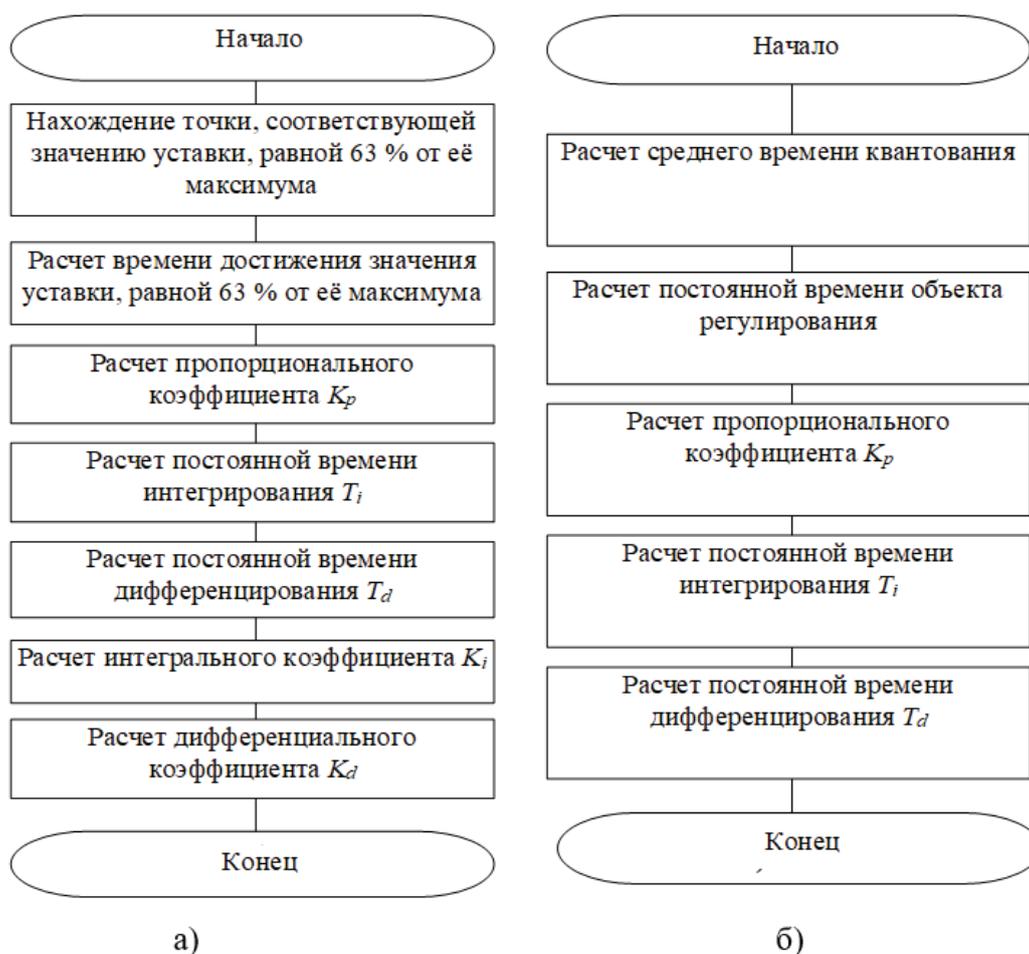


Рис. 1. Алгоритм расчета коэффициентов для ПИД-регулятора: а) классического; б) модифицированного

```

private double prevError;
private double prevCur;
private double prevPrevCur;
private double out;
private double sumError;
private int PidNumber;
public double PIDcontroll(double Kp, double Ti, double Td, double dT, double setP, double Cur)
{
    if (Cur > 100) Cur = 100;
    if (Cur < 0) Cur = 0;
    if (setP > 100) Cur = 100;
    if (setP < 0) Cur = 0;
    double curError = setP - Cur;
    double velocity = curError - prevError;
  
```

```
switch (PidNumber) {
    case (0):
        out = setP;
        break;
    case (1):
        sumError = sumError + curError;
        out = Kp * (curError + dT / Ti * sumError + Td / dT * velocity);
        break;
    case (2):
        out += Kp * (-Cur + prevCur + dT / Ti * (curError) + Td / dT * (-Cur + 2 * prevCur -
prevPrevCur));
        break;
}
prevError = curError;
prevPrevCur = prevCur;
prevCur = Cur;
if (out > 100) { out = 100; }
if ((out < 0)) { out = 0; }
return out;
}
```

Проведенные на стенде испытания показали работоспособность предложенного алгоритма и его программной реализации.

Литература:

1. ПИД-регулятор. [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/ПИД-регулятор> (дата обращения 20.04.1019).
2. Назначение регуляторов. [Электронный ресурс]. URL:
3. <https://elektronhic.ru/avtomatika/pid-regulyator-polnoe-opisanie-primenenie.html> (дата обращения 20.04.1019).
4. Сазонов, А. А. Микропроцессорное управление технологическим оборудованием микроэлектроники: учеб. пособие / А. А. Сазонов. — М.: Радио и связь, 1998. — 264 с.
5. Радионов, В. Д. Технические средства АСУ ТП: учеб. пособие для вузов / В. Д. Радионов. — М.: Высшая школа, 1989. — 263 с.

Исследование современных методов тестирования и диагностики компьютерных систем

Добржинская Татьяна Юрьевна, аспирант;
Рогова Олеся Сергеевна, аспирант;
Алентьева Екатерина Романовна, аспирант
Дальневосточный федеральный университет (г. Владивосток)

По мере того как сфера применения цифровых вычислительных систем расширялась и нашла своё применение в промышленности, сельском хозяйстве, транспортной сфере, значение задачи обеспечения правильности их функционирования резко возросло.

В прошлом вычислительные системы использовались главным образом в неоперативном режиме для пакетной обработки, и если неисправности аппаратных средств своевременно не обнаруживались, это не имело серьезных последствий. В настоящее время вычислительные машины все больше используются в оперативном режиме, работая в реальном времени в качестве средств управления химическими агрегатами и ядерными реакторами, а также в автоматизированных системах управления транспортом [1, с. 6].

В результате необходимости стабильной работы всех вышеперечисленных систем возникает необходимость в диагностическом контроле автоматизированных систем.

Основными задачами контроля и диагностики ЭВМ является обеспечение безопасности, функциональной надёжности и эффективности работы технического объекта, а также сокращение затрат на его техническое обслуживание и уменьшение потерь от простоев в результате отказов и преждевременных выводов в ремонт [2].

Способы борьбы с неисправностями компонентов автоматизированных систем:

избыточность — нейтрализующая эффект, обусловленный неисправностями;

диагностические процедуры — позволяющие обнаружить неисправность.

Нейтрализация неисправностей служит лишь для того, чтобы отдалить неизбежное. С течением времени накопится столько неисправностей, что средства нейтрализации не будут способны компенсировать их отрицательный эффект. Поэтому такой способ полезен в первую очередь в таких применениях, как управление космическими кораблями, где правильную работу машины необходимо гарантировать на относительно короткий период времени, а ремонт затруднен или невозможен.

С другой стороны, метод, основанный на обнаружении неисправностей в сочетании с ремонтом, полезен тогда, когда остановки машины допустимы, однако неисправности нельзя оставлять необнаруженными. Это соответствует случаю работы в режиме постоянной готовности. В определенном смысле нейтрализация неисправности и диагностика неисправности несовместимы, поскольку эффект нейтрализованной неисправности нельзя наблюдать

или диагностировать, по крайней мере до тех пор, пока не проявятся дальнейшие неисправности.

Для начала стоит разобраться с причинами, которые могут вызвать такое явление. Как известно и пыль, и неблагоприятные климатические условия ухудшают состояние компонентов ПК. Соответственно, выход железа из строя может быть вызван окислением контактов, попаданием пыли (и следственно, статического электричества) на микросхемы и разъемы, их перегрев. Перегрев также может быть вызван и плохим охлаждением.

Однако были разработаны многочисленные методы использования избыточности, когда для целей испытания можно отключить соответствующий механизм нейтрализации. Это позволяет объединять методы нейтрализации и диагностики [1, с. 7].

Такое объединение представляет собой как раз то, что необходимо для работы в условиях управления процессами в реальном времени и для обработки информации в оперативном режиме. В таких условиях очень важно предотвратить полный выход системы из строя в течение весьма значительной части ее срока службы. Это вызывает необходимость использования избыточности для обеспечения правильной работы системы при наличии неисправностей, а также обуславливает необходимость использования средств обнаружения и диагностики для обеспечения эффективного ремонта во время технического обслуживания.

Трудность диагностирования неисправностей в современных системах усугубляется тем, что используются интегральные микросхемы (чипы), в которых нельзя наблюдать сигналы на внутренних точках схемы. В то же время введение многочисленных контрольных выводов явно не рационально. Следовательно, необходимы такие тест-процедуры, при использовании которых, была бы возможность идентифицировать саму неисправность используя только нормальные входы и выходы схемы. Заметим, что такая интегральная схема вполне может содержать несколько сотен логических элементов (вентилей) [3]. Необходимо также, чтобы эти процедуры были бы в значительной мере автоматизированными. Поэтому процесс идентификации и диагностики систем связан с разработкой технических устройств и алгоритмов оценки и распознавания состояний объектов диагностирования с требуемой точностью детализации.

Последние годы процессы идентификации и диагностики объектов и явлений ознаменовались внедрением систем распознавания. Для данного направления это ха-

рактируется в сферах промышленного и сельского хозяйства.

На промышленных предприятиях методы распознавания нашли применение при построении систем технической диагностики технологического и оборудования, разработке «интеллектуальных» роботов, в автоматизированных системах управления предприятиями, в частности оперативного управления войсками и оружием. Тем не менее, основная задача идентификации и диагностики отождествляется с построением в том, или другом смысле

оптимальных алгоритмов распознавания, исследованием условий, обеспечивающих возможность построения таких алгоритмов [1, с. 8].

Поэтому при разработке любых устройств технической диагностики необходим системный подход, смысл которого в данном случае состоит в следующем. Система диагностирования должна строиться так, чтобы в условиях неизбежных ограничений результаты ее работы обеспечивали возможность системе управления реализовать потенциально достижимую эффективность.

Литература:

1. Ольшанский, В. В., Мартемьянов С. В. Идентификация и диагностика систем. Учебное пособие. Институт водного транспорта имени Г. Я. Седова — филиал «Государственный морской университет имени адмирала Ф. Ф. Ушакова». 2016.
2. Основные задачи контроля и диагностики ЭВМ Режим доступа: <https://studopedia.org/4-95564.html>
3. Техническая диагностика Режим доступа: https://ru.wikipedia.org/wiki/Техническая_диагностика

МЕТАЛЛУРГИЯ

Задачи и методы обеднения шлаков на ОАО «Алмалыкский ГМК»

Маткаримов Сохибжон Турдалиевич, старший преподаватель;

Сафаров Аскар Хайруллаевич, студент

Ташкентский государственный технический университет имени Ислама Каримова (Узбекистан)

Одной из главных задач на современном этапе развития Узбекистана является формирование системы мер по обеспечению устойчивого развития, базирующейся на разработке и внедрении новых и максимальном использовании имеющихся экологически безопасных энерго и ресурсосберегающих мало и безотходных технологий. Это относится и к технологиям переработки отходов.

Главными проблемами горной промышленности являются истощение минеральных ресурсов в доступной части планеты и загрязнение окружающей среды отходами горного и обогатительного производства. Так, если в 1913 г. на одного жителя Земли добывалось 5 т минерального сырья, то в 1980-х годах — более 30 т [1]. В горной отрасли это создает усиленную экологическую опасность и социальную напряженность в районах таких предприятий, самый высокий уровень выбросов вредных веществ в атмосферу и водоемы, образование твердых отходов. Внедрение относительно чистых технологий и создание безотходных производств может снизить экологическую напряженность. Отходы производства и потребления следует рассматривать не только как источник загрязнения окружающей среды и негативного воздействия на человека, но и как источник вторичных минеральных энергетических ресурсов. При этом ресурсосберегающая политика производителей в условиях рыночной экономики должна обеспечиваться государственным регулированием на основе экономических механизмов, причем так, чтобы предприятию было выгодно производить не только конечные продукты, но и обеспечивать благоприятную среду обитания. Конечной целью в данном случае следует считать оптимизацию производства одновременно по энерготехнологическим, экономическим и экологическим параметрам. Основным путем достижения этой цели являются разработка новых и усовершенствование существующих технологических процессов и производств. Одним из примеров такого подхода к организации безотходного производства является утилизация отходов тяжелой промышленности.

Медная промышленность является одной из ведущих подотраслей цветной металлургии. Выпуск меди постоянно растет, а технология ее получения непрерывно совершенствуется. Дальнейшее увеличение производства меди возможно за счет вовлечения в металлургиче-

скую переработку новых видов сырья, повышения комплексности его использования и интенсивного освоения новых прогрессивных технологических процессов и аппаратов.

На медеплавильном заводе ОАО «Алмалыкский ГМК» при проведении технологического процесса производства меди образуются следующие виды шлаков: конвертерный шлак первого периода, шлак кислородно-факельной плавки, шлак отражательной плавки, шлак отражательной плавки при модифицированном технологическом процессе. Все шлаковые отходы имеют отрицательное отношение к положительным действиям комбината. Это, в частности, будет отражаться на себестоимости и росте затрат [2].

Основной задачей комбината является комплексное извлечение ценных компонентов, а также обезвреживание этих отходов. Общая схема комплексной переработки шлаков в этом случае будет включать в себя неэлектротермическое предварительное обеднение шлаков, грануляцию и измельчение предварительно обезмеженных шлаков, флотационное глубокое обеднение измельченных шлаков, твердофазное восстановление шлаков, измельчение полученного клинкера и отделение железа магнитной сепарации, брикетирование магнитного концентрата и его переплавка с рафинированием от примесей для получения товарного железа или чугуна, переработка железосиликатного остатка с получением каменного литья, предварительное обезуглероживание клинкера цинкового производства, переработка клинкера цинкового производства в составе медной шихты с переводом его ценных компонентов в штейн и железосиликатных компонентов в шлак, предварительное неэлектротермическое обезмеживание шлаков перед флотацией [3].

Следует только отметить, что технологически вопросы отходов металлургии решаются стандартными методами, за исключением обезмеживания, которое при низком содержании меди можно и не проводить.

Рассмотренные выше примеры комплексного использования шлаков руд решают только часть проблем по утилизации отходов на комбинате. На АГМК существует ряд объектов, представляющих интерес для получения дополнительных металлов методом кучного выщелачивания: отвалы забалансовых сульфидных руд, окисленных и сме-

шанных руд, отвальные хвосты обогащения. Последние являются особым видом сырья для кучного выщелачивания. В отличие от руды, для вовлечения отвальных хвостов в процесс кучного выщелачивания не требуются операции дробления и окомкования по принятым технологиям. Бактериальные методы относятся к одному из современных направлений научно-технического прогресса в области переработки минерального сырья биотехнологии металлов, которая позволяет значительно повысить ком-

плексность его использования и обеспечить эффективную защиту окружающей среды [4].

Все вышеперечисленные задачи и методы могут дать экономический эффект в промышленной деятельности этого предприятия. Налаженное малоотходное технология производства упростит логистику и снизит затраты на сырьевые запасы. Важным считается то, что при таких процессах не залеживаются отходы, и оно не приходит в негодность.

Литература:

1. Трима, R. E. Molten Salts edited by Mamantov // New York, Marsel, 1969. P. 81–119
2. Юсупходжаев, А. А., Хасанов А. С. Некоторые вопросы переработки шлаков МПП АГМК // ЦНИИ цвет мет эконом. информации М. 1986. № 3 (173) 158–163 с.
3. Хасанов, А. С. Физическая свойства жидких шлаков и штейнов // Горный вестник Узбекистана, 2004. № 3/18 с. 84–93
4. Хасанов, А. С. Физическая химия медного производства. Навои. 2003. 176–180 с.

СТРОИТЕЛЬСТВО

Современное состояние жилищно-коммунального комплекса в РФ и странах ЕС

Наседкин Владислав Эдуардович, студент магистратуры
Орловский государственный университет имени И. С. Тургенева

Жилищно-коммунальное хозяйство, является одной из главных отраслей муниципальной экономики. От ее эффективности зависит качество населения в целом. Целью совершенствования управления жилищно-коммунальным хозяйством является создание условий для приведения жилищного фонда и коммунальной инфраструктуры в соответствие со стандартами качества, обеспечивающими комфортные условия проживания. Таким образом, актуальность исследования заключается в кризисном состоянии жилищно-коммунального комплекса, что способствует изучению тенденций развития отрасли жилищно-коммунального хозяйства и обоснование экономической целесообразности внедрения информационных технологий на предприятии.

Составление рынка ЖКХ в России — это одно из наиболее сложных направлений социально-экономического реформирования. На протяжении многих лет Жилищно-коммунальная сфера формировалась, как сложная и многоотраслевая система в условиях централизованного управления, в составе которой было порядка 30 видов различной деятельности.

К основным составляющим российского ЖКХ можно отнести содержание и эксплуатацию жилищного фонда, водоснабжение и водоотведение, тепло и электроснабжение, са-

нитарная очистка городов, дорожно-мостовое хозяйство, содержание и благоустройство территорий и т. д. В ЖКХ РФ сосредоточено примерно 25% основных фондов государства, лишь коммунальной энергетикой потребляется больше 20% электроэнергии и 45% тепловой энергии. И поэтому реформирование ЖКХ представляется весьма трудным, социально важными болезненным процессом, так как затрагивает все слои людей в стране. Наряду с этим, количество и масштаб существующих проблем в ЖКХ является значимым индикатором нынешнего состояния общества.

К сожалению, как и вся экономика страны, сфера ЖКХ значительно деградировала за последнее время. Для поддержания ее рабочего состояние, на сегодняшний момент, требуется все больше и больше средств. И если учесть тот факт, что в жилищно-коммунальном комплексе страны работает более 52000 предприятий находятся на грани банкротства, не составляет благоприятного исхода сложившейся ситуации.

Высокий уровень обеспеченности России энергоресурсами должен заметно упростить долю коммунальных платежей для отечественных потребителей. Если сравнить цены на электроэнергию для российский семей, то можно сделать вывод, что на сегодняшний день, этот показатель является самым низким из представленных.

Таблица 1

Внутренние цены на электроэнергию для населения, по данным за 2017 год

Место	Страна	Стоимость электроэнергии руб./кВт. ч.
1	Швеция	7,94
2	Италия	7,97
3	Нидерланды	6,88
4	Португалия	6,76
5	Франция	5,23
6	Германия	9,89
7	Испания	7,51
8	Чехия	5,85
9	Польша	5,61
10	Болгария	3,37
11	Великобритания	5,88
12	Латвия	4,25
13	Хорватия	4,68
14	Турция	5,57

15	Молдова	3,71
16	Румыния	4,27
17	Белоруссия	1,01
18	Россия	2,17

Создание удобных критерий проживания и качества предоставляемых коммунальных услуг основополагающей, как для человека, так и для общества в целом. Следовательно, возникает прямая потребность в реформировании системы ЖКХ, ее переход на абсолютно иную ступень взаимоотношений.

Анализ иностранного опыта к управлению ЖКХ, показывает, что коммунальные услуги тесно связаны с жильем и рассматриваются, как жизненно-важные. Следовательно, обеспечение населения коммунальными услугами всегда будет сопровождаться активным участием органов государства. У стран США, Англии, Франции и Германии возникли четыре общих подхода к регулированию и управлению ЖКХ муниципальных образований и сходятся они в следующем: регулирующие органы определяют правила работы частных предприятий-конкурентов, защищают права потребителей, гарантируют выполнение стандартов качества на жизненно важные услуги.

В России же, дотируемая отрасль и перекрестное субсидирование, является важным критерием, который торжествует применение накопленного за рубежом опыта. В

следствии, модель ЖКХ должна содержать основу для удаления данных недостатков из сферы управлению отраслью.

В настоящее время, рыночные отношения базируются на возможной передаче жилого фонда от одной организации к другой. При данной передаче стоит уделить важное внимание передачи информационной поддержке, необходимой для выполнения определенных функций.

На рисунке 1 приведены средние данные жилищно-коммунальных платежей домашних хозяйств России, Чехии и США. Такая выборка стран представляется наиболее удачной, и позволяет осуществить анализ сложившейся ситуации с жилищно-коммунальными платежами, как в государстве классического капитализма, так и постсоветских странах, находящихся на разных уровнях развития.

В странах Восточной Европы с переходной экономикой наиболее высокие доли расходов на ЖКУ у домашних хозяйств Эстонии (10,2%), Венгрии (10,8%), Польши (9,9%). В Австрии эта доля составляет порядка 9,2%, в Финляндии и Швеции — 3,4–3,7%.

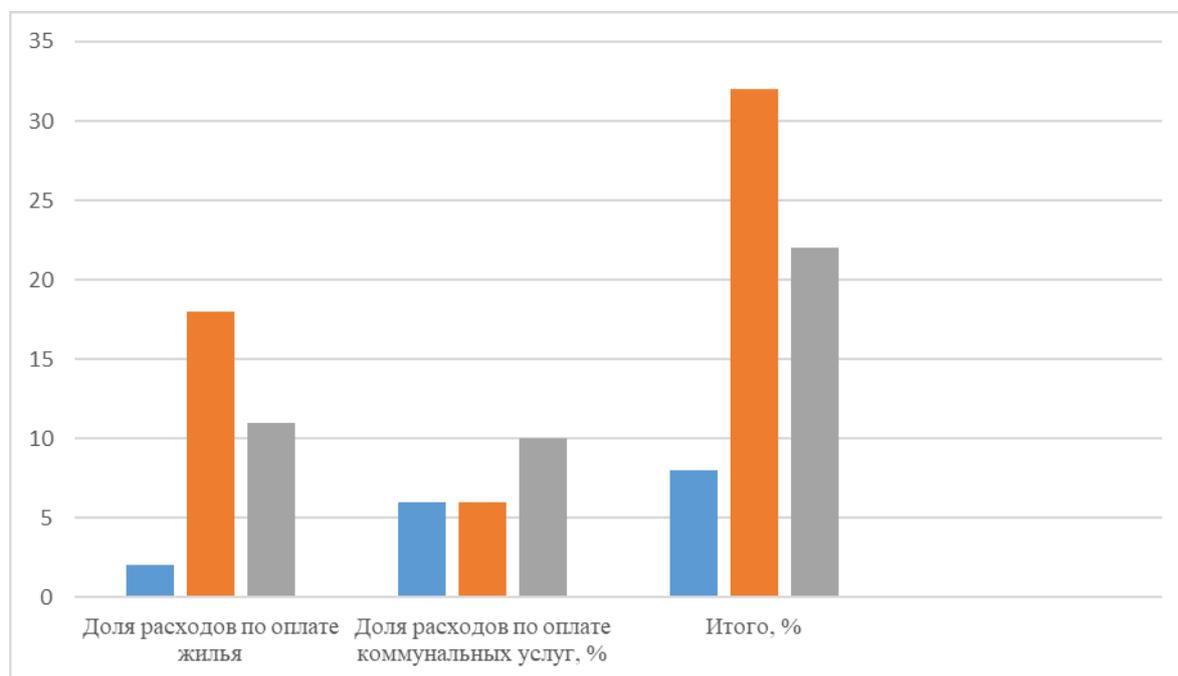


Рис. 1. Доля жилищно-коммунальных платежей в структуре потребительских расходов домашнего хозяйства за 2018 год. (Россия — голубой цвет, США — оранжевый, Чехия — серый)

Рисунок 1 наглядно иллюстрирует размер доли расходов по оплате коммунальных услуг в России и за рубежом. Так в России эта доля (6,9%) на сегодняшний день находится на уровне США, но значительно уступает Чехии (10,9%), не обеспеченной собственными минеральными

ресурсами. Однако жилищные расходы среднестатистической чешской (12,3%) и американской семьи (19,0%) на порядок превышают аналогичный показатель среднестатистического российского домашнего хозяйства (1,4%). Так же несоответствие соотношений доли расходов на

оплату жилья к доле расходов на коммунальные услуги в России (0,2) и США (2,75). По данному показателю США в 10 раз обгоняет Россию. Эти соотношения показывают национальные приоритеты в жилищных и коммунальных вопросах.

В соответствии с логикой американцев, на первом месте стоит качественное благоустроенное жилье как фундаментальный актив семьи и залог социальной стабильности, а коммунальные услуги являются всего лишь необходимым к нему приложением. В крайнем случае, жилье без коммунальных услуг пригодно к эксплуатации, а коммунальные услуги без их приложения теряют смысл.

Таким образом, на сегодняшний день развитие жилищно-коммунального хозяйства России значительно отстает от зарубежного развития. Долговечность электросетей на

сегодняшний день в 2–3 раза ниже, чем за рубежом. В то же время инвестиционная привлекательность отрасли значительно ниже по отношению к зарубежным странам. Повышение цен на электроэнергию в России без решения задачи повышения энерго-эффективности приведет к резкому снижению конкурентоспособности российской промышленности. При этом с учетом высокой степени износа основных фондов в российской промышленности, рост цен на энергию приведет скорее не к снижению энергоемкости производств, а к их закрытию.

Решение проблемы низкой энерго-эффективности требует от государства в первую очередь мер поддерживающего характера, которые бы, в частности, позволили предприятиям направлять дополнительные средства на модернизацию оборудования.

Литература:

1. Федеральный закон Российской Федерации от 23 ноября 2009 г. N 261-ФЗ «Об энергосбережении и о повышении энергетической эффективности и о внесении изменений в отдельные законодательные акты Российской Федерации». — СПС «Консультант плюс»
2. Зарукина, Е. В. Оценка уровня развития ЖКХ в России [по сравнению с развитыми зарубежными странами] // Муниципальная власть. 2003. — № 6.
3. Щербанин, Ю. Проблемы развития жилищно-коммунальной инфраструктуры России [Электронный ресурс] / Ю. Щербанин // Жилищно-коммунальная инфраструктура. — Режим доступа: <http://viktorvoksanaev.narod.ru/02012012.pdf>
4. Годовой отчет 2012 [Электронный ресурс]: Жилищная стратегия РФ на период до 2030 года. — Режим доступа: <http://ar2012.rzd.ru/development-strategy/transport-development-strategy/>
5. Жилищно-коммунальная инфраструктура России [Электронный ресурс] // Российский бизнес форум. — Режим доступа: <http://ros.biz/events/railways/>
6. Жилищно-коммунальная инфраструктура России [Электронный ресурс] // Транс регион. — Режим доступа: <http://trreg.ru/zheleznodorozhnaya-infrastruktura-v-rossii>

Научное издание

ТЕХНИЧЕСКИЕ НАУКИ В РОССИИ И ЗА РУБЕЖОМ

Выпускающий редактор Г. А. Кайнова
Ответственные редакторы Е. И. Осянина, О. А. Шульга
Оформление обложки Е. А. Шишков
Подготовка оригинал-макета П. Я. Бурьянов

Материалы публикуются в авторской редакции.

Подписано в печать 24.06.2019. Формат 60x84/8. Усл. печ. л. 4,18. Тираж 300 экз.

Издательство «Новация», 350911, г. Краснодар, ул. им. Фадеева, д. 429.

Отпечатано в типографии издательства «Молодой ученый», 420029, г. Казань, ул. Академика Кирпичникова, д. 25