

Федеральное государственное бюджетное
образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

На правах рукописи

Ганджа Тарас Викторович
РАЗВИТИЕ МЕТОДА КОМПОНЕНТНЫХ ЦЕПЕЙ
ДЛЯ РЕАЛИЗАЦИИ КОМПЛЕКСА ПРОГРАММ МОДЕЛИРОВАНИЯ
ХИМИКО-ТЕХНОЛОГИЧЕСКИХ СИСТЕМ

05.13.18 – Математическое моделирование,
численные методы и комплексы программ

Диссертация на соискание степени доктора технических наук

Научный консультант
д.т.н., профессор
Дмитриев Вячеслав Михайлович

2017

Оглавление

Список сокращений	9
Введение	11
Глава 1. Системный анализ и задачи компьютерного моделирования сложных технических управляемых систем	30
1.1 Задачи исследования и функционального проектирования сложных технических управляемых систем	30
1.2 Формализованное представление сложной технической управляемой системы.....	32
1.3 Анализ структуры и связей химико-технологических систем	36
1.4 Структура компьютерной модели СТУС для решения задач на основе вычислительного эксперимента	40
1.5 Постановка задач исследования и функционального проектирования СТУС на основе автоматизированного эксперимента	43
1.6 Обобщенный алгоритм автоматизированного эксперимента над сложными техническими управляемыми системами	46
1.7 Многоуровневая схема автоматизированного эксперимента на основе компьютерных моделей СТУС	53
1.8 Выводы	59
Глава 2. Многоуровневая интерпретация метода компонентных цепей для моделирования сложных технических управляемых систем	61
2.1 Метод компонентных цепей	61
2.1.1 Назначение метода компонентных цепей	61
2.1.2 Основные понятия метода компонентных цепей	63
2.1.3 Методика построения компонентной цепи технических объектов с энергетическими и информационными связями	66
2.1.4 Алгоритм автоматического построения модели компонентной цепи	72
2.1.5 Алгоритм вычислительного эксперимента	75
2.2 Модель сложной технической управляемой системы в формате метода компонентных цепей	79

2.2.1 Компьютерная модель сложной технической управляемой системы	79
2.2.2 Вычислительная модель СТУС	81
2.2.3 Принцип разделения уравнений в модели СТУС	83
2.3 Метод многоуровневого компьютерного моделирования сложных технических управляемых систем	84
2.3.1 Многоуровневая компонентная цепь СТУС	84
2.3.2 Методика формирования многоуровневой компьютерной модели СТУС ...	85
2.3.3 Основы языка многоуровневого компьютерного моделирования	90
2.4 Графические нотации языка моделирования управляемых технологических объектов	93
2.4.1 Компоненты-источники	93
2.4.2 Компоненты-преобразователи	96
2.4.3 Компоненты-измерители	98
2.5 Графические нотации языка моделирования алгоритмических конструкций	100
2.5.1 Алгоритмические компонентные цепи	100
2.5.2 Правила формирования алгоритмических компонентов	101
2.6 Классификация визуальных компонентов языка виртуальных инструментов и приборов	104
2.7 Выводы	110
Глава 3. Компьютерное моделирование процессов функционирования химико-технологических систем	113
3.1 Формализованное представления объектов с неоднородными векторными связями в языке моделирования ХТС	114
3.1.1 Структура неоднородной векторной связи	114
3.1.2 Правила коммутации компонентов с НВС	117
3.2 Обобщенная модель физико-химического компонента	119
3.3 Автоматическое построение и решение вычислительной модели компонентной цепи химико-технологической системы	127
3.4 Набор компонентов химико-технологических систем.....	130
3.4.1 Источник смеси веществ	130

3.4.2 Измерительные компоненты характеристик многокомпонентных вещественных потоков	131
3.4.3 Комплексное гидротермодинамическое сопротивление многокомпонентному вещественному потоку	134
3.4.4 Смеситель	137
3.4.5 Теплообменник	138
3.4.6 Сепаратор	141
3.4.6 Абсорбер	145
3.5 Выводы	149
Глава 4. Язык моделирования алгоритмических конструкций	151
4.1 Грамматика языка моделирования алгоритмических конструкций	151
4.1.1 Операторы языка МАК	155
4.1.2 Операнды языка МАК	156
4.1.3 Правила формирования выражения языка МАК	158
4.2 Правила построения дерева вывода синтаксических конструкций язык МАК ..	161
4.3 Формализованное отображение дерева вывода математико-алгоритмического выражения в алгоритмическую компонентную цепь	165
4.4 Отображение скалярных математических конструкций языка МАК в формат алгоритмических КЦ	166
4.4.1 Типы данных. Источники констант и определенных переменных	166
4.4.2 Отображение операторов языка МАК	169
4.4.3 Отображение функций языка МАК	171
4.5 Отображение векторно-матричных конструкций языка МАК	176
4.6 Отображения математико-алгоритмических конструкций языка МАК в формат алгоритмических компонентных цепей	179
4.7 Компоненты для представления алгоритмов решения задач многовариантного анализа и параметрической оптимизации	185
4.8 Автоматическая параметризация компьютерной модели СТУС с помощью баз данных	190
4.9 Принципы использования атрибутивных связей компонентов	192

4.10 Выводы	194
Глава 5. Язык виртуальных инструментов и приборов	196
5.1 Многоуровневая структура виртуального прибора	196
5.2 Формализованное представление виртуального прибора	198
5.3 Модели виртуальных приборов с атрибутивными связями	200
5.4 Последовательность действий пользователя при формировании виртуального прибора	201
5.5 Формирование функциональных блоков виртуальных инструментов и приборов	208
5.6. Погрешности виртуальных приборов	212
5.7 Выводы	219
Глава 6. Комплекс программ автоматизации экспериментов над сложными техническими управляемыми системами	221
6.1 Назначение и структура комплекса программ	221
6.2 Опции многослойного редактора	224
6.3 Интерпретатор языка моделирования ХТС	231
6.3.1 Структура входных и выходных данных интерпретатора языка моделирования ХТС	231
6.3.2 Алгоритм формирования компонентной цепи исследуемого объекта с неоднородными векторными связями	234
6.4 Универсальное вычислительное ядро	237
6.4.1 Объектно-ориентированная структура универсального вычислительного ядра	239
6.4.2 Методики повышения быстродействия вычислительного эксперимента	249
6.4.2.1 Методика и алгоритм повышения быстродействия анализа функционирования химико-технологических систем	251
6.4.2.2 Методика формирования системы уравнений с учетом их класса	259
6.4.2.3 Методика анализа КЦ ХТС с постоянной обратной матрицей	262
6.5. Интерпретатор математико-алгоритмических конструкций	264

6.6. Имитационное ядро	268
6.6.1 Структура передаваемого сообщения	268
6.6.2 Алгоритм передачи сообщений	270
6.6.3 Структура алгоритмических компонентов	272
6.7 Библиотека моделей компонентов	274
6.7.1 Методы взаимодействия компонента с универсальным вычислительным ядром	275
6.7.2 Набор численных методов расчета параметров и коэффициентов уравнений компонентов химико-технологических систем	278
6.7.3 Методы взаимодействия компонента с имитационным ядром	280
6.7.4 Методы взаимодействия пользователя с визуальными компонентами	281
6.7.5 Интерфейс компонентов с переменным числом связей	283
6.8 Генератор моделей компонентов	284
6.9 Интерактивные панели	285
6.9.1 Интерактивная схемная панель	285
6.9.2 Интерактивная математическая панель	286
6.10 Выводы.....	292
Глава 7. Построение многоуровневых компьютерных моделей сложных технических управляемых систем для проведения вычислительных экспериментов ...	296
7.1 Алгоритм решения задач исследования и функционального проектирования СТУС	296
7.2 Разработка и отладка алгоритмов управления на компьютерных моделях химико-технологических систем	300
7.3 Решение задачи проектирования теплообменного аппарата	304
7.4 Определение точки росы газа путем проведения многовариантного анализа в многоуровневой компьютерной модели	306
7.5 Построение компьютерной модели управления режимами функционирования химико-технологической системы.....	310
7.5.1 Схема коррекции параметров компонентной цепи химико-технологической системы	310

7.5.2 Многоуровневая компонентная цепь минимизации расхода ингибитора при абсорбционной осушке природного газа	314
7.6 Построение сетевых компьютерных тренажеров для обучения и переподготовки операторов-технологов	317
7.7 Выводы.....	321
Заключение	323
Список использованных источников	327
Приложение 1. Набор моделей визуальных компонентов	351
Приложение 2. Пример исходного кода модели визуального компонента	356
Приложение 3. Принципы работы пользователя с компонентами «График».....	359
Приложение 4. Принципы использование компонента «Таблица»	362
Приложение 5. Структура базы данных свойств веществ	364
Приложение 6. Реализация численного метода решения уравнения Соава-Редлиха-Квонга	365
Приложение 7. Набор компонентов математических операций в языке моделирования алгоритмических конструкций	369
Приложение 8. Реализация компонента <i>SCComponent</i>	376
Приложение 9. Программно-алгоритмическая реализация компонента «Абсорбер»	380
Приложение 10. Программно-алгоритмическая реализация компонента «Асборбер», адаптированная к методике формирования системы уравнений с учетом их класса	388
Приложение 11. Программная реализация измерителей потенциальных и потоковых переменных	395
Приложение 12. Программная реализация компонента-преобразователя данных	401
Приложение 13. Программная реализация копонента с методами обработки действий пользователя	404
Приложение 14. Шаблоны файлов и методов для генератора моделей компонентов	406
Приложение 15. Руководство пользователя генератором моделей компонентов	415

Приложение 16. Свидетельство СибРУМЦ	426
Приложение 17. Акты внедрения.....	428
Приложение 18. Свидетельства о государственной регистрации программ для ЭВМ	447

СПИСОК СОКРАЩЕНИЙ

БД	база данных
БОД	блоки обработки данных
БМК	библиотека моделей компонентов
ВИП	виртуальные инструменты и приборы
ВМ	вычислительная модель
ВП	виртуальный прибор
ВЭ	вычислительный эксперимент
ГКБ	глобальный координатный базис
ГМК	генератор моделей компонентов
ДСМ	дискретно-событийное моделирование
ИзмУ	измерительное устройство
ИМ	имитационная модель
ИМАП	интерактивная математико-алгоритмическая панель
ИМП	интерактивная математическая панель
ИСП	интерактивная схемная панель
ИспУ	исполнительное устройство
КМ	компьютерная модель
КЦ	компонентная цепь
ЛАРМ	лабораторное автоматизированное рабочее место
ЛВС	локальная вычислительная сеть
ЛКБ	локальный координатный базис
ЛПФ	логарифмоподобная функция
МАК	моделирование алгоритмических конструкций
МК	модель компонента
МКЦ	многоуровневая компонентная цепь
ММ	математическая модель
ММК	математическая модель компонента
НВС	неоднородная векторная связь
ООП	объектно-ориентированное программирование

СТУС	сложные технические управляемые системы
УКПГ	установка комплексной подготовки газа
УУ	устройства управления
ФБ	функциональный блок
ФВ	физическая величина
ФКБ	физический координатный базис
ХТС	химико-технологические системы

Введение

Актуальность. При добыче, переработке и использовании природного газа или его отдельных компонентов (веществ) находят широкое применение сложные технические управляемые системы (СТУС). Входящие в них химико-технологические системы (ХТС) допускают декомпозицию на взаимосвязанные элементы, между которыми под управлением устройств управления (УУ) протекают физически неоднородные, далее мультифизические, энергетические и многокомпонентные вещественные потоки. Принцип действия УУ включает в себя получение, математическую (численную) обработку и визуализацию данных измерения наблюдаемых переменных ХТС и выработку на их основе управляющих воздействий на исполнительные устройства. В отличие от обычных алгоритмов алгоритм функционирования УУ, называемый далее *сценарием управления*, представляет собой несколько параллельных (возможно взаимосвязанных) цепочек обработки поступающих данных и получения управляющих воздействий. Он может быть реализован в виде программы на некотором низкоуровневом языке программирования и зашит в контроллер. В этом случае управление будет считаться *автоматическим*. В случае *автоматизированного управления* результаты измерения передаются в компьютер, где с привлечением оператора осуществляется их обработка с целью выработки управляющих воздействий, удовлетворяющих различным критериям. При этом способе управления могут быть задействованы различные средства интеллектуализации: база данных (БД), экспертные системы с базами знаний, нейронные сети и т.п. Процессы, протекающие в ХТС зачастую чувствительны к изменениям параметров окружающей среды и подвержены её возмущающим воздействиям.

Как правило, ХТС представляет собой совокупность элементов и аппаратов химической технологии. В каждом из них наблюдаются изменения физических свойств (давления и температуры), а также фазовые переходы и химические преобразования некоторых веществ, называемых далее *компонентами потока*, каждая из которых характеризуется своей концентрацией.

Для автоматизации решения задач научно-исследовательского, проектно-конструкторского и технологического характера, а также для проведения учебных экспериментов, целесообразным является использование компьютерных моделей СТУС, адекватно описывающих протекающие в реальной системе процессы. Наличие энергетических обратных связей, зависимость коэффициентов уравнений математической модели (ММ) каждой отдельной ФХС от гидравлических, термодинамических характеристик, концентраций и параметров компонент потока усложняют процессы реализации компьютерных моделей ХТС и проведения вычислительных экспериментов (ВЭ). Помимо них КМ СТУС должны включать функциональные модели, адекватно описывающие сценарии проведения всех этапов вычислительного эксперимента и работу УУ, а также развитые панели визуализации результатов и интерактивного изменения значений параметров модели. Наибольший эффект при использовании таких компьютерных моделей (КМ) может быть достигнут при их интеграции с реальной ХТС и её устройством управления, представленным измерительно-управляющим контроллером. Это позволит поддерживать модели в адекватном реальной системе состоянии и при необходимости применять к нему результаты ВЭ.

С применением таких КМ СТУС на основе автоматизированного эксперимента решению подлежат следующие задачи:

- формирования структуры, выбора элементов ХТС и определения значений их параметров, при которых наблюдаются требуемые режимы его функционирования;
- выбора исполнительных устройств, оказывающих непосредственные энергетические воздействия на ХТС, пропорциональные информационным сигналам УУ и предназначенные для перевода системы из текущего в желаемое состояние;
- формирования и отладки сценариев управления, лежащих в основе работы УУ исследуемой СТУС;
- определения значений управляющих воздействий, позволяющих устанавливать в ХТС режимы, удовлетворяющие заданным требованиям;

– разработки и отладки сценариев функционирования SCADA-систем, обеспечивающих интеллектуальное управление ХТС с включением компьютерных моделей объекта в контур его управления для автоматизированного решения задач определения, установления и поддержания требуемых режимов функционирования;

– разработки на основе КМ СТУС компьютерных учебных программ (интерактивных учебников, компьютерных задачник, тренажеров и виртуальных лабораторий) по различным техническим дисциплинам кибернетического направления и сетевых компьютерных тренажеров операторов-технологов, деятельность которых направлена на управление протекающими в ХТС процессами с целью поддержания требуемых режимов функционирования.

Созданная для решения одной из перечисленных задач модель СТУС может быть использована для решения остальных задач исследования данной системы.

Для решения обозначенного круга задач в настоящее время применяются различные комплексы программ автоматизации вычислительного эксперимента с реализованными в них моделями и методами компьютерного моделирования. Зачастую реализующие их комплексы программ предназначены для моделирования объектов конкретной физической природы, не допускают интеграции друг с другом, а их механическое соединение не приносит желаемого эффекта с точки зрения автоматизации решения перечисленных задач.

Таким образом, **актуальность проводимого исследования** обусловлена необходимостью разработки нового подхода к автоматизированному решению задач исследования и функционального проектирования сложных технических управляемых систем газовой промышленности, многоуровневой интерпретации метода компонентных цепей и реализации её на основе комплекса программ моделирования химико-технологических систем. Для эффективного решения задач многоуровневая интерпретация метода компонентных цепей должна позволять формировать и анализировать многоуровневые компьютерные модели СТУС (КМ СТУС), в которых математические модели ХТС объединены с имитационными

моделями автоматизированных экспериментов и устройств управления, а также с визуальными моделями панелей визуализации и управления.

Степень разработанности проблемы. Вопросы развития фундаментальных основ методологии и технологии вычислительного эксперимента рассматривались в работах Г.И. Марчука [1], А.А. Самарского [2–4], П.В. Петрова и В.А. Целищева [5, 6]. В настоящее время разработаны различные методы и основанные на них комплексы программ для моделирования механических [7, 8], гидравлических [9, 10], тепловых [11, 12], электрических и электромеханических объектов [13, 14], а также систем автоматического управления [15]. Помимо этого, разработаны комплексы алгоритмов и программ компьютерного моделирования химико-технологических систем [16], позволяющие определять и исследовать режимы работы элементов и аппаратов химической технологии. Однако построенные в них компьютерные модели не допускают включения в них моделей исполнительных и измерительных устройств. Также в них отсутствует возможность построения функциональных моделей устройств управления для целей исследования сценариев его работы, направленных на выработку требуемых режимов функционирования ХТС.

Одним из универсальных методов компьютерного моделирования и автоматизации ВЭ над мультифизическими объектами является метод компонентных цепей (КЦ) [17, 18]. Предложенный В.М. Дмитриевым и Е.А. Арайсом, в настоящее время он адаптирован к моделированию технических устройств и систем в работах Т.Н. Зайченко [19], А.В. Шутенкова и др. На его базе разработана среда компьютерного моделирования МАРС [20], в которой реализованы виртуальные лаборатории учебного и научно-исследовательского назначения [21, 22, 23]. Помимо этого метод КЦ и основанные на нем программно-инструментальные средства были адаптированы к моделированию и управлению химико-технологическими и эколого-экономическими системами в работах Ю.А. Журавского и Э.В. Балакиной [24], И.Я. Клепака [25] и О.С. Затик [26], а также к моделированию и построению виртуальных инструментов и приборов в работе Т.Ю. Коротинной [27] при непосредственном участии автора диссертации.

Цель исследования: Развитие метода компонентных цепей для автоматизации решения задач исследования и функционального проектирования сложных технических управляемых систем и реализация на его основе комплекса программ моделирования химико-технологических систем.

Объект исследования: химико-технологические системы с многокомпонентными вещественными потоками в связях, взаимодействующие с устройством управления посредством измерительных и исполнительных устройств.

Предмет исследования: методы, численные алгоритмы и комплекс программ моделирования химико-технологических систем.

Задачи исследования:

1. Формирования формализованного представления сложных технических управляемых систем и математическая постановка задач их исследования на базе нового подхода к автоматизированному решению задач путём проведения вычислительного эксперимента над компьютерными моделями СТУС, включающими математические модели химико-технологических систем, имитационные (функциональные) модели устройств управления и визуальные модели панелей визуализации и управления экспериментом.

2. Построения многоуровневой компонентной цепи СТУС из трех взаимосвязанных уровней – объектного, логического и визуального, и разработки графического языка многоуровневых компонентных цепей, включающего три подязыка: язык химико-технологических систем, язык моделирования алгоритмических конструкций и язык виртуальных инструментов и приборов, и реализации их программно-алгоритмического аппарата.

3. Формирования графических нотаций языка моделирования химико-технологических систем, предназначенного для формирования их моделей на объектном уровне многоуровневой компонентной цепи. Разработки алгоритмов автоматизированного формирования и численного анализа вычислительных моделей ХТС с неоднородными векторными связями, обеспечивающими протекание между их компонентами информационных, мультифизических энергетических и многокомпонентных вещественных потоков, со встроенными в эти алгоритмы

численными методами расчета характеристик многокомпонентных вещественных потоков.

4. Разработки структуры универсального вычислительного ядра и реализации в нем новых численных методов явно-неявного анализа, основанных на непосредственном определении значений единственной неопределенной переменной на этапе формирования уравнений для целей обеспечения повышения его быстродействия при анализе вычислительных моделей с неоднородными векторными потоками.

5. Разработки нотаций языка моделирования алгоритмических конструкций для построения имитационных алгоритмов проведения автоматизированных экспериментов с возможностями численной обработки результатов анализа моделей ХТС и интерактивного изменения значений параметров модели. Формирование системы отображения лексем языка моделирования алгоритмических конструкций в формат алгоритмических компонентных цепей.

6. Построения и исследования языка виртуальных инструментов и приборов, представленного набором визуальных компонентов для формирования на визуальном уровне многоуровневой компонентной цепи СТЭС лицевых панелей приборов, к классу которых относятся панели визуализации результатов и интерактивного управления вычислительным экспериментом.

7. Формирования и реализации многоуровневой структуры библиотеки моделей компонентов, обеспечивающей хранение и использование компонентов трех подязыков языка многоуровневого компьютерного моделирования, а также создания программно-алгоритмических средств – генератора моделей компонентов, предназначенного для формирования программно-алгоритмического кода компонентов, и интерактивных панелей, оснащенных редактором математико-алгоритмических конструкций для автоматизации исследования новых моделей компонентов.

8. Разработки комплекса программ «Среда моделирования химико-технологических систем», основанного на развитии метода компонентных цепей и предназначенного для автоматизации вычислительных экспериментов над

СТУС, а также его апробация на задачах научно-исследовательского, производственного и учебного характера.

Методологическая, теоретическая и эмпирическая база исследования.

Методологической основой исследований является метод компонентных цепей, предложенный профессором В.М. Дмитриевым и Е.А. Арайсом. При решении поставленных задач, направленных на его развитие и многоуровневую интерпретацию для целей моделирования химико-технологических систем, применялись методы системного анализа, теории управления, математического, имитационного и визуального моделирования, теории графов. При построении компьютерных моделей ХТС применялись методы системного анализа объектов и процессов химической технологии, предложенные В.В. Кафаровым. Реализация новых численных методов явно-неявного анализа компьютерных моделей химико-технологических систем основана на алгоритмах неявного анализа, реализованных в рамках вычислительного ядра системы MAPS. Практическая реализация комплекса программ выполнялась с применением методов структурного и объектно-ориентированного программирования, унифицированного языка моделирования UML, языка программирования Visual C++.

Научная новизна диссертации заключается в том, что:

В области математического моделирования:

– предложен и обоснован новый подход к автоматизированному решению задач исследования и функционального проектирования сложных технических управляемых систем, заключающийся в многократном проведении вычислительного эксперимента над компьютерными моделями ХТС при варьировании значений параметров их компонентов согласно сценарию решения задачи, формируемому пользователем в графическом виде и связанному с графической панелью визуализации и интерактивного управления параметрами исследуемой модели (п. 5 паспорта специальности – комплексные исследования научных и технических проблем с применением современной технологии математического моделирования и вычислительного эксперимента);

– осуществлено развитие метода компонентных цепей, открывшее возможности формирования и анализа многоуровневых компонентных цепей химико-технологических систем, которые помимо математических моделей аппаратов химической технологии с неоднородными векторными связями содержат имитационные модели сценариев проведения вычислительных экспериментов, включающие функциональные модели устройств управления, и визуальные модели панелей визуализации и интерактивного управления. (п. 1 паспорта специальности – разработка новых математических методов моделирования объектов и явлений).

– сформирован новый язык виртуальных инструментов и приборов, представленный набором визуальных компонентов с едиными программно-алгоритмическими принципами их функционирования и обеспечивающий формирование и моделирование виртуальных приборов, обладающих визуальными панелями отображения результатов и интерактивного управления параметрами модели. В отличие от моделей, построенных в других системах, нелинейные модели виртуальных приборов не входят в общую систему уравнений исследуемой ХТС, что позволяет включать в них блоки численно-алгоритмического анализа результатов моделирования с целью их подготовки к визуализации. За счет такого разделения модели увеличивается точность расчета значений первичных переменных модели ХТС и значительно расширяются возможности численно-алгоритмической обработки результатов моделирования (соответствует п. 8 паспорта специальности – разработка систем компьютерного и имитационного моделирования).

В области численных методов:

– Предложен схемотехнический язык моделирования химико-технологических систем и разработаны программно-алгоритмические средства в виде интерпретатора языка ХТС и универсального вычислительного ядра. В основу их работы положены новые алгоритмы формирования вычислительной модели ХТС с неоднородными векторными связями и новый численный метод ее явно-неявного анализа, основанный на непосредственном определении значений единственной неопределенной переменной, входящей в уравнение, на этапе их формирования. Повышение быстродействия алгоритма обеспечивается снижением общего числа

элементарных операций, выполняемых в ходе формирования и решения неполной системы уравнений, а повышение его точности основано на отсутствии необходимости задания начальных условий (п. 4 паспорта специальности – разработка эффективных численных методов и алгоритмов в виде программно-ориентированных программ для проведения вычислительных экспериментов).

В области комплексов программ:

– сформированы лексемы и грамматические правила языка моделирования алгоритмических конструкций, отличительным признаком которого является возможность построения функциональных моделей устройств управления и имитационных моделей сценариев проведения экспериментов, включающих этапы первичной и циклической параметризации моделей ХТС, получения и численной обработки результатов их анализа. Предложена и реализована система отображения лексем языка моделирования алгоритмических конструкций в формат алгоритмических компонентных цепей, взаимосвязанных с моделью ХТС, и реализован алгоритм передачи сообщений с данными различных типов между компонентами логического уровня многоуровневой компонентной цепи СТУС (п. 7 паспорта специальности – разработка новых математических методов и алгоритмов интерпретации натурального эксперимента на основе его математической модели)

– сформирована структура и разработан комплекс программ «Среда моделирования химико-технологических систем», основанный на развитии метода компонентных цепей. В отличие от существующих комплексов программ он включает в себя построенные на единых принципах метода компонентных цепей и объектно-ориентированного программирования средства математического, имитационного и визуального моделирования, обеспечивающие решение задач исследования и функционального проектирования сложных технических управляемых систем предприятий газовой промышленности (п. 4 паспорта специальности – реализация эффективных численных методов и алгоритмов в виде комплекса проблемно-ориентированных программ для проведения вычислительного эксперимента).

– предложена и спроектирована новая многоуровневая структура библиотеки моделей компонентов. Основанная на принципах объектно-ориентированного

программирования, она позволяет хранить и использовать в многоуровневых компонентных цепях компоненты трех подязыков языка многоуровневого компьютерного моделирования. Для автоматического формирования и исследования новых моделей компонентов реализованы интерактивные панели, содержащие в своем составе редактор математико-алгоритмических конструкций для ввода модели, и «Генератор моделей компонентов», обеспечивающий автоматизацию формирования программного кода моделей с целью их добавления в библиотеку моделей компонентов (п. 3 паспорта специальности – разработка, обоснование и тестирование эффективных вычислительных методов с применением современных компьютерных технологий).

Теоретическая значимость. Теоретическая значимость диссертации заключается в том, что в ней решена крупная народнохозяйственная задача создания нового подхода к автоматизированному решению задач исследования и функционального проектирования сложных технических управляемых систем газовой промышленности. Ее решение достигнуто путем теоретического развития метода компонентных цепей и реализации на его основе комплекса программ моделирования химико-технологических систем. На базе предложенной структуры многоуровневой компонентной цепи предложены и разработаны современные компьютерные учебные программы и сетевые компьютерные тренажеры операторов-технологов газовой промышленности.

Теоретическая значимость результатов диссертации также состоит в развитии методов компьютерного моделирования сложных технических управляемых систем, обеспечивающих одновременное моделирование непрерывных процессов, протекающих в ХТС, и дискретно-событийных процессов, описывающих сценарии проведения экспериментов и наблюдаемых в устройствах управления. Предложенные в диссертации принципы и методики построения многоуровневых компонентных цепей ХТС, а также реализованные графические языки их формирования и программно-алгоритмический аппарат их анализа направлены на развитие общей теории моделирования. Разработанные новые численные методы явно- неявного решения систем уравнений обеспечивают повышение эффективности и

быстродействия многовариантного анализа и параметрической оптимизации моделей химико-технологических систем.

Практическая значимость

1. Предложенная многоуровневая компонентная цепь СТус позволяет автоматизировать процессы исследования и функционального проектирования химико-технологических систем и устройств управления ими.

2. В разработанном комплексе программ, интегрированном с реальным технологическим объектом, может быть построена интеллектуальная система управления, в которой компьютерная модель ХТС используется для анализа поведения объекта и выработки управляющих воздействий, позволяющих устанавливать и поддерживать желаемые режимы функционирования.

3. В комплексе программ «Среда моделирования химико-технологических систем» в виде многоуровневой компонентной цепи была построена компьютерная модели стерилизатора для обработки наполненных растворами ампул, инструмента и одежды, используемых в федеральном государственном унитарном предприятии «Научно-производственное объединение по медицинским иммунобиологическим препаратам «Микроген» министерства здравоохранения Российской Федерации в филиале ФГУП «НПО «Микроген» Минздрава России в г. Томск «НПО «Вирион».

4. Результаты диссертационного исследования были использованы при разработке ЗАО НПФ «Сибнефтекарт» программного обеспечения автоматизированных систем управления автозаправочными станциями и нефтебазами в части анализа процессов хранения, реализации и учета нефтепродуктов; разработки компонентов языка моделирования, построения и отладки алгоритмов управления технологическими объектами; визуализации результатов вычислительных экспериментов с элементами интерактивного управления параметрами.

5. Результаты диссертационного исследования были использованы ОАО «ТомскНИПИНефть» при разработке сценариев функционирования автономных компьютерных тренажеров операторов в рамках исполнения договора В042316/0760Д на выполнение проектных работ по объекту «Обустройство Кую-

мбинского месторождения. Интегрированная автоматизированная система управления и безопасности (Фаза 1.1)».

6. На базе комплекса программ «Среда автоматизации экспериментов над сложными техническими управляемыми системами» разработаны учебно-иллюстративные модели, компьютерные тренажеры, виртуальные и реально-виртуальные лаборатории по различным техническим дисциплинам, внедрённые в учебный процесс Томского государственного университета систем управления и радиоэлектроники, Оренбургского государственного университета и национально-исследовательского Томского политехнического университета.

7. В системе виртуальных инструментов и приборов, входящей в состав разработанного комплекса программ, реализованы клиентские части имитационной динамической модели «Виртуальный промысел», для использования которых реализовано приложение «MARS-Engine», предназначенное для использования подготовленных многоуровневых компонентных цепей отдельно от «Среды моделирования химико-технологических систем». Имитационная динамическая модель «Виртуальный промысел», реализованная совместно с научными сотрудниками кафедры химической технологии топлива и химической кибернетики Национального исследовательского Томского политехнического университета, внедрена в процесс обучения и переподготовке специалистов ОАО «Томскгазпром».

Научные результаты, выносимы на защиту

1. Новый подход к автоматизированному решению задач исследования и функционального проектирования сложных технических управляемых систем позволяет автоматизировать решение задач их исследования и функционального проектирования путём многократного вычислительного эксперимента над многоуровневыми компонентными цепями с возможностями интерактивного варьирования значений параметров компонентов химико-технологических систем, численно-алгоритмической обработкой и визуализацией результатов их анализа.

2. Развитие метода компонентных цепей обеспечивает графическое построение и автоматизированное функционирование многоуровневых компонентных цепей СТЭС, представленных на трех взаимосвязанных уровнях с примене-

нием трех схмотехнических языков построения моделей: языка управляемых технологических объектов, языка моделирования алгоритмических конструкций и языка виртуальных инструментов и приборов. Структура многоуровневой компонентной цепи сложных технических управляемых систем допускает проведение многовариантного вычислительного эксперимента над СТУС с применением математических моделей ХТС, имитационных моделей сценариев проведения экспериментов и устройств управления со встроенными в них блоками численной обработки результатов и расчета коэффициентов моделей ХТС, а также визуальных моделей панелей визуализации и интерактивного управления.

3. Схмотехнический язык химико-технологических систем обеспечивает построение компонентных цепей ХТС с неоднородными векторными связями, а его программно-алгоритмический аппарат в виде интерпретатора языка ХТС и адаптированного универсального вычислительного ядра позволяет анализировать процессы преобразования мультифизических энергетических и многокомпонентных вещественных потоков, протекающие в объектах газовой промышленности.

4. Новый численный метод явно-неявного анализа вычислительных моделей ХТС с неоднородными векторными связями с заложенной в нем возможностью непосредственного определения значений переменных на этапе формирования системы алгебро-дифференциальных уравнений и реализованный на его основе алгоритм анализа обеспечивают повышение точности и быстродействия анализа компьютерных моделей химико-технологических систем на 25-50 % по сравнению с неявными методами анализа. Это достигается за счет определения значений ряда переменных на этапе формирования вычислительной модели ХТС и их использования для вычисления значений коэффициентов и свободных членов формируемых уравнений.

5. Язык моделирования алгоритмических конструкций и его программно-алгоритмические средства обеспечивают построение и работу имитационных моделей алгоритмов автоматизированных экспериментов, включающих блоки параметризации и численной обработки результатов анализа компьютерных моде-

лей ХТС, а также содержащих функциональные модели устройств управления протекающими в них процессами.

6. Язык виртуальных инструментов и приборов предназначен для построения лицевых панелей виртуальных инструментов и приборов и представлен совокупностью визуальных компонентов, осуществляющих визуализацию результатов измерения и моделирования и варьирование значений параметров исследуемых компьютерных моделей.

7. Сформированная новая структура библиотеки моделей компонентов обеспечивает загрузку, выбор и работу компонентов трех взаимосвязанных подязыков языка многоуровневых компонентных цепей, а также включает в себя средства исследования и формирования новых моделей компонентов, представленные генератором моделей компонентов и совокупностью интерактивных панелей.

8. Структура комплекса программ «Среда моделирования химико-технологических систем», основанного на развитии метода компонентных цепей для моделирования сложных технических управляемых систем предприятий газовой промышленности, и совокупность методик и практических примеров его применения обеспечивают автоматизацию решения задач производственного, научно-исследовательского и учебного характера.

Достоверность результатов.

Исследования адекватности построенных моделей подтверждают достоверность полученных результатов. Это было установлено: путем сравнения результатов с результатами, полученными в известных аналогичных программных комплексах, таких как Aspen Hysys, ChemCAD и прочие. Разработанные компьютерные модели в виде компонентных цепей химико-технологических систем являются непротиворечивыми, так как реализованы на основе известных уравнений, описывающих протекающие в них процессы. Численные методы сравнивались с результатами теоретических расчетов, а также с результатами, полученными в аналогичных комплексах программ. Достоверность разработанного программного комплекса подтверждается непротиворечивостью использования основных поло-

жений объектно-ориентированного программирования, а также полученными актами о его использовании для решения различных задач народнохозяйственного значения, в том числе и для автоматизации процесса обучения и переподготовки специалистов газотранспортной промышленности. Возможность практического использования подтверждается внедрением комплекса программ в учебный процесс Национального исследовательского Томского политехнического университета, Томского государственного университета систем управления и радиоэлектроники, Оренбургского государственного университета, а также реализацией сетевого тренажера, внедренного в ОАО «Востокгазпром».

Апробация результатов. Результаты докладывались и представлялись в материалах конференций:

1. Электронные средства и системы управления. Международная научно-практическая конференция (г. Томск, 2004, 2015);
2. Современная техника и технологии. Международная научно-практическая конференция (г. Томск, 2001, 2012, 2013);
3. Природные и интеллектуальные ресурсы Сибири. Международная научно-практическая конференция (г. Абакан, 2010, г. Новосибирск, 2012, г. Новокузнецк, 2013, г. Томск, 2015, 2016);
4. Методы и алгоритмы прикладной математики в технике, медицине и экономике. Международная научно-практическая конференция (г. Новочеркасск, 2013 г.);
5. Управление большими системами. Всероссийская школа-конференция молодых ученых (г. Уфа, 2013);
6. Радиотехнические и информационные системы и устройства. Региональная научно-техническая конференция (г. Томск, 2000);
7. Моделирование: Теория, методы и средства. Международная научно-практическая конференция (г. Новочеркасск, 2010, 2011);
8. Современное образование: актуальные проблемы профессиональной подготовки и партнерства с работодателями. Международная научно-методическая конференция (г. Томск, 2014, 2015);

9. Научная сессия ТУСУР. Международная (всероссийская до 2010 года) научно-техническая конференция студентов, аспирантов и молодых ученых (г. Томск, 2007, 2011, 2015);

10. Перспективы развития информационных технологий. Международная научно-практическая конференция (г. Новосибирск, 2014);

11. Информатизация процессов формирования открытых систем на основе СУБД, САПР и систем искусственного интеллекта. Международная научно-техническая конференция (г. Вологда, 2011);

12. Научная дискуссия: вопросы технических наук. Международная заочная научно-практическая конференция (г. Москва, 2013);

13. Химия и химическая технология XXI века. Международная научно-практическая конференция (г. Томск, 2008, 2014, 2017).

Публикации. По теме диссертации опубликовано 58 работ, в том числе: 5 монографий, 19 статей в рецензируемых научных журналах, рекомендованных ВАК; 4 статьи в журналах, индексируемых в системе SCOPUS, 5 свидетельств о регистрации программ для ЭВМ; 19 тезисов докладов на Российских и международных конференциях.

Использование результатов исследований.

1. НИР «Метод компонентных цепей для компьютерного моделирования и управления эколого-экономическими системами», грант РФФИ 10-06-00084, 2010-2012 годы.

2. НИР «Метод многоуровневого моделирования алгоритмов управления технологическими процессами в сложных системах», грант РФФИ 11-07-00384, 2011-2013 г.

3. НИР «Многоуровневые компьютерные модели для интеллектуального управления эколого-экономическими системами предприятий нефтегазового комплекса», грант РФФИ 13-07-00378, 2013-2015 г.

4. Хозяйственный договор № 18/914-С1 от 16.04.2007 г. «Создание автоматизированной системы управления технологическими процессами системы об-

ратного снабжения метанолом УКПГ Мыльджинского газоконденсатного месторождения ООО «Томскгазпром».

5. Хозяйственный договор № 18-9/07 Фр. от 12.09.2007 г. «Разработка и изготовление стендов для учебно-научных лабораторий».

Структура и объем диссертации. В состав диссертации входит введение, 7 глав, заключение, списка литературы из 212 наим., приложения на 107 страницах. Объем диссертации без приложений – 350 с., в т.ч. 166 рисунков, 6 таблиц.

Личный вклад. Результаты диссертационной работы, соответствующие поставленным задачам и сформулированные в положениях, выносимых на защиту, получены автором лично. Вклад автора состоит в развитии метода компонентных цепей и его адаптации к компьютерному моделированию сложных технических управляемых систем предприятий газовой промышленности, непосредственном участии в разработке моделей химико-технологических систем и программно-алгоритмического аппарата подязыков языка многоуровневых компонентных цепей, а также в обработке и интерпретации данных численного моделирования, подготовке основных публикаций по диссертационной работе. Формирование теоретических основ развития метода компонентных цепей проводилось совместно с научным консультантом В.М. Дмитриевым и Т.Н. Зайченко. Программная реализация комплекса программ «Среда моделирования химико-технологических систем» выполнялась совместно с А.Н. Кураколовым, М.А. Ерошкиным, С.А. Пановым, В.В. Ганджой. Построение и реализация компьютерных моделей химико-технологических систем осуществлялось совместно с И.М. Долгановым и М.О. Писаревым. Некоторые результаты исследований получены совместно с соавторами опубликованных работ.

Краткое содержание работы. В первой главе проводится системный анализ сложных технических управляемых систем с целью постановки задач компьютерного моделирования и последующим построением их компьютерной модели. Сформулированы задачи исследования и функционального проектирования СТУС, для решения которых целесообразно применение компьютерных моделей. Она

содержит алгоритм вычислительного эксперимента, лежащего в основе решения таких задач.

Вторая глава содержит теоретическое обоснование и основные положения многоуровневой интерпретации метода компонентных цепей, а также сформирована и обоснована структура многоуровневой компонентной цепи. Для её формирования предложен язык, состоящий из трех подязыков – языка управляемых химико-технологических систем, языка моделирования алгоритмических конструкций и языка виртуальных инструментов и приборов. Для каждого из них сформированы основные классы компонентов.

В третьей главе рассмотрены вопросы компьютерного моделирования химико-технологических систем газовой промышленности. Для построения и анализа их компонентных цепей предложено формализованное представление компонентов с неоднородными векторными связями, на основе которого разработана обобщенная модель химико-технологического компонента. Она является базовой для разработки набора компонентов химико-технологических систем, к которым относятся: источник смеси веществ, совокупность измерителей, осуществляющих извлечение из модели значений переменных, а также модели таких аппаратов ХТС как теплообменник, сепаратор и абсорбер.

Четвертая глава посвящена построению и реализации языка моделирования алгоритмических конструкций (МАК), предназначенного для построения сценариев проведения вычислительного эксперимента и решения на его основе задач исследования и функционального проектирования СТЭС. Выражения языка формируются в виде алгоритмических компонентных цепей на логическом уровне многоуровневых компонентных цепей, которые включают функциональные модели устройств управления. Набор компонентов данного языка включает в себя компоненты основных скалярных и векторно-матричных операций, а также компоненты для представления алгоритмов многовариантного анализа и параметрической оптимизации.

В пятой главе приводятся основные нотации языка виртуальных инструментов и приборов, представляющие собой визуальные компоненты для форми-

рования виртуальных приборов и панелей визуализации и интерактивного управления на визуальном уровне многоуровневых компонентных цепей. Каждый из них помимо визуальных компонентов, включает в себя компоненты языка МАК, реализующие его основную функциональность. Глава содержит описание многоуровневой структуры виртуального прибора (ВП), его формализованное представление в формат многоуровневых компонентных цепей, а также алгоритм его формирования. Приводится описание методики формирования функциональных блоков ВП, а также исследование погрешности виртуального прибора.

Шестая глава включает в себя вопросы построения и реализации комплекса программ моделирования химико-технологических систем, описание алгоритмов функционирования её основных модулей, а также описание основных интерфейсных классов, позволяющих модулям обмениваться информацией. При описании объектно-ориентированной структуры универсального вычислительного ядра основное внимание уделено методикам и алгоритмам повышения его быстродействия при анализе моделей ХТС с неоднородными векторными связями. Помимо этого рассмотрены алгоритмы построения и функционирования алгоритмических компонентных цепей, между компонентами которых передаются данные различных типов.

Седьмая глава содержит описания алгоритма решения задач исследования и функционального проектирования ХТС с использованием многоуровневых компонентных цепей, а также перечень построенных многоуровневых компьютерных моделей, позволяющих их решение. Также она включает в себя описание принципов реализации сетевого компьютерного тренажера, разработанного на базе распределенной многоуровневой компьютерной модели.

Глава 1. Системный анализ и задачи компьютерного моделирования сложных технических управляемых систем

Рассмотрены сложные технические управляемые системы газовой промышленности, предназначенные для преобразования физических свойств и компонентного состава многокомпонентных вещественных потоков и использования их свойств. В данной главе сформулированы задачи исследования и функционального проектирования сложных технических управляемых систем, для решения которых предлагается использовать многоуровневые компонентные цепи.

1.1 Задачи исследования и функционального проектирования сложных технических управляемых систем

Технологические объекты газовой промышленности относятся к классу химико-технологических систем (ХТС), допускающих декомпозицию на взаимосвязанные элементы и аппараты химической технологии [37]. Между ними циркулируют *многокомпонентные вещественные потоки*, изменение физических свойств и компонентных составов которых происходит с использованием или выделением энергии различной физической природы. Протекающие в них процессы подвержены сильному влиянию параметров и характеристик окружающей среды (возмущающие воздействия).

Для управления протекающими в ХТС процессами применяются комплексы технических и программно-алгоритмических средств, осуществляющие сбор и первичную обработку результатов измерений наблюдаемых характеристик ХТС, а также позволяющие применять при необходимости управляющие воздействия. Такой комплекс совместно с управляемой им ХТС образуют *сложную техническую управляемую систему*. Входящее в СТУС устройство управления может функционировать в автоматическом или автоматизированном режимах. В случае автоматического режима оно реализуется на базе универсального измерительно-управляющего контроллера, в котором на специализированном языке реализуется управляющая программа, основанная на сценарии управления и направленная на определение величин управляющих воздействий. Система, образованная ХТС

совместно с автоматическим устройством управления, называется системой автоматического управления.

Когда в процессе функционирования ХТС управляющие решения принимаются оператором, такая система является автоматизированной системой управления. В настоящее время для частичной или полной автоматизации процесса принятия управленческих решений находят применения различные средства интеллектуализации. К ним относятся экспертные системы [28] с наборами продукционных моделей знаний, средства автоматизации математических вычислений [29, 30] и компьютерного моделирования [31, 32].

При разработке и функционировании СТУС их компьютерные модели могут быть применены для автоматизации решения следующих важных научно-технических задач:

- 1) экспериментального определения требуемых режимов функционирования при формировании технического задания на разрабатываемую ХТС;
- 2) исследования и функционального проектирования ХТС с определением его структуры и значений параметров, удовлетворяющих требованиям технического задания;
- 3) формирования и отладки сценариев функционирования УУ, работающих в автоматическом режиме;
- 4) разработки виртуальных лабораторий [21], компьютерных тренажеров и имитаторов работы ХТС [33, 34], используемых в процессе обучения и переподготовки специалистов в области их исследования, функционального проектирования и управления химико-технологическими системами;
- 5) моделирования и разработки интеллектуальных систем управления ХТС [35], в которых принятие управленческих решений, направленных на установление и поддержание безопасных и эффективных режимов функционирования, осуществляется с применением компьютерных (математических и имитационных) моделей объекта и системы управления.

Автоматизация решения перечисленных задач может быть достигнута путем развития универсального метода компьютерного моделирования физически

неоднородных технических объектов и его адаптации к компьютерному моделированию ХТС. Реализация на его основе комплекса программ моделирования химико-технологических систем откроет возможности автоматизации вычислительного эксперимента (ВЭ) над математическими моделями (ММ) ХТС.

Применяемый для этого метод компьютерного моделирования должен позволять объединять в единой компьютерной модели СТУС структурно-функциональные модели ХТС, функциональные модели сценариев работы УУ и визуальные модели средств отображения результатов моделирования и интерактивного управления параметрами модели. Разработка на его основе нового комплекса программ «Среда моделирования химико-технологических систем» позволит автоматизировать задачи исследования и функционального проектирования ХТС на базе многовариантного анализа их моделей при варьировании параметров. При его интеграции с реальными объектами могут быть разработаны интеллектуальные системы управления технологическими объектами газовой промышленности, в которых в процессе принятия управленческих решений задействована компьютерная модель ХТС.

1.2 Формализованное представление сложной технической управляемой системы

Сложная техническая управляемая система включает в себя управляемую химико-технологическую систему, в которой протекают мультифизические энергетические и многокомпонентные вещественные потоки, и взаимосвязанное с ней устройство управления, функционирование которого основано на преобразовании информационных потоков и сигналов.

Для адекватного описания процессов, протекающих в ХТС и УУ, а также для адекватного проведения вычислительных экспериментов используемая в них модель СТУС должна соответствовать реальной системе управления. Кроме того, используемая в ней КМ ХТС, с помощью которой автоматизируется вычислительный эксперимент, будет применяться как для проектирования сценариев функционирования УУ, так и для решения задач научно-исследовательского и проектно-конструкторского характера, а также для реализации программных мо-

дулей обучения и переподготовки управляющего персонала промышленных предприятий.



Рисунок 1.1 – Структурно-функциональная схема сложной технической (технологической) управляемой системы

Представленная на рисунке 1.1 структурно-функциональная схема СТУС, полученная на основе ее формализованного представления [36], допускает декомпозицию на взаимосвязанные элементы. ХТС может быть декомпозирована на *элементы и аппараты химической промышленности*, каждому из которых ставится в соответствие *компонент структуры*, реализующий его математическую модель (ММ). Работа УУ может быть описана сценарием информационных преобразований данных измерения наблюдаемых характеристик ХТС (или результатов моделирования ее ММ) в управляющие воздействия на реальные исполнительные устройства (ИспУ) или их модели. Он может быть декомпозирован на *отдельные компоненты сценария*, каждый из которых описывает определенное действие с информационными потоками данных, получаемыми с измерительных устройств (ИзмУ).

Структурно-функциональная схема СТУС, представленная на рисунке 1.1, включает в себя декомпозированную на отдельные элементы структуры химико-технологическую систему и взаимосвязанное с ней УУ, декомпозированное на отдельные элементарные операции преобразования результатов измерения $y = [y_E, y_S]$ в управляющие воздействия $u = [u_E, u_S]$. Компьютерная модель СТУС, построенная на основе представленной на рисунке 1.1 ее структурно-функциональной схемы, содержит следующие классы компьютерных моделей элементов и функциональных блоков:

1) *класс моделей исполнительных устройств*, оказывающих непосредственные энергетические и вещественные воздействия на ХТС $U = [U_E, U_S]$, пропорциональные информационным сигналам УУ. К их числу относятся:

– *источники энергии*, представляющие собой компоненты, каждый из которых позволяет описать в модели подведение к определенным элементам ХТС энергии различной физической природы U_E , пропорциональной управляющим воздействиям УУ u_E ;

– *источники вещества* – это компоненты, описывающие в модели ХТС подачу к его элементам многокомпонентных смесей веществ, каждое вещество которых называется *компонентой потока*, характеризующихся вектором характеристик U_S и предназначенных для преобразования и использования их физических свойств и компонентного состава с использованием или генерацией энергии различной физической природы. Управления их процессами осуществляется УУ с помощью информационных сигналов u_S ;

2) *класс моделей элементов ХТС*, описывающих протекающие в нем процессы преобразования входящих в него энергетических и (или) вещественных потоков U в соответствующие выходные потоки $Y = [Y_E, Y_S]$. Процессы, протекающие в этих элементах, подвержены возмущающему воздействию $F = [F_E, F_S]$ со стороны окружающей среды. К числу элементов ХТС, для которых должны быть построены соответствующие модели, относятся:

– *преобразователи энергии*, отражающие в модели ХТС протекание процессов накопления входных энергетических потоков U_E , их преобразование в выходные потоки энергии Y_E , в том числе отличной от входного потока природы. Так, например, электрическая энергия может быть преобразована в механическую, гидравлическую или тепловую. В моделях преобразователей энергии должны учитываться потери, наблюдаемые при изменении ее физического вида или ее передаче на значительные расстояния. Кроме того, необходим учет энергии при помощи расходомеров, счетчиков тепловой и электрической энергии. Характеристики протекающих в преобразователях энергии процессов зависят от значений параметров и характеристик окружающей среды F_E . Выходная энергия может подаваться к различным преобразователям веществ E_S и быть направленной на изменение скорости протекания процессов преобразования веществ, а также быть выделенной из них;

– *преобразователи вещества (многокомпонентных вещественных потоков)*, выражающие изменение физических характеристик и компонентных составов входных вещественных потоков U_S с целью получения совокупности выходных потоков многокомпонентных вещественных потоков Y_S . Описываемые ими процессы подлежат управлению подводимой к ним энергии E_S . Характеристики процессы, направленных на преобразование многокомпонентных вещественных потоков, зависят от возмущающих воздействий окружающей среды F_S , которыми являются изменение атмосферного давления, температуры воздуха, концентраций различных веществ в объектах окружающей среды и т.п.;

3) *класс моделей измерительных устройств*, осуществляющих выборку значений соответствующих переменных модели ХТС и их передачу в модель УУ, реализующую алгоритмы его работы. К числу моделей ИзМУ, называемых далее *измерительными компонентами*, относятся:

– *измерители энергии*, измеряющие в определенные моменты времени и передающие УУ временные характеристики потенциальных и потоковых перемен-

ных, характеризующих энергетический поток определенной физической природы Y_E , проходящий по связи между некоторыми компонентами ХТС;

– измерители характеристик веществ (вещественных потоков), осуществляющие измерение характеристик многокомпонентных вещественных потоков Y_S и их передачу УУ. Первичными характеристиками вещественных потоков являются мгновенные значения потенциальных и потоковых переменных, характеризующих запасенную энергию потока, и характеристики его компонентного состава, представленные концентрациями входящих в него веществ;

4) класс моделей объектов окружающей среды, оказывающих на ХТС возмущающие воздействия $F = [F_E, F_S]$;

5) устройство управления, представленное совокупностью сценариев управления, которые лежат в основе его работы и направлены на выработку управляющих воздействий на ИспУ. Оно включает в себя панель визуализации результатов экспериментов и средства интерактивного управления параметрами модели ХТС и управляющих воздействий на реальные ИспУ.

Для целей построения компьютерных моделей ХТС, входящих в состав КМ СТУС, с использованием подхода системного анализа, предложенного В.В. Кафаровым [111], рассмотрена структура и характер связей управляемых технических объектов, допускающих декомпозицию на взаимосвязанные элементы и аппараты, между которыми протекают мультифизические энергетические и многокомпонентные вещественные потоки.

1.3 Анализ структуры и связей химико-технологических систем

В химико-технологических системах [37] предприятий газовой промышленности под управлением операторов-технологов, взаимодействующих с АСУ ТП и SCADA-системами, протекают физические преобразования и химические превращения многокомпонентных вещественных потоков. Изменению подлежат гидравлические и термодинамические потоков, а также преобразование их компонентных составов. К их числу относятся фазовые переходы соответствующих компонентов (веществ) из одной фазы в другую (выпаривание, конденсирование,

абсорбция, десорбция, ректификация) [38, 39], а также разнообразные химические реакции с получением новых веществ, называемых далее *компонентами потока*. ХТС представляют собой совокупность элементов и аппаратов с протекающими в них процессами взаимодействия нескольких многокомпонентных потоков. Для получения желаемого эффекта данные потоки со скоростью G должны подаваться в соответствующий аппарат под давлением P и с определенной температурой T . В обобщенном виде ХТС, обладающая исполнительными устройствами изменения давления и температуры, может быть представлена виде, приведенном на рисунке 1.2. Она содержит следующие элементы:

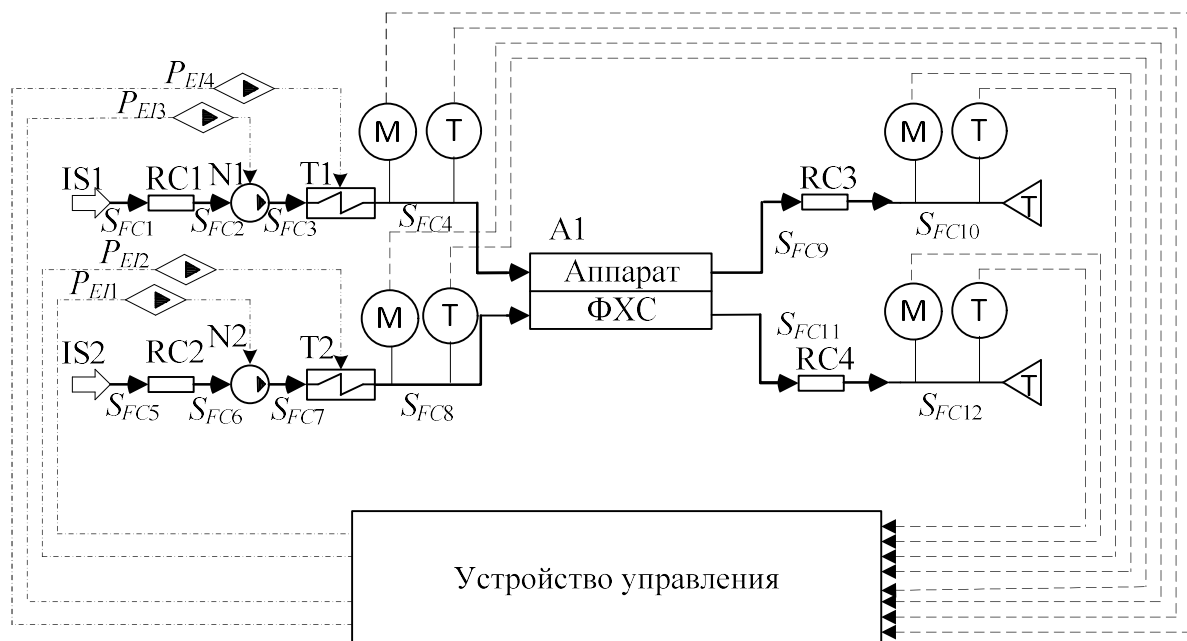


Рисунок 1.2 – Структурно-функциональная схема обобщенной химико-технологической системы

- IS1, IS2 – управляемые источники многокомпонентных смесей веществ, в которых регулируется скорость потока вещества G , выраженная в моль/с;
- RC1, RC2, RC3, RC4 – трубы, оказывающие гидравлическое и термодинамическое сопротивление протекающему по ним многокомпонентному вещественному потоку;
- N1, N2 – управляемые насосы, осуществляющие увеличение давления входящего в ХТС потока на определенную величину ΔP ;
- T1, T2 – управляемые нагреватели, повышающие температуру подаваемой в ХТС многокомпонентной смеси веществ на величину ΔT .

Наблюдение за процессами, протекающими в рассматриваемой ХТС, осуществляется манометрами М и термометрами Т, измеряющими давление и температуру смеси соответственно в точках соединения элементов, к которым они подключены. Преобразование результатов измерений в информационные сигналы, передаваемые в УУ, осуществляют энергоинформационные преобразователи P_{EI1}, \dots, P_{EI6} , которые являются датчиками соответствующих величин [40].

Управление процессами, протекающими в каждом аппарате ХТС, осуществляется с помощью управляемых насосов N1, N2, осуществляющих изменение давления смеси на входе аппарата, а также управляемыми нагревателями Т1, Т2, функционирование которых направлено на изменение температур входящих в аппарат вещественных потоков. Преобразование информационных сигналов УУ в энергетические воздействия на управляемые насосы и нагреватели осуществляется информационно-энергетическими преобразователями P_{IE1}, \dots, P_{IE4} .

Связи компонентов ХТС S_{FC1}, \dots, S_{FC12} , являясь физико-химическими, описывают энергетические характеристики и концентрации веществ, входящих в описываемый связью многокомпонентный поток. Каждая физико-химическая связь $S_{FCi}, i = \overline{1, 12}$ можно представить в виде

$$S_{FCi} = \{S_{Gi} \cup S_{Ti} \cup S_{Ci}\} \rightarrow \{\{P_i, G_i\} \cup \{T_i, Q_i\} \cup \{C_i\}\}, \quad (1.1)$$

где $S_{Gi} \rightarrow \{P_i, G_i\}$ – гидравлическая связь, в которой P_i – давление, а G_i – объемный расход вещества; $S_{Ti} \rightarrow \{T_i, Q_i\}$ – термодинамическая связь, состоящая из T_i – температуры и Q_i – теплового потока; $S_{Ci} \rightarrow \{C_i\}$ – однородная векторная связь, содержащая концентрации всех компонентов (веществ), составляющих данный поток.

Проблемы построения и исследования моделей ХТС освещены в работах В.В. Кафарова, В.П. Мешалкина, И.Н. Дорохова [37, 41, 42]. Средствами компьютерного моделирования ФХС и ХТС в настоящее время являются комплексы программ *HYSYS* [43], *ChemCAD* [44, 45] и другие. Вопросы применения этих систем в научно-исследовательской деятельности и учебном процессе рассмотрены в ра-

ботах Т.Н. Гартмана [46]. Проблемами разработки компьютерных моделей ХТС для управления технологическими процессами занимаются ученые национально-исследовательского Томского политехнического университета под руководством Иванчиной Э.Д. и Ивашкиной Е.Н. [47–51].

В виде химико-технологических систем могут быть представлены управляемые технологические объекты газовой промышленности, в том числе и технологическое оборудование газодобывающих промыслов Российской Федерации [52], предназначенных для сбора, подготовки и транспорта нефти, газа и воды. Их деятельность должна осуществляться в соответствии с требованиями, установленными действующими стандартами в области системы менеджмента качества, промышленной и экологической безопасности [53, 54].

Как правило, для управления протекающими в ХТС процессами используются АСУ, функциональное проектирование и интеллектуализация процессов управления [55] которых являются актуальными задачами. Масштабность объектов и сложность протекающих в них процессов в совокупности с требованиями поддержания соответствующих режимов технологических процессов не позволяют производить подготовку и повышение квалификации персонала предприятий различных отраслей с использованием непосредственного функционирующего оборудования. Для решения обозначенных задач могут быть применены компьютерные модели, разработанные в рамках универсального метода компьютерного моделирования. Такой метод позволяет в единой многоуровневой компьютерной модели объединять непрерывные процессы преобразования мультифизических энергетических и многокомпонентных вещественных потоков ХТС с дискретно-событийными информационными процессами УУ, направленными на определение управляющих воздействий на исполнительные устройства, позволяющие устанавливать в объекте управления необходимый режим функционирования. Для разработки нового или развития существующего метода и реализации на его основе программно-алгоритмического аппарата компьютерного моделирования ХТС с протекающими в них энергетическими и многокомпонентными вещественными потоками проведен системный анализ управляемых ими технологиче-

ских объектов. На его основе представлено формализованное представление СТУС, позволяющее сформировать его компьютерную модель, предназначенную для решения задач исследовательского, проектно-конструкторского и учебного назначения на основе автоматизированного эксперимента.

1.4 Структура компьютерной модели СТУС для решения задач на основе вычислительного эксперимента

Для эффективного решения задач исследования и функционального проектирования СТУС, в основе которых лежит вычислительный эксперимент, может быть применена компьютерная модель СТУС, включающая в себя модель ХТС и интегрированная с реальным объектом. Ее структурная схема представлена на рисунке 1.3 и включает в себя:

– *реальную химико-технологическую систему*, на которой должны быть апробированы результаты эксперимента, полученные с помощью её КМ. Непосредственные воздействия на ХТС осуществляются *исполнительными устройствами*, преобразующими информационные сигналы модели УУ \widetilde{u}_o в пропорциональные им управляющие энергетические воздействия \widetilde{U}_o на компоненты ХТС. *Измерительные устройства* выполняют мониторинг наблюдаемых характеристик ХТС \widetilde{y}_o . На реальную химико-технологическую систему оказывают влияние объекты окружающей среды в виде возмущающих воздействия \widetilde{F} , которые зачастую носят стохастический характер;

– *компьютерная модель ХТС*, состоящая из моделей исполнительных устройств K_{IS} , непосредственной модели ХТС K_o и моделей измерительных устройств K_{IZ} . Она формируется относительно первичных параметров и переменных входящих в неё элементов. Структурно подобная реальной ХТС, её КМ позволяет эффективно получать требуемые результаты анализа в статическом и динамическом режимах с возможностью варьирования значений определенных параметров ее компонентов и моделей и источников возмущающих воздействий K_F ;

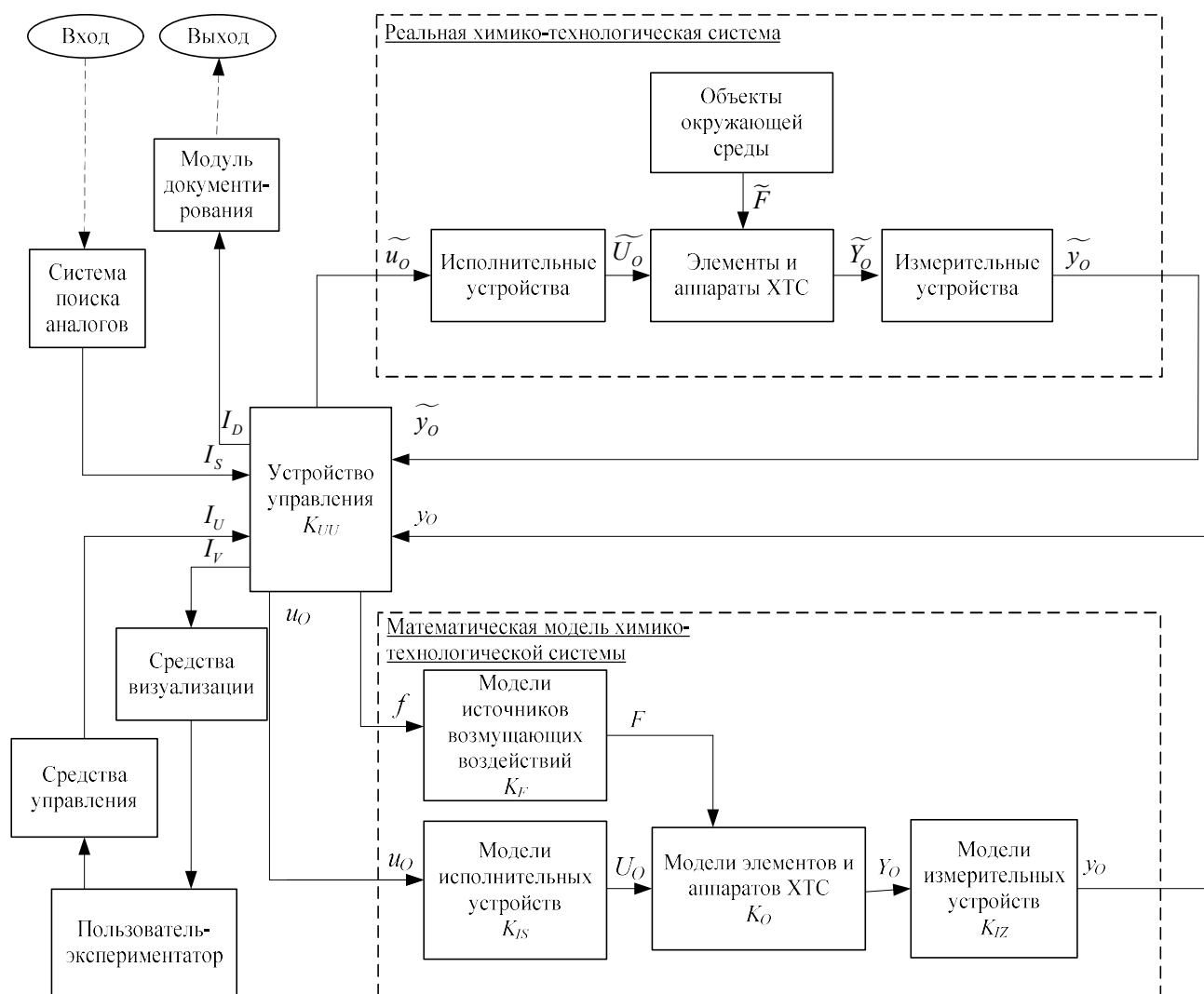


Рисунок 1.3 – Структура компьютерной модели сложной технической управляемой системы

– *устройство управления*, осуществляющее сбор и анализ результатов моделирования и данных измерения, а также передачу необходимых значений на средства визуализации, прием и обработку задающих воздействий, вырабатываемых оператором с помощью средств управления, и вычисления на их основе управляющих воздействий на реальную ХТС и на её КМ. Взаимодействия с библиотекой прототипов-аналогов организуется на уровне передачи необходимой информации через модуль документирования и получения необходимых прототипов и их значений;

– *система поиска аналогов* [56], осуществляющая поиск в библиотеке прототипов-аналогов ранее выполненных проектов, представляющих собой сами компьютерные модели с подробным описанием, сформированным автоматически с помощью модуля документирования;

– *модуль документирования* [57], предназначенный для автоматизированной подготовки текущей и отчетной информации о решенных задачах исследования, проектирования и функционирования СТУС для ее передачи в библиотеку прототипов-аналогов с целью хранения и использования системой поиска аналогов;

– *библиотека прототипов-аналогов*, представляющая собой серверное приложение, к которому подключается комплекс программ «Среда моделирования химико-технологических систем» для помещения выполненных проектов в архив и поиска прототипов среди ранее выполненных проектов;

– *средства визуализации*, представляющие собой программно-реализованные модули отображения на экране монитора в понятном виде данных изменения характеристик реальной ХТС, результатов анализа её КМ и решения задач исследования, проектирования и функционирования СТУС;

– *средства управления* – визуальные компоненты, предназначенные для воздействия на ХТС и её КМ непосредственно с экрана монитора.

Совокупность средств визуализации и средств управления образуют *панели визуализации и управления*, с которыми взаимодействует оператор-исследователь (далее: *оператор*) во время проведения эксперимента. Его задачами является наблюдение за протекающими в КМ ХТС и реальной системе процессами, определение, установление и поддержание требуемых режимов её функционирования. Для эффективной работы оператора должны быть заранее реализованы необходимые модели для их исследования в ходе выполнения эксперимента. Они должны содержать в своем составе целевые функции, предназначенные для повышения адекватности моделей и решения задач исследования, проектирования и управления, а также необходимые продукционные модели знаний экспертной системы.

На базе предложенной структуры компьютерной модели СТУС может быть реализована интеллектуальная система управления техническим (технологическим) объектом [58, 59] и интеллектуальных SCADA-систем [60] с последующим ее внедрением в производство.

1.5 Постановка задач исследования и функционального проектирования СТус на основе автоматизированного эксперимента

При исследовании и функционального проектирования СТус на основе автоматизированного эксперимента, реализуемого с помощью компьютерных моделей ХТС и УУ, решению подлежат следующие задачи:

1. Задача идентификации моделей химико-технологической системы и подключенных к ней моделей исполнительных и измерительных устройств:

1.1. Задача формирования структуры модели ХТС и определения значений параметров её элементов \mathbf{P}_o , при которых модель адекватно описывает процессы, протекающие в реальной системе. При этом минимизации подлежит функция

$$\varepsilon_1(\mathbf{P}_o) = \sum_{i=1}^{N_o} (y_i(\mathbf{P}_o) - \tilde{y}_i)^2 \rightarrow 0, \quad (1.2)$$

где y_i – значение текущей наблюдаемой переменной модели ХТС; \tilde{y}_i – значение наблюдаемой характеристики, измеренное на функционирующем реальной системы; N_o – количество наблюдаемых значений ХТС.

При достижении целевой функцией (1.2) нулевого значения, модель будет адекватно описывать процессы, протекающие в реальной системе.

1.2. Формирование моделей исполнительных устройств, оказывающих непосредственные воздействия на ХТС для её перевода из текущего в желаемое состояние, и определение их параметров \mathbf{P}_{IS} , удовлетворяющих показателям точности и быстродействия. Решение данной задачи осуществляется путем минимизации целевой функции

$$\varepsilon_2(\mathbf{P}_{IS}) = \sum_{i=1}^{N_{IS}} (Y_i(\mathbf{P}_{IS}) - \tilde{Y}_i)^2 \rightarrow \min, \quad (1.3)$$

где $Y_i(\mathbf{P}_{IS})$ – значение текущих показателей точности и быстродействия управления, которые зависят от значений параметров ИспУ; \tilde{Y}_i – желаемые значения показателей точности и быстродействия, приведенные в техническом задании на проектируемую СТус, в которую помимо ХТС входят ИспУ и ИзмУ; N_{IS} – коли-

чество исполнительных устройств, задействованных в исследовании, выполняемом с помощью целевой функции (1.3).

1.3. Формирование моделей измерительных устройств, осуществляющих измерения значений наблюдаемых переменных ХТС с передачей их значений устройству управления, и определение их параметров \mathbf{P}_{IZ} осуществляется на основе минимизации разности показателей точности и быстродействия между реальными ИзМУ и их компьютерными моделями при варьировании значений параметров последних. При этом минимизации подлежит целевая функция вида

$$\varepsilon_3(\mathbf{P}_{IZ}) = \sum_{i=1}^{N_{IZ}} (Y_i(\mathbf{P}_{IZ}) - \tilde{Y}_i) \rightarrow \min, \quad (1.4)$$

где $Y_i(\mathbf{P}_{IZ})$ – совокупность показателей точности и быстродействия ИзМУ, влияющих на качество работы УУ; \tilde{Y}_i – желаемые показатели точности и быстродействия ИзМУ, которые приведены в техническом задании на формируемую СТУС; N_{IZ} – количество измеряемых устройств в рассматриваемой системе.

2. Задача анализа протекающих в ХТС процессов при изменении значений параметров компонентов с определением диапазона изменений наблюдаемых переменных, для выбора которых из общего решения модели и передачи на панели визуализации и управления применяются измерительные приборы. Математически она может быть описана как

$$\mathbf{y}(t) = \mathbf{y}(t, \mathbf{P}_{IS}). \quad (1.5)$$

При этом визуализации подлежат сами временные характеристики $\mathbf{y}(t) = [y_1(t) \ y_2(t) \ \dots \ y_{N_{IS}}(t)]^T$, где N_{IS} – количество ИспУ, подключенных к ХТС.

3. Задача функционального проектирования ХТС заключается в нахождении таких значений варьируемых параметров системы, выделенных из общего множества параметров компонентов объекта $P_o^* \subset P_o$, при которых достигаются требуемые значения точностных и динамических показателей качества работы СТУС [61], относящихся к классу параметров-функционалов [62]. При этом решается

задача параметрического синтеза технических объектов [63] путем минимизации целевой функции

$$F(P_o^*) = (Y_o^* - Y_o(P_o^*))^2 \rightarrow \min, \quad (1.6)$$

где Y_o^* – требуемые значения точностных и динамических показателей качества работы СТУС.

Данный класс задач решается с помощью реализации методов многовариантного анализа и параметрической оптимизации [64], базирующихся на многократном вычислении значений целевой функции (1.6) при различных значениях варьируемых параметров P_o^* .

4. Формирование и отладка сценариев управления, на основе которых должно функционировать УУ в рамках исследуемой и проектируемой СТУС, осуществляется на основе полной компьютерной модели СТУС с включенными в нее компьютерными моделями ХТС, ИзмУ и ИспУ, которая адекватно описывает поведение реальной или проектируемой СТУС. При этом каждый формируемый сценарий управления предполагает установление конкретного значения наблюдаемых переменных ХТС при воздействии на неё конкретным ИспУ. В этом случае минимизации подлежит целевая функция вида

$$\varepsilon_4(u) = \sum_{i=0}^{N_{из}} (Y_{oi}(U_o(u_o)) - \tilde{Y}_i)^2 \rightarrow \min, \quad (1.7)$$

где $U_o(u_o)$ – непосредственное энергетическое воздействие на ХТС со стороны ИспУ, пропорциональное вырабатываемой УУ команде управления u_o ; $Y_{oi}(U_o(u_o))$ – реакция рассматриваемой ХТС на команду управления, посылаемую устройством управления.

Функция (1.7) минимизируется для каждой команды управления u , формирования которой осуществляется отдельной ветвью сценария управления.

Задача синтеза ХТС, формирования и отладки сценариев управления могут решаться как без, так и с ограничениями на выходные переменные. Во втором случае задаются диапазоны изменения наблюдаемых переменных

$$y_i^{\min} \leq y_i \leq y_i^{\max} \quad (1.8)$$

определять диапазоны изменения варьируемых параметров

$$P_j^{\min} \leq P_j \leq P_j^{\max} \quad (1.9)$$

и (или) управляющих воздействий

$$u_k^{\min} \leq u_k \leq u_k^{\max}, \quad (1.10)$$

при которых не нарушаются условия (1.8)

Представленные целевые функции (1.2)-(1.7) и взаимосвязанные с ними методы их минимизации, а также ограничения на наблюдаемые переменные (1.8), варьируемые параметры (1.9) и управляющие воздействия (1.10) лежат в основе разработки соответствующих сценариев автоматизированных экспериментов, которые могут проводиться над компьютерными моделями ХТС. Наибольший эффект от решения этих задач достигается при одновременном использовании реальных химико-технологических систем и адекватных им компьютерных моделей. Такая структура комплекса программ открывает возможности реализации интеллектуальных систем управления реальными ХТС, в которых их модели используются для определения, установления и поддержания эффективных и безопасных режимов их функционирования.

Для автоматизированного решения поставленных задач исследования и функционального проектирования СТУС рассмотрен обобщенный алгоритм эксперимента, основанный на многовариантном анализе компьютерной модели СТУС с различными значениями параметров входящих в нее компонентов, для целей его реализации в формате многоуровневых компонентных цепей.

1.6 Обобщенный алгоритм автоматизированного эксперимента над сложными техническими управляемыми системами

Решение задач исследования и функционального проектирования СТУС, математическая постановка которых приведена в пункте 1.5, может быть автоматизировано с помощью их КМ, содержащих в себе модели ХТС и УУ, а также развитые средства визуализации экспериментов и интерактивного варьирования значений параметров моделей. С их использованием проведение автоматизиро-

ванного эксперимента может осуществляться на основе обобщенного алгоритма, приведенного на рисунке 1.4 и включающего следующие шаги:

Шаг 1 (Ш1). Формирование компьютерной модели сложной технической управляемой системы. Полученная в результате данного этапа КМ должна отражать топологическую структуру ХТС с подключенными к ней ИспУ и ИзмУ. В нее должны входить модели источников возмущающих воздействий. КМ ХТС должна быть взаимосвязанной с функциональной моделью УУ и с реализованными в ней сценариями управления и автоматизированного эксперимента, направленного на решение задач исследования и проектирования СТУС. Данная взаимосвязь должна быть выражена в виде средств параметризации компонентов модели ХТС и получения результатов ее анализа, подлежащих дальнейшей численной обработке и визуализации. Наиболее удобным способом представления КМ ХТС и моделей сценариев функционирования УУ является графическая форма, составленная из взаимосвязанных компонентов структуры или компонентов, выражающих определенные математико-алгоритмические операции сценария эксперимента и управления. В этом случае такая модель будет легко модифицируемой, допускать декомпозицию на модели отдельных блоков и подсистем. Уравнения входящих в модель ХТС компонентов, каждый из которых является моделью (формализованным отображением) определенного элемента, должны быть сформированы относительно переменных их связей и включать в себя переменные внешних связей ХТС с другими функциональными блоками СТУС, структура которой приведена на рисунке 1.3:

$$\mathbf{V}_s = [u, U, Q, F, Y, y]. \quad (1.11)$$

Совокупность уравнений всех компонентов, входящих в КМ СТУС, и топологических уравнений их связей образуют математическую (вычислительную) модель, *вектором решения* которой является вектор \mathbf{V}_s . Для целей эффективного решения задач в КМ СТУС в явном виде должны входить первичные параметры компонентов ХТС P_o , исполнительных P_{IS} и измерительных P_{IZ} устройств, соответствующие параметрам элементов реального объекта.

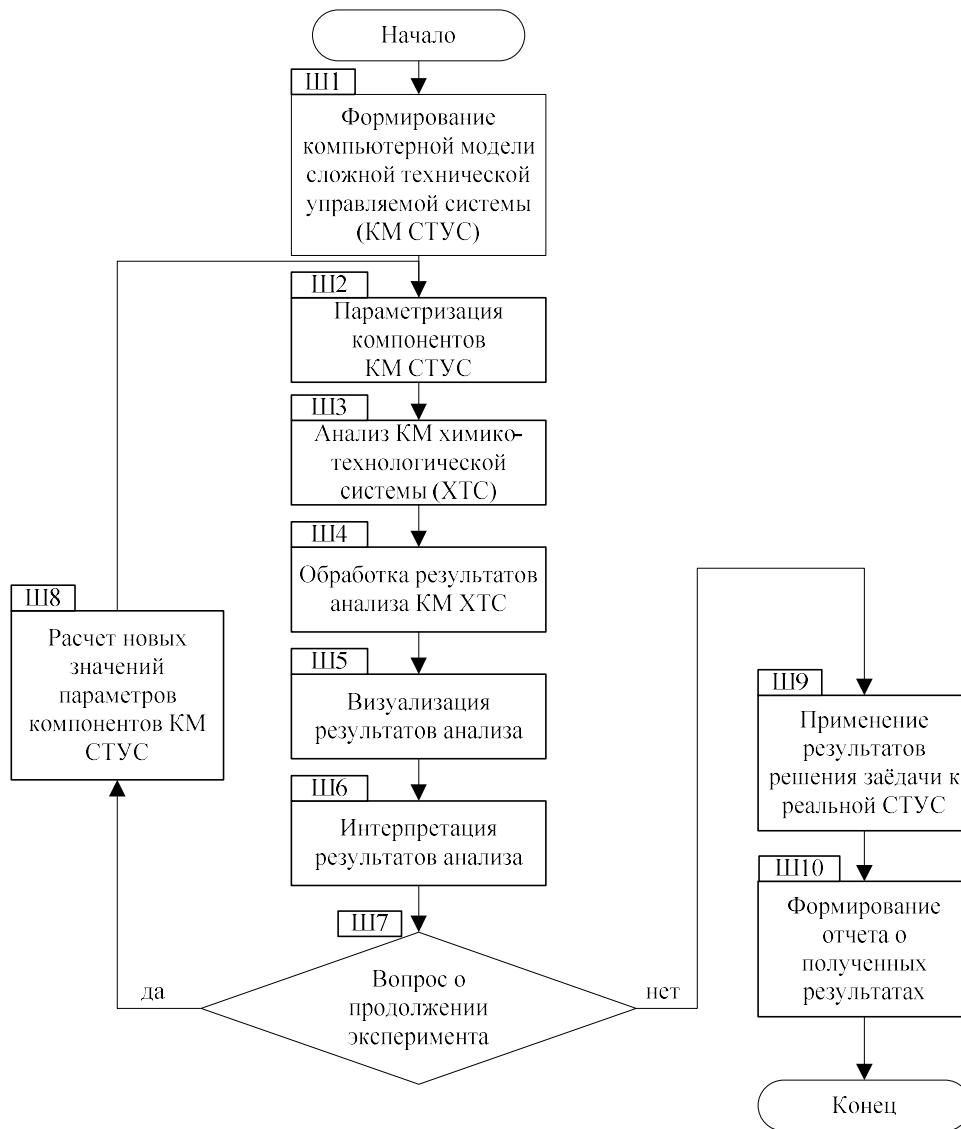


Рисунок 1.4 – Алгоритм проведения автоматизированного эксперимента над компьютерными моделями СТУС

Помимо этого, на данном шаге формируется функциональная (имитационная) модель УУ, описывающая работу реального устройства управления и включающая сценарий проведения автоматизированного эксперимента. В случае решения задач анализа, которые являются самыми простыми задачами, в формируемую КМ СТУС устанавливаются соответствующие измерительные компоненты, осуществляющие вывод результатов анализа для последующей обработки и визуализации. При решении задач исследования и функционального проектирования СТУС, а также выполняются ВЭ учебного характера, из всего набора параметров

$$P_{STUS} = (P_{IS} \cup P_O \cup P_{IS}) \quad (1.12)$$

выделяется подлежащий варьированию набор параметров

$$P_{STUS}^* \in P_{STUS}. \quad (1.13)$$

Их значения могут изменяться пользователем или автоматически в ходе решения задач, основанных на алгоритмах многовариантного анализа и параметрической оптимизации [65] с указанием метода оптимизации сформированной целевой функции (1.2)-(1.7) и начальных значений параметров множества P_{STUS}^* .

Шаг 2 (Ш2). Параметризация компонентов КМ СТУС. Устанавливаются такие значения параметров ХТС P_O , исполнительных P_{IS} и измерительных P_{IZ} устройств, при которых КМ СТУС будет адекватно описывать процессы, протекающие в реальной системе. Совокупности перечисленных параметров образуют множество параметров КМ СТУС $P_{STUS} = P_O \cup P_{IS} \cup P_{IZ}$. Значения параметров могут быть взяты из различных информационных источников: БД технических (технологических) параметров, БД характеристик веществ, ГИС-систем и др. При этом допускается основанный на использовании информационных источников предварительный расчет значений параметров с помощью средств компьютерной математики [30] с реализованными в них функциями статистической обработки информации [66]. В случае, если полученная КМ не достаточно адекватно описывает протекающие в СТУС процессы, может быть произведена коррекция значений выбранных параметров ее компонентов $P_{STUS}^* \subset P_{STUS}$, относящихся в том числе и к классу неизмеряемых параметров [67, 68]. Она заключается в минимизации целевой функции, составленной относительно наблюдаемых переменных реальной ХТС с подключенными к ней ИспУ и ИзмУ, а также их единой КМ

$$F(P_{UTO}^*) = \sum_{i=1}^{N_y} (\bar{y}_i - y_i)^2, \quad (1.14)$$

где \bar{y}_i – изменяемая характеристика реальной ХТС; y_i – значение соответствующей переменной сформированной КМ ХТС; N_y – количество наблюдаемых характеристик.

Шаг 3 (Ш3). Анализ КМ ХТС. На данном этапе выполняется вычислительный эксперимент над КМ ХТС с установленными текущими значениями ее варьируемых параметров. Он может проводиться в статическом или динамическом

режиме (во временной или частотной области). При этом для каждого из режимов задаются значения определенных параметров. Результатом ВЭ является вектор решения КМ СТУС \mathbf{V} (1.11), элементы которого в зависимости от режима анализа могут представлять собой как статические значения переменных, так и их временные, частотные или параметрические характеристики.

Шаг 4 (Ш4). Обработка результатов анализа КМ ХТС. Данный шаг выполняется средствами компьютерной математики и блоками обработки результатов моделирования [69] с целью получения значений параметров-функционалов Y_p входящих в вектор решения \mathbf{V} наблюдаемых переменных y , представленных временными, частотными или параметрическими характеристиками. К параметрическим характеристикам относятся зависимости наблюдаемых переменных от различных параметров компонентов КМ, управляющих или возмущающих воздействий (например: температура и давление окружающей среды, давление и температура вещественного потока на входе исследуемого аппарата ХТС).

Шаг 5 (Ш5). Визуализация результатов анализа. Отображению для пользователя могут подлежать полученные на ШЗ первичные результаты анализа СТУС y , а также результаты их численной обработки Y , полученные на Ш4. Их визуализация осуществляется с помощью различных средств отображения информации, представленной как в числовой форме, так и в виде временных, частотных и (или) параметрических характеристик.

Шаг 6 (Ш6). Интерпретация результатов анализа. Интерпретация заключается в анализе полученных на шагах ШЗ и Ш4 первичных и вторичных результатов моделирования, т.е. в анализе сформированных неравенств и вычислении целевых функций (1.2)-(1.7), а также их градиентов (в случае использования градиентных методов оптимизации) [64]. Помимо этого, на данном шаге может осуществляться проверка технической возможности применения результатов моделирования к реальной ХТС. Для выполнения этого шага может быть задействована реальная система, взаимодействующая с моделью при помощи универсального измерительно-управляющего контроллера [70], а также могут быть использованы продукционные модели знаний, хранящиеся в базе знаний [71].

Шаг 7 (Ш7). Вопрос о продолжении эксперимента. На данном шаге на основе результатов ряда выполненных шагов и итераций принимается решение о продолжении или завершении эксперимента. Если его цель не достигнута, то производится переход на Ш8, иначе осуществляется переход к Ш9.

Шаг 8 (Ш8). Расчет новых значений параметров компонентов КМ СТУС. Новые значения варьируемых параметров P_{STUS}^* определяются пользователем с помощью средств автоматизации математических вычислений и интеллектуализации либо согласно выбранному на Ш1 методу минимизации целевой функции (1.2)-(1.7) на основе ее текущего значения и значений измеряемых переменных модели СТУС y . Шаги изменения параметров могут задаваться пользователем или определяться автоматически на основе вычисления градиента целевой функции. Определяемые на данном шаге новые значения варьируемых параметров P_{STUS}^* передаются на этап параметризации компонентов КМ СТУС (Ш2). Путем данного перехода осуществляется циклическое выполнение автоматизированного эксперимента, лежащего в основе решения задач исследования, проектирования и функционирования СТУС.

Шаг 9 (Ш9). Применение результатов решения задач к реальной СТУС. На данном шаге с помощью соответствующей экспертной системы выявляется возможность применения полученных результатов к реальной СТУС. В случае положительного решения данного вопроса специально сформированными командами осуществляется установка соответствующих параметров в реальной системе, полученных при решении задач проектирования, а в ходе управления – осуществляется применение к реальной ХТС сформированных управляющих воздействий \tilde{u}_0 . При этом наблюдение за состоянием реальной системы должно выполняться с помощью того же комплекса программ моделирования, в котором производилось решение соответствующих задач. Это позволит автоматически сравнивать значения наблюдаемых переменных с полученными ранее результатами моделирования. Помимо этого открываются возможности проведения дополнительного моделирования и анализа, а также формулирования и решения другой задачи исследования, проектирования и функционирования СТУС.

Шаг 10 (Ш10). Формирование отчета о полученных результатах. В процессе проведения исследований над СТУС и ее КМ появляются подлежащие протоколированию и документированию результаты. Они могут быть результатами анализа модели и решения задач на основе моделирования, выраженными схемами, алгоритмами и программами функционирования УУ, командами управления реальным объектом или протоколами выполнения практических занятий учебного назначения. Эти данные должны быть переданы специальному модулю, задачей которого является формирование отчета о выполнении экспериментов, представленного документом и предназначенным для его хранения, копирования и использования.

Представленный на рисунке 1.4 алгоритм автоматизированного эксперимента позволяет с помощью КМ СТУС осуществлять решение задач, направленных на исследование и функциональное проектирование ХТС и систем управления ими. Для его основе реализуется основанный на развитии метода компонентных цепей [17] комплекс программ компьютерного моделирования химико-технологических систем. На его основе открываются возможности разработки различных виртуальных и реально-виртуальных лабораторий, компьютерных тренажеров и имитаторов, предназначенных для автоматизации процессов обучения и переподготовки технического персонала предприятий газовой и других отраслей промышленности.

Для реализации предложенного алгоритма выполнения автоматизированных экспериментов над моделями СТУС предлагается использовать обобщенную структуру компьютерной модели сложных технических (технологических) управляемых систем. Сформированная в формате универсального метода компьютерного моделирования, предназначенного для анализа СТУС, КМ должна включать ММ ХТС, имитационные модели сценариев выполнения автоматизированных экспериментов и функционирования УУ, а также визуальные модели средств отображения результатов и интерактивного управления параметрами модели и управляющих воздействий на реальные объекты.

1.7 Многоуровневая схема автоматизированного эксперимента на основе компьютерных моделей СТЭС

Автоматизированный эксперимент направлен на изучение свойств и характеристик процессов, протекающих в исследуемых и проектируемых сложных технических управляемых системах. Он также позволяет с помощью их КМ определять значения ХТС и управляющих воздействий на реальную систему для установки и поддержания в ней требуемых (желаемых) режимов функционирования. Протекающие в ХТС процессы носят преимущественно непрерывный характер. Процессы УУ являются дискретно-событийными.

При выполнении ВЭ для описания непрерывных процессов, протекающих в ХТС, применяются математические модели и осуществляющие их построение и решение методы математического моделирования. ММ представляют собой системы алгебро-дифференциальных уравнений в обыкновенных и/или частных производных, сформированных относительно его внутренних и внешних параметров и переменных. Применяемые для их решения методы математического моделирования включают в себя численные методики линеаризации нелинейных и алгебраизации дифференциальных уравнений [72], а также численного решения систем линейных алгебраических уравнений [73]. К классу методов математического моделирования относятся табличный метод, метод передаточных функций [74], метод переменных состояний [75], отличающиеся друг от друга набором базисных переменных модели. На базе этих методов реализованы комплексы программ компьютерного моделирования объектов конкретной физической природы: электрической [76, 77, 78, 79]; пневматической [80]; гидравлической [81]; тепловой и теплоэнергетической [82, 83, 84].

Основными пакетами моделирования и анализа химико-технологических систем являются Aspen HYSYS [43] и ChemCAD [44, 45]. Данные комплексы программ осуществляют анализ процессов, протекающих в элементах и аппаратах химической промышленности. Однако в них отсутствует возможность подключения к моделям ХТС, представленным в графической форме, моделей средств автоматизации – исполнительных и измерительных устройств, а также функции фор-

мирования, отладки и тестирования сценариев управления. К тому же эти пакеты не позволяют автоматизировать решение задач многовариантного анализа и параметрической оптимизации, а также не допускают своей интеграции с реальными химико-технологическими системами для применения к ним результатов моделирования.

Для построения компьютерных моделей СТУС, предназначенных для проведения автоматизированных экспериментов научно-исследовательского, проектно-конструкторского и учебного назначения, необходимо применять метод, позволяющий формировать и решать ММ ХТС с энергетическими и многокомпонентными вещественными потоками в связях, а также интегрировать её с дискретно-событийными моделями УУ и сценариями решения задач исследования и проектирования СТУС на базе многовариантного анализа ММ ХТС.

Дискретные процессы описываются разностными уравнениями, представленными в рамках динамических систем дискретного времени [85]. Явное представление алгебро-дифференциальных и разностных уравнений в силу ряда причин (многочисленные энергетические обратные связи, зависимость потенциальных и потоковых переменных одной физической природы друг от друга и переменных, принадлежащих другим физическим природам) в таких системах затруднено, а в некоторых случаях невозможно. Помимо разностных уравнений объекты с протекающими дискретными процессами могут быть представлены имитационными моделями (ИМ) [86, 87], характеризующимися возможностью представления выражений и уравнения в явном виде относительно выходных переменных. В этом случае они могут быть описаны как математическими, так и математико-алгоритмическими выражениями, допускающими вычисления значений выходных переменных непосредственно в компоненте с передачей полученных результатов компонентам, подсоединенным к его выходным выводам (узлам). К классу имитационного относится дискретно-событийное моделирование (ДСМ) [88, 89]. Оно представляет собой подход, предполагающий абстрагирование от непрерывной природы объектов с рассмотрением только отдельных событий моделируемой системы. ДСМ наиболее развито и имеет огромную сферу приложений – от логи-

стики и систем массового обслуживания [90] до транспортных и производственных систем. С помощью ИМ могут быть описаны объекты с дискретно-событийным поведением, а также последовательность действий, представленная в виде алгоритмов и сценариев. На основе их отображения с помощью ИМ организована работа системы LabView [91] фирмы National Instruments [92]. Она позволяет строить и использовать виртуальные инструменты и приборы, являющиеся имитационными (функциональными) моделями своих реальных аналогов. Также на этом подходе функционируют большинство SCADA-систем [93–97], применяемых для автоматизации контроля и управления различными промышленными объектами. Несмотря на выполнение ими своих основных задач, в современных SCADA-системах отсутствуют или слабо представлены функции прогнозирования динамики процессов и интеллектуализации определения управляющих воздействий, требующие анализа моделей объектов при различных значениях ее параметров. Для решения этих проблем могут быть разработаны интеллектуальные системы управления с компьютерными моделями в контуре управления [36].

Выполняемый над ХТС или её моделью автоматизированный эксперимент (натурный или вычислительный), обобщённый алгоритм которого приведен на рисунке 1.4, представляет собой последовательность действий (сценарий), каждое из которых может быть описано своим математическим или математико-алгоритмическим выражением. Совокупность этих выражений, представленных в явном виде относительно результатов каждого предыдущего шага и обрабатываемых выполняемых пользователем действия (события), могут быть представлены дискретно-событийными моделями. Для их описания в настоящее время находят распространение алгоритмы и средства имитационного моделирования [98].

Для отображения пользователю результатов эксперимента и осуществления возможностей управления его ходом, а также для задания и изменения варьируемых параметров исследуемой СТЭС или ее модели, предусматривается панель визуализации и интерактивного управления. Она состоит из графических элементов отображения результатов эксперимента в понятной пользователю форме и формирования управляющих воздействий, позволяющих изменять величины входных

сигналов и варьировать значения изменяемых параметров модели ХТС. В случае реализации ВЭ соответствующая панель должна быть реализована виртуально и представлять собой набор графических элементов, визуализирующих на экране монитора характеристики соответствующих процессов.

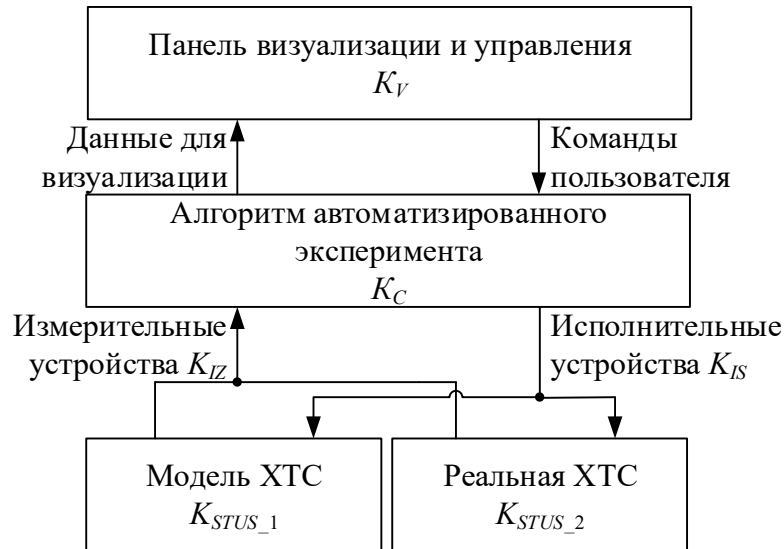


Рисунок 1.5 – Многоуровневая схема автоматизированного эксперимента

Для автоматизации эксперимента, лежащего в основе автоматизации решения задач исследования и функционального проектирования СТУС, алгоритм которого приведен на рисунке 1.4, предложена многоуровневая схема, представленная на рисунке 1.5. Ее программно-алгоритмической реализацией является многоуровневая компьютерная модель СТУС, состоящая из компонентов следующих классов:

- K_V – класс компонентов, предназначенных для отображения результатов натуральных и вычислительных экспериментов в понятной исследователю форме, а также компонентов интерактивного изменения значений управляющих воздействий и варьируемых параметров модели исследуемой СТУС. Данный класс компонентов далее будем называть *визуальными компонентами*;

- K_A – класс компонентов, называемых далее *алгоритмическими* и предназначенных для построения и выполнения сценариев автоматизированных экспериментов, обобщенный вид которых приведен на рисунке 1.4;

– K_O – класс *объектных компонентов*, образующих компьютерную модель СТУС K_{STUS_V} , которую составляет модель ХТС с подключенными к ней моделями ИспУ и ИзмУ, а также средства подключения к реальному объекту с помощью универсальных измерительно-управляющих контроллеров K_{STUS_R} .

Приведенная классификация компонентов основана на различии процессов, протекающих в связях компонентов соответствующих классов:

1. Компоненты множества K_O , из которых формируются КМ исследуемой ХТС с подключенными к ней моделями ИспУ и ИзмУ, при своем соединении образуют узлы вида N_e , общий вид которых приведен на рисунке 1.6. Они описывают процессы обмена энергией между компонентами, которые являются моделями элементов ХТС. В каждом из узлов должны выполняться два топологических закона: закон равенства потенциальных переменных всех связей, подсоединенных к одному узлу, и закон равенства нулю алгебраической суммы потоковых переменных, входящих в один узел. К узлам данного вида могут подключаться энергетические $S_{E1}, S_{E2}, \dots, S_{En}$ и информационные $S_{I1}, S_{I2}, \dots, S_{Im}$ связи компонентов множества K_O .

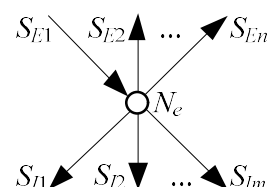


Рисунок 1.6 – Обобщенный узел N_e , образованный соединением компонентов множества K_O

2. Соединениями связей компонентов множества K_A образуются узлы вида N_a , представленные на рисунке 1.7. К ним подсоединяются информационные связи компонентов. С помощью узлов данного вида осуществляется передача сигнал-переменных от компонентов-источников к компонентам-получателям данного сигнала, называемого далее *сообщением*.

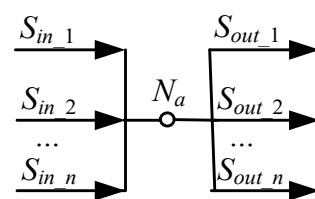


Рисунок 1.7 – Обобщенный узел N_a , образованный соединением компонентов множества K_A

3. Компоненты множества K_V присоединяются к узлам вида N_a , осуществляя тем самым прием подлежащих визуализации данных или передачу в

сценарий эксперимента воздействий пользователя на общую компьютерную модель СТУС.

Представленная на рисунке 1.5 многоуровневая схема автоматизированного эксперимента может быть реализована в виде многоуровневой структуры, каждый уровень которой включает в себя компоненты определенных множеств:

- *Визуальный уровень* организует взаимодействие модели с пользователем во время ее работы, осуществляя визуализацию результатов анализа и предоставляя возможность пользователю изменять значения варьируемых параметров компонентов и управляющих воздействий на КМ и реальную ХТС. Он формируется из компонентов множества K_V .

- *Логический уровень* включает в себя составленную из компонентов множества K_A имитационную модель сценария автоматизированного эксперимента, реализуемого на основе обобщенного алгоритма, представленного на рисунке 1.4. В нее также входит функциональная модель устройства управления исследуемой СТУС, которая взаимодействует с реальной ХТС и её моделью;

- *Объектный уровень* содержит составленную из компонентов множества K_O компьютерную модель ХТС с подключенными к ней моделями ИспУ и ИзмУ. Он также включает в себя средства подключения модели к реальному объекту с помощью универсальных измерительно-управляющих контроллеров.

Для реализации приведенной на рисунке 1.5 многоуровневой схемы автоматизированного эксперимента осуществлено развитие метода компонентных цепей [17], являющегося универсальным методом компьютерного моделирования, позволившее:

- формировать и анализировать компьютерные (математические) модели ХТС, допускающих декомпозицию на элементы, каждый из которых может быть представлен *компонентом* – моделью-заменителем, описывающим протекающие в элементе процессы преобразований мультифизических энергетических и многокомпонентных вещественных потоков с помощью систем алгебро-дифференциальных уравнений;

– составлять имитационные модели сценариев автоматизированных экспериментов, направленных на решение задач исследования и функционального проектирования СТУС, в основе которых лежит представленный на рисунке 1.4 алгоритм;

– представлять набранные из визуальных компонентов средства отображения результатов экспериментов в понятном пользователю виде, и компонентов интерактивного изменения значений параметров компонентов КМ СТУС и управляющих воздействий на реальный объект.

Реализованный на основе метода комплекс программ моделирования химико-технологических систем позволит автоматизировать решение задач исследования и проектирования СТУС, а также реализовывать режимы интеллектуального управления ХТС, предполагающие использование их моделей в процессе управления протекающими процессами.

1.8 Выводы

1. На основе системного анализа сложных технических управляемых систем, предназначенных для управления химико-технологическими системами, сформировано их формализованное представление, расширяющее понятийно-определяющий аппарат МКЦ и позволяющее декомпозировать СТУС на самостоятельные взаимосвязанные элементы, между которыми протекают мультифизические энергетические и многокомпонентные вещественные потоки, управляемые информационными сигналами устройства управления.

2. Сформирована структура компьютерной модели сложной технической управляемой системы, позволяющая с использованием средств компьютерного моделирования, автоматизированных вычислений и элементов поддержки принятия управляющих решений автоматизировать процессы выполнения экспериментов научно-исследовательского, проектного и учебного назначения, а также автоматизировать процесс управления объектами с применением их компьютерных моделей.

3. Осуществлена математическая постановка задач исследования и функционального проектирования сложных технических управляемых систем, решение

которых основано на автоматизированном эксперименте с использованием компьютерных моделей СТУС.

4. Разработан обобщенный алгоритм эксперимента, предназначенного для автоматизированного решения задач исследования и функционального проектирования СТУС, который основан на использовании их компьютерных моделей, включающих математические модели химико-технологических систем, исполнительных и измерительных устройств, а также имитационную (функциональную) модель устройства управления.

5. Сформирована и обоснована многоуровневая схема автоматизированного эксперимента, предназначенного для исследования компьютерных моделей сложных технических управляемых систем с возможностью применения результатов к реальному объекту и организации режима интеллектуального управления с выработкой управляющих воздействий с применением компьютерных моделей. Предложенный программно-алгоритмический аппарат позволяет осуществлять функциональное проектирование и отладку сценариев функционирования SCADA-систем, в которых компьютерные модели управляемых ХТС используются для автоматизированного определения, установления и поддержания эффективных и безопасных режимов функционирования.

Глава 2. Многоуровневая интерпретация метода компонентных цепей для моделирования сложных технических управляемых систем

Глава посвящена вопросам теоретического развития и многоуровневой интерпретации метода компонентных цепей. Данный подход откроет новые возможности исследования сложных технических управляемых систем предприятий газовой промышленности с исследованием протекающих в ХТС процессов, разработки и отладки сценариев, лежащих в основе функционирования устройств управления.

2.1 Метод компонентных цепей

2.1.1 Назначение метода компонентных цепей

К классу универсальных методов компьютерного моделирования технических объектов и систем относится метод компонентных цепей [17]. Его главными отличительными особенностями являются:

1. Метод относится к классу объектно-ориентированных языков [99] и предназначен для моделирования сложных объектов и систем с информационными и энергетическими потоками в связях.

2. Элементы моделируемых с его помощью объектов и систем могут иметь различную физическую природу (электротехника/электромеханика, механика/мехатроника, робототехника, гидравлика, термодинамика, теплотехника, химические технологии и т.п.) и входить в единую модель исследуемой мультифизической системы.

3. Модель каждого элемента, называемая *компонентом*, формируется автоматически с учетом четырех основных аспектов: геометрического, топологического, физического и математического (математико-алгоритмического). Она представляет собой систему алгебро-дифференциальных уравнений в обыкновенных и/или частных производных. В случае наличия в модели логических соотношений и алгоритмических блоков она может быть представлена в явном виде и относиться к классу имитационных моделей.

4. Для объектов и систем с функционально обособленными подсистемами в рамках МКЦ введено понятие *подцепи* [100] – структуры, допускающей автономное решение. Она позволяет четко разделить объекты и системы с непрерывным и дискретно-событийным поведением.

5. Форма уравнений (выражений) модели и ее топологическая структура могут изменяться в зависимости от поведения переменных и наступления определенных событий.

Метод компонентных цепей позволяет представить сложный, физически неоднородный технических (технологический) объект с информационными и энергетическими связями, допускающий декомпозицию на взаимосвязанные элементы, в виде *компонентной цепи (КЦ)* – КМ в формате данного метода, предназначенной для проведения его анализа в статическом и динамическом режиме. Кроме того, на основе метода разработана система формализованного представления математических выражений в формат метода КЦ [101] и реализован интерпретатор, осуществляющий автоматическое построение соответствующих им КЦ [102] для целей расчета математических выражений и их использования при моделировании объектов в качестве моделей компонентов. Для автоматизированного анализа КЦ технических и технологических объектов и расчета КЦ математических выражений разработано универсальное вычислительное ядро [103]. Оно лежит в основе разработки системы компьютерного моделирования МАРС [21] и системы компьютерной математики «Макрокалькулятор» [30]. Обе системы в совокупности образуют комплекс программ «Расчетно-моделирующая среда» [104]. На ее основе реализуются компьютерные задачки и тренажеры [105], виртуальные лаборатории [21] по различным техническим дисциплинам: основы электротехники и электроники [23, 106], теория автоматического управления [22], компьютерное моделирование, а также система компьютерного моделирования физических задач [107].

С целью максимального приближения реализуемого в среде моделирования МАРС вычислительного эксперимента к натурному в её рамках разработан редактор виртуальных инструментов и приборов (редактор ВИП) [108]. Он позволяет

представить в формате метода КЦ виртуальные генераторы и измерительные приборы, состоящие из лицевых панелей, алгоритмов функционирования и интерфейсов подключения к исследуемой модели в виде специализированных КМ, а затем использовать их для визуализации результатов моделирования и интерактивного управления параметрами исследуемой модели.

Рассмотрим основные понятия метода КЦ для целей его развития и многоуровневой интерпретации, образующей открывающей возможности построения многоуровневых компонентных цепей (МКЦ), предназначенных для исследования и функционального проектирования сложных технических управляемых систем газовой промышленности.

2.1.2 Основные понятия метода компонентных цепей

Базовыми понятиями метода КЦ [17, 18] являются понятие компонента и компонентной цепи.

Компонент является формализованным отображением элемента или функционального блока моделируемого объекта или системы. Он имеет свою математическую или математико-алгоритмическую модель, уравнения или выражения

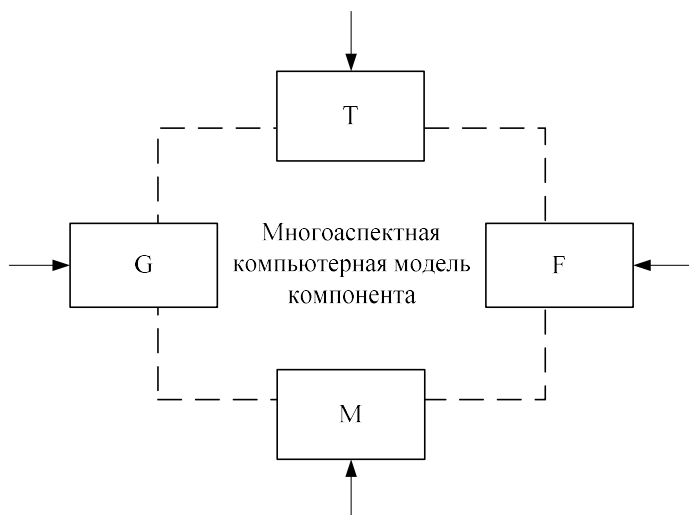


Рисунок 2.1 – Многоаспектная компьютерная модель компонента

которой составлены относительно его первичных параметров и переменных. В рамках метода КЦ модель компонента (МК), структура которой представлена на рисунке 2.1, формируется на основе совокупности следующих аспектов:

G – геометрического аспекта, включающего в себя условное графическое и (или) буквенное обозначение компонента на

принципиальных, структурных или технологических схемах, а также правила отображения результатов на панелях визуализации и интерактивного управления;

T – топологического аспекта, определяющего количество и типы всех связей S , их ориентацию относительно компонента и совокупность переменных, принадлежащих каждой связи, называемых далее переменными связей компонента. В зависимости от вида моделирования (математического, имитационного или визуального) вводятся различные типы связей с соответствующими им переменными;

F – физического аспекта, определяющего протекающие в соответствующем элементе физические процессы, а также включающего в себя его свойства, называемые атрибутами компонента A, и их значения – параметры P;

M – математического аспекта, представляющего собой математическое или математико-алгоритмическое описание протекающих в отображаемом элементе процессов. Математическая МК формируется в локальном координатном базисе (ЛКБ) компонента в виде системы выражений или уравнений, записанных относительно первичных параметров и переменных его связей в физическом координатном базисе (ФКБ). Математико-алгоритмическая МК – это выражение или совокупность выражений, составленных в явном виде из переменных входных связей для вычислений значений переменных соответствующих выходных связей компонента.

Формируемая программно-алгоритмическая реализация МК состоит из блоков, каждый из которых соответствует определённому аспекту:

– *геометрического блока*, включающего в себя реализацию изображения компонента в виде условного графического изображения и полюсов – специально обозначенных мест, предназначенных для его коммутации с другими компонентами. Каждому полюсу соответствует определенная связь из множества S ;

– *топологического блока*, реализующего информацию о направлениях потоков в связях компонента. В каждой связи множества S поток может быть направлен от полюса к компоненту или от компонента к полюсу. В первом случае поток будет являться входящим и приниматься со знаком «+», во втором он будет исходящим и приниматься со знаком «-». Помимо этого допускается и неориентированные связи, по которым передаются информационные потоки;

– *функционального (вычислительного) блока*, содержащего реализацию математической или математико-алгоритмической МК. ММ присуща компонентам моделируемого объекта, в котором протекают непрерывные энергетические процессы и процессы преобразования многокомпонентных вещественных потоков под управлением информационных сигналов. Она представляет собой систему алгебро-дифференциальных уравнений, сформированных в неявном виде относительно параметров компонента и переменных всех его связей. Математико-алгоритмические МК представляют собой совокупность выражений, записанных в явном виде относительно переменных выходных связей и предназначенных для расчета их значений. С помощью таких моделей могут быть описаны сценарии эксперимента и функционирования УУ, входящих в состав СТУС;

– *измерительного блока*, осуществляющего выборку соответствующих значений из общего решения КЦ и их передачу другим компонентам для целей математико-алгоритмической обработки и визуализации;

– *визуального блока*, реализующего функции принятия и отображения в понятном пользователю виде результатов моделирования и измерения, а также осуществления интерактивных действий пользователя с моделью объекта.

В зависимости от вида компонента и от типа связей в его программно-алгоритмическую реализацию может входить произвольная совокупность описанных блоков. Каждый из них представляется одной или несколькими функциями, которые вызываются соответствующими модулями комплекса программ, реализующими алгоритмы моделирования.

Компонентной цепью [17], являющейся основной КМ исследуемого объекта или системы в формате метода КЦ, в общем случае будет произвольная совокупность компонентов, связи которых, именуемые *ветвями цепи*, соединены в общих точках, называемых *узлами цепи*. С формальной точки зрения КЦ представляет собой совокупность трех взаимосвязанных множеств

$$C = (K, B, N), \quad (2.1)$$

где K – множество компонентов; B – множество связей (ветвей); N – множество узлов КЦ, образованных при коммутации связей компонентов S .

Каждой связи S_j компонента множества K (2.1) с номером b_j , являющимся номером ее ветви, заданным в ЛКБ компонента, ставится в соответствие ее уникальный в рамках КЦ C (2.1) номер B_j , где $J = 1..CB$, CB – количество ветвей цепи. Каждый узел множества N КЦ C (2.1) характеризуется своим уникальным номером N_I ($I = 1..CN$, CN – количество узлов цепи), который сообщается связям всех компонентов, подключенных к данному узлу. Таким образом, массив номеров ветвей \mathbf{B} размерностью CB и массив номеров узлов \mathbf{N} размерностью CN представляют в совокупности глобальный координатный базис (ГКБ) КЦ C (2.1).

Расширением метода КЦ является метод подцепей [100], согласно которому КЦ C (2.1) может быть представлена совокупностью подцепей

$$C = (C_1 \cup C_2 \cup \dots \cup C_n), \quad (2.2)$$

где C_1, C_2, \dots, C_n – компонентные цепи вида (2.1).

Как для отдельных подцепей C_1, C_2, \dots, C_n , так и для всей КЦ (2.1) на основе четырех основных аспектов метода КЦ могут быть автоматически построены и решены их вычислительные модели (ВМ), позволяющие определять статические значения переменных связей входящих в них компонентов, их временные и частотные характеристики. В рамках метода КЦ предусматриваются различные виды многовариантного анализа [109] и параметрического синтеза объектов [63], для которых на основе компонентных цепей исследуемых объектов могут быть построены как семейства выходных характеристик, так и другие характеристики.

2.1.3 Методика построения компонентной цепи технических объектов с энергетическими и информационными связями

Метод КЦ [17] является универсальным методом компьютерного моделирования технических объектов с информационными и энергетическими связями. В таких объектах под управлением информационных сигналов протекают процессы получения, преобразования и использования энергии различной физической природы.

С позиции топологического аспекта технический объект, между компонентами которого осуществляется обмен физически неоднородной энергией и информацией, имеет свое отображение в виде КЦ объекта с энергетическими и информационными связями

$$C_o = (K_o, B_o, N_o), \quad (2.3)$$

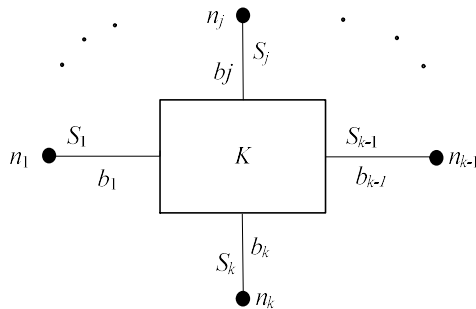


Рисунок 2.2 – Обобщенный вид компонента

где K_o – множество компонентов технических объектов; B_o – множество ветвей (связей) всех компонентов множества K_o ; N_o – множество узлов, образованных путем соединения связей всех компонентов множества K_o .

Каждый компонент множества K_o ,

обобщенный вид которого представлен на рисунке 2.2, имеет множество связей S_j ($j = 1..k$). Для построения и анализа моделей технических объектов в формате МКЦ введены три типа связей:

– *энергетический* тип связи. Каждой связи данного типа S_j ставится в соответствие пара топологических координат – узел n_j и ветвь b_j – с переменными (V_{n_j}, V_{b_j}) , с переменными V_{n_j} – потенциальная, V_{b_j} – потоковая переменные, называемые далее *переменными связи*. Такую связь будем называть *энергетической*

$$S_j = S_{Ej} = (b_j \eta n_j) \rightarrow \{V_{n_j}, V_{b_j}\}; \quad (2.4)$$

– *информационный* тип связи, согласно которому каждой связи S_j ставится в соответствие пара топологических координат – узел n_j и ветвь b_j – с одной потенциальной переменной V_{n_j} . Связь данного типа будем называть *информационной*

$$S_j = S_{Ij} = (b_j \eta n_j) \rightarrow \{V_{n_j}\}; \quad (2.5)$$

– *однородный векторный* тип связи, согласно которому каждой связи ставится в соответствие пара топологических координат – узел n_j и ветвь b_j – с вектором потенциальных переменных, например, координат точек кинематиче-

ской системы. Связь данного типа будет называться *однородной векторной связью*

$$S_j = S_{IVj}. \quad (2.6)$$

Формирование моделей компонентов множества K_O (2.3) производится с учетом четырех основных аспектов метода КЦ, каждый из которых характеризуется набором определенных предикатов:

Геометрический аспект предписывает каждому компоненту, который является моделью определенного элемента моделируемого технического объекта, свое условное графическое и буквенное обозначение в графическом редакторе, регламентированное соответствующим ГОСТом [110]. Кроме того, данный аспект предписывает каждой связи компонента свое изображение, называемое *полюсом*. С их помощью организуются связи между компонентами.

Топологический аспект накладывает ограничения на типы всех связей компонента, каждой из которых ставит в соответствие определенную совокупность переменных связей. Каждая переменная (потенциальная или потоковая) характеризуется определенным порядковым номером в ЛКБ компонента. Для каждой энергетической связи S_{Ej} задается направление ее потока парой значений: от узла к компоненту $(b_j \eta^- n_j) = (-b_j, n_j)$ и $(b_j \eta^+ n_j) = (b_j, n_j)$ в обратном направлении. Коммутация связей компонентов согласно топологическому аспекту производится на основании правила 2.1.

Правило 2.1. Согласно топологическому аспекту соединение допустимо только между компонентами, имеющими один тип примыкающей связи.

Физический аспект позволяет рассмотреть физические процессы, протекающие в элементе, с различной степенью их детализации. Он накладывает физический смысл на все параметры компонента и переменные каждой его связи в виде определенного субаспекта. Каждый энергетический субаспект F_E физического аспекта для каждой энергетической связи представляется в виде множества свойств

$$F_E = \left\{ \begin{array}{l} N_A, Name_nVar, Symbol_nVar, Value_nVar, SI_nVar, \\ Name_bVar, Symbol_bVar, Value_bVar, SI_bVar \end{array} \right\}, \quad (2.7)$$

где N_A – наименование энергетического субаспекта физического аспекта (например: динамика); $Name_nVar$ – имя потенциальной переменной данного субаспекта (например: скорость); $Symbol_nVar$ – символ, которым данная переменная выражается в уравнениях, определяющих ММ компонента (например: V); $Value_nVar$ – значение потенциальной переменной, полученное при анализе модели на конкретном шаге анализа; SI_nVar – единица измерения потенциальной переменной (например: м/с); $Name_bVar$ – имя потоковой переменной данного субаспекта (например: сила); $Symbol_bVar$ – символ, которым данная потоковая переменная выражена в уравнениях ММ компонента (например: F); $Value_bVar$ – значение потоковой переменной, полученное при анализе модели на конкретном шаге анализа; SI_bVar – единица измерения значений потоковой переменной (например, Н).

Энергетическими субаспектами физического аспекта являются механический, гидравлический, термодинамический, электрический, магнитный и другие, мощность потоков которых равна произведению мгновенных значений соответствующих потенциальной и потоковой переменных.

Информационный субаспект F_I физического аспекта для каждой информационной связи компонента представляется в виде

$$F_I = \{N_A, Name_nVar, Symbol_nVar, Value_nVar, SI_nVar\}, \quad (2.8)$$

где N_A – наименование информационного субаспекта физического аспекта (например, уровень заполнения резервуара); $Name_nVar$ – имя потенциальной переменной данного субаспекта (например, количество жидкости в резервуаре); $Symbol_nVar$ – символ, которым данная потенциальная переменная выражается в уравнениях, образующих математическую модель данного компонента; $Value_nVar$ – значение потенциальной переменной, полученное при анализе мо-

дели при конкретном шаге ВЭ (например: h); SI_nVar – единица измерения значения потенциальной переменной (например, m^3).

Представителями информационного субаспекта физического аспекта являются:

- математический, выражающий значение аргумента или результат выполнения каждой операции в рассматриваемом математическом выражении, представленном в формате КЦ [101];

- вещественный компонентный, выражающий в долях или процентах занимаемую веществом часть многокомпонентного вещественного потока [111];

- экономический, позволяющий рассматривать стоимостные характеристики информационных и энергетических потоков [112].

Согласно физическому аспекту коммутация связей компонентов осуществляется на основе правила 2.2

Правило 2.2. Соединение допустимо между связями компонентов, имеющих одинаковый субаспект F физического аспекта.

Проверка их соответствия производится на этапе построения компьютерной (вычислительной) модели исследуемого объекта.

Математический аспект является наиболее актуальным при компьютерном моделировании объектов и систем. Он связан с представлением математической или вычислительной МК, а также с формированием и решением ВМ цепи целиком. Системный анализ компонентов моделируемого объекта позволяет упростить формализованное представление объекта в формате КЦ за счет выбора удобного топологического представления компонентов и автоматического формирования уравнений законов сохранения для потоковых переменных.

Математическая модель компонента (ММК) – это уравнение или система уравнений, описывающая законы его функционирования и составленная относительно параметров компонента и переменных его связей. Такие законы могут быть представлены в аналитическом, алгоритмическом или таблично-графическом виде. Согласно методу КЦ для своего использования при анализе технических объектов модель компонента должна быть преобразована к аналити-

ческому виду, то есть для нее должно быть сформировано уравнение или система уравнений.

Вычислительная модель компонента – это его ММК, представленная в формате рассматриваемого метода. Методом КЦ допускаются следующие типы уравнений, составленные относительно переменных связей компонента $V^k = V_n^k \cup V_b^k$, называемых *компонентными уравнениями*:

– линейные уравнения вида

$$\sum_i a_i \cdot V_i^k = b, \quad (2.9)$$

где a_i, b – постоянные коэффициенты;

– нелинейные уравнения

$$f(V^k) = 0, \quad (2.10)$$

где $f(V^k)$ – произвольная функция;

– дифференциальные уравнения

$$\frac{d\Psi(V^k)}{dt} = f(V^k, t), \quad (2.11)$$

где $\Psi(V^k) = \sum_i \psi_i \cdot V_i^k$ – линейная форма с постоянными коэффициентами;

$f(V^k, t)$ – произвольная функция;

– уравнения источников

$$\sum_i a_i \cdot V_i = b(t), \quad (2.12)$$

где a_i – постоянные коэффициенты, $b(t)$ – произвольная функция времени.

Для целей формирования ВМ компонентной цепи программно-алгоритмическая реализация каждого компонента множества K_o (2.3) должна включать реализацию соответствующих блоков. Передача данных, хранящихся в описанных блоках компонентов, производится путем опроса вычислительным аппаратом математического моделирования соответствующего блока всех компонентов.

Формирование ВМ КЦ технического объекта производится на основе алгоритма автоматического построения модели цепи.

2.1.4 Алгоритм автоматического построения модели компонентной цепи

Построение КЦ моделируемого объекта C_o (2.3) заключается в преобразовании ЛКБ каждого из компонентов множества K_o

$$(n_1, n_2, \dots, n_k, b_1, b_2, \dots, b_k) \quad (2.13)$$

в ГКБ цепи

$$(N_0, N_1, \dots, N_{CN-1}, B_1, B_2, \dots, B_{CB}), \quad (2.14)$$

где CN – количество узлов КЦ; CB – количество ее ветвей. Каждому узлу и каждой ветви ставится в соответствие свой уникальный номер в ЛКБ цепи.

Анализ технических объектов с энергетическими и информационными связями программно-алгоритмическими средствами метода КЦ предполагает автоматическое построение ВМ исследуемого объекта на основе его КЦ C_o (2.3), линеаризацию нелинейных (2.10) и алгебраизацию дифференциальных уравнений (2.11), а также определение методов линейной алгебры значений переменных связей всех компонентов, образующих вектор решения цепи

$$V = [[V_N], [V_B]]. \quad (2.15)$$

Решением КЦ является совокупность значений всех потенциальных $[V_N]$ и потоковых $[V_B]$ переменных связей КЦ при статическом или динамическом анализе во временной области. Анализ КЦ объекта в частотной области предполагает расщепление каждой потенциальной и потоковой переменной на ее вещественную и мнимую части

$$V_i = V_i' + 1j \cdot V_i'', \quad (2.16)$$

где $V_i' = \text{Re}(V_i)$, $V_i'' = \text{Im}(V_i)$.

Построение математической модели КЦ заключается в объединении вычислительных моделей компонентов и узловых топологических законов сохранения для потенциальных и потоковых переменных

$$M_C = \left[\bigcup_i M_{Ki} \cup \bigcup_k M_{Nk} \right], \quad (2.17)$$

где M_{Ki} – множество ВМ компонентов, состоящих из уравнений (2.9)–(2.12); M_{Nk} – множество ММ узлов цепи, включающих топологические уравнения вида

$$\sum_j \alpha_j \cdot V_{Bj} = 0, \quad (2.18)$$

где α_j – числовой коэффициент, равный $+1$, если поток в ветви B_j направлен к узлу, и -1 – если от узла, для которого формируется конкретное топологическое уравнение.

Алгоритм построения модели КЦ зависит от выбранного режима ее анализа (статический или динамический, во временной или в частотной области) и уравнений моделей входящих в КЦ компонентов. На его основе формируется матрично-топологическая и вычислительная модели КЦ. Алгоритм построения модели КЦ состоит из следующих этапов [18]:

Этап 1. Формирование отношений инцидентности $[T]$ типа «номер узла – номер ветви со знаком» для всех узлов каждого компонента. Номер ветви принимается со знаком минус, если она ориентирована к узлу, и со знаком «плюс», если от узла.

Этап 2. Построение матрицы инцидентности $[T]$ размерности $b \times n$, каждый элемент которой равен

$$T_{ij} = \begin{cases} 0, & \text{если ветвь } B_i \text{ не инцидентна узлу } N_j \\ 1, & \text{если ветвь } B_i \text{ ориентирована от узла } N_j \\ -1, & \text{если ветвь } B_i \text{ ориентирована к узлу } N_j \end{cases}. \quad (2.19)$$

Этап 3. Построение топологических уравнений для КЦ с энергетическими связями ($[T] \neq 0$)

$$[T] \cdot [V_B] = 0 \quad (2.20)$$

для всех узлов КЦ, имеющих инцидентные ветви.

Этап 4. Построение компонентных матриц (матриц линейных форм относительно потенциальных и потоковых переменных) $[A_{Mi}]$ и векторов правых ча-

стей $[W_M]$, $i = 1..q$ для всех q компонентов КЦ параллельно с линеаризацией нелинейных и алгебраизацией дифференциальных уравнений моделей компонентов.

Этап 5. Построение матричных моделей компонентов

$$[A_{Mi}] \cdot \begin{bmatrix} [V_N] \\ [V_B] \end{bmatrix} = [W_{Mi}]. \quad (2.21)$$

Этап 6. Построение общей компонентной матрицы

$$[A_M] = \left[[A_{M1}] [A_{M2}] \dots [A_{Mq}] \right]^T \quad (2.22)$$

и вектора правых частей

$$[W_M] = \left[[W_{M1}] [W_{M2}] \dots [W_{Mq}] \right]. \quad (2.23)$$

Этап 7. Построение матричной модели КЦ. Для компонентной цепи, имеющей энергетические связи, модель формируется в виде

$$\begin{aligned} V_{N_0} &= 0 \\ \left[\begin{array}{c|c} [A_M] & \\ \hline \mathbf{0} & [T] \end{array} \right] \cdot \begin{bmatrix} [V_N] \\ [V_B] \end{bmatrix} &= \begin{bmatrix} [W_M] \\ \mathbf{0} \end{bmatrix}, \end{aligned} \quad (2.24)$$

где V_{N_0} – узел отсчета, обычно $N_0 = 0$. При этом узлы КЦ и соответствующие им потенциальные переменные $[V_N]$ нумеруются автоматически, а узел отсчета КЦ указывается пользователем.

В КЦ без энергетических связей отсутствуют потоковые переменные $V_B = \emptyset$, не требуется формирование законов сохранения для потоковых переменных. По сравнению с КЦ, имеющей энергетические связи (2.24), в данном случае не требуется задание значения переменной в точке отсчета V_{N_0} . В КЦ, имеющей только информационные связи, необходимо задавать нулевой узел и значение потенциальной переменной в этом узле путем подключения в нему источника потенциальной переменной. Обобщенная схема формирования матрично-топологической модели КЦ во временной области представлена на рисунке 2.3.

Таким образом, после автоматической линеаризации нелинейных и алгебраизации дифференциальных уравнений модель КЦ в общем виде имеет вид

$$[\Phi] \cdot [V] = [W], \quad (2.25)$$

где $[\Phi]$ – матрица коэффициентов; $[V]$ – вектор-столбец решения КЦ (2.15); $[W]$ – вектор-столбец правых частей.

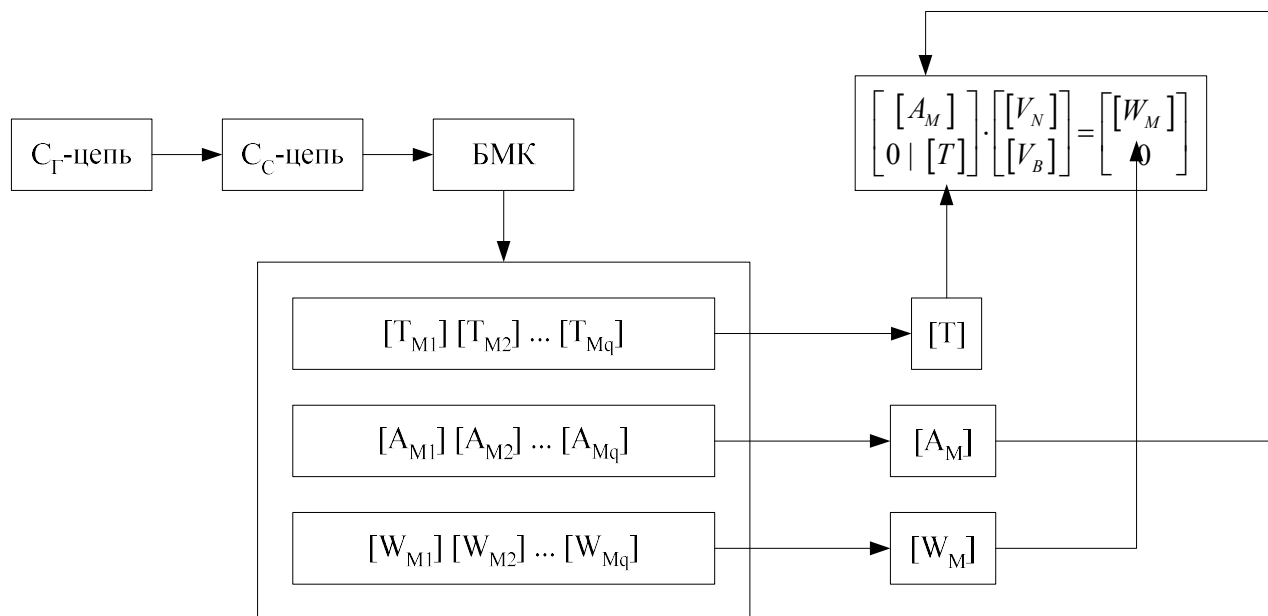


Рисунок 2.3 – Схема формирования матрично-топологической модели КЦ во временной области

Представленный на рисунке 2.3 алгоритм автоматического построения модели КЦ лежит в основе ВЭ, алгоритм которого на основе автоматически сформированной ВМ КЦ (2.21) позволяет произвести линеаризацию нелинейных и алгебраизацию входящих в нее дифференциальных уравнений и решение полученной системы линейных алгебраических уравнений (2.25) с целью получения значений переменных, входящих в вектор решения КЦ (2.15).

2.1.5 Алгоритм вычислительного эксперимента

На базе алгоритма построения модели КЦ, приведенного в пункте 2.1.4, может быть реализован *вычислительный эксперимент*. Согласно определению А.А. Самарского ВЭ можно рассматривать как метод изучения объектов, устройств и систем с протекающими в них физическими процессами с помощью средств математического моделирования [2–4, 113, 114].

ВЭ предполагает вслед за построением ММ исследуемого объекта или системы производить ее численное решение, позволяющее «проиграть» поведение

объекта в различных условиях и модификациях. Кроме того, современные средства автоматизации ВЭ должны включать развитые средства визуализации его результатов и взаимодействия с пользователем, позволяющие ему вносить топологические изменения в модель исследуемого объекта, а также интерактивно изменять значения выбранных параметров непосредственно во время проведения ВЭ.

С точки зрения реализации программно-алгоритмического аппарата, обеспечивающего автоматизацию ВЭ, он может быть определен как совокупность действий, направленных на преобразование входной информации, включающей в себя КЦ исследуемого объекта (2.3) и задание на моделирование, в выходную информацию, представленную в виде некоторых числовых значений на измерительных компонентах, либо в виде временных или частотных зависимостей.

Процесс получения выходных данных основан на решении системы алгебро-дифференциальных уравнений. При использовании различных методов линеаризации нелинейных и алгебраизации дифференциальных уравнений появляется возможность представленных в виде КЦ анализа объектов во временной, и частотной области. Последовательность действий ВЭ, представленная на рисунке 2.4, легла в основу реализации универсального вычислительного ядра [103] и состоит из следующих шагов:

Шаг 1. Формирование КЦ исследуемого объекта (системы) заключается в установке и соединении всех компонентов, являющихся моделями его элементов.

Шаг 2. Нумерация потенциальных и потоковых переменных заключается в преобразовании ЛКБ всех компонентов (2.13) в ГКБ сформированной КЦ (2.14).

Шаг 3. Установка начального значения времени (частоты) и выбор шага производится согласно выбранному режиму анализа и используемым средствам визуализации результатов.

Шаг 4. Формирование топологических уравнений вида (2.18) выполняется на основе информации о топологии цепи, полученной путем опроса топологического блока всех входящих в исследуемую КЦ компонентов. Она представляет собой данные об инцидентности ветвей каждого компонента соответствующим узлам цепи.

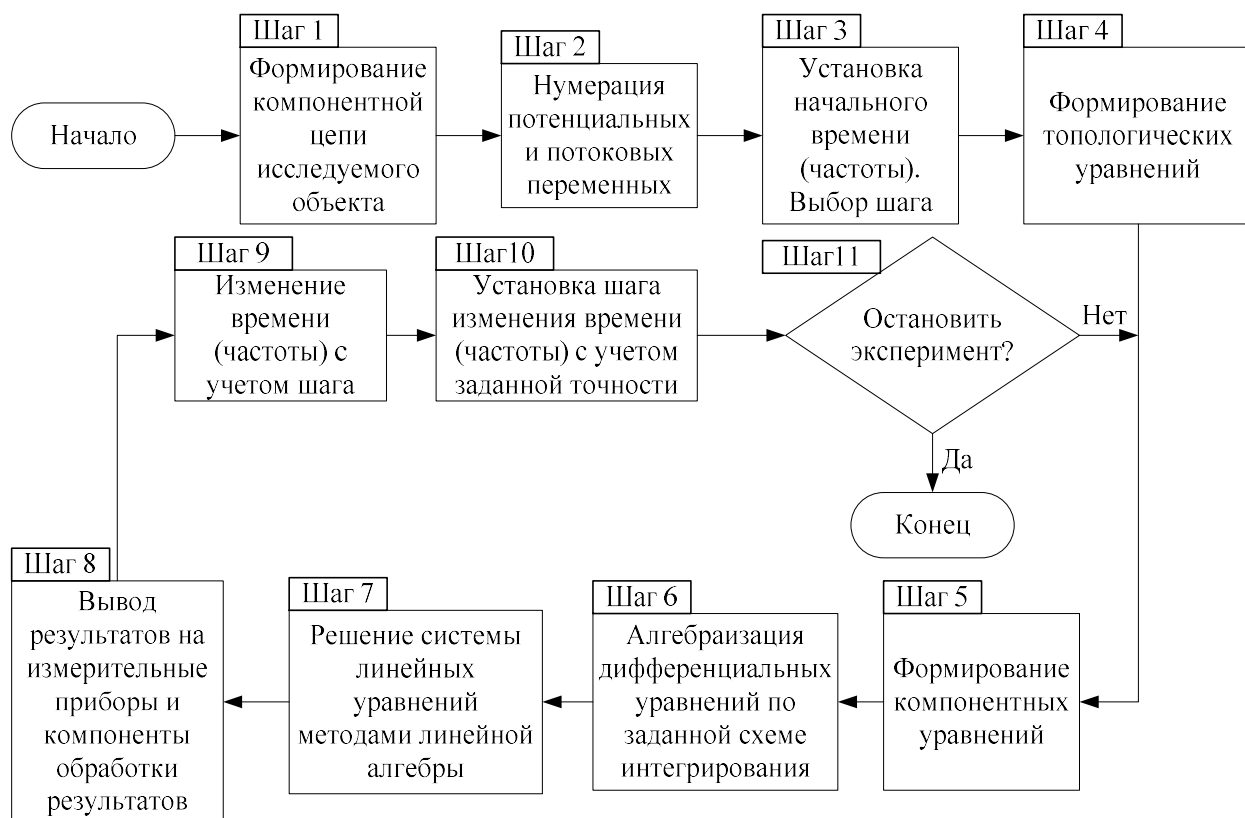


Рисунок 2.4 – Алгоритм вычислительного эксперимента над КМ технического объекта в формате МКЦ

Шаг 5. Формирование компонентных уравнений вида (2.9)-(2.12) осуществляется путем опроса вычислительного блока входящих в КЦ компонентов. На этом шаге производится линеаризация нелинейных алгебраических уравнений вида (2.10) на основе схемы Ньютона

$$\sum_i \left. \frac{\partial f}{\partial V_i} \right|_{V(t-h)} \cdot V(t) = f(V) \Big|_{V(t-h)}. \quad (2.26)$$

Шаг 6. Линеаризация дифференциальных уравнений [72] осуществляется на основе заданной схемы интегрирования. В универсальном вычислительном ядре [103], являющемся программно-алгоритмической реализацией алгоритма ВЭ, реализованы следующие схемы алгебраизации дифференциальных уравнений (2.11):

Явная схема Эйлера позволяет осуществлять преобразование дифференциального уравнения (2.11) к виду

$$\sum_i \Psi_i \cdot V_i(t+h) = \sum_i \Psi_i \cdot V_i + h \cdot f(V(t)). \quad (2.27)$$

Неявная схема Эйлера предполагает, что в результате алгебраизации уравнение (2.11) примет вид

$$\sum_i \left(\Psi_i - h \cdot \frac{\partial f(V)}{\partial V_i} \Big|_{V=V(t)} \right) \cdot V(t+h) = \sum_i \Psi_i \cdot V_i(t) + h \cdot f(V(t)). \quad (2.28)$$

Согласно неявной схеме трапеций дифференциальное уравнение (2.11) после линеаризации будет иметь вид

$$\sum_i \left(\Psi_i - \frac{h}{2} \cdot \frac{\partial f(V)}{\partial V_i} \Big|_{V=V(t)} \right) \cdot V(t+h) = \sum_i \Psi_i \cdot V_i(t) + \frac{h}{2} \cdot f(V(t)). \quad (2.29)$$

Шаг 7. Решение системы линейных алгебраических уравнений может выполняться различными методами линейной алгебры. В настоящее время в качестве основного выбран метод Жордана [115] с выбором главного элемента по строке.

Шаг 8. Вывод результатов на измерительные приборы и компоненты обработки результатов выполняется с помощью измерительных компонентов, осуществляющих выборку значений соответствующих переменных из общего решения КЦ (2.15). Обработка результатов производится основанными на численных методах алгоритмами [62]. Полученные результаты передаются на визуальные компоненты. Также они могут быть использованы при решении задач параметрической оптимизации [63].

Шаг 9. Изменение времени (частоты) производится с учетом заданного или выбранного на основе заданной точности шага.

Шаг 10. Установка шага изменения времени (частоты) выполняется с учетом заданной точности на основе результатов двух или более предыдущих итераций полного вектора решения.

Шаг 11. Проверка останова эксперимента выполняется как на основе текущего значения времени (частоты), достигнутой точности решения нелинейных уравнений при постоянных входных воздействиях или по команде останова ВЭ.

2.2 Модель сложной технической управляемой системы в формате метода компонентных цепей

2.2.1 Компьютерная модель сложной технической управляемой системы

Для построения в формате МКЦ компьютерной модели СТУС, структура которой приведена на рисунке 1.3, требуется каждому ее объекту и функциональному блоку поставить в соответствие компонент, обобщенный вид которого представлен на рисунке 2.2, или КЦ (2.3) [116]. Данная модель, основанная на методе подцепей (2.2), может быть представлена КЦ СТУС вида

$$C_{STUS} = \{C_{ChTS} \cup C_{IZ} \cup C_C \cup C_{IS} \cup C_V\} \quad (2.30)$$

где КЦ ХТС, в которой предусмотрены его взаимодействия с ИспУ и ИзмУ, имеет вид

$$C_{ChTS} = (K_{ChTS}, B_{ChTS} \cup B_{ChTSUIS} \cup B_{ChTSUIZ}, N_{ChTS} \cup N_{ChTSUIS} \cup N_{ChTSUIZ}); \quad (2.31)$$

КЦ измерительных устройств имеет вид

$$C_{IZ} = (K_{IZ}, B_{IZ}, N_{IZ}); \quad (2.32)$$

КЦ исполнительных устройств имеет вид

$$C_{IS} = (K_{IS}, B_{IS}, N_{IS}); \quad (2.33)$$

КЦ устройства управления, в которой реализуются алгоритмы его функционирования, представлена в виде

$$C_C = (K_C, B_C \cup B_{CUIZ} \cup B_{CUIZ}, N_C \cup N_{CUIZ} \cup N_{CUIZ}); \quad (2.34)$$

КЦ средств визуализации и интерактивного управления, которые, образуя множество визуальных компонентов K_V , позволяют формировать панели визуализации и интерактивного управления СТУС

$$C_V = (K_V, B_V = \emptyset, N_V = \emptyset), \quad (2.35)$$

которая называется *визуальной компонентной цепью*.

Для построения КЦ (2.30) необходимо сформировать каждую из представленных подцепей, предусмотрев при этом связи между ними $S_{ChTSUIZ}$, $S_{ChTSUIS}$, S_{CUIZ} , S_{CUIZ} . Подлежащая исследованию ХТС, между компонентами которой протекают

мультифизические энергетические и многокомпонентные вещественные потоки, может быть представлена КЦ вида (2.31), если каждому его элементу будет поставлен в соответствие компонент, являющийся его моделью в формате метода КЦ. Ей будет соответствовать векторно-матричная модель (2.24) с вектором решения (2.15).

Для отображения в модели ХТС возмущающих воздействий со стороны окружающей среды, используются компоненты-источники возмущающих воздействий, каждый из которых может быть описан обобщенной математической моделью

$$V_F = f_F(t), \quad (2.36)$$

где V_F – вектор возмущающих воздействий, отображающий в модели вектор параметров и характеристик окружающей среды; $f_F(t)$ – вектор-функция времени, характеризующая полную совокупность возмущающих воздействий.

Для измерения значений наблюдаемых переменных $Y = [Y_S, Y_E, Y_I]$, носящих энергетический и вещественный характер, с последующим их преобразованием в информационные сигналы $y = [y_S, y_E, y_I]$, в КЦ СТУС включаются измерительные компоненты множества K_{IZ} . Каждый из них позволяет измерять значения либо одной потенциальной $V_{Ni} \subset V_N$, либо одной потоковой переменной $V_{Bi} \subset V_B$ из вектора переменных цепи (2.15). Обобщенная модель каждого измерительного компонента множества K_{IZ} может быть представлена в виде

$$f_Z(V_i) - V_Z = 0, \quad (2.37)$$

где V_i – потенциальная или потоковая переменная, принадлежащая компоненту $K_{IZ} \subset C_{ХТС}$.

Переменная V_i КЦ $C_{ХТС}$ входит в вектор измеряемых переменных ХТС Y : $V_i \in Y$; $V_Z \in y$ – вектор результатов функции измерения, причем $V_Z \notin V$; f_Z – функция измерения, реализуемая в компоненте K_{IZ} . Она в общем случае может иметь сложный нелинейный характер, предполагающий как непосредственное измерение текущих значений, так и их обработку, основанную на численных методах

анализа данных и записанную либо в аналитическом, либо в алгоритмическом виде.

Измеренные в модели текущие значения наблюдаемых переменных ХТС V_Z используются в функциональной модели УУ, которая представлена совокупностью компонентов-контроллеров $K_C \notin K_{ChTS}$, предназначенных для построения КМ устройства управления C_C в формате метода КЦ. Эти компоненты могут быть описаны ММ вида

$$f_u(V_Z) - V_u = 0, \quad (2.38)$$

где $V_u \notin V_{UTO}$ – вектор информационных сигналов, отображающий в КЦ СТУС информационные сигналы u ; f_u – в общем случае нелинейная функция преобразования результатов измерений V_Z в информационные сигналы управления V_u , вырабатываемые УУ и посылаемые им на ИспУ.

Исполнительным устройствам в КЦ СТУС (2.30) ставятся в соответствие компоненты множества K_{IS} , каждый из которых описывается обобщенной математической моделью

$$f_{IS}(V_u) - V_U = 0, \quad (2.39)$$

где V_U – вектор управляющих воздействий на ХТС, в который входят управляющие воздействия, направленные на прямые изменения значений его параметров; $f_{IS}(V_u)$ – функция преобразования управляющих информационных сигналов V_u в энергетические воздействия на ХТС V_U .

2.2.2 Вычислительная модель СТУС

Математическая модель СТУС, сформированная в формате метода КЦ (2.30), может быть представлена в виде

$$M_{STUS} = \left[\bigcup_i M_{ChTSi} \cup \bigcup_j M_{IZj} \cup \bigcup_k M_{ISk} \cup \bigcup_l M_{Cl} \right], \quad (2.40)$$

где M_{ChTSi} – математические модели компонентов множества K_{ChTS} (2.31) в совокупности с математическими моделями узлов $N_{ChTS} \cup N_{ChTS \cup IS} \cup N_{ChTS \cup IZ}$, уравнения ко-

торых составлены согласно основному топологическому закону (2.20); M_{Iz} – ММК множества K_{Iz} ; M_{Isk} – ММК множества K_{Is} ; M_{Cl} – ММК множества K_C , представляющая собой модель УУ.

Математической модели СТУС (2.40) соответствует сформированная алгоритмом построения модели цепи (п. 2.1.4) векторно-матричная модель вида

$$\begin{bmatrix} \Phi_{KN}^{ChTS} & \Phi_{KB}^{ChTS} & 0 & 0 & \Phi_{S2}^{IS} \\ 0 & \Phi_T^{ChTS} & 0 & 0 & 0 \\ \Phi_{ZN}^{ChTS} & \Phi_{ZB}^{ChTS} & \Phi_{Z1}^{IZ} & 0 & 0 \\ 0 & 0 & \Phi_{Z2}^{IZ} & \Phi_{U1}^C & 0 \\ 0 & 0 & 0 & \Phi_{U2}^C & \Phi_{S1}^{IS} \end{bmatrix} \cdot \begin{bmatrix} V_N^{ChTS} \\ V_B^{ChTS} \\ V_Z^{IZ} \\ V_u^C \\ V_U^{IS} \end{bmatrix} = \begin{bmatrix} W_K^{ChTS} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (2.41)$$

где $\Phi_{KN}^{ChTS}, \Phi_{KB}^{ChTS}$ – матрицы коэффициентов при потенциальных и потоковых переменных в уравнениях (2.9)-(2.12) компонентов множества K_{ChTS} ; Φ_{S2}^{IS} – коэффициенты при переменных исполнительных устройств множества K_{Is} в уравнениях компонентов множества K_{UO} ; Φ_T – матрица коэффициентов топологических уравнений (2.20), составленных для узлов множества N_{ChTS} ; $\Phi_{ZN}^{ChTS}, \Phi_{ZB}^{ChTS}$ – матрицы коэффициентов в ММ (2.37) компонентов множества K_{Iz} при потенциальных V_N^{ChTS} и потоковых переменных V_B^{ChTS} ; Φ_{Z1}^{IZ} – матрица коэффициентов в моделях (2.37) при переменных V_Z^{IZ} измерительных компонентов; Φ_{Z2}^{IZ} – матрица коэффициентов в уравнениях компонентов УУ множества C_C вида (2.38) при измерительных переменных V_Z^{IZ} ; Φ_{U1}^C – матрица коэффициентов в уравнениях (2.38) компонентов множества C_C при переменных V_u^C , характеризующих подаваемых на ИспУ информационные сигналы; Φ_{U2}^C – матрица коэффициентов в ММ исполнительных устройств вида (2.39) при информационных переменных V_u^C ; Φ_{S1}^{IS} – матрица коэффициентов в моделях ИспУ (2.39) при переменных V_U^{IS} , выражающих непосредственные энергетические воздействия ИспУ на ХТС.

Построенная в формате метода КЦ компьютерная модель СТУС (2.30) с несколькими наблюдаемыми переменными ХТС $V_Z^{IZ} = y$, выраженными потенци-

альными и потоковыми переменными его модели, управляемой с помощью нескольких управляющих воздействий $V_U^{IS} = U$, описывается в общем случае системой нелинейных уравнений. Если ХТС удастся описать моделью линейных алгебро-дифференциальных уравнений, то в общую модель СТУС (2.40) нелинейность будут вносить модели других ее блоков: измерительных и исполнительных устройств, а также устройства управления.

Чтобы повысить эффективность процесса автоматизированного исследования и функционального проектирования СТУС требуется упростить представленную модель СТУС (2.40)-(2.41). Для этого необходимо выделить из нее блоки, допускающие представление в явной форме, и реализовать аппарат обмена информацией между этими блоками, не прибегая к построению и решению их единой системы уравнений.

2.2.3 Принцип разделения уравнений в модели СТУС

В силу явного вида уравнений (2.37)-(2.39) матрицы $\Phi_{Z1}^{ZZ}, \Phi_{U1}^C, \Phi_{S1}^{IS}$ векторно-матричной модели (2.41) будут иметь вид отрицательной диагональной единичной матрицы

$$-I = \begin{bmatrix} -1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -1 \end{bmatrix}. \quad (2.42)$$

Из полученного на основе такой матрицы матричного уравнения

$$\Phi \cdot V_1 - I \cdot V_2 = 0 \quad (2.43)$$

элементы вектора V_2 могут быть выражены в явном виде

$$V_2 = \Phi \cdot V_1 \cdot (-I)^{-1}. \quad (2.44)$$

Матричное уравнение (2.44) может быть представлено совокупностью независимых друг от друга алгебраических уравнений, которые могут быть нелинейными. Каждое из них позволяет определять конкретный элемент вектора V_2 . В этом случае более эффективным является представление уравнений (2.37)-(2.39) в явном виде (2.44) и реализация математических выражений для вычислений зна-

чений элементов векторов $V_Z^{IZ}, V_u^C, V_U^{IS}$ непосредственно в компонентах множеств K_{IZ}, K_C, K_{IS} (2.30). Для обмена информацией между компонентами этих множеств и множества K_{CHTS} , анализ которых производится универсальным вычислительным ядром [103] на основе алгоритма ВЭ, представленного на рисунке 2.4, может быть применен способ обмена информацией, основанный на формировании, пересылке и обработке сообщений с данными различных типов. Этот способ относится к классу методов имитационного моделирования [86, 87] и реализован в виде алгоритма передачи сообщений [116], аналогичного интерфейсу передачи данных между компьютерами, образующими один кластер при решении задач параллельного программирования [117].

Математические модели ХТС, представленные в формате КЦ (2.31), взаимосвязанные с ними имитационные модели измерительных (2.37), исполнительных устройств (2.39) и устройства управления (2.38) совместно с компонентами визуализации и интерактивного изменения значений параметров компонентов множества K_V , входящие в общую КЦ СТУС, будут образовывать ее многоуровневую компонентную цепь (МКЦ). Она может быть построена и проанализирована в рамках развития метода компонентных цепей и его адаптации к моделированию сложных технических (технологических) управляемых систем на основе методики стратификации и интеграции компонентной цепи СТУС [118].

2.3 Формирование многоуровневых компонентных цепей сложных технических управляемых систем

2.3.1 Многоуровневая компонентная цепь СТУС

Развитие метода КЦ [17], проведенное в данной диссертационной работе, представляет собой его расширение на класс сложных технических управляемых систем. В управляемых ими химико-технологических системах протекают мультифизические энергетические и многокомпонентные вещественные потоки. Являясь универсальным методом компьютерного моделирования, метод КЦ после его развития позволяет в единой многоуровневой компонентной цепи (МКЦ) совме-

щать КЦ различных видов моделирования (математического, имитационного и визуального), представляя их различными подцепями в рамках МКЦ вида

$$C_M = (C_V \cup C_A \cup C_O), \quad (2.45)$$

где C_V – КЦ средств визуализации результатов моделирования, а также средств интерактивного управления значениями параметров модели ХТС (2.35); C_A – КЦ сценариев эксперимента, включающая в свой состав имитационную (функциональную) модель устройства управления СТУС и состоящая из компонентов множества K_C (2.34). Называемая далее *алгоритмической компонентной цепью*, она может быть представлена в графическом компонентном формате в виде алгоритмических подцепей

$$C_A = (C_{A1} \cup C_{A2} \cup \dots \cup C_{An}), \quad (2.46)$$

где $C_{A1}, C_{A2}, \dots, C_{An}$ – визуально несвязанные между собой алгоритмические КЦ, каждая из которых представляет собой набор взаимосвязанных компонентов, называемых далее *алгоритмическими компонентами*. Для них справедливо $C_{Ai} \cap C_{Aj} = \emptyset$ для любых $i = 1..n, j = 1..n, i \neq j$. Каждая алгоритмическая КЦ C_i , представленная в графической форме в виде совокупности взаимосвязанных компонентов, предназначена для выполнения некоторого шага или набора шагов алгоритма автоматизированного эксперимента, представленного на рисунке 1.4;

C_O – компонентная цепь ХТС (2.31) с подключенными к ней компонентами ИзМУ (2.32) и ИспУ (2.33), осуществляющие ее согласование с алгоритмической компонентной цепью устройства управления C_A .

2.3.2 Методика формирования многоуровневых компонентных цепей СТУС

Многоуровневая компонентная цепь СТУС (2.45) может быть представлена графически в виде трех взаимосвязанных уровней [118]. Она включает в себя КЦ ХТС и средства подключения к реальной химико-технологической системе, имитационную модель сценария вычислительного эксперимента, в которую входит

функциональная модель устройства управления СТУС, а также взаимосвязанную с ними панель визуализации и интерактивного управления.

Под *уровнем представления модели (уровнем моделирования)* будем понимать степень абстракции, способ представления и обработки информации об исследуемой СТУС, ее моделях, методах обработки данных, средствах визуализации промежуточных и итоговых результатов экспериментов.

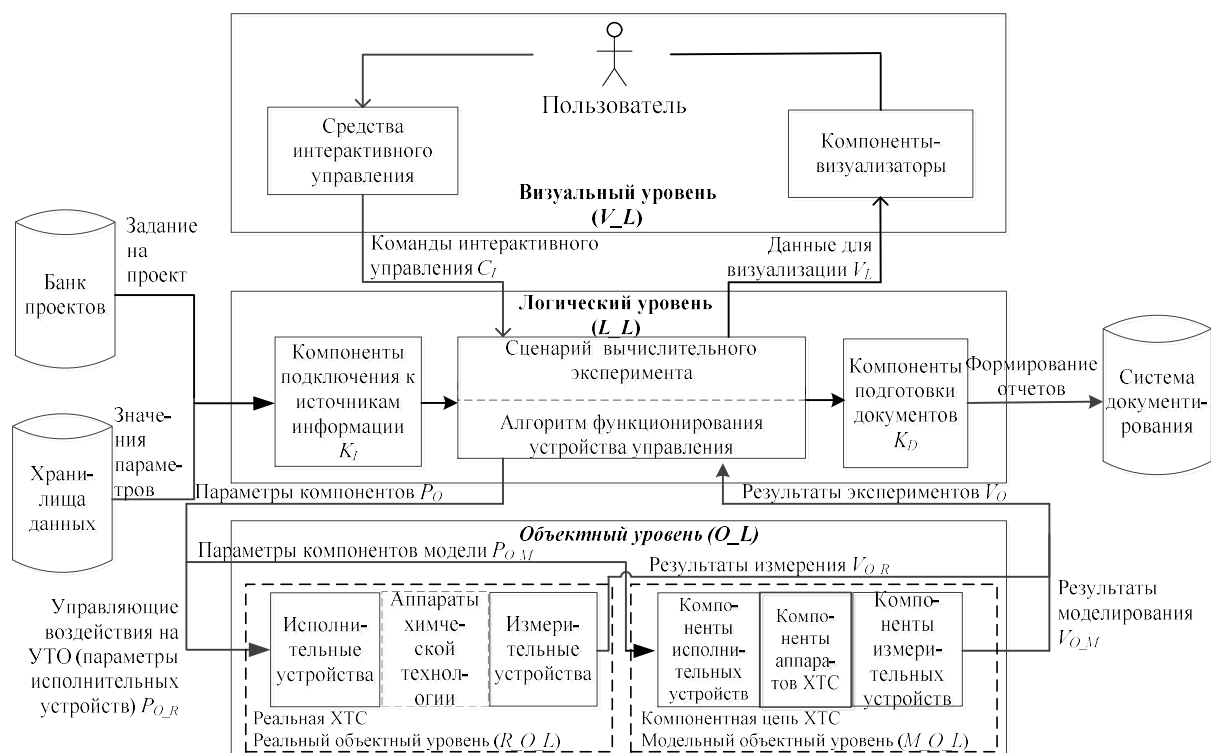


Рисунок 2.5 – Структура многоуровневой компонентной цепи СТУС

Многоуровневая КЦ СТУС, предназначенная для выполнения автоматизированных экспериментов научно-исследовательского, проектного и учебного назначения, представлена на рисунке 2.5 и состоит из следующих взаимосвязанных уровней:

– *объектного уровня* (O_L), включающего в себя компонентную цепь СТУС O_M , состоящая из КМ ХТС C_{ChTS} (2.31) с подключенными к ней компьютерными моделями ИзмУ C_{IZ} (2.32) и ИспУ C_{IS} (2.33), а также реальную ХТС O_R , на которой апробируются результаты, полученные с помощью КЦ. Представление КЦ ХТС производится на *модельном объектном уровне* (M_O_L) – $O_M \in M_O_L$. Реальная химико-технологическая система совместно с платами

сбора данных и измерительными контроллерами образуют *реальный объектный уровень* (R_O_M) – $O_R \in R_O_M$;

– *логического уровня* (L_L), на котором в виде алгоритмической КЦ C_A (2.46) отображается сценарий исследования и функционального проектирования СТУС, интегрированный со сценарием функционирования устройства управления. Её работа основана на использовании результатов анализа модели ХТС V_{O_M} и данных измерения реальной системы V_{O_R} , передаваемых с объектного уровня. Алгоритмическая КЦ сценария включает в себя команды параметризации КЦ ХТС, математические средства анализа, обработки результатов моделирования и измерения, их визуализации, а также средства принятия управленческих решений, направленных на изменение значений параметров компонентов КЦ ХТС и управляющих воздействий на реальную систему. Каждая команда сценария исследования и функционального проектирования СТУС должна быть представлена алгоритмической КЦ $C_A \in L_L$, принадлежащей логическому уровню, графически в виде набора компонентов, входные связи каждого из которых соответствуют аргументам описываемой им команды, а выходные – результатам ее выполнения. Ее работа основана на рассматриваемом далее алгоритме передачи сообщений [119], относящемся к классу алгоритмов ИМ или ДСМ;

– *визуального уровня* (V_L) отображения информации о текущих результатах выполнения шагов сценария исследования и функционального проектирования СТУС, включающего сценарий функционирования УУ. На данном уровне реализуется *панель визуализации и интерактивного управления*, представленная визуальной КЦ C_V (2.35). Она включает в себя *компоненты-визуализаторы*, предназначенные для отображения пользователю подлежащих визуализации данных V_L , а также средства интерактивного управления, называемые *компонентами-регуляторами*, и позволяющие формировать соответствующие команды управления C_I . Команды-визуализаторы и компоненты-регуляторы образуют множество K_V (2.35).

С целью первичной параметризации КМ СТУС (2.30) значения элементов реальной ХТС (2.31) и взаимосвязанных с ней ИзМУ (2.32) и ИспУ (2.33) должны быть доставлены в формируемую на логическом уровне модель сценария исследования и проектирования СТУС C_C (2.34) из внешних информационных источников (баз данных, баз знаний и т.п.), называемых далее *хранилищами данных*. Средства подключения к ним K_I принадлежат логическому уровню $K_I \in L_L$. Помимо этого модели реальных элементов СТУС $P_{O_R}^0$ и ее модели $P_{O_M}^0$, образующие вектор начальных значений параметров компонентов

$$P_O = [P_{O_R}, P_{O_M}] \quad (2.47)$$

могут быть рассчитаны с помощью средств компьютерной математики. Их значения должны передаваться с логического уровня на объектный

$$P_O : L_L \rightarrow O_L. \quad (2.48)$$

После их установки в модели СТУС запускается процесс решения задачи, представленной своим сценарием на логическом уровне МКЦ. В случае решения задачи слежения за состоянием реального объекта одновременно с запуском модели начинается процесс наблюдения за характеристиками функционирующего реального объекта.

Результаты анализа V_{O_M} КМ СТУС O_M , принадлежащей модельному объектному уровню $O_M \in M_O_L$, и (или) измерения V_{O_R} наблюдаемых характеристик реального объекта O_R , составляющего реальный объектный уровень $O_R \in R_O_L$, образуют вектор результатов моделирования и измерения

$$V_O = \begin{bmatrix} V_{O_M} \\ V_{O_R} \end{bmatrix}. \quad (2.49)$$

Сформированный на объектном уровне, он подлежит передаче на логический уровень МКЦ

$$V_O : O_L \rightarrow L_L, \quad (2.50)$$

где производится его математическая обработка. Ее целью является определение параметров-функционалов временных, частотных и (или) параметрических харак-

теристик, а также подготовка полученных результатов, образующих вектор V_L (см. рисунок 2.5) к визуализации.

Вектор V_L , подготовленный на логическом уровне, передается на визуальный для визуализации содержащихся в нем значений пользователю

$$V_L : L_L \rightarrow V_L . \quad (2.51)$$

Содержащиеся в нем данные с помощью средств визуализации должны быть отображены в виде статических значений, различных функциональных зависимостей и их параметров-функционалов, полученных в результате применения численных методов анализа. Кроме них визуальному уровню принадлежат компоненты-регуляторы значений параметров компонентов КЦ ХТС и управляющими воздействиями на реальную систему, на которой осуществляется апробация результатов экспериментов, проведенных на компьютерных моделях. Эти инструменты позволяют передавать команды интерактивного управления C_I с визуального на логический уровень

$$C_I : V_L \rightarrow L_L . \quad (2.52)$$

Наряду с результатами моделирования и измерения V_O команды C_I используются на логическом уровне многоуровневой модели для расчета новых значений параметров компонентов

$$P_O = \begin{bmatrix} P_{O_M} \\ P_{O_R} \end{bmatrix}, \quad (2.53)$$

которые передаются с логического на объектный уровень

$$P_O : L_L \rightarrow O_L . \quad (2.54)$$

Они используются для параметризации компонентов КЦ ХТС O_M , в которую помимо моделей элементов и аппаратов химической промышленности включены модели измерительных и исполнительных устройств. Также вектор P_O содержит значения управляющих воздействий на реальные ИспУ, которые воздействуют на реальную ХТС O_R .

Результаты, получаемые на различных этапах работы представленной на логическом уровне многоуровневой КЦ модели сценария исследования и функцио-

нального проектирования СТУС, с помощью компонентов множества K_D могут быть переданы в систему документирования [120, 121]. Она позволяет формировать документы, представляющие собой отчеты об исследованиях, проводимых в рамках МКЦ СТУС.

Представленная на рисунке 2.5 структура многоуровневых компонентных цепей сложных технических (технологических) управляемых систем является развитием метода КЦ. Она позволяет формировать многоуровневые КЦ СТУС (2.45) в графическом виде. С ее помощью при построении соответствующих алгоритмов будет автоматизирован процесс исследования и функционального проектирования СТУС предприятий газовой промышленности. Кроме того, разработанные в ее формате многоуровневые КЦ позволяют реализовать разнообразные виртуальные и реально-виртуальные лаборатории, а также компьютерные тренажеры, направленные на обучение и переподготовку управляющего персонала ХТС. Ход и результаты выполнения определенных шагов сценариев эксперимента, представленного на рисунке 1.4, могут быть оформлены в виде документов различных текстовых редакторов (*MS Word, Latex, PDF* и других).

Формирование многоуровневых КЦ СТУС осуществляется пользователем на языке многоуровневых компонентных цепей (язык МКЦ), в котором реализуются функции построения и преобразования во внутримашинное представление моделей каждого уровня многоуровневой КЦ СТУС с целью из последующей обработки определенными программно-алгоритмическими средствами.

2.3.3 Основы языка многоуровневых компонентных цепей

Язык многоуровневых компонентных цепей, являясь графическим языком, предназначен для формирования многоуровневых КЦ СТУС в графическом (не текстовом) формате. Он объединяет в себе три языка, являющихся его подязыками:

- 1) язык моделирования химико-технологических систем (язык ХТС), позволяющий на объектном уровне МКЦ СТУС представить компонентные цепи химико-технологических систем с подключенными к ним компонентами ИзмУ и

ИспУ в графическом виде. КЦ ХТС должны адекватно отображать протекающие в управляемых ХТС процессы преобразования мультифизических энергетических и многокомпонентных вещественных потоков, находящихся под управлением информационных сигналов устройства управления;

2) *язык моделирования алгоритмических конструкций (язык МАК)*, являющийся расширением языка компьютерного моделирования математических выражений [101, 102, 122]. Он предназначен для построения моделей сценариев исследования и проектирования СТУС, включающих сценарии функционирования устройств управления. Эти модели принадлежат логическому уровню многоуровневой КЦ СТУС. Являясь графическим языком представления сценариев, его выражения могут быть представлены в традиционном текстовом или математическом представлении;

3) *язык построения виртуальных инструментов и приборов (язык ВИП)*, являющийся подязыком языка МКЦ, предназначенным для формирования и функционирование виртуальных инструментов и приборов [27, 108]. К их числу относятся панели визуализации и интерактивного управления, представленные на визуальном уровне многоуровневой КЦ СТУС.

Формально язык МКЦ можно представить в виде объединения перечисленных языков

$$\text{Язык МКЦ} = (\text{Язык ХТС}) \cup (\text{Язык МАК}) \cup (\text{Язык ВИП}). \quad (2.55)$$

Пересечениями пар подязыков языка МКЦ образуются межуровневые взаимодействия:

1. При пересечении языка ХТС и языка МАК образуются межуровневые взаимодействия (2.48) и (2.50):

$$(\text{Язык ХТС}) \cap (\text{Язык МАК}) = (P_{STUS}^0 : L_L \rightarrow O_L) \cup (V_o : O_L \rightarrow L_L); \quad (2.56)$$

2. Пересечением языка МАК и языка ВИП являются межуровневые взаимодействия (2.51) и (2.52):

$$\begin{aligned}
 (\text{Язык МАК}) \cap (\text{Язык ВИП}) = \\
 (V_L : L_L \rightarrow V_L) \cup (C_I : V_L \rightarrow L_L) .
 \end{aligned}
 \tag{2.57}$$

Развитие метода КЦ включает в себя методики построения многоуровневых КЦ (2.45) СТУС, их обобщенную структуру, представленную на рисунке 2.5, и язык МКЦ (2.55). Его программно-алгоритмической реализацией является комплекс программ «Среда моделирования химико-технологических систем». Он позволяет автоматизировать задачи исследования и функционального проектирования СТУС, а также позволяющий формировать интеллектуальные системы управления со встроенными в них компьютерными моделями ХТС. Такой комплекс программ включает в себя необходимые программно-инструментальные средства, интерпретирующие графическое представление моделей, расположенных на различных уровнях многоуровневой КЦ СТУС, во внутримашинные представления, соответствующие определенным подязыкам языка МКЦ (2.55).

Для создания такой модели в графическом виде в комплексе программ моделирования ХТС реализован многослойный редактор, на каждом слое которого отображается соответствующий уровне многоуровневой КЦ СТУС:

- На *визуальном слое* формируется принадлежащая визуальному уровню панель визуализации и интерактивного управления, состоящая из визуальных компонентов отображения результатов моделирования и данных измерения, а также компонентов интерактивного управления значениями параметров КЦ ХТС и управляющих воздействий на реальную систему.

- *Логический слой* многослойного редактора соответствует логическому уровню многоуровневой КЦ СТУС. На нем в виде алгоритмической КЦ C_A (2.46) разрабатываются сценарии исследования и функционального проектирования СТУС, включающие сценарии функционирования УУ. В него также входят сценарии взаимодействия МКЦ с реальными объектами, реализующие операции (2.50) и (2.54). Помимо этого на логическом слое редактора отображение находят компоненты визуального уровня, которые принимают подлежащие визуализации данные (2.51) и передают в алгоритмы результаты интерактивного воздействия пользователя на многоуровневую КЦ СТУС (2.52);

– На *объектном слое*, соответствующем объектному уровню многоуровневой КЦ СТУС, располагается компонентная цепь ХТС с входящими в нее компонентами ИзМУ и ИспУ (2.31).

Для построения соответствующих модулей комплекса программ моделирования ХТС рассмотрим основы каждого из подязыков языка многоуровневого компьютерного моделирования.

2.4 Графические нотации языка моделирования

химико-технологических систем

В первой главе сформирована представленная на рисунке 1.1 структурно-функциональная схема сложной технической (технологической) управляемой системы, предназначенной для управления химико-технологическими системами, допускающими декомпозицию на элементы и аппараты химической промышленности, по связям между которыми протекают мультифизические энергетические и многокомпонентные вещественные потоки. Ее основными структурными элементами являются: исполнительные устройства – источники энергии и источники многокомпонентных вещественных потоков; управляемая ХТС, включающая преобразователи энергии и веществ, а также измерительные устройства соответствующих классов. Для формирования КЦ химико-технологической системы каждому ее элементу ставится в соответствие графическое отображение на объектном слое многослойного редактора КЦ СТУС, которое является геометрическим аспектом данного компонента. Далее рассмотрим классы компонентов языка моделирования ХТС.

2.4.1 Компоненты-источники

В качестве *источника энергии* могут выступать источник потенциальной переменной (источник потенциала), изображенный на рисунке 2.6, а, источник разности потенциальных переменных, приведенный на рисунке 2.6, б, и источник потоковой переменной, представленный на рисунке 2.6, в.

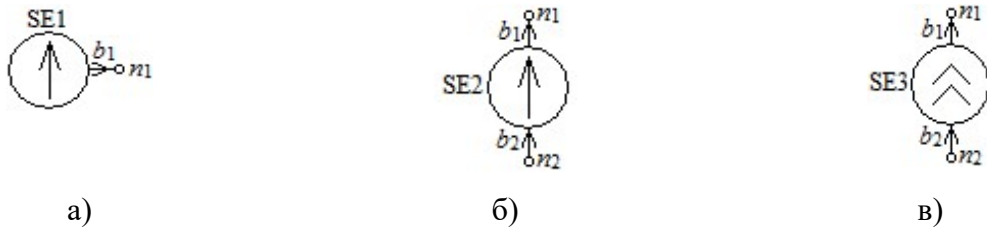


Рисунок 2.6 – Источники энергии
 а) источник потенциала;
 б) источник разности потенциалов;
 в) источник потока

В качестве источника потенциала, представленного на рисунке 2.6, может выступать источник скорости, давления, температуры, электрического или магнитного потенциала. Данный источник имеет единственную выходную связь

$$S_1 = (b_1 \eta^+ n_1) \rightarrow \{V_{n1}, V_{b1}\} \quad (2.58)$$

и описывается математической моделью вида

$$V_{n1} = E_n, \quad (2.59)$$

где E_n – значение потенциальной переменной.

Источники разности потенциалов в компонентных цепях представляют насосы, нагреватели, источники напряжения и т.п. Они имеют одну входную $S_1 = (b_1 \eta^- n_1) \rightarrow \{V_{n1}, V_{b1}\}$ и одну выходную связи $S_2 = (b_2 \eta^+ n_2) \rightarrow \{V_{n2}, V_{b2}\}$. Математическая модель источника разности потенциалов включает в себя следующие уравнения

$$\begin{aligned} V_{n1} - V_{n2} &= \Delta E_n \\ V_{b2} - V_{b1} &= 0 \end{aligned}, \quad (2.60)$$

где ΔE_n – разность потенциальных переменных.

Источник потоковой переменной, обладая входной $S_1 = (b_1 \eta^- n_1) \rightarrow \{V_{n1}, V_{b1}\}$ и выходной связями $S_2 = (b_2 \eta^+ n_2) \rightarrow \{V_{n2}, V_{b2}\}$, будет иметь математическую модель вида

$$\begin{aligned} V_{b1} &= I_b \\ V_{b1} - V_{b2} &= 0 \end{aligned}, \quad (2.61)$$

где I_b – значение потоковой переменной, которое может выражать скорость вещества или теплового потока, электрический ток, магнитный поток и т.п.

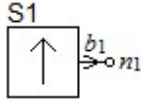


Рисунок 2.7 – Источник
вещественного потока

Источником вещественного потока (источником вещества), изображенным на рисунке 2.7, называется компонент, отражающий подачу в моделируемую ХТС многокомпонентной смеси веществ. В общем случае она

будет характеризоваться давлением (объемным расходом), температурой (тепловым потоком), а также вектором концентраций входящих в ее состав веществ (компонентов вещества). Образованная единственным выходом компонента связь будет иметь структуру

$$S_1 = (b_1 \eta^+ n_1) \rightarrow \{\{P_1, G_1\}, \{T_1, Q_1\}, \{C_1\}\}, \quad (2.62)$$

где P_1 – давление многокомпонентной вещественной смеси; G_1 – вещественный поток смеси в связи S_1 ; T_1 – температура смеси в узле n_1 ; Q_1 – тепловой поток в связи S_1 ; C_1 – вектор концентраций веществ в потоке.

Математическая модель источника вещества включает в себя уравнение задания давления или вещественного потока, уравнение задания температуры (теплового потока), а также совокупность уравнений формирования вектора концентраций веществ, составляющих многокомпонентный поток. В данном случае связь S_1 может быть представлена совокупностью двух энергетических связей S_{G_1} и S_{T_1} , а также однородной векторной связью S_{C_1} , размер которой определяется количеством веществ, входящих в многокомпонентный поток. Моделирование объектов с такими связями в формате метода КЦ может быть проведено путем искусственного расщепления одной векторной связи S_1 на совокупность энергетических (2.4) и информационных (в том числе однородных векторных) связей (2.5). Но в этом случае полученная КЦ будет иметь виртуальные (не отражающие непосредственно объекта) связи, усложняющие процесс формирования КЦ исследуемой ХТС.

Для устранения этого недостатка в работе рассмотрена методика и реализованы алгоритмы компьютерного моделирования КЦ ХТС с неоднородными векторными связями [123], которые рассмотрены в третьей главе диссертации. Они позволяют описывать процессы функционирования ХТС, между элементами ко-

торых протекают мультифизические энергетические и многокомпонентные вещественные потоки.

Математические модели источника энергии (2.59)-(2.61) могут быть сведены к виду модели источников (2.12) в формате метода КЦ. При реализации методики представления моделей объектов с неоднородными векторными связями открываются возможности представления моделей источников вещественного потока в виде совокупности уравнений источников (2.12).

2.4.2 Компоненты-преобразователи

Управляемая химико-технологическая система совместно с УУ образуют сложную техническую (технологическую) управляемую систему, модель которой в формате МКЦ строится на основе структурно-функциональной схемы, представленной на рисунке 1.1. На ее объектном уровне формируется модель ХТС с взаимодействующими с ним исполнительными и измерительными устройствами. Она допускает декомпозицию на взаимосвязанные преобразователи энергии и вещества (вещественных потоков).

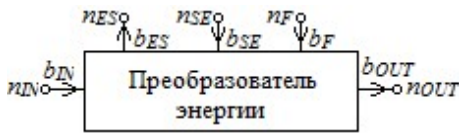


Рисунок 2.8 – Обобщенный компонент «Преобразователь энергии»

Модели преобразователей энергии, входящие в состав ХТС, реализуются на основе обобщенного компонента «Преобразователь энергии», представленного на рисунке 2.8. Его связями являются:

$$- S_{IN} = (b_{IN} \eta^- n_{IN}) \rightarrow \{V_{n_IN}, V_{b_IN}\} - \text{связь, от-}$$

ражающая поступающую в компонент энергию определенной физической природы;

$$- S_{OUT} = (b_{OUT} \eta^+ n_{OUT}) \rightarrow \{V_{n_OUT}, V_{b_OUT}\} - \text{связь, выражающая передачу рас-}$$

считываемым компонентом энергии определенной физической природы другим подключенным в этой связи компонентам;

$$- S_{ES} = (b_{ES} \eta^+ n_{ES}) \rightarrow \{V_{n_ES}, V_{b_ES}\} - \text{связь, выражающая энергетическое воз-}$$

действие рассматриваемого компонента на один или несколько преобразователей вещества;

– $S_{SE} = (b_{SE} \eta^- n_{SE}) \rightarrow \{V_{n_{SE}}, V_{b_{SE}}\}$ – связь для отображения в модели энергетического воздействия компонента на процессы, протекающие в преобразователях вещества;

– $S_F = (b_F \eta^- n_F) \rightarrow \{V_{n_F}, V_{b_F}\}$ – связь для отображения в модели энергетического влияния окружающей среды на протекающие в компоненте процессы.



Рисунок 2.9 – Обобщенный компонент «Преобразователь вещества»

Для представления на объектном уровне многоуровневой КМ СТУС преобразователей вещества должны быть разработаны соответствующие компоненты на основе обобщенного компонента «Преобразователь вещества», представленного на рисунке 2.9. Его связями являются:

– $S_{IN} = (b_{IN} \eta^- n_{IN}) \rightarrow \{\{P_{IN}, G_{IN}\}, \{T_{IN}, Q_{IN}\}, \{C_{IN}\}\}$ – входная физико-химическая связь вида (1.1), отражающая характеристики многокомпонентного вещественного потока, поступающего в преобразователь вещества;

– $S_{OUT} = (b_{OUT} \eta^+ n_{OUT}) \rightarrow \{\{P_{OUT}, G_{OUT}\}, \{T_{OUT}, Q_{OUT}\}, \{C_{OUT}\}\}$ – выходная физико-химическая связь (1.1), характеризующая выходящий из преобразователя вещества многокомпонентный вещественный поток;

– $S_{SE} = (b_{SE} \eta^+ n_{SE}) \rightarrow \{V_{n_{SE}}, V_{b_{SE}}\}$ – выходная связь, выражающая энергетическое воздействие вещественного потока на процессы, протекающие в преобразователе энергии;

– $S_{ES} = (b_{ES} \eta^- n_{ES}) \rightarrow \{V_{n_{ES}}, V_{b_{ES}}\}$ – входная связь, выражающая воздействие преобразователя энергии на процессы, протекающие в рассматриваемом преобразователе вещества;

– $S_F = (b_F \eta^- n_F) \rightarrow \{\{P_F, G_F\}, \{T_F, Q_F\}, \{C_F\}\}$ – входная связь, выражающая вещественное влияние окружающей среды на процессы, протекающие в преобразователях вещества.

Большинство связей обобщенного компонента «Преобразователь вещества» являются физико-химическими (1.1). Каждая из них может быть представлена совокупностью нескольких энергетических (2.4) и одной однородной векторной связью (2.6), представляющей собой набор информационных связей скалярного типа.

В настоящее время в рамках метода КЦ разработаны методики и алгоритмы моделирования технических объектов и устройств с энергетическими и информационными связями, а также допускается моделирование информационно-энергетических систем. К объектам с энергетическими связями (2.4) относятся электрические цепи и электронные устройства [23]; механические (динамические) системы [100], на связях которых учитываются силовые потоки; гидравлические и термодинамические объекты; электромеханические системы и устройства преобразовательной техники. Объектами с информационными (скалярными (2.5) и однородными векторными (2.6)) связями являются кинематические преобразователи [18], системы автоматического управления [22], а также скалярные и векторно-матричные математические выражения [101], используемые для формирования новых моделей компонентов и расчета значений выражений [30].

Для моделирования ХТС, в которых преобладают многокомпонентные вещественные потоки, введены *неоднородные векторные связи* [123]. Каждая из них должна представлять собой произвольную совокупность энергетических и информационных связей, что позволит в каждом соединении компонентов учитывать несколько видов энергии (гидравлическую и термодинамическую) многокомпонентного вещественного потока, а также совокупность концентраций входящих в него веществ. Принципы построения компьютерных моделей ХТС с неоднородными векторными связями, а также алгоритмы получения их вычислительных моделей, адаптированных для анализа универсальным вычислительным ядром [103], подробно рассмотрены в третьей главе.

2.4.3 Компоненты-измерители

При рассмотрении структурно-функциональной схемы СТУС, представленной на рисунке 1.1, в группе измерительных устройств было выделено два типа

измерителей: измерители характеристик энергии и измерители характеристик вещества. Такие ИзмУ, называемые далее *измерительными компонентами*, предназначены для измерения значений потенциальных или потоковых переменных связей моделируемой ХТС и осуществления передачи полученных значений в модель УУ, расположенную на логическом уровне многоуровневой КЦ СТУС. Тем самым они выполняют передачу результатов моделирования с ее объектного уровня на логический (2.50). Для этих целей измерительные компоненты размещаются на объектном уровне многоуровневой КЦ СТУС и имеют свои отображения одновременно на объектном и логическом слое многослойного редактора, в котором осуществляется формирование модели. Совокупность реализованных измерительных компонентов представлена в таблице 2.1

Таблица 2.1 – Компоненты-измерители языка моделирования ХТС

Измеритель	Отображение на объектном слое	Отображение на логическом слое	Математическая модель
Измеритель потенциальной переменной V_n			$V_{n2} = V_{n1}$
Измеритель потоковой переменной V_b			$V_{n3} = V_{b1}$
Измеритель разности потенциальных переменных dV_n			$V_{n3} = V_{n1} - V_{n2}$

Для применения представленных в таблице 2.1 измерительных компонентов к измерению значений переменных в цепях с неоднородными векторными связями каждый из них обладает набором следующих параметров, которые устанавливаются во вкладке «Свойства»:

– *Количество энергетических связей* – указывается количество энергетических связей, входящих в число конкретной неоднородной векторной связи. По умолчанию этот параметр имеет значение 1;

– *Количество информационных связей* – указывается количество информационных связей, входящих в состав неоднородной векторной связи, к которой присоединяется данный компонент. По умолчанию – 0.

– *Измеряем энергетическую (разность энергетических) связь (-ей)* – устанавливается значение «Истина», если производится измерение потенциальной или потоковой переменной энергетической связи. Если измерению подлежит потенциальная переменная информационной связи, следует указать значение «Ложь». По умолчанию данный параметр имеет значение «Истина»

– *Номер измеряемой переменной* – указывается номер связи измеряемой переменной отдельно для энергетических и информационных связей в ЛКБ измерительного компонента. По умолчанию имеет значение 0.

Совокупность обобщенных компонентов-источников, представленных на рисунках 2.6–2.7, и компонентов-преобразователей, приведенных на рисунках 2.8–2.9, образуют основу языка моделирования химико-технологических систем. На базе введенных компонентов в третьей главе будут рассмотрены реализованные компоненты с неоднородными векторными связями, позволяющие моделировать протекающие в ХТС процессы преобразования мультифизических энергетических и многокомпонентных вещественных потоков.

2.5 Графические нотации языка моделирования алгоритмических конструкций

2.5.1 Алгоритмические компонентные цепи

Логический уровень многоуровневой КМ СТУС, представленной на рисунке 2.5, реализует математико-алгоритмическую модель сценария исследования и проектирования сложных технических (технологических) управляемых систем, в состав которых входят сценарии функционирования их устройств управления. Она может быть представлена в графической форме в виде одной или нескольких алгоритмических КЦ (2.46), либо в текстовой форме, выраженной несколькими подпрограммами, написанными на языке моделирования алгоритмических конструкций и имеющими однозначные отображения в соответствующие алгоритмические компонентные цепи. Каждая подпрограмма C_{Ai} из C_A (2.46) может быть представлена в виде алгоритмической КЦ

$$C_A = (K_A, B_A, N_A, M), \quad (2.63)$$

где K_A – множество алгоритмических компонентов логического уровня. В него также будут входить компоненты-средства визуализации результатов моделирования и измерения и управляющие компоненты, образующие множество визуальных компонентов K_V (2.35), а также компоненты измерители потенциальных I_Vn1 и потоковых переменных I_Vb1 , представленные в таблице 2.1; B_A – множество связей компонентов алгоритмической КЦ C_A (2.63); N_A – множество узлов алгоритмической КЦ C_A (2.63); M – множество передаваемых по связям сообщений с данными, полученными путем расчета записанной в явной форме математико-алгоритмической модели компонента

$$Y = F_A(P_A, X), \quad (2.64)$$

где P_A – параметры компонента множества K_A (2.63); X – входные переменные компонента множества K_A ; Y – выходная переменная, являющаяся результатом работы компонента.

Работа компонентов множества K_A заключается в получении входных данных X , их преобразовании в соответствие с предписанным математико-алгоритмическим выражением, определяемым функцией F_A (2.64), а также в передаче полученных результатов Y другим компонентам, подключенным к выходным выводам рассматриваемого компонента.

2.5.2 Принципы формирования алгоритмических компонентов

Каждый компонент множества K_A представляет собой совокупность множеств

$$K_A = (P_A, b, n, M), \quad (2.65)$$

где P_A – набор параметров компонента, которые не зависят от данных, принятых в сообщении; b – массив номеров ветвей компонента, заданный в его ЛБК; n – массив номеров полюсов компонента в его ЛБК; M – массив сообщений, размер которого соответствует количеству ветвей компонента.

Каждый компонент множества K_A формируется с учетом 4-х основных аспектов, введенных в п. 2.1.2.

Согласно *геометрическому аспекту* каждый компонент множества K_A имеет свое графическое и (или) текстовое обозначение на логическом уровне многоуровневой КЦ СТУС. Для визуальных компонентов, имеющих свое отображение одновременно на логическом и визуальном уровнях, он предписывает правила отображения полученных результатов, а также правила преобразований действий пользователя с управляющими компонентами на визуальном уровне модели.

Топологическим аспектом устанавливаются типы связей компонента, которые могут быть входными или выходными. По входным связям данные, переменными связей которых являются аргументы реализуемой компонентом функции, поступают в него для их дальнейшего преобразования. *Выходные связи* позволяют передавать результаты обработки входных данных другим, подключенным к ним компонентам. Каждая *алгоритмическая связь* S_{Aj} задается парой значений $(\pm b_j, n_j)$, где b_j – номер соответствующей ветви, n_j – номер узла, заданные в ЛКБ данного компонента. Знак номера ветви b_j определяет тип данной связи. Если номер ветви указан со знаком «минус», то она будет считаться входной. Для всех выходных ветвей их номера указаны со знаком «плюс». Коммутация связей компонентов множества K_A производится на основании правила 2.3.

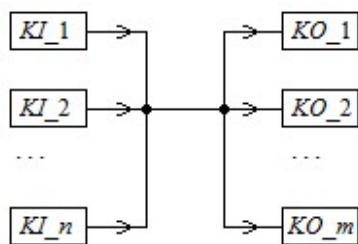


Рисунок 2.10 – Правило соединения алгоритмических компонентов

Правило 2.3. Согласно топологическому аспекту в одном узле, образованном связями компонентов, допускается соединение нескольких входных KI_1, KI_2, \dots, KI_n и нескольких выходных алгоритмических связей KO_1, KO_2, \dots, KO_m (рисунок 2.10). При этом не допускается, чтобы к одному узлу были подсоединены либо только входные, либо только

выходные связи компонентов множества K_A .

Физический аспект, накладывающий согласно методу КЦ физический смысл на переменные связей, для алгоритмических компонентов определяет типы передаваемых по связям данных. К ним относятся числовые и нечисловые типы. К числовым типам данных относятся целочисленный, вещественный и комплекс-

ный тип, являющиеся скалярными типами данных, а также массивы (вектора и матрицы), элементами которых являются переменные числовых типов. Нечисловыми типами данных являются строковые, табличные типы данных, а также графические типы, определяющие параметры визуализации данных (форму, цвет, цветовой градиент и т.п.) на визуальном слое многослойного редактора. Физический аспект накладывает ограничения на соединения этих компонентов, определяемые правилом 2.4.

Правило 2.4. Соединение в одном узле допустимо между связями компонентов, по которым передаются и принимаются данные одного и того же типа.

Математический аспект определяет математико-алгоритмические выражения, предназначенные для преобразования входных данных в выходные. Обобщенно математико-алгоритмическую модель компонентов данного множества можно представить в явном виде

$$V_{I2} = F(P, V_{I1}) \quad (2.66)$$

где P – параметры компонента, значения которых задаются и изменяются пользователем и не зависят от работы компонента; V_{I1} – входные данные, поступающие на вход компонента посредством связей S_{I1} ; V_{I2} – выходные данные, являющиеся результатами работы алгоритмической функции F и предназначенные для передачи другим компонентам, подключенным к связям S_{I2} данного компонента.

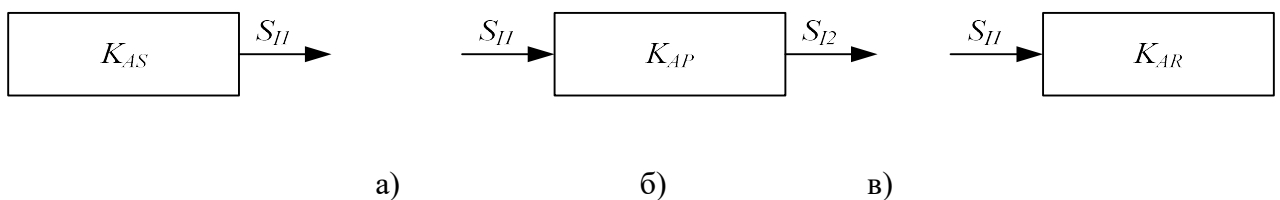


Рисунок 2.11 – Типы компонентов для построения и моделирования алгоритмических компонентных цепей

а) компонент-источник, б) компонент-преобразователь, в) компонент-приемник

Множество алгоритмических компонентов K_A позволяет формировать принадлежащие логическому уровню многоуровневой КЦ СТУС один или несколько сценариев, представленных своими алгоритмическими КЦ (2.46). Оно включает в себя компоненты следующих типов, представленных на рисунке 2.11

$$K_A = (K_{AS}, K_{AP}, K_{AR}), \quad (2.67)$$

где K_{AS} – *компоненты-источники данных* (рисунок 2.11, а), предназначенные для подачи в сценарии исследования и проектирования СТУС данных различных типов. К ним относятся компоненты, осуществляющие передачу данных с объектного (2.50) или визуального (2.52) на логический уровень многоуровневой КЦ СТУС, либо компоненты, принимающие информацию от внешних источников данных; K_{AP} – *компоненты-преобразователи данных* (рисунок 2.11, б), работа которых заключается в преобразовании входных данных, поступающих на их входные связи S_{I1} , в результаты определенного типа данных, передаваемых другим компонентам через выходные связи S_{I2} ; K_{AR} – *компоненты-приемники данных* (рисунок 2.11, в), осуществляющие прием данных с помощью связей S_{I1} с целью их передачи с логического на визуальный (2.51) или объектный уровень (2.54) многоуровневой КЦ СТУС.

Каждая ветвь b_j , заданная в ЛКБ любого компонента множества K_A (2.67), в рамках ГКБ алгоритмической КЦ (2.63) идентифицируется своим уникальным номером B_j , хранящимся в массиве \mathbf{B} размером CB . Каждый полюс n любого компонента множества K_A может быть соединен только с одним из узлов $N_j \in \mathbf{N}$, заданным уникальным номером в ГКБ алгоритмической КЦ C_A (2.63).

Грамматика языка моделирования алгоритмических конструкций, а также принципы их интерпретации в формат алгоритмических компонентных цепей, являющихся внутримашинным представлением данного языка, рассмотрены в четвертой главе. Функционирование алгоритмических КЦ осуществляется путем получения данных различных типов и их передачи с помощью алгоритма передачи сообщений, рассмотренного в пятой главе.

2.6 Классификация визуальных компонентов языка виртуальных инструментов и приборов

Множество компонентов визуального уровня многоуровневой K_V , называемых далее *визуальными компонентами*, предназначено для формирования лице-

вых панелей виртуальных приборов (ВП) [124], а также панелей визуализации и интерактивного управления многоуровневой КЦ СТУС. К подлежащим визуализации данным относятся результаты анализа модели ХТС с подключенными к ней компонентами ИзмУ и ИспУ и данные измерения характеристик реального объекта, а также данные, полученные в результате работы любого шага модели сценария исследования и проектирования СТУС, в который включена функциональная модель УУ. Множество визуальных компонентов, каждый из которых принадлежит визуальному уровню многоуровневой КЦ СТУС, образует *панели визуализации и интерактивного управления* [108] и включает в себя компоненты следующих подмножеств

$$K_V = (K_{V1} \cup K_{V2} \cup K_{V3} \cup K_{V4}) \quad (2.68)$$

где K_{V1} – множество компонентов статической информации, предназначенных для представления на панели ВП статической информации с целью повышения информативности их лицевых панелей и панелей визуализации; K_{V2} – множество компонентов-визуализаторов результатов моделирования и измерения, представленных значениями числовых и нечисловых типов данных; K_{V3} – множество компонентов-регуляторов, позволяющих пользователю вводить данные числовых и нечисловых типов для их использования в сценариях исследования и функционального проектирования СТУС, в том числе и в качестве значений параметров компонентов КЦ ХТС и подключенных к нему компонентов ИзмУ и ИспУ, принадлежащих объектному уровню многоуровневой КЦ СТУС; K_{V4} – множество компонентов, обладающих специальными оконными интерфейсами для более детального исследования динамических характеристик, получаемых в результате моделирования КЦ ХТС и при измерении величин реального объекта.

Множество компонентов статической информации предназначено для повышения информативности лицевых панелей ВИП. К таким компонентам относится представленный на рисунке 2.13 компонент «Надпись», в котором в качестве параметра «Текст» строкового типа данных

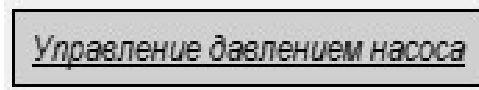


Рисунок 2.13 – Компонент «Надпись» на визуальном слое

вводится строка, статически отображаемая в панели визуализации и интерактивного управления. Помимо этого для изменения визуального представления данного компонента варьированию подлежат следующие его параметры: использование/не использование рамки, цвет рамки, цвет фона, цвет текста, использование/не использование фона, шрифт и выравнивание текста (по центру, по левому краю, по правому краю).

Для визуализации результатов моделирования и измерения, представленных значениями числовых типов данных, используются *компоненты-визуализаторы* множества K_{V2} . На визуальном уровне модели СТУС (рисунок 2.12, б) каждый из них представляет собой поле, на котором в понятной пользователю форме выводятся соответствующие результаты моделирования и (или) измерения. Компоненты данного множества, принадлежащие визуальному уровню многоуровневой КЦ СТУС, также отображаются на логическом слое редактора (рисунок 2.12, б) в виде компонента с единственной связью

$$S_1 = (b_1 \eta n_1) \rightarrow \{H\}, \quad (2.69)$$

с помощью которой они принимают подлежащее отображению значение H . Одновременное отображение компонентов множества K_{V2} на визуальном и логическом слоях редактора реализует функцию передачи подлежащих визуализации значений с логического уровня многоуровневой КЦ на ее визуальный уровень (2.51).

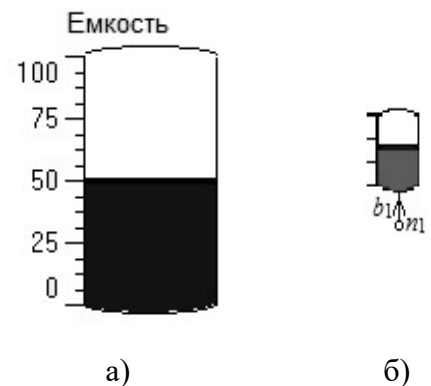
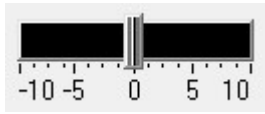
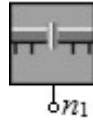


Рисунок 2.12 – Компонент-визуализатор скалярных значений (емкость)
а) отображение на визуальном слое;
б) отображение на логическом слое

Компоненты-регуляторы множества K_{V3} предоставляют пользователю возможность вносить изменения в процесс функционирования алгоритмических мо-



а)



б)

Рисунок 2.14 – Управляющий компонент

- а) отображение на визуальном слое;
б) отображение на логическом слое

делей, а также задавать различные значения, которые могут использоваться для расчета значений параметров КЦ ХТС и подключенных к ней компонентов ИзмУ и ИспУ, а также управляющих воздействий на реальную систему. К таким компонентам отно-

сятся разнообразные кнопки, бегунки, цифровые табло с возможностью задания значений и др. Компоненты множества K_{V3} , принадлежащие визуальному уровню многоуровневой КЦ СТУС, имеют свои отображения одновременно на визуальном (рисунок 2.14, а) и на логическом слоях (рисунок 2.14, б) многослойного редактора. Они реализуют функцию (2.52) межуровневого взаимодействия многоуровневой КЦ СТУС. С помощью единственной выходной связи

$$S_1 = (n_1 \eta^+ b_1) \rightarrow \{U\} \quad (2.70)$$

данные компоненты передают значение U с визуального на логический уровень КЦ СТУС другим компонентам, для его использования в сценариях исследования и функционального проектирования СТУС.

Компоненты визуализации результатов анализа КЦ ХТС и данных измерения ее реальных характеристик приведены в приложении 1, а пример программно-инструментальной реализации визуального компонента представлен в приложении 2.

Во множество K_{V4} включены компоненты для визуализации временных, частотных, параметрических и других характеристик. Такие компоненты, принадлежащие визуальному уровню многоуровневой КЦ СТУС, помимо отображений на логическом и визуальном слоях редактора имеют специальные интерфейсные окна, позволяющие более детально визуализировать и исследовать характеристики. В них реализованы различные функции, позволяющие производить исследование отображаемых характеристик. К компонентам множества K_{V4} относятся компоненты «График» и «Таблица».

Компонент «График» предназначен для отображений массивов данных (зависимостей) в графическом виде (в форме графиков, диаграмм, годографов и т.д.). Принадлежа визуальному уровню многоуровневой КЦ СТУС, на логическом слое редактора он имеет свое отображение в виде компонента с переменным числом связей. Каждая из них имеет вид

$$S_i = (b_i \eta^{-n_i}) \rightarrow \{G_i\}, \quad (2.71)$$

где G_i – подлежащая визуализации зависимость, $i = \overline{1, N}$.

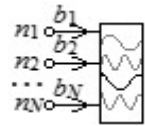


Рисунок 2.15 –
Компонент
«График»



Рисунок 2.16 – Окно исследования характеристик

На визуальном слое многослойного редактора многоуровневой КЦ СТУС компонент «График» имеет отображение в виде поля, в котором отображаются построенные временные или частотные характеристики, а также разнообразные зависимости переменных (корневые годографы, вольтамперные и другие характеристики). Для их исследования с помощью различных средств (сканеров, луп и т.п.) требуется выполнить двойной щелчок мыши по изображению данного ком-

понента на визуальном слое редактора. При этом будет открыто отдельное окно, представленное на рисунке 2.16, в котором пользователю предоставляются различные опции для исследования визуализируемых характеристик. Перечень опций работы пользователя с графиком и пояснения к ним приведены в приложении 3.

Для вывода данных различных типов в табличной форме во множестве компонентов K_{V4} предусмотрен компонент «Таблица», обладающий специальным окном, представленным на рисунке 2.17, для вывода информа-

№	Перв...	Пин	Назв...	Втор...	Пин	Назв...
1	T2	1	Эмитер	T1	1	Эмитер
2	T2	0	База	T3	1	Эмитер
3	T1	0	База	T2	2	Колле...
4	T3	2	Колле...	T3	0	База
5	T3	2	Колле...	T1	2	Колле...
6	T3	0	База	T1	2	Колле...

Рисунок 2.17 – Вид документа Таблица

ции различных типов данных: строкового, целочисленного, логического, вещественного, комплексного и т.п. Количество строк отображаемой в окне таблицы соответствует количеству организованных связей компонента «Таблица», представленного на рисунке 2.17 и отображаемого на логическом слое редактора

$$S_i = (b_i \eta^{-n_i}) \rightarrow \{V_i\}, \quad (2.72)$$

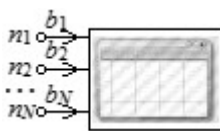


Рисунок 2.18 –
Компонент
«Таблица»

где V_i – значение в столбце i очередной формируемой строки таблицы. Тип данного значения может быть любым, но он должен оставаться постоянным для всех значений определенного столбца таблицы.

Принципы использования компонента «Таблица» приведены в приложении 4.

Таким образом, в сценариях исследования и функционального проектирования СТус, представленных имитационными моделями на логическом уровне многоуровневой КЦ СТус, компоненты множеств K_{V2} и K_{V4} являются приемниками, а компоненты множества K_{V3} – источниками данных различных типов (целочисленного, вещественного, комплексного, строкового и т.п.). Принадлежа визуальному уровню многоуровневой КЦ СТус, данные компоненты имеют свои

отображения как на визуальном, так и на логическом слое редактора, где они обмениваются данными с другими компонентами. Визуальные компоненты множества K_{V1} отображаются только на визуальном слое редактора и предназначены для повышения информативности лицевых панелей ВИП и панелей визуализации и интерактивного управления КЦ СТУС.

Визуальные компоненты, реализованные в комплексе программ моделирования химико-технологических систем, приведены в приложении 1, а пример программного кода реализации визуального компонента – в приложении 2.

2.7 Выводы

1. Для автоматизированного решения поставленных в первой главе задач исследования и функционального проектирования СТУС было осуществлено развитие метода компонентных цепей, заключающееся в построении многоуровневых компьютерных моделей сложных технических управляемых систем. Метод КЦ относится к классу объектно-ориентированных языков моделирования сложных объектов и систем с информационными и энергетическими потоками в связях между их элементами. Модель каждого элемента, называемая компонентом, формируется на основе четырех основных аспектов: геометрического, топологического, физического и математического.

2. На основе формирования и исследования компонентных цепей СТУС было установлено, что в их состав входят компоненты исполнительных и измерительных устройств, а также устройства управления, уравнения которых могут быть представлены в явном виде относительно переменных своих выходных связей. На основе этого утверждения предложен принцип разделения уравнений компонентной цепи СТУС на явную и неявную составляющие. Неявная часть цепи СТУС подлежит расчету универсальным вычислительным ядром, в основе которого лежит алгоритм ВЭ, представленный в пункте 2.1.5. Выражения, представленные в явной форме, подлежат расчету отдельно друг от друга на основе результатов, полученных при анализе неявной части модели ХТС.

3. Предложены методика формирования многоуровневых компонентных цепей СТУС, их обобщенная структура, а также язык МКЦ. Это позволяет уменьшить размерность решаемой системы модели управляемого объекта, а также снизить вычислительные нагрузки при расчете нелинейных моделей устройства управления, представляемых в явном виде. Также эта методика позволяет вносить изменения в модель управляемого объекта без изменения модели устройства управления, либо изменять его модель без изменения модели объекта. Она открывает возможности формирования многоуровневых компонентных цепей, состоящих из визуальных компонентов, алгоритмических КЦ сценариев исследования и функционального проектирования СТУС с включенными в них сценариями функционирования устройств управления, а также компонентной цепи ХТС с неоднородными векторными связями между ее компонентами. Структура многоуровневых компонентных цепей позволяет расположить МКЦ СТУС на трех взаимосвязанных уровнях: визуальном, логическом и объектом, а также включает в себя правила межуровневого взаимодействия. Являясь графическим языком моделирования, язык многоуровневых компонентных цепей включает в себя три подязыка: язык моделирования ХТС, язык моделирования алгоритмических конструкций и язык виртуальных инструментов и приборов, пересечением пар которых образуются соответствующие межуровневые взаимодействия.

4. Разработаны графические нотации языка моделирования ХТС, согласно которым каждому источнику, преобразователю и измерителю мультифизической энергии или многокомпонентных вещественных потоков ставится в соответствие определенный компонент, представляющий его в формате данного языка. Для соединения компонентов используются неоднородные векторные связи, позволяющие отражать в модели ХТС все перечисленные виды потоков.

5. Приведены графические нотации языка моделирования алгоритмических конструкций, с использованием которых формируются сценарии исследования и функционального проектирования СТУС на логическом слое редактора многоуровневых компонентных цепей. С учетом четырех основных аспектов метода компонентных цепей предложены принципы формирования алгоритмиче-

ских компонентов, которые могут быть компонентами-источниками, компонентами-преобразователями и компонентами-приемниками данных различных типов. Часть из этих компонентов реализуют в себе межуровневые взаимодействия.

6. В рамках языка виртуальных инструментов и приборов предложена структура обобщенного виртуального прибора, приведено и обосновано его многоуровневое представление, а также приведена классификация визуальных компонентов. Они подразделяются на четыре основных множества: множество компонентов статической информации; множество компонентов-визуализаторов; множество компонентов-регуляторов и множество компонентов, обладающих специальными оконными интерфейсами для исследования характеристик и зависимостей.

7. Сформулирована общая концепция языка многоуровневых компонентных цепей, позволяющая в единой модели объединять математические модели, описывающие протекающие в ХТС непрерывные процессы преобразования мультифизических энергетических и многокомпонентных вещественных потоков, имитационные модели УУ, которое с точки зрения своего функционирования относится к классу дискретно-событийных систем, и визуальные модели панелей визуализации результатов и интерактивного управления параметрами модели и управляющими воздействиями на реальные объекты.

Глава 3. Компьютерное моделирование процессов функционирования химико-технологических систем

Метод компонентных цепей [17] является универсальным методом компьютерного моделирования физически неоднородных объектов в статическом и динамическом (во временной и частотной областях) режимах. Основанная на нем среда моделирования MAPC [20, 125] (свидетельство о государственной регистрации программы для ЭВМ представлено в приложении 18) позволяет автоматизировать анализ простых технических объектов с выводом его результатов на различные средства визуализации в виде статических значений наблюдаемых переменных и их временных, частотных и параметрических характеристик. К их классу относятся механические [100], гидравлические, термодинамические, электрические цепи [126], электромеханические преобразователи [18, 19], описываемые компьютерными моделями с энергетическими связями (2.4), а также системы автоматического управления, средства телекоммуникации [127] и математические выражения [101], для моделирования которых достаточно применения информационных (2.5) и однородных векторных связей (2.6), представляющих собой совокупность нескольких информационных связей.

Формирование и анализ моделей химико-технологических систем, обобщенные структурно-функциональные схемы которых рассмотрены в пункте 1.3, потребовало расширения и развития формализма метода компонентных цепей. Для этого в него было введено понятие неоднородной векторной связи, разработаны алгоритмы формирования и повышения быстродействия вычислительных моделей для компонентных цепей, содержащих такие связи.

Представленная на объектном уровне многоуровневой КЦ СТУС в виде компонентов с неоднородными векторными связями компонентная подцепь ХТС включает в себя модели исполнительных и измерительных устройств. Значения их параметров наряду со значениями управляющих воздействий должны определяться в процессе работы представленного на логическом уровне сценария исследования и функционального проектирования СТУС, работа которого основана на многократном анализе модели ХТС, представленной на объектном уровне.

3.1 Формализованное представление объектов

с неоднородными векторными связями в языке моделирования ХТС

Для целей построения компонентных цепей ХТС с протекающими в них процессами преобразования мультифизических энергетических и многокомпонентных вещественных потоков построена методика и реализованы алгоритмы автоматизированного формирования цепей с неоднородными векторными связями (НВС). Каждая НВС позволяет сочетать в себе гидро- и термодинамические характеристики, а также вектор концентраций входящих в многокомпонентный поток веществ. На их основе открываются новые возможности компьютерного моделирования химико-технологических систем, анализ структуры и связей которых был проведен в пункте 1.3.

3.1.1 Структура неоднородной векторной связи

Для целей моделирования в формате метода КЦ преобразования мультифизических энергетических и многокомпонентных вещественных потоков, протекающих с ХТС, применены компонентные цепи с неоднородными векторными связями.

Каждая *неоднородная векторная связь* $\mathbf{S}_{\mathbf{v}_k}$ [123], структура и содержание которой представлено на рисунке 3.1, представляет собой совокупность ряда взаимосвязанных друг с другом энергетических и информационных связей, *векторная ветвь* \mathbf{b}_k которой инцидентна ее *векторному узлу* \mathbf{n}_k , и имеет вид

$$\mathbf{S}_{\mathbf{v}_k} = (\mathbf{b}_k \eta \mathbf{n}_k), \quad (3.1)$$

где $\mathbf{b}_j = [b_1 \ b_2 \ \dots \ b_{C_e}]$ – векторная ветвь, являющаяся вектором размера C_e ($\mathbf{b}_j \in R^{C_e}$) номеров ветвей скалярных энергетических связей (2.4), входящих в состав НВС $S_{E_{-1j}}, S_{E_{-2j}}, \dots, S_{E_{-C_ej}} e \subset \mathbf{S}_{\mathbf{v}_k}$; η – отношение инцидентности ветви \mathbf{b}_j узлу \mathbf{n}_j (указывается η^+ – когда ветвь \mathbf{b}_j направлена к узлу \mathbf{n}_j , и η^- – если ветвь \mathbf{b}_j ориентирована от узла \mathbf{n}_j);

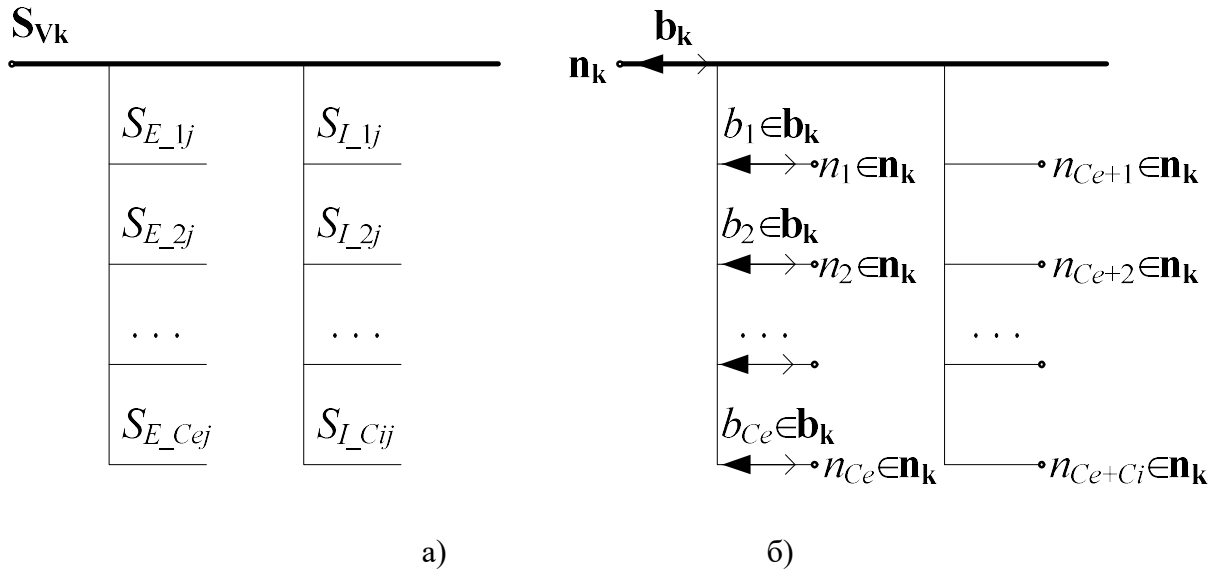


Рисунок 3.1 – Структура и содержание неоднородной векторной связи

- а) структура неоднородной векторной связи;
- б) содержание неоднородной векторной связи

$\mathbf{n}_k = [n_1, n_2, \dots, n_{Ce}, n_{Ce+1}, n_{Ce+1}, \dots, n_{Ce+Ci}]$ – векторный узел ($\mathbf{n}_k \in R^{Ce+Ci}$), представляющий собой массив номеров узлов энергетических (2.4) и информационных (2.5) связей, входящих в состав неоднородной векторной связи S_{V_k} ; Ce – количество энергетических связей, а Ci – количество информационных связей в составе НВС S_{V_k} .

В каждой энергетической связи $S_{E_{-i}} \subset S_{V_k}$ циркулирует энергетический поток определенной физической природы. Каждая информационная связь $S_{I_{-j}} \subset S_{V_k}$ соответствует определенному наблюдаемому в ней информационному сигналу.

Для всех энергетических связей задается единое направление потока, которое может быть указано для всей НВС S_{V_k} , которой они принадлежат. В случае, если поток в связи будет направлен от узла к компоненту, то соотношение инцидентности будет записываться в виде

$$S_{V_k} = (\mathbf{b}_k \eta^- \mathbf{n}_k), \tag{3.2}$$

а если он направлен от компонента к узлу, то в виде

$$S_{V_k} = (\mathbf{b}_k \eta^+ \mathbf{n}_k). \tag{3.3}$$

Формально НВС S_{V_k} можно представить в виде

$$\mathbf{S}_{\mathbf{vk}} = [S_{E_{-1k}}, S_{E_{-2k}}, \dots, S_{E_{-Cek}}, S_{I_{-1k}}, S_{I_{-2k}}, \dots, S_{I_{-Cik}}], \quad (3.4)$$

где $S_{E_{-1k}}, S_{E_{-2k}}, \dots, S_{E_{-Cek}}$ – совокупность энергетических связей (2.4), каждая из которых принадлежит определенному энергетическому субаспекту физического аспекта F_E (2.7)

$$S_{E_{-i}} \in F_E. \quad (3.5)$$

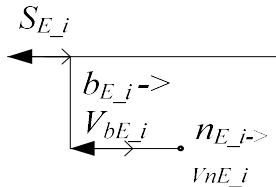


Рисунок 3.2 – Структура и содержание энергетической связи $S_{E_{-j}}$

Каждой энергетической связи $S_{E_{-i}}$, структура которой представлена на рисунке 3.2, ставится в соответствие пара топологических координат – узел $n_{E_{-i}}$ и ветвь $b_{E_{-i}}$ – соответственно с потенциальной $n_{E_{-i}} \rightarrow V_{n_{E_{-i}}}$ и потоковой переменной $b_{E_{-i}} \rightarrow V_{b_{E_{-i}}}$

$$S_{E_{-i}} = (b_{E_{-i}} \eta n_{E_{-i}}) \rightarrow \{n_{E_{-i}} \rightarrow V_{n_{E_{-i}}}, b_{E_{-i}} \rightarrow V_{b_{E_{-i}}}\} \rightarrow \{V_{n_{E_{-i}}}, V_{b_{E_{-i}}}\}, \quad (3.6)$$

$S_{I_{-1j}}, S_{I_{-2j}}, \dots, S_{I_{-Cij}}$ – совокупность информационных связей (2.5), каждая из которых принадлежит определенному информационному субаспекту физического аспекта F_I (2.8)

$$S_{I_{-j}} \in F_I. \quad (3.7)$$

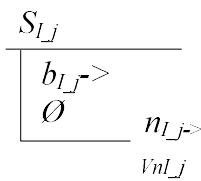


Рисунок 3.3 – Структура информационной связи

$$n_{I_{-j}} \rightarrow V_{n_{I_{-j}}}, b_{I_{-j}} \rightarrow \emptyset$$

Каждой информационной связи $S_{I_{-j}}$, структура которой представлена на рисунке 3.3, ставится в соответствие пара топологических координат – узел $n_{I_{-j}}$ и ветвь $b_{I_{-j}}$ с одной потенциальной переменной

$$S_{I_{-j}} = (b_{I_{-j}} \eta n_{I_{-j}}) \rightarrow \{n_{I_{-j}} \rightarrow V_{n_{I_{-j}}}, b_{I_{-j}} \rightarrow \emptyset\} \rightarrow \{V_{n_{I_{-j}}}\}. \quad (3.8)$$

Формирование компонентных цепей ХТС, имеющих НВС, производится на основании правил коммутации компонентов. При этом векторам \mathbf{b}_j и \mathbf{n}_j , все элементы которых заданы в ЛКБ компонента, каждой неоднородной векторной связи

(3.1) ставятся в соответствие вектора \mathbf{V}_j и \mathbf{N}_j , характеризующие их потенциальные и потоковые переменные в рамках формируемой КЦ.

3.1.2 Правила коммутации компонентов с НВС

В результате соединения между собой двух и более неоднородных векторных связей компонентов в КЦ ХТС, образуется *векторный узел*. В зависимости от структуры входящих в него связей ему может быть поставлена в соответствие одна или несколько потенциальных переменных. Процесс соединения связей компонентов и образование узлов называется *коммутацией*. При формировании математической модели рассматриваемой цепи каждому узду будет поставлено в соответствие C_e топологических уравнений (где C_e – число энергетических связей в НВС, вступающих в операцию коммутации). Коммутация компонентов и построения вычислительной модели цепи, включающей неоднородные векторные связи, осуществляется на основании следующих правил:

Правило 3.1. Общее правило коммутации связей по классу. Коммутация между двумя НВС S_1 и S_2 компонентов K_1 и K_2 , где $S_1 \in K_1$ и $S_2 \in K_2$, допускается, если данные связи являются одного типа

$$S_1 = S \wedge S_2 = S, \quad (3.9)$$

где $S = S_I \vee S_E \vee S_V$.

Правило 3.2. Правило коммутации информационных связей. Если связи S_1 и S_2 являются информационными (2.5), т.е. $(S_1 = S_I) \wedge (S_2 = S_I)$, то соединение между данными связями компонентов возможно, если

$$F_1 = F_2, \quad (3.10)$$

где F_1 – субаспект физического аспекта (2.8) связи S_1 ; F_2 – субаспект физического аспекта (2.8) информационной связи S_2 .

Правило 3.3. Правило коммутации однородных векторных связей. Когда связи S_1 и S_2 являются однородными векторными (2.6) $(S_1 = S_{IV}) \wedge (S_2 = S_{IV})$, соединение между ними возможно в случае

$$(F_1 = F_2) \wedge (C_1 = C_2), \quad (3.11)$$

где F_1 – физический субаспект (2.8) всех информационных связей, входящих в связь S_1 ; F_2 – физический субаспект (2.8) всех информационных связей, входящих в связь S_2 ; C_1 – размерность векторной связи S_1 , то есть количество входящих в нее информационных связей (2.5); C_2 – размерность векторной связи S_2 .

Правило 3.4. Правило коммутации энергетических связей. Соединение двух энергетических связей вида (2.4) $S_1 = S_E$ и $S_2 = S_E$ допустимо, если их физические аспекты $F_1 \in S_1$ и $F_2 \in S_2$ совпадают

$$F_1 = F_2, \quad (3.12)$$

где F_1, F_2 – физические аспекты (2.7) энергетических связей S_1 и S_2 соответственно.

Правило 3.5. Правило коммутации неоднородных векторных связей. Две НВС $S_{V1} = S_V$ и $S_{V2} = S_V$ могут быть соединены между собой при выполнении следующих условий:

– количество C_{e1} энергетических связей, входящих в НВС S_{V1} , должно быть равно количеству C_{e2} энергетических связей НВС S_{V2}

$$C_{e1} = C_{e2}; \quad (3.13)$$

– равенства количества информационных связей, включенных в векторные связи S_{V1} и S_{V2}

$$C_{i1} = C_{i2}, \quad (3.14)$$

где C_{i1} – количество информационных связей (2.5), включенных в НВС S_{V1} ; C_{i2} – количество информационных связей (2.5) в НВС S_{V2} ;

– совпадение физических субаспектов энергетических связей $S_{E_i}^1$, входящих в НВС S_{V1} , и энергетических связей $S_{E_i}^2$, образующих НВС S_{V2}

$$F_{E_i}^1 = F_{E_i}^2, \quad (3.15)$$

где $F_{E_i}^1$ – физический субаспект (2.7) энергетической связи $S_{E_i}^1 \in S_{V1}$; $F_{E_i}^2$ – физический субаспект (2.7) энергетической связи $S_{E_i}^2 \in S_{V2}$; $i = \overline{1, C_{e1}}$;

– совпадение физических субаспектов информационных связей $S_{I_j}^1 \in S_{V_1}$ с физическими субаспектами соответствующих информационных связей $S_{I_j}^2 \in S_{V_2}$

$$F_{I_j}^1 = F_{I_j}^2, \quad (3.16)$$

где $F_{I_j}^1$ – физический субаспект (2.8) информационной связи $S_{I_j}^1 \in S_{V_1}$; $F_{I_j}^2$ – физический субаспект (2.8) информационной связи $S_{I_j}^2 \in S_{V_2}$; $j = \overline{1, Ci}$.

В случае нарушения правил 1-5 хотя бы для одной связи в рассматриваемой КЦ должно быть выведено соответствующее сообщение и прекращен процесс формирования КЦ ХТС, содержащей НВС. Данные правила положены в основу алгоритма формирования компонентных цепей химико-технологических систем.

3.2 Обобщенная модель физико-химического компонента

Технические и технологические объекты, в которых помимо мультифизических потоков энергии циркулируют многокомпонентные вещественные потоки, относятся к классу ХТС.

Химико-технологическими системами, обобщенная структурно-функциональная схема которых приведена на рисунке 1.2, представляют собой совокупность элементов и аппаратов химической промышленности, в которых протекают разнообразные процессы, направленные на получение конкретной продукции с заданными характеристиками. Каждый аппарат ХТС, в котором производятся изменения физических характеристик и компонентных составов вещественных потоков, представляет собой *физико-химическую систему*. Циркулирующие между элементами и аппаратами вещественные потоки представляют собой многокомпонентные смеси веществ, обладающие различными видами энергии. Определенный вид энергии в каждом узле ХТС характеризуется парой дуальных переменных – потенциальной V_{nj} и потоковой V_{bj} , произведение текущих значений которых в любой момент времени определяет мгновенную мощность конкретной физической природы.

Вещественный поток может быть образован одним веществом или смесью веществ, находящихся в различных агрегатных состояниях. Образованный одним веществом в газовой или жидкой фазе поток будет называться *однокомпонентным*. Концентрация входящего в него вещества будет равна 1, а концентрации других веществ в данном потоке будут равны 0.

Когда вещественный поток образован несколькими веществами, он будет считаться *многокомпонентным* [128]. Конкретное вещество, находящееся в потоке, будет являться *компонентой вещества*, характеризующейся своей *концентрацией*, определяющей массовую, объемную или молярную долю вещества в потоке.

Протекающие в ХТС процессы направлены на изменение физических свойств и компонентных составов циркулирующих потоков. Оказывая техногенные воздействия на объекты окружающей среды, они зависят:

- 1) от компонентного состава входных вещественных потоков и запасенной в них энергии различной физической природы;
- 2) от физических и химических свойств веществ, составляющих многокомпонентный поток и определяющих агрегатное состояние, называемой *фазой вещества*;
- 3) от количества подводимой и отводимой энергии различной физической природы без изменения компонентного состава потока;
- 4) от геометрических форм, размеров и других параметров химических аппаратов.

Для описания многокомпонентных химико-технологических систем введен обобщенный компонент ХТС [129], представленный на рисунке 3.4. Рассмотрим методику его построения с точки зрения четырех основных аспектов метода КЦ, которые были приведены в пункте 2.1.3.

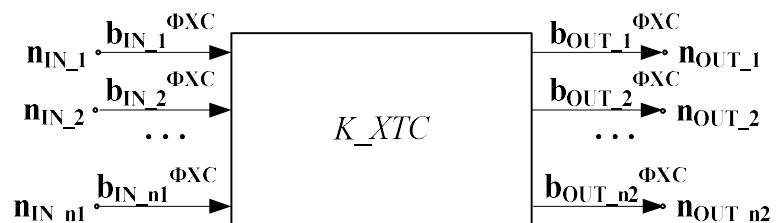


Рисунок 3.4 – Обобщенный компонент химико-технологической системы

Геометрическим аспектом задается условное графическое и (или) буквенное обозначение компонента на объектном уровне многоуровневой КЦ СТУС. Также определяется геометрическая форма и размеры описываемого компонентом аппарата, от которых зависят характеристики протекающих в нем процессов.

С точки зрения *топологического аспекта* формируются следующие совокупности связей компонента:

– совокупность *входных физико-химических связей*

$$\begin{aligned} \mathbf{S}_{\text{IN}_1}^{\text{ФХС}} = (\mathbf{b}_{\text{IN}_1} \eta^- \mathbf{n}_{\text{IN}_1}), \mathbf{S}_{\text{IN}_2}^{\text{ФХС}} = (\mathbf{b}_{\text{IN}_2} \eta^- \mathbf{n}_{\text{IN}_2}), \dots, \\ \mathbf{S}_{\text{IN}_{n1}}^{\text{ФХС}} = (\mathbf{b}_{\text{IN}_{n1}} \eta^- \mathbf{n}_{\text{IN}_{n1}}) \end{aligned}, \quad (3.17)$$

находящихся слева от компонента, по которым в него поступают входные вещественные потоки для преобразования их физических свойств и (или) компонентных составов;

– совокупность *выходных физико-химических связей*

$$\begin{aligned} \mathbf{S}_{\text{OUT}_1}^{\text{ФХС}} = (\mathbf{b}_{\text{OUT}_1} \eta^- \mathbf{n}_{\text{OUT}_1}), \mathbf{S}_{\text{OUT}_2}^{\text{ФХС}} = (\mathbf{b}_{\text{OUT}_2} \eta^- \mathbf{n}_{\text{OUT}_2}), \dots, \\ \mathbf{S}_{\text{OUT}_{n1}}^{\text{ФХС}} = (\mathbf{b}_{\text{OUT}_{n1}} \eta^- \mathbf{n}_{\text{OUT}_{n1}}) \end{aligned}, \quad (3.18)$$

расположенных справа от изображения компонента и предназначенных для вывода преобразованных вещественных потоков из компонента ХТС.

С позиций *физического аспекта* каждой связи ставится в соответствие совокупность переменных, которыми в математической модели описываются протекающие в компоненте процессы. Каждая входная (3.17) и выходная (3.18) физико-химические связи представляют собой совокупность энергетических (2.4) и однородных векторных связей (2.6)

$$\mathbf{S}_j^{\text{ФХС}} = \{S_j^G, S_j^T, S_j^C\}, \quad (3.19)$$

где $S_j^G = (b_j^G \eta n_j^G) \rightarrow \{n_j^G \rightarrow V_{nj}^G, b_j^G \rightarrow V_{bj}^G\}$ – энергетическая связь, принадлежащая гидравлическому субаспекту физического аспекта. Она характеризуется переменными: потенциальной $V_{nj}^G = P_j$ – давлением и потоковой $V_{bj}^G = G_j$ – объемным, массовым или молярным расходом вещества. Далее такую связь будем называть *гидравлической*; $S_j^T = (b_j^T \eta n_j^T) \rightarrow \{n_j^T \rightarrow V_{nj}^T, b_n^T \rightarrow V_{bj}^T\}$ – энергетическая связь термо-

динамического субаспекта физического аспекта, характеризующая запас тепловой энергии в вещественном потоке. Она включает следующие переменные связи: $V_{nj}^T = T_j$ – температура, $V_{bj}^T = Q_j$ – поток теплоты и далее называется *термодинамической связью*; $S_j^C = (b_j^C \eta \mathbf{n}_j^C) \rightarrow \{\mathbf{n}_j^C \rightarrow \mathbf{V}_{nj}^C, b_j^C \rightarrow \emptyset\}$ – однородная векторная связь с вектором переменных $\mathbf{V}_{nj}^C = \mathbf{C}_j$, характеризующим концентрации веществ в описываемой им связи. Размерность данного вектора соответствует количеству веществ, циркулирующих во всей моделируемой ХТС. Далее такую связь будем называть *связью концентраций*.

По виду исследуемых концентраций веществ в потоке предусматриваются два режима моделирования ХТС методом МКЦ:

1) режим с полным компонентным составом, когда рассматриваются все вещества, действующие в любом потоке моделируемой системы. В этом случае сумма концентраций всех веществ в каждом из узлов в любой момент времени должна равняться 1. Если вещество не попадает в данную связь, то его концентрация принимается равной 0;

2) режим с частным компонентным составом. В этом случае рассматриваются отдельные вещества и в векторе концентраций находятся лишь абсолютные концентрации рассматриваемых веществ, имеющие единицы измерения гр./м^3 . Такой режим может быть применен для исследования биотехнологических систем и их взаимодействия с окружающей средой, когда определенное количество вещества, введенное в аппарат ХТС, оказывает воздействие на параметры и характеристики окружающей среды. Также с помощью данного режима может исследоваться пропускная способность сердечно-сосудистой системы человека [130] за счет изменения вязкости циркулируемой в ней жидкости.

С точки зрения *математического аспекта* процессы, протекающие в любом компоненте ХТС, будут описываться линейными, нелинейными и дифференциальными уравнениями (2.9)–(2.12), составленными относительно переменных связей данного компонента, которые образуют вектор

$$V^{K_ХТС} = [\mathbf{P}_K \quad \mathbf{G}_K \quad \mathbf{T}_K \quad \mathbf{Q}_K \quad \mathbf{C}_K]^T \quad (3.20)$$

где \mathbf{P}_k – вектор давлений во всех узлах, образованных соединением связей компонента со связями других компонентов; \mathbf{G}_k – вектор расходов вещества во всех его входных и выходных потоках; \mathbf{T}_k – вектор температур вещественных потоков; \mathbf{Q}_k – вектор тепловых потоков; \mathbf{C}_k – вектор концентраций веществ во входных и выходных потоках компонента ХТС.

Так как в ряде аппаратов ХТС протекают процессы, распределенные по длине аппарата, то помимо традиционных для метода компонентных цепей линейных (2.9), нелинейных (2.10) и обыкновенных дифференциальных уравнений (2.11), вводятся формы уравнений с частными производными по длине аппарата:

– линейные уравнения с частными производными по длине аппарата l

$$\sum_{i=1}^N K_i \cdot \frac{dV_i^{K_XTC}}{dl} = \sum_{i=1}^N \alpha_i \cdot V_i^{K_XTC} + b, \quad (3.21)$$

где K_i, α_i, b – постоянные коэффициенты;

– нелинейные уравнения с линейно входящими частными производными

$$\sum_{i=1}^N K_i \cdot \frac{dV_i^{K_XTC}}{dl} = f(\mathbf{V}^{K_XTC}) \quad (3.22)$$

где K_i – постоянные коэффициенты, f – нелинейная функция;

– обыкновенные дифференциальные уравнения с частными производными по длине аппарата

$$\frac{d\Psi(V^{K_XTC})}{dt} = \sum_{i=1}^N K_i \cdot \frac{dV_i^{K_XTC}}{dl} + f(\mathbf{V}^{K_XTC}) \quad (3.23)$$

Математическую модель обобщенного компонента ХТС, содержащую уравнения материального и теплового баланса, а также уравнения преобразований физических характеристик и компонентных составов вещественных потоков, можно представить в виде совокупности следующих уравнений вида (2.9)-(2.11), (3.21)-(3.23)

$$M_K^{\Phi X C} = \{M_K^G \cup M_K^{GT} \cup M_K^T \cup M_K^{GTC} \cup M_K^C \cup I_T\}, \quad (3.24)$$

где M_K^G – множество уравнений модели, в которые входят только переменные гидравлических связей S_j^C , $\forall j = \overline{1, n}$ (где $n = n_1 + n_2$) НВС компонента ХТС; M_K^{GT} – множество уравнений модели, каждой из которых содержит переменные гидравлических S_j^G и термодинамических связей S_j^T компонента ХТС; M_K^T – множество уравнений, включающих в себя только переменные термодинамических связей S_j^T компонента; M_K^{GTC} – совокупность уравнений компонента, в которых выражена зависимость концентраций веществ в потоках от гидравлических и термодинамических переменных связей данного компонента; M_K^C – уравнения, в которые входят только переменные, характеризующие концентрации веществ в потоках компонента; I_T – совокупность топологических соотношений компонента, предназначенных для формирования топологических уравнений для узлов, образованных его связями.

Векторно-матричная форма модели (3.24), получаемая на каждом шаге ВЭ после линеаризации нелинейных (2.10) и алгебраизации дифференциальных уравнений (2.11) всех компонентов, будет иметь вид

$$[\Phi_{K_ФХС}] \cdot \begin{bmatrix} \mathbf{P}_K \\ \mathbf{G}_K \\ \mathbf{T}_K \\ \mathbf{Q}_K \\ \mathbf{C}_K \end{bmatrix} = \begin{bmatrix} W_K^G \\ 0 \\ W_K^T \\ 0 \\ W_K^C \end{bmatrix} \quad (3.25)$$

где $\Phi_{K_ФХС}$ – прямоугольная матрица коэффициентов с компонентных уравнениях рассматриваемого компонента ФХС; W_K^G – вектор свободных членов в компонентных уравнениях, описывающих гидравлические процессы; W_K^T – вектор свободных членов в компонентных уравнениях, описывающих термодинамические процессы; W_K^C – вектор свободных членов в компонентных уравнениях, позволяющих рассчитывать концентрации веществ в выходных потоках компонента ФХС.

Линеаризация частных производных в дифференциальных уравнениях (3.21)-(3.23) производится на этапе формирования программно-алгоритмического вида компонентов. При этом длина аппарата L разбивается на ряд элементарных ячеек длиной l , для каждой из которых формируются линеаризованные по неявной схеме Эйлера (2.28) уравнения относительно переменных связей начала и конца рассматриваемой ячейки.

Гидравлические, термодинамические и массообменные процессы, протекающие в элементах и аппаратах ХТС, зависят от геометрических и физических свойств самого элемента или аппарата K_A , которые являются постоянными для каждого конкретного элемента или аппарата, а также от физико-химических свойств многокомпонентного вещественного потока K_S . Как правило, для каждого вещества они являются справочными данными или являются функциями давления и/или температуры

$$K_{Si} = F(\mathbf{P}_K, \mathbf{T}_K), \quad (3.26)$$

расчет которых производится по приводимым в литературе [131] по формулам с использованием значений входящих в них коэффициентов.

Свойства многокомпонентного вещественного потока зависят от соответствующих свойств составляющих его веществ и их концентраций

$$K_{SN} = F_N(\mathbf{K}_{S1}, \mathbf{C}), \quad (3.27)$$

где \mathbf{K}_{S1} – вектор физико-химических свойств веществ, составляющих многокомпонентный поток; \mathbf{C} – вектор концентраций веществ многокомпонентного вещественного потока.

Зависимость коэффициентов уравнений математической модели обобщенного компонента ХТС (3.24) от переменных (3.26)-(3.27) позволяют отнести соответствующие модели к классу нелинейных.

Структура разработанной базы данных, предназначенной для хранения свойств веществ и их использования для расчета коэффициентов модели (3.26) и (3.27) компонентных уравнений ХТС, приводится в приложении 5. Далее эту БД будем называть базой данных характеристик компонент потока.

На основе рассмотренных аспектов компьютерная модель ХТС, сформированная в формате языка моделирования ХТС, имеет структуру, представленную на рисунке 3.5. Блок расчета коэффициентов K_S и его взаимосвязь с базой данных характеристик компонент потока являются общими для всех компонентов ХТС и реализуются в общем интерфейсном классе, от которого наследуются все классы, реализующие конкретные компоненты.

Обобщенный компонент ХТС и его программно-алгоритмическая реализация состоят из следующих блоков:

– *топологического блока*, с помощью которого компонент осуществляет передачу универсальному вычислительному ядру совокупность топологических соотношений компонента I_T (3.24), в каждом из которых топологическая пара представлена в ГКБ сформированной КЦ, содержащей рассматриваемый компонент;

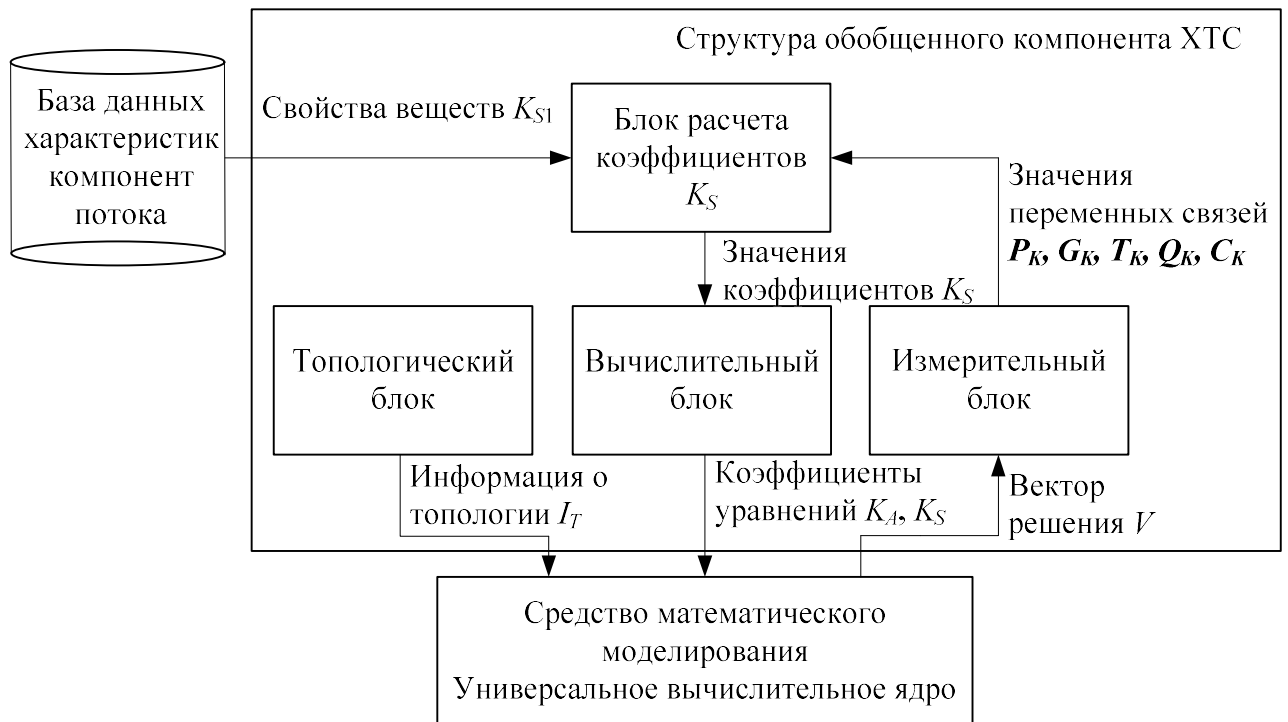


Рисунок 3.5 – Блочная структура обобщенного компонента ХТС

– *вычислительного блока*, осуществляющего формирование компонентных уравнений M_K^G , M_K^{GT} , M_K^T , M_K^{GTC} , M_K^C математической модели компонента ХТС (3.24);

– *блока расчета коэффициентов*, в котором осуществляется расчет коэффициентов K_S , значения которых зависят от гидравлических P_K, G_K и термоди-

намических характеристик T_k, Q_k входных связей компонентов (3.26), а также их концентраций C_k (3.27), с которыми каждое вещество входит в многокомпонентный вещественный поток. Он может быть встроен в модель или в класс, от которого наследуется класс модели, а также может быть представлен в виде интерактивной математической панели логического уровня многоуровневой КЦ. Рассчитанные коэффициенты используются для формирования уравнений, по которым будут рассчитаны переменные цепи ХТС;

– *измерительного блока*, с помощью которого из общего вектора решения цепи (2.15) выбираются значения давления P_k , объемного расхода вещества G_k , температуры T_k , теплового потока Q_k и вектора концентраций веществ C_k каждой связи компонента, на основе которых в блоке расчета коэффициентов осуществляется вычисление текущих значений коэффициентов K_{Si} (3.26) и K_{SN} (3.27).

Блок расчета коэффициентов является общим для всех моделей компонентов ХТС. Он содержит методики обработки результатов компьютерного моделирования и расчета коэффициентов уравнений моделей, которые в своем составе содержат решение полиномиальных уравнений, поиск коэффициентов на основе минимизации целевых функций и т.п. Каждая методика реализуется в виде некоторого метода интерфейсного класса, общего для всех компонентов ХТС.

На основе обобщенных моделей ХТС сформирован набор типовых методик для моделирования компонентов химико-технологических систем, к классу которых могут быть отнесены гидравлические [132, 133], термодинамические [134], теплоэнергетические [135] и тепло- и массообменные аппараты [39]. Они позволяют формировать компьютерные модели соответствующих систем, анализировать протекающие в них мультифизические процессы и автоматизировать решение задач исследования и функционального проектирования систем управления.

3.3 Автоматическое построение и решение вычислительной модели компонентной цепи химико-технологической системы

Для автоматизированного анализа процессов, протекающих в ХТС, являющихся объектами управления в рамках СТУС, используется универсальное вы-

числительное ядро [103]. Согласно методу КЦ, на котором основан язык моделирования ХТС, вычислительная модель анализируемого объекта с энергетическими (2.4) и информационными (2.5) связями включает в себя топологические (2.18) и компонентные (2.9)–(2.12) уравнения.

Математическая модель КЦ ХТС представляет собой объединение математических моделей следующих множеств

$$M_C^{ХТС} = \left[\bigcup_{i^G} M_{Ki}^G \cup \bigcup_{k^G} M_{Nk}^G \cup \bigcup_{i^T} M_{Ki}^T \cup \bigcup_{k^T} M_{Nk}^T \cup \bigcup_{i^C} M_{Ki}^C \right], \quad (3.28)$$

где M_{Ki}^G – множество компонентных уравнений, описывающих гидравлическую составляющую исследуемых процессов, являющуюся независимой в ХТС; M_{Nk}^G – множество топологических уравнений, связывающих между собой гидравлические потоки в узлах, образованных соединением связей компонентов; M_{Ki}^T – множество компонентных уравнений, предназначенных для описания протекающих в ХТС тепловых процессов, зависящих от значений соответствующих гидравлических переменных компонентов; M_{Nk}^T – совокупность топологических уравнений, составленных относительно тепловых потоков в узлах КЦ рассматриваемой ХТС; M_{Ki}^C – множество компонентных уравнений, учитывающих концентрации веществ в выходящих связях компонентов ХТС.

Математической модели КЦ ХТС (3.28) соответствует ее векторно-матричное представление, получаемое универсальным вычислительным ядром на каждой итерации ВЭ, алгоритм которого представлен на рисунке 2.4

$$\begin{bmatrix} \Phi_{KN}^G & \Phi_{KB}^G & 0 & 0 & 0 \\ 0 & A_T^G & 0 & 0 & 0 \\ \Phi_{KN}^{GT} & \Phi_{KB}^{GT} & \Phi_{KN}^T & \Phi_{KB}^T & 0 \\ 0 & 0 & 0 & A_T^T & 0 \\ \Phi_{KN}^{GC} & \Phi_{KB}^{GC} & \Phi_{KN}^{TC} & \Phi_{KB}^{TC} & \Phi_{KN}^C \end{bmatrix} \cdot \begin{bmatrix} P \\ G \\ T \\ Q \\ C \end{bmatrix} = \begin{bmatrix} W^G(t) \\ 0 \\ W^T(t) \\ 0 \\ W^C(t) \end{bmatrix}, \quad (3.29)$$

где Φ_{KN}^G, Φ_{KB}^G – коэффициенты при потенциальных P и потоковых G переменных в компонентных уравнениях, описывающих гидравлические процессы в исследуемой ХТС; A_G^T – матрица коэффициентов в топологических уравнениях гидравли-

ческой составляющей ХТС, составленных относительно потоков вещества G , которые далее будут называться *компонентными гидравлическими уравнениями*; $\Phi_{KN}^{GT}, \Phi_{KB}^{GT}$ – матрицы коэффициентов при гидравлических переменных в уравнениях, описывающих протекающие в ХТС термодинамические процессы и называемых далее *компонентными термодинамическими уравнениями*; Φ_{KN}^T, Φ_{KB}^T – матрицы коэффициентов при термодинамических переменных (температурах T и тепловых потоках Q) в компонентных термодинамических уравнениях; A_T^T – матрица коэффициентов в топологических уравнениях вида (2.18), связывающих в узлах входящие и выходящие термодинамические потоки; $\Phi_{KN}^{GC}, \Phi_{KB}^{GC}$ – матрицы коэффициентов при гидравлических потенциальных и потоковых переменных в уравнениях, описывающих динамику изменения концентраций веществ в каждом компоненте ХТС в результате протекающих в нем массообменных и (или) химических процессов; $\Phi_{KN}^{TC}, \Phi_{KB}^{TC}$ – матрицы коэффициентов при термодинамических переменных в уравнениях динамики изменения концентраций веществ в компонентах ХТС; Φ_{KN}^C, Φ_{KB}^C – матрицы коэффициентов при переменных, описывающих концентрации веществ в потоках между компонентами ХТС.

Коэффициенты матриц компонентных уравнений $\Phi_{KN}^G, \Phi_{KB}^G, \Phi_{KN}^{GT}, \Phi_{KB}^{GT}, \Phi_{KN}^T, \Phi_{KB}^T, \Phi_{KN}^{GC}, \Phi_{KB}^{GC}, \Phi_{KN}^{TC}, \Phi_{KB}^{TC}, \Phi_{KN}^C, \Phi_{KB}^C$ зависят от времени, схемы и шага интегрирования.

При решении системы уравнений (3.29) на каждом шаге ВЭ, реализуемого в универсальном вычислительном ядре, соответствующие переменные вектора решения

$$V^{\Phi_{XC}} = \begin{bmatrix} P \\ G \\ T \\ Q \\ C \end{bmatrix} \quad (3.30)$$

получают искомые значения.

Таким образом, на основе алгоритма формирования и анализа КЦ с НВС реализуется аппарат компьютерного моделирования ХТС.

3.4 Набор компонентов химико-технологических систем

Согласно описанному в пункте 2.4 языку моделирования ХТС в набор компонентов химико-технологических систем входят компоненты-источники, компоненты-преобразователи и компоненты-измерители информационных, мультифизических энергетических и многокомпонентных вещественных потоков. Ранее в рамках метода компонентных цепей [17] и основанной на нем среды моделирования MARC [20] были достаточно подробно рассмотрены и реализованы источники, преобразователи и измерители информационных сигналов и энергетических потоков различной физической природы. В данном разделе рассмотрены компоненты, позволяющие анализировать преобразование физических свойств и компонентных составов многокомпонентных вещественных потоков.

3.4.1 Источник смеси веществ

Для моделирования подачи в моделируемую ХТС многокомпонентной смеси веществ используется компонент «Источник смеси веществ», представленный на рисунке 3.6. Он обладает единственной связью $S_1 = (b_1 \eta^+ n_1) \rightarrow \{\{P_1, G_1\}, \{T_1, Q_1\}, \{C_1\}\}$, где P_1 – давление в узле n_1 , G_1 – объемный расход многокомпонентной смеси веществ в потоке, образованном ветвью b_1 , T_1 – температура в узле n_1 ; Q_1 – тепловой поток в потоке ветви b_1 , C_1 – вектор концентраций веществ, составляющих многокомпонентный поток.

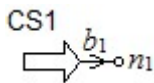


Рисунок 3.6 –
Компонент
«Источник смеси
веществ»

В свойствах компонента задаются следующие параметры и характеристики: P^{DEF} – значение давления в узле n_1 ; G^{DEF} – значение объемного потока многокомпонентной смеси в ветви b_1 ; T^{DEF} – значение температуры в узле n_1 ; N_c – количество компонент (веществ), образующих многокомпонентный вещественный поток; **Names** – совокупность имен веществ, составляющих поток, каждому из которых соответствует определенная строка параметров в базе данных компонент потока;

C^{DEF} – вектор концентрации веществ, образующих многокомпонентный вещественный поток.

Математическая модель компонента «Источник смеси веществ» включает в себя следующие уравнения

$$\begin{aligned} P_1 &= P^{\text{DEF}} \\ G_1 &= G^{\text{DEF}} \\ T_1 &= T^{\text{DEF}}, \\ Q_1 &= Q^{\text{DEF}} \\ C_{li} &= C_{li}^{\text{DEF}} \end{aligned} \quad (3.31)$$

где $Q^{\text{зад}}$ – тепловой поток, который рассчитывается по формуле

$$Q^{\text{DEF}} = G \cdot T \cdot \sum_{i=1}^{N_c} \left(\rho_i \cdot CP_i \cdot (C_i^{\text{DEF}})^2 \right), \quad (3.32)$$

где ρ_i – плотность вещества $Name_i$, выбираемая по его имени из БД компонент потока; CP_i – теплоемкость соответствующего вещества, которая представляет собой зависимость температуры T

$$CP_i = CPVAP_A_i + CPVAP_B_i \cdot T + CPVAP_C_i \cdot T^2 + CPVAP_D_i \cdot T^3, \quad (3.33)$$

где $CPVAP_A_i, CPVAP_B_i, CPVAP_C_i, CPVAP_D_i$ – константы в уравнении идеально-газовой плотности, в котором CP выражена в кал/(моль*К). Данные константы задаются для каждого вещества в одноименных полях БД компонент потока.

3.4.2. Измерительные компоненты характеристик

МНОГОКОМПОНЕНТНЫХ ВЕЩЕСТВЕННЫХ ПОТОКОВ

В любом узле, образованном соединением двух и более компонентов ХТС, многокомпонентный вещественный поток характеризуется давлением, массовым (объемным или молярным) расходом вещества, температурой и тепловым потоком, а также вектором концентраций веществ, входящих в рассматриваемый поток. Каждый измерительный компонент должен осуществлять выбор значения соответствующей переменной из вектора решения (2.15) и его передачу с объектного уровня многоуровневой КЦ на её логический уровень (2.50).

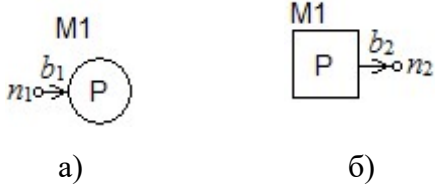


Рисунок 3.7 – Компонент «Манометр»
 а) отображение на объектном уровне
 б) отображение на логическом уровне

Измерение давления осуществляется измерительным компонентом «Манометр». На объектном уровне он отображается в виде, представленном на рисунке 3.7, а, а также на логическом уровне в виде, приведенном на рисунке 3.7, б. Единственная связь компонента на объектном уровне, являясь неоднородной векторной

$$S_1 = (b_1 \eta^- n_1) \rightarrow \{\{P_1, \emptyset\}, \{\emptyset, \emptyset\}, \{\emptyset\}\}, \quad (3.34)$$

имеет одну потенциальную переменную P_1 . Его связь на логическом уровне

$$S_2 = (b_2 \eta^+ n_2) \rightarrow \{V\} \quad (3.35)$$

содержит переменную, которой осуществляется передача значения другим подсоединенным к данной связи компонентам логического уровня. Математическая модель компонента «Манометр» имеет вид

$$V = P_1. \quad (3.36)$$

Для измерения скорости протекания многокомпонентного вещественного

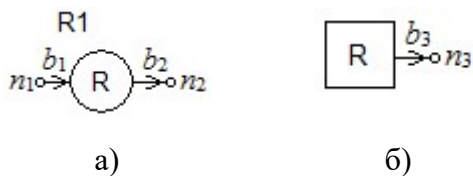


Рисунок 3.8 – Компонент «Расходомер»
 а) отображение на объектном уровне
 б) отображение на логическом уровне

потока используется компонент «Расходомер». На объектном уровне многоуровневой КЦ он имеет отображение, представленное на рисунке 3.8, а. Его отображение на логическом уровне имеет вид,

представленный на рисунке 3.8, б. Отображение на объектном уровне имеет две связи

$$\begin{aligned} S_1 &= (b_1 \eta^- n_1) \rightarrow \{\{P_1, G_1\}, \{T_1, Q_1\}, \{C_1\}\}, \\ S_2 &= (b_2 \eta^+ n_2) \rightarrow \{\{P_2, G_2\}, \{T_2, Q_2\}, \{C_2\}\}. \end{aligned} \quad (3.37)$$

Для передачи измеренного значения расхода вещества на логическом уровне многоуровневой КМ компонент «Расходомер» имеет связь

$$S_3 = (b_3 \eta^+ n_3) \rightarrow \{V\}. \quad (3.38)$$

Его математическая модель включает в себя следующие соотношения

$$\begin{aligned}
 P_1 - P_2 &= 0 \\
 G_1 - G_2 &= 0 \\
 T_1 - T_2 &= 0 \\
 Q_1 - Q_2 &= 0 \\
 C_{1i} - C_{2i} &= 0, \quad \forall i = \overline{1, CN} \\
 V &= G_1
 \end{aligned}
 \tag{3.39}$$

Измерение текущего значения температуры протекающего между компо-

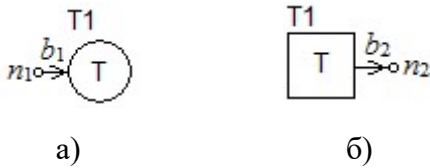


Рисунок 3.9 – Компонент «Термометр»
 а) отображение на объектном уровне
 б) отображение на логическом уровне

нентами многокомпонентного вещественного потока осуществляется измерительным компонентом «Термометр». Его отображение на объектном уровне многоуровневой КЦ имеет вид, представленный на

рисунке 3.9, а, на её логическом уровне он имеет вид, приведенный на рисунке 3.9, б. Связь компонента «Термометр» на объектном уровне

$$S_1 = (b_1 \eta^- n_1) \rightarrow \{ \{ \emptyset, \emptyset \}, \{ T, \emptyset \}, \{ \emptyset \} \}
 \tag{3.40}$$

включает в себя только одну переменную T – температуру, измеряемую в Кельвинах (К). На логическом уровне многоуровневой КЦ он имеет связь вида

$$S_2 = (b_2 \eta^+ n_2) \rightarrow \{ V \},
 \tag{3.41}$$

где V – переменная, с помощью которой текущее значение температуры передается другим компонентам логического уровня.

Математическая модель компонента «Термометр» имеет вид

$$V = T.
 \tag{3.42}$$

Измерение и передача с объектного на логический уровень теплового потока осуществляется компонентом «Датчик теплового потока». На объектном уровне он имеет вид, представленный на рисунке 3.10, а, его вид на логическом уровне приведен на рисунке 3.10, б. Связи компонента «Датчик теплового потока» на объектному

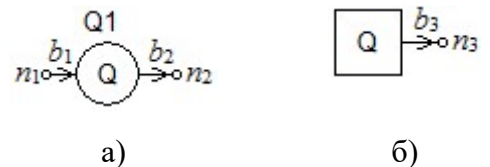


Рисунок 3.10 – Компонент «Датчик теплового потока»
 а) отображение на объектном уровне
 б) отображение на логическом уровне

уровню многоуровневой КМ имеют вид (3.37). Его связь на её логическом уровне

имеет вид (3.38), а математическая модель совпадает с моделью (3.39) за исключением последнего уравнения, которое имеет вид

$$V = Q_1 . \quad (3.43)$$

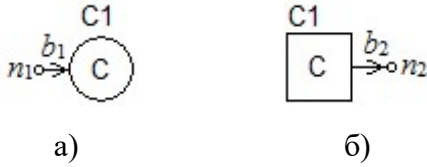


Рисунок 3.11 – Компонент «Датчик концентрации»

- а) отображение на объектном уровне
б) отображение на логическом уровне

Компонент «Датчик концентрации» осуществляет выборку из вектора решения (2.15) и передачу с объектного уровня многоуровневой КЦ на её логический уровень текущее значение концентрации выбранного вещества в узле, к которому он прикреп-

лен своим изображением, представленном на рисунке 3.11, а, на объектном уровне. На логическом уровне данный компонент имеет вид, представленный на рисунке 3.11, б. Связь компонента «Датчик концентрации» на объектном уровне

$$S_1 = (b_1 \eta^- n_1) \rightarrow \{ \{ \emptyset, \emptyset \}, \{ \emptyset, \emptyset \}, \{ C \} \} \quad (3.44)$$

включает в себя только вектор концентраций. Из него выбирается концентрация вещества, указанного в поле «Вещество» свойств данного компонента. По порядковому номеру этого вещества i осуществляется выборка соответствующего значения из вектора решения (2.15)

$$V = C_i \quad (3.45)$$

и его передача другим компонентам логического уровня МКЦ с помощью связи

$$S_2 = (b_2 \eta^+ n_2) \rightarrow \{ V \} . \quad (3.46)$$

3.4.3. Комплексное гидротермодинамическое сопротивление многокомпонентному вещественному потоку

Гидротермодинамическое сопротивление многокомпонентному вещественному потоку, представленное на рисунке 3.12, выражает собой трубопровод, по которому осуществляется передача потока в режиме ламинарного или турбулентного течения [38]. Его связи описываются следующими переменными:

– входная связь $S_1 = (b_1 \eta^- n_1) \rightarrow \{\{P_1, G_1\}, \{T_1, Q_1\}, \{C_1\}\}$ описывается входным давлением P_1 , объемным расходом многокомпонентной смеси веществ G_1 , температурой T_1 , тепловым потоком Q_1 и вектором концентраций входящих в смесь веществ C_1 ;

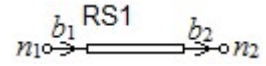


Рисунок 3.12 – Гидротермодинамическое сопротивление потоку

– выходная связь $S_2 = (b_2 \eta^+ n_2) \rightarrow \{\{P_2, G_2\}, \{T_2, Q_2\}, \{C_2\}\}$ описывается аналогичными переменными связи.

Математическая модель компонента «Гидротермодинамическое сопротивление» имеет вид

$$\begin{aligned} P_1 - P_2 - \lambda \cdot \rho \cdot \frac{L}{d} \cdot \frac{G_1^2}{(\pi \cdot d^2)^2} &= 0 \\ T_1 - T_2 - R_t \cdot Q_1 &= 0 \\ G_1 - G_2 &= 0 \\ Q_1 - Q_2 &= 0 \\ C_{1i} - C_{2i} &= 0 \end{aligned} \quad , \quad (3.47)$$

где L – длина трубы; d – диаметр; λ – коэффициент потерь на трение по длине трубы

$$\lambda = \begin{cases} \frac{64}{\text{Re}} & \text{Re} < 2300 \\ \frac{0.316}{\sqrt[4]{\text{Re}}} & 2300 < \text{Re} < 10^5 \\ 0.0032 + \frac{0.221}{\text{Re}^{0.273}} & 10^5 < \text{Re} < 10^6 \end{cases} \quad , \quad (3.48)$$

Re – число Рейнольдса, рассчитываемое по формуле

$$\text{Re} = \frac{G \cdot D}{\nu \cdot A} \quad , \quad (3.49)$$

где D – гидравлический диаметр, рассчитываемый для каждого вида труб по определенной формуле [136]; ν – динамическая вязкость среды, которая для каждого вещества является справочной величиной и задается в базе данных компо-

нент потока в поле *Viscosity*. Для многокомпонентного вещественного потока она вычисляется по формуле

$$v = \sum_i^{N_c} v_i \cdot C_{li}; \quad (3.50)$$

A – площадь поперечного сечения трубопровода, которая зависит от его геометрических свойств (например, для круглого – $A = \pi \cdot d^2$); ρ – плотность многокомпонентного вещественного потока, которая рассчитывается по формуле

$$\rho = \sum_{i=1}^{N_c} \rho_i \cdot C_i, \quad (3.51)$$

где ρ_i – плотность соответствующего вещества, входящего в многокомпонентный вещественный поток с концентрацией C_i . Она задается в БД компонент потока в поле Ro ; R_i – тепловое сопротивление.

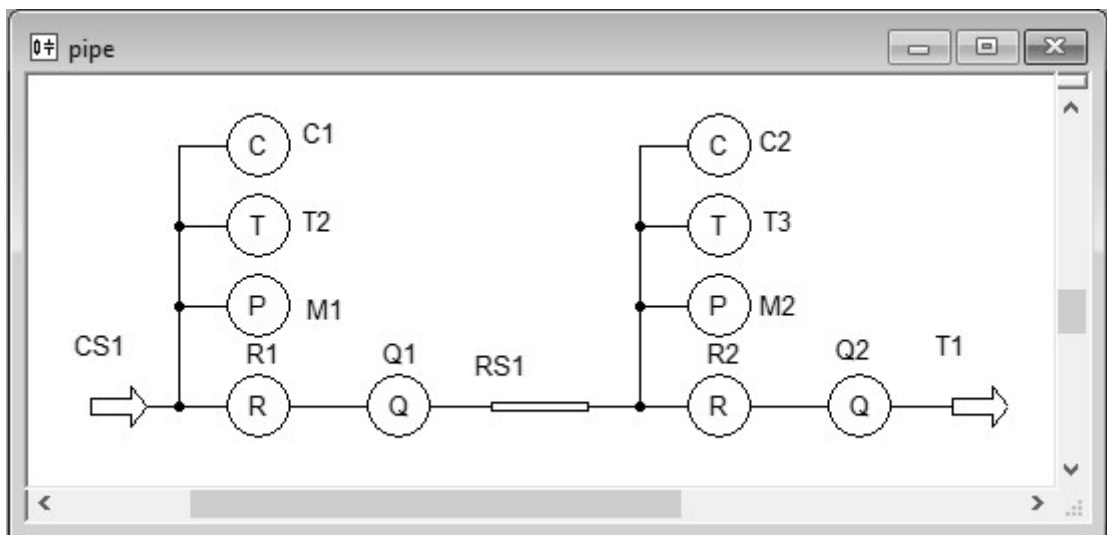


Рисунок 3.13 – Компьютерная модель участка трубопровода

На рисунке 3.13 приведена компьютерная модель участка трубопровода, в котором труба выражена гидротермосопротивлением $RS1$. Его геометрическими параметрами являются длина $L = 5$ м, диаметр $d = 0.1$ м, тип сечения – круглое сечение, термосопротивление – $10 \frac{^{\circ}C \cdot c}{Дж}$. С помощью источника $CS1$ подается

вода под давлением $P = 100$ кПа, температурой $T = 373$ К и расходом $G = 500$ моль/с. Тепловой поток рассчитывается по формуле (3.32).

На рисунке 3.14 представлены результаты анализа представленного участка трубопровода.

P_1	R_1	T_1	Q_1	C_1
1.000e+006	5.000e+002	373.000	1.525e+009	1.000
P_2	R_2	T_2	Q_2	C_2
8.701e+005	5.000e+002	373.000	1.525e+009	1.000

Рисунок 3.14 – Результаты анализа компьютерной модели участка трубопровода

3.4.4 Смеситель

Для смешивания двух многокомпонентных потоков в химической технологии используются смесители. В данный аппарат под различным давлением и обладающие различными температурами поступают два потока различного молярного расхода.

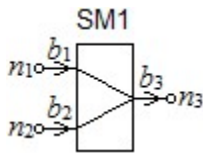


Рисунок 3.15 –
Компонент
«Смеситель»

Компонент «Смеситель», представленный на рисунке 3.15. Он включает в себя следующие связи:

$$- S_1 = (b_1 \eta^- n_1) = \{ \{P_1, G_1\}, \{T_1, Q_1\}, \{C_1\} \} - \text{связь, по которой}$$

подается первый подлежащий смешению поток;

$$- S_2 = (b_2 \eta^- n_2) = \{ \{P_2, G_2\}, \{T_2, Q_2\}, \{C_2\} \} - \text{связь, по кото-}$$

рой в смеситель поступает второй поток;

$$- S_3 = (b_3 \eta^+ n_3) = \{ \{P_3, G_3\}, \{T_3, Q_3\}, \{C_3\} \} - \text{связь, по которой из смесителя}$$

выводится поток, содержащий компоненты первого и второго потока.

Математическая модель компонента «Смеситель» включает в себя следующие уравнения

$$\begin{aligned}
 P_1 + P_2 - P_3 &= 0 \\
 G_1 + G_2 - G_3 &= 0 \\
 G_1 \cdot T_1 + G_2 \cdot T_2 - G_3 \cdot T_3 &= 0 \quad . \\
 Q_1 \cdot T_1 + Q_2 \cdot T_2 - Q_3 \cdot T_3 &= 0 \\
 C_{1i} \cdot G_1 + C_{2i} \cdot G_2 - C_{3i} \cdot G_3 &= 0
 \end{aligned}
 \tag{3.52}$$

На рисунке 3.16 представлена компьютерная модель секции смешения двух вещественных потоков, находящихся в газовой фазе. С помощью источника CS1 в смеситель со скоростью 200 моль/с под давление 200 кПа подается смесь метана и пропана температурой 300 К в пропорциях 0,75:0,25. Источником CS2 под давлением 100 кПа со скоростью 100 моль/с поступает смесь метана и этана в пропорциях 0,4:0,6, которая имеет температуру 350 К. На рисунке 3.17 приведены результаты анализа модели секции смешения

двух вещественных потоков.

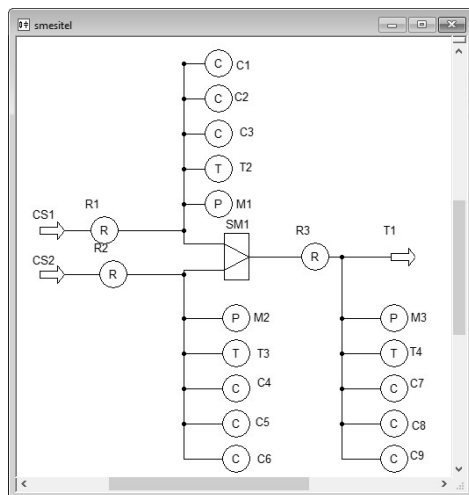


Рисунок 3.16 – Компьютерная модель секции смешения двух вещественных потоков

P_1	G_1	T_1	C1_CH4	C1_C2H5	C1_C3H8
2.000e+005	200.000	300.000	0.750	0.250	0.000
P_2	G_2	T_2	C2_CH4	C2_C2H5	C2_C3H8
1.000e+005	100.000	359.000	0.400	0.000	0.600
P_out	G_out	T_3	Cout_CH4	Cout_C2H5	Cout_C3H8
3.000e+005	300.000	319.667	0.633	0.167	0.200

Рисунок 3.17 – Результаты анализа компьютерной модели секции смешения двух вещественных потоков

3.4.5 Теплообменник

Теплообменник – это аппарат ХТС, в котором осуществляется теплообмен между двумя теплоносителями – многокомпонентными вещественными потоками, находящимися в газовой или жидкой фазе и имеющими различную температуру. В нем осуществляется перенос теплоты (теплоперенос) от более нагретого теплоносителя, называемого далее *горячим*, к менее нагретому – холодному, через разделяющую их стенку. Принимая во внимание, что теплообмен происходит по всей длине прямоугольного теплообменника, изменение температур горячего и

холодного теплообменника описывается системой дифференциальных уравнений в частных производных по длине теплообменника [38]

$$\begin{aligned} \frac{\partial T^H}{\partial l} &= \frac{K \cdot d}{G^H \cdot CP^H(T^H(l)) \cdot \rho^H} \cdot (T^H(l) - T^C(l)) \\ \frac{\partial T^C}{\partial l} &= -\frac{K \cdot d}{G^C \cdot CP^C(T^C(l)) \cdot \rho^C} \cdot (T^H(l) - T^C(l)) \end{aligned} \quad (3.53)$$

где K – коэффициент теплопередачи; d – ширина площади разделяющей стенки; G^H, G^C – молярные расходы горячего и холодного теплоносителя; ρ^H, ρ^C – плотности горячего и холодного теплоносителя соответственно, вычисляемые по формуле (3.51); CP^H, CP^C – удельные молярные теплопроводности теплоносителей, рассчитываемые по формуле [131]

$$CP(T) = \sum_{i=1}^{N_c} CP_i(T) \cdot C_i, \quad (3.54)$$

где CP_i – удельные теплопроводности веществ, входящих в соответствующий теплоноситель, рассчитываемые по формуле (3.32); C_i – концентрации веществ в этом теплоносителе.

Изменение тепловых потоков горячего и холодного теплоносителей соответственно по всей длине разделяющей стенки описывается уравнениями

$$\begin{aligned} Q^H(0) - Q^H(L) &= \rho^H \cdot G^H \cdot (CP^H(T^H(L)) \cdot T^H(L) - CP^H(T^H(0)) \cdot T^H(0)) \\ Q^C(0) - Q^C(L) &= \rho^C \cdot G^C \cdot (CP^C(T^C(L)) \cdot T^C(L) - CP^C(T^C(0)) \cdot T^C(0)) \end{aligned} \quad (3.55)$$

Реализованный на основе уравнений (3.53)–(3.55) компонент «Теплообменник» представлен на рисунке 3.18. Он обладает следующими связями:



Рисунок 3.18 – Компонент «Теплообменник»

- $S_1 = (b_1 \eta^- n_1) \rightarrow \{ \{ P^H(0), G^H(0) \}, \{ T^H(0), Q^H(0) \} \}$
- $C^H(0)$ – связь, по которой в теплообменник поступает горячий теплоноситель;
- $S_2 = (b_2 \eta^- n_2) \rightarrow \{ \{ P^H(L), G^H(L) \}, \{ T^H(L), Q^H(L) \}, \{ C^H(L) \} \}$ – связь, через которую горячий теп-

лонооситель выходит из аппарата;

$$- S_3 = (b_3 \eta^- n_3) \rightarrow \left\{ \left\{ P_1^C(0), G_1^C(0) \right\}, \left\{ T^C(0), Q^C(0) \right\}, \left\{ C_1^C(0) \right\} \right\} - \text{связь подачи в}$$

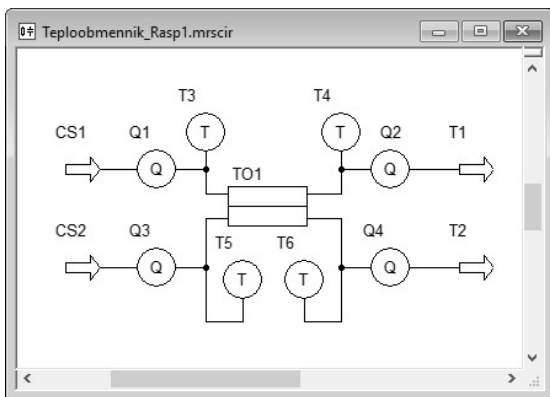
теплообменник холодного теплоносителя;

$$- S_4 = (b_4 \eta^- n_4) \rightarrow \left\{ \left\{ P_2^C(L), G_2^C(L) \right\}, \left\{ T^C(L), Q^C(L) \right\}, \left\{ C_2^C(L) \right\} \right\} - \text{связь, выра-$$

жающая вывод холодного теплоносителя из теплообменника.

Для формирования дифференциального уравнения в частных производных (3.53) воспользуемся описанной ранее методикой линеаризации уравнений (3.21)-(3.23). Разобьем всю длину теплообменника L на ряд отрезков Δl , на каждом из которых теплообмен производится с постоянной скоростью. На каждом из них дифференциальные уравнения вида (3.53) на основе явной схемы Эйлера (2.27) могут быть представлены в алгебраизованном виде

$$\begin{aligned} G^H(0) \cdot CT(T^H(l + \Delta l)) \cdot T^H(l + \Delta l) - (1 - K \cdot d \cdot \Delta l) \cdot T^H(l) - K \cdot d \cdot \Delta l \cdot T^C(l) &= 0 \\ G^C(0) \cdot CT(T^C(l + \Delta l)) \cdot T^C(l + \Delta l) - (1 - K \cdot d \cdot \Delta l) \cdot T^C(l) + K \cdot d \cdot \Delta l \cdot T^H(l) &= 0 \\ Q^H(0) - Q^H(L) - \rho^H \cdot G^H(0) \cdot (CP^H(T^H(L)) \cdot T^H(L) - CP^H(T^H(0)) \cdot T^H(0)) &= 0 \\ Q^C(0) - Q^C(L) - \rho^C \cdot G^C(0) \cdot (CP^C(T^C(L)) \cdot T^C(L) - CP^C(T^C(0)) \cdot T^C(0)) &= 0 \quad . \quad (3.56) \\ P^H(0) - P^H(L) &= 0 \\ G^H(0) - G^H(L) &= 0 \\ C_i^H(0) - C_i^H(L) &= 0 \\ P^C(0) - P^C(L) &= 0 \\ G^C(0) - G^C(L) &= 0 \\ C_i^C(0) - C_i^C(L) &= 0 \end{aligned}$$



а)

Q горячий ДО	Q горячий ПОСЛЕ
5.923e+009	5.791e+009
Q холодный ДО	Q холодный ПОСЛЕ
4.877e+009	5.007e+009
T горячий ДО	T горячий ПОСЛЕ
363.000	348.044
T холодный ДО	T холодный ПОСЛЕ
303.000	318.110

б)

Рисунок 3.19 – Моделирование теплообменника

а) компьютерная модель теплообменника

б) результаты моделирования

На рисунке 3.19, а представлена компьютерная модель теплообменника, в который с помощью источника SC1 под давлением 100000 кПа, объемным расходом 2000 м³/ч и температурой 363 К подается горячий теплоноситель. С помощью источника SC2 под давлением 100000 кПа, объемным расходом 2000 м³/ч и температурой 303 К подается холодный теплоноситель. В качестве горячего и холодного теплоносителя выбрана вода (H₂O) без примесей. Остальные параметры данного вещества приводятся в базе данных веществ. Длина поверхности контакта фаз 10 м, ширина 0.1 м. Коэффициент теплопередачи 3600. Результаты анализа компьютерной модели теплообменника представлены на рисунке 3.19, б.

Совпадение результатов с результатами, полученными в комплексе программ Aspen Hysys, доказывают адекватность реализованной модели теплообменника. В отличие от аналога, реализованный в среде моделирования ХТС компонент допускает интерактивное изменение значений параметров компонента «Теплообменник» во время проведения вычислительного эксперимента.

3.4.6 Сепаратор

Поступающий на установку комплексной подготовки газа (УКПГ) [138] добытый природный газ содержит в себе пластовую воду, различные механические примеси, а также различные компоненты (вещества), находящиеся в газовой или жидкой фазе. Для разделения газовой смеси на различные компоненты потока она проходит последовательно несколько ступеней сепарации: сепаратор первой ступени сепарации (С-1), промежуточный (С-4) и низкотемпературный (С-2). Перед

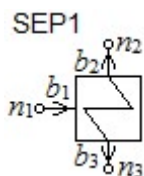


Рисунок 3.20 –
Компонент
«Сепаратор»

сепараторами С-4 и С-2 газ охлаждается в теплообменниках, модель которых приведена в пункте 3.4.4.

Для описания модели сепарации газа при различных давлениях и температурах используется компонент «Сепаратор», представленный на рисунке 3.20, который обладает следующими связями:

– $S_1 = (b_1 \eta^- n_1) \rightarrow \{\{P_{In}, G_{In}\}, \{T_{In}, Q_{In}\}, \{C_{In}\}\}$ – входная связь, по которой газожидкостная смесь под определенным давлением P_{In} и температурой T_{In} поступает в сепаратор;

– $S_2 = (b_2 \eta^+ n_2) \rightarrow \{\{P_G, G_G\}, \{T_G, Q_G\}, \{C_G\}\}$ – выходная связь, предназначенная для вывода газа из сепаратора под давлением P_G температурой T_G . Компонентный состав газа представлен вектором концентраций веществ C_G ;

– $S_3 = (b_3 \eta^+ n_3) \rightarrow \{\{P_L, G_L\}, \{T_L, Q_L\}, \{C_L\}\}$ – выходная связь, по которой из сепаратора отводится жидкость под давлением P_L , которая обладает температурой T_L . Вектор C_L содержит концентрации веществ, составляющих данный поток.

Математическая модель компонента «Сепаратор» представляет систему уравнений относительно переменных его связей вида

$$\begin{aligned}
 P_G - P_{In} &= 0 \\
 P_L - P_{In} &= 0 \\
 G_{In} - G_G - G_L &= 0 \\
 e \cdot G_G - G_L &= 0 \\
 T_G - T_{In} &= 0 \\
 T_L - T_{In} &= 0 \\
 Q_G - Q_{In} &= 0 \\
 Q_L - Q_{In} &= 0 \\
 C_{In_i} - e \cdot C_{G_i} - (1 - e) \cdot C_{L_i} &= 0 \\
 C_{G_i} - K_i \cdot C_{L_i} &= 0
 \end{aligned}
 \tag{3.57}$$

где $i = \overline{1, CF}$, CF – количество компонент (веществ) во входном потоке сепаратора; e – доля пара, рассчитываемая методом золотого сечения путем минимизации целевой функции вида

$$F(e) = \sum_{i=1}^{CF} C_{In_i} \cdot \frac{K_i - 1}{1 + e \cdot (K_i - 1)}, \tag{3.58}$$

K_i – константы фазового равновесия системы «Пар-жидкость», рассчитываемые путем решения уравнения Соава-Редлиха-Квонга [137]. Данная методика реализована в виде метода, приведенного в приложении 6.

На рисунке 3.21 приведена модель секции низкотемпературной сепарации газожидкостной смеси, подающейся под давлением 2 МПа и температурой -80°C . Она имеет следующий компонентный состав: метан – 0.7812; этан – 0.03; пропан – 0.02; изобутан – 0.0048; н-бутан – 0.001; азот – 0.163.

В таблице 3.1 приведено назначение измерителей, включенных в компьютерную модель низкотемпературной сепарации газожидкостной смеси.

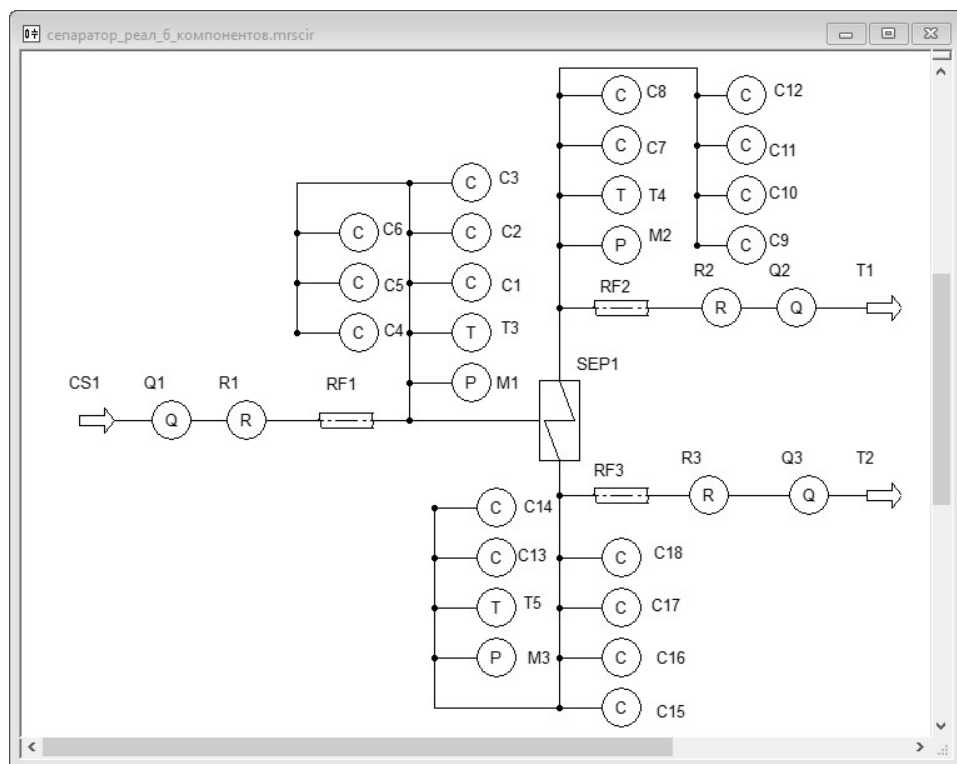


Рисунок 3.21 – Компьютерная модель секции низкотемпературной сепарации природного газа

Таблица 3.1 – Назначение измерителей компьютерной модели секции низкотемпературной сепарации газожидкостной смеси

Физическая величина	Вход	Выход газовой фазы	Выход жидкой фазы
Давление, Па	M1	M2	M3
Молярный расход вещества, моль/с	R1	R2	R3
Температура, К	T1	T2	T3
Тепловой поток, Дж/с	Q1	Q2	Q3

Продолжение таблицы 3.1

Физическая величина	Вход	Выход газовой фазы	Выход жидкой фазы
Концентрация метана	C1	C7	C13
Концентрация этана	C2	C8	C14
Концентрация пропана	C3	C9	C15
Концентрация изобутана	C4	C10	C16
Концентрация н-бутана	C5	C11	C17
Концентрация азота	C6	C12	C18

На рисунке 3.22 приведены результаты анализа секции низкотемпературной сепарации газожидкостной смеси, которые совпали с результатами, приведёнными в статье Р.Р. Акберова [137], при одинаковых значениях входных параметров. Данное совпадение показывает правильность реализации компонента «Сепаратор».

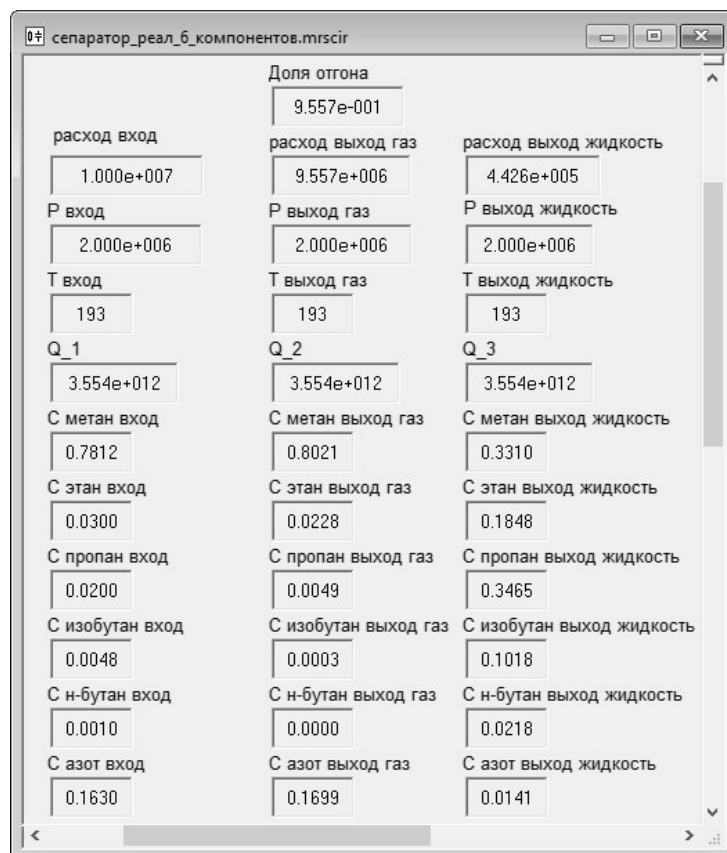


Рисунок 3.22 – Результаты моделирования секции низкотемпературной сепарации газожидкостной смеси

3.4.7 Абсорбер

Для осушки природного газа от паров влаги в технологическом процессе предприятий газовой промышленности нашли широкое применение *абсорберы* [39, 138]. Они являются аппаратами химической технологии, в которых осушка газа производится путем массообмена абсорбируемых компонент между газом и жидким абсорбентом (ингибитором). В его качестве могут использоваться метанол, дэтиленгликоль или другие жидкие вещества.

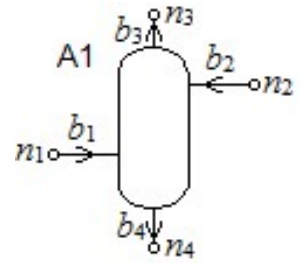


Рисунок 3.23 –
Компонент
«Абсорбер»

Компонент «Абсорбер» [139], представленный на рисунке 3.23, имеет следующие физико-химические связи типа (3.19):

– $S_1 = (n_1 \eta^- b_1) \rightarrow \{\{P_1, G_1\}, \{T_1, Q_1\}, \{C_1\}\}$ – связь, по которой в абсорбер поступает подлежащий осушке газ;

– $S_2 = (b_2 \eta^- n_2) \rightarrow \{\{P_2, G_2\}, \{T_2, Q_2\}, \{C_2\}\}$ – связь, по которой в абсорбер подается ингибитор;

– $S_3 = (b_3 \eta^+ n_3) \rightarrow \{\{P_3, G_3\}, \{T_3, Q_3\}, \{C_3\}\}$ – связь, предназначенная для вывода осушенного газа из абсорбера;

– $S_4 = (b_4 \eta^+ n_4) \rightarrow \{\{P_4, G_4\}, \{T_4, Q_4\}, \{C_4\}\}$ – связь, через которую осуществляется вывод из абсорбера насыщенного влагой ингибитора.

Математическим описанием используемых в абсорберах процессов являются следующие уравнения:

– *уравнение массопередачи*

$$G = \sum_{i=1}^{N_F} \min(\beta_{iy} (y_i - y_{ip}^\Gamma) \cdot F, \beta_{ix} (x_{ip}^\Gamma - x_i) \cdot F), \quad (3.59)$$

где β_{iy}, β_{ix} – коэффициенты массопередачи в отдающей (газовой) и принимающей (жидкой) фазах, выражающие количество вещества, подводимого к единице площади раздела фаз в единицу времени при единичной движущей силе процесса; y_i, x_i – содержание i -го компонента соответственно в газовой и жидкой фазах;

$y_{ip}^\Gamma, x_{ip}^\Gamma$ – равновесные концентрации на границе раздела фаз; F – площадь межфазной поверхности; N_F – количество компонентов-веществ в рассматриваемой ХТС, подлежащих абсорбции.

– уравнения материального баланса абсорбции, выраженного равенством количеств веществ обеих фаз, поступающих в абсорбер и выходящих из него:

$$\begin{aligned} G_1 - G_3 &= G \\ G_4 - G_2 &= G \quad ; \\ P_1 + P_2 - P_3 - P_4 &= 0 \end{aligned} \quad (3.60)$$

– уравнения теплового баланса абсорбции, составленные на основании того, что процесс абсорбции протекает без подвода и отвода сторонней тепловой энергии

$$\begin{aligned} G_1 \cdot Q_1 + G_2 \cdot Q_2 - G_3 \cdot Q_3 - G_4 \cdot Q_4 &= 0 \\ T_1 \cdot Q_1 + T_2 \cdot Q_2 - T_3 \cdot Q_3 - T_4 \cdot Q_4 &= 0 \quad ; \end{aligned} \quad (3.61)$$

– уравнения концентраций на выходе газовой и жидкой фаз абсорбера

$$\begin{aligned} C_{3i} - \frac{G_1 \cdot C_{1i} - G_{Ci}}{G_1 - G} &= 0 \\ C_{4i} - \frac{G_2 \cdot C_{2i} + G_{Ci}}{G_2 + G} &= 0 \end{aligned} \quad (3.62)$$

Уравнения (3.59)–(3.62) лежат в основе реализации математической модели компонента «Абсорбер» в рамках языка моделирования ХТС.

Модель блока абсорбционной осушки природного газа. В функционирующих установках комплексной подготовки газа (УКПГ) Российской Федерации [138] в качестве ингибитора используется метанол или деэтиленгликоль. Рассмотрим пример при использовании метанола. Пусть на вход компонента «Абсорбер», составляющего основу представленной на рисунке 3.24 модели блока абсорбционной осушки природного газа, с помощью источника CS1 под давлением 9,5 МПа молярным расходом 1000 моль/с подается сырой природный газ. Соотношение концентраций газа и воды в подаваемой газовой смеси составляет 0,85:0,15. С целью ее осушки, то есть максимального уменьшения концентрации воды на выходе газовой фазы абсорбера, на вход его жидкой фазы источником

CS2 под давлением 200 кПа молярным расходом 75 моль/с подается ингибитор (метанол), разбавленный водой в соотношении 0,9:0,1. Скорость его подачи составляет 150 моль/с.

Осушка газа проходит в абсорбере с площадью контакта фаз 3,16 м². Коэффициент массопередачи в отдающей и принимающей фазах для воды составляет 0,03. Равновесная концентрация в отдающей фазе равна 0, в принимающей фазе – 0,35.

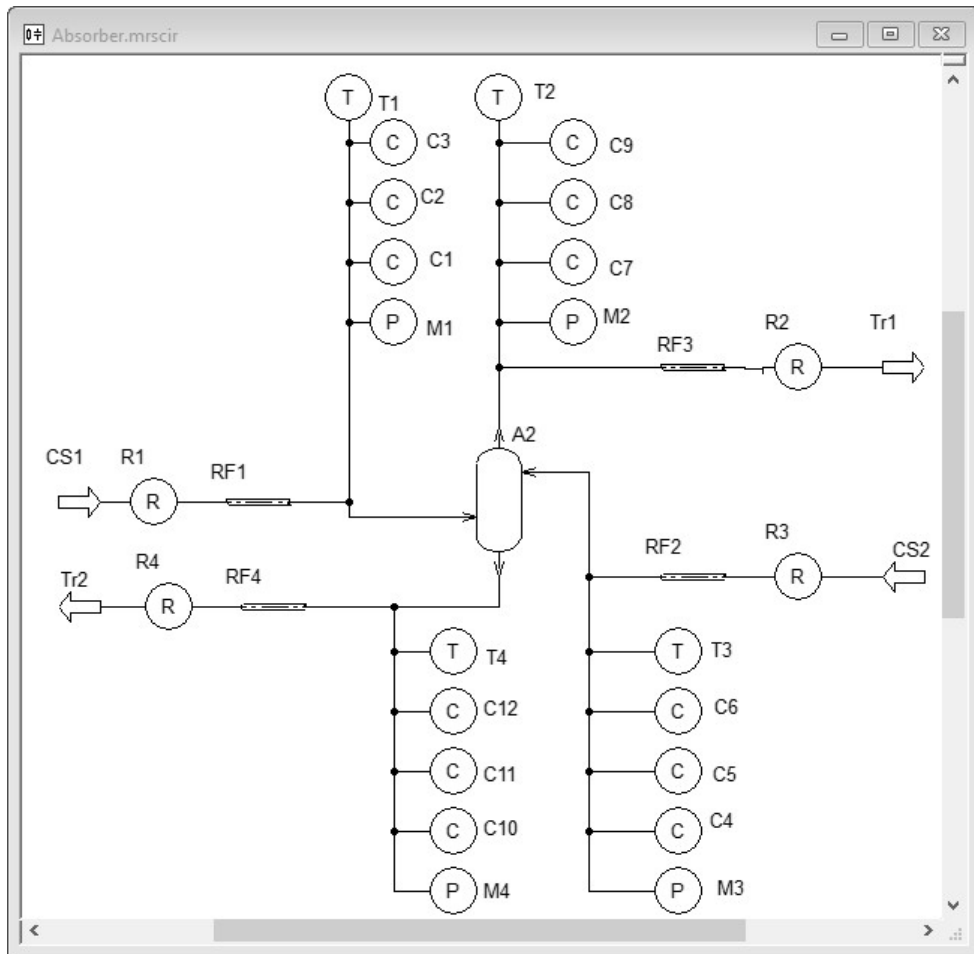


Рисунок 3.24 – Компонентная цепь секции абсорбционной осушки природного газа

В КЦ секции абсорбционной осушки природного газа, представленную на рисунке 3.24, включены манометры, расходомеры и измерители концентраций компонентов во входных и выходных потоках. физический смысл измерительных компонентов приведён в таблице 3.2.

Таблица 3.2 – Назначение измерительных компонентов в компьютерной модели секции абсорбционной осушки природного газа

Назначение компонента	Вход газовой фазы	Вход жидкой фазы	Выход газовой фазы	Выход жидкой фазы
Давление	M1	M2	M3	M4
Объемный расход	R1	R2	R3	R3
Температура	T1	T2	T3	T4
Концентрация газа	C1	C4	C7	C10
Концентрация воды	C2	C5	C8	C11
Концентрация метанола	C3	C6	C9	C12

Результаты анализа приведённой на рисунке 3.24 компонентной цепи секции абсорбционной осушки природного газа представлены на рисунке 3.25

P_G_in	P_L_in	P_G_out	P_L_out
9.500e+006	2.000e+006	9.116e+006	2.384e+006
G_G_in	G_L_in	G_G_out	G_L_out
1.000e+003	9.596e+002	7.500e+001	1.154e+002
T_G_in	T_L_in	T_G_out	T_L_out
3.730e+002	3.730e+002	3.730e+002	3.730e+002
C_CH4_G_in	C_CH4_L_in	C_CH4_G_out	C_CH4_L_out
0.850	0.000	0.886	0.000
C_H2O_G_in	C_H2O_L_in	C_H2O_G_out	C_H2O_L_out
0.150	0.000	0.114	0.350
C_C2H5OH_G_in	C_C2H5OH_L_in	C_C2H5OH_G_out	C_C2H5OH_L_out
0.000	1.000	0.000	0.650

Рисунок 3.25 – Результаты компьютерного моделирования секции абсорбционной осушки природного газа

На примере представленных в данном пункте компьютерных моделей аппаратов химической технологии продемонстрированы возможности языка ХТС, основанного на выполненном развитии метода компонентных цепей, моделирования объектов с распределенными параметрами и переменными. Их особенностью также является зависимость коэффициентов уравнений моделей от значений переменных на каждом шаге анализа. На основе реализованных принципов моделирования могут быть построены компьютерные модели других аппаратов химико-технологических систем, расширяющие возможности программно-алгоритмических средств языка ХТС.

3.5 Выводы

1. На основе системного анализа процессов, протекающих в химико-технологических системах, предложено формализованное представление объектов с неоднородными векторными связями в формате языка ХТС.
2. Разработана структура неоднородной векторной связи, представляющей собой произвольную совокупность энергетических и информационных связей и предназначенной для описания характеристик многокомпонентных веществных потоков. При их соединении образуется векторный узел, через который протекает векторный поток, характеризующийся несколькими видами энергии и информации.
3. Сформирован набор правил коммутации компонентов с неоднородными векторными связями, предусматривающих проверку размерности векторных связей и совпадение физических аспектов определенных энергетических и информационных связей, входящих в состав вступающих в соединение НВС.
4. Предложена обобщенная структура химико-технологического компонента, включающая в себя совокупности входных и выходных физико-химических связей, относящихся к классу неоднородных векторных. Помимо традиционных для метода компонентных цепей топологического, вычислительного и измерительного блоков, она включает в себя блок расчета коэффициентов, являющийся общим для всех компонентов ХТС и связанный в базой данных характеристик компонент потока.
5. Сформирована математическая модель и вектор переменных обобщенного химико-технологического компонента, которая помимо традиционных для метода КЦ линейных, нелинейных и дифференциальных уравнений с обыкновенными производными, входят уравнения в частных производных. Предложена новая методика их линеаризации на этапе формирования программно-алгоритмического вида компонента ХТС.
6. Сформирована и исследована векторно-матричная модель химико-технологической системы, в которой выявлены блоки гидравлических, термоди-

намических уравнений, а также блок уравнений расчета концентраций веществ в выходных потоках каждого компонента.

7. Разработан набор компонентов химико-технологических систем, который включает компоненты-источники, компоненты-преобразователи многокомпонентных вещественных потоков, а также компоненты-измерители их гидравлических и термодинамических характеристик и значений концентраций веществ, составляющих поток. Выявлено, что коэффициенты уравнений, составляющих модели данных компонентов, зависят от значений гидравлических и термодинамических характеристик и рассчитываются на основе справочных данных для веществ, составляющих многокомпонентные потоки.

8. На примере модели теплообменника показаны принципы построения компьютерных моделей компонентов и аппаратов ХТС с распределенными параметрами, уравнения которых содержат частные производные характеристик по геометрическим параметрам аппарата.

9. Модель сепаратора представляет собой пример модели со встроенным в неё блоком расчетов коэффициентов уравнений, которые зависят от значений гидравлических и термодинамических характеристик входного вещественного потока.

10. Разработанные и представленные в данной главе компьютерные модели апробированы путем проведения вычислительного эксперимента и подтвердили свою адекватность при сравнении с литературными источниками и существующими программными комплексами компьютерного моделирования химико-технологических систем.

Глава 4. Язык моделирования алгоритмических конструкций

В главе рассматривается язык моделирования алгоритмических конструкций (язык МАК), предназначенный для формирования принадлежащих логическому уровню многоуровневой компонентной цепи СТУС сценариев решения задач исследования и функционального проектирования сложных технических управляемых систем, а также сценариев функционирования устройств управления, представленных алгоритмическими КЦ. Язык МАК, являясь подязыком языка многоуровневых компонентных цепей, расширяет его на класс дискретно-событийных систем, позволяя формировать и анализировать математико-алгоритмические конструкции. Они могут быть представлены в графической форме, в виде совокупности алгоритмических КЦ (2.63), состоящих из представленных на рисунке 2.11 компонентов-источников, компонентов-преобразователей и компонентов-приемников данных различных типов, так и в текстовой (естественно-математической) форме – в виде совокупности некоторых подпрограмм, которые должны быть автоматически транслированы в формат алгоритмических КЦ.

4.1 Грамматика языка моделирования алгоритмических конструкций

Логический уровень представленной на рисунке 2.5 многоуровневой компонентной цепи СТУС описывает сценарии выполнения экспериментов над КЦ ХТС и включает в себя функциональную модель устройства управления, реализующую сценарии его работы. С целью их выполнения, представляющих собой совокупность шагов преобразования входных данных в совокупность выходных данных [140], эти сценарии должны быть представлены в виде подпрограмм на определенном языке программирования или моделирования.

В настоящее время помимо широко распространенных языков программирования, таких как *C++* [141], *Pascal* [142], *Assembler* [143], *Structure Text (ST)* международного стандарта IEC61131-3 [144, 145] и других, представляющих выполняемые на них программы в текстовой форме в виде набора грамматически правильных конструкций, находят широкое применение графические языки программирования. К ним могут быть отнесены *G*-язык [146], применяющийся при

формировании кодов программ в рамках комплекса программ *LabView* [91, 147], универсальный язык моделирования *UML* [148, 149], используемый при моделировании и проектировании программных комплексов и информационных систем, язык функционального моделирования IDEF-0 [150] и другие. К графическим языкам относятся также языки программирования промышленных контроллеров [151]:

– *Sequential Function Chart (SFC)* – язык программирования промышленных контроллеров, аналогом которого может служить сеть Петри [152] с цветными фишками [153];

– *Ladder Diagram (LD)* – язык релейной (лестничной) логики стандарта МЭК 61131-3 [154], представляющий наглядный интерфейс логики контроллера и облегчающий задачи программирования и ввода в эксплуатацию контроллеров, а также быстрый поиск неполадок в подключаемом к ним оборудовании;

– *Functional Block Diagram (FBD)* – графический язык программирования, предназначенный для формирования алгоритмов и сценариев для программируемых логических контроллеров. Программа представляет собой список цепей, выполняемых последовательно сверху вниз. Цепи могут иметь метки перехода, позволяющие изменять последовательность выполнения цепей при программировании условий и циклов. Допускается использование наборов библиотечных блоков (компонентов) и собственных блоков, написанных на FBD и других языках МЭК 61131-3. Под блоком понимается подпрограмма, функция или функциональный блок. Каждая отдельная цепь представляет собой выражение, составленное из взаимосвязанных графических элементов. К выходу каждого блока подключается следующий блок, образуя цепь. Внутри цепи блоки выполняются строго в порядке их следования. Результат вычисления цепи записывается во внутреннюю переменную или подается на выход программируемого логического контроллера.

Все перечисленные языки программирования и моделирования в той или иной степени позволяют реализовать основные алгоритмические конструкции. Однако их интерпретаторы являются закрытыми и не позволяют интегрировать

написанные в их формате программы со средствами компьютерного моделирования ХТС, язык построения моделей которых был рассмотрен в главе 3.

Таким образом, требуется разработать специальный **язык моделирования алгоритмических конструкций (язык МАК)**, являющийся одним из подязыков языка многоуровневых компонентных цепей, рассмотренного в пункте 2.3.3. Язык МАК предназначен для реализации программ, положенных в основу функционирования устройств управления СТУС, сценарии которых располагаются на логическом уровне многоуровневой КЦ СТУС. Для описания языка МАК использованы нотации Бэкуса-Наура [155], нотации языка представления математических выражений [101, 102, 156] и рассмотренные в главе графические нотации языка МАК, разработанные на основе метода компонентных цепей [17].

Входными данными программы, сформированной на языке МАК, являются (см. рисунок 2.5):

– данные измерения характеристик V_{O_R} реальной ХТС, принадлежащей реальному объектному уровню R_O_L , и результаты анализа модели химико-технологической системы V_{O_M} , составляющей модельный объектный уровень M_O_L . Совместно образуя вектор V_O (2.49), осуществляется их передача с объектного уровня на логический уровень (2.50);

– команды интерактивного управления C_I , поступающие в сценарий с визуального уровня (2.52);

– данные D внешних хранилищ данных, с помощью которых осуществляется параметризация многоуровневой КЦ СТУС.

В результате работы программы или ее отдельных составных частей должны быть получены следующие выходные результаты:

– значения параметров компонентов КМ ХТС P_{O_M} и значения управляющих воздействий на реальную химико-технологическую систему P_{O_R} . Совместно образуя вектор P_O (2.53), они передаются с логического уровня на объектный (2.48);

– подлежащие визуализации результаты выполнения шагов сценария функционирования УУ V_L , сформированного конструкциями языка МАК, передаваемые с логического уровня многоуровневой КЦ СТУС на ее визуальный уровень (2.51);

– информация I , полученная в результате выполнения шагов сценария и предназначенная для добавления ее в документы, формируемые автоматически с помощью генератора отчетных форм [157].

Формально программу, написанную на языке МАК, можно представить в виде

$$\mathbf{V}_{\text{OUT}} = \langle F_{\text{MAC}} \rangle (\mathbf{V}_{\text{IN}}) \quad (4.1)$$

где $\mathbf{V}_{\text{IN}} = [\mathbf{V}_0 \quad \mathbf{C}_1 \quad \mathbf{D}]^T$ – вектор входных данных, являющихся аргументами операторов и функций программы $\langle F_{\text{MAC}} \rangle$; $\mathbf{V}_{\text{OUT}} = [\mathbf{P}_0 \quad \mathbf{V}_L \quad \mathbf{I}]^T$ – вектор выходных данных, получаемых в результате работы программы $\langle F_{\text{MAC}} \rangle$.

В вектора \mathbf{V}_{IN} и \mathbf{V}_{OUT} входят числовые данные (целочисленные, вещественные и комплексные значения), называемые далее *данные численных типов*, а также данные *нечисловых типов* (логические, символьные (строковые) и массивы данных перечисленных типов).

Программа $\langle F_{\text{MAC}} \rangle$, сформированная на языке МАК, представляет собой совокупность последовательно выполняемых процедур

$$\langle F_{\text{MAC}} \rangle \rightarrow \langle F_{\text{MAC}_1} \rangle, \langle F_{\text{MAC}_2} \rangle, \dots, \langle F_{\text{MAC}_N} \rangle, \quad (4.2)$$

позволяющих определять элементы вектора \mathbf{V}_{OUT} . Каждая отдельная процедура $\langle F_{\text{MAC}_i} \rangle$ выражается математико-алгоритмическим выражением

$$\langle F_{\text{MAC}_i} \rangle \rightarrow \langle Expr \rangle, \quad (4.3)$$

где

$$\langle Expr \rangle \rightarrow \langle \text{Операнд} \rangle \langle \text{Оператор} \rangle \langle \text{Операнд} \rangle \langle \text{Операнд} \rangle \quad (4.4)$$

Рассмотрим множества операторов $\langle \text{Оператор} \rangle$ и операндов $\langle \text{Операнд} \rangle$, являющиеся нетерминальными символами грамматики языка МАК. При построе-

нии программ на любом формальном языке программирования любой нетерминальный символ грамматики должен быть представлен одним или совокупностью терминальных символов данного языка.

4.1.1 Операторы языка МАК

Множество операторов $\langle \text{Оператор} \rangle$ включает в себя арифметические $\langle OA \rangle$, логические $\langle LO \rangle$ операторы, операторы сравнения $\langle OC \rangle$, а также оператор присваивания $:=$ и оператор вычисления $=$, определяющие операции для операндов различных типов данных

$$\langle \text{Оператор} \rangle \rightarrow \langle AO \rangle | \langle OC \rangle | \langle LO \rangle | := | = . \quad (4.5)$$

Во множество арифметических операторов $\langle AO \rangle$ из (4.5) входят следующие терминальные символы языка МАК

$$\langle AO \rangle \rightarrow + | - | \cdot | \div , \quad (4.6)$$

реализующие основные арифметические операции над операндами различных типов данных.

Результатами сравнения двух операндов числового типа данных, кроме комплексного, выполняемого с помощью операторов сравнения множества

$$\langle OC \rangle \rightarrow < | > | \leq | \geq | = | \neq , \quad (4.7)$$

являются данные логического типа, принимающие одно из двух значений: *true* – истина, *false* – ложь.

Операторы, принимающие значения логического типа данных, могут вступать в операции отрицания, дизъюнкции и конъюнкции, выраженные в грамматике языка МАК следующими логическими (булевыми) операторами

$$\langle LO \rangle \rightarrow \wedge | \vee | \neg , \quad (4.8)$$

результатами выполнения которых являются данные логического типа.

Ряд из представленных операторов могут распространяться на данные нечисловых типов (например, оператором $+$ из множества $\langle AO \rangle$ (4.6) может обозначаться операция конкатенации строк или сложение соответствующих членов числовых массивов равной длины).

Таким образом, *операторы* множества (4.5) являются специфическими символами языка МАК, указывающими на выполнение тех или иных операций над данными, выраженными *операндами*.

4.1.2 Операнды языка МАК

Каждый операнд, входящий во множество нетерминальных символов $\langle \text{Операнд} \rangle$ языка МАК, может быть одним из следующих нетерминальных символов

$$\langle \text{Операнд} \rangle \rightarrow \langle \text{Константа} \rangle | \langle \text{Переменная} \rangle | \langle \text{Функция} \rangle | \langle \text{Expr} \rangle. \quad (4.9)$$

Целое математико-алгоритмическое выражение $\langle \text{Expr} \rangle$ (4.4), стоящее слева или справа от некоторого оператора множества $\langle \text{Оператор} \rangle$ (4.5), будет являться составным операндом по отношению к данному оператору.

$\langle \text{Константа} \rangle$ является операндом языка МАК, имеющим постоянное значение соответствующего типа данных

$$\langle \text{Константа} \rangle \rightarrow \langle \text{Целочисленная_K} \rangle | \langle \text{Вещественная_K} \rangle | \langle \text{Комплексная_K} \rangle | \langle \text{Строковая_K} \rangle. \quad (4.10)$$

Целочисленная константа $\langle \text{Целочисленная_K} \rangle$ из (4.10) представляет собой набор цифр, который может начинаться со знака «минус»

$$\langle \text{Целочисленная_K} \rangle \rightarrow [-] \langle \text{Цифра} \rangle | [-] \langle \text{Цифра} \rangle \langle \text{Конец_Ц_K} \rangle, \quad (4.11)$$

где

$$\langle \text{Цифра} \rangle \rightarrow 0|1|2|3|4|5|6|7|8|9, \quad (4.12)$$

$$\langle \text{Конец_Ц_K} \rangle \rightarrow \langle \text{Цифра} \rangle | \langle \text{Цифра} \rangle \langle \text{Конец_Ц_K} \rangle. \quad (4.13)$$

Лексема $\langle \text{Цифра} \rangle$ в случае (4.12) определена для десятичной константы. Двоичная константа должна состоять только из цифр 0 и 1. Шестнадцатеричная константа помимо всех цифр (4.12) может содержать в себе следующие буквы латинского алфавита: A, B, C, D, E, F .

Вещественная константа $\langle \text{Вещественная_K} \rangle$ может быть определена через нетерминальный символ $\langle \text{Конец_Ц_K} \rangle$ (4.13) следующим образом

$$\langle \text{Вещественная_}K \rangle \rightarrow [-] \langle \text{Конец_Ц_}K \rangle . \langle \text{Конец_Ц_}K \rangle | \pi | e. \quad (4.14)$$

Комплексной константой является нетерминальный символ грамматики языка МАК, представленный в виде

$$\langle \text{Комплексная_}K \rangle \rightarrow \langle \text{Вещественная_}K \rangle + j \langle \text{Вещественная_}K \rangle. \quad (4.15)$$

В отличие от константы переменная является операндом языка МАК, который имеет определенное имя и некоторое численное значение, независящее от закона ее изменения. В грамматике языка МАК она выражается своим именем

$$\langle \text{Переменная} \rangle \rightarrow \langle \text{Имя} \rangle, \quad (4.16)$$

которое представляет собой совокупность прописных и строчных букв латинского алфавита, цифр и знака подчеркивания «_», начинающуюся с буквы

$$\langle \text{Имя} \rangle \rightarrow \langle \text{Буква} \rangle | \langle \text{Буква} \rangle \langle \text{Конец_имени} \rangle, \quad (4.17)$$

где

$$\langle \text{Буква} \rangle \rightarrow A|B|C|D|\dots|Z|a|b|c|d|\dots|z \quad (4.18)$$

$$\begin{aligned} \langle \text{Конец_имени} \rangle \rightarrow \langle \text{Буква} \rangle | \langle \text{Цифра} \rangle | _ | \langle \text{Буква} \rangle \langle \text{Конец_Имени} \rangle \\ \langle \text{Цифра} \rangle \langle \text{Конец_имени} \rangle | _ | \langle \text{Конец_имени} \rangle \end{aligned} \quad (4.19)$$

Третьим операндом языка МАК является *функция*, являющаяся отображением множества аргументов A на множество значений функции B [155]. В описываемом языке по визуальному отображению функцию в редакторе математико-алгоритмических выражений можно классифицировать на [156]:

1) логарифмоподобные (ЛПФ), которыми являются функции, имеющие определенное символьное обозначение и список аргументов

$$\langle \text{Функция} \rangle \rightarrow \langle \text{Имя_функции} \rangle (\langle \text{список_аргументов} \rangle), \quad (4.20)$$

где

$$\langle \text{Имя_функции} \rangle \rightarrow \langle \text{Имя} \rangle, \quad (4.21)$$

$$\begin{aligned} \langle \text{Список_аргументов} \rangle \rightarrow \langle \text{Аргумент} \rangle \\ | \langle \text{Аргумент} \rangle \langle \text{Список_Аргументов} \rangle \\ \langle \text{Аргумент} \rangle \rightarrow \langle \text{Expr} \rangle \end{aligned} \quad (4.22)$$

2) функции общего вида – функции, каждая из которых имеет специфический графический вид в редакторе математико-алгоритмических конструкций. К ним в частности относятся функции, определенные для операторов числовых типов данных и представленные в таблице 4.1. В них черными квадратиками помечены аргументы, т.е. места ввода соответствующих математико-алгоритмических конструкций. Правила их грамматического разбора и интерпретации в алгоритмические КЦ рассмотрены далее.

Таблица 4.1

Таблица функций общего вида

Знак	Название	Знак	Название
$\sqrt{\blacksquare}$	Корень	$\frac{d}{d\blacksquare}$	Полная производная
$\blacksquare^{\blacksquare}$	Возведение в степень	$\int \blacksquare d\blacksquare$	Определенный интеграл
$\log_{\blacksquare} \blacksquare$	Логарифм по основанию	$\sum_{\blacksquare=\blacksquare} \blacksquare$	Сумма
$\blacksquare!$	Факториал	$\prod_{\blacksquare=\blacksquare} \blacksquare$	Произведение
(\blacksquare)	Взятие в скобки	$\begin{pmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{pmatrix}$	Матрица
$ \blacksquare $	Взятие модуля	$\blacksquare_{\blacksquare}$	Индекс
$\frac{\blacksquare}{\blacksquare}$	Деления (дробь)	$\left\{ \begin{array}{l} \blacksquare \\ \blacksquare \\ \blacksquare \end{array} \right.$	Ветвление

В рассмотренной грамматике языка МАК выделены основные нетерминальные символы и показаны правила их формирования из соответствующих терминальных символов (4.5)-(4.22), а также приведены основные грамматические конструкции (4.2)-(4.4), на основе которых формируются сценарии в формате языка МАК.

4.1.3 Правила формирования выражений языка МАК

Каждое выражение (4.4), записанное в формате языка МАК, принадлежит логическому уровню многоуровневой компонентной цепи СТус, приведенной на

рисунке 2.5. Оно предназначено для вычисления значения некоторой переменной для последующей его передачи на объектный или визуальный уровень многоуровневой КЦ СТУС, либо для отправки в генератор отчетных форм для помещения в отчет.

Вычислить значение такой переменной с помощью математико-алгоритмических конструкций можно следующими способами:

1) *выражением присваивания* – высказыванием в языке МАК, включающим оператор присваивания «:=»

$$\langle \text{Переменная} \rangle := \langle \text{Expr} \rangle; \quad (4.23)$$

2) *выражением условного присваивания*, которое имеет вид

$$\langle \text{Переменная} \rangle := \begin{cases} \langle \text{Expr}_1 \rangle & \langle L_Expr_1 \rangle \\ \langle \text{Expr}_2 \rangle & \langle L_Expr_2 \rangle \\ \dots & \dots \\ \langle \text{Expr}_N \rangle & \langle L_Expr_N \rangle \end{cases}, \quad (4.24)$$

где $\langle \text{Expr}_1 \rangle, \langle \text{Expr}_2 \rangle, \dots, \langle \text{Expr}_N \rangle$ – совокупность математико-алгоритмических выражений, тип результата выполнения которых совпадает с типом переменной, которой присваивается результат операции; $\langle L_Expr_1 \rangle, \langle L_Expr_2 \rangle, \dots, \langle L_Expr_N \rangle$ – совокупность логических выражений, в которых должны быть использованы операторы сравнения (4.7) и могут быть использованы логические операторы (4.8). В результате выполнения выражения (4.24) переменной $\langle \text{Переменная} \rangle$ присваивается результат вычисления выражения $\langle \text{Expr}_i \rangle$ в случае, если логическое выражение $\langle L_Expr_i \rangle$ имеет истинное значение. При этом последующие логические выражения на истинность не проверяются и выражения не вычисляются;

3) *решением уравнения*, сформированного на основе правила

$$\langle \text{Expr}_1 \rangle == \langle \text{Expr}_2 \rangle, \quad (4.25)$$

где $\langle \text{Expr}_1 \rangle$ и (или) $\langle \text{Expr}_2 \rangle$ включают в себя подлежащую определению переменную, сформированную по правилам (4.16);

4) в результате решения *системы уравнений*, включающей в себя N уравнений, сформированных по правилу (4.25), где $N > 1$ – количество подлежащих одновременному определению переменных)

Для случаев многократного выполнения одних и тех же действий (например, для обработки массивов данных) в языке МАК используются циклические конструкции. В широко известных языках программирования [141-145] используются три вида циклов: цикл с параметром (*for*), цикл с предусловием (*while*) и цикл с постусловием (*do while*).

При формировании выражений для циклических конструкций в языке МАК используется синтаксис языка C++ [141], так как в настоящее время данный язык является наиболее распространенным. Его синтаксические конструкции используются во многих языках программирования: php [158], Ruby [159] и других.

В языке МАК выражения для циклов записываются в следующем виде:

1) цикл с параметром

$$\begin{aligned} &for(\langle \text{инит_пар_цикла} \rangle; \langle \text{условие_продолжения} \rangle; \langle \text{шаг} \rangle) \\ &\quad \langle \text{тело_цикла} \rangle \end{aligned} \quad (4.26)$$

где $\langle \text{инит_пар_цикла} \rangle$ – выражение вида (4.23), которым производится инициализация параметра цикла, то есть присваивание ему первоначального значения; $\langle \text{условие_продолжения} \rangle$ – вычисляемое логическое выражение, результат которого принадлежит логическому типу данных. Если он имеет истинное значение, то цикл продолжает работу, если ложное – то цикл прекращает работу и выполняется команда, стоящая после тела цикла; $\langle \text{шаг} \rangle$ – выражение, численный результат которого присваивается параметру цикла после выполнения очередного шага; $\langle \text{тело_цикла} \rangle$ – набор математико-алгоритмических конструкций вида (4.4), вычисляемых (выполняемых) при каждом значении параметра цикла;

2) выражение цикла с предусловием имеет формат

$$\begin{aligned} &while(\langle \text{условие_продолжения} \rangle) ; \\ &\quad \langle \text{тело_цикла} \rangle \end{aligned} \quad (4.27)$$

3) выражением цикла с постусловием:

$$\begin{aligned}
 &do\{ \\
 &\quad \langle \text{тело_цикла} \rangle \quad . \\
 &\}while(\langle \text{условие_продолжения} \rangle)
 \end{aligned}
 \tag{4.28}$$

Циклы с предусловием (4.27) и с постусловием (4.28) выполняются, пока логическое выражение $\langle \text{условие_продолжения} \rangle$ имеет истинное значение. Как только оно приобретает ложное значение *false*, цикл прекращает выполнение тела и управления передается команде, стоящей непосредственно за циклом.

В данном пункте описаны все синтаксические конструкции языка МАК (выражение присваивания, выражение условного присваивания, уравнение, система уравнений, цикл с параметром, цикл с предусловием, цикл с постусловием), необходимые для формирования на логическом уровне МКЦ СТУС сценариев вычислительных экспериментов различного назначения и функционирования устройств управления. С целью их выполнения они должны быть представлены в формате алгоритмических КЦ (2.63). Основной принцип построения алгоритмических КЦ математико-алгоритмических конструкций, сформированных в формате языка МАК, сформулирован и доказан в виде теоремы.

4.2 Правила построения дерева вывода синтаксических конструкций языка МАК

Для выполнения расчетов и сценариев, включающих в себя сценарии функционирования УУ, их необходимо представить в виде взаимосвязанных алгоритмических компонентов.

Для построения алгоритмической компонентной цепи сценария, представленного в формате языка МАК, необходимо, чтобы входящие в него математические операции и алгоритмические конструкции могли быть представлены в виде моделей (2.63).

Любую математико-алгоритмическую конструкцию, содержащую N математических и алгоритмических операций, можно разложить на совокупность N выражений, каждое из которых будет содержать только одну операцию. Такая операция – оператор (4.5) или функция (4.20) – в общем случае производится над

несколькими операндами (переменными или константами определенного типа данных). Она также определяет тип данных получаемого результата. Такую операцию можно представить в виде

$$F(\mathbf{V}) = R, \quad (4.29)$$

где \mathbf{V} – вектор операндов, вступающих в операцию; F – вид выполняемой операции; R – результат операции (4.29).

Операция (4.29) может быть представлена в виде

$$R = F(\mathbf{V}), \quad (4.30)$$

что соответствует обобщенной математической модели компонента алгоритмического уровня (2.64) при $P_A = \emptyset$.

Циклические алгоритмические структуры (4.26)-(4.28) могут быть представлены совокупностью арифметических (4.6), логических операторов (4.8), операторов сравнения (4.7) и функций (4.20), которые представляются в виде (4.30). Следовательно, данные структуры могут быть представлены совокупностью моделей компонентов логического уровня (2.66). Последовательность выполнения таких операций зависит от их ранга, значений ранее выполненных операций сравнения и логических операций.

Таким образом, на основе сходства выражения (4.30) с выражением

$$Y = F_A(P_A, X)$$

можно сделать вывод о том, что каждой операции языка МАК может быть поставлен в соответствие определенный компонент, реализующий ее в формате алгоритмических КЦ. Объединив связи компонентов в соответствии с рангом входящих в выражение операций, на логическом уровне МКЦ СТУС будет построена алгоритмическая КЦ (2.63), предназначенная для вычисления данного выражения.

Поставив на основе теоремы каждому оператору и функции, входящим в рассматриваемое выражение, определенный компонент, этому выражению по определенным правилам можно поставить в соответствие алгоритмическую КЦ (2.63). Ее формирование может производиться пользователем непосредственно в графической форме на логическом уровне многоуровневой КЦ СТУС. Кроме того

это выражение может быть записано в естественно-математической форме для целей его применения в интерактивных математико-алгоритмических панелях логического уровня. В этом случае для построения алгоритмических КЦ применяется интерпретатор МАК, работа которого основана на правилах построения дерева вывода синтаксических конструкций языка МАК и его формализованном отображении в формат алгоритмических компонентных цепей.

Выполнение операций языка МАК осуществляется по следующим правилам:

1. Каждому оператору из множества (4.5) присваивается определенный целочисленный ранг. Выполнение выражения начинается с оператора, имеющего самый низкий ранг при условии, если значения всех его операндов известны, либо для них получены ранее выражения, вычисляющие их при заданных значениях операндов.

2. Вычисление функций (4.20) производится в случае, когда все ее аргументы определены, либо имеются выражения для их нахождения.

3. Операция присваивания (4.23)-(4.24) выполняется в случае, когда определено значение, подлежащее присваиванию переменной, то есть, получены численные или логические значения правых частей выражений присваивания.

4. Уравнение (4.25) может быть представлено в виде алгоритмической КЦ (2.63), если его левому $\langle Expr_1 \rangle$ и правому $\langle Expr_2 \rangle$ выражениям будут поставлены в соответствие определенные алгоритмические компонентные подцепи.

5. Каждая операция цикла (4.26)-(4.28) выполняется по определенному алгоритму, в котором выполнению подлежат все введенные в него нетерминальные символы грамматики.

Последовательно применяя сформулированные правила, а также правила грамматики (4.3)-(4.28) в ходе разбора каждого выражения (4.4) можно получить дерево его синтаксического разбора. В таком дереве узлы помечаются определенными нетерминальными символами грамматики, листья – ее терминальными символами, а каждая связь между узлами или между узлами и соответствующими листьями – выражена одним из правил грамматики. Полученное дерево вывода содержит избыточную информацию, включающую в себя все нетерминальные символы, находящиеся в узлах дерева. После того, как дерево построено, они не несут никакой смысловой нагрузки и не представляют интереса для дальнейшей работы. Для автоматизированного построения выполняемого кода, в данном случае – алгоритмической КЦ (2.63), может быть использовано дерево операций [155, 160]. Его узлами являются математико-алгоритмические операции – операторы (4.5) и функции (4.20), а листьями – его операнды: константы (4.10) и переменные (4.16)

С целью построения алгоритмов отображения дерева операций, пример которого представлен на рисунке 4.1, в формат алгоритмических КЦ выделим его следующие множества:

- 1) множество листьев дерева

$$L = \{L_1, L_2, L_3, L_4, L_5\} \tag{4.31}$$

где $L_1 = 3, L_2 = x, L_3 = 2, L_4 = 2, L_5 = x$;

- 2) множество операций рассматриваемого выражения, образованное находившимися в вершинах дерева вывода операторами и функциями рассматриваемого выражения

$$A = \{A_1, A_2, A_3, A_4, A_5\} \tag{4.32}$$

где $A_1 = ".", A_2 = "/", A_3 = "\cdot", A_4 = "sin", A_5 = "="$;

- 3) множество узлов дерева вывода, каждый из которых образован на соответствующей дуге графа

$$N = \{N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8, N_9\}. \tag{4.33}$$

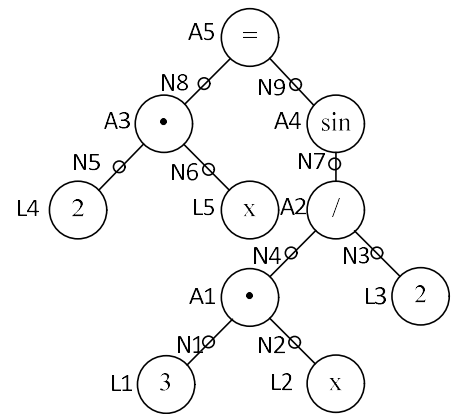


Рисунок 4.1 – Дерево вывода математического выражения

$$2 \cdot x = \sin\left(\frac{3 \cdot x}{2}\right)$$

Каждому из них соответствует переменная V_i , $i = 1..9$ вектора результатов

$$\mathbf{V} = [V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9] \quad (4.34)$$

промежуточных операций, содержащая результат конкретной промежуточной операции.

Таким образом, дерево вывода математико-алгоритмического выражения содержит множество листьев (4.31), представленных константами и переменными, множество вершин (4.32), каждая из которых определяет определенную операцию над соответствующими операндами, а также множество узлов (4.33), обозначающих результаты каждой выполняемой промежуточной операции. При построении алгоритмических КЦ для каждого из представленных множеств (4.31)-(4.33) найдено однозначное отображение во множества K_A, B_A и N_A алгоритмической КЦ (2.63).

4.3 Формализованное отображение дерева вывода математико-алгоритмического выражения в алгоритмическую компонентную цепь

Каждому математико-алгоритмическому выражению (4.3), сформированному в формате языка МАК, будет соответствовать определенное дерево вывода. На его основе для такого выражения может быть построена алгоритмическая КЦ (2.63), если:

1) каждой константе (4.10) или переменной (4.16) из множества L (4.31), значение которой известно, будет поставлен в соответствие компонент-источник, представленный на рисунке 2.11, а;

2) каждой переменной (4.16) из множества L (4.31), значение которой подлежит определению, будет соответствовать компонент-приемник, представленный на рисунке 2.11, в;

3) каждой операции множества A (4.32), к которым относятся операторы (4.5) и функции (4.20), будет найдено отображение в виде компонента-преобразователя, представленного на рисунке 2.11, б, или алгоритмической КЦ (2.63), содержащей произвольное количество взаимосвязанных компонентов множества K_A ;

4) множеству узлов дерева вывода N (4.33) будет соответствовать множество узлов алгоритмической КЦ N_A , а каждому значению вектора V (4.34) – сообщение множества M , передаваемое по определенной связи цепи.

Если всем переменным, входящим в рассматриваемое математико-алгоритмическое выражение (4.4), соответствуют некоторые заданные значения определенного типа данных, то по сформированной на его основе алгоритмической КЦ (2.63) можно рассчитать значение данного выражения, не прибегая к формированию и решению системы алгебро-дифференциальных уравнений, лежащих в основе алгоритма ВЭ, представленного на рисунке 2.4. В результате вычисления значения выражения, каждой операции которого поставлена в соответствие записанная в явном виде математико-алгоритмическая модель, будет найдено искомое значение переменной, которое присваивается ей с помощью оператора присваивания \doteq . Для этих целей к соответствующему ей узлу должен быть присоединен компонент-приемник, представленный на рисунке 2.11, в.

С его помощью это значение может быть передано на объектный уровень многоуровневой КЦ СТУС, представленной на рисунке 2.5, в качестве значений команды управления реальным объектом или значения параметра компонента ХТС. Оно также может быть передано на визуальный уровень для его последующей визуализации. Также компонент-приемник может передавать результат выполнения сценария в генератор отчетных форм для помещения его в формируемый отчет.

4.4 Отображение скалярных математических конструкций языка

МАК в формат алгоритмических КЦ

4.4.1 Типы данных. Источники констант и определенных переменных

Для построения и выполнения сценариев исследования и функционирования СТУС, включающих в себя сценарии функционирования устройств управления, требуется ввод и преобразование данных следующих типов:

1. *Данные целочисленного типа* используются для задания количественных параметров команд управления реальной ХТС, в качестве индексов при обра-

ботке массивов данных. Они также применяются в других целях, где не допускается использование вещественных или комплексных данных. Также допустимо задание целочисленных данных в двоичной или шестнадцатеричной форме для формирования команд платам сбора данных и управляющим контроллерам, взаимодействующим с реальным объектом. Соответствующие данным целочисленного типа источники представлены на рисунке 4.2

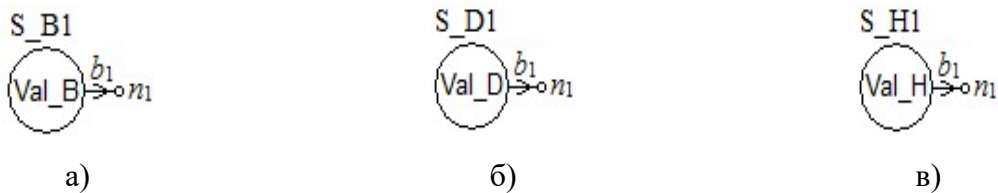


Рисунок 4.2 – Источники данных целочисленного типа,
 а) источник двоичной константы
 б) источник десятичной константы
 в) источник шестнадцатеричной константы

Им соответствует математическая модель вида

$$V_{n1} = I, \quad (4.35)$$

где I – заданная константа целочисленного типа (4.11). Ветвь b_1 инцидентна узлу n_1 и направлена от компонента к нему. Они в совокупности образуют связь

$$S_1 = (b_1 \eta^{+n_1}) \rightarrow \{V_{n1}\}. \quad (4.36)$$

Данные целочисленного типа также могут вступать в математические операции с данными других численных типов. При этом производится их преобразование по определенным правилам.

2. *Данными вещественного типа* являются подлежащие обработке и визуализации результаты анализа КЦ ХТС и данные измерения характеристик реальной системы. Для задания вещественных констант используется источник константы вещественного типа, представленный на рисунке 4.3, с математической моделью

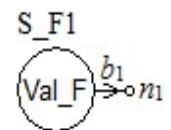


Рисунок 4.3 –
 Источник константы
 вещественного типа

$$V_{n1} = F \quad (4.37)$$

где F – константа вещественного типа (4.14).

3. *Данные комплексного типа* используются при частотных видах анализа КМ ХТС, а также могут применяться для подготовки графиков функций к визуализации. При этом в качестве вещественной части каждой точки указывается значение ее абсциссы, а в качестве мнимой части – значение ординаты точки. Источник комплексной константы, представленный на рисунке 4.4, имеет математическую модель вида

$$V_{n1} = C, \quad (4.38)$$

где C – комплексная константа.

4. *Данные логического типа* являются результатами операции сравнения данных числовых типов и позволяют организовать ветвления и циклы в сценариях исследования и функционального проектирования СТУС. переменные этого типа данных могут принимать только два значения: 0 – *false* (ложь) и 1 – *true* (истина). Источник логического значения, представленный на рисунке 4.5, описывается математической моделью вида

$$V_{n1} = B, \quad (4.39)$$

где B – константа логического типа.

5. *Данные строкового типа* применяются в сценариях исследования и функционального проектирования СТУС для формирования запросов к внешним хранилищам данных, строк и текстов, выводимых на визуальном уровне, и текстовых материалов для их помещения в формируемые с помощью генератора отчетных форм документы. Математическая модель источника строкового типа, представленного на рисунке 4.6, имеет вид

$$V_{n1} = S, \quad (4.40)$$

где S – константа строкового типа, состоящая из заключенных в кавычки разнообразных видимых символов, предусмотренных таблицей печатных символом ASCII [161].

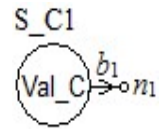


Рисунок 4.4 – Источник константы комплексного типа

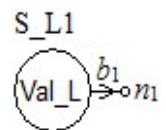


Рисунок 4.5 – Источник константы логического типа

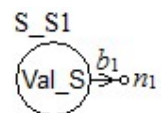


Рисунок 4.6 – Источник постоянной строки

Преобразование данных представленных типов осуществляется с помощью соответствующих операторов (4.5) и функций (4.20) языка МАК, которые отображаются в формат алгоритмических КЦ в виде соответствующих компонентов.

4.4.2 Отображение операторов языка МАК

Множество операторов (4.5), к которым относятся арифметические операторы $\langle AO \rangle$ (4.6), операторы сравнения $\langle OC \rangle$ (4.7) и логические операторы $\langle LO \rangle$ (4.8) отображаются в формате алгоритмических КЦ во множество соответствующих компонентов.

Отображение арифметических операторов. Каждому из арифметических операторов может быть поставлена в соответствие математическая модель вида

$$y = x_1 \langle A \rangle x_2 \quad (4.41)$$

где x_1, x_2 – операнды, вступающие в математическую операцию, предписанную оператором $\langle A \rangle$; $\langle A \rangle$ – один из арифметических операторов множества $\langle AO \rangle$ (4.6).

Операнды, вступающие в операцию, отображаемую оператором $\langle A \rangle$ (4.41), могут иметь один из численных типов данных: целочисленного, вещественного или комплексного. Если в нее вступают два операнда, имеющие разные типы данных, то производится преобразование данных по следующим правилам:

Правило 4.1. Если в операцию, соответствующую оператору $\langle A \rangle$, вступают данные целочисленного и вещественного типа, то производится операция над ними, результат которой будет принадлежать вещественному типу данных.

Правило 4.2. Если один из операндов x_1 или x_2 , вступающих в операцию $\langle A \rangle$ (4.41), принадлежит комплексному типу данных, то второй операнд преобразуется к комплексному типу, для которого

$$\begin{aligned} \operatorname{Re}(x_i) &= x_i \\ \operatorname{Im}(x_i) &= 0 \end{aligned} \quad (4.42)$$

где $i = 1 \vee 2$. Результат операции $\langle A \rangle$ (4.41) в этом случае будет принадлежать комплексному типу данных.

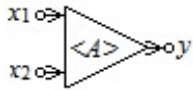


Рисунок 4.7 –
Обобщенный
компонент
арифметической
операции

Математической модели вида (4.41) ставится в соответствие компонент, обобщенный вид которого приведен на рисунке 4.7, отображающий соответствующую арифметическую операцию в формате алгоритмических КЦ.

Множество арифметических операторов (4.6) отображается во множество компонентов арифметических операций: сложением, вычитанием, умножением, делением. Графическое обозначение арифметически операций, их имитационные модели и описание принципов работы приведены в таблице П7.1. приложения 7.

Отображение операторов сравнения. Каждый оператор сравнения $\langle C \rangle$ множества $\langle OC \rangle$ (4.7) будет описываться математико-алгоритмической моделью вида

$$y = \begin{cases} true & \text{если } x_1 \langle C \rangle x_2 \\ false & \text{если } \neg(x_1 \langle C \rangle x_2) \end{cases}, \quad (4.43)$$

где x_1, x_2 – операнды целочисленного или вещественного типа данных, вступающие в операцию сравнения, предписанную оператором $\langle C \rangle$; y – переменная логического типа данных, принимающая значение результата операции сравнения.

Каждой операции сравнения, отображаемой в языке МАК оператором $\langle C \rangle$ (4.43) из множества операторов сравнения $\langle OC \rangle$ (4.7), ставится в соответствие

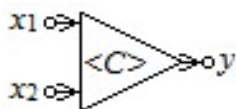


Рисунок 4.8 – Обобщенный
компонент операции
сравнения

компонент, обобщенный вид которого представлен на рисунке 4.8. Множество операторов сравнения $\langle OC \rangle$ (4.7) отображается во множество компонентов операций сравнения. К ним относятся компоненты, отображающие на языке МАК операции «Больше», «Меньше», «Больше или равно», «Меньше или равно», «Равно» и «Не равно».

Графическое обозначение алгоритмических компонентов операторов сравнения, их имитационные модели и описание принципов работы приведены в таблице П7.2 приложения 7.

Отображение логических операторов. Каждому логическому оператору $\langle L \rangle$ из множества $\langle LO \rangle$ (4.8), кроме оператора отрицания \neg , ставится в соответствие математическая модель вида

$$y = x_1 \langle L \rangle x_2, \quad (4.44)$$

где x_1 и x_2 – операнды логического типа данных; y – результат выполнения операции, предписанной оператором $\langle L \rangle$, который также приобретает значение логического типа данных.

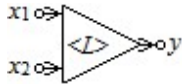


Рисунок 4.9 –
Обобщенный
компонент
логической
операции

Каждая логическая операция, представленная в языке МАК оператором $\langle L \rangle$ из множества логических операторов $\langle LO \rangle$ (4.8), отображается в формате алгоритмических КЦ компонентом, вид которого представлен на рисунке 4.9.

Операции отрицания

$$y = \neg x_1 \quad (4.45)$$

ставится в соответствие оператор «Отрицание», представленный на рисунке 4.10.

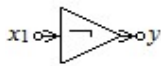


Рисунок 4.10 –
Компонент операции
«Отрицание»

Множество логических операторов $\langle LO \rangle$ (4.8), в составе оператора отрицания – логического «Не», конъюнкции «И» и дизъюнкции «ИЛИ», отображается во множество компонентов логических операций. Графическое обозначение алгоритмических компонентов логических операций, их имитационные

модели и описание принципов работы приведены в таблице П7.3 приложения 7.

4.4.3 Отображение функций языка МАК

К множеству функций языка МАК относятся:

- *математические функции*, аргументами которых являются скалярные переменные числовых типов данных (целочисленного, вещественного и комплексного);

- *функционалы и блоки обработки* [69], позволяющие оценивать функции на заданном интервале выбранного аргумента (времени, частоты и др.) некоторым интегральным значением, называемым *параметром-функционалом*;

– функции пользователя [156], имеющие имя, которому ставится в соответствие некоторое математическое выражение относительно аргументов данной функции.

Отображение стандартных математических функций. К стандартным математическим функциям относятся степенные, логарифмические, тригонометрические, обратные тригонометрические, прямые и обратные гиперболические тригонометрические функции. Они подразделяются на логарифмоподобные и функции общего вида. Каждая ЛПФ в языке МАК представляется в виде (4.20). Для нее задается определенное уникальное имя $\langle \text{Имя_функции} \rangle$ (4.21), которое должно быть зафиксировано в статической таблице идентификаторов. К стандартным функциям общего вида, представленным в таблице 4.1, относятся – корень, возведение в степень, логарифм по основанию, факториал, взятие в скобки, взятие модуля, деление (дробь). Остальные функции общего вида будут относиться к функционалам или матрицам, которые рассмотрены в следующих разделах.

Каждой стандартной математической функции ставится в соответствие компонент, отображающий ее в формате алгоритмических КЦ (2.63), с математической моделью вида

$$y = f(\mathbf{X}), \quad (4.46)$$

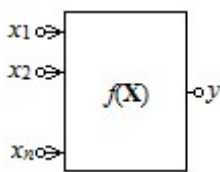


Рисунок 4.11 – Обобщенный компонент стандартной математической функции

где $\mathbf{X} = [x_1 \ x_2 \ \dots \ x_n]^T$ – массив аргументов функции; f – реализованная математическая или математико-алгоритмическая функция; y – результат выполнения функции.

Графическое обозначение алгоритмических компонентов стандартных математических функций, их имитационные модели и описание принципов работы приведены в таблице П7.4 приложения 7.

Отображение блоков обработки данных. Параметрами-функционалами [62] называются численные интегральные значения, характеризующие некоторую функцию на заданном интервале ее аргумента. Их определение осуществляется по

определенным алгоритмам, называемым *блоками обработки данных* (БОД) [69]. Каждый БОД в формате языка МАК представляется в виде

$$S = F(x, P_1, P_2, f(x, V)), \quad (4.47)$$

где x – варьируемый аргумент функции $f(x, V)$, называемый далее *параметром функционала*, при изменении которого определяется соответствующий параметр-функционал (например, время, частота, сопротивление и т.п.); P_1 – нижний предел изменения параметра функционала x ; P_2 – верхний предел изменения параметра функционала x ; $f(x, V)$ – функция, параметр-функционал которой подлежит определению; V – вектор аргументов функции $f(x, V)$, значения которых остаются постоянными при определении параметра-функционала; S – подлежащее определению значение параметра-функционала.

При задании функционала пределы изменения его параметров, а также исследуемая функция $f(x, V)$ могут представляться некоторыми математико-алгоритмическими выражениями, для которых могут быть построены отдельные алгоритмические КЦ.



Рисунок 4.12 – Структура компонента «Функционал»

Кроме этого функция $f(x, V)$ может представлять собой временную, частотную или параметрическую зависимость некоторой выходной величины модели исследуемой ХТС или данные измерения определенной характеристики реальной системы. Обобщенная структура компонента функционала представлена на рисунке 4.12. Она лежит в основе разработки компо-

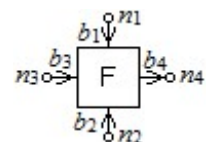


Рисунок 4.13 – Обобщенный компонент «Функционал»

нента «Функционал». Его обобщенный вид приведен на рисунке 4.13. Обобщенный компонент «Функционал» имеет следующие входные и выходные связи:

– $S_1 = (b_1 \eta^- n_1) \rightarrow \{P_1\}$ – связь, с помощью которой в компонент поступает значение нижнего предела параметра функционала, который рассчитывается математико-алгоритмическим выражением $Expr_1$;

– $S_2 = (b_2 \eta^- n_2) \rightarrow \{P_2\}$ – связь для приема значения верхнего предела параметра функционала, для расчета которого применяется математико-алгоритмическое выражение $Expr_2$;

– $S_3 = (b_3 \eta^- n_3) \rightarrow \{f(x, \mathbf{V})\}$ – связь для приема значений функции, заданной выражением $Expr_3$ и составленной относительно результатов анализа КМ ХТС объектного уровня многоуровневой КМ СТЭС и (или) данных измерения характеристик реальной системы;

– $S_4 = (b_4 \eta^+ n_4) \rightarrow \{S\}$ – связь для передачи вычисленного значения функционала другим компонентам, подключенным к узлу n_4 .

К параметрам-функционалам, реализованным на основе обобщенного компонента «Функционал», представленного на рисунке 4.13, относятся такие функции общего вида: полная производная, определенный интеграл, сумма, произведение, а также другие подобные функции, называемые *общематематическими блоками обработки данных*.

Кроме того, к компонентам-функционалам относятся блоки обработки данных, которыми являются временные характеристики наблюдаемых переменных реальной ХТС \widetilde{y}_o (см. рисунок 1.3), а также временные, частотные или параметрические характеристики результатов анализа её КЦ y_o . Для определения параметров-функционалов временных, частотных и параметрических характеристик реализован набор компонентов, позволяющих анализировать:

- общие временные, частотные и параметрические характеристики;
- стационарные колебательные процессы;
- переходные аperiodические и колебательные процессы.



Рисунок 4.14 – Обобщенный компонент блоков обработки данных

ний временной, частотной или параметрической характеристики f , параметр-функционал которой подлежит определению;

– $S_2 = (b_2 \eta^+ n_2) \rightarrow \{S\}$ – связь, с помощью которой компонент передает найденное значение параметра функционала другим компонентам.

К общематематическим БОД относятся суммирование, произведение, максимальное и минимальное значения, значения абсциссы (времени, частоты и т.п.) при максимальном или минимальном значении. Основанными на разложении временного сигнала в ряд Фурье [162] блоками обработки данных установившихся периодических процессов во временной области являются амплитуда и фаза заданной гармоники, частота и период сигнала. Для анализа периодических процессов разработаны БОД для определения установившегося значения, времени переходного процесса, перерегулирования, периода свободных колебаний. Анализ частотных характеристик подразумевает автоматическое определение резонансной частоты, частоты среза, запасов устойчивости по амплитуде и фазе. Компоненты БОД, предназначенные для определения параметров-функционалов временных, частотных и параметрических характеристик реальной ХТС и её КЦ, приведены в таблице П7.5 приложения 7.

Отображение функций пользователя. Функцией пользователя в языке МАК называется любая функция, сформированная на основании правила (4.20), имя которой $\langle \text{Имя_функции} \rangle$ не совпадает с именами стандартных ЛПФ, приведенных в таблице П7.4 приложения 7. Определение функции пользователя в формате языка МАК производится по правилу

$$\langle \text{Имя_функции} \rangle (\langle \text{Список_аргументов} \rangle) := \langle \text{Expr} \rangle, \quad (4.48)$$

Обобщенный компонент определения БОД, представленный на рисунке 4.14 и называемый далее *компонентом БОД*, содержит следующие связи:

$$- S_1 = (b_1 \eta^- n_1) \rightarrow \{f\} - \text{связь для приема значе-}$$

где $\langle Expr \rangle$ – математико-алгоритмическое выражение, составленное из ранее известных лексем и аргументов функции, входящих в список $\langle Список_аргументов \rangle$.

При определении функции пользователя по правилу (4.48) для математико-алгоритмического выражения $\langle Expr \rangle$, записанного справа от знака присваивания $:=$, формируется собственная алгоритмическая КЦ (2.63). В случае ее использования для вычислений сформированная ранее алгоритмическая КЦ встраивается в КЦ выражения, в котором использована данная функция.

4.5 Отображение векторно-матричных конструкций языка МАК

Для целей передачи по одной связи, организованной между компонентами логического уровня многоуровневой КЦ СТУС, упорядоченного набора значений одного скалярного типа данных в язык МАК введены векторные и матричные типы данных. При этом в общем случае по векторно-матричной связи будет осуществляться передача матрицы вида

$$[M_{n,m}] = \begin{bmatrix} V_{11} & V_{12} & \dots & V_{1m} \\ V_{21} & V_{22} & \dots & V_{2m} \\ \dots & \dots & \dots & \dots \\ V_{n1} & V_{n2} & \dots & V_{nm} \end{bmatrix}, \quad (4.49)$$

где n – число строк матрицы; m – число ее столбцов.

Если $n = 1$, будем говорить о вектор-строке, если $m = 1$ – то о вектор-столбце.

Каждый элемент матрицы (вектора) представляет собой переменную или константу определенного типа данных. Для матрицы, все элементы которой, относящиеся к одному скалярному типу данных (смотри пункт 4.4), определены, может быть поставлен в соответствие компонент «Источник матрицы», приведенный на рисунке 4.15. Его параметрами являются количество строк n и столбцов m . Математическая модель компонента «Источник матрицы» включает в себя $n \cdot m$ уравнений вида

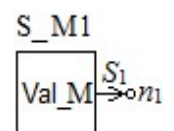


Рисунок 4.15 –
Источник
матрицы

$$V_{n_1}[i][j] = M[i][j], \tag{4.50}$$

где $i = \overline{1, n}, j = \overline{1, m}$, M – матрица размером $n \times m$, все элементы которой определены.

Для матрицы определен оператор «Взятие элемента», который в формате языка МАК имеет вид

$$M_{i,j} \tag{4.51}$$

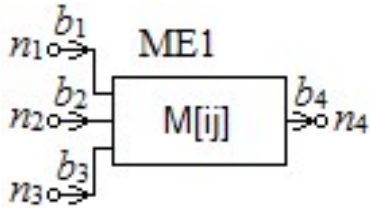


Рисунок 4.16 – Компонент «Элемент матрицы»

где i – номер строки, в которой находится необходимый элемент; j – номер столбца.

Оператору (4.51) ставится в соответствие компонент «Элемент матрицы», приведенный на рисунке 4.16, с математической моделью

$$V = M_{i,j}, \tag{4.52}$$

где M – матрица, поступающая в компонент по $S_1 = (b_1 \eta^- n_1) \rightarrow \{M\}$; i – номер строки матрицы, в которой находится элемент ($S_2 = (b_2 \eta^- n_2) \rightarrow \{i\}$); j – номер столбца, в котором находится элемент ($S_3 = (b_3 \eta^- n_3) \rightarrow \{j\}$); V – подлежащий взятию элемент матрицы для его дальнейшего использования отдельно от других элементов матрицы ($S_4 = (b_4 \eta^+ n_4) \rightarrow \{V\}$).

Для присваивания заданному элементу матрицы (вектора) некоторого значения, что применяется при построении сценариев, работающих с одномерными и многомерными массивами, используется основное выражение присваивания значения элементу матрицы

$$M_{i,j} := V, \tag{4.53}$$

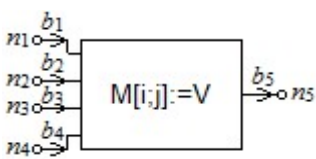


Рисунок 4.17 – Компонент «Присваивание значения элементу матрицы»

где M – численно определенная матрица, каждый элемент которой имеет определенное численное значение; i, j – индексы изменяемого элемента матрицы M ; V – присваиваемое значение, тип которого должен совпадать с типом элементов матрицы.

Отображением операции (4.53), представленной в формате языка МАК, в формат алгоритмических КЦ является компонент «Присваивание значения элементу матрицы», приведенный на рисунке 4.17. Его связями являются:

– $S_1 = (b_1 \eta^- n_1) \rightarrow \{M_1\}$ – связь, с помощью которой компонент получает матрицу M_1 для изменения значения его элемента;

– $S_2 = (b_2 \eta^- n_2) \rightarrow \{i_1\}$ – связь для получения индекса строки матрицы с подлежащим изменению элементов;

– $S_3 = (b_3 \eta^- n_3) \rightarrow \{j_1\}$ – связь для получения индекса столбца матрицы, в котором находился подлежащий изменению элемент;

– $S_4 = (b_4 \eta^- n_4) \rightarrow \{V\}$ – связь, через которую в компонент поступает значение, которое должно быть присвоено элементу матрицы;

– $S_5 = (b_5 \eta^+ n_5) \rightarrow \{M_2\}$ – связь для передачи матрицы с измененным значением элемента с индексами i и j .

Математическая модель элемента «Присваивание значения элементу матрицы» описывается выражением

$$M_2[i, j] = \begin{cases} V & \text{если } i = i_1 \wedge j = j_1 \\ M_1[i, j] & \text{если } i \neq i_1 \vee j \neq j_1 \end{cases}, \quad (4.54)$$

которое выполняется для $\forall i = \overline{1..n}, \forall j = \overline{1..m}$, где n, m – размеры матрицы M_1 .

После выполнения выражения (4.54) размер матрицы M_2 будет равен размеру матрицы M_1 .

Над матрицами, удовлетворяющими определенным условиям, могут быть выполнены операции сложения, вычитания, умножение матрицы на скаляр, умножение двух матриц, нахождение определителя и ранга матрицы, а также возведение матрицы в целочисленную степень. Компоненты, отображающие перечисленные операции языка МАК в формат алгоритмических КЦ, приведены в таблице П7.6 приложения 7.

4.6 Отображение математико-алгоритмических конструкций языка МАК в формат алгоритмических компонентных цепей

В пункте 4.1.3 были приведены правила формирования основных математико-алгоритмических конструкций языка МАК. Они могут быть отображены в соответствующие алгоритмические компонентные цепи (2.63) с использованием компонентов, приведенных в приложении 7.

Отображение математических выражений. Выражение присваивания (4.23) будет отображено в алгоритмическую КЦ, где к выходному узлу выражения $\langle Expr \rangle$ будет присоединен компонент, осуществляющий передачу рассчитанного значения на объектный и/или визуальный уровень многоуровневой КЦ СТУС, либо передачу этого значения в генератор отчетных форм [120].

В качестве примера на рисунке 4.18 приведен пример алгоритмической КЦ, соответствующей выражению присваивания (4.23). С его помощью производится вычисление гидросопротивления трубы по ее заданным геометрическим размерам – длине l и диаметру d , а также по введенному коэффициенту вязкости λ проходящей по трубе жидкогазовой смеси. Вычисление гидросопротивления осуществляется по формуле

$$R_G = \lambda \cdot \frac{L}{d}. \quad (4.55)$$

Алгоритмическая (математическая) компонентная цепь, предназначенная для

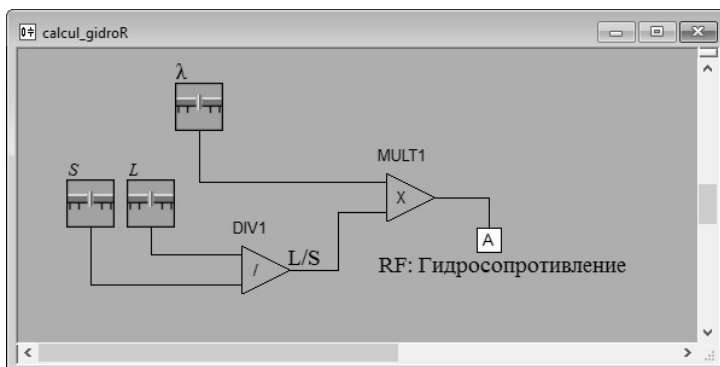


Рисунок 4.18 – Алгоритмическая (математическая) компонентная цепь для вычисления гидросопротивления трубы по формуле (4.55)

вычисления гидросопротивления по формуле (4.55), приведена на рисунке 4.18.

При изменении хотя бы одного из параметров λ, L или d с помощью сформированной алгоритмической КЦ, приведенной на рисунке 4.18, вычисляется соответствующее значение гидросо-

противления трубы, которое с помощью компонента-атрибута устанавливается в качестве значения параметра компонента RF.

Если значение некоторого параметра или переменной должно быть вычислено по некоторому условию, применяется выражение условного присваивания (4.24). В этом случае используются операторы сравнения $\langle OC \rangle$ (4.7) и логические операторы $\langle LO \rangle$ (4.8). Их результатами являются значения логического типа данных. Для построения алгоритмической КЦ для выражения условного присваивания (4.24) применяется компонент «Условное присваивание», представленный на рисунке 4.19. На

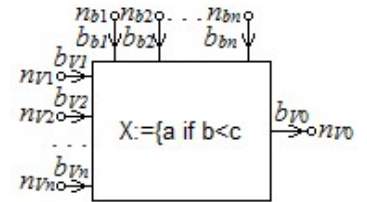


Рисунок 4.19 – Компонент «Условное присваивание»

его входные связи $S_{vi} = (b_{vi} \eta^- n_{vi}) \rightarrow \{V_i\}$ (где $i = \overline{1, n}$) поступают значения вещественного типа, полученные в результате вычисления выражений $\langle Expr1 \rangle, \langle Expr2 \rangle, \dots, \langle ExprN \rangle$. На входные связи логического типа $S_{bi} = (b_{bi} \eta n_{bi}) \rightarrow \{V_{Li}\}$ подаются результаты выполнения логических выражений $\langle L_Expr1 \rangle, \langle L_Expr2 \rangle, \dots, \langle L_ExprN \rangle$ выражения (4.24). Если первое попавшееся J -е логическое выражение имеет значение *true*, то значение соответствующего математико-алгоритмического выражения $\langle ExprJ \rangle$ присваивается выходной переменной V_0 (где $S_{v0} = (b_{v0} \eta^+ n_{v0}) \rightarrow \{V_0\}$).

С помощью компонента «Условное присваивание» может быть рассчитан коэффициент лобового сопротивления [38], который зависит от значения числа Рейнольдса следующим образом:

$$\xi = \begin{cases} \frac{24}{Re} & \text{если } Re < 2 \\ \frac{18.5}{Re^{0.6}} & \text{если } 2 < Re < 500 \\ 0.5 & \text{если } Re > 500 \end{cases} \quad (4.56)$$

Алгоритмическая КЦ, соответствующая выражению (4.56), представлена на рисунке 4.20.

Если подлежащая определению переменная не может быть явно выражена из выражения, что бывает, например, в квадратном уравнении, в языке МАК используется выражение «уравнение» (4.25), в левую $\langle Expr_1 \rangle$ и (или) в правую часть $\langle Expr_2 \rangle$ которого входит подлежащая определению переменная. Для решения такого уравнения должны быть сформированы компонентные цепи для обеих его частей, которые соединяются

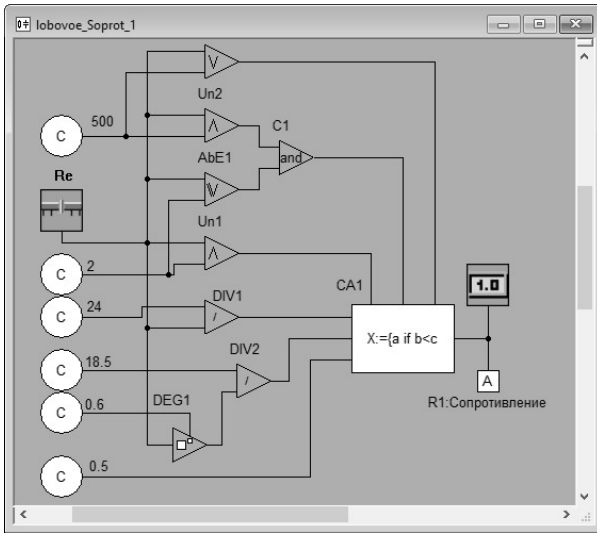


Рисунок 4.20 – Алгоритмическая КЦ для расчета коэффициента лобового сопротивления по формуле (4.56)

компонентом «Равенство», приведенном на рисунке 4.21. Связи данного компонента соединяют две части алгоритмической КЦ, сформированной на основе уравнения:

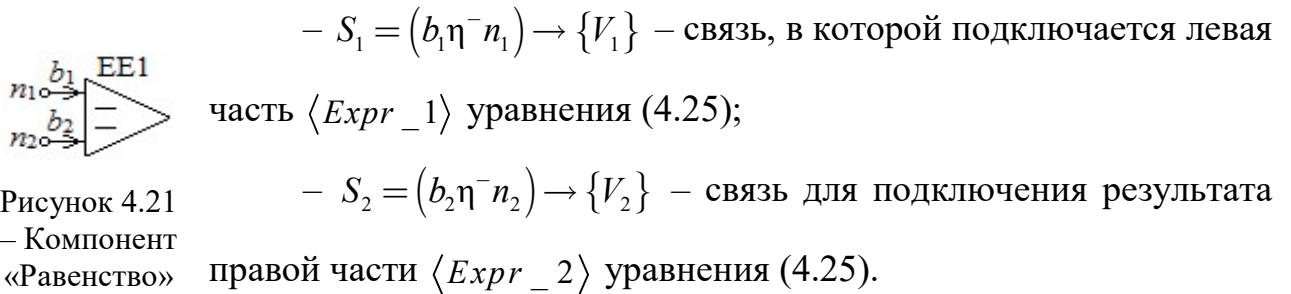


Рисунок 4.21 – Компонент «Равенство»

Математическая модель компонента «Равенство», приведенного на рисунке 4.21, имеет вид

$$V_1 - V_2 = 0. \tag{4.57}$$

В качестве примера рассмотрим алгоритмическую КЦ характеристического уравнения замкнутой системы автоматического управления второго порядка

$$T_1^2 p^2 + T_2 p + 1 + K = 0, \tag{4.58}$$

имеющего два корня, вид которых (вещественный или комплексный) обуславливает тип переходных процессов в системах автоматического управления второго порядка [61, 85]. Алгоритмическая КЦ уравнения (4.58), составленная из матема-

тических компонентов, приведенных в приложении 7, и компонента «Равенство», изображенного на рисунке 4.21, приведена на рисунке 4.22

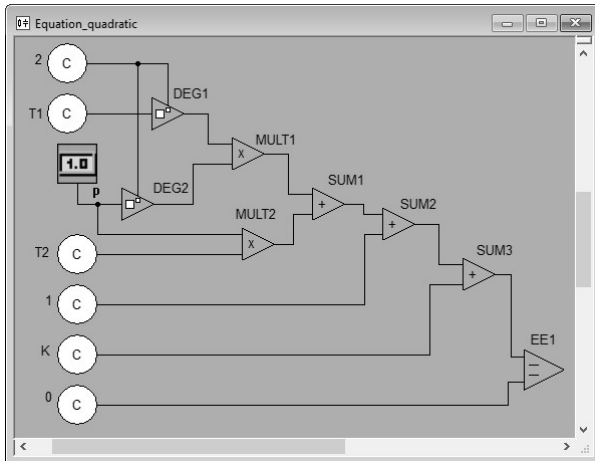


Рисунок 4.22 – Алгоритмическая компонентная цепь уравнения (4.58)

Для решения уравнения (4.58) компоненты, входящие в ее алгоритмическую КЦ, представленную на рисунке 4.22, должны описываться моделями, представленными в неявной форме относительно переменных своих связей. В этом случае решение уравнения может быть получено соответствующими режимами универсального вычислительного ядра [103].

Аналогичным образом может быть составлена алгоритмическая КЦ из математических операторов и функций, предназначенная для решения систем линейных, нелинейных и дифференциальных уравнений [30, 163].

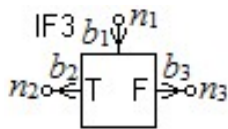


Рисунок 4.23 – Компонент «Ветвление»

Отображение алгоритмических конструкций. В отличие от математических выражений (4.23), вычисления которых

осуществляются по линейному алгоритму в соответствии с предписанными приоритетами операций, алгоритмические конструкции предполагают ветвления по заданным условиям с образованием алгоритмических обратных связей, называемых *циклами*. Для организации ветвления и циклов языка МАК (4.26)-(4.28) в формат алгоритмических КЦ введен компонент «Ветвление». Представленный на рисунке 4.23, он имеет следующие связи:

– $S_1 = (b_1 \eta^- n_1) \rightarrow \{V_1\}$ – связь для получения результата вычисления выражения логического типа данных;

– $S_2 = (b_2 \eta^+ n_2) \rightarrow \{V_2\}$ – связь для передачи управления в случае, когда переменная $V_1 \neq 0$, то есть когда на узел n_1 поступает значение «Истина»;

– $S_3 = (b_3 \eta^+ n_3) \rightarrow \{V_3\}$ – связь для передачи управления в случае, когда переменная $V_2 \neq 0$, то есть когда на узел n_2 поступает значение «Истина»;

– $S_3 = (b_3\eta^+n_3) \rightarrow \{V_3\}$ – связь для передачи управления, если $V_1 = 0$, то есть когда на узел n_1 передано значение «Ложь».

Математическая модель компонента «Ветвление» включает в себя два уравнения

$$\begin{aligned} V_2 &= \begin{cases} 1 & \text{если } V_1 \neq 0 \\ 0 & \text{если } V_1 = 0 \end{cases} \\ V_3 &= \begin{cases} 0 & \text{если } V_1 \neq 0 \\ 1 & \text{если } V_1 = 0 \end{cases} \end{aligned} \quad (4.59)$$

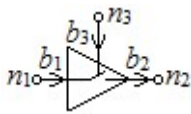


Рисунок 4.24 –
Компонент
«Накопитель»

Для передачи значений необходимых переменных в компоненты, выполняющие вычисления по результатам проверки логического значения в компоненте «Ветвление», разработан компонент «Накопитель», представленный на рисунке 4.24. Его задачей является пропускать значение, пришедшее на вход последним, если на управляющий вход подан сигнал со значением «Истина». Для осуществления этих возможностей компонент «Ветвление» имеет следующие связи:

– $S_1 = (b_1\eta^-n_1) \rightarrow \{V_1\}$ – предназначена для получения некоторого значения любого типа данных, которое должно быть передано другим компонентам;

– $S_2 = (b_2\eta^+n_2) \rightarrow \{V_2\}$ – связь, через которую осуществляется передача значения, полученного по связи S_1 при условии, что по управляющей связи было принято истинное значение;

– $S_3 = (b_3\eta^-n_3) \rightarrow \{V_3\}$ – управляющая связь, с помощью которой компонент «Накопитель» получает управляющий сигнал.

Математико-алгоритмической моделью компонента «Накопитель» является уравнение

$$V_2 = \begin{cases} V_1 & \text{если } V_3 \neq 0 \\ 0 & \text{если } V_3 = 0 \end{cases} \quad (4.60)$$

Совокупность компонентов «Ветвление», представленного на рисунке 4.23, и «Накопитель», приведенного на рисунке 4.24, реализует в формате алгоритми-

ческих КЦ операцию ветвления, которая в отличие от компонента «Условное присваивание» позволяет на основе единого условия присваивать соответствующие значения различным переменным.

В качестве примера рассмотрим модель управления клапанами C_1 и C_2 в зависимости от количества газа G в баллоне [164]. Если его текущее количество равно или меньше минимального $G \leq G_{\min}$, то клапан C_1 , по которому газ из баллона подается потребителю, должен закрыться, то есть $C_1 = 0$, а если равен или больше максимального $G \geq G_{\max}$, то открыться – $C_1 = 1$.

Клапан C_2 предназначен для подачи газа в баллон. Он должен открываться, когда количество газа в баллоне стало равно или меньше минимального значения $G \leq G_{\min}$, и закрываться, когда количество газа в баллоне достигло максимального значения $G \geq G_{\max}$.

Сценарий функционирования УУ в формате языка МАК представлен в виде двух выражений условного присваивания (4.24)

$$C_1 = \begin{cases} 1 & \text{если } G \geq G_{\max} \\ 0 & \text{если } G \leq G_{\min} \end{cases} \quad (4.61)$$

$$C_2 = \begin{cases} 1 & \text{если } G \leq G_{\min} \\ 0 & \text{если } G \geq G_{\max} \end{cases}$$

Алгоритмическая КЦ, соответствующая выражению (4.61), приведена на

рисунке 4.25.

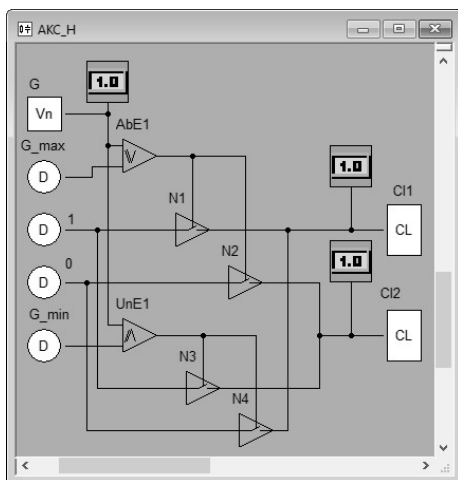


Рисунок 4.25 – Алгоритмическая компонентная цепь выражения (4.61)

С помощью компонента «Ветвление», представленном на рисунке 4.23, и приведенном на рисунке 4.24 компонента «Накопитель» могут быть сформированы алгоритмические КЦ алгоритмических структур (4.26)-(4.28).

Цикл с параметром (4.26) может быть выполнен только в случае, когда количество его итераций заранее известно. В литературе по теории алгоритмов и программ [165] его

рекомендуют использовать в основном для работы с массивами информации, к классу которых относятся вектора и матрицы (4.49).

На рисунке 4.26 приведена алгоритмическая КЦ алгоритма поиска суммы элементов матрицы M размерности $S_x \times S_y$.

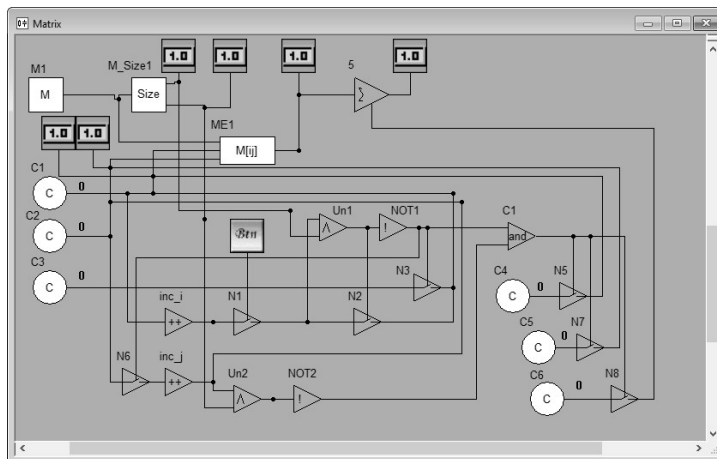


Рисунок 4.26 – Алгоритмическая КЦ вычисления суммы элементов матрицы

Аналогичным образом могут быть реализованы алгоритмические КЦ для сценариев, включающих различные виды циклов (4.26)-(4.28).

4.7 Компоненты для представления алгоритмов решения задач многовариантного анализа и параметрической оптимизации

В процессе исследования и функционального проектирования СТЭС решению могут подлежать задачи параметрической оптимизации, направленные на определение значений аргументов целевых функций (1.2)-(1.7), при которых они имеют минимум [166]. К данному классу также относятся задачи идентификации модели, при которых подлежащие минимизации целевые функции (1.14) составляются относительно данных измерения характеристик реальной химико-технологической системы и результатов анализа соответствующей ей КЦ при варьировании параметров. Их решение основано на многократном анализе и параметрическом синтезе модели ХТС в статическом или динамическом режиме, результаты которого, включая параметры-функционалы полученных зависимостей, используются при формировании целевых функций.

Для выполнения различных видов многовариантного анализа [65] в рамках алгоритмических КЦ применяются следующие компоненты:

1. Для построения семейства выходных характеристик или зависимости некоторого параметра-функционала от значений варьируемого параметра модели используется компонент «Построитель семейства характеристик», приведенный на рисунке 4.27. Его связи являются:

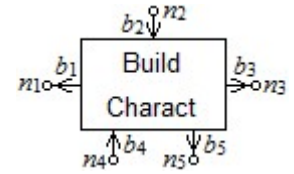


Рисунок 4.27 – Компонент «Построитель семейств характеристик»

– $S_1 = (b_1 \eta^- n_1) \rightarrow \{P\}$ – связь для передачи текущего значения варьируемого параметра;

– $S_2 = (b_2 \eta^- n_2) \rightarrow \{C_1\}$ – для получения от измерительных компонентов или блоков обработки результатов текущего значения или самой характеристики, полученной в результате анализа сформированной на объектном уровне многоуровневой КЦ СТУС при текущем значении варьируемого параметра;

– $S_3 = (b_3 \eta^+ n_3)$ – для передачи построенного семейства выходных характеристик или зависимости параметра-функционала от варьируемого параметра КЦ ХТС;

– $S_4 = (b_4 \eta^+ n_4) \rightarrow \{Start\}$ – связь для передачи сигнала начала или окончания очередного одновариантного анализа, проводимого универсальным вычислительным ядром;

– $S_5 = (b_5 \eta^- n_5) \rightarrow \{Begin\}$ – связь, с помощью которой компонент получает сигнал начала выполнения многовариантного анализа.

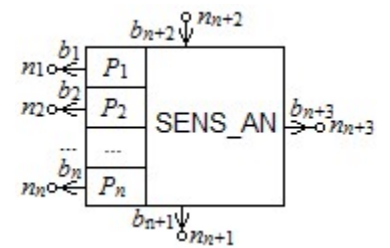


Рисунок 4.28 – Компонент «Анализ чувствительности»

2. Для проведения анализа чувствительности, который вычисляет коэффициенты чувствительности некоторой выходной переменной или ее параметра-функционала от изменения значения каждого из варьируемых параметров исследуемой ХТС по формуле

$$K_{P_i} = \frac{\partial F}{\partial P_i}, \quad (4.62)$$

где ∂F – изменение исследуемой переменной или ее параметра-функционала $\partial F = F(P_i + \partial P_i) - F(P_i)$; ∂P_i – изменение параметра;

используется компонент «Анализ чувствительности», представленный на рисунке 4.28. Он содержит следующие связи:

$$- S_1 = (b_1 \eta^+ n_1) \rightarrow \{P_1\}, S_2 = (b_2 \eta^+ n_2) \rightarrow \{P_2\}, \dots S_n = (b_n \eta^+ n_n) \rightarrow \{P_n\} - \text{связи}$$

для передачи значений параметров, варьируемых при проведении анализа чувствительности;

$$- S_{n+1} = (b_{n+1} \eta^+ n_{n+1}) \rightarrow \{Start\} - \text{связь, передающая сигнал } Start \text{ о начале}$$

очередного одновариантного анализа при установленных значениях параметров анализируемой модели;

$$- S_{n+2} = (b_{n+2} \eta^- n_{n+2}) \rightarrow \{F\} - \text{связь для приема значения анализируемой}$$

функции F , полученного в процессе одновариантного анализа, т.е. значения некоторой переменной связи в статическом или ее параметра-функционала при анализе в динамическом режиме;

$$- S_{n+3} = (b_{n+3} \eta^+ n_{n+3}) \rightarrow \{G\} - \text{связь для передачи другим компонентам сце-}$$

нария решения задачи исследования и функционального проектирования СТус результата анализа чувствительности, которым является вектор-градиент исследуемой переменной или параметра-функционала F исследуемой ХТС.

3. Проведение статистического анализа, результатом которого является распределение некоторой наблюдаемой переменной или ее параметра-функционала при заданных распределениях варьируемых параметров исследуемой модели ХТС, осуществляется компонентом «Статистический анализ» [167], пред-

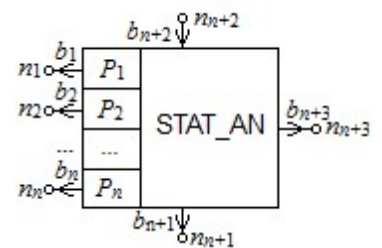


Рисунок 4.29 – Компонент «Статистический анализ»

ставленным на рисунке 4.29. Назначение его связей аналогично назначению связей компонента «Анализ чувствительности», который был рассмотрен на рисунке 4.28.

В процессе исследования, проектирования и функционирования СТУС автоматизированному решению подлежат задачи параметрического синтеза (1.6), в основе которых лежит минимизация целевых функций, сформированных относительно измеряемых переменных модели и наблюдаемых характеристик реальной ХТС [166]. Их решение основано на многократном анализе КЦ ХТС в статическом или динамическом режиме, результаты которого, включая параметры-функционалы полученных зависимостей, используются при формировании целевых функций. Для этого в формате языка МАК реализованы следующие программно-алгоритмические возможности построения сценариев для решения задач параметрической оптимизации:

- 1) использование стандартных компонентов-критериев [18], к которым от-

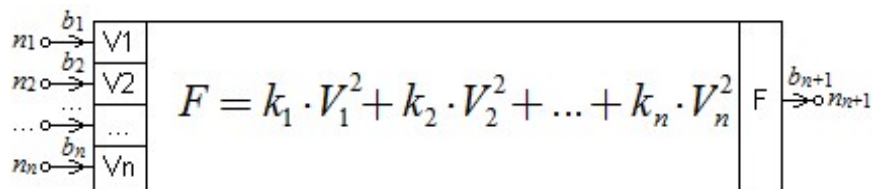


Рисунок 4.30 – Компонент «Целевая функция»

носятся средневзвешенный аддитивный, среднеквадратический, мультипликативный критерии, а также критерии формы;

- 2) использование компонента «Целевая функция», представленного на рисунке 4.30. С его помощью пользователем вводится целевая функция любого доступного математического вида относительно переменных его входных связей

$$S_i = (b_i \bar{n}_i) \rightarrow \{V_i\} \quad (4.63)$$

где $i = \overline{1, n}$.

Введенная в данный компонент, основанный на интерактивной математической панели [168], целевая функция автоматически преобразуется в компонентную подцепь на основе системы отображения математических выражений [102]. Полученная КЦ подлежит расчету с помощью универсального вычислительного ядра [103].

Полученные значения целевой функции передаются другим компонентам, представленным в приложении 7 и реализующим сценарии методов оптимизации, с помощью выходной связи

$$S_{n+1} = (b_{n+1}\eta^+ n_{n+1}) \rightarrow \{F\}, \tag{4.64}$$

где F – вычисленное значение целевой функции.

Любой метод оптимизации, градиентный или безградиентный, может быть реализован с помощью специального компонента [169], на входную связь которого

$$S_{n+1} = (b_{n+1}\eta^- n_{n+1}) \rightarrow \{F\} \tag{4.65}$$

поступает значение целевой функции, вычисленное с помощью компонента «Целевая функция» или других компонентов, к которым могут относиться компоненты-блоки обработки результатов, предназначенные для вычисления параметров-функционалов временных, частотных или параметрических характеристик.

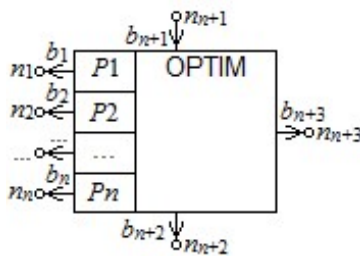


Рисунок 4.31 –
Компонент
«Метод оптимизации
нулевого порядка»

Компонент, реализующий метод оптимизации нулевого порядка, приведен на рисунке 4.31. Его связями являются:

$$- S_1 = (b_1\eta^+ n_1) \rightarrow \{P_1\}, S_2 = (b_2\eta^+ n_2) \rightarrow \{P_2\}, \dots, S_n = (b_n\eta^+ n_n) \rightarrow \{P_n\} - \text{связи}$$

для установки текущих значений варьируемых параметров P_1, P_2, \dots, P_n компонентов компьютерной модели ХТС, сформированной на объектном уровне многоуровневой КЦ СТУС;

$$- S_{n+1} = (b_{n+1}\eta^- n_{n+1}) \rightarrow \{F\} - \text{связь для получения текущего значения целевой функции } F, \text{ полученное путем численной обработки результатов анализа КЦ ХТС с помощью компонентов-блоков обработки результатов анализа и компонентов математических и алгоритмических функций};$$

– $S_{n+2} = (b_{n+2}\eta^+ n_{n+2}) \rightarrow \{Start\}$ – сигнал о начале очередного одновариантного анализа с установленными значениями варьируемых параметров;

$-S_{n+3} = (b_{n+3}\eta^+ n_{n+1}) \rightarrow \{Stop\}$ – сигнал об окончании процесса оптимизации.

В случае, когда метод оптимизации получил минимальное значение целевой функции с заданной точностью, передается сигнал «Истина» (*true*). Если реализованный метод оптимизации не достиг минимального значения целевой функции, а окончание процесса оптимизации произошло по другой причине – передается сигнал «ложь» (*false*).

В случае реализации градиентного метода оптимизации в компонент добавляется связь $S_4 = (b_4\eta^- n_4) \rightarrow \{G\}$, с помощью которой он получает значение градиента целевой функции, полученное с помощью компонента «Анализ чувствительности», который представлен на рисунке 4.28.

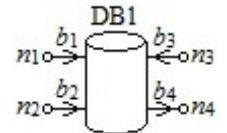
При реализации описанного набора компонентов [170], представленных на рисунках 4.27–4.31, открываются возможности формирования сценариев для автоматизированного решения задач исследования и функционального проектирования СТЭС, включая сценарии функционирования входящих в них устройств управления. При этом методы оптимизации могут быть использованы для определения оптимальных значений параметров компонентов ХТС или управляющих воздействий, при которых в них наблюдаются эффективные режимы функционирования. Практическое решение указанных задач на основе методов многовариантного анализа и параметрической оптимизации приведено в главе 7.

4.8 Автоматическая параметризация компьютерной модели СТЭС с помощью баз данных

С целью первичной параметризации компонентов компонентной цепи ХТС, расположенной на объектном уровне многоуровневой КЦ СТЭС, приведенной на рисунке 2.5, а также для своевременной выработки управляющих воздействий, адекватной текущей ситуации, должны использоваться актуальные данные, хранящиеся в базах данных технологических параметров [171], а также с ГИС-системами [172] в случае исследования компьютерных моделей эколого-экономических систем [173].

Считая, что многоуровневая КЦ СТУС будет использовать базы данных только в качестве источников информации различных типов данных, на ее логическом уровне должны быть реализованы компоненты, выполняющие следующие основные операции [174]:

- осуществление подключения к БД;
- поиск информации в БД на основе запросов, сформированных в формате языка SQL [158, 175];
- закрытие подключения к БД.



Для реализации функций подключения к базам данных, представленных в виде файлов на компьютере или развернутых на удаленных серверах, применяется компонент «База данных» [176], представленный на рисунке 4.32. В нем реализуются методы и алгоритмы доступа к физической БД, реализованной средствами различных систем управления базами данных [177] (Access, Oracle, MySQL и др.). Путь к базе данных указывается в качестве параметра данного компонента. Связи компонента «База данных» предназначены:

Рисунок 4.32 –
Компонент
«База данных»

- $S_1 = (b_1 \eta \bar{n}_1) \rightarrow \{Start\}$ – для осуществления подключения к БД, которое устанавливается, когда переменная $Start \neq 0$, т.е. когда компонент получит истинное (ненулевое) значение;
- $S_2 = (b_2 \eta \bar{n}_2) \rightarrow \{Stop\}$ – для прекращения подключения к БД в случае, когда переменная $Stop \neq 0$;
- $S_3 = (b_3 \eta \bar{n}_3) \rightarrow \{Query\}$ – для получения запроса $Query$, представляющего собой строку, сформированную на языке запросов SQL и предназначенную для реализации запроса к БД для получения значений, направленных для их присвоения соответствующим параметрам цепи ХТС, принадлежащей объектно-му уровню многоуровневой КЦ СТУС;

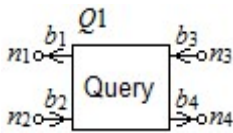


Рисунок 4.33 –
Компонент
«Запрос»

– $S_4 = (b_4\eta^+n_4) \rightarrow \{Answer\}$ – для передачи ответа

компоненту, отправившему запрос к БД.

Для формирования SQL-запросов к БД, представляющих собой строку определенного формата на языке SQL, используется представленный на рисунке 4.33 компонент «Запрос». Для

формирования запроса, его отправки в БД и получения ответа данных компонент содержит следующие связи:

– $S_1 = (b_1\eta^+n_1) \rightarrow \{Query\}$ – связь, по которой от компонента в компонент (компоненты) «База данных» посылается запрос, сформированный в формате языка SQL;

– $S_2 = (b_2\eta^-n_2) \rightarrow \{Answer\}$ – связь для получения ответа на запрос из БД;

– $S_3 = (b_3\eta^-n_3) \rightarrow \{Control\}$ – связь для получения значения, по которому должен осуществляться поиск в базе данных;

– $S_4 = (b_4\eta^+n_4) \rightarrow \{Value\}$ – связь для передачи найденного в БД значения *Value* для его дальнейшего использования в компонентах логического уровня многоуровневой КЦ СТУС.

Разработанные отдельно друг от друга компоненты «База данных» и «Запрос» позволяют организовать различные запросы к одной и той же БД либо адресовать один и тот же запрос одновременно к нескольким базам данных.

4.9 Принцип использования атрибутивных связей компонентов

При работе сформированных на логическом уровне многоуровневой КЦ



А1: площадь контакта фаз

Рисунок 4.34 – Компонент-атрибут «Площадь контакта фаз» компонента Абсорбер (рис. 3.23)

СТУС сценариев их исследования и функционального проектирования возникает необходимость изменения значений параметров компонентов объектного или визуального уровня, которые не являются переменными их связей. Такие параметры, относящиеся к данным числовых и нечисловых типов, называются

атрибутами. Их значения могут задаваться пользователем с помощью специальных интерфейсных элементов до начала работы многоуровневой КЦ СТУС. Для изменения значений атрибутов в процессе ее функционирования применяются так называемые *атрибутные связи*. Являясь виртуальными, эти связи представляются на логическом уровне многоуровневой КЦ в виде специального компонента «Атрибут», приведенного на рисунке 4.34 и связанного с компонентом объектного или визуального уровня. Его параметр получает значение X , принятое по единственной связи компонента «Атрибут»

$$S_1 = (b_1 \eta^{-1} n_1) \rightarrow \{X\}. \quad (4.66)$$

С помощью трех представленных выше компонентов: «База данных» (рис. 4.32), «Запрос» (рис. 4.33) и «Атрибут» (рис. 4.34) реализуются механизмы параметризации компонентной цепи ХТС, сформированной на объектном уровне многоуровневой КЦ СТУС. Пусть на её объектном уровне на основе компонента «Абсорбер», представленного на рисунке 3.23, реализована модель блока абсорбционной осушки природного газа, приведенная на рисунке 3.24. Уравнения математической модели компонента «Абсорбер» (3.60)-(3.62) включают следующие параметры:

– β_{iy}, β_{ix} – коэффициенты массопередачи абсорбируемого вещества в отдающей и принимающей фазах;

$y_{ip}^\Gamma, x_{ip}^\Gamma$ – равновесные концентрации абсорбируемого вещества соответственно в отдающей и принимающей фазах;

S – площадь поверхности контакта фаз.

Принимая во внимание, что коэффициенты массопередачи для отдающей и принимающей фазы одинаковые и хранятся в БД параметров и характеристик веществ, а площадь контакта фаз, являясь конструктивным параметром, хранится в БД характеристик аппаратов химической технологии, алгоритмическая КЦ параметризации модели абсорбера $A1$, принадлежащая логическому уровню многоуровневой КЦ СТУС, будет иметь вид, представленный на рисунке 4.35.

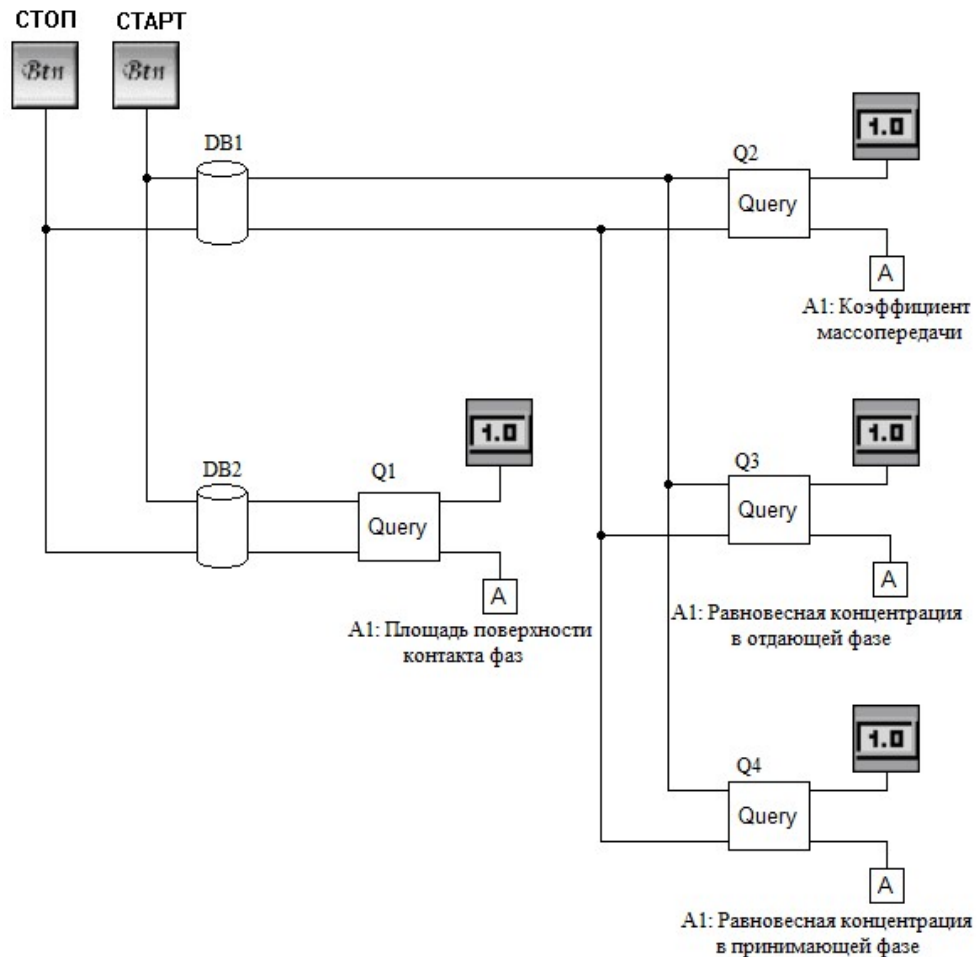


Рисунок 4.35 – Принцип параметризации модели компонентов объектного уровня с использованием баз данных

4.10 Выводы

1. На основе рассмотрения и анализа существующих языков программирования, в том числе языков программирования промышленных контроллеров международного стандарта МЭК 61131-3, предложен язык моделирования алгоритмических конструкций, являющийся одним из подязыков языка многоуровневых компонентных цепей. Язык МАК предназначен для формирования моделей сценариев исследования и функционального проектирования сложных технических (технологических) управляемых систем, включающих в себя сценарии функционирования устройств управления. Разработана грамматика языка МАК, в которой выделены основные операторы и операнды (константы, переменные и функции).

2. Разработаны правила формирования выражений языка МАК, которыми являются математико-алгоритмические конструкции, принадлежащие логическо-

му уровню многоуровневых компонентных цепей сложных технических управляемых систем.

3. Сформулирована и доказана теорема о представлении математико-алгоритмических выражений в формате языка алгоритмических компонентных цепей. В соответствии с ней каждому нетерминальному символу грамматики языка МАК ставится в соответствие определенный компонент, отображающий его в формате алгоритмических КЦ.

4. Сформулированы правила формализованного отображения дерева вывода математико-алгоритмических конструкций в алгоритмические компонентные цепи на основании правил преобразования нетерминальных символов языка МАК в его терминальные символы.

5. На основе сформированных правил разработано формализованное отображение скалярных и векторно-матричных математических конструкций языка МАК в формат алгоритмических компонентных цепей. Согласно ему каждому операнду, значение которого известно, ставится в соответствие компонент-источник значения соответствующего типа данных. Каждому скалярному или векторно-матричному оператору или функции ставится в соответствие компонент-преобразователь, отображающий и выполняющий эту операцию в формате алгоритмических компонентных цепей. Каждому неизвестному операнду, значение которого подлежит определению и передаче на визуальный или объектный уровень МКЦ, ставится в соответствие компонент данного уровня или компонент-атрибут.

6. Разработаны компоненты интеграции многоуровневых компонентных цепей СТУС с базами данных и предложен принцип использования атрибутивных связей для параметризации компонентов визуального и объектного уровней многоуровневой компонентной цепи СТУС с помощью моделей сценариев, представленных на ее логическом уровне.

Глава 5. Язык виртуальных инструментов и приборов

В главе описываются нотации языка виртуальных инструментов и приборов, включающие в свой состав визуальные компоненты и позволяющие формировать виртуальные приборы, предназначенные для автоматизации виртуальных (вычислительных) и реальных (натурных) экспериментов. Каждый виртуальный прибор (ВП), включающий в себя компоненты языков ВИП и МАК, может быть обособлен в конкретную подцепь и добавлен в библиотеку моделей компонентов среды моделирования химико-технологических систем. Для разработки ВП разработан комплекс программ «Система виртуальных инструментов и приборов» [178], являющийся подсистемой разработанного в диссертации комплекса программ.

5.1. Многоуровневая структура виртуального прибора

В данное время на рынке существует множество программных продуктов, применяемых для проведения автоматизированных экспериментов различных видов и назначения. Большинство из них, имея достаточно широкую гамму виртуальных инструментов и приборов [179, 180, 181], позволяют исследовать физические объекты электрической природы, предоставляя измеренную информацию через специализированный контроллер, но не допускают автоматического соединения с системами компьютерного моделирования, позволяющими производить исследование компьютерных моделей физически неоднородных технических объектов, в том числе и химико-технологических систем. Для автоматизации экспериментов, направленных на исследование и функциональное проектирование СТУС, должны быть разработаны как стандартные виртуальные приборы, так и предоставлены средства для формирования различных приборов пользователем на основе многоуровневых компонентных цепей.

В рамках метода компонентных цепей моделирования *виртуальным прибором* будем называть оконный интерфейс, имитирующий на компьютере логику работы реального прибора и предназначенный для генерации сигналов на реальную ХТС и/или её КМ, являющуюся подцепью многоуровневой компонентной

цепи, а также для отображения данных измерения сигналов и результатов анализа модели [182].

Для реализации поставленных перед собой целей ВП по аналогии с реальным источником, генератором или измерительным прибором [183] должен содержать:

- *лицевую панель*, представляющую собой замкнутую экранную область, в которой производится визуализация результатов моделирования и измерения, а также располагаются органы управления параметрами и характеристиками прибора;

- *модель алгоритма функционирования прибора*, осуществляющую обработку поступающих в прибор данных для их вывода на лицевую панель или подготовку информации для ее передачи на исследуемые с помощью прибора объект или модель;

- *интерфейс взаимодействия с объектом или моделью*, представляющий собой выводы прибора для его подсоединения к необходимым для исследования точкам объекта или модели.

Виртуальный прибор формируется из *виртуальных инструментов*, каждый из которых представляет собой неделимую часть и осуществляет некоторую функцию виртуального прибора. К ним относятся алгоритмические компоненты множества K_A (2.67), реализующие алгоритмы работы ВП, а также *визуальные компоненты множества K_V* (2.68), из которых формируется лицевая панель прибора.

Для построения виртуальных приборов может быть применена структура многоуровневой КЦ СТУС. В этом случае она имеет вид, представленный на рисунке 5.1, а каждый её уровень содержит:

- ***Визуальный уровень*** многоуровневой КЦ ВП содержит его лицевую панель, состоящую из визуальных компонентов отображения данных в понятном пользователю виде и органов интерактивного управления параметрами и характеристиками прибора.

– На ее *логическом уровне* располагается сценарий функционирования прибора, представляющий собой одну или несколько визуально не взаимосвязанных между собой алгоритмических компонентных подцепей (2.63).

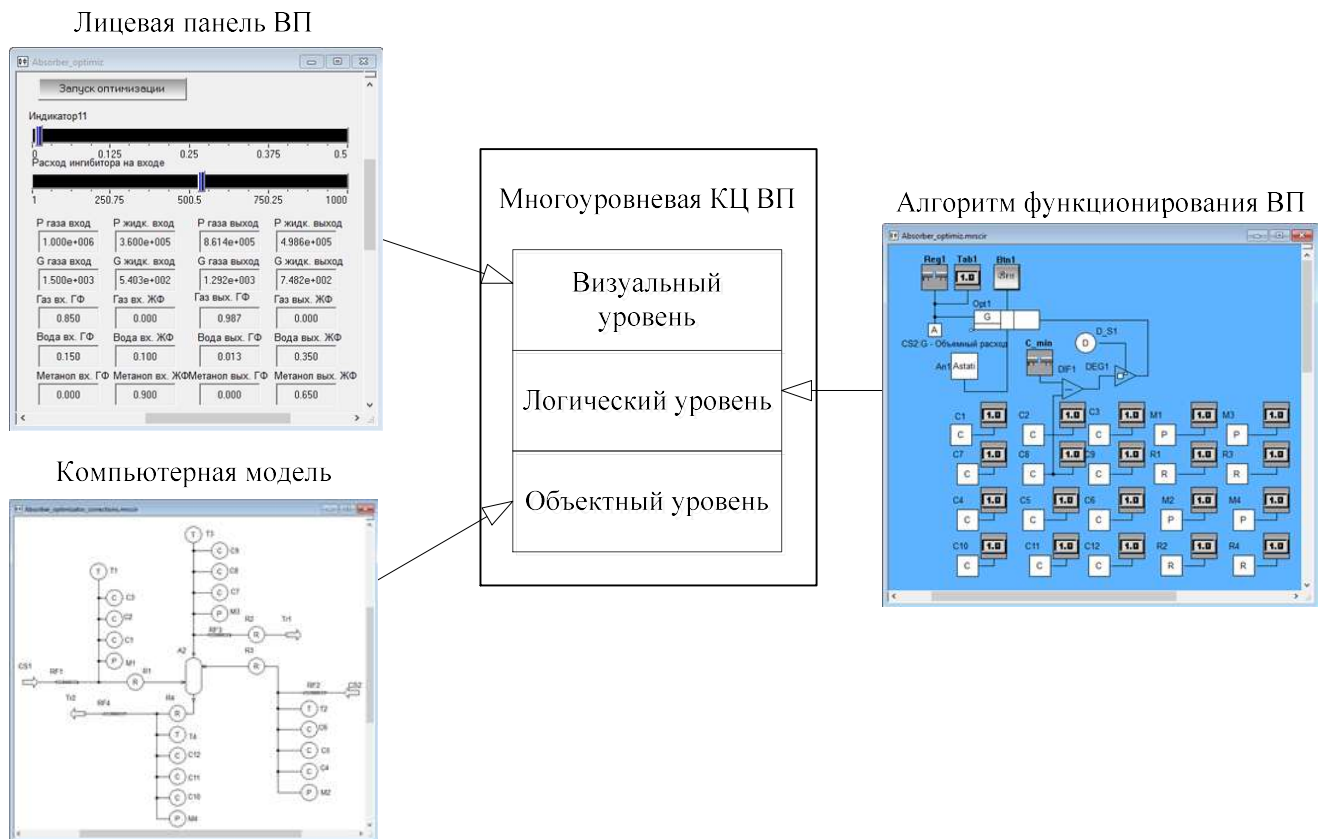


Рисунок 5.1 – Многоуровневое представление виртуального прибора

– *Объектный уровень* представляет собой интерфейс взаимодействия ВП с исследуемой КЦ или реальной ХТС.

Представленная на рисунке 5.1 структура может быть обособлена в конкретный виртуальный прибор, представляющий собой макрокомпонент, сформированный на основе метода подцепей (2.2). Используемые для создания его лицевой панели компоненты визуального уровня используются также для разработки панелей визуализации и интерактивного управления, входящих в состав МКЦ СТУС.

5.2. Формализованное представление виртуального прибора

Любой виртуальный прибор генераторного или измерительного типа, предназначенный для проведения реального (натурного) эксперимента или макси-

мального приближения виртуального (вычислительного) эксперимента к натурному [125], в формате МКЦ формально можно представить в виде совокупности трех подцепей

$$C^{ВП} = (C_o^{ВП} \cup C_L^{ВП} \cup C_V^{ВП}) \quad (5.1)$$

где $C_o^{ВП}$ – компонентная цепь объектного уровня ВП, на котором формируется интерфейс его подключения к исследуемой реальной химико-технологической системе и (или) её КЦ, представленной в формате языка ХТС; $C_L^{ВП}$ – алгоритмическая КЦ виртуального прибора вида (2.63), включающая подцепи сценария его функционирования, сформированные из компонентов множества K_A (2.67); $C_V^{ВП} = \{K_V^{ВП}\}$ – совокупность визуальных компонентов, составляющих лицевую панель виртуального прибора.

Компонентная цепь объектного уровня ВП $C_o^{ВП}$ представляет собой множество взаимосвязанных компонентов объектного уровня, с помощью которых прибор интегрируется с исследуемым объектом или его компьютерной моделью. Интеграция с реальным объектом осуществляется аппаратно-программным комплексом «Лабораторное автоматизированное рабочее место» (ЛАРМ) [106], инкапсулирующим в себе совокупность источников постоянного и генераторов переменного тока и множество измерительных приборов, предназначенных для исследования объектов электрической природы. Исследование объектов неэлектрической природы, в том числе и химико-технологических систем, осуществляется путем подключения входов реальных исполнительных устройств к источникам или генераторам аппаратно-программного комплекса ЛАРМ и выходов датчиков различных величин к входам его измерительных приборов (вольтметру, амперметру, двухканальному осциллографу, а также к измерителям цифровых сигналов).

С помощью совокупности алгоритмических компонентных подцепей логического уровня $C_L^{ВП}$ описывается функционирование виртуального прибора. Оно заключается в получении, необходимой обработки и подготовки к визуализации получаемых данных, которые могут являться результатами измерения характеристик реального объекта или значениями переменных исследуемой модели, полу-

ченными в статическом или динамическом (во временной или частотной области) режимах. В алгоритмические КЦ могут быть включены блоки обработки данных, обобщенный компонент которых приведен на рисунке 4.14, а также блоки и подцепи вычисления значений параметров прибора, исследуемого с его помощью объекта или модели, подаваемые из ВП с помощью специализированных команд.

Совокупность визуальных компонентов $C_V^{ВП} = \{K_V^{ВП}\}$, из которых формируется лицевая панель прибора, включает в себя: компоненты статической информации множества K_{V1} из (2.68); компоненты-визуализаторы результатов моделирования и данных измерения множества K_{V2} ; компоненты-регуляторы множества K_{V3} , с которыми оперирует пользователь во время проведения эксперимента; компоненты множества K_{V4} , обладающие специальным оконным интерфейсом для более детального исследования временных, частотных или параметрических характеристик.

Из представленных совокупностей компонентов может быть сформирован любой прибор генераторного или измерительного типа, предназначенный для исследования ХТС и (или) их компьютерных моделей, представленных подцепями объектного уровня МКЦ в формате языка моделирования ХТС. Разработка прибора осуществляется по стандартному алгоритму.

5.3. Модели виртуальных приборов с атрибутивными связями

При формировании виртуального прибора в алгоритмической КЦ, описывающей алгоритмы функционирования ВП, происходит обмен информацией различных типов данных. У визуальных компонентов может быть множество атрибутов. Для изменения их значений не целесообразно выносить соответствующие узлы к изображению компонента на логическом уровне многоуровневой КЦ ВП.

Для варьирования значения некоторого атрибута визуального компонента множества K_V (2.68), используется компонент-атрибут, представленный на рисунке 4.34.



Рисунок 5.2 – Возможность изменения атрибута компонента

- а) отображение компонента на визуальном слое;
- б) отображение компонента на логическом слое;
- в) атрибутивный компонент визуального компонента на логическом слое

На рисунке 5.2 показан пример изменения максимального значения (смотри рисунок 5.2, в) компонента «Стрелочник», представленный на рисунках 5.2, а и б. При получении сообщения, тип данных которого совпадает с типом данных атрибута «Максимальное значение шкалы регулятора», производится изменение значения соответствующего атрибута компонента «Стрелочник».

5.4. Последовательность действий пользователя при формировании виртуального прибора

Виртуальный прибор генераторного или измерительного типа формируется пользователем из визуальных компонентов K_V , составляющих лицевую панель прибора, а также из логических компонентов множества K_A (2.63), реализующих сценарии его функционирования на логическом уровне его КЦ. Помимо этого, для проведения виртуальных экспериментов формируется подцепь его сопряжения с исследуемой КЦ ХТС на объектном уровне. При исследовании реального объекта в сценарий функционирования прибора добавляются компоненты сопряжения с аппаратно-программными комплексами, например, функционирующими на базе языка управления механизмами X-Robot [184].

Последовательность действий пользователя по формированию виртуального прибора, представленная на рисунке 5.3, состоит из следующих этапов:

1. На этапе *Составления технического задания на прибор* выявляются задачи, на решение которых направлен формируемый ВП, а также производится выявление его параметров, подлежащих варьированию с помощью компонентов-

регуляторов на лицевой панели прибора, их интервалов варьирования и единиц измерения.



Рисунок 5.3 – Последовательность действий пользователя по формированию виртуального прибора

2. На основании сформированного технического задания на прибор производится **Формирование лицевой панели прибора**. Оно заключается в выборе необходимых визуальных компонентов множества K_V (2.68) и их параметров. На данном этапе производится задание геометрических размеров компонентов на визуальном уровне, шрифтов отображения числовой и строковой информации, а

также взаимное расположение визуальных компонентов, составляющих лицевую панель прибора.

3. После того, как сформирована лицевая панель виртуального прибора, ее визуальные компоненты будут иметь свои отображения на логическом уровне многоуровневой КЦ ВП. Путем добавления на этот уровень необходимых логических компонентов множества K_A (2.63) и их соединения в соответствующие алгоритмические подцепи, осуществляющих преобразование входных данных в выходные, производится *формирование сценариев работы прибора*.

4. *Формирование интерфейсных связей прибора с реальным объектом или его моделью* производится путем оформления данного прибора в макрокомпонент. Для этого применяется специальный компонент «Интерфейс», пред-

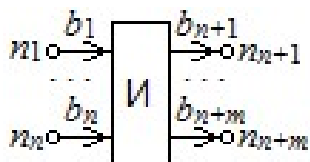


Рисунок 5.4 – Компонент «Интерфейс» для создания виртуальных приборов в виде макрокомпонентов

ставленный на рисунке 5.4. Его задачами являются формирование необходимых связей прибора с другими компонентами, а также задание параметров для создания лицевой панели прибора в виде отдельного окна. Любая компонентная цепь, включающая в себя компонент «Интерфейс», может быть сохранена в библиотеку моделей компонентов как макрокомпонент ВП. В этом случае он будет обладать специальным интерфейсным окном, содержащем в себе лицевую панель прибора, а также в виде многоуровневой подцепи входить в МКЦ исследуемой СТУС.

5. *Тестирование прибора* заключается в проверке работоспособности всех функциональных узлов ВП. Для этого прибор подвергается функционированию во всех режимах работы, заявленных в техническом задании, проверяя при этом правильность задания параметров с помощью компонентов-регуляторов множества K_{V3} (2.68) и отображения измеряемых переменных на компонентах-визуализаторах K_{V2} .

На основании приведенной последовательности действий любой пользователь имеет возможность формирования виртуального генератора или измерительного прибора для его использования при исследовании реальных управляемых

технических объектов и их компьютерных моделей, сформированных на объектном уровне многоуровневой модели ВП.

Предложенную последовательность действий продемонстрируем на примере создания виртуального прибора «Мультиметр», предназначенного для измерения текущих значений основных характеристик электрических цепей, которые могут входить в исполнительные или измерительные устройства ХТС.

1. Составление технического задания на прибор «Мультиметр».

Мультиметр – это виртуальный измерительный прибор, предназначенный для измерения текущих (для постоянного тока) и действующих (для переменного тока) значений напряжения и тока, активного сопротивления и активной мощности. Создаваемый прибор должен обладать следующими характеристиками: диапазон измерения напряжения – 1 мкВ – 1 МВ; диапазон измерения тока от 1 мкА до 1 МА; диапазон измерения активного сопротивления от 1 пОм до 1 ТОм; диапазон измерения активной мощности от 1 пВт до 1 ТВт.

2. **Формирование лицевой панели прибора.** Лицевая панель ВП «Мультиметр», представленная на рисунке 5.5, включает следующие элементы:

– *Цифровое табло «Значение»*, выводящее необходимое измеряемое

или рассчитываемое на основе измерений значение;

– *Комбинированный список «Величина»*,

позволяющий выбирать необходимую для измерения и визуализации характеристику исследуемого участка электрической цепи. К таким величинам относятся: напряжение, ток, сопротивление, мощность.

– *Комбинированный список «Форма представления»*,

позволяющий выбирать форму представления вещественного числа: с фиксированной запятой или с плавающей запятой;

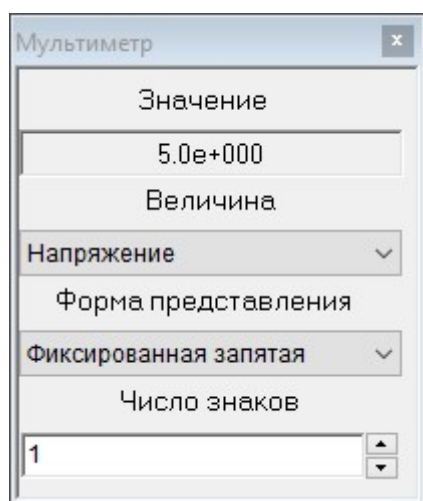


Рисунок 5.5 – Лицевая панель виртуального прибора «Мультиметр»

– *цифровое число со спином «Число знаков»*, позволяет изменять количество знаков после запятой в визуализируемом значении.

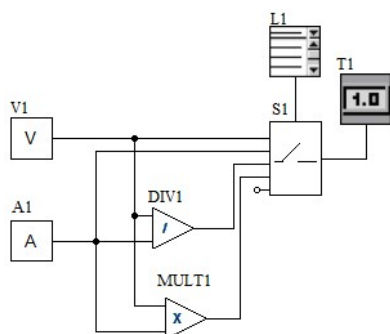


Рисунок 5.6 – Алгоритмическая компонентная цепь функционирования ВП «Мультиметр»

3. Формирование сценариев работы прибора осуществляется с применением описанных в главе 4 диссертации компонентов языка моделирования алгоритмических конструкций и измерительных компонентов языка моделирования ХТС, осуществляющих первичные измерения текущих значений напряжения и тока.

В представленной алгоритмической КЦ функционирования ВП «Мультиметр» используются следующие компоненты:

– *Вольтметр V1 и амперметр A1* используются для измерения напряжения $U1$ и тока ветви $I1$ соответственно ветви, в которую включен виртуальный прибор «Мультиметр» для исследования ее характеристик;

– *Деление DIV1* применяется для определения сопротивления

$$R1 = \frac{U1}{I1} \quad (5.2)$$

исследуемой ветви электрической цепи, с которой включен ВП;

– *Умножение MULT1* позволяет вычислить мощность, потребляемую исследуемой схемой с помощью ВП «Мультиметр» ветвью

$$P1 = U1 \cdot I1; \quad (5.3)$$

– С помощью *комбинированного списка L1* пользователем выбирается измеряемая характеристика обозначенной ветви электрической цепи. Каждой характеристике соответствует определенный порядковый номер: 0 – напряжение; 1- ток; 2 – сопротивление; 3 – мощность;

– *Алгоритмический ключ S1* осуществляет передачу на свой единственный выход значения, пришедшего на информационный вход, номер которого был передан последним комбинированным списком $L1$;

– *Цифровое табло T1* предназначено для визуализации полученного значения на лицевой панели виртуального прибора.



Рисунок 5.7 – Установка формата числа

Для настройки параметров цифрового табло *T1*, предназначенного для вывода результата на лицевую панель прибора, используются:

– *комбинированный список L2*, позволяющий выбрать одну из форм представления вещественного числа: 0 – фиксированная запятая, 1 – плавающая запятая. Его выход подключен к компоненту-атрибуту T1:формат числа. Соответствующая алгоритмическая КЦ приведена на рисунке 5.7.

– *цифровое табло со спином T2* предназначено для установки количества знаков после запятой в визуализируемом результате. Установка соответствующего значения в качестве атрибута цифрового табло *T1* производится с помощью алгоритмической КЦ, представленной на рисунке 5.8.



Рисунок 5.8 – Установка количества знаков после запятой

4. Формирование интерфейсных связей с исследуемой моделью осуществляется на объектном уровне многоуровневой КЦ ВП. Её компонентная подцепь объектного уровня, полученная в результате установки связей измерительных компонентов – вольтметра *V1* и амперметра *A1* с компонентом Интерфейс, представленном на рисунке 5.4, приведена на рисунке 5.9.

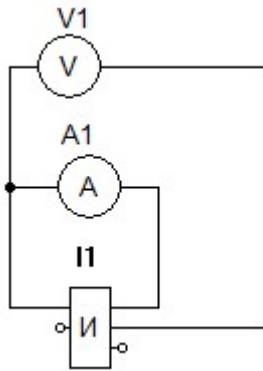


Рисунок 5.9 – Компонентная цепь объектного уровня ВП «Мультиметр»

После сохранения полученной многоуровневой КЦ как макрокомпонента в библиотеку макрокомпонентов среды моделирования ХТС будет добавлен макрокомпонент «Мультиметр», представленный на рисунке 5.10.

5. Для тестирования прибора сформирована КЦ электрической цепи постоянного тока, представленная на рисунке 5.11. С помощью виртуального прибора «Мультиметр» измерению подлежат все характеристики



Рисунок 5.10 – Компонент «ВП Мультиметр»

сопротивления R2. Результаты измерения представлены на рисунках 5.12,а-г.

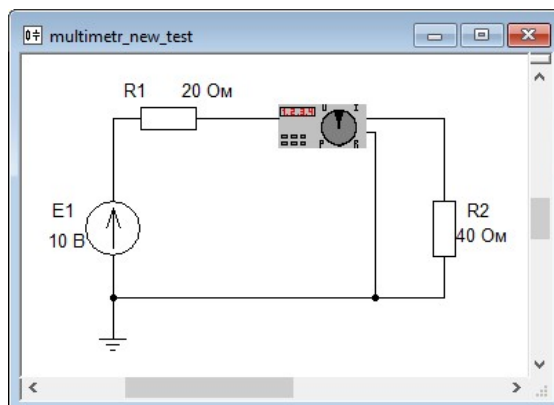
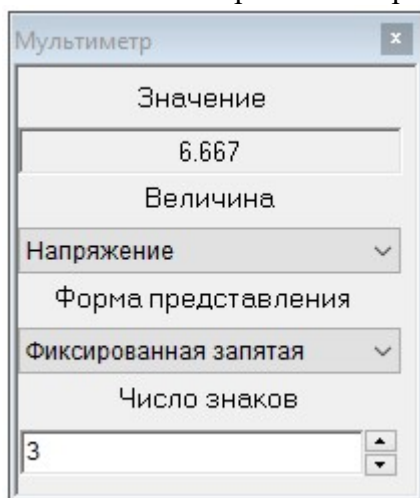
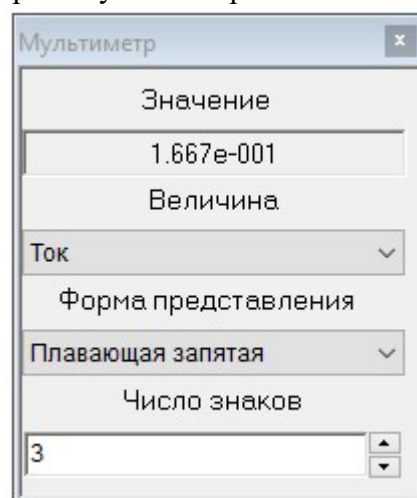


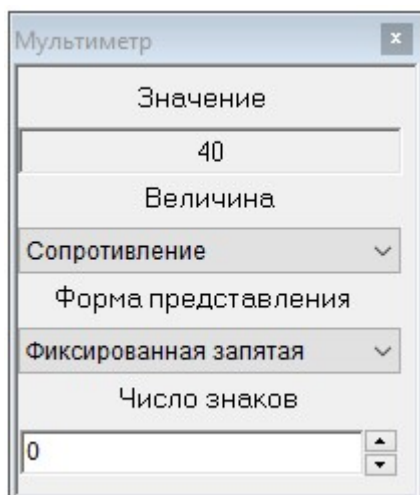
Рисунок 5.11 – Компьютерная модель электрической цепи для тестирования виртуального прибора «Мультиметр»



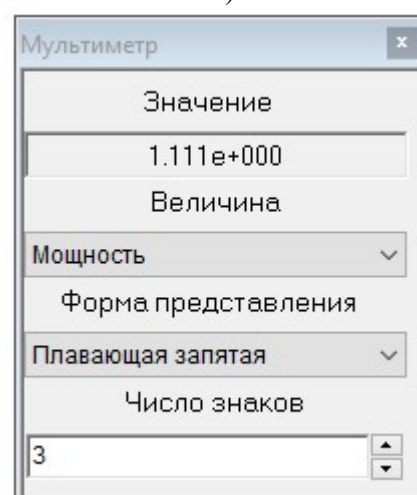
а)



б)



в)



г)

Рисунок 5.12 – Тестирование виртуального прибора «Мультиметр»

- а) в режиме измерения напряжения;
- б) в режиме измерения тока;
- в) в режиме измерения сопротивления;
- г) в режиме измерения мощности

Аналогичным образом могут быть разработаны любые виртуальные приборы генераторного и измерительного типа, предназначенные для исследования реальных объектов и (или) их компьютерных моделей.

5.5. Формирование функциональных блоков виртуальных инструментов и приборов

При формировании ряда виртуальных приборов генераторного и измерительного типа было выявлено, что большинство приборов содержит одинаковые функциональные части, выполняющие одни и те же функции и различающиеся лишь параметрами пределов измерения визуальных компонентов множества K_V (2.68). Такие части могут быть представлены в библиотеке моделей компонентов как функциональные блоки виртуальных приборов.

Функциональным блоком (ФБ) будем называть типовую структуру виртуальных приборов, представляющую собой многоуровневую подцепь, состоящую из визуальных компонентов и связанного с ними сценария обработки информации. Основным назначением ФБ является встраивание в ряд ВП, в которых требуется выполнение данной функции. С формальной точки зрения ФБ можно представить в виде виртуального прибора (5.1), у которого $C_O^{ВП} = \emptyset$, $C_L^{ВП} \neq \emptyset$ и $C_V^{ВП} \neq \emptyset$.

По своему назначению функциональные блоки подразделяются на:

- *задающие ФБ*, предназначенные для вычисления значений параметров компонентов модели исследуемого объекта или задающих воздействий на реальную ХТС, представленных в виде передаваемых сигналов;

- *визуализирующие ФБ*, осуществляющие сборку результатов измерений и моделирования, их математическую обработку и визуализацию.

Задающие функциональные блоки представляют собой совокупность визуальных компонентов, с помощью которых осуществляется задание каких-либо параметров ВП на визуальном уровне его многоуровневой КЦ, значения которых передаются на её логический уровень (2.52), где осуществляется их обработка сценариями функционирования ВП с передачей ее результатов на объектный уро-

вень (2.54). Типовая структура задающего ФБ и протекающий в нем информационный поток представлены на рисунке 5.13.

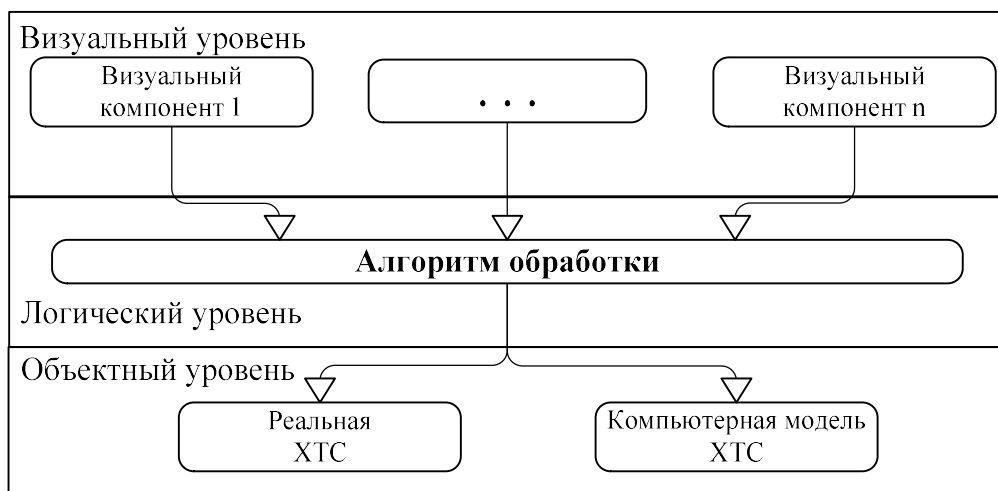


Рисунок 5.13 – Структура задающего функционального блока

Визуализирующие функциональные блоки представляют собой типовые сценарии обработки данных измерения и результатов моделирования с целью их визуализации в числовой или графической форме. Для этого необходимые данные поступают с объектного уровня многоуровневой КЦ на её логический уровень (2.50), где производится их обработка и последующая передача на визуальный уровень (2.51) с целью их визуализации. Структура визуализирующего функционального блока и передача информации между уровнями модели приведены на рисунке 5.14.

Типовые функциональные блоки позволяют определять характеристики, присущие большинству виртуальных генераторов и измерительных приборов. Например, к таким блокам в области исследования электротехнических исполнительных устройств и датчиков, осуществляющих преобразование физических величин различной физической природы в электрические сигналы, относятся:

- амплитуда сигнала (напряжения или тока), его частота, фазовый сдвиг относительно некоторого опорного сигнала той же частоты, задаваемые для виртуальных генераторов сигналов различных форм;

- параметры-функционалы сигналов во временной области (период и частота сигнала, время переходного процесса, установившееся значение, перерегулирование и др.);

– параметры-функционалы частотных характеристик (резонансная частота, частота среза, значение фазы на частоте среза, запасы устойчивости по амплитуде и фазе и пр.);

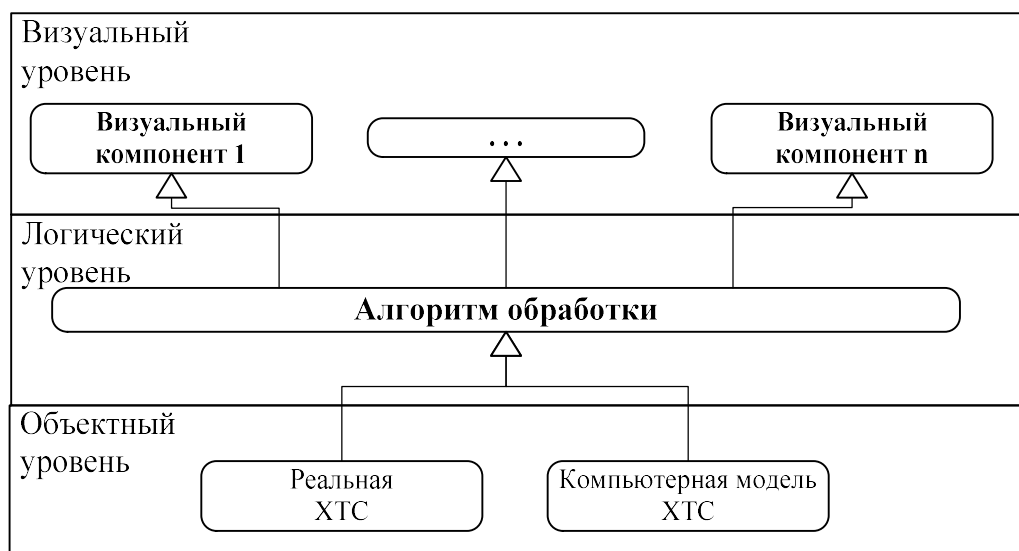


Рисунок 5.14 – Структура визуализирующего функционального блока

– параметры-функционалы параметрических характеристик, полученных в результате многовариантного анализа (см. пункт 4.7). К характеристикам данного класса относятся коэффициент передачи, коэффициент полезного действия, рабочая точка нелинейных элементов и др.;

– характеристики виртуальных приборов визуализирующего типа (развертка прибора, усиление и смещение сигналов, синхронизация).

Каждая их перечисленных характеристик является универсальной и может быть добавлена в любой прибор. Далее рассмотрим некоторые типовые функциональные блоки, реализованные в среде компьютерного моделирования ХТС.

Функциональный блок задания значения (амплитуды) напряжения. Значение (амплитуда) напряжения в виртуальных приборах генераторного типа задается в виде числового значения *амплитуды* и единицы измерения – *порядка амплитуды*. На визуальном уровне многоуровневой КЦ для задания этих величин используется цифровое табло со спином, предназначенное для задания целочисленных значений, и комбинированный список, позволяющий выбирать из списка необходимую единицу измерения. Каждому порядку значения амплитуды соответствует определенный порядковый номер, относительно которого производится

расчет значения амплитуды. В представленном на рисунке ФБ микровольтам (мкВ) – 0, милливольтам (мВ) – 1, вольтам (В) – 2, киловольтам (кВ) – 3, мега-вольтам (МВ) – 4. Результат работы данного ФБ, которым является вычисленное значение амплитуды, устанавливается в атрибут источника гармонического сигнала ES1 и передается в цифровое табло T2 для визуализации на лицевой панели прибора.

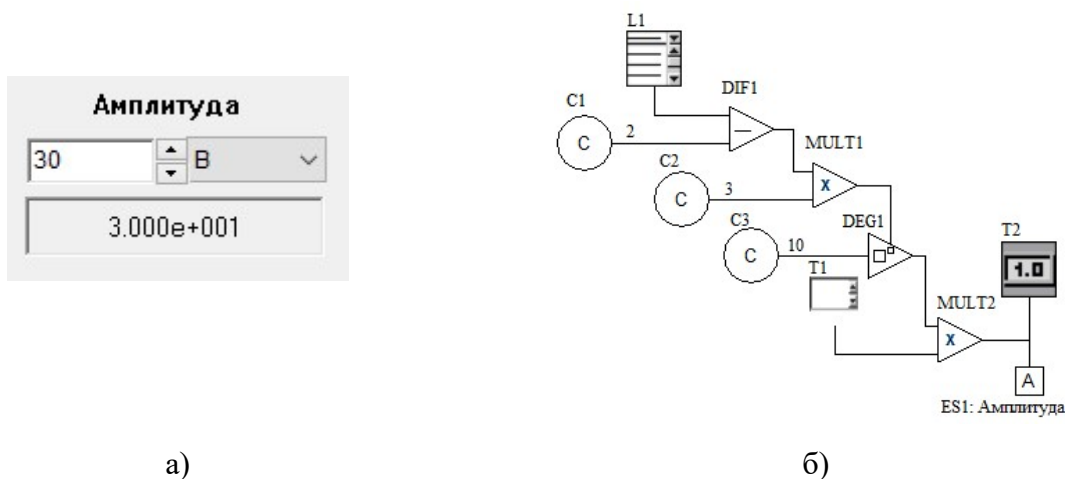
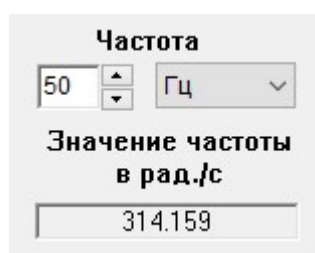


Рисунок 5.15 – Функциональный блок задания значения (амплитуды) сигнала
 а) лицевая панель
 б) алгоритмическая компонентная цепь

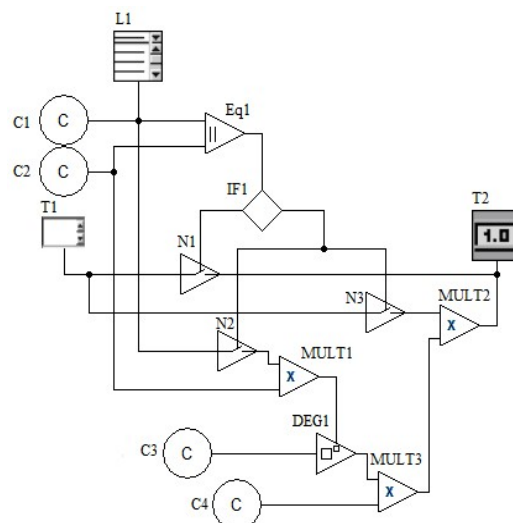
С помощью цифрового табло T1 задается значение напряжения, которое измеряется в единицах измерения, устанавливаемых с помощью комбинированного списка L1. Так как Вольтам соответствует порядковый номер 2, то он вычитается из выбранного пункта из списка с помощью компонента DIF1. Полученная разность умножается на 3, и их произведение будет являться показателем степени, в которую нужно возвести число 10. Результат возведения в степень компонентом MULT2 умножается на значение напряжения. Тем самым это значение переводится в вольты и затем устанавливается в качестве значения амплитуды напряжения компонента ES1.

Функциональный блок задания частоты. Частота сигнала, формируемая в генераторе, задается значением и соответствующей единицей измерения. Значение представляет собой целое число, для ввода которого в ФБ вводится числовое табло со спином T1. Единицы измерения частоты задаются с помощью комбинированного списка L1. В нем соответствующие поля имеют свои порядковые но-

мера: Герцы (Гц) – 0, килогерцы (кГц) – 1, мегагерцы (МГц) – 2, радианы в секунду (рад./с) – 3. Числовое табло T1 и комбинированный список L1 в совокупности с цифровым табло T2, визуализирующим передаваемое в генератор значение, представленное в радианах в секунду, образуют лицевую панель ФБ, представленную на рисунке 5.16, а. Его функционирование обеспечивает алгоритмическая КЦ, представленная на рисунке 5.16, б.



а)



б)

Рисунок 5.16 – Функциональный блок задания частоты
а) лицевая панель
б) алгоритмическая компонентная цепь

Аналогичным образом могут быть сформированы функциональные блоки для любых виртуальных приборов генераторного и измерительного типа.

5.6. Погрешности виртуальных приборов

Для целей введения и обоснования погрешностей виртуальных приборов рассмотрим основные понятия, связанные с метрологией, измерениями физических величин и погрешностей их изменений.

Физические величины (ФВ) – это свойства объекта, для которых могут быть установлены и воспроизведены градации определенного размера. Метрология [185], как наука об измерениях, имеет дело и изучает только физические величины, т.е. величины, для которых может существовать физически реализуемая и воспроизводимая единица величины. В теории измерений вводятся понятия истинного измеренного и действительного значений физической величины [186].

Основной постулат метрологии заключается в том, что истинное значение ФВ существует, однако определить его путем измерения невозможно.

Действительное значение – значение ФВ, найденное экспериментальным путем настолько близкое к истинному, что для данной цели может быть принято вместо него.

Под *измеренным значением* понимается значение величины, отложенное по отсчетному устройству средства измерения.

Измеряемая физическая величина – ФВ, подлежащая измерению в соответствии с поставленной измерительной задачей.

Влияющая физическая величина – ФВ, оказывающая влияние на объект или на средство измерения, что приводит к искажению результата измерений.

Физический параметр – это ФВ, характеризующая частичную особенность измеряемой величины, например, амплитуда колебаний, действующее или среднее значение напряжения.

Постоянная величина – ФВ, не изменяющаяся за время интервала измерения.

Переменная величина – ФВ, изменяющаяся по размеру за время интервала измерения.

Измерения – значения ФВ, которые находятся опытным путем с помощью специальных технических средств и характеризующиеся уравнением

$$X = n[x], \quad (5.4)$$

где X – измеряемая величина; $[x]$ – единица величины; n – число измерений.

По способу нахождения искомого значения измеряемой величины различают прямые (непосредственно по показаниям средств измерений) и косвенные измерения, рассчитываемые по формулам на основе прямых измерений. Например, измерение мощности является косвенным, так как её значение вычисляется по прямым измерениям напряжения и тока $P = U \cdot I$.

По режиму работы средства измерения разделяются на *статические* и *динамические*. Последние связаны, как правило, с наличием у измерительных

средств инерции (механической, тепловой, электрической) или трения, люфтов, остаточной жесткости и т.п.

Общим термином «Средства измерения» называют технические средства, используемые при измерениях и имеющие *нормированные метрологические характеристики*, влияющие на результаты и точность измерений.

Измерительный прибор – средство измерений, предназначенное для выработки сигнала измерительной информации в форме, доступной для непосредственного восприятия наблюдателем. В зависимости от формы представления информации различают аналоговые и дискретные (цифровые) приборы.

У аналоговых приборов показания являются непрерывной функцией измеряемой величины (например: стрелочный прибор).

В цифровых приборах осуществляется преобразование непрерывной функции в цифровой код, результат измерения отражается на цифровом табло.

Выше было дано определение виртуального прибора, который в зависимости от своей реализации может быть компьютерной моделью аналогового или цифрового прибора.

Количественной характеристикой качества измерения (качества функционирования прибора) является погрешность измерений, которая может быть одного из следующих типов:

– абсолютная погрешность, значение которой может быть вычислено по формуле

$$\Delta x = x_{\text{изм}} - x_{\text{д}} ; \quad (5.5)$$

– относительная погрешность, рассчитываемая по формуле

$$dx = \frac{x_{\text{изм}} - x_{\text{д}}}{x_{\text{изм}}} , \quad (5.6)$$

где $x_{\text{изм}}$ – измеренное с помощью прибора значение измеряемой величины; $x_{\text{д}}$ – её действительное значение.

Позитивной характеристикой качества измерений является *точность измерений*. Она связана с погрешностью обратной зависимостью – измерение тем более точно, чем меньше погрешность.

Погрешность как степень доверия к результатам измерений должна быть оценена. Различают измерения с *точным* (с наивысшей достижимой точностью) и *приближенным* оцениванием погрешности. При измерениях с точным оцениванием погрешности учитываются индивидуальные характеристики средств измерения, анализируется метод измерений, учитывается влияние условий измерения. При измерениях с приближенным оцениванием погрешностей оцениваются лишь нормативные, типовые метрологические характеристики средств измерения и оценивается влияние на результат лишь отклонений условий измерения от нормальных.

Классификация погрешностей. В целях единообразия подхода к анализу погрешностей в метрологии принята классификация погрешностей, приведенная в таблице 5.1

Таблица 5.1

Основные типы погрешностей

№	Условия проявления	Тип погрешности
1	Во времени	Систематические, случайные
2	По источнику возникновения	Методические, инструментальные и личные
3	По условиям возникновения	Основная и дополнительная
4	Изменяющиеся пропорционально измеряемой величине	Аддитивные и мультипликативные

1. По характеру проявления во времени выделяют *систематические и случайные (по знаку или величине) погрешности*:

Систематической называется такая погрешность, которая при повторных измерениях одной и той же величины в одних и тех же условиях остается постоянной или закономерно изменяется. Пример: динамическая погрешность прибора при детерминированных сигналах.

Случайная – это погрешность, которая при повторных измерениях одной и той же величины в одних и тех же условиях изменяется случайным образом по знаку и (или) величине. Случайная погрешность прибора возможна по ряду причин: из-за неточной разметки шкалы; деформации стрелки;– трения в опорах, колебаний температуры, изменения давления и пр.

2. По источнику возникновения различают *инструментальные, методические* и *личные* погрешности. Например: наличие внутреннего конечного значения сопротивления у прибора – амперметра – приводит к появлению методической погрешности. В общей погрешности прибора могут присутствовать как систематические, так и случайные погрешности, оказывающие свое влияние на результат измерения, их классифицируют как инструментальные.

3. По условиям возникновения у средств измерения различают *основную погрешность*, когда внешние условия в норме, и *дополнительную погрешность*, когда внешние условия вышли за пределы нормы.

4. *Аддитивной*, например, является *систематическая погрешность*, вызываемая неточной установкой нуля у стрелочного прибора, а *мультипликативной* – погрешность измерения времени отстающими или спешащими часами.

Используя введенные понятия и определения для традиционных средств и процессов измерений, рассмотрим далее виртуальные средства измерения и связанные с ними погрешности.

Виртуальный прибор генераторного или измерительного типа формируется пользователем из визуальных компонентов, представляющих собой лицевую панель прибора, и компонентов языка моделирования алгоритмических конструкций, реализующих сценарии его функционирования на логическом уровне многоуровневой КЦ ВП. Он предназначен для отображения данных измерения наблюдаемых характеристик реального объекта и результатов анализа его КЦ. На рисунке 5.17 приведена схема процесса образования и накопления ошибок измерения в ВП в результате выполнения ВЭ над компьютерной моделью ХТС, сформированной на объектном уровне многоуровневой КЦ СТУС.

Суммарная погрешность измерений ВП ε_{Σ} , согласно представленной на ри-

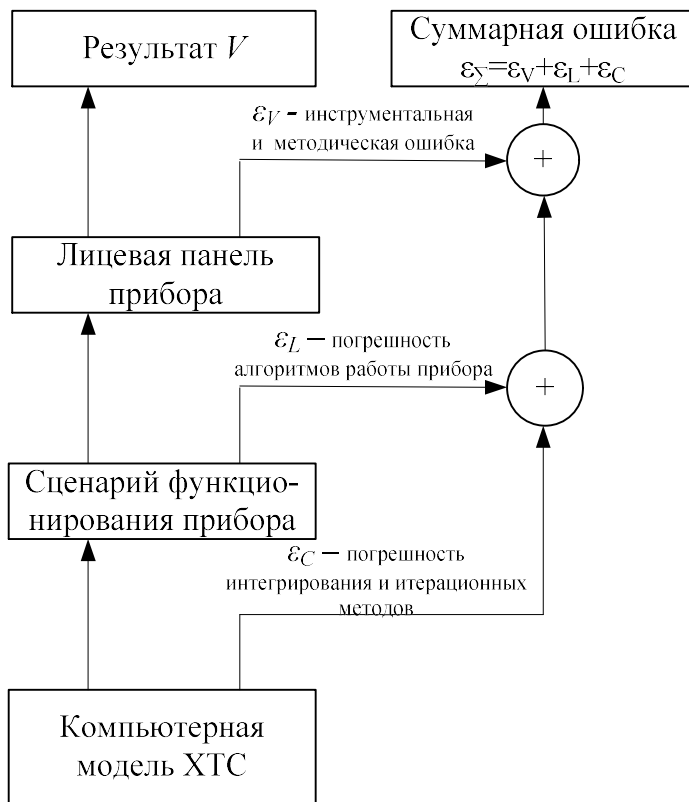


Рисунок 5.17 – Схема формирования погрешностей в виртуальном приборе при исследовании компьютерных моделей ХТС

сунке 5.17 схеме, складывается из погрешности моделирования универсальным вычислительным ядром [103] ε_C , образуемой объектом уровне многоуровневой КЦ, погрешностей сценариев функционирования прибора ε_L , расположенных на логическом уровне МКЦ ВП, и инструментальной погрешности ε_V , связанной с графическими возможностями персонального компьютера, на котором производится функционирование разработанного комплекса программ.

Погрешности процесса анализа КЦ ХТС универсальным вычислительным ядром ε_C связаны с ошибками интегрирования при неявной схеме Эйлера, схеме трапеций и других, которые реализованы в рамках вычислительного ядра.

Сценарии функционирования ВП, представленные на логическом уровне многоуровневой КЦ ВП, обычно формируются на явных математических и математико-алгоритмических выражениях. В их задачи входят определение физических параметров-функционалов измеряемых величин, таких как максимальные, минимальные, среднее, действующие значения периодических сигналов, их период и частота колебаний.

Величина локальной ошибки при интегрировании с помощью неявного метода Эйлера, являющегося основным методом алгебраизации дифференциальных уравнений в универсальном вычислительном ядре [103], вычисляется по формуле

$$\varepsilon_H = -\frac{h^2}{2} \frac{d^2\tau}{dt^2}. \quad (5.7)$$

Величина ошибки при интегрировании с помощью явного метода Эйлера рассчитывается на основании формулы

$$\varepsilon_{\text{я}} = \frac{h^2}{2} \frac{d^2(\tau)}{dt^2}. \quad (5.8)$$

В обоих случаях локальная ошибка пропорциональна квадрату шага интегрирования, но имеет разные знаки.

Рассмотрим поведение глобальной ошибки при использовании рассмотренных методов. Поскольку локальные ошибки могут принимать различные знаки, глобальная ошибка может бесконечно возрастать либо оставаться конечной. Это зависит от метода интегрирования, величины шага, а также поведения функции $f(t)$. Численная неустойчивость заключается в том, что при большом числе шагов решение расходится, т.е. начинает неограниченно возрастать, хотя истинное решение конечно.

Чтобы решение уравнений модели исследуемого с помощью ВП ХТС было устойчивым, требуется выполнение условия, накладываемого на шаг интегрирования h

$$|1 - h\lambda| \leq 1 \text{ или } h \leq \frac{2}{\lambda} = 2\tau, \quad (5.9)$$

где λ – минимальный корень характеристического уравнения или наименьшее собственное значение матрицы решения.

При управлении параметрами ВЭ, проводимого над КЦ ХТС, необходимо учитывать следующее ограничение:

К интерфейсным ошибкам в процессе визуализации и восприятия информации с экрана персонального компьютера можно отнести:

- 1) неточности шкал визуальных инструментов ВП;
- 2) ошибки визирования, связанные с точностью установки курсоров;

3) ошибки взаимного преобразования физических координат, получаемых с логического уровня модели прибора, в экранные координаты визуальных компонентов его лицевой панели.

Обобщенная погрешность измерений ε представляет собой алгебраическую сумму перечисленных выше погрешностей:

$$\varepsilon = \pm \varepsilon_C \pm \varepsilon_L \pm \varepsilon_V' \pm \varepsilon_V'' , \quad (5.10)$$

где ε_C – погрешность первичных результатов моделирования, вызванная наличием нелинейных и/или инерционных элементов, описываемых нелинейными (2.10) и/или дифференциальными (2.11) уравнениями, шагом моделирования и точностью задания начальных условий. Данная погрешность порождается на объектном уровне МКЦ СТУС при анализе модели ХТС одним из режимов анализа, реализованном в универсальном вычислительном ядре [103]. В случае измерения наблюдаемых характеристик реального объекта эта погрешность будет связана как с погрешностью самого измерительного прибора, ошибкой, порожденной АЦП измерительного контроллера, осуществляющего оцифровывание измеренных данных и их передачу в компьютер; ε_L – погрешность, порождаемая сценариями и блоками обработки данных экспериментов; ε_V' – погрешность визуального отображения информации, порождаемая переводом результатов измерения и моделирования, а также данных их численной обработки в экранные формы, численные значения, выводимые на экран, а также в формируемые на их основе временные, частотные или параметрические характеристики; ε_V'' – погрешность задания параметров исследуемой КЦ ХТС, а также управляющих воздействий на реальный объект.

5.7. Выводы

1. Предложена многоуровневая структура виртуального прибора, которая интерпретирована в структуру многоуровневой компонентной цепи СТУС. Согласно ей любой виртуальный прибор состоит из лицевой панели, формируемой из визуальных компонентов на визуальном уровне многоуровневой компонентной цепи; сценария функционирования прибора, представленного алгоритмической

КЦ на логическом уровне МКЦ; интерфейса взаимодействия виртуального прибора с исследуемым объектом и/или моделью, который формируется на её объектном уровне.

2. Разработано формализованное представление виртуального прибора, представленное в виде компонентной цепи, которая на основе метода подцепей включает в себя подцепи объектного, логического и визуального уровней МКЦ ВП.

3. Предложена методика использования моделей виртуальных приборов с атрибутивными связями, согласно которой каждому варьируемому атрибуту визуального компонента ставится в соответствие компонент «Атрибут» логического уровня. На его единственный узел поступает сообщение с новым значением варьируемого атрибута, которое устанавливается в компоненте.

4. Разработана и обоснована последовательность действий пользователя при формировании виртуального прибора. В неё помимо разработки лицевой панели, сценария функционирования и интерфейса взаимодействия ВП с исследуемым объектом и модели входят этапы формирования технического задания на прибор, его тестирование и добавление в библиотеку виртуальных приборов комплекса программ «Среда моделирования химико-технологических систем»

5. Сформирована методика формирования задающих и визуализирующих функциональных блоков, являющихся типовыми для большинства виртуальных приборов генераторного и измерительного типа. Реализован ряд примеров функциональных блоков, которые были применены при разработке виртуальных приборов, предназначенных для исследования компьютерных моделей и реальных объектов.

6. Разработана схема анализа погрешностей виртуальных приборов при выполнении вычислительных экспериментов над компонентными цепями ХТС с помощью универсального вычислительного ядра, входящего в состав комплекса программ «Среда моделирования химико-технологических систем».

Глава 6. Комплекс программ моделирования химико-технологических систем

Глава посвящена описанию структуры комплекса программ моделирования химико-технологических систем, предназначенного для автоматизации решения задач исследования и функционального проектирования сложных технических управляемых систем. В ней приводится описание алгоритмов работы основных его модулей, осуществляющих анализ математических моделей ХТС, представленных компонентными подцепями объектного уровня МКЦ, и имитационных моделей сценариев проведения экспериментов, направленных на исследование и функциональное проектирование СТУС на базе ХТС и являющихся алгоритмическими КЦ логического уровня.

6.1. Назначение и структура комплекса программ

На основе рассмотренной во второй главе структуры многоуровневой компонентной цепи, расширяющей возможности метода КЦ на сложные технические управляемые системы, разработан комплекс программ «Среда моделирования химико-технологических систем». Он предназначен для автоматизированного решения задач исследования и функционального проектирования на основе автоматизированных экспериментов над многоуровневыми компонентными цепями СТУС. Разработанный комплекс программ позволяет:

– формировать представленные в графическом виде многоуровневые КЦ СТУС, включающие в себя подцепи ХТС с неоднородными векторными связями (3.4) [123], модели сценариев автоматизированных экспериментов для решения задач исследования и функционального проектирования СТУС, представленных алгоритмическими подцепями на логическом уровне МКЦ, а также панели визуализации и интерактивного управления, представляющие собой наборы визуальных компонентов (2.68);

– производить анализ функционирования ХТС, представленных компонентными подцепями объектного уровня, сформированными в формате языка моделирования химико-технологических систем, между компонентами которой проте-

кают информационные, мультифизические энергетические и многокомпонентные вещественные потоки;

– выполнять сценарии экспериментов, представленные на логическом уровне многоуровневой КЦ СТУС в виде алгоритмических подцепей (2.63) и направленные на исследование и функциональное проектирование СТУС, в том числе на разработку и отладку сценариев функционирования устройства управления;

– осуществлять визуализацию результатов анализа КЦ ХТС и наблюдаемых характеристик реального объекта, а также осуществлять формирование и изменение воздействий пользователя на модель и на реальный объект;

– производить автоматизированное документирование результатов решения задач исследования и функционального проектирования СТУС с возможностью автоматизированного синтеза сценариев, на основе которых функционирует измерительно-управляющий контроллер.

Комплекс программ «Среда моделирования химико-технологических систем» функционирует под управлением операционной системы *Windows*. Для его разработки исследовались следующие языки и средства объектно-ориентированного программирования, формирования и обработки запросов к базам данных, представления, визуализации и документирования результатов моделирования и измерения:

– объектно-ориентированный язык программирования Visual C++ с библиотекой классов и функций MFC [187], применяемый для непосредственной разработки основных моделей комплекса программ;

– динамически загружаемая библиотека *ftd2xx.dll* – библиотека функций для работы с FDTI-устройствами [188], подключаемыми к компьютеру по шине USB. В комплексе программ таким устройством является универсальный измерительно-управляющий контроллер X-Mega [189], функционирующий на основе сценариев, написанных на языке X-Robot [184].

Для автоматизации вычислительных и натуральных экспериментов, направленных на решение задач исследования и функционального проектирования

СТУС в состав комплекса программ «Среда моделирования химико-технологических систем» [190], структура которого приведена на рисунке 6.1, входят:

– *многослойный редактор многоуровневой компонентной цепи сложной технической управляемой системы*, позволяющий в графическом виде формировать многоуровневые КЦ СТУС на трех взаимосвязанных слоях с применением трёх подязыков языка МКЦ;

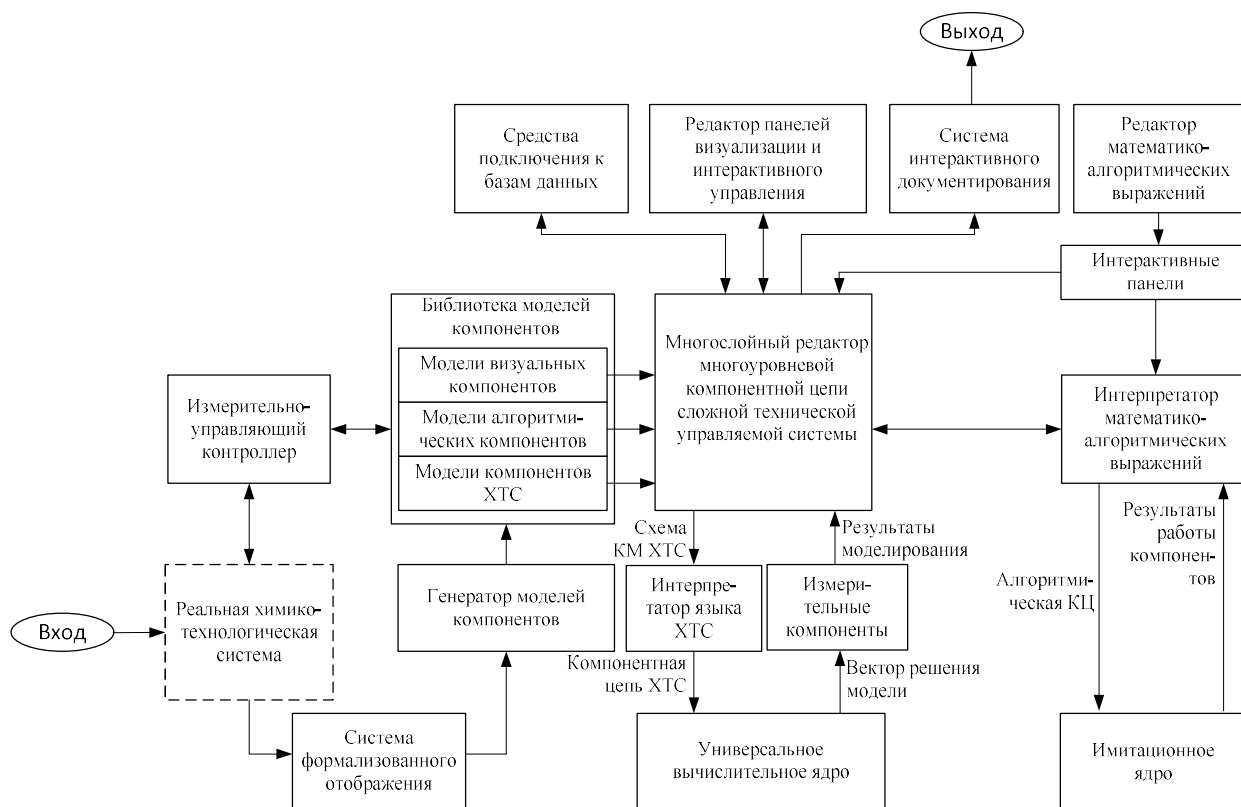


Рисунок 6.1 – Структура комплекса программ моделирования химико-технологических систем

– *программно-алгоритмические средства языка моделирования ХТС*, представленные интерпретатором языка ХТС, осуществляющим построение КЦ исследуемой химико-технологической системы на основе её графической модели, представленной на объектном слое редактора МКЦ; универсального вычислительного ядра [103], реализующего представленный на рисунке 2.4 алгоритм анализа КЦ ХТС, измерительные компоненты, приведенные в таблице 2.1 и осуществляющие выборку соответствующих значений из вектора решения (2.15) и их передачу с объектного уровня МКЦ СТУС на ее логический уровень (2.50);

– *программно-алгоритмические средства языка МАК*, реализованные в виде интерпретатора математико-алгоритмических выражений, имитационного ядра, в основу реализации которого положен алгоритм передачи сообщений [116], и интерактивных панелей, включающих в свой состав редактор математико-алгоритмических выражений. Помимо этого они включают в себя средства подключения к базам данных и средства интерактивного документирования;

– *средства языка ВИП*, представленные редактором панелей визуализации и интерактивного управления и библиотекой моделей визуальных компонентов;

– *библиотека моделей компонентов (БМК)*, в которой реализованы инструментальные средства хранения и использования компонентов всех трех подязыков языка многоуровневых КЦ, а также средства ее пополнения, представленные в виде генератора моделей компонентов [191] и интерактивных панелей.

Таким образом, комплекс программ моделирования химико-технологических систем, структура которого представлена на рисунке 6.1, позволяет автоматизировать решение задач исследования и функционального проектирования СТУС (1.2)-(1.10). Его интеграция с реальным объектом позволяет применять полученные с помощью модели результаты к реальной ХТС, тем самым проводя апробацию результатов и окончательную отладку сценариев управления до окончательного их программирования в контроллер.

6.2. Опции многослойного редактора

Многослойный редактор предназначен для формирования многоуровневых компонентных цепей СТУС, предназначенных для исследования и функционального проектирования сложных технических (технологических) управляемых систем. В нем реализованы стандартные для большинства текстовых и графических редакторов опции – создания новых многоуровневых КЦ, их сохранения в файл и открытия файла с ранее сохраненной моделью.

Редактор включает в себя три слоя, на каждом из которых пользователь формирует соответствующий уровень многоуровневой компонентной цепи с использованием нотаций определенного подязыка языка МКЦ:

– *Объектный слой* соответствует объектному уровню многоуровневой КЦ СТУС. На нем в нотациях графического языка моделирования ХТС (п. 2.4) формируется модель управляемой химико-технологической системы, компоненты которой, принадлежащие множеству K_{ChTS} (2.31), соединены неоднородными векторными связями (3.4). На ее основе интерпретатором языка ХТС формируется вычислительная модель подцепи объектного уровня МКЦ, адаптированная к анализу универсальным вычислительным ядром [103] в статическом и динамическом режимах.

– *Логический слой* соответствует логическому уровню многоуровневой КЦ СТУС. Он предоставляет пользователю возможности формирования сценариев автоматизированных экспериментов, каждый из которых представляет собой цепочку взаимосвязанных компонентов множества K_A (2.63), отображающих операторы и функции языка моделирования алгоритмических конструкций (см. главу 4). На основе имитационных КЦ сценариев, сформированных на логическом слое, интерпретатором математико-алгоритмических выражений автоматически формируется алгоритмическая КЦ (2.63), анализ которой производится имитационным ядром, работа которого основана на алгоритме передачи сообщений [116].

– *Визуальный слой редактора* соответствует визуальному уровню многоуровневой КЦ СТУС. На нем из визуальных компонентов множества K_V (2.68) формируются лицевые панели виртуальных приборов [108], а также панели визуализации и интерактивного управления СТУС. Для получения подлежащей визуализации информации (выполнения межуровневого взаимодействия (2.51)) компоненты множества $K_{V2} \subset K_V$ имеют свои отображения одновременно на визуальном и логическом слоях редактора. А множество управляющих компонентов $K_{V3} \subset K_V$ отображаются на двух слоях редактора (визуальном и логическом) для выполнения межуровневого взаимодействия (2.52).

Процесс разработки многоуровневой КЦ СТУС в многослойном редакторе включает в себя следующие действия пользователя:

1) выбор определенного компонента из дерева компонентов (1 на рисунке 6.2) и его установку в соответствующее место определенного слоя рабочего поля редактора (2 на рисунке 6.2), в котором производится формирование многоуровневой КЦ в графическом виде;

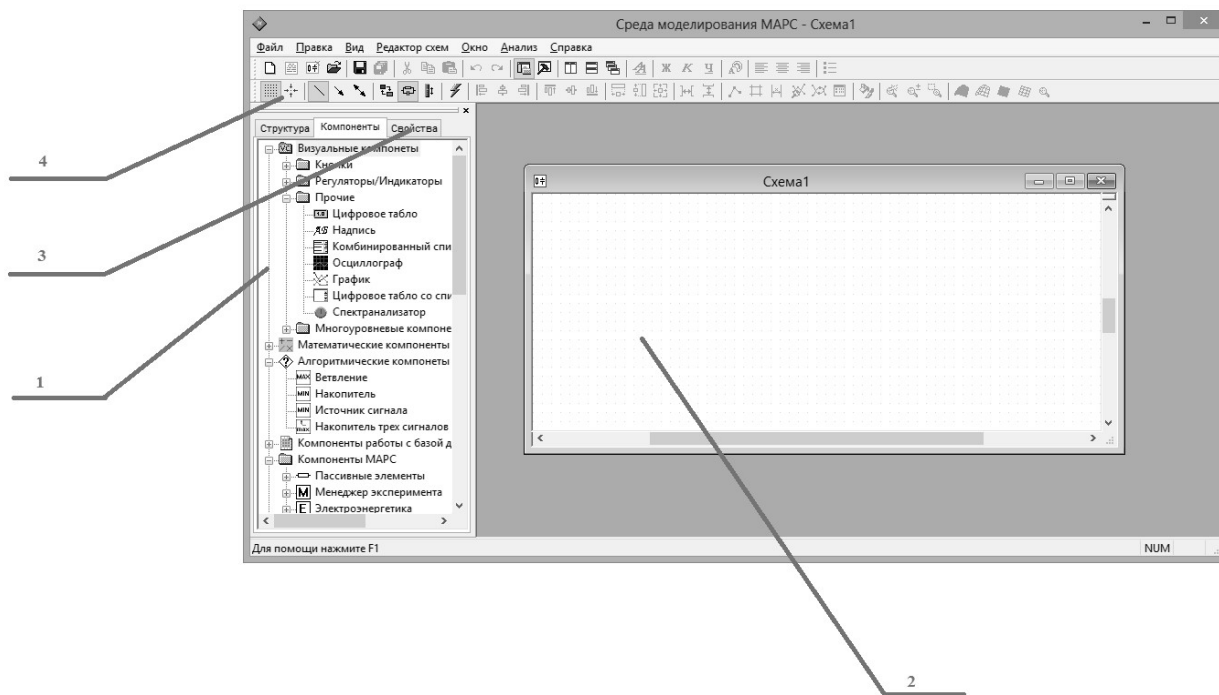






Рисунок 6.2 – Главное окно комплекса программ моделирования химико-технологических систем

2) соединение компонентов, которое производится путем установки связей между их соответствующими узлами или присоединением узла некоторого компонента к существующей связи;

3) параметризация компонентов, которая выполняется с помощью вкладки «Свойства» на панели вкладок (3 на рисунке 6.2). Каждому свойству отводится своя строка таблицы. В её левой колонке указывается наименование свойства, а в правой – значение;

4) переключение между слоями многослойного редактора, которое производится с помощью расположенных на панели инструментов (4 на рисунке 6.2)

кнопок:  – объектный;  – логический;  – визуальный.

После того, как модель сформирована на всех слоях многослойного редактора, ее анализ можно запустить, нажав кнопку  на панели инструментов (4 на рисунке 6.2) или воспользовавшись пунктом «Запуск» меню «Анализ».

Во время анализа многоуровневой КЦ СТУС пользователь может наблюдать результаты анализа модели ХТС и значения наблюдаемых характеристик реальной системы, а также имеет возможность варьирования значений определенных параметров модели и управляющих воздействий.

Таким образом, в многослойном редакторе многоуровневой КЦ СТУС над компонентами, которые могут располагаться на всех его слоях, выполняются следующие опции:

- предоставление информации редактору для формирования дерева компонентов, из которого по указанию пользователя будут устанавливаться определенные компоненты в рабочую область многослойного редактора;

- установка компонента в указанное пользователем место и его прорисовка на предписанных слоях редактора. Компоненты, принадлежащие визуальному или объектному уровням многоуровневой КЦ СТУС, могут иметь свои отображения на визуальном или объектном слоях редактора, а также на его логическом слое;

- возможность изменения размера изображения компонента на соответствующем слое редактора;

- инициализация свойств и параметров компонента;

- поворот изображения компонента на определенном слое редактора на 90^0 по и против часовой стрелки;

- соединение изображений компонентов на логическом и объектном слоях редактора;

- параметризация компонента (изменение его свойств) до и во время выполнения автоматизированного эксперимента;

- инициализация определенных свойств компонента в начале эксперимента и обнуление его определенных свойств в конце эксперимента;

– отображение компонентов на визуальном слое редактора и возможность манипулирования с визуальными образами компонентов-регуляторов во время работы модели;

– сохранение текущего положения компонентов и текущих значений его параметров.

Для выполнения этих опций каждый компонент содержит в себе:

1) *информацию для установления в дерево компонентов*, которая включает в себя имя компонента; название папки дерева, в которую должен быть установлен компонент; уникальный идентификационный номер компонента; иконку, с которой он будет отображаться в дереве компонентов;

2) *сведения о связях компонентов*, которые для каждой связи включают ее ориентацию относительно компонента, координаты, её название, тип и порядковый номер связи в ЛКБ компонента;

3) *информацию об изображениях компонента* на определенных слоях редактора. В случае статического (неизменяемого) изображения оно должно быть прописано один раз. Для визуальных компонентов, предназначенных для визуализации данных измерения и результатов моделирования, изображение должно меняться в зависимости от поступающих данных или действий пользователя на визуальном слое редактора, оно должно быть динамическим. Такие компоненты должны обладать функцией перерисовки компонента;

4) *сведения о параметрах компонента*, доступных для варьирования пользователю из редактора параметров компонента, приведенного на рисунке 6.3. Для каждого параметра указывается его имя, тип и значение по умолчанию;

5) *информацию о входных и выходных редакторах*, которые должны открываться у некоторых компонентов по двойному щелчку мыши по их изображению

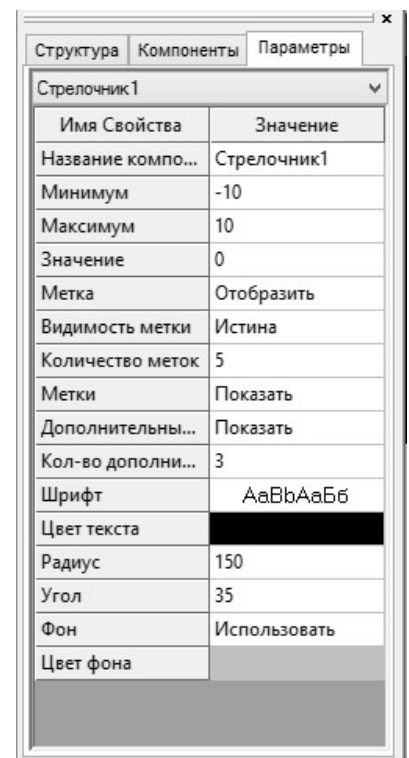


Рисунок 6.3 – Редактор параметров компонента

на определённом слое редактора.

Сформированную в многослойном редакторе многоуровневую КЦ СТУС с формальной точки зрения можно представить в виде

$$C = (K, Q), \quad (6.1)$$

где K – множество компонентов; Q – множество связей между ними.

Наиболее близкой к языку МКЦ (см. пункт 2.3.3) формой описания информационных систем, к классу которых относится проектируемый комплекс программ «Среда моделирования химико-технологических систем», является объектно-ориентированное программирование (ООП) [192]. Для описания информационных систем и комплексов программ, разработанных по технологии ООП, наиболее подходящим языком является визуальный язык представления информационных систем UML [148, 149] и входящие в его состав диаграммы. Далее его нотации будут использованы для описания классов и алгоритмов, лежащих в основе реализации спроектированного и разработанного комплекса программ «Среда моделирования химико-технологических систем».

Основным интерфейсным классом, от которого наследуются все компоненты, имеющие отображение на любом слое многослойного редактора, является интерфейсный класс *CComponent*, структура которого с основными необходимыми полями представлена на рисунке 6.4. Полное описание класса с пояснением полей и реализованных в нем методов приведено в приложении 8.

CComponent
+PinsMassiv : CPin
+CountPins : int
+ParametersMassiv : SAttribute
+CountParameters : int

Рисунок 6.4 – Класс компонента *CComponent*

Входной информацией для интерпретатора языка ХТС и интерпретатора языка МАК является совокупность компонентов C и совокупность всех связей компонентов Q (6.1). Для описания алгоритмов работы перечисленных интерпретаторов интерес представляют следующие поля приведенного на рисунке 6.4 класса компонента *CComponent*:

– *PinsMassiv* – массив выводов компонента размера *CountPins*, с помощью которых образуются связи с другими компонентами на определенном слое редактора многоуровневых КЦ. Этот массив регламентирует порядок выводов,

задавая каждому свой порядковый номер в ЛКБ и геометрическое место в редакторе относительно графического изображения компонента;

- *ParametersMassiv* – массив параметров компонента, значения которых не зависят от результатов моделирования и входят в состав математических выражений, предназначенных для расчета значений коэффициентов при переменных в уравнениях. Размер данного массива задается полем *CountParameters*.

Каждое соединение, образованное между двумя компонентами, входит во множество Q (6.1) и представляет собой экземпляр класса *CWire*, приведенного на рисунке 6.5. Он включает в себя следующие поля:

- *pComponentFirst* – первый компонент, входящий в соединение и являющийся экземпляром класса *CComponent*;

- *nPinFirst* – номер вывода в ЛКБ компонента *pComponentFirst*, с помощью которого данный компонент входит в рассматриваемое соединение. Он также определяет порядковый номер этого узла в массиве *PinsMassiv* класса *CComponent*;

- *pComponentSecond* – второй компонент, входящий в рассматриваемое соединение;

- *nPinSecond* – номер вывода в ЛКБ компонента *pComponentSecond*, определяющий данный узел в его массиве *PinsMassiv*.

Совокупность компонентов, каждый из которых реализуется в определённом классе, наследуемым от представленного на рисунке 6.4 класса *CComponent*, и совокупность всех связей между компонентами, каждая из которых является экземпляром представленного на рисунке 6.5 класса *CWire*, являются входной информацией интерпретатора языка моделирования ХТС и интерпретатора языка МАК. Интерпретатор языка ХТС формирует компонентную цепь объекта, обладающего неоднородными векторными связями, для последующего её анализа универсальным вычислительным ядром [103]. Задачей интерпретатора языка

<i>CWire</i>
+pComponentFirst : CComponent
+nPinFirst : int
+pComponentSecond : CComponent
+nPinSecond : int

Рисунок 6.5 – Класс соединения между компонентами *CWire*

МАК является построение алгоритмической КЦ вида (2.63) на основе алгоритмических подцепей МКЦ, представленных на её логическом уровне.

6.3. Интерпретатор языка моделирования ХТС

6.3.1 Структура входных и выходных данных интерпретатора языка ХТС

Основной задачей интерпретатора языка ХТС является построение вычислительной модели КЦ объектного уровня МКЦ, обладающей НВС (3.4). Компоненты химико-технологических систем, представленные на рисунках 2.6–2.9 и описывающие определенные элементы и аппараты химической технологии, реализованы в классах, наследуемых от интерфейсного класса *IBaseChTSMModel*, представленного на рисунке 6.6. Его полями являются:

– *ConnectionMassiv* – массив связей компонента, каждая из которых является объектом класса *CVectorConnection*. Его размер равен размеру массива *PinsMassiv* класса *CComponent* и задается полем *CountConnection*.

IBaseChTSMModel
+ConnectionMassiv : CVectorConnection
+CountConnection : long

Рисунок 6.6 – Класс компонентов с неоднородными векторными связями

Каждая связь компонента, являясь НВС (3.4), входит в массив *ConnectionMassiv* класса *IBaseChTSMModel* в виде экземпляра класса *CVectorConnection*, представленного на рисунке 6.7. Его полями являются:

– *EnergyMassiv* – массив энергетических связей (3.6), каждая из которых является экземпляром класса *CEnergyConnection*;

CVectorConnection
+EnergyMassiv : CEnergyConnection
+EnergySize : int
+InformationMassiv : CInformationConnection
+InformationSize : int

Рисунок 6.7 – Класс векторной связи *CVectorConnection*

– *EnergySize* – количество энергетических связей (3.6), входящих в рассматриваемую НВС;

– *InformationMassiv* – массив информационных связей (3.8), включенных в рассматриваемую векторную связь;

– *InformationSize* – количество информационных связей в рассматриваемой НВС.

Каждая энергетическая связь (3.6), которой ставится в соответствие пара топологических координат – узел n_j и ветвь b_j – с потенциальной V_{n_j} и потоковой V_{b_j} переменными, выражается объектом класса *CEnergyConnection*, представленного на рисунке 6.8. Его полями являются:

CEnergyConnection
+PhysicAspect : CEnergyPhysicAspect
+NameNVariable : char
+NumberNVariable : int
+NameBVariable : char
+NumberBVariable : int

Рисунок 6.8 – Класс энергетической связи *CEnergyConnection*

– *PhysicAspect* – энергетический субаспект физического аспекта, представляющий собой совокупность предикатов (2.7);

– *NameNVariable* – имя потенциальной переменной, которым она выражена в уравнениях модели, записанных в редакторе математических выражений;

– *NumberNVariable* – номер потенциальной переменной к ГКБ, которым она выражена в формируемой вычислительной модели КЦ ХТС;

– *NameBVariable* – имя потоковой переменной, выражающее ее в уравнениях математической модели, составленных в физическом КБ компонента;

– *NumberBVariable* – номер потоковой переменной в ГКБ формируемой КЦ.

CInformationConnection
+PhysicAspect : CInformationPhysicAspect
+NameNVariable : char
+NumberNVariable : int

Рисунок 6.9 – Класс информационной связи *CInformationConnection*

Каждая информационная связь (3.8) с единственной потенциальной переменной выражается в виде объекта класса *CInformationConnection*, который представлен на рисунке 6.9. Он включает в себя следующие поля:

– *PhysicAspect* – информационный субаспект физического аспекта, определяющий совокупность предикатов связи (2.8);

– *NameNVariable* – имя потенциальной переменной, выражающее ее в математической модели, составленной в ФКБ компонента;

– *NumberNVariable* – номер потенциальной переменной, заданный в ГКБ формируемой КЦ и выражающий в ней данную переменную в вычислительной модели, формируемой универсальным вычислительным ядром.

Для взаимодействия с универсальным вычислительным ядром компоненты языка ХТС помимо интерфейсного класса *IBaseChTSMModel*, описывающего компоненты с неоднородными векторными связями, должны наследоваться от интерфейсного класса *IMarsInterface*, реализующего методы взаимодействия компонентов с универсальным вычислительным ядром. Интерфейсный класс *IMarsInterface* рассмотрен далее.

Входной информацией алгоритма формирования КЦ исследуемой ХТС, реализованного в рамках интерпретатора языка химико-технологических систем, является массив связей, являющихся экземплярами класса *CWire*. В нём проинициализированы номера всех потенциальных и потоковых переменных, то есть им присвоены значения -1 в ГKB формируемой цепи. Так как нумерация переменных начинается с номера 0, то первоначально для работы алгоритма необходимо каким-либо образом обозначить непрономерованные переменные. Для этих целей было использовано значение -1. Сначала нумеруются все потенциальные переменные КЦ с номерами от 0 до значения $CountNV-1$, а затем потоковые переменные – начиная с номера $CountNV$ и заканчивая номерами $CountNV+CountNB-1$ (где $CountNV$ – количество потенциальных, $CountNB$ – количество потоковых переменных в формируемой КЦ).

CSchema
+ComponentMassiv : IBaseChTSMModel
+CountComponent : int
+CountPotenzVariables : int
+CountPotokVariables : int

Рисунок 6.10 – Класс схемы *CSchema*

Результатом работы интерпретатора языка ХТС является компонентная цепь химико-технологической система, являющая объектом класса *CSchema*, представленного на рисунке 6.10 и включающего в себя следующие поля:

- *ComponentMassiv* – массив компонентов – объектов класса *IBaseChTSMModel*, в каждом из которых реализованы функции предоставления топологической информации о связях компонентов и формирования их вычислительных моделей;
- *CountComponent* – количество компонентов, находящихся в данный момент в массиве *ComponentMassiv*;

- *CountPotenzVariables* – счетчик количества потенциальных переменных связей всех компонентов, входящих в формируемую КЦ;
- *CountPotokVariables* – счетчик количества потоковых переменных связей всех компонентов, входящих в рассматриваемую КЦ исследуемого объекта.

6.3.2 Алгоритм формирования компонентной цепи исследуемого объекта с неоднородными векторными связями

Входной информацией алгоритма формирования КЦ исследуемого объекта с НВС является массив связей всех компонентов Q (6.1), соединенных в редакторе между собой связями. Каждая связь представляет собой экземпляр класса *CWire*, представленного на рисунке 6.5. Потенциальные и потоковые переменные всех компонентов должны быть проинициализированы, то есть полям *NumberNVariable* и *NumberBVariable* всех экземпляров классов *CEnergyConnection*, представленного на рисунке 6.8, и *CInformationConnection*, представленного на рисунке 6.9, всех компонентов должны быть присвоены значения -1 .

Счетчики количества компонентов *CountComponent*, потенциальных *CountPotenzVariables* и потоковых *CountPotokVariables* переменных формируемой ВМ *pNewCircuit* класса *CSchema*, представленного на рисунке 6.10, должны иметь значения 0.

Процесс формирования ВМ *pNewCircuit* производится в цикле, в котором для каждого соединения *pWire[j]* класса *CWire* между узлом *nPinFirst* компонента *pComponentFirst* и узлом *pPinSecond* компонента *pComponentSecond* выполняется алгоритм, состоящий из следующих шагов:

Шаг 1. Проверка совпадения размерностей векторных связей соединенных компонентов. Между векторной связью *ConnectionMassiv[nPin1]* компонента *pComponentFirst* и связью *ConnectionMassiv[nPin2]* компонента *pComponentSecond* осуществляется проверка условий (3.13) и (3.14), то есть проверка равенства количества энергетических *EnergySize* и информационных *InformationSize* связей, входящих в соединенные в одном узле НВС компонентов.

Шаг 2. Проверка соответствия субаспектов физического аспекта соответствующих энергетических связей, входящих в векторные связи. На основе выражения (3.15) осуществляется проверка совпадения энергетических субаспектов физических аспектов *PhysicAspect* между каждой энергетической связью *EnergyMassiv[I]*, входящей в НВС *ConnectionMassiv[nPin1]*, принадлежащей компоненту *pComponentFirst* соединения *pWire[j]*, и соответствующей энергетической связью *EnergyMassiv[I]* векторной связи *ConnectionMassiv[nPin2]* компонента *pComponentSecond*, для всех $I=1..EnergySize$. В случае несовпадения физических аспектов на экран должно выводиться соответствующее информационное сообщение об ошибке соединения компонентов и работа алгоритма должна быть прервана.

Шаг 3. Проверка соответствия информационных субаспектов физических аспектов информационных связей, входящих в соединяемые векторные связи. На основе выражения (3.16) проверяется совпадение информационных субаспектов физического аспекта *PhysicAspect* между каждой информационной связью *InformationMassiv[I]*, входящей в векторную связь *ConnectionMassiv[nPin1]* компонента *pComponentFirst*, и соответствующей ей информационной связи *InformationMassiv[nPin2]* компонента *pComponentSecond*. Данная проверка осуществляется для всех $j=1..InformationSize$. В случае несовпадения хотя бы одной пары информационных субаспектов производится вывод информационного сообщения об ошибке и прекращение работы алгоритма.

Шаг 4. Нумерация потенциальных переменных векторных связей компонентов. При правильном соединении двух связей компонентов оно содержит *EnergySize* энергетически и *InformationSize* информационных связей. Такому соединению будет соответствовать $EnergySize+InformationSize$ потенциальных переменных. Если потенциальные переменные обеих связей, вступающих в соединение, равны -1 (то есть они еще не пронумерованы), то последовательно все $I=1..EnergySize+InformationSize$ переменных получают значения от *CountPotenzVariables* до $CountPotenzVariables+EnergySize+InformationSize-1$, где *CountPotenzVariables* – количество ранее пронумерованных потенциальных пере-

менных компонентов ВМ, находящихся в массиве *ComponentMassiv* объекта *pNewCircuit* класса *CSchema*. после этого значение *CountPotenzVariables* увеличивается на количество пронумерованных в данном шаге переменных.

Если все потенциальные переменные какой-либо связи одного компонента уже пронумерованы, то переменные связи второго компонента получают номера соответствующих потенциальных переменных связи первого компонента.

Если потенциальные переменные обеих векторных связей в рассматриваемом соединении пронумерованы, они будут иметь различные номера. Тогда потенциальной переменной с большим номером N одной связи присваивается меньший номер M соответствующей переменной второй связи. Также производится поиск потенциальных переменных с номером N в связях других компонентов, добавленных ранее в массив *ComponentMassiv*, и их замена на номер M . В связи с удалением одной потенциальной переменной производится уменьшение общего количества потенциальных переменных *CountPotenzVariable* на единицу, а также уменьшение на единицу номеров потенциальных переменных, превышающих число N . В результате данной операции общее количество переменных формируемой ВМ КЦ должно уменьшиться на количество потенциальных переменных рассматриваемой связи, то есть на число *EnergySize+InformationSize*.

В результате такой нумерации потенциальные переменные векторных связей всех компонентов будут иметь номера от 0 до значения *CountPotenzVariable-1*.

Шаг 5. Нумерация потоковых переменных связей. Производится нумерация потоковых переменных вступающих в соединение векторной связи *ConnectionMassiv[nPin1]* компонента *pComponentFirst* и векторной связи *ConnectionMassiv[nPin2]* компонента *pComponentSecond*. Каждая потоковая переменная всех энергетических связей *CEnergyConnection*, имеющая номер *NumberBVariable*, равный -1 , получает свой порядковый номер *CountPotokVariable*. После этих действий общее число потоковых переменных *CountPotokVariable* объекта *pNewCircuit* класса *CSchema* увеличивается на единицу.

Шаг 6. Добавление компонента в массив. После того, как все потенциальные и потоковые переменные всех связей компонента, входящих в массив

ConnectionMassiv, инициализированы, то есть получили свои порядковые номера, данный компонент может быть добавлен в массив компонентов *ComponentMassiv* формируемой компонентной цепи *pNewCircuit* класса *CSchema*. При его добавлении увеличивается количество находящихся в массиве компонентов *CountComponent*.

После того, как ко всем связям компонентов класса *CWire* были применены приведенные шаги алгоритма, полученную ВМ *pNewCircuit* можно считать сформированной и соответствующей построенной пользователем компонентной цепи ХТС на объектном слое многослойного редактора МКЦ СТУС.

Проверка правильности формирования вычислительной модели КЦ *pNewCircuit* производится с целью выявления неподключенных связей входящих в модели компонентов. Для этого у всех векторных связей класса *CVectorConnection*, входящих в массив связей *ConnectionMassiv* каждого компонента, наследуемого от интерфейсного класса *IBaseChTSMModel*, осуществляется проверка номеров всех потенциальных *NumberNVariable* и потоковых *NumberBVariable* переменных. Равенство значению -1 этих номеров означает, что данная связь компонента не была подключена ни к какой связи другого компонента. При этом должен производиться вывод сообщения об ошибке на экран и прекращение работы интерпретатора.

Если при проверке правильности формирования КЦ не выявлено переменных, номера которых равны -1 , сформированная цепь передается универсальному вычислительному ядру [103], где производится ее анализ согласно заданному режиму с соответствующими ему параметрами.

6.4. Универсальное вычислительное ядро

При выполнении различных видов анализа КЦ технических (технологических) объектов в рамках комплекса программ моделирования ХТС используется универсальное вычислительное ядро [103], которое положено в основу разработки расчетно-моделирующей среды [104, 193, 194]. Она предназначена для полного комплексного исследования сложных технических устройств и систем учебного и научно-исследовательского назначения. Для этих целей она включает в себя

предназначенную для исследования СТУС среду моделирования MAPC [20] и систему компьютерной математики «Макрокалькулятор» [30], позволяющую автоматизировать вычисления математических выражений любой сложности. Спроектированное и реализованное в рамках расчетно-моделирующей среды [104] универсальное вычислительное ядро удовлетворяет следующим условиям:

1. Универсальное вычислительное ядро представляет собой самостоятельный модуль в виде динамически загружаемой библиотеки (dll) и используется в рамках среды моделирования ХТС для автоматизации статического и динамического одновариантного анализа КЦ ХТС, являющихся подцепями объектного уровня МКЦ, а также для вычисления математических выражений в рамках интерактивных математических панелей объектного и логического уровней многоуровневой КЦ.

2. Оно позволяет производить расчет КЦ математических выражений наряду с анализом КЦ ХТС, используя один и тот же программно-алгоритмический аппарат.

3. Универсальное вычислительное ядро открыто к его пополнению новыми методами анализа и расчета.

4. Оно имеет модульную структуру, спроектированную согласно представленной на рисунке 2.4 схеме ВЭ. Это позволяет легко и быстро исправлять при необходимости существующие методики анализа и добавлять новые, направленные на повышение эффективности работы универсального вычислительного ядра.

5. Отсутствуют ограничения на размер модели исследуемого ХТС по числу компонентов, числу узлов, для которых формируются топологические уравнения (2.18), а также по числу и размерности формируемых вычислительных моделей компонентов.

Выделение основных этапов выполнения ВЭ, алгоритм которого представлен на рисунке 2.4, ряд из которых допускают вариативность методов, позволяют сформировать следующую объектно-ориентированную структуру универсального вычислительного ядра.

6.4.1 Объектно-ориентированная структура универсального вычислительного ядра

Одним из наиболее оптимальных средств разработки универсального вычислительного ядра, имеющего объектно-ориентированную структуру и реализующего рассмотренный в пункте 2.1.5 алгоритм ВЭ, является язык программирования высокого уровня *Visual C++* [187, 195], основанный на ООП [148]. Он позволяет:

- реализовывать отдельные модули в виде динамически загружаемых библиотек и использовать их различными приложениями путем вызовов экспортируемых функций;
- использовать между всеми модулями единое адресное пространство рабочей памяти с расположенными в ней различными динамическими структурами;
- использовать один и тот же модуль универсального вычислительного ядра различными приложениями, реализованными на основе метода КЦ [17] и его различных расширений.

Для задействования в ВЭ, реализованном в универсальном вычислительном

IMarsComponent
+Nodes : int
+Branch : int
+Param : double
+CountNodes : int
+CountBranch : int
+CountParam : int
+CountEquation : int
+CountMembers : int
+fChangeOfParam : bool
+TopologQuery() : bool
+CalculatQuery() : bool
+CalculatQuery1() : bool
+CalculatQuery2() : bool
+MeasureQuery() : bool
+AbsoluteTermQuery() : bool
+SetVectorConnections() : void

Рисунок 6.11 – Класс
вычислительных
компонентов

ядре, каждый компонент объектного уровня, включая компоненты ХТС наследуемые от *IBaseChTSMModel*, представленного на рисунке 6.6, должен наследоваться от интерфейсного класса *IMarsComponent*, структура которого приведена на рисунке 6.11. Его полями, задействованными при формировании топологических и компонентных уравнений на этапах опросов топологического и вычислительного блоков компонентов (смотри пункт 2.1.2), являются:

- *Nodes* – массив номеров потенциальных переменных, заданных в ГКБ формируемой цепи. Порядковый номер каждого члена данного массива соответствует номеру потенциальной переменной в ЛКБ рассматриваемого компонента;

– *Branch* – массив номеров потоковых переменных в ГKB формируемой КЦ, расположенных в порядке, заданном ЛКБ компонента. Массивы *Nodes* и *Branch* формируются на этапе формирования КЦ исследуемого объекта интерпретатором языка моделирования ХТС;

– *Param* – массив параметров компонента, задействованных в формировании его вычислительной модели;

– *CountNodes* – количество потенциальных переменных компонента, определяющее размер массива *Nodes*;

– *CountBranch* – количество потоковых переменных компонента, задающее размер массива *Branch*;

– *CountParam* – количество параметров компонента, соответствующее размеру массива *Param*;

– *CountEquation* – количество компонентных уравнений, формируемых данным компонентом и добавляемых в общую систему уравнений анализируемой КЦ;

– *CountMembers* – количество слагаемых во всех компонентных уравнениях рассматриваемого компонента. Данное поле наряду с полем *CountEquation* используется для расчета размера формируемой ВМ и инициализации соответствующей массивов системы уравнений.

Методами класса *IMarsComponent*, которые должны быть перегружены во всех наследуемых от него классах, реализующих определенные компоненты ХТС, являются:

– *SetLKBVariables()* – формирование ЛКБ компонента на основе его физического КБ, то есть присвоение каждой потенциальной и потоковой переменной компонента определенного порядкового номера в рамках переменных компонента. При этом также производится подсчет количества потенциальных *CountNodes* и потоковых переменных *CountBranch*, которые будут являться размерами массивов *Nodes* и *Branch* соответственно;

– *TopologQuery()* – опрос топологического блока компонента, на котором каждый компонент о каждой своей энергетической связи, входящей в состав НВС (3.4), формирует пару значений вида

$$(\pm B_j, N_j), \quad (6.2)$$

где B_j – номер потоковой переменной, заданный ГКБ цепи; N_j – номер соответствующей ей потенциальной переменной в ГКБ цепи;

– *CalculatQuery()* – метод опроса вычислительного блока компонентов, направленный на формирование компонентных уравнений рассматриваемого объекта или системы (2.9)-(2.12) и их добавление в формируемую систему уравнений анализируемой КЦ ХТС;

– *MeasureQuery()* – метод опроса измерительного блока компонентов, заключающийся в выделении значений переменных, отмеченных измерительными компонентами, из общего вектора решения (2.15) и в их передаче на логический уровень многоуровневой КЦ СТУС для последующей обработки и визуализации.

Такая реализация моделей компонентов основных элементов и функциональных блоков ХТС позволяет унифицировать доступ к компонентам за счет единообразия заголовков всех методов опроса соответствующих блоков компонентов. Наследуемые от интерфейсного класса *IMarsComponent* классы компонентов образуют библиотеку моделей компонентов ХТС. В приложении 9 приведен пример модели абсорбера с реализацией всех перечисленных методов.

Компонентная цепь ХТС, как объекта анализа, в универсальном вычислительном ядре представлена объектом класса *CCircuit*, приведенном на рисунке 6.12 и содержащем следующие поля:

CCircuit
+ComponentList : IMarsInterface
+CountNodes : int
+CountBranch : int
+TopologQuery() : bool
+CalculatQuery() : bool
+MeasureQuery() : bool

Рисунок 6.12 – Класс компонентной цепи

– *ComponentList* – список компонентов, входящих в исследуемую КЦ ХТС;

– *CountNodes* – количество потенциальных переменных исследуемой КЦ, соответствующее количеству его энергетических и информационных узлов, входящих в состав векторных узлов;

– *CountBranch* – количество потоковых переменных цепи, соответствующее количеству ее энергетических связей всех НВС.

Для формирования ВМ КЦ, имеющей векторно-матричное представление вида (2.24), класс *CCircuit* включает в себя методы осуществляющие опросы соответствующих блоков компонентов, входящих в список *ComponentList*.

Входной информацией ВЭ, реализуемого в рамках универсального вычислительного ядра, является КЦ ХТС, в каждом компоненте которой пронумерованы потенциальные и потоковые переменные.

SDirectParam
+ICC : int
+PrintSystemEq : int
+PerIter : bool
+StopExperiment : bool

Рисунок 6.13 –
Структура параметров
настройки
вычислительного ядра
уравнений;

Настройка ядра на определенный вид ВЭ формируется в виде следующих полей представленной на рисунке 6.13 структуры *CDirectParam*:

– *ICC* – слово состояния ядра), определяющее режим анализа КЦ, метод алгебраизации дифференциальных уравнений, а также метод расчета полученной системы линейных

уравнений;
– *PrintSystemEq* – режим вывода линеаризованной системы уравнений (0 – система уравнений не выводится; 1 – выводится на первом шаге линеаризации; *n* – выводится на *n* первых шагах линеаризации);

– *StopExperiment* – флаг остановки эксперимента по требованию пользователя;

– *PerIter* – флаг повтора формирования и решения системы уравнений на данном шаге во времени или частоте при изменении положения любого коммутационного компонента (ключа, реле времени и т.п.).

Структура *CDirectParam* представляет собой набор данных, формирующийся до начала ВЭ (одновариантного анализа) и содержащий статические настройки ядра, использующийся другими классами при проведении эксперимента.

Информация о временных (частотных) интервалах, шагах измерения времени (частоты), являющейся независимым общим для всех компонентов исследуемой КЦ параметром, задается до начала ВЭ и формируется в виде структуры данных *SMarsTime*, представленной на рисунке 6.14. Её полями являются:

SMarsTime
+T : double
+TMin : double
+TMax : double
+H : double
+HMin : double
+HMax : double
+EpsVn : double
+EpsVb : double

Рисунок 6.14 –
Структура
данных
SMarsTime

– T – текущее время (частота) моделирования, номер итерации для анализа нелинейных КЦ ХТС в статическом режиме;

– $TMin$ – начальное (минимальное) время (частота) моделирования;

– $TMax$ – конечное (максимальное) время (частота) моделирования;

– $HMin$ – минимальный шаг изменения времени (частоты);

– $HMax$ – максимальный шаг измерения времени (частоты);

– H – текущий шаг изменения времени (частоты). Если

$HMin=HMax$, то моделирование производится с постоянным шагом, в противном случае применяется схема моделирования с переменным шагом при учете заданной точности;

– $EpsN$ – точность решения для всех потенциальных переменных модели V_{Ni} , где $i = \overline{0, CountNodes - 1}$;

– $EpsB$ – точность решения для потоковых переменных модели V_{Bj} для всех $j = \overline{1, CountBranch}$.

Исходной информацией к ВЭ, проводимому алгоритмами вычислительного ядра, являются:

– сформированная интерпретатором языка ХТС, описанным в пункте 6.3, компонентная цепь исследуемой химико-технологической системы с неоднородными векторными связями (3.4), которая является экземпляром класса *CCircuit*;

– параметры настройки вычислительного ядра, представленные в виде структуры *SDirectParam*;

– настройки времени (частоты) выполняемого режима анализа, находящиеся в структуре *SMarsTime*.

Процесс анализа КЦ ХТС заключается в автоматическом формировании общей системы алгебро-дифференциальных уравнений (2.24), линеаризации нелинейных (2.10), алгебраизации дифференциальных (2.11) уравнений и решении полученной системы линейных алгебраических уравнений (2.25). Представленная

в плотном виде без переменных с нулевыми коэффициентами, она формируется в виде структуры *SSystemEquations*, представленной на рисунке 6.15. В нее входят следующие массивы данных и скалярные поля:

<i>SSystemEquations</i>
+m1 : int
+m2 : int
+am3 : double
+IB : int
+KL : int
+W : double
+j : int
+j1 : int
+V : double

Рисунок 6.15 –
Структура системы
уравнений
SSystemEquations

– $m1$ – топологический массив, формирующийся при опросе топологического блока всех компонентов, входящих в КЦ ХТС;

– $m2$ – массив номеров переменных, входящих в формируемые уравнения с ненулевыми коэффициентами. В данный массив заносятся номера потенциальных и потоковых переменных в ГKB анализируемой КЦ ХТС;

– $am3$ – массив ненулевых коэффициентов уравнений системы алгебро-дифференциальных уравнений (2.24), который после процессов линеаризации и алгебраизации будет содержать ненулевые коэффициенты матрицы Φ из (2.25);

– IB – указательный целочисленный массив, включающий количества неизвестных членов с ненулевыми коэффициентами в каждом уравнении системы (2.24);

– KL – массив указателей классов уравнений, которые будут рассмотрены в пункте 6.4.2;

– W – массив свободных членов уравнений системы (2.24);

– j – указатель переменных (всегда указывает на первый свободный элемент в массивах $m2$ и $am3$);

– $j1$ – указатель уравнений (указывает на первый свободный член в массивах W , KL и IB);

– V – вектор решения компонентной цепи (2.15).

Система уравнений, представленная в виде структуры данных *CSystemEquations*, формируется в плотном виде (без нулевых элементов). В случае, если переменная, помещаемая в массив $m2$, находится под знаком дифференцирования, что характерно для дифференциальных уравнений (2.11), то к ее номеру прибавляется соответствующий признак.

Формирование топологических и компонентных уравнений ВМ КЦ ХТС осуществляется путем опросов соответствующих блоков входящих в цепь компо-

CFormirEquations
+InitEquations() : SSystemEquations
+DeleteEquations()
+SetSystemEquations() : SSystemEquations
+SetTopologEquations()
+SetComponentEquation()
+ExplEulerMethod()
+ImplEulerMethod()
+TrapeziumMethod()
+PrintSystemEquations()

Рисунок 6.16 – Класс формирования системы уравнений

нентов. Алгебраизация дифференциальных уравнений (2.11) осуществляется по одной из схем (2.27)–(2.29), заданной пользователем. Для формирования системы уравнений в рамках универсального вычислительного ядра реализован класс *CFormirEquations*, представленный на рисунке 6.16. Его методами являются:

– *InitEquation()* – метод инициализации структуры *SSystemEquations*, в котором рассчитываются размеры и выделяется память под все её динамические массивы;

– *DeleteEquation()* – метод освобождения памяти, выделенной под массивы структуры *SSystemEquations*;

– *SetSystemEquations()* – основной метод формирования системы уравнений, линейаризации нелинейных (2.10) и алгебраизации дифференциальных уравнений (2.11) формируемой системы;

– *SetTopologEquation()* – метод формирования топологических уравнений (2.18) на основе информации, полученной в результате опроса топологического блока всех компонентов анализируемой КЦ ХТС;

– *SetComponentEquation()* – метод формирования компонентных уравнений путем опроса вычислительного блока всех компонентов, входящих в КЦ ХТС;

– *ExplEulerMethod()* – метод алгебраизации дифференциальных уравнений (2.11) по явной схеме Эйлера (2.27);

– *ImplEulerMethod()* – метод алгебраизации дифференциальных уравнений (2.11) по неявной схеме Эйлера (2.28);

– *TrapeziumMethod()* – метод алгебраизации дифференциальных уравнений (2.11) согласно схеме трапеций (2.29);

– *PrintSystemEquations()* – метод вывода сформированной системы уравнений в понятном пользователю виде с целью отладки и верификации моделей КОМПОНЕНТОВ.

CLinAlgebra
+InitMassive() +DeleteMassive() +ASLM()

Рисунок 6.17 – Класс алгоритмов линейной алгебры

Линеаризованная в результате работы метода *SetSystemEquations()* класса *CFormirEquations()* система уравнений на каждом шаге по времени (частоте) разрешается относительно вектора решения (2.15) с помощью алгоритмов линейной алгебры, реализованных в виде методов представленного на рисунке 6.17 класса *CLinAlgebra*:

- *InitMassive()* – инициализация массивов, используемых при решении систем линейных алгебраических уравнений;
- *DeleteMassive()* – освобождение памяти, занятой динамическими массивами под необходимые вектора и матрицы;
- *ASLM()* – реализация алгоритма решения систем линейных алгебраических уравнений с плотным (без нулевых элементов) представлением на основе ал-

CAnaliz
+DirPar : SDirectParam -MarsTime : SMarsTime -SystEq : SSystemEquations -FormirEq : CFormirEquations -LinAlgebra : CLinAlgebra -Circuit : CCircuit
+SelectAnaliz() +InitFreq() +AnMin1() +AnMin3() +ALINST() +ASTATI() +ADAN() +ADIN() +ADINT() +Iterazia() +FreqAnaliz()

Рисунок 6.18 – Интерфейсный класс видов анализа ХТС

горитма исключения по схеме Жордана [115]. Результаты работы алгоритма заносятся в вектор V системы уравнений, представленной структурой *SSystemEquations*.

Интерфейсным классом, включающим реализованные методы анализа КЦ ХТС с неоднородными векторными связями, является класс *CAnaliz*, структура которого приведена на рисунке 6.18. Его полями-членами являются объекты ранее введенных классов и структур:

- *DirPar* – структура типа *SDirectParam* (рисунок 6.13), включающая в себя набор параметров настройки универсального вычислительного ядра на конкретный режим анализа модели ХТС;
- *MarsTime* – структура типа *SMarsTime* (рисунок 6.14), объединяющая в себе информацию о временном (частотном) интервале анализа КЦ ХТС;

– *SystEq* – структура типа *SSystemEquations* (рисунок 6.15), хранящая в себе данные формируемой системы уравнений, представленной в плотном виде (без нулевых элементов);

– *FormirEq* – объект класса *CFormirEquations* (рисунок 6.16), включающий в себя методы формирования системы уравнений, линеаризации нелинейных и алгебраизации дифференциальных уравнений;

– *LinAlgebra* – объект класса *CLinAlgebra* (рисунок 6.17) с реализованными в нем методами решения систем линейных алгебраических уравнений;

– *Circuit* – объект класса *CCircuit* (рисунок 6.12), представляющий собой набор данных об анализируемой компонентной цепи ХТС.

Методы класса *CAnaliz* реализуют различные виды анализа КЦ ХТС:

– *SelectAnaliz()* – метод выбора режима анализа ХТС и выполнения функций инициализации основных структур данных универсального вычислительного ядра;

– *InitFreq()* – метод инициализации данных в случае выполнения различных видов частотного анализа, предусматривающих расщепление переменных связей компонентов на действительную и мнимую части (2.16);

– *AnMin1()* – реализация схемы интегрирования с переменным шагом по времени (частоте);

– *AnMin3()* – реализация схемы интегрирования с постоянным шагом по времени (частоте);

– *ALINST()* – реализация режима анализа линейных статических объектов с постоянными входными воздействиями;

– *ASTATI()* – реализация режима анализа нелинейных статических объектов с постоянными входными воздействиями;

– *ADAN()* – реализация режима анализа динамических объектов, в моделях которых присутствуют дифференциальные уравнения вида (2.11), подвергающиеся линеаризации согласно явной схеме Эйлера (2.27);

– *ADIN()* – реализация режима анализа динамических объектов с дифференциальными уравнениями (2.11) с их алгебраизацией согласно неявной схеме Эйлера (2.28);

– *ADINT()* – реализация режима анализа динамических объектов, согласно которой дифференциальные уравнения вида (2.11) подвергаются алгебраизации согласно неявной схеме трапеций (2.29);

– *Iterazia()* – выполнение одной итерации анализа, включающей этапы формирования, линеаризации нелинейных, алгебраизации дифференциальных уравнений и расчета системы линейных уравнений при текущем значении времени (частоты) с использованием значений вектора решений (2.15), полученных на нескольких предыдущих итерациях;

– *FreqAnaliz()* – реализация частотного анализа объектов в заданном диапазоне частот.

С целью приближения ВЭ к натурному эксперименту в рамках данного класса реализуются методы статического и динамического эксперимента, который заканчивается по требованию пользователя. Кроме стандартных средств визуализации в таком эксперименте могут быть использованы виртуальные приборы и панели визуализации результатов и интерактивного управления, состоящие из компонентов языка ВИП.

Реализация универсального вычислительного ядра с применением технологии ООП [192] делает его достаточно гибким к пополнению новыми методами алгебраизации дифференциальных уравнений и решения систем линейных алгебраических уравнений. Также она позволяет сделать открытой библиотеку моделей компонентов, содержащую в себе модели компонентов, не привязанных к ядру, но взаимодействующих с ним путем вызова определенных методов, реализованных в классах компонентов. Такой подход к реализации позволяет представить библиотеку моделей компонентов в виде набора модулей, в каждом из которых реализуется определенная группа компонентов. При этом любой модуль БМК может быть использован совместно с различными редакторами входной информации и системами визуализации результатов.

Модуль «Универсальное вычислительное ядро» зарегистрирован в Отраслевом фонде алгоритмов и программ. Свидетельство о регистрации № 4584 приведено в приложении 18.

Для адаптации универсального вычислительного ядра к анализу компонентных цепей ХТС, включающих НВС, были разработаны методики и алгоритмы повышения быстродействия анализа: методика и алгоритм явно-неявного анализа КЦ ХТС; методика формирования системы уравнений с учетом их класса; применение быстродействующих алгоритмов решения систем линейных алгебраических уравнений.

6.4.2 Методики повышения быстродействия вычислительного эксперимента

Размерность компонентных цепей, исследуемых ХТС, теоретически не ограничена. Она возрастает как за счет увеличения числа компонентов структуры, добавляемых в исследуемую модель, так и за счет ввода новых компонент потоков, выражающих вещества, входящие в многокомпонентные потоки. Для оперативного получения проектных и управленческих решений актуальным является вопрос о сокращении времени, затрачиваемом универсальным вычислительным ядром на выполнение одной итерации анализа. Это позволит уменьшить время решения задач многовариантного анализа и параметрического синтеза ХТС, основанных на многократном анализе их КЦ. Решение данного вопроса может быть найдено за счет сокращения количества операций и, как следствие, времени ВЭ, затрачиваемого на формирование и линеаризацию компонентных уравнений, а также времени расчета системы линейных алгебраических уравнений, выполняющихся циклически в представленном на рисунке 2.4 алгоритме вычислительного эксперимента.

Система уравнений, формируемая вычислительным ядром на каждом шаге ВЭ, в общем случае состоит из топологических и компонентных уравнений. Топологические уравнения вида (2.18) формируются один раз при первом вызове метода формирования системы уравнений *SetSystemEquation* класса *CFormirEquations*. Компонентные уравнения формируются при каждой итерации ВЭ на основе вектора решений V (2.15), который является членом структуры *SSystemEquations*. В [17] приведена следующая классификация компонентных уравнений (таблица 6.1).

Таблица 6.1 – Классификация компонентных уравнений

Класс уравнения	Вид модели
KL=10	Линейное уравнение (2.9) с постоянными a_i и b
KL=11	Линейное уравнение (2.9) с постоянными a_i и переменными $b(t)$
KL=12	Линейное уравнение (2.9) с переменными a_i и $b(t)$
KL=13	Нелинейное уравнение (2.10)
KL=14	Дифференциальное уравнение (2.11), ψ – постоянные, $f(V^k, t)$ – линейная без свободного члена
KL=15	Дифференциальное уравнение (2.11), ψ – постоянные, $f(V^k, t)$ – линейная с постоянным свободным членом
KL=16	Дифференциальное уравнение (2.11), ψ – постоянные, $f(V^k, t)$ – линейная с переменным свободным членом
KL=17	Дифференциальное уравнение (2.11), ψ – постоянные, $f(V^k, t)$ – нелинейная
KL=18	Дифференциальное уравнение (2.11), ψ – переменные, $f(V^k, t)$ – линейная без свободного члена
KL=19	Дифференциальное уравнение (2.11), ψ – переменные, $f(V^k, t)$ – линейная с постоянным свободным членом
KL=20	Дифференциальное уравнение (2.11), ψ – переменные, $f(V^k, t)$ – линейная с переменным свободным членом
KL=21	Дифференциальное уравнение (2.11), ψ – переменные, $f(V^k, t)$ – нелинейная

По результатам опроса вычислительных блоков всех компонентов, входящих в анализируемую КЦ, система уравнений будет иметь произвольный набор компонентных уравнений типов 10-21. Она подвергается алгоритму алгебраизации, который реализован в методе *SetSystemEquations* класса *CFormirEquations*, приведенного на рисунке 6.16. При алгебраизации дифференциальных уравнений могут появляться нулевые коэффициенты. По правилам формирования системы уравнений в плотной форме [17] требуется перемещать без изменения коэффициенты линейных и нелинейных уравнений в массивах $m2$ и $am3$ объекта *SystEq*, являющегося структурой приведенного на рисунке 6.15 типа *SSystemEquations*, влево на количество освободившихся нулевых элементов. При такой реализации универсального вычислительного ядра во время анализа моделей ХТС значительное время затрачивается на формирование системы уравнений и алгебраизацию входящих в нее дифференциальных уравнений. Для повышения эффективности анализа таких объектов предложена методика формирования системы с учетом клас-

са уравнений. А для анализа КЦ ХТС, в которых все вещественные потоки упорядоченные, применяется методика и алгоритм явно-неявного анализа.

6.4.2.1 Методика и алгоритм повышения быстродействия анализа функционирования химико-технологических систем

При рассмотрении математических моделей компонентов ХТС установлено, что алгоритм вычислительного эксперимента, представленный на рисунке 2.4 и реализованный в рамках универсального вычислительного ядра [103], вносит избыточность в систему уравнений, связанную с тем, что каждое уравнение может быть включена одна неопределенная ранее переменная. При решении такой системы уравнений приходится выполнять большое количество операций, которые можно избежать путем предварительного определения значений переменных, допускающих явное представление из уравнений математических моделей компонентов химико-технологических систем, рассмотренных в главе 3.

Вектор решения модели компонента ХТС (3.30) включает в себя переменные трех множеств

$$V^{\Phi XC} = [V_{IN}^{\Phi XC}, V_{OUT_1}^{\Phi XC}, V_{OUT_2}^{\Phi XC}], \quad (6.3)$$

где $V_{IN}^{\Phi XC}$ – набор переменных, принадлежащих входным связям компонента; $V_{OUT_1}^{\Phi XC}$ – группа переменных выходных связей, допускающих свое явное определение в моделях соответствующих компонентов; $V_{OUT_2}^{\Phi XC}$ – группа переменных, не допускающих своего явного представления в моделях компонентов.

Переменные единственной выходной связи компонента-источника многокомпонентной смеси веществ, приведенного на рисунке 3.6, относятся к множеству $V_{OUT_1}^{\Phi XC}$. Их значения могут непосредственно в компоненте быть добавленными в соответствующие элементы вектора решения цепи (2.15) без формирования и добавления уравнений (3.31) в общую формируемую систему уравнений.

Связь S_1 компонента «Гидротермодинамическое сопротивление потоку», представленного на рисунке 3.12, является входной. Все её переменные принад-

лежат множеству $V_{IN}^{\Phi XC}$ вектора $V^{\Phi XC}$ (6.3), а переменные его выходной связи S_2 могут быть явно выражены из уравнений математической модели (3.47) в виде

$$\begin{aligned} P_2 &= P_1 - \lambda \cdot \rho \cdot \frac{L}{d} \cdot \frac{G^2}{(\pi \cdot d^2)^2} \\ G_2 &= G_1 \\ T_2 &= T_1 - R_t \cdot Q_1 \\ Q_2 &= Q_1 \end{aligned} \quad (6.4)$$

На основе выражений (6.4) можно сделать вывод, что модель компонента «Гидротермодинамическое сопротивление потоку» является полностью представимой к явному виду. Следовательно, все переменные её единственной выходной связи S_2 принадлежат группе переменных $V_{OUT_1}^{\Phi XC}$ вектора $V^{\Phi XC}$ (6.3).

Модель теплообменника (3.56) одновременно может описывать реальный аппарат, в котором теплоносители движутся в одном направлении или навстречу друг другу. Её первые два уравнения составляются для каждого участка аппарата, на которые он делится при алгебраизации дифференциальных уравнений (3.53), описывающих распределенный процесс теплообмена. Такие уравнения не могут быть представлены в явном виде, так как в них одновременно присутствуют переменные, характеризующие состояние обоих теплоносителей. Каждое из остальных уравнений модели (3.56) содержит одну переменную, значение которой может быть определено путем его явного решения.

Учитывая, что вектор входных переменных приведенного на рисунке 3.20 компонента «Сепаратор» включает в себя следующие переменные

$$V_{IN} = [P_{in}, G_{in}, T_{in}, Q_{in}, C_{in}]^T, \quad (6.5)$$

каждая переменная обеих его выходных связей S_1 и S_2 может быть выражена в явном виде из соответствующего уравнения системы (3.57)

$$\begin{aligned}
P_L &= P_{In} \\
G_L &= e \cdot G_{In} \\
T_L &= T_{In} \\
Q_L &= Q_{In} \\
C_{L_i} &= \frac{C_{In_i}}{K_i \cdot e + (1 - e)} \\
P_G &= P_{In} \\
G_G &= \frac{G_{In}}{1 + e} \\
T_G &= T_{In} \\
Q_G &= Q_{In} \\
C_{G_i} &= K_i \cdot C_{L_i}
\end{aligned} \tag{6.6}$$

Таким образом, переменные обеих выходных связей компонента «Сепаратор» принадлежат группе переменных $V_{OUT_1}^{\Phi XC}$ вектора $V^{\Phi XC}$ (6.3)

Ни одна переменная двух выходных связей представленного на рисунке 3.23 компонента «Абсорбер» не может быть явно выражена из уравнений его математической модели (3.60)-(3.62). Поэтому все переменные его выходных связей S_3, S_4 входят в группу переменных $V_{OUT_2}^{\Phi XC}$. Для анализа протекающих в нем процессов требуется применение методов неявного анализа.

На основе рассмотренных математических моделей компонентов ХТС выявлено, что часть этих моделей может быть полностью ($V_{OUT_1}^{\Phi XC} \neq \emptyset, V_{OUT_2}^{\Phi XC} = \emptyset$) или частично ($V_{OUT_1}^{\Phi XC} \neq \emptyset, V_{OUT_2}^{\Phi XC} \neq \emptyset$) представлена в явном виде. Наиболее эффективным методом анализа таких моделей будет численный метод явно-неявного анализа, основанный на алгоритме формирования упакованной вычислительной модели ХТС. Он предполагает определение порядка опросов вычислительных блоков компонентов от источников к терминаторам при формировании компонентных уравнений. При этом каждое формируемое линейное или нелинейное уравнение будет проверяться на возможность непосредственного вычисления значения единственной входящей в него неопределенной переменной. Для реали-

зации алгоритма помимо вектора решения цепи (2.15) вводится вектор флагов решения

$$D = \left[\left[D_N \right] \left[D_B \right] \right]. \quad (6.7)$$

Каждый элемент вектора D (6.7) $D_j = \text{TRUE}$ ($j = CN + CB$, где CN – количество потенциальных переменных, CB – количество потоковых переменных исследуемой цепи), если соответствующий ему элемент V_j вектора V (2.15) на данной итерации вычислительного эксперимента определен на этапе формирования топологических и компонентных уравнений.

Формирование упакованной вычислительной модели ХТС реализуется в рамках универсального вычислительного ядра [103] и предполагает, что компоненты каждого уравнения представлены в виде структуры

$$Eq = \{KL, IB, m2, m3, W\}, \quad (6.8)$$

где KL – признак типа уравнения (основные для универсального вычислительного ядра типы уравнений и их признаки указаны в таблице 6.1; IB – количество переменных в уравнении, включая и их производные; $m2$ – массив номеров переменных (для обыкновенных производных к номеру переменных добавляется кодовый признак); $m3$ – массив коэффициентов; W – свободный член уравнения.

Если при формировании компонентного уравнения (6.8) в него входит только одна ранее не определенная переменная, то это уравнение автоматически разрешается относительно этой переменной с номером j согласно следующему выражению

$$V_{m2[j]} = \frac{W}{m3_j} - \sum_{\substack{i=0 \\ i \neq j}}^{IB} \frac{m3_i}{m3_i} \cdot V_{m2[i]}. \quad (6.9)$$

Алгоритм формирования упакованной вычислительной модели ХТС, в которой источниками многокомпонентных смесей веществ задаются значения потенциальных и потоковых переменных связей, состоит из следующих шагов:

Шаг 1. Формирование топологического массива $m1$ всех компонентов исследуемой цепи, содержащего топологические пары вида

$$T_j = \{\pm B_j, N_j\}, \quad (6.10)$$

где B_j – номер ветви, заданный в ГКБ рассматриваемой КЦ, N_j – номер узла, инцидентного ветви B_j , представленный в ГКБ КЦ.

Шаг 2. Сортировка компонентов по направлениям потоков на основе топологических пар (6.10). Она осуществляется по выходным топологическим парам, у которых номер ветви указан со знаком «минус». В ходе проведения сортировки компоненты, входящие в исследуемую КЦ ХТС и отображаемые на рисунке 6.19 узлами дерева, должны быть отсортированы для вызова опроса их вычислительных блоков в порядке, приведенном на рисунке 6.19. В узлах дерева установлен порядковый номер, согласно которому должны проводиться опросы вычислительных блоков компонентов.

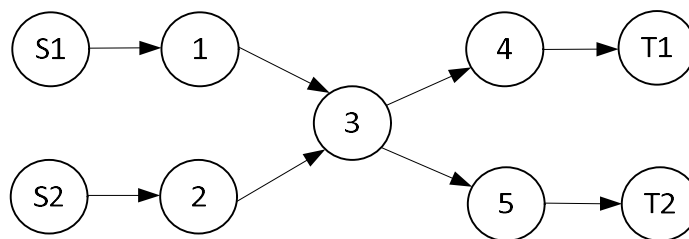


Рисунок 6.19 – Правило сортировки компонентов для вычислительного опроса

Шаг 3. Опрос вычислительного блока однополюсных компонентов-источников и терминаторов $S_1, S_2, \dots, T_1, T_2, \dots$. Так как такие компоненты формируют уравнения вида (6.8) с $IB = 1$, то на этапе преобразования уравнения из ЛКБ компонента в ГКБ компонентной цепи производится непосредственное определение значений соответствующих переменных. В этом случае их значения непосредственно заносятся в вектор решения (2.15), а соответствующие им члены вектора D (6.7) приобретают значения TRUE.

Шаг 4. Формирование топологических уравнений на основе массива топологических пар (6.10).

Шаг 5. Явное разрешение топологических уравнений, сформированных на шаге 4, по формуле (6.9).

Шаг 6. Шаг 5 выполняется до тех пор, пока на нем разрешению подлежит хотя бы одно уравнение. Разрешенное уравнение исключается из системы

уравнений. Неразрешенные уравнения остаются в системе для последующего разрешения.

Шаг 7. Осуществляется опрос вычислительного блока компонентов в установленном на рисунке 6.19 порядке с попыткой непосредственного разрешения каждого линейного и нелинейного уравнения по формуле (6.9). Для каждого дифференциального уравнения непосредственно из компонента осуществляется выход соответствующей процедуры, входящей в представленный на рисунке 6.16 класс *CFormirEquations*, с рассмотрением возможности его разрешения по формуле (6.9). Разрешенные уравнения исключаются из рассмотрения.

Шаг 8. Для оставшейся системы линейных алгебраических уравнений, если в ней есть хотя бы два уравнения, вызывается метод их решения, реализованный в представленном на рисунке 6.17 классе алгоритмов линейной алгебры *CLinAlgebra*. Эта система уравнений будет содержать ранее не разрешенные уравнения.

В качестве примера работы данного алгоритма рассмотрим модель ХТС смешения двух многокомпонентных вещественных потоков, реализуемую на базе

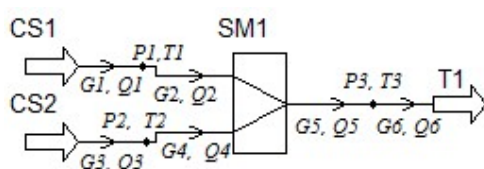


Рисунок 6.20 – Компонентная цепь ХТС смешения двух вещественных потоков

рассмотренного в пункте 3.4.4 компонента «Смеситель». Её КЦ с обозначением всех переменных связей компонентов представлена на рисунке 6.20.

Пусть источник CS1 осуществляет подачу в ХТС многокомпонентной смеси веществ, состоящей из метана и пропана в пропорциях 0,75:0,25. Смесь температурой $T_1=300$ К поступает в смеситель под давление $P_1=200$ кПа со скоростью $G_1=200$ моль/с.

Источником CS2 под давлением $P_2=100$ кПа со скоростью $G_2=100$ моль/с на второй выход смесителя подается смесь метана и этана в пропорциях 0,4:0,6, которая имеет температуру $T_2=350$ К.

Выполним действия алгоритма на основе расстановки потенциальных и потоковых переменных, приведенных на рисунке 6.20.

Шаг 1. Осуществляется опрос топологического блока всех компонентов. Его результатами являются топологические пары вида

$$\left[\begin{array}{l} \{-G1, P1\}, \{-Q1, T1\}, \{G2, P1\}, \{Q2, T1\}, \\ \{-G3, P2\}, \{-Q3, T2\}, \{G2, P2\}, \{Q4, T2\} \\ \{-G5, P3\}, \{-Q5, T3\}, \{G6, P3\}, \{Q6, T3\} \end{array} \right]. \quad (6.11)$$

Шаг 2. На основе массива топологических пар (6.11) опрос вычислительных блоков компонентов будет проводиться в следующем порядке: *CS1*, *CS2*, *T1*, *SM1*.

Шаг 3. Осуществляется опрос вычислительных блоков компонентов-источников *CS1* и *CS2*, модель которых описана в пункте 3.4.1. В ходе его проведения соответствующие переменные связей получают конкретные значения

$$\begin{array}{ll} P1 = 200000 & P2 = 100000 \\ G1 = 200 & G3 = 100 \\ T1 = 300 & T2 = 350 \\ Q1 = 1.224 \cdot 10^8 & Q3 = 4.936 \cdot 10^7 \\ C_{1CH4} = 0.75 & C_{2CH4} = 0.4 \\ C_{1C3H8} = 0.25 & C_{2C3H8} = 0 \\ C_{1C2H6} = 0 & C_{2C2H6} = 0.6 \end{array} \quad (6.12)$$

Соответствующие этим переменным элементы вектора *D* (6.7) получают значение TRUE.

Шаг 4. Формирование топологических уравнений на основе топологических пар компонентов (6.11)

$$\begin{array}{l} G_1 - G_2 = 0 \\ Q_1 - Q_2 = 0 \\ G_3 - G_4 = 0 \\ Q_3 - Q_4 = 0 \\ G_5 - G_6 = 0 \\ Q_5 - Q_6 = 0 \end{array} \quad (6.13)$$

Шаг 5. На основе (6.12) из топологических уравнений (6.13) получается:

$$\begin{aligned}
 G_2 &= G_1 = 200 \\
 Q_2 &= Q_1 = 1.224 \cdot 10^8 \\
 G_4 &= G_3 = 100 \\
 Q_4 &= Q_3 = 4.936 \cdot 10^7
 \end{aligned}
 \tag{6.14}$$

Два последних топологических уравнения из системы (6.13) остаются неразрешенными и на данном шаге добавляются в основную систему уравнений

$$\begin{aligned}
 G_5 - G_6 &= 0 \\
 Q_5 - Q_6 &= 0
 \end{aligned}
 \tag{6.15}$$

Шаг 6. Осуществляется опрос вычислительного блока компонентов, в котором производится формирование компонентных уравнений с их возможным разрешением относительно выходных переменных. В результате опроса смесителя в систему уравнений будут добавлены следующие уравнения

$$\begin{aligned}
 P_1 + P_2 - P_3 &= 0 \\
 G_2 + G_4 - G_5 &= 0 \\
 G_2 \cdot T_1 + G_4 \cdot T_2 - G_5 \cdot T_3 &= 0 \\
 Q_2 \cdot T_1 + Q_4 \cdot T_2 - Q_5 \cdot T_3 &= 0 \\
 C_{1i} \cdot G_2 + C_{2i} \cdot G_4 - C_{3i} \cdot G_5 &= 0
 \end{aligned}
 \tag{6.16}$$

В каждое из уравнений системы (6.16) входит одна ранее неопределенная переменная. Поэтому ее уравнения могут быть разрешены относительно каждой из переменных в следующем виде

$$\begin{aligned}
 P_3 &= P_1 + P_2 \\
 G_5 &= G_2 + G_4 \\
 T_3 &= \frac{G_2 \cdot T_1 + G_4 \cdot T_2}{G_5} \quad . \\
 Q_5 &= \frac{Q_2 \cdot T_1 + Q_4 \cdot T_2}{T_3} \\
 C_{3i} &= \frac{C_{1i} \cdot G_2 + C_{2i} \cdot G_4}{G_5}
 \end{aligned}
 \tag{6.17}$$

При подстановке ранее полученных в (6.12) и (6.14) значений получается, что

$$\begin{aligned}
 P_3 &= 300000 \\
 G_5 &= 300 \\
 T_3 &= 316,667 \\
 Q_5 &= 1.705 \cdot 10^8 \cdot \\
 C_{3CH_4} &= 0.633 \\
 C_{3C_2H_5} &= 0.167 \\
 C_{3C_3H_8} &= 0.2
 \end{aligned}
 \tag{6.18}$$

При подстановке значений (6.18) в топологические уравнения (6.15) получается, что

$$\begin{aligned}
 G_6 &= G_5 = 300 \\
 Q_6 &= Q_5 = 1.216 \cdot 10^8 \cdot
 \end{aligned}
 \tag{6.19}$$

Полученные значения совпадают с результатами моделирования секции смешения двух многокомпонентных вещественных потоков, представленных на рисунке 3.17, что доказывает правильность формирования и реализации алгоритма.

Рассмотренный алгоритм повышения быстродействия анализа функционирования ХТС позволяет за счет непосредственного определения значений ряда переменных цепи, допускающих свое явное выражение, на этапе формирования вычислительной модели цепи сократить размер решаемой алгоритмами линейной алгебры системы линейных алгебраических уравнений, а также минимизировать время проведения анализа цепей, допускающих явное выражения ряда переменных связей цепи.

6.4.2.2 Методика формирования системы уравнений с учетом их класса

Для исключения ситуаций, возникающих при линеаризации нелинейных (2.10) и алгебраизации дифференциальных уравнений (2.11), требующих перемещения влево коэффициентов уравнений, находящихся в массивах $m2$ и $am3$ объекта структуры *SSystemEquations*, представленной на рисунке 6.15, предлагается использовать алгоритм формирования системы уравнений с учетом их класса. Суть его заключается в следующем:

1. Линейные уравнения (2.9) с постоянными коэффициентами ($KL = 10$) формируются один раз за весь вычислительный эксперимент и не подвергаются алгоритму линеаризации нелинейных и алгебраизации дифференциальных уравнений.

2. Линейные уравнения (2.9), коэффициенты которых зависят от времени (частоты) ($KL = 11,12$) формируются на каждом шаге, но не подвергаются алгоритму линеаризации и алгебраизации.

3. Нелинейные ($KL = 13$) и дифференциальные ($KL = 14 - 21$) уравнения формируются на каждом шаге и обрабатываются основным алгоритмом, реализованном в методе *SetSystemEquations()* класса

IMarsComponent
+Nodes : int
+Branch : int
+Param : double
+CountNodes : int
+CountBranch : int
+CountParam : int
+CountEquation : int
+CountMembers : int
+fChangeOfParam : bool
+TopologQuery() : bool
+CalculatQuery() : bool
+MeasureQuery() : bool
+CalculatQuery1() : bool
+CalculatQuery2() : bool

Рисунок 6.21 – Изменение в классе вычислительных компонентов для реализации методики

свободным членом ($KL = 10$);

2) *CalculatQuery1()* – опрос вычислительного блока компонентов, предназначенный для формирования линейных уравнений с изменяемыми во времени коэффициентами и/или свободным членом ($KL = 11,12$);

лизованном в методе *SetSystemEquations()* класса *CFormirEquations*, структура которого приведена на рисунке 6.16.

Реализация приведенной методики требует осуществления следующих шагов:

– дополнение родительского класса вычислительных компонентов *IMarsComponent* (рисунок 6.21), а также всех наследуемых от него классов компонентов, следующими методами:

1) *CalculatQuery2()* – опрос вычислительного блока компонентов, в котором формируются только его линейные уравнения с независимыми от времени

(частоты) коэффициентами и сво-

CCircuit
+ComponentList : IMarsInterface
+CountNodes : int
+CountBranch : int
+TopologQuery() : bool
+CalculatQuery() : bool
+CalculatQuery1() : bool
+CalculatQuery2() : bool
+MeasureQuery() : bool

Рисунок 6.22 – Изменения в классе *CCircuit* для реализации методики

– в методе *CalculatQuery()* согласно методике будут формироваться только нелинейные ($KL = 13$) и дифференциальные уравнения ($KL = 14 - 21$).

В приложении 11 приведен пример реализации класса компонента «Абсорбер», приведенного на рисунке 3.23, с учетом описанных изменений.

Введение новых видов опросов вычислительного блока в анализируемую КЦ компонентов требует введения новых методов в класс *CCircuit*, представленный на рисунке 6.12. В данный класс включены следующие методы (изменения в классе приведены на рисунке 6.22):

– *CalculatQuery1()* – метод формирования линейных уравнений с изменяющимися во времени коэффициентами, формируемых компонентами исследуемой КЦ;

– *CalculatQuery2()* – метод формирования линейных уравнений, коэффициенты которых не зависят от времени (частоты).

В соответствии с описанными принципами формирования систем уравнений, основной алгоритм формирования уравнений, реализованный в методе *SetSystemEquations* класса *CFormirEquations*, представлен на рисунке 6.23. Он работает во временной области быстрее

обычного, так как при исследовании ХТС количество дифференциальных уравнений вида (2.11) зачастую не превышает 25 % от общего количества уравнений в системе, составленной универсальным вычислительным ядром для исследуемой компонентной цепи.

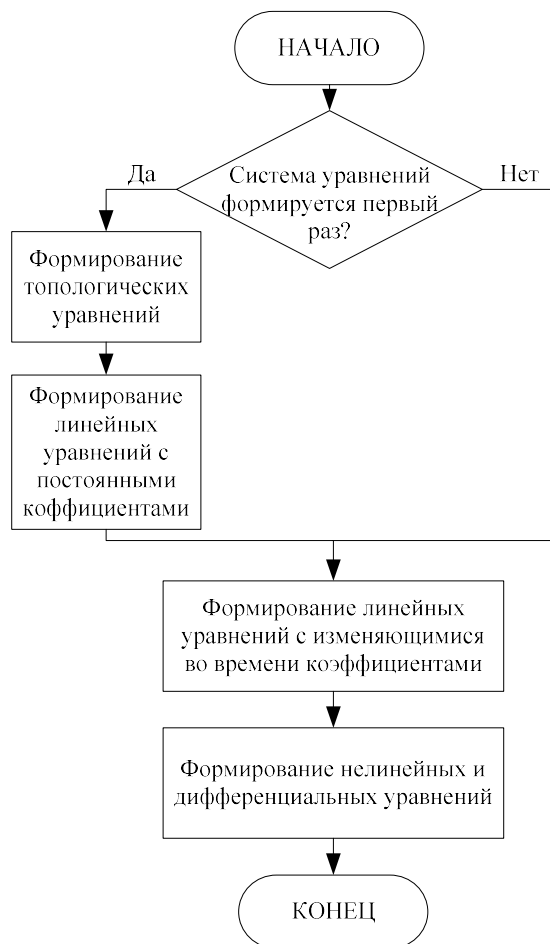


Рисунок 6.23 – Алгоритм формирования системы уравнений с учетом их класса

6.4.2.3 Методика анализа КЦ ХТС с постоянной обратной матрицей

Зачастую при решении задач исследования и функционального проектирования СТус приходится производить различные виды многовариантного анализа и параметрического синтеза. Алгоритмические компоненты для их проведения приведены в пункте 4.7. Обычно при решении таких задач осуществляется многократных анализ исследуемого ХТС во временной или частотной области с постоянным шагом. В этом случае основная матрица $[\Phi]$ линеаризованной и алгебраизованной системы уравнений (2.25) имеет постоянные коэффициенты, а компоненты вектора свободных членов W изменяются в зависимости от значений времени и получаемых значений вектора решения V . Анализ КЦ ХТС данного класса может осуществляться на основе методики, в основе которой лежит алгоритм решения систем единой алгебраизованных дифференциальных уравнений. Постоянство значений коэффициентов позволяет один раз осуществить обращение матрицы $[\Phi]$

$$[\Phi_1] = [\Phi]^{-1} \quad (6.20)$$

и хранить ее в разреженном виде для быстрого многократного умножения на изменяющийся вектор правых частей $[W]$.

При этом на каждом шаге по времени (частоте) будут производиться только следующие действия:

IMarsInterface
+Nodes : int
+Branch : int
+Param : double
+CountNodes : int
+CountBranch : int
+CountParam : int
+CountEquation : int
+CountMembers : int
+fChangeOfParam : bool
+TopologQuery() : bool
+CalculatQuery() : bool
+CalculatQuery1() : bool
+CalculatQuery2() : bool
+MeasureQuery() : bool
+AbsoluteTermQuery() : bool

Рисунок 6.24 – Изменения в классе *IMarsInterface* для реализации алгоритма

CCircuit
+ComponentList : IMarsInterface
+CountNodes : int
+CountBranch : int
+TopologQuery() : bool
+CalculatQuery() : bool
+CalculatQuery1() : bool
+CalculatQuery2() : bool
+MeasureQuery() : bool
+AbsoluteTermQuery() : bool

Рисунок 6.25 – Изменения в классе *CCircuit* для реализации алгоритма

1. С помощью специального введенного блока компонентов будет осуществляться их опрос, в результате которого должны рассчитываться только значения соответствующих компонентов вектора правых частей $[W]$ системы уравнений (2.25).

2. Полученный в результате данного опроса вектор W будет умножаться справа на матрицу $[\Phi]$ с целью получения вектора решения V на конкретном шаге анализа

$$V = [\Phi_1] \cdot W . \quad (6.21)$$

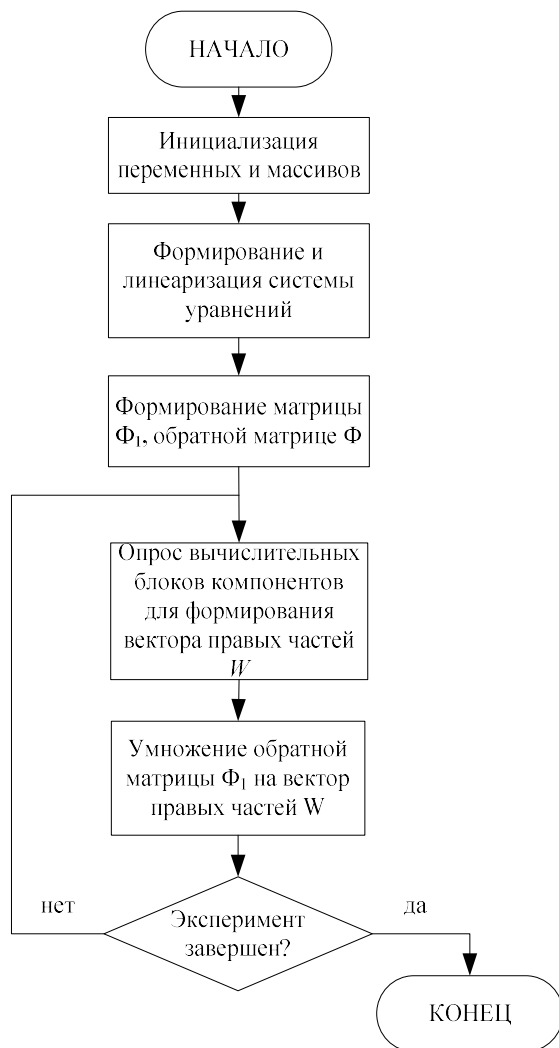


Рисунок 6.26 – Блок схема алгоритма анализа компонентных цепей ХТС с постоянной обратной матрицей

Для реализации данного алгоритма потребовалось выполнить следующие модификации классов универсального вычислительного ядра [103]:

1. Ввести новый метод опроса вычислительного блока компонентов *AbsoluteTermQuery()* в интерфейсный класс вычислительных компонентов *IMarsInterface*, приведенный на рисунке 6.24, и в класс компонентной цепи *CCircuit*, изменения в котором представлены на рисунке 6.25. В результате введенных методов будет формироваться только вектор правых частей W в объекте структуры *SSystemEquations*, приведенной на рисунке 6.15.

2. В классе *CAnaliz* ввести метод анализа КЦ *AnalizWithInverseMatrix*, позволяющий анализировать ХТС с неоднородными векторными связями на основе постоянной обратной матрицы. Блок-схема алгоритма данного метода приведена на рисунке 6.26.

родными векторными связями на основе постоянной обратной матрицы. Блок-схема алгоритма данного метода приведена на рисунке 6.26.

Описанный алгоритм применим для КЦ химико-технологических систем, протекающие процессы в которых описываются линейными алгебраическими $KL = 10 - 11$ и дифференциальными $KL = 14 - 17$ уравнениями с постоянными коэффициентами.

6.5. Интерпретатор математико-алгоритмических конструкций

Целью функционирования интерпретатора языка МАК является построение алгоритмической КЦ (2.63) на основе моделей сценариев исследования и функционального проектирования СТУС, сформированных в графической форме на логическом уровне многоуровневой КЦ СТУС. Для работы имитационного ядра, задачей которого является осуществление функционирования сценариев исследования и функционального проектирования СТУС в виде совокупностей соединенных между собой алгоритмических компонентов (2.65), информацию об алгоритмической КЦ достаточно представить в виде узловой матрицы [196]. Каждый её элемент может принимать одно из следующих значений

$$A_{i,j} = \begin{cases} 1 & \text{ветвь } i \text{ инцидентна узлу } j \text{ и информация} \\ & \text{передается от компонента к узлу} \\ -1 & \text{ветвь } i \text{ инцидентна узлу } j \text{ и информация} \\ & \text{передается от узла к компоненту} \\ 0 & \text{ветвь } i \text{ не инцидентна узлу } j \end{cases}, \quad (6.22)$$

где $i = \overline{1, CB}$, $j = \overline{1, CN}$, CB – количество связей компонентов формируемой алгоритмической КЦ; CN – количество ее узлов.

IAlgorithmicComponent
+Nodes : int
+Branch : int
+Attributes : SAttributes
+CountAlgorithmicWires : int
+GetDirectionWire() : int

Рисунок 6.27 –
Интерфейсный класс
алгоритмических
компонентов

Формирование узловой матрицы A (6.22) осуществляется на основе перебора всех связей алгоритмических компонентов (2.65), являющихся экземплярами приведенного на рисунке 6.5 класса $CWire$. Такие компоненты, формирующую подцепь логического уровня МКЦ, должны наследоваться от интерфейсного класса $IAlgorithmicComponent$, который представлен на рисунке 6.27. Его полями

являются:

- *Nodes* – массив номеров узлов, заданных в ГКБ алгоритмической КЦ;
- *Branch* – массив номеров ветвей в глобальном координатном базисе алгоритмической КЦ;
- *Attributes* – массив атрибутов-параметров компонента, каждый из которых представляет собой структуру, хранящую данные любого типа;
- *CountLogicWires* – количество алгоритмических (логических) связей компонента. Данный параметр определяет размерность массивов *Nodes* и *Branch*.

Методом класса *IAlgorithmicComponent*, вызываемым интерпретатором математико-алгоритмических конструкций при формировании КЦ сценариев, представленных компонентными подцепями логического уровня МКЦ, является метод

CAlgorithmicCircuit
+ComponentsList : IAlgorithmicComponent
+CountNodes : int
+CountBranch : int
+A : int
+AddComponent()
+GetComponent() : IAlgorithmicComponent
+SearchComponent() : IAlgorithmicComponent

Рисунок 6.28 – Класс алгоритмической компонентной цепи

GetDirectionWire(). Он возвращает тип направления потока сообщений в связи, номер которой *nWire* является аргументом данного метода. Если направление связи – от узла к компоненту, то метод возвращает значение -1 , а если от компонента к узлу – то значение 1 .

Формируемая интерпретатором алгоритмическая КЦ представляет собой класс *CAlgorithmicCircuit*, представленный на рисунке 6.28 и содержащий в себе следующие поля:

- *ComponentList* – список алгоритмических компонентов класса *IAlgorithmicComponent*, которые помещаются в него в процессе работы алгоритма формирования алгоритмической КЦ;
- *CountNodes* – количество узлов алгоритмической КЦ (2.63), которое соответствует количеству строк матрицы *A*;
- *CountBranch* – количество связей (ветвей) всех компонентов, входящих в формируемую алгоритмическую КЦ. Данный параметр соответствует количеству столбцов матрицы *A*;
- *A* – узловая матрица, каждый элемент которой формируется по правилу (6.22).

Класс алгоритмической КЦ *CAgorithmicCircuit* включает в себя следующие методы:

- *AddComponent()* – метод добавления компонента, программно-алгоритмическая реализация которого наследуется от класса *IAlgorithmicComponent*, в список *ComponentList*;
- *GetComponent()* – метод выдачи компонента по номеру его связи, который указывается в качестве параметра данного метода;
- *SearchComponent()* – метод, возвращающий компонент с входной ветвью, номер которой, указанный в ГКБ алгоритмической КЦ, передается в качестве аргумента данного метода.

Алгоритм формирования алгоритмической КЦ класса *CAgorithmicCircuit*, представленного на рисунке 6.28, заключается в переборе всех связей типа *CWire*, приведенного на рисунке 6.5, с учетом того, все члены массивов *Nodes* и *Branch* всех компонентов, наследуемых от класса *IAlgorithmicComponent*, проинициализированы, то есть каждому их элементов присвоено значение -1 . Для каждой связи с номером i , с помощью которой соединены два алгоритмических компонента $CA1$ и $CA2$ соответственно выводами $nPin1$ и $nPin2$, осуществляются следующие шаги:

Шаг 1. Если $Nodes[nPin1]$ компонента $CA1$ (далее $CA1.Nodes[nPin1]$) равен -1 и $CA2.Nodes[nPin2] == -1$, то выполняются:

- 1) присваивание указанным полям значения *CountNodes* класса *CAgorithmicCircuit*;
- 2) присваивание полям $CA1.Branch[nPin1]$ и $CA2.Nodes[nPin2]$ очередных значений *CountBranch*, увеличение всякий раз это поле класса *CAgorithmicCircuit* на единицу;
- 3) присваивание элементу матрицы $A[CA1.Nodes[nPin1], CA1.Branch[nPin1]]$ результата выполнения метода *GetDirectionWires(nPin1)*, вызванного для объекта $CA1$ (далее $CA1.GetDirectionWires(nPin1)$);
- 4) присваивание элементу матрицы $A[CA2.Nodes[nPin2], CA2.Branch[nPin2]]$ результата выполнения метода *GetDirectionWires(nPin2)*.

Шаг 2. Если $CA1.Nodes[nPin1] \neq -1$, а элемент $CA2.Nodes[nPin2] = -1$, то выполняются следующие действия:

1. Элементу $CA2.Nodes[nPin2]$ присваивается значение $CA1.Nodes[nPin1]$;
2. Полю $CA2.Branch[nPin2]$ присваивается очередное значение *Count-Branch* класса *IAlgorithmicComponent*, которое затем увеличивается на 1;
3. Элементу матрицы $A[CA2.Nodes[nPin2], CA2.Branch[nPin2]]$ присваивается результат выполнения метода $CA2.GetDirectionWires(nPin2)$.

Шаг 3. Если $CA1.Nodes[nPin1] = -1$, а элемент $CA2.Nodes[nPin2] \neq -1$, то выполняются следующие действия:

1. Элементу $CA1.Nodes[nPin1]$ присваивается значение $CA2.Nodes[nPin2]$;
2. Полю $CA1.Branch[nPin1]$ присваивается очередное значение *Count-Branch* класса *IAlgorithmicComponent*, которое затем увеличивается на 1;
3. Элементу матрицы $A[CA1.Nodes[nPin1], CA1.Branch[nPin1]]$ присваивается результат выполнения метода $CA1.GetDirectionWires(nPin1)$.

Шаг 4. Если $CA1.Nodes[nPin1] \neq -1$ и $CA2.Nodes[nPin2] \neq -1$, то

1. К номеру наибольшего узла присваивается номер наименьшего;
2. Номера узлов в формируемой алгоритмической КЦ уменьшаются на единицу;
3. Строка с номером большего элемента удаляется из матрицы A со сдвигом вверх строк, лежащих ниже ее.

В результате обработки каждой связи по приведенному алгоритму будет сформирована алгоритмическая КЦ в виде объекта класса *CAlgorithmicCircuit*, позволяющая осуществлять функционирование сценариев исследования и функционального проектирования сложных технических (технологических) управляемых систем, располагающиеся на логическом уровне многоуровневой КЦ СТУС и включающие в себя сценарии функционирования их устройств управления.

6.6. Имитационное ядро

6.6.1 Структура передаваемого сообщения

Основная задача имитационного ядра – обеспечение функционирования сценариев исследования и функционального проектирования СТУС, являющихся алгоритмическими подцепями МКЦ СТУС вида (2.63), сформированных интерпретатором математико-алгоритмических конструкций. Такое функционирование заключается в осуществлении возможности передачи результатов выполнения отдельных операций сценариев, каждая из которых выражена своим компонентом. Представленные информацией различных типов данных результаты выполнения функций каждого компонента передаются подключенным к ним компонентам с помощью *сообщений*. Каждое сообщение множества M (2.63) представляется в виде

$$M = (D, T, B, N), \quad (6.23)$$

где D – данные любого типа, передаваемые в сообщении M между компонентами; T – тип данных, передаваемых в сообщении; B – номер ветви в ГКБ алгоритмической КЦ C_A (2.63), с которой передается сообщение M узлу N ; N – номер узла, заданный в ГБК алгоритмической КЦ, которому передается сообщение.

CMessage
+Data : void
+Type : int
+nPin : int
+nBranch : int

Рисунок 6.29 –
Структура
сообщения
CMessage

В программной реализации имитационного ядра сообщение M реализуется в виде структуры данных *CMessage*, представленной на рисунке 6.29 и включающей в себя следующие поля:

- *Data* – указатель на передаваемые в сообщении данные различного типа;
- *Type* – идентификатор типа данных, необходимый для осуществления операции разыменования данных компонентами-приемниками сообщения;
- *nPin* – номер узла компонента-отправителя, с которого было отправлено сообщение, заданный в глобальном координатном базисе алгоритмической КЦ;
- *nBranch* – заданный в ГКБ алгоритмической КЦ номер связи (ветви), в которого отправляется сообщение. Имитационное ядро изменяет его на номер связи

компонента-получателя для идентификации связи, по которой он получил сообщение.

Процесс формирования и отправки компонентом сообщения производится в ряде случаев, которые далее называются *срабатыванием компонента*. Срабатывать определенный компонент множества K_A (2.65) может в следующих случаях:

- во время запуска или останова работы МКЦ СТУС;
- в начале и по окончании очередной итерации анализа модели ХТС во время решения задач исследования и функционального проектирования СТУС;
- при поступлении сообщения хотя бы на один из его входных узлов от других компонентов алгоритмической КЦ C_A (2.63);
- когда компонент получил сообщение с данными на каждый из своих входных узлов;

IAlgorithmicComponent
+Nodes : int
+Branch : int
+Attributes : SAttributes
+CountAlgorithmicWires : int
+GetDirectionWire() : int
+SendMessage()
+OnMessage()
+OnRun()
+OnStop()
+OnRunExperiment()
+OnStopExperiment()
+StepTime()

Рисунок 6.30 – Класс алгоритмических компонентов с методами работы с сообщениями

– если произошло взаимодействие с визуальным образом компонента-регулятора множества K_{V3} (2.68).

Для осуществления возможностей класс *IAlgorithmicComponent*, представленный на рисунке 6.30, помимо полей и методов, описанных в пункте 6.5, включает следующие методы, которые могут быть перегружены в наследуемых от него компонентах:

- *SendMessage()* – метод, в котором компонент формирует сообщение из данных, хранящихся в его параметрах и (или) полученных из ранее принятых сообщений;

– *OnMessage()* – метод приема и обработки сообщений, каждое из которых представляет собой объект представленной на рисунке 6.29 структуры *CMessage*;

– *OnRun()* – метод, срабатывающий при запуске МКЦ СТУС на выполнение. В нем инициализируются необходимые компоненту структуры данных и устанавливаются первоначальные значения его определенных полей, не являющихся параметрами, значения которых задаются пользователем в редакторе параметров компонента, приведенном на рисунке 6.3;

– *OnStop()* – метод, вызываемый у каждого компонента при окончании эксперимента. В нем производится установка значений соответствующих компонентов в первоначальное положение, а также освобождение памяти, занятой динамическими структурами данных;

– *OnRunExperiment()* – метод, реализуемый у компонентов логического уровня МКЦ СТЭС и вызываемый перед началом очередного анализа модели ХТС, являющейся подцепью объектного уровня МКЦ;

– *OnStopExperiment()* – метод, вызываемый у компонентов по окончании очередного анализа КЦ ХТС;

– *StepTime()* – метод, вызываемый у алгоритмических компонентов перед очередной итерацией анализа КЦ ХТС по времени или частоте.

С целью оправки сообщения с данными любого типа алгоритмический компонент формирует объект структуры *CMessage* и передает его имитационному ядру путем вызова метода *SendMessage()*. В этом случае в ядре вызывается метод, реализующий алгоритм передач сообщений [116], работа которого основывается на анализе узловой матрицы A (6.22).

6.6.2 Алгоритм передачи сообщений

Алгоритм передачи сообщений, представленный на рисунке 6.31, лежит в основе работы имитационного ядра комплекса программ моделирования химико-технологических систем.

Он позволяет организовать передачу сообщений с данными различных типов между компонентами логического уровня многоуровневой КЦ, к которым относятся математические и алгоритмические компоненты, описываемые представленными в явном виде математико-алгоритмическими выражениями относительно выходных переменных.

Во время запуска эксперимента один раз производится формирование узловой матрицы A , каждый элемент которой формируется по формуле (6.22) для всех компонентов множества $K_A \subset C_A$ (2.63).

В компоненте K_{Ai} , в котором произошло срабатывание, формируется сообщение M (6.23). Такой компонент будет являться компонентом-отправителем сообщения. Сообщение M содержит информацию $M.D$ типа $M.T$. Согласно алгоритму передачи сообщений, приведенному на рисунке 6.31, оно передается другим компонентам подцепи C_A (2.63) МКЦ, подключенным к определенному выходному узлу компонента K_{Ai} .

В сообщении M указываются топологические признаки компонента-отправителя: ветвь $M.B$, с которой отправляется сообщение, и узел $M.N$, которому оно передается. Указанное сообщение M передается компонентом-отправителем имитационному ядру в виде экземпляра структуры $CMessage$, приведенной на рисунке 6.29. Путем проверки всех элементов столбца $M.N$ матрицы A (6.22) на равенство 1

производится выявление всех ветвей NB , подключенных к узлу с номером $M.N$ и являющихся компонентами-приемниками сообщений. При обнаружении таких связей путем вызова функции $SearchComponent$ производится поиск компонента

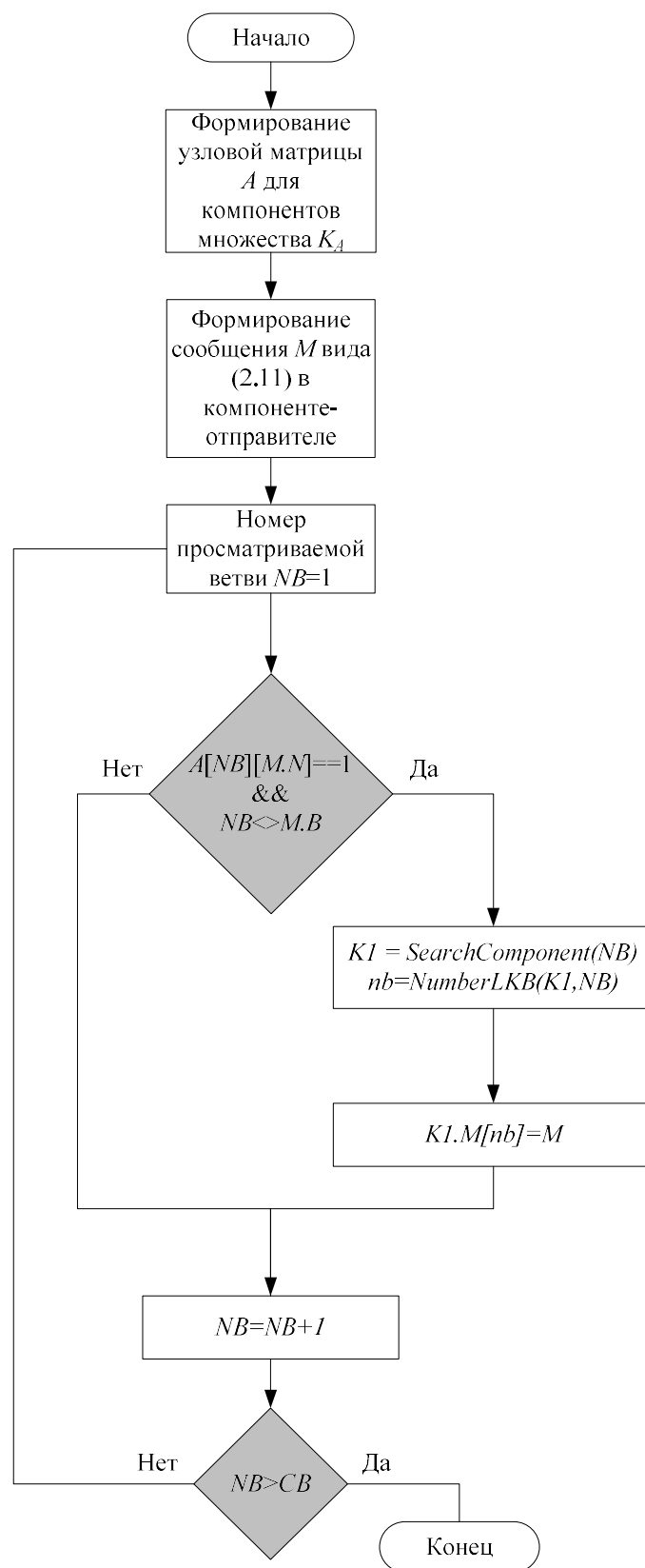


Рисунок 6.31 – Алгоритм передачи сообщений

K_{Aj} , которому принадлежит ветвь с номером NB , представленным в ГКБ алгоритмической КЦ C_A (2.63). С целью передачи сообщения M на определенную ветвь компонента K_{Aj} с помощью функции

$$nb = \text{NumberLKB}(K_{Aj}, NB) \quad (6.24)$$

производится поиск ветви с номером nb , заданным в ЛКБ компонента K_{Aj} . После этого путем присваивания

$$K_{Aj}.M[nb] = M \quad (6.25)$$

производится непосредственная передача сообщения компоненту K_{Aj} , подключенному к узлу $M.N$.

Описанный алгоритм заканчивает работу после того, как были просмотрены все элементы столбца с номером $M.N$ матрицы A .

6.6.3 Структура алгоритмических компонентов

Согласно языку моделирования алгоритмических конструкций, описанному в главе 4, каждый компонент множества K_A алгоритмической КЦ C_A (2.63) представляет собой совокупность геометрического, топологического и вычислительного (функционального) блоков. Его структура приведена на рисунке 6.32.

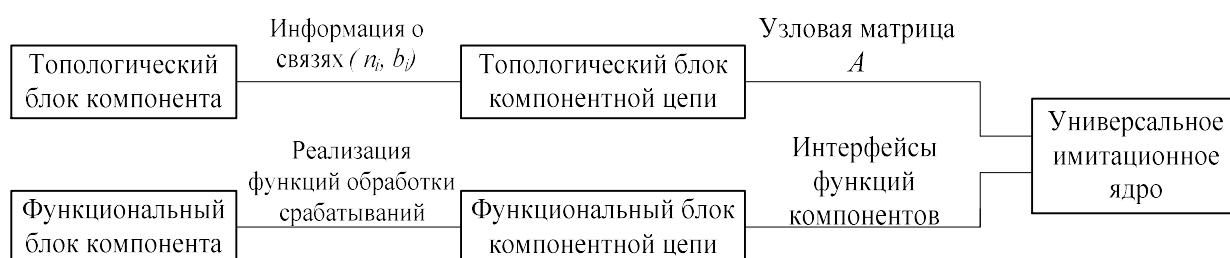


Рисунок 6.32 – Структура алгоритмических компонентов и их связь с имитационным ядром

Топологический блок каждого из компонентов множества K_A включает информацию обо всех его связях, предназначенных для обмена сообщениями с другими алгоритмическими компонентами. Функциональный блок представляет собой совокупность функций (методов класса компонента) преобразования входных данных компонента в выходные. Все компоненты множества K_A должны взаимо-

действовать с имитационным ядром, в рамках которого реализуется представленный на рисунке 6.31 алгоритм передачи сообщений. Если с помощью геометрического блока каждый компонент множества K_A взаимодействует с графическим редактором МКЦ СТУС, то взаимодействие с имитационным ядром осуществляется с помощью топологического и функционального блоков, которые в программно-алгоритмической реализации компонентов представлены соответствующими методами классов, реализующих данные блоки компонентов.

Топологический блок каждого компонента включает в себя информацию обо всех его связях $s_i = (b_i \eta^\pm n_i)$. С их помощью организуются соединения с другими компонентами и преобразование топологических координат компонента из ЛКБ каждого компонента в ГКБ рассматриваемой алгоритмической КЦ $S_i = (N_i, B_i)$. На основе ГКБ цепи формируется ее топологический блок, включающий в себя узловую матрицу A . На ее основе в имитационном ядре реализуется алгоритм передачи сообщений.

Функциональный блок компонентов множества K_A (2.63) включает в себя реализацию функций приема сообщений, обработки содержащихся в них данных, а также методов формирования и отправки сообщений другим компонентам. В него также входят функции обработки срабатываний компонента. С целью взаимодействия с имитационным ядром определенные функции всех компонентов имеют одинаковые прототипы, совокупность которых образуют функциональный блок алгоритмической КЦ.

В компонентах-источниках множества K_{AS} , обобщенный компонент которых представлен на рисунке 2.11, а, реализуются функции обработки всех видов срабатываний компонента и преобразования информации в данные, предназначенные для передачи. К таким компонентам относятся:

- измерительные компоненты, осуществляющие измерения текущих значений потенциальных и потоковых переменных КМ ХТС и наблюдаемых переменных реального объекта и их передачу с объектного на логический уровень многоуровневой КЦ СТУС (2.50);

– компоненты-регуляторы множества K_{V3} из (2.68), являющие компонентами языка МАК и реализующие операцию передачи информации с визуального на логический уровень (2.52);

– компоненты логического уровня, взаимодействующие с внешними аппаратными средствами с целью доставки в МКЦ СТУС актуальных данных, хранящихся в информационных ресурсах.

В компонентах преобразователей множества K_{AP} (2.67), обобщенный вид которых представлен на рисунке 2.11, б, должны быть реализованы функции преобразования входной информации, поступающей по совокупности входных связей, и функции формирования выходной информации, предназначенной для пересылки другим компонентам множества K_A . К ним относятся математические, логические, алгоритмические и другие компоненты.

Компоненты множества K_{AR} (2.67), типовой компонент которых представлен на рисунке 2.11, в, у которых реализуются только функции приема и обработки сообщений, будут являться компонентами-приемниками. К ним относятся компоненты, предназначенные для параметризации компонентов модели ХТС, компоненты-команды управления реальным объектом, а также компоненты-визуализаторы результатов моделирования K_V (2.68) и компоненты, осуществляющие взаимодействие МКЦ СТУС с генератором отчетных форм.

6.7. Библиотека моделей компонентов

Библиотека моделей компонентов (БМК) комплекса программ моделирования химико-технологических систем представляет собой совокупность модулей, содержащих программно-алгоритмические реализации компонентов всех трех подязыком языка МКЦ. Все содержащиеся в них компоненты распределены по разделам. Каждый компонент может быть наследован от различных интерфейсных классов [197], предоставляющих ему возможность взаимодействия с различными моделями комплекса программ.

Центральным модулем БМК является динамически загружаемая библиотека *ВМКManager.dll*, называемая менеджером БМК. Она позволяет подгружать дина-

мические библиотеки с непосредственными реализациями компонентов. Она взаимодействует с многослойным редактором на уровне вызова функций формирования дерева компонентов и инициализации конкретного компонента, реализованного в определенном модуле БМК. Для предоставления информации о компоненте в его программной реализации представлена функция *SetInfoComponent()*, в которой с помощью определенной структуры передается уникальный идентификационный номер компонента; его имя, с которым он отображен в дереве компонентов; имя каталога, в котором он должен находиться, а также наименование модуля БМК, в котором реализован компонент.

В конструкторе компонента, вызываемом при создании каждого объекта определенного класса, реализуются блок формирования метки компонента, блок идентификации выводов компонента, предназначенных для соединения с другими компонентами в редакторе, блок инициализации информации о слоях редактора, на которых имеет свои отображения компонент. Если компонент имеет динамическое изображение на каком-либо слое редактора, изменяющееся во время редактирования или анализа модели, то в нем реализуется метод *Paint()*.

Компоненты, предполагающие сохранение/чтение некоторых данных при сохранении/чтении схемы, в которую они включены, должны включать в себя реализацию функции сериализации *Serialize()*.

Перечисленный набор функций, реализуемых в компоненте, позволяет организовать его взаимодействие с графическим редактором многоуровневых КЦ СТУС. Более подробно рассмотрим методы взаимодействия программно-алгоритмических реализаций компонентов с универсальным вычислительным и имитационным ядрами, а также пользователем в режиме визуализации результатов моделирования и измерения.

6.7.1 Методы взаимодействия компонента с универсальным вычислительным ядром

Компоненты, являющиеся компьютерными моделями элементов химико-технологических систем, принадлежат объектному уровню многоуровневой КЦ СТУС, представленной на рисунке 2.5, являются элементами языка моделирова-

ния ХТС, нотации которого приведены в пункте 2.4, и отображаются на объектном слое многослойного редактора. Такие компоненты, помимо интерфейсного класса *CComponent*, представленного на рисунке 6.4, должны наследоваться от классов *IBaseChTSMModel*, приведенного на рисунке 6.6 и реализующего в себе структуры неоднородных векторных связей компонента, и *IMarsComponent*, представленного на рисунке 6.11, который включает в себя основные виды опросов топологического, вычислительного и измерительного блоков компонента. Для формирования информации об НВС компонента в каждом из них реализуется метод *SetVectorConnection*, в котором:

– Для каждой энергетической связи, входящей в состав любой НВС компонента, вызывается метод

$$\text{SetEnergyConnection}(\langle \text{NumberPins} \rangle, \langle \text{NumberWires} \rangle, \langle \text{NameAspect} \rangle, \langle \text{NameNVar} \rangle, \langle \text{NumberNVar} \rangle, \langle \text{NameBVar} \rangle, \langle \text{NumberBVar} \rangle)$$

где $\langle \text{NumberPins} \rangle$ – номер вывода компонента – он же номер НВС компонента; $\langle \text{NumberWires} \rangle$ – номер энергетической связи в рамках НВС, соответствующей выводу с номером $\langle \text{NumberPins} \rangle$; $\langle \text{NameAspect} \rangle$ – имя аспекта, которому принадлежит данная энергетическая связь; $\langle \text{NameNVar} \rangle$ – номер потенциальной переменной данной энергетической связи, заданный в ЛКБ компонента; $\langle \text{NameBVar} \rangle$ – имя потоковой переменной, которым она выражена в математической модели компонента; $\langle \text{NumberBVar} \rangle$ – номер потоковой переменной, заданный в ЛКБ компонента.

– Для каждой информационной связи в составе определенной НВС вызывается метод

$$\text{SetInformationConnection}(\langle \text{NumberPins} \rangle, \langle \text{NumberWires} \rangle, \langle \text{NameAspect} \rangle, \langle \text{NameNVar} \rangle, \langle \text{NumberNVar} \rangle)$$

С целью опроса топологического блока компонентов, содержащих хотя бы одну элементарную связь в составе любой НВС, в классе компонента реализуется метод *TopologQuery()*. Для каждой элементарной связи в нем вызывается метод

$$SetTopologQuery(\langle pEq \rangle, \pm \langle NumberBVar \rangle, \langle NumberNVar \rangle),$$

где $\langle pEq \rangle$ – структура представленного на рисунке 6.15 типа *SSystemEquations*, в которой формируется топологический массив $m1$; $\langle NumberBVar \rangle$ – номер потоковой переменной связи компонента в его ЛКБ, которая берется со знаком «минус», если поток в ней направлен от узла к компоненту, и со знаком «плюс», если – от компонента к узлу; $\langle NumberNVar \rangle$ – номер потенциальной переменной (номер узла), для которой (-ого) формируется конкретное уравнение.

Формирование компонентных уравнений модели компонента осуществляется в трех методах:

- 1) *CalculatQuery2()*, в котором формируются линейные уравнения (2.9);
- 2) *CalculatQuery1()*, формирующий уравнения источников (2.12);
- 3) *CalculatQuery()*, предназначенный для формирования нелинейных (2.10) и дифференциальных уравнений (2.11).

Для формирования каждого уравнения в этих методах должен быть вызван метод

$$SetEquation(\langle pEq \rangle, \langle KL \rangle, \langle W \rangle, \langle CC \rangle, \langle Var \rangle, \langle Coeff \rangle), \quad (6.26)$$

где $\langle pEq \rangle$ – структура типа *SSystemEquation*, в которую добавляется формируемое уравнение; $\langle KL \rangle$ – тип уравнения, устанавливаемый в соответствии с таблицей 6.1; $\langle W \rangle$ – свободный член уравнения; $\langle CC \rangle$ – количество переменных в формируемом уравнении, в том числе стоящие под знаком дифференциала, и коэффициентов при них; $\langle Var \rangle$ – массив номеров переменных, заданных в ЛКБ компонента (если переменная стоит под знаком дифференциала в уравнениях вида (2.11), то к ее номеру добавляется числовой признак); $\langle Coeff \rangle$ – массив коэффициентов при переменных в уравнениях.

Для реализации методики повышения быстродействия ВЭ, описанной в пункте 6.4.2.3, в соответствующих компонентах, все уравнения которых имеют постоянные коэффициенты при переменных, реализуется метод *Absolute-*

TermQuery(). В нем осуществляется формирование массива свободных членов W переданной в него структуры типа *SSystemEquation*.

Пример исходного кода компонента языка моделирования ХТС с реализацией перечисленных методов приведен в приложении 12.

В классах измерительных компонентов, описанных в пункте 2.4.3, задачей которых является выборка определенных значений потенциальных и (или) потоковых переменных и их передача с объектного на логический уровень МКЦ СТУС (2.50), реализуется метод *MeasureQuery()*. В него передается структура *SSystemEquation* с вычисленным на данном шаге решения вектором V (2.15). Выборка из него соответствующих значений переменных осуществляется по номерам потенциальных и потоковых переменных компонента, заданных в ГКБ цепи.

Реализация измерителей потенциальных и потоковых переменных с учетом задания размера НВС, к которой они подключены, и указания номера измеряемой переменной приведена в приложении 11.

6.7.2 Набор численных методов расчета параметров и коэффициентов уравнений компонентов физико-химических систем

Особенностью моделей компонентов химико-технологических систем, которые были рассмотрены в главе 3, является зависимость их коэффициентов от давлений и температуры многокомпонентных вещественных потоков, а также от концентраций веществ, составляющих эти потоки, которые являются переменными связей рассматриваемого компонента. Набор реализованных численных методов, с помощью которых осуществляется вычисление значений соответствующих коэффициентов, реализован в виде класса *IFizChimInterface*, представленного на рисунке 6.33. Он содержит следующие поля:

- *PathOfBD* – путь к базе данных характеристик компонент потока, в которой хранятся характеристики веществ, составляющих многокомпонентные вещественные потоки;

- *CountSubstance* – количество веществ в рассматриваемой модели ХТС;

IFizChimInterface
+PathOfBD : String
+CountSubstance : int
-NamesSubstance : String
+GetCountSubstance() : long
+CalculateOfFriction()
+SetVectorOfConzentracion()
+CalculateOfDensity()
+CalculateOfViscosity()
+CalculateOfHeatCapacity()
+CalculateOfCubeEquation()
+MetodicsOfSoavaRedlichCvong()
+InitMethodicsSoavaRedlichCvong()
+CloseMetodicsSoavaRedlichCvong()
+ShilovsMetodics()

Рисунок 6.33 – Класс методов определения характеристик моделей физико-химических систем

– *NamesSubstance* – массив имен веществ, составляющих многокомпонентные потоки в анализируемой модели ХТС.

В классе *IFizChimInterface* реализованы следующие численные методы:

– *GetCountSubstance()* – возвращает количество компонент-веществ, действующих во всех компонентах анализируемой модели ХТС;

– *CalculateOfFriction()* – расчет коэффициента потерь на трение многокомпонентного вещественного потока о стенки элементов и аппаратов ХТС по формуле (3.48). Он используется в математической модели гидротермодинамического сопротивления трубы (3.47);

– *SetVectorOfConzentracion()* – метод формирования вектора концентраций веществ на некотором входе или выходе рассматриваемой модели элемента или аппарата ХТС. В нем производится выборка соответствующих значений из общего вектора решения модели (2.15) и распределение по порядковым номерам веществ, имена которых расположены в массиве *NamesSubstance*;

– *CalculateOfDensity()* – метод вычисления плотности многокомпонентного потока веществ по формуле (3.51) по их плотностям, заданным в базе данных веществ, и вычисленным концентрациям в исследуемом потоке;

– *CalculateOfViscosity()* – метод расчета вязкости многокомпонентного вещественного потока по формуле (3.50);

– *CalculateOfHeatCapacity()* – метод расчета теплоемкости многокомпонентного вещественного потока по формуле (3.54) с использованием хранящихся в базе данных веществ удельных теплоемкостей отдельных компонент и их концентраций в исследуемом многокомпонентном потоке;

– *CalculateOfCubeEquation()* – численный метод решения кубического уравнения, используемый в методике Соава-Редлиха-Квонга [198] для расчета коэффициентов фазового равновесия системы «жидкость-газ»;

– *MethodicsSoavaRedlichCvong()* – программная реализация методики Соава-Редлиха-Квонга расчета коэффициентов фазового равновесия K_i , которые используются в уравнениях модели газового сепаратора (3.57);

– *InitMethodicsSoavaRedlichCvong()* – метод инициализации массивов данных, используемых для хранения результатов промежуточных вычислений, выполняемых в рамках методики расчета коэффициентов фазового равновесия Соава-Редлиха-Квонга;

– *CloseMethodicsSoavaRedlichCvong()* – метод завершения методики Соава-Редлиха-Квонга, в котором освобождается память, занимаемая массивами результатов промежуточных вычислений;

– *ShilovsMethodics()* – программная реализация численной методики Шилова [199] расчета коэффициентов фазового равновесия K_i , применяемых в уравнениях моделей компонентов ХТС.

Реализованные в интерфейсном классе *IFizChimInterface* численные методы вычисления характеристик многокомпонентных вещественных потоков и коэффициентов фазового равновесия используются различными классами, в которых реализуются модели компонентов ХТС. Из них основе в объектном уровне многоуровневой КЦ формируется компьютерная модель исследуемой ХТС, в которой имена компонентов циркулирующих в ней потоков задаются компонентами «Источник смеси веществ», который приведен на рисунке 3.6.

Определяемые на основе методов характеристики и коэффициенты применяются для формирования уравнений модели ХТС, анализ которой производится универсальным вычислительным ядром.

6.7.3 Методы взаимодействия компонента с имитационным ядром

Каждый компонент языка МАК, имеющий свое отображение на логическом слое многослойного редактора, взаимодействует с имитационным ядром, основанном на алгоритме передачи сообщений, работа которого описана в пункте 6.6. Это взаимодействие осуществляется путем отправки компонентом сообщений, а

также путем их приема от других компонентов, подключенных к его входным связям.

Отправка компонентом сообщений производится во время его срабатываний, которые были приведены в пункте 6.6.1. Для этого из любого метода, который обрабатывает события срабатывания компонента, осуществляется вызов метода

$$\text{SendMessage}(\langle pData \rangle, \langle nWires \rangle, \langle type \rangle),$$

где $\langle pData \rangle$ – указатель на данные различных типов, предназначенные для передачи другим компонентам; $\langle nWires \rangle$ – номер связи в ЛКБ компонента, по которой отправляется сообщение; $\langle type \rangle$ – тип передаваемого по связи сообщения.

При отправке компонентом-отправителем сообщения имитационному ядру в нем срабатывает алгоритм передачи сообщений, представленный на рисунке 6.31. Он передает сформированное сообщение всем компонентам-приемникам, подключенным к выходному выводу компонента-отправителя.

Получение компонентом-приемником сообщения от имитационного ядра осуществляется методом

$$\text{OnMessage}(\text{SMessage}^* \text{mes}),$$

в котором с учетом указанного в сообщении типа передаваемых данных $\text{mes} \rightarrow type$ и номера связи компонента $\text{mes} \rightarrow pWires$, на которую пришло сообщение, производится разбор данных $\text{mes} \rightarrow pData$.

Реализация класса компонента-преобразователя данных, сочетающего в себе свойства компонента-отправителя и компонента-приемника данных, приведена в приложении 12.

6.7.4 Методы взаимодействия пользователя с визуальными компонентами

При выполнении экспериментов с многоуровневой КЦ СТУС активным для пользователя является визуальный уровень, расположенный на визуальном слое редактора. На нем располагаются визуальные компоненты множества K_v (2.68), с помощью которых в формате языка ВИП формируются панели визуализации и интерактивного управления [108]. Имея свои отображения на логическом слое ре-

дктора, в программно-алгоритмических реализациях данных компонентов присутствуют функции отправки *SendMessage()* и приема сообщений от других компонентов *OnMessage()*. Для визуализации на визуальном слое редактора информации, принятой от компонентов логического уровня многоуровневой КЦ СТУС, в данных компонентах реализуется функции перерисовки изображения компонента

$$OnPaint(<dc>, <Layer>),$$

где *<dc>* – класс контекста устройства (экрана монитора, принтера и т.п.), на которое выводится графическая информация, сформированная методами данного класса [200]; *<Layer>* – идентификатор слоя редактора, на котором производится отображение данного компонента (в данном случае - визуального слоя).

Помимо визуальных компонентов метод *OnPaint()* может быть реализован для перерисовки отображаемых на объектном или логическом слоях редактора компонентов с изменяемым изображением. В частности, к ним относятся компоненты с переменным числом связей.

Для реакции компонента на действия пользователя на его изображения с помощью мыши, что актуально для управляющих компонентов множества K_{v3} (2.68), реализующий его класс должен быть наследован от интерфейсного класса *IMouseInterface*, который приведен на рисунке 6.34 и содержит следующие методы:

IMouseInterface
+OnLButtonDown() +OnLButtonUp() +OnRButtonDown() +OnRButtonUp() +OnDblClick() +OnMouseMove()

Рисунок 6.34 – Интерфейсный класс обработки действий мыши пользователя

– *OnLButtonDown()* – реакция компонента на нажатие левой кнопки мыши по его изображению (изображениям) на любом слое редактора;

– *OnLButtonUp()* – реакция компонента на отпускание левой кнопки мыши, нажатой на изображении компонента;

– *OnRButtonDown()* – реакция компонента на нажатие правой кнопки мыши по его изображениям;

– *OnRButtonUp()* – реакция компонента на отпускание правой кнопки мыши, нажатой на его изображении;

- *OnDblClick()* – реакция компонента на двойной щелчок левой кнопки мыши по его изображению;
- *OnMouseMove()* – реакция компонента на подведение мыши к его изображениям на различных слоях редактора.

Классы, реализующие компоненты логического и объектного уровней многоуровневой КЦ СТУС, могут быть наследованы от представленного на рисунке 6.34 класса *IMouseInterface* для реализации одного или нескольких методов обработки действия пользователя с компонентом. Пример реализации класса компонента с описанными методами интерфейсного класса *IMouseInterface*, приведен в приложении 13.

6.7.5 Интерфейс компонентов с переменным числом связей

Ряд компонентов объектного и логического уровней многоуровневой КЦ СТУС могут обладать переменным числом связей. При организации связи с одним из его выводов, находящимся с определенной стороны компонента, должен появляться новый свободный вывод, предназначенный для организации новой связи. При удалении связи с некоторым выводом компонента он должен быть удален, а все последующие выводы должны быть сдвинуты на одну позицию. При добавлении и удалении вывода должен изменяться размер изображения данного компонента на слое, на которого производится эти операции. Для реализации данных функций такой компонент, называемый *компонентом с переменным числом связей*, должен наследоваться от интерфейсного класса *IVarWiresInterface*,

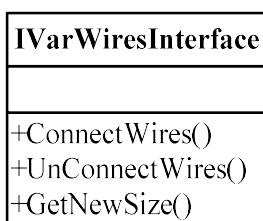


Рисунок 6.35 –
Интерфейсный класс
компонентов с
переменным числом
связей

представленный на рисунке 6.35 и содержащий следующие

методы:

- *ConnectWires()* – метод, срабатывающий в случае присоединения связи к выводу, номер которого передается в качестве аргумента данного метода;

- *UnConnectWires()* – метод обработки события удаления связи от вывода, номер которого является аргументом метода;

– *SetNewSize()* – метод вычисления и установки нового размера изображения компонента на слое с добавленным или удаленным выводом.

В данном пункте описаны интерфейсные методы, от которых может быть наследован класс определенного компонента, реализованного в рамках БМК комплекса программ моделирования ХТС, а также их основные методы, которые могут быть перегружены в определенных классах компонентов. Классы компонентов реализуются на языке объектно-ориентированного программирования C++ [141]. Для автоматизации их создания в комплекс программ включен генератор моделей компонентов (ГМК) [191], который на основе шаблонов классов, их методов и информации, введенной пользователем в интерфейсные окна ГМК, позволяет получить программный код компонента.

6.8. Генератор моделей компонентов

Генератор моделей компонентов [191] представляет собой отдельный программный модуль, входящий в комплекс программ моделирования ХТС, структура которого представлена на рисунке 6.1. ГМК предназначен для формирования программно-алгоритмического вида компонентов с последующей его компиляцией на языке Visual C++. Он позволяет на своих вкладках формировать информацию о классе компонента, использующуюся при формировании его программно-алгоритмического вида. Для его формирования разработаны специальные шаблоны *заголовочного* (*.h) и *исполнительного файлов* (*.cpp), в которых прописаны шаблоны общих для всех классов методов, а также шаблоны методов интерфейсных классов, которые приведены в приложении 14. В зависимости от назначения и функциональных задач компонента они могут быть включены в его программно-алгоритмическую реализацию. В них *тегами*, то есть специальными метками, заключенными в угловые скобки < >, помечены места заполнения определенных данных о компоненте.

В приложении 15 приведено руководство пользователя ГМК, в котором приведено описание всех его вкладок и информации, которая должна быть введена в каждую из них при формировании программно-алгоритмического вида соответствующего компонента.

6.9. Интерактивные панели

6.9.1 Интерактивная схемная панель

Интерактивная схемная панель (ИСП) – это макрокомпонент, состоящий из совокупности подцепей объектного и (или) логического уровней многоуровневой КЦ СТУС, сформированный в формате языка моделирования ХТС (смотри пункт 2.4), языка МАК, описанных в пункте 2.5, и визуальных компонентов языка ВИП, классификация которых представлена в пункте 2.6. С формальной точки зрения ИСП представляет собой подцепь вида

$$C^{ICP} = (K_V^{ICP} \cup C_L^{ICP} \cup C_C^{ICP}), \quad (6.27)$$

где K_V^{ICP} – совокупность визуальных компонентов множества K_V (2.68), образующая визуальный уровень ИСП, который может быть обособлен в отдельное интерфейсное окно, с которым взаимодействует пользователь при проведении экспериментов; C_L^{ICP} – алгоритмическая компонентная подцепь вида (2.63), представляющие собой сценарии функционирования ИСП; K_C^{IPC} – подцепь объектного уровня МКЦ (2.1), описывающие протекающие в формируемом с помощью ИСП процессы.

Компонент, создаваемый с помощью интерактивной схемной панели, может быть отображен на объектном или на логическом слое редактора. Если он должен находиться на объектном слое, то компонент «Интерфейс», представленный на рисунке 5.4, должен находиться на этом слое и иметь связи с компонентами языка моделирования ХТС. Когда с помощью ИСП создается компонент логического уровня, компонент «Интерфейс» располагается на этом уровне многоуровневой КЦ и соединяется с алгоритмическими компонентами множества K_A (2.63).

Создаваемый таким образом компонент может содержать самостоятельное окно, в котором осуществляется визуализация протекающих в таком компоненте процессов. В этом случае в параметрах компонента «Интерфейс» указывается слой, подлежащий отображению в таком окне.

6.9.2 Интерактивная математическая панель

Интерактивная математическая панель (ИМП) [168] – это компонент объектного или логического уровня МКЦ СТУС, предназначенный для тестирования и отладки новых моделей компонентов, представленных уравнениями или математическими выражениями относительно переменных связей компонента [201]. Математическая или математико-алгоритмическая модель ИМП формируется пользователем в редакторе математико-алгоритмических конструкций. В случае ИМП объектного уровня ММ данного компонента представляет собой совокупность алгебро-дифференциальных уравнений (2.9)-(2.12), записанных относительно всех переменных его связей. Для ИМП логического уровня в редакторе математико-алгоритмических конструкций формируются выражения, представленные в явном виде относительно единственной выходной связи данного компонента.

Интерактивная математическая панель объектного уровня. Для экспериментирования с математическими моделями компонентов ХТС, которые требуется добавить в БМК, в её составе реализована ИМП объектного уровня. Её

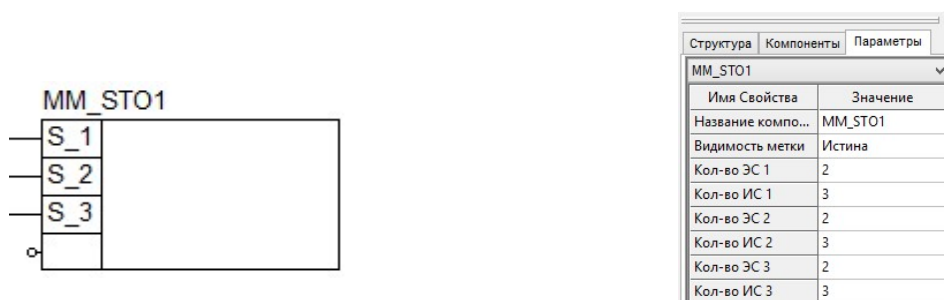


Рисунок 6.36 – Структура интерактивной математической панели и ее взаимосвязь с другими модулями комплекса программ МКМ СТУС

структура, оснащенная редактором математических выражений, и взаимосвязь с другими моделями комплекса программ представлена на рисунке 6.36. При формировании модели СТУС в многослойном редакторе КМ интерактивная математическая панель

тическая панель объектного уровня представляет собой расположенный на объектном слое редактора компонент с динамически изменяющимся числом выводов для организации связей с другими компонентами.

Являясь потомком приведенного на рисунке 6.35 интерфейсного класса *IVarWiresInterface*, данный компонент всегда будет иметь один пустой, не подсоединенный, вывод. При образовании связи с ним у ИМП образуется новый вывод без ограничения на число его связей. Вид интерактивной математической панели на объектном уровне многоуровневой КЦ СТУС приведен на рисунке 6.37, а.



а) б)
Рисунок 6.37 – Компонент «Интерактивная математическая панель»
а) вид компонента на объектном слое редактора;
б) таблица свойств компонента

Все связи ИМП объектного уровня многоуровневой КЦ СТУС являются неоднородными векторными (3.4) и для каждой из них в свойствах компонента, представленных на рисунке 6.37, б, указывается: N_e – количество энергетических связей; N_i – количество информационных связей.

Узел	Связь	Потенц. переменная	Поток. переменная	Направление
Связь 1	ES_1	P1	G	К компоненту
Связь 1	ES_2	T1	Q	К компоненту
Связь 1	IS_1	C11	---	
Связь 1	IS_2	C12	---	
Связь 1	IS_3	C13	---	
Связь 2	ES_1	P2	G	От компонента
Связь 2	ES_2	T2	Q	От компонента
Связь 2	IS_1	C21	---	
Связь 2	IS_2	C22	---	

Рисунок 6.38 – Таблица формирования физического координатного базиса компонента ИМП

Физический координатный базис компонента ИМП формируется пользователем в таблице «Физический КБ ИМП», которая представлена на рисунке 6.38 и открывается при активизации пункта «Показать переменные» в сплывающем меню компонента.

Для каждой энергетической связи ES_i , включенной в состав любой НВС

ИМП «Связь j », должны быть заданы имена потенциальной и потоковой переменных, с которыми они входят в математическую модель формируемого с помощью ИМП компонента. Каждая информационная связь IS_k , включенная в состав любой НВС ИМП «Связь j », представлена именем потенциальной переменной. Имена потоковых и потенциальных переменных формируются по правилу (4.16) языка МАК в столбце «Потенц. переменная». По этому же правилу в столбце «Поток. переменная» задается имя потоковой переменной. Для каждой энергетической связи в столбце «Направление» задается направление потока энергии – «от узла к компоненту» или «от компонента к узлу».

Математическая модель компонента ИМП формируется в редакторе математико-алгоритмических конструкций, который приведен на

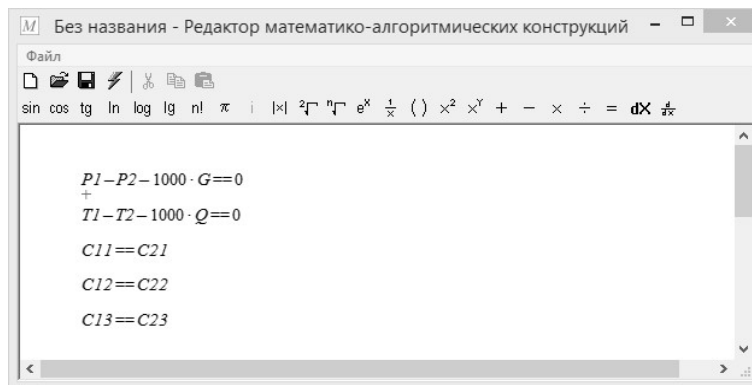


Рисунок 6.39 – Редактор математико-алгоритмических конструкций

математико-алгоритмических конструкций, который приведен на рисунке 6.39. Она представляет собой совокупность алгебро-дифференциальных уравнений, составленную относительно всех переменных связей ИМП, заданных своими именами в физическом КБ компонента.

После того, как закрывается редактор математико-алгоритмических конструкций, в котором формируются уравнения математической модели ИМП, они передаются интерпретатору, который формирует алгоритмическую компонентную цепь. Её пример для совокупности уравнений, приведенной на рисунке 6.39, представлен на рисунке 6.40. После того, как алгоритмическая КЦ была сформирована в локальном КЦ математической подцепи, производится ее преобразование в глобальный КБ компонентной цепи, в которую включена ИМП.

На этапе опроса топологического блока ИМП формируются пары значений

$(\pm b_j, n_j)$ для всех энергетических связей, входящих в состав любой НВС.

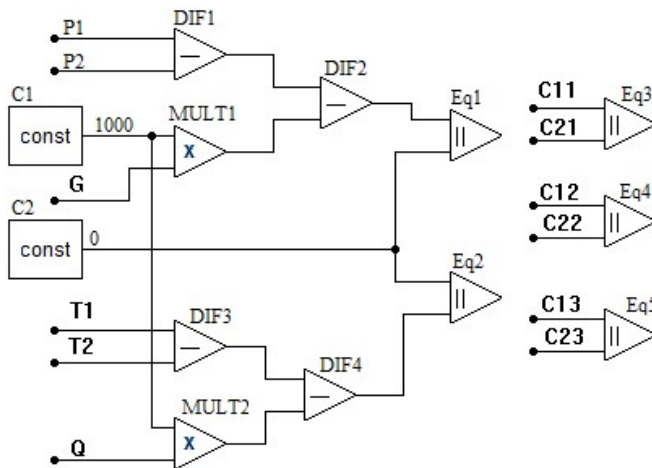


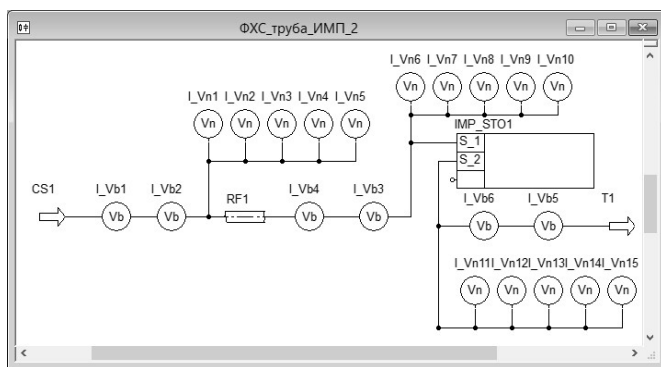
Рисунок 6.40 – Алгоритмическая компонентная цепь системы уравнений ИМП

Этап опроса вычислительного блока заключается в формировании уравнений всех компонентов математической подцепи, сформированной интерпретатором на основе уравнений, введенных пользователем в редактор математико-алгоритмических

конструкций.

Результат анализа представленной на рисунке 6.41 многокомпонентной гидравлической системы, состоящей из двух последовательно соединенных труб, одна из которых представлена компонентом ИМП, приведен на рисунке 6.37.

Равномерное распределение давлений и температур, а также равенство концентраций веществ во всех измеряемых точках подтверждают правильность анализа представленной гидравлической компонентной цепи.



ФХС_труба_ИМП_2		
В точке источника	Между трубами	В точке терминатора
P 1	P 2	P 3
2000.000	1000.000	0.000
T 1	T 2	T 3
373.000	323.000	273.000
C1 нефть	C2 нефть	C3 нефть
0.400	0.400	0.400
C1 газ	C2 газ	C3 газ
0.400	0.400	0.400
C1 вода	C2 вода	C3 вода
0.200	0.200	0.200
G 1	G 2	G 3
1.000	1.000	1.000
Q 1	Q 2	Q 3
0.050	0.050	0.050

а)

б)

Рисунок 6.41 – Использование ИМП при моделировании ХТС

а) компьютерная модель
б) результаты моделирования

Интерактивная математико-алгоритмическая панель. Интерактивная математико-алгоритмическая панель (ИМАП) логического уровня позволяет с использованием правил грамматики языка МАК (4.23)-(4.28) сформировать совокупность выражений, включая циклы, из нескольких операций данного языка (4.5)-(4.22). Введенные в редактор выражения предназначены для вычисления результата, значения которого выводится из ИМАП по ее выходной связи. В состав интерактивной математико-алгоритмической панели, представленный на рисунке 6.42, включены следующие программно-алгоритмические блоки:



Рисунок 6.42 – Внутренняя структура интерактивной математической панели логического уровня

- редактор математико-алгоритмических конструкций, позволяющий вводить и редактировать выражения на языке МАК, который описан в главе 4;
- интерпретатор математико-алгоритмических конструкций, осуществляющий построение алгоритмической КЦ (2.63) выражений, введенных в редакторе;
- имитационное ядро, которое на основе алгоритма передачи сообщений, описанного в пункте 6.6.2, осуществляет вычисление результата введенного в редактор выражения.

ИМАП логического уровня представляет собой компонент, изображенный на рисунке 6.43, с переменным числом входных связей

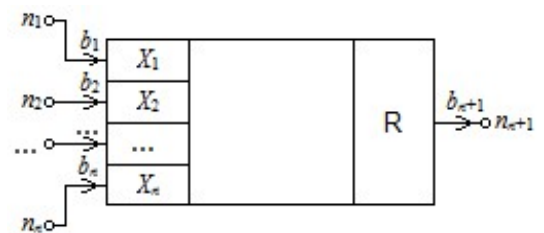


Рисунок 6.43 – Компонент «Интерактивная математико-алгоритмическая панель» логического уровня

$$S_i = (b_i \eta^{-n_j}) \rightarrow \{X_i\}, \quad (6.28)$$

где X_i – входная переменная с уникальным в рамках компонента именем, представленным в совокупности математико-алгоритмических конструкций, формируемых пользователем в редакторе. При присоединении к свободному узлу компонента ИМАП, находящемуся слева от компонента, некоторой связи другого компонента, у ИМАП автоматически появляется новый свободный узел.

Каждому узлу в свойствах компонента соответствует атрибут «Имя X_i », с помощью которого пользователем задается имя переменной, принимающей значения, поступающие по данной входной связи. Помимо этого в поле «Значение X_i » пользователь имеет возможность задавать первоначальное значение соответствующей переменной. Если значение не будет задано, оно при расчетах будет приниматься равным 0. К связям компонента (6.28) могут быть подключены информационные выходы представленных в таблице 2.1 измерительных компонентов языка моделирования ХТС, представленные на рисунках 4.2–4.6 источники значений числовых типов и другие компоненты-преобразователи языка МАК, результатом работы которых данных числовых типов, а также компоненты-регуляторы множества K_{V3} (2.68) языка ВИП.

Формирование выражений осуществляется в формате языка МАК (смотри главу 4) в представленном на рисунке 6.44 редакторе математико-алгоритмических конструкций, который открывается двойным щелчком мыши по

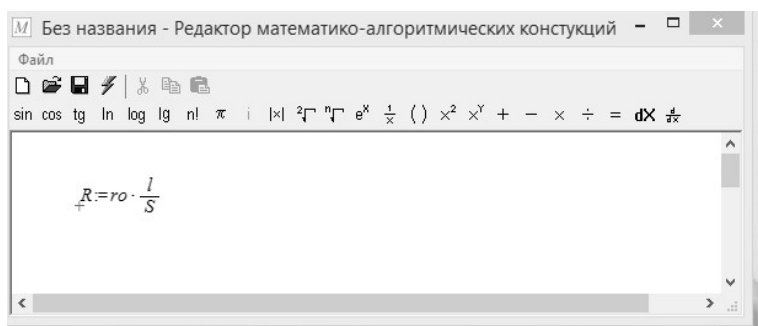


Рисунок 6.44 – Редактор математико-алгоритмических конструкций

изображению ИМАП в логическом слое редактора.

При закрытии редактора сформированное в нем выражение передается интерпретатору языка МАК, описанному в пункте 6.5, который на основе системы

отображения математико-алгоритмических конструкций [101] формирует алгоритмическую КЦ, пример которой представлен на рисунке 6.45.

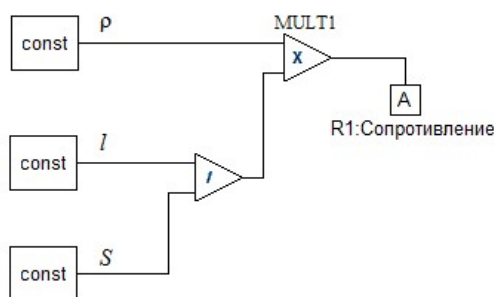


Рисунок 6.45 – Алгоритмическая компонентная цепь записанного в редакторе выражения

По ней имитационным ядром, функционирующим на основе алгоритма передаче сообщений [116], осуществляется вычисление результата, передаваемого другим компонентам по выходной связи

$$S_{i+1} = (b_{i+1} \eta^+ n_{i+1}) \rightarrow \{R\}. \quad (6.29)$$

Расчет введенного выражения осуществляется как в начале эксперимента на основе значений, заданных в свойствах ИМАП в качестве начальных, так и при получении сообщения на любой из входных выводов компонента ИМАП (6.28).

Таким образом, ИМАП логического уровня МКЦ СТУС представляет собой компонент, математико-алгоритмическая модель которого формируется на языке МАК и может содержать несколько его операций. Являясь компонентом алгоритмических компонентных цепей, она располагается на логическом слое редактора и может быть применена для первичного расчета значений компонентов исследуемой модели ХТС, обработки результатов измерения и моделирования, а также для построения сложных математико-алгоритмических конструкций. В них могут быть включены условные операторы, циклы и вызовы функций. Такая модель может быть использована для формирования блоков обработки результатов натурального или вычислительного экспериментов [202].

6.10. Выводы

1. Рассмотрено назначение и сформирована структура комплекса программ «Среда компьютерного моделирования химико-технологических систем», в основе разработки которого лежит метод компонентных цепей, развитый в данной работе для компьютерного моделирования и исследования сложных технических управляемых систем предприятий газовой промышленности.

2. Определены опции многослойного графического редактора, позволяющие формировать в нем многоуровневые компонентные цепи СТУС на трех своих взаимосвязанных слоях многослойного редактора, каждый из которых со-

ответствует определенному уровню МКЦ, представленному в графических нотациях определенного подязыка языка многоуровневого компьютерного моделирования.

3. Разработан интерпретатор языка моделирования ХТС, входной информацией которого является совокупность компонентов, расположенных на объектном слое редактора и связей между ними, выходной – компонентная цепь, состоящая из компонентов языка моделирования ХТС. Согласно алгоритму формирования компонентной цепи исследуемого объекта с НВС каждой потенциальной и потоковой переменной всех компонентов, входящих в формируемую КЦ, ставятся в соответствие определенные номера в глобальном координатном базисе составляемой КЦ. В основу работы данного алгоритма положены правила коммутации компонентов с неоднородными векторными связями, сформулированные в третьей главе.

4. Предложена и реализована объектно-ориентированная структура универсального вычислительного ядра, предназначенная для выполнения одновариантного анализа компьютерных моделей управляемых химико-технологических систем, представленных в формате метода компонентных цепей с неоднородными векторными связями.

5. Разработаны методики и алгоритмы повышения быстродействия вычислительного эксперимента, проводимого над моделями ХТС без потерь адекватности получаемых результатов. В их основе лежит принцип формирования уравнений на основе их классификации и алгоритмы явно-неявного анализа, предусматривающего непосредственное разрешение уравнений относительно входящих в них единственной переменной, значение которой не определено на данной итерации.

6. Спроектирован и создан интерпретатор математико-алгоритмических конструкций, задачей которого является построение алгоритмических КЦ выражений алгоритмов исследования и функционального проектирования СТУС, представленных на логическом уровне многоуровневой компьютерной модели в

формате описанного в главе 4 языка моделирования алгоритмических конструкций.

7. Предложена универсальная структура передаваемого сообщения и разработан алгоритм передачи сообщений, на базе которых реализовано имитационное ядро. Оно обеспечивает механизм обмена данными различных типов между компонентами алгоритмической КЦ логического уровня многоуровневой КМ СТУС, сформированной в формате графического языка моделирования алгоритмических конструкций, с использованием ее узловой матрицы. Разработана структура алгоритмических компонентов и реализована их унифицированная взаимосвязь с универсальным имитационным ядром.

8. Сформирована универсальная структура и на ее основе создана библиотека моделей компонентов, обеспечивающая хранение компонентов трех подязыков языка многоуровневых компонентных цепей: языка моделирования ХТС, языка моделирования алгоритмических конструкций и языка виртуальных инструментов и приборов. В интерфейсных классах реализованы методы взаимодействия компонентов с универсальным вычислительным и имитационным ядрами, а также методы взаимодействия пользователя с визуальными компонентами во время выполнения экспериментов над МКЦ СТУС.

9. Спроектирован и реализован генератор моделей компонентов, представляющий собой отдельный модуль комплекса программ и позволяющий формировать программно-алгоритмический вид компонентов для компиляции на объектно-ориентированном языке программирования и их добавления в БМК.

10. Разработана интерактивная схемная панель, позволяющая обособливать многоуровневую компонентную подцепь в отдельный макрокомпонент с последующим его использованием в объектном или логическом уровнях многоуровневой КЦ СТУС.

11. Созданы интерактивные математические панели объектного и логического уровней, оснащенные редактором математико-алгоритмических выражений и позволяющие формировать модели компонентов в этом редакторе относительно переменных в физическом координатном базисе. ИМП объектного уровня пред-

назначена для быстрого формирования компонентов ХТС с неоднородными векторными связями, относительно переменных которых формируется модель в виде системы алгебро-дифференциальных уравнений. Модель ИМАП логического уровня формируется пользователем в редакторе на языке моделирования алгоритмических конструкций, рассмотренном в главе 4 диссертации.

Глава 7. Построение многоуровневых компьютерных моделей сложных технических управляемых систем для проведения вычислительных экспериментов

В данной главе рассматриваются алгоритм решения задач исследования и функционального проектирования СТУС на базе ХТС, в которых циркулируют одно- и многокомпонентные вещественные потоки, управляемые путем изменения давления и температуры, а также принципы формирования их многоуровневых компонентных цепей. В таких системах управляющее воздействие,работанное с помощью компьютерной модели, может быть применено к реальному объекту управления автоматически или быть рекомендательным для управляющего персонала. В последнем случае такая многоуровневая компонентная цепь СТУС позволит реализовать интеллектуальные системы управления.

7.1 Алгоритм решения задач исследования и функционального проектирования СТУС

В основе решения задач исследования и функционального проектирования сложных технических управляемых систем лежит вычислительный эксперимент, алгоритм которого представлен на рисунке 7.1. Он состоит из следующих шагов [203]:

Шаг 1 (Ш1). Формирование компонентной цепи ХТС объектного уровня. Осуществляется вызов интерпретатора языка моделирования ХТС, который на основе компьютерной модели химико-технологических систем, формируемой из компонентов языка моделирования ХТС, описанных в главе 3, на объектном уровне многоуровневой КЦ СТУС, по алгоритму формирования КЦ ХТС, описанному в пункте 6.3.2, формирует компонентную цепь исследуемой химико-технологической системы вида (2.3), каждый компонент которой обладает неоднородными векторными связями (3.4).

Шаг 2. (Ш2). Формирование компонентной цепи сценария исследования и функционирования СТУС. Производится вызов интерпретатора языка

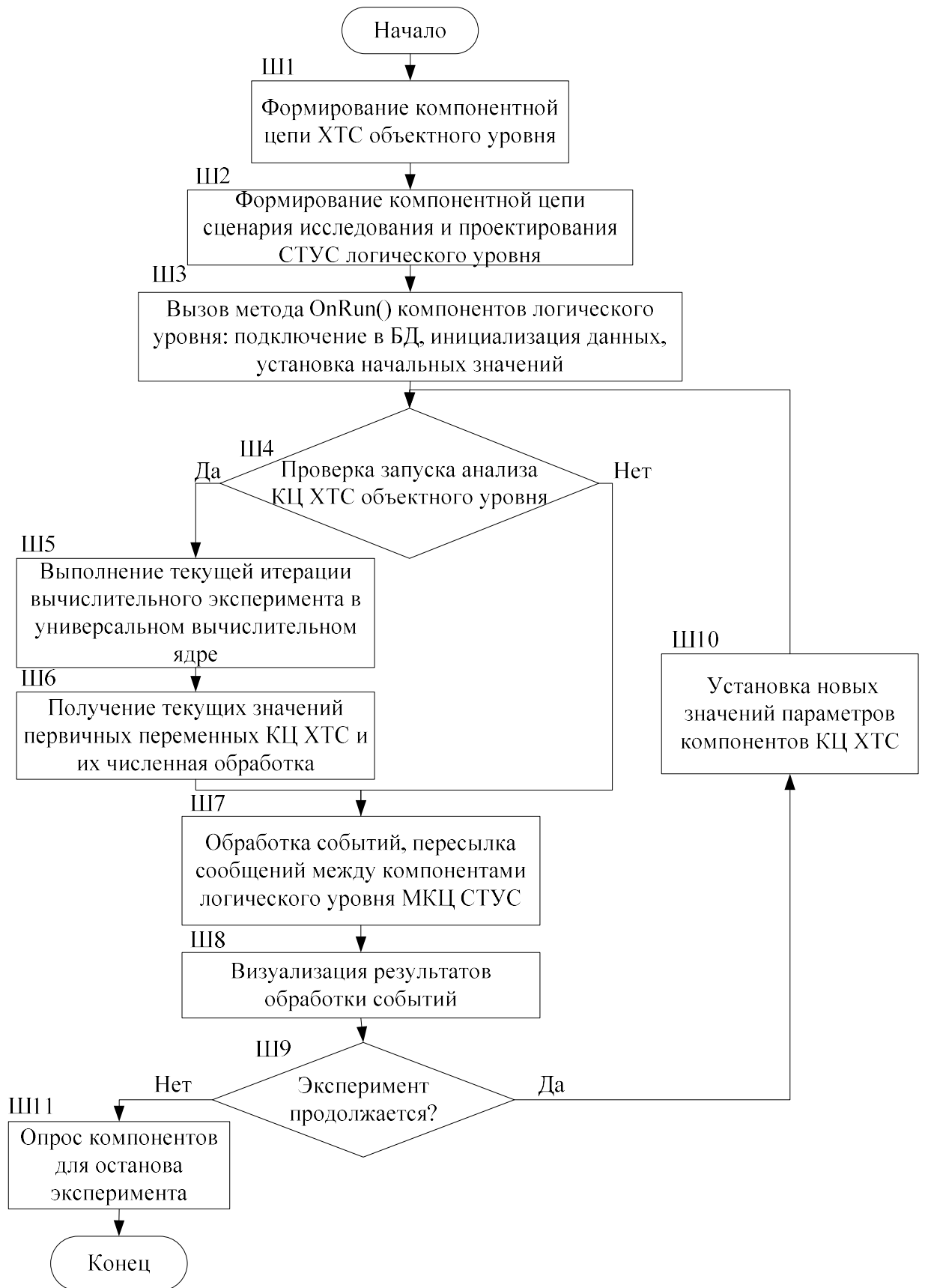


Рисунок 7.1 – Алгоритм решения задач исследования и функционального проектирования СТУС

МАК, осуществляющего формирование алгоритмической КЦ сценария исследования и функционального проектирования СТУС, представленного алгоритмической подцепью логического уровня МКЦ. Результатом работы интерпретатора является сформированная алгоритмическая КЦ (2.63) и соответствующая ей матрица инцидентности, построенная по правилу (6.22) и лежащая в основе функционирования алгоритма передачи сообщений имитационного ядра.

Шаг 3 (Ш3). Вызов метода *OnRun()* компонентов логического уровня МКЦ. На этом шаге в различных компонентах логической подцепи МКЦ, у которых реализован метод *OnRun()*, производится подключение к базам данных, инициализация структур данных и выделение под них оперативной памяти, представленных в виде массивов и матриц, а также расчет и установка начальных значений параметров компонентов КЦ ХТС, сформированной на объектном уровне МКЦ. Он осуществляется компонентами её логического уровня на основе выставленных значений компонентов-регуляторов, находящихся на визуальном уровне МКЦ.

Шаг 4 (Ш4). Проверка запуска анализа КЦ ХТС объектного уровня. Данный шаг является первым в итерационном процессе алгоритма решения задач исследования и функционального проектирования СТУС. На нём осуществляется проверка, была ли запущена на анализ компонентная цепь химико-технологической системы, принадлежащая объектному уровню МКЦ СТУС. Если ее запуск во время выполнения эксперимента состоялся, то производится переход к шагу 5. Если вычислительное ядро не было запущено или запущенный ранее режим анализа к этому шагу завершен, то осуществляется переход к шагу 7.

Шаг 5 (Ш5). Выполнение текущей итерации вычислительного эксперимента. В этом случае осуществляется одна итерация алгоритма вычислительного эксперимента над компьютерной моделью ХТС, описанного в пункте 6.4.2.1.

Шаг 6 (Ш6). Получение первичных значений переменных КЦ ХТС и их численная обработка. Выборка первичных значений, подлежащих передаче с объектного уровня МКЦ СТУС на ее логический уровень, производится измерительными компонентами, реализующими межуровневые взаимодействия (2.50)и

описанными в пункте 3.4.2. Численная обработка результатов осуществляется компонентами БОД, обобщенный компонент которых приведен на рисунке 4.14, с использованием других компонентов языка МАК, приведенных в приложении 7.

Шаг 7 (Ш7). Обработка событий, пересылка сообщений между компонентами логического уровня МКЦ СТУС. На данном шаге осуществляется обработка событий, поступивших в МКЦ от внешних программных и аппаратных модулей, действий пользователя с компонентами регуляторами K_{V3} , входящими во множество визуальных компонентов K_V (2.68) и реализующими межуровневое взаимодействие (2.52). Функционирование данного шага осуществляется имитационным ядром, в основе которого лежит представленный на рисунке 6.31 алгоритм передачи сообщений. Данный шаг заканчивает работу после того, как отрабатывают все цепочки алгоритмических подцепей логического уровня МКЦ, на которые поступили результаты моделирования КЦ ХТС объектного уровня и события действий пользователя с компонентами визуального уровня. При выполнении данного шага осуществляется расчет значений, подлежащих визуализации, а также расчет новых значений параметров компонентов КЦ ХТС, являющейся подцепью объектного уровня МКЦ.

Шаг 8 (Ш8). Визуализация результатов обработки сообщений. Получив сообщения от компонентов логического уровня МКЦ во время выполнения шага 7, компоненты-визуализаторы множества K_{V2} из (2.68) реализуют межуровневое взаимодействие (2.51) и осуществляют преобразование поступившей к ним в сообщениях информации к графическому выводу. Также они осуществляют её вывод на экран с помощью графических функций класса *CPaintDC* [200] в виде числовых значений, временных, частотных или параметрических зависимостей.

Шаг 9 (Ш9). Проверка продолжения эксперимента. На данном шаге производится анализ того, нажата ли кнопка «СТАРТ» пользователем на панели инструментов редактора многоуровневой компонентной цепи. Если она нажата, то производится переход к шагу 10. В противном случае производится переход к шагу 11.

Шаг 10 (Ш10). Установка новых значений компонентов КЦ ХТС. С помощью компонентов-атрибутов, приведенных на рисунке 4.34, производится параметризация компонентов объектного уровня КЦ ХТС значениями, которые были рассчитаны на шаге 7, реализуя межуровневые взаимодействия (2.48)

Шаг 11 (Ш11). Опрос компонентов для останова эксперимента. На данном шаге производится освобождение памяти под динамические структуры данных, которая была выделена под них на шаге 3. В случае использования компонентов подготовки документации производится передача накопленной ими информации в систему документирования, которая формирует итоговый документ о проведении эксперимента.

На основе описанного алгоритма может быть автоматизировано решение различных задач исследования и функционального проектирования СТУС, требующих многократного анализа КЦ химико-технологических систем при различных значениях параметров их компонентов. К таким задачам относятся различные виды многовариантного анализа и параметрического синтеза, в основе которых лежат методы оптимизации, а также задачи проведения практических и лабораторных занятий с помощью компьютерных тренажеров, реализованных на базе многоуровневых компонентных цепей.

7.2 Разработка и отладка сценариев управления на компьютерных моделях химико-технологических систем

Предложенная структура многоуровневой компонентной цепи, представленная на рисунке 2.5, открывает возможности отладки сценариев управления, представленных своими имитационными моделями на логическом уровне многоуровневой КЦ. Их основной задачей является анализ результатов моделирования ХТС, представленных значениями наблюдаемых переменных, и выработка на их основе соответствующих управляющих воздействий на модели ИспУ. При решении данных задач помимо компонентов ХТС в подцепь объектного уровня МКЦ входят компоненты исполнительных и измерительных устройств.

Рассмотрим процесс формирования и отладки сценариев управления на примере многоуровневой компонентной цепи поддержания давления газа в баллоне (газохранилище) [204]. Пусть изначально баллон пустой. Он заполняется по входной линии, в которой газовый поток проходит через клапан С11. Выходная линия доставляет газ потребителю. Она управляется клапаном С12. Доля открытия каждого из клапанов изменяется в пределах от закрытой до полностью открытой соответствующими сервоприводами, работающими по апериодическому закону с коэффициентом усиления $K = 1$ и варьируемой постоянной времени [205]. Изменение состояния обоих клапанов производится по сигналам, когда давление газа в баллоне достигает определенного порогового давления. Пусть имеется два пороговых значения:

- $P_1 = d_1 \cdot P_{\max}$, при котором должен начинаться закрываться клапан С12 и открываться клапан С11;
- $P_2 = d_2 \cdot P_{\max}$, являющееся началом закрытия входного клапана С11 и открытия выходного клапана С12.

Основной задачей отладки сценариев управления является нахождение таких пороговых значений d_1 и d_2 (где $d_i \in [0,1]$, $i = 1, 2$), при которых давление газа в баллоне не будет превышать максимального значения P_{\max} , а также будет соблюдаться режим бесперебойного снабжения газом потребителей. То есть давление газа не должно стать равным 0.

На рисунке 7.1 представлена объектная подцепь МКЦ системы поддержания газа в баллоне. Помимо этого компонентами апериодических звеньев AZ1 и AZ2, входящие в библиотеку моделей компонентов структурных блоков САУ [22], представлены модели электромеханических приводов, подключенных к вентилям клапанов С11 и С12 и управляющих степенью их открытия. Применяемые электромеханические привода [206], осуществляющие управление долей открытия клапанов, могут быть представлены апериодическими звеньями первого порядка AZ1 и AZ2 с заданной постоянной времени T и единичным коэффициентом передачи $K = 1$. При этом постоянные времени приводов могут быть в общем

случае различными. С помощью измерительных компонентов I_Vn1 и I_Vn2 степень открытия клапанов на логическом уровне многоуровневой КЦ передается атрибуту «Доля открытия клапана» соответствующих клапанов.

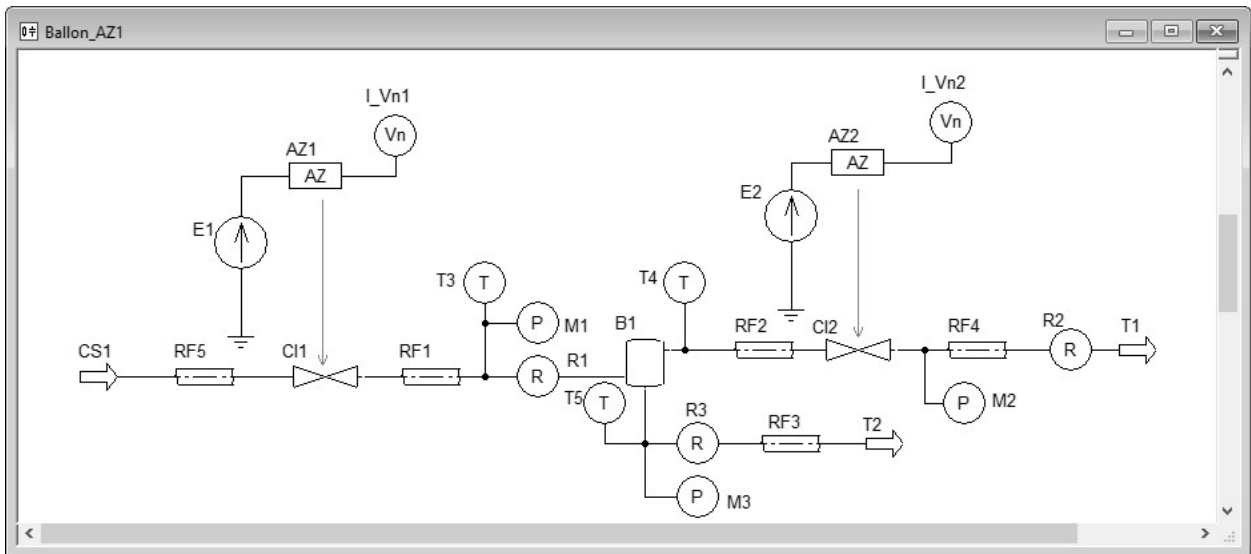


Рисунок 7.2 – Компьютерная модель системы поддержания давления газа в баллоне на объектном уровне многоуровневой КМ

В КЦ объектного уровня МКЦ системы поддержания давления газа в баллоне входят следующие измерительные компоненты: манометры M1, M2 и M3, осуществляющие давление газа на входе, выходе и внутри баллона соответственно; T3, T4 и T4 – термометры, измеряющие температуру в соответствующих точках модели системы; R1, R2 – расходомеры, передающие на логический уровень многоуровневой КЦ соответствующие текущие скорости расходов газа; расходомер R3 – для измерения текущего количества газа в баллоне.

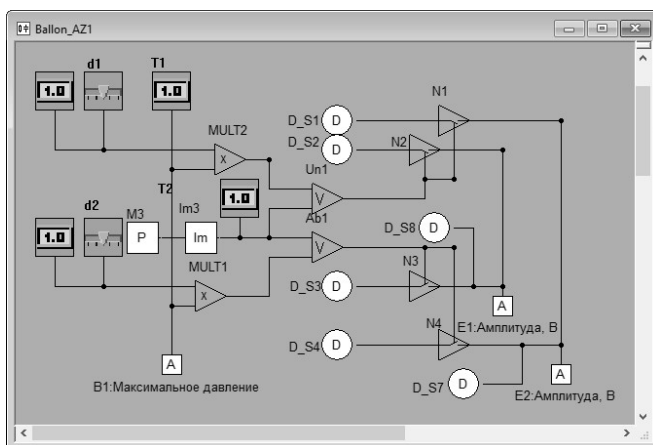


Рисунок 7.3 – Алгоритмическая компонентная цепь функционирования устройства управления поддержания давления газа в баллоне

расходомер R3 – для измерения текущего количества газа в баллоне.

На логическом уровне многоуровневой КЦ с использованием значения текущего давления газа, передаваемого с объектного слоя манометром M3, построена алгоритмическая компонентная подцепь сценария функционирования устройства управления, представленная на рисунке 7.3. Его за-

дачей является поддержание давления газа в баллоне в заданных пределах. Это значение, с учетом заданной инерционности клапанов, не должно превышать максимального значения, для предотвращения аварийной ситуации в баллоне, а также не должно быть меньше минимального с целью бесперебойной поставки газа потребителям.

Минимальное значение давления газа в баллоне, при котором должен начать закрываться выходной клапан и одновременно открываться входной, определяется умножением максимального значения давления на пороговое значение, задаваемое регулятором $d1$. Эта операция осуществляется компонентом-умножителем MULT2. Если текущее значение давления газа в баллоне меньше полученного значения от источника D_S1 на компонент-атрибут «E1: Амплитуда, В» подается значение 0, а от источника D_S2 на компонент-атрибут «E2: Амплитуда, В» – значение 1. При этом начинается закрытие выходного клапана и открытие входного.

Когда текущее значение давления в газе становится выше значения, задаваемого регулятором $d2$, что проверяется с помощью компонента Ab1, производится закрытие выходного клапана и открытие входного. Для этого от источника D_S3 на компонент «E1: Амплитуда, В» подается значение 1 и от источника D_S4 на компонент-атрибут «E2: Амплитуда, В» – значение 0. Это является началом закрытия входного и открытия выходного клапана.

Визуализация процесса управления состоянием баллона и клапанов осуществляется на визуальном уровне многоуровневой КЦ, представленного на рисунке 7.4. На нем с помощью графического компонента V1 изменением цвета отображается степень заполнения баллона. Аналогично изменение цвета индикаторов C11 и C12, а также с помощью регуляторов «Клапан входной» и «Клапан выходной» осуществляется отображение текущего значения степени открытия клапанов. Соответствующими цифровыми табло в числовой форме визуализируются текущие значения переменных и параметров компьютерной модели.

С целью отладки сценариев управления пользователь имеет возможность изменения пороговых значений $d1$ и $d2$. Для этого на визуальном уровне преду-

смотрены соответствующие компоненты-регуляторы. Для безопасного и эффективного использования баллона (газохранилища) они должны быть установлены таким образом, чтобы при заданных постоянных времени сервоприводов, осу-

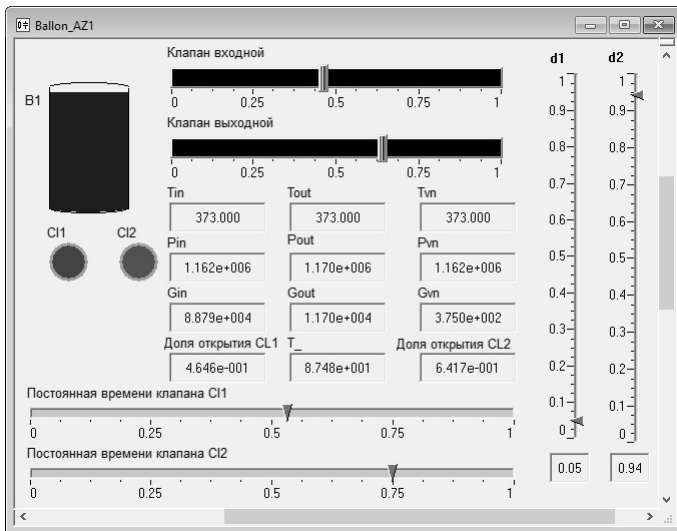


Рисунок 7.4 – Панель отображения результатов анализа системы поддержания давления газа в баллоне на визуальном уровне многоуровневой

ществляющих управление состоянием клапанов, баллон полностью не опустошался, а также давление в нем не превышало максимального значения, но достигало этого значения при заполнении баллона.

Регуляторами «Постоянная времени клапана C11» и «Постоянная времени клапана C12» пользователь имеет возможность варьировать постоянные времени аperiodических

звеньев AZ1 и AZ2, отражающих в модели привода управления состоянием входного и выходного клапанов соответственно.

7.3 Решение задачи проектирования теплообменного аппарата

Многоуровневые компонентные цепи позволяют решать задачи функционального проектирования СТУС [65], когда по заданным входным данным необходимо определить значения параметров некоторого аппарата химической промышленности, при которых соответствующие выходные переменные будут иметь заданные значения. Решение таких задач основано на использовании алгоритмов параметрической оптимизации [63] при варьировании значений подлежащих определению параметров аппарата.

Пусть требуется определить минимальную длину теплообменного аппарата, в который под давлением 10 МПа объемным расходом 1 кмоль/с подаются два теплоносителя – горячий при температуре 353 К и холодный температурой 303 К. В качестве теплоносителей используется добытый природный газ, коэффициенты

для расчета теплоемкости для которого определяются по формуле (3.54). Диаметр теплообменника 0.2 м, коэффициент теплопередачи – 1600 Вт/(м²*К).

При решении данной задачи минимизации подлежит целевая функция вида

$$F(L) = (T_H - T_C)^2, \quad (7.1)$$

где T_H – температура горячего теплоносителя на выходе теплообменника длиной L , T_C – температура холодного теплоносителя на его выходе.

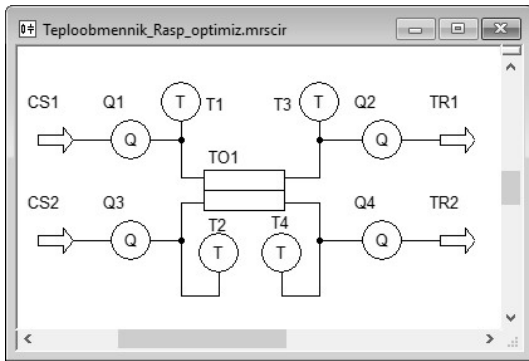


Рисунок 7.5 – Модель теплообменной секции установки комплексной подготовки газа

Решение данной задачи выполнено с помощью многоуровневой КЦ, на объектном уровне которой сформирована представленная на рисунке 7.5 подцепь теплообменной секции типовой установки комплексной подготовки газа [138], которая эксплуатируется на газовых и газоконденсатных месторождениях Российской Федерации. С помощью источника CS1 в теплообменник TO1 подается горячий теплоноситель, а источником CS2 – холодный теплоноситель.

Температуры теплоносителей на выходе теплообменника передаются с объектного уровня МКЦ на логический соответствующими термометрами: T3 измеряет температуру горячего теплоносителя, а T4 – холодного теплоносителя. Относительно передаваемых значений с помощью компонентов DIF1 и DEG1 осуществляется расчет целевой функции (7.1) при различных значениях длины теплообменника. Её значение передается компоненту-оптимизатору Optim1, который на основе метода покоординатного спуска, адаптированного

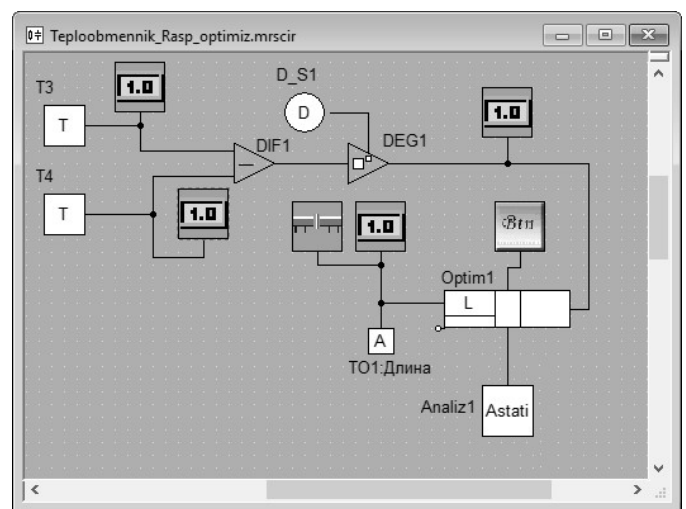


Рисунок 7.6 – Алгоритмическая компонентная цепь решения задачи поиска минимальной длины теплообменника

для минимизации целевых функций с одним аргументом, осуществляет поиск минимума целевой функции. Задав определенное значение длине теплообменника с помощью компонента-атрибута «ТО1: Длина», запускается нелинейный статический анализ КЦ объектного уровня. Для запуска очередной итерации её анализа, сформированной на объектном уровне многоуровневой КЦ, компонент-оптимизатор Optim1 передает команду запуска на нелинейный статический анализ компоненту Analiz1. Алгоритмическая компонентная подцепь, соответствующая описанному сценарию, приведена на рисунке 7.6.

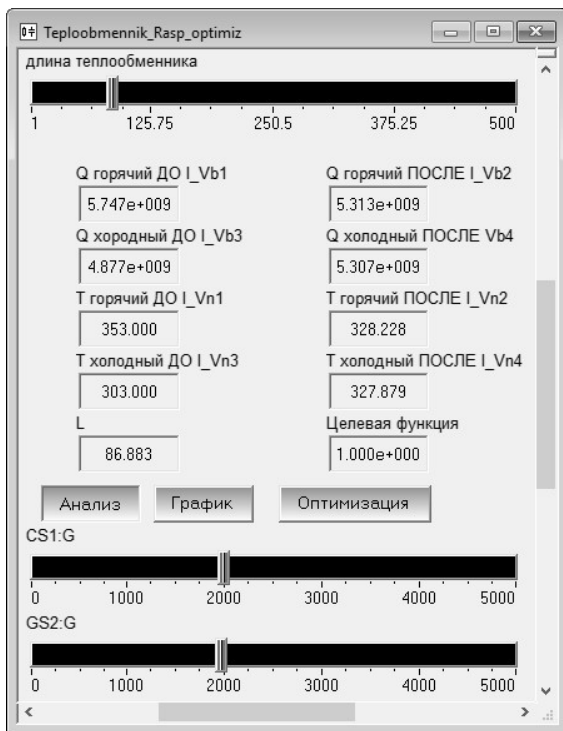


Рисунок 7.7 – Лицевая панель решения задачи определения минимальной длины теплообменного аппарата

На рисунке 7.7 представлена лицевая панель для решения задачи определения минимальной длины теплообменника, являющаяся визуальной подцепью МКЦ. С помощью регуляторов CS1:G и CS2:G устанавливаются молярные расходы горячего и холодного теплоносителей соответственно. При нажатии на кнопку СТАРТ запускается процесс минимизации, который определяет минимальную длину теплообменника 128,9 м., при которой будут наблюдаться равные температуры теплоносителей на выходе аппарата.

Аналогичную МКЦ можно применять для определения значений параметров различных элементов и аппаратов ХТС, удовлетворяющих некоторым условиям, заданным в виде целевых функций.

7.4 Определение точки росы газа путем проведения многовариантного анализа в многоуровневой компьютерной модели

Актуальной проблемой при транспортировке газа по трубопроводам является проблема уменьшения гидратообразования в газотранспортной системе. Для

снижения образования гидратов и поддержки пропускной способности газ в установках комплексной подготовки газа проходит ряд ступеней сепарации, а также осушку газа в абсорберах [39]. Наибольший эффект от сепарации газа может быть получен в случае, когда газ на входе сепаратора имеет температуру, соответствующую точке росы по углеводородам. При температуре точки росы, выполняется соотношение

$$\sum_{i=1}^N \frac{y_i}{K_i} = 1, \quad (7.2)$$

где y_i – концентрации компонентов в паровой фазе; K_i – константы фазового равновесия, определяемые путем численного решения уравнений Соава-Редлиха-Квонга [137].

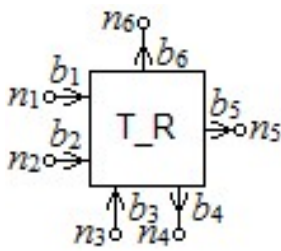


Рисунок 7.8 – Компонент «Определение точки росы»

Для определения температуры точки росы по углеводородам на основе обобщенного компонента БОР и компонентов многовариантного анализа, описанных в пункте 4.7 в рамках языка МАК разработан компонент «Определение точки росы», представленный на рисунке 7.8. Его связями являются:

$$- S_1 = (b_1 \eta^- n_1) \rightarrow \{ \mathbf{K} \} - \text{связь для получения вектора}$$

коэффициентов фазового равновесия всех компонент, действующих в рассматриваемой ХТС;

$$- S_2 = (b_2 \eta^- n_2) \rightarrow \{ \mathbf{y} \} - \text{связь для получения вектора концентраций компо-}$$

нент, составляющих многокомпонентные потоки рассматриваемой ХТС;

$$- S_3 = (b_3 \eta^- n_3) \rightarrow \{ Start \} - \text{связь для получения сигнала о начале поиска}$$

температуры точки росы;

$$- S_4 = (b_4 \eta^+ n_4) \rightarrow \{ Analiz \} - \text{связь для передачи сигнала о проведении оче-}$$

редней итерации анализа модели исследуемой ХТС;

– $S_5 = (b_5 \eta^+ n_5) \rightarrow \{T_T\}$ – связь для передачи текущей температуры соответствующему источнику многокомпонентного потока, установленного на входе рассматриваемого аппарата ХТС;

– $S_6 = (b_6 \eta^+ n_6) \rightarrow \{T_R\}$ – связь для отправки найденной температуры точки росы другим компонентам для использования найденного результата.

В параметрах компонента «Определение точки росы» задается начальная температура, от которой осуществляется старт многовариантного анализа по температуре. Алгоритм работы компонента начинается в момент, когда по связи S_3 придет сообщение. Текущее значение температуры передается источнику многокомпонентного потока или другому компоненту, задающему температуру потока, по связи S_5 . Сигнал о проведении очередной итерации анализа посылается компонентам одновариантного анализа по связи S_4 . Когда компонент по связям S_1 и S_2 получает вектора коэффициентов фазового равновесия и концентраций всех веществ в исследуемой ХТС, осуществляется расчет суммы, указанной в левой части соотношения (7.2). Если оно выполняется, то осуществляется повышение температуры до значения, при котором это соотношение станет меньше 1. Когда соотношение (7.2) не выполняется, что характерно для температур, превышающих точку росы по углеводородам, осуществляется понижение температуры и ее передача соответствующему источнику через связь S_5 . Итерационный процесс продолжается до тех пор, пока соотношение (7.2) не будет выполняться.

Максимальная температура, при которой выполняется соотношение (7.2), будет являться точкой росы T_{TR} по углеводородам. В качестве результата полученное значение T_{TR} будет передано по связи S_6 .

На основе КЦ секции низкотемпературной сепарации природного газа, представленной на рисунке 3.18, и приведенного на рисунке 7.8 компонента «Определение точки росы» разработана алгоритмическая компонентная цепь определения температуры точки росы газа при заданном его давлении на входе сепаратора. Она представлена на рисунке 7.9.

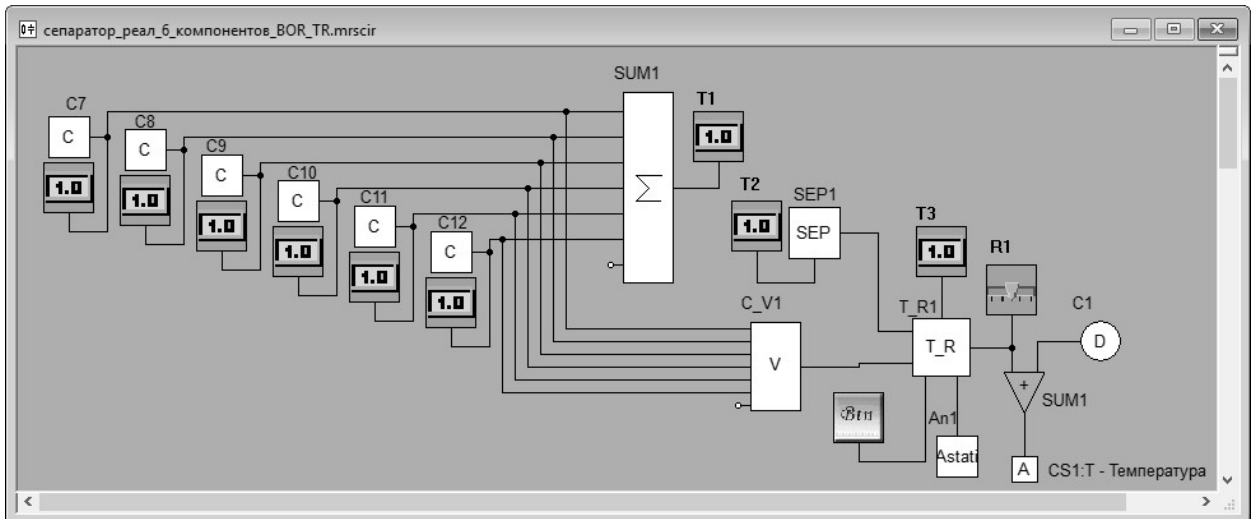


Рисунок 7.9 – Алгоритмическая компонентная цепь определения температуры точки росы

Компонентом-сумматором SUM1 осуществляется контроль суммы всех концентраций на выходе паровой фазы сепаратора. С помощью конструктора вектора C_V1 формируется вектор концентраций веществ, которые передаются с

объектного уровня многоуровневой КЦ на её логический уровень компонентами C7-C12. Сформированный вектор передается на компонент T_R1, осуществляющий поиск точки росы при заданном давлении вещественного потока на выходе сепаратора. Найденное значение температуры точки росы в результате многовариантного анализа при изменении температуры T источника CS1 передается на цифровое табло T3.

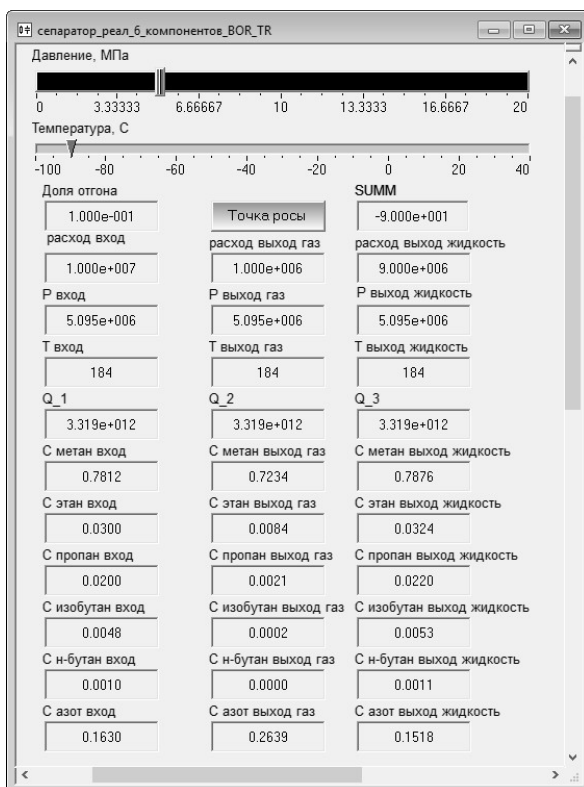


Рисунок 7.10 – Панель визуализации и интерактивного управления многоуровневой КМ определения температуры точки росы

При решении данной задачи на визуальном уровне многоуровневой КЦ сформирована панель визуализации и управления, представленная на рисунке 7.10.

На рисунке 7.11 приводится зависимость температуры точки росы газа от его давления, построенная в результате выполнения многовариантного анализа при изменении давления газа на входе сепаратора в пределах 3-7 МПа с шагом 1 МПа. Возрастание температуры точки росы газа при увеличении его давления на входе сепаратора подтверждает правильность работы модели и полученных результатов.

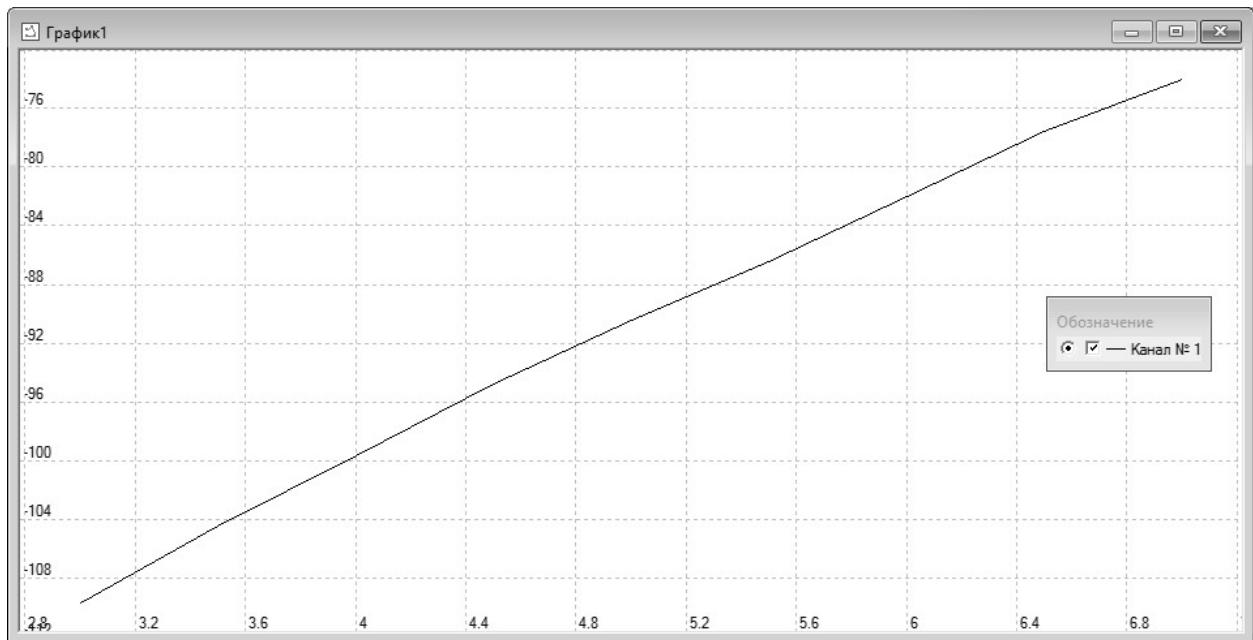


Рисунок 7.11 – График зависимости температуры T точки росы газа по углеводородам от давления P (P выражено в МПа, T – в К)

Сформированная многоуровневая КЦ может применяться для определения температуры точки росы газа при заданном давлении, а также значения давления газа, позволяющее обеспечить точку росы по углеводородам при известной его температуре.

7.5 Построение многоуровневой компонентной цепи системы управления режимами функционирования химико-технологической системы

7.5.1. Коррекция параметров компонентной цепи химико-технологической системы

Важной задачей газовой промышленности является минимизация расходов на добычу и первичную переработку добытого природного газа. Большая часть

газа добывается на Крайнем Севере Российской Федерации, где введены в эксплуатацию ряд установок комплексной подготовки газа (УКПГ) [138]. В них газ проходит несколько ступеней сепарации для освобождения от тяжелых углеводородов. Для снижения себестоимости транспорта газа и уменьшения проблемы гидратообразования в трубах он должен содержать минимальное количество воды. Для его осушки, то есть уменьшения концентрации воды в нем, используются абсорберы. В них используется эффект массообмена между газом и жидким ингибитором. Основной проблемой эксплуатации абсорберов на УКПГ является задача минимизации расхода ингибитора при абсорбционной осушке газа.

Для определения давления ингибитора, при котором выходящий из абсорбера газ будет иметь заданную минимальную степень осушки и при этом будет наблюдаться минимальный расход ингибитора может быть применена КЦ секции абсорбционной осушки природного газа, приведенная на рисунке 3.24. Для получения с её помощью актуальных результатов, позволяющих устанавливать в реальном объекте оптимальные режимы функционирования, компонентная цепь должна адекватно описывать процессы, протекающие с реальной ХТС.

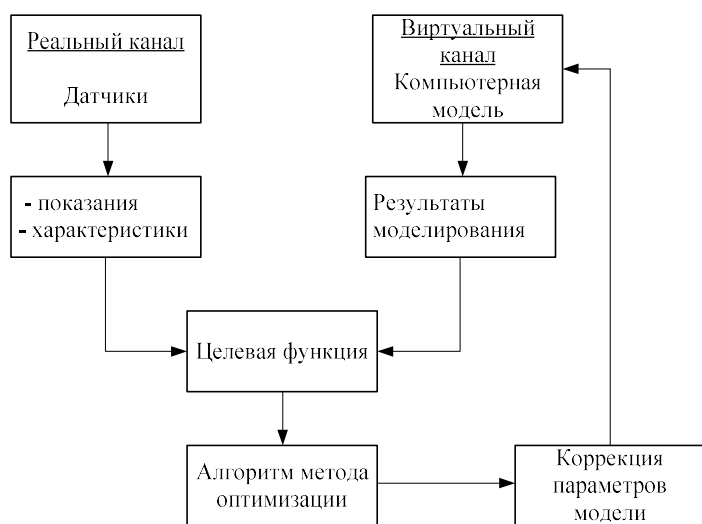


Рисунок 7.12 – Схема коррекции параметров модели

Для того, чтобы КЦ ХТС, являющаяся подцепью объектного уровня МКЦ, адекватно описывала протекающие в реальном объекте процессы, её параметры, от которых зависят коэффициенты уравнений ММ, должны непрерывно или периодически корректироваться на основе накапливаемой истории процесса. В этом случае решению подлежит

задача минимизации целевой функции вида (1.14), сформированная из приведенных на рисунке двух составляющих:

- данных, поступающих с датчиков, образующих реальный канал;

– результатов компьютерного моделирования, которые представляют собой виртуальный канал.

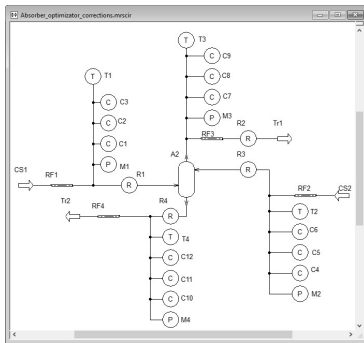
Параметрическая оптимизация [64] предполагает, что по известным откликам объекта на заданные воздействия при неизменной структуре его модели производится поиск значений её параметров, удовлетворяющих заданным критериям. Задача оптимизации формулируется как задача поиска минимума целевой функции $\min(F(X, P))$ в допустимой области варьирования значений параметров модели P . Результатом оптимизации является вектор значений параметров P^* , при которых наблюдается минимальное значение целевой функции F . Она реализует заданный пользователем критерий оптимизации, формируется на основании различных критериев оптимизации и вычисляется на основе полученных результатов анализа модели объекта, сформированной на объектном уровне МКЦ.

Формирование сценариев параметрической оптимизации производится на логическом уровне МКЦ СТУС с помощью стандартных компонентов-критериев, компонента «Целевая функция», представленного на рисунке 4.30, а также представленного на рисунке 4.31 компонента, реализующего различные методы оптимизации.

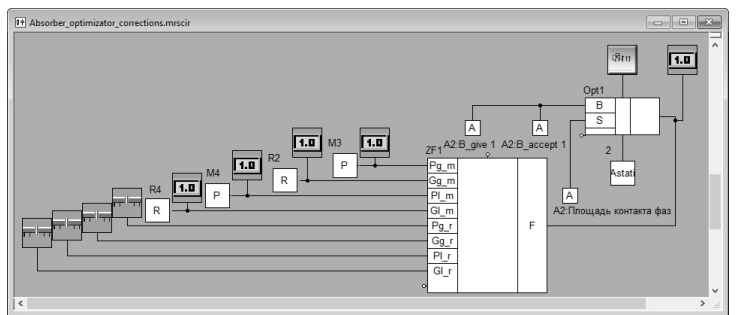
Задача коррекции параметров модели абсорбера заключается в поиске таких её параметров, при которых при тех же входных воздействиях переменные на выходе модели имели бы те же значения, что и выходные характеристики реального абсорбера. Для её решения были использованы данные выходных характеристик абсорбера, функционирующего на УКПГ Ямбургского газоконденсатного месторождения. При его исследовании были проведены замеры и получены следующие значения входных и выходных характеристик: входное давление газа – 4,25 МПа; входное давление ингибитора – 500 кПа; содержание воды в газе – 15 %; давление на выходе газовой фазы абсорбера $P_G^R = 4,234$ МПа; давление на выходе жидкой фазы абсорбера – $P_L^R = 516,2$ кПа; содержание воды в газе на выходе абсорбера – 14,7 %; содержание воды в ингибиторе на выходе абсорбера – 3,1 %. Они используются в процессе коррекции параметров компонента «Абсорбер» с целью более

точного отображения ею характеристик реального аппарата. Данная задача заключается в определении таких значений параметров модели абсорбера, при которых при тех же значениях входных воздействий (входные давления газа и ингибитора, а также концентрации воды в поступающем в абсорбер газе) переменные на выходе модели имели бы те же значения, что и выходные характеристики реального абсорбера. Для решения данной задачи на основе среднеквадратичного критерия была сформирована целевая функция

$$F(\beta, S) = (P_G^M \cdot G_G^M - P_G^R \cdot G_G^R)^2 + (P_L^M \cdot G_L^M - P_L^R \cdot G_L^R)^2, \quad (7.3)$$



а)



б)

	P_G_in	P_L_in	P_G_out	P_L_out
Начало	1.000e+006	5.000e+005	9.354e+005	5.646e+005
Fzel	1.7e+006			
	G_G_in	G_L_in	G_G_out	G_L_out
B	1.000e+003	1.200e+002	9.354e+002	1.846e+002
	T_G_in	T_L_in	T_G_out	T_L_out
S	6.378e-002	3.730e+002	3.730e+002	3.730e+002
	C_CH4_G_in	C_CH4_L_in	C_CH4_G_out	C_CH4_L_out
	0.700	0.000	0.748	0.000
	C_H2O_G_in	C_H2O_L_in	C_H2O_G_out	C_H2O_L_out
	0.300	0.000	0.252	0.350
	C_C2H5OH_G_in	C_C2H5OH_L_in	C_C2H5OH_G_out	C_C2H5OH_L_out
	0.000	1.000	0.000	0.650

в)

Рисунок 7.13 – Многоуровневая компьютерная модель коррекции параметров модели абсорбера

где P_G^M, G_G^M – давление и расход газа на выходе модели абсорбера; P_G^R, G_G^R – давление и расход газа на выходе реального абсорбера; P_L^M, G_L^M – давление и расход ингибитора на выходе модели абсорбера; P_L^R, G_L^R – давление и расход ингибитора на выходе реального абсорбера.

Независимыми аргументами целевой функции, изменяющимися в за-

данных пределах в процессе её минимизации, являются:

S – площадь поверхности контакта фаз;

β – коэффициент массопередачи в отдающей и принимающей фазах (в соответствии с физическими условиям и равными температурами газа и ингибитора коэффициенты массопередачи в отдающей и принимающей фазах принимаются равными друг другу). Для коррекции параметров компонента «Абсорбер» разработана многоуровневая КЦ, приведённая на рисунке 7.13.

В результате применения метода покоординатного спуска, реализованного в представленном на рисунке 7.13, б компоненте Opt1, были найдены следующие значения параметров модели абсорбера $\beta = 0.06378$ и $S = 8.922 \text{ м}^2$, что соответствует характеристикам реального абсорбера УКПГ Ямбургского газоконденсатного месторождения. Аналогичным образом могут быть найдены значения параметров моделей других компонентов, при которых они будут адекватно отражать процессы, протекающие в реальных элементах и аппаратах химической промышленности.

После того, как модель скорректирована по предложенной методике, она может считаться прогнозной и использоваться для определения оптимальных режимов функционирования химико-технологических систем.

7.5.2. Многоуровневая компонентная цепь минимизации расхода ингибитора при абсорбционной осушке природного газа

В УКПГ [138], эксплуатируемых на газовых и газоконденсатных месторождениях Крайнего Севера Российской Федерации, добытый природный газ проходит несколько степеней очистки от примесей и тяжелых углеводородов в сепараторах. Его осушка от влаги осуществляется в абсорберах путем массообмена воды между газовой и жидкой фазами в системе с подвижным контактом фаз. При этом актуальной проблемой является задача оптимального расхода ингибитора.

Сформируем целевую функцию. Требуется определить минимальный расход ингибитора на входе абсорбера, при котором концентрация воды в газовой

фазе на его выходе будет меньше или равна некоторой заданной (минимального) значения

$$\begin{aligned} C_{H_2O}^{G-out} &\leq C_{\min} \\ G^{G-in} &\rightarrow \min \end{aligned} \quad (7.4)$$

Таким образом, данная задача может быть сведена к задаче одномерной оптимизации с целевой функцией вида

$$F(G^{G-in}) = (C_{H_2O}^{G-out} - C_{\min})^2 \rightarrow \min \quad (7.5)$$

при увеличении расхода ингибитора на входе жидкой фазы абсорбера. Как только целевая функция достигнет своего минимума, будет найдено минимальное значение ингибитора G^{G-in} , при котором концентрация воды на выходе газовой фазы абсорбера будет иметь значение C_{\min} .

Решение поставленной задачи может быть осуществлено с помощью многоуровневой КЦ.

Пусть на вход газовой фазы абсорбера подается сырой очищенный от механических примесей газ с содержанием воды

$C_{H_2O}^{C-in} = 0,15$ под давлением 9.5 МПа и молярным расходом 1500 моль/с. Необходимо

определить минимальный расход ингибитора на входе жидкой фазы абсорбера, при котором газ на выходе будет иметь заданную минимальную концентрацию воды C_{\min} . Для

решения данной задачи сформирована многоуровневая КЦ, на объектном уровне которой, представленном на рисунке 7.14, располагается компонентная цепь секции абсорбционной осушки природного газа. Физический смысл измерительных компонентов, осуществляющих передачу результатов моделирования с объектного уровня

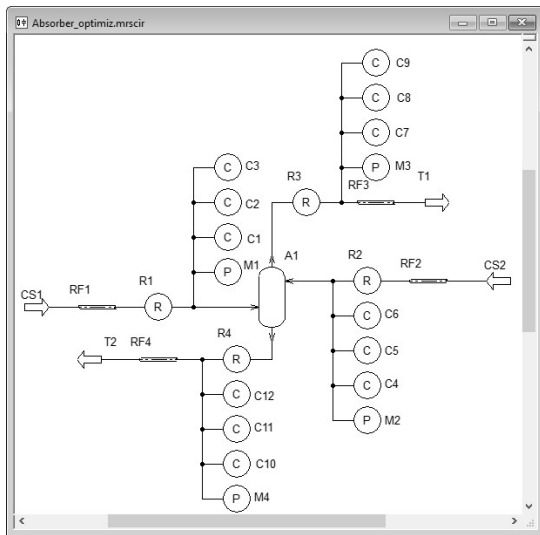


Рисунок 7.14 – Объектный уровень многоуровневой КМ минимизации расхода ингибитора

на её логический уровень, приведен в таблице 7.1.

На основе передаваемых значений на логическом уровне МКЦ сформирована алгоритмическая подцепь минимизации расхода ингибитора на входе абсорбера, представленная на рисунке 7.15.

Таблица 7.1 – Физический смысл измерительных компонентов

Физическая величина	Вход газовой фазы	Вход жидкой фазы	Выход газовой фазы	Выход жидкой фазы
Давление	M1	M2	M3	M4
Расход	R1	R2	R3	R4
Концентрация газа	C1	C4	C7	C10
Концентрация воды	C2	C5	C8	C11
Концентрация ингибитора	C3	C6	C9	C12

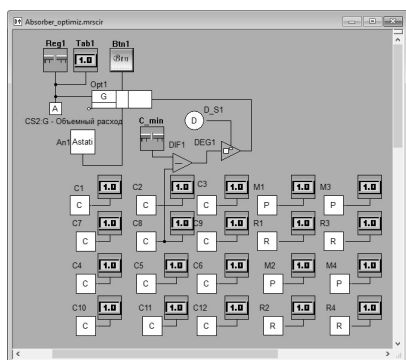


Рисунок 7.15 – Алгоритмическая компонентная цепь минимизации расхода ингибитора при абсорбционной осушке природного газа

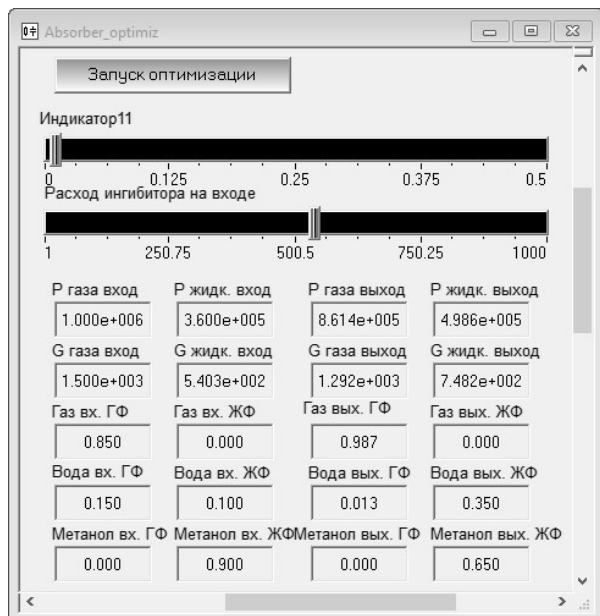


Рисунок 7.16 – Результаты решения задачи минимизации расхода ингибитора при абсорбционной осушке природного газа на визуальном уровне многоуровневой КЦ

Процесс оптимизации запускается по нажатию на кнопку Btn1 и осуществляется компонентом-оптимизатором Opt1. Целевая функция (7.5) сформирована с помощью компонентов DIF1 и DEG1, который возводит разность между требуемой минимальной C_{min} и текущей концентрацией воды на выходе газовой фазы абсорбера, передаваемой с объектного на логический уровень многоуровневой КЦ измерителем концентраций C8. В ходе работы компонента-оптимизатора текущее значение расхода ингибитора на входе абсорбера передается на визуальный уровень с помощью регулятора Reg1 и отображается цифровым табло Tab1.

На рисунке 7.16 представлен визуальный уровень многоуровневой КЦ минимизации расхода ингибитора при абсорбционной осушке природного газа. С ее помощью получено, что если газ подается

в абсорбер под давлением $P_{G_IN} = 1 \text{ МПа}$ с объемным расходом $G_{G_IN} = 1500 \text{ моль} / \text{с}$, то минимальный расход ингибитора будет составлять $G_{L_IN} = 540.3 \text{ моль} / \text{с}$ при его входном давлении $P_{L_IN} = 360 \text{ кПа}$. Полученные данные были приняты технологическим персоналом УКПГ Ямбургского газоконденсатного месторождения, где получили практическое подтверждение при эксплуатации секции абсорбционной осушки природного газа.

Аналогичным образом могут быть решены другие задачи проектно-конструкторского характера и интеллектуального управления, в процессе решения которых многократно выполняется анализ КЦ исследуемой ХТС при различных значениях параметров входящих в неё компонентов.

7.6 Построение сетевых компьютерных тренажеров для обучения и переподготовки операторов-технологов

Для предотвращения предаварийных и аварийных ситуаций на объектах повышенной опасности, к классу которых относятся химико-технологические системы газовой промышленности, а также для приобретения навыков оперативного действия в условиях возникновения таких ситуаций управляющий персонал предприятий должен помимо получения теоретических знаний проходить и практические занятия. Их целью является приобретения навыков совместного управления ХТС в сложившихся условиях, ликвидация последствий возникающих ситуаций и приведение химико-технологических систем к штатному режиму функционирования. По различным причинам проведение таких занятий на действующих промышленных объектах недопустимо. Наиболее оптимальным способом их проведения является применение комплексом программ компьютерного моделирования, позволяющих строить модели ХТС с протекающими в них процессами с последующей реализацией на их основе сетевых компьютерных тренажеров [207]. Они должны быть реализованы на основе стандартной архитектуры «клиент-сервер» на основе системы компьютерного моделирования, включающей в себя библиотеку моделей компонентов сетевого взаимодействия.

Для реализации сетевых компьютерных тренажеров на базе среды моделирования химико-технологических систем в рамках языка моделирования алгоритмических конструкций разработана библиотека сетевого взаимодействия моделей, реализованная в виде виртуальных интерфейсов передачи данных [208]. Она позволяет двум или нескольким приложениям, работающим в локальной вычислительной сети (ЛВС), осуществлять обмен данными различных типов. Применение таких компонентов позволило реализовать структуру сетевой многоуровневой КЦ, приведенную на рисунке 7.17. Она легла в основу реализации сетевого тренажера для подготовки операторов-технологов ХТС предприятий газовой промышленности [34, 209].

В этом случае сетевая структура многоуровневой КЦ состоит из серверной (Сервер) и клиентских (Клиент) частей, которые с помощью компонентов сетевого взаимодействия подключаются к одной серверной части. В этом случае серверная часть сетевой многоуровневой компонентной цепи состоит из:

– *объектного уровня*, где формируется интерфейс взаимодействия с реальным исследуемым объектом или его компонентная цепь, являющаяся объектной подцепью МКЦ, представленная в формате языка моделирования ХТС и анализируемая универсальным вычислительным ядром. Также на этом уровне может быть использована модель технического или технологического объекта, реализованная в виде динамически загружаемой библиотеки (DLL) на любом языке программирования;

– *логического уровня*, который содержит представленный в виде алгоритмической КЦ сценарий проведения эксперимента, состоящий из блоков обработки результатов, подцепей задания и вычисления значений параметров компонентов, средств подготовки и передачи клиентам результатов моделирования.

Клиентские части сетевой многоуровневой КМ включают:

– логический уровень, где в формате языка моделирования алгоритмических конструкций формируются сценарии получения, подготовки и передачи визуальным компонентам подлежащих отображению результатов моделирования.

Также на данном уровне располагаются компоненты формирования и отправки серверной части команд на изменение значений параметров КЦ исследуемой ХТС;

– *визуального уровня*, на котором из визуальных компонентов множества K_v (2.67) сформирована панель визуализации и управления параметрами исследуемой КМ.



Рисунок 7.17 – Структура распределенной многоуровневой компьютерной модели для реализации сетевого тренажера

В качестве клиентских приложений используются виртуальные приборы, созданные ранее в системе виртуальных инструментов и приборов [178], которые открываются специальным движком MARS-Engine [210]. Он позволяет открывать ВП как отдельные приложения, не подгружая при этом весь сложный интерфейс среды моделирования ХТС. Такие приложения представляют собой приборы, пример которых представлен на рисунке 7.18. С их помощью пользователи (преподаватели и обучаемые) могут наблюдать соответствующие результаты моделирования и в интерактивном режиме изменять значения параметров компонентов анализируемой на сервере КЦ.

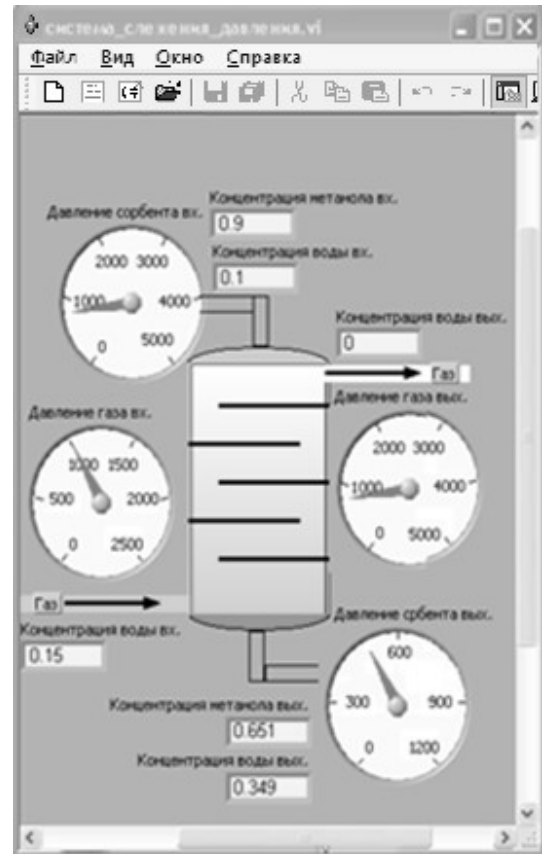


Рисунок 7.18 – Клиентская часть сетевого компьютерного тренажера

Преподаватели также имеют возможность изменять топологию анализируемой модели с помощью специальных коммутационных компонентов для создания различных внештатных ситуаций.

Сетевое взаимодействие между серверным и клиентскими приложениями осуществляется с помощью компонентов «Передатчик» (Transmitter), который отправляет данные и изображен на рисунке 7.19, а, и компонента «Приемник» (Receiver), представленного на рисунке 7.19, б.

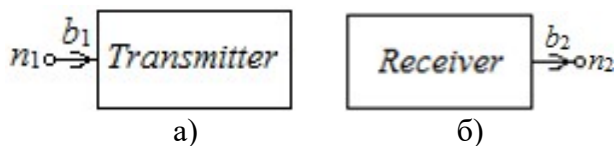


Рисунок 7.19 – Компоненты взаимодействия серверного и клиентских приложений сетевого компьютерного тренажера

а) компонент «Передатчик»; б) компонент «Приемник»

Компонент «Передатчик» осуществляет отправку данных D_1 , принятых по связи $S_1 = (b_1 \eta^{-1} n_1) \rightarrow \{D_1\}$ по локальной вычислительной сети компьютеру, чей IP-адрес указан в качестве параметра данного компонента. Компонент

нент «Приемник» принимает адресованные ему и широковещательные сообщения из ЛВС и передает принятые данные по связи $S_2 = (b_2\eta^+n_2) \rightarrow \{D_2\}$.

Реализованный таким образом сетевой компьютерный тренажер на базе среды моделирования ХТС предназначен для проведения компьютеризированных занятий, направленных на приобретение навыков группового управления технологическим оборудованием предприятий нефтегазовой промышленности и отработки совместных действий персонала как в штатном режиме, так и при возникновении внештатных, предаварийных и аварийных ситуаций. Используемые в них компонентные цепи ХТС допускают изменение своей топологии и варьирование параметров входящих в них компонентов. С помощью таких тренажеров осуществляется проведение занятий по отработке действий персонала в различных производственных ситуациях без вывода реального оборудования из технологического процесса, не подвергая опасности персонал предприятия, объекты окружающей природной среды и проживающее население.

7.7 Выводы

1. Сформирован алгоритм решения задач исследования и функционального проектирования СТус на базе многоуровневой компонентной цепи, в основе которого лежит многократный анализ КЦ ХТС при различных значениях параметров её компонентов.

2. Предложена методика разработки и отладки сценариев управления протекающими в ХТС процессами с помощью многоуровневой компонентной цепи. В качестве примера рассмотрена МКЦ системы управления состоянием баллона (газохранилища), предоставляющая возможности варьирования уставками, относительно которых должны срабатывать исполнительные устройства, а также их постоянными времени для исследования динамики процессов наполнения газохранилища и бесперебойной поставки газа потребителям.

3. С использованием модели теплообменника реализована многоуровневая компонентная цепь, позволяющая при заданных гидравлических и термодинамических характеристиках теплоносителей на входе теплообменника опреде-

лять минимальную его длину, при которой на выходе теплообменника температуры теплоносителей будут одинаковыми. С помощью данной модели продемонстрирована возможность решения задач функционального проектирования с помощью программно-алгоритмического аппарата метода компонентных цепей, развитого для компьютерного моделирования управляемых химико-технологических систем.

4. На основе введенных в главе 4 обобщенных компонентов для проведения различных видов многовариантного анализа разработан компонент «Определение точки росы», который на основе реализованного в нём численного метода определяет температуру точки росы газа при варьировании давления на входе исследуемой модели аппарата ХТС. На его основе реализована многоуровневая компонентная цепь определения температуры точки росы газа, входящего в сепаратор, при различных значениях его входного давления.

5. На базе многоуровневой компьютерной модели решена актуальная для газовой промышленности задача минимизации расхода ингибитора при абсорбционной осушке природного газа. Для этого сформирована схема коррекции параметров компонента «Абсорбер» как аппарата ХТС, а также непосредственно многоуровневая компонентная цепь для минимизации расхода ингибитора при абсорбционной осушке природного газа.

6. Предложена структура распределенной многоуровневой компонентной цепи, являющейся основой для реализации сетевых компьютерных тренажеров. Они предназначены для проведения групповых занятий с использованием КЦ ХТС по отработке навыков персонала совместного управления в штатном режиме его функционирования и при возникновении внештатных, предаварийных и аварийных ситуаций.

Заключение

Диссертационная работа явилась обобщением научных исследований и программно-алгоритмических разработок автора. В ней осуществлено развитие метода компонентных цепей и его адаптация к моделированию управляемых химико-технологических систем. Проведенные исследования положены в основе реализации комплекса программ моделирования химико-технологических систем предприятий газовой промышленности. Протекающие в них процессы направлены преимущественно на преобразование физических свойств и составов многокомпонентных вещественных потоков. Для целей построения многоуровневых компонентных цепей СТус и автоматизации ВЭ, направленного на автоматизацию решения задач научно-исследовательского, проектно-конструкторского и технологического характера, были поставлены и успешно решены соответствующие задачи, результаты которых имеют самостоятельное научное и практическое значение:

1. На основе нового подхода к автоматизированному решению задач исследования и функционального проектирования СТус, предназначенных для управления химико-технологическими системами, предложено их новое формализованное представление, лежащее в основе формирования компьютерных моделей сложных технических управляемых систем.

2. Осуществлено развитие метода компонентных цепей, обеспечившее формирование многоуровневых компонентных цепей СТус в виде трех взаимосвязанных уровней в формате языка их графического формирования, состоящего из трех взаимосвязанных подязыков: языка химико-технологических систем, языка моделирования алгоритмических конструкций и языка виртуальных инструментов и приборов, алгоритмический аппарат которых построен на единых принципах объектно-ориентированного программирования.

3. Сформированы графические нотации языка химико-технологических систем для формирования их компьютерных моделей на объектном уровне многоуровневой компонентной цепи. Разработаны алгоритмы автоматического формирования и численного анализа их вычислительных моделей с неоднородными

векторными связями, описывающими передачу между компонентами ХТС информационных, мультифизических энергетических и многокомпонентных вещественных потоков.

4. Разработана новая структура универсального вычислительного ядра, адаптированного к анализу вычислительных моделей химико-технологических систем. В его составе реализованы новые численные методы явно-неявного анализа вычислительных моделей ХТС с неоднородными векторными связями с возможностями непосредственного определения значений переменных на этапе формирования системы алгебро-дифференциальных уравнений. Он обеспечивает повышение точности получаемых результатов, а также сокращение времени решения задач на 25-50 % по сравнению с методами неявного анализа, реализованными ранее в рамках универсального вычислительного ядра.

5. Сформирована грамматика языка моделирования алгоритмических конструкций, состоящая из лексем и грамматических правил построения имитационных моделей сценариев экспериментов, включая элементы численной обработки результатов анализа компьютерных моделей ХТС и интерактивного изменения значений параметров модели. Предложена и разработана система отображения лексем языка МАК в формат алгоритмических компонентных цепей и библиотека моделей компонентов основных операций, входящих в алгоритмы проведения вычислительных экспериментов.

6. Предложен и исследован язык виртуальных инструментов и приборов, а также реализован набор визуальных компонентов, предназначенных для формирования на визуальном уровне многоуровневых компонентных цепей лицевых панелей виртуальных приборов, к классу которых относятся панели визуализации и интерактивного управления вычислительным экспериментом.

7. Разработана многоуровневая структура библиотеки моделей компонентов, в которой хранятся компоненты трех подязыков языка многоуровневых компонентных цепей, обеспечивающие построение МКЦ СТУС на соответствующих слоях многослойного редактора.

8. Сформирована структура и реализован комплекс программ «Среда моделирования химико-технологических систем», основе которого лежит развитие метода компонентных цепей и его адаптация к моделированию химико-технологических систем. Данный комплекс программ обеспечивает проведение автоматизированных вычислительных экспериментов над сложными техническими управляемыми системами предприятий газовой промышленности, а также лежит в основе реализации сетевых компьютерных тренажеров операторов-технологов.

Представленные результаты свидетельствуют о том, что все поставленные задачи диссертационного исследования успешно решены, а сформулированные цели достигнуты.

Создание отечественного комплекса программ моделирования химико-технологических систем способствует повышению научного потенциала страны и решению задач импортозамещения дорогостоящих программных продуктов.

Промышленное использование комплекса программ позволяет автоматизировать решение задач исследования и функционального проектирования технологических объектов предприятий газовой промышленности, а также систем управления ими с возможностью формирования и отладки алгоритмов функционирования управляющих устройств. Внедрение в него алгоритмов и средств компьютерного моделирования объектов с неоднородными векторными связями открывает более широкие возможности автоматизации решения задач анализа и параметрического синтеза химико-технологических систем.

Разработанный на основе комплекса программ сетевой тренажер позволяет автоматизировать проведение практических занятий и лабораторных работ, направленных на приобретение навыков совместного управления объектами предприятий газодобывающей и газоперерабатывающей промышленности. Он позволяет отработать совместные действия персонала в штатных режимах работы промышленного оборудования, а также при возникновении внештатных, предаварийных и аварийных ситуаций. Его использование в научных исследованиях открывает более широкие возможности отработки на моделях инновационных под-

ходов к управлению технологическими объектами, а также обеспечивает определение и поддержание режимов их функционирования, удовлетворяющих различным критериям оптимизации.

Дальнейшим развитием исследований видится реализация программно-алгоритмического аппарата формирования распределенных SCADA-систем со встроенными в них компьютерными моделями для управления технологическими объектами предприятий химических отраслей промышленности, допускающих декомпозицию на элементы с многокомпонентными вещественными потоками между ними.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Марчук Г.И. Методы вычислительной математики. – М.: Наука, 1977. – 455 с.
2. Самарский А.А. Математическое моделирование и вычислительный эксперимент // Вестник АН СССР. 1979. – № 5. – С.38-49.
3. Самарский А.А. Математическое моделирование. Идеи. Методы. Примеры. – М.: Физматлит, 2001. – 320 с.
4. Самарский А.А., Вабищевич П.Н. Математическое моделирование и вычислительный эксперимент, Институт математического моделирования РАН, 2000 [Электронный ресурс]. – Режим доступа: <http://www.imamod.ru/~vab/matmod/MatMod.htm>, свободный
5. Петров П.В. Технология вычислительного эксперимента / П.В. Петров, Р.А. Сумарчин, В.А. Целищев // Вестник Уфимского государственного авиационного технического университета. – 2008. – Т. 10. – № 1. – С. 30-35.
6. Коева А.А. Автоматизация вычислительного эксперимента с помощью программного комплекса HMAR / А.А. Коева, П.В. Петров, В.А. Целищев // Вестник Уфимского государственного авиационного технического университета. – 2013. – Т. 17. – № 3. – С. 166-173
7. SimMechanics – моделирование механических систем для Simulink [Электронный ресурс] – Режим доступа: <http://matlab.ru/products/simmechanics>, свободный. Дата последнего обращения 30.12.2015 г.
8. Универсальный механизм – программный комплекс для моделирования механических систем: Главная страница [электронный ресурс] – Режим доступа: <http://www.umlub.ru/pages/index.php?id=1>, свободный. Дата последнего обращения 30.12.2015 г.
9. Sevice – Netafim [электронный ресурс] – Режим доступа: <https://www.netafim.com/service/hydrocalc-pro>, свободный. Дата последнего обращения 30.12.2015 г.
10. Комплекс программ «Гидравлических расчет водопроводной сети с графическим представлением расчетной системы «HydroCalc». ЭСКО. Электрон-

ный журнал энергосервисной компании «Экологические системы. – 2004. – № 11. [Электронный ресурс]. – Режим доступа: http://esco-ecosys.narod.ru/2004_11/art56.htm, свободный. Дата последнего обращения 30.12.2015 г.

11. Тимофеев В.Ю., Зайцев А.А. Моделирование тепловых полей в сложных динамических системах средствами САПР // Физико-математические науки. Физика. – 2009. – № 2 (10), 2009. – С. 115-122

12. Теплоэнергетические системы и энергетические балансы промышленных предприятий. ЭСКО. Электронный журнал энергосервисной компании «Экологические системы», 2004, № 1. [Электронный ресурс]. – Режим доступа: http://esco-ecosys.narod.ru/2004_1/art100.htm, свободный. . (дата обращения: 30.01.2015).

13. James W. Nilsson, Susan Riedel. Introduction to Multisim for Electric Circuits. – Prentice Hall. – 2011. – 144 pp.

14. Аветисян Д.А. Автоматизация проектирования электромеханических систем. – М.: Высш. шк., 1998. – 331 с.

15. Знакомство с системой CLASSIC. Исследование линейных САУ в CLASSIC. Методические указания для выполнения лабораторных работ по курсу «Моделирование систем» для студентов направлений 550200 – Автоматизация и управления и 657900 – Автоматизация технологических процессов и производств (в нефтегазовой отрасли) Института дистанционного обучения / А.В. Воронин. – Томск: Издательство ТПУ, 2004. – 18 с. [Электронный ресурс] – Режим доступа: <http://window.edu.ru/resource/120/57120/files/tpu019.pdf>, свободный. Дата последнего обращения 30.12.2015 г.

16. Сибирцев В.С. Экспериментальные методы исследования физико-химических систем. Ч. 1. Основы теории строения вещества и физико-химических превращений: Учебное пособие. – СПб.: Университет ИТМО, 2016. – 78 с.

17. Дмитриев В.М. Автоматизация моделирования промышленных роботов / В.М. Дмитриев, Л.А. Арайс, А.В. Шутенков. – М.: Машиностроение, 1995. – 304 с.

18. Дмитриев В.М. Автоматизация функционального проектирования электромеханических систем и устройств преобразовательной техники / В.М. Дмитриев, Т.Н. Зайченко, А.Г. Гарганеев, Ю.А. Шурыгин. – Томск: Изд-во Том. ун-та, 2000. – 292 с.

19. Зайченко Т.Н. Методы, алгоритмы и программные средства моделирования электротехнических устройств и систем: Диссертация на соискание степени доктора технических наук. 05.13.18 / Т.Н. Зайченко; науч. конс.: Ю.А. Шурыгин; Федеральное агентство по образованию, Томский государственный университет систем управления и радиоэлектроники. – Томск, 2007. – 445 с.

20. МАРС – среда моделирования технических устройств и систем. Монография / В.М. Дмитриев, А.В. Шутенков, Т.Н. Зайченко, Т.В. Ганджа. – Томск: В-Спектр, 2011. – 278 с.

21. Дмитриев В.М. Виртуальные лаборатории и программно-инструментальное обеспечение для их разработки / В.М. Дмитриев, А.В. Шутенков // Компьютерные технологии в образовании. Под ред. В.М. Дмитриева. – Томск: Изд-во Том. ун-та, 2001. – Вып. 1. – С. 86-94.

22. Зайченко Т.Н. Виртуальная учебная лаборатория по курсу «Теория автоматического управления» / Т.Н. Зайченко, Т.В. Ганджа // Компьютерные технологии в образовании. Под ред. В.М. Дмитриева. – Томск: Изд-во Том. ун-та, 2004. – Вып. 2. – с. 79-87.

23. Зайченко Т.Н. Виртуальная учебная лаборатория по курсу «Электротехника и электроника» / Т.Н. Зайченко, Т.В. Ганджа // Электронные средства и системы управления. Материалы международной научно-практической конференции. – Томск: Изд-во Института оптики атмосферы СО РАН, 2004. – В трех частях. – Ч. 2. – С. 183-186.

24. Балакина Э.В., Дмитриев В.М., Журавский Ю.А. Экономико-экологическое регулирование в процессе российских реформ: Монография. – Кемерово: Кемеровский институт пищевой промышленности, 1999. – 108 с.

25. Клепак И.Я. Алгоритмы автоматизации и управления технологическими процессами газопромысловых объектов с применением компьютерных мо-

делей: Диссертация на соискание степени кандидата технических наук. 05.13.06 / И.Я. Клепак; науч. рук. В.М. Дмитриев; Томский государственный университет систем управления и радиоэлектроники. – Томск, 2009. – 155 с.

26. Затик О.С. Алгоритмы и программные средства имитационного моделирования для управления эколого-экономическими системами на основе метода компонентных цепей [Текст]: диссертация на соискание ученой степени кандидата технических наук, 05.13.06 / О.С. Затик; науч. рук. В.М. Дмитриев; Томский государственный университет систем управления и радиоэлектроники. – Томск, 2011. – 186 с.

27. Коротина Т.Ю. Алгоритмы и программное обеспечение моделирования приборов и устройств для воздания автоматизированных лабораторных комплексов [Текст]: диссертация на соискание ученой степени кандидата технических наук, 05.13.18 / Т.Ю. Коротина; науч. рук. В.М. Дмитриев; Томский государственный университет систем управления и радиоэлектроники. – Томск, 2013. – 193 с.

28. Джарратано Дж., Райли Г. Экспертные системы: принципы разработки и программирование: Пер. с англ. — М. : Издательский дом «Вильямс», 2006. – 1152 с.

29. Охорзин В.А. Прикладная математика в системе MathCAD. Учебное пособие. 3-е изд.- СПб.: Лань, 2009. – 352 с.

30. Математика на Макрокалькуляторе: Учебное пособие / В.М. Дмитриев, Т.В. Ганджа, Е.В. Истигечева; Федеральное агентство по образованию. Томск. гос. ун-т систем управления и радиоэлектроники, Высший колледж информатики, электроники и менеджмента. – Томск: Томск. гос. ун-т систем управления и радиоэлектроники, 2007. – 110 с.

31. Тарасевич Ю.Ю. Математическое и компьютерное моделирование. Вводный курс: Учебное пособие. Изд. 4-е, испр. – М.: Едиториал УРСС, 2004. – 152 с.

32. Дмитриев В.М., Шутенков А.В, Дмитриев И.В. Компьютерное моделирование устройств и систем / Том. гос. ун-т систем управления и радиоэлектроники. – Томск: ТМЛ-Пресс, 2010. – 293 с.
33. Тема 4.3 Компьютерные тренажеры в обучении персонала нефтегазовой отрасли [Электронный ресурс] – Режим доступа: imeloet.ru/epa/docs/ITiOvNGO/4.3.pdf, свободный. . (дата обращения: 30.01.2015).
34. Dmitriev V.M., Gandzha T.V., Dolganov I.M., Pisarev M.O., Dolganova I.O., Sizova E.N., Ivashkina E.N. Structure of network simulator for training and re-training of operators of controlled technological objects of oil and gas industry // *Petroleum and Coal*. – Vol. 57. – 2015. – Issue 6. – PP. 691-695
35. Кулаков С.М. Интеллектуальные системы управления технологическими процессами: теория и практика (монография) / С.М. Кулаков, В.Б. Трофимов // *Успехи современного естествознания*. – 2010. – № 2. – С. 101-102.
36. Ганджа Т.В. Формализованное представление технически сложного объекта с компьютерной моделью в контуре управления // *Приборы и системы. Управление. Контроль. Диагностика*. – 2012. – № 2. – С. 29-35.
37. Кафаров В.В. Анализ и синтез химико-технологических систем. Учебник для вузов. / В.В. Кафаров, В.П. Мешалкин. – М.: Химия, 1991. – 432 с.
38. Дытнерский Ю.И. Процессы и аппараты химической технологии: Учебник для вузов. Изд. 2-е. В 2 кн.: Часть 1. Теоретические основы процессов химической технологии. Гидромеханические и тепловые процессы и аппараты. – М.: Химия, 1995. – 400 с.
39. Дытнерский Ю.И. Процессы и аппараты химической технологии: Учебник для вузов. Изд. 2-е. В 2 кн. Часть 2. Массообменные процессы и аппараты. – М.: Химия, 1995. – 368 с.
40. Датчики: Справочное пособие / В.М. Шарапов, Е.С. Полищук, Н.Д. Кошевой, Г.Г. Ишанин, И.Г. Минаев, А.С. Совлуков. – Москва: Техносфера, 2012. – 624 с.

41. Кафаров В.В. Основы построения операционных систем в химической технологии / В.В. Кафаров, В.Н. Ветохин. – М.: Наука, 1980. – 430 с.
42. Программирование и вычислительные методы в химии и химической технологии. / В.В. Кафаров, В.Н. Ветохин, А.И. Бояринов. – М.: Наука, 1972. – 489 с.
43. Michael E. Henyak, Ir. Chemical Process Simulation and the Aspen HYSYS Software. // Department of Chemical Engineering. Bucknell University. Lewisburg, PA, 17837. – 1998. – 111 P. [Электронный ресурс] – Режим доступа: http://www.departments.bucknell.edu/chem_eng/cheg200/HYSYS_Manual_a_BlueHYSYS.pdf, свободный. . (дата обращения: 30.01.2015).
44. CHEMCAD Suite [Электронный ресурс] – Режим доступа: <http://www.chemcad.co.uk/>, свободный. (дата обращения: 30.01.2015).
45. Математическое моделирование химико-технологических систем с использованием программы ChemCAD: Учебно-методическое пособие / Казан. гос. технол. ун-т. Сост.: Н.Н. Зиятдинов, Т.В. Лаптева, Р.А. Рыжов. – Казань, 2008. – 160 с. [Электронный ресурс]. – Режим доступа: http://www.kstu.ru/ft/978-5-7882-XXX-Zijatdinow_matmod.pdf свободный. (дата обращения: 30.01.2015).
46. Гартман Т.Н., Бояринов А.И. Применение ЭВМ в химической технологии: Учебн. пособие. – М.: МХТИ им. Д.И. Менделеева, 1987. – 83 с.
47. Кравцов А.В., Иванчина Э.Д. Интеллектуальные системы в химической технологии и инженерном образовании. – Новосибирск: Наука, 1996. – 200 с.
48. Кравцов А.В. Системный анализ и повышение эффективности нефтеперерабатывающих производств методом математического моделирования / А.В. Кравцов, Э.Д. Иванчина, С.А. Галушин, Д.С. Полубоярцев. – Томск: Изд-во ТПУ, 2004. – 170 с.
49. Кравцов А.В. Системный анализ химико-технологических процессов: учебное пособие / А.В. Кравцов, Э.Д. Иванчина, Е.Н. Ивашкина, Е.С. Шарова. – Томск: Изд-во ТПУ, 2008. – 96 с.
50. Kravtsov A.V. Thermodynamic stability of coke-generation compounds formed on the surface of platinum dehydrogenation catalysts in their oxidation with wa-

ter / A.V. Kravtsov, E.D. Ivanchina, E.N. Ivashkina, E.V. Frantsina, S.V. Kiseleva, R.V. Romanivskii. // *Petroleum Chemistry*. – 2013. – Т. 53. – № 4. – С. 267-275.

51. Frantsina E.V. Developing of mathematical model for controlling the operation of alkane dehydrogenation catalyst on production of linear benzene / E.V. Frantsina, E.N. Ivashkina, E.D. Ivanchina, R.V. Romanocskii // *Chemical Engineering Journal*. – 2014. – Т. 238. – P. 129-139.

52. Унифицированные технологические схемы сбора, транспорта и подготовки нефти, газа и воды нефтедобывающих районов [Текст]: РД 39-0148311-605-86, утвержден министерством нефтяной промышленности 3.10.1986: введ в действие 01.01.1987. – М: Министерство нефтяной промышленности, 1986. [Электронный документ]. – Режим доступа: http://www.ohranatruda.ru/ot_biblio/normativ/data_normativ/10/10184/index.php, свободный. (дата обращения: 30.01.2015).

53. ГОСТ ISO 9001-2011. Системы менеджмента качества. Требования. – М.: ФГУП «СТАНДАРТИНФОРМ», 2012. – 36 с.

54. ГОСТ ISO 14001-2004. Системы экологического менеджмента – требования и руководство по применению. – Международная организация по стандартизации (ИСО), 2004. – 30 с.

55. Дмитриев В.М. Интеллектуализация управления технологическими процессами на углеводородных месторождениях / В.М. Дмитриев, Т.В. Ганджа, Е.В. Истигечева, И.Я.Клепак. – Томск: В-Спектр, 2012. – 212 с.

56. Кочергин М.И. Поиск прототипов сложных технических систем и их элементов на основе автоматического анализа текста / М.И. Кочергин, В.В. Ганджа, С.А. Панов // *Перспективы развития фундаментальных наук. Сборник трудов XI Международной конференции студентов и молодых ученых под редакцией Е.А. Вайтулевич*. – Томск: Национальный исследовательский Томский политехнический университет, 2014 – С.1039-1041.

57. Ганджа Т.В. Место подсистемы документирования в интеллектуальной SCADA-системе / Т.В. Ганджа, С.А. Панов // *Современные техника и технологии: Сборник трудов XIX Международной научно-практической конференции студентов, аспирантов и молодых ученых*. В 3 т. Т. 2. / Томский политехнический

университет. – Томск: Изд-во Томского политехнического университета, 2013. – С. 317-318.

58. Дмитриев В.М. Компьютерные модели в контуре интеллектуального управления технологическими процессами нефтегазодобывающих предприятий для минимизации техногенного воздействия на окружающую среду / В.М. Дмитриев, Т.В. Ганджа // Природные и интеллектуальные ресурсы Сибири (СИБРЕСУРС-2012): доклады 18-й Международной научно-практической конференции, Новосибирск, 15-17 октября 2012 г. / Отв. Ред. Л.С. Петрова, В.Н. Масленников. – Томск: САН ВШ; В-Спектр, 2012. – С- 203-207.

59. Дмитриев В.М. Структурно-функциональная схема SCADA-системы для управления технологическими процессами предприятий нефтегазовой промышленности / В.М. Дмитриев, Т.В. Ганджа // Методы и алгоритмы прикладной математики в технике, медицине и экономике: Материалы XIII Междунар. науч.-практ. конф. г. Новочеркасск, 12 марта 2013 г. / Юж.-Рос. гос. техн. ун-т (НПИ). – Новочеркасск, ЮРГТУ (НПИ), 2013. – С. 17-21.

60. Дмитриев В.М. Компьютерное моделирование SCADA-систем / В.М. Дмитриев, Т.В. Ганджа // Управление большими системами. Материалы X Всероссийской школы-конференции молодых ученых. Том 3. / Уфимск. гос. техн. ун-т. 5-7 июня 2013 года. – Уфа: УГАТУ, 2013. – с. 75-78.

61. Теория автоматического управления: Учеб. пособие для ВУЗов / А.С. Востриков, Г.А. Французова. – М.: Высш. шк., 2004. – 365 с.

62. Функциональный контроль и диагностика электротехнических и электромеханических систем и устройств по цифровым отсчетам мгновенных значений тока и напряжения / В.С. Аврамчук, Н.Л. Бацева, Е.И. Гольдштейн, И.Н. Исаченко, Д.В. Ли, А.О. Сулейманов, И.В. Цапко // Под ред. Е.И. Гольдштейна. – Томск: Печатная мануфактура, 2003. – 240 с.

63. Ганджа Т.В. Параметрический синтез технических объектов // Компьютерные технологии в современном образовании / Под ред. В.М. Дмитриева. – Томск: изд-во Том. ун-та, 2001. – Вып. 1. – С. 194-200.

64. Банди Б. Методы оптимизации. Вводный курс: Пер. с англ. – М.: Радио и связь, 1988. – 128 с.
65. САПР. Системы автоматического проектирования: учеб. пособие для техн. Вузов. В 9 кн. Кн. 5. Автоматизация функционального проектирования / П.К. Кузьмик, В.Б. Маничев; Под ред. Норенкова. – Мн.: Выш. шк., 1988. – 141 с.: ил.
66. Бобенко А.В. Статистический анализа в среде многоуровневого моделирования МАРС // Научная сессия ТУСУР-2012: Материалы Всероссийской научно-технической конференции студентов, аспирантов и молодых ученых, Томск, 16-18 мая 2012 г. – Томск: В-Спектр, 2012: В 5 частях. – Ч. 5. – С. 136-139.
67. Лаходьнов В.С. Модели оптимальной оценки и прогноза неизмеряемой переменной состояния электромеханического объекта [Текст]: Диссертация на соискание степени кандидата технических наук, 05.13.18 / В.С. Лаходьнов; науч. рук. В.Г. Букреев; Национальный исследовательский Томский политехнический университет. – Томск, 2010. – 157 с.
68. Колесникова С.И. Исследование качества распознавания состояния стохастической системы / С.И. Колесникова, В.С. Лаходьнов, Ю.Р. Цой // Информационные технологии. – 2010. – № 6. – С. 56-
69. Зайченко Т.Н. Блок обработки результатов вычислительного эксперимента в системе автоматизации функционального моделирования электромеханических систем / Т.Н. Зайченко, Т.В. Ганджа // Современная техника и технологии. СТТ-2001. Труды VII международной научно-практической конференции студентов, аспирантов и молодых специалистов. 26 февраля – 2 марта 2001 г. – Томск: Издательство Томского политехнического университета, 2001. – С. 23-25.
70. Дмитриев В.М. Интеллектуализация управления технологическими процессами на углеводородных месторождениях / В.М. Дмитриев, Т.В. Ганджа, Е.В. Истигечева, И.Я.Клепак. – Томск: В-Спектр, 2012. – 212 с.
71. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. Учебник. – СПб: Питер, 2000. – 384 с.

72. Современные численные методы решения обыкновенных дифференциальных уравнений / Дж. Холл, Дж. Уатт. – М.: Издательство «Мир», 1979. – 312 с.
73. Ильин В.А., Позняк Э.Г. Линейная алгебра: учебник для ВУЗов. – 6-е изд., стер. – М.: ФИЗМАТЛИТ, 2004. – 280 с.
74. Алгоритмы и программы проектирования автоматических систем / П.Д. Крутько, А.И. Максимов, Л.М. Скворцов; Под ред. П.Д. Крутько. – М.: Радио и связь, 1988. – 306 с.
75. Калабеков Б.А. Методы автоматизированного расчета электронных схем в технике связи: Учеб. пособие для вузов / Б.А. Калабеков, В.Ю. Лапидус, В.М. Малафеев. – М.: Радио и связь, 1990. – 272 с.
76. James W. Nilsson, Susan Riedel. Introduction to Multisim for Electric Circuits. – Prentice Hall. – 2011. – 144 pp.
77. Spice Circuit HandBook / Steven M. Sandler, Charles Hymowitz. – McGraw-Hill Company, 2006. – 311 s.
78. Амелина М.А., Амелин С.А. Программа схемотехнического моделирования MiCro-CAP. Версии 9, 10. – Смоленск, Смоленский филиал НИУ МЭИ, 2013. – 618 с.
79. Максимов А. Моделирование устройств на микроконтроллерах с помощью программы ISIS из пакета PROTEUS VSM // Радио. – 2005. – № 4. – С. 30-32.
80. Схиртладзе А.Г., Иванов В.И., Кареев В.Н. Гидравлические и пневматические системы. – Издание 2-е, дополненное. – М.: ИЦ МГТУ «Станкин», «Янус-К», 2003. – 544 с
81. SimHydraulics – гидравлика для Simulink [Электронный ресурс] – Режим доступа: <http://matlab.ru/products/simhydraulics>, свободный. Дата последнего обращения 22.01.2016 г.
82. Евроинтех – Thermal Solution – Пакет моделирования тепловых процессов SAUNA [Электронный ресурс] – Режим доступа: <http://www.eurointech.ru/sauna>, свободный

83. Thermal Software for heat sinks, circuit, board, enclosures, boxes, transformers, IC's, power transistors and other electronics components [Электронный ресурс] – Режим доступа: <http://www.thermalsoftware.com/>, свободный. Дата последнего обращения 22.01.2016 г.
84. Малыгин А. SolidWorks simulation. Термический анализ. [Электронный ресурс]. – Режим доступа: <http://cadregion.ru/solidworks-simulation/termicheskij-analiz.html>, свободный. Дата последнего обращения: 22.01.2016 г.
85. Мирошник И.В. Теория автоматического управления. Линейные системы. – СПб.: Питер, 2005. – 336 с.
86. Бусленко Н.П. Моделирование сложных систем. – М.: Издательство «Наука», 1978. – 400 с.
87. Максимей И.В. Имитационное моделирование на ЭВМ. – М.: Радио и связь, 1988. – 232 с.
88. Кельтон В. Имитационное моделирование / В. Кельтон, А. Лоу. – 3-е изд. – СПб.: Питер, 2004. – 847 с
89. Дискретно-событийное моделирование в логистике | Пекарня идей [Электронный ресурс] – Режим доступа: <http://www.i-bakery.ru/page/diskretno-sobotiinoe-modelirovanie-v-logistike>, свободный. Дата последнего обращения: 04.01.2016 г.
90. Гнеденко Б.В., Коваленко И.Н. Введение в теорию массового обслуживания. – М.: Издательство ЛКИ, 2007. – 400 с.
91. Баран Е.Д. LabVIEW FPGA. Реконфигурируемые измерительные и управляющие системы. – М.: ДМК Пресс, 2009. – 448 с.
92. National Instruments Россия, СНГ и Балтия [Электронный ресурс] – Режим доступа: <http://russia.ni.com/>, свободный. Дата последнего обращения 04.01.2016 г.
93. Украинская SCADA система КОНТУР [Электронный ресурс]. – Режим доступа: http://www.industrialauto.ru/modules/myarticles/article.php?mode=0&order=0&item_id=9, свободный. Дата последнего обращения 04.01.2016 г.

94. SCADA-система для электроэнергетики «АТЛАНТ» [Электронный ресурс]. – Режим доступа: <http://www.isup.ru/articles/2/6661/>, свободный. . Дата последнего обращения 04.01.2016 г.
95. Медведев С. Структура и основные функции Trace Mode 6 и T-Factory 6 [Электронный ресурс]. – Режим доступа [http://www.pcweek.ru /themes/detail.php?ID=65152&sphrase_id=24244](http://www.pcweek.ru/themes/detail.php?ID=65152&sphrase_id=24244), свободный. Дата последнего обращения 04.01.2016 г.
96. SCADA TRACE MODE. Российская SCADA для АСУ ТР [Электронный ресурс]. – Режим доступа <http://www.adastra.ru/>, свободный. Дата последнего обращения 04.01.2016 г.
97. OpenSCADA: Главная [Электронный ресурс]. – Режим доступа <http://oscada.diyaorg.dp.ua>, свободный. Дата последнего обращения 04.01.2016 г.
98. Карпов Ю.Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. – СПб.: БХВ-Петербург, 2006. – 400 с.
99. Грехем И. Объектно-ориентированные методы. Принципы и практика – Object-Oriented Methods: Principles & Ptactice. – 3-е изд. – М.: Вильямс, 2004. – 880 с.
100. Арайс Е.А. Автоматизация моделирования многосвязных механических систем / Е.А. Арайс, В.М. Дмитриев. – М.: Машиностроение, 1987. – 240 с.
101. Дмитриев В.М. Алгоритм формирования и вычисления математических выражений методом компонентных цепей / В.М. Дмитриев, Т.В. Ганджа // Математические машины и системы. – 2010. – № 3. –С. 9-21.
102. Дмитриев В.М. Система отображения математических выражений в язык компонентных цепей / В.М. Дмитриев, Т.В. Ганджа, М.А. Ерошкин / Компьютерные технологии в образовании. Под ред. В.М. Дмитриева. –Томск: Изд-во Том. ун-та, 2004. – С. 29-39.
103. Дмитриев В.М. Архитектура универсального вычислительного ядра для реализации виртуальных лабораторий / В.М. Дмитриев, А.В. Шутенков, Т.В. Ганджа // Приборы и системы. Управление. Контроль. Диагностика. –2004. – № 2. – С. 24-28.
104. Дмитриев В.М. Расчетно-моделирующая среда для учебных и научных лабораторий / В.М. Дмитриев, Т.В. Ганджа // Вестник Московского городско-

го педагогического университета. Серия «Информатика и информатизация образования». – 2004. – № 3. – С. 40-45

105. Дмитриев В.М. Задачи построения и конфигурирования компьютерных тренажеров / В.М. Дмитриев, Т.В. Ганджа // Дистанционные образовательные технологии. Выпуск 1. Пути развития. Сборник научных трудов. – Томск: ТУСУР, 2004. – С. 70-77.

106. Дмитриев В.М. ЛАРМ: Автоматизированный лабораторный комплекс по электротехнике и электронике: Учебное пособие для ВУЗов / В.М. Дмитриев, А.В. Шутенков, Т.В. Ганджа, А.Н. Кураколов. – Томск: В-Спектр, 2010. – 186 с.

107. Дмитриев В.М. Компьютерное моделирование физических задач / В.М. Дмитриев, А.Ю. Филиппов, Т.В. Ганджа, И.В. Дмитриев. – Томск: В-Спектр, 2010. – 248 с.

108. Дмитриев В.М. Редактор виртуальных инструментов и приборов / В.М. Дмитриев, Т.В. Ганджа, Т.Ю. Коротина // Приборы и системы. Управление. Контроль. Диагностика. – 2009. – № 6. – С. 19-24.

109. Зайченко Т.Н. Многовариантный анализ характеристик технических объектов в системе МАРС / Т.Н. Зайченко, Т.В. Ганджа // Радиотехнические и информационные системы и устройства. Тезисы докладов региональной научно-технической конференции. – Томск, 2000. – С. 118-119.

110. Условные графические обозначения в электрических схемах (действующие и отмененные). Краткий обзор [Электронный ресурс]. – Режим доступа: http://www.electromonter.info/handbook/symbol_all.html, свободный. (дата обращения 03.02.2015).

111. Кафаров В.В. Системный анализ химической технологии. Топологический анализ формализации / В.В. Кафаров, И.Н. Дорофов. – М.: Наука, 1979. – 304 с.

112. Цисарь И.Ф. Компьютерное моделирование экономики / И.Ф. Цисарь, В.Г. Нейман. – М.: Диалог-МИФИ, 2002. – 304 с.

113. Горбунов-Посадов М.М. Системное обеспечение пакетов прикладных программ / М.М. Горбунов-Посадов, Д.А. Корягин, В.В. Мартынюк – М.: Наука, 1990. – 208 с.

114. Горбунов-Посадов М.М. Расширяемые программы. – М.: Политех, 1999. – 336 с.
115. Lipschutz, Seymour, and Lipson, Mark. «Schaum's Outlines: Linear Algebra». Tata McGraw-hill edition. Delhi 2001. pp. 69-80.
116. Дмитриев В.М., Ганджа Т.В. Компьютерная модель управляемых технически сложных объектов // Информатика и системы управления. – 2012. – № 3 (33). – С. 47-59.
117. MPI: A Message-Passing Interface Standard. Version 3.1- Message Passing Interface Forum, September 21, 2021. – 822 P. [Электронный ресурс]. – Режим доступа: <http://www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf>, свободный. (дата обращения 06.02.2015)
118. Дмитриев В.М. Методика стратификации и интеграции компьютерной модели сложной технической управляемой системы / В.М. Дмитриев, Т.В. Ганджа, Т.Н. Зайченко // Информатика и системы управления. – 2016. – № 4. – С. 11-22.
119. Дмитриев В.М. Принцип формирования многоуровневых компьютерных моделей SCADA-систем для управления сложными технологическими объектами / В.М. Дмитриев, Т.В. Ганджа // Информатика и системы управления. – 2013. – № 2 (36). – С. 24-35.
120. Григорьева Т.Е. Дискретно-событийное моделирование в СМ МАРС для курса «Системы массового обслуживания» // Доклады ТУСУРа. – 2014. – № 1 (31). – С. 152-155.
121. Ганджа Т.В. Задачи и архитектура подсистемы документирования исследований в среде многоуровневого моделирования МАРС / Т.В. Ганджа, С.А. Панов // Доклады ТУСУРа. – 2011. – 2 (24). – Ч. 2. – С. 334-338.
122. Панов С.А. Автоматизация документирования в научно-исследовательской деятельности / С.А. Панов, Т.В. Ганджа // Современная техника и технологии: сборник трудов XVIII Международной научно-практической конференции студентов, аспирантов и молодых ученых. В 3 т. Т. 2. / Томский по-

литехнический университет. – Томск: Изд-во Томского политехнического университета, 2012. – С. 291-292.

123. Ганджа Т.В. Алгоритм формирования компонентных цепей систем математических выражений // Труды выпускников аспирантуры ТУСУР. Под ред. В.И. Карнышева. – Томск: Том. гос. ун-т систем управления и радиоэлектроники, 2005. – С. 49-57.

124. Дмитриев В.М. Принципы построения моделей сложных технологических объектов с неоднородными векторными связями / В.М. Дмитриев, Т.В. Ганджа, С.К. Важенин // Современные технологии. Системный анализ. Моделирование. – 2014. – № 1. – С. 104-111.

125. Дмитриев В.М. Компьютерное моделирование визуальных интерфейсов виртуальных инструментов и приборов / В.М. Дмитриев, Т.В. Ганджа, В.В. Ганджа, С.А. Панов // Научная визуализация. – 2016. – Т. 8. – № 3. – С. 111-131.

126. Свидетельство о государственной регистрации программы для ЭВМ № 2013612164. Среда компьютерного моделирования MAPC / В.М. Дмитриев, Т.Н. Зайченко, А.В. Шутенков, Т.В. Ганджа, В.В. Ганджа. – 15.02.2013. – М.: Роспатент, 2013.

127. Арайс Е.А., Дмитриев В.М. Моделирование неоднородных цепей и систем на ЭВМ. – М.: Радио и связь, 1982. – 160 с.

128. Буданов А.Н. Синтез алгоритмов обработки информации с использованием виртуальных интерфейсов в составе преобразовательных элементов сети передачи данных [Текст]: диссертация на соискание степени кандидата технических наук. Спец. 05.13.05 – Элементы и устройства вычислительной техники и систем управления / А.Н. Буданов; науч. рук. В.М. Дмитриев; Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Томский государственный университет систем управления и радиоэлектроники» (ТУСУР). – Томск: 2015. – 179 с.

129. Кафаров В.В. Основы автоматизированного проектирования химических производств / В.В. Кафаров, В.Н. Ветехин. – М.: Наука, 1987. – 623 с.

130. Дмитриев В.М. Построение компьютерных моделей многофракционных физико-химических систем газопромысловых объектов в формате метода компонентных цепей / В.М. Дмитриев, А.В. Шутенков, Т.В. Ганджа // Доклады ТУСУРа. – 2012. – № 2 (26). – Ч. 1. – С. 145-150.

131. Дмитриев В.М. Исследование механики левого желудочка при сердечной недостаточности для компьютерного моделирования / В.М. Дмитриев, Т.В. Ганджа, В.Х. Ваизов // Моделирование: Теория, методы и средства. Материалы XI междунар. науч.-практ. конф. г. Новочеркасск, 31 марта 2011 г. / Юж-Рос. техн. ун-т (НПИ). – Новочеркасск, ЮРГТУ, 2011. – С. 69-75.

132. Рид Р., Праусниц Дж., Шервуд Т. Свойства газов и жидкостей: Справочное пособие / Пер. с англ. под ред. Б.А. Соколова. – 3-е изд., перераб. и доп. – Л.: Химия, 1982. – 592 с., ил. – Нью-Йорк, 1977.

133. Маршалов Е.Д. Имитационное моделирование гидравлических систем с регулирующими органами / Е.Д. Маршалов, О.А. Нечаева // Вестник Ивановского государственного энергетического университета. – 2007. – № 4. – С. 84.

134. Кассина Н.В. Математическое моделирование разветвленных гидравлических систем / Н.В. Кассина, Л.В. Смирнов // Компьютерные исследования и моделирование. – 2009. – Т. 1. – № 2. – С. 173-179.

135. Терехов С.В. Моделирование тепловых и кинематических свойств реальных систем. – Донецк: «Вебер» (Донецкое отделение), 2007. – 30 с.

136. Теплотехника: учебн. для ВУЗов/ А.П. Баскаков, Б.В. Берг, О.К. Витт и др.: под ред. А.П. Баскакова. – 2-е изд., перераб. – М.: Энергоатомиздат, 1991. – 224 с., ил.

137. Гидравлический расчет трубопровода [Электронный ресурс]. – Режим доступа: <http://ars.gubkin.ru/rasthet.htm>, свободный. Дата обращения: 05.02.2017 г.

138. Акберов Р.Р. Особенности расчета фазового равновесия пар-жидкость многокомпонентных систем при использовании уравнения Соава-Редлиха-Квонга // Теоретические основы химической технологии. – 2011. – Т. 45. – № 3. – С.329-335.

139. Ананенков А.Г., Ставкин Г.П., Талыбов Э.Г. АСУ ТП промыслов газоконденсатного месторождения Крайнего Севера. – М.: ООО «Недра-Бизнесцентр», 1998. – 231 с.
140. Дмитриев В.М. Автоматическое регулирование расхода метанола при абсорбционной осушке природного газа в установке комплексной подготовки газа / В.М. Дмитриев, Т.В. Ганджа, И.Я. Клепак // Приборы. – 2008. – № 6. – С . 52-58.
141. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 3-е издание – Introduction to Algorithms, Third Edition. – М.: Вильямс, 2013. – 1328 с.
142. Страуструп Б. Язык программирования C++. Специальное издание. – М.: Бином, 2011. – 1136 с.
143. Грызлов В.И., Грызлова Т.П. Турбо Паскаль 7.0. – М.: «ДМК», 2000. – 416 с.
144. Пильщиков В.Н. Assembler. Программирование на языке ассемблера IBM PC. – М.: Диалог-МИФИ, 1999. – 288 с.
145. Introduction into IEC 61131-3 Programming Languages [Электронный ресурс]. – Режим доступа: http://www.plcopen.org/pages/tc1_standards/iec_61131_3/, свободный. (дата обращения 14.02.2015)
146. Петров И.В. Программируемые контроллеры. Стандартные языки и приемы прикладного программирования / под ред. В.П. Дьяконова. – М.: СОЛОН-Пресс, 2004. – 256 с.
147. Introduction to G Programming: Publish Date: Jan 29, 2009 [Электронный ресурс]. – Режим доступа: <http://www.ni.com/white-paper/7668/en/pdf>, свободный. (дата обращения 02.02.2015).
148. Пейч Л.И., Точилин Д.А., Поллак Б.П. LabVIEW для новичков и специалистов. – М.: Горячая линия – Телеком, 2004. – 384 с.
149. Ларман К. Применение UML 2.0 и шаблонов проектирования – Applying UML and Pattern: An Introduction to Object-Oriented Analysis and Design and Interactive Development. – 3-е изд. – М.: Вильямс, 2006. – 736 с.

150. Шмуллер Дж. Освой самостоятельно UML 2 за 24 часа. Практическое руководство. Sams Teach Yourself UML in 24 Hours. Complete Starter Kit. – М.: Вильямс, 2005. – 416 с.
151. Моделирование и анализ систем. IDEF-технологии: практикум / С. Черемных, И. Семенов, В. Ручкин. – М.: Финансы и статистика, 2006. – 192 с.
152. Минаев И.Г. Программируемые логические контроллеры в автоматизированных системах управления / И.Г. Минаев, В.М. Шарапов, В.В. Самойленко, Д.Г. Ушкур. 2-е изд. перераб. и доп. – Ставрополь: АГРУС, 2010. – 128 с.
153. Мараховский В.Б. Моделирование параллельных процессов. Сети Петри. Курс для системных архитекторов, программистов, системных аналитиков, проектировщиков сложных систем управления / В.Б. Мараховский, Л.Я. Розенблюм, А.В. Яковлев. – СПб.: Профессиональная литература, АйТи-Подготовка, 2014. – 400 с.
154. Венгжин А. Использование раскрашенных сетей Петри для моделирования и верификации параллельных устройств логического управления / А. Венгжин, А.Г. Короткевич // Радиоэлектроника и информатика. – 2000. – № 4(13). – С. 70-74
155. Парр Э. Программируемые контроллеры: руководство для инженера. – М.: БИНОМ. Лаборатория знаний, 2007. – 516 с.
156. Ахо А. Теория синтаксического анализа, перевода и компиляции: в 2 т.: пер. с англ. / А.В. Ахо, Дж.Д. Ульман. – М.: Мир. 1978. – Т. 1: Синтаксический анализ. – М.: Мир, 1978. – 612 с.
157. Ерошкин М.А. Язык представления математических выражений для реализации редактора Макрокалькулятора / М.А. Ерошкин, Т.В. Ганджа // Компьютерные технологии в образовании / Под ред. В.М. Дмитриева. – Томск: Изд-во Том. ун-та, 2004. – Вып. 2. – С. 23-28.
158. Панов С.А. Интерактивные отчетные формы для информационного обеспечения лабораторного эксперимента // Современное образование: актуальные проблемы профессиональной подготовки и партнерства с работодателем. Ма-

териалы международной научно-методической конференции. Томск, 30-31 января 2014 г. – Томск: Издательство ТУСУРа, 2014. – С. 84-85.

159. Суэринг С. Конверс Т. Парк Дж. PHP и MySQL. Библия программиста, 2-е издание = PHP 6 and MySQL 6 Bible. – М.: Диалектика, 2010. – 912 с.

160. Флэнаган Д., Мацумото Ю. Язык программирования Ruby = The Ruby Programming Language / пер. с англ. Н. Вильчинский. – 1-е изд. – СПб.: Питер, 2011. – 496 с.

161. Системное программное обеспечение / А.В. Гордеев, А.Ю. Молчанов. – СПб.: Питер, 2002. – 736 с.

162. Standard ECMA-6 7-bit coded Character Set. 6th edition (December 1991) [Электронный ресурс] – Режим доступа: <http://www.ecma-international.org/publications/standards/Ecma-006.htm>, свободный

163. Афонский А.А., Дьяконов В.П. Цифровые анализаторы спектра, сигналов и логики / Под ред. проф. В.П. Дьяконова. – М.: СОЛОН-Пресс, 2009. – 248 с.

164. Ганджа Т.В. Единый подход к решению дифференциальных уравнений и систем на Макрокалькуляторе // Научная сессия ТУСУР-2007. Материалы докладов Всероссийской научно-технической конференции студентов, аспирантов и молодых ученых. Томск, 3-7 мая 2007 г. – Ч. 5. – С. 221-223.

165. Квакернаак Х., Сиван Р. Линейные оптимальные системы управления. – М.: Издательство «Мир, 1976. – 652 с.

166. Кнут Д.Э. Искусство программирования : пер. с англ. / Д.Э. Кнут; ред. пер. Ю.В. Козаченко: Стэнфордский университет. – М.: Вильямс, 2005. – Т.1 Основные алгоритмы: монография / пер. С.Н. Тригуб, пер. Ю.Г. Гордиенко, пер. И.В. Красикова. – 3-е изд. – М.: Вильямс, 2005. – 712 с

167. Ахназарова С.Л. Методы оптимизации эксперимента в химической технологии: Учеб. пособие для хим.-технол. спец. вузов. – 2-е изд., перераб. и доп. – М.: Высш. шк., 1985. – 327 с., ил.

168. Григорьева Т.Е. Методика статистического анализа технических объектов в среде многоуровневого компьютерного моделирования / Т.Е. Григорьева, Т.В. Ганджа // Научная сессия ТУСУР-2016: материалы Международной научно-

технической конференции студентов, аспирантов и молодых ученых. Томск, 25-27 мая 2016 г. – Томск: В-Спектр, 2015. В 6 частях. – Ч. 3. – С. 97-99.

169. Дмитриев В.М. Интерактивная математическая панель для моделирования систем с информационными и энергетическими связями / В.М. Дмитриев, Т.В. Ганджа // Электронные средства и системы управления: Материалы Международной научно-практической конференции. – Томск: Изд-во Института оптики атмосферы СО РАН, 2004. – В трех частях. – Ч. 2. – С. 62-65.

170. Ганджа Т.В. Применение макрокомпонентов для решения задач функционального моделирования и проектирования // Моделирование. Теория, методы и средства: Материалы X Междунар. науч.-практ. конф., г. Новочеркасск, 5 апр. 2010 г. / Юж.-Рос. гос. техн. ун-т (НПИ). – Новочеркасск: ЮРГТУ, 2010. – С. 4-8.

171. Свидетельство об отраслевой регистрации разработки № 10233. Библиотека моделей компонентов среды моделирования MAPS для автоматизированного решения задач параметрического синтеза технических объектов / Т.В. Ганджа, И.Я. Клепак. – М.: ОФАП, 2008. Оpubл. Компьютерные учебные программы и инновации. – 2008. – № 9. – С. 139.

172. Корнеев В.В. Интеллектуальная обработка информации / В.В. Корнеев, А.Ф. Гареев, С.В. Васютин, В.В. Райх. – М.: «Нолидж», 2000. – 352 с.

173. Терещенко А.Г. Геоинформационные системы для мониторинга и анализа окружающей среды / А.Г. Терещенко, И.А. Сухаленцев, В.В. Соколов, А.Ю. Карпов, О.Н. Лаздина // Экология и промышленность России. – 2005. – № 1. – С. 22-24.

174. Ганджа Т.В. Модели компонентов эколого-экономической системы нефтегазодобывающего комплекса в формате метода компонентных цепей / Т.В. Ганджа, О.С. Затик // Известия Томского политехнического университета. – 2009. – Т. 314. – № 5. Управление, вычислительная техника и информатика. – С. 114-116.

175. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. = DataBase System: A Practical Approach to Design, Implementation and Management. – 3-е изд. – М.: Вильямс, 2003. – 1436 с.

176. Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель. SQL: полное руководство, 3-е издание = SQL: The Complete Reference, Third Edition. – М.: «Вильямс», 2014. – 960 с.

177. Ганджа Т.В. Компоненты для работы с базами данных в среде многоуровневого компьютерного моделирования MAPC / Т.В. Ганджа, С.А. Панов // Перспективы развития информационных технологий: сборник материалов XIX Международной научно-практической конференции / Под общ. ред. С.С. Чернова. – Новосибирск: Издательство ЦРНС, 2014. – С. 40-44.

178. Дейт К.Дж. Введение в системы баз данных = Introduction to Database Systems. – 8-е изд. – М.: Вильямс, 2005. – 1328 с.

179. Датчики: Справочное пособие / В.М. Шарапов, Е.С. Полищук, Н.Д. Кошевой, Г.Г. Ишанин, И.Г. Минаев, А.С. Совлуков. – Москва: Техносфера, 2012. – 624 с.

180. Федосов В.П., Нестеренко А.К. Цифровая обработка сигналов в LabView: учеб. пособие / под ред. В.П. Федосова. – М.: ДМК-Пресс, 2007. – 456 с.

181. Автоматизация физических исследований и эксперимента: компьютерные измерения и виртуальные приборы на основе Lab VIEW 7 / под ред. П.А. Бутырина. – М.: ДМК Пресс, 2005. – 264 с.

182. Блюм П. LabView: стиль программирования: пер. с англ.; под ред. П. Михеева. – М.: ДМК Пресс, 2008. – 400 с.

183. Дмитриев В.М. Система визуализации и управления вычислительным экспериментом в среде многоуровневого моделирования MAPC / В.М. Дмитриев, Т.В. Ганджа, Т.Ю. Коротина // Доклады ТУСУРа. – 2010. – № 1 (21). – Часть 2. – С. 149-155.

184. Вознесенский А.С. Электротехника и измерительная техника: Учебник для вузов / А.С. Вознесенский, В.Л. Шкуратник. – М.: Издательство Московского государственного горного университета, 2008. – 461 с.

185. Мальцев Ю.И. Язык управления механизмами X-Robot // Электронные средства и системы управления: Материалы докладов IX Международной

научно-практической конференции (30-31 октября 2013 г.): В 2 ч. – Ч. 2. – Томск: В-Спектр, 2013. – С. 114-118.

186. Сергеев А.Г. Метрология: учебное пособие для вузов / А.Г. Сергеев, В.В. Крохин. – М.: Логос, 2001. – 408 с.

187. Клаассен К.Б. Основы измерений. Электронные методы и приборы в измерительной технике. – М.: Постмаркет, 2000. – 352 с.

188. Мешков А.В., Тихомиров Ю.В. Visual C++ и MFC. В 2-х т. – СПб: БХВ-СПб, 2001. – 468 с., 482 с.

189. FTDI Chip Home Page [Электронный ресурс] – Режим доступа: <http://www.ftdichip.com/>, свободный (дата обращения 20.02.2015)

190. Микроконтроллеры семейства AVR XMEGA [Электронный ресурс] – Режим доступа: http://www.atmel.com/ru/ru/products/microcontrollers/avr/avr_xmega.aspx, свободный (дата обращения: 19.02.2015)

191. Дмитриев В.М. Система виртуальных инструментов и приборов для автоматизации учебных и научных экспериментов / В.М. Дмитриев, Т.В. Ганджа, С.А. Панов. // Программные продукты и системы / Software & System. – 2016. – № 3. – Т. 29. – С. 154-162.

192. Свидетельство о государственной регистрации программы для ЭВМ № 2014662641. Генератор моделей компонентов среды моделирования MAPC / Т.В. Ганджа. – 04.12.2014. – М.: Роспатент, 2014.

193. Синтес А. Освой самостоятельно объектно-ориентированное программирование на 21 день = Sams Teach Yourself Object-Oriented Programming in 21 Days. – М.: Вильямс, 2002. – 672 с.

194. Дмитриев В.М. Архитектура расчетно-моделирующей среды для виртуальных лабораторий / В.М. Дмитриев, Т.В. Ганджа // Дистанционные технологии в образовании. Выпуск 1. Пути развития. Сборник научных трудов. – ТУСУР, Томск, 2004. – С. 62-69.

195. Свидетельство о государственной регистрации программы для ЭВМ № 2009612036. Программное обеспечение «Расчетно-моделирующая среда» /

Т.В. Ганджа, Т.Н. Зайченко, А.В. Шутенков, А.Н. Кураколов. – 22.04.2009. – М.: Роспатент, 2009.

196. Дейтел Х.М., Дейтел П.Дж.. Как программировать на С++: Пер. с англ. – М.: ЗАО «Издательство БИНОМ», 1999 г. – 1024 с.: ил.

197. Distel R. Graph Theory. Electronic Edition. – NY.: Springer-Verlag, 2005. – 422 с.

198. Павловская Т.А. С/С++. Программирование на языке высокого уровня. – СПб.: Питер, 2003. – 461 с.

199. Nasri Z, Binous H. Application of the Soave-Redlich-Kwong Equation of State Using Mathematica // J. Chem. Eng. Japan. 2007. V. 40. № 6. 3. 534.

200. Шилов В.И., Клочков А.А., Ярышев Г.М. Расчет констант фазового равновесия компонентов природных нефтегазовых смесей // Нефтяное хозяйство. – 1987. – № 1. – С. 50-55.

201. CPaintDC Class [Электронный ресурс] –Режим доступа: <http://msdn.microsoft.com/ru-ru/library/a48eab8d.aspx>, свободный. Дата последнего обращения 28.02.2015г

202. Дмитриев В.М. Генератор моделей компонентов физически неоднородных цепей на базе интерактивной математической панели / В.М. Дмитриев, Т.В. Ганджа, Т.Ю. Коротина // Доклады ТУСУРа. – 2009. – № 2 (20). – С. 94-99.

203. Dmitriev V.M. An algorithm to improve the speed and accuracy of analysis of chemical process system operation / V.M. Dmitriev, T.V. Gandzha, I.M. Dolganov, Natalia V. Aksenova // Petroleum and Coal. – 2017. – Vol. 59. – Issue 4. – P. 429–441.

204. Дмитриев В..М. Методика построения многоуровневых компонентных цепей для моделирования химико-технологических систем / В.М. Дмитриев, Т.В. Ганджа // Доклады ТУСУРа. – 2017. – Т. 20. – № 3. – С. 82-87.

205. Дмитриев В.М. Определение значений параметров регулятора с помощью многоуровневой компьютерной модели / В.М. Дмитриев, Т.В. Ганджа, Т.Н. Зайченко // Доклады ТУСУРа. – 2017. – Т. 20. – № 2. – С. 91-95.

206. Ганджа Т.В. Применение интерактивной математической панели для реализации блоков обработки результатов / Т.В. Ганджа, Т.Ю. Коротина // Ин-

формационные технологии в социально значимых отраслях экономики (ИТСОЭ-3): Межвузовский сборник научных трудов / Том 3. Информационные технологии в обеспечении качества персонифицированных услуг. – Новосибирск: Редакционно-издательский центр Новосибирского государственного университета. –2009. – С. 50-59.

207. Карнаухов Н.Ф. Электромеханика и мехатронные системы. – Ростов н/Д: Феникс, 2006. – 320 с.

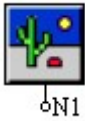


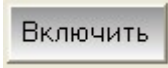






208. Тема 4.3. Компьютерные тренажеры в обучении персонала нефтегазовой отрасли [Электронный ресурс] – Режим доступа: <http://sstu.syzran.ru/epa/docs/ITiOvNGO/4.3.pdf>, свободный. Дата обращения – 12.02.2017.



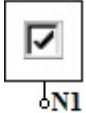
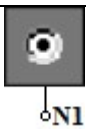
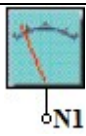
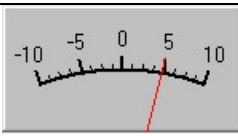
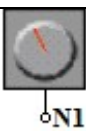

209. Дмитриев В.М., Буданов А.Н. Виртуальные интерфейсы для цифровых систем передачи мультисервисного трафика // Инфокоммуникационные технологии. – 2013. – Т. 11. – № 3. – С. 27-30.





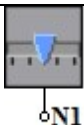
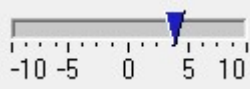

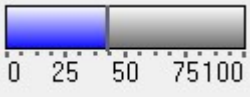
210. Дмитриев В.М. Структура сетевого тренажера для обучения и переподготовки операторов управляемых технологических объектов нефтегазовой промышленности / В.М. Дмитриев, Т.В. Ганджа, И.М. Долганов, М.О. Писарев. – Электронные средства и системы управления: Материалы докладов XI Международной конференции (25-27 ноября 2015 г.): В 2-х частях. – Ч. 2. – Томск: В-Спектр, 2015. – С. 279-284.

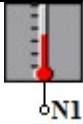
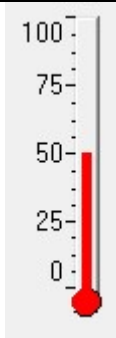
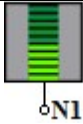






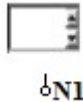
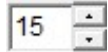
211. Ганджа Т.В. MARS-ENGINE – средство использования виртуальных приборов // Современное образование: практико-ориентированные технологии подготовки инженерных кадров: материалы междунар. науч.-метод. конф., 29-30 января 2015 г., Россия, Томск. – Томск: Изд-во Томск. гос. ун-та систем управления и радиоэлектроники, 2015. – С. 49-50.

Приложение 1. Набор моделей визуальных компонентов

Название компонента	Представление на логическом слое	Представление на визуальном слое	Атрибуты компонента	Принимаемое (отправляемое) сообщение
Кнопка с изображением			Количество состояний; Текущее состояние; Изображение при выключенном состоянии; Изображение при включенном состоянии.	Принимает сообщение об изменении состояния логического типа; Посылает сообщение другим компонентам при изменении состояния
Кнопка с надписью			Надпись; Количество состояний; Текущее состояние; Цвет фона 1; Цвет фона 2; Направление градиента.	Принимает сообщение об изменении состояния логического типа; Посылает сообщение другим компонентам при изменении состояния
Кнопка-переключатель			Состояние; Количество состояний; Текущее состояние; Цвет фона; Цвет переключателя 1; Цвет переключателя 2; Положение (вертикально или горизонтально)	Принимает сообщение об изменении состояния логического типа; Посылает сообщение другим компонентам при изменении состояния.
Круглая кнопка			Надпись; Количество состояний; Текущее состояние; Цвет кнопки в положении «Выключено»; Цвет кнопки в положении «Включено»	Посылает сообщение другим компонентам при изменении состояния путем нажатия на кнопку мышью; Принимает сообщение об изменении состояния логического типа
Флажковый переключатель			Надпись; Количество состояний (одно или два); Текущее состояние; Положение размещения флажка	Отправляет сообщение подключенным к узлу NI компонентам, в котором содержится информация о положении флажка

Название компонента	Представление на логическом слое	Представление на визуальном слое	Атрибуты компонента	Принимаемое (отправляемое) сообщение
Пружинный переключатель	 N1		Надпись; Количество состояний; Текущее состояние; Горизонтально	Отправляет сообщение подключенным к узлу N1 компонентам, в котором содержится информация о положении переключателя
Флажок	 N1	<input checked="" type="checkbox"/> Флажок1	Надпись; Количество состояний; Текущее состояние;	Отправляет сообщение подключенным к узлу N1 компонентам, в котором содержится информация о состоянии флажка
Группа радиокнопок	 N1	<input type="radio"/> Синусоида <input type="radio"/> Прямоугольник <input type="radio"/> Треугольник <input checked="" type="radio"/> Пила	Количество кнопок; Название каждой кнопки	На узел N1 отправляет сообщение с номером радиокнопки, которая была выбрана пользователем
Стрелочник	 N1		Минимальное значение; Максимальное значение; Текущее значение; Количество и видимость основных (с цифрами) и дополнительных (без цифр) меток; Шрифт; Цвет текста; Радиус шкалы; Цвет фона	Принимает сообщение на узел N1 и устанавливает стрелку на значение, принятое в сообщении вещественного или целочисленного типа
Регулятор	 N1	Регулятор1 	Минимальное значение; Максимальное значение; Текущее значение; Количество и видимость основных (с цифрами) и дополнительных (без цифр) меток; Шрифт; Цвет текста; Радиус шкалы; Цвета фона, ручки, шкалы, меток, указателя	Позволяет пользователю управлять круговым регулятором и отправляет сообщение на узел N1 с вещественным значением, соответствующим текущему положению рабочего органа регулятора

Название компонента	Представление на логическом слое	Представление на визуальном слое	Атрибуты компонента	Принимаемое (отправляемое) сообщение
Измеритель			Минимальное значение; Максимальное значение; Текущее значение; Количество и видимость основных (с цифрами) и дополнительных (без цифр) меток; Шрифт.	Принимает сообщение на узел <i>NI</i> и устанавливает стрелку на значение, принятое в сообщении вещественного или целочисленного типа
Регулятор с прямоугольным бегунком			Минимальное значение; Максимальное значение; Текущее значение; Количество и видимость основных (с цифрами) и дополнительных (без цифр) меток; Шрифт; Цвет текста, фона бегунка	Предназначен для задания пользователем с помощью органа управления (бегунка синего цвета) вещественного значения и его передачи другим компонентам с помощью узла <i>NI</i>
Регулятор с треугольным бегунком			Минимальное значение; Максимальное значение; Текущее значение; Количество и видимость основных (с цифрами) и дополнительных (без цифр) меток; Шрифт; Цвет текста, фона бегунка	Предназначен для задания пользователем с помощью органа управления (бегунка синего цвета) вещественного значения и его передачи другим компонентам с помощью узла <i>NI</i>
Прогресс			Минимальное значение; Максимальное значение; Текущее значение; Количество и видимость основных (с цифрами) и дополнительных (без цифр) меток; Шрифт; Цвет индикации пустой области; Цвет индикации заполненной области	Позволяет визуализировать вещественное значение, принятое на узел <i>NI</i> , путем заполнения области от минимального до текущего значения

Название компонента	Представление на логическом слое	Представление на визуальном слое	Атрибуты компонента	Принимаемое (отправляемое) сообщение
Термометр			Минимальное значение; Максимальное значение; Текущее значение; Количество и видимость основных (с цифрами) и дополнительных (без цифр) меток; Шрифт; Цвет индикации пустой области; Цвет индикации заполненной области	Позволяет визуализировать в виде термометра вещественное значение, принятое на узел <i>NI</i>
Эквалайзер			Минимальное значение; Максимальное значение; Текущее значение; Количество и видимость меток; Цвета заполненных и незаполненных меток; Положение (вертикально или горизонтально)	Предназначен для визуализации в виде эквалайзера принятого на узел <i>NI</i> вещественного значения
Цифровое табло			Вид числа (целое, вещественное или комплексное число); Формат числа (с фиксированной или с плавающей запятой); Количество знаков после запятой; Начальное значение действительной и мнимой части числа	Предназначено для визуализации целых, вещественных и комплексных чисел, принимаемых на узел <i>NI</i>
Надпись	-		Цвет и фон рамки, цвет и шрифт текста, текст	Предназначена для визуализации статической текстовой информации на лицевой панели прибора
Комбинированный список			Шрифт, набор строк, активная строка	Позволяет выбрать из списка один элемент и передает его номер на узел <i>NI</i>
Цифровое табло со спином			Минимальное, максимальное и текущее значения, шрифт	Позволяет задавать целочисленные значения в пределах от ми-

Название компонента	Представление на логическом слое	Представление на визуальном слое	Атрибуты компонента	Принимаемое (отправляемое) сообщение
				нимального до максимального значения и передавать его с помощью узла <i>NI</i>

Приложение 2. Пример исходного кода модели визуального компонента

```

//Конструктор класса CThermometerComponent
CThermometerComponent::CThermometerComponent(void)
{
    SetName("Термометр");
#ifdef SCHEMEVIEW
    VisualLayer.Init(VISUAL_LAYER,dtPaint);
    AddLayer(&VisualLayer);
    SchemeLayer.InitIcon(EVALUATE_LAYER,LoadIcon(AfxGetInstanceHandle(),
        MAKEINTRESOURCE(IDI_ICON_THERM)),CSize(40,40));
#endif
#ifdef SCHEMEVIEW
    SchemeLayer.Init(EVALUATE_LAYER,dtPaint);
#endif
    VisualLayer.InitialSize=CSize(50,150);
    AddLayer(&SchemeLayer);
    Pin.Init("Выход/Вход",CPoint(50,100),nsBottom,EVALUATE_LAYER);
    AddPin(&Pin);
    ATIN_LONG(Radius,"Радиус %");
    Radius=8;
    ATIN_LONG(H2,"Метки %");
    H2=5;
    ATIN_LONG(H3,"Надписи %");
    H3=0;
    Min=0;
    Max=100;
    Pos=50;
    Moving=false;
}

//Деструктор класса CThermometerComponent
CThermometerComponent::~CThermometerComponent(void)
{
}

//Функция прорисовки компонента на визуальном слое многослойного редактора
void CThermometerComponent::Paint(CDC &dc, long layer)
{
    if (GetSize(layer).cy<100)
        SetSize(CSize(30,100),layer);
    CRect Rect(CPoint(0,0),GetSize(layer));
    CString Str;
    Str.Format("%g",Max);
    CRect RectPr(0,0,0,0);
    dc.DrawText(Str,&RectPr,DT_CALCRECT);
    RealRad=Rect.Height()/100*Radius;
    DrawRect.SetRect(CPoint(Rect.right-(long)RealRad-(long)(RealRad*0.8),
    Rect.top),CPoint(Rect.right-(long)(RealRad*0.2),Rect.bottom-(long)RealRad));
    CRect DrawLabelRect(DrawRect.left,DrawRect.top,DrawRect.right,DrawRect.bottom);
    CRect EllRect(Rect.right-2*(long)RealRad,Rect.bottom-2*(long)RealRad,Rect.right,
    Rect.bottom-1);
    long _Value=(long)((DrawLabelRect.Height()-(long)RealRad)/(Max-Min)*(Pos-Min));
    CRect ActiveRect;
    ActiveRect=CRect(DrawLabelRect.left+4,DrawLabelRect.bottom-
    (long)RealRad-_Value,DrawLabelRect.right-4,DrawLabelRect.bottom);
}

```

```

CBrush Br,*oldbr;
Br.CreateSolidBrush(RGB(255,0,0));
CPen DarkPen,LightPen,EllPen,*oldpen;
EllPen.CreatePen(PS_SOLID,1,RGB(255,0,0));
DarkPen.CreatePen(PS_SOLID,1,RGB(0,0,0));
LightPen.CreatePen(PS_SOLID,1,RGB(255,255,255));
oldbr=dc.SelectObject(&Br);
dc.DrawEdge(DrawRect,EDGE_RAISED,BF_RECT);
oldpen=dc.SelectObject(&EllPen);
dc.Ellipse(EllRect);
dc.SelectObject(&LightPen);
dc.Arc(EllRect,CPoint(EllRect.CenterPoint().x,EllRect.top),CPoint(EllRect.left,
EllRect.bottom));
dc.SelectObject(&DarkPen);
dc.Arc(EllRect,CPoint(EllRect.left,EllRect.bottom),CPoint(EllRect.CenterPoint().x,
EllRect.top));
dc.FillRect(ActiveRect,&Br);
//Метки
if ((ShowTextLabel.Active==1)&&(NumberLabel>0))
{
    CRect rect(DrawLabelRect.left,DrawLabelRect.top,DrawLabelRect.right,
    DrawLabelRect.bottom-(long)RealRad);
    if (AddLab.Active==0)
        DrawLabelsVert(dc, rect, H2, H3, true, false);
    else
        DrawLabelsVert(dc, rect, H2, H3, true, true);
}
dc.SelectObject(oldpen);
dc.SelectObject(oldbr);
}

//Функция обработки события изменения значения атрибута
void CThermometerComponent::OnChangeAttribute(Attribute *pAttribute)
{
    SEMars->InvalidateAllView(this);
    CDoubleComponent::OnChangeAttribute(pAttribute);
}

//Функция обработки события перемещения мыши
void CThermometerComponent::OnMouseMove(UINT nFlags, CPoint point, View *view)
{
    if (Moving)
    {
        Pos=Min+(double)(Max-Min)/(double)((DrawRect.Height()+
        3-(long)RealRad)/(double)(DrawRect.Height()+3-(long)RealRad-point.y));
        if (Pos<Min)
            Pos=Min;
        if (Pos>Max)
            Pos=Max;
        SEMars->InvalidateAllView(this);
        SendKernel();
    }
}

//Функция обработки события нажатия левой кнопки мыши
void CThermometerComponent::OnLButtonDown(UINT nFlags, CPoint point, View *view)
{

```

```
CRect Rect(CPoint(0,0),GetSize(view->Layer));
if (PtInRect(Rect,point))
{
    Moving=true;
    view->pMainView->SetCapture();
}
}

//Функция обработки события отпущания левой кнопки мыши
void CThermometerComponent::OnLButtonUp(UINT nFlags, CPoint point, View *view)
{
    Moving=false;
    ReleaseCapture();
}

//функция обработки события изменения размера термометр на визуальном слое //редактора
void CThermometerComponent::OnSizing(int nHandle,CSize &size,long layer)
{
    if (layer==VISUAL_LAYER)
    {
        size.cy=1+(size.cy/10)*10;
        if (abs(size.cy)<10) size.cy=11;
        size.cx=15+size.cy/6;
    }
}
```

Приложение 3. Принципы работы пользователя с компонентом «График»

Специальный тип документа – **График** предназначен для вывода данных в виде различных кривых или в виде годографов с использованием усовершенствованного метода интерполяции, прекрасно работающего и на большом количестве точек. При активизации данного документа в главном меню появляется новый пункт – **График** (рис. ПЗ.1).

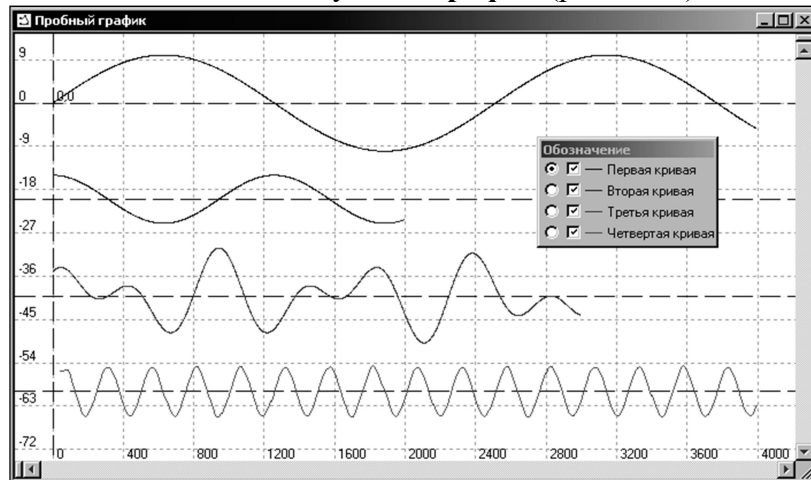


Рисунок ПЗ.1 – Окно **График**

Меню **График**, реализованное в редакторе среды многоуровневого компьютерного моделирования, содержит следующие команды управления графическим выводом выходной информации:

1. **Отображать метки** – управляет отображением точек, по которым строится график. При включенном режиме эти точки обводятся дополнительной окружностью или другой фигурой (в зависимости от установок). При отключенном режиме точки, по которым строится график, не выделяются.

2. **Отображать линейные метки** – управляет отображением линейных сканеров. Для изменения положения сканеров необходимо при нажатой клавише **Ctrl** левой кнопкой мыши (первый сканер – левый) или правой (второй сканер – правый) перетащить сканер на необходимую позицию. В окне **Обозначение** будет происходить отображение значений характеристик, соответствующих положению сканеров первого и второго, а также разностного значения между показаниями двух сканеров. Для того чтобы увидеть эти значения, необходимо изменить размер окна **Обозначения**.

3. **В одном масштабе** – управляет режимом масштабирования графиков. При выключенном режиме единого масштаба каждый график масштабируется на все окно, иначе все графики строятся в одном масштабе.

4. **Сглаживать сплайном** – включает/выключает режим кубической интерполяции. При включенном режиме производится кубическая интерполяция графика. При отключенном режиме точки просто соединяются по прямой линии, в этом случае график рисуется быстрее.

5. **Отображать окно Обозначение** – показывает/скрывает окно **Обозначение**. При включенном режиме окно **Обозначение** отображается в окне графика. При отключенном режиме – скрыто.

6. **Сетка** – отображает/скрывает масштабную сетку. При включенном режиме отображается масштабная сетка и масштабные метки. При отключенном режиме отображение графика производится без линий масштабной сетки и без масштабных меток. В случае если не выбран активный график в окне **Обозначение**, то масштабные метки выводятся только при включенном режиме единого масштаба. При выбранном активном графике отображаются масштабные метки и значения в соответствии с активным графиком.

7. **Свойства** – устанавливает свойства графика.

8. **Начальный размер** – устанавливает начальное значение масштаба. Это необходимо для возврата в исходное масштабирование, после отказа от дополнительного масштабирования.

9. **Режим точечного масштабирования** – управляет включением/отключением данного режима. При включенном режиме масштабирование изображения происходит с помощью нажатий левой и правой кнопок мыши. Для увеличения определенной части графика необходимо в точке этой части нажать правую кнопку мыши. Для уменьшения достаточно в любом месте графика нажать левую кнопку мыши. При увеличении или уменьшении точка, на которую указывает мышь, остается неподвижной, т.е. увеличение или уменьшение производится относительно точки, в которой нажата левая или правая кнопка мыши.

10. **Режим прямоугольного масштабирования** – управляет включением/отключением данного. При включенном режиме масштабирование изображения происходит в соответствии с прямоугольником, указанным с помощью мыши. Для увеличения определенной прямоугольной части графика необходимо нажать левую кнопку мыши в верхнем углу прямоугольника, и, не отпуская, перевести указатель мыши в противоположный угол. После этого график масштабируется таким образом, что видна только прямоугольная область графика, которая указана с помощью мыши.

11. **Импорт данных** – аналогична одноименной команде из меню **Таблица**.

Окно **Обозначение** (рис П3.2) предназначено для:

- отображения соответствия линии графика названию;
- выбора активного графика;
- управления видимостью графиков;
- выбора типа линии, меток, цвета графика.

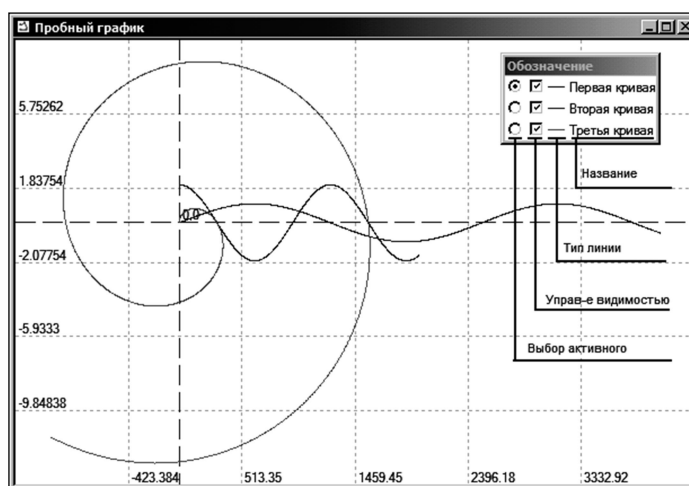


Рисунок П3.2 – График с окном **Обозначение**

Для выбора графика, по масштабу которого будут отображаться масштабные метки, предназначена первая колонка элементов управления. Для того чтобы выбрать активный график, достаточно щелкнуть мышкой на необходимый кружок.

Для управления видимостью графиков предназначена вторая колонка элементов управления – флажки. Поставленный флажок соответствует отображению графика, а снятый – скрытию.

Третья колонка отображает тип линии графика. Для изменения типа линии необходимо щелкнуть левой кнопкой мыши, в результате чего будет вызвано диалоговое окно **Выбор характеристик линии**.

И в последней колонке располагаются имена графиков.

Диалоговое окно **Выбор характеристик линии** (рис. П3.3) предназначено для выбора типа линии, меток и цвета графика.

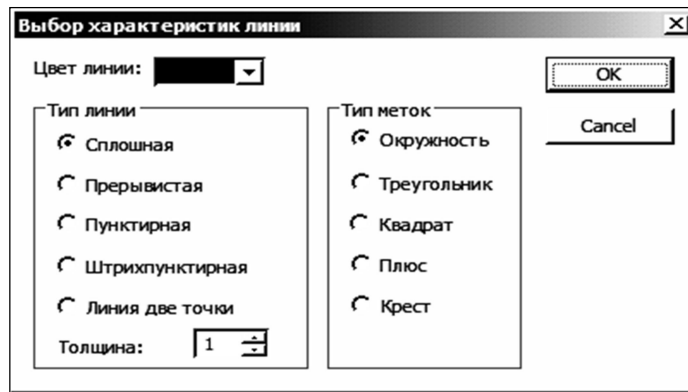


Рисунок П3.3 – Окно **Выбор характеристик линии**

В данном диалоговом окне располагаются следующие элементы:

- кнопка выбора **цвета линии**;
- группа **тип линии**, которая позволяет указать тип и толщину линии (толщина используется только при сплошном типе линии);
- группа **тип меток**, которая позволяет выбрать различную форму меток графика.

После выбора необходимых параметров линии нужно нажать кнопку **ОК**. Для отмены изменений – **Cancel**.

Приложение 4. Принципы использования компонента «Таблица»

Для вывода табличных данных в программе предусмотрен специальный тип документа – **Таблица** (рис. П4.1). Таблица позволяет выводить строковые, целочисленные, вещественные, логические и комплексные данные. При активизации таблицы в главном меню появляется новый пункт – **Таблица**.

№	Перв...	Пин	Назв...	Втор...	Пин	Назв...
1	T2	1	Эмитер	T1	1	Эмитер
2	T2	0	База	T3	1	Эмитер
3	T1	0	База	T2	2	Колле...
4	T3	2	Колле...	T3	0	База
5	T3	2	Колле...	T1	2	Колле...
6	T3	0	База	T1	2	Колле...

Рисунок П4.1 – Вид документа **Таблица**

Меню **Таблица** содержит команды управления графическим выводом выходной информации:

1. **Свойства таблицы** – меню, открывающее окно (рис. П4.2), в котором можно изменить характеристику столбцов таблицы, то есть задать имя, тип отображаемого значения и длину столбца.

2. **Импорт данных** – позволяет осуществить импортирование данных из других документов. Данная команда имеется также в меню **График** и **3D график**. В результате вызова этой команды на экране появится диалоговое окно **Импорт данных** (рис. П4.3).

№	Имя столбца	Тип данных	Длина столбца
1	Имя компонента	Строковой	100
2	Напряжение	Комплексный	100
3	Размер	Целочисленный	100
4	Отображать	Логический	100

Рисунок П4.2 – Окно **Свойства таблицы**

Доступные документы:
Пробный график

Характеристики:
Первая кривая
Вторая кривая
Третья кривая

Выбранные характеристики:
Пробный график:Первая кривая
Пробный график:Третья кривая

Рисунок П4.3 – Диалоговое окно **Импорт данных**

В данном диалоговом окне в верхней левой части располагается выпадающий список, в котором содержатся доступные для импортирования документы. После выбора документа в окне **Выбранные характеристики** появятся имеющиеся в нем характеристики. Для выбора пунктов в данном окне необходимо воспользоваться одним из следующих приемов:

1) для выбора одной характеристики достаточно один раз нажать левую кнопку мыши на требуемом пункте;

2) для выбора несколько подряд идущих характеристик необходимо один раз нажать левую кнопку мыши на первом пункте, затем нажать клавишу **Shift** и, не отпуская ее, кликнуть левой кнопкой мыши на последнем требуемом для выделения пункте;

3) для выбора нескольких характеристик необходимо воспользоваться клавишей **Ctrl**, в этом случае (при удерживаемой клавише **Ctrl**) левой кнопкой мыши можно изменять состояние выделения каждого пункта окна **Характеристики**.

Для добавления выбранных характеристик в текущий документ необходимо нажать кнопку **>>**, расположенную по центру диалогового окна.

Для того чтобы сохранить внесенные изменения, необходимо нажать кнопку **Ok**, иначе **Cancel**.

Приложение 5. Структура базы данных свойств веществ

Table1	
Имя поля	Тип данных
ID	Счетчик
SubstanceName	Текстовый
MM	Числовой
T	Числовой
C	Числовой
H	Числовой
Viscosity	Числовой
Ro	Числовой
O	Числовой
CT	Числовой
CPVAP_A	Числовой
CPVAP_B	Числовой
CPVAP_C	Числовой
CPVAP_D	Числовой
Pc	Числовой
Tc	Числовой
w	Числовой

Приложение 6. Реализации численного метода решения уравнения

Соава-Редлиха-Квонга

```
#define GET_COUNT_FRACTION int CountFraction = GetCountFraction();

//Метод решения кубического уравнения с поиском максимального вещественного корня
double IFizChimInterfece::CalcOfCubeEquations_MaxRoot
    (double a, double b, double c, double d)
{
#ifdef _FIZCHIM_DLL
    b *= -1;
    d *= -1;
    double Q = (b*b - 3 * c) / 9,
        R = (2 * b*b*b - 9 * b*c + 27 * d) / 54;
    complex x1, x2, x3;
    if (R*R > Q*Q*Q)
    {
        double V = (R / fabs(R));
        double s = fabs(R) + pow(R*R - Q*Q*Q, 0.5);
        double step = 1./3.;
        double A = -V*pow(s,step),
            B = (A != 0) ? Q / A : 0;
        x1 = A + B - (b / 3);
        x2 = -(A + B) / 2 - b / 3 + complex(0, 1)*pow(3., 0.5)*(A-B)/2;
        x3 = -(A + B) / 2 - b / 3 - complex(0, 1)*pow(3., 0.5)*(A - B) / 2;
        return x1.x;
    }
    else
    {
        complex t = acos(R / pow(Q*Q*Q, 0.5)) / 3;
        x1 = -2 * pow(Q, 0.5)*cos(t) - a / 3;
        x2 = -2 * pow(Q, 0.5)*cos(t + 2 * M_PI / 3) - a / 3;
        x3 = -2 * pow(Q, 0.5)*cos(t - 2 * M_PI / 3) - a / 3;
        double MAX = -1000000;
        if (x1.y == 0)
            MAX = x1.x;
        if ((x2.y == 0) && (MAX < x2.x))
            MAX = x2.x;
        if ((x3.y == 0) && (MAX < x3.x))
            MAX = x3.x;
        return MAX;
    }
#else
    return 0;
#endif
}

//Инициализация массивов для методики Соава-Редлиха-Квонга
void IFizChimInterfece::InitMetodicsCoavaRedlichCvong()
{
    GET_COUNT_FRACTION;

    m = new double[CountFraction];
}
```

```

    alfa = new double[CountFraction];
    Ap = new double[CountFraction];
    Pri = new double[CountFraction];
    Bpi = new double[CountFraction];
    Ab = new double*[CountFraction];
    Pri = new double[CountFraction];
    Tri = new double[CountFraction];
    FV = new double[CountFraction];
    Fl = new double[CountFraction];
    for (int f = 0; f < CountFraction; f++)
        Ab[f] = new double[CountFraction];

}
//Заккрытие методики Соава-Редлиха-Квонга
void IFizChimInterfece::CloseMetodicsCoavaRedlichCvong()
{
    GET_COUNT_FRACTION;
    delete[] Fl;
    delete[] FV;
    for (int f = 0; f < CountFraction; f++)
    {
        delete[] Ab[f];
    }
    delete[] Ab;
    delete[] Bpi;
    delete[] Pri;
    delete[] Ap;
    delete[] alfa;
    delete[] Tri;
    delete[] m;
}

//Реализация методики Соава-Редлиха-Квонга
void IFizChimInterfece::MetodicsCoavaRedlichCvong(double P, double T, double* x, double*
y, double* K)
{
    if ((T != T_old) || (P != P_old))
        CountIteration = 0;
    switch (CountIteration++)
    {
    case 1:
    {
        //Расчет начальных коэффициентов
        GET_COUNT_FRACTION;
        for (int f = 0; f < CountFraction; f++)
        {
            Pri[f] = /*0.145**/P / (Pci[f]**0.145*/);
            double tt = Tci[f] - 273.15, tt1 = T - 273.15;
            Tri[f] = (tt*1.8 + 491.67) / (tt1*1.8 + 491.67);
            K[f] = (Pci[f] / P) * exp(5.37*(1 + wi[f]) * (1 - Tri[f]));
            T_old = T; P_old = P;
        }
    }
}

```

```

}break;
case 0:
{
  //расчет коэффициентов
  GET_COUNT_FRACTION;
  for (int f = 0; f < CountFraction; f++)
  {
    K[f] = (1. / CountFraction);
  }
  T_old = T; P_old = P;
}break;
default:
{
  GET_COUNT_FRACTION;
  for (int f = 0; f < CountFraction; f++)
  {
    m[f] = 0.48 + 1.574*wi[f] - 0.176*pow(wi[f], 2);
    Tri[f] = ((T - 273.15)*1.8 + 491.67) / ((Tci[f] - 273.15)*1.8 + 491.67);
    Pri[f] = P / (Pci[f]);
    alfa[f] = pow((1 + m[f] * (1 - pow(Tri[f], 0.5))), 2);
    Ap[f] = 0.42747*alfa[f] * Pri[f] / pow(Tri[f], 2);
    Bpi[f] = 0.08664*Pri[f] / Tri[f];
  }
  for (int f = 0; f < CountFraction; f++)
  {
    for (int f1 = 0; f1 < CountFraction; f1++)
    {
      Ab[f][f1] = pow(Ap[f] * Ap[f1], 0.5);
    }
  }
  //Для газовой фазы
  double AvG = 0, BvG = 0; double A1 = 0, B1 = 0;
  for (int f = 0; f < CountFraction; f++)
  {
    for (int f1 = 0; f1 < CountFraction; f1++)
    {
      AvG += Ab[f][f1] * y[f] * y[f1];
      A1 += Ab[f][f1] * x[f] * x[f1];
    }
    BvG += Bpi[f] * y[f];
    B1 += Bpi[f] * x[f];
  }
  double A1 = (AvG - BvG - pow(BvG, 2)), B1 = AvG*BvG,
    A2 = (A1 - B1 - pow(B1, 2)), B2 = A1*B1;
  double Zv = CalcOfCubeEquations_MaxRoot(1., 1, A1, B1);
  double Z1 = CalcOfCubeEquations_MaxRoot(1, 1, A2, B2);
  for (int f = 0; f < CountFraction; f++)
  {
    double ar1 = (Zv - 1)*Bpi[f] / BvG - log(Zv - BvG) - (AvG / BvG)*
      (2 * pow(Ap[f], 0.5) / pow(AvG, 0.5) - Bpi[f] / BvG)*log((Zv + BvG) / Zv);
    FV[f] = pow(M_EXP, ar1); //exp(ar1);
    double ar2 = (Z1 - 1)*(Bpi[f] / B1) - log(Z1 - B1) - (A1 / B1)*(2 * pow(Ap[f], 0.5) /

```

```
        pow(A1, 0.5) - Bpi[f] / B1)*log((Z1 + B1) / Z1);
    F1[f] = pow(M_EXP, ar2); //exp(ar2);
    K[f] = (FV[f] != 0) ? (F1[f] / FV[f]) : F1[f];
}
}break;
}
}
```


**Приложение 7. Набор компонентов математических операций
в языке моделирования алгоритмических конструкций**

Таблица П7.1

Компоненты арифметических операторов языка МАК

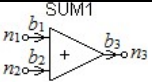
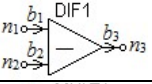
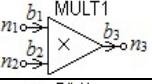
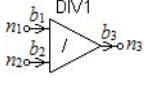

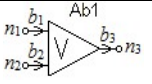
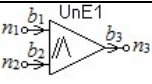
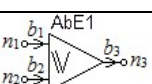
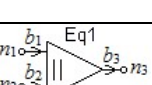
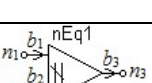
Название и обозначение	Визуальный вид отображающего компонента	Переменные связей компонента	Математическая модель	Типы данных и условия
Сложение «+»			$V_3 = V_1 + V_2$	V_1, V_2, V_3 – одного любого типа данных
Вычитание «-»		$S_1 = (b_1 \eta^- n_1) = \{V_1\}$	$V_3 = V_1 - V_2$	V_1, V_2, V_3 – одного любого типа данных, кроме строкового и массива строк. При действиях с массивами, они должны быть одной размерности
Умножение «·»		$S_2 = (b_2 \eta^- n_2) = \{V_2\}$	$V_3 = V_1 \cdot V_2$	
Деление «/»		$S_3 = (b_3 \eta^+ n_3) = \{V_3\}$	$V_3 = \frac{V_1}{V_2}$	

Таблица П7.2

Компоненты операторов сравнения языка МАК

Название и обозначение	Визуальный вид отображающего компонента	Переменные связей компонента	Математическая модель	Типы данных и условия
Меньше «<»			$V_3 = \begin{cases} 1 & \text{если } V_1 < V_2 \\ 0 & \text{если } V_1 \geq V_2 \end{cases}$	V_1, V_2 – целочисленного или вещественного типа данных V_3 – логического типа данных
Больше «>»			$V_3 = \begin{cases} 1 & \text{если } V_1 > V_2 \\ 0 & \text{если } V_1 \leq V_2 \end{cases}$	
Меньше или равно «≤»		$S_1 = (b_1 \eta^- n_1) = \{V_1\}$	$V_3 = \begin{cases} 1 & \text{если } V_1 \leq V_2 \\ 0 & \text{если } V_1 > V_2 \end{cases}$	
Больше или равно «≥»		$S_2 = (b_2 \eta^- n_2) = \{V_2\}$	$V_3 = \begin{cases} 1 & \text{если } V_1 \leq V_2 \\ 0 & \text{если } V_1 > V_2 \end{cases}$	
Равно «=»		$S_3 = (b_3 \eta^+ n_3) = \{V_3\}$	$V_3 = \begin{cases} 1 & \text{если } V_1 = V_2 \\ 0 & \text{если } V_1 \neq V_2 \end{cases}$	
Не равно «≠»			$V_3 = \begin{cases} 1 & \text{если } V_1 = V_2 \\ 0 & \text{если } V_1 \neq V_2 \end{cases}$	

Компоненты логических операторов языка МАК

Название и обозначение	Визуальный вид отображающего компонента	Переменные связей компонента	Математическая модель	Типы данных и условия
Отрицание « \neg »		$S_1 = (b_1 \eta^- n_1) = \{V_1\}$ $S_2 = (b_2 \eta^+ n_2) = \{V_2\}$	$V_2 = \neg V_1$	V_1, V_2, V_3 – логического, целочисленного или вещественного типа данных
Конъюнкция (логическое «И») « \wedge »		$S_1 = (b_1 \eta^- n_1) = \{V_1\}$	$V_3 = V_1 \wedge V_2$	
Дизъюнкция (логическое «ИЛИ») « \vee »		$S_2 = (b_2 \eta^- n_2) = \{V_2\}$ $S_3 = (b_3 \eta^+ n_3) = \{V_3\}$	$V_3 = V_1 \vee V_2$	

Таблица П7.4

Компоненты стандартных математических функций

Название и обозначение	Визуальный вид отображающего компонента	Переменные связей компонента	Математическая модель	Типы данных и условия
Степенные, показательные и иррациональные функции				
Степень		$S_1 = (b_1 \eta^- n_1) = \{V_1\}$ $S_2 = (b_2 \eta^- n_2) = \{V_2\}$ $S_3 = (b_3 \eta^+ n_3) = \{V_3\}$	$V_3 = V_1^{V_2}$	
Экспонента		$S_1 = (b_1 \eta^- n_1) = \{V_1\}$ $S_2 = (b_2 \eta^+ n_2) = \{V_2\}$	$V_2 = e^{V_1}$	
Корень		$S_1 = (b_1 \eta^- n_1) = \{V_1\}$ $S_2 = (b_2 \eta^- n_2) = \{V_2\}$ $S_3 = (b_3 \eta^+ n_3) = \{V_3\}$	$V_3 = \sqrt[n_2]{V_1}$	
Квадратный корень		$S_1 = (b_1 \eta^- n_1) = \{V_1\}$ $S_2 = (b_2 \eta^+ n_2) = \{V_2\}$	$V_2 = \sqrt{V_1}$	
Логарифм		$S_1 = (b_1 \eta^- n_1) = \{V_1\}$ $S_2 = (b_2 \eta^- n_2) = \{V_2\}$ $S_3 = (b_3 \eta^+ n_3) = \{V_3\}$	$V_3 = \log_{V_2} V_1$	
Натуральный логарифм		$S_1 = (b_1 \eta^- n_1) = \{V_1\}$ $S_2 = (b_2 \eta^+ n_2) = \{V_2\}$	$V_2 = \ln(V_1)$	
Десятичный логарифм			$V_2 = \lg(V_1)$	

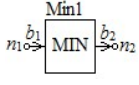
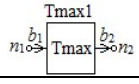
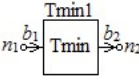
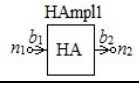
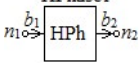
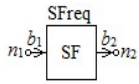
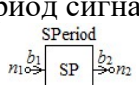
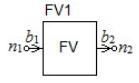
Тригонометрические функции				
Синус		$S_1 = (b_1 \eta^- n_1) = \{V_1\}$ $S_2 = (b_2 \eta^+ n_2) = \{V_2\}$	$V_2 = \sin(V_1)$	
Косинус			$V_2 = \cos(V_1)$	
Тангенс			$V_2 = \operatorname{tg}(V_1)$	
Котангенс			$V_2 = \operatorname{ctg}(V_1)$	
Секанс			$V_2 = \operatorname{sec}(V_1)$	
Косеканс			$V_2 = \operatorname{cosec}(V_1)$	
Обратные тригонометрические функции				
Арксинус		$S_1 = (b_1 \eta^- n_1) = \{V_1\}$ $S_2 = (b_2 \eta^+ n_2) = \{V_2\}$	$V_2 = \arcsin(V_1)$	
Аркосинус			$V_2 = \arccos(V_1)$	
Арктангенс			$V_2 = \operatorname{arctg}(V_1)$	
Арккотангенс			$V_2 = \operatorname{arcctg}(V_1)$	
Арксеканс			$V_2 = \operatorname{arcsec}(V_1)$	
Аркосеканс			$V_2 = \operatorname{arccsc}(V_1)$	
Гиперболические тригонометрические функции				
Синус гиперболический		$S_1 = (b_1 \eta^- n_1) = \{V_1\}$ $S_2 = (b_2 \eta^+ n_2) = \{V_2\}$	$V_2 = \sinh(V_1)$	
Косинус гиперболический			$V_2 = \cosh(V_1)$	
Тангенс гиперболический			$V_2 = \operatorname{tgh}(V_1)$	
Котангенс гиперболический			$V_2 = \operatorname{ctgh}(V_1)$	

Секанс гиперболический			$V_2 = \operatorname{sech}(V_1)$	
Косеканс гиперболический			$V_2 = \operatorname{cosech}(V_1)$	
Обратные гиперболические тригонометрические функции				
Арксинус гиперболический		$S_1 = (b_1 \eta^- n_1) = \{V_1\}$ $S_2 = (b_2 \eta^+ n_2) = \{V_2\}$	$V_2 = \operatorname{arcsinh}(V_1)$	
Арккосинус гиперболический			$V_2 = \operatorname{arccosh}(V_1)$	
Арктангенс гиперболический			$V_2 = \operatorname{arctgh}(V_1)$	
Арккотангенс гиперболический			$V_2 = \operatorname{arcctgh}(V_1)$	
Арксеканс гиперболический			$V_2 = \operatorname{arcsech}(V_1)$	
Арккосеканс гиперболический			$V_2 = \operatorname{arccosech}(V_1)$	

Таблица П7.5

Компоненты блоков обработки данных

Название и графическое представление	Переменные связей компонента	Математическая модель	Физический смысл параметров и переменных
Общематематические блоки обработки			
Суммирование 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{V}\}$ $S_1 = (b_1 \eta^+ n_2) = \{S\}$	$S = \sum_{i=1}^n V_i$	\mathbf{V} – вектор значений функции; n – размер вектора \mathbf{V}
Произведение 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{V}\}$ $S_1 = (b_1 \eta^+ n_2) = \{P\}$	$P = \prod_{i=1}^n V_i$	\mathbf{V} – вектор значений функции; n – размер вектора \mathbf{V}
Максимальное значение 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{V}\}$ $S_1 = (b_1 \eta^+ n_2) = [M]$	$M = \max_{i=1, n} (V_i)$	\mathbf{V} – вектор значений функции; n – размер вектора \mathbf{V}

Название и графическое представление	Переменные связей компонента	Математическая модель	Физический смысл параметров и переменных
Минимальное значение 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{V}\}$ $S_1 = (b_1 \eta^+ n_2) = \{M\}$	$M = \min_{i=1,n} (V_i)$	\mathbf{V} - вектор значений функции; n - размер вектора \mathbf{V}
Значение абсциссы при максимальном значении 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{V}\}$ $S_1 = (b_1 \eta^+ n_2) = \{T_{\max}\}$	$T_{\max} = t _{V(t)=\max_{i=1,n}(V_i)}$	\mathbf{V} - вектор значений функции; n - размер вектора \mathbf{V}
Значения абсциссы при минимальном значении 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{V}\}$ $S_1 = (b_1 \eta^+ n_2) = \{T_{\min}\}$	$T_{\min} = t _{V(t)=\min_{i=1,n}(V_i)}$	\mathbf{V} - вектор значений функции; n - размер вектора \mathbf{V}
Блоки обработки установившихся периодических процессов во временной области			
Амплитуда гармоника 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{V}(t)\}$ $S_1 = (b_1 \eta^+ n_2) = \{A\}$	$A = \text{Ampl}(F(V(t), n))$	$\mathbf{V}(t)$ - вектор значений временной функции; n - заданный номер гармоники; $F(V(t))$ - функция разложения сигнала $V(t)$ в ряд Фурье
Фаза гармоника 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{V}(t)\}$ $S_1 = (b_1 \eta^+ n_2) = \{\varphi\}$	$\varphi = \text{Phase}(F(V(t), n))$	
Частота сигнала 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{V}(t)\}$ $S_1 = (b_1 \eta^+ n_2) = \{\omega\}$	$\omega = \text{Freq}(F(V(t)))$	
Период сигнала 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{V}(t)\}$ $S_1 = (b_1 \eta^+ n_2) = \{T\}$	$T = \frac{1}{\text{Freq}(F(V(t)))}$	
Блоки обработки переходных процессов			
Установившееся значение 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{V}(t)\}$ $S_1 = (b_1 \eta^+ n_2) = \{FV\}$	$FV = \lim_{\substack{a \rightarrow 0 \\ i \rightarrow \infty}} \frac{\text{LMAX}(V_{i+1}) + \text{LMIN}(V_i)}{2}$ $a = \text{LMAX}(V_i) - \text{LMIN}(V_i)$	i - номер локального максимума или минимума; LMAX - функция определения локального

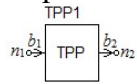
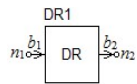
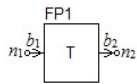
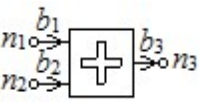
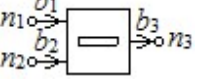
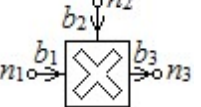
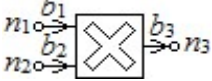
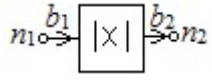
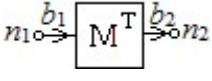
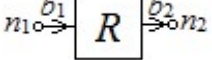
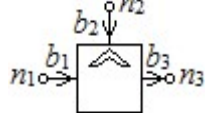
Название и графическое представление	Переменные связей компонента	Математическая модель	Физический смысл параметров и переменных
			максимума; LMIN – функция определения локального минимума
Время переходного процесса 	$S_1 = (b_1 \eta^- n_1) = \{V(t)\}$ $S_1 = (b_1 \eta^+ n_2) = \{TPP\}$	$TPP =$ $\min_{T_p} \{T_p : V_{n1}(t) - FV(V_{n1}) > eps, t\}$	eps – заданная точность нахождения;
Перерегулирование 	$S_1 = (b_1 \eta^- n_1) = \{V(t)\}$ $S_1 = (b_1 \eta^+ n_2) = \{DR\}$	$DR = \frac{MAX(V_{n1}) - FV(V_{n1})}{FV(V_{n1})} \cdot 100\%$	
Период свободных колебаний 	$S_1 = (b_1 \eta^- n_1) = \{V(t)\}$ $S_1 = (b_1 \eta^+ n_2) = \{DR\}$	$FP(V(t)) = MAX_L_T(V(t), i) - MAX_L_T(V_{n1}, i - 1)$	$MAX_L_T(V(t), i)$ – функция определения времени наступления максимума с номером i в массиве V_{n1}

Таблица П7.6

Компоненты векторно-матричных операций

Оператор	Переменные связей компонента	Математическая модель	Типы данных и условия выполнения
Сложение матриц 	$S_1 = (b_1 \eta^- n_1) = \{M_1\}$ $S_2 = (b_2 \eta^- n_2) = \{M_2\}$ $S_3 = (b_3 \eta^+ n_3) = \{M_3\}$	$M_3 = M_1 + M_2$	Выполняется, если размеры матриц M_1 и M_2 совпадают, в этом случае матрица M_3 будет иметь те же размеры
Вычитание матриц 	$S_1 = (b_1 \eta^- n_1) = \{M_1\}$ $S_2 = (b_2 \eta^- n_2) = \{M_2\}$ $S_3 = (b_3 \eta^+ n_3) = \{M_3\}$	$M_3 = M_1 - M_2$	Выполняется, если размеры матриц M_1 и M_2 совпадают, в этом случае матрица M_3 будет иметь те же размеры
Умножение матрицы на скаляр 	$S_1 = (b_1 \eta^- n_1) = \{M_1\}$ $S_2 = (b_2 \eta^- n_2) = \{C_2\}$ $S_3 = (b_3 \eta^+ n_3) = \{M_3\}$	$M_3 = C_2 \cdot M_1$	Выполняется при любых условиях, размер матрицы M_3 будет совпадать с размером матрицы M_1

<p>Умножение матриц</p> 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{M}_1\}$ $S_2 = (b_2 \eta^- n_2) = \{\mathbf{M}_2\}$ $S_3 = (b_3 \eta^+ n_3) = \{\mathbf{M}_3\}$	$\mathbf{M}_3 = \mathbf{M}_1 \cdot \mathbf{M}_2$	<p>Выполняется, если число столбцов матрицы \mathbf{M}_1 равно количеству строк матрицы \mathbf{M}_2. В этом случае размер матрицы \mathbf{M}_3 будет соответствовать числу строк матрицы \mathbf{M}_1 и числу столбцов матрицы \mathbf{M}_2</p>
<p>Определитель матрицы</p> 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{M}_1\}$ $S_2 = (b_2 \eta^+ n_2) = \{D\}$	$D = \det(\mathbf{M}_1)$	<p>Выполняется, если матрица \mathbf{M}_1 - квадратная, то есть число ее строк, равно числу ее столбцов</p>
<p>Транспонирование матрицы</p> 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{M}_1\}$ $S_2 = (b_2 \eta^+ n_2) = \{\mathbf{M}_2\}$	$\mathbf{M}_2 = \mathbf{M}_1^{-1}$	<p>Выполняется, если $\det(\mathbf{M}_1) \neq 0$. Размер матрицы \mathbf{M}_2 равен размеру матрицы \mathbf{M}_1</p>
<p>Ранг матрицы</p> 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{M}_1\}$ $S_2 = (b_2 \eta^+ n_2) = \{r\}$	$r = \text{rank}(\mathbf{M}_1)$	<p>Выполняется, если матрица \mathbf{M}_1 - квадратная, то есть число ее строк, равно числу ее столбцов</p>
<p>Возведение матрицы в степень</p> 	$S_1 = (b_1 \eta^- n_1) = \{\mathbf{M}_1\}$ $S_2 = (b_2 \eta^- n_2) = \{C_2\}$ $S_3 = (b_3 \eta^+ n_3) = \{\mathbf{M}_3\}$	$\mathbf{M}_3 = (\mathbf{M}_1)^{C_2}$	<p>Выполняется, если матрица \mathbf{M}_1 - квадратная, а C_2 - целое</p>

Приложение 8. Реализация класса *CComponent*

```

class CComponent :public IComponent
{
public:
    //Конструктор класса
    CComponent();
    //Деструктор класса
    virtual ~CComponent();
    //Метод инициализации основных полей компонента
    virtual void Init(long code, ISEmars* semars=NULL);
    //Добавление метки для имени компонента
    void AddLabelName(long Layer);
    //метод отрисовки компонента в редакторе
    virtual void OnPaint(CDC& dc,long layer);
    //Реакция компонента на изменение значение атрибута
    virtual void OnChangeAttribute(Attribute* pAttribute);
    //Установка размера изображения компонента на соответствующем слое
    virtual void SetSize(CSize size,long layer);
    //Установка точки начала прорисовки компонента
    virtual void SetPosition(CPoint point,long layer);
    //Одновременная установка позиции и размера компонента
    virtual void SetSizePosition(CSize size,CPoint point,long layer);
    //Установка единого размера компонента на всех слоях
    virtual void SetSizeForAllLayers(CSize size);
    //Установка позиции компонента на всех слоях
    virtual void SetPositionForAllLayers(CPoint point);
    //Получение размера компонента на указанном слое редактора
    virtual CSize GetSize(long layer);
    //Получение позиции компонента на указанном слое редактора
    virtual CPoint GetPosition(long layer);
    //Получение коэффициента вращения компонента на указанном слое редактора
    virtual long GetRotate(long layer);
    //Установка коэффициента вращения компонента на указанном слое редактора
    virtual void SetRotate(long rotate,long layer);
    //Доступность вращения на указанном слое редактора
    virtual bool AllowRotate(long layer);
    //Получение имени компонента
    virtual CString GetName();
    //Установка имени компонента
    virtual void SetName(CString name);
    //Получение слоев, на который отображен компонент
    virtual long GetLayers() {return Layers;};
    //Добавление изображение компонента на определенном слое редактора
    virtual void AddView(View *pView);
    //Удаление изображения компонента на определенном слое редактора
    virtual void RemoveView(View *pView){};
    //Удаление изображений компонента на всех слоях редактора
    virtual void RemoveAllView(){};
    //Метод инициализации перед запуском модели
    virtual void InitOnRun(){Run=true;};
    //Метод освобождения памяти после завершения эксперимента
    virtual void DestroyOnStop(){Run=false;};

```



```

//Получение количества пинов компонента
virtual int GetCountPins() {return (int)ArrayPins.GetSize();};
//Получение количества меток компонента
virtual int GetCountLabels() {return (int)ArrayLabels.GetSize();};
//Получение количества атрибутов компонента
virtual int GetCountAttributes() {return (int)ArrayAttributes.GetSize();};
//Получение определенного узла компонента
virtual IPin* GetIPin(long number){return (IPin*)ArrayPins[number];};
Получение определенной метки компонента
virtual ILabel* GetILabel(long number){return (ILabel*)ArrayLabels[number];};
//Получение определенного атрибута компонента по его номеру
virtual Attribute* GetAttribute(long number);
//Получение определенного атрибута компонента по его имени
virtual Attribute* GetAttribute(CString NameAttr);
//Установка параметров вращения компонента
virtual CPoint Rotate(CPoint p,long rotate,long dx,long dy);
//Установка номера динамически загружаемой библиотеки
virtual void SetDllNumber(long nDll){DllNumber=nDll;};
//Получение номера динамически загружаемой библиотеки
virtual long GetDllNumber(){return DllNumber;};
//Установка имени динамически загружаемой библиотеки
virtual void SetDllName(CString name){DllName=name;};
//Получение имени динамически загружаемой библиотеки
virtual CString GetDllName(){return DllName;};
protected:
//Номер динамически загружаемой библиотеки
long DllNumber;
//Имя динамически загружаемой библиотеки
CString DllName;
//Массив слоев, на которых отображается компонент
CArray <CLayer*,CLayer*> ArrayLayers;
//Массив узлов компонента
CArray <CPinComponent*,CPinComponent*> PinsMassive;
//Массив меток (текстовых надписей)
CArray <CLabelComponent*,CLabelComponent*> ArrayLabels;
//Массив атрибутов
CArray <Attribute*,Attribute*> ArrayAttributes;
//Массив параметров
double* ParametersMassiv;
//Количество параметров
Int CountParameters;
//Добавление слоя отображения компонента
virtual void AddLayer(CLayer* pCLayer);
//Добавление узла компонента
void AddPin(CPinComponent* pPin);
//Добавление метки компонента
void AddLabel(CLabelComponent* pLabel);
//Добавление атрибута компонента
virtual void AddAttribute(Attribute* pAttribute);
//Флаг запуска эксперимента
bool Run;
//Получение слоя компонента по его номеру

```

```

CLayer* GetLayer(long layer);
//Получение типа отображения компонента на соответствующем слое
virtual EDrawType GetDrawType(long Layer);
//Метод динамической прорисовки компонента
virtual void Paint(CDC& dc,long layer){};
//Массив метод компонента
CArray <CLabelComponent*,CLabelComponent*> LabelsName;
//Уникальный идентификационный номер компонента
long UID;
public:
//Реакция на движение указателя мыши на компоненте
virtual void OnMouseMove(UINT nFlags, CPoint point,View *view){};
//Реакция на нажатие левой кнопки мыши
virtual void OnLButtonDown(UINT nFlags, CPoint point,View *view){};
//Реакция на нажатие правой кнопки мыши
virtual void OnRButtonDown(UINT nFlags, CPoint point,View *view){};
//Реакция на отпуск левой кнопки мыши
virtual void OnLButtonUp(UINT nFlags, CPoint point,View *view){};
//Реакция на отпуск правой кнопки мыши
virtual void OnRButtonUp(UINT nFlags, CPoint point,View *view){};
//Реакция на прокручивание колеса мыши вперед
virtual void OnMouseOver(UINT nFlags, CPoint point,View *view){};
//Реакция на прокручивание колеса мыши назад
virtual void OnMouseWheel(UINT nFlags, CPoint point,View *view){};
//Реакция на установку курсора мыши на компоненте
virtual BOOL OnSetCursor(CPoint point,View *view){return FALSE;};
//Реакция на двойной щелчок левой кнопки мыши
virtual void OnLButtonDblClk(UINT nFlags, CPoint point,View *view){OnDblClk();};
//Реакция на изменение размера
virtual void OnSize(long layer){};
//Метод, вызываемый в начале эксперимента
virtual void OnRun(){};
//Метод, вызываемый в конце эксперимента
virtual void OnStop(){};
//Метод сериализации (записи/чтения) данных компонента
virtual void Serialize(CArchive &ar);
private:
//Набор поддерживаемых слоев
long Layers;
//Номер класса компонента
long ClassNumber;
public:
//Обновление меню компонента
virtual void UpdateMenu(CMenu*Menu);
//Реакция на вызов команд меню компонента
virtual void OnCommand(long number);
//Реакция на двойной щелчок по изображению компонента
virtual void OnDblClk(void);
//Получение места компонента в дереве компонента
virtual HTREEITEM GetItem() {return item;};
//Установка компонента в дерево компонентов
virtual void SetItem(HTREEITEM Item) {item=Item;};

```

```
    //Запрос интерфейса компонента
    virtual void GetInterface(long IDInterface,void **pInterface){};
protected:
    //Место компонента в дереве компонентов
    HTREEITEM item;
public:
    //Указатель на документ, в который добавлен компонента
    ISEMars* SEMars;
    //Реакция на присоединение связи к компоненту
    virtual void ConnectPin(long number){};
    //Реакция на удаление связи компонента
    virtual void UnConnectPin(long number){};
    //Установка уникального идентификационного номера компонента
    virtual void SetUID(long _UID){UID=_UID;};
    //Получение уникального идентификационного номера компонента
    virtual long GetUID() {return UID;};
};
```

Приложение 9. Программно-алгоритмическая реализация компонента «Абсорбер»

1. Описание класса *CAbsorber*

```

class CAbsorber :
    public IRedactComponent, ISerialize, IBaseCTOModel, IFizChimInterfece
{
    //Площадь поверхности контакта фаз
    double S;
    //Количество абсорбируемых веществ
    long CountAbsFraction;
    //Старое количество абсорбируемых веществ, применяется
//для изменения количества атрибутов компонента
    long OldCountAbsFraction;
//Объем вещества, переходящего через границу раздела фаз
    double G;
    //Массив информации об абсорбируемых веществах
    CArray <sAbsorbFraction*, sAbsorbFraction*> mAbsorbFraction;
    //флаг начала эксперимента
    bool First;
    //Массивы для вычисления концентраций веществ в фазах на выходе абсорбера
    double* GC, *pG, *pF;
public:
    //Конструктор класса
    CAbsorber(void);
    //Деструктор класса
    virtual ~CAbsorber(void);
    //Информация о компоненте для формирования дерева компонентов
    static SComponentInfo* SetInfoComponent(HINSTANCE pInst);
    //Запрос интерфейсов компонента
    virtual void GetInterface(long ID, void** pInterface);
    //Метод начала эксперимента
    virtual void OnRun();
    //Метод реакции компонента на изменение значений атрибутов компонента
    virtual void OnChangeAttribute(Attribute* pAttr);
    //Метод сериализации (чтения и записи данных в файл)
    virtual void Serialize(CArchive& ar);
    //Метод формирования локального координатного базиса компонента
    virtual void SetMarsVarsConnection();
    //Метод опроса топологического блока компонента
    virtual bool TopologQuery(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);
    //Метод опроса вычислительного блока компонента
    virtual bool CalculatQuery(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);
};

```

2. Реализация методов класса *CAbsorber*

```

#include "StdAfx.h"
#include "Absorber.h"
#include "resource.h"

#define P1 pEq->V[Nodes[0]]
#define P2 pEq->V[Nodes[1]]
#define P3 pEq->V[Nodes[2]]
#define P4 pEq->V[Nodes[3]]

#define G1 pEq->V[Branch[0]]
#define G2 pEq->V[Branch[1]]
#define G3 pEq->V[Branch[2]]
#define G4 pEq->V[Branch[3]]

#define T1 pEq->V[Nodes[4]]
#define T2 pEq->V[Nodes[5]]
#define T3 pEq->V[Nodes[6]]
#define T4 pEq->V[Nodes[7]]

#define Q1 pEq->V[Branch[4]]
#define Q2 pEq->V[Branch[5]]
#define Q3 pEq->V[Branch[6]]
#define Q4 pEq->V[Branch[7]]

CAbsorber::CAbsorber(void)
{
    SetName("A");
    AddNodes(eLeft,CPoint(0,60),"Вход газовой фазы",L_Calculate);
    AddNodes(eRight,CPoint(100,30),"Вход жидкой фазы",L_Calculate);
    AddNodes(eTop,CPoint(50,0),"Выход газовой фазы",L_Calculate);
    AddNodes(eBottom,CPoint(50,100),"Выход жидкой фазы",L_Calculate);

    ar_0= new CStringArray;
    ar_0->Add("Line(35,25,35,75)");
    ar_0->Add("Line(65,25,65,75)");
    ar_0->Add("Arc(65,35,35,14,65,25,35,25)");
    ar_0->Add("Arc(65,84,35,65,35,75,65,75)");
    ar_0->Add("Line(50,5,50,15)");
    ar_0->Add("Line(50,0,53,10)");
    ar_0->Add("Line(50,0,47,10)");
    ar_0->Add("Line(50,84,50,95)");
    ar_0->Add("Line(53,90,50,100)");
    ar_0->Add("Line(47,90,50,100)");
    ar_0->Add("Line(65,30,100,30)");
    ar_0->Add("Line(65,30,75,27)");
    ar_0->Add("Line(65,30,75,33)");
    ar_0->Add("Line(0,60,35,60)");
    ar_0->Add("Line(35,60,25,57)");
    ar_0->Add("Line(35,60,25,63)");
    AddLayer(L_Calculate,CSize(100,100),ar_0);
}

```

```

S=1.000000e+002;
CountAbsFraction=0;
OldCountAbsFraction=0;
G=0.000000e+000;
//cg=1000;
//cf=1000;

AddParam(_double,"Площадь контакта фаз",&S);
AddParam(_long,"Количество абсорбируемых веществ",&CountAbsFraction);
//AddParam(_double,"Коэффициент массопередачи в газовой фазе",&cg);
//AddParam(_double,"Коэффициент массопередачи в жидковой фазе",&cf);
GC = pG = pF = NULL;
}

CAbsorber::~CAbsorber(void)
{
if (GC)
delete GC;
if (pG)
delete pG;
if (pF)
delete pF;
for (int i = 0, c = mAbsorbFraction.GetCount(); i < c; i++)
{
sAbsorbFraction* pAF = mAbsorbFraction.GetAt(0);
mAbsorbFraction.RemoveAt(0);
delete pAF->pNComboNameF;
delete pAF;
}
}

SComponentInfo* CAbsorber::SetInfoComponent(HINSTANCE pInst)
{
SComponentInfo* pInfo = new SComponentInfo;
pInfo->ID=40024;
pInfo->Name="Абсорбер";
pInfo->PathOfTree="химико-технологическая система";
//pInfo->pIcon=LoadIcon(pInst,MAKEINTRESOURCE());
//pInfo->pIconFolder=LoadIcon(pInst,MAKEINTRESOURCE());
return pInfo;
}
//Запускается в начале эксперимента
void CAbsorber::SetMarsVarsConnection()
{
CountNodes=CountBranch=0;
CountVectorConnections = 4;
InitVectorsConnection();
//VectorConnections = new CVectorConnection[CountVectorConnections];
SetEnergyConnection(0,0,"", "P1", 0, "G1", 1);
SetEnergyConnection(1,0,"", "P2", 1, "G2", 2);
SetEnergyConnection(2,0,"", "P3", 2, "G3", 3);
SetEnergyConnection(3,0,"", "P4", 3, "G4", 4);
}

```

```

SetEnergyConnection(0,1,"", "T1", 4, "Q1", 5);
SetEnergyConnection(1,1,"", "T2", 5, "Q2", 6);
SetEnergyConnection(2,1,"", "T3", 6, "Q3", 7);
SetEnergyConnection(3,1,"", "T4", 7, "Q4", 8);
int num = 8;
for(int i=0,c=GetCountFraction();i<c;i++)
{
    CString s;
    s.Format("%i",i+1);
    SetInformationConnection(0,2, "", "C1_" +s, num++);
    SetInformationConnection(1,2, "", "C2_" +s, num++);
    SetInformationConnection(2,2, "", "C3_" +s, num++);//ВЫХОД
    SetInformationConnection(3,2, "", "C4_" +s, num++);//ВЫХОД
}
InitVariables();
CountEquations=7;
CountMembers=36;
};

```

```

bool CAbsorber::TopologQuery(CSystemEquations* pEq, CMarsTime* MarsTime, CDirect-
Param* DirPar)
{
    SetTopologWires(pEq,-1,0);
    SetTopologWires(pEq,-2,1);
    SetTopologWires(pEq,+3,2);
    SetTopologWires(pEq,+4,3);
    SetTopologWires(pEq,-5,4);
    SetTopologWires(pEq,-6,5);
    SetTopologWires(pEq,+7,6);
    SetTopologWires(pEq,+8,7);
    ///G1=P1=G2=P2=G3=P3=G4=P4=1;
    /*if (GC)
        delete GC;
    if (pG)
        delete pG;
    if (pF)
        delete pF;*/
    int CountFraction = GetCountFraction();
    if (GC==NULL)
        GC = new double[CountFraction];
    if (pG==NULL)
        pG = new double[CountFraction];
    if (pF==NULL)
        pF = new double[CountFraction];
    return true;
}
bool CAbsorber::CalculatQuery2(CSystemEquations* pEq, CMarsTime* MarsTime, CDirect-
Param* DirPar)
{
    int m2_2[4] = { 0, 1, 2, 3 };
    double am3_2[4] = { 1, 1, -1, -1 };
    SetEquation(pEq, 10, 0, 4, m2_2, am3_2);
}

```

```
int m2_4[2] = { 4, 6 };
double am3_4[2] = { 1, -1 };
SetEquation(pEq, 10, 0, 2, m2_4, am3_4);
```

```
int m2_5[2] = { 5, 7 };
double am3_5[2] = { 1, -1 };
SetEquation(pEq, 10, 0, 2, m2_5, am3_5);
```

```
int m2_6[2] = { -5, -7 };
double am3_6[2] = { 1, -1 };
SetEquation(pEq, 10, 0, 2, m2_6, am3_6);
```

```
int m2_7[2] = { -6, -8 };
double am3_7[2] = { 1, -1 };
SetEquation(pEq, 10, 0, 2, m2_7, am3_7);
return true;
}
```

```
bool CAbsorber::CalculatQuery1(CSystemEquations* pEq, CMarsTime* MarsTime, CDirect-
Param* DirPar)
{
double t = MarsTime->T;
G=0;
int CountFraction = GetCountFraction();
for(int i=0;i<CountFraction;i++)
    GC[i]=pG[i]=pF[i]=0;
if(!First)
{
int CountAbs = mAbsorbFraction.GetCount();
for(int i=0;i<CountAbs;i++)
{
sAbsorbFraction* pFract = mAbsorbFraction.GetAt(i);
CString NameF = pFract->pNComboNameF->Items.GetAt(pFract->pNComboNameF-
>Active);
//TRACE(NameF);
int NumberF = GetNumberFraction(NameF);
//Отдает
if (NumberF == -1)
    continue;
double Gx =min(pFract->BetaG*(pEq->V[Nodes[8+4*NumberF]]-pFract->CRG)*S,
                pEq->V[Branch[0]]*pEq->V[Nodes[8+4*NumberF]]);
//Принимаем
double Gy=pFract->BetaL*(pFract->CRL-pEq->V[Nodes[8+4*NumberF+1]])*S;
if (NumberF<0 || NumberF>=CountFraction)
    AfxMessageBox("Memory error");
GC[NumberF]=(Gx<Gy)?Gx:Gy;
G+=GC[NumberF];
}
//нормировка концентраций
bool B=true;
int SS=0;
```



```

double OutG = pEq->V[Branch[0]]-G, InL = pEq->V[Branch[1]]+G;
int CountIter = 0;
while(B)
{
    B=false;
    if(DirPar->StopExperiment)
        return true;
    for(int i=0;i<CountFraction;i++)
    {
        int C0=Nodes[8+4*i], C1=Nodes[8+4*i+1];
        pG[i]=(pEq->V[Branch[0]]*pEq->V[C0]-GC[i])/OutG; //газовая фаза
        pF[i]=(pEq->V[Branch[1]]*pEq->V[C1]+GC[i])/InL; //жидкая фаза
    }
    Normir(pF);
    Normir(pG);
    if (CountIter++>10)
        break;
    for(int i=0;i<CountAbs;i++)
    {
        sAbsorbFraction* pFract = mAbsorbFraction.GetAt(i);
        CString Name = pFract->pNComboNameF->Items.GetAt(pFract-
>pNComboNameF->Active);
        int NumberFraction = GetNumberFraction(Name);
        double DeltaC = pF[NumberFraction]-pFract->CRL;
        static double OldDeltaC = DeltaC;
        if(DeltaC>10e-6)
        {
            double S = (pF[NumberFraction]-pFract->CRL)*(pEq->V[Branch[1]]+G);
            GC[NumberFraction]-=S;
            G-=S;
            B=true;
            if(OldDeltaC==DeltaC)
            {
                //AfxMessageBox("Абсорбция не идет");
                //TRACE("Абсорбция не идет");
                B=false;
                //DirPar->StopExperiment=!DirPar->StopExperiment;
            }
        }
    }
}
}
else
    First=false;
int m2_0[2]={-1,-3};
double am3_0[2]={1,-1};
SetEquation(pEq,11,G,2,m2_0,am3_0);

int m2_1[2]={-2,-4};
double am3_1[2]={-1,1};
SetEquation(pEq,11,G,2,m2_1,am3_1);

```

```

int num=7;
for(int i=0,c=GetCountFraction();i<c;i++)
{
    TRACE(GetNameFraction(i));
    int m2_5[1]={(num+=3)};
    double am3_5[1]={1};
    SetEquation(pEq,11,pG[i],1,m2_5,am3_5);

    int m2_6[1]={++num};
    double am3_6[1]={1};
    SetEquation(pEq,11,pF[i],1,m2_6,am3_6);
}
return true;
};

bool CAbsorber::CalculatQuery(CSystemEquations* pEq, CMarsTime* MarsTime, CDirect-
Param* DirPar)
{
    double t=MarsTime->T;
    /*не считается
    int m2_3[8]={0,-1,1,-2,2,-3,3,-4};
    double am3_3[8]={G1,P1,G2,P2,-G3,-P3,-G4,-P4};
    SetEquation(pEq,13,P1*G1+P2*G2-P3*G3-P4*G4,8,m2_3,am3_3);
    */ int m2_3[4] = { 0, -3, 2, -1 };
    double am3_3[4] = { G3, P1, -G1, -P3 };
    SetEquation(pEq, 13, G3*P1 - G1*P3, 4, m2_3, am3_3);

    return true;
};

void CAbsorber::OnRun()
{
    First=true;
}

void CAbsorber::OnChangeAttribute(Attribute* pAttr)
{
    if(CountAbsFraction>OldCountAbsFraction)
    {
        for(int i=OldCountAbsFraction;i<CountAbsFraction;i++)
        {
            sAbsorbFraction* pAF = new sAbsorbFraction;
            pAF->pNComboNameF = new sCombo;
            SetComboNameSubstants(pAF->pNComboNameF);
            CString str;
            str.Format("Абс. вещество %i",i+1);
            pAF->pAttrComboNameF=AddParam(_sCombo,str,pAF->pNComboNameF);
            pAF->BetaG=1;
            str.Format("Коэффициент массопередачи в отдающей фазе вещества %i",i+1);
            pAF->pAttrBetaG=AddParam(_double,str,&pAF->BetaG);
        }
    }
}

```

```

    pAF->BetaL=1;
    str.Format("Коэффициент массопередачи в принимающей фазе вещества %i",i+1);
    pAF->pAttrBetaL=AddParam(_double,str,&pAF->BetaL);
    pAF->CRG=0;
    str.Format("Концентрация в отдающей фазе вещества %i",i+1);
    pAF->pAttrCRG=AddParam(_double,str,&pAF->CRG);
    pAF->CRL=0;
    str.Format("Концентрация в принимающей фазе вещества %i",i+1);
    pAF->pAttrCRL=AddParam(_double,str,&pAF->CRL);
    mAbsorbFraction.Add(pAF);
}
OldCountAbsFraction=CountAbsFraction;
}
}

void CAbsorber::GetInterface(long ID, void** pInterface)
{
    if(ID==Interface_IRedactComponent)
        *pInterface=(IRedactComponent*)this;
    if(ID==Interface_ISerialize)
        *pInterface=(ISerialize*)this;
    if(ID==Interface_IMarsInerfsace)
        *pInterface=(IMarsInerfsace*)this;
}

```

**Приложение 10. Программно-алгоритмическая реализация компонента
«Абсорбер», адаптированная к методике формирования
системы уравнений с учетом их класса**

1. Описание класса *CAbsorber*

```

class CAbsorber :
    public IRedactComponent, ISerialize, CBaseCTOModel, IFizChimInterfece
{
    //Площадь поверхности контакта фаз
    double S;
    //Количество абсорбируемых веществ
    long CountAbsFraction;
    //Старое количество абсорбируемых веществ, применяется
//для изменения количества атрибутов компонента
    long OldCountAbsFraction;
//Объем вещества, переходящего через границу раздела фаз
    double G;
    //Массив информации об абсорбируемых веществах
    CArray <sAbsorbFraction*, sAbsorbFraction*> mAbsorbFraction;
    //флаг начала эксперимента
    bool First;
    //Массивы для вычисления концентраций веществ в фазах на выходе абсорбера
    double* GC, *pG, *pF;
public:
    //Конструктор класса
    CAbsorber(void);
    //Деструктор класса
    virtual ~CAbsorber(void);
    //Информация о компоненте для формирования дерева компонентов
    static SComponentInfo* SetInfoComponent(HINSTANCE pInst);
    //Запрос интерфейсов компонента
    virtual void GetInterface(long ID, void** pInterface);
    //Метод начала эксперимента
    virtual void OnRun();
    //Метод реакции компонента на изменение значений атрибутов компонента
    virtual void OnChangeAttribute(Attribute* pAttr);
    //Метод сериализации (чтения и записи данных в файл)
    virtual void Serialize(CArchive& ar);
    //Метод формирования локального координатного базиса компонента
    virtual void SetMarsVarsConnection();
    //Метод опроса топологического блока компонента
    virtual bool TopologQuery(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);
    //Метод опроса вычислительного блока компонента для формирования
//линейных уравнений с постоянными коэффициентами
    virtual bool CalculatQuery2(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);
    //Метод опроса вычислительного блока компонента для формирования
//линейных уравнений с переменными во времени коэффициентами

```

```

    virtual bool CalculatQuery1(CSystemEquations* pEq, CMarsTime* MarsTime =
NULL, CDirectParam* DirPar = NULL);
    //Метод опроса вычислительного блока компонента для формирования
//нелинейных и дифференциальных уравнений
    virtual bool CalculatQuery(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);

};

```

2. Реализация методов класса *CAbsorber*

```

#define P1 pEq->V[Nodes[0]]
#define P2 pEq->V[Nodes[1]]
#define P3 pEq->V[Nodes[2]]
#define P4 pEq->V[Nodes[3]]

#define G1 pEq->V[Branch[0]]
#define G2 pEq->V[Branch[1]]
#define G3 pEq->V[Branch[2]]
#define G4 pEq->V[Branch[3]]

#define T1 pEq->V[Nodes[4]]
#define T2 pEq->V[Nodes[5]]
#define T3 pEq->V[Nodes[6]]
#define T4 pEq->V[Nodes[7]]

#define Q1 pEq->V[Branch[4]]
#define Q2 pEq->V[Branch[5]]
#define Q3 pEq->V[Branch[6]]
#define Q4 pEq->V[Branch[7]]

```

```

CAbsorber::CAbsorber(void)

```

```

{
    SetName("A");
    AddNodes(eRight,CPoint(100,30),"Вход газовой фазы",L_Calculate);
    AddNodes(eLeft,CPoint(0,60),"Вход жидкой фазы",L_Calculate);
    AddNodes(eTop,CPoint(50,5),"Выход газовой фазы",L_Calculate);
    AddNodes(eBottom,CPoint(50,95),"Выход жидкой фазы",L_Calculate);
    ar_0= new CStringArray;
    ar_0->Add("Line(35,25,35,75)");
    ar_0->Add("Line(65,25,65,75)");
    ar_0->Add("Arc(65,35,35,14,65,25,35,25)");
    ar_0->Add("Arc(65,84,35,65,35,75,65,75)");
    ar_0->Add("Line(50,5,50,15)");
    ar_0->Add("Line(50,0,53,10)");
    ar_0->Add("Line(50,0,47,10)");
    ar_0->Add("Line(50,84,50,95)");
    ar_0->Add("Line(53,90,50,100)");
    ar_0->Add("Line(47,90,50,100)");
    ar_0->Add("Line(65,30,100,30)");
    ar_0->Add("Line(65,30,75,27)");
    ar_0->Add("Line(65,30,75,33)");
}

```

```

ar_0->Add("Line(0,60,35,60)");
ar_0->Add("Line(35,60,25,57)");
ar_0->Add("Line(35,60,25,63)");
AddLayer(L_Calculate,CSize(100,100),ar_0);
S=1.000000e+002;
CountAbsFraction=0;
OldCountAbsFraction=0;
G=0.000000e+000;
AddParam(_double,"Площадь контакта фаз",&S);
AddParam(_long,"Количество абсорбируемых веществ",&CountAbsFraction);
GC = pG = pF = NULL;
}

```

```

CAbsorber::~CAbsorber(void)
{
if (GC)
delete GC;
if (pG)
delete pG;
if (pF)
delete pF;
for (int i = 0, c = mAbsorbFraction.GetCount(); i < c; i++)
{
sAbsorbFraction* pAF = mAbsorbFraction.GetAt(0);
mAbsorbFraction.RemoveAt(0);
delete pAF->pNComboNameF;
delete pAF;
}
}

```

```

SComponentInfo* CAbsorber::SetInfoComponent(HINSTANCE pInst)
{
SComponentInfo* pInfo = new SComponentInfo;
pInfo->ID=40024;
pInfo->Name="Абсорбер";
pInfo->PathOfTree="Физико-химические системы";
return pInfo;
}

```

```

void CAbsorber::SetMarsVarsConnection()
{
CountNodes=CountBranch=0;
CountVectorConnections = 4;
InitVectorsConnection();
SetEnergyConnection(0,0,"", "P1", 0, "G1", 1);
SetEnergyConnection(1,0,"", "P2", 1, "G2", 2);
SetEnergyConnection(2,0,"", "P3", 2, "G3", 3);
SetEnergyConnection(3,0,"", "P4", 3, "G4", 4);
SetEnergyConnection(0,1,"", "T1", 4, "Q1", 5);
SetEnergyConnection(1,1,"", "T2", 5, "Q2", 6);
SetEnergyConnection(2,1,"", "T3", 6, "Q3", 7);
SetEnergyConnection(3,1,"", "T4", 7, "Q4", 8);
}

```

```

int num = 8;
for(int i=0,c=GetCountFraction();i<c;i++)
{
    CString s;
    s.Format("%i",i+1);
    SetInformationConnection(0,2, "", "C1_"+s, num++);
    SetInformationConnection(1,2, "", "C2_"+s, num++);
    SetInformationConnection(2,2, "", "C3_"+s, num++);
    SetInformationConnection(3,2, "", "C4_"+s, num++);
}
InitVariables();
CountEquations=7;
CountMembers=36;
};

```

```

bool CAbsorber::TopologQuery(CSystemEquations* pEq, CMarsTime* MarsTime, CDirect-
Param* DirPar)

```

```

{
    SetTopologWires(pEq,-1,0);
    SetTopologWires(pEq,-2,1);
    SetTopologWires(pEq,+3,2);
    SetTopologWires(pEq,+4,3);
    SetTopologWires(pEq,-5,4);
    SetTopologWires(pEq,-6,5);
    SetTopologWires(pEq,+7,6);
    SetTopologWires(pEq,+8,7);
    G1=P1=G2=P2=G3=P3=G4=P4=1;
    int CountFraction = GetCountFraction();
    if (GC==NULL)
        GC = new double[CountFraction];
    if (pG==NULL)
        pG = new double[CountFraction];
    if (pF==NULL)
        pF = new double[CountFraction];
    return true;
}

```

```

bool CAbsorber::CalculatQuery2(CSystemEquations* pEq, CMarsTime* MarsTime, CDirect-
Param* DirPar)

```

```

{
    int m2_2[4]={0,1,2,3};
    double am3_2[4]={1,1,-1,-1};
    SetEquation(pEq,11,0,4,m2_2,am3_2);

    int m2_4[2] = { 4, 6 };
    double am3_4[2] = { 1, -1 };
    SetEquation(pEq, 10, 0, 2, m2_4, am3_4);

    int m2_5[2] = { 5, 7 };
    double am3_5[2] = { 1, -1 };
    SetEquation(pEq, 10, 0, 2, m2_5, am3_5);
}

```

```

int m2_6[2] = { -5, -7 };
double am3_6[2] = { 1, -1 };
SetEquation(pEq, 10, 0, 2, m2_6, am3_6);

int m2_7[2] = { -6, -8 };
double am3_7[2] = { 1, -1 };
SetEquation(pEq, 10, 0, 2, m2_7, am3_7);
}

bool CAbsorber::CalculatQuery1(CSystemEquations* pEq, CMarsTime* MarsTime, CDirect-
Param* DirPar)
{
double t = MarsTime->T;
G=0;
int CountFraction = GetCountFraction();
for(int i=0;i<CountFraction;i++)
    GC[i]=pG[i]=pF[i]=0;
if(!First)
{
int CountAbs = mAbsorbFraction.GetCount();
for(int i=0;i<CountAbs;i++)
{
sAbsorbFraction* pFract = mAbsorbFraction.GetAt(i);
CString NameF = pFract->pNComboNameF->Items.GetAt(pFract->pNComboNameF
->Active);
int NumberF = GetNumberFraction(NameF);
double Gx =min(pFract->BetaG*(pEq->V[Nodes[8+4*NumberF]]-pFract->CRG)*S,
pEq->V[Branch[0]]*pEq->V[Nodes[8+4*NumberF]]);
double Gy=pFract->BetaL*(pFract->CRL-pEq->V[Nodes[8+4*NumberF+1]])*S;
GC[NumberF]=(Gx<Gy)?Gx:Gy;
G+=GC[NumberF];
}
//нормировка концентраций
bool B=true;
int SS=0;
double OutG = pEq->V[Branch[0]]-G, InL = pEq->V[Branch[1]]+G;
while(B)
{
B=false;
if(DirPar->StopExperiment)
return true;
for(int i=0;i<CountFraction;i++)
{
int C0=Nodes[8+4*i], C1=Nodes[8+4*i+1];
pG[i]=(pEq->V[Branch[0]]*pEq->V[C0]-GC[i])/OutG;
pF[i]=(pEq->V[Branch[1]]*pEq->V[C1]+GC[i])/InL;
}
Normir(pF);
Normir(pG);
for(int i=0;i<CountAbs;i++)
{
sAbsorbFraction* pFract = mAbsorbFraction.GetAt(i);

```



```

CString Name = pFract->pNComboNameF->Items.GetAt(
    pFract->pNComboNameF->Active);
int NumberFraction = GetNumberFraction(Name);
double DeltaC = pF[NumberFraction]-pFract->CRL;
static double OldDeltaC = DeltaC;
if(DeltaC>10e-6)
{
    double S = (pF[NumberFraction]-pFract->CRL)*(pEq->V[Branch[1]]+G);
    GC[NumberFraction]-=S;
    G-=S;
    B=true;
    if(OldDeltaC==DeltaC)
    {
        B=false;
    }
}
}
}
else
    First=false;

int m2_0[2]={-1,-3};
double am3_0[2]={1,-1};
SetEquation(pEq,11,G,2,m2_0,am3_0);

int m2_1[2]={-2,-4};
double am3_1[2]={-1,1};
SetEquation(pEq,11,G,2,m2_1,am3_1);

int num=7;
for(int i=0,c=GetCountFraction();i<c;i++)
{
    int m2_5[1]={(num+=3)};
    double am3_5[1]={1};
    SetEquation(pEq,11,pG[i],1,m2_5,am3_5);

    int m2_6[1]={++num};
    double am3_6[1]={1};
    SetEquation(pEq,11,pF[i],1,m2_6,am3_6);
}
return true;
};

```

```

bool CAbsorber::CalculatQuery(CSystemEquations* pEq, CMarsTime* MarsTime, CDirect-
Param* DirPar)
{
    double t=MarsTime->T;
    int m2_3[4] = { 0, -3, 2, -1 };
    double am3_3[4] = { G3, P1, -G1, -P3 };
    SetEquation(pEq, 13, G3*P1 - G1*P3, 4, m2_3, am3_3);
    return true;
}

```

```

};

void CAbsorber::OnRun()
{
    First=true;
}

void CAbsorber::OnChangeAttribute(Attribute* pAttr)
{
    if(CountAbsFraction>OldCountAbsFraction)
    {
        for(int i=OldCountAbsFraction;i<CountAbsFraction;i++)
        {
            sAbsorbFraction* pAF = new sAbsorbFraction;
            pAF->pNComboNameF = new sCombo;
            SetComboNameSubstants(pAF->pNComboNameF);
            CString str;
            str.Format("Абс. вещество %i",i+1);
            pAF->pAttrComboNameF=AddParam(_sCombo,str,pAF->pNComboNameF);
            pAF->BetaG=1;
            str.Format("Коэффициент массопередачи в отдающей фазе вещества %i",i+1);
            pAF->pAttrBetaG=AddParam(_double,str,&pAF->BetaG);
            pAF->BetaL=1;
            str.Format("Коэффициент массопередачи в принимающей фазе вещества %i",i+1);
            pAF->pAttrBetaL=AddParam(_double,str,&pAF->BetaL);
            pAF->CRG=0;
            str.Format("Концентрация в отдающей фазе вещества %i",i+1);
            pAF->pAttrCRG=AddParam(_double,str,&pAF->CRG);
            pAF->CRL=0;
            str.Format("Концентрация в принимающей фазе вещества %i",i+1);
            pAF->pAttrCRL=AddParam(_double,str,&pAF->CRL);
            mAbsorbFraction.Add(pAF);
        }
        OldCountAbsFraction=CountAbsFraction;
    }
}

void CAbsorber::Serialize(CArchive& ar)
{
}

void CAbsorber::GetInterface(long ID, void** pInterface)
{
    if(ID==Interface_IRedactComponent)
        *pInterface=(IRedactComponent*)this;
    if(ID==Interface_ISerialize)
        *pInterface=(ISerialize*)this;
    if(ID==Interface_IMarsInerfsace)
        *pInterface=(IMarsInerfsace*)this;
}

```

Приложение 11. Программная реализация измерителей потенциальных и потоковых переменных

1. Описание класса измерителя потенциальной переменной *CPotenzMeter*

```

class CPotenzMeter :
    public IRedactComponent,IMessengerComponent, IBaseCTOModel
{
    double Vn;
    long CountEnergy;
    long CountInformation;
    bool bEnergy;
    long NumberVar;

public:
    CPotenzMeter(void);
    virtual ~CPotenzMeter(void);
    static SComponentInfo* SetInfoComponent(HINSTANCE pInst);
    virtual void GetInterface(long ID, void** pInterface);
    virtual void SendMessage(LPVOID Data, int pPin, eTypes Type);
    virtual void SetMarsVarsConnection();
    virtual bool CalculatQuery1(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);
    virtual bool CalculatQuery(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);
    virtual bool IzmeritQuery(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);
};

```

2. Реализация методов класса *CPotenzMeter*

```

CPotenzMeter::CPotenzMeter(void)
{
    SetName("I_Vn");
    AddNodes(eBottom,CPoint(50,100),"Вход",L_Calculate);
    AddNodes(eRight,CPoint(100,50),"Выход",L_Logic);
    ar_1= new CStringArray;
    ar_1->Add("Rect(10,10,90,90)");
    ar_1->Add("Text(20,20,80,80,Vn)");
    AddLayer(L_Logic,CSize(40,40),ar_1);
    ar_2= new CStringArray;
    ar_2->Add("Ellipse(10,10,90,90)");
    ar_2->Add("Line(50,90,50,99)");
    ar_2->Add("");
    ar_2->Add("Text(20,21,80,79,Vn)");
    AddLayer(L_Calculate,CSize(40,40),ar_2);
    Vn=0.000000e+000;
    CountEnergy=1;
    CountInformation=0;
    bEnergy=true;
    NumberVar=1;
    AddParam(_long,"Кол-во элементарных связей",&CountEnergy);
    AddParam(_long,"Кол-во информационных связей",&CountInformation);
}

```

```

AddParam(_bool, "Измеряем энергетическую",&bEnergy);
AddParam(_long, "Номер изменяемой переменной",&NumberVar);
}

CPotenzMeter::~CPotenzMeter(void)
{

}

SComponentInfo* CPotenzMeter::SetInfoComponent(HINSTANCE pInst)
{
    SComponentInfo* pInfo = new SComponentInfo;
    pInfo->ID=40402;
    pInfo->Name="Измеритель Vn";
    pInfo->PathOfTree="Обобщенные измерители";
    return pInfo;
}

void CPotenzMeter::SetMarsVarsConnection()
{
    CountVectorConnections = 1;
    InitVectorsConnection();
    int CountW=0;
    for(int j=0; j<CountEnergy;j++)
    {
        CString s1, s2;
        s1.Format("VEn_%i",j+1);
        s2.Format("VEb_%i",j+1);
        SetEnergyConnection(0,CountW++, "", s1,j,s2,j);
    }
    for(int i=0; i<CountInformation;i++)
    {
        CString str;
        str.Format("Vi_%i",i+1);
        SetInformationConnection(0,CountW++, "", str, CountEnergy+i);
    }
    InitVariables();
    for(int i=0;i<CountBranch;i++)
        Branch[i]=-2;
    CountEquations=1;
    CountMembers=1;
};

bool CPotenzMeter::CalculatQuery1(CSystemEquations* pEq, CMarsTime* MarsTime,
CDirectParam* DirPar)
{
    return true;
};

bool CPotenzMeter::CalculatQuery(CSystemEquations* pEq, CMarsTime* MarsTime,
CDirectParam* DirPar)
{

```

```

double t=MarsTime->T;
return true;
}

```

```

bool CPotenzMeter::IzmeritQuery(CSystemEquations* pEq, CMarsTime* MarsTime,
CDirectParam* DirPar)
{
if(NumberVar==0)
{
return true;
}
if (pEq->&pEq->V)
{
int NumLKB = (bEnergy) ? NumberVar - 1 : CountEnergy + NumberVar - 1;
ASSERT(NumLKB < CountNodes);
double Val = pEq->V[Nodes[NumLKB]];
SendMessage(&Val, 1, _double);
}
return true;
}

```

```

void CPotenzMeter::SendMessage(LPVOID Data, int pWires, eTypes Type)
{
CMessage* mes = new CMessage;
mes->Data=Data;
mes->nWires=pWires;
mes->type=Type;
mes->pComponent=(IComponent*)pMComp;
OutMessage(mes);
delete mes;
}

```

```

void CPotenzMeter::GetInterface(long ID, void** pInterface)
{
if(ID==Interface_IRedactComponent)
*pInterface=(IRedactComponent*)this;
if(ID==Interface_IMessengerComponent)
*pInterface=(IMessengerComponent*)this;
if(ID==Interface_IMarsInerfsace)
*pInterface=(IMarsInerfsace*)this;
}

```

3. Описание класса измерителя потоковой переменной *CPokotMeter*

```

class CPotokMeter :
public IRedactComponent,IMessengerComponent, IBaseCTOModel
{
double Vb;
long CountEnergy;
long CountInformaion;
long NumberVar;
public:
CPotokMeter(void);

```

```

virtual ~CPotokMeter(void);
static SComponentInfo* SetInfoComponent(HINSTANCE pInst);
virtual void GetInterface(long ID, void** pInterface);
virtual void SendMessage(LPVOID Data, int pPin, eTypes Type);
virtual void SetMarsVarsConnection();
virtual bool TopologQuery(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);
virtual bool CalculatQuery1(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);
virtual bool CalculatQuery(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);
virtual bool IzmeritQuery(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar= NULL);
};

```

4. Реализация методов класса *CPotokMeter*

```

CPotokMeter::CPotokMeter(void)
{
SetName("I_Vb");
AddNodes(eLeft, CPoint(0,50), "Вход", L_Calculate);
AddNodes(eRight, CPoint(100,50), "Выход", L_Calculate);
AddNodes(eRight, CPoint(100,50), "Измеритель", L_Logic);
ar_1= new CStringArray;
ar_1->Add("Rect(10,10,90,90)");
ar_1->Add("Text(20,20,80,80,Vb)");
ar_1->Add("Line(90,50,100,50)");
AddLayer(L_Logic, CSize(40,40), ar_1);
ar_2= new CStringArray;
ar_2->Add("Ellipse(10,10,90,90)");
ar_2->Add("Line(0,50,10,50)");
ar_2->Add("Line(90,50,100,50)");
ar_2->Add("Text(20,20,80,80,Vb)");
AddLayer(L_Calculate, CSize(40,40), ar_2);
Vb=0;
CountEnergy=1;
CountInformaion=0;
NumberVar=1;
AddParam(_long, "Кол-во элементарных связей", &CountEnergy);
AddParam(_long, "Кол-во информационных связей", &CountInformaion);
AddParam(_long, "Номер изменяемой переменной", &NumberVar);
}

CPotokMeter::~~CPotokMeter(void)
{

}

SComponentInfo* CPotokMeter::SetInfoComponent(HINSTANCE pInst)
{
SComponentInfo* pInfo = new SComponentInfo;

```

```

pInfo->ID=40403;
pInfo->Name="Измеритель Vb";
pInfo->PathOfTree="Обобщенные измерители";
return pInfo;
}

```

```

void CPotokMeter::SetMarsVarsConnection()
{
    CountNodes=CountBranch=0;
    CountVectorConnections = 2;
    InitVectorsConnection();
    int nVnVar=0;
    for(int i=0;i<CountEnergy;i++)
    {
        CString s1,s2,s3;
        s1.Format("Ve_In%i",i+1);
        s2.Format("Ve_Out%i",i+1);
        s3.Format("Ve_b%i",i+1);
        SetEnergyConnection(0,i,"",s1, 2*i, s3, i+1);
        SetEnergyConnection(1,i,"",s2, 2*i+1, s3, i+1);
    }
    for(int j=0;j<CountInformaion;j++)
    {
        CString s1,s2,s3;
        s1.Format("Vi_In%i",CountEnergy+2*j);
        s2.Format("Vi_Out%i",CountEnergy+2*j+1);
        SetInformationConnection(0,CountEnergy+j,"",s1,CountEnergy*2+2*j);
        SetInformationConnection(1,CountEnergy+j,"",s2,CountEnergy*2+2*j+1);
    }
    InitVariables();
    CountEquations=CountEnergy+CountInformaion;
    CountMembers=2*(CountEnergy+CountInformaion);
}

```

```

bool CPotokMeter::TopologQuery(CSystemEquations* pEq, CMarsTime* MarsTime,
CDirectParam* DirPar)
{
    for(int i=0;i<CountEnergy;i++)
    {
        SetTopologWires(pEq,-(i+1),2*i);
        SetTopologWires(pEq,+(i+1),2*i+1);
    }
    return true;
}

```

```

bool CPotokMeter::CalculatQuery1(CSystemEquations* pEq, CMarsTime* MarsTime,
CDirectParam* DirPar)
{
    double t = MarsTime->T;
    for(int i=0;i<CountEnergy;i++)
    {

```

```

        int m2_0[2]={2*i,2*i+1};
        double am3_0[2]={1,-1};
        SetEquation(pEq,10,0,2,m2_0,am3_0);
    }
    for(int i=0;i<CountInformaion;i++)
    {
        int m2_1[2]={CountEnergy*2+2*i,CountEnergy*2+2*i+1};
        double am3_1[2]={1,-1};
        SetEquation(pEq,10,0,2,m2_1,am3_1);
    }
    return true;
}
bool CPotokMeter::CalculatQuery(CSystemEquations* pEq, CMarsTime* MarsTime,
CDirectParam* DirPar)
{
    double t=MarsTime->T;
    return true;
}
bool CPotokMeter::IzmeritQuery(CSystemEquations* pEq, CMarsTime* MarsTime , CDirect-
Param* DirPar)
{
    if(NumberVar==0||NumberVar>CountBranch)
    {
        return true;
    }
    double Val = pEq->V[Branch[NumberVar-1]];
    SendMessage(&Val,2,_double);
    return true;
}
void CPotokMeter::SendMessage(LPVOID Data, int pWires, eTypes Type)
{
    CMessage* mes = new CMessage;
    mes->Data=Data;
    mes->nWires=pWires;
    mes->type=Type;
    mes->pComponent=(IComponent*)pMComp;
    OutMessage(mes);
    delete mes;
}
void CPotokMeter::GetInterface(long ID, void** pInterface)
{
    if(ID==Interface_IRedactComponent)
        *pInterface=(IRedactComponent*)this;
    if(ID==Interface_IMessengerComponent)
        *pInterface=(IMessengerComponent*)this;
    if(ID==Interface_IMarsInerfsace)
        *pInterface=(IMarsInerfsace*)this;
}

```


Приложение 12. Программная реализация компонента-преобразования данных

1. Описание класса оператора сложения двух комплексных чисел *CSumma*

```
class CSumma :
public IRedactComponent,IMessengerComponent
{
    complex S1;
    complex S2;
    complex R;
    bool F1, F2;
public:
    CSumma(void);
    virtual ~CSumma(void);
    static SComponentInfo* SetInfoComponent(HINSTANCE pInst);
    virtual void GetInterface(long ID, void** pInterface);
    virtual void AcceptMessage(CMessage* pMessage);
    virtual void SendMessage(LPVOID Data, int pPin, eTypes Type);
    virtual void OnRun();
};
```

2. Реализация методов класса *CSumma*

```
CSumma::CSumma(void)
{
    SetName("SUM");
    AddNodes(eLeft,CPoint(0,25),"Слагаемое 1",L_Logic);
    AddNodes(eLeft,CPoint(0,75),"Слагаемое 2",L_Logic);
    AddNodes(eRight,CPoint(100,50),"Сумма",L_Logic);
    ar_1= new CStringArray;
    ar_1->Add("Line(10,10,10,90)");
    ar_1->Add("Line(90,50,10,90)");
    ar_1->Add("Line(90,50,10,10)");
    ar_1->Add("Line(90,50,100,50)");
    ar_1->Add("Line(0,25,10,25)");
    ar_1->Add("Line(0,75,10,75)");
    ar_1->Add("Text(20,40,50,60,+)");
    AddLayer(L_Logic,CSize(40,40),ar_1);
    S1=complex(0,0);
    S2=complex(0,0);
    R=complex(0,0);
    F1=F2=false;
}

CSumma::~CSumma(void)
{
}

SComponentInfo* CSumma::SetInfoComponent(HINSTANCE pInst)
{
    SComponentInfo* pInfo = new SComponentInfo;
```

```

pInfo->ID=30104;
pInfo->Name="Сумма";
pInfo->PathOfTree="Математические компоненты";
pInfo->pIcon=LoadIcon(pInst,MAKEINTRESOURCE(IDI_MC_SUMMA));
pInfo->pIconFolder=LoadIcon(pInst,MAKEINTRESOURCE(IDI_MC_FOLDER));
return pInfo;
}

```

```

void CSumma::AcceptMessage(CMessage* pMessage)
{
switch(pMessage->nWires)
{
case 0: {
F1=true;
if (pMessage->type == _double)
{
S1 = complex(*(double*)pMessage->Data, 0);
R = complex(S1.x + S2.x, S2.y);
if (R.y == 0)
SendMessage(&R.x,2,_double);
else
SendMessage(&R,2,_complex);
}
if (pMessage->type == _complex)
{
S1 = *(complex*)pMessage->Data;
R = complex(S1.x + S2.x, S1.y + S2.y);
if (R.y == 0)
SendMessage(&R.x,2,_double);
else
SendMessage(&R,2,_complex);
}
if(pMessage->type==_long)
{
S1=complex(*(long*)pMessage->Data,0);
R=complex(S1.x + S2.x, S1.y + S2.y);
if (R.y == 0)
SendMessage(&R.x,2,_double);
else
SendMessage(&R,2,_complex);
}
};
break;
case 1: {
F2=true;
if (pMessage->type == _double)
{
S2 = complex(*(double*)pMessage->Data, 0);
R = complex(S1.x + S2.x, S1.y);
if (R.y == 0)
SendMessage(&R.x,2,_double);
else

```

```

        SendMessage(&R,2,_complex);
    }
    if (pMessage->type == _complex)
    {
        S2 = *(complex*)pMessage->Data;
        R = complex(S1.x + S2.x, S1.y + S2.y);
        if (R.y == 0)
            SendMessage(&R.x,2,_double);
        else
            SendMessage(&R,2,_complex);
    }
    if(pMessage->type==_long)
    {
        S2=complex(*(long*)pMessage->Data,0);
        R=complex(S1.x + S2.x, S1.y + S2.y);
        if (R.y == 0)
            SendMessage(&R.x,2,_double);
        else
            SendMessage(&R,2,_complex);
    }
};
break;
}
}

void CSumma::SendMessage(LPVOID Data, int pPin, eTypes Type)
{
    if(F1&&F2)
    {
        CMessage* mes = new CMessage;
        mes->Data=Data;
        mes->nWires=pPin;
        mes->type=Type;
        mes->pComponent=(IComponent*)pMComp;
        OutMessage(mes);
        delete mes;
    }
}

void CSumma::GetInterface(long ID, void** pInterface)
{
    if(ID==Interface_IRedactComponent)
        *pInterface=(IRedactComponent*)this;
    if(ID==Interface_IMessengerComponent)
        *pInterface=(IMessengerComponent*)this;
}

void CSumma::OnRun()
{
    S1=S2=complex(0,0);
    F1=F2=false;
}

```

Приложение 13. Программная реализация компонента с методами обработки действий пользователя

1. Описание класса компонента обработки действий мыши

```

class CColorSource :
    public IRedactComponent, IMouseInterface, IMessengerComponent, IBaseImitationComponent
{
    COLORREF col;
    COLORREF col2;
    COLORREF col3;

public:
    CColorSource(void);
    virtual ~CColorSource(void);
    static SComponentInfo* SetInfoComponent(HINSTANCE pInst);
    virtual void GetInterface(long ID, void** pInterface);
    virtual void OnRun();
    virtual void OnMouseMove(CPoint point);
    virtual void OnLButtonDown(CPoint point);
    virtual void OnRButtonDown(CPoint point);
    virtual void SendMessage(LPVOID Data, int pPin, eTypes Type);
    virtual void OnRunExperiment();
    virtual void StepTime(double T);
};

```

2. Реализация методов класса компонента обработки действий мыши

```

CColorSource::CColorSource(void)
{
    AddNodes(eRight,CPoint(100,50),"Выход",L_Logic);
    ar_1= new CStringArray;
    ar_1->Add("Rect(10,10,90,90)");
    ar_1->Add("Line(90,50,100,50)");
    AddLayer(L_Logic,CSize(50,50),ar_1);
    col = RGB(255,0,0);
    col2 = RGB(0,0,255);
    col3 = RGB(255,215,0);
    AddParam(_COLORREF,"Передаваемый в начале цвет",&col);
    AddParam(_COLORREF,"Передаваемый по кнопке цвет",&col2);
}
CColorSource::~CColorSource(void)
{
}
SComponentInfo* CColorSource::SetInfoComponent(HINSTANCE pInst)
{
    SComponentInfo* pInfo = new SComponentInfo;
    pInfo->ID=30009;
    pInfo->Name="Источник цвета";
}

```

```

    pInfo->PathOfTree="тестирование";
    return pInfo;
}
void CColorSource::OnRun()
{
    col = RGB(60,179,113);
}
void CColorSource::SendMessage(LPVOID Data, int pPin, eTypes Type)
{
    CMessage* mes = new CMessage;
    mes->Data=Data;
    mes->nWires=pPin;
    mes->type=Type;
    mes->pComponent=(IComponent*)pMComp;
    OutMessage(mes);
    delete mes;
}
void CColorSource::OnRunExperiment()
{
    SendMessage(&col,0,_COLORREF);
}
void CColorSource::StepTime(double T)
{
    SendMessage(&col,0,_COLORREF);
}
void CColorSource::OnMouseMove(CPoint point)
{
    SendMessage(&col3,0,_COLORREF);
}
void CColorSource::OnLButtonDown(CPoint point)
{
    SendMessage(&col2,0,_COLORREF);
}
void CColorSource::OnRButtonDown(CPoint point)
{
    SendMessage(&col,0,_COLORREF);
}
void CColorSource::GetInterface(long ID, void** pInterface)
{
    if(ID==Interface_IRedactComponent)
        *pInterface=(IRedactComponent*)this;
    if(ID==Interface_IMouseInterface)
        *pInterface=(IMouseInterface*)this;
    if(ID==Interface_IMessengerComponent)
        *pInterface=(IMessengerComponent*)this;
    if(ID==Interface_IBaseImitationComponent)
        *pInterface=(IBaseImitationComponent*)this;
}

```

Приложение 14. Шаблоны файлов и методов для генератора моделей компонентов

1. Шаблон заголовочного файла *.h

```
#pragma once
#include "..\InterfaceComponent\InterfaceComponent.h"
#include "..\InterfaceComponent\struct.h"
// #include "..\marscomponent\marscomponentclass.h"

class <NameClass> :
public <NameBaseClass>
{
    <VARIABLES_H>
public:
    <NameClass>(void);
    virtual ~<NameClass>(void);
    static SComponentInfo* SetInfoComponent(HINSTANCE pInst);
    virtual void GetInterface(long ID, void** pInterface);
    <FUNCTION_H>
};
```

2. Шаблон исполняемого файла *.cpp

```
#include "StdAfx.h"
#include "<FILE_NAME_H>"
#include "resource.h"

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

<NameClass>::~<NameClass>(void)
{
    <LABEL>
    <NODES>
    <IMAGE_LAYER>
    <VARIABLES_CPP>
    <PARAMETER_CPP>
}

<NameClass>::~~<NameClass>(void)
{
    <DELETE_POINTS>
}

SComponentInfo* <NameClass>::SetInfoComponent(HINSTANCE pInst)
{
    SComponentInfo* pInfo = new SComponentInfo;
    pInfo->ID=<ID_COMPONENT>;
    pInfo->Name=<NameComponent>;
```

```

pInfo->PathOfTree=<NamePath>;
//pInfo->pIcon=LoadIcon(pInst,MAKEINTRESOURCE(<IconComponent>));
//pInfo->pIconFolder=LoadIcon(pInst,MAKEINTRESOURCE(<IconFolder>));
return pInfo;
}
<SET_CONNECTIONS>
<TOPOLOG_LAYER_CPP>
<CALCULAT2_LAYER_CPP>
<CALCULAT1_LAYER_CPP>
<CALCULAT_LAYER_CPP>
<BEGIN_VALUE_CPP>
<ACCEPT_MESSAGE_CPP>
<SEND_MESSAGE_CPP>
<SEND_MESSAGE_BEGIN>
<SEND_MESSAGE_ITERATION>
<FUNCTION_CPP>

```

```

void <NameClass>::GetInterface(long ID, void** pInterface)
{
<ALL_INTERFACE>
}

```

3. Шаблон методов интерфейсных классов

<f1>IRedactComponent::OnRun-Действие компонента при запуске моделирования

<h>

```
virtual void OnRun();
```

</h>

<cpp>

```
void <NameClass>::OnRun()
```

```
{
```

```
  ^<VarName>=<Value>;^
```

```
  //Впишите свой текст
```

```
}
```

</cpp>

</f>

<f>IRedactComponent::OnStop-Действие компонента при остановке моделирования

<h>

```
virtual void OnStop();
```

</h>

<cpp>

```
void <NameClass>::OnStop()
```

```
{
```

```
  //Впишите свой текст
```

```
}
```

</cpp>

</f>

<f>IRedactComponent::OnChangeAttribute-Реакция компонента при изменении значения атрибута

<h>

```
virtual void OnChangeAttribute(Attribute* pAttr);
```

</h>

```

<cpp>
void <NameClass>::OnChangeAttribute(Attribute* pAttr)
{
    //Впишите свой текст
}
</cpp>
</f>
<f>ISerialize::Serialize-Функция файловых записи и чтения
<h>
virtual void Serialize(CArchive& ar);
</h>
<cpp>
void <NameClass>::Serialize(CArchive& ar)
{
    //Впишите свой текст
}
</cpp>
</f>
<f>INotVectorImage::Paint-Функция прорисовки компонента (используется при динами-
ческом изображении компонента)
<h>
virtual void Paint(CDC& dc, long Layer);
</h>
<cpp>
void <NameClass>::Paint(CDC& dc, long Layer)
{
    //Впишите свой текст
}
</cpp>
</f>
<f>INotVectorImage::GetSize-Функция запроса геометрических размеров компонента на
слое
<h>
virtual CSize GetSize(long Layer);
</h>
<cpp>
CSize <NameClass>::GetSize(long Layer)
{
    CSize size;
    //Впишите свой текст
    return size;
}
</cpp>
</f>
<f1>CComponent::GetInterface - Обработчик интерфейсов
<h>
virtual void GetInterface(long ID, void **&pInterface);
</h>
<cpp>
void <NameClass>::GetInterface(long ID, void** &pInterface)
{

```



```

}
</cpp>
</f>

```

<f>IMouseInterface::OnMouseMove-Обработчик подвода мыши к компоненту

<h>

```
virtual void OnMouseMove(CPoint point);
```

</h>

```
<cpp>
```

```
void <NameClass>::OnMouseMove(CPoint point)
```

```
{
```

```
    //Впишите свой текст
```

```
}
```

```
</cpp>
```

```
</f>
```

<f>IMouseInterface::OnLButtonDown-Обработчик нажатия левой кнопки мыши

<h>

```
virtual void OnLButtonDown(CPoint point);
```

</h>

```
<cpp>
```

```
void <NameClass>::OnLButtonDown(CPoint point)
```

```
{
```

```
    //Впишите свой текст
```

```
}
```

```
</cpp>
```

```
</f>
```

<f>IMouseInterface::OnDbcClk(void)-Обработчик двойного нажатия левой кнопки мыши

без эксперимента

<h>

```
virtual void OnDbcClk(void);
```

</h>

```
<cpp>
```

```
void <NameClass>::OnDbcClk(void)
```

```
{
```

```
    //Впишите свой текст
```

```
}
```

```
</cpp>
```

```
</f>
```

<f>IMouseInterface::OnLButtonDbcClk-Обработчик двойного нажатия левой кнопки мы-

ши при эксперименте

<h>

```
virtual void OnLButtonDbcClk();
```

</h>

```
<cpp>
```

```
void <NameClass>::OnLButtonDbcClk()
```

```
{
```

```
    //Впишите свой текст
```

```
}
```

```
</cpp>
```

```
</f>
```

<f>IRedactComponent::OnRButtonDown-Обработчик нажатия правой кнопки мыши

<h>

```
virtual void OnRButtonDown(CPoint point);
```

```

</h>
<cpp>
void <NameClass>::OnRButtonDown(CPoint point)
{
    //Впишите свой текст
}
</cpp>
</f>
<f>IRedactComponent::OnSize-Обработчик изменения размера компонента
</f>
<f>IVarWiresInterface::ConnectNode-Обработчик присоединения связи к узлу
<h>
virtual SNodes* ConnectNode(int Number);
</h>
<cpp>
SNodes* <NameClass>::ConnectNode(int Number)
{
    SNodes* pNodes = new SNodes;
    //Впишите свой текст
    return pNodes;
}
</cpp>
</f>
<f>IVarWiresInterface::UnConnectNode-Обработчки отсоединения связи от узла
<h>
virtual SNodes* UnConnectNode(int Number);
</h>
<cpp>
SNodes* <NameClass>::UnConnectNode(int Number)
{
    SNodes* pNodes = new SNodes;
    //Впишите свой текст
    return pNodes;
}
</cpp>
</f>
<f>IVarWiresInterface::GetNewSize-Установка размера компонента после добавления уз-
ла
<h>
virtual CSize GetNewSize(long Layer);
</h>
<cpp>
CSize <NameClass>::GetNewSize(long Layer)
{
    CSize NewSize;
    //Заполните позиции структуры NewSize
    return NewSize;
}
</cpp>
</f>
<f>IMenuComponent::UpdateMenu-Формирование контекстного меню компонента
</f>

```

<f>IMenuComponent::OnCommand-Обработчик команд меню

</f>

<f1>IMessengerComponent::AcceptMessage-Обработчик сообщений

<h>

virtual void AcceptMessage(CMessage* pMessage);

</h>

<cpp>

void <NameClass>::AcceptMessage(CMessage* pMessage)

```
{
    switch(pMessage->nWires)
    {
        ^case <nWires>:
        {
            if(pMessage->type==_<type>)
            {
                <Var_Name>=*(<type>*)pMessage->Data;
                <Handler>;
                <Sendler>;
            }
        };break;^
    }
}
```

</cpp>

</f>

<f1>IMessengerComponent::SendMessage-Отправитель сообщений

<h>

virtual void SendMessage(LPVOID Data, int pPin, eTypes Type);

</h>

<cpp>

void <NameClass>::SendMessage(LPVOID Data, int pWires, eTypes Type)

```
{
    CMessage* mes = new CMessage;
    mes->Data=Data;
    mes->nWires=pWires;
    mes->type=Type;
    mes->pComponent=(IComponent*)pMComp;
    OutMessage(mes);
    delete mes;
}
```

</cpp>

</f>

<f1>IMessengerComponent::OnRunExperiment-начало имитационного эксперимента

<h>

virtual void OnRunExperiment();

</h>

<cpp>

void <NameClass>::OnRunExperiment()

```
{
    ^SendMessage(<Data>,<nWires>,_<Type>);^
}
```

</cpp>

</f>

```

<f>IBaseImitationComponent::StepTime-Обработчик начала очередного шага по времени
<h>
virtual void StepTime(double T);
</h>
<cpp>
void <NameClass>::StepTime(double T)
{
//впишите свой текст
^SendMessage(<Data>,<nWires>,<Type>);^
}
</cpp>
</f>
<f1>IMarsInerfsace::SetMarsVarsConnection-Формирование вычислительных связей ком-
понента
<h>
virtual void SetMarsVarsConnection();
</h>
<cpp>
//Запускается в начале эксперимента
void <NameClass>::SetMarsVarsConnection()
{
CountNodes=CountBranch=0;
CountVectorConnections = <CountVectorConnections>;
InitVectorsConnection();
^ SetEnergyConnection(<s>,<nw>,<pA>, <NameNV>, <NLV>, <NameBV>, <BLV>);^
^ SetInformationConnection(<s>,<nw>,<pA>, <NameNV>, <NLV>);^
InitVariables();
CountEquations=<CountEquations>;
CountMembers=<CountMembers>;
};
</cpp>
</f>
<f1>IMarsInerfsace::TopologQuery-Опрос топологического слоя компонентов
<h>
virtual bool TopologQuery(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);
</h>
<cpp>
bool <NameClass>::TopologQuery(CSystemEquations* pEq, CMarsTime* MarsTime,
CDirectParam* DirPar)
{
^ SetTopologWires(pEq,<znak><BLV>,<NLV>);^
return true;
}
</cpp>
</f>
<f1>IMarsInerfsace::CalculatQuery2-Формирование линейных уравнений с постоянными
коэффициентами
<h>
virtual bool CalculatQuery2(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);
</h>

```

```

<cpp>
bool <NameClass>::CalculatQuery2(CSystemEquations* pEq, CMarsTime* MarsTime,
CDirectParam* DirPar)
{
^ int m2_<numEQ>[<CountMem>]={<NumberVariables>};
double am3_<numEQ>[<CountMem>]={<Coefficients>};
SetEquation(pEq,<Type>,<W>,<CountMem>,m2_<numEQ>,am3_<numEQ>);^
return true;
};
</cpp>
</f>

```

<f1>IMarsInerfsace::CalculatQuery1-Формирование линейных уравнений с переменными коэффициентами

```

<h>
virtual bool CalculatQuery1(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);
</h>

```

```

<cpp>
bool <NameClass>::CalculatQuery1(CSystemEquations* pEq, CMarsTime* MarsTime,
CDirectParam* DirPar)
{
double t = MarsTime->T;
^ int m2_<numEQ>[<CountMem>]={<NumberVariables>};
double am3_<numEQ>[<CountMem>]={<Coefficients>};
SetEquation(pEq,<Type>,<W>,<CountMem>,m2_<numEQ>,am3_<numEQ>);^
return true;
};
</cpp>
</f>

```

<f1>IMarsInerfsace::CalculatQuery-Формирование нелинейных и дифференциальных уравнений

```

<h>
virtual bool CalculatQuery(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);
</h>

```

```

<cpp>
bool <NameClass>::CalculatQuery(CSystemEquations* pEq, CMarsTime* MarsTime,
CDirectParam* DirPar)
{
double t=MarsTime->T;
^ int m2_<numEQ>[<CountMem>]={<NumberVariables>};
double am3_<numEQ>[<CountMem>]={<Coefficients>};
SetEquation(pEq,<Type>,<W>,<CountMem>,m2_<numEQ>,am3_<numEQ>);^
return true;
};
</cpp>
</f>

```

<f>IMarsInerfsace::IzmeritQuery-Измерительный опрос

```

<h>
virtual bool IzmeritQuery(CSystemEquations* pEq, CMarsTime* MarsTime = NULL,
CDirectParam* DirPar = NULL);
</h>

```

```

<cpp>
    bool <NameClass>::IzmeritQuery(CSystemEquations* pEq, CMarsTime* MarsTime,
CDirectParam* DirPar)
    {
        double V;//закончите выражение
        SendMessage(&V, ___, _double);
        return true;
    }
</cpp>
</f>
<f>ICalculateControlsInterface::SetStepAnaliz-Установка шага моделирования
<h>
void SetStepAnaliz(double H);
</h>
<cpp>
void <NameClass>::SetStepAnaliz(double h)
{
    ICalculateControlsInterface::SetStepAnaliz(h);
}
</cpp>
</f>
<f>InteractivePanel::CreateWindow-Открытие окна
<h>
virtual void CreatePanelWindow(CWnd* pParentWnd);
</h>
<cpp>
void <NameClass>::CreatePanelWindow(CWnd* pParentWnd)
{

}
</cpp>
</f>

```

Приложение 15. Руководство пользователя генератором моделей

КОМПОНЕНТОВ

П15.1. Общая информация о классе компонента

Общая для всех компонентов информация, предоставляемая редактору многоуровневой КМ для формирования дерева компонентов (рис. 5.3), формируется во вкладке «Главная» (рис. П15.1). На этой вкладке пользователем должны быть заполнены следующие поля:

– **Имя класса** – наименование класса, который реализует данный компонент в рамках БМК. Оно должно быть сформировано из заглавных и строчных букв латинского алфавита, цифр и знака подчеркивания «_». В шаблонах класс компонентов представлен тегом `<NameClass>`;

– **Имя компонента** – строка, с которой формируемый компонент идентифицируется в дереве компонентов (рис. 5.2, 1). Данному полю соответствует тег `<NameComponent>`;

– **Имя папки** – наименование папки в дереве компонентов, в которой должен находиться данный компонент. В шаблонах компонента это поле представлено тегом `<NameFolder>`;

– **Иконка компонента** представляет собой уникальный идентификационный номер иконки, включенный в проект, в котором будет реализовываться разрабатываемый с помощью ГКМ компонент. В шаблонах компонента ей соответствует тег `<IconComponent>`;

– **Иконка папки** указывает уникальный идентификационный номер иконки, включенной в проект, которая будет отображаться около папки, содержащей данный компонент. Она представлена тегом `<IconFolder>`;

– **UID** – уникальный идентификационный номер компонента, представляющий собой число, с помощью которого данный компонент идентифицируется в дереве компонентов и библиотеке моделей компонентов.

На основе информации, представленной на вкладке «Главная» формируется функция `SetInfoComponent()`, в которой заполняется экземпляр структуры `SComponentInfo` (рис. 6.1).

П15.2. Информация о пинах компонента

Формирование информации о пинах компонента, с помощью которых он соединяется с другими компонентами на объектном и логическом слоях редактора многоуровневой КМ СТУС, производится на вкладке «Пины» (рис. П15.2). В столбцы приведенной таблице о каждом пине заносится следующая информация:

Рисунок П15.1 – Вкладка Главная

– **Ориентация** – поле, определяющее положение пина относительно компонента. Оно может принимать одно из следующих значений: *Left* – узел находится слева от изображения компонента, *Right* – справа, *Top* – сверху, *Bottom* – снизу;

– **X** – абсцисса пина относительно компонента;

– **Y** – ордината пина относительно компонента;

– **Имя** – наименование пина, предназначенное для его идентификации в рамках компо-

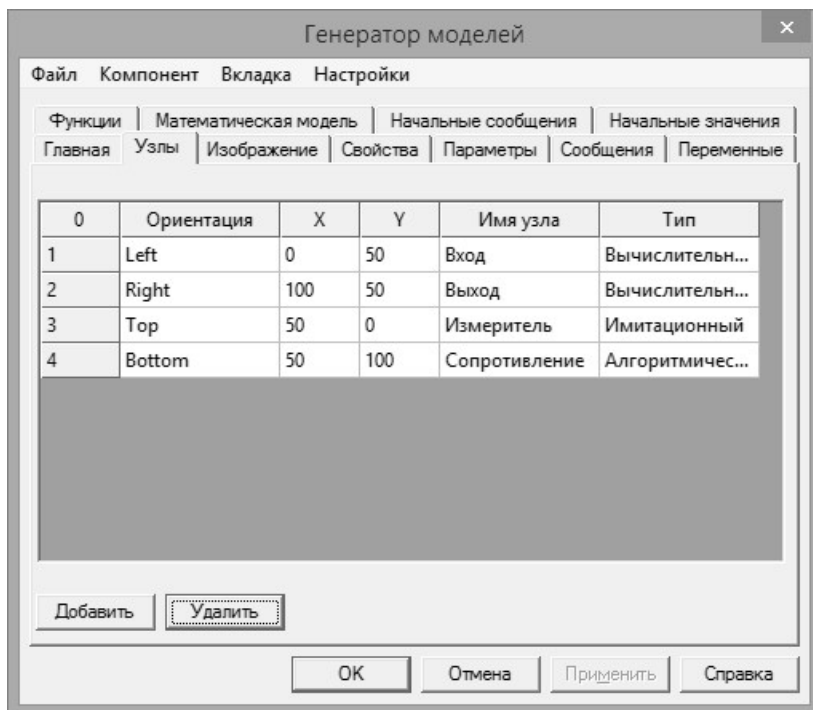


Рисунок П15.2 – Вкладка Пины

– **алгоритмический** – тип пина компонента, имеющего свое отображение на логическом слое редактора и задействованного при формировании и выполнении алгоритмов интеллектуального управления ИСУ СТО. Пину данного типа соответствуют переменные различных типов данных, принимающие или передающие сообщения другим компонентам с помощью универсального имитационного ядра (п. 5.7).

Для каждого пина компонента формируется вызов метода *AddNodes* с соответствующим набором значений его параметров, взятым из заполненной таблицы на вкладке «Пины» (рис. П15.2).

П15.3. Информация об изображениях компонента на слоях редактора

Если изображение компонента на некотором слое редактора, на котором он отображается, не изменяется, то оно называется *статическим* или *векторным*. Если же оно изменяется на каком либо слое, то оно называется *динамическим* и реализуется в методе *OnPaint()* пользователем в виде текста на языке *Visual C++* после того, как основные файлы компонента будут сформированы в ГМК.

В случае векторного изображения компонента его формирование производится в квадрате размером 100x100 точек, приведенном во вкладке «Изображение» (рис. П15.3) из следующих геометрических примитивов:

– **линия**, для которой задаются координаты начала (X_b, Y_b) и конца (X_e, Y_e);

– **Желательно**, чтобы оно было уникальным в рамках имен всех пинов формируемого компонента;

– **Тип** – тип пина, определяющий слой редактора многоуровневой КМ ИСУ СТО, на котором будет отображаться этот пин, а также правила его использования при моделировании. Введены следующие типы пинов компонентов:

– **вычислительный** – тип пина компонента, отображаемого на объектном слое редактора и задействованном при математическом моделировании СТО. Пину данного типа ставятся в соответствие определенные потенциальные и потоковые переменные во вкладке «Переменные», которая будет описана ниже;

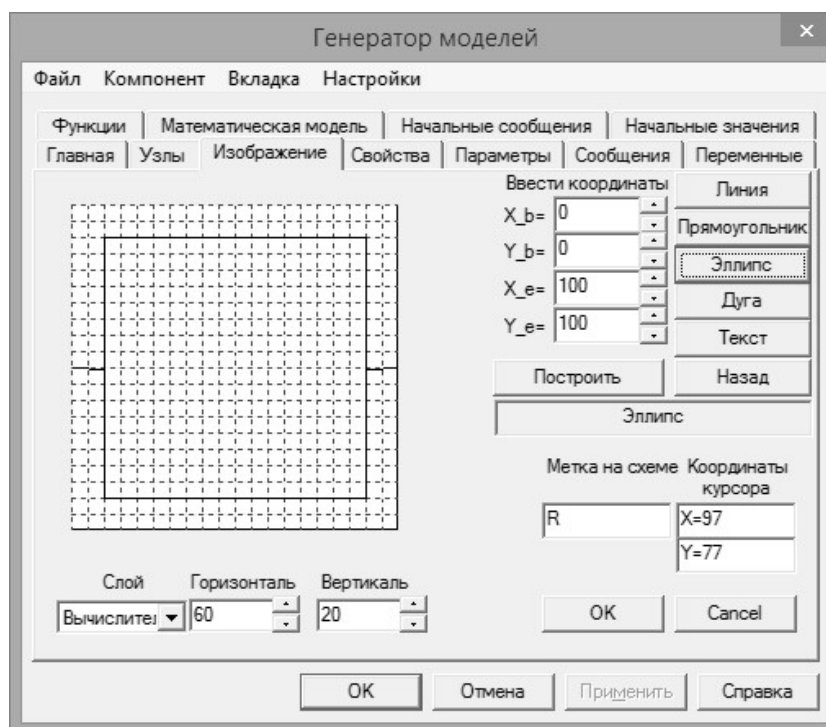


Рисунок П15.3 – Вкладка Изображение

осуществляется при нажатии на соответствующие кнопки на форме. При этом координаты точек могут быть указаны мышью при ее перемещении на поле формируемого изображения, либо в соответствующих полях ввода координат. В этом случае для добавления соответствующего примитива к формируемому изображению необходимо нажать на кнопку **Построить**. При перемещении курсора на поле формируемого изображения его координаты показываются в полях **Координаты курсора**.

На вкладке «**Изображение**» существует возможность выбора конкретного слоя, что выполняется с помощью комбинированного списка «**Слой**». Размеры изображения, в которое должен быть масштабирован рисунок, задаются с помощью полей ввода **Горизонталь** и **Вертикаль**. Для идентификации изображения компонентов может быть задана символическая метка, которая вводится в поле ввода **Метка на схеме**.

На основе введенной информации формируется массив строк, каждая из которых соответствует определенному примитиву (табл. П15.1).

Таблица П15.1

Формирование информации о графических примитивах

Графический примитив	Строка	Параметры
Линия	Line(X_b, Y_b, X_e, Y_e)	(X_b, Y_b) – координаты начала линии; (X_e, Y_e) – координаты конца линии;
Прямоугольник	Rect(X_b, Y_b, X_e, Y_e)	(X_b, Y_b) – координаты левого верхнего угла; (X_e, Y_e) – координаты правого нижнего угла;
Эллипс	Ellipse(X_b, Y_b, X_e, Y_e)	(X_b, Y_b) – координаты левого верхнего угла прямоугольника, в который списан эллипс; (X_e, Y_e) – координаты правого нижнего угла прямоугольника, в который списан эллипс;
Дуга	Arc($X_b, Y_b, X_e, Y_e, X_{b1}, Y_{b1}, X_{e1}, Y_{e1}$)	(X_b, Y_b) – координаты левого верхнего угла прямоугольника, в который списана дуга; (X_e, Y_e) – координаты правого нижнего угла пря-

– *прямоугольник*, для которого задаются координаты левого верхнего (X_l, Y_l) и правого нижнего (X_r, Y_r) углов;

– *эллипс*, для которого задаются координаты левого верхнего (X_l, Y_l) и правого нижнего (X_r, Y_r) углов прямоугольника, в который вписывается формируемый эллипс;

– *дуга*, для которой задаются координаты прямоугольника, в котором должна быть вписана дуга, а также координаты ее начала и конца;

– *текст*, который указывается в специальном диалоговом окне и после чего мышью устанавливаются координаты прямоугольника, в котором должен быть изображен введенный текст.

Выбор соответствующих графических примитивов осу-

		моугольника, в который списана дуга; (X_b1, Y_b1) – координаты начала дуги; (X_e, Y_e) – координаты конца дуги;
Текст	Text(X_b,Y_b,X_e,Y_e,Str)	(X_b, Y_b) – координаты левого верхнего угла прямоугольника, в который вписан текст; (X_e, Y_e) – координаты правого нижнего угла прямоугольника, в который вписан текст; Str – подлежащий отображению текст.

П15.4. Информация о свойствах компонента

Свойствами будем называть члены-данные класса, характеризующие данный компонент. Формирование свойств класса компонента осуществляется во вкладке «Свойства» (рис. П15.4). На ней в табличной форме заносится информация о каждом входящем в класс свойстве:

- *Тип* – тип свойства, который выбирается комбинированным списком и может принимать одно из следующих значений: *bool* – логический тип данных; *long* – целочисленный тип данных; *double* – вещественный тип данных; *complex* – комплексный тип данных; *ArrayBool* – массив значений логического типа данных; *ArrayLong* – массив целочисленных значений; *ArrayDouble* – массив вещественных значений; *ArrayComplex* – массив комплексных значений; *string* – строка; *strings* – массив строк; *combo* – комбинированный список, позволяющий пользователю выбора одного значения из имеющегося набора; *file* – некоторый файл на диске или в локальной сети с полным путем к нему; *pen* – карандаш, т.е. структурный тип данных, определяющий правила прорисовки на экране различных геометрических примитивов; *brush* – кисть – структурный тип данных, определяющий правила закрашивания замкнутых графических областей на экране; *gradient* – вид заливки замкнутой графической области, в которой указываются два цвета, плавно переходящие один в другой; *font* – шрифт, представляющий собой структурный тип данных, определяющий параметры вывода текстовых элементов на экран; *colorref* – цвет, задаваемый в формате RGB (Red-Green-Blue);

- *указатель* – метка, показывающая, является ли вводимая переменная статической (метка не устанавливается) или динамической (метка устанавливается). Под статическую переменную любого типа память выделяется на этапе компиляции программы. Для динамической переменной (массива переменных) память должна быть выделена в процессе работы программы;

- *имя переменной* представляет собой строку из строчных и прописных букв латинского алфавита, цифр и знака подчеркивания (), которая начинается с буквы;

- *количество* – поле, в котором указывается размерность массива. Если это поле пустое, то соответствующее свойство будет считаться простой переменной;

- *значение* – ука-

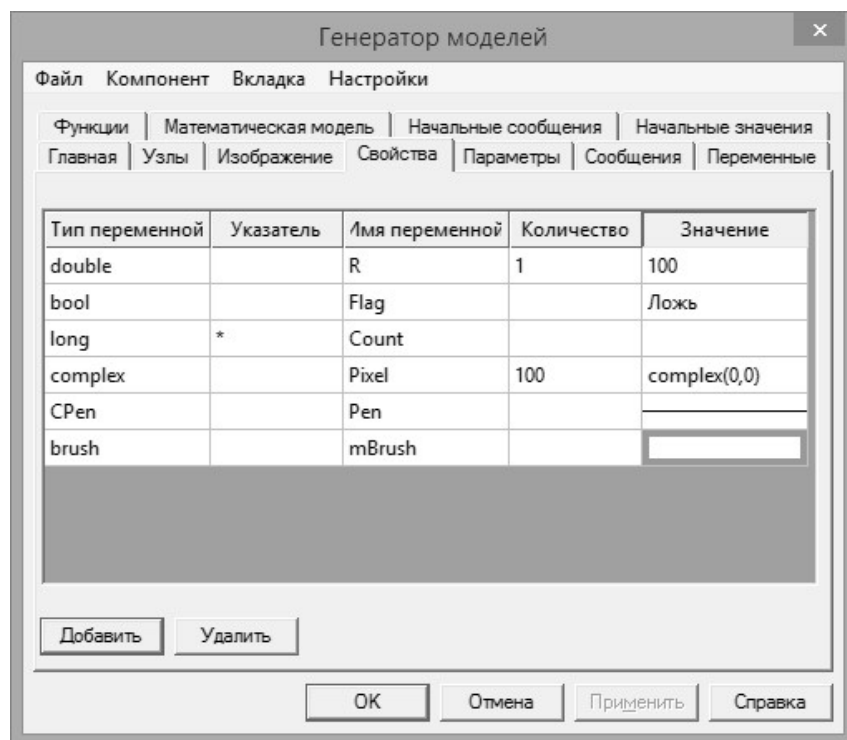


Рисунок П15.4 – Вкладка Свойства

зывается значение соответствующей переменной, которой должно быть присвоено ей в конструкторе класса компонента.

Для каждого свойства компонента в заголовочном классе компонента формируется строка вида

```
<type> <name>;
```

где *<type>* – тип свойства, указанный в первом столбце таблицы; *<name>* – имя свойства.

Если у свойства задано начальное значение *<value>*, то в конструкторе класса формируется строка вида

```
<name>=<value>;
```

П15.5. Информация о параметрах компонента

Параметрами называются свойства компонента, значения которых могут быть заданы пользователем в редакторе атрибутов компонента редактора многоуровневой МК ИСУ СТО. Для того, чтобы некоторое свойство компонента стало его параметром, необходимо во вкладке

Параметры в ГМК (рис.П15.5) напротив этого свойства ввести **Имя параметра**.

Для каждого параметра, для которого было введено имя, в конструкторе класса компонента прописывается вызов функции

```
AddParam(<type>,
<name>, <var>);
```

где *<type>* – тип свойства, которое должно стать параметром компонента, *<name>* – имя данного свойства, *<var>* – переменная, за которой закреплено данное свойство.

Таким образом, свойство любого типа данных, кроме массивов, может стать параметром компонента.

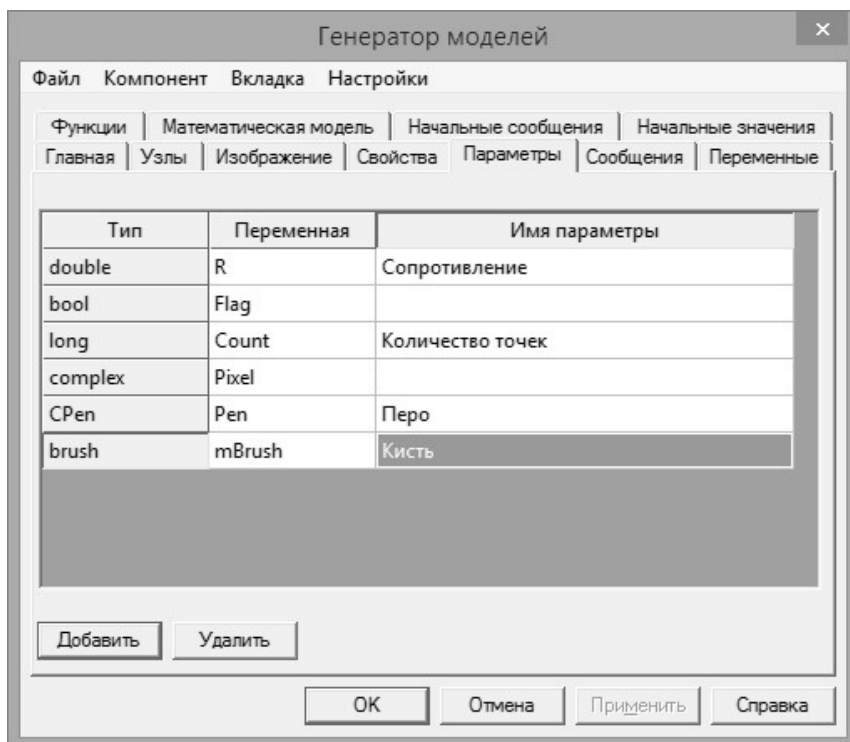


Рисунок П15.5 – Вкладка Параметры

П15.6. Начальные значения переменных

Если какой-либо переменной класса (например, счетчику сообщения или счетчику нажатия левой кнопки мыши.) должно быть присвоено значение в начале эксперимента (начальное значение), то его можно задать на вкладке **Начальные значения** (рис. П15.6). В столбце «Задание» напротив данного свойства необходимо выбрать пункт «Задание», а в столбце «Значение»

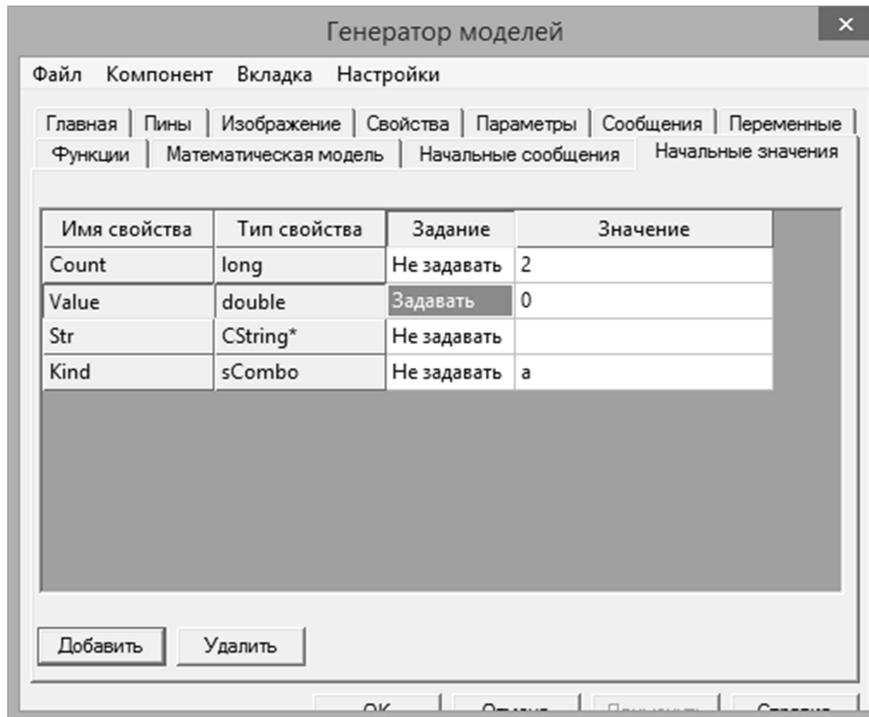


Рисунок П15.6 – Вкладка Начальные значения

следует ввести значение соответствующего типа данных, которое должна принять переменная.

Присваивание указанным переменным значений, заданных на вкладке **Начальные значения**, производится в методе *OnRun()*, который вызывается в начале эксперимента для каждого компонента, находящегося на любом слое редактора многоуровневой КМ ИСУ СТО. В этом методе для каждой такой переменной прописывается строка кода

$$\langle Var \rangle = \langle Value \rangle;$$

где $\langle Var \rangle$ – имя переменной; $\langle Value \rangle$ – значение, которое следует присвоить переменной.

П15.7. Формирование математической модели компонента СТО

Перед непосредственным формированием математической модели компонента СТО, которая представляет собой совокупность уравнений вида (2.9)-(2.12), каждой связи компонента, для которой на вкладке **Пины** указан тип **Вычислительный**, должны быть поставлены в соответствии определенные (потенциальные и потоковые) переменные связи. Это осуществляется во вкладке **Переменные** (рис. П15.7). На данной вкладке для каждой связи, идентифицируемой по имени пина, задается:

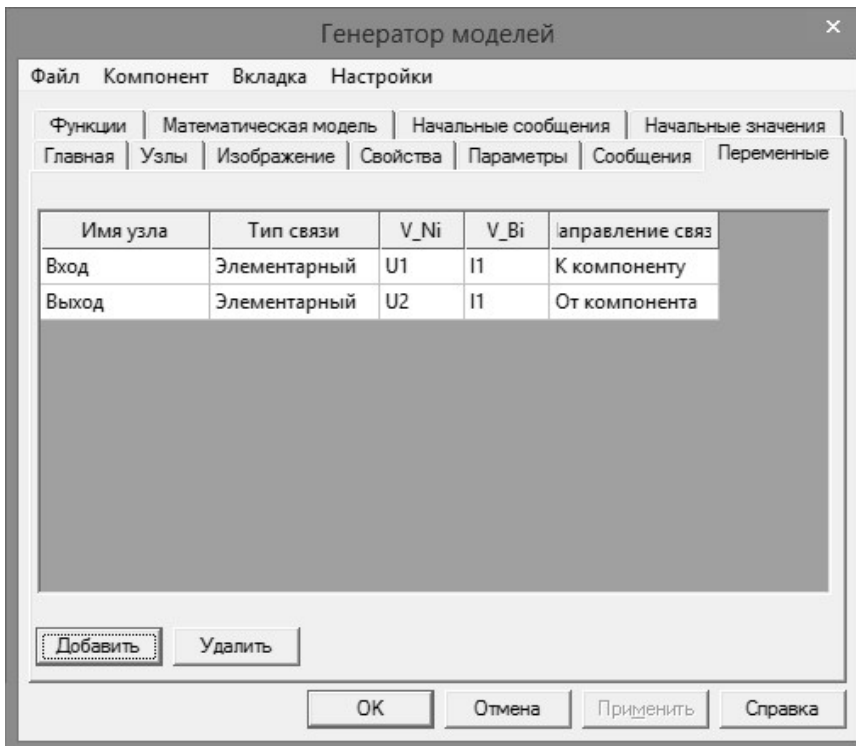


Рисунок П15.7 – Вкладка Переменные

– тип связи – элементарная (2.4), информационная (2.5) или однородная векторная связь (2.6), представляющая собой совокупность информационных связей;

– тип связи – элементарная (2.4), информационная (2.5) или однородная векторная связь (2.6), представляющая собой совокупность информационных связей;

– V_Ni – имя потенциальной переменной связи, которым она идентифицируется в математической модели компонента. Указывается как для информационных, так и для элементарных связей;

- V_{Bi} – имя потоковой переменной связи, которым она идентифицируется в математической модели компонента. Указывается только для энергетических связей;
- *Направление связи* – указывает направление потока данного вида энергии в компоненте.

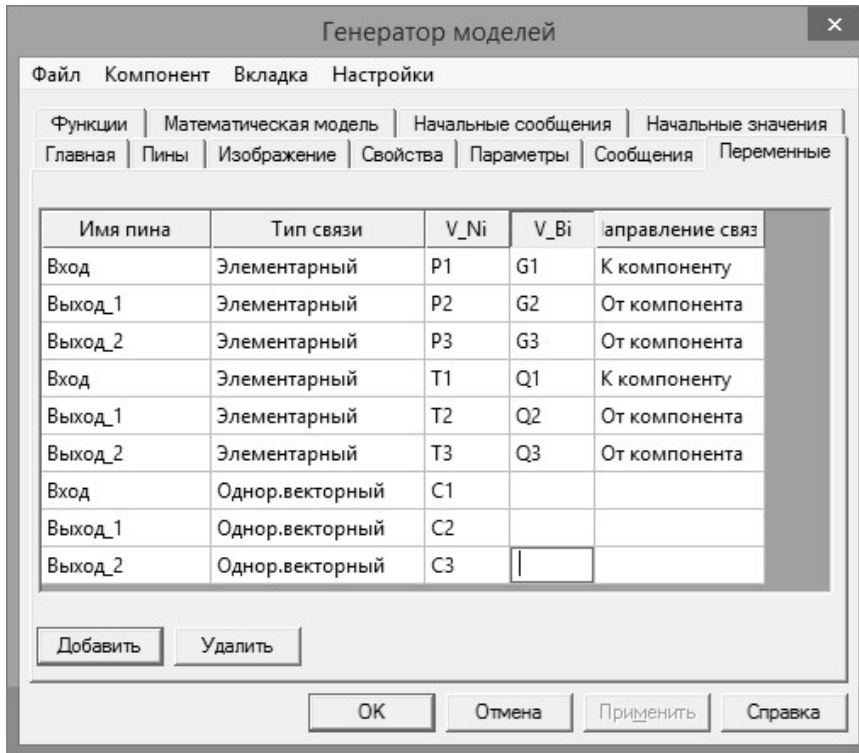


Рисунок П15.8 – Формирование переменных связей компонента с неоднородными векторными связями

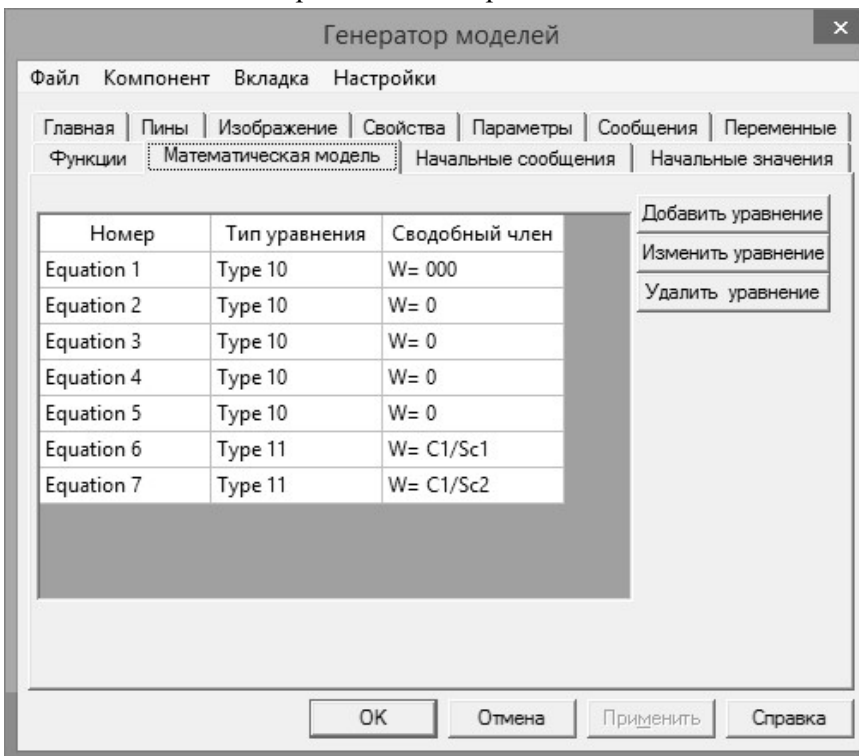


Рисунок П15.9 – Вкладка Математическая модель

Для формирования неоднородных векторных связей следует добавить соответствующее количество элементарных и (или) информационных связей. Для этого следует нажать кнопку **Добавить** и в появившейся строке выбрать *имя пина*, для которого формируется еще одна связь. Это действие необходимо выполнить для каждого пина с НВС.

Пример вкладки **Переменные** для формирования компонента «Сепаратор» (рис. 3.11), в каждой связи которого (3.40) должны учитываться гидравлический и термодинамический энергетические потоки, а также вектор концентраций входящих в многокомпонентный поток веществ, представлен на рис. П15.8.

Формирование уравнений математической модели компонента производится в ГМК во вкладке **Математическая модель** (рис. П15.9). На данной вкладке с помощью расположенных на ней кнопок пользователь имеет возможность выполнять действия, соответствующие одноименным кнопкам:

- *Добавить уравнение*, при этом сформированное в специальной форме уравнение добавляется в конец списка уравнения;

- *Изменить уравнение*. Этому действию подлежит уравнение, строка которого является активной в момент нажатия на кнопку;

– *Удалить уравнение*. При нажатии на эту кнопку удалению подлежит уравнение, строка которого является активной.

При нажатии на кнопку **Добавить уравнение** открывается пустая, а при нажатии на

Переменная	Производная	Коэффициент
G1	Переменная	Sc1
G2	Переменная	-1

Рисунок П15.10 – Форма информации уравнения компонента

кнопку **Изменить уравнение** открывается заполненная форма **Уравнение** (рис. П15.10), в полях которой пользователем приводятся следующие данные об уравнении компонента:

– в поле **Класс уравнения** с помощью комбинированного списка выбирается класс уравнения, соответствующий табл. 5.1. При формировании компонентных уравнений он заносится в массив *KL* структуры системы уравнений *SSystemEquation*;

– в поле **Свободный член уравнения** формируется выражение для свободного члена уравнения, который при формировании системы уравнений заносится в массив *W* структуры *SSystemEquation*;

– в поле **Количество членов** указывается количество переменных с коэффициентами, которые входят в уравнение линейно или под знаком дифференцирования;

– в представленной на форме таблице указываются **Переменные**, которые выбираются из комбинированного списка, для каждой из них указывается вид вхождения в уравнение (переменная или производная), а также вводится коэффициент, с которым эта переменная входит в уравнение. Коэффициент может представлять собой число, некоторый параметр компонента или составленное относительно них математическое выражение.

При завершении формирования уравнения следует нажать кнопку **Ок**. Если пользователь нажмет кнопку **Отмена** введенные данные и сделанные изменения не сохранятся.

На основе сформированной во вкладках **Переменные** и **Математическая модель** информации формируются функции компонента по опросам его топологического и вычислительного блоков. Такой компонент должен быть наследован от интерфейсного класса *IMarsInterface*. При этом в формируемом ГМК методе *SetLKBVariables()* на основе физического координатного базиса, относительно которого в ГМК формируются компонентные уравнения, осуществляется формирование ЛКБ компонента. С этой целью для каждой элементарной связи, входящей в любую НВС компонента, осуществляется вызов метода

$$\text{SetEnergyConnection}(\langle nNVS \rangle, \langle NameNVar \rangle, \langle NumberNVar \rangle, \langle NameBVar \rangle, \langle NumberBVar \rangle),$$

где $\langle nNVS \rangle$ – номер соответствующей НВС, которая соответствует порядковому номеру пина компонента на объектном слое редактора; $\langle NameNVar \rangle$ – имя потенциальной переменной, которым она выражена в физическом КБ компонента; $\langle NumberNVar \rangle$ – номер потенциальной переменной данной энергетической связи в ЛКБ компонента; $\langle NameBVar \rangle$ – имя потоковой переменной, которым она выражена в физическом КБ компонента; $\langle NumberBVar \rangle$ – номер потоковой переменной в ЛКБ компонента. В данном методе производится формирование структуры *SEnergyConnection*, содержащей информацию о данной энергетической связи компонента.

Для каждой информационной связи, входящей в состав какой-либо НВС компонента производится вызов метода

$$\text{SetInformationConnection}(\langle nNVS \rangle, \langle NameNVar \rangle, \langle NumberNVar \rangle)$$

В данном методе осуществляется формирование структуры *CInformationConnection*, которая включает в себя данные о рассматриваемой информационной связи, включенной в состав НВС. Он же вызывается для каждой информационной связи, входящей в состав однородной векторной связи.

Сформированная с помощью данных методов информация применяется для формирования КЦ, то есть интерпретации ЛКБ каждого входящего в нее компонента в ГКБ формируемой цепи.

С целью опроса топологического блока компонента, включающего в свои НВС энергетические связи, в его программно-алгоритмический код включается метод *TopologQuery*. В данном методе для каждой энергетической связи производится вызов метода

$$\text{SetTopologWires}(\langle pEq \rangle, \pm \langle NumberBVar \rangle, \langle NumberNVar \rangle)$$

где $\langle pEq \rangle$ – структура системы уравнений *SSystemEquations*, в которой формируется топологический массив $m1$; $\langle NumberBVar \rangle$ – номер потоковой переменной в ЛКБ компонента; $\langle NumberNVar \rangle$ – номер потенциальной переменной в ЛКБ компонента.

В реализующие опрос вычислительного блока компоненте методы формируемого класса добавляются функции формирования соответствующих уравнений вычислительной модели компонента. Для составления каждого уравнения в методах *CalculatQuery()*, *CalculatQuery1()* и *CalculatQuery2()* осуществляется вызов метода

$$\text{SetEquation}(\langle pEq \rangle, \langle KL \rangle, \langle W \rangle, \langle CV \rangle, \langle m2 \rangle, \langle am3 \rangle)$$

где $\langle pEq \rangle$ – структура системы уравнений *SSystemEquations*, в которую добавляется информация о формируемом уравнении; $\langle KL \rangle$ – класс уравнения в соответствии с табл. 5.1; $\langle W \rangle$ – свободный член уравнения; $\langle CV \rangle$ – количество переменных, в том числе стоящих под знаком дифференциала, и коэффициентов к ним, которое определяет размерность массивов $m2$ и $am3$; $m2$ – массив номеров входящих в уравнение переменных, заданных в ЛКБ компонента; $am3$ – массив коэффициентов, стоящих перед переменными в формируемом с помощью метода уравнении.

П15.8. Принципы формирования компонентов языка моделирования алгоритмических конструкций

Компоненты, задействованные в формировании и выполнении алгоритмов ИУ СТО и принадлежащие логическому уровню многоуровневой КМ ИСУ СТО (рис. 2.5), должны иметь свое отображение на логическом слое многослойного редактора (рис. 5.2). Процесс моделирования сформированных таким образом алгоритмических КЦ заключается в приеме и отправке данных различных типов с помощью сообщений *CMessage* (рис. 5.27).

Подготовка и отправка сообщений компонентов может проводиться как в начале эксперимента, в начале очередного шага анализа КМ СТО, принадлежащей объектному уровню, а также при получении определенного сообщения от других компонентов, подключенных к входным пинам данного компонента.

Для формирования программного кода алгоритмического компонента, позволяющего подготавливать и отправлять сообщения в начале эксперимента и в начале очередного шага анализа КМ СТО, используется вкладка **Начальные сообщения** (рис. П15.11). На данной вкладке напротив пина, с которого должно быть отправлено сообщение, пользователем указывается:

– **Пин** – указывается пин, для которого на вкладке **Пины** (рис. П15.2) указан тип алгоритмический и с которого должно отправляться сообщение либо в начале эксперимента, либо же в начале очередной итерации анализа КМ СТО, принадлежащей объектному уровню многоуровневой КМ ИСУ СТО;

– **Тип сообщения** указывается момент, когда требуется отправлять сообщение: В начале

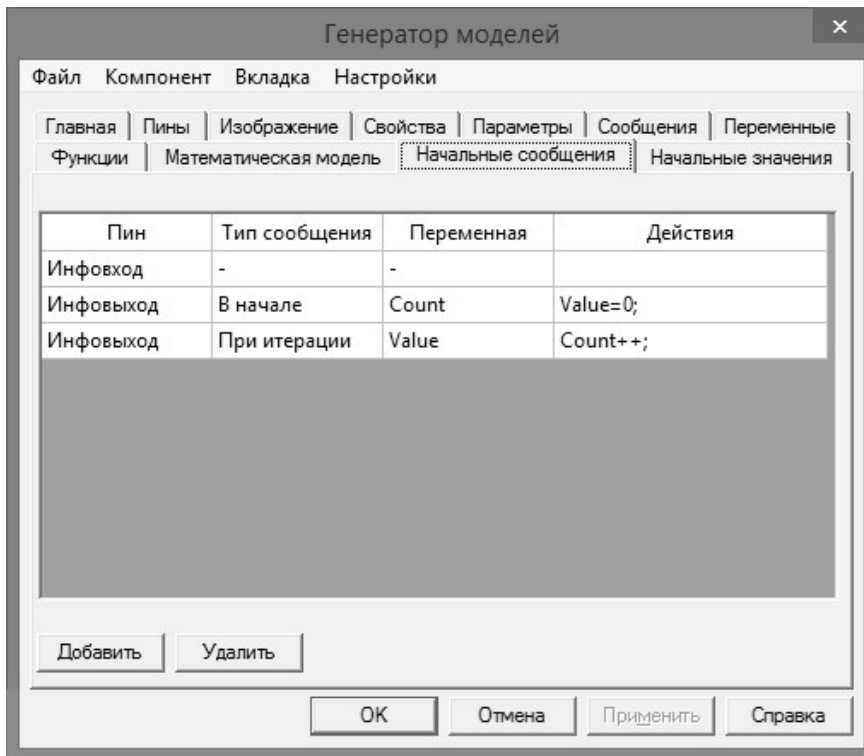


Рисунок П15.11 – Вкладка Начальные сообщения

рования и отправления сообщений в начале эксперимента;

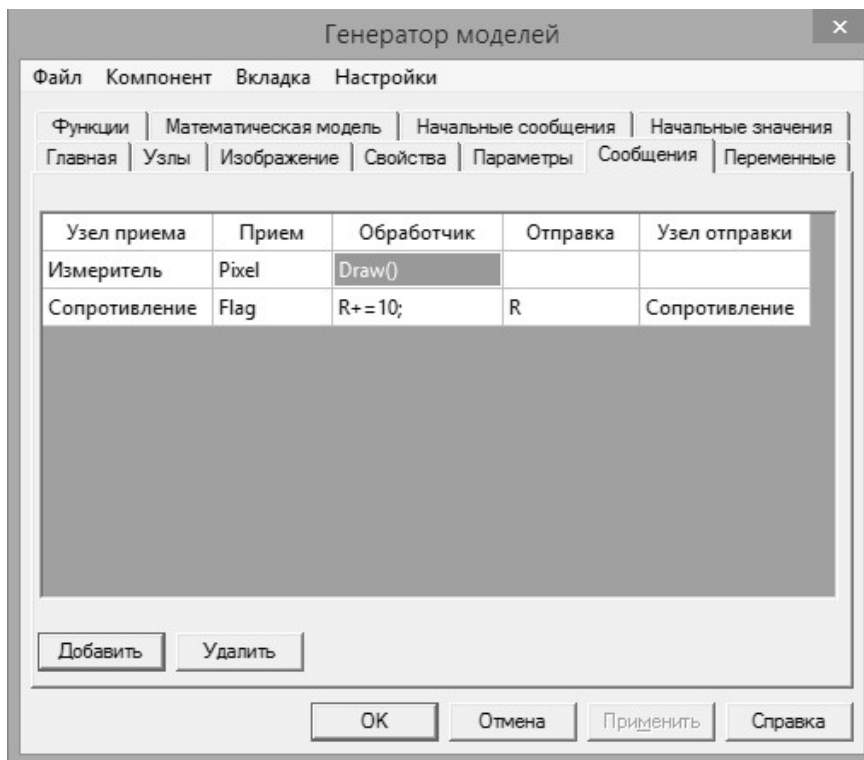


Рисунок П15.12 – Вкладка Сообщения

эксперимента («В начале») либо же в начале очередной итерации («При итерации»);

– **Переменная** – из комбинированного списка выбирается одно из свойств компонента, введенных на вкладке **Свойства** (рис. П15.4), значение которого должно быть передано в сообщении;

– **Действия** – указываются программные фрагменты, которые должны быть выполнены непосредственно перед отправкой сообщения.

На основе введенных во вкладку «Начальные сообщения» данных формируются следующие методы класса формируемого компонента:

1) *OnRunExperiment()* – предназначенный для формирования и отправления сообщений в начале эксперимента;

2) *StepTime()*, предназначенный для формирования и отправки сообщения при каждой итерации анализа КМ СТО, принадлежащей объектному уровню многоуровневой компьютерной модели ИСУ СТО.

Формирование метода получения сообщений, обработки содержащихся в них информации и отправки результата другим компонентам, что характерно для компонентов-преобразователей (рис. 2.13, б), производится во вкладке **Сообщения** (рис. П15.12). На данной вкладке для каждого **пина приема**, имеющего тип **Алгоритмический**, в полях таблицы указывается:

– в поле **Прием** указы-

вается свойство компонента, которому будут присвоены данные, содержащиеся в принимаемом сообщении, при условии совпадения типа свойств с типом поступающей информации;

– в поле **Обработчик** указывается программный код на языке *Visual C++*, который должен выполняться между моментами приема и отправки сообщений;

– в поле **Отправка** – переменная, значение которой должно быть передано в сообщении, отправляемом по результатам приема и обработки поступивших данных;

– поле **Пин отправки** должно содержать имя выходного пина, имеющего тип **Алгоритмический**, с которого будет отправлено сообщение с данными, хранящимися в переменной, указанной в поле **Отправка**.

По умолчанию ГМК предоставляет возможность сформировать обработчики для одного сообщения, пришедшего на каждый из пинов компонента. Если для какого-либо пина необходимо сформировать еще несколько обработчиков сообщения, то следует нажать на кнопку **Добавить** и в появившейся строке ввести все данные, в том числе и **Пин приема**, на который компонент должен получить сообщение.

По результатам обработки данных, введенных пользователем во вкладке **Сообщения**, производится формирование метода *OnMessage()* компонента, где производится разбор поступающих сообщений по номерам пинов, на которые поступают входящие в компонент сообщения.

П15.9. Включение функций в программно-алгоритмическую реализацию компонента

При формировании программно-алгоритмической реализации компонента в нее можно сразу включить методы различных интерфейсных классов, описанных в п.6.1. и обеспечивающих интерфейсные возможности компонента. Для включения определенного метода на вкладке **Методы** (рис. П15.13) следует напротив его имени и пояснения установить комбинированный список **Включение** в положение **Включить**. В этом случае в программно-алгоритмический код компонента будет добавлен пустой метод, который пользователь должен реализовать по своему усмотрению. Полный список таких методов и наименование интерфейсных классов, в которых реализованы их прототипы, приведены в приложении 13.

Выбрав пункт **Генерировать** в меню **Компонент**, на основе шаблонов заголовочного (*template.h*) и исполняемого (*template.cpp*) файлов, а также на основе шаблонов основных методов, включенных в различные интерфейсные классы, формируются заголовочный и исполняемый файлы компонента. Подключив данные файлы к проекту, в котором реализованы компоненты БМК, сгенерированный компонент будет доступен для использования в комплексе программ многоуровневого компьютерного моделирования интеллектуальных систем управления сложными технологическими объектами.

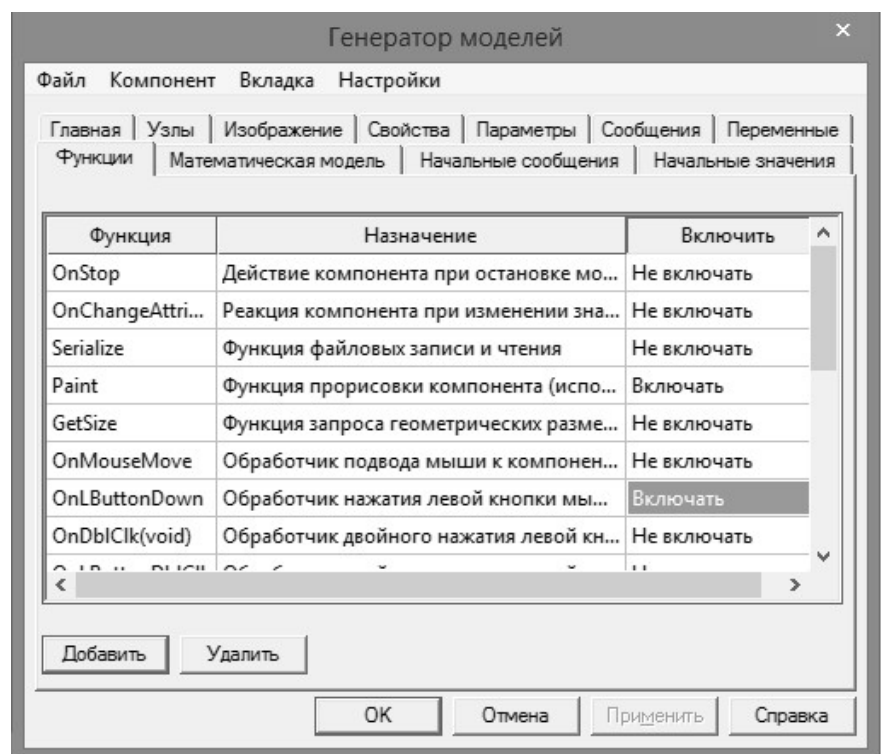


Рисунок П15.13 – Вкладка Методы

Приложение 16. Свидетельство СибРУМЦ



УТВЕРЖДАЮ:

Зам. председателя СибРУМЦ,
профессор

Сидор С. А. С. А. Подлесный
«14» февраля 2015 г.

ЗАКЛЮЧЕНИЕ

Сибирского регионального учебно-методического центра (СибРУМЦ) высшего профессионального образования

Учебно-методический совет СибРУМЦ по электротехнике, электромеханике и электротехнологии рассмотрел возможность издания учебного пособия «Теоретические основы электротехники. Часть I. Установившиеся режимы в линейных электрических цепях». Авторы – д.т.н., профессор зав. кафедрой Дмитриев Вячеслав Михайлович, к.т.н., с.н.с., доцент кафедры Шутенков Александр Васильевич, к.т.н., доцент кафедры Хатников Валентин Иванович, к.т.н., доцент кафедры Ганжда Тарас Викторович – сотрудники МИСА ФГБОУ ВПО «Томский государственный университет систем управления и радиоэлектроники». Объем пособия – 187 с.

Представленное учебное разработано по курсу «Электротехника и электроника» для студентов, обучающихся по направлениям подготовки бакалавров 27.03.03 (220100.62) «Системный анализ и управление» и 27.03.04 (220400.62) «Управление в технических системах».

Структура учебного пособия соответствует установившимся нормам, содержание соответствует примерной учебной программе ТОО и ОПЦ, а также действующим ФГОС ВПО.

Учебное пособие содержит достаточное количество практических примеров, задач, вопросов для самоконтроля знаний обучающихся, контрольных заданий и ответов, что выгодно отличает его от изданий других авторов. Примеры и задачи подобраны удачно. Учитывая непрерывное уменьшение количества часов на изучение курсов ТОО и ОПЦ, пособие охватывает все разделы линейных электрических цепей в обоснованном сокращенном виде.

Учебное пособие написано четким, лаконичным и ясным языком с высоким методическим уровнем.

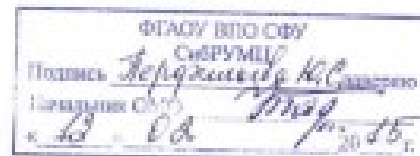
Пособие имеет хорошее полиграфическое качество и удобно для пользователя, соответствует типовой программе учебных дисциплин ТОЭ и ОТЦ и ФГОС ВПО.

На основании вышесказанного СибРУМЦ считает возможным издать учебное пособие «Теоретические основы электротехники. Часть I. Установившиеся режимы в линейных электрических цепях» (авторы – В. М. Дмитриев, А. В. Шутенков, В. И. Хатников, Т. В. Ганжда; ФГБОУ ВПО «Томский государственный университет систем управления и радиоэлектроники») с рекомендацией: «Рекомендовано Сибирским региональным учебно-методическим центром высшего профессионального образования для межвузовского использования в качестве учебного пособия для студентов, обучающихся по электротехническим и радиотехническим направлениям и специальностям».

Член УМС СибРУМЦ по электротехнике,
электромеханике и электротехнологии,
профессор кафедры «Электротехнология
и электротехника» ФГАОУ ВПО «Сибирский
федеральный университет», к.т.н., профессор,
Лауреат премии Правительства РФ
в области образования 2013 г.



Ю. С. Перфильев



Приложение 17. Акты внедрения



ТОМСКНИПИНЕФТЬ
ИНСТИТУТ НЕФТЯНОГО ПРОЕКТИРОВАНИЯ

ОТКРЫТОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО
 «ТОМСКИЙ НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ И ПРОЕКТИВНЫЙ ИНСТИТУТ НЕФТИ И ГАЗА»
 (ОАО «ТОМСКНИПИНЕФТЬ»)

634027, Российская Федерация, г. Томск, Мира пр., д. 72,
 Тел.: (3822) 61-19-90, Факс: (3822) 61-18-80
 nipi@tomsknpi.ru
 ОКПО 44235454, ОГРН 1027000658170, ИНН/КПП 7021049088/701750001



от _____ № _____



УТВЕРЖДАЮ
 Генеральный директор
 ОАО «ТомскНИПИнефть»

Д.О. Окшнев

АКТ ВНЕДРЕНИЯ

результатов диссертационной работы Ганджи Т.В. по теме
 «Развитие метода компонентных цепей для реализации комплекса программ
 моделирования химико-технологических систем» (соискание ученой степени д.т.н. по
 специальности 05.13.18 – Математическое моделирование, численные методы и
 комплексы программ).

Мы, нижеподписавшиеся, составили настоящий акт об использовании указанных далее результатов диссертационной работы Ганджи Тараса Викторовича по теме «Развитие метода компонентных цепей для реализации комплекса программ моделирования химико-технологических систем», представленной на соискание ученой степени доктора технических наук по специальности 05.13.18 – Математическое моделирование, численные методы и комплексы программ.

Результаты диссертационного исследования Ганджи Тараса Викторовича были использованы при разработке сценариев функционирования автономных компьютерных тренажеров операторов в рамках исполнения договора В042316/0760Д на выполнение проектных работ по объекту «Обустройство Кузюбинского месторождения. Интегрированная автоматизированная система управления и безопасности (Фаза 1.1)». В частности, в процессе проектирования объекта был использован комплекс программ «Среда моделирования химико-технологических

систем» для моделирования и исследования технологических процессов в штатных и аварийных режимах функционирования автоматизированной системы управления технологическими процессами.

Начальник Управления по ИТ и
развитию бизнес-процессов, к.т.н.



А.В. Лыпин

Ученый секретарь, к.т.н.



А.Г. Чернов

Начальник отдела автоматизации
технологических процессов



И.В. Федченко

И.о. главного инженера проекта,
ведущий инженер отдела автоматизации
технологических процессов, к.т.н.



И.С. Грехов

Ведущий инженер отдела автоматизации
технологических процессов



А.Г. Зибзен



Федеральное государственное унитарное предприятие «Научно-производственное объединение по медицинским иммунобиологическим препаратам «Микроген»
Министерства здравоохранения Российской Федерации
Филиал ФГУП «НПО «Микроген» Минздрава России в г. Томск «НПО «Вирион»

ОГРН 1037722027727, ИНН 7722292838, КПП 701702001, ОКПО 14407886, ОКВЭД 72.19
Место нахождения: ул. Ивановского, 8, г. Томск, 634040 Почтовый адрес: ул. Ивановского, 8, г. Томск, 634040
e-mail: info.tomsk@microgen.ru Тел.: 8 (3822) 905 808, факс: 8 (3822) 905 809

УТВЕРЖДАЮ

Главный инженер филиала
ФГУП «НПО «Микроген»
Минздрава России

г. Томск «НПО «Вирион»
Шарахов



Шарахов
» 02 2017 г.

АКТ

о внедрении результатов диссертационного исследования Ганджи Тараса Викторовича

Составлен комиссией:

- председатель - Черноусов В.В., начальник отдела метрологии;
- член комиссии - Гурьянов О.В., ведущий инженер по КИПиА;
- член комиссии - Скрынченко А.В., ведущий инженер по КИПиА.

В комплексе программ «Среда многоуровневого компьютерного моделирования ИАРС», разработанном Ганджой Т.В., была построена многоуровневая компьютерная модель термостата для обработки наполненных растворами ампул, инструмента и одежды, используемых на предприятии. С её помощью определены оптимальные режимы работы термостата, позволившие без ухудшения качества стерилизации сократить затраты времени и электрической энергии на использование автоклава.

Основными преимуществами разработанной многоуровневой компьютерной модели являются:

- 1) возможность совместного моделирования пневматических и термодинамических процессов, протекающих в термостате, в динамическом режиме;

2) возможность нахождения оптимальных по времени работы и затратам электрической энергии режимов функционирования различных технологических объектов и аппаратов, основанных на измерении давления и температуры рабочей среды (воздушной или жидкой);

3) построение и отработка на модели стерилизатора различных алгоритмов его работы, обеспечивающих применение безопасных режимов на реальном оборудовании;

4) наличие варьирования параметров моделей исполнительных устройств, позволяющих выработать оптимальные режимы стерилизации;

5) развитые средства визуализации результатов моделирования и интерактивного управления параметрами компонентов исследуемой компьютерной модели стерилизатора в интерактивном режиме.

Заключение:

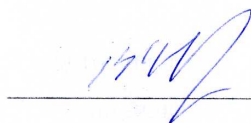
Разработка новых режимов работы стерилизатора для обработки ампул с инъекционными растворами, инструмента и одежды, обеспечила:

– оптимальное использование стерилизатора за счет сокращения времени на изменение температурных режимов;

– безопасность стерилизации ампул, чувствительных к резкому перепаду температуры в автоклаве;

– сокращение энергозатрат на использование стерилизатора.

Председатель комиссии

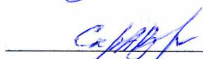


В.В. Черноусов

Члены комиссии



О.В. Гурьянов



А.В. Скрынченко

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Томский государственный университет систем управления и радиоэлектроники»

Акт ввода в эксплуатацию (внедрения) № _____
«30» 09 2014 г.

Программное обеспечение «Среда многоуровневого компьютерного моделирования интеллектуальных систем управления сложными техническими и технологическими объектами с их компьютерными моделями в контуре управления»

Комиссия в составе:

Председатель А.А.Шелупанов,
Н.Д. Малютин;
В.И. Авдзейко;
П.Е. Троян;
Д.Д. Зыков.

созданная на основании распоряжения ректора № 67 от 15 сентября 2014 г., составила настоящий акт о нижеследующем:

1. Комиссия в период с 29 по 30 201__ года провела приемку работ временного трудового коллектива № 02/14.
2. Комиссии при приемке были предъявлены:
 - Комплекс программ «Среда многоуровневого компьютерного моделирования интеллектуальных систем управления сложными техническими и технологическими объектами (ИСУ СТО)», включающий в себя:
 - многослойный редактор компьютерной модели ИСУ СТО;
 - интерпретатор языка СТО;
 - универсальное вычислительное ядро;
 - набор измерительных компонентов;
 - интерпретатор математико-алгоритмических выражений;
 - универсальное имитационное ядро;
 - библиотеку моделей компонентов;
 - интерактивные панели;
 - генератор моделей компонентов;
 - измерительно-управляющий контроллер;
 - средства подключения к хранилищам данных и знаний;
 - редактор панелей визуализации и интерактивного управления;
 - систему интерактивного документирования.
- 2.1. Комиссия установила:

В рамках реализации мероприятия выполнены следующие работы:

 - Разработан многослойный редактор компьютерных моделей, позволяющий формировать многоуровневые компьютерные модели интеллектуальных систем управления сложными техническими и технологическими объектами;
 - Реализовано универсальное вычислительное ядро, предназначенное для анализа компьютерных моделей сложных технических и технологических объектов с неоднородными векторными связями;
 - Разработаны измерительные компоненты, предназначенные для передачи результатов моделирования с объектного уровня компьютерной модели на ее логический уровень;

- Реализован интерпретатор математико-алгоритмических выражений, осуществляющий построение алгоритмической компонентной цепи схем алгоритмов, представленных на логическом уровне многоуровневой компьютерной модели;
- Сформировано универсальное имитационное ядро, предназначенное для анализа алгоритмов интеллектуального управления;
- Разработана библиотека моделей компонентов, позволяющая хранить в себе компоненты объектного, логического и визуального уровня многоуровневой компьютерной модели ИСУ СТО;
- Реализованы интерактивные панели, предоставляющие интерфейс для ввода и редактирования моделей компонентов объектного и логического уровня многоуровневой компьютерной модели в естественно математическом виде;
- Разработан генератор моделей компонентов для формирования программно-алгоритмического вида моделей компонентов объектного, логического и визуального уровней многоуровневой компьютерной модели ИСУ СТО;
- Созданы средства подключения к хранилищам данных, представляющие собой отдельную группу компонентов логического слоя многоуровневой компьютерной модели ИСУ СТО для подключения к представленным в виде файлов на компьютере или расположенных в сети Интернет базам данных и базам знаний;
- Разработан редактор панелей визуализации и интерактивного управления, предназначенный для построения из визуальных компонентов панелей, на которых будут располагаться графические средства отображения результатов моделирования и измерения, а также данных, полученных на различных этапах работы алгоритмов интеллектуального управления, представленных алгоритмическими компонентными цепями на логическом слое редактора;
- Реализована система интерактивного документирования, позволяющая формировать отчеты о работе многоуровневой компьютерной модели ИСУ СТО, а также другие документы, содержащие результаты моделирования и измерения.

2.2. Разработанное программное обеспечение соответствует требованиям служебного задания и прикладного программного обеспечения

Комиссия постановила:

- Принять работы в целом;
- Ввести в опытную эксплуатацию на кафедре МиСА;
- Установить срок опытной эксплуатации продукта до 30.06.2015.
- Установить срок полезного использования 5 лет.

Комиссия считает целесообразным:

- Разработку интеллектуальных систем управления сложными техническими и технологическими объектами на базе разработанной среды многоуровневого компьютерного моделирования

Члены комиссии:

 / А.А. Шелупанов

 / Н.Д. Малютин

 / В.И. Авдзейко

 / П.Е. Троян

 / Д.Д. Зыков

АКТ
апробации имитационной динамической модели
«Виртуальный промысел»

Настоящий акт составлен в том, что имитационная динамическая модель «Виртуальный промысел» для определения параметров установки низкотемпературной сепарации газа, позволяющая производить численные эксперименты по установлению откликов параметров на управляющие и нештатные воздействия, а также рассмотреть работу установки в режиме реального времени, опробована в ОАО «Томскгазпром». «Виртуальный промысел» разработан сотрудниками кафедры Химической технологии топлива и химической кибернетики национального исследовательского Томского политехнического университета.

Реализация продукта представлена в виде компьютерной программы с использованием клиент-серверной технологии и выполнена совместно с сотрудниками кафедры Моделирования и системного анализа Томского государственного университета систем управления и радиоэлектроники.

На серверной части осуществляется расчёт установки, сформированной из выбранных моделей аппаратов, соединённых между собой в соответствии с технологической схемой, с передачей его результатов подключенным к серверу клиентам. Серверная часть программы представлена в виде расчетного комплекса, в котором реализованы математические модели технологических объектов предприятий нефтегазовой промышленности. Клиентская часть реализована в виде автоматического рабочего места оператора, максимально приближенного к виду реального места оператора на производстве.

При создании клиентской части использована система виртуальных инструментов и приборов (СВИП). Работа клиентских частей происходит на базе системы «MARS-Engine». Существует возможность отображения и контроля дополнительных параметров, позволяющая анализировать работу установки.

Авторские права на модели элементов установки принадлежат сотрудникам кафедры химической технологии топлива и химической кибернетики Национального исследовательского Томского политехнического университета (Писарев М.О., Долганов И.М.).

Система «MARS-Engine», на базе которой реализованы клиентские части, разработана сотрудниками кафедры моделирования и системного анализа Томского государственного университета систем управления и радиоэлектроники при непосредственном участии доцента, к.т.н. Ганджи Т.В.

Данная модель может быть широко использована при выполнении исследований работоспособности и безопасности технологических установок в рамках методик HAZOP/HAZID.

РЕЗУЛЬТАТЫ

Результатами, апробированными в ОАО «Томскгазпром» и включенными в разработанную модель, предназначенную для анализа работы установки подготовки газа и подготовки операторов автоматизированных систем управления технологическими объектами предприятий нефтегазовой промышленности, являются:

1. Расчетный комплекс, включающий модели технологических объектов предприятий нефтегазовой промышленности и разработанный научным коллективом кафедры химической технологии топлива и химической кибернетики (ХТТХК)

Национального исследовательского Томского политехнического университета (Писарев М.О., Долганов И.М.);

2. Автоматическое рабочее место оператора, позволяющее в режиме реального времени отслеживать изменение параметров системы и вносить управляющие воздействия (Писарев М.О., Долганов И.М.);

3. Приложение «MARS-Engine», предназначенное для использования виртуальных инструментов и приборов, к классу которых относятся клиентские приложения имитационной динамической модели, осуществляющие сетевое взаимодействие с математическими моделями технологических объектов, анализ которых осуществляется на сервере. Данное приложение было реализовано доцентом Томского государственного университета систем управления и радиоэлектроники Ганджой Т.В.

ЗАКЛЮЧЕНИЕ

Использование имитационной динамической модели в расчетах на устойчивость работы установки низкотемпературной сепарации позволит:

- 1) Давать оценку о принимаемых проектных решениях при создании и модернизации установок низкотемпературной сепарации;
- 2) Принимать во внимание динамические особенности работы установки низкотемпературной сепарации при составлении правил и ограничений, включаемых в регламент эксплуатации установки;
- 3) Выявлять оптимальные параметры работы установки низкотемпературной сепарации с учетом установления стационарного режима.

Со стороны ОАО «Томскгазпром»

Главный специалист отдела подготовки, переработки и сдачи УВС, к.т.н.

Станкевич В.С.

Подпись заверяю:

Начальник Управления по кадровой политике ОАО «Томскгазпром»



Козырева Г.С.

Со стороны ТПУ

Директор ИКИ, аспирант

Писарев М.О.

Доц. каф. ХТТХК, к.т.н.

Долганов И.М.

Со стороны ТУСУР

Зав. каф. МиСА, д.т.н., проф.

Дмитриев В.М.

Доц. каф. МиСА, к.т.н.

Ганджа Т.В.



Ministry of Education and Science of the Russian Federation
Federal State Autonomous Educational Institution of Higher Education
"National Research Tomsk Polytechnic University" (TPU)
30, Lenin ave., Tomsk, 634050, Russia
Tel. +7-3822-606333, +7-3822-701779,
Fax +7-3822-563865, e-mail: tpu@tpu.ru, tpu.ru
OKPO (National Classification of Enterprises and Organizations):
02069303,
Company Number: 1027000890168,
VAT / KPP (Code of Reason for Registration)
7018007264/701701001, BIC 046902001

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский
Томский политехнический университет» (ТПУ)
Ленина, пр., д. 30, г. Томск, 634050, Россия
тел.: +7-3822-606333, +7-3822-701779,
факс +7-3822-563865, e-mail: tpu@tpu.ru, tpu.ru
ОКПО 02069303, ОГРН 1027000890168,
ИНН/КПП 7018007264/701701001, БИК 046902001

АКТ

«04» 09 2017 № б/н

г. Томск

ОБ ИСПОЛЬЗОВАНИИ РЕЗУЛЬТАТОВ ДИССЕРТАЦИИ ГАНДЖИ ТАРАСА ВИКТОРОВИЧА

Составлен комиссией:

Председатель - /зав. кафедрой ХТТ, к.т.н./ Е.М. Юрьев

Члены комиссии: /ассистент каф. ХТТ, к.т.н./ М.В. Киргина
/ассистент каф. ХТТ/ Е.С. Хлебникова

составили настоящий акт о том, что комплекс программ «Среда моделирования химико-технологических систем», созданный на базе развитого метода компонентных цепей, наряду с другими комплексами программ компьютерного моделирования химико-технологических систем может быть включен в процесс обучения студентов кафедры химической технологии топлива и химической кибернетики Национального исследовательского Томского политехнического университета. На основе его возможностей в области компьютерного моделирования управляемых химико-технологических систем разработаны практические и лабораторные работы, апробированные на аудиторных занятиях по дисциплинам «Газохимия» и «Химическая технология нефти и газа».

Отличительными особенностями разработанного комплекса программ являются:

– возможность проведения одно- и многовариантного анализа над химико-технологическими системами предприятий газовой промышленности, для анализа которых применяются компьютерные модели управляемых химико-технологических систем с неоднородными векторными потоками в связях, а также решение задач параметрической оптимизации на основе безградиентных и градиентных методов;

– реализованные блоки обработки данных, включающие численные методы определения точки росы и точки кипения газа при различных давлениях в исследуемых аппаратах химической технологии;

– разработанные компоненты логического уровня, обеспечивающие построение функциональных моделей устройств управления системами транспорта и переработки многокомпонентных газовых потоков, а также возможность отладки алгоритмов их функционирования;

– широкий набор визуальных компонентов, обеспечивающий построение различных панелей визуализации результатов вычислительных экспериментов над химико-технологическими системами и управления параметрами компонентов исследуемых моделей в интерактивном режиме;

– наличие интерактивных математических и математико-алгоритмических панелей, обеспечивающих исследование новых математических моделей компонентов и проведение над ними вычислительных экспериментов исследовательского характера;

– включение генератора моделей компонентов в комплекс программ, обеспечивающего создание моделей новых компонентов различными пользователями и пополнение ими библиотеки моделей компонентов.

Заключение

Все вышесказанное характеризует комплекс программ моделирования химико-технологических систем как законченное программное обеспечение, предназначенное для обучения химиков-технологов газовой промышленности современным подходам к автоматизации решения задач научно-исследовательского, производственного и учебного характера.

Председатель комиссии

Е.М. Юрьев

Члены комиссии

М.В. Киргина

Е.С. Хлебникова

Подписи председателя и членов комиссии заверяю:

Ученый секретарь ректората ЦТУ



О.А. Ананьева



Министерство образования и науки Российской Федерации
 Обособленное подразделение
**«Научно-исследовательский институт автоматики и электромеханики
 Томского государственного университета систем управления и
 радиоэлектроники»
 («НИИ АЭМ ТУСУР»)**

634034, г. Томск, ул. Белинского 53, НИИ АЭМ ТУСУР,
 тел: 8-(3822) 55-61-96, факс: 55-77-03, E-mail: aem@tusur.ru
 ОКПО 02070246, ОГРН 1027000867068, ИНН 7021000043, КПП 701745001

«Утверждаю»

Зам. директора по научной
 работе, к.т.н.

Целебровский И.В.
 «12» мая 2017г.

СПРАВКА

об использовании теоретических и практических
 разработок Ганджи Тараса Викторовича,
 представленных в его докторской диссертации
**«Метод многоуровневого компьютерного моделирования и
 комплекс программ автоматизированного эксперимента
 над сложными техническими управляемыми системами»**

При выполнении х/д № 18/914-С1 от 16.04.2007г. «Создание автоматизированной системы управления технологическими процессами системы оборотного снабжения метанолом УКПГ Мыльджинского газоконденсатного месторождения ОАО «Томскгазпром» были использованы следующие наработки:

1. Сформирован набор моделей компонентов системы, включающий в себя источник многокомпонентного вещественного потока, отрезки трубопроводов, управляемые клапана, емкости хранения газа, нагнетающие насосы, смеситель, сепаратор, абсорбер и т.д.;
2. Произведен многовариантный анализ модели автоматизированной системы управления оборотного снабжения метанолом. Результатами анализа явились найденные диапазоны изменения расхода метанола на входе абсорбера, обеспечивающие эффективное его использование при осушке природного газа.

При выполнении х/д 18-9/07Фр. от 12.09.2007г. «Разработка и изготовление стендов для учебно-научных лабораторий» с помощью компонентов языка виртуальных инструментов и приборов разработан макет стенда автоматизированной системы управления системой оборотного снабжения метанолом.

Зав.отд.18 НИИ АЭМ ТУСУР,
 К.Т.Н.

Ю.А.Андреев





ЗАКРЫТОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО
НАУЧНО-ПРОИЗВОДСТВЕННАЯ ФИРМА

СибНефтеКарт

Юридический адрес: Россия, г. Томск, ул. Нахимова, 8, строение 1, оф. 4.
Почтовый адрес: 634034, г. Томск, ул. Нахимова, 8, строение 1, офис 4
тел.: (382 2) 41-65-11, (3822) 41-69-11
E-mail: sncard@sncard.ru

А К Т

О внедрении (о возможности использования) разработок

Ганджи Тараса Викторовича,

представленных в его докторской диссертации

**«Метод многоуровневого компьютерного моделирования и комплекс программ
автоматизированного эксперимента над сложными техническими
управляемыми системами»**

Результаты диссертационного исследования Ганджи Тараса Викторовича были использованы при разработке ЗАО НПФ "Сибнефтекарт" программного обеспечения автоматизированных систем управления автозаправочными комплексами и нефтебазами, в части:

- 1) анализа процессов хранения, реализации и учета нефтепродуктов;
- 2) разработки компонентов языка моделирования, построения и отладки алгоритмов управления технологическими объектами;
- 3) визуализации результатов вычислительных экспериментов с элементами интерактивного управления параметрами.

Представленный в докторской диссертации Ганджи Т.В. комплекс программ использовался при:

- 1) построении компьютерной модели процессов, протекающих в резервуарах для хранения нефтепродуктов. Эта модель позволяет прогнозировать изменения характеристик нефтепродуктов в емкостях при изменении их объема и параметров окружающей среды;
- 2) разработке имитационной модели прогнозирования процесса бесперебойного снабжения автозаправочных станций топливом различных марок в зависимости от его потребления;
- 3) разработке многоуровневой компьютерной модели систем управления топливораздаточными колонками, наливными пунктами и систем измерения параметров нефтепродуктов в горизонтальных резервуарах.

Начальник клиентского
отдела, д-р техн. наук



А.Я. Клименко



**ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
«ТОМСКНЕФТЕПЕРЕРАБОТКА»
(ООО «ТНП»)**

Юридический адрес: 634530, Томская область, Томский р-он, с. Семилужки, ул. Нефтепровод, 2
Почтовый адрес: 634041, г. Томск, ул. Карташова, 23А
Тел: 8 (3822) 43 40 40, факс: 43 07 06, www.tomnpz.ru, e-mail: mail@tomnpz.ru
ИНН 7017135873, КПП 701401001, ОГРН 1067017007144, ОКПО 79198169

Исх. № 3943 от 30.12. 2013 г.
На № _____ от _____ 2013 г.

В диссертационный совет Д 212.268.02 при
ФГБОУ ВПО «Томский государственный
университет систем управления и
радиоэлектроники»

634050, г. Томск, пр. Ленина, д. 40
8 (3822) 510-530
office@tusur.ru

АКТ О ВНЕДРЕНИИ

Результатов диссертационной работы Ганджи Тараса Викторовича
«Методология построения интеллектуальных SCADA-систем с компьютерной моделью в
контуре управления технологическими процессами предприятий нефтегазовой
промышленности»

Настоящим подтверждается, что результаты диссертационного исследования Т.В. Ганджи «Методология построения интеллектуальных SCADA-систем с компьютерной моделью в контуре управления технологическими процессами предприятий нефтегазовой промышленности» применимы к технологическим процессам переработки нефти на нефтеперерабатывающем заводе Общество с ограниченной ответственностью «Томскнефтепереработка» в виде:

- многоуровневой компьютерной модели систем управления, позволяющей на основе математической модели объекта управления автоматически определять экономически эффективные и экологически безопасные режимы работы установок переработки нефти УПН-100А,Б с выпуском качественных нефтепродуктов (бензин газовый стабильный, ГАЗОЙЛЬ ИСО-F-D 1.2.3.4, топочный мазут 100);
- целевых функций и реализованных в виде компонентов-методов параметрической оптимизации, предназначенных для определения требуемых режимов функционирования объектов управления;
- компонентов взаимодействия многоуровневой компьютерной модели с реальным объектом управления, с помощью которых модель получает данные измерения наблюдаемых характеристик реального объекта и позволяет осуществлять управление исполнительными устройствами (двигателями, насосами, подогревателями);
- компонентов связи с информационными источниками данных, представленными базами данных технологических параметров и геоинформационными системами экологических характеристик объектов окружающей природной среды.

Предложенные автором методология и архитектура многоуровневых компьютерных моделей интеллектуальных SCADA-систем, включающих компьютерную модель объекта управления, возможно внедрить в контур управления технологическим процессом переработки нефти и получения основных продуктов переработки. Разработанная методология интеллектуального управления позволяет учитывать текущие характеристики окружающей среды наравне с использованием данных измерений технологических характеристик. На их основе с использованием аппарата компьютерного моделирования и параметрической оптимизации, реализуемых в единой среде многоуровневого компьютерного моделирования, решаются задачи определения,

установления и поддержания экологически безопасных и экономически эффективных режимов функционирования технологического оборудования с выводом данных измерений, результатов моделирования и всей необходимой технологической информации для пользователя на графический монитор. Помимо этого он также содержит органы формирования и изменения задающих воздействий.

Внедрение разработанной интеллектуальной SCADA-системы позволяет предприятию выполнять требования ст. 10 Федерального закона N 116-ФЗ "О промышленной безопасности опасных производственных объектов" в части создания тренажеров.

Разработанное программное обеспечение направлено на эффективное и безопасное управление технологическими процессами, протекающими в производственных объектах предприятий нефтегазовой промышленности. Также предложенные принципы могут применяться для контроля и управления технологическими процессами отраслей химической промышленности, взаимодействующих с объектами окружающей природной среды

Материалы докторского исследования Т.В. Ганджи имеют большую практическую и теоретическую значимость.

Первый заместитель генерального директора



Н.А. Шелехова



МИНОБРНАУКИ РОССИИ

Федеральное государственное
бюджетное образовательное учреждение
высшего профессионального образования
«Оренбургский государственный
университет»
(ОГУ)

УТВЕРЖДАЮ

Проректор по учебной работе

А.Д. Проскурин

20__ г.



АКТ

№

г. Оренбург

О внедрении результатов диссертационной работы
в учебный процесс кафедры ПЭИИТ ЭЭФ ОГУ

Комиссия в составе:

председатель комиссии: декан электроэнергетического факультета, кандидат технических наук, доцент Вакулюк Владимир Максимович;

члены комиссии: заведующий кафедрой промышленной электроники и информационно-измерительной техники, кандидат технических наук, доцент Художков Олег Викторович, профессор кафедры промышленной электроники и информационно-измерительной техники доктор технических наук, профессор Булатов Виталий Николаевич,

составили настоящий акт о том, что результаты диссертационной работы Ганджи Тараса Викторовича, представленной на соискание степени доктора технических наук, внедрены в учебный процесс кафедры *промышленной электроники и информационно-измерительной техники Оренбургского государственного университета* в виде реально-виртуальной лаборатории по дисциплине «Системы автоматического регулирования и управления». Она создана на базе среды многоуровневого компьютерного моделирования, интегрированной с аппаратно-программным комплексом «Лабораторное автоматизированное рабочее место студента» (ЛАРМ).

Среда компьютерного моделирования с использованием компонентного подхода позволяет формировать многоуровневые компьютерные модели для проведения экспериментов на трех взаимосвязанных уровнях:

– на *объектном уровне* составляется схема взаимодействий многоуровневой компьютерной модели с драйвером ЛАРМ, компоненты которой предназначены для формирования и передачи команд управления аппаратно-программным комплексом, а также получения от него значений измеряемых характеристик исследуемого объекта в реальном масштабе времени.

– на *логическом уровне* из компонентов, представляющих математические и алгоритмические операции, разрабатывается модель алгоритма проведения эксперимента, включающая в себя компоненты формирования команд управления комплексом ЛАРМ, блоки обработки полученных результатов и их подготовку к визуа-

лизации, а также компоненты автоматизированного документирования результатов исследования;

– на *визуальном уровне* из графических компонентов представлена панель визуализации результатов измерения и изменения параметров источников и генераторов сигналов, входящих в комплекс ЛАРМ.

Отличительной особенностью представленной реально-виртуальной лаборатории является возможность математической обработки результатов и возможность построения разнообразных экспериментов. Для исследования систем автоматического управления с помощью аппаратно-программного комплекса ЛАРМ разработаны следующие блоки математической обработки результатов:

– блоки построения частотных характеристик, позволяющие определять амплитуду и фазу сигнала на выходе системы или звена относительно входного синусоидального сигнала;

– блоки определения показателей качества работы САУ в статическом и динамическом режиме. Для исследования статических режимов САУ разработан блок определения установившегося значения сигнала (ошибки), а для исследования переходных процессов блоки определения времени переходного процесса, перерегулирования, количества и частоты свободных колебаний.

Кроме исследования реальных систем автоматического управления, представленных реальными электрическими цепями, созданная в рамках диссертационной работы среда многоуровневого компьютерного моделирования позволяет исследовать модели систем автоматического управления, которые формируются на объектном уровне многоуровневой модели из соответствующих компонентов, представляющих собой структурные звенья систем автоматического управления. Основным достоинством среды многоуровневого компьютерного моделирования является то, что для исследования реальных объектов и их компьютерных моделей используются одни и те же наборы компонентов логического уровня. Для визуализации данных измерения и результатов моделирования используются унифицированные визуальные компоненты, что значительно расширяет возможности проведения компьютерных экспериментов и сокращает время на изучение принципов работы с приборами.

Разработанное программное обеспечение в совокупности с аппаратно-программным комплексом ЛАРМ увеличивает возможности экспериментирования, позволяет существенно уменьшить время проведения каждого исследования, а следовательно, позволяет за одно и то же выделенное рабочей программой время под лабораторные работы выполнить большее количество экспериментов.

Все выше сказанное повышает уровень подготовки бакалавров и магистров по направлению подготовки электроника, радиотехника и системы связи.

Председатель комиссии
декан ЭЭФ



В.М. Вакулюк

Члены комиссии:
заведующий кафедрой ПЭиИИТ
профессор кафедры ПЭиИИТ



О.В. Худорожков
В.Н. Булатов



ОАО «ГАЗПРОМ»

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ГАЗПРОМ ДОБЫЧА ЯМБУРГ»

(ООО «Газпром добыча Ямбург»)

Ф И Л И А Л

УПРАВЛЕНИЕ АВТОМАТИЗАЦИИ И МЕТРОЛОГИЧЕСКОГО ОБЕСПЕЧЕНИЯ

« ____ » _____ 20 ____ г.

№ _____

А К Т О В Н Е Д Р Е Н И И

результатов диссертации на соискание степени доктора технических наук Ганджи Тараса Викторовича

Настоящим подтверждается, что результаты диссертационного исследования Т.В. Ганджи «Метод и комплекс программ многоуровневого компьютерного моделирования систем интеллектуального управления сложными технологическими объектами» использованы при разработке алгоритма оптимального управления приборами освещения на производственной площадке УППГ-4А Ямбургского газоконденсатного месторождения ООО «Газпром добыча Ямбург».

Разработанная автором исследования многоуровневая компьютерная модель систем интеллектуального управления сложными технологическими объектами позволяет:

- сформировать компьютерную модель объекта управления с неоднородными векторными связями между ее компонентами и повысить адекватность ее анализа для решения задач интеллектуального управления технологическим оборудованием предприятий нефтегазовой промышленности;

- разработать в графическом схемотехническом виде алгоритм решения задачи интеллектуального управления технологическим оборудованием, в ходе которого неоднократно используется компьютерная модель оборудования при варьировании параметров входящих в нее компонентов;

- создать из визуальных компонентов стенд отображения результатов моделирования и измерения характеристик реального объекта с возможностями интерактивного управления параметрами его модели и управляющими воздействиями на реальный объект.

Гибкость разработанной автором компьютерной модели и ее адаптация к физически различным подсистемам сложных технических систем технологических объектов обеспечивается за счет использования неоднородных векторных связей. Имея универсальный характер, алгоритмы формирования и анализа компьютерных моделей сложных технологических объектов со связями данного класса позволяют анализировать

мультифизические многокомпонентные потоки в сложных технологических процессах.

Разработанный новый метод формирования многоуровневых компьютерных моделей систем интеллектуального управления представляет собой теоретическую значимость выполненного диссертационного исследования, а возможность определения и установки оптимальных режимов функционирования оборудования представляет большую практическую значимость для предприятий нефтегазовой промышленности.

**Начальник Управления автоматизации
и метрологического обеспечения
ООО «Газпром добыча Ямбург»**



С.И. Гункин

Министерство образования и науки
Российской Федерации
федеральное государственное автономное
образовательное учреждение
высшего профессионального образования

**«СЕВЕРНЫЙ (АРКТИЧЕСКИЙ)
ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
ИМЕНИ М.В. ЛОМОНОСОВА»
(САФУ имени М.В. Ломоносова)**

набережная Северной Двины, д. 17,
г. Архангельск, Россия, 163002
http://www.narfu.ru
e-mail: public@narfu.ru
тел./факс: 8(8182) 28-76-14
телефон: 8(8182) 21-89-20

28.04.2015 № ДП.1-04/48

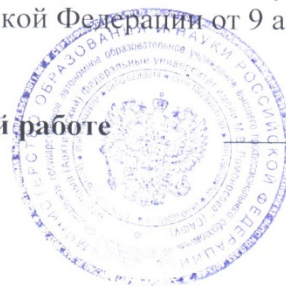
На № _____ от _____

**В диссертационный
совет Д212.268.02
Томского государственного
университета систем управления
и радиоэлектроники**

СПРАВКА ОБ ИСПОЛЬЗОВАНИИ РЕЗУЛЬТАТОВ ДИССЕРТАЦИОННОГО ИССЛЕДОВАНИЯ

выдана Ганджа Тарасу Викторовичу, кандидату технических наук, в том, что результаты его диссертационного исследования в виде среды многоуровневого компьютерного моделирования систем интеллектуального управления сложными технологическими объектами, вошедшие в материалы диссертации на соискание степени доктора технических наук, использованы при выполнении научного проекта «Разработка технологии выявления кризисных ситуаций и определения путей их разрешения. Создание модели опережающего стратегического управления», выполняемого Федеральным государственным автономным образовательным учреждением высшего профессионального образования «Северный (Арктический) федеральный университет имени М.В. Ломоносова» (САФУ имени М.В. Ломоносова) в кооперации с обществом с ограниченной ответственностью «ИБС Экспертиза» при субсидировании работ проектом Министерством образования и науки Российской Федерации, в соответствии с Правилами предоставления субсидий на государственную поддержку кооперации российских высших учебных заведений, государственных научных учреждений и организаций, реализующих комплексные проекты по созданию высокотехнологичного производства, утвержденных постановлением Правительства Российской Федерации от 9 апреля 2010 г. № 218.

Проректор по научной работе



Б.Ю. Филиппов

Б.Ю. Филиппов

Степанова В.В.
(8182) 21-61-00 доб. 13-83

**Приложение 18. Свидетельства о государственной
регистрации программ для ЭВМ**

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2013612164

Среда компьютерного моделирования МАРС

Правообладатель(ли): **Общество с ограниченной ответственностью
«Научно-производственное предприятие «РЕВИКОМ» (RU)**

Автор(ы): **Дмитриев Вячеслав Михайлович (RU),
Зайченко Татьяна Николаевна (RU), Шутенков Александр
Васильевич (RU), Ганджа Тарас Викторович (RU),
Ганджа Василий Викторович (RU)**

Заявка № **2012661599**

Дата поступления **25 декабря 2012 г.**

Зарегистрировано в Реестре программ для ЭВМ

15 февраля 2013 г.

*Руководитель Федеральной службы
по интеллектуальной собственности*

Б.П. Симонов



РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2014662641

Генератор моделей компонентов среды
моделирования MARCПравообладатель: *Ганджа Тарас Викторович (RU)*Автор: *Ганджа Тарас Викторович (RU)*

Заявка № 2014660400

Дата поступления 13 октября 2014 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 04 декабря 2014 г.

Врио руководителя Федеральной службы
по интеллектуальной собственности

Л.Л. Кирий



РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2009612036

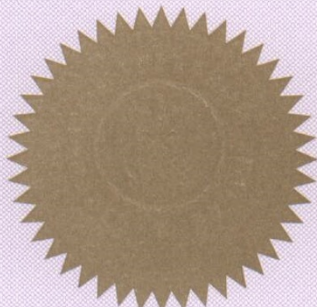
Программное обеспечение
«Расчетно-моделирующая среда»Правообладатель(ли): *Общество с ограниченной ответственностью
«АСОТ» (RU)*Автор(ы): *Ганджа Тарас Викторович,
Зайченко Татьяна Николаевна, Шутенков Александр Васильевич,
Кураколов Алексей Николаевич (RU)*

Заявка № 2009611461

Дата поступления 7 апреля 2009 г.

Зарегистрировано в Реестре программ для ЭВМ
22 апреля 2009 г.*Руководитель Федеральной службы по интеллектуальной
собственности, патентам и товарным знакам*

Б.П. Симонов



РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2014617748

Библиотека моделей компонентов системы
автоматизированного документированияПравообладатели: *Панов Сергей Аркадьевич (RU), Ганджа Тарас
Викторович (RU)*Авторы: *Панов Сергей Аркадьевич (RU),
Ганджа Тарас Викторович (RU)*

Заявка № 2014615349

Дата поступления 02 июня 2014 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 01 августа 2014 г.

Руководитель Федеральной службы
по интеллектуальной собственности

Б.П. Симонов



РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2015663196

Библиотека моделей компонентов бизнес-процессов для
среды моделирования MAPCПравообладатели: *Ганджа Тарас Викторович (RU), Панов Сергей
Аркадьевич (RU), Григорьева Татьяна Евгеньевна (RU)*Авторы: *Ганджа Тарас Викторович (RU), Панов Сергей Аркадьевич
(RU), Григорьева Татьяна Евгеньевна (RU)*

Заявка № 2015660407

Дата поступления 28 октября 2015 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 14 декабря 2015 г.

Руководитель Федеральной службы
по интеллектуальной собственности
 Г.П. Ивлиев


РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2014617663

**Библиотека моделей компонентов «Телеком» для среды
моделирования MARC**

Правообладатели: **Буданов Алексей Николаевич (RU), Ганджа
Василий Викторович (RU), Ганджа Тарас Викторович (RU)**

Авторы: **Буданов Алексей Николаевич (RU), Ганджа Василий
Викторович (RU), Ганджа Тарас Викторович (RU)**

Заявка № 2014615619

Дата поступления 11 июня 2014 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 30 июля 2014 г.

Руководитель Федеральной службы
по интеллектуальной собственности

A handwritten signature in black ink, appearing to read 'B.P. Simonov', is written over a light blue background.

Б.П. Симонов



РОССИЙСКАЯ ФЕДЕРАЦИЯ

**СВИДЕТЕЛЬСТВО**

о государственной регистрации программы для ЭВМ

№ 2010617248**Автоматизированный учебно-методический комплекс**Правообладатель(ли): *Общество с ограниченной ответственностью «АСОТ» (RU)*Автор(ы): *Ганджа Тарас Викторович,
Ганджа Василий Викторович, Ильичева Яна Юрьевна,
Кураколов Алексей Николаевич, Онищук Максим Владимирович,
Филиппов Алексей Юрьевич (RU)*

Заявка № 2010614890

Дата поступления 9 августа 2010 г.

Зарегистрировано в Реестре программ для ЭВМ
29 октября 2010 г.Руководитель Федеральной службы по интеллектуальной
собственности, патентам и товарным знакам

Б.П. Симонов



ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННЫЙ КООРДИНАЦИОННЫЙ ЦЕНТР ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
ОТРАСЛЕВОЙ ФОНД АЛГОРИТМОВ И ПРОГРАММ

СВИДЕТЕЛЬСТВО ОБ ОТРАСЛЕВОЙ
РЕГИСТРАЦИИ РАЗРАБОТКИ

№ 4584

Настоящее свидетельство выдано на разработку:

Универсальное вычислительное ядро среды
моделирования МАРС

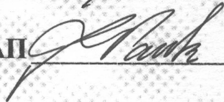
зарегистрированную в Отраслевом фонде алгоритмов и программ.

Дата регистрации: **08 апреля 2005 года**

Авторы: **Шутенков А.В., Ганджа Т.В.**

Организация-разработчик: **Томский государственный университет**
систем управления и радиоэлектроники

Директор  **Е.Г. Калинин**

Руководитель ОФАП  **А.И. Галкина**

Дата выдачи 29.04.2005



ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ФГНУ «ГОСУДАРСТВЕННЫЙ КООРДИНАЦИОННЫЙ ЦЕНТР ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

ОТРАСЛЕВОЙ ФОНД АЛГОРИТМОВ И ПРОГРАММ

**СВИДЕТЕЛЬСТВО ОБ ОТРАСЛЕВОЙ
РЕГИСТРАЦИИ РАЗРАБОТКИ**

№ 8902

Настоящее свидетельство выдано на разработку:

**Программная система
«Среда моделирования МАРС»**

зарегистрированную в Отраслевом фонде алгоритмов и программ.

Дата регистрации: **09 августа 2007 года**

Авторы: **Дмитриев В.М., Ганджа Т.В., Зайченко Т.Н., Кураколов А.Н.**

Организация-разработчик: **Томский государственный университет
систем управления и радиоэлектроники**



Директор  **Е.Г. Калинин**

Руководитель ОФАП  **А.И. Галкина**

Дата выдачи **30.08.2007**



ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ФГНУ «ГОСУДАРСТВЕННЫЙ КООРДИНАЦИОННЫЙ ЦЕНТР ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

ОТРАСЛЕВОЙ ФОНД АЛГОРИТМОВ И ПРОГРАММ

**СВИДЕТЕЛЬСТВО ОБ ОТРАСЛЕВОЙ
РЕГИСТРАЦИИ РАЗРАБОТКИ**

№ 10232

Настоящее свидетельство выдано на разработку:

**Библиотека моделей компонентов среды
моделирования МАРС для анализа и синтеза
характеристик газопромысловых объектов**

зарегистрированную в Отраслевом фонде алгоритмов и программ.

Дата регистрации: 25 марта 2008 года

Авторы: Дмитриев В.М.; Ганджа Т.В., Клёпак И.Я.

Организация-разработчик: **ФГУ ВПО Томский государственный
университет систем управления и
радиоэлектроники**

Директор

Е.Г. Калинин
Е.Г. Калинин

Руководитель ОФАП

А.И. Галкина
А.И. Галкина

Дата выдачи 01.04.2008



ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ФГНУ «ГОСУДАРСТВЕННЫЙ КООРДИНАЦИОННЫЙ ЦЕНТР ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

ОТРАСЛЕВОЙ ФОНД АЛГОРИТМОВ И ПРОГРАММ

**СВИДЕТЕЛЬСТВО ОБ ОТРАСЛЕВОЙ
РЕГИСТРАЦИИ РАЗРАБОТКИ**

№ 10233

Настоящее свидетельство выдано на разработку:

**Библиотека моделей компонентов среды
моделирования МАРС для автоматизированного решения
задач параметрического синтеза технических объектов**
зарегистрированную в Отраслевом фонде алгоритмов и программ.

Дата регистрации: 25 марта 2008 года

Авторы: Ганджа Т.В., Клепак И.Я.

Организация-разработчик: **ГОУ ВПО Томский государственный
университет систем управления и
радиоэлектроники**

Директор

Е.Г. Калинин

Руководитель ОФАН

А.И. Галкина

Дата выдачи 01.04.2008