

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

Одесский национальный политехнический университет

На правах рукописи

Нестерюк Александр Геннадиевич

УДК: 004.942

**ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ МОДЕЛИРОВАНИЯ И
АНАЛИЗА ДИСКРЕТНО-НЕПРЕРЫВНЫХ
АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ**

05.13.06 – Информационные технологии

Диссертация

на соискание научной степени

кандидата технических наук

Научный руководитель

Денисенко Владимир Анатольевич

доктор технических наук, профессор

Одесса, 2016

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ	5
ВВЕДЕНИЕ	6
РАЗДЕЛ 1 АНАЛИЗ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ МОДЕЛИРОВАНИЯ И ИССЛЕДОВАНИЯ СВОЙСТВ СЛОЖНЫХ СИСТЕМ В ДИСКРЕТНО-НЕПРЕРЫВНОМ ПРОСТРАНСТВЕ	12
1.1. Основные задачи и принципы проектирования систем управления гибридными техническими объектами	12
1.2. Анализ математических схем представления ДН систем.....	17
1.2.1. Интегрированные системы на основе модели конечного автомата	18
1.2.2. Интегрированные системы на основе модели сети Петри	21
1.3. Методы определения управляемости (достижимости) динамических систем	28
1.4. Визуализированные средства моделирования и анализа сложных технических автоматизированных систем.....	31
1.4.1. Средства моделирования систем, разработанных на основе модели конечного автомата	31
1.4.2. Средства моделирования систем, разработанных на основе модели сети Петри.....	32
Выводы	37
РАЗДЕЛ 2 РАЗРАБОТКА ИНФОРМАЦИОННОЙ МОДЕЛИ СЛОЖНОГО ТЕХНИЧЕСКОГО ОБЪЕКТА	40
2.1. Разработка модифицированного сигнального графа.....	41
2.2. Математическая модель модифицированного сигнального графа.....	46

2.3. Расширение теоретико-множественного представления ДНАСУ.....	53
2.3.1. Переходы – результаты наложения элементарных сетей.....	53
2.3.2. Переходы – результат наложения фрагментов сетей.....	56
2.4. Формализованное представление расширенной ДНАСУ.....	60
Выводы.....	71

РАЗДЕЛ 3 ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ ОПРЕДЕЛЕНИЯ

ДОСТИЖИМОСТИ.....	72
3.1 Задача достижимости с позиций теории управления.....	72
3.2 Определение критериев достижимости в структурных состояниях НС части ДНАСУ.....	79
3.3 Общая постановка задачи достижимости ДНАСУ и пути ее решения.....	86
3.4 Особенности построения моделей.....	92
3.5 Аналитический подход к определению достижимости ДНАСУ.....	94
3.6 Определение правил редукции непрерывно-событийной части ДНАСУ.....	95
3.7 Определение правил редукции дискретной части ДНАСУ.....	99
Выводы.....	101

РАЗДЕЛ 4 РАЗРАБОТКА ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ МОДЕЛИРОВАНИЯ И АНАЛИЗА СВОЙСТВА ДОСТИЖИМОСТИ

ДНАСУ.....	102
4.1 Базовая информационная технология моделирования ДНАСУ.....	102
4.2 Базовая информационная технология анализа достижимости ДНАСУ.....	104
4.3 Информационная технология общего назначения DCNET.....	111
4.3.1 Общая структура ИТ моделирования и анализа достижимости ДНАСУ.....	111
4.3.2 Требования к среде моделирования.....	113
4.3.3 Разработка модулей ИТ моделирования и анализа достижимости ДНАСУ.....	115
4.3.4 Разработка архитектуры программной среды.....	120
4.3.5 Требования к программным и аппаратным ресурсам.....	127

	4
4.3.6 Реализация пользовательского интерфейса	127
4.3.7 Работа с программным комплексом.....	134
4.4 Моделирование и анализ свойства достижимости АСУ.....	136
Выводы	142
ЗАКЛЮЧЕНИЕ	143
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	146
ПРИЛОЖЕНИЕ А ПРИМЕР ИССЛЕДОВАНИЯ ДНАСУ	158
ПРИЛОЖЕНИЕ Б КРАТКОЕ СПРАВОЧНОЕ РУКОВОДСТВО ПРОГРАММНОЙ СРЕДЫ DCNET.....	169
Б.1 Интерфейс программы DC-Net.....	169
Б.2 Выбор режимов работы	174
Б.3 Главное меню.....	175
Б.4 Панель инструментов.....	182
Б.5 Поиск элемента.....	183
Б.6 Редактирование схемы.....	185
Б.7 Режим установки элементов	186
Б.8 Режим задания связей	190
Б.9 Режим работы с группой выделенных элементов	191
Б.10 Ввод и редактирование информации о параметрах элемента.	194
Б.11 Режим коррекции разметки сети	206
Б.12 Моделирование.....	208
Б.13 Анализ свойства достижимости	210
Б.14 Опции памяти	212
ПРИЛОЖЕНИЕ В АКТЫ ВНЕДРЕНИЯ РЕЗУЛЬТАТОВ ДИССЕРТАЦИИ	213

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

- АСУ – автоматизированная система управления;
- АСУТП – автоматизированная система управления технологическим процессом;
- БД – база данных;
- ГС – гибридная система;
- ДН АСУ – дискретно-непрерывная АСУ;
- ДН-сеть – дискретно-непрерывная сеть;
- ДН система - дискретно-непрерывная система;
- ДС часть – дискретно-событийная часть;
- ИТ – информационная технология;
- ЛДС – логико-динамическая система;
- МСГ – модифицированный сигнальный граф;
- Н/Д – непрерывно-дискретный преобразователь;
- НС часть – непрерывно-событийная часть;
- ОУсУС – объект управления с управляемой структурой;
- ПСГ – переходный граф состояний;
- САПР – система автоматического проектирования;
- САУ – система автоматического управления;
- СГ – сигнальный граф;
- СП – сеть Петри;
- СТАУ – современная теория автоматического управления;
- СТС – сложная техническая система;
- СУС – система с управляемой структурой;
- ТАУ – теория автоматического управления;
- ТК – технический комплекс;
- ТС – техническая система;
- УЛУ – устройство логического управления;

ВВЕДЕНИЕ

Актуальность темы. Автоматизация отраслей народного хозяйства требует применения АСУ двух типов, а именно, построенных по аналоговым и цифровым принципам. К преимуществам аналогового способа управления относится оперативность реакции на изменение внешних условий, а к недостаткам – сложность настройки, низкая помехозащищенность, что снижает его эффективность. В противоположность аналоговым, цифровые системы достаточно просты в настройке и более эффективны, однако они требуют больших затрат времени для проведения расчетов, что снижает оперативность. Для устранения противоречия между оперативностью и эффективностью часто используют аналого-цифровые АСУ, относящиеся к классу дискретно-непрерывных систем (ДН АСУ), в которых наиболее затратные по времени процедуры реализуют в виде аналоговых устройств. Примерами таких систем являются системы управления полетами, управления процессами в некоторых отраслях химической промышленности и другие.

При разработке ДН АСУ применяются различные методы и ИТ моделирования и анализа информационных систем для автоматизированного управления, основанные на эквивалентных непрерывных, или на эквивалентных дискретных моделях. В результате применения эквивалентных моделей и соответствующих ИТ (непрерывных или дискретных, например, таких, как MATLAB, MvStudium, VisSim) снижается качество моделирования, под которым будем понимать нормированную среднеквадратичную ошибку между реакциями объекта моделирования и модели, что влияет на эффективность разработки системы. Одной из основных задач ИТ моделирования и анализа ДН АСУ является доказательство свойства достижимости дискретно-непрерывной системы, которое проводится путем численного эксперимента или с помощью аналитических методов.

Существующие методы анализа свойства достижимости имеют существенные недостатки. Во-первых, численные методы не позволяют проводить доказательство достижимости на ранних этапах разработки АСУ. Во-вторых, для анализа ДН АСУ используют отдельно непрерывные и дискретные аналитические методы исследования свойства достижимости, что увеличивает как время анализа, так и время синтеза системы.

Для устранения этих недостатков предлагается разработать ИТ моделирования и анализа свойства достижимости на основе использования сетей Петри и ДН-сетей, что позволит повысить качество моделирования. Также для анализа свойства достижимости ДН АСУ предлагается использовать редукцию-декомпозицию ДН-сети и СГ, что позволит уменьшить сложность вычислений, и, следовательно, время анализа.

Таким образом, разработка ИТ для моделирования и анализа дискретно-непрерывной динамики средствами ДН-сетей является **актуальной**.

Связь работы с научными программами, планами, темами. Диссертационные исследования выполнены в соответствии с планами научно-исследовательских работ ОНПУ № 559-64 «Цифрові узгоджені фільтри сигналів з розширеним спектром в системах керування збирання інформації» (0107U001961), № 664-64 «Частотно-залежні компоненти цифрових систем які перебудуються» (0109U008464), а также № 54-64 «Дискретно-безперервні мережі в автоматизованих системах управління» (0111U009482).

Цель и задачи исследования. Целью диссертационной работы является повышение качества моделирования и сокращение времени анализа свойства достижимости путем разработки и исследования моделей, методов и информационной технологии моделирования и анализа дискретно-непрерывных автоматизированных систем управления на базе сетей Петри и дискретно-непрерывных сетей.

Для достижения этой цели в работе решены следующие задачи.

1. Проведен анализ прикладных областей применения дискретно-непрерывных АСУ, моделей, методов и существующих ИТ их моделирования и анализа.

2. Разработана дискретно-непрерывная модель, которая описывает дискретно-непрерывные АСУ.
3. Разработан модифицированный сигнальный граф, который представлен в формате сетей Петри.
4. Разработан метод исследования свойства достижимости дискретно-непрерывных АСУ.
5. Разработаны базовые ИТ моделирования и анализа достижимости дискретно-непрерывных АСУ на основе ДН-сетей.
6. Проведено моделирование и исследовано свойство достижимости модели пакетной установки для выпаривания раствора хлористого натрия.

Объект исследования – процесс имитационного моделирования и анализа сложных дискретно-непрерывных АСУ.

Предмет исследования – модели, методы и ИТ моделирования и анализа дискретно-непрерывных АСУ на основе ДН-сетей.

Методы исследования. При разработке моделей использованы методы теории многосвязных САР и гибридного пространства состояний, при разработке методов преобразования сигнального графа и анализа свойства достижимости – теория графов и сетей Петри, а также структурные матрицы. При разработке программного обеспечения использовались методы объектно-ориентированного программирования и теории баз данных.

Научная новизна полученных результатов исследования заключается в разработке методов и моделей для ИТ моделирования и анализа ДН АСУ. В работе получены следующие научные результаты.

1. Получила дальнейшее развитие дискретно-непрерывная модель дискретно-непрерывных автоматизированных систем управления, которая отличается наличием дополнительных компонентов: дискретно-непрерывных и стохастических переходов, что позволило повысить качество моделирования.
2. Впервые разработан метод преобразования сигнального графа объекта моделирования в модель системы, представленной сетью Петри, который отличается наличием формального соответствия между элементами

сигнального графа и элементами сети Петри, что позволило разработать метод анализа свойства достижимости.

3. Впервые разработан метод и информационная технология анализа достижимости состояния дискретно-непрерывной системы, отличающиеся использованием сигнального графа и выполнением редукции-декомпозиции сетей в дискретно-непрерывном пространстве, что обеспечивает проведение исследования достижимости дискретно-непрерывных АСУ на ранних этапах разработки системы и позволяет уменьшить время анализа.
4. Получили дальнейшее развитие метод и информационная технология моделирования дискретно-непрерывных объектов, которые базируются на разработанной дискретно-непрерывной модели и отличаются наличием этапа исследования достижимости заданных состояний, что позволяет повысить качество моделирования и сократить время моделирования и анализа.

Практическое значение полученных результатов. Практическая ценность работы заключается в том, что на основе разработанных методик, алгоритмов и программ впервые стало возможным в рамках ИТ моделировать сложные ДН системы, представленные с помощью ДН-сети, и исследовать их свойства. Результаты исследований апробированы и внедрены в учебный процесс кафедры «Компьютерные системы» Одесского национального политехнического университета, а также в НПП «ХАРТРОН-ЕНКОС» и НВКЦ «ОДИС-W», что позволило сократить время разработки на 30% и улучшить качество продукции на 6-7%.

Личный вклад соискателя заключается в развитии методов моделирования и моделей ДО систем, основанных на базе ДН-сетей, определение способов их декомпозиции с целью решения задачи достижимости, а именно: реализация операций в программной среде DCNET [19], постановка задач достижимости [33, 37], разработка алгоритмов анализа достижимости [20, 26 – 28, 31, 33, 35, 37]; проведение моделирования системы испарителя [24]; разработка алгоритмов свертки ДН-сетей [30, 44], преобразования МСГ, и их практическая реализация [30]; участие в проведении моделирования [21, 39, 76] и их сравнительный анализ [76];

обзор [17], получение характеристик [83] и анализ свойств среды DCNET, а также возможности ее интеграции в систему дистанционного обучения [16]; верификация программной среды моделирования [21]; введение нечеткого перехода и модификации программной среды [36]; разработка структур хранения данных, их динамического обновления, структур файлов и баз данных, систем моделирования, анализа достижимости, отображение графиков и таблиц, графического редактора [18, 20, 29, 31 – 33, 39, 90, 102]. Разработанная ИТ для моделирования и исследования ДН систем реализована в программной среде DCNET. Соискатель принимал участие в проведении ее испытаний и анализе результатов.

Апробация результатов диссертации. Основные теоретические положения и результаты диссертационной работы докладывались на следующих конференциях и семинарах: IV семинаре «Моделирование в прикладных научных исследованиях» (Одесса, 1997), 1-й международной школе по моделированию (Алушта, 1996), 45th International Scientific Colloquium, (Ilmenau Technical University, 2000), Международных конференциях «Автоматика» (Одесса, 2001, Винница, 2006, Одесса, 2008), Международных научно-практических конференциях «Современные информационные и электронные технологии» (Одесса, 2002, 2003, 2007 – 2012), семинаре «Информационные системы и технологии» (Одесса, 2003), Международной конференции «Кораблестроение: образование, наука, производство» (Николаев, 2002), Международной конференции «Современные технологии высшего образования» (Одесса, 2003), Международной научно-практической конференции «Системный анализ и информационные технологии» (Киев, 2004), 61-й научно-технической конференции профессорско-преподавательского состава ОНПУ (Одесса, 2004), XV международной научно-методической конференции «Управление качеством подготовки специалистов» (Одесса, 2010), Европейской выставке информационных технологий CeBIT'98 (Ганновер, Германия, 1998), а также на расширенном научном семинаре кафедры компьютерных систем ОНПУ (Одесса, 2016).

Публикации. Результаты диссертации опубликованы в 31 публикации, в том числе в 10 статьях в специальных научных журналах и сборниках научных трудов,

определенных МОН Украины как профессиональных изданий по техническим наукам, и 20 сборниках трудов и тезисов международных конференций. Из этих публикаций 1 – в издании, которое включено в международные наукометрические базы Index Copernicus International, Ulrich's Periodicals Directory, Electronic Journals Library, РИНЦ, ВИНТИ РАН.

РАЗДЕЛ 1

АНАЛИЗ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ МОДЕЛИРОВАНИЯ И ИССЛЕДОВАНИЯ СВОЙСТВ СЛОЖНЫХ СИСТЕМ В ДИСКРЕТНО- НЕПРЕРЫВНОМ ПРОСТРАНСТВЕ

1.1. Основные задачи и принципы проектирования систем управления гибридными техническими объектами

Продолжающийся в настоящее время на Украине сложный экономический кризис, выдвигает на первое место проблемы разработки новых технологий, позволяющих ускорить перестройку технологических процессов на производстве в связи с резким повышением требований к качеству готовой продукции и введением новых европейских стандартов. Повышение уровня производства может быть получено путем создания новых, перевооружением и интенсификацией действующих предприятий в таких отраслях промышленности как металлургическая, станкостроительная, химическая, нефтеперерабатывающая, пищевая и др. Выход из сложившегося кризиса на Украине возможен при широком внедрении прогрессивных ИТ. Без существенной модернизации экономики Украина не состоится как государство. Без повышения доли высокотехнологических и энергоэффективных производств страна не только не начнет жить лучше, а продолжит деградацию. Известно, что создание новых производств и процессов потребует ощутимого роста уровня автоматизации, и, как следствие, значительного усложнения соответствующих проектных решений. Постоянное ужесточение требований к производствам по устойчивости работы, безаварийности, защите окружающей среды, энергетическим показателям приводит к повышению сложности современных технологических систем и объектов, а также автоматизированных систем управления. В этом случае чрезмерно высокая цена

ошибочных проектных решений входит в явное противоречие с существующим инструментарием и технологиями проектирования.

Методическое обеспечение тесно связано с формированием задач теории автоматизированного проектирования, теории сложных систем и созданием теоретических основ разрешения этих задач. К общим задачам теории относятся задачи декомпозиции объекта проектирования, определения таких свойств как достижимость и управляемость, анализа общего проектного решения, построения законов функционирования объекта в целом как сложной технологической системы. К локальным задачам теории относятся задачи построения законов функционирования подсистем на всех уровнях и этапах проектирования, представленные для различной физической природы таких систем.

Темпы научно-технического прогресса в развитых странах требуют повышения качества и сокращения сроков проектирования сложных технологических объектов. Данная проблема может быть решена на основе широкого использования средств вычислительной техники.

Процесс проектирования систем автоматизации производственных или технологических систем имеет целью определение совокупности организационных и технологических мероприятий, обеспечивающих передачу ряда функций по управлению процессами автоматическим устройствам (контроля, регулирования, сигнализации, защиты и др.) для обеспечения оптимального ведения процессов. Следует учесть, что стоимость проектирования АСУ сложными ТО составляет от 5 до 15 процентов стоимости проектирования, а стоимость работ по сооружению АСУ – от 1.0 до 1.5 процента общего объема строительно-монтажных работ по сооружению объектов. Все это превращает автоматизацию производственных процессов в наиболее дорогостоящую часть проекта.

Современная новая техника, включающая ГПС, ГАП, АСУТП, автономно функционирующие объекты представляет собою основную составляющую материально-технической базы развитого общества. В состав новой техники входят подразделения, цеха. В частности, в энергетическое подразделение тепловой станции входят различные энергетические установки (турбины, компрессора,

теплообменники), электрические установки (генераторы, преобразователи напряжения), вспомогательное оборудование. Станция – это система, которая функционирует, прежде всего, за счет сложного взаимодействия, характеризуется большим числом различной природы частей, взаимодействующих между собой сложным образом. Важнейшим требованием, предъявляемым к этим системам, является оптимальное использование на каждом этапе или режиме функционирования ресурсов (энергетических, информационных, вычислительных и др.) для достижения цели при соблюдении множества ограничений.

Переход от проектирования отдельных агрегатов (тепловых, гидравлических) к проектированию сложных технологических комплексов совместно с автоматизированными системами управления порождает проблемы, качественно отличающиеся от проблем разработки суммы отдельных агрегатов [49,50,56]. Возникает необходимость рассматривать взаимосвязи в системе. С учетом того что в технические комплексы входят аппараты и установки различной физической природы (дискретные и непрерывные) резко усложняются вопросы математического моделирования систем и вопросы их анализа [115, 116].

Важнейшей составляющей современной прогрессивной ИТ, входящей в состав методологического, математического и программного обеспечения, является модель объекта проектирования.

На практике при автоматизации сложных технических систем приходится часто решать задачи управления системами, параметры и структура которых подвержены внезапным скачкообразным изменениям, а также системами, цели функционирования которых могут изменяться в процессе функционирования. Поэтому в указанных технических системах необходимо выделить дискретно-непрерывные процессы, состоящие из подпроцессов, которые функционируют как непрерывным, так и дискретным образом. Такие системы достаточно широко распространены в промышленности и получили на западе название гибридных.

Подобные системы характеризуются особенностями:

1. Составной характер системы. Подобные системы состоят из подсистем, некоторых достаточно автономных частей всей системы. Расчленение

сложной системы на подсистемы зависит от принятых технических решений и целей создания системы.

2. Разнородность подсистем и элементов, составляющих системы. Выделим несколько видов разнородности. Прежде всего – это физическая разнородность. Вторая разнородность – это разнородность математических схем, описывающих функционирование различных элементов. Минимальный состав системы может включать две подсистемы с различными математическими схемами и средствами интерфейса между ними.
3. Многорежимный фактор функционирования системы. Это значит, что достижение цели может быть выполнено за счет набора выполнения различных операций. Это происходит потому, что для реализации глобальной цели необходимо реализовать промежуточные подцели, и, как следствие, многократность оценок процессов, протекающих в подсистеме.
4. Случайность и неопределенность факторов, действующих в системе. Примерами подобных факторов могут служить погодные условия, паводки, случайные отказы оборудования, транспорта, укладочных устройств и т.д. Учет этих факторов приводит к резкому усложнению задач и увеличивает трудоемкость исследований (необходимость получения представленных наборов данных).
5. Многокритериальность оценок процессов, протекающих в системе. Невозможность однозначной оценки диктуется следующими обстоятельствами: наличием множества подсистем, каждая из которых, вообще говоря, оценивается по своим критериям; множественностью показателей (иногда противоречивых), характеризующих работу всей системы (например, темпы укладки бетона, качество строительных работ); наличием неформализуемых критериев, используемых при принятии решений (в случае, когда решения основаны, например, на практическом опыте лиц, принимающих решения).
6. Большая размерность системы.
7. Дискретно-непрерывное пространство состояний.

Указанная особенность технологических систем требует разработки новых математических средств, которые могли бы адекватно отражать дискретный и непрерывный характер функционирования технологических процессов.

Для описания динамических процессов, происходящих в сложных системах, таких как цех, отделения, предложено использовать основы теории дискретно-непрерывных систем с управляемой структурой и составом подсистем, в основу которой положена теория ДН-сетей.

При переходе объекта из начального состояния (x_0^c, x_0^d) в заданное (x_3^c, x_3^d) необходимо сформировать алгоритм функционирования при известных ограничениях на элементы конструкции ОУ и переменные и попытаться определить управляющее воздействие $u = F(x^c, x^d, t)$, обеспечивающее перевод системы из начального состояния (x_0^c, x_0^d) в конечное (x_3^c, x_3^d) и при этом обеспечивающее критерий эффективности.

Разработка алгоритма управления сложными системами, используя аналитические методы, во многих случаях практически невозможна.

Известные SCADA системы удачно решают задачу диспетчеризации подразделения и формирования распределенной АСУТП. Однако они не предназначены для формирования алгоритмов функционирования технологического комплекса. Считается, что эта проблема была решена ранее.

Анализ процесса проектирования таких систем показывает, что задачи построения законов функционирования и анализа динамических свойств таких систем наиболее сложны в своем решении. И необходима ИТ, позволяющая решать данную проблему. Основой ИТ является математическая модель объекта или системы. Поэтому развитие ИТ должно идти по пути объективных инноваций, их анализа и одновременно построения математических моделей объекта, средств моделирования и анализа.

Инновации могут быть и со стороны математических моделей (математических схем) объекта, и со стороны средств моделирования и построения промышленных систем с новыми возможностями. В качестве объекта может быть деталь, а моделью объекта – ее геометрическая модель.

Далее будем рассматривать и изучать объекты, такие как цех, участок цеха, автономно функционирующие объекты, «вводимые» в виде математической модели.

Одной из основных задач, которую требуется решить при разработке информационной технологии, является задача декомпозиции объекта, а также задача построения закона функционирования объекта в целом.

В состав методологического обеспечения информационной технологии целесообразно включить модель объекта проектирования – формализованный элемент, категориями которого должны быть отображены цели функционирования объекта проектирования.

Распространение системного подхода на интересующие нас системы, состоящие из непрерывно-событийной и дискретно-событийной подсистем, а также интерфейса соединения между ними, возможно при слиянии логических и динамических уровней абстракции и рассмотрения комплексов управления различной физической природы с единых позиций. Ниже рассмотрим:

- различные уровни абстракции начиная с 1965г., работы [5, 10, 14, 23, 36] и по настоящее время;
- подходы к определению задачи достижимости в непрерывно-дискретном пространстве состояния;
- развитие и применение ИТ и программных средств моделирования и исследования сложных систем;
- наметим дальнейшие задачи разработки средств визуализированного моделирования и теоретическое обоснование определения достижимости ДН систем.

1.2. Анализ математических схем представления ДН систем

Рассмотрим два примера использования математических моделей для описания процессов различной природы, в которых используется дифференциальная динамика и логические условия изменения свойства систем.

Первый пример относится к описанию энергосистем.

Пусть дано уравнение системы

$$\dot{x}^c = Ax^c + Bu^c \quad (1.1)$$

где $x^c \in R^n$ – непрерывное состояние, $u^c \in R^l$ – непрерывное управляющее воздействие, A – постоянная $(n \times n)$ – матрица, B – постоянная $(n \times l)$ – матрица. Матрица A разбивается на блоки так, чтобы диагональные блоки были квадратными.

Тогда, относя переменные соотношения, множителями при которых являются элементы матрицы A_i i -блока, к вектору состояния x_i подсистемы S_i , уравнение (1.1) разбивается на S подсистем и может быть представлено уравнением

$$\dot{x}_i^c = A_i x_i^c + \sum_{\substack{i=1 \\ i \neq j}}^S A_{ij} x_j^c + B_i u^c, \quad i = \overline{1, p},$$

где $x_i^c \in R^{n_i}$ – вектор состояния, $u_i^c \in R^{l_i}$ – вектор управляющего воздействия подсистемы S_i .

1.2.1. Интегрированные системы на основе модели конечного автомата

Впервые элементарная модель ГС были разработаны в институте Кибернетики им. М.В. Глушкова в конце шестидесятых годов проф. Жуком К.Д. и получила название логико-динамическое звено.

Логико-динамическое звено представлено совокупностью систем дифференциальных уравнений, моделирующих отдельные режимы функционирования системы [50]:

$$\begin{aligned} \dot{x}^c(t) &= f_v(x^c, u^c, t) - \text{состояния}, \\ y^c(t) &= \eta_v(x^c, t) - \text{выхода}, \quad v = \overline{1, N}, \end{aligned} \quad (1.3)$$

$$x^c(t_0) = x_0^c$$

где: $x^c \in R^n$, $u^c \in R^m$, $y^c \in R^r$ – вектора состояния, управляющего воздействия и выхода, t – время, f_v и η_v – переходная и выходная функции системы в v -ом режиме. Здесь и далее признаком непрерывной переменной есть индекс c в верхней части обозначения.

Каждая из векторных функций $f_v(u^c, x^c, t)$ представляет локальную динамику системы в v -ом режиме, свойства таких режимов отличаются между собой численными характеристиками параметров и траекторией. Например, динамика полета летательного аппарата в области дозвуковых и сверхзвуковых скоростей представляют собой дискретную последовательность характерных режимов системы.

Каждому режиму v однозначно соответствует структура Σ_v системы $\Sigma\{\Sigma_v\}_{v=1}^N$

Для описания порядка смены структуры в системе $\Sigma\{\Sigma_v\}_{v=1}^N$ используется конечный автомат, определенный характеристическими функциями:

$$\alpha_v^f = \lambda(a_{v-1}^f, \omega_v) \text{ – переходной,} \quad (1.4)$$

$$\beta_v = \delta(a_v^f, \beta_v) \text{ – выходной,} \quad (1.5)$$

и множествами $A = \{a\}$ состояний, $\Omega = \{\omega\}$ – входов, $B = \{\beta\}$ – выходов.

ЛДС представляется уравнением состояния

$$\dot{x}^c(t) = a_v^f \cdot f_v(x^c(t), u^c(t), t) \quad (1.6)$$

при условии единственности $a_\alpha^f \cap a_\gamma^f = 0, \alpha \neq \gamma$ и полноты $\bigcup_{v=1}^N a_v^f = 1$.

Аналогично записано уравнение выхода системы

$$y^c(t) = a_v^f \cdot q_v(x^c(t), t) \quad (1.7)$$

при тех же условиях единственности и полноты.

Для формализации непрерывно-событийной части логико-динамической системы использована гибридная функция, приведенная в работе [50].

Для создания интегрированной системы модель конечного автомата применяется в системах с управляемой структурой ОУ и составом подсистем [50], а также при создании модели гибридной сети. В первой системе предложена векторно-матричная гибридная функция Ξ , названная вектор-функцией управления структурой. Во второй используются понятия генератора событий, отражающего неконтролируемые события χ , и контроллер обратной связи γ_c . Поведение приведенных систем следующее.

Когда система находится в дискретном состоянии a_v^c , непрерывное состояние $x^c(t)$ развивается согласно динамическому уравнению с разрывной правой частью [50]. Как только непрерывное состояние входит в одну из областей $x^c(t) \in \bar{X}_i$, вырабатывается управляющее воздействие генератором событий (конечным автоматом), изменяется правая часть системы уравнений и функционирование происходит в соответствии с измененной производящей функцией $f_{v+1}^c(x^c(t))$. Приведенные модели должны быть дополнены условием единственности и полноты. Должно быть определено начальное структурное состояние $\langle f_1, a_1, q_1 \rangle$, тогда допустимое входное слово $\delta = (\omega_1, \omega_2, \dots, \omega_r)$ автомата единственным определит один из переходов на множестве структур $\Sigma_{v=1}^N$.

Хотелось бы обратить внимание на то, что в работах, посвященных созданию теории ЛДС [50], определены перспективные в будущем разработки. Так теория систем с управляемой структурой охватывает широкий класс объектов, функционирующих в режимах взаимодействия, координации и целенаправленной смены структурных состояний. В этом классе систем измеряемая дифференциальная динамика «вкладывается» как элементарный подпроцесс в дискретную динамику структурных состояний. На втором уровне в сложной системе представлено

управление составом взаимодействующих подсистем. Модель третьего уровня есть развивающаяся логическая сеть.

1.2.2. Интегрированные системы на основе модели сети Петри

При разработке модели системной динамики системы с управляемой структурой [34] впервые была использована СП для описания событийной динамики. Термин системная динамика несет нагрузку объединяющего начала НС подсистемы, представленной СГ (непрерывными переходами и позициями) и ДС подсистемы, представленной СП. Данная модель нашла развитие в работах [16 – 33, 35 – 45, 56, 79 – 83, 101, 102] при формализации ДН-сетей.

ДН-сеть состоит из позиций двух видов, переходов трех видов, функций следования и предшествования и может быть представлена кортежем

$$ДНС = \langle R, H, M_0 \rangle,$$

где H – оператор пометок;

M_0 – маркировка;

$R = (P, T, I, O)$ – набор вершин сети, состоящий из конечного набора позиций P , конечного набора переходов T , причем $P \cap T = 0$ и $P \cup T = 0$, а также входных I и выходных O .

Виды переходов, позиций и дуг, используемые в ДН-сетях [58] приведены в табл. 1.1.

Условия срабатывания структурно-управляемых переходов t_j^{cs} следующие:
 $\forall p_i^c \in I^{cs}(t_j^{cs}): \mu(p_i^c) = \alpha$, $\forall p_i^d \in I_i^{cs}(p_i^d, t_j^{cs}): \mu(p_i^d) = 0$, где $I_i^{cs}(p_i^d, t_j^{cs})$ – ингибиторная дуга.
 Новая маркировка определяется по правилу: $\forall p_i^c \in O^{cs}(t_j^{cs}): \mu_1(p_i^c) = \mu_0(p_i^c) + \alpha$,
 $\forall p_r^d \in O^{cs}(t_j^{cs}): \mu_1(p_r^d) = \mu_0(p_r^d) + 1$, где $\mu_0(p_i^c) = 0$, $\mu_0(p_r^d) = 0$.

Таблица 1.1 – Виды переходов, позиций и дуг

t_1^d 	p_1^d 	t_1^c 	p_1^c 	t_1^{cs} 	
Дискретный переход	Дискретная позиция	Непрерывный переход	Непрерывная позиция	Структурно-управляемый переход	Дуги: Нормальная Ингибиторная

Переходы вида t^c и t^{cs} помечаются операторами (дифференцирования, интегрирования, предикатной функцией, нелинейными преобразователями). Переходы вида t_j^c срабатывают мгновенно, а t_j^d – в соответствии с правилами СП.

Маркировка ДН-сети позволяет характеризовать функционирование соответствующего дискретно-непрерывного процесса (системы). Динамика изменения событийной части состояний моделируется движением меток по позициям p_i^c и p_i^d ДН-сети.

Основным преимуществом ДН-сети является возможность описания единым математическим аппаратом как дискретной, так и непрерывной частей системы. Далее, граф описываемой системы своей структурой полностью повторяет собой внутреннюю структуру системы (в отличие, например, от гибридных СП), что позволяет исследовать достижимость системы при помощи исследования достижимости графа. Также предельно прост переход к описанию при помощи ДН-сетей от традиционного описания НС части системы, поскольку они имеют аналогичные элементы описания, расширяющиеся только при введении непрерывных позиций.

Дискретное состояние $x_a^d(t_k) \in Z^x \subseteq N^{Z_x}$ эквивалентно маркированию интерпретированной сети Петри с P^d позициями и T^d переходами и зависит от предшествующего состояния $x_a^d(t_{k-1})$:

$$x_n^d(t_k) = x_n^d(t_{k-1}) + \bar{A}_n v_n^d(t_k) + u_n^d(t_k) + W_n^d(t_k), \quad (1.16)$$

где $x_n^d(t_k)$ представляет число маркеров в позиции p_i^d , $\mu(p_i^d) = x_n^d(t_k)$ в момент времени t_k , $\bar{A}_n = [\bar{A}_{n1}, \dots, \bar{A}_{nm}]$ – матрица инцидентий сети Петри с векторами-столбцами \bar{A}_{ni} , соответствующими модификации маркировки, если ассоциированный с соответствующим столбцом переход t_i^d срабатывает; $v_n^d(t_k)$ – вектор срабатывания. Изменение одной из компонент $v_n^d(t_k)$ происходит при попадании $x^c(t)$ в переключающее множество $\gamma_{ij} = X_i \cap X_j$. С целью получения определенной модели общая граница нескольких смежных переключающих множеств $\bar{X}_i \cap \bar{X}_j, \dots, \bar{X}_i \cap \bar{X}_l$ назначается исключительно одному из них.

Таким образом, динамика СУС описывается вышеприведенными уравнениями (1.15) – (1.18) при условии «сшивания» решения $x(t, |t_k|)$ на отдельных участках. Структурное состояние ОУСУС определено как пятиместный предикат $\Sigma_i = (x_0^c, x^c(t_i), M_i^d, t_i^c, P_i, Q_i)$.

Непрерывная функция $x^c(t)$ удовлетворяет решению уравнения (1.3) абсолютно. Объединение непрерывного состояния с дискретным состоянием сети даст гибридное состояние системы. Поэтому гибридное состояние сети представлено шестиэлементным кортежем $Z = (\mu, q, \dot{x}^c, x^c, x_0^c, \tau_0)$. Z принимает значения из множества $H = Z^{|x|} \times \Delta \times C \times R^n \times R$.

Особое внимание обращает на себя аппарат гибридной динамической системы, разработанный в рамках выполнения проекта «Непрерывно-дискретная динамика технических систем» в техническом университете г. Ильменау. Данная система является расширением гибридных СП. Основная идея по гибридным СП, на которой базируется гибридная динамическая система, состоит в том, что непрерывные и дискретные величины процесса отображаются с помощью непрерывных и, соответственно, дискретных состояний. При этом были использованы элементы сети, приведенные в таблице 1.1.

В то время как дискретные состояния могут принимать только целочисленное количество маркеров, при непрерывных состояниях происходит реально значимая интерпретация маркеров. Это является предпосылкой к формированию непрерывных (реально значимых) величин процесса.

Моделирование дискретных динамических систем происходит как и в обычных СП. Иллюстрация времени является обязательной предпосылкой при моделировании динамики системы. Это происходит благодаря непрерывным переходам, которые подчиняется функции скорости. Этим способом могут описываться линейные и нелинейные непрерывные системы.

Гибридные системы учитывают всё комбинациями непрерывных частей системы (таких как химические процессы) или смешанными дискретно-непрерывными (химические процессы, связанные с логикой) совместно с дискретной частью, которая реагирует на внешние события (запланированные или непредвиденные). Полная структура такой системы изображает следящие системы управления классической иерархической структуры в три уровня: исполнительный, следящий, координирующий. Распределения решений дискретны по природе, в то время как технологические процессы могут быть непрерывны, дискретны или смешаны.

Существуют два основных пути, по которым гибридный процесс и система управления могут быть сближены. Первый – добавление дискретной компоненты вокруг основного непрерывного решения; другой – добавление непрерывной величины в основной дискретный каркас – концепция, которую мы используем. Широко используется метод решения включения дискретно-непрерывной части.

Модель состояния сети содержит интерпретированную сеть Петри, представляет дискретно-событийную подсистему и расширенную модель пространства состояний, описывающую непрерывную подсистему.

Обе части системы объединены средствами интерфейсов. Более того, могут иметься как дискретные, так и непрерывные входы и выходы. Общая форма модели дана в [113]. Непрерывная динамика будет кусочно-линейной с целью допустимого

упрощения в течение моделирования или с целью подходящей аппроксимации нелинейной непрерывной динамики [113].

Непрерывное состояние $x^c(t) \in X^c \subseteq R^{n_c}$ изменяется в соответствии с моделью пространства состояний

$$\begin{aligned} \dot{x}^c(t) &= \underline{f}_{Cx}(x^c(t), \underline{v}_c(t), u^c(t)) = \underline{A}(\underline{v}_c)x^c(t) + \underline{a}_0(\underline{v}_c) + \underline{B}(\underline{v}_c)u^c, \\ x^c(0) &= x_0^c \end{aligned} \quad (1.21)$$

с вектором параметров $\underline{v}_c(t) = \underline{f}_{DC}(x^d(t)) \in R^{m_c}$ и действующими извне непрерывными входами $u^c(t) \in u^c \subseteq R^{p_c}$.

Дискретное состояние $x^d(t_k) \in X^d \subseteq N^{n_d}$ представлено в математической форме аналогично (1.16). Изменение дискретного состояния $x^d(t_k)$ соответствует управляющему событию, если оно непосредственно инициировано изменением хотя бы одного компонента в $u^d(t_k)$. Внутреннее событие происходит, если изменение дискретного состояния инициировано переключением вектора $\underline{v}_D(t) = \underline{f}_{CD}(x^c(t)) \in \{0,1\}^{m_D}$ с $x^c(t) \in \Omega_i$. Если непрерывное состояние достигает многообразия $\Omega_i \subseteq X^c$, так называемого переключающего множества, в результате внутреннего события, то Ω_i указывает на активное переключающее множество при текущем дискретном состоянии x^d , обозначенное как $\Omega_i^*(x^d)$.

С целью получения чисто определенной модели общая граница нескольких смежных переключающих множеств Ω_i назначается исключительно одному из них, также как в предыдущей работе.

Анализ структурных схем и математического описания рассмотренных гибридных систем позволили определить общность в их представлении:

1. Используются алгоритмические языки, которые позволяют связать в единый комплекс как числовые, так и логические зависимости. В первом случае в качестве такого языка использован язык гибридных функций [90], во втором –

матричное представление такого языка с использованием матрицы линейки и вектор функции управления структурой [50].

2. В структурном состоянии Σ_i непрерывно-событий части ГС для f_i базисными функциями являются

$$\{\mathcal{U}\}: (+, -, \cdot, \frac{1}{P}, P, \tau),$$

где $\frac{1}{P}$, P , τ – операторы дифференцирования, интегрирования и запаздывания.

3. Нелинейные операторы – Система логических функций состоит из:

а. $\{L\}: (\vee, \wedge, -, \rightarrow),$

где $\vee, \wedge, -, \rightarrow$ – символы операторов дизъюнкции, конъюнкции, отрицания и импликации соответственно, в работах [5, 56] реализованная моделью конечного автомата.

б. $\{G\}: \langle P, T, I, O, M \rangle -$

логических отключений, представленных СП [56, 67, 73].

4. Отмечается тенденция смены логических функций $\{L\}$, реализующихся моделью конечного автомата на функции $\{G\}$, реализующих сеть Петри. Последнее связано с тем, что, в отличие от конечного автомата, сеть Петри можно описать асинхронные и параллельные процессы обработки информации, присущие сложным системам [75].

Анализ структурных схем и математического описания различных видов гибридных систем позволили предложить обобщенную схему гибридной системы.

Из структурной схемы, приведенной на рис. 1.1 видно, что для представления объекта управления с управляемой структурой (гибридного процесса) и регулятора (контроллера) с управляемой структурой может быть использована одна и та же математическая схема, такая как [56]. Аналогичное положение и с моделью логических устройств (они так же могут быть представлены одной схемой).

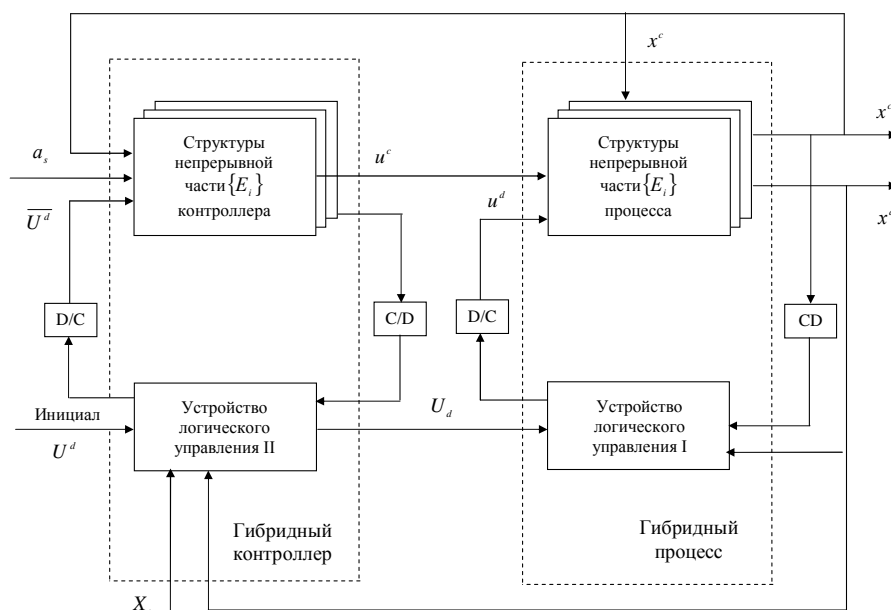


Рисунок 1.1 – Обобщенная схема ГС

В этой схеме управляемая часть системы состоит из гибридного процесса, включающего структуры непрерывной части процесса $\{\Sigma_i\}_{i=1}^N$, и устройства логического уравнения I. Данное представление гибридного процесса связано с тем, что процесс существенно нелинейный, или же в результате кусочной линеаризации нелинейного процесса представлен линейными структурами $\{\Sigma_i\}_{i=1}^N$ и логикой их переключения. В этой части основные динамические свойства отдельных операций $\{\Sigma_i\}_{i=1}^N$ гибридного технологического объекта управления (процесса) определены логическим устройством управления I. В свою очередь, в гибридном процессе «ведущим» является УЛУ I, а «ведомым» – НС часть.

Управляющая часть (гибридный контроллер) состоит из непрерывных структур $\{R_i\}$ регулятора (законов регулирования) и устройства логического управления II. Логическое устройство I обеспечивает выполнение отдельных операций в гибридном процессе, а логическое устройство II – реализацию режима функционирования (набора отдельных операций), а также формирование соответствующих им наборов законов регулирования $\{R_i\}_{i=1}^N$. «Ведущим» является УЛУ II, а «ведомым» – УЛУ I при реализации режимов функционирования в гибридном процессе.

Хотя гибридные СП дали новый моделирующий аппарат для систем, содержащих в себе как дискретно-событийную, так и непрерывную часть, всё же, при сравнении с ДН-сетями, описать апериодические и более сложные процессы, которые характерны для химических, пищевых и других производств, с помощью гибридных сетей Петри не возможно. Это определено ограничениями гибридной СП – маркировка непрерывной позиции может быть только положительным вещественным числом не больше целого и скорость срабатывания перехода должна быть постоянной. ДН-сети также обладают некоторыми недостатками, такими как повышенная размерность задачи при преобразованиях дискретных сигналов в непрерывные. Математический аппарат ДН-сетей также не полностью разработан. В частности, не разработаны методы исследования таких свойств ДН-сетей как достижимость, покрываемость и др. Рассмотренные методы построения и исследования СУС позволяют описывать их как сложные системы, причем большинство из них требуют разбиения на дискретную и непрерывную составляющие. ДН-сети лишены данного недостатка, причем наиболее прогрессивными в данном отношении являются ДН-сети, поскольку обладают более гибким математическим описанием. Применение математического аппарата ДН-сетей как наиболее полно отражающего внутреннюю структуру СУС позволит решить такие задачи исследования как выход системы в заданный режим путем наименьших затрат уже на этапах проектирования.

С современной точки зрения все достижения классической теории управления объясняются тем, что в ней изучались только такие модели, для которых свойство управляемости и идентифицируемости были естественными.

1.3. Методы определения управляемости (достижимости) динамических систем

Определение. Событие (τ, x^c) для линейной системы Σ называется управляемым (относительно начала координат) тогда и только тогда, когда найдется

такое $t \geq \tau$ и такое входное воздействие ω , которое переведет систему из состояния (τ, x^c) в состояние $(t, 0)$.

Одним из основополагающих свойств динамических систем является свойство управляемости (достижимости). Данное понятие связано с переводом системы посредством управления из одного состояния в другое. При рассмотрении структурно-качественной стороны достижимости интересуются принципиальной возможностью перевода системы из одного заданного множества состояний в другое заданное множество, как правило, за конечное время [27, 30, 31, 56]. При количественном изучении достижимости рассматриваются те или иные характеристики переходных процессов при простейших типовых управляющих воздействиях. Может быть рассмотрена достижимость как динамического объекта, не оснащенного регулятором, так и систем, содержащих замкнутые контура регулирования.

Наиболее широкое распространение для исследования критерия управляемости получило использование критерия Калмана, разработанного для линейных стационарных систем, представленных в пространстве состояний. Необходимое и достаточное условие полной управляемости в v -м структурном состоянии в соответствии с определением Калмана имеет вид:

$$\text{rank } G_v = \text{rank} \begin{vmatrix} B_v & A_v & B_v & \dots & A_v^{n-1} B_v \end{vmatrix} = n.$$

Для СУС выведены условия управляемости в работе [56] и они имеют следующие критерии:

$$\text{rank} \{ B_1, A_1 B_1, \dots, A_1^{(n-1)} B_1, B_2, A_2 B_2, \dots, A_2^{(n-1)} B_2, \dots, B_n, \dots, A_n^{(n-1)} B_n \} = n. \quad (1.22)$$

В данном случае критерий Калмана может быть не выполнен для каждой из структур, а в целом условие (1.22) выполнимо, и тогда СУС управляема. Похожие

условия управляемости, только другим методом, были получены в работе [56] и могут быть представлены для системы, состоящей из двух структур (Σ_i, Σ_{i+1}) :

$$\text{rank } S_{ij+1} = \text{rank}(B_i, A_i B_i, \dots, A_i^{n-1} B_i, B_{i+1} A_{i+1} B_{i+1}, \dots, A_{i+1}^{n-1} B_{i+1}) = n.$$

Исследование достижимости на основе построения матрицы всех путей в орграфе рассмотрены в работе [13]. Для этого в состав переменных z введём три группы переменных: переменные состояния $x_i = z_i, i = 1, 2, \dots, n$, управления $u_j = z_{n+j}, j = 1, 2, \dots, l$, и выходные переменные $y_k = z_{n+l+k}, k = 1, 2, \dots, m$. Для вершин $z_r, r = 1, 2, \dots, N$, где $N = n+l+m$, составим матрицу единичных и нулевых путей $R_1 = I + M$. Вершина u обозначает позицию-исток, то есть позицию, у которой есть только выходные дуги. Вершина y – позицию-сток, то есть позицию, у которой есть только входные дуги. И вершина x обозначает промежуточную позицию, у которой есть как входные дуги, так и выходные.

Если в системе отсутствуют обратные связи ко входам u от выходов y и состояний x , то есть $R_1^{xu} = 0, R_1^{yu} = 0$, и нет прямых связей от y к x , то есть $R_1^{yx} = 0$. Пусть также нет связей от u_i к u_j и от y_k к y_l , то есть соответствующие матрицы M^u и M^y будут нулевыми, а матрицы R_1^u и R_1^y равными $I + M$ – единичными. Таким образом, матрица всех единичных и нулевых путей орграфа принимает вид:

$$R_1 = \begin{pmatrix} R_1^x & 0 & R_1^{xy} \\ R_1^{ux} & 1 & R_1^{uy} \\ 0 & 0 & 1 \end{pmatrix} \quad (1.23)$$

Если рассматривается система, у которой все передаточные функции – правильные дроби, то y не будет зависеть от u и нулевым станет ещё один блок R_1^{yu} .

Для получения матрицы всех связей между вершинами возведём матрицу R_1 в степень $N-1$. Условия достижимости можно сформулировать следующим образом [13]: *множество вершин X достижимо из множества входов U , если и только если в матрице R_1^{uxN} нет нулевых столбцов.*

В данном подходе есть одна особенность: исследуемая система не должна иметь сокращения нулевых полюсов.

По аналогии с исследованием сетей Петри в работах [3, 4, 95] предложено использовать подход определения достижимости на основе построения дерева достижимости дискретно-непрерывной сети [31, 37] или эволюционного графа.

1.4. Визуализированные средства моделирования и анализа сложных технических автоматизированных систем

1.4.1. Средства моделирования систем, разработанных на основе модели конечного автомата

При разработке АСУ значительное внимание уделяется автоматизации динамических расчетов (моделированию рабочих процессов установок, построению законов функционирования и обоснованию выбранных систем управления). Одним из первых средств моделирования АСУ являются программы СИАМ и «Эскиз», разработанные для выполнения в среде MSDOS. В настоящее время существует достаточно большое количество пакетов, предназначенных для моделирования отдельно, как непрерывных [56, 62 – 64, 102], так и дискретных систем [56, 62, 102, 106]. В то же время моделирование систем, сочетающих в себе непрерывные и дискретные компоненты, и, в частности, систем с управляемой структурой, гибридных систем, с их помощью является затруднительным и зачастую не является возможным.

Ниже остановимся подробно только на тех пакетах, которые позволяют моделировать как непрерывные, так и гибридные системы, формировать алгоритм функционирования, и реализованы как визуализированные среды. Такие системы

охватывают три основных класса: системы визуализации научных исследований, системы информационной визуализации, визуализированные тренажеры и SCADA-системы. Широко используется в научных исследованиях математический пакет MATLAB, разработанный в MathWorks, Inc. (Toronto). Данный пакет фактически является интерпретатором собственного языка программирования, на котором реализовано большое количество прикладных задач из различных областей науки и техники. Данная версия является очень удобной для моделирования САУ за счет включения в состав пакета программы Simulink, позволяющей осуществлять графический ввод и моделирование структур САУ. Simulink позволяет достаточно быстро и просто ввести схему моделируемой системы, причем, поскольку эта схема отображается на экране, то уменьшается вероятность ошибок, связанных с неправильным подключением связей, кроме того, упрощается исправление параметров блоков. Метафорой визуализации системы SIMULINK является блок-схема линейной или нелинейной САУ. Выбор элементов, предоставляемый для построения моделей САУ, также достаточно велик. Сюда входят линейные, нелинейные, дискретные элементы, различные генераторы сигналов, блоки, реализующие дискретно-цифровые преобразования, и логические элементы. К сожалению, в число блоков, которые предоставляет пользователям MATLAB, не входят элементы, описывающие сети Петри, поэтому для моделирования СУС необходимо использовать блоки переключатели (блоки умножения) для подключения–отключения непрерывных частей системы при изменении её структуры, а для задания поведения дискретной части – использовать логические автоматы Мили или Мура.

1.4.2. Средства моделирования систем, разработанных на основе модели сети Петри

Расширением пакета MATLAB/Simulink является пакет SIKOSS, позволяющий проводить моделирование гибридных систем, представленных двумя подсистемами: непрерывно-событийной и дискретно-событийной, последняя

представлена моделью сети Петри. Эти два пакета работают как отдельные процессы, обменивающиеся данными при помощи механизма Dynamic Data Exchange.

Следующей программой, предназначенной для моделирования сложных систем, описываемых на основе СП, является программа Visual Object Net++ [106]. Написанная с использованием принципов объектно-ориентированного программирования, она имеет два уровня представления гибридных объектных сетей. Первый уровень – уровень СП, включает в себе гибридные СП. Элементами данного уровня являются классы, описывающие непрерывные и дискретные позиции и переходы гибридной СП, а также обычные, ингибиторные и тестовые дуги. В данном случае метафорой визуализированного языка является гибридная СП, которая позволяет представлять модели различных подсистем, имеющих входы и выходы, представленных специальным классом входных и выходных непрерывных и дискретных позиций, наследованных от объектов соответствующего класса.

Для анализа и проектирования сложных технологических систем и комплексов разработана программа DCNET v.1.0 [90]. Данная программа позволяет осуществить графический ввод сложных технологических систем и комплексов, представленных языковыми средствами ДН-сетей, нейронных сетей, что дает возможность на всех этапах разработки системы применять единое программное средство, объединяющее непрерывные и дискретные компоненты исследуемых систем. В этой среде метафорой моделирования является ДН-сеть, элементы которой приведены в табл. 1.1. Программа позволяет осуществлять моделирование с наглядным отображением структурной и непрерывной динамики и возможностью оперативной корректировки параметров моделирования. Важным достоинством программы является возможность реализовать алгоритмы управления структурой системы, логические законы управления и логику поведения без предварительной разработки функциональных схем элементов логического управления. Имеющаяся версия программы включает около сотни стандартных типов блоков (линейных, нелинейных, дискретных и др.) и предоставляет средства для создания блоков,

определяемых пользователем. Программа поддерживает работу на нескольких языках (русский, английский, немецкий), обладает развитой системой помощи, а также развитыми средствами визуализации и документирования результатов исследования. Работа с данной программой не требует знания языков программирования, а лишь основных понятий ДН-сетей и СУС и навыков работы в среде Windows.

Анализ работ [1, 6, 12, 13, 16 – 33, 35 – 46, 53, 54, 56, 61, 70 – 72, 75, 95, 96 – 116] говорит о прикладной проблеме, а именно, визуализации представления структурных схем гибридных систем и результатов их моделирования. Последние представляются графиками изменения во времени непрерывных переменных и динамикой изменения маркировки непрерывных и дискретных позиций, а также представлений в табличных формах. Таким образом, визуальной моделью гибридной динамической системы является ее представление в виде структурной схемы (двудольного ориентированного графа с двумя видами вершин), представленной средствами ДН-сетей.

Общей физической сущностью в рассматриваемых случаях, которая представляется компьютерной моделью и является изучаемым объектом, являются АСУ в технических и экономико-социальных сферах человеческой деятельности.

Анализируя системы представления [47, 56, 62 – 64, 102, 106] можно выявить их общность, состоящую в том, что они поддерживают очень близкие графические метафоры, в частности те, формализованное представление которых есть ДН-сеть, ГС, дискретные, непрерывные и гибридные СП, и т.п. Вследствие чего существует много общего между собственными языками визуализированного программирования. В них используются подобные языки визуализации, словарь которых составляют наборы видов отображений (двумерные образы переходов и позиций) и изменения их графических и неграфических атрибутов. Рассмотренные языки визуализации дают возможность пользователю манипулировать визуализированными объектами на экране.

Существует собственное поведение графических моделей в рассмотренных примерах, а именно, движение маркеров по позициям и изменение их числа.

Грамматические правила рассмотренных языков визуализации существенно отличаются. Так, в языках различны задания последовательностей смены видов отображения.

Следует отметить, что вид отображения в различных системах существенно отличается. Так, в Simulink дается представление в виде блоков, представленных прямоугольниками в которых можно «записать» характеристику передаточной функции или звена. В других системах (VisSim, DCNet) язык визуализации содержит представление переходов в виде прямоугольника, размеры которых не позволяют «записать» внутри них параметры передаточной функции, но в этом случае программа дает возможность «раскрыть» изображение и «установить» в нем параметры.










Работа над общей и близкой проблемой [62-64, 106] определила выбор метафоры – очень близкой знаковой системы, которая используется при визуализации.




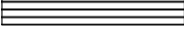
Анализ прикладной проблемы показывает, что большинство их разработано и реализовано для моделирования и исследования непрерывно-дискретных динамических систем, имеющих широкое распространение и использование в различных сферах человеческой деятельности (табл. 1.2).

Поскольку данная программа, в отличие от рассмотренных выше, представляет возможности для проектирования СУС на базе ДН-сетей, то будет целесообразным ее использование при разработке модели.

На практике в последнее время экономическая эффективность ведения процесса становится вторичной характеристикой. Сначала необходимо обеспечить такие свойства технологических систем как гибкость, т.е. возможность противостоять изменяющимся условиям внешней среды; стабилизируемость – свойство системы, связанное с качеством переходных процессов и устойчивостью; живучесть – возможность управления процессом при неисправностях в системе управления или агрегатах, с помощью которых реализуется технологический процесс; надежность – исключение аварийных ситуаций [10].

Таблица 1.2 – Возможности моделирования ДН АСУ существующими пакетами

Наименование пакета	Тип системы
СИАМ (МАИ, Москва)	
Эскиз (ИКАН, Киев)	
Simulink (MathWorks, Toronto)	
Simulink/SIKOSS (MathWorks, Toronto / VTU, Leipzig)	
Visual Object Net++ (TU Ilmenau)	
FlightSim (ЦАГИ, Москва)	
VisSim (Visual Solutions, Westford)	
MvStudium (MVSTUDIUM Group)	
МВТУ (МВТУ, Москва)	

-  – непрерывная система (совокупность дифференциальных уравнений)
-  – конечная система (модель автомата)
-  – конечная система (модель сети Петри)
-  – непрерывная система (непрерывная сеть Петри)

В связи с этим большое внимание уделяется разработке различных типов компьютерных тренажеров. Особую важность приобретает качественное обучение и постоянное поддержание квалификации и готовности персонала, его противоаварийные тренировки.

Динамический тренажер в своей основе имеет математическую модель реальных физических процессов и поэтому наиболее полезен для качественного обучения персонала.

Динамические тренажеры разработаны в ВТИ, МЭИ (Россия), ЦНТУ «Динамика». Разработанная высококачественная математическая модель динамики полета самолетов и вертолетов является серьезным преимуществом ЦНТУ

«Динамика», созданным на базе ЦАГИ (Центральный аэрогидродинамический институт). Для создания математической модели летательного аппарата с системой управления используется программный пакет FlightSim, разработанный на основе широко известной программной системы MATLAB/Simulink путем ее адаптации к авиационным задачам реального времени. Создание модели производится в интерактивном режиме из специальных блоков комплекса FlightSim, что обеспечивает простоту и гибкость его использования.

FlightSim получил широкое распространение в авиационных ОКБ России и других стран и их службах летных испытаний.

Задача интеграции в одном тренажере режимной (непрерывной) и коммутационной (дискретной) моделей энергосистемы и функций управления ими впервые была успешно решена во ВНИИЭ (Москва). Трудность этой задачи заключалась в необходимости обеспечить надежное синхронное функционирование двух сложных, разработанных независимо одна от другой и математически разнотипных моделей одной технической системы. Решение этой задачи позволило распространить на интегрированную коммутационно-режимную модель энергосистемы методики управления и режимом, и переключениями.

Выводы

1. Анализ технологических процессов, присущих техническим комплексам, указывает на то, что входящее в них оборудование и установки математически могут быть описаны на основе взаимосвязи числовых вычислительных алгоритмов и логических алгоритмов. В данном случае используются алгоритмы численного решения системы линейных (нелинейных) дифференциальных уравнений с разрывной правой частью. Логические алгоритмы могут быть представлены математическими схемами: системой логических функций, моделью конечного автомата, расширением сети Петри. Данный перечень является основой создания алгоритмических языков. В

частности при создании языка дискретно-непрерывных сетей использован алгоритм расширенных сетей Петри.

2. В большинстве из рассмотренных математических схем гибридных систем язык функций, заданных кусочной схемой, представлен термальной записью на основе гибридной функции [56] или в матричной форме с использованием вектор-функции управления структурой [56], последняя является наиболее перспективной, так как позволяет представлять многосвязные объекты, представленные в гибридном пространстве состояния.
3. Анализ и представление ДН систем в виде обобщенной схемы позволил определить дальнейшее направление в развитии описания математической схемы ДН системы: развитие языка дискретно-непрерывных сетей (введение дополнительных переходов и позиций, расширяющих возможности сетей), что позволит отразить эти свойства при определении матрицы инцидентов и классификации ГС.
4. Метод, используемый при анализе достижимости, тесно связан с формой описания (математической схемой), которая отражает имеющиеся знания о процессе, указывая объект автоматизации. Так как гибридные системы являются специализированным классом нелинейных систем, то не существует в общем виде критерия, при помощи которого достижимость всех комбинаций начального состояния и состояний заданного конечного (цели) может одновременно устанавливаться.
5. Анализ возможностей ИТ моделирования сложных технических систем (Simulink, MATLAB, VISSIM, Visual Object Net++) позволил сделать вывод о целесообразности создания новой ИТ на основе языка ДН-сетей, реализующего более широкую метафору визуализации, чем выше перечисленные системы.
6. Определение одного из основополагающих свойств – управляемости динамических систем – возможно только для линейных и некоторых нелинейных систем. Исследование свойства достижимости ДН АСУ в выбранном дискретно-непрерывном пространстве состояний на основе

аналитического метода, а в отдельных случаях – путем численного моделирования в предложенной визуализированной среде, является актуальной задачей.

РАЗДЕЛ 2

РАЗРАБОТКА МОДЕЛИ ДИСКРЕТНО-НЕПРЕРЫВНОЙ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ

Проведенный в разделе 1 анализ математических схем и комбинаций представлений АСУ указывают на то, что системы данного класса состоят из частей: непрерывно-событийной, событийно-дискретной и интерфейса между ними. НС часть представляется системой дифференциальных уравнений с разрывной правой частью, ДС часть представляется блок-схемой алгоритмов логического управления

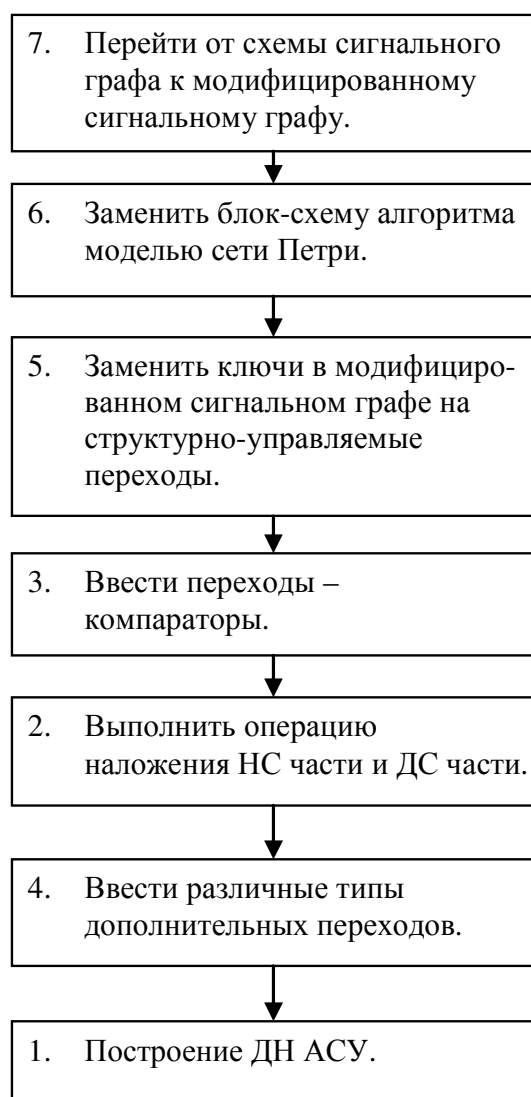


Рисунок 2.1 – Схема формирования ДН АСУ.

(моделью конечного автомата, сетью Петри). ДС часть является модулирующей по отношению к НС части. В известных работах по разработке вариантов математических схем ДН АСУ вначале определяют типы переходов и позиций, а затем предлагается структура ДН АСУ.

Анализ показывает, что единой схемой для двух частей может быть теоретико-множественное представление (СГ, ПГС). В данной работе будем отталкиваться от того, что НС часть представлена сигнальным графом (переходным графом состояния, диаграммой состояния) с логическими ключами, а ДС часть – блок-схемой алгоритма логического управления. Определим следующую последовательность, которая позволит подойти к формированию

математической схемы ДНАСУ. Данная схема представлена на рис. 2.1.

2.1. Разработка модифицированного сигнального графа

Известно, что для определения передаточных функций линейных непрерывных систем может быть использован метод графов (сигнальных графов) [13]. Для использования систем в пространстве состояний – переходный граф состояния или диаграмма состояния [56]. В обоих случаях оператором, преобразующим входной сигнал в выходной сигнал помечена дуга, соединяющая вершины, помеченные соответственно сигналами в одном случае непрерывными, в другом – дискретными. Исключение составляет оператор суммирования, им помечается узел, в котором реализуется суммирование сигналов. Это связано с тем, что граф системы строится на основе структурной схемы системы управления.

Определим последовательность процедур, позволяющих преобразовать СГ в форму, удобную для представления НС части ДНАСУ, так называемой модифицированной ДНАСУ.

Рассмотрим СГ $\hat{N}^c = \{P^c, Q^c, \hat{F}^c\}$, где $P^c = \{p_i^c\}$ – множество вершин графа; $Q^c = \{q_r^c\}$ – множество направленных дуг графа; $\hat{F}^c = \{\hat{f}_s^c\}$ – их пометочные функции (рис. 2.2). Каждой вершине p_i^c сопоставлен непрерывный сигнал x_i^c , а дугам $q_r^c(p_i^c, p_k^c)$ – функции преобразования непрерывных сигналов $f_s^c(q_r^c(p_i^c, p_k^c))$, кроме суммирования. В СГ операция алгебраического суммирования выполняется в вершине, в которую входят несколько дуг. Как известно, СГ могут быть представлены линейная [13] и некоторые нелинейные системы автоматического управления, однако в ПСГ отсутствует механизм изменения состава элементов и связей между ними. Наша задача состоит в том, чтобы реализовать этот механизм.

Процедуру перехода от ПСГ к МПСГ рассмотрим на примере графа, представленного на рис. 2.2. Граф содержит вершины-истоки p_1^c , p_2^c , промежуточные дуги, обеспечивающие преобразование сигналов, вершину-сток p_5^c . Будем считать, что в данном примере дуги помечены некоторыми операторами a_i ,

$i=1,2,3,\dots,5$, конкретные значения приведены на рис. 2.2 (символ S – оператор Лапласа).

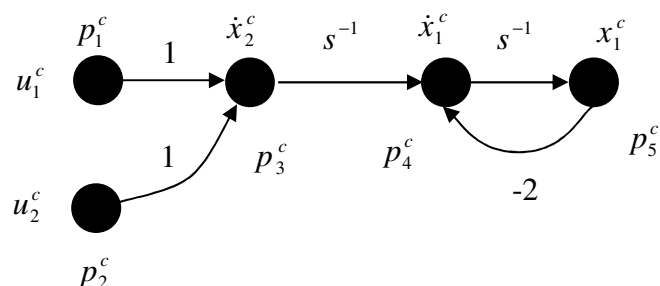


Рисунок 2.2 – Пример СГ.

Выделим фрагмент p_1^c, p_2^c, p_3^c , СГ приведенного на рис. 2.3а. Дополним его единичной дугой как показано на рис. 2.3б, тогда в узле p_6^c реализуется операция суммирования сигналов u_1^c и u_2^c , а в узле p_3^c – операция дифференцирования.

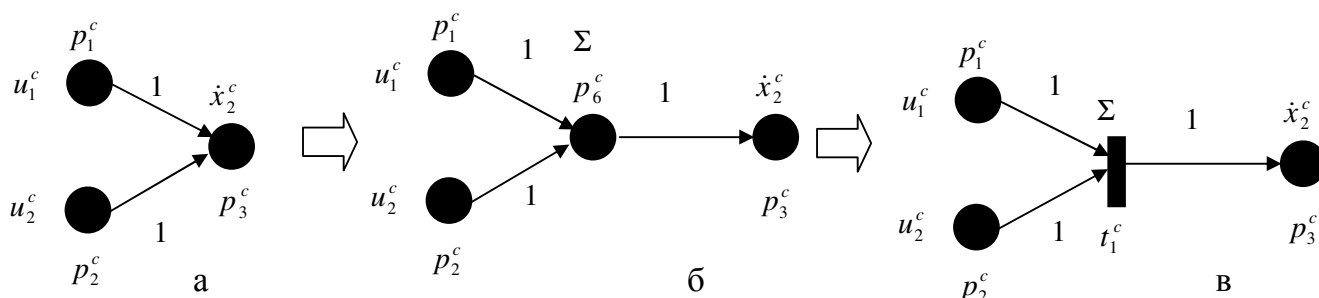


Рисунок 2.3 – Операции преобразования СГ: а – исходный СГ, б – введение сумматора, в – конечный МСГ.

Заменяем узел p_6^c на переход t_1^c с тем же оператором (суммирование) – получим подграф, изображенный на рис. 2.3.в.

Выполним замену элементарной дуги СГ на элементарную дугу МПСГ. Как показано на рис. 2.4, вершины p_1^c и p_2^c , соединенные дугой, дополним вершиной t_1^c , которой поставим в соответствие операцию преобразования s^{-1} , ранее закрепленную за дугой. Объединим последовательно включенные операторы суммирования и интегрирования – получим граф приведенный на рис. 2.5.а.

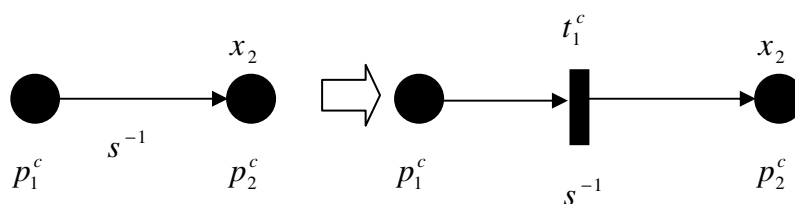
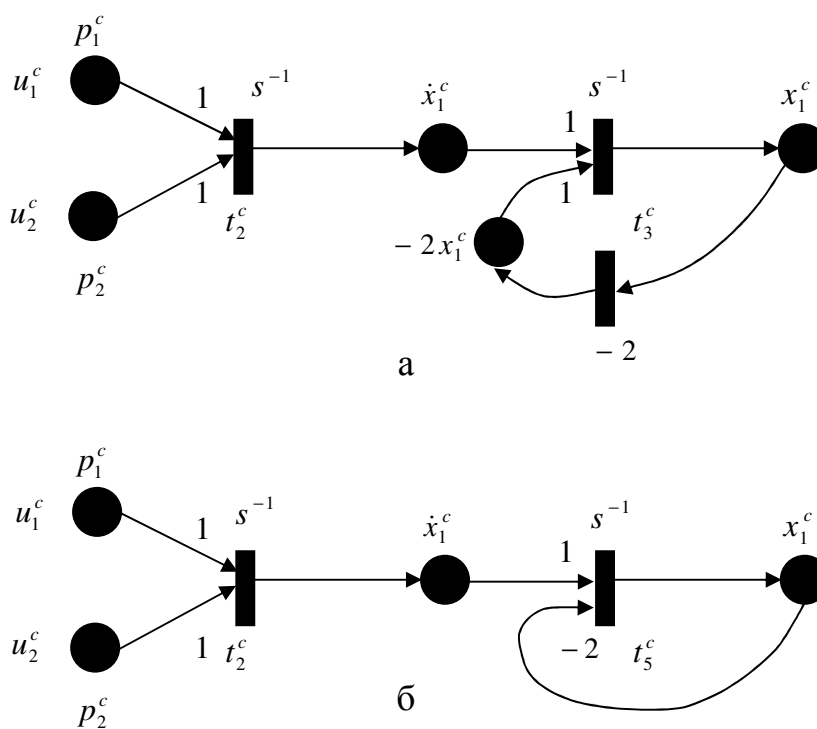


Рисунок 2.4 Преобразование элементарного СГ.

Рисунок 2.5 – Полученный МСГ: а – с переходом t_3^c в обратной связи; б – без перехода

В результате данного преобразования МПСГ содержит на одну единицу больше дуг и вершин, чем в исходном СГ. В данном случае это незначительное увеличение числа дуг связано с представлением операции суммирования.

Таким образом, метод перехода от СГ к МСГ, представленном СП, состоит в следующем.

1. Проводится проверка вершин СГ на количество входящих дуг. Если есть вершины, которые имеют более одной входной дуги, то каждая из таких вершин

обрабатывается в соответствии с шагом 2, после чего следует переход к шагу 3. Иначе – переход к шагу 3.

2. Найденная вершина заменяется на вершину-переход, ставится в соответствие функция преобразования сигнала – сумматор с вершиной-позицией, связанной с ним единичной дугой. При этом выходные дуги замененной вершины становятся выходными дугами вводимой вершины-позиции.

3. Каждая дуга СГ, за исключением введенных на шаге 2 выходных дуг, заменяется на вершину-переход, соединенный входной дугой с той вершиной, из которой исходила заменяемая дуга, и выходной дугой с той вершиной, в которую входила эта дуга. Вводимому переходу ставится в соответствие функция преобразования сигнала, которая была назначена заменяемой дуге.

Формально МСГ может быть представлен кортежем:

$$N^c = \{P^c, T^c, I^c, O^c, F^c\},$$

где $P^c = \{p_i^c\}$ – множество непрерывных позиций;

$T^c = \{t_j^c\}$ – множество непрерывных переходов;

$I^c = \{i^c(p_i^c, t_j^c)\}$ – множество входных дуг переходов, т.е. множество дуг, проведенных от позиций к переходам;

$O^c = \{o^c(t_j^c, p_i^c)\}$ – множество выходных дуг переходов;

$F^c = \{f^c(t_j^c)\}$ – пометочная функция переходов.

В данном случае пометочная функция относится к переходам, а не к дугам, как было ранее и может представлять собой операторы интегрирования, дифференцирования, суммирования и др. Данное графическое представление позволит в дальнейшем использовать процедуру надления переходов функцией «включить-выключить» оператора преобразования либо связи.

С целью маркировки данного графа используется понятие бесконечное число меток a , которым может быть помечена любая входная вершина-позиция (позиция-исток) $p_i^c : o^c(\bullet, p_i^c) = \emptyset$ графа. При наличии бесконечного числа меток a в хотя бы в

одной из входных, по отношению к данному переходу позициям, переход срабатывает и помещает бесконечное число меток в выходную позицию. При этом маркировка входных позиций не меняется, т.е. выполняются следующие правила: $0 - a = 0$, $0 + a = a$, $a + a = a$, $a - a = a$. Отсюда следует определение входной и выходной матрицы:

Определение. Входной A^{lc} (выходной A^{oc}) матрицей МПСГ называется матрица размерностью $(m \times n)$, каждому столбцу $a_{(\bullet)i}^{lc}$ входной матрицы ($a_{(\bullet)i}^{oc}$ выходной матрицы) поставлена в соответствие позиция p_i^c , а каждой соответственно строке $a_{j(\bullet)}^{lc}$ ($a_{j(\bullet)}^{oc}$) – переход t_j^c . Элементами данной матрицы является « a », если переход t_j^c соединен с позицией p_i^c входной (для входной матрицы) или выходной (для выходной матрицы) дугой или «0» в противном случае, т.е.

$$A^{lc} = \left\{ a_{ij}^{lc} = \begin{cases} a, & \text{если } \exists i^c(p_i^c, t_j^c); \\ 0, & \text{иначе;} \end{cases} \right\}, \quad A^{oc} = \left\{ a_{ij}^{oc} = \begin{cases} a, & \text{если } \exists o^c(t_j^c, p_i^c); \\ 0, & \text{иначе;} \end{cases} \right\}.$$

Матрица инцидентий МСГ A^c может быть представлена как разность матриц выходной A^{lc} и входной A^{oc} , при этом поставив вместо элементов « a » входной матрицы элементы « $-a$ ». Таким образом, динамика изменения маркировки полученной сети может быть записана как $M_{i+1}^c(t^+) = M_i^c(t) + A^{cT}U^c(t)$, где $U^c(t)$ – вектор, определяющий какой из переходов должен сработать.

Математически это может быть записано как введение специального оператора преобразования $\vec{F}^{\hat{N}^c \rightarrow N^c}$, который может также рассматриваться как набор из операторов

$$\vec{F}^{\hat{N}^c \rightarrow N^c} = \left\{ \vec{F}^{\hat{P}^c \rightarrow P^c}, \vec{F}^{\hat{Q}^c \rightarrow I^c, T^c, O^c}, \vec{F}^{\hat{F}^c \rightarrow F^c} \right\},$$

где $\vec{F}^{\hat{P}^c \rightarrow P^c}$ – оператор преобразования множества позиций;

$\vec{F}^{\hat{Q}^c \rightarrow I^c, T^c, O^c}$ - оператор преобразования множеств дуг СГ во множества входных дуг, переходов и выходных дуг;

$\vec{F}^{\hat{F}^c \rightarrow F^c}$ - оператор преобразования множества функций сигнала.

Т.е. может быть записано $P^c = \vec{F}^{\hat{P}^c \rightarrow P^c}(\hat{P}^c)$, $I^c = \vec{F}^{\hat{Q}^c \rightarrow I^c}(\hat{Q}^c)$, $T^c = \vec{F}^{\hat{Q}^c \rightarrow T^c}(\hat{Q}^c)$, $O^c = \vec{F}^{\hat{Q}^c \rightarrow O^c}(\hat{Q}^c)$, $F^c = \vec{F}^{\hat{F}^c \rightarrow F^c}(\hat{F}^c)$. Матрица инцидентий МСГ размерностью $(m^c \times n^c)$

может быть также получена из матрицы единичных путей СГ путем применения к ней оператора преобразования $\vec{F}^{\hat{A}^c \rightarrow A^c}$. Она также может быть получена из матрицы единичных путей СГ по следующим формулам: размерность: $m^c = m^{\hat{c}} + m^{\hat{\Sigma}}$,

$$n^c = \sum_{i=1}^{n^{\hat{c}}} \sum_{j=1}^{n^{\hat{c}}} \hat{a}_{ij}^c + n^{\hat{\Sigma}}, \quad \text{где} \quad m^{\hat{\Sigma}} = 2 \sum_{i=1}^{n^{\hat{c}}} \left(\begin{cases} 1, & \text{если } \sum_{j=1}^{n^{\hat{c}}} \hat{a}_{ij}^c > 1 \\ 0, & \text{иначе} \end{cases} \right), \quad n^{\hat{\Sigma}} = \sum_{i=1}^{n^{\hat{c}}} \left(\begin{cases} 1, & \text{если } \sum_{j=1}^{n^{\hat{c}}} \hat{a}_{ij}^c > 1 \\ 0, & \text{иначе} \end{cases} \right) -$$

соответственно компоненты, учитывающие количество переходов, помеченных операторами суммирования. $A^c = \{a_{ij}^c\}$.

2.2. Математическая модель модифицированного сигнального графа

С целью уточнения особенностей гибридных систем целесообразно разработать обобщенную математическую модель ДНАСУ, которая позволила бы уточнить такие понятия как гибридное пространство состояния, оператор изменения непрерывной маркировки и определиться в необходимости использования в модели аналого-цифрового и цифро-аналогового преобразователей [28].

Объединяющим началом для слияния логических и динамических уровней абстракции ДНАСУ является теория алгоритмов. Основополагающими или системными составляющими данных систем являются структура системы и механизм переключения взаимосвязей между элементами системы. Данный механизм поясним на приведенном ниже примере перехода от представления системы СГ с логическими элементами (ключами) к представлению обобщенной ДНАСУ.

Каждому режиму ν однозначно соответствует структура Σ_ν НС части ДНАСУ, а каждой структуре Σ_ν – производящая функция f_ν . В общем случае НС часть может иметь несколько структур $\{\Sigma_i\}_{i=1}^N$. Представим НС традиционным средством – событийным графом с ключами К1 и К2, приведенную на рис. 2.6. Под действием непрерывного управляющего воздействия $u^c(t) = (u_1^c(t), u_2^c(t))^T$ и переключения положения ключей изменяется значение непрерывного состояния $x^c(t) = (x_1^c(t), x_2^c(t))^T$.

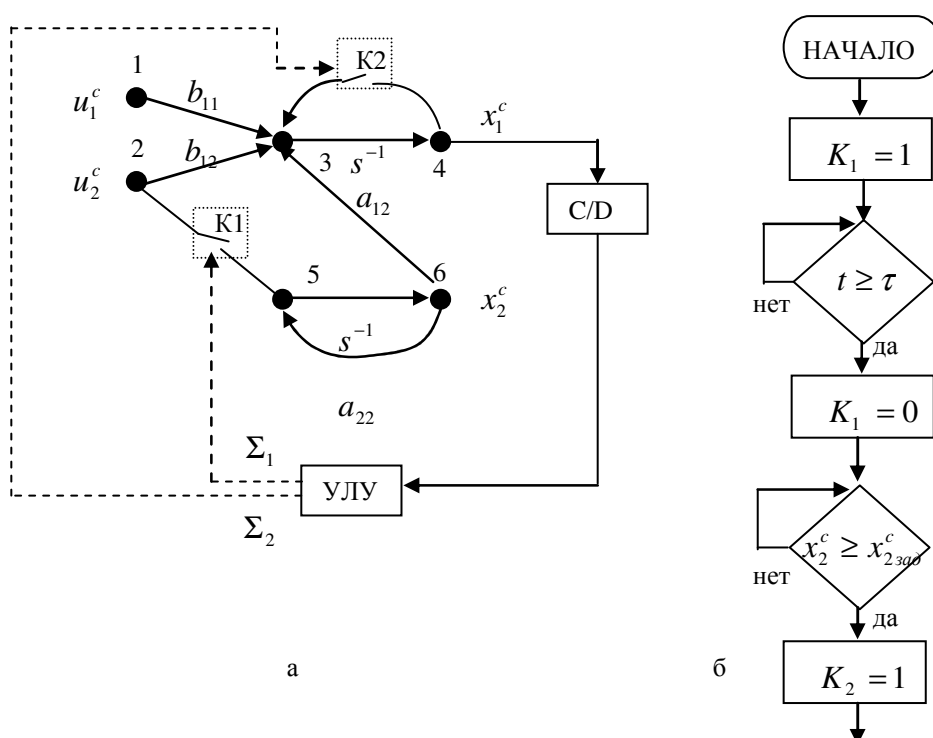


Рисунок 2.6 – Схема и алгоритм управления гибридной системы: а – схема ДНАСУ, б – алгоритм логического управления

Запишем уравнение состояния НС части в структурном состоянии Σ_1 , приведенном на рис. 2.6.

$$\Sigma_1 \rightarrow \dot{x}^c(t) = \left\{ \frac{dx^c(t)}{dt} = f_1[x^c(t), u^c(t), t] \right\} = A_1 x^c(t) + B_1 u^c(t) = \begin{vmatrix} 0 & a_{12} \\ 0 & a_{22} \end{vmatrix} \begin{vmatrix} x_1^c(t) \\ x_2^c(t) \end{vmatrix} + \begin{vmatrix} b_{11} & b_{12} \\ 0 & 0 \end{vmatrix} \begin{vmatrix} u_1^c(t) \\ u_2^c(t) \end{vmatrix}. \quad (2.1)$$

На сигнальном графе могут быть представлены и другие структуры. Наиболее сложно математически описывается формализм, обеспечивающий механизм срабатывания ключей K_i , $i=1,2$, т.е. управление структурой системы.

Эта проблема особенно актуальна при математическом представлении сложных интегрированных систем типа гибких производственных систем (ГПС). Неоднократно подчеркивалось [5, 56], что в математическую модель, описывающую многорежимные процессы функционирования, свойственные ГПС, помимо непрерывного управляющего воздействия $u^c(t)$ должно быть введено дискретное управляющее воздействие $u^d(t_k)$, обеспечивающие изменение структуры системы. В традиционном представлении (2.1) не показан формализм управления структурой. Определим «природу» $u^d(t_k)$ и место его приложения в системе.

Модифицированный СГ (рис. 2.7) получим в результате замены вершин 3, 5 СГ непрерывными переходами t_1^c и t_2^c , реализующими операторы суммирования-интегрирования, ключей К1 и К2 – структурно-управляемыми переходами t_1^{cs} и t_2^{cs} , вершин 1, 2, 4, 6 непрерывными позициями p_i^c , $i=\overline{1,4}$ соответственно. Модель ДНАСУ, состоящая из модифицированного СГ, описывающего НС и ДС части,

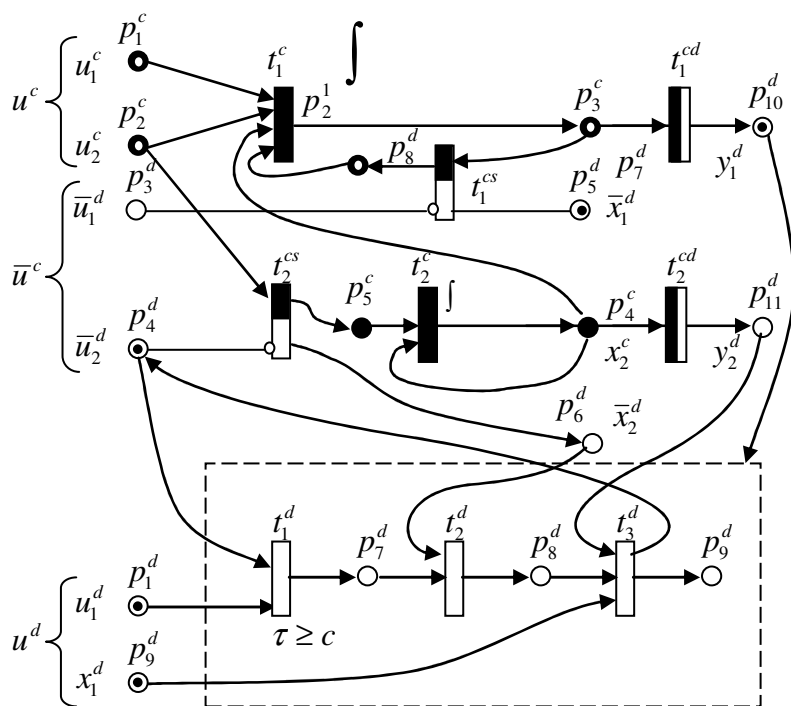


Рисунок 2.7 – Пример ДН-сети ДНАСУ

представленной сетью Петри, изображена на рисунке 2. Для представления ДС части, которая выделена пунктиром, используются дискретные переходы t_i^d и позиции p_j^d . Интерфейс между частями ДНАСУ обеспечивается за счет использования переходов-компараторов t_1^{cd} и t_2^{cd} .

Введем некоторые определения определения модифицированного СГ [28].

Определение. Позиция-исток p_i^c всегда содержит вещественное значение сигнала $u_i^c(t)$, а это значит, что $\mu(p_i^c) = a$, т.е. маркировка позиции $\mu(p_i^c)$ есть метка a . Метка изображается белой точкой на фоне черной непрерывной позиции p_i^c .

Условия срабатывания структурно-управляемого перехода t_i^{cs} приведены в таблице.

Условие срабатывания перехода-компаратора t_i^{cd} есть

$$x_e^c \geq k, \quad e = 1, \dots, r, \quad (2.2)$$

где x_e^c – значение сигнала, которым помечена позиция $p_e^c \in I(t_i^{cd})$;

k – постоянное значение, которым помечен переход t_i^{cd} .

В результате срабатывания t_i^{cd} появляется маркер в позиции $p_r^d \in O(t_i^{cd})$, т.е. $\mu(p_r^d) = 1$.

Переходы t_i^c и t_i^{cs} могут быть помечены операторами преобразования непрерывных сигналов (суммирование, интегрирование, передаточная функция, уравнение состояния и др.).

В данном примере можно выделить: непрерывное состояние $x^c(t) = (x_1^c(t), x_2^c(t))$, дискретное состояние $\bar{x}^d(t_k) = (\bar{x}_1^d(t_k), \bar{x}_2^d(t_k))^T = (\mu(p_5^d), \mu(p_6^d))^T$, непрерывное управляющее воздействие $u^c(t) = (u_1^c(t), u_2^c(t))$ и внутреннее дискретное управляющее воздействие $\bar{u}^d(t_k) = (\bar{u}_1^d(t_k), \bar{u}_2^d(t_k))^T = (\mu(p_3^d), \mu(p_4^d))^T$.

Внутреннее воздействие $\bar{u}^d(t_k)$ обеспечивает управление структурой $\{\Sigma_i\}_{i=1}^N$ НС части и формируется ДС частью на основе внешнего дискретного управляющего воздействия $u^d(t_k) = (u_1^d(t_k), u_2^d(t_k))^T = (\mu(p_1^d), \mu(p_2^d))^T$, состояния $\bar{x}^d(t_k)$ НС части и сигнала обратной связи $y^d(t_k) = (y_1^d(t_k), y_2^d(t_k))^T = (\mu(p_{10}^d), \mu(p_{11}^d))^T$. Правила срабатывания переходов t_i^d ДС части – такие же как для сетей Петри.

Запишем уравнение непрерывно-событийного состояния НС части

$$\frac{dx^c(t)}{dt} = A\Xi_1(\bar{u}^d(t_k))x^c(t) + B\Xi_2(\bar{u}^d(t_k))\mu^c(t) \quad (2.3)$$

$$x^c(t_0) = x_0^c; \quad \bar{u}^d(t_0) = \bar{u}_0^d$$

где $\Xi_i = \left| \xi_1^i, \xi_2^i \right|^T$ – вектор-функция управления структурой, $i=1,2$, $\xi_{i(\cdot)} = \{E, 0\}$;

$E, 0$ – единичная и нулевая матрицы размерностью (2×2) ;

$\xi_1^i \cap \xi_2^i = \mathbf{0}$, $\xi_1^i \cup \xi_2^i = E$ – условие единственности и полноты;

$A = |A_1, A_2|$; $B = |B_1, B_2|$ – постоянные матрицы:

$$A_2 = \begin{vmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{vmatrix}, \quad B_2 = \begin{vmatrix} b_{11} & b_{12} \\ 0 & b_{22} \end{vmatrix}.$$

Матрицы A_1 и B_1 определены выше.

Изменения элементов вектора-функции Ξ_i в зависимости от значения внутреннего управляющего воздействия $\bar{u}^d(t_k)$ приведены в таблице 2.1.

Таблица 2.1 – Изменения элементов вектора-функции Ξ_i в зависимости от значения $\bar{u}^d(t_k)$

$\bar{u}^d(t_k) = (\mu(p_3^d), \mu(p_4^d))^T$	ξ_1^1	ξ_2^1	ξ_1^2	ξ_2^2
$(0,1)^T$	E	0	E	0
$(0,0)^T$	E	0	0	E
$(1,0)^T$	0	E	0	E
$(1,1)^T$	0	E	E	0

Выполнение действий в правой части уравнения (2.3) происходит в следующем порядке, в начале выполняется умножение матрицы A на функцию Ξ_i , полученный результат умножается на $x^c(t)$. Например,

$$A\Xi_1(\bar{u}^d(t_k) = (0,1)^T)x^c(t) = \begin{vmatrix} 0 & a_{12} & : & a_{11} & a_{12} \\ 0 & a_{22} & : & 0 & a_{22} \end{vmatrix} \times \begin{vmatrix} E \\ 0 \end{vmatrix} \times x^c(t) = \begin{vmatrix} 0 & a_{12} \\ 0 & a_{22} \end{vmatrix} \times \begin{vmatrix} x_1^c(t) \\ x_2^c(t) \end{vmatrix}$$

Уравнение состояние ДС части можно представить в следующем виде

$$x^d(t_k) = x^d(t_{k-1}) + H\delta^d(t_k) + u^d(t_k) + y^d(t_k), \quad (2.4)$$

где $x^d(t_k) = u^d(t_k) \cup \bar{u}^d(t_k) \cup \bar{x}^d(t_k) \cup \theta^d(t_k) \cup y^d(t_k)$ – дискретное состояние ДС подсистемы;

$u^d(t_k) = (u_1^d(t_k), u_2^d(t_k))^T = (\mu(p_1^d), \mu(p_2^d))^T$ – дискретное управляющее воздействие;

$\theta^d(t_k) = (\theta_1^d(t_k), \theta_2^d(t_k), \theta_3^d(t_k))^T = (\mu(p_7^d), \mu(p_8^d), \mu(p_9^d))^T$ – вектор состояния внутренних позиций p_c^d ДС части;

$x^d(t_k) = (\mu(p_1^2), \mu(p_2^2), \dots, \mu(p_{11}^d))^T$ – дискретное состояние ДНАСУ, представленной на рис. 2.6;

$H = |h_1, h_2, h_3|$ – матрица инцидентности дискретных позиций p_i^d , $i=1, \dots, 11$ и дискретных переходов t_j^d , $j=1, \dots, 3$ с вектор-столбцами h_j , соответствующих модификациям маркировки $\{1, -1, 0\}$;

$\delta^d(t_k) = |\delta_1^d, \delta_2^d, \delta_3^d|^T$ – вектор управления $\delta_i^d \cap \delta_j^d = 0$; $\delta_i^d \cup \delta_j^d = 1$ для $i \neq j$.

Изменение с состояния $x^d(t_{k-1})$ на состояние $x^d(t_k)$ может произойти в результате срабатывания одного из переходов t_1^d , t_2^d , t_3^d (соответствующим образом формируется вектор управления $\delta^d(t_k)$) или под действием $u^d(t_k)$ и $y^d(t_k)$. Условие срабатывания дискретного перехода $t_{(.)}^d$ такое же как и в СП.

В столбцах матрицы G , соответствующих переходам типа $t_{(.)}^{cs}$, значащие элементы разделены по диагонали на верхнюю и нижнюю части. В верхней части приведены элементы матрицы G_o , в нижней – G_s . Первые два столбца являются общими для указанных матриц. Выделенные элементы на пересечении столбца,

помеченного переходом t_i^{cs} , и строки, помеченной позицией p_j^d , указывают на то, что p_j^d является входной позицией для перехода t_i^{cs} , но связана с ним ингибиторной дугой, а значения в них есть алгебраическое действие над маркером позиции p_j^d . Для того, чтобы «открыть» переход t_i^{cs} необходимо вычесть из позиции p_i^d маркер, в противном случае – добавить в нее маркер в результате срабатывания переходов t_i^d в ДС части (формирование вектора $\bar{u}^d(t_k) = (\mu(p_3^d), \mu(p_4^d))^T$).

$G =$

$-\alpha$				p_1^c
$-\alpha$			$-\alpha$ $+\alpha$	p_2^c
$+\alpha$		$-\alpha$ $+\alpha$		p_3^c
	$+\alpha$			p_4^c
	$-\alpha$		$+\alpha$ 0	p_5^c
		$+\alpha$ 0		p_6^c
		-1 $+1$		p_3^d
			-1 $+1$	p_4^d
		$+1$ 0		p_5^d
			$+1$ 0	p_6^d

2.3. Расширение теоретико-множественного представления ДНАСУ

2.3.1. Переходы – результаты наложения элементарных сетей

ДНАСУ является динамической системой, которая функционирует на основе использования информации о событиях, ситуациях, процессах (тепло-массо переносе и преобразовании энергии), происходящих внутри, а также вне рассматриваемой системы. Пространством состояния ДНАСУ есть объединение $X_{ГС}^c \cup X_{ГС}^d$, где $X_{ГС}^c$ и $X_{ГС}^d$ соответственно векторы состояний НС части и ДС части. В процессе функционирования ДНАСУ реализуется взаимодействие как между подсистемами, так и с внешней непрерывной системой S_i^c и с дискретной системой $S_{i=1}^d$. Математическая модель такого взаимодействия может быть доопределена вновь введенными переходами, реализующими операторы $a^{(\bullet\bullet)}(t)$, отображающие подмножество входных сигналов на подмножество выходных сигналов. При этом входные и выходные сигналы могут относиться как к одной форме представления, так и к разным формам представления. Верхние индексы перехода указывают формы представления (аналоговая, дискретная) входного и выходного сигналов. Схема отображения различной формы сигналов интерфейса приведена на рис. 2.8.

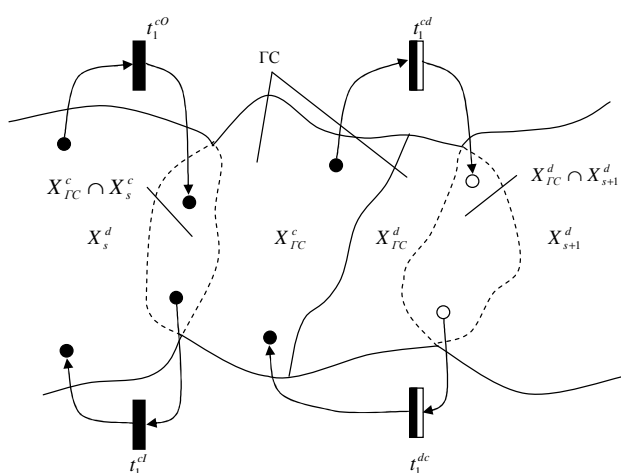


Рисунок 2.8 – Схема отображения сигналов при взаимодействии ДНАСУ с другими системами

Прежде чем внести средства отображения различных форм сигналов определим МСГ, дополним его переходами, реализующими различные функции интерфейса, и сетями, полученными в результате выполнения операций наложения элементарных сетей (фрагментов сетей).

Для получения модифицированной ДНС объединим сначала МСГ, полученный ранее с ДСП. Рассмотрим различные операции таких объединений.

Определим операцию наложения как операцию теоретико-множественного объединения графов, дополненного правилом формирования разметки, при помощи которой одна сеть накладывается на другую [19]. Необходимо отметить, что операции наложения могут быть двух типов [19]. К первому типу относятся операции наложения, производимые над элементарными сетями различных типов, в результате которых образуются смешанные виды элементарных сетей. Ко второму типу относятся операции наложения, производимые над несколькими сетями одного типа, помеченные одинаковыми операторами преобразования, результатом которых является некоторая составная сеть. Для этого введем понятия элементарная сеть и ее фрагмент, примеры которого приведены в таблице 2.2.

Определение. Элементарной сетью $N^{(\bullet)e}$ будем называть такую сеть $N^{(\bullet)}$, которая содержит только один переход $T^{(\bullet)e} = \{t_j^{(\bullet)}\}$, $j=1$ соответствующего вида и две позиции $P^{(\bullet)e} = \{p_i^{(\bullet)}\}$, $i=1,2$, одна из которых $p_1^{(\bullet)}$ является входной, а другая $p_2^{(\bullet)}$ – выходной для данного перехода, т.е.:

$$I^{(\bullet)}(p_1^{(\bullet)}, t_1^{(\bullet)}) \neq \emptyset \text{ или } \hat{I}^{(\bullet)}(p_1^{(\bullet)}, t_1^{(\bullet)}) \neq \emptyset; O^{(\bullet)}(t_1^{(\bullet)}, p_2^{(\bullet)}) \neq \emptyset; I^{(\bullet)}(p_2^{(\bullet)}, t_1^{(\bullet)}) = \emptyset; O^{(\bullet)}(t_1^{(\bullet)}, p_1^{(\bullet)}) = \emptyset.$$

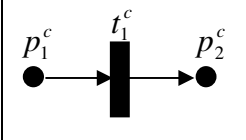
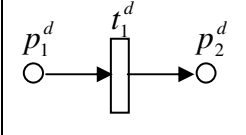
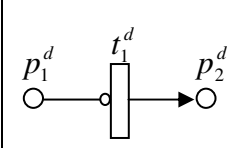
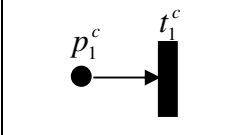
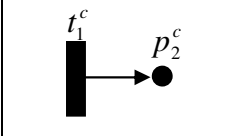
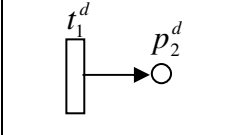
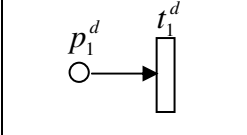
Данное определение является общим для СГ и МСГ.

Определение. Фрагмент элементарной сети есть подсеть элементарной сети, состоящая из одного перехода $T^{(\bullet)e} = \{t_j^{(\bullet)}\}$, $j=1$ и одной позиции $P^{(\bullet)e} = \{p_i^{(\bullet)}\}$, $i=1$.

Примеры элементарных сетей и их фрагментов приведены в таблице 2.2.

Опишем формирование некоторых операций над элементарными сетями и их фрагментами.

Таблица 2.2 – Примеры элементарных сетей и их фрагментов

Тип элементарной сети (фрагмента)	Граф сети	Обозначение сети
Непрерывная сеть		N^{ce}
Сеть Петри		N^{de}
Сеть Петри с ингибиторной дугой		\hat{N}^{de}
Фрагмент N^{ce} сети		N^{cle}
Фрагмент N^{ce} сети		N^{cOe}
Фрагмент N^{de} сети		N^{dOe}
Фрагмент N^{de} сети		N^{dle}

Определение. Операцией наложения элементарной непрерывной сети N^{ce} на элементарную дискретную сеть \hat{N}^{de} (со входной ингибиторной дугой) называется такая операция, результатом которой является элементарная структурно-управляемая сеть $N^{cse} = N^{ce} \cap \hat{N}^{de}$.

Характеристики сети N^{cse} , полученной в результате наложения приведены в таблице 2.2.

Полученный в результате такой операции структурно-управляемый переход t_j^{cs} может быть помечен оператором преобразования $f(u, t)$ входного непрерывного

сигнала $u_i^c(p_1^c)$ в выходной непрерывный сигнал $x_i^c(p_2^c)$, которыми помечены соответственно вершины-позиции p_1^c и p_2^c . Данный переход срабатывает только в случае отсутствия маркеров во всех входных дискретных позициях p_k^d (так как они соединены с ним ингибиторными дугами) и наличия метки α во входных непрерывных позициях p_i^c , если оператор $f(u,t)$ нелинейный (при использовании линейного оператора, в частности, суммирования, это требование отсутствует). При его срабатывании в выходные непрерывные p_r^c и дискретные p_s^d позиции поступает соответственно бесконечное число α и количество маркеров, равное весу выходной дуги $f^d(o^{cs}(t_j^{cs}, p_s^d))$, а при его закрытии – изымаются оттуда. Правила срабатывания структурно-управляемого перехода приведены в таблице 2.3.

2.3.2. Переходы – результат наложения фрагментов сетей

Исходя из цели объединения расширенной СП и МСГ дополним последний новыми переходами, реализующими релейные (переключательные) функции.

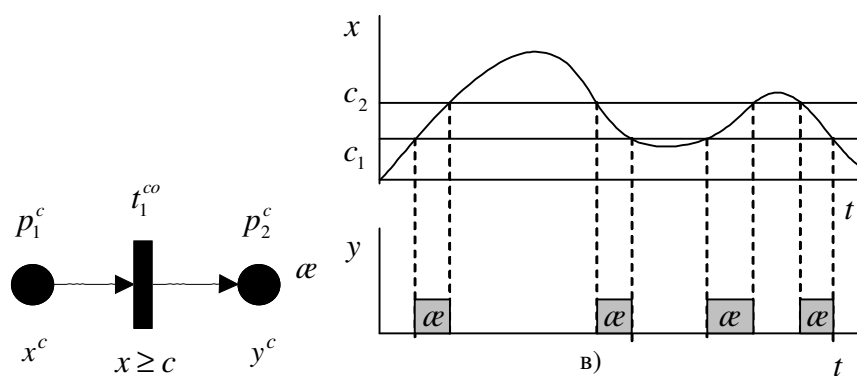
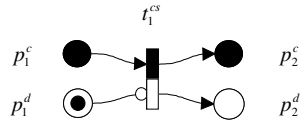
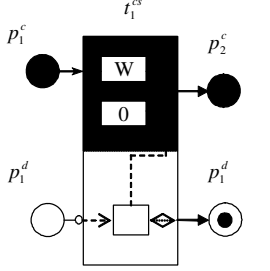
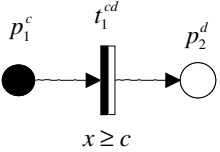
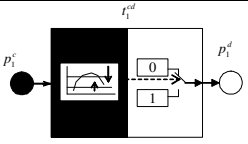
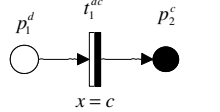
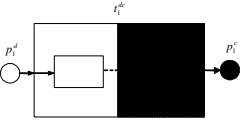


Рисунок 2.9 – Входной переход и его работа: а – представление входного перехода; б – работа входного перехода

Таблица 2.3 – Операции наложения

Операции наложения	Результирующая сеть	Правила срабатывания	Внутренняя структура перехода	Матрица инцидентности и правила определения элементов
<p>Сеть N^{ce} на сеть \hat{N}^{de}</p>		<p>Условия срабатывания</p> $\forall p_k^d \in \hat{I}^{cs}(t_j^{cs}): \mu_i^d(p_k^d) < f^d(\hat{i}^d(p_k^d, t_j^{cs}))$ $\forall p_l^c \in \tilde{I}^{cs}(t_j^{cs}): \mu_i^c(p_l^c) = \alpha.$ <p>Новая разметка</p> $\forall p_k^d \in O^{cs}(t_j^{cs}): \mu_{i+1}^d(p_k^d) = f^d(O^d(t_j^{cs}, p_k^d))$ $\forall p_l^c \in O^{cs}(t_j^{cs}): \mu_{i+1}^c(p_l^c) = \alpha$		$A^{cs} = \begin{matrix} & P^d & & P^c & & P^{cs} \\ P^d & \begin{bmatrix} A^d & \emptyset & A^{lcs} & A^{ocs} \end{bmatrix} \\ P^c & \begin{bmatrix} \emptyset & A^c & A^{lcs} & A^{ocs} \end{bmatrix} \end{matrix}$ $A^{lcs} = \left\{ a_{ij}^{lcs} = \begin{cases} \alpha, & \text{если } \exists i^c(p_i^c, t_j^{cs}); \\ 1, & \text{если } \exists \hat{i}^d(p_i^d, t_j^{cs}); \\ 0, & \text{иначе;} \end{cases} \right\},$ $A^{ocs} = \left\{ a_{ij}^{ocs} = \begin{cases} \alpha, & \text{если } \exists o^c(t_j^{cs}, p_i^c); \\ 1, & \text{если } \exists o^{cs}(t_j^{cs}, p_i^d); \\ 0, & \text{иначе;} \end{cases} \right\}$

Продолжение таблицы 2.3

Операции наложения	Результирующая сеть	Правила срабатывания	Внутренняя структура перехода	Матрица инцидентности и правила определения элементов
<p>Фрагмент сети N^{coe} на фрагмент сети N^{de}</p>		<p>Условия срабатывания</p> $\forall p_i^c \in I^{cd}(t_j^{cd}): \mu_i^c(p_i^c) = \alpha; f(u_i^c(p_i^c), t) = 1$ <p>Новая разметка</p> $\forall p_l^d \in O^{cd}(t_j^{cd}): \mu_{l+1}^d(p_l^d) = 1$		$A^{cd} = \begin{matrix} & T^d & T^c & T^{cd} \\ \begin{matrix} P^d \\ P^c \end{matrix} & \begin{bmatrix} A^d & \emptyset & A^{Ocd} \\ \emptyset & A^c & A^{Icd} \end{bmatrix} \end{matrix}$ $A^{lcd} = \left\{ a_{ij}^{lcd} = \begin{cases} \alpha, & \text{если } \exists i^c(p_i^c, t_j^{cd}); \\ 0, & \text{иначе;} \end{cases} \right\},$ $A^{ocd} = \left\{ a_{ij}^{ocd} = \begin{cases} 1, & \text{если } \exists o^c(t_j^{cd}, p_i^d); \\ 0, & \text{иначе;} \end{cases} \right\}$
<p>Фрагмент сети N^{de} на фрагмент сети N^{cie}</p>		<p>Условия срабатывания</p> $\forall p_k^d \in I^{dc}(t_j^{dc}): \mu_i^d(p_k^d) \geq f^d(\hat{i}^d(p_k^d, t_j^{dc}))$ <p>Новая разметка</p> $\forall p_k^d \in I^{dc}(t_j^{dc}): \mu_{i+1}^d(p_k^d) = \mu_i^d(p_k^d) - f^d(\hat{i}^d(p_k^d, t_j^{dc}))$ $\forall p_k^d \in I^{dc}(t_j^{dc}): \mu_{i+1}^d(p_k^d) = \alpha$		$A^{dc} = \begin{matrix} & T^d & T^c & T^{dc} \\ \begin{matrix} P^d \\ P^c \end{matrix} & \begin{bmatrix} A^d & \emptyset & A^{Odc} \\ \emptyset & A^c & A^{I dc} \end{bmatrix} \end{matrix}$ $A^{l dc} = \left\{ a_{ij}^{l dc} = \begin{cases} 1, & \text{если } \exists i^d(p_i^d, t_j^{dc}); \\ 0, & \text{иначе;} \end{cases} \right\},$ $A^{ocs} = \left\{ a_{ij}^{ods} = \begin{cases} \alpha, & \text{если } \exists o^c(t_j^{dc}, p_i^c); \\ 0, & \text{иначе;} \end{cases} \right\}$

Введем переход t_j^{co} , помеченный функцией $f^{co}(x^c) = y^c = \begin{cases} h, & \text{если } x^c \geq c; \\ 0, & \text{иначе.} \end{cases}$, где x^c –

переменная, которой помечена входная позиция p_i^c ; c – порог, при равенстве или превышении которого в выходной позиции формируется постоянное значение сигнала, равное h (рис. 2.9.а). Данный тип перехода может быть использован как генератор постоянного структурного значения сигнала y^c амплитуды сигнала h , модулированного во времени в зависимости от входного сигнала. При этом разметка формируется по следующим правилам:

$\mu_s^c(p_k^c) = \begin{cases} \alpha, & \text{если } \mu_s^c(p_i^c) = \alpha, \forall p_i^c : i^c(p_i^c, t_j^{co}) \neq \emptyset, \text{ и } f^{co}(t_j^c) = 1 \\ 0, & \text{иначе} \end{cases}$. Данный переход может быть

использован при формировании широтно-импульсной модуляции. Подобно данной, получим разметку (рис. 2.9) при реализации пометочной функции

$f^{co}(x^c) = \begin{cases} h, & \text{если } c_1 \leq x^c \leq c_2; \\ 0, & \text{иначе.} \end{cases}$. В данном случае множество пометочных функций F

расширяется за счет функций F^{co} , а множество переходов T – за счет переходов T^{co} .

Таким образом, имеем: $N^{co} = \{P^c, T^c, T^{co}, F^c, F^{co}, I^c, O^c, I^{co}, O^{co}\}$, где I^{co} , O^{co} – множество входных и выходных дуг, соединенных со вновь введенным типом переходов.

Выше были введены выходные переходы, позволяющие выдавать состояние системы, представленного МСГ, для внешнего устройства логического управления. Теперь поставим обратную задачу: пользуясь изменением маркировки МСГ, изменять значения входных сигналов. Введем своеобразный входной переход T^{ci} , помеченный специальной функцией F^{ci} , который при подаче на вход сигнала в виде появления бесконечного числа меток во входной позиции, помимо выдачи маркировки в выходную позицию, мог бы изменять значение выходного сигнала на некоторую величину $c \neq 0$, служа при этом своеобразным счетчиком изменений входной маркировки $\alpha - 0 - \alpha - \alpha - \dots$. Т.е.,

$f_j^{ci} : y(t_{k+1}) = \begin{cases} y(t_k) + c, & \text{если } \mu_k^c(p_i^c) = \alpha, \forall p_i^c : i^c(t_j^{ci}, p_i^c) \neq \emptyset, \\ y(t_k), & \text{иначе} \end{cases}$. Маркировка выходных позиций

для данного типа переходов $\mu_s^c(p_i^c) = \alpha, \forall p_i^c : o^c(t_j^{ci}, p_i^c) \neq \emptyset$.

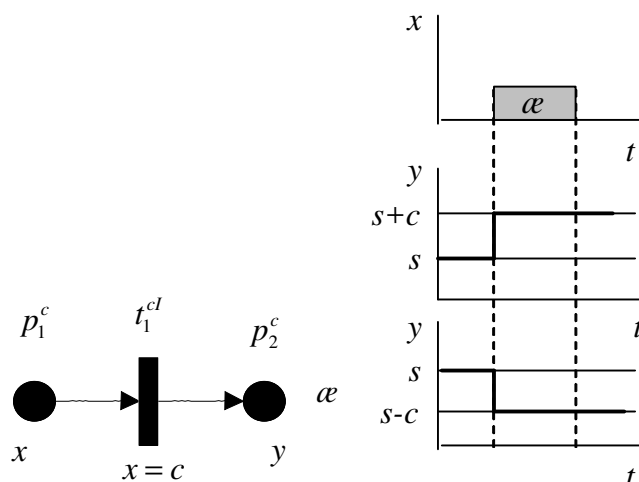


Рисунок 2.10 – Входной переход и его работа: а – представление входного перехода; б – работа входного перехода

Работа данного перехода представлена на рис. 2.10. Тогда, МСГ со входными переходами будет иметь следующий вид: $N^{cl} = \{P^c, T^c, T^{cl}, F^c, F^{cl}, I^c, O^c, I^{cl}, O^{cl}\}$, где I^{cl} , O^{cl} – множество входных и выходных дуг, соединенных с входными переходами. Переходы помечены разного вида функциями f^{co} в другом F^{cl} .

2.4. Формализованное представление расширенной ДНАСУ

Структурная схема ДНАСУ, отражающая различные математические схемы представления НС части системы, приведена на рис. 2.11. В данном случае ДС подсистема определена использованием математической схемы СП. Стратификация НС подсистемы проведена путем рассмотрения различных особенностей подсистемы [36].

Запишем математическую схему первой страты НС1 подсистемы, используя матрично-дифференциальное представление, дополненное механизмом изменения правой части системы уравнений:

НС подсистема:

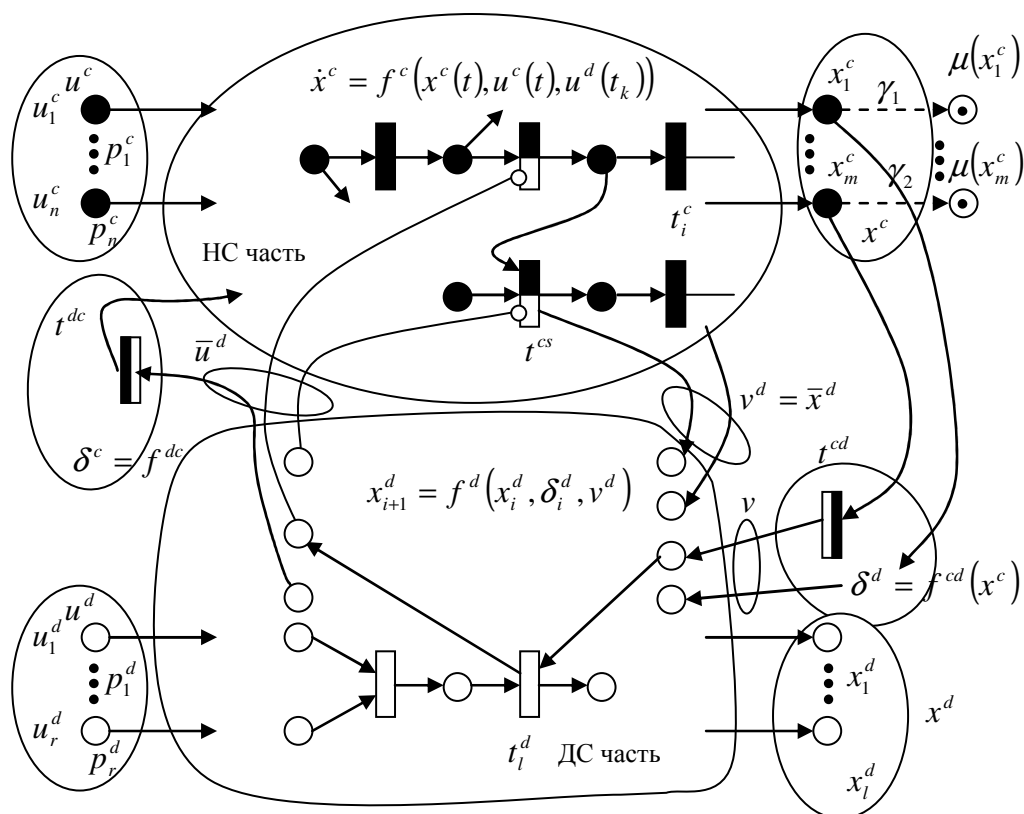


Рисунок 2.11 – Пример ДНАСУ

$$\dot{x}^c(t) = (A \Xi^x(\bar{u}^d(t_k)) x^c(t) + b \Xi^b(t_k)) + B \Xi(\bar{u}^d(t_k)) u^c(t), \quad (2.5)$$

$$y^c(t) = C x^c(t), \quad (2.6)$$

$$x^c(t) = x^c(t_0) = x_0^c, \quad u^d(t_0) = \bar{u}_0^d, \quad u^c(t_0) = u_0^c, \quad y^c(t_0) = y_0^c;$$

где $u^c(t) \in R^m = \bigcup_{i=1}^N \bar{W}_i$ – пространство непрерывного входа входного воздействия;

$u^d(t_k) \in Z^w = \bigcup_{i=1}^N \bar{Z}^w_i$ – пространство дискретного входного воздействия;

$x^c(t) \in R^n = \bigcup_{i=1}^N \bar{X}_i$ – пространство состояния непрерывной составляющей;

$y^c(t) \in R^r = \bigcup_{i=1}^N \bar{Y}_i$ – пространство непрерывного выхода;

$\Xi(t_k): DHC_o^i \rightarrow f_i: (A_i^o, B^o, C^o)$, оператор, преобразующий структурное состояние

Σ_i ДН-сети в производящий оператор f_i $i=1,2,\dots,N$, $\Xi(t_k) = \Phi(y_o^d(t_k))$;

$\Xi(t_k) = |\xi_1^j, \xi_2^j, \dots, \xi_N^j|^T$, $j=1,2,\dots,N$, $\xi_i^j = \{E, 0\}$, $\xi_i^j \cap \xi_k^j = \emptyset$, $\xi_i^j \cup \xi_k^j = 1$ для $i \neq j$.

Ступенчатое изменение структуры НС1 подсистемы обеспечивает ДС подсистема, представленная уравнением состояния СП с учетом введенных параметров τ^d , s^d , расширяющих алгоритмические свойства ДС подсистемы:

ДС подсистема:

$$x^d(t_k) = x^d(t_{k+1}) + H\delta^d(t_k, \tau^d, s^d) + u^d(t_k, \tau^d, s^d), \quad (2.7)$$

$$\delta^d(t_k, \tau^d, s^d) = \Delta(u^d(t_k, \tau^d, s^d), \eta^d(t_k)), \quad (2.8)$$

$$x^d(t_0) = x_0^d, \quad u^d(t_0) = u_0^d,$$

где $x^d(t_k) \in Z^X = \bigcup_{i=1}^N \bar{Z}^X_i$ – пространство состояния дискретной составляющей,

зависящей от маркировки дискретных позиций $x_i^d(t_k = \mu(p_i^d))$;

$U = R^m \times Z^W$ – расширенное (гибридное) пространство входа;

$u(t, |t_k|) = (u(t), u^d(t_k))$ – гибридное входное воздействие;

$X = R^n \times Z^X$ – расширенное состояние;

$x(t, |t_k|) = (x(t), x^d(t_k))$ – расширенное (гибридное) состояние;

$y^d(t_k) \in Z^Y = \bigcup_{i=1}^N \bar{Z}^Y_i$ – пространство дискретного выхода;

$Y = R^r \times Z^Y$ – расширенное пространство выхода;

$y(t, |t_k|) = (y(t), y^d(t_k))$ – расширенный (гибридный) выход;

H – матрица инцидентности ДС части с элементами $h_{ij} = \{+1, -1, 0\}$;

τ^d – время задержки срабатывания $\tau^d = (\tau_1^d, \dots, \tau_h^d)$ перехода типа $t_i^d \in T^d$,

$$\left. \begin{array}{l} \tau_1^d = \zeta_1, \\ \vdots \\ \tau_h^d = \zeta_h, \end{array} \right\}; \quad (2.9)$$

s^d – время, в течении которого могут быть реализованы условия срабатывания перехода $s^d = (s_1^d, \dots, s_h^d)$

$$\left. \begin{array}{l} s_1^d \in [t_1^{разр}, t_1^{запр}] \\ \vdots \\ s_1^d \in [t_h^{разр}, t_h^{запр}] \end{array} \right\}; \quad (2.10)$$

$\delta^d(t_k, \tau^d, s^d)$ – управляющий сигнал для переходов ДС подсистемы;

$\eta^d(t_k) = f_v^d(x^d(t_k), v^d(t_k)) \in \{0,1\}$ – обобщенное входное воздействие на ДС подсистему (управляющий сигнал для переходов $t_k^{(*)}$);

$v^d(t_k) = V(h^d(t_k), \bar{x}_H^d(t_k))$ – внутренний управляющий сигнал срабатывания НС подсистемы переходов $T \in T^c \cup T^{cs} \cup T^{cd}$:

$$v_i^d = \begin{cases} 1, & \text{если } x(t) \in \bar{X}_i, \\ \text{иначе } 0, \end{cases};$$

$h^d(t_k) = f(x^c(t)) \in \{0,1\}$ – предикатная функция, реализующая интерфейс обратной связи С/D.

$\bar{u}^d(t_h) = \pi(x^d(t_h))$; где π – оператор проекции $x^d(t_h)$ в $\bar{u}^d(t_h)$.

В зависимости от текущего значения дискретной составляющей $x^d(t_k)$ (2.7), которое формируется воздействием $u^d(t_k)$, вырабатывается управляющее воздействие $\bar{u}^d(t_k)$, обеспечивающее «выбор» структуры правой части (2.5) обратной связи вектора начального состояния $x^c(t_0)$; формируется текущее состояние $x(t, t_k) = (x^c(t_0), x^d(t_k))$.

Модель ДНАСУ может быть представлена путем объединения топологического и абстрактно-алгебраических уровней представления системы. В данном подходе с использованием абстрактно-алгебраической схемы (2.5) – (2.8) представим НС подсистему. Применение теоретико-множественной схемы (топологической), в частности, ДН-сети, позволяет представить ДНАСУ в целом. Рассмотрим кортеж, в котором приведены элементы вышеназванного объединения:

$$\Gamma = \langle \Sigma, R, H, \bar{F}, M_0, \Pi, time, tdugi \rangle, \quad (2.11)$$

где Σ – абстрактно-алгоритмический уровень ДНАСУ, представленный системами уравнений (2.5) – (2.10);

$R = \langle P, T, I, O \rangle$ – ДН-сеть;

P – конечный набор мест;

T – конечный набор переходов;

$P \cap T = 0$ и $P \cup T = 0$;

$I(p_i, t_j)$ – функция, которая определяет дуги от места до перехода;

$O(p_i, t_j)$ – функция, которая определяет дуги от перехода до места;

$H : P \cup T \rightsquigarrow \{d, c, cs, cd, dc, \Pi\}$ – оператор, названный назначающей функцией, указывающий для каждого узла, является он дискретным, непрерывным, непрерывно-дискретным первого типа, непрерывно-дискретным второго типа, дискретно-непрерывным первого типа, дискретно-непрерывным второго типа соответственно, а также каким оператором (сигналом) он помечен;

M_0 – начальная маркировка;

Π – оператор преобразования ДНАСУ, представленной системой уравнений (2.5) – (2.8) в теоретическо-множественное представление в виде ДН-сети;

\bar{F} – матрица инцидентности позиций и переходов НС части ДНАСУ;

$Time(t)$ – функция назначения времени τ^d , которая связывает рациональное число или задержку срабатывания каждого перехода;

$Time(p, t)$ – связывает рациональное число (время s^d) с каждой дугой.

Виды переходов, позиций и дуг, используемых в расширенной ДН-сети, приведены в таблице 2.4 [20]. В ней индексом * обозначены вновь введенные элементы. При этом время задержки τ и промежутки готовности срабатывания $[t_{нач}, t_{кон}]$ вновь введенного перехода с задержкой и ограничением времени срабатывания определяются константами либо для стохастического подвида дискретного перехода могут определяться стохастической функцией $f^{stoh}(t)$.

Таблица 2.4 – Виды переходов, позиций и дуг в расширенной ДН-сети

t_1^c 	Непрерывный переход	p_1^c 	Непрерывная позиция
t_1^d 	Дискретный переход	p_1^d 	Дискретная позиция
t_1^d  τ	Дискретный переход с запаздыванием срабатывания*	t_1^d  $[t_{нач}, t_{кон}]$	Дискретный переход с ограничением по времени срабатывания*
t_1^{cl}  $x \geq c$	Входной непрерывный переход*	t_1^{co}  $x = c$	Выходной непрерывный переход*
t_1^{cs} 	Структурно-управляемый переход	t_1^{dc}  $x = c$	Дискретно-непрерывный переход*
t_1^{cd}  $x \geq c$	Непрерывно-дискретный переход* первого типа	t_1^{cd}  $c_1 \leq x \leq c_2$	Непрерывно-дискретный переход* второго типа
	Нормальная дуга		Ингибиторная дуга
	Макро переход*	i_1^1 	Исток в макропереходе*

Запишем условия формирования структуры НС подсистемы. Условия формирования элементов \bar{f}_{ij} матрицы \bar{F} :

$$\bar{f}_{ij} = \begin{cases} +1, \text{ если } p_i^d \in O(t_j^d), \\ -1, \text{ если } p_i^d \in I(t_j^d), \\ -\alpha, \text{ если } p_i^c \in I(t_j^c) \text{ или } p_i^c \in I(t_j^{cs}), \\ +\alpha, \text{ если } p_i^c \in O(t_j^c) \text{ или } p_i^c \in O(t_j^{cs}), \\ 0, \text{ если нет связи между вершинами.} \end{cases}$$

Условия формирования управляющего сигнала η^d :

$$\eta_i^d = \begin{cases} 1, \text{ если } \begin{cases} \forall p_s^d \in I^d(t_j^d): \mu(p_s^d) = 1, \\ \forall p_s^c \in I^{cs}(t_j^{cs}): \mu(p_s^c) = \alpha \text{ и } O_t(p_i^d, t_j^{cs}) = 0, \end{cases} \\ \text{иначе } 0, \end{cases}$$

Введенная ранее разметка непрерывных позиций $\mu(p_i^c) \in \{\emptyset, \alpha\}$, а также наличие в НС подсистеме структурно-управляемых переходов $t_j^{cs} \in T^{cs}$, обеспечивающих формирование последовательности структур $\Sigma_1 \propto \Sigma_2 \propto \dots \Sigma_N$, и, тем самым, перемещение меток α по позициям $p_i^c \in P^c$, позволили записать уравнение их динамики:

$$\bar{x}_1^d(t_{k+1}) = \bar{x}_1^d(t_k) + \bar{F}_o v_1^d(t_k) + u_o^d(t_k),$$

где $\bar{x}_1^d(t_k) \in Z^\alpha = \bigcup_{i=1}^k Z_i^\alpha$ – пространство состояний.

В зависимости от значений текущего состояния дискретной составляющей состояния $x^d(t_k) = M_k(p_i^{(*)}) \in \{\mathcal{X}, 0, N\}$, $i = 1, 2, \dots, R$ вектора управления $U = (u^c(t), u^d(t_k))$; вектор-функции управления структурой $\Xi(u_o^d(t_k))$; внешних управляющих воздействий $W^d(t_k)$; топологии ДН-сети, представляющей ОУсУС и описанной матрицей инцидентности \tilde{A}_0 ; вектора начального состояния $x(t_0) = (x^c(t_0), x^d(t_0))$, формируется текущее состояние $x(t, |t_k|)$.

Таблица 2.5 – Представление маркировки непрерывных позиций с описателями

$p_i^{(*)}$	Имя позиции	Значение сигнала	Метка
1	Температура	20 °С	α
2	Температура	\emptyset	0

Для ДНАСУ предполагается, что $x^c(t) \in R^n = \bigcup_{i=1}^N \bar{x}_i(t)$ – пространство непрерывного состояния, где \bar{x}_i – подпространство, в котором изменяется $x^c(t)$ при условии, что НС часть находится в Σ_i структурном состоянии. Если $x^c(t)$ достигает многообразия переключающего множества $\bar{x}_i \cap \bar{x}_k, \dots, \bar{x}_i \cap \bar{x}_l$ в результате воздействий $\bar{u}^d(t_k)$ и $u^d(t_k)$, то предполагается, что общая граница $\Gamma_{ki} = \Gamma(\bar{x}_k \cap \bar{x}_i)$ двух смежных подпространств \bar{x}_k и \bar{x}_i задается уравнением $\gamma_{ki}(x^c, x^d) = 0$, $k, l = 1, 2, \dots, N$. С целью получения определенной модели общая граница $\Gamma(\bar{x}_i \cap \bar{x}_k, \dots, \bar{x}_j \cap \bar{x}_l)$ нескольких смежных переключающих множеств $\bar{x}_i, \bar{x}_k, \dots, \bar{x}_l$ назначается исключительно одному из них. Последнее связано с тем, что промежуточные цели s_{om} , могут быть формально представлены предикатными функциями P_m , и упорядочены в форме единственной последовательности

$$R_s : \{s_{o1}, P_1\} \infty \{s_{o2}, P_2\} \infty \dots \infty \{s_{om}, P_m\}.$$

Условие перехода можно выполнить, представив границы переключающих множеств \bar{x}_i и \bar{x}_k предикатами P_i , определенными через набор

$$P_i(x^d(t)): \gamma_{ik}(x^c(t)) \leq 0. \quad (2.12)$$

И тогда $(x^d(t), x^c(t))$ удовлетворяет (или не удовлетворяет) предикату P_i если $P_i(x^d(t_k), x^c(t)) = 1$ (соответственно $P_i(x^d(t_k), x^c(t)) = 0$).

Поскольку вершины-позиции результирующей сети при применении данной операции не претерпевают изменений, объединенный граф может быть записан как $N^{cse} = \{P^{ce}, P^{de}, T^{cse}, I^{cse}, O^{cse}, F^{cse}\}$, где I^{cs} и O^{cs} – соответственно множества входных и выходных дуг, соединяющих T^{cs} с позициями P^c и P^d , а индексом e обозначена принадлежность их элементарной сети. Отсюда также при объединении графов N^d и N^c с учетом N^{cse} получаем гибридную сеть со структурно-управляемыми переходами: $N^{cs} = \{P^c, P^d, T^c, T^d, T^{cs}, I^c, I^d, I^{cs}, O^c, O^d, O^{cs}, F^c, F^d, F^{cs}\}$.

Матрица инцидентности гибридной сети N^{cs} может быть представлена как объединенная структура матриц инцидентностей графов дискретной, непрерывной и структурно-управляемой сетей и приведена в таблице 2.4.

Определение. Операцией наложения элементарной непрерывной сети N^{coe} на элементарную дискретную сеть N^{de} называется такая операция, результатом которой является элементарная непрерывно-дискретная сеть $N^{cde} = N^{coe} \cap N^{dOc}$.

Полученный в результате такой операции непрерывно-дискретный переход t_j^{cd} может быть помечен оператором преобразования $f(u, t)$ входного непрерывного сигнала $u_i^c(p_1^c)$ в выходной дискретный сигнал $x_i^d(p_2^d)$, которыми помечены соответственно вершины-позиции p_1^c и p_1^d . Данный переход срабатывает только в случае наличия бесконечного числа меток a во входных непрерывных позициях p_1^c и выполнения оператора преобразования $f(u, t)$. Правила срабатывания непрерывно-дискретного перехода приведены в таблице 2.4.

Поскольку выходные непрерывные и входные дискретные вершины-позиции результирующей сети при применении данной операции отсутствуют, объединенный граф может быть записан как $N^{cde} = \{P^{ce}, P^{de}, T^{cde}, I^{cde}, O^{cde}, F^{cde}\}$, где I^{cd} и O^{cd} – соответственно множества входных и выходных дуг, соединяющих T^{cd} с позициями P^c и P^d , а индексом e обозначена принадлежность их элементарной сети. Отсюда также при объединении графов N^d и N^{co} с учетом N^{cde} получаем гибридную сеть с непрерывно-дискретными переходами, полученными за счет объединения перехода типа T^{co} и T^d путем наложения их друг на друга (рис. 2.9). При таком наложении отпадает необходимость во входных дискретных и выходных

непрерывных позициях, поэтому множества F^{co} , I^{co} и O^{co} заменяются соответственно на F^{cd} , I^{cd} и O^{cd} :

$$N^{cd} = \{P^c, P^d, T^c, T^d, T^{cd}, I^c, I^d, I^{cd}, O^c, O^d, O^{cd}, F^c, F^d, F^{cd}\}.$$

Матрица инцидентности гибридной сети N^{cd} может быть представлена как объединенная структура матриц инцидентностей графов дискретной, непрерывной и дискретно-непрерывной сетей и приведена в таблице 2.4.

Определение. Операцией наложения элементарной дискретной сети N^{de} на элементарную непрерывную сеть N^{ce} называется такая операция, результатом которой является элементарная дискретно-непрерывная сеть $N^{dce} = N^{de} \cap N^{ce}$.

Правила срабатывания перехода t_i^{dc} , внутренняя структура, матрица инцидентностей A^{dc} и правила определения ее элементов приведены в таблице 2.4.

Поскольку выходные дискретные и входные непрерывные вершины-позиции результирующей сети при применении данной операции отсутствуют, объединенный граф может быть записан как $N^{dce} = \{P^{ce}, P^{de}, T^{dce}, I^{dce}, O^{dce}, F^{dce}\}$, где I^{cd} и O^{dc} – соответственно множества входных и выходных дуг, соединяющих T^{dc} с позициями P^d и P^c , а индексом e обозначена принадлежность их элементарной сети. Отсюда также при объединении графов N^{ci} и N^d с учетом N^{dce} получаем гибридную сеть с дискретно-непрерывными переходами, полученными за счет объединения перехода типа T^d и T^{ci} путем наложения их друг на друга (рис. 2.11). При таком наложении отпадает необходимость во входных непрерывных и выходных дискретных позициях, поэтому множества F^{cl} , I^{cl} и O^{cl} заменяются соответственно на F^{dc} , I^{dc} и O^{dc} :

$$N^{dc} = \{P^c, P^d, T^c, T^d, T^{dc}, I^c, O^c, I^d, O^d, I^{dc}, O^{dc}, F^c, F^d, F^{dc}\}.$$

Для получения ДНАСУ (модифицированной ДН-сети) остается только произвести объединения всех полученных типов сетей: $N = \{N^{cs}, N^{cd}, N^{dc}\}$. Здесь

необходимо отметить, что во всех трех частях имеются наборы одинаковых элементов: T^c , T^d , P^c , P^d и F^c , F^d , I^d , O^d , I^c , O^c , которые при объединении поглощают друг друга. По этой же причине в данное объединение не включены остальные простые типы сетей – N^d , N^c . Тогда модифицированная ДН-сеть может быть записана как

$$N = \{P^c, P^d, T^c, T^d, T^{cs}, T^{cd}, T^{dc}, I^c, O^c, I^d, O^d, I^{cs}, O^{cs}, I^{cd}, O^{cd}, I^{dc}, O^{dc}, F^c, F^d, F^{cs}, F^{cd}, F^{dc}\}.$$

Отсюда, матрица инцидентности модифицированной ДН-сети N может быть представлена как объединенная структура матриц инцидентностей графов дискретной, непрерывной и непрерывно-дискретной, дискретно-непрерывной и структурно-управляемой сетей. Матрица инцидентности модифицированной ДН-сети позволяет произвести классификацию исследуемой ДНАСУ, представленной ДН-сетью.

	T^c	T^d	T^{cs}	T^{cd}	T^{dc}
P^c	×		×	×	×
P^d		×	×	×	×

В данном случае в ДНАСУ реализуется прямая и обратная связь между НС и ДС подсистемами.

В случае получения матрицы инцидентности вида

	T^c	T^d	T^{cs}	T^{cd}	T^{dc}
P^c	×		×	×	×
P^d		×	×	×	×

исследуемая ДНАСУ является разомкнутой программной системой. Знак \times обозначает что матрица $T^{(*)} \neq \emptyset$. Матрицы T^{cs} состоят из подматриц.

Для классификации исследуемой ДНАСУ вначале представим ее с использованием языковых средств ДН-сети, затем заполним матрицу инцидентности, в которой учтем все структурные свойства ДНАСУ.

Выводы

1. Предложена процедура перехода системы, представленной событийным графом СГ с ключами совместно с алгоритмом, к модифицированной диаграмме состояния, которая позволила естественным образом ввести в структуру различного вида переходы, которые позволили выполнить объединение подсистем, функционирующих в различных подпространствах состояния: непрерывном, дискретном и дискретно-непрерывном.
2. Разработка метода преобразования позволила произвести переход от представления НС части ДНАСУ к представлению в формате СП, которые являются частным случаем ДН-сетей. Данный метод позволяет использовать методы анализа свойства достижимости СГ для анализа свойства достижимости НС части ДНАСУ, представленных ДН-сетью.
3. Введение дополнительных переходов, реализующих различные функции преобразования сигналов, позволило доопределить интерфейс и разработать обобщенную модель ДНАСУ.
4. Проведение разработки по определению формализованного представления ДНАСУ является основанием для разработки компонентов прогрессивной ИТ моделирования и анализа ДНАСУ.
5. Выполненный анализ ДН систем на основе матрицы инцидентности модифицированной диаграммы состояния позволил классифицировать ДНАСУ на программные, с обратной связью и комбинированные системы.
6. Проведенные исследования качества моделирования дискретно-непрерывных АСУ с использованием нормированной среднеквадратической ошибки (которая составила для эквивалентных аналоговых АСУ 5 – 10%, эквивалентных цифровых – 7 – 15%) показали, что для разработанных дискретно-непрерывных систем она составила менее 1%.

РАЗДЕЛ 3

ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ ОПРЕДЕЛЕНИЯ ДОСТИЖИМОСТИ

3.1. Задача достижимости с позиций теории управления

Усилиями ученых, перечисленных в разделе 1, произошло становление теории управления дискретно-непрерывными системами. Однако можно заметить, что решая задачи управления дискретно-непрерывными системами, авторы заведомо принимали допущение: исследуемая ДНАСУ (математическая модель) обладает свойством достижимости. Причина такого допущения состоит в том, что нет хорошо зарекомендованного метода определения достижимости ДНАСУ. Об отдельных работах было сказано выше.

Разработка и эксплуатация комплексных технических систем ставит задачу разработки устройства управления такими системами. Существование устройства управления предопределяется одним из важнейших свойств объекта управления, а именно, достижимостью. Тип и точность разрабатываемой модели исследуемого реального процесса выбирается в зависимости от целей управления.

В работе будет рассмотрено два типа процессов.

- Систему необходимо перевести из одной зоны пространства состояний в другую без учета точного значения состояния непрерывной составляющей x^c , для достижения цели достаточно чисто дискретного управления, основанного на дискретном приближении смешанного дискретно-непрерывного режима работы [28].
- Систему необходимо перевести из дискретно-непрерывной зоны пространства состояний в другую зону дискретно-непрерывного состояния, обратив внимание на непрерывное поведение во времени.

В первом случае процесс целесообразно представить дискретно-событийной моделью – а именно – СП. И, как следствие, исследовать свойства процесса, используя наработки теоретического базиса СП.

Во втором случае в частях процесса (системы) дискретный и непрерывный режимы работы имеют примерно одинаковый вес. В этом случае идеи, приведенные ранее, должны быть интегрированы, дополняя друг друга, и задача дискретно-непрерывного управления решается допустимым сочетанием дискретных и непрерывных подходов.

Прежде чем поставить задачу достижимости, обратим внимание на некоторые особенности пространства состояний и математической модели ДНАСУ. Выберем пространство состояний, в котором рассматривается задача достижимости.

Выше были представлены в описании ДНАСУ и их пространства состояния: $X = (a_v^f, x^c)$ в работах [40, 56], $\Sigma = (x_0^c, x^c(t_i), M_i^d, t_i^c, P_i, O_i)$ в работе [28] и $Z = (\mu, g, \dot{x}^c, x^c, x_0^c, \tau_0)$. Проанализировав данные представления отметим, что непрерывное состояние $\dot{x}^c(t) = f_i(x^c, u^c, t) = f(t, P_i(t, x^c, u^c), u^c)$ (для линейных стационарных систем) и дискретное состояние $a_v^f \equiv \mu \equiv M_i^d \equiv x_i^d$ в конечном итоге определяет формирование следующей математической схемы $f_i^c = \{P_i, Q_i\}$. Воспользуемся представлением пространства состояния ДНАСУ в форме $X = (x^c(t), x^d(t_k))$ [56].

В дискретно-непрерывной системе промежуточные цели s_{om} , $m=1, \dots, M$ являются результатом декомпозиции общей цели задания управления S_o . Промежуточные цели s_{om} могут быть записаны при помощи предикатных функций P_m и упорядочены в форме единственной последовательности

$$R_s : \{s_{o1}, P_1\} \infty \{s_{o2}, P_2\} \infty \dots \infty \{s_{om}, P_m\},$$

где ρ_i – границы переключающих множеств представлений, которые определены через набор $\rho_i(x^d(t)) : \gamma_{ji}(x^c(t)) \leq 0$, и тогда $(x^d(t), x^c(t))$ удовлетворяет (соответственно

не удовлетворяет) предикату P если $P_i(x^d(t_k), x^c(t))=1$ (соответственно $P_i(x^d(t_k), x^c(t))=0$).

В ДНАСУ в общем случае имеется несколько путей с различной последовательностью срабатывания переходов (как дискретных $t_{(\bullet)}^d$, так и непрерывных $p_{(\bullet)}^c$), переводящих систему из начального состояния $(x^d(0), x^c(0))$ в конечное состояние $(x^d(l), x^c(l))$.

Большинство методов определения достижимости основано на определении локального многообразия $x(t) \in \bar{X}_1$, достижимого при $x^d(t_k)$ определении границ переключающего множества $\Omega = \bar{X}_i \cap \bar{X}_j$ и формировании $u^d(t_l)$. Этот подход основан на вычислении так называемой дискретно-непрерывной траектории, начиная с $x(t_0) = x_0 = (x^d(0), x^c(0))$ во временном интервале $[t_0, t_l]$. Нелинейная динамическая система (2.5) – (2.8) есть гладкая в пределах подинтервала $[t_v, t_{v+1}] \subset (T_1, T_2)$, описываемая на этом подинтервале производящей функций f_v и соответствующими переходной P_v^c и выходной Q_v^c функциями.

Пусть S_0 – целевое множество. Система (2.5) – (2.8) не обладает достижимостью из $|t_0, x_0, x_1^d|$ в S_0 по отношению $U = (u^c(t), u^d(t_k))$ для любого множества траекторий

$$x^c(t) = P_{(\bullet)}(t_{s-1}, t_s), \{P_{(\bullet)}(t_{s-2}, t_{s-1})(t_{s-2}, \dots, P_{(\bullet)}(t_0, t_1)(t_1, t_0, x^c(t_0), u^c(t_0, t_1)), \dots, u^c(t_{s-2}, t_{s-1}))u^c(t_{s-1}, t_s)\}$$

$$P_{(\bullet)}(t_{s-1}, t_s) = \bar{P}\Xi(u^d(t_s)), \bar{P} = (P_1, P_2, \dots, P_N), s = 1, 2, \dots,$$

порождаемых последовательностью срабатывания переходов $t_j^{cs} \in T^{cs}$ в НС части ДНАСУ под действием $u^d(t_s)$, сформированных ДС частью.

ДНАСУ ограничено конструируема относительно S_0 , если одна или несколько переходных функций P_v , $v = 1, 2, \dots, N$, удовлетворяют условию $P(t, x^c(t), u^c(t), u^d(t_s)) \in \Pi_{x^c}(S_0)$, Π_{x^c} – оператор проектирования.

Существования пересечений $X_v^k \cap X_{(v+1)}^k$ означает возможность достичь в ДНАСУ гладкости траектории движения при управлении структурой, в смысле

$$P^v(t^{(-0)}, t_{s-1}^{(0)}, x_{s-1}^{(0)}, u(\bullet)) = P^{v+1}(t_s^{(0)}, x_s^{(0)}, u(\bullet)).$$

В отличие от абстрактной задачи управления детерминированной системой, где однажды полученный вывод об управляемости системы справедлив при рассмотрении вопроса достижимости множеств $X_{t_1} \subset \dots \subset X_{t_n}$, для ДНАСУ характерно при переходе к новой локальной структуре необходимость вновь исследовать достижимость промежуточного целевого множества S_m^0 , $m = 1, 2, \dots, n$. Причем возможны случаи, когда ни одна из структур Σ_v^k , $v = 1, \dots, N$, соответствующая v -му состоянию $u^d(t_s)$ не обеспечивает достижимость S_m^0 и лишь заданием одной из последовательностей переходов $t_i^d \in T^d$ ДС части оператор переходов ДНАСУ приобретает вид

$$P_{(\bullet)}(t_{s-1}, t_s), \{P_{(\bullet)}(t_{s-2}, t_{s-1})(t_{s-2}, \dots, P_{(\bullet)}(t_0, t_1)(t_1, t_0, x^c(t_0), u^c(t_0, t_1)), \dots, u^c(t_{s-2}, t_{s-1}))u^c(t_{s-1}, t_s)\}: t \geq t_0$$

и тем самым обеспечивается достижимость промежуточного целевого множества S_{0m} .

Прежде чем определить задачу достижимости для системы (2.5) – (2.8) определим некоторые особенности системы (2.5) – (2.8), представленные топологическим уровнем, ДН-сетью. Сначала воспользуемся некоторыми понятиями ДН-сетей из работы [56].

Позиция-исток p_i^c всегда содержит вещественное значение сигнала $u_i(t)$, а это значит $\mu(p_i^c) = \alpha$, т.е. маркировка позиции $\mu(p_i^c)$ есть α .

Маркировка непрерывных позиции p_i^c есть функция, отражающая множество P^c во множество $M^c = \{\alpha, 0\}$, где 0 – нуль, α – признак вещественного значения сигнала.

Определение 1. Позиция p_m^c достижима из позиции p_l^c , если существует такое управляющее воздействие $u(t) = (u^c(t), u^d(t_k))$, которое формирует путь $t_l^{(\bullet)} \rightarrow t_2^{(\bullet)} \rightarrow \dots \rightarrow t_r^{(\bullet)}$, $t_i^{(\bullet)} \in T^c \cup T^{cs}$, $i = l, k, \dots, r$, из позиции p_l^c в позицию p_m^c , $p_l^c = I(t_l^{(\bullet)})$ и $p_m^c = O(t_r^{(\bullet)})$. При условии, если позиция $p_l^{(\bullet)}$ имеет метку α , то и позиция p_m^c имеет метку α , если путь активный, в противном случае $\mu(p_m^c) = 0$.

Определение 2. Если структурно-управляемые переходы $t_j^{cs} \in T^{cs}$, входящие в сформированный $u(t) = (u^c(t), u^d(t_k))$ путь $t_l^{(\bullet)} \rightarrow t_2^{(\bullet)} \rightarrow \dots \rightarrow t_r^{(\bullet)}$, $t_j^{(\bullet)} \in T^c \cup T^{cs}$, $j = l, k, \dots, r$, из позиции $p_l^c : \exists I(p_l^c, t_l^{(\bullet)}) \neq 0$ в позицию $p_m^c : \exists O(t_r^{(\bullet)}, p_m^c) \neq 0$, «открыты», то путь активный. Последнее свойство определено тем, что для $\forall p_i^d : I^{cs}(p_i^d, t_j^{cs}) \neq 0$ и $\mu(p_i^d) = 0$.

Поясним данные определения на примере, рассмотренном в работе [30]. Рассмотрим математическую модель системы:

$$\begin{cases} \dot{x}_1^c(t) = b_{11}u_1^c(t) + b_{12}u_2^c(t); \\ \dot{x}_2^c(t) = b_{22}\gamma(u_1^d(t_k))u_2^c(t); \end{cases}$$

$$x_1^c(0) = x_2^c(0) = 0, \quad \gamma(u_1^d(t_k)) \in \{1, 0\}.$$

С учетом значения $\gamma(u_1^d(t_k))$ система может изменять свою структуру. Представим систему (2.5) – (2.8) с использованием языковых средств ДН-сетей на рис. 3.1.

Как видно из схемы, представленной на рис. 3.1, реализован случай когда $\gamma(u_1^d(t_k)) = 1$. В этом структурном состоянии структурно-управляемый переход t_1^{cs} открыт и позиции p_5^c и p_8^c достижимы из входных позиций p_1^c и p_2^c , что подтверждается наличием маркеров во входных и выходных позициях. Таким образом, система, представленная на рис. 3.1, обладает свойством достижимости, т.е. найдется такое $\tau_1 \leq \tau$ и такое входное воздействие, которое переведет состояние $(x^c = (x_1^c(0), x_2^c(0))^T, \tau_0)$ в состояние $(x^c = (x_1^c(\tau_1), x_2^c(\tau_1))^T, \tau_1)$.

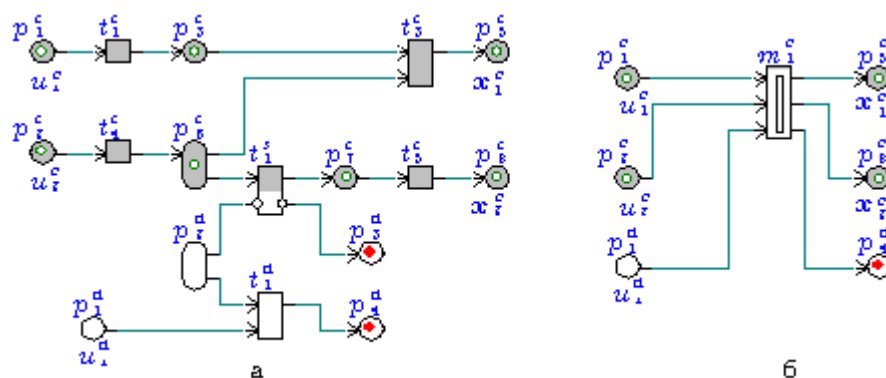


Рисунок 3.1 – Пример первой структуры ДНАСУ. а – структура; б – замещающий макропереход

Что проиллюстрировано рис. 3.2.

$$\begin{cases} \dot{x}_1^c(t) \\ \dot{x}_2^c(t) \end{cases} = \begin{vmatrix} b_{11} & b_{12} \\ 0 & b_{22} \end{vmatrix} \begin{cases} u_1^c(t) \\ u_2^c(t) \end{cases}$$

Рисунок 3.2 – Схема достижимой системы

Формирование следующей структуры обеспечивает $\gamma(u_1^d(t_k))=0$ и реализуется переходом t_1^d в зависимости от маркировки позиций p_1^d , p_2^d . На рис. 3.3,а изображено структурное состояние, когда структурно-управляемый переход t_1^{cs} закрыт и выходная позиция p_8^c становится недостижимой из входной позиции p_2^c .

В данном структурном состоянии система является частично достижимой и может быть представлена следующей структурной схемой (рис. 3.4).

Для анализа сложных систем с большим числом позиций и переходов используется процедура «свертывания» МСГ и автоматическая выдача списка недостижимых позиций-стоков системы. Примеры для соответствующих структурных состояний показаны на рис. 3.2,б и 3.3,б. В первом случае переход t_5^c вошел в макро переход m_1^c , а во втором – нет, что указывает на частичную достижимость в Σ_1 структурном состоянии [19].

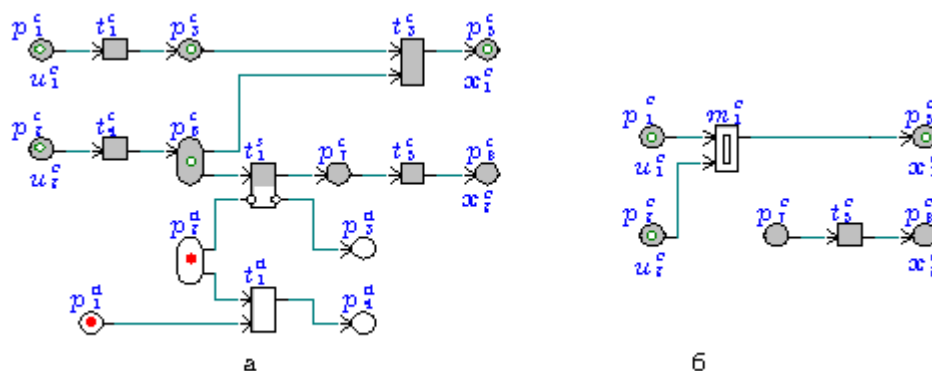


Рисунок 3.3 – Пример второй структуры ДНАСУ. а – структура; б – замещающий макропереход

$$\dot{x}_1^c(t) = \begin{vmatrix} b_{11} & b_{12} \\ u_1^c(t) \\ u_2^c(t) \end{vmatrix}$$

$$\dot{x}_2^c(t) = 0$$

Рисунок 3.4 – Структурная схема частично достижимой системы

Сформулируем задачу достижимости в следующем виде.

Определение 3. Необходимо перевести систему из дискретно-непрерывного состояния (x_0^c, x_0^d) в дискретно-непрерывное состояние (x_r^c, x_r^d) . Система будет называться достижимой, если найдется такое конечное время $\tau \leq t$, в течении которого некоторое управляющее воздействие $U = (u^c(t), \bar{u}_i^d(t_k))$ сделает такой переход возможным.

Остановимся на двух подходах, которые были частично обоснованы ранее и которые будут реализованы ниже.

1.а. Определение алгебраического критерия достижимости Σ_i структуры НС части, проверка достижимости вектора x_i^c , редукция-декомпозиция ДС части и заключение о достижимости i -го режима;

1.б.Выполнение декомпозиции (определение классов (подграфов)) НС части и отдельно ДС части, редукция Σ_i структуры НС части, проверки достижимости вектора x_i^c , редукция ДС части и заключение о достижимости i -го режима;

2 Моделирование i -го режима ДНАСУ, формирование текущего состояния $(x^c(t), x^d(t_k))^T$ разметок и сравнение с заданным состоянием $(x_s^c(t), x_s^d(t_l))^T$. В случае совпадения текущего состояния $(x^c(t), x^d(t_k))^T$ с заданным состоянием $(x_s^c(t), x_s^d(t_l))^T$ делается вывод о том, что заданное состояние достижимо.

3.2. Определение критериев достижимости в структурных состояниях НС части ДНАСУ

Как отмечалось ранее, НС часть описывается системой дифференциальных уравнений с разрывной правой частью, которая формируется алгоритмом, сформированным сетью Петри и интерфейсными функциями. Представление НС части МСГ позволило выделить структурные состояния НС части Σ_i , неизменности правой части системы дифференциальных уравнений на отдельных временных интервалах $\Sigma_{i[t_0, t_1]}, \Sigma_{i+1[t_1, t_2]}, \dots, \Sigma_{k[t_{k-1}, t_k]}$.

Представление ДНАСУ средствами МСГ позволяет проводить декомпозицию НС части на структурные состояния Σ_i путем «включить-выключить» связи между элементами МСГ за счет применения переходов t^{cs} , t^{cd} в МСГ. При этом структурное состояние $\Sigma_i \xrightarrow{\pi} f_i$ отображается в производящую функцию f_i . В результате декомпозиции каждое структурное состояние Σ_i НС части представляется подсетью π_i , для которой справедливо наличие позиций истоков $p_{i_r}^o$, помеченных входными сигналами u_γ^c , $\gamma=1,2,\dots,r$, и позиций стоков $p_{i_k}^c$, $k=1,2,\dots,k$ помеченных выходными сигналами $x_{i_k}^c$.

С учетом того, что можно перейти от теоретико-множественного представления подсетей $(\pi_i)^{ДНС}$, представленных ДН-сетью, к представлению

подсетью $(\pi_i)^{CG}$ на основе СГ, и для $(\pi_i)^{CG}$ определить матрицу единичных путей R_i^{ux} , то можно определить:

если в каждом структурном состоянии $\Sigma_i(\pi_i)^{CG}$ НС части ДНАСУ в матрице $(R_i^{ux})^N$ нет нулевых столбцов, то НС часть ДНАСУ достижима.

Приведем этапы перехода от матрицы инцидентности A_i^c подсети $(\pi_i)^{DHC}$ к матрице единичных путей R_i^{ux} :

1. Получим матрицу единичных путей \tilde{A}_i^c подсети $(\pi_i)^{DHC}$. Для этого из матрицы инцидентностей A_i^c выделим входную A^{cl} и выходную A^{co} матрицы.

2. Построим матрицу R_i единичных путей подсети $(\pi_i)^{CG}$.

Для реализации пункта 1 необходимо из матрицы инцидентности МСГ A_i^c выделить входную A_i^{cl} и выходную A_i^{co} матрицы. Матрица A_i^{cl} может быть получена из матрицы A_i^c путем замены всех ее элементов a на 0 и заменой $-a$ на a . Матрица A_i^{co} может быть получена из матрицы A_i^c путем замены всех ее элементов $-a$ на 0. Поскольку при работе с СГ не используется понятие бесконечного числа маркеров a , то необходимо также при переходе от МСГ к СГ заменить все элементы a матриц A_i^{cl} и A_i^{co} на эквивалентные им 1, означающие наличие связи между узлами СГ, также как и a в МСГ. После данной обработки матриц A_i^{cl} и A_i^{co} получим соответственно матрицы $A_i^{cl'}$ и $A_i^{co'}$.

При построении матрицы единичных путей R_i необходимо объединить матрицы $A_i^{cl'}$ и $A_i^{co'}$ следующим образом [30]:

$$R_i = \begin{bmatrix} 0 & A_i^{cl'} \\ A_i^{co'} & 0 \end{bmatrix}. \quad (3.1)$$

Для того чтобы получить матрицу R_i^c воспользуемся следующим выражением

$$R_i^c = R_i + E.$$

Матрицу R_i можно получить также путем использования следующей теоремы.

Утверждение. Матрица R_i единичных путей СГ может быть получена при помощи бинарного умножения транспонированной входной $A^{cl'}$ и выходной $A^{co'}$ матриц:

$$R_i = A^{cl'}{}^T \cdot A^{co'}$$

Доказательство. Рассмотрим правило бинарного умножения матриц:

$\tilde{a}_y^c = \sum_{k=1}^m a_{ik}^{cl'}{}^T \cdot a_{kj}^{co'}$. Так, для получения «1» необходимо, чтобы хотя бы одно

произведение $a_{ik}^{cl'}{}^T \cdot a_{kj}^{co'}$ равнялось «1», что означает наличие «1» в обоих элементах

$a_{ik}^{cl'}{}^T$ и $a_{kj}^{co'}$ матрицы, а это, в свою очередь означает, что i -я позиция является

входной к какому-либо переходу $i^c(p_i^c, t_k^c) \neq 0$, а j -я позиция – выходной: $o^c(t_k^c, p_j^c) \neq 0$.

Т.е. устанавливается связь, минуя данный переход t_k^c МСГ между его входными и выходными позициями, что является аналогом передачи функций, принадлежащим переходам, дугам и позициям.

При этом подходе размерность матрицы R_i будет почти в два раза меньше чем при получении ее по способу, описанному выше (3.1) [80].

Рассмотрим подробнее на примере.

Пусть в результате преобразования ДНС мы получили МСГ, изображённый на рис. 3.5.

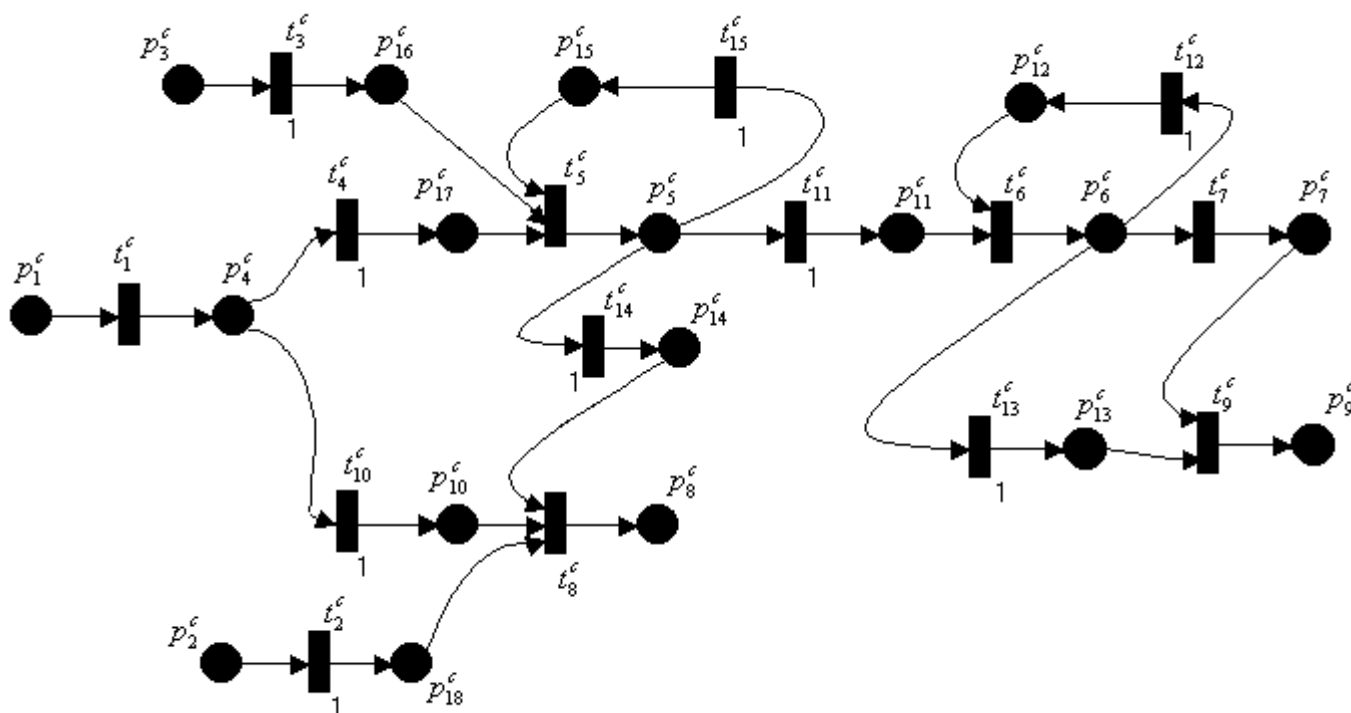


Рисунок 3.5 – Пример ДНАСУ, представленной ДН-сетью

Тогда, после применений преобразований, представленных на рис. 3.5, сокращаем количество вершин МСГ и получаем его в виде, представленном на рис. 3.6.

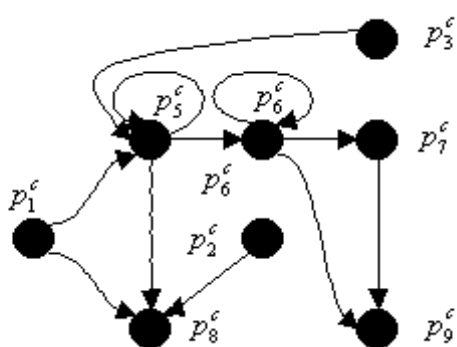


Рисунок 3.6 – Модифицированный сигнальный граф

Составим матрицу инцидентности для МСГ, представленного на рис. 3.5, произведя для удобства расчёта замену « α » на «1». Она имеет вид

	p_1^c	p_2^c	p_3^c	p_4^c	p_5^c	p_6^c	p_7^c	p_8^c	p_9^c	p_{10}^c	p_{11}^c	p_{12}^c	p_{13}^c	p_{14}^c	p_{15}^c	p_{16}^c	p_{17}^c	p_{18}^c
$\tilde{A}^c = p_1^c$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_2^c	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_3^c	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_4^c	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_5^c	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
p_6^c	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
p_7^c	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
p_8^c	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1
p_9^c	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
p_{10}^c	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{11}^c	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{12}^c	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
p_{13}^c	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
p_{14}^c	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{15}^c	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{16}^c	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{17}^c	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{18}^c	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Или соответствующая ей матрица единичных путей:

	p_1^c	p_2^c	p_3^c	p_4^c	p_5^c	p_6^c	p_7^c	p_8^c	p_9^c	p_{10}^c	p_{11}^c	p_{12}^c	p_{13}^c	p_{14}^c	p_{15}^c	p_{16}^c	p_{17}^c	p_{18}^c
$R^c = p_1^c$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_2^c	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_3^c	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_4^c	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_5^c	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
p_6^c	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
p_7^c	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
p_8^c	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1
p_9^c	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
p_{10}^c	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{11}^c	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{12}^c	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
p_{13}^c	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
p_{14}^c	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{15}^c	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{16}^c	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{17}^c	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_{18}^c	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Либо с использованием (3.1):

$$\dot{x}^c(t) = f^c(x^c(t), \bar{u}^d(t_k), u^c(t), t), \quad (3.2)$$

$$x^d(t_k) = x^d(t_{k+1}) + H\delta^d(t_k, \tau^d, s^d) + u^d(t_k, \tau^d, s^d), \quad (3.3)$$

$$\delta^d(t_k, \tau^d, s^d) = \Delta(u^d(t_k, \tau^d, s^d), \eta^d(t_k)), \quad (3.4)$$

$$x^d(t_0) = x_0^d, \quad u^d(t_0) = u_0^d.$$

Динамику поведения ДНАСУ в пространстве состояний $(\bar{x}_I^d(t_k), \bar{x}_{II}^d(t_k))$ можно представить системой:

$$\dot{x}^c(t) = f^c(x^c(t), \bar{u}^d(t_k), u^c(t), t), \quad (3.5)$$

$$\bar{x}_I^d(t_k) = \bar{x}_I^d(t_{k-1}) + \bar{A}_I^d v_I^d(t_k) + \bar{u}_I^d(t_k), \quad (3.6)$$

$$\bar{x}_{II}^d(t_k) = \bar{x}_{II}^d(t_{k-1}) + \bar{A}_o^d v_o^d(t_k) + \bar{u}_o^d(t_k). \quad (3.7)$$

Как видно из рис. 2.9 алгебраическому представлению НС части соответствует уравнение (3.5), теоретико-множественному представлению НС части соответствует уравнение (3.6) и теоретико-множественному представлению ДС части – уравнение (3.7).

Определение 4. Необходим перевод системы (3.5) – (3.7) из состояния $(x_I^c = \alpha, x_0^d)$ в состояние $(x_{I,r}^c = \alpha, x_r^d)$. Система будет называться достижимой, если найдутся такие качественные свойства НС и ДС подсистем при выполнении которых при управляющем воздействии $U = (u_I^d(t_k), u_o^d(t_k))$ сделает такой переход возможным.

В определении достижимости мы несколько упростим задачу и будем рассматривать проблему достижимости для i -го режима функционирования. Тогда, если ДНАСУ обладает свойством достижимости в каждом отдельном режиме O_j , то она достижима в целом.

Данное условие можно выполнить, представив границы переключающих множеств предикатами, определенными через набор

$$\rho_i(x^d(t)): \gamma_{ji}(x^c(t)) \leq 0,$$

и тогда $(x^d(t), x^c(t))$ удовлетворяет (соответственно не удовлетворяет) предикату Р, если $\rho_i(x^d(t_k), x^c(t)) = 1$ (соответственно $\rho_i(x^d(t_k), x^c(t)) = 0$).

Определение 5. Предикатная функция $\rho_m(x^d(t^h), x^c(t))$ будет истинной, если описатель $d_1(x_i^c(t))$ переменной $x_i^c(t)$, такой, что $d_1(x_i^c(t)): \mu(P_i^c) = \chi$, при этом формируются условия перехода из Σ_i в Σ_{i+1} , которые формируются в ДС части системы.

Поясним работу алгоритма на примере дискретно-непрерывного процесса, представленного на рис. 3.7.

Дискретно-непрерывный процесс состоит из НС части, в которой формируются различные производящие функции из $\Delta = \{f_1, f_2, \dots, f_n\}$ под действием вектора u_0^d , формирующего связи между элементами НС части. Переходы типа t_i^{cs} , реализующие связи между элементами, представлены непосредственно после позиций истоков p_i^c , помеченных непрерывным сигналом u_i^c , т. е. $p_i^c \in I(p_i^c, t_i^{cs})$ и $|\mu(p_1^c) = \mu(p_3^c) = \mu(p_7^c)| = a$. В начальном структурном состоянии $u_o^d = |u_{o1}^d, u_{o2}^d, u_{o3}^d|^T = |\mu(p_1^d), \mu(p_4^d), \mu(p_7^d)|^T$, значения $\mu(p_i^d) \in \{0, 1\}$, $i = 1, 4, 7$.

Составляющая дискретно-непрерывного процесса, ДС часть, представлена двумя режимами в нижней половине рисунка. В ДС части формируются отдельные операции или режимы (набор операций, входящих в режим). В данном процессе могут быть реализованы отдельные операции.

Например:

$$u_1^c = \frac{dx_1^c}{dt} > 0 \rightarrow x_1^c \uparrow \text{ (увеличение значения сигнала } x_1^c \text{),}$$

$$u_1^c = \frac{dx_1^c}{dt} = 0 \rightarrow x_1^c = const \text{ (неизменное значение сигнала),}$$

$$u_2^c = const > 0 \rightarrow x_2^c \uparrow \text{ (увеличение значения сигнала } x_2^c \text{),}$$

$$u_2^c = const < 0 \rightarrow x_2^c \downarrow \text{ (уменьшение значения сигнала } x_2^c \text{).}$$

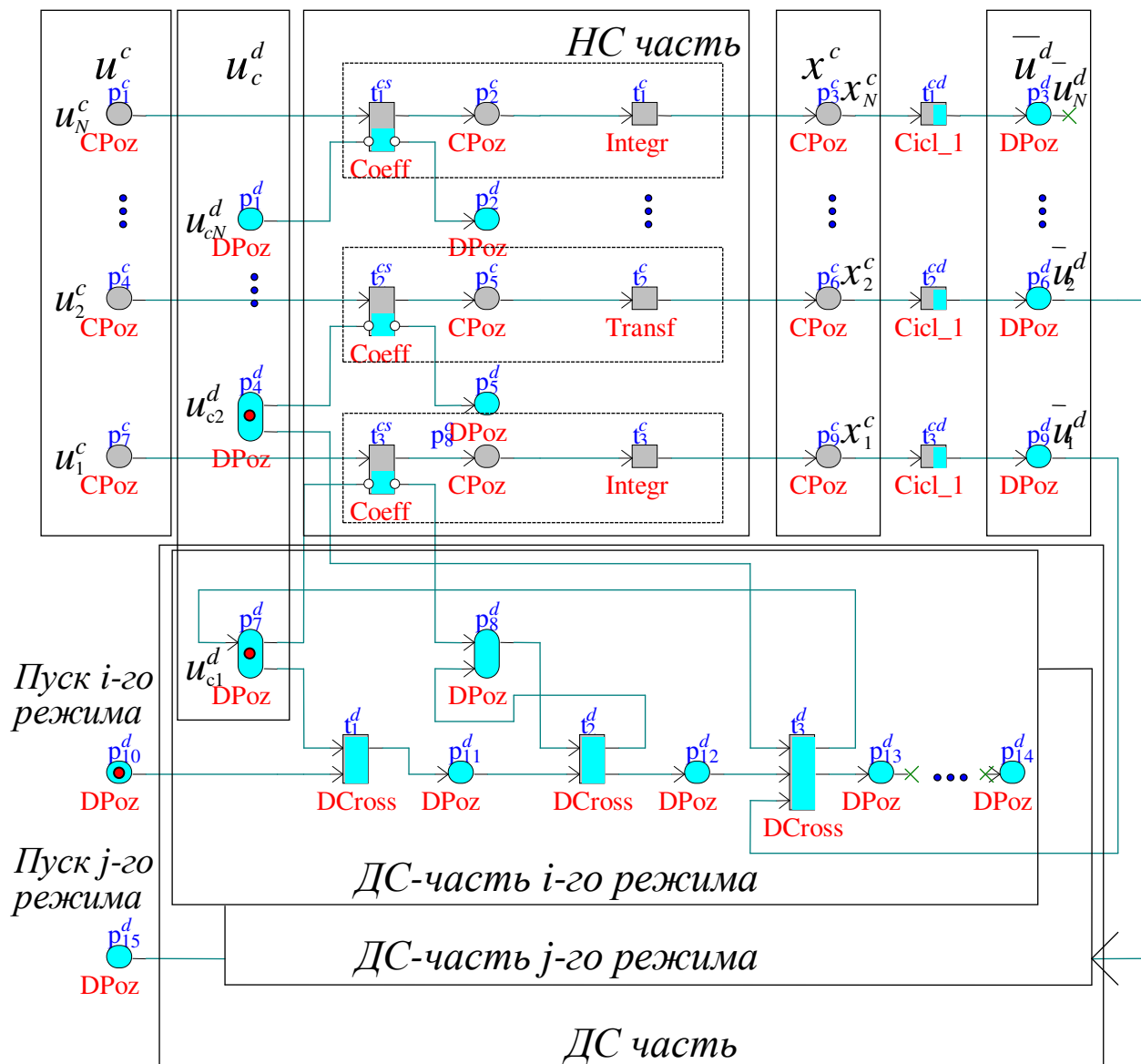


Рисунок 3.7 – Пример ДНАСУ

Поясним реализацию i -го режима функционирования, состоящего из набора операций $\langle x_1^c \uparrow, x_1^c = const, x_2^c \uparrow \rangle$, при этом $\langle x_1^c \uparrow \langle x_1^c = const \rangle x_2^c \uparrow \rangle$. Как было определено выше, в начальном состоянии все переходы типа t_i^{cs} закрыты. Вектор состояния $x^c(t_0) = (x_2^c(t_0), \dots, x_n^c(t_0))$, при этом описатели непрерывных сигналов $d_1^c(\mu(p_9^c)) = d_2^c(\mu(p_6^c)) = d_3^c(\mu(p_3^c)) = \emptyset$. Это справедливо в силу того, что между позициями-стоками p_7^c, p_4^c, p_1^c (помеченных соответственно сигналами u_1^c, u_2^c и u_n^c) и позициями-истоками p_9^c, p_6^c и p_3^c не существует путей. Данное свойство вытекает

из определения 1. Реализуем операцию o_1 ($x_1^c \uparrow$). Каждой операции o_i соответствует свое структурное состояние Σ_i НС части. Инициализирует начало i -го режима функционирования, а также операции o_1 воздействие $u^d(t)$ (занесение маркера в позицию p_{10}^d). При $\mu(p_{10}^d)=\mu(p_7^d)=1$ переход t_1^d срабатывает. В результате чего маркировка станет следующей: $\mu(p_{10}^d)=\mu(p_7^d)=0$ и $\mu(p_{11}^d)=1$. И как следствие этого сработает (откроется) переход t_3^{cs} . В результате срабатывания которого $\mu(p_8^d)=1$.

Сформируется путь $p_7^c \rightarrow t_3^{cs} \rightarrow p_8^c \rightarrow t_3^c \rightarrow p_9^c$, т.е. выполняется условие достижимости сигнала x_1^c по отношению к входному воздействию u_1^c , и как следствие, $d_1^c(\mu(p_9^c))=\infty$. Переход t_2^d предназначен для проверки выполнения требования – переход t_3^{cs} «открыт», что достигается благодаря наличию входной и выходной дуг, проведенных между ним и сигнальной позицией p_8^d перехода t_3^{cs} . В результате данной проверки маркировка изменится следующим образом: $\mu(p_{11}^d)=0$, $\mu(p_{12}^d)=1$. Далее становится возможным реализация операции o_2 ($x_1^c = const$) и одновременно операции o_3 ($x_2^c \uparrow$). При $\mu(p_4^d)=\mu(p_{12}^d)=\mu(p_9^d)=1$ срабатывает переход t_2^{cs} , и устанавливается маркировка $\mu(p_4^d)=\mu(p_{12}^d)=\mu(p_9^d)=0$ и $\mu(p_7^d)=\mu(p_{13}^d)=1$; сформируется путь $p_4^c \rightarrow t_2^{cs} \rightarrow p_5^c \rightarrow t_2^c \rightarrow p_6^c$ и $d_1^c(\mu(p_6^c))=\infty$. В результате появления маркера в позиции p_7^d переход t_3^{cs} опять закрывается, устанавливая маркировку $\mu(p_8^d)=0$ и разрывая путь $p_7^c \rightarrow t_3^{cs} \rightarrow p_8^c \rightarrow t_3^c \rightarrow p_9^c$ и $d_1^c(\mu(p_9^c))=\emptyset$. Данное условие будет выполнено в результате того, что $\rho(x_1^d(t_3^{cd}), d_1^c(\mu(p_9^d)))=1$. Поскольку при закрытии перехода t_3^{cs} значение выходного сигнала p_8^c устанавливается в 0, выполняется операция $x_1^c = const$. Операция $x_2^c \uparrow$ реализуется в результате срабатывания перехода t_2^{cs} .

Описание формирования отдельных операций и, как следствие, режимов функционирования, а также определение достижимости состояние x^c позволит представить обобщенный алгоритм проверки достижимости дискретно-непрерывной системы.

Структурная схема обобщенного алгоритма определения достижимости ДНАСУ приведена на рис. 3.8.

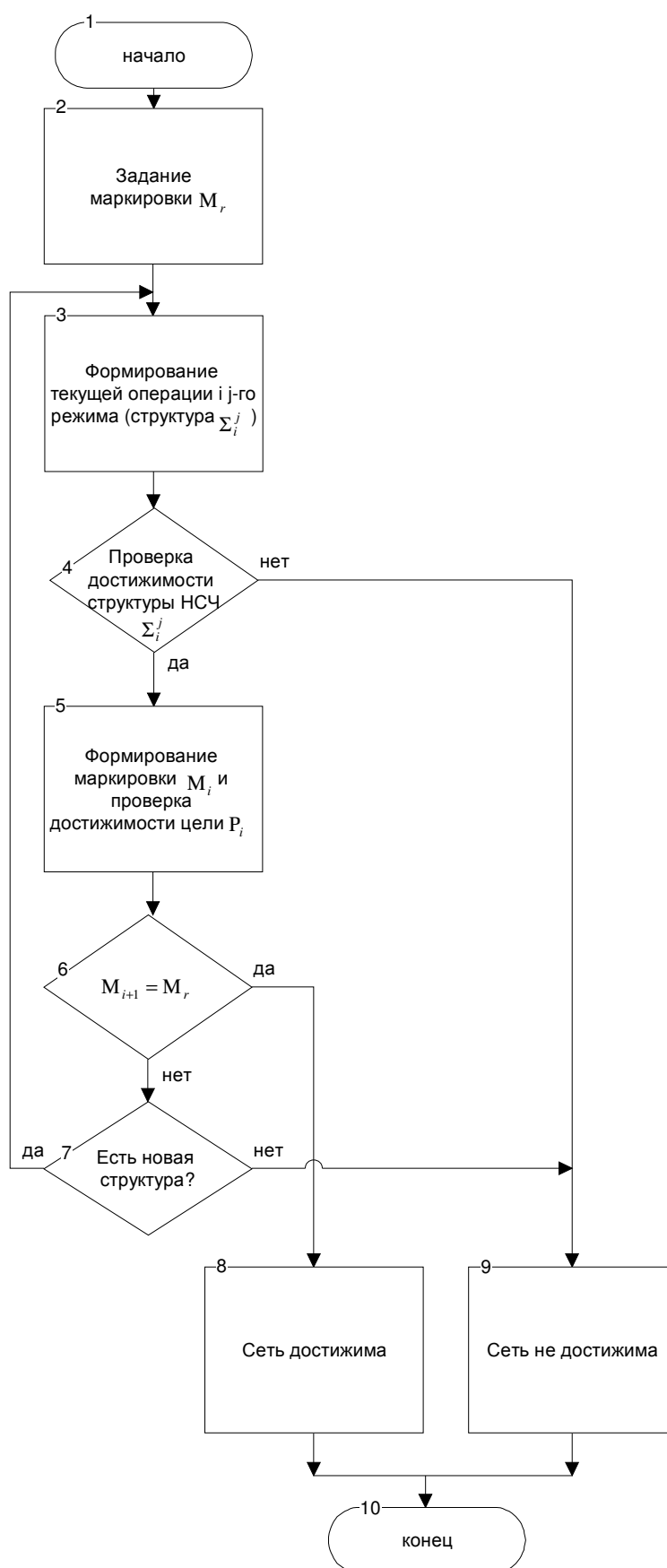


Рисунок 3.8 – Общий алгоритм проверки достижимости модифицированной ДН-сети

3.4. Особенности построения моделей

Прежде чем перейти к рассмотрению алгоритмического метода определения достижимости обратим внимание на следующие особенности представленной в разделе обобщенной модели, в том случае, если данная модель претендует на адекватное отображение процессов, происходящих в реальных дискретно-непрерывных процессах. Дискретным алгоритмом управления, реализуемым в дискретно-непрерывном процессе, назовем конечное множество операций, на котором заданы бинарные отношения следования и параллелизма, режимом логического управления – упорядоченное подмножество операций. Дискретно-непрерывный процесс всегда имеет начало и конец, и, соответственно, в нем фиксируется подмножество начальных и подмножество финальных операций. Дискретный алгоритм управления будем называть циклическим, если любая начальная операция в нем непосредственно следует за финальными, и кроме того выполняется условие: если две операции находятся в отношении параллелизма, то они принадлежат таким частям процесса (подпроцессам), которые одновременно запускаются и одновременно заканчиваются [53]. Если же в процессе с отмеченными выше свойствами начальные операции могут инициироваться до завершения процесса, т.е. непосредственно следовать за нефинальными операциями, то дискретный процесс назовем конвейерным.

Выделим условия, которым должен удовлетворять дискретный алгоритм управления, как циклический, так и конвейерный.

1. Любая операция может быть повторно начата только после своего завершения, т.е. исключается инициирование операции во время ее выполнения.

2. Дискретный процесс, начавшись, всегда может быть доведен до конца: в ходе его выполнения не должно возникать «тупиковых» ситуаций, не имеющих продолжения.

Дискретно-непрерывный процесс представляет собой один из возможных вариантов изменения дискретно-непрерывного состояния $X = (x^c(t), x^d(t))$.

Корректность дискретно-непрерывного процесса будет обеспечена если, во-первых, алгоритм логического управления соответствует приведенным выше двум условиям, и, во-вторых, согласован с возможностями непрерывной составляющей α -состояния, выраженной через предикатную функцию, связывающую непрерывную и дискретную части системы.

Модель многорежимного дискретно-непрерывного процесса должна отображать все переходы от режима к режиму для каждой технологической величины $X^c(t)$ под действием всех возможных значений управляющих воздействий $u^c(t)$, $u^d(t)$ и технологических условий $\rho_1(x^c(t), x^d(t))$. Модель дискретно-событийного процесса, реализуемого в многорежимном объекте, должна вскрывать взаимосвязь между режимами изменения технологических величин и операциями процесса, а также отображать все переходы между операциями при конкретном поведении объекта. Согласование моделей дискретно-непрерывного процесса объекта заключается в определении возможности появления недопустимых режимов изменения технологических величин и нахождения условий, исключающих подобные ситуации [37].

Необходимо подчеркнуть еще одну особенность предложенных моделей гибридных процессов в работах [55, 95, 96, 103, 105, 106], но не отмеченных авторами. Это связано с компьютерной реализацией моделей, разработанных на основе СП.

При компьютерной реализации функционирования СП не выполняется условие Петри, что может сработать любой из переходов p_i^d , $i = 1, 2, \dots, n$, если для них выполняются условия срабатывания. В компьютерной реализации сработает раньше тот переход $t_i^d(\cdot)$, который был раньше набран или сформулирован в программной реализации.

Данную особенность СП можно устранить, если при формировании переходов t_i^d в маркировке M_i^d , использовать датчик случайных чисел для определения, какой из t_i^d сработал в новой маркировке $M_{i=1}^d$, затем возврат в маркировку M и формирование нового перехода, используя датчик случайных чисел.

Если рассматривать реальные дискретно-непрерывные процессы, то в них такие ситуации не допускаются, а формируются четкие условия набора операций и взаимосвязи режимов. Условия определяются регламентом ведения дискретно-непрерывного технологического процесса, который прописан технологом. Формирование условий смены операций обеспечивается проверкой выполнения предикатных функций.

Обратим внимание еще на одну особенность дискретно-непрерывного процесса: при реализации алгоритмов логического управления не допускается наличие в них конфликтов.

3.5. Аналитический подход к определению достижимости ДНАСУ

Определим множество дискретно-непрерывной достижимости. В общем случае существует несколько путей с различной последовательностью срабатывания переходов из начального множества $\bar{X}_0(0,0)$ в конечное множество $\bar{X}_r(x_r^c, x_r^d)$. Переход из одного структурного состояния Σ_i в другое структурное состояние Σ_j произойдет, если непрерывное состояние $x^c(t)$ попадает в переходное множество $x(t) \in \bar{X}_{ij}$, где $\bar{X}_{ij} = \bar{X}_i \cap \bar{X}_j$, $x^c = R^n = \bigcup_{i=0}^N \bar{X}_i$ – пространство состояний непрерывной составляющей, \bar{X}_0 – область пространства x^c , достижимая из начального состояния x_0^c в структурном состоянии $\Sigma_0 = \langle f_0, P_0, G_0 \rangle$ или $f_0 = A_0 x^c(t) + B u^c(t)$. Обозначим область достижимости ДНАСУ в начальном структурном состоянии Σ_0 следующим образом: $\bar{X}_0 = \bar{X}_0(x_{n,0}^c, x^d(t_0))$. После попадания в переходное множество \bar{X}_{01} произойдет изменение структурного состояния НС подсистемы на $\Sigma_1 = \langle f_1, P_1, G_1 \rangle$. Тогда область достижимости для этой структуры представим в следующем виде: $\bar{X}_1 = P_1(\bar{X}_0(x_{n,0}^c), x^d(t_1))$. В общем виде $\bar{X}_i = P_i(\bar{X}_{i-1}(x_{H,i}^c), x^d(t_i))$

Математическая система ДНАСУ, основанная на представлении средствами ДН-сетей, позволяет воспользоваться уникальной возможностью работать с разными срезами (стратами) представления ДНАСУ:

- алгебраической моделью.
- теоретико-множественной моделью.

Исследование свойств ДН-сетей, а именно: позиция p_i^c достижима из позиции p_r^c , если между ними существует путь и он активен, позволяет перейти к определению достижимости ДНАСУ на основе алгоритмического подхода, опирающегося на теоретико-множественное представление ДНАСУ с учетом пометочной функции ДН-сети.

Определение 6. Дискретно-непрерывное множество достижимости ДН-сети есть операционное вычисляемое множество трех кортежей

$$(x_v^d(t_k); x^d(t_l)(\Sigma_{H,v}); \rho(x^d, x^e(t_l))=1).$$

Этапы реализации множества кортежей (метод анализа достижимости) [26]:

- 1.а. Формирование структуры Σ_i НС подсистемы.
- 1.б. Формирование матрицы инцидентности A_{ii}^e для структуры Σ_i .
2. Проверка достижимости структуры Σ_i (на основе анализа R_i или на основе декомпозиции-редукции НС части).
3. Проверка истинности $\rho(x^d(t_k), x_i^c(t))=1$. При выполнении истинности предиката – проверка условия $\rho(x^d(t_k), x_i^c(t))=\rho_m$, при выполнении которого – остановка, в противном случае – определение вектора состояния $x^d(t_k)$ (проведение редукции-декомпозиции ДС подсистемы), при этом автоматически формируется $\bar{u}^d(t_k)$; увеличение i на 1 и переход к пункту 1.а.

3.6. Определение правил редукции непрерывно-событийной части ДНАСУ

Механизм формирования структурного состояния и переход от структурного состояния Σ_i в другое структурное состояние Σ_j был описан выше. В данном подразделе определим, какие преобразования можно производить над подграфами

НС части и каково влияние входных сигналов $u_i^c(t)$ на выходные сигналы $a_i^c(t)$, т.е. определим свойство достижимости для МСГ. Предположим, что в результате декомпозиции возможно формирование ограниченного количества типов подграфов. Для каждого подграфа построим матрицу инцидентности N^u . По матрице N^u определим путь от позиции-истока к позиции-стоку. Если позиция-сток достижима из позиции-истока, то производим преобразование подграфа [80].

Выделим 1 тип подграфа (А-элемент) (рис. 3.9).

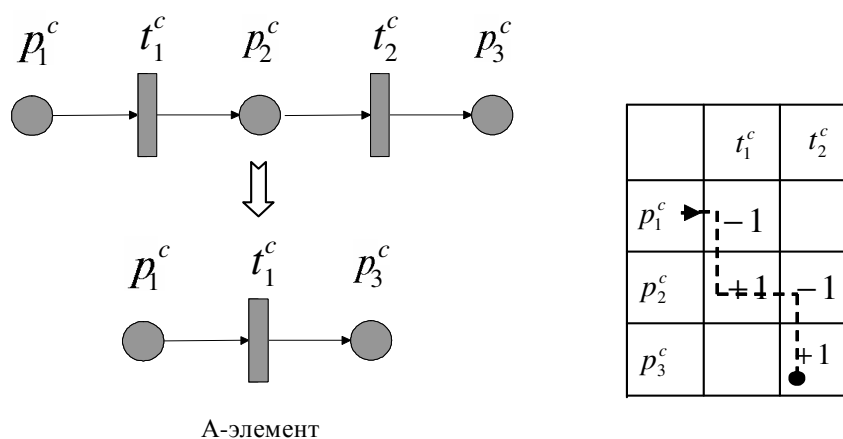


Рисунок 3.9 – А-элемент и матрица инцидентности

Для подграфа 1 типа построим матрицу инцидентности (рис. 3.9).

Анализ матрицы N^u позволяет сделать вывод, что существует путь из позиции сток p_i^c в позицию исток p_3^c . Путь отмечен пунктирной линией. Тогда подсеть может быть заменена элементарной сетью, показанной на рис. 3.9.

Рассмотрим 2-й подграф (П-элемент) и его матрицу инцидентности (рис. 3.10).

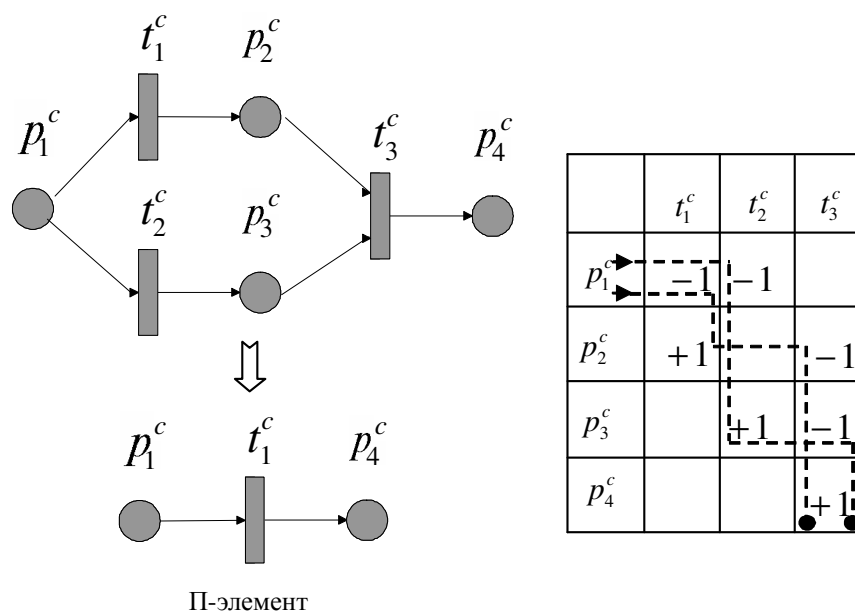


Рисунок 3.10 – П-элемент и матрица инцидентности

Можно удалить переходы t_1^c , t_2^c и позиции p_2^c и p_3^c – получим элементный подграф (подсеть).

Анализ матрицы инцидентности подграфа, приведенного ниже (рис. 3.11), позволяет сделать возможным замену его следующим подграфом.

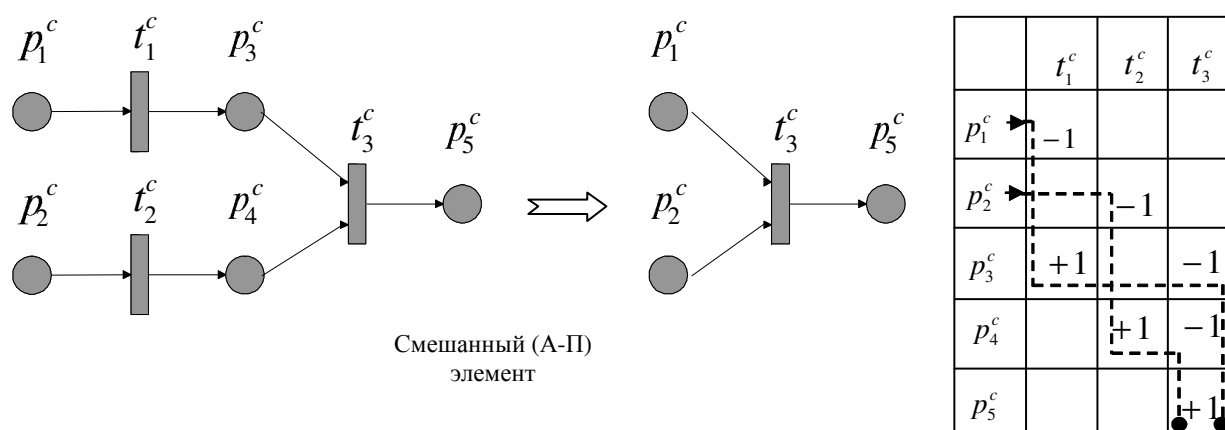


Рисунок 3.11 – Смешанный элемент и матрица инцидентности

Анализируя подграф с обратной связью и его матрицу инцидентности, можно сделать вывод, что данный подграф (рис. 3.12) также может быть заменен на сокращенный.

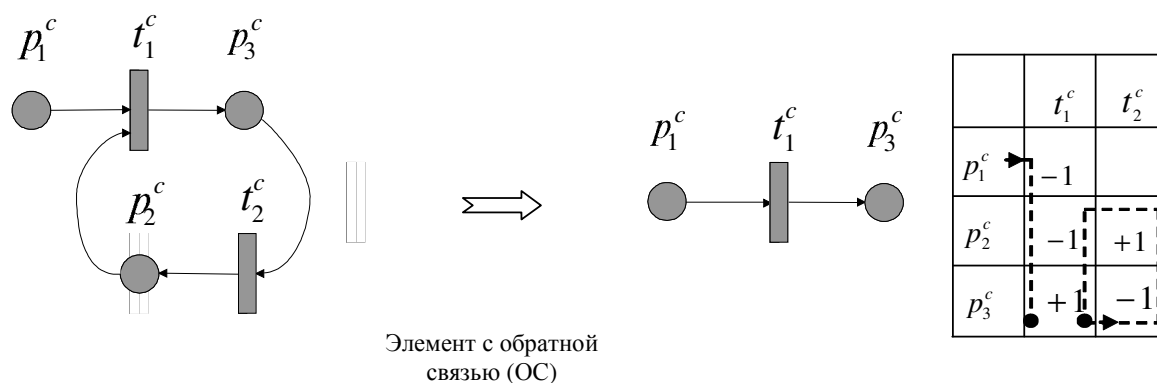


Рисунок 3.12 – Элемент с обратной связью и матрица инцидентий

Данные преобразования могут быть произведены над подграфами, если непрерывные переходы помечены линейными операторами (интегрирования, дифференцирования, суммирования, коэффициентом передачи). Для случая, когда они помечены нелинейными операторами необходима процедура линеаризации либо представление нелинейного элемента в виде подсистемы ДН-сети.

На основе матриц инцидентности по аналогии с рассмотрением достижимости НС части также не трудно убедиться в следующих заменах (рис. 3.13).

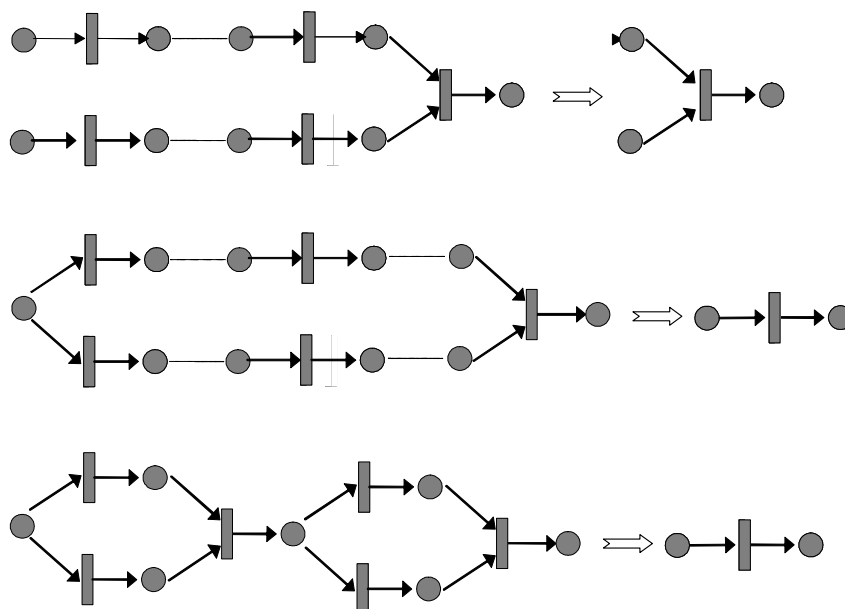


Рисунок 3.13 – Возможные замены НС части ДН-сети

Рассмотренный метод анализа достижимости при помощи редукции-декомпозиции имеет свои ограничения применения из-за ограничений на правила срабатывания непрерывных переходов, которые не рассматриваются в данном методе и которые в общем случае могут отличаться от стандартных, а также из-за конечного множества рассмотренных вариантов редукции сети. Несколько расширить область его применения позволяет использование матрицы единичных путей соответствующего СГ.

3.7. Определение правил редукции дискретной части ДНАСУ

Процедура декомпозиции-редукции сети Петри обсуждалась в нескольких работах. Впервые упоминается в публикации [4], более широкое обсуждение и применение выполнено в работе [95].

Основная идея метода редукции сети Петри заключается в замене отдельных подсетей общей модели макро позициями [95]. В диссертационной работе отдельные подсети будут заменены макро переходами [56], по той причине, что этот подход позволяет проводить декомпозицию и дискретно-непрерывных сетей. Замена подсистем производится макро переходами таким образом, чтобы свойства преобразованной модели оставались неизменными.

Основные свойства СП, такие как живучесть и ограниченность, рассмотрим ниже.

Определение 7. СП называется живучей, если в ней может быть запущен каждый переход t_j^d .

Изображенная на рис. 3.14 сеть живуча, так как для маркировки $\mu = (1,0,0)$ возможны запуски переходов, в последовательность которых входят все переходы, имеющиеся в сети.

Определение 8. Элементарная сеть Петри – это такая сеть, которая состоит из одного перехода, одной входной позиции и одной выходной позиции.

Определение 9. Макропереход – это переход, представляющий подсеть. Обозначается прямоугольником с планкой внутри.

Суть метода заключается в следующем: если после преобразования сети (когда невозможно выделить поддающиеся преобразованию блоки) получилась элементарная сеть, содержащая один переход и две позиции (входную и выходную), то все позиции в данной сети являются достижимыми; в противном случае, необходимо воспользоваться алгоритмом построения дерева достижимости и с его помощью решить вопрос достижимости [80].

Предлагаются следующие правила редукции СП (по аналогии с [4, 52 – 55, 95]) с целью сведения ее к типовым локальным подсетям (рис. 3.14).

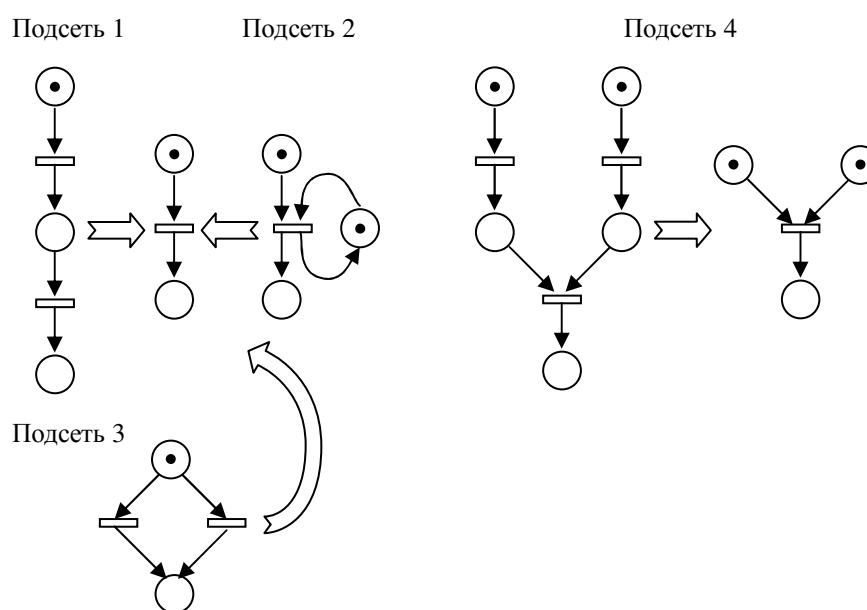


Рисунок 3.14 – Правила редукции СП

Необходимо отметить, что применение данного метода редукции-декомпозиции ДС части ДН-сети не всегда позволяют определить свойство достижимости. Это может происходить из-за наличия конфликтов или если сеть имеет сложную структуру. В данном случае метод позволяет упростить анализ, сократив размерность анализируемой системы.

Выводы

1. В результате анализа подходов к определению достижимости ДН системы установлено, что определение свойства достижимости ДН системы самым тесным образом связано с формализацией задачи управления как итерационной процедуры, обеспечивающей перебор структур на множестве $\{\Sigma\}_{i=1}^N$ и оценку эффективности системы по заданному критерию.
2. Разработанный метод анализа свойства достижимости НС части ДНАСУ основан на методе преобразования СГ в формат представления СП и на анализе свойства достижимости соответствующего СГ.
3. Разработанный метод анализа достижимости ДНАСУ основан на редукции-декомпозиции ДН-сетей, языковыми средствами которых описывается ДНАСУ.
4. Установлено, что в некоторых случаях, таких как сложность анализируемой системы, наличие конфликтов ДН-сети, а также нелинейностей и нестандартных правил срабатывания переходов, аналитический метод не может дать ответ на вопрос о достижимости заданного состояния. В этих случаях необходимо применение численного метода.
5. В случае невозможности дать однозначный ответ на вопрос о достижимости заданного состояния аналитический метод позволяет упростить задачу путем снижения размерности системы.

РАЗДЕЛ 4

РАЗРАБОТКА ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ МОДЕЛИРОВАНИЯ И АНАЛИЗА СВОЙСТВА ДОСТИЖИМОСТИ ДНАСУ

4.1. Базовая информационная технология моделирования ДНАСУ

При рассмотрении модели ДНАСУ, описываемой системами (2.5) – (2.10) можно отметить следующие особенности. ДН-сеть, описывающая ДНАСУ, содержит помимо непрерывной части еще и дискретную часть, состояние которой определяет текущую структуру системы Σ_i . Перед началом моделирования необходимо провести выделение текущей структуры, для моделирования которой имеется возможность использовать известные методы численного интегрирования. Дискретная часть описывается при помощи СП с некоторыми отличиями от классической, что позволяет использовать для моделирования ее работы известные методы [66, 67]. Обе части системы вместе с интерфейсом между ними описаны с помощью единых требований языка ДН-сетей и хранятся в едином массиве. Каждый из элементов системы имеет свои входы и выходы, информация о соединении которых хранится в специальном массиве. Это потребовало разработки метода моделирования ДНАСУ, описанных при помощи языковых средств ДН-сетей, в который необходимо изначально встроить этапы, связанные с проверкой, при необходимости, свойства достижимости исследуемой системы. Последнее вызвано сложностью систем и необходимостью сокращения временных затрат на разработку всей системы в целом [79].

Разработанный метод моделирования ДН объектов заключается в следующем.

1. Проводится инициализация системы с исчислением начальной маркировки дискретной и непрерывной частей системы.

2. При необходимости проводится анализ свойства достижимости с помощью аналитического метода.

3. Производится выделение текущей структуры непрерывной части системы, при этом вычисляются условия срабатывания структурно-управляемых переходов.

4. Для активных непрерывных, дискретно-непрерывных, а также структурно-управляемых переходов проводится вычисление значений выходных сигналов на текущий момент времени.

5. Значения непрерывных сигналов, вычисленных на предыдущем этапе передаются позициям.

6. Обрабатываются условия срабатывания непрерывно-дискретных переходов, формируется связанные с этим изменения дискретных маркеров. Если изменений нет, то тогда этапы 4 и 5 повторяются с новыми значениями текущего времени до выявления таких изменений, иначе проводится обработка дискретных переходов и позиций. В результате получается новая дискретная разметка, на основании которой формируется новая структура, которая и обрабатывается на втором этапе. При исследовании свойства достижимости на данном этапе проводится сравнение текущей разметки с заданной, на основании результатов которого делается вывод о достижимости заданного состояния.

Для моделирования событийной составляющей ДН-сети, представляющей из себя дискретную сеть Петри, используется вычисление вектора ее срабатывания и вычисления дискретного управляющего воздействия $u^d(t_k)$ на структурно-управляемые переходы. На основании вычисленного вектора формируется текущая структура непрерывной части системы. При этом структурно-управляемые и дискретно-непрерывные переходы на основании дискретного управляющего воздействия формируют начальные установки моделирования текущей структуры и разметки непрерывной части.

Для моделирования непрерывной части исследуемой системы были реализованы методы Эйлера и Рунге-Кутта четвертого порядка, при помощи которых производится моделирование сигналов, обрабатываемых непрерывными либо активными структурно-управляемыми переходами ДН-сети. Обратная связь между непрерывной и дискретной частями обеспечивается при помощи работы непрерывно-дискретных и дискретно-непрерывных переходов, логика работы

которых описана ранее. При проведении моделирования производится визуализация выходных сигналов состояния (непрерывного $x^c(t)$ и дискретного $x^d(t_k)$), а также изменения состояния маркировки дискретных позиций, изменения маркировки a в непрерывных позициях с отображением ее на схеме системы. Таким образом, базовая ИТ моделирования ДНАСУ на базе ДН-сетей может быть представлена в виде, показанном на рис. 4.1.

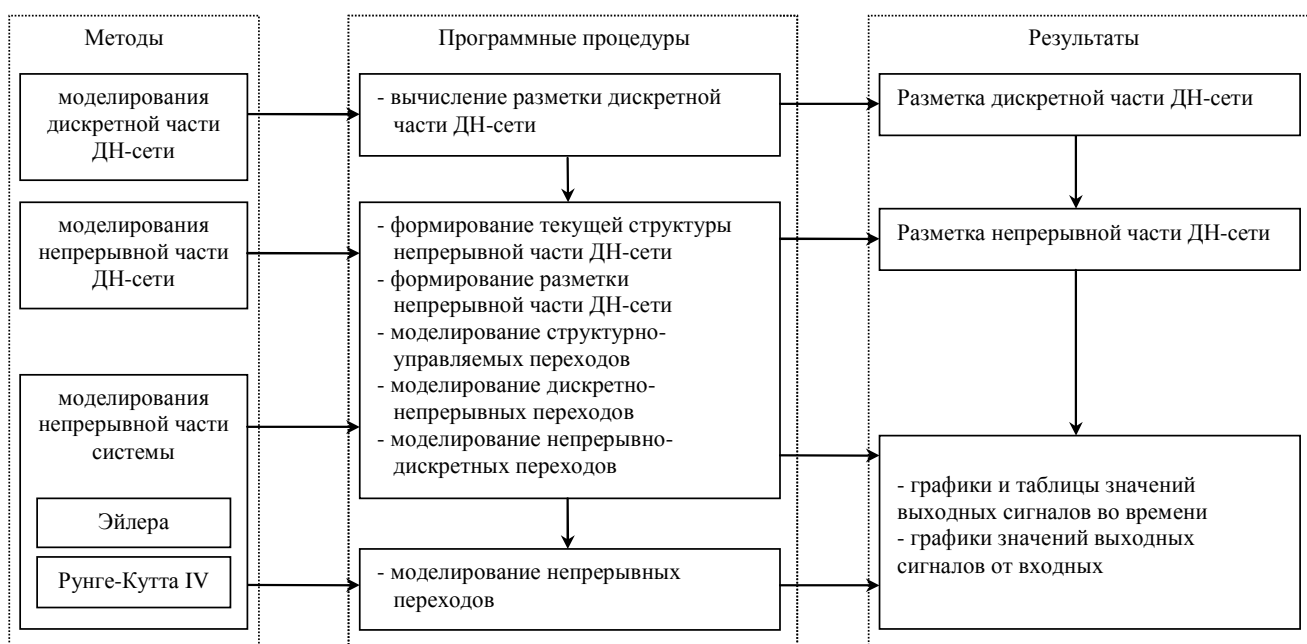


Рисунок 4.1 – Базовая ИТ моделирования дискретно-непрерывных систем

4.2. Базовая информационная технология анализа достижимости ДНАСУ

При исследовании проблемы достижимости гибридной системы принимаем, что данная система спроектирована, а именно:

- определена глобальная цель S_M и текущие цели S_{Mi} ;
- под каждую текущую цель s_{i0} определена операция O_{iM} и структура Σ_i , реализующая операцию;
- определен режим (или режимы) (набор операций), реализующийся в ДНАСУ и алгоритм логического управления;
- разработана математическая модель ДНАСУ;

- модель гибридной системы (НС часть, ДС часть и интерфейс) реализована в ИТ моделировании и анализа ДНАСУ DCNET и заданы начальные условия;
- сформирована матрица связей LinkMatrix подпрограммой CreateLinkMatrix формирования данной матрицы.

В общем виде алгоритм определения достижимости [33, 35] ДНАСУ приведен на рис. 4.2. Рассмотренные выше процессы, такие как: декомпозиция-редукция НС части в текущем структурном состоянии Σ_i ; определение достижимости в данном состоянии; декомпозиция-редукция ДС частей с предварительным выделением позиции p_i^d , определяющей «глубину» редукции – позволяют перейти к представлению более подробного алгоритма определения достижимости ГС, для реализации которого разработана программа.

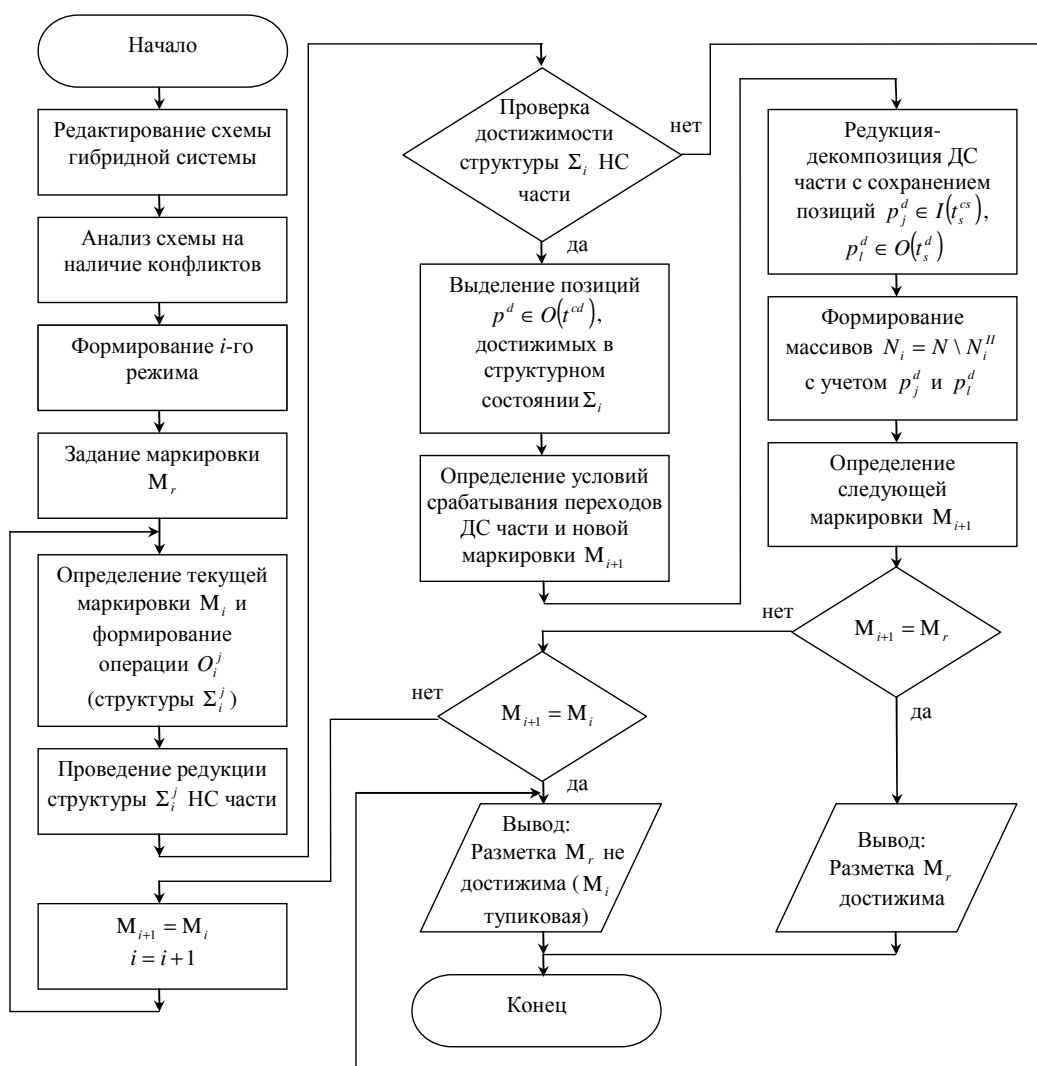


Рисунок 4.2 – Алгоритм анализа достижимости

В представленной на рис. 4.2 блок-схеме алгоритма выделим основные этапы.

1. Формирование текущей структуры и редукция-декомпозиция данной структуры (блоки 1, 2, 3).
2. Формирование связей между ДС и НС частями (блоки 4, 5, 6, 7).
3. Редукция-декомпозиция дискретной части с предварительным формированием горизонта декомпозиции (блоки 7, 8, 9).
4. Формирование следующей маркировки с анализом ее на достижимость путем сравнения ее с заданной и с текущей.

Определение редукции НС части ДН-сети предлагается выполнить при помощи алгоритма, приведенного на рис. 4.3.

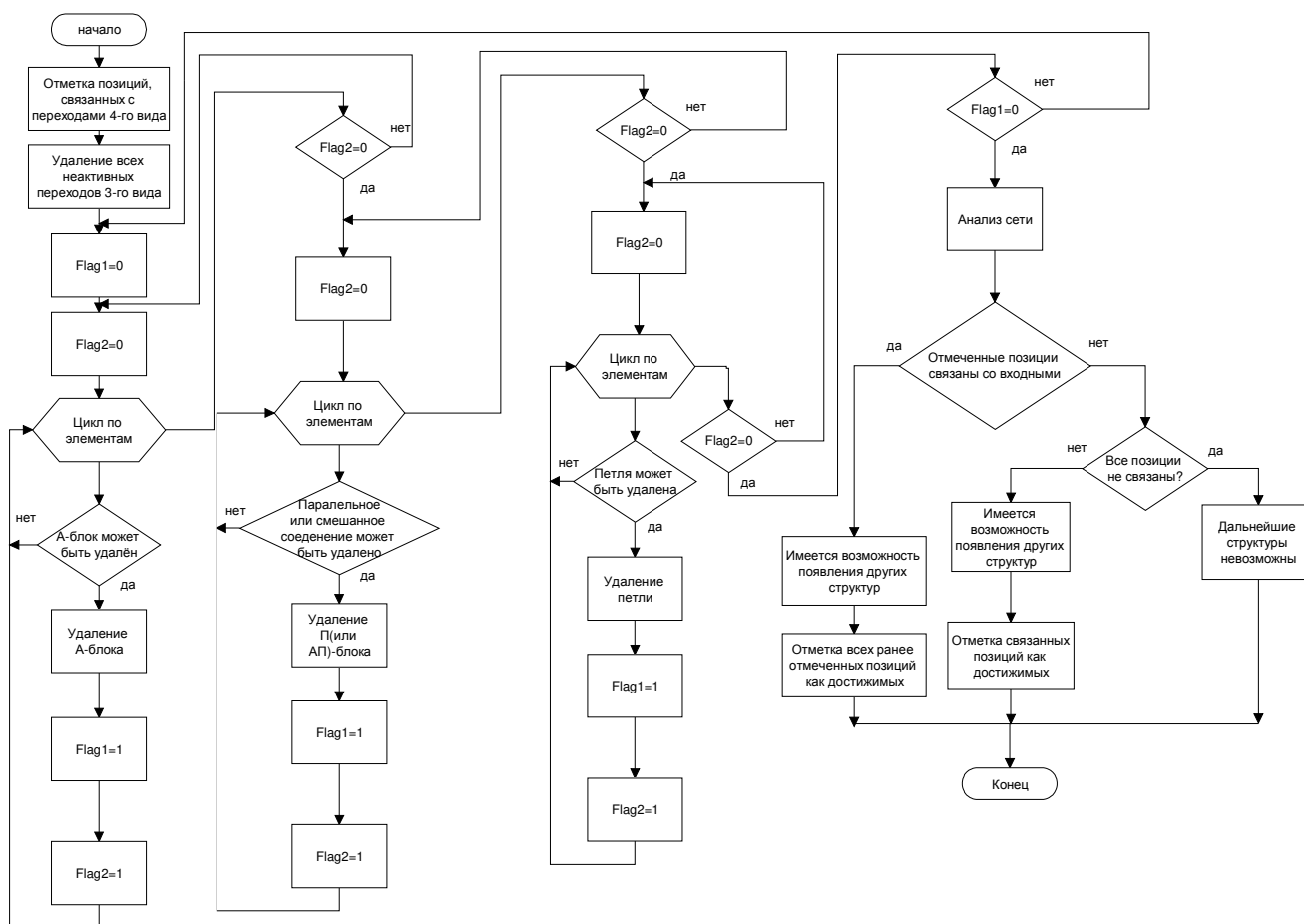


Рисунок 4.3 – Алгоритм проведения редукции текущей структуры Σ_i НС части ГС

В данном алгоритме использована процедура выделения элементов A^c , P^c , или $A^c - P^c$ в текущих структурах Σ_i следующим образом.

1. Производится поиск и выделение $p_i^c \in I(t_i^{cd})$, поиск t_i^{cs} , для которых выполняется условие $p_i^d \in I(t_i^{cs})$, $\mu(p_i^d) = 1$, с последующим удалением t_i^{cs} .
2. Присвоение первому флагу FlagC1, представляющего наличие преобразованных блоков, значения нуля.
3. Присвоение второму флагу FlagC2, обозначающему на данном этапе алгоритма наличие преобразованных A^c элементов (для проверки необходимости последующих итераций), значения нуля.
4. Проводится цикл по всем переходам t_i^c , t_i^{cs} и позициям p_i^c с проверкой, являются ли они частью A^c элемента. Если это так, то реализуется удаление данного элемента. При этом производится корректировка матрицы инцидентности. Значениям обоих флагов присваивается единица. Если после окончания цикла значение FlagC2 не равно нулю, то выполняется переход к шагу 3, поскольку это означает возможность наличия A^c элементов, в противном случае – переход к шагу 5.
5. Присвоение флагу FlagC2 значения нуля (в данном случае FlagC2 является счетчиком преобразованных P^c элементов).
6. Проводится цикл по всем переходам t_i^c , t_i^{cs} и позициям p_i^c с проверкой являются ли они частью P^c элемента. Если это так, то реализуется удаление данного элемента. При этом производится корректировка матрицы инцидентности. Значениям обоих флагов присваивается единица. Если после окончания цикла значение FlagC2 не равно нулю, то выполняется переход к шагу 5, поскольку это означает возможность наличия P^c элементов, в противном случае – переход к шагу 7.
7. Присвоение флагу FlagC2 значения нуля (в данном случае FlagC2 является счетчиком преобразованных L^c элементов).
8. Проводится цикл по всем переходам t_i^c , t_i^{cs} и позициям p_i^c с проверкой являются ли они частью L^c элемента. Если это так, то реализуется удаление

данного элемента. При этом производится корректировка матрицы инцидентности. Значениям обоих флагов присваивается единица. Если после окончания цикла значение FlagC2 не равно нулю, то выполняется переход к шагу 7, поскольку это означает возможность наличия L^c элементов, в противном случае – переход к шагу 9.

9. Производится проверка равенства нулю флага FlagC1. Если это так, то в ходе данной итерации никаких преобразований не было, т.е. элементов типа A^c , P^c и L^c преобразуемая сеть не содержит и возможен переход к следующему шагу. В противном случае производится переход к шагу 2.
10. Производится анализ полученной в результате работы алгоритма непрерывной части сети. Если отмеченные на первом шаге позиции $p_i^c \in I(t_{(\bullet)}^{cd})$ связаны с позициями-истоками, содержащими метки a , то они отмечаются как достижимые. В противном случае делается вывод о недостижимости всех отмеченных выходных позиций.

Алгоритм реализации интерфейса между НС и ДС частями ГС опишем следующим образом.

1. Формирование предикатной функции: если для $p_i^d \in O(t_j^{cd})$ выполняется условие $\mu(p_i^d) = 1$, то тогда $\rho_i(x_i^c, \mu(p_i^d)) = 1$.
2. Выделение для текущей структуры Σ_i позиций $p_j^d \in I(t_s^{cs})$, $p_j^d \in I(t_s^d)$ и $p_i^d \in O(t_s^{cs})$, $p_i^d \in I(t_q^d)$, а также $p_k^d \in O(t_v^{cd})$

Алгоритм редукции дискретной части (рис. 4.4) опишем следующим образом:

1. Присвоение флагу FlagD1, обозначающему наличие преобразованных элементов (для проверки необходимости последующих итераций), значения нуля.
2. Присвоение флагу FlagD2, обозначающему на данном этапе алгоритма наличие преобразованных A^d элементов (для проверки необходимости последующих итераций), значения нуля.
3. Проводится цикл по всем позициям p_i^d и переходам t_j^d с проверкой являются ли они частью A^d элемента. Если это так, то реализуется удаление данного

элемента. При этом производится корректировка матрицы инцидентности. Значениям обоих флагов присваивается единица. Если после окончания цикла значение FlagD2 не равно нулю, то выполняется переход к шагу 2, поскольку это означает возможность наличия A^d элементов, в противном случае – переход к шагу 4.

4. Присвоение флагу FlagD2, обозначающему на данном этапе алгоритма наличие преобразованных Π^d элементов, значения нуля.

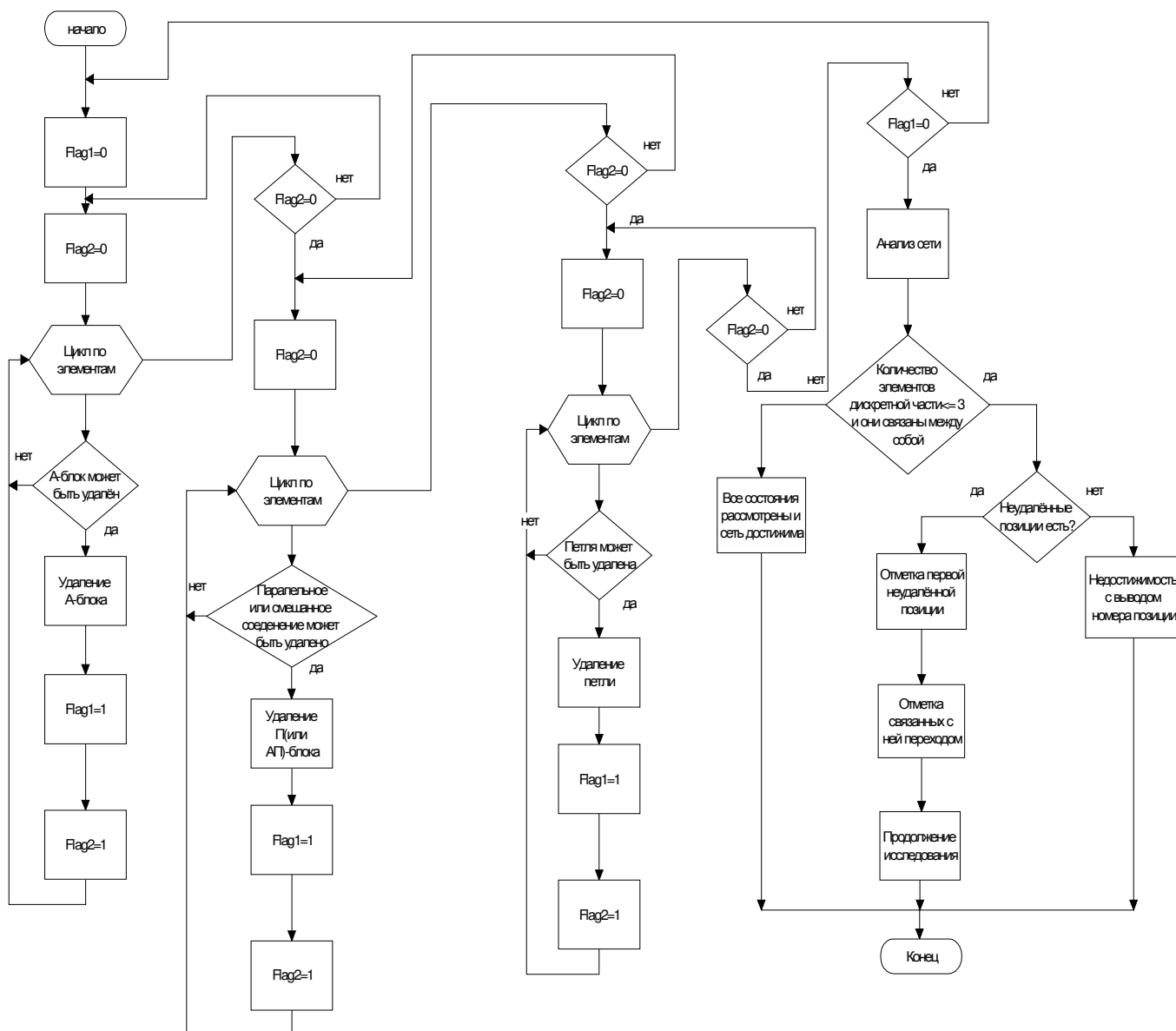


Рисунок 4.4 – Алгоритм редукции дискретной части сети

5. Проводится цикл по всем позициям p_i^d и переходам t_j^d с проверкой являются ли они частью Π^d элемента. Если это так, то проводится редукция данного элемента, при этом производятся изменения матрицы инцидентности. При этом значениям флагов FlagD1 и FlagD2 присваивается единица. Если после окончания цикла значение флага FlagD2 не равно нулю, то выполняется снова проверка всех позиций и переходов, для чего происходит переход к шагу 4. В противном случае, сеть более не содержит Π^d элементов, и производится переход к шагу 6.
6. Присвоение второму флагу, обозначающему на данном этапе алгоритма наличие преобразованных L^d элементов, значения нуля.
7. Проводится цикл по всем оставшимся позициям p_i^d и переходам t_j^d с проверкой являются ли они частью L^d элемента. Если это так, проводится редукция данного L^d элемента. При этом значения обоих флагов, FlagD1 и FlagD2, становятся равными единице. Если после окончания цикла значение флага FlagD2 отлично от нуля, то происходит переход к шагу 6, поскольку это означает возможность наличия L^d элементов, иначе – к шагу 8.
8. Производится проверка равенства нулю флага FlagD1. Если это так, то в ходе данной итерации никаких преобразований выполнено не было, и выполняется переход к следующему шагу, поскольку полученная на данном этапе сеть не содержит элементов, поддающихся преобразованию. В противном случае производится переход к шагу 1.
9. Производится анализ полученной в результате работы алгоритма дискретной части сети. Если количество дискретных элементов меньше либо равно 3 и имеются связи между ними, то делается вывод о достижимости всех состояний. В противном случае проверяется, есть ли неудаленные позиции. Если таковые отсутствуют, то делается вывод о не достижимости конечного состояния. Если есть неудаленные позиции, то производится отметка первой неудаленной позиции и отметка связанных с ней переходов и исследования продолжаются.

Таким образом, полученные алгоритмы позволяют построить базовую ИТ, схема которой приведена на рис. 4.5.

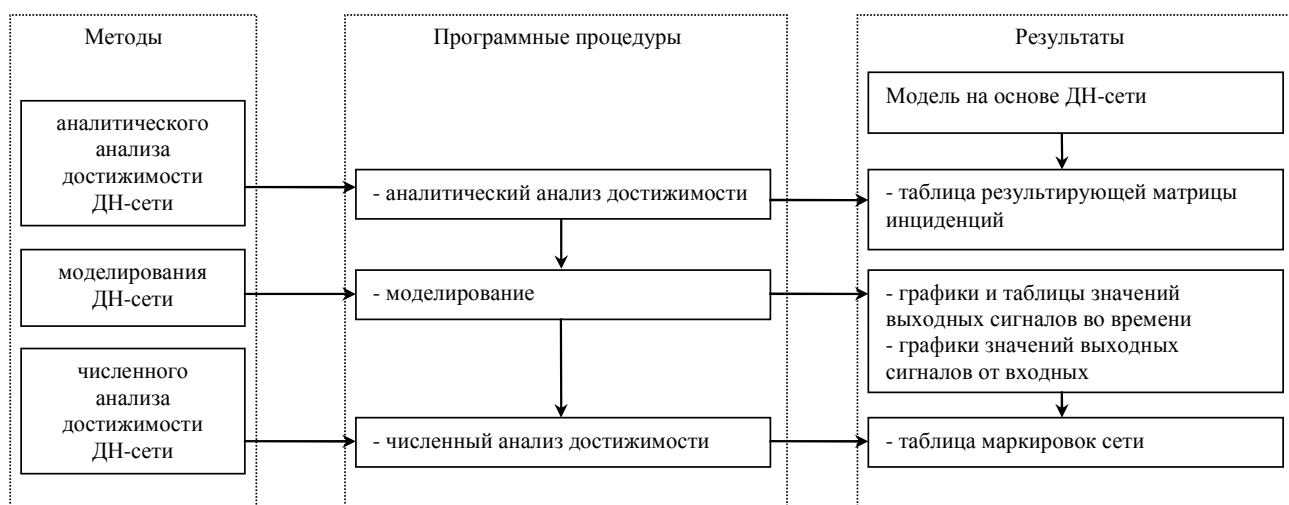


Рисунок 4.5 – ИТ анализа свойства достижимости дискретно-непрерывных систем

Согласно рис. 4.5, в результате проведения анализа достижимости аналитическим методом получаем результирующую матрицу инцидентности, при помощи которой делается вывод о достижимости системы. При применении численного метода [31] анализа вывод о достижимости делается на основании анализа таблицы маркировок ДН-сети либо графиков или таблиц значений выходных сигналов системы.

4.3. Информационная технология общего назначения DCNET

4.3.1. Общая структура ИТ моделирования и анализа достижимости ДНАСУ

Разработанные базовые ИТ моделирования и анализа свойства достижимости использованы при разработке ИТ общего назначения DCNET, предназначенной для моделирования и анализа свойства достижимости ДН АСУ. Упрощенная схема данной ИТ [35] приведена на рис. 4.6. Как видно из рисунка, модель исследуемой системы хранится в файлах моделей, за работу с которыми отвечает модуль работы

с файлами моделей, который совместно с сформированной библиотекой элементов формирует первичную модель исследуемой системы. На первичной модели, которая непосредственно отображается на экране, имеется возможность отображать результаты и ход моделирования, которые также могут сохраняться в графических либо текстовых файлах. Модуль компиляции модели отвечает за предварительную проверку и компиляцию первичной модели в скомпилированную модель, которая служит исходной моделью для проведения моделирования и анализа достижимости с помощью соответствующих модулей. При оптимальном синтезе системы модуль оптимального синтеза проводит неоднократное моделирование скомпилированной модели, решая на каждом шаге оптимизации задачу достижимости. Модуль проверки корректности модели обеспечивает проверку модели на наличие конфликтных ситуаций. Состояние системы может быть сохранено и восстановлено в любой момент времени в БД DCNET при помощи модуля взаимодействия с БД. Также данный модуль обеспечивает взаимодействие с остальными системами комплекса при использовании его в многопользовательском режиме.

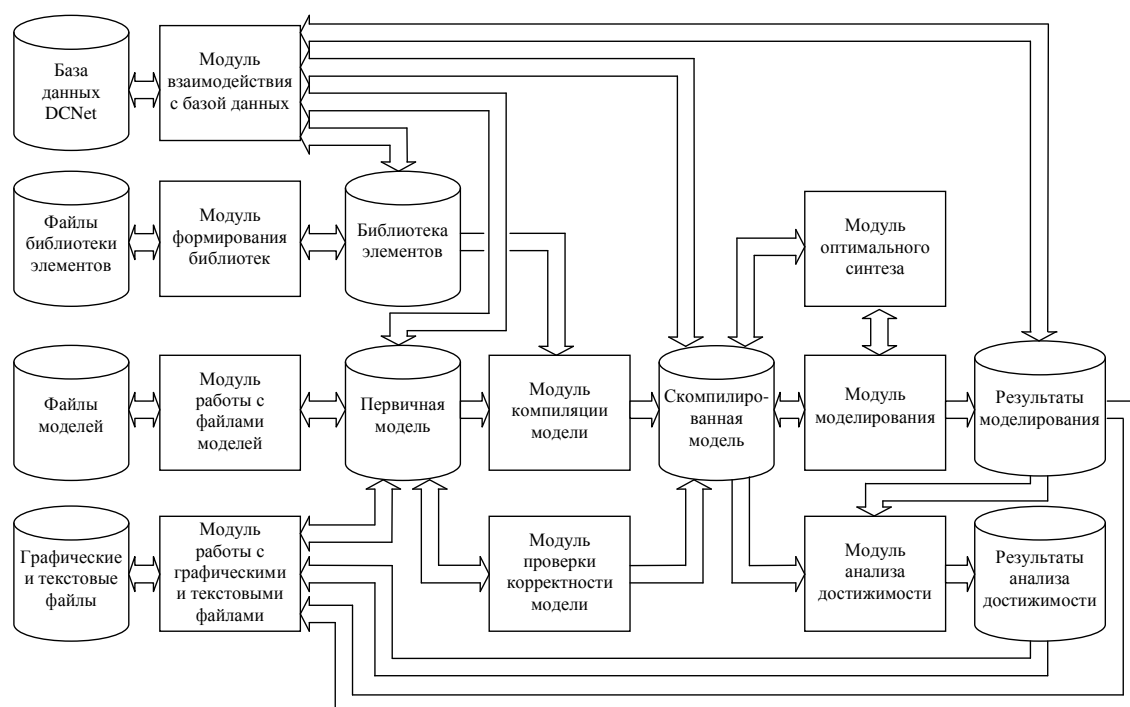


Рисунок 4.6 – ИТ моделирования и анализа ДН АСУ

4.3.2. Требования к ИТ моделирования

Принятая концепция разработки интерфейсов позволили выделить основные режимы управления системой и работы ИТ и системы, разработанной в ее составе, определяющие направления работ при создании интерфейсов системы. Из них можно выделить следующие основные режимы: работы с базой данных, проведение расчетов, анализа, настройки, помощи. Рассмотрим данные режимы более подробно.

Режим работы с БД позволяет проводить подключение с авторизацией пользователя на сервере БД; загрузку и сохранение информации о схеме исследуемой системы, о результатах исследований; проведение поиска и классификации информации об имеющихся в базах моделях, сохраненных результатов работы системы; производить сохранение и восстановление, как исследуемой модели системы, так и результатов ее моделирования.

Режим проведения расчетов позволяет проводить моделирование исследуемой системы как в пошаговом режиме, при котором моделирование приостанавливается при обнаружении требуемого события, так и в режиме, когда моделирование проводится от начала и до конца исследуемого периода времени системы. В данных режимах моделирования автоматически производится построение всех необходимых графиков и таблиц, а также производится отрисовка в реальном времени всех изменений структурного состояния системы, а также разметки дискретно-непрерывной сети системы. В процессе приостановки моделирования возможны просмотр и коррекция параметров моделирования системы, а также изменения разметки дискретно-непрерывной сети.

Режим анализа позволяет проводить анализ исследуемой дискретно-непрерывной сети на корректность, достижимость, как аналитическим методом, так и при помощи смены последовательностей структур, записанных в процессе моделирования.

Режим настройки проводит настройку параметров системы, таких как параметры интерфейса, а также настройки методов и времени, шага моделирования,

шага фиксации результатов, варьируемых параметров модели, методов моделирования.

Режим помощи вызывает контекстно-зависимую справочную информацию обо всех диалогах, режимах работы системы, имеющихся возможностях интерфейса системы.

При создании системы была использована среда разработки Turbo C++ фирмы Borland. Данная среда разработки основана на версии языка программирования C++ фирмы Borland. Она имеет гибкий перенастраиваемый интерфейс и мощную справочную систему. Предпосылками для использования данной среды разработки явились известные достоинства, личное предпочтение автором языка C++ другим языкам и средам разработки, а также опыт работы с языком C++ данной фирмы.

Одним из важнейших критериев, повлиявших на выбор данной среды разработки, явилось то, что она распространяется совершенно бесплатно и в соответствии с условиями лицензирования позволяет разрабатывать коммерческое программное обеспечение.

При разработке ИТ учитывались следующие требования.

1. Относительная независимость систем предусматривает наличие централизованного банка данных и информационный обмен между системами с помощью управления базой этих данных.

2. Эволюционность систем и комплекса в целом позволяет в некоторой степени автоматизировать элементы творчества, т.е. создавать проект на основе накопленного системой опыта.

3. Минимум взаимодействия системы с внешней средой дает возможность создавать максимально автоматизированные системы.

4. Оперативность взаимодействия система – проектировщик предполагает наличие простых для освоения, но развитых процедур взаимодействия между проектировщиком и системой, предлагающей ему альтернативные варианты для продолжения решения, с которым он либо соглашается, либо предлагает собственный вариант.

5. Универсальность автоматизированной системы для групп родственных по характеру работы объектов позволяет снизить затраты на создание системы за счет объединения ресурсов проектирования.

Таким образом, при разработке ИТ основное внимание было сосредоточено на обеспечении все большей независимости ее от разработчика и повышении универсальности в определенном классе задач. Усилия по разработке проектируемой системы были направлены на решение трех основных взаимосвязанных задач: создание и объединение ресурсов системы; разработка информационной базы системы; разработка средств обмена информации.

Использование модульного принципа позволило сформировать иерархическую схему системы применительно конкретной решаемой задачи. Поскольку разработанная ИТ предназначена для проведения расчетов для различных математических моделей, на первое место выходит проблема достоверности математического описания исследуемой системы.

Решение задачи построения ИТ требует использования типовых модулей. Поэтому представление прикладных программ в виде модулей и отдельных процедур по существу является формой унификации правил их составления. Это упрощает их объединение с другими модулями. Чтобы исключить необходимость детального анализа структуры модуля, каждый из них должен сопровождаться информацией о характере решаемой задачи и методе, требуемых ресурсах памяти и внешних устройствах, о взаимодействии модуля с другими, способах модификации, обрабатываемых им исключительных ситуациях и т.д.

4.3.3. Разработка модулей ИТ моделирования и анализа достижимости ДНАСУ

Анализ существующих принципов построения позволил разработать структуру системы, представленную на рис. 4.7 и состоящую из модулей и процедур. Данные модули могут быть разделены на основные, вспомогательные и сервисные. Основные модули служат для решения самостоятельных задач,

вспомогательные – для обеспечения работы основных модулей, сервисные – для вывода и обработки дополнительной информации, облегчающей работу проектировщика. Общий список модулей с их краткой характеристикой каждого из модулей приведены в табл. 4.1, где тёмно-серым цветом выделены модули, разработанные соискателем, модули, в разработке которых соискатель принимал участие, отмечены светло-серым цветом.

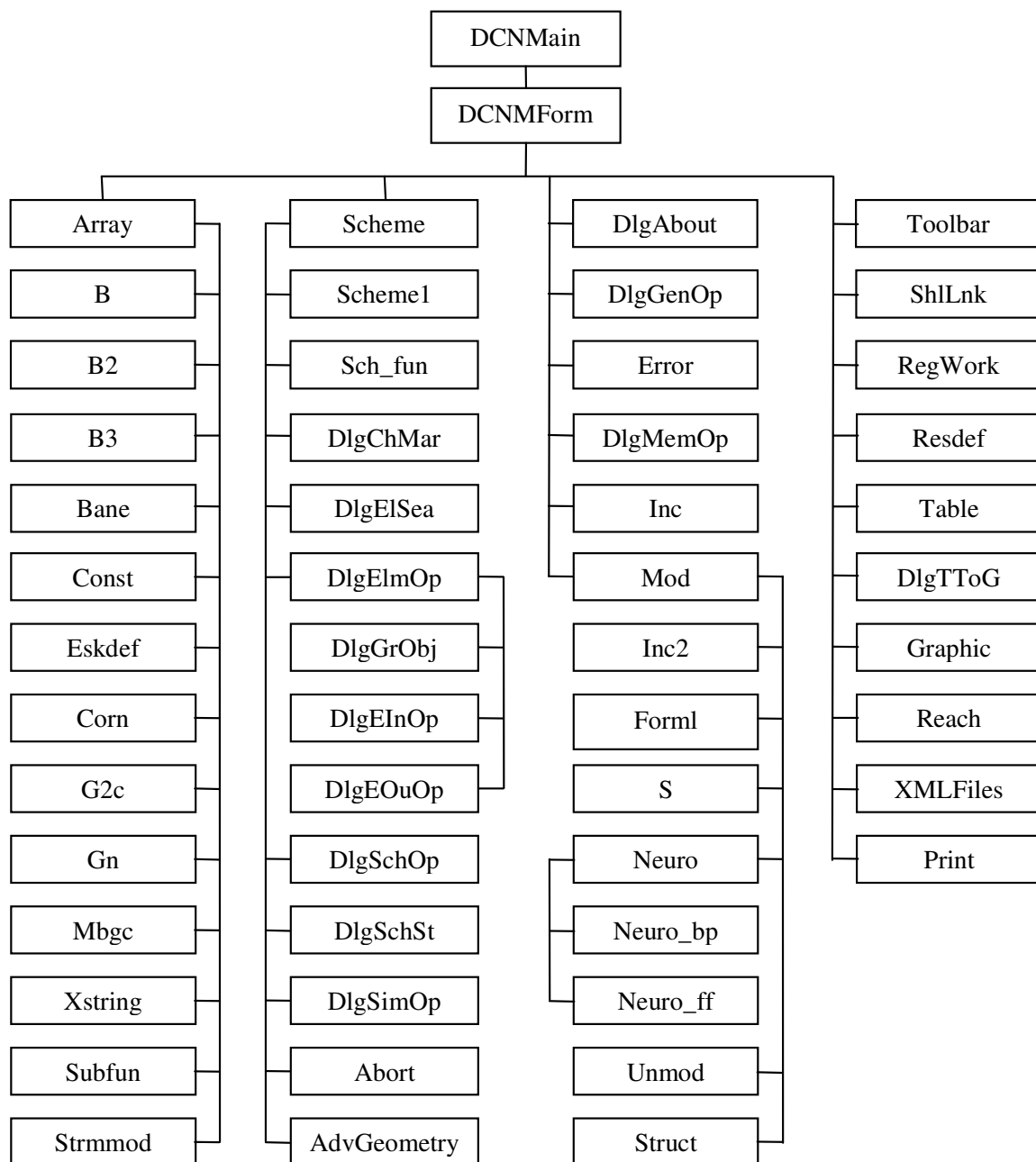


Рисунок 4.7 – Структурная схема системы DCNET

Таблица 4.1 – Основные модули DCNET

Обозначение	Функциональное назначение
Abort	Прерывание текущей операции
AdvGeometry	Обеспечение отрисовки графических элементов
Array	Работа с коллекциями
B	Анализ модели на корректность
B2	Анализ параметров элемента на корректность
B3	Реализация поведения типовых элементов
Bane	Основной модуль компиляции модели
BWCCL	Обеспечение внешнего вида диалоговых окон
Const	Описание основных констант
Corn	Синтаксический анализ полей элемента
DCNet	Основной модуль
DCNMForm	Главное окно программы
DlgAbout	Справка о программе
DlgChMar	Изменение числа маркеров позиций
DlgElInOp	Редактирование информации о входах элемента
DlgElmOp	Редактирование информации о элементе
DlgElSea	Поиск элемента на схеме
DlgEOuOp	Редактирование информации о выходах элемента
DlgGenOp	Редактирование настроек программы
DlgGrObj	Редактирование информации о графическом элементе
DlgMemOp	Редактирование настроек памяти
DlgSchOp	Редактирование настроек схемы
DlgSchSt	Редактирование информации о состоянии системы
DlgSimOp	Редактирование настроек моделирования
DlgTToG	Передача данных из таблиц в графики
Error	Сообщения об ошибках
Eskdef	Константы и настройки, относящиеся к моделировщику

Продолжение табл. 4.1

Обозначение	Функциональное назначение
XMLFiles	Работа с XML
Forml	Реализация поведения элементов - логических формул
G2c	Работа с варьируемыми параметрами
Gn	Работа с параметрами элемента
Graphic	Работа с графиками
Inc	Описание структур связей модели для моделирования
Inc2	Реализация поведения элементов при моделирование
Main	Основной модуль
Mbgc	Заполнение полей элемента
Mod	Моделирование схемы
Neuro	Основной модуль работы с нейронными сетями
Neuro_bp	Вспомогательный модуль нейронных сетей
Neuro_ff	Вспомогательный модуль нейронных сетей
Print	Обработка вывода на принтер
Reach	Исследование ДН-сети на достижимость, корректность и пр.
RegWork	Работа с реестром
Resdef	Определение ресурсов системы
S	Реализация поведения элементов - математических формул
Sch_fun	Вспомогательный модуль работы со схемой
Scheme	Основной модуль работы со схемой
Scheme1	Вспомогательный модуль работы со схемой
ShlLnk	Работа со связями Windows
Strmmod	Работа с потоками
Struct	Описание основных структур модели
Subfun	Вспомогательный модуль
Table	Работа с таблицами
Toolbar	Управление панелями инструментов

Продолжение табл. 4.1

Обозначение	Функциональное назначение
Unmod	Очистка после моделирования
Xstring	Работа со строками

Модульностью программы обеспечивается работа с библиотекой моделей, представленной программными модулями, реализующими описания различных элементов модели, а также информационными модулями, описывающими различные параметры элементов.

Системный принцип, положенный в основу разработанной программы потребовал использования единой информационной базы данных, хранящих информацию о модели, что позволило унифицировать хранимую информацию об элементах модели и связях между ними. Таким образом, информационное обеспечение является функций управляющей программы.

Программное обеспечение ИТ может быть разделено на два уровня: обеспечивающие и функциональное. Обеспечивающие ПО отвечает за ввод и коррекцию модели системы, коррекцию разметки сети, сохранение и восстановление модели в файлах либо на сервере баз данных, отображение и работу с результатами моделирования, а также за настройку параметров интерфейса путем управления работой различных систем. Функциональное ПО состоит из модулей решения задач расчета и анализа всех характеристик ТО и САУ в целом, а также проведения имитации функционирования ДНАСУ. Информационное обеспечение систем состоит из баз данных библиотеки элементов, моделей, результатов моделирований. Общая структура приведена на рис. 4.8.

Анализ связей модулей и систем позволяет сделать вывод о возможности независимой разработки при условии организации общего доступа к банку данных системы. Данная особенность учитывалась в дальнейшем при организации архитектуры системы и разработке банка данных.

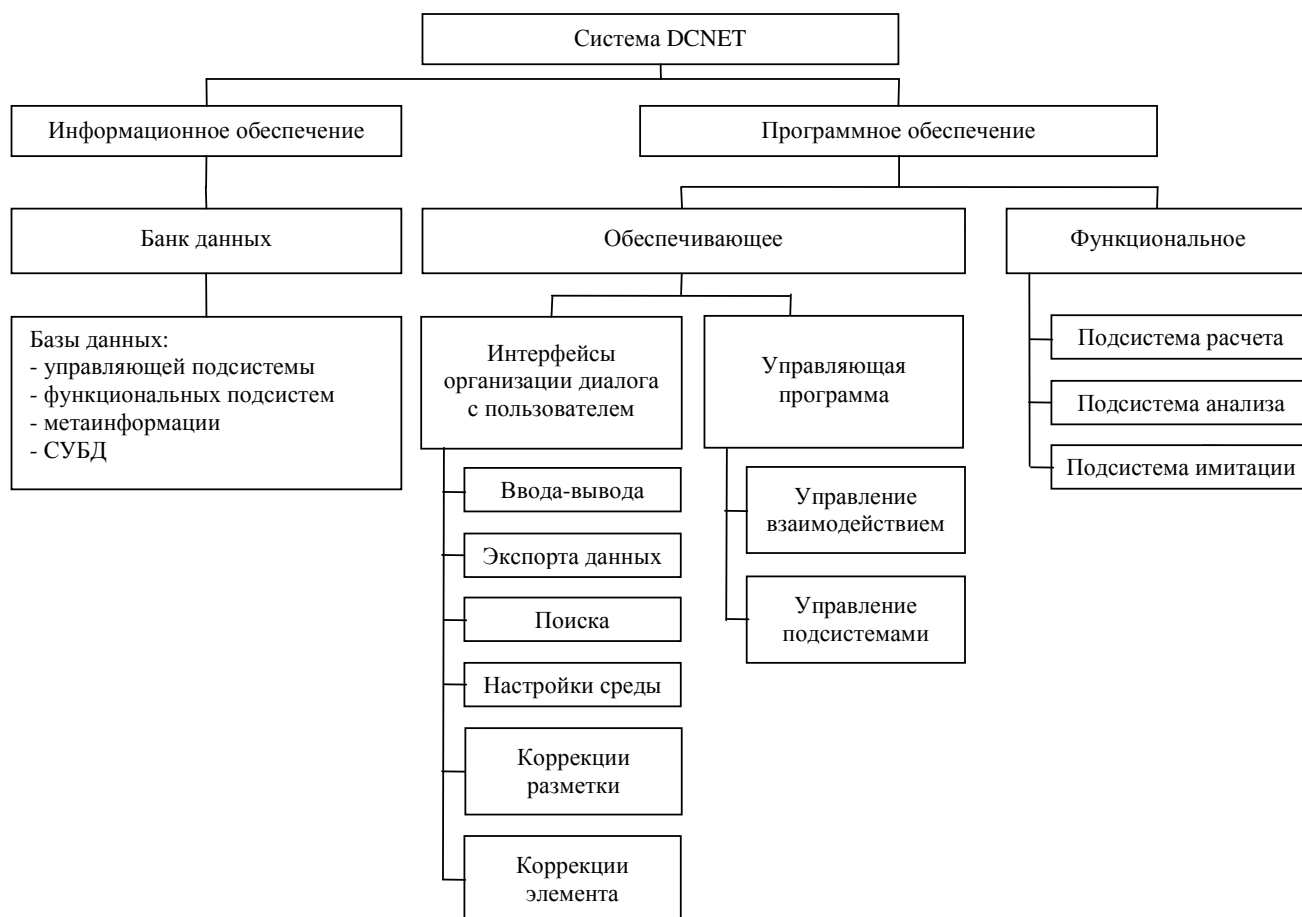


Рисунок 4.8 – Общая структура системы DCNET

4.3.4. Разработка архитектуры системы

Анализ существующих подходов разработки программных средств показывает, что при разработке современных программных комплексов возможно использование следующих подходов, отличающихся по своей архитектуре.

1. Традиционный подход, когда при разработке программного комплекса разрабатывается несколько отдельных, независимых программных модулей и связь между ними проводится при помощи различных процедур экспорта и импорта данных.

2. Подход, основанный на централизованной архитектуре, когда разрабатывается один основной модуль, а все необходимые дополнения реализуются при помощи подключения дополнительных библиотек динамической компоновки.

Каждый из вышеупомянутых подходов имеет свои достоинства и недостатки, обусловленные реализованными особенностями строения системы. Так, при традиционном подходе необходимо обеспечивать наборы различных конверторов данных, для преобразований одного типа в другой необходимо запускать отдельные исполняемые файлы, что замедляет работу всей системы, заставляет пользователя запоминать какой модуль для чего предназначен, увеличивает нагрузку на дисковую систему компьютера и т.п. Напротив, подход, основанный на централизованной архитектуре, практически лишен вышеперечисленных недостатков, однако из-за наличия единой системы требует для выполнения какого-либо действия с данными наличие как самой системы, так и дополнительной библиотеки динамической компоновки, что вызывает необходимость наличия на каждом компьютере иногда значительных ресурсов для выполнения даже наипростейших операций из-за повышенных требований самой оболочки.

При разработке программного комплекса «Дискретно-непрерывные сети» (DCNET) было принято решение воспользоваться подходом, основанным на централизованной архитектуре, используя при этом некоторые элементы традиционного подхода. Использование централизованной архитектуры позволяет предоставить пользователю единую среду, имеющую некоторый минимальный набор функциональности и позволяющую производить моделирование исследуемой системы, а также исследование ее на достижимость, а всю остальную функциональность возложить на дополнительные внешние модули, выполненные в виде библиотек динамической компоновки, а также в виде отдельных исполняемых модулей. Данный подход позволяет, сохранив преимущества централизованного подхода, легко наращивать функциональность системы, а также использовать некоторые ее элементы по отдельности. С другой стороны, при реализации центральный элемент архитектуры получается небольшим по размеру и не требующим для исполнения значительных ресурсов [41, 81, 83].

Использованная при разработке системы архитектура представлена на рис. 4.9. Как видно из рисунка 4.9 каждая из систем комплекса является полностью автономной, однако может объединяться с другими системами для решения каких-

либо сложных задач, связываясь с ними при помощи передачи данных от одной системе к другой. При этом обмен данными на себя берут специальные сервера баз данных. При этом любая из систем, участвующая в комплексе может при необходимости взять на себя основную роль по управлению всем комплексом. При этом ее основная задача состоит в объединении информационных ресурсов систем в единый банк данных, создающий единое информационное пространство для работы систем функционального программного обеспечения.

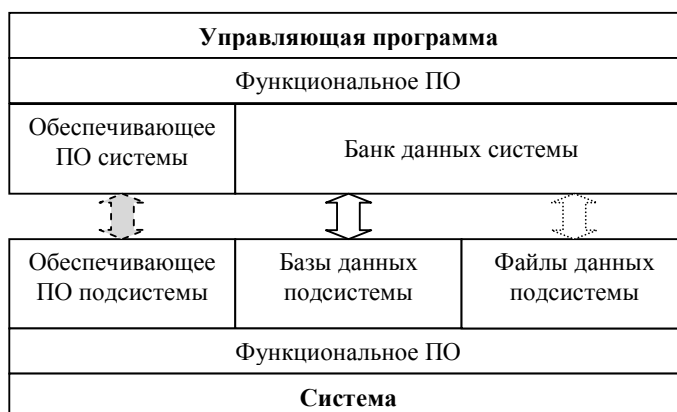


Рисунок 4.9 – Архитектура программного комплекса

Обеспечивающее программное обеспечение (ПО) управляющей системы вызывает необходимую систему для решения конкретной задачи. Обеспечивающее ПО системы организует процесс решения задачи: диалог с пользователем, ввод и вывод данных, запуск функционального ПО системы. Исходные данные и результаты работы функционального ПО системы сохраняются в БД системы. После окончания работы результаты работы системы сохраняются в основной БД, которая отвечает за репликацию изменившихся данных на все компьютеры, входящие в комплекс. Таким образом, может быть гарантирована работа всех частей комплекса как единого целого для решения конкретных задач.

Архитектура каждой из систем является единой для всего комплекса и приведена на рис. 4.10.

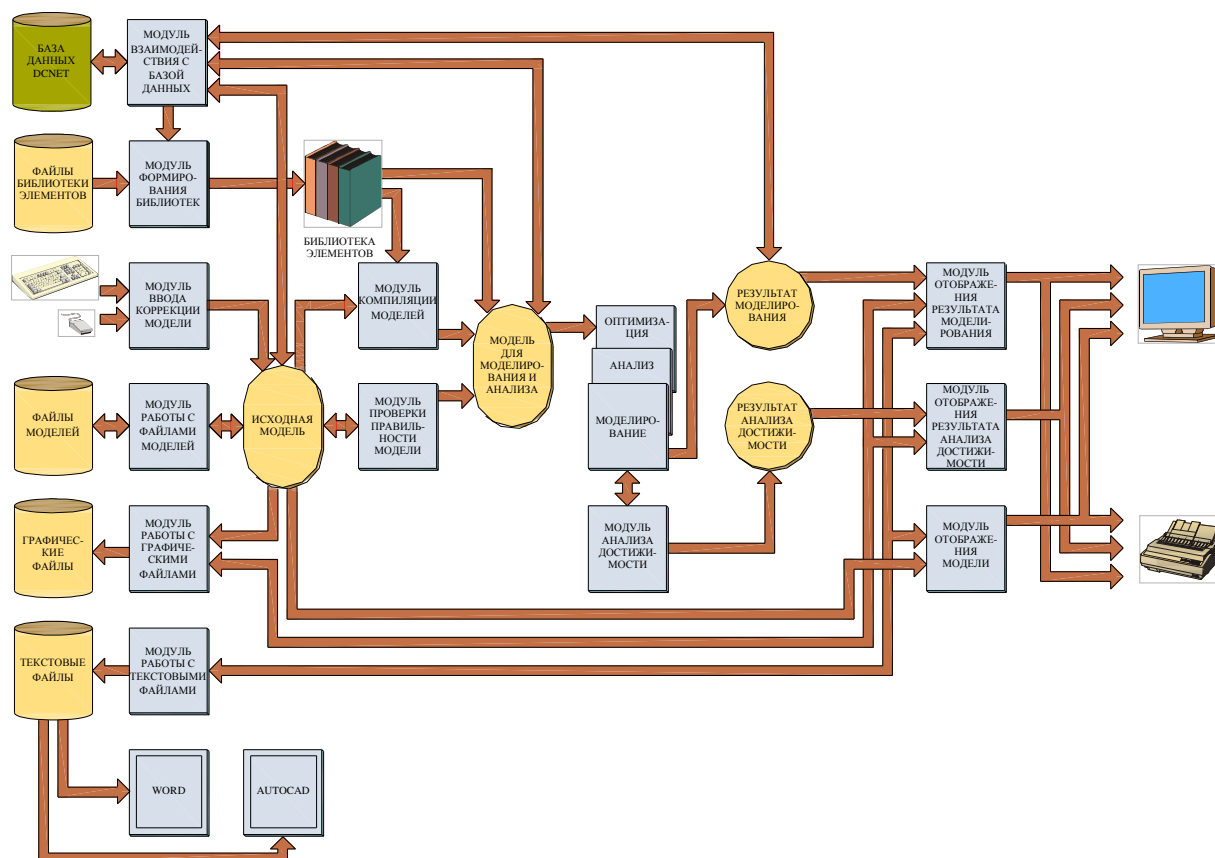


Рисунок 4.10 – Архитектура системы DCNET

Для доступа к данным в системе DCNET используется реляционная база данных [77, 78], работающая под управлением свободно распространяемого сервера PostgreSQL, обеспечивающего физическую и логическую независимость данных. Совокупность таблиц образует единую базу данных DCNET, которая является единой базой для всех систем комплекса. Основными задачами отдельных систем являются:

- просмотр списков объектов системы (списков нормативов, справочников, оперативных массивов, сформированных отчетов);
- просмотр и редактирование любого справочника системы в едином стиле (режим таблицы, режим формы);
- обеспечение сервисных функций (работа с буфером обмена, печать справочника и т.п.);
- вызов информации справочной системы, относящейся к текущему открытому объекту;

- вывод дополнительной, формируемой пользователем, информации по текущему объекту в виде подсказок и информационной панели;
- контроль вводимой пользователем информации и вывод диалоговых сообщений.

Разделение программной логики системы на уровни является результатом функциональной декомпозиции задач системы, при которой группы функций сгруппированы в четыре большие группы:

1. функции представления данных и организации диалога с пользователем;
2. функции взаимодействия системы с базой данных;
3. функции обработки данных в системе; 4) функции управления работой системы.

Каждая группа функций предполагает реализацию в виде совокупности функций и процедур (программной логики), взаимосвязанных между собой для решения своей области задач. Логика представления данных и организации диалога, логика взаимодействия с базой данных, логика управления системой представляет собой часть обеспечивающего программного обеспечения, единого для всех систем.

Логика взаимодействия системы с базой данных организована на основе архитектуры универсального доступа к данным (Microsoft Universal Data Access architecture), включающей в себя следующие элементы.

- Microsoft ActiveX Data Objects (ADO) представляет собой интерфейс прикладного программирования для доступа к данным, хранящимся в различных источниках. ADO и его расширения являются упрощенным высокоуровневым объектно-ориентированным интерфейсом для работы с OLE DB;

- OLE DB – метод доступа ко всем имеющимся данным через стандартный COM-интерфейс, независимо от того где и как эти данные сохраняются. OLE DB представляет собой низкоуровневый интерфейс доступа к данным (рис. 4.11);

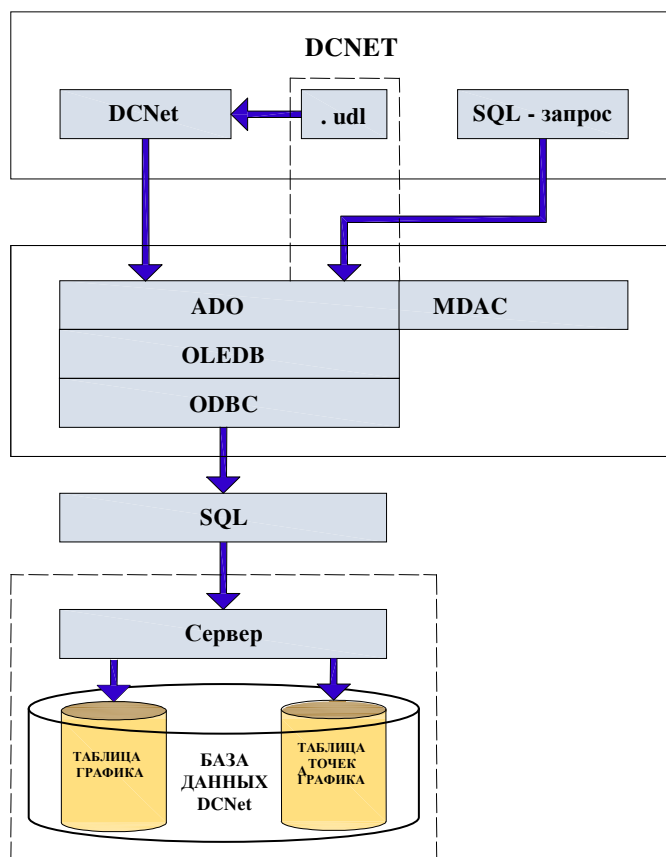


Рисунок 4.11 – Архитектура доступа к базе данных ИТ DCNET

- Open Database Connectivity (ODBC) является стандартом Microsoft для работы с реляционными базами данных. ODBC используется для совместимости с более ранними разработками.

Необходимо также отметить, что для организации автономного функционирования каждой из отдельных систем DCNET архитектура предусматривает наличие помимо интерфейса работы с СУБД интерфейс работы с отдельными файлами. В файлах может быть сохранена вся информация, обычно сохраняемая в базе данных системы, а затем восстановлена в DCNET. Такой подход позволяет повысить при работе надежность всего комплекса в целом, поскольку он представляет собой распределенную систему и в случае аварии линий связи либо центрального сервера базы данных позволит сохранить результаты работы на локальном компьютере для последующей передачи их на центральный сервер после устранения аварии. Также такой подход позволяет временно сохранить набор

вариантов моделей и результатов, когда разработчик не уверен в окончательности и правильности их и не желает делать их доступными для других пользователей комплекса.

Разработанная архитектура учитывалась при разработке базы данных и интерфейса системы DCNET и ее модулей.

Для исследуемого объекта характерным является, как было отмечено ранее, большая размерность дискретно-непрерывной сети, описывающей данный объект, а также сложность взаимосвязей и разнородность информационных элементов. Также необходимо отметить отсутствие однозначных алгоритмов решения отдельных задач в рамках системы, и, как следствие, необходимость их модификаций в процессе разработки исследуемого объекта. Кроме того необходимо предусмотреть инвариантность функционирования системы DCNET к изменяющейся структуре и составу информационной базы при расширении функциональных возможностей отдельных ее модулей. Исходя из этого, можно сделать вывод, что эффективное функционирование всего комплекса возможно только при использовании единых подходов к его информационному обеспечению, а также при взаимодействии отдельных систем через общесистемные интерфейсы доступа к данным. Таким образом, на основе общей архитектуры системы можно представить архитектуру информационного обеспечения в виде, представленном на рис. 4.9.

База данных управляющей программы представляет собой хранилище общей для всех систем информации, а также информации, используемой для управления работой как системы в целом, так и для отдельных систем DCNET.

Банк данных включает в себя систему управления базами данных, базу метainформации и данные управляющей системы.

В качестве системы управления базами данных выбрана СУБД PostgreSQL как обладающая необходимыми инструментами, являющаяся свободно распространяемой и позволяющая развернуть разрабатываемую систему, как на отдельных рабочих станциях, так и на серверах сети.

4.3.5. Требования к программным и аппаратным ресурсам

Система DCNET разрабатывалась для использования под ОС Microsoft Windows. При разработке системы были использованы стандартные библиотеки, механизмы и технологии, входящие в состав Windows 98SE/NT/2000/XP. Система не предъявляет жестких требований к аппаратному обеспечению компьютера (благодаря механизмам самой ОС), занимает всего несколько мегабайт дискового пространства (основной модуль) и может быть запущена в минимальном варианте практически на любом компьютере, позволяющем по своим характеристикам установить и запустить саму ОС серии MS Windows. Также возможен запуск системы под управлением любой другой ОС, имеющей в своем составе эмуляторы среды ОС Windows. Установленное количество оперативной памяти и мощность процессора влияют только на скорость вычислений. Необходимо также отметить, что при использовании всего комплекса в распределенном варианте налагаются естественные требования на наличие аппаратуры связи между компьютерами комплекса, а также требования, налагаемые установкой используемой версии СУБД PostgreSQL. На основании анализа, проведенного в разделе 1, вытекает, что прикладной проблемой, для реализации которой разрабатывается визуализированная система, являются обеспечение автоматизации процесса моделирования и анализа свойств системы, решение отдельных задач проектирования и эксплуатации сложных технических и социально-экономических систем управления. Модель прикладной области разрабатывается на основе свойств и особенностей ДН-сетей.

4.3.6. Реализация пользовательского интерфейса

Одной из основных задач при разработке системы является задача разработки программного интерфейса. При решении ее в процессе разработки программы DCNET было установлено, что наиболее приемлемым вариантом интерфейса является привязка элементов управления непосредственно к источникам данных.

Таким образом, для реализации выбранного подхода необходимо разработать набор отдельных интерфейсов для каждого информационного элемента.

Метафора ДН-сетей определяет язык визуализации, построенный на ее базе. Это ограничение определяется следующим.

1. В ДН-сетях непрерывная динамика реализуется функциональными элементами, имеющими вход и выход; структура и связи между ними определены структурной схемой, представленной МСГ.
2. Дискретно-событийная динамика представлена блок-схемой, описанной расширенной СП.

Метафорой ДН-сети является метафора, основанная на объединении метафоры функциональных схем и метафоры блок-схем. Если метафора функциональных схем позволяет отразить непрерывную сущность объекта исследования, то метафора блок-схемы – отразить событийную сущность и представить поток управления НС системой и экспериментом в целом.

Проектируемый на базе ДН-сетей визуальный язык должен быть основан на потоке преобразования непрерывного сигнала, потоке преобразования дискретно-событийного сигнала и потоке управления (изменение маркировки дискретных позиций), а также на потоке, связанном с движением меток a по непрерывным позициям. Если программы, основанные на визуализации двух первых потоков, известны [62 – 64], то разработка программы визуализации, основанной на визуализации трех потоков выполняется впервые.

Словарь языка визуализации состоит из единичных элементов языка, представляющих собой условное двумерное графическое представление составляющих структурных схем САР и блок-схем АСУТП. Представление и число видов отображений можно получить из информационной схемы, приведенной на рис. 4.12. Словарь первичных элементов содержит записи о каждом типе элемента.

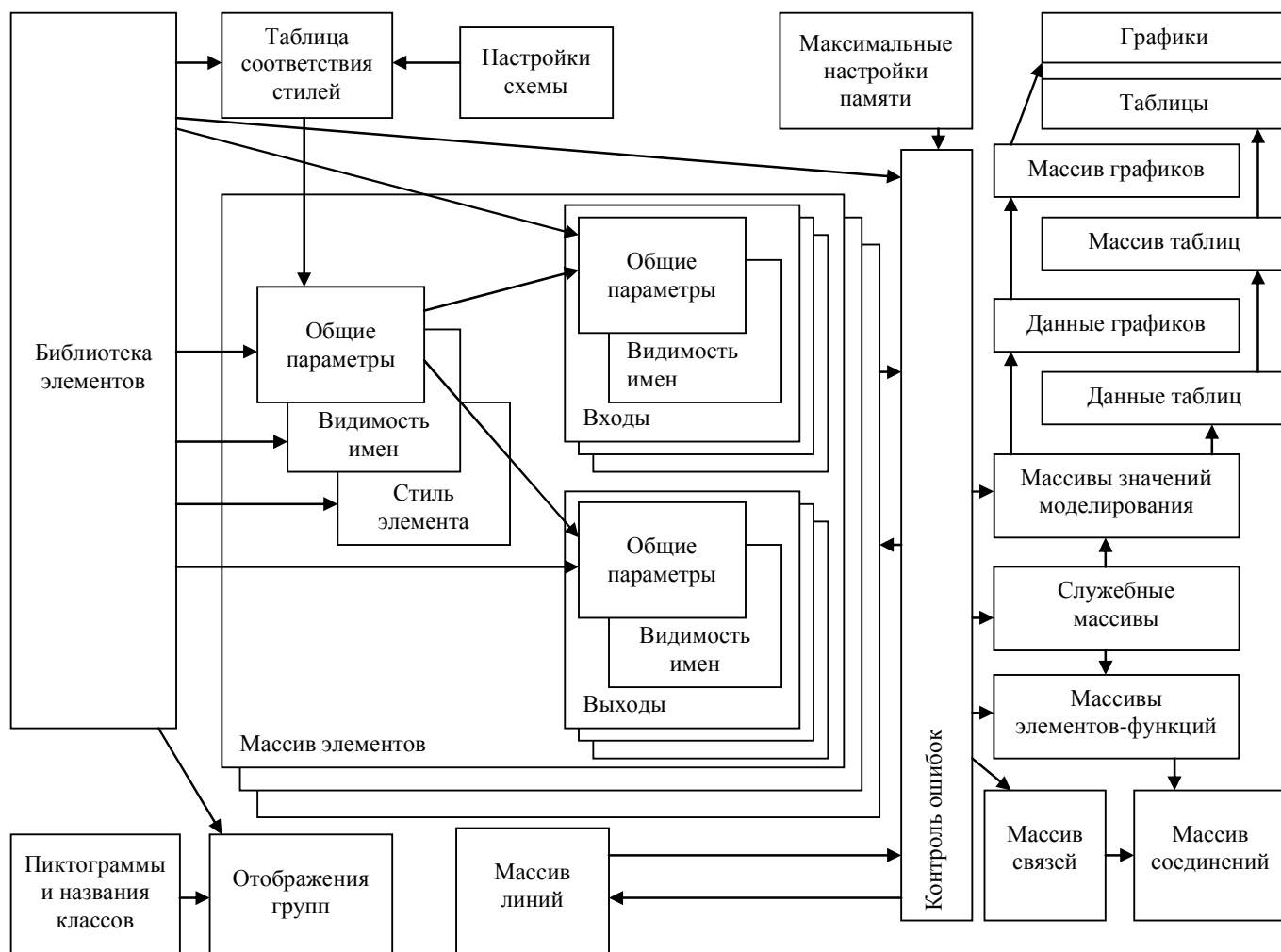


Рисунок 4.12 – Информационная схема связей классов и массивов хранения информации

Каждая запись имеет структуру, приведенную на рис. 4.13. Полный формат полей данных структуры ВТВ приведен в табл. 4.2.

Уникальное имя				Тип				Стиль				...				Процедура обработки			
1	2	...	14	15	...	18	19	...	22	...	263	...	277						

Рисунок 4.13 – Структура записи первичных элементов

Таблица 4.2 – Поля структуры записи данных элемента

Тип	Имя	Предназначение
char	bna [LNAME]	Уникальное имя
int	tip	Тип
unsigned int	Style	Стиль
unsigned int	ccStyle	Возможность изменения стиля
unsigned int	Visible	Видимость частей элемента
int	Control	Управляемость элемента
int	nuu	Количество входов
int	minnuu	Минимальное количество входов
int	maxnuu	Максимальное количество входов
int	cnuu	Количество управляющих входов
int	mincnuu	Минимальное количество управляющих входов
int	maxcnuu	Максимальное количество управляющих входов
unsigned int	ciuu	Вход может быть ингибиторным непрерывным
unsigned int	ciduu	Вход может быть ингибиторным дискретным
unsigned int	cssu	Вход может изменить непрерывный флаг
unsigned int	cduu	Вход может изменить дискретный флаг
unsigned int	ciuu	Вход может изменить ингибиторный флаг
unsigned int	csu	Вход – непрерывный флаг
unsigned int	dsu	Вход – дискретный флаг
unsigned int	isu	Вход – ингибиторный флаг
int	nxx	Количество выходов
int	minnxx	Минимальное выходов
int	maxnxx	Максимальное количество выходов
int	cnxx	Количество управляющих выходов
int	mincnxx	Максимальное количество управляющих выходов
int	maxcnxx	Максимальное количество управляющих выходов
unsigned int	ciexx	Выход может быть ингибиторным непрерывным

Продолжение табл. 4.2

Тип	Имя	Предназначение
unsigned int	cidxx	Выход может быть ингибиторным дискретным
unsigned int	csxx	Выход может изменить непрерывный флаг
unsigned int	cdxx	Выход может изменить дискретный флаг
unsigned int	cixx	Выход может изменить ингибиторный флаг
unsigned int	sxx	Выход – непрерывный флаг
unsigned int	dxx	Выход – дискретный флаг
unsigned int	ixx	Выход – ингибиторный флаг
int	npp	Количество параметров
int	minnpp	Минимальное количество параметров
int	maxnpp	Максимальное количество параметров
char	SchName[LTXT]	Имя элемента на схеме
char	nar[LTXT]	Имя элемента в меню

Полная длина записи составляет 277 байт. Дополнительно к этой информации хранятся пиктограммы для каждого элемента и откомпилированные процедуры обработки. Примеры пиктограмм, представленные по классам, приведены на рис 4.14. Более подробная расшифровка пиктограмм элементов и их классов приведена в приложении Б.

Каждый элемент в словаре первичных элементов должен быть отнесен к одному из восьми приведенных классов (число их может быть увеличено). Каждый из этих классов имеет свои атрибуты, которыми являются название класса, его номер, пиктограмма, а также начальный и конечный номера элементов, принадлежащих данному классу. Как видно из рисунка, каждый класс визуально представлен в виде соответствующей панели инструментов, содержащих кнопки, соответствующие элементам класса. Отображение панелей управляется кнопками, содержащими соответствующие пиктограммы классов. В левом верхнем углу приведен класс «ДНС» общего представления позиций и переходов. В нижнем левом углу – класс «Результаты», содержащий элементы визуализации результатов

моделирования (графики и таблицы). Отображения конкретных переходов приведено в приложении Б.

Рассмотрим отображение, появляющееся после установки перехода типа Transf (Передающая функция) в окне схемы, приведенное в приложении Б. Обратим внимание на структуру записи, приведенной на отображении «Входы» (рис. 4.14).

Тип				Маркировка				Имя				Коэффициент				Номер входа			
1	2	...	4	5	...	12	13	...	22	23	...	26	27	...	30				

Рисунок 4.14 – Структура записи «входы»

В поле «Тип» записывается признак входа (непрерывный, дискретный, ингибиторный), в поле «Маркировка» – единицы измерения входного сигнала ($^{\circ}\text{C}$, $\text{м}^3/\text{ч}$ и др.), в поле «Имя» – название физического параметра (температура, расход и др.), в поле «Коэффициент» – коэффициент передачи входного сигнала, в поле «Номер входа» – номер входа на пиктограмме перехода (сверху вниз). Следует обратить внимание, что такие признаки, как маркировка и имя, впервые введены в автоматизированную систему решения смешанных дискретно-непрерывных уравнений.

Исходя из требований по управлению вышеперечисленными полями, внешний вид интерфейса редактирования параметров элементов удобно сгруппировать по функциональному признаку: общие настройки (ответственные за количество соединений, номер элемента и параметры, определяемые функциональным назначением элемента), видимость имен (ответственные за возможность отображения имени элемента, его параметров, а также их месторасположение) и стиль (определяющий поведенческие свойства элемента). Также удобно вынести аналогичные свойства входов и выходов в отдельные диалоговые окна. В результате получаем интерфейс редактирования параметров элемента, описанный в приложении Б.

При изменении различных режимов работы системы курсоры в окне редактирования схемы также должны изменять свой вид с целью дополнительного информирования пользователя о текущем режиме работы системы. Виды курсоров окна схемы в различных режимах сведены в приложении Б.

Указатель «Элемент» показывает, что система находится в режиме редактирования «Установка элемента» (перехода или позиции); указатель «Карандаш» – в режиме «Проведение соединений», и в случае, если карандаш прозрачный, означает что в данный момент не проводится линия, иначе – карандаш зачернен; указатель «Кисть» – в режиме коррекции маркировки дискретных позиций ДН-сети.

При нажатии кнопки Help на окне параметров элемента Transf получим описание параметров элемента.

Примеры отображений позиций различных типов и их параметров приведены на рисунке 4.15. Слева показан (рис. 4.15,а) пример непрерывной позиции p_5^c , помеченной непрерывным сигналом x_1^c , значение которого в текущий момент времени составляет 43,2952. Внутри дискретной позиции p_{11}^d (рис. 4.15,б) установлен признак наличия маркеров, число которых указано справа в круглых скобках [39, 90].

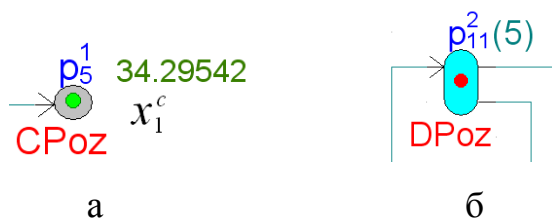


Рисунок 4.15 – Пример представления позиции и их атрибутов: а – непрерывная позиция, б – дискретная позиция

С целью упрощения работы со схемой исследуемой системы в системе DCNET реализована система поиска элемента. Она позволяет проводить поиск элементов по различным признакам и устанавливает курсор мыши на найденный элемент, либо выдает сообщение об ошибке в случае неудачи.

Поскольку хранение информации о модели для моделирования организовано в виде массивов, то возникает необходимость управления их размером путем задания максимально возможного числа элементов массивов.

Грамматические правила построения языка моделирования построены на требованиях по использованию словаря и установления правил формирования схемы ГС на основе единиц языка: две единицы языка, такие как элементы, не могут соприкасаться друг с другом в точках входа/выхода; два или более выхода одного или нескольких элементов не могут быть соединены между собой; позиции, относящиеся к разным типам, не могут быть соединены между собой; установка внутренних параметров элементов должна быть обеспечена строго в соответствии с разработанными правилами для каждого из элементов; внешние атрибуты элементов должны соответствовать графическому изображению типа элемента и могут задаваться в относительных координатах элемента. Реализация вышеперечисленных требований возлагается на модули: ввода/коррекции модели, проверки правильности модели, компиляции модели с использованием библиотеки элементов. Результатом работы модулей является исходная модель ДНАСУ. Модуль ввода/коррекции модели формирует исходную модель, исходя из информации, получаемой из библиотеки элементов, введенная пользователем информация об элементе интерпретируется в его параметры, которые впоследствии могут быть изменены путем двойного щелчка мышью на пиктограмме требуемого элемента.

4.3.7. Работа с программным комплексом

Процесс работы с системой представлен на рис. 4.16 в виде следующей последовательности: сначала производится набор и редактирование элементов схемы исследуемой ДНАСУ при помощи редактора схемы и библиотеки элементов, при этом редактор позволяет проводить редактирование схемы исследуемой ДНАСУ с использованием библиотеки составных элементов (как блоков, так и готовых систем) для работы с иерархическими ДНАСУ. Затем при помощи

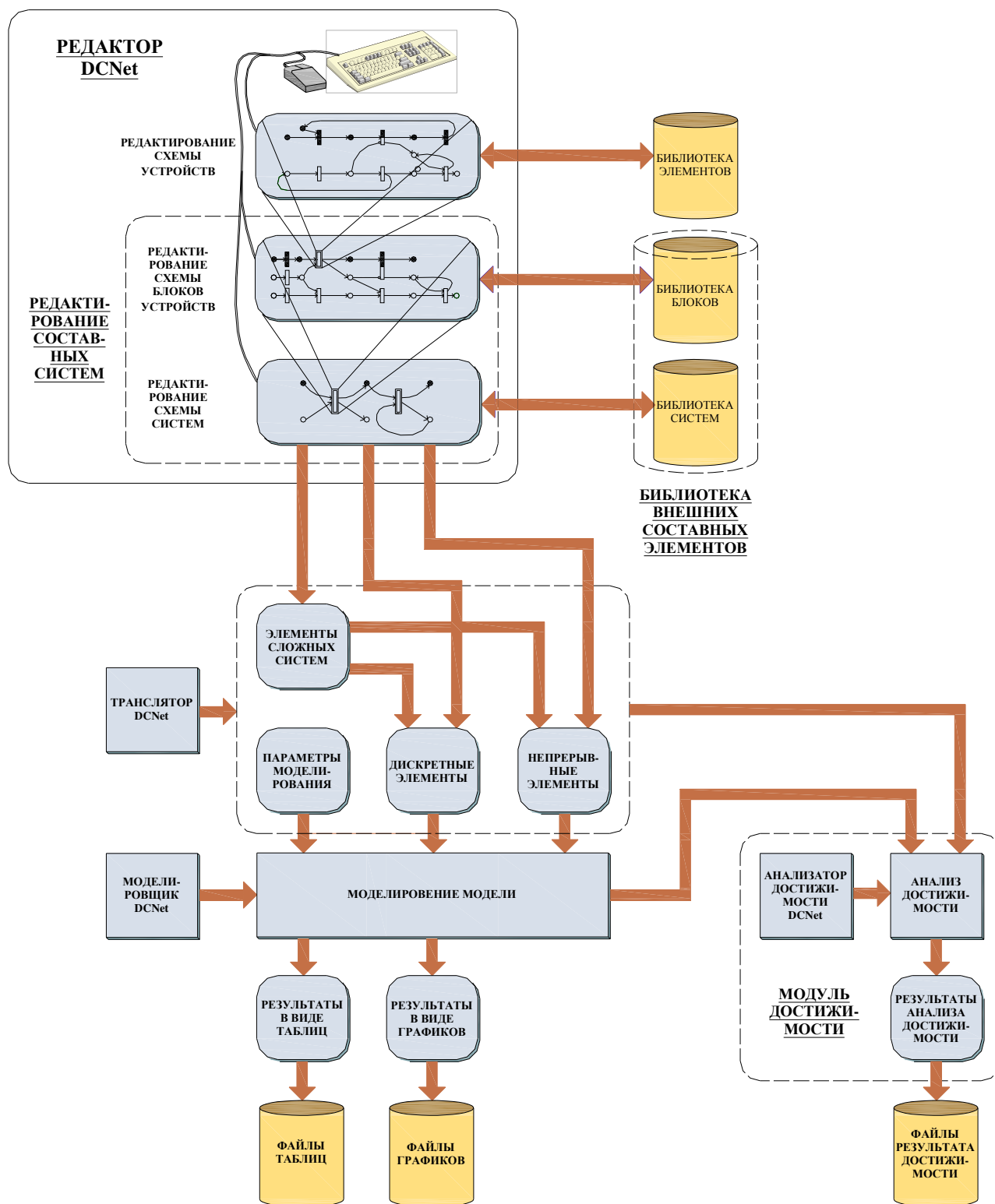


Рисунок 4.16 – Структурная схема редактирования и моделирования АСУ

транслятора получаем рабочую модель исследуемой системы. Необходимо отметить, что на данном этапе производится преобразование иерархической модели в простую, представленную одним уровнем иерархии. Это позволяет упростить

модуль моделирования. Производим ее моделирование полученной таким образом модели и сохраняем результаты в виде файлов графиков и таблиц. Как видно из рис. 4.16, для исследования достижимости необходимо произвести компиляцию модели. Это связано с тем, что подпрограмме исследования достижимости необходима информация о типе элемента и связях между ними [82]. Моделирование для получения результата анализа свойства достижимости необходимо только для численного метода.

4.4. Моделирование и анализ свойства достижимости АСУ

В качестве примера исследования свойства достижимости АСУ выбрана пакетная установка для выпаривания раствора хлористого натрия [24, 110].

Рассмотрим процесс моделирования на примере модели системы выпарной установки, приведенной в приложении А.

При помощи редактора вводим схему системы в DCNET. При этом получаем схему, приведенную на рис. 4.17. Как видно из рисунка 4.17, система представлена при помощи пиктограмм, используемых в системе DCNET и описанных в приложении Б.

Математическая модель объекта управления, описываемая представленными в приложении А формулами, является многомерной, нелинейной, имеет выходные параметры m_1 , m_2 , T и представлена с использованием средств ДН-сетей на рис. 4.3. Входными воздействиями для объекта являются:

- управление положением клапана V12 по каналу позиция p_4^2 переход t_4^3 ;
- управление положением клапана V15 по каналу позиция p_3^2 переход t_3^3 ;
- управление положением клапана V18 по каналу позиция p_2^2 переход t_2^3 ;
- управление положением ввода количества тепловой энергии Q_h по каналу позиция p_1^2 переход t_1^3 .

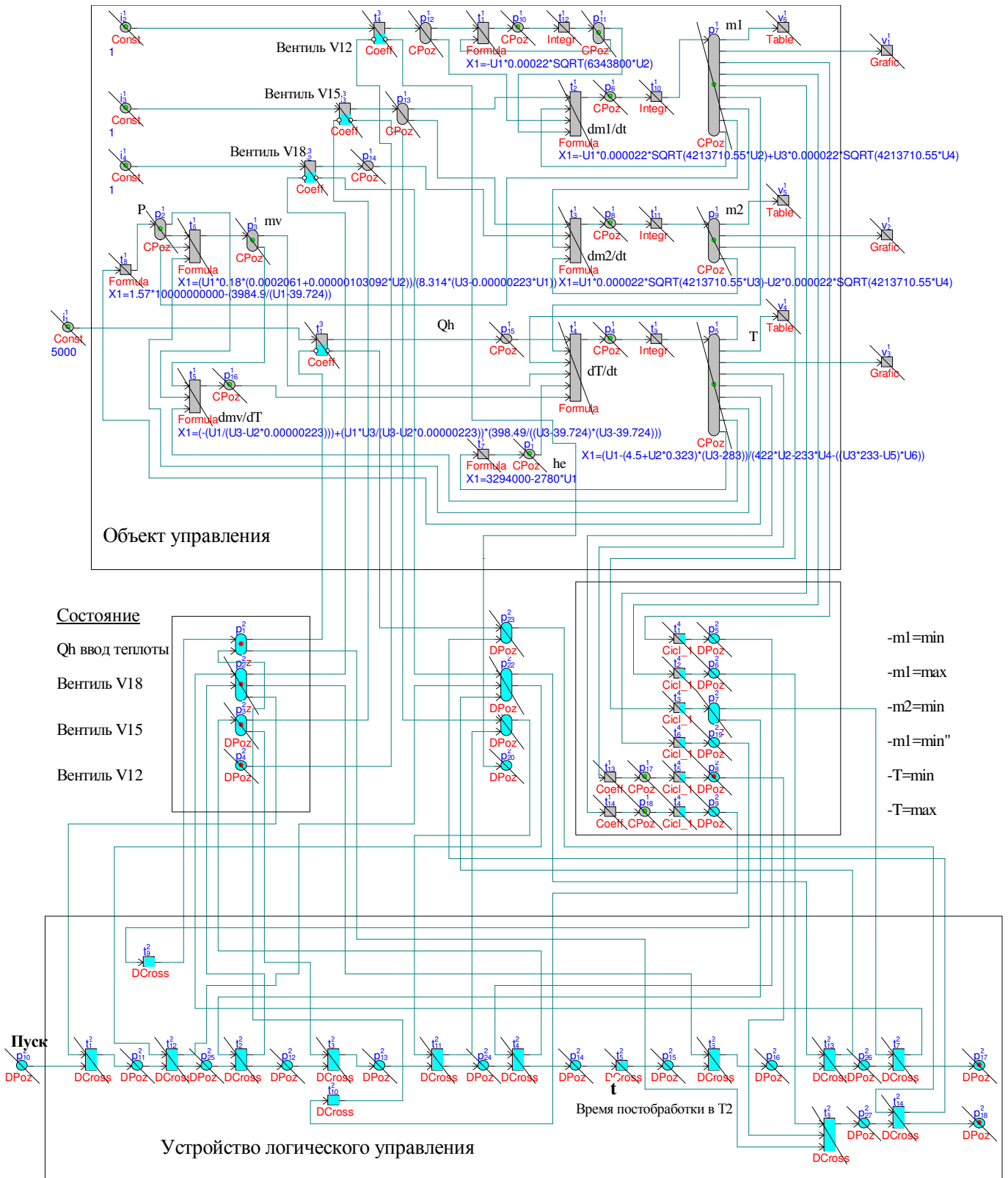


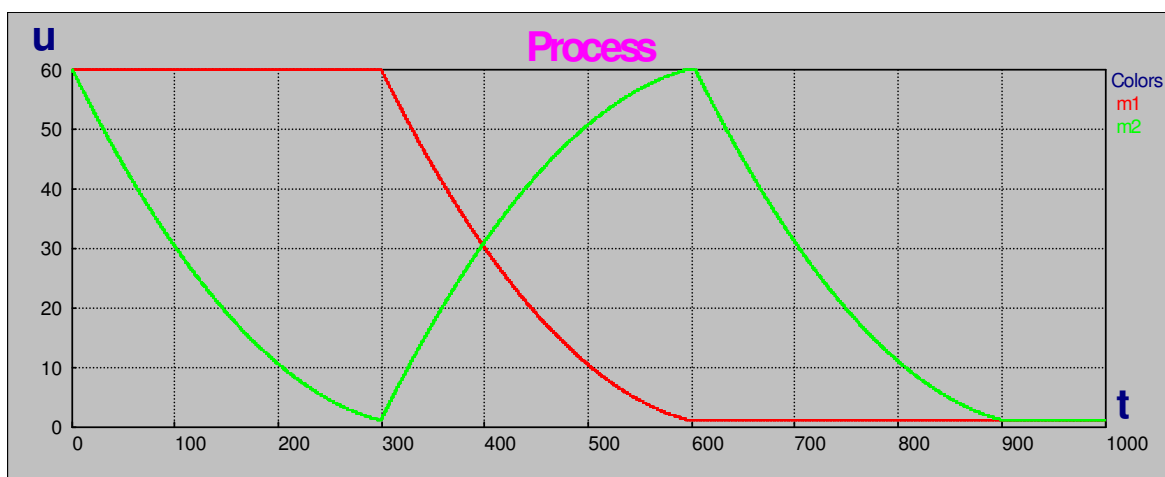
Рисунок 4.17 – Пример АСУ пакетной установки выпаривания раствора хлористого натрия

Переходы $t_1^3, t_2^3, t_3^3, t_4^3$ являются структурно-управляемыми и их срабатывание зависит от наличия маркера во входных дискретных позициях $p_1^2, p_2^2, p_3^2, p_4^2$ соответственно, соединенных с вышеназванными переходами ингибиторной дугой. При наличии маркера в указанных позициях, данные переходы закрываются, в противном случае они открыты. Маркировка выходных позиций $p_{23}^2, p_{22}^2, p_{21}^2, p_{20}^2$ структурно-управляемых переходов свидетельствует о правильности их срабатывания. Выходные непрерывные сигналы: масса жидкости в первом танке m_1 , масса жидкости во втором танке m_2 , температура жидкости в первом танке T формируются соответственно в непрерывных позициях p_7^1, p_9^1 и p_5^1 . Для формирования выходных дискретных сигналов используются непрерывно-дискретные переходы. Выходной дискретной позицией данного вида переходов появляется маркер в том случае, если текущее значение входного сигнала будет больше либо равно заданному значению. Выходные дискретные сигналы формируются в позициях $p_5^2, p_6^2, p_7^2, p_{19}^2, p_8^2, p_9^2$ и появление маркеров в них означает соответственно выполнение следующих условий: достижение массы в емкости Т1 минимального значения $m_1 = \min$, достижение массы в емкости Т1 максимального значения $m_1 = \max$, достижение массы в емкости Т2 минимального значения $m_2 = \min$, достижение массы в емкости Т1 минимального значения $m_1 = \min''$, достижение температуры в емкости Т1 минимального значения $T = \min$, достижение температуры в емкости Т1 максимального значения $T = \max$. Данные дискретные сигналы будут использованы для формирования алгоритма управления.

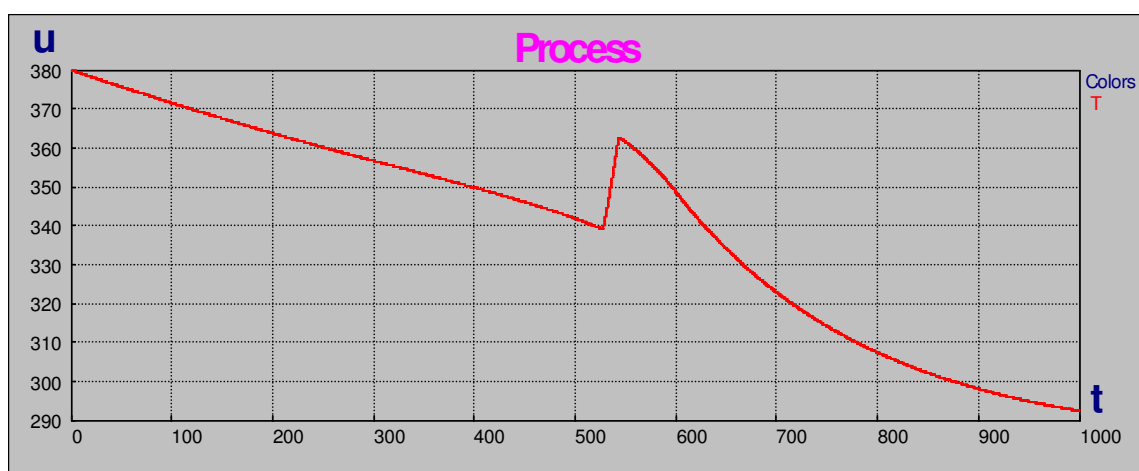
На рис. 4.19 отображена начальная разметка ДН-сети, описывающей установку. После задания необходимых параметров в результате проведения моделирования получаем графики зависимости уровня заполнения танков от времени, представленные на рис. 4.18. На данном рисунке отображаются графики. Проведем анализ свойства достижимости при помощи аналитического и численных методов для системы, представленной на рис. 4.17.

В модуле «Анализ достижимости» реализуется два режима.

1. Аналитический (декомпозиция-редукция ДН-сети).



а



б

Рисунок 4.18 – Результаты моделирования установки: а – зависимость массы m_1 и m_2 от времени; б – зависимость температуры T от времени

2. Маркерный (моделирование).

Для каждого режима можно реализовать «Полный» или «Шаговый» тип запуска.

Для исследования пакетной установки выпаривания раствора хлористого натрия выбираем режим – аналитический, тип запуска – шаговый.

Выполнение программы обеспечивается в интерактивном режиме нажатием кнопки «Пуск». В этом случае на экране дисплея формируется начальное состояние (см. приложение А), последующие нажатия кнопки «Пуск» обеспечивают формирование программой следующих итераций.

Визуализированное представление результата исследования представлено на рис. 4.19. Перечеркивание наклонной линией изображений всех типов переходов $t^{(*)}$ и позиций $p^{(*)}$ является признаком того, что исследуемая АСУ является достижимой.

При выборе режима «Аналитический» тип запуска «Полный» нажатие кнопки «Пуск» реализует все итерации.

При реализации режима «Маркерный» запускается анализ достижимости численным методом. При этом производится моделирование системы с накоплением разметки. Анализ достижимости при этом производится исходя из конечного состояния системы. Данное состояние показано в приложении А.

В результате проведения анализа достижимости при помощи аналитического и численного методов получены результаты времени, затраченного на проведение одного прогона исследования на различных моделях. Для каждой модели исследование выполнялось по 10 раз с одинаковыми начальными данными, после чего результаты усреднялись. Усредненные результаты временных затрат представлены в табл. 4.3.

Таблица 4.3 – Затраты времени на проведение анализа свойства достижимости

Название модели	Число элементов модели	Время для численного метода	Время для аналитического метода
Двигатель постоянного тока [26, 35, 56, 102]	43	4.3 с	170 мс
Реактор производства поливинилхлорида [38, 56, 76]	187	2.3 с	250 мс
Модель сообщающихся сосудов [106]	33	300 мс	1 мс
Установка выпаривания раствора хлористого натрия	93	700 мс	50 мс

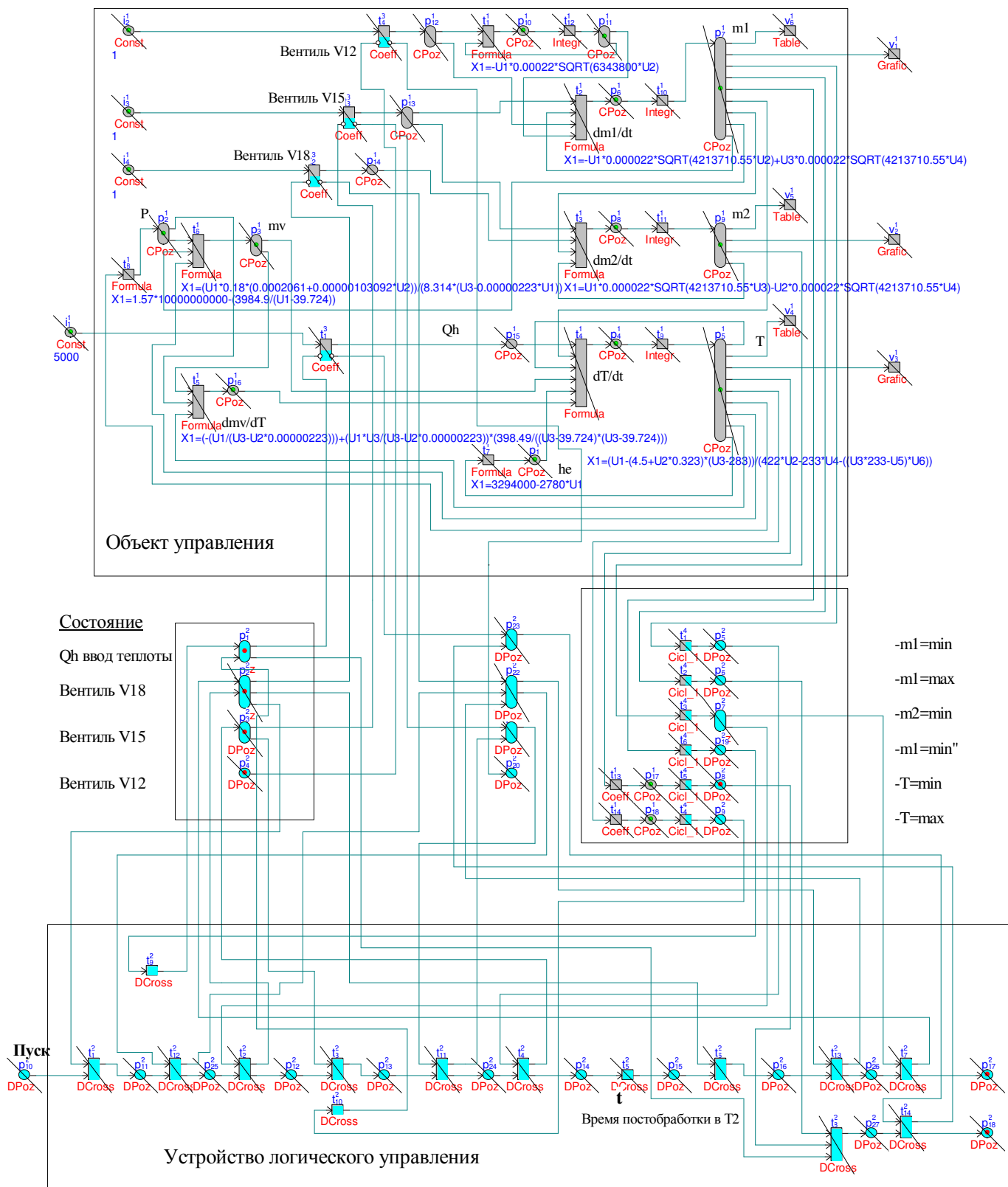


Рисунок 4.19 – Визуализация результата исследования достижимости АСУ пакетной установки выпаривания хлористого натрия

Выводы

1. Проведена разработка базовых ИТ моделирования и анализа свойства достижимости ДН АСУ, представленных ДН-сетью.
2. Разработан модуль проверки свойства достижимости исследуемой системы при помощи аналитического и численного подходов использует разработанные базовые ИТ.
3. ИТ моделирования и анализа свойства достижимости ДН АСУ базируется на разработанной дискретно-непрерывной модели и внедрении специального этапа исследования достижимости заданных состояний, что позволяет повысить качество моделирования и сократить время анализа.
4. Разработана система моделирования и исследования некоторых свойств АСУ, представленных ДН-сетью, и использующая для моделирования НС части системы модифицированный метод Эйлера или Рунге-Кутты четвертого порядка.
5. Проведена проверка разработанной программной среды путем моделирования процесса работы пакетной установки выпаривания раствора хлористого натрия.
6. Проведенный сравнительный анализ результатов моделирования с системой MATLAB показал нормированную среднеквадратичную ошибку менее 1%.
7. Анализ потраченного времени исследования свойства достижимости при исследовании пакетной установки выпаривания раствора хлористого натрия показал его уменьшение при применении аналитического метода в 14 раз по сравнению с численным методом.

ЗАКЛЮЧЕНИЕ

В диссертационной работе решена важная научная задача, которая заключается в разработке и исследовании моделей и методов для информационной технологии моделирования и оценки достижимости дискретно-непрерывных автоматизированных систем, модели которых синтезированы на базе сетей Петри и дискретно-непрерывных сетей.

Основные результаты и выводы диссертационной работы состоят в следующем.

1. Современные автоматизированные системы управления часто являются дискретно-непрерывными, что позволяет сочетать в себе преимущества как аналоговых, так и цифровых систем, нивелируя их недостатки. Эти системы применяются в различных отраслях народного хозяйства, где требуется высокая оперативность и эффективность, таких как системы реального времени.
2. Показано, что информационные технологии моделирования и анализа дискретно-непрерывных систем, имеющиеся на сегодняшний день, не обеспечивают необходимое для потребностей практики качество моделирования. Кроме того, одна из важных задач анализа требует много времени при оптимальном синтезе системы. В связи с этим было выбрано направление, которое позволяет уменьшить эти недостатки посредством использования методов редукции-декомпозиции. Разработка моделей и информационных технологий моделирование проводится на базе ДН-сетей.
3. Разработана и исследована дискретно-непрерывная модель путем расширения форм представления дискретно-непрерывной сети при помощи введения дополнительных дискретно-непрерывного и дискретного нечеткого переходов, что позволяет улучшить качество моделирования по показателю нормированной среднеквадратической ошибки в 5 раз по сравнению с

аналоговой и в 7 раз по сравнению с цифровой. Данная модель позволила провести разработку метода преобразования сигнального графа объекта моделирования в модель, представленную сетью Петри.

4. Разработанный модифицированный сигнальный граф и методы представления его в виде сетей Петри, позволил разработать метод анализа свойства достижимости и расширил область его применения по сравнению с использованием только редукции-декомпозиции.
5. Разработаны аналитический метод анализа достижимости дискретно-непрерывной сети на основе редукции-декомпозиции дискретно-событийной части и декомпозиции непрерывно-событийной части и отдельно дискретно-событийной части, а также численный метод анализа достижимости, который предусматривает реализацию моделирования i -го режима системы до совпадения текущего состояния с заданным состоянием. Разработка аналитического метода позволила обеспечить проведение анализа свойства достижимости на ранних этапах разработки и уменьшить время анализа в среднем в 14 раз по сравнению с численным методом.
6. Разработанные базовые информационные технологии, направленные на моделирование и анализ сложных технических объектов с многорежимным характером функционирования, встроены в информационную технологию DCNET для моделирования дискретно-непрерывных систем и анализа свойства достижимости. Информационная технология DCNET базируется на разработанной дискретно-непрерывной модели и внедрении специального этапа исследования достижимости заданных состояний, что позволяет повысить качество моделирования и сократить время анализа. Проведенный сравнительный анализ результатов моделирования с системой MATLAB показал нормированную среднеквадратичную ошибку менее 1%. Анализ потраченного времени исследования свойства достижимости показал его уменьшение при применении аналитического метода в 14 раз по сравнению с численным методом.

7. Информационная технология DCNET использована при разработке дискретно-непрерывной автоматизированной системы управления установки для выпаривания раствора хлористого натрия. Для разработанной системы было проведено моделирование системы и анализ свойства достижимости с помощью аналитического и численного методов. Результаты диссертационной работы апробированы и внедрены в учебный процесс кафедры «Компьютерные системы» Одесского национального политехнического университета, а также НПП «ХАРТРОН-ЕНКОС» и НВКЦ «ОДИС-W», где использованы при разработке и оптимизации системы контроля городского освещения СПРУТ. Применение разработанных информационных технологий позволило сократить время разработки на 30% и улучшить качество оборудования на 6 – 7%.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Айзерман, М. А. Логика. Автоматы. Алгоритмы. / М.А. Айзерман, А.А. Гусев. – Физматгиз, 1963.
2. Айзерман, М. А. Основы теории разрывных систем. / М.А. Айзерман, Е.С. Пятницкий // Автоматика и телемеханика. – 1974. – № 7, № 8.
3. Анисимов, Н. А. Алгебра структур протоколов на основе сетей Петри / Н.А. Анисимов // Автоматика и вычислительная техника. – 1987. – № 1. С. 9 – 13.
4. Анисимов, Н. А. Редуцируемость сетей Петри / Н.А. Анисимов // Программирование. – 1984. – №4. С. 36 – 42.
5. Арбиб, М. О единой концепции в теории конечных автоматов и теории управления / М. Арбиб // Экспресс-информ. Сер. «Техн. кибернетика» – 1965. – № 12. – С. 1 – 18.
6. Бадман, М. К. Территориально-производственные комплексы: прогнозирование процесса функционирования с использованием сетей Петри. / М.К. Бадман, О.Л. Бадман, Т.Н. Есикова – Новосибирск: Наука, 1990. – 303 с.
7. Баранов, А. А. Структурно-топологический подход к исследованию управляемости и наблюдаемости систем управления / А.А. Баранов // Управляющие системы и машины, 2007, №4. С. 10 – 13.
8. Блауберг, Н. В. Становление и сущность системного подхода. / Н.В. Блауберг, Э.Г. Юдин – М.: Наука, 1973.
9. Бусленко, В. Н. Автоматизация имитационного моделирования сложных систем. /В.Н. Бусленко – М.: Наука, 1977. – 240 с.
10. Бусленко, Н. П. Моделирование сложных систем. / Н.П. Бусленко – М.: Наука, 1978. – 399 с.
11. Вавилов, А. А. Структурный и параметрический синтез сложных систем. / А.А. Вавилов – Л.: ЛЭТИ, 1979. – 96 с.

12. Веклин, Н. П. Математическое моделирование химикотехнологических комплексов периодического типа с использованием сетей Петри / Н.П. Веклин // Механизация и автоматизация управления. – Киев, 1990. – №1. С. 13 – 19.
13. Воронов, А. А. Введение в динамику сложных управляемых систем / А.А Воронов – Наука. Гл. ред. физ.-мат. лит. 1985. – 370 с.
14. Головицына, М.В. Проектирование автоматизированных технологических комплексов / М.В. Головиціна – М.: Изд-во МГОУ, 2001. – 321 с.
15. Груднев, А. И. Моделирование динамики обрабатывающего участка ГАП / А.И. Груднев, А.А Меликян // Техническая кибернетика, 1987, №3. с. 31 – 37.
16. Денисенко, А. В. К вопросу о включении САПР сложных технических объектов в систему дистанционного обучения / В.А. Денисенко, А.Г. Нестерюк // Матеріали XV міжнар. наук.-метод. конф. «Управління якістю підготовки фахівців», м. Одеса, Україна, 22 – 23 квітня 2010 р. Ч. I. – Одеса: ОДАБА, 2010. – С. 41 – 43.
17. Денисенко, А. В. САПР гибридных систем /А.В. Денисенко, А.Г. Нестерюк // Тез. докл. СИЭТ-2002 Одесса, 2002. – С. 92.
18. Денисенко, В. Программа моделювання дискретно-непрервних мереж в середовищі MS Windows. / В. Денисенко, Т. Денисенко, А. Нестерюк, В. Ляхин // Тезис. докл. 1-й международной школы по моделированию (AUTUMN'96), г. Алушта, Украина, 12 – 14 сентября 1996 г. – Алушта, 1996. – С. 39.
19. Денисенко, В. А. Аналитическое представление дискретно-непрерывных сетей в системах автоматического проектирования / В.А. Денисенко, А.Г. Нестерюк // Холодильная техника и технология. – 1998. – № 1. Вып. 58. – С. 88 – 92.
20. Денисенко, В. А. Визуализированная программная среда моделирования дискретно-непрерывных сетей / В.А. Денисенко, А.Г. Нестерюк // Електромашинобудування та електробладнання. – 2004. – № 63. – С. 116 – 121.
21. Денисенко, В. А. Визуальная среда моделирования производственных процессов / В.А. Денисенко, А.Г. Нестерюк // Современные информационные

- и электронные технологии (СИЭТ-2003): труды 4-й междунар. научно-практ. конф., г. Одесса, Украина, 19 – 23 мая 2003 г. – Одесса: ОНПУ, 2003. – С. 140.
22. Денисенко, В. А. Виртуальный лабораторный стенд для исследования “реактивных” систем / В.А. Денисенко, А.Г. Нестерюк // Вторая международная научно-методическая конференция «Сучасні технології вищої освіти». – Одесса, 2003. – С. 43.
 23. Денисенко, В. А. Гибридное моделирование в системах анимации / В.А. Денисенко, А.Г. Нестерюк, А.В. Денисенко, И.П. Сорока // XV Міжнародна конференція з автоматичного управління (Автоматика-2008): Тези доповідей 15-ї міжнар. наук.-техн. конф., м. Одеса, Україна, 23 – 26 вересня, 2008 р. – Одесса, 2008. – С. 156 – 158.
 24. Денисенко, В. А. Дискретно-непрерывная сеть как средство моделирования сложных технических систем / В.А. Денисенко, А.Г. Нестерюк, А.А. Гурский // Холодильная техника и технология. – 2004. – № 4. – С. 54–58.
 25. Денисенко, В. А. Дискретно-непрерывные и нейронные сети в системах автоматического управления / В.А. Денисенко, А.Г. Нестерюк, В.В. Ляхин // Холодильная техника и технология. – 1999. – №1. Вып. 59. – С. 96-100.
 26. Денисенко, В. А. Достижимость гибридной системы / В.А. Денисенко, А.Г. Нестерюк // Холодильная техника и технология. – 2005. – № 3. – С. 67 – 74.
 27. Денисенко, В. А. Достижимость одного класса систем с управляемой структурой / В.А. Денисенко, Х.-М. Ханиш, В.В. Вяткин, А.Г. Нестерюк // IX Міжнародна конференція з автоматичного управління (Автоматика-2001): Тези доповідей 9-ї міжнар. наук.-техн. конф., м. Одеса, Україна, 10 – 14 вересня, 2001 р. – Одесса, 2001. – С. 26 – 27.
 28. Денисенко, В. А. Задача достижимости гибридной системы / В.А. Денисенко, А.Г. Нестерюк // Тр. Одес. политехн. ун-та. – Одесса: ОНПУ, 2005. – Вып. 1 (23). – С. 67 – 74.
 29. Денисенко, В. А. Использование программной среды DCNET для исследования систем с управляемой структурой / В.А. Денисенко,

- А.Г. Нестерюк, В.В. Ляхин // В кн. Моделирование в прикладных научных исследованиях. Материалы IV семинара. – Одесса: ОГПУ. – 1997. – С. 9 – 11.
30. Денисенко, В. А. Исследование достижимости непрерывной части гибридной системы / В.А. Денисенко, А.Г. Нестерюк // Тр. Одес. политехн. ун-та. – Одесса: ОНПУ, 2003. – Вып. 1 (19). – С. 112 – 115.
31. Денисенко, В. А. Исследование задачи достижимости сложной системы с использованием графического программирования. / В.А. Денисенко, А.Г. Нестерюк // Тр. Одес. политехн. ун-та. – Одесса: ОНПУ, 1999. – Вып. 2 (8). – С. 10 – 13.
32. Денисенко, В. А. Компьютерное моделирование гибридных систем / В.А. Денисенко, А.Г. Нестерюк // Друга міжнар. наук.-метод. конф. «Сучасні технології вищої освіти». – Одеса, 2003. – С. 37.
33. Денисенко, В. А. Математическое и программное обеспечения определения достижимости гибридных систем / В.А. Денисенко, А.Г. Нестерюк // XIII Міжнародна конференція з автоматичного управління (Автоматика-2006): Тези доповідей 13-ї міжнар. наук.-техн. конф., м. Вінниця, Україна, 25 – 28 вересня, 2006 р. – Вінниця: УНІВЕРСУМ-Вінниця, 2006. – С. 19.
34. Денисенко, В. А. Модель системной динамики системы с управляемой структурой / В.А. Денисенко // Тез. докл. 50-й юбилейной научной конференции. – Одесса.: ОТИППЛ, Ротапринт, 1990. – 152 с.
35. Денисенко, В. А. Модуль проверки достижимости дискретно-непрерывных сетей / В.А. Денисенко, А.Г. Нестерюк // Наукові праці ОНАХТ. – Одеса, 2004. – Вып. 27. – С. 130 – 137.
36. Денисенко, В. А. Нечеткие дискретно-непрерывные сети и среда их моделирования / В.А. Денисенко, А.В. Денисенко, А.Г. Нестерюк // XV Міжнародна конференція з автоматичного управління (Автоматика-2008): Тези доповідей 15-ї міжнар. наук.-техн. конф., м. Одеса, Україна, 23 – 26 вересня, 2008 р. – Одесса, 2008. – С. 154 – 155.
37. Денисенко, В. А. Определение условий достижимости в гибридных системах / В.А. Денисенко, А.Г. Нестерюк // Кораблебудування: освіта, наука,

виробництво: Матеріали міжнар. конф., Миколаїв, Україна, 24 – 25 вересня 2002 р. Т.ІІ. – Миколаїв, 2002. – С. 232 – 233.

38. Денисенко, В. А. Программная среда исследования систем с управляемой структурой / В.А. Денисенко, А.Г. Нестерюк // IX Міжнародна конференція з автоматичного управління (Автоматика-2001): Тези доповідей 9-ї міжнар. наук.-техн. конф., м. Одеса, Україна, 10 – 14 вересня, 2001 р. – Одеса, 2001. – С. 84.
39. Денисенко, В. А. Программная среда моделирования систем с управляемой структурой / В.А. Денисенко, А.Г. Нестерюк, В.В. Ляхин // Тр. Одес. политехн. ун-та. – Одесса, 1998. Вып. 2 (6). – С. 17 – 20.
40. Денисенко, В. А. Разработка программных средств моделирования систем с управляемой структурой / Т.А. Денисенко, А.Г. Нестерюк, В.В. Ляхин и др. // Одес. гос. политехн. ун-т. Одесса. 1996. – 9 с. – Ден. в ГАСНТИ Украины. 18.06.97, №366 – Ук 97.
41. Денисенко, В. А. САПР гибридных систем. / В.А. Денисенко, А.Г. Нестерюк // Международная конференція «Радиотехнические устройства и материалы» – Одесса, 2002. – С. 43.
42. Денисенко, В. А. Сравнение графических программных средств моделирования дискретно-непрерывных систем. / В.А. Денисенко, А.Г. Нестерюк // Холодильная техника и технология. – 1998. – №1. Вып. 58. – С. 96 – 100.
43. Денисенко, В. А. Сравнение программных средств моделирования систем с управляемой структурой. / В.А. Денисенко, А.Г. Нестерюк, В.В. Ляхин // В кн. Моделирование в прикладных научных исследованиях. Материалы IV семинара. – Одесса: ОГПУ. – 1997. – С. 11 – 12.
44. Денисенко, В. А. Средства моделирования сложных технологических процессов / В.А. Денисенко, А.Г. Нестерюк, А.А. Гурский // Тез. докл. 61-й научно-техни. конф. профессорско-преподавательского состава. – Одесса, 2004.

45. Денисенко, В. А. Формализм свертки сложных систем / В.А. Денисенко, А.В. Кобушко, А.Г. Нестерюк, Т.А. Денисенко // Тез. докл. семинара «Информационные системы и технологии». – Одесса, 2003. – С. 33.
46. Додонов, А. Г. Компьютерная информация: Хранение и управление / А.Г. Додонов, П.И. Иваненко, В.Г. Клименко – К.: Наук. думка, 1992. – с. 232.
47. Дьяконов, В. П. Simulink 4. Специальный справочник. / В.П. Дьяконов – СПб.: Питер, 2002.
48. Емельянов, С. В. Системы автоматического регулирования с управляемой структурой объекта / С.В. Емельянов, Н.В. Буровой, Н.В. Карпущина // Докл. АН СССР 1979. – Т224, №5. – с. 134 – 139.
49. Емельянов, С. В. Системы автоматического управления с переменной структурой. / С.В. Емельянов – Наука, 1967.
50. Жук, К. Д. Автоматизированное проектирование логико-динамических систем. / К.Д. Жук, А.А. Тимченко – К.: Наук. думка, 1981. – 320 с.
51. Заде, Л. Теория линейных систем. Метод пространства и состояний / Л. Заде, И. Дезоер. – Наука, 1970.
52. Зайцев, Д. А. Визуализация производственных процессов в инструментальной системе диспетчера машиностроительного предприятия / Д.А. Зайцев, А.И. Слепцов // Автометрия. – 1990. – № 4. – С. 90 – 93.
53. Зайцев, Д. А. Декомпозиция сетей Петри / Д.А. Зайцев // Кибернетика и системный анализ. – 2004. – № 5. – С. 131 – 140.
54. Зайцев, Д. А. Композиционный анализ сетей Петри / Д.А. Зайцев // Кибернетика и системный анализ. – 2006, № 1. – С. 143 – 154.
55. Зайцев, Д. А. Эквивалентные преобразования временных сетей Петри / Д.А. Зайцев, А.И. Слепцов // Кибернетика и системный анализ. – 1997. – № 5. – С. 59 – 76.
56. Згуровский, М. З. Дискретно-непрерывные системы с управляемой структурой. Теория моделирования, применение. / М.З. Згуровский, В.А. Денисенко – К.: Наукова думка, 1998. – 350 с.

57. Згуровский, М. З. Задачи управления гибкими химико-технологическими системами дискретно-непрерывного типа / М.З. Згуровский, Ю.Н. Власенко, З. Банашек // Химическая технология. – 1988. – № 4. – С. 56-64.
58. Згуровский, М. З. Интегрированные системы оптимального управления и проектирования: Учеб. Пособие. / М.З. Згуровский – К.:Выща шк., 1990. – 351 с.
59. Калман, Р. Очерки по математической теории систем. / Р. Калман, П. Фалб, М. Арбиб – М.:Мир, 1971. – 400 с.
60. Капустин, Н. М. Математическое моделирование технологических объектов и процессов. / Н.М. Капустин – М.: Изд-во МГОУ, 2003. – 247 с.
61. Касти, Дж. Большие системы: Связность, сложность и катастрофы. / Дж. Касти – М.: Мир, 1982. – С. 216.
62. Клиначев, Н. В. Программа VisSim и ее расширения. [Электронный ресурс] / Н.В. Клиначев – Режим доступа: <http://model.exponenta.ru>
63. Козлов, О. С. Исследование и проектирование автоматических систем с помощью программного комплекса «МВТУ» / О.С. Козлов, Л.М. Скворцов // Информационные технологии. – 2006. №8.
64. Козлов, О. С. Программный комплекс для исследования динамики и проектирования технических систем / О.С. Козлов и др.// Информационные технологии. – 2005. №9.
65. Кондаков, В. И. Логический словарь-справочник. / В.И. Кондаков – М.: Наука, 1975. – 719 с.
66. Котов, В. Е. Алгебра регулярных сетей / В.Е. Котов // Кибернетика. – 1980. - №5. С. 10 – 18.
67. Котов, В. Е. Сети Петри. / В.Е. Котов – М.: Наука, 1984. – 160 с.
68. Куо, Б. Теория и проектирование цифровых систем управления. / Б. Куо – 445 с.
69. Легасов, В. А. Гибкие системы в химической технологии / В.А. Легасов, М.С. Сафронов, В.К. Бельнов, Н.С. Рабовская // Журн. всесоюз. хим. о-ва им. Д. И. Менделеева. – 1987. – Т.32, №3. – С. 252 – 258.

70. Ли, К. Основы САПР (CAD/CAM/CAE) / К. Ли – Спб.: Питер, 2004. – 560 с.
71. Месарович, М. Теория иерархических многоуровневых систем /М. Месарович, Д. Мако, Я. Такахара; пер. с англ. – Мир, 1973.
72. Морозов, А. А. и др. Проектирование интегрированных АСУ цехами с периодической технологией / А.А. Морозов и др // Управляющие системы и машины. – 1983. – №6.
73. Мурата, Т. Сети Петри: Свойства, анализ, приложения / Т. Мурата // ТИНЭР, 1989. – т.77, №4. – С. 127 – 135.
74. Негойце, Э. Применение теории систем к проблемам управления. / Э. Негойце – М.: Мир, 1981. – 180 с.
75. Нестеренко, Б. Б. Алгебра процессов для моделирования параллельных асинхронных вычислительных структур / Б.Б. Нестеренко, М.А. Новаторский // Электронное моделирование. – 2006. – № 4. С. 47 – 64.
76. Нестерюк, А. Г. Анализ моделирования химического реактора как логико-динамической системы. / А.Г. Нестерюк, А.А. Гурский, А.В. Денисенко // Электронные и компьютерные системы. – 2014. – № 14(90). – С. 124 – 130.
77. Нестерюк, А. Г. Использование сервера баз данных Postgres для организации взаимодействия элементов САПР DCNET / А.Г. Нестерюк // Современные информационные и электронные технологии (СИЭТ-2012): труды 13-й междунар. научно-практ. конф., г. Одесса, Украина, 4 – 8 июня 2012 г. – Одесса: ОНПУ, 2012. – С. 67.
78. Нестерюк, А. Г. Использование серверов баз данных при разработке САПР сложных систем / А.Г. Нестерюк // Современные информационные и электронные технологии (СИЭТ-2011): труды 12-й междунар. научно-практ. конф., г. Одесса, Украина, 23 – 27 мая 2011 г. – Одесса: ОНПУ, 2011. – С. 66.
79. Нестерюк, А. Г. Модуль моделирования САПР технических систем / А.Г. Нестерюк // Современные информационные и электронные технологии (СИЭТ-2010): труды 11-й междунар. научно-практ. конф., г. Одесса, Украина, 24 – 28 мая 2010 г. Т. I. – Одесса: ОНПУ, 2010. – С. 176.

80. Нестерюк, А. Г. Подходы к сокращению объема вычислений при решении задачи достижимости систем с управляемой структурой / А.Г. Нестерюк // Современные информационные и электронные технологии (СИЭТ-2009): труды 10-й междунар. научно-практ. конф., г. Одесса, Украина, 18 – 22 мая 2009 г. Т. I. – Одесса: ОНПУ, 2009. – С. 70.
81. Нестерюк, А. Г. Построение САПР с управляемой структурой / А.Г. Нестерюк // Современные информационные и электронные технологии (СИЭТ-2008): труды 9-й междунар. научно-практ. конф., г. Одесса, Украина, 19 – 23 мая 2008 г. Т. I. – Одесса: ОНПУ, 2008. – С. 170.
82. Нестерюк, А. Г. Практическая реализация модуля исследования достижимости дискретно-непрерывных сетей / А.Г. Нестерюк // Современные информационные и электронные технологии (СИЭТ-2007): труды 8-й междунар. научно-практ. конф., г. Одесса, Украина, 21 – 25 мая 2007 г. – Одесса: ОНПУ, 2007. – С. 87.
83. Нестерюк, А. Г. Разработка программной среды анализа достижимости сложных систем / А.Г. Нестерюк // Тези доп. IV міжнар. наук.-практ. конф. студентів, аспірантів та молодих вчених «Системний аналіз та інформаційні технології», м. Київ, Україна, 1 – 3 липня 2004 г. – Київ, 2004. – С. 147 – 148.
84. Норенков, И. П. Автоматизированное проектирование / И.П. Норенков – М.: Изд-во МГТУ им. Н.Э. Баумана, 2000. – 376 с.
85. Норенков, И. П. Информационные технологии в образовании / И.П. Норенков, А.М. Зимин – М.: Изд-во МГТУ им. Н.Э. Баумана, 2004.
86. Норенков, И. П. Основы автоматизированного проектирования: уч. для ВУЗов / И.П. Норенков – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 423 с.
87. Питерсон, Дж. Теория сетей Петри и моделирование систем. / Дж. Питерсон – М.: Мир. 1984. – 264 с.
88. Портер, У. Современные основания общей теории систем /У. Портер: пер. с англ. – М.: Наука. 1971. – С. 556.
89. Постон, Т. Теория катастроф / Т. Постон, И. Стюарт: пер. с англ.-М.: Мир.1983. – С. 607.

90. Свідоцтво про реєстрацію авторського права на твір. Комп'ютерна програма «Дискретно-непрерывные сети (DCNET)» / В.А. Денисенко, А.Г. Нестерюк. – №33728; дата реєстрації 15.06.2010.
91. Справочник по теории автоматического управления / под ред. А.А. Красовского. – М.: Наука. Гл. ред. физ.-мат. лит. 1987. – 712 с.
92. Терно, О. Р. Гибридные функции – новый метод описания сложных систем / О.Р. Терно // Техническая кибернетика, №6, 1965
93. Трудоношин, В. А. Моделирование электромеханических систем с помощью программно-методического комплекса «ПА9» / В.А. Трудоношин, И.В. Трудоношин // Информационные технологии. – 2006. №4.
94. Эрлих, Г. Системы с управляемой структурой. Отчет. / Г. Эрлих, А.А. Ильченко – Лейпциг: ВТУ, 1996. – 30 С.
95. Юдницкий, С. А. Логическое управление дискретными процессами: модели, анализ, синтез. / С.А. Юдницкий, В.З. Магергут – М.: машиностроение, 1987. – 176 с.
96. Alla, H. Hybrid Petri Nets / H. Alla, R. David and J. Bail / ECC European Control Conference – Grenoble – 1991.
97. Alur, R. Automatic symbolic verification of embedded systems. / R. Alur, T.A. Henzinger and Ho P.-H: In: Proceedings of the 14th Annual Real-Time System Symposium. S. 2 – 11, EWWW Computer Society Press, 1993.
98. Alur, R. Hybrid Automata: An Algorithm Approach to the Specification and Verification of Hybrid Systems / R. Alur, C. Coucoubetis, T.A. Henzinger and Ho P.-H – LNCS 736, Springer Verlag, Berlin, 1995.
99. Branicky, M. S. A unified framework for hybrid control: model and optimal control theory / M,S. Branicky, V.S. Borkar, S.K. Mitter // IEEE Transactions on Automatic Control, 43(1):31 – 45 – January 1998.
100. Chouikha, M. Modelling and Analysis of Continuous-discrete Systems with Hybrid Petri Nets / M. Chouikha and E. Schnieder // Proceeding of the IFAC Congress '99 – Beijing, 1999.

101. Denisenko, V. Grundlagen der Theorie diskret-kontinuierlicher Systeme mit steuerbarer Struktur / V. Denisenko, H. Erlich // 45th International Scientific Colloquium, Ilmenau Technical University – 2000 – C. 460 – 465.
102. Denisenko, V. Vergleich grafischer Programmtools zur Simulation diskret-kontinuierlicher System / V. Denisenko, H. Erlich, A. Ilchenko, A. Nesteryuk// 45th International Scientific Colloquium, Ilmenau Technical University – 2000 – C. 472 – 479.
103. Drath, R. Modellierung diskret-kontinuierlicher System mit Pery-Netzen / R. Drath, S. Schwuchow // In: Schnieder E. (Hrsg.): Entwurf komplexer Automatisierungssysteme. 5. Fachtagung, S/ 265-283 – Braunschweig, 1997.
104. Drath, R. Modellierung hybrider Systeme auf der Basis modifizierter Petri-Netze: Dissertation an der TU-Ilmenau, Fachgebiet Automatisierungsanlagen und Prozeßletztechnik. ISBN-Nr.: 3-932633-40-7./ R. Drath – Ilmenau, 1999.
105. Drath, R. Prozeßmodellierung mittels attributierter Hybrider Dynamischer Netze / R. Drath // 45th International Scientific Colloquium, Ilmenau Technical University – 2000 – C. 453-459.
106. Drath, R. Visual Object Net++. [Электронный ресурс] / R. Drath – Режим доступа: <http://www.systemtechnik.tu-ilmenau.de/~drath/visual.htm>
107. Ehrlich, H. Dynamische Systeme mit steuerbarer Struktur in der Automatisierungstechnik / H. Ehrlich, V. Denisenko // Moderne informationstechnische Methoden für die Meß-, regelungs- und Automatisierungstechnik/ Jahrestagung 1995 — 1, DFMRS. Bremen C. 11 – 25.
108. Erlich, H. Systeme mit steuerbarer Struktur in der Automatisierungstechnik. / H. Erlich, C. Nitu, V. Denisenko // The third international conference on development and application systems. MAY 23 – 25, 1996. TV “STEFAN CEL MARE” SUCEAVA, ROMANIA, 1996, – c. 11 – 13.
109. Fdanke, D. 2D-Approach to Stability of Hybrid Systems. / D. Fdanke //In: Hybrid Systems. 3rd International Conference on Automation of Mixed Processes, S. 159 – 163, Reims, 1998.

110. Hanisch, H.-M. Timestamp Petri Nets in Technical Applications / H.-M Hanisch, K. Lautenbach, C. Simon, J. Thieme // International Workshop on Discrete Event Systems, WODES '98, 26 – 28 August 1998 – Cagliari – Proceedings pp. 321 – 326.
111. Ichikawa, A. Reachability and control of discrete event systems represented by conflict-free Petri nets. / A. Ichikawa, K. Yokoyama, S. Kurosi – Proc. of ISCAS 85. – P. 487 – 490.
112. Licht, Th. Modellierung von gemischt ereignis-/zeitdiskreten Systemen mittels Hybrider Dynamischer Netze / Th. Licht, R. Drath // 45th International Scientific Colloquium, Ilmenau Technical University – 2000 – C. 466 – 471.
113. Nenninger, G. Modeling and Analysis of Hybrid Systems: A New Approach Integrating Petri Nets and Differential Equations / G. Nenninger, V. Krebs // In: IEEE Workshop on Parallel and Distributed Real Time Systems, S. 234 – 238 – Genf, 1997.
114. Nenninger, G. M. Modellierung, Simulation und Analyse hybrider dynamischer Systeme mit Nets-Zustands-Modellen / G.M. Nenninger, M.K. Schnabel and V.G Krebs// In Automatisierungstechnik, Vol. 47, pp. 118 – 126 – March 1999.
115. Rovetto, C. Deadlock Control Software for Two Automated Guided Vehicles using Petri Nets / C. Rovetto, E. Cano and J.M. Colom // In Proceedings of the International Workshop on Petri Nets and Software Engineering (PNSE'10). Braga, Portugal, jun 2010. – pp. 125 – 140.
116. Thomas Schlegl. State Reconstruction and Error Compensation in Discrete-Continuous Control Systems / Thomas Schlegl, TU München, Mark K. Schnabel, Universität Karlsruhe, Martin Buss, TU München, Volker G. Krebs, Universität Karlsruhe // Automatisierungstechnik Theoretische Grundlagen, Methoden, Anwendungen 48, Jahrgang 2000, Heft 9, Seite 411 – 468.

ПРИЛОЖЕНИЕ А

ПРИМЕР ИССЛЕДОВАНИЯ ДНАСУ

В качестве примера исследования свойства достижимости гибридного объекта выбрана пакетная установка для выпаривания раствора хлористого натрия, предложенная как тестовая задача [24, 110].

Технологический процесс (рис. А.1) реализуется следующим образом. Заполнение резервуара Т1 обеспечивается открытием клапана V12. Информация о достижении заданного уровня регистрируется уровнемером LIS. Для обеспечения необходимого уровня концентрации раствора к резервуару Т1 подводится теплоноситель. Пар охлаждается в конденсаторе С1, к которому подается охлаждающая вода.

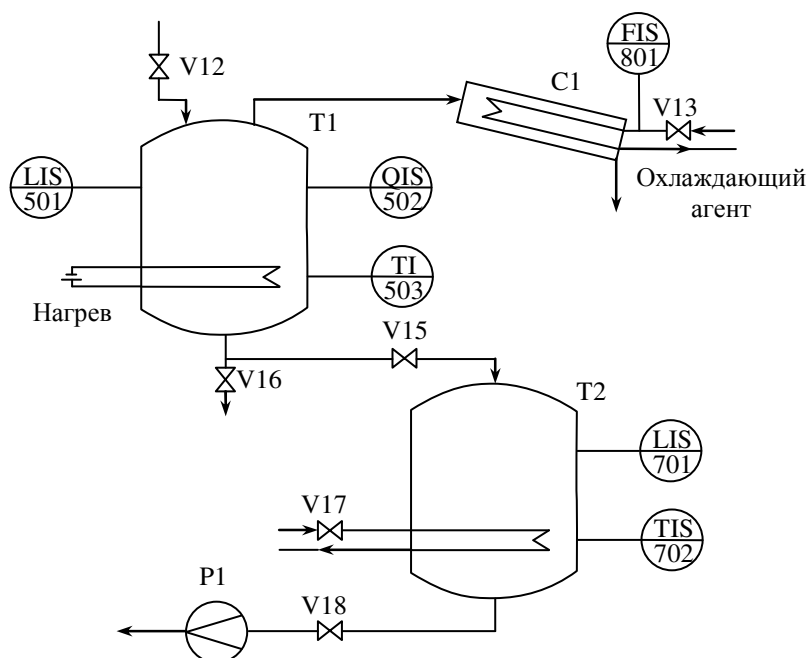


Рисунок А.1 – Функциональная схема установки

Когда желательная концентрация раствора в резервуаре T1 достигнута и резервуар T2 пуст, сконцентрированный раствор сливается в T2, где происходит этап постобработки. Готовое вещество выкачивается из T2.

Если в случае аварии системы охлаждения конденсаторов в T1 продолжается процесс испарения, то через некоторое время достигается опасно высокое давление в C1, что приводит к разрушению конденсатора. Следовательно, процесс испарения в случае такой аварии должен быть прерван.

Если вещество в резервуаре T1 не нагрето, и его нельзя сливать в емкость T2, потому что предыдущая партия все еще обрабатывается в T2, то через некоторое время вещество в T1 преобразуется в гель.

Это, в свою очередь, вызывает разрушение испарителя-конденсатора T1. Следовательно, опасная ситуация создается как при аварии (потеря охлаждающей воды), так и в случае занятости T2 обработкой предыдущей партии.

Проблема, рассматриваемая далее, состоит в том, чтобы расширить (или изменить) управляющую программу, для случая, гарантирующего безопасную остановку процесса в случае нарушения охлаждения в конденсаторе. Такой сбой ведет к опасно высокому давлению в T1, если процесс испарения продолжается слишком долго. С другой стороны, отключение нагревания немедленно может вызывать другой нежелательный сценарий: если температура в T1 становится слишком низкой, кристаллизация портит партию. Следовательно, необходимо осушать T1 перед тем, как это случится. Осушение, однако, может проводиться слишком долго, и оно не может быть начато прежде, чем T2 опустеет, так как нельзя допустить, чтобы две партии были смешаны. Следовательно схема управления должна обеспечить "период ожидания", в течение которого температура в емкости T1 не повысится выше или упадет ниже заданных порогов.

Математическое описание установки выпаривателя

Процесс описывается системой нелинейных дифференциальных уравнений,

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

где t представляет время (непрерывное), $x = (T, m_1, m_2)$ является вектором состояния, где T – температура в емкости Т1, m_1 – общая масса в емкости Т1, m_2 – масса в емкости Т2.

$u_d = (Q_h, Y_{V15}, Y_{V18})$ является вектором управления, где Q_h – количество тепловой энергии, поданной в реактор, Y_{V15} – положение клапана V15, и Y_{V18} – положение клапана V18. Клапаны V15 и V18 могут занимать положение «открыт» или «закрыт». В системе управления управляющие воздействия $u_d \in U_d$ могут принимать четыре состояния:

$$u_d \in U_d = \{u_d^{(1)}, u_d^{(2)}, u_d^{(3)}, u_d^{(4)}\},$$

где

$$u_d^{(1)} = \begin{pmatrix} 5000 \\ 0 \\ 1 \end{pmatrix}, u_d^{(2)} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, u_d^{(3)} = \begin{pmatrix} 5000 \\ 1 \\ 0 \end{pmatrix}, u_d^{(4)} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

Модель системы (A.1) в развернутом виде может быть представлена системой уравнений (A.2) – (A.4), соответственно описывающих балансы массы и энергии:

$$\frac{dT}{dt} = \frac{Q_h - k \left(A_1 + \frac{\pi D m_1}{\rho l A_1} \right) (T - T_e)}{c_{p,L} m_1 - \hat{c}_p m_V - (T \hat{c}_p - \Delta h_{ev}) \frac{dm_V}{dt}}, \quad (\text{A.2})$$

$$\frac{dm_1}{dt} = -Y_{V15} A_R \sqrt{\frac{2 \rho_L m_1 g}{A_1}}, \quad (\text{A.3})$$

$$\frac{dm_2}{dt} = -Y_{V15} A_R \sqrt{\frac{2 \rho_L m_1 g}{A_1}} - Y_{V18} A_R \sqrt{\frac{2 \rho_L m_2 g}{A_2}}, \quad (\text{A.4})$$

где $\hat{c}_p = c_{p,L} - c_{p,V}$ и

$$m_v = \frac{pM_\omega \left(\frac{m_1(t_0)}{pL} + V_v(t_0) - \frac{m_1}{pL} \right)}{R_m \left(T - \frac{pM_\omega}{R_m pL} \right)}, \quad (\text{A.5})$$

$$\frac{dm_v}{dT} = -\frac{m_v}{T - \frac{pM_\omega}{R_m pL}} + \frac{Tm_v}{T - \frac{pM_\omega}{R_m pL}} \cdot \frac{B_p}{(T + C_p)^2} \quad (\text{A.6})$$

Следующие физические величины могут быть использованы: p , давление насыщения воды, моделируется при помощи уравнения Антоина

$$p = e^{A_p} - \frac{B_p}{T + C_p} \quad (\text{A.7})$$

и Δh_{ev} – специфическая энтальпия испарения

$$\Delta h_{ev} = b_1 - b_2 T. \quad (\text{A.8})$$

Параметры модели (A.2) – (A.8) представлены в таблице А.1.

Во избежание возникновения опасного давления, температура T в емкости Т1 не должна превышать 385 К. T может понизиться ниже 338 К, только когда резервуар Т1 почти пустой (уровень жидкости меньше, чем 1 см, или, эквивалентно, m_1 – меньше чем 2.91 – 100g). Кроме того, перелив партии из емкости Т1 в емкость Т2 может начинаться только тогда, когда Т2 пуста. В случае аварии начавшийся перелив не должен быть прерван, прежде чем Т1 опустеет. В заключение, всякий раз, когда резервуар Т1 пуст, нагревание должно быть выключено.

Разработка математической модели СУС проводится в три этапа. На первом этапе разрабатывается модель ОУСУС, на втором – модель УЛУ и на последнем этапе они объединяются между собой.

После подстановки численных значений в уравнения, описывающие УЛУ, получаем следующие формулы, которые являются операторами непрерывных переходов ДН-сети:

Таблица А.1 – Параметры детальной модели

Величина	Значение
Поперечный разрез Т1	$A_1 = 0.03 \text{ m}^2$
Поперечный разрез Т2	$A_2 = 0.06 \text{ m}^2$
Поперечный разрез трубы	$A_R = 2.2 \cdot 10^{-5} \text{ m}^2$
Диаметр Т1	$D = 0.2 \text{ m}$
Коэффициент тепловой передачи	$K = 150.0 \text{ W / K / m}^2$
Гравитационная постоянная	$G = 9.81 \text{ m / s}^2$
Газовая постоянная	$R_m = 8.314 \text{ J / K / mol}$
Молекулярный вес воды	$M_\omega = 0.18 \text{ 100 g / mol}$
Емкость жидкой теплоты	$C_{p,L} = 422 \text{ J / 100g / K}$
Емкость испаряющейся теплоты	$C_{p,v} = 189 \text{ J / 100g / K}$
Плотность жидкости	$P_L = 9700 \text{ 100g / m}^3$
Начальный объем испарения	$V_{V(t_0)} = 0.02 \text{ m}^3$
Начальная масса жидкости	$m_{L(t_0)} = 6.0 \text{ m}^3$
Коэффициент давления	$A_p = 23.478$
Коэффициент давления	$B_p = 3984.9$
Коэффициент давления	$C_p = -39.724$
Коэффициент энтальпии	$B_1 = 3.294 * 10^6 \text{ J / kg}$
Коэффициент энтальпии	$B_2 = 2.78 * 10^3 \text{ J / kg / K}$
Температура окружения	$T_e = 283 \text{ K}$

$$\frac{dT}{dt} = \frac{Qh - (4,5 + 0,323 \cdot m1) \cdot (T - 283)}{422 \cdot m1 - 233 \cdot m_v - (T \cdot 233 - \Delta h_{ev}) \cdot \frac{dm_v}{dt}} \quad (\text{A.9})$$

$$\frac{dm1}{dt} = -Y15 \cdot 22 \cdot 10^{-5} \cdot \sqrt{6343800 \cdot m1} \quad (\text{A.10})$$

$$\frac{dm2}{dt} = Y15 \cdot 22 \cdot 10^{-5} \cdot \sqrt{6343800 \cdot m1} - Y18 \cdot 22 \cdot 10^{-5} \cdot \sqrt{3171900 \cdot m2} \quad (\text{A.11})$$

$$m_v = \frac{P \cdot 0,18 \cdot \left(6,185 \cdot 10^{-4} + 0,02 - \frac{m1}{9700} \right)}{8,314 \cdot (T - 2,23 \cdot 10^{-6} \cdot P)} \quad (\text{A.12})$$

$$\frac{dm_v}{dt} = -\frac{m_v}{T - P \cdot 2,23 \cdot 10^{-6}} + \frac{T \cdot m_v}{T - 2,23 \cdot 10^{-6}} \cdot \frac{3984,9}{(T - 39,724)^2} \quad (\text{A.13})$$

$$P = 1,57 \cdot 10^{10} - \frac{3984,9}{T - 39,724} \quad (\text{A.14})$$

Математическая модель объекта управления, описываемая формулами (A.9) – (A.14), является многомерной, нелинейной, имеет выходные параметры $m1$, $m2$, $T1$ и представлена с использованием средств ДН-сетей на рис. A.2. Входными воздействиями для объекта являются:

- управление положением клапана V12 по каналу позиция p_4^2 переход t_4^3 ;
- управление положением клапана V15 по каналу позиция p_3^2 переход t_3^3 ;
- управление положением клапана V18 по каналу позиция p_2^2 переход t_2^3 ;
- управление положением ввода количества тепловой энергии Q_h по каналу позиция p_1^2 переход t_1^3 .

Выходные непрерывные сигналы: масса жидкости в первом танке m_1 , масса жидкости во втором танке m_2 , температура жидкости в первом танке T - формируются соответственно в непрерывных позициях p_7^1 , p_9^1 и p_5^1 . Выходные дискретные сигналы формируются в позициях p_5^2 , p_6^2 , p_7^2 , p_{19}^2 , p_8^2 , p_9^2 и появление

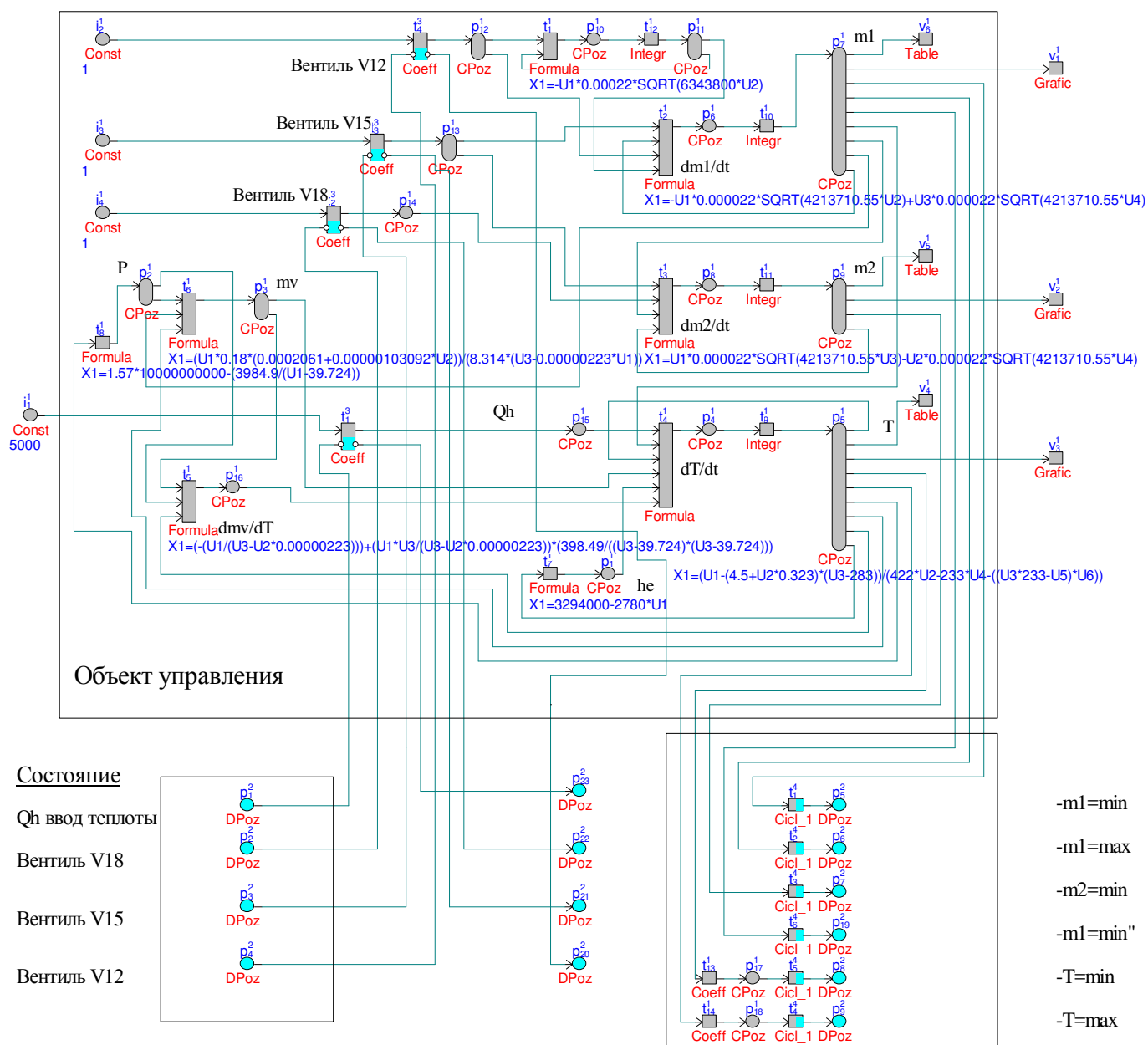


Рисунок А.2 – Представление объекта управления языком ДН-сетей

маркеров в них означает соответственно выполнение следующих условий: достижение массы в емкости T1 минимального значения $m_1 = \min$, достижение массы в емкости T1 максимального значения $m_1 = \max$, достижение массы в емкости T2 минимального значения $m_2 = \min$, достижение массы в емкости T1 минимального значения $m_1 = \min''$, достижение температуры в емкости T1 минимального значения $T = \min$, достижение температуры в емкости T1 максимального значения $T = \max$.

Данными дискретными сигналами помечен алгоритм управления, представленный на рис. А.3.

Процесс управления выпарной установки может быть представлен в виде графсета, показанного на рис. А.3. Составим на основании алгоритма управления УЛУ в соответствии использованием представление средствами ДН-сетей (приложение Б). Получаем УЛУ, представленное на рис. А.4. Здесь за пуск процесса отвечает дискретная позиция p_{10}^2 , а за останов – конечные дискретные позиции p_{17}^2 и p_{18}^2 . Соответственно, наличие маркеров в результате моделирования будет означать успешность проведения процесса и достижимость заданного состояния.

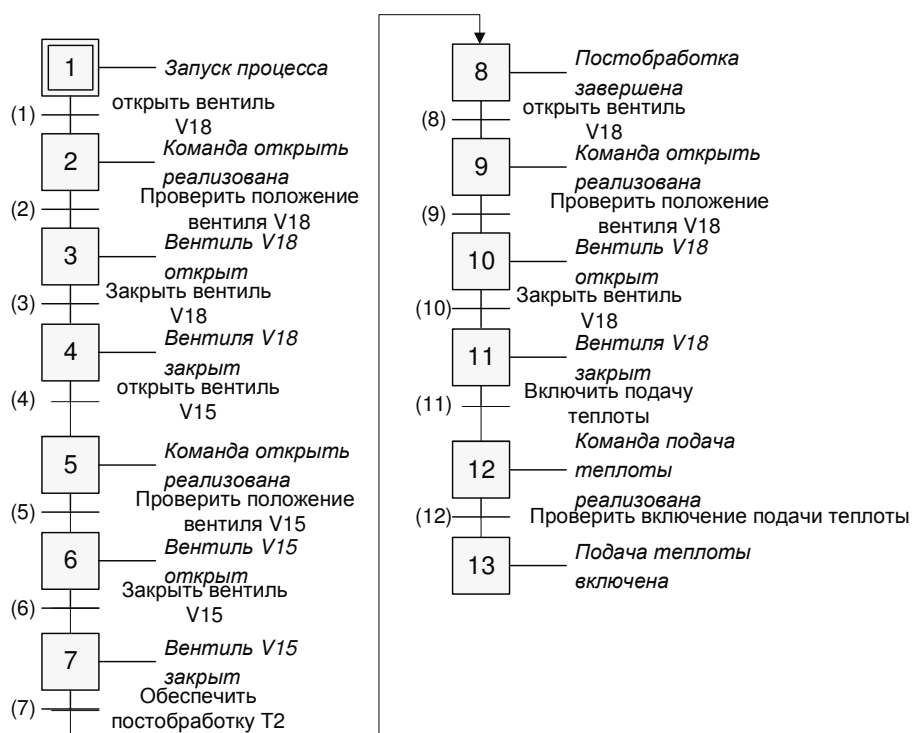


Рисунок А.3 – Графсет алгоритма управления пакетной установки выпаривания раствора хлористого натрия

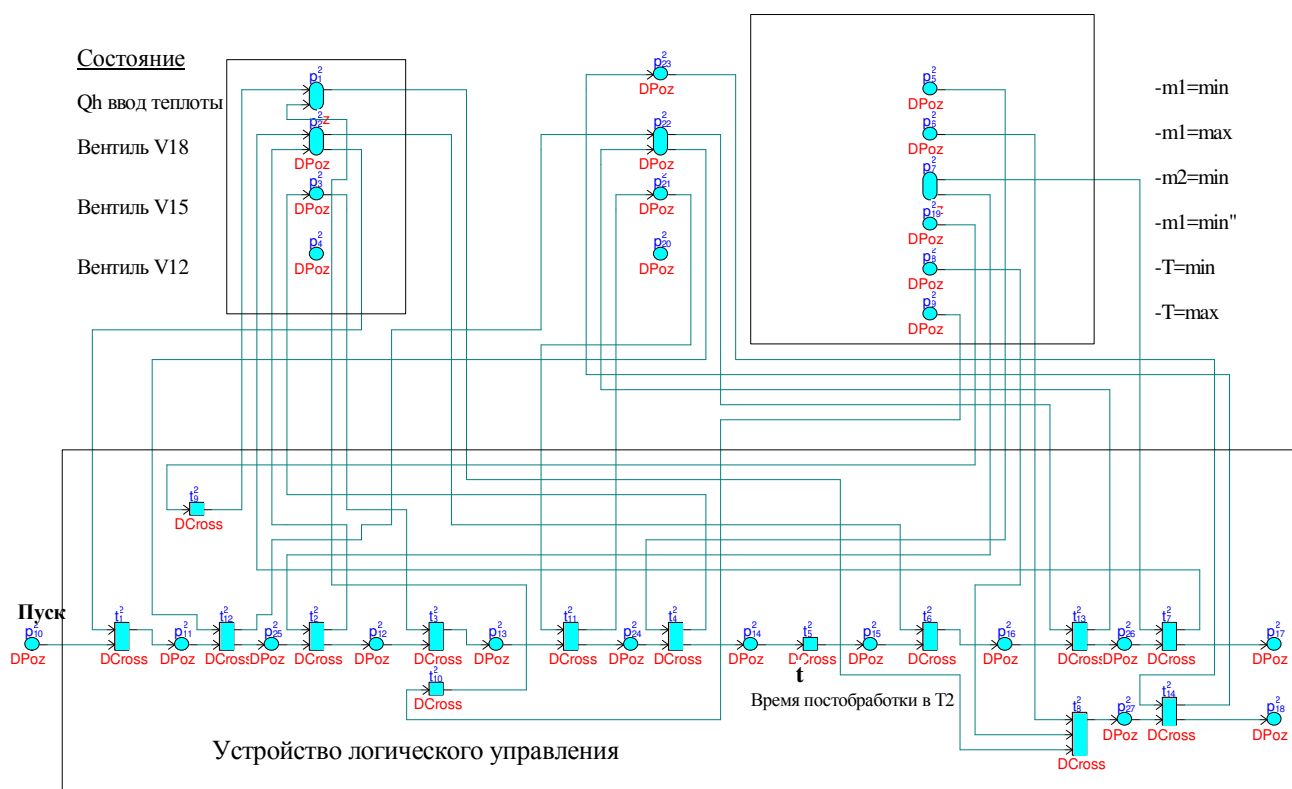


Рисунок А.4 – Представление УЛУ языком ДН-сетей

Запуск процедуры анализа свойства достижимости при помощи аналитического метода обеспечивается в интерактивном режиме нажатием клавиши «Пуск». При этом выбор метода должен находиться в положении «Аналитический». В этом случае если был выбран «Шаговый» тип запуска, то в таблице формируется начальное состояние (см. рис. А.5), последующие нажатия клавиши «Пуск» обеспечивают формирование следующей итерации текущей матрицы системы. Промежуточные результаты итераций анализа свойства достижимости представлены на рис. А.6. На рис. А.7 показана конечная итерация редукции ДН-сети. Как можно определить, из рисунка, ДН-сеть была преобразована в сеть, которая является достижимой. При реализации типа запуска «Полный» нажатие клавиши «Пуск» реализует все итерации и таблица принимает вид, показанный на рис. А.7.

При выполнении численного метода анализа свойства достижимости необходим запуск моделирования исследуемой системы с накоплением векторов маркировок, накапливаемых в таблице.

Начальное состояние

	t_1^1	t_2^1	t_3^1	t_4^1	t_5^1	t_6^1	t_7^1	t_8^1	t_9^1
p_1^1	0	0	0	$-\infty$	0	0	∞	0	
p_2^1	0	0	0	0	$-\infty$	$-\infty$	0	∞	
p_3^1	0	0	0	$-\infty$	$-\infty$	∞	0	0	
p_4^1	0	0	0	∞	0	0	0	0	
p_5^1	0	0	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	
p_6^1	0	∞	0	0	0	0	0	0	
p_7^1	0	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	0	0	
p_8^1	0	0	∞	0	0	0	0	0	
p_9^1	0	0	$-\infty$	0	0	0	0	0	
p_{10}^1	∞	0	0	0	0	0	0	0	
p_{11}^1	$-\infty$	$-\infty$	0	0	0	0	0	0	
p_{12}^1	$-\infty$	$-\infty$	0	0	0	0	0	0	
p_{13}^1	0	$-\infty$	$-\infty$	0	0	0	0	0	
p_{14}^1	0	0	$-\infty$	0	0	0	0	0	
p_{15}^1	0	0	0	$-\infty$	0	0	0	0	
p_1^2	0	0	0	0	0	0	0	0	
p_2^2	0	0	0	0	0	0	0	0	
p_3^2	0	0	0	0	0	0	0	0	
p_4^2	0	0	0	0	0	0	0	0	
p_6^1	0	0	0	$-\infty$	∞	0	0	0	

Режим
 Аналитический
 Маркерный

Тип запуска
 Полный
 Шаговый

Пуск

CAPS NUM OVR

Рисунок А.5 – Начальное состояние исследование свойства достижимости аналитическим методом

Состояние после сокр. непрерывной части

	t_1^2	t_2^2	t_3^2	t_4^2	t_5^2	t_6^2	t_7^2	t_8^2	t_9^2
p_1^2	0	0	0	0	0	0	0	-1	1
p_2^2	-1	1	0	0	0	-1	1	0	0
p_3^2	0	0	-1	1	0	0	0	0	0
p_4^2	0	0	0	0	0	0	0	0	0
p_{10}^2	-1	0	0	0	0	0	0	0	0
p_{11}^2	1	-1	0	0	0	0	0	0	0
p_{12}^2	0	1	-1	0	0	0	0	0	0
p_{13}^2	0	0	1	-1	0	0	0	0	0
p_{14}^2	0	0	0	1	-1	0	0	0	0
p_{15}^2	0	0	0	0	1	-1	0	0	0
p_{16}^2	0	0	0	0	0	1	-1	0	0
p_{17}^2	0	0	0	0	0	0	1	0	0
p_{18}^2	0	0	0	0	0	0	0	1	0
p_6^2	0	0	0	0	0	0	0	0	0

Режим
 Аналитический
 Маркерный

Тип запуска
 Полный
 Шаговый

Пуск

CAPS NUM OVR

Рисунок А.6 – Состояние после редукции НС части

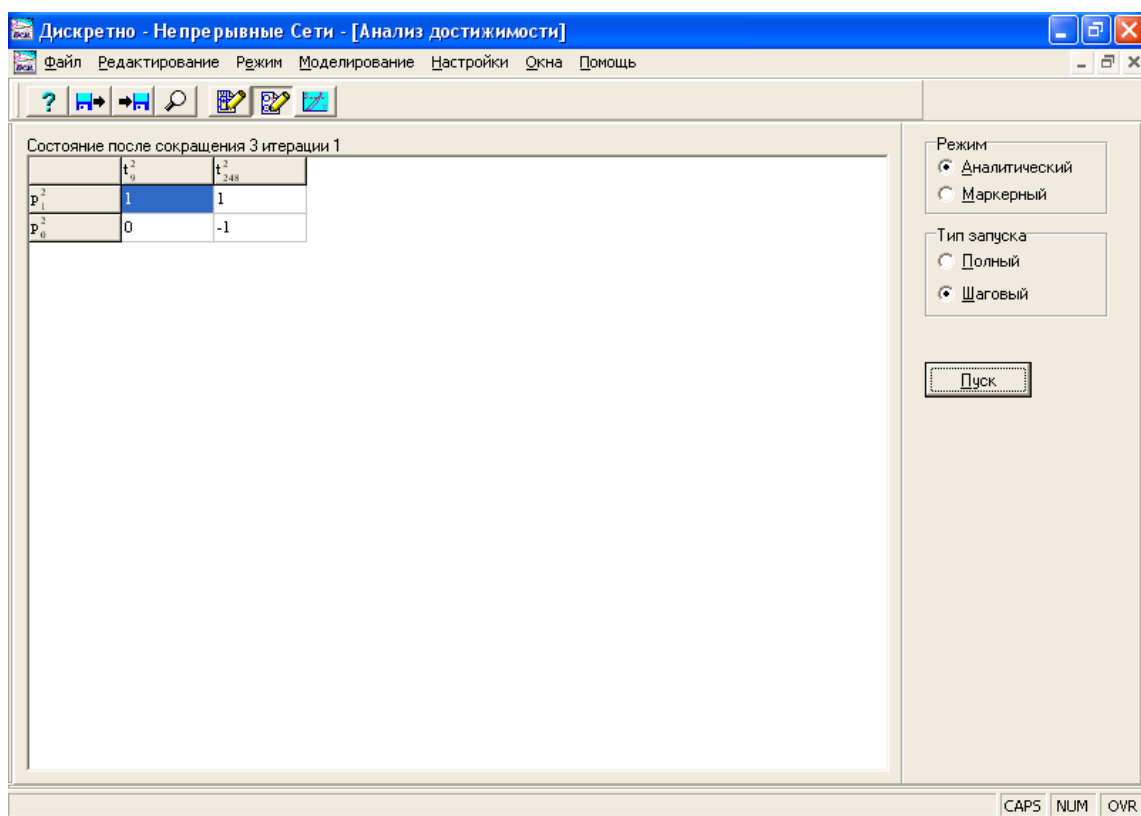


Рисунок А.7 – Конечное состояние после редукции ДС части

Окончательный вид накопленных маркеров показан на рис. А.8. Как видно из рисунка, конечное состояние при проведении моделирования было достигнуто, поскольку в обеих позициях p_{17}^2 и p_{18}^2 имеются метки.

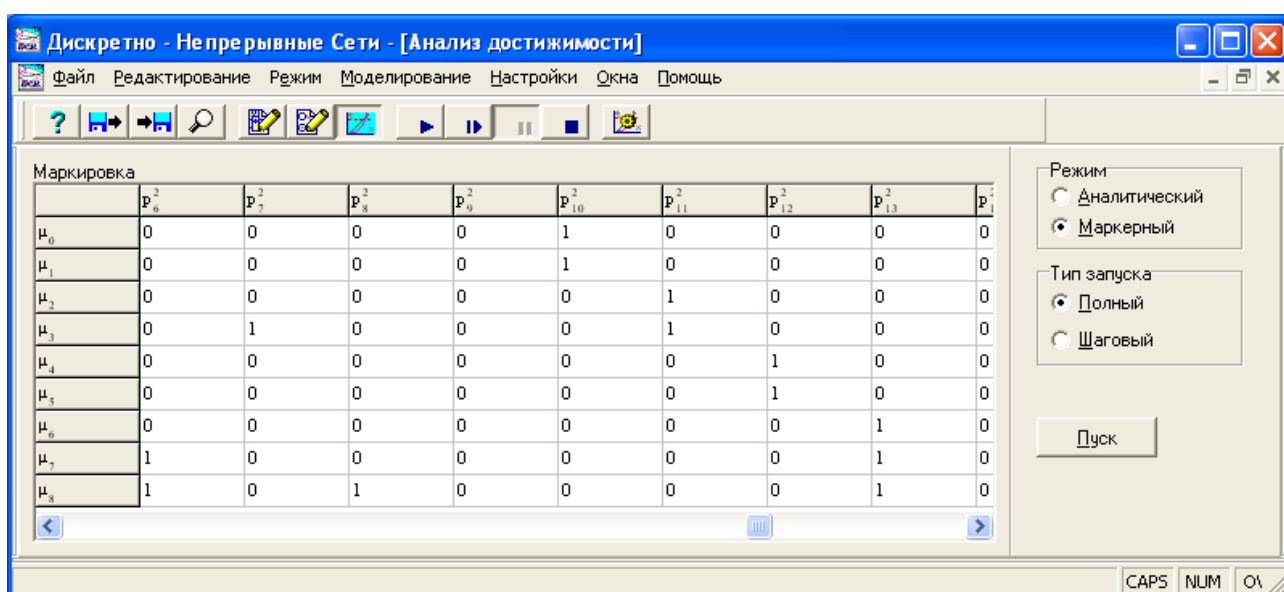


Рисунок А.8 – Конечная маркировка

ПРИЛОЖЕНИЕ Б

КРАТКОЕ СПРАВОЧНОЕ РУКОВОДСТВО ПРОГРАММНОЙ СРЕДЫ DCNET

Б.1 Интерфейс программы DC-Net

В системе DC-Net используется многооконный интерфейс с ниспадающими и разворачивающимся меню. Объясним назначение отдельных элементов программы, основное окно которой изображено на рис. Б.1.

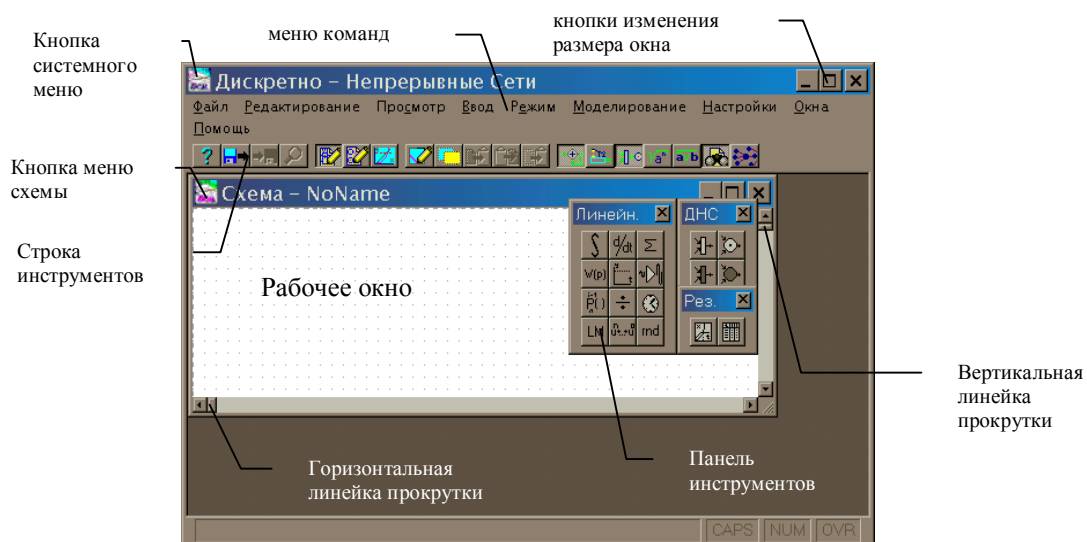


Рисунок Б.1 – Основное окно программной среды DCNET

Кнопка системного меню. Расположена в верхнем левом углу окна и является стандартной кнопкой управления приложениями Windows. С её помощью изменяется и восстанавливается размер окон, они перемещаются, сворачиваются и закрываются.

Кнопка меню схемы. Аналогична кнопке системного меню, но применяется при работе с окнами схемы.

Меню команд. Меню выбранной курсором команды разворачивается вниз. Некоторые команды, например **File-Save** (Файл /Сохранить), выполняются немедленно после их выбора. Другие команды, отмеченные многоточием "...", ведут к появлению соответствующего диалогового окна.

Вертикальная линейка прокрутки. Служит для перемещения по листу схемы в вертикальном направлении.

Горизонтальная линейка прокрутки. Служит для перемещения по листу схемы в горизонтальном направлении.

Демонстрацию возможностей программы **DC-NET** проведём на одной из схем примеров, приведенных в каталоге **Examples**. Загружаем схему по команде **File-Open...** или нажав на кнопку на панели инструментов. В открывшемся окне (см. рис.2) выбираем каталог **C:\DCNET\EXAMPLES**, и выбираем файл **drath.dcn**. В результате загружается схема, показанная на рис. Б.2.

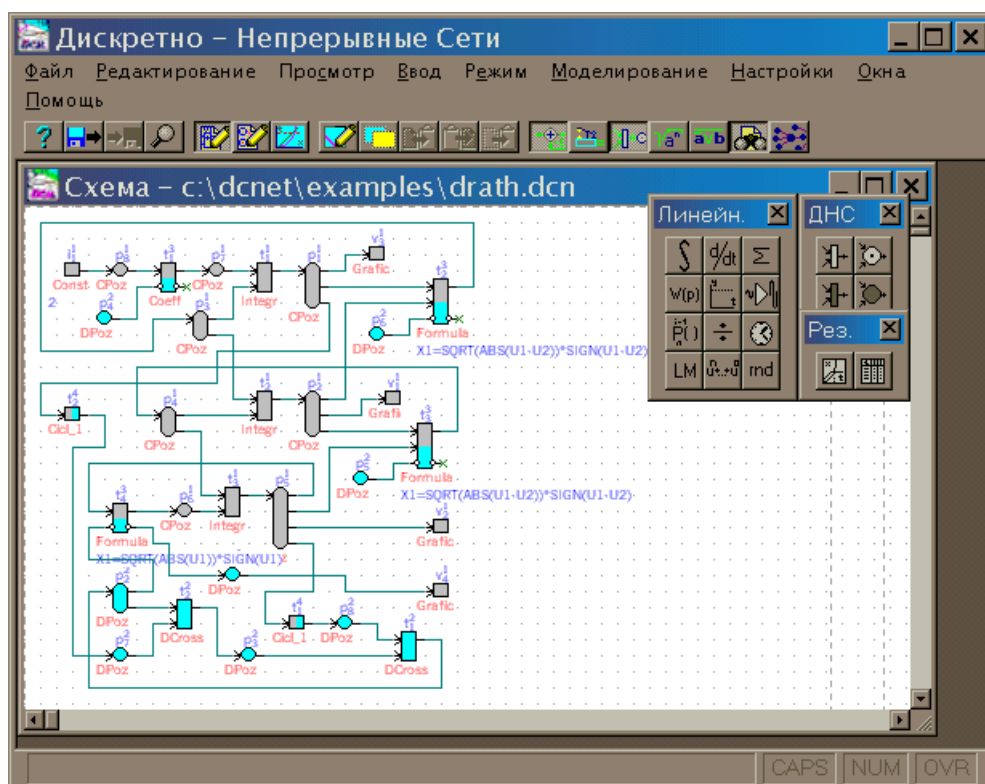


Рисунок Б.2 – Пример отображения схемы исследуемой системы в режиме редактирования

После загрузки схемы программа находится в режиме корректировки схемы. Режим корректировки схемы предназначен для изменения параметров элементов системы, удаления элементов и элементов, а также их объединения в группы. Курсор в этом режиме имеет вид стандартной стрелки. Переход к режиму редактирования схемы, из любого другого режима (*режим установки элементов, режим задания связей*) осуществляется нажатием правой клавиши мыши либо клавиши *Esc*.

Для ввода элементов исследуемой модели используется *режим установки элементов*. Для входа в *режим установки элементов* необходимо выбрать устанавливаемый элемент либо нажав левую клавишу мыши на требуемом элементе в одной из панелей элементов, либо воспользовавшись пунктом главного меню *Ввод*. При этом выбранный элемент подсвечивается на панели другим цветом, и курсор принимает форму элемента. После этого нажатие левой клавиши мыши, в области редактирования схемы, вызывает запрос параметров элементов и устанавливает его. Для перехода к вводу элементов другого типа необходимо либо произвести нажатие левой клавиши мыши на одной из панелей элементов, либо воспользоваться меню. Нажатие правой клавиши мыши возвращает пользователя в *режим корректировки схемы*.

Информацию о топологии исследуемой модели позволяет ввести *режим задания связей*. Для перехода в режим задания связей необходимо нажать кнопку панели инструментов, изображающую карандаш, либо воспользоваться пунктом *Линия* в подменю *Ввод* главного меню.

При этом курсор меняет свое изображение на изображение не закрашенного карандаша, и работа осуществляется следующим образом. При нажатии на левую клавишу мыши курсор, изменяет свой вид на изображение закрашенного карандаша и от точки, где было сделано нажатие, за ним начинает тянуться линия. Для того чтобы повернуть линию необходимо снова нажать на левую клавишу мыши, а для того чтобы завершить рисование линии необходимо сделать быстро два нажатия на левую клавишу (*Double Click*). При этом правая клавиша мыши во время проведения линии, т.е. когда курсор имеет форму закрашенного карандаша,

осуществляет отмену рисования текущей линии (карандаш становится не закрашенным), а когда линия не проводится, также как и при установке элементов, возвращает пользователя в режим корректировки схемы.

После ввода схемы или загрузки готовой схемы, как в нашем случае, следующим этапом является переход в *режим коррекции разметки сети*. Активизация режима осуществляется при нажатии на кнопку *Режим коррекции разметки сети* панели инструментов с помощью мыши. При активизации данного режима из *режима редактирования схемы* происходит проверка схемы модели на наличие в ней ошибок, и при обнаружении таковых выдаются соответствующие сообщения, и активизации режима не происходит. Предупреждения также выдаются при несовпадении имён подсоединённых друг к другу входов и выходов, а также их типов (непрерывный или дискретный). При удачной активизации режима выдаётся звуковой сигнал, а на схеме появляется разметка сети (с указанием количества маркеров в позиции для дискретных позиций). Вид схемы в этом режиме показан на рис. Б.3.

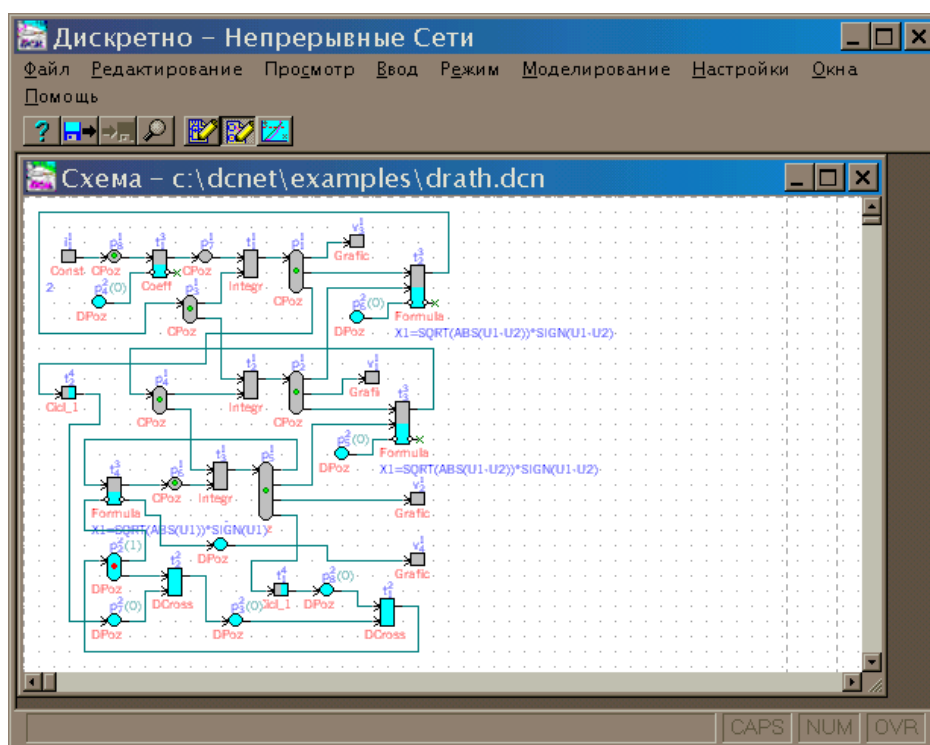


Рисунок Б.3 – Пример отображения схемы исследуемой системы в режиме коррекции разметки

Для коррекции разметки необходимо в режиме коррекции разметки сети "щелкнуть" по позиции левой клавишей мыши для установки маркера в выбранную позицию, или правой клавишей мыши – для изъятия маркера из позиции.

При двойном "щелчке" левой клавишей мыши на дискретной позиции вызывается диалоговое окно, в котором необходимо задать целое число – количество маркеров в позиции.

Следующим этапом является моделировка схемы. Для этого необходимо перейти в *режим моделирования*. Активизация режима моделирования осуществляется при нажатии на кнопку *Режим моделирования* панели инструментов с помощью мыши. При активизации данного режима *из режима редактирования схемы* происходит проверка схемы модели на наличие в ней ошибок, и при обнаружении таковых выдаются соответствующие сообщения, и активизации режима не происходит. Предупреждения также выдаются при несовпадении имён подсоединённых друг к другу входов и выходов, а также их типов (непрерывный или дискретный) (Если соответствующие параметры заданы в *Общих настройках* программы). Необходимо отметить, что одним из необходимых условий для моделирования схемы является подключённость всех входов элементов к выходам, и при обнаружении такой ошибки курсор мыши устанавливается в местонахождения элемента или связи (окно схемы открывается, если оно было закрыто). При удачной активизации режима становятся доступными кнопки управлением режима на панели инструментов, и выдаётся звуковой сигнал, а на схеме появляется разметка сети.

Для запуска моделирования необходимо выполнить команду *пуск моделирования*. Состояние пуск моделирования активируется путём:

- нажатия кнопки *Пуск моделирования* на панели инструментов;
- нажатия клавиши *F9*;
- выбора команды *Пуск* меню *Моделирование*.

После запуска моделирования появляется окно, в котором задаются параметры моделирования: начальное время, конечное время, шаг интегрирования,

шаг фиксации и выбирается метод моделирования. После нажатия кнопки **ОК** получаем результаты моделирования, приведенные на рис. Б.4.

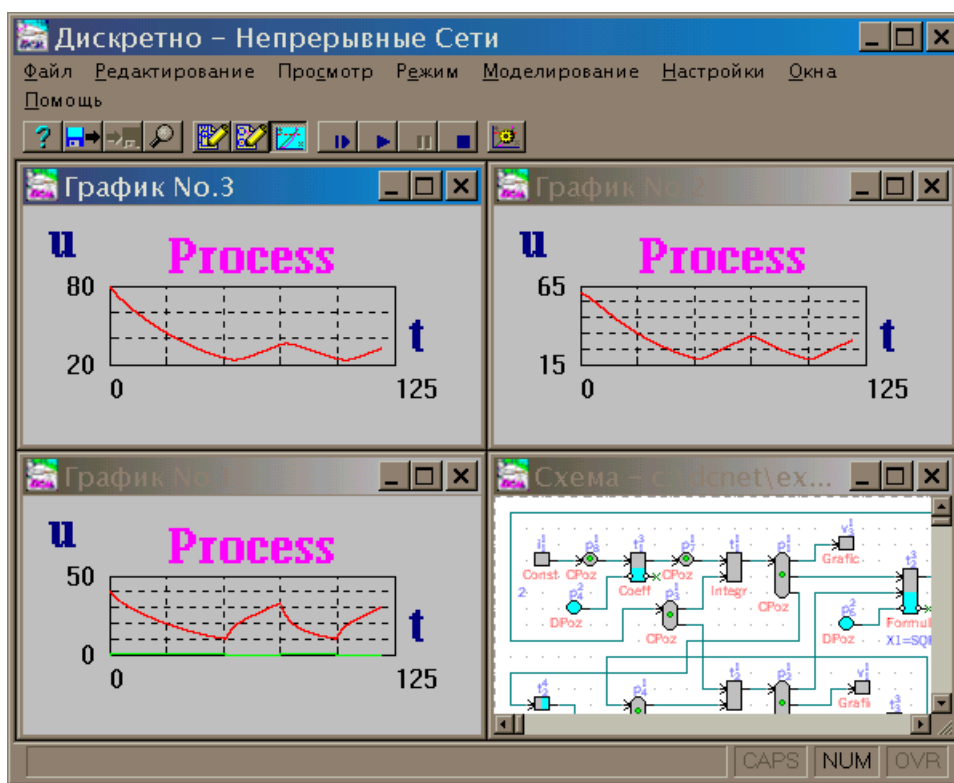


Рисунок Б.4 – Пример работы среды в режиме моделирования

Б.2 Выбор режимов работы

Окно выбора режима работы представлено на рис. Б.5. Оно отображается при запуске программной среды и позволяет запустить процесс подключения к серверу БД, переключиться в режим редактирования и моделирования схемы исследуемой системы, провести анализ на достижимость, настроить параметры среды, а также получить справочную информацию о программной среде.

При выборе работы с БД производится подключение к выбранному серверу БД, после чего определяется режим взаимодействия с группой пользователей, подключенной к данному серверу на основании предыдущих настроек системы и текущего состояния БД.



Рисунок Б.5 – Окно выбора режима работы

При запуске анализа на достижимость предварительно загружается последняя открытая модель, которая предварительно компилируется с проверкой ее корректности и наличия ошибок, после чего проводится автоматический запуск анализа аналитически методом. При необходимости возможен анализ на достижимость с помощью моделирования.

Б.3 Главное меню

Главное меню программы содержит следующие меню, изменяющиеся в зависимости от текущего активного окна и состояния программы и состоит из следующих пунктов.

- **Файл** – Содержит команды работы со схемой модели, вывода на печать и выхода из программы.
- **Редактирование** – Содержит команды редактирования модели, экспорта графических файлов и работы с буфером обмена.
- **Просмотр** – Содержит команды управления видимостью панелей инструментов.
- **Режим** – Содержит команды изменения текущего режима программы.

- **Моделирование** – Содержит команды управления режимом моделирования.
- **Настройки** – Содержит команды изменения текущих настроек программы и модели.
- **Окна** – Содержит команды управления видимостью, размерами и расположением окон программы.
- **Помощь** – Содержит команды вызова помощи.
- При активном окне схемы модели имеются следующие дополнительные меню:
- **Ввод** – Содержит подменю команд управления установки элементов, а также проведения линий и переключения в режим работы с элементами.
- При активном окне таблиц имеются следующие дополнительные меню:
- **Сервис** – Содержит сервисные команды работы с таблицами.

Рассмотрим каждое из них более подробно.

Меню **Файл** содержит команды работы со схемой модели, вывода на печать и выхода из программы.

- **Новый** – Создает новую модель путем закрытия текущего редактируемого файла модели и очистки схемы и имени файла.
- **Открыть** – Выводит диалоговое окно открытия существующей модели путем чтения соответствующего файла модели, заданного пользователем.
- **Сохранить** – Сохраняет данную модель. Если модель сохраняется впервые, то действие данной команды совершенно аналогично команде **Сохранить как....**
- **Сохранить как...** – Выводит диалоговое окно сохранения данной модели под новым именем и местоположением.
- **Комментарий...** – Выводит диалоговое окно для ввода и/или изменения комментария к модели.
- **Печать...** – Выводит диалоговое окно для распечатки содержимое окна.
- **Настройка принтера...** – Выводит диалоговое окно для настройки вида страницы для печати и изменения настроек принтера.
- **Выход** – Закрывает приложение, предварительно запрашивая сохранение не сохраненной модели.

При наличии активного соединения с сервером базы данных данное меню дополняется следующими пунктами.

- **Подключиться к ДВ...** – Подключиться к серверу базы данных.
- **Открыть из ДВ...** – Открыть существующую модель из базы данных. При этом восстанавливаются все открытые окна графиков и таблиц, а также данные состояния модели.
- **Сохранить ДВ как...** – Сохраняет текущую модель под новым именем в базе данных. При этом сохраняется все текущие открытые окна графиков и таблиц, а также данные состояния модели.
- **Импортировать таблицу из ДВ...** – Открывает окно импорта данных в окно Таблица из базы данных.
- **Экспортировать таблицу в ДВ...** – Открывает окно экспорта данных из окна Таблицы в базу данных.
- **Импортировать график из ДВ...** – Открывает окно импорта данных в окно Графика из базы данных.
- **Экспортировать график в ДВ...** – Открывает окно экспорта данных из окна График в базу данных.

Меню **Редактирование** Содержит команды редактирования модели, экспорта графических файлов и работы с буфером обмена.

- **Отменить** – Отменяет последнюю операцию, произведенную пользователем, если она привела к изменению вида схемы модели.
- **Вырезать** – Вырезает выделенное и помещает его в *Буфер Обмена*.
- **Копировать** – Копирует выделенное в текущем окне в *Буфер Обмена*. Если текущее окно – окно графиков, то данная команда помещает график в *Буфер Обмена*.
- **Вставить** – Вставляет содержимое *Буфера Обмена* в текущую позицию.
- **Удалить** – Удаляет выделенное.
- **Копировать в...** – Копирует выделенное в графический файл. Если текущее окно – окно графиков, то данная команда помещает график в графический файл.

- **Подменю Выделить** – Содержит команды выделения содержимого текущего окна: выделяет все содержимое окна, все линии или все элементы на схеме.
- **Подменю Развыделить** – Содержит команды снятия выделения с содержимого текущего окна.
- **Перенумерация элементов** – Перенумеровывает все элементы схемы в порядке их обнаружения во внутренних массивах схемы модели (т.е. в порядке их установки). Данная команда изменяет только цифровые номера элементов и не затрагивает их буквенные обозначения или верхние индексы.
- **Поиск элемента...** – Выводит диалоговое окно поиска элемента по номеру или названию.

Меню **Просмотр** содержит команды управления видимостью панелей инструментов. Текущее состояние панелей инструментов отображается возле соответствующих строк меню – видимые в данный момент панели инструментов отмечены.

- **Линейн.** – Показывает / прячет панель установки линейных элементов.
- **Нелин.** – Показывает / прячет панель установки нелинейных элементов.
- **ДНС** – Показывает / прячет панель установки элементов дискретно-непрерывных сетей и сетей Петри.
- **Мат.** – Показывает / прячет панель установки математических элементов.
- **Логич.** – Показывает / прячет панель установки логических элементов.
- **Рез.** – Показывает / прячет панель установки элементов отображения результатов моделирования.
- **Нейро** – Показывает / прячет панель установки элементов нейронных сетей и элементов оптимизации.
- **Граф. ред.** – Показывает / прячет панель установки элементов графического редактора.

Меню **Режим** содержит команды изменения текущего режима программы. Текущее состояние программы (текущий режим) отображается возле соответствующей строки меню – текущий режим отмечен.

- **Редактирование** – Переключает программу в режим **Редактирования**.
- **Изменение маркировки** – Переключает программу в режим **Изменение маркировки**.
- **Моделирование** – Переключает программу в режим **Моделирования**.

Меню **Моделирование** содержит команды управления режимом моделирования.

- **Пуск** – Запускает моделирование до окончания времени моделирования или до ввода новых команд пользователя, переключаясь в режим **Моделирования**, если текущий режим какой-либо другой.
- **Шаг** – Делает **Шаг** моделирования до окончания времени моделирования или до наступления какого-либо события в процессе моделирования или до ввода новых команд пользователя, переключаясь в режим **Моделирования**, если текущий режим какой-либо иной.
- **Пауза** – Приостанавливает моделирование до ввода новых команд пользователя, переключаясь в режим **Моделирования**, если текущий режим какой-либо иной.
- **Стоп** – Останавливает моделирование до ввода новых команд пользователя.

Меню **Настройки** содержит команды изменения текущих настроек программы и модели.

- **Основные...** – Вызывает диалоговое окно для изменения **Основных** настроек программы, таких как язык, реакцию на обнаруженные ошибки, автоинициализация разметки.
- **Память...** – Вызывает диалоговое окно для изменения настроек **Памяти**, таких как максимальное количество элементов, связей, окон таблиц и графиков и т.п.
- **Моделирование...** – Вызывает диалоговое окно для изменения настроек **Моделирования**, таких как начальное и конечное время и шаг моделирования и накопления результатов, а также значения варьируемых параметров.
- **Схема...** – Вызывает диалоговое окно для изменения текущих настроек **Схемы**, таких как размеры и увеличение схемы, видимость сетки и разбиения на страницы при печати, а также способ нумерации элементов модели.

Меню **Окна** содержит команды управления видимостью, размерами и расположением окон программы.

- **Каскад** – Упорядочивает открытые окна каскадом.
- **Черепица** – Упорядочивает открытые окна черепицей.
- **Упорядочить значки** – Упорядочивает свернутые окна вдоль нижнего края.
- **Заккрыть все** – Закрывает все открытые окна.
- **Схема** – Открывает окно схемы модели, если оно не было открыто ранее.

Меню **Помощь** содержит команды вызова помощи.

- **Текущая** – Выводит текущую страницу контекста системы помощи.
- **Индекс** – Выводит индекс помощи.
- **Клавиатура** – Выводит помощь о работе с клавиатурой.
- **Работа с помощью** – Выводит информацию об использовании системы помощи
- **О программе...** – Выводит диалоговое окно с информацией о версии программы.

Меню **Ввод** доступно только при активном окне схемы модели. Содержит подменю команд управления установки элементов, а также проведения линий и переключения в режим работы с блоками. Текущее состояние программы (текущий режим редактирования) отображается возле соответствующей строки меню (или подменю – если режим установки какого-либо элемента) – текущий режим (работа с блоками, проведение связей или устанавливаемый элемент) отмечен. Команды, доступные из данного меню следующие.

- **Блок** – Переключается в режим **Работы с Блоками** или выйти из него.
- **Линия** – Переключается в режим установки **Соединений (Линий)** в модели или выйти из него.
- Подменю **Линейн.** – Содержит линейные элементы.
- Подменю **Нелин.** – Содержит нелинейные элементы.
- Подменю **ДНС** – Содержит элементы дискретно-непрерывных сетей и сетей Петри.
- Подменю **Мат.** – Содержит математические элементы.
- Подменю **Логич.** – Содержит логические элементы.

- Подменю **Рез.** – Содержит элементы отображения.
- Подменю **Нейро** – Содержит элементы нейронных сетей и оптимизации.
- Подменю **Граф. ред.** – Содержит элементы графического редактора.

Меню **Сервис** доступно только при активном окне редактора схемы исследуемой системы или окне таблиц.

При активном окне редактора схемы содержит следующие команды работы с исследуемой системой.

- **Анализ Корректности** – Произвести анализ схемы на корректность. В данном режиме проводится анализ исследуемой схемы на наличие конфликтов. При обнаружении конфликтной ситуации вершины ДН-сети, имеющие конфликты будут помечены двумя линиями в виде буквы «V» поверх изображения соответствующего элемента. Запуск данного анализа возможен только при наличии откомпилированной модели, т.е. из режима коррекции разметки либо моделирования.
- **Снять метки** – Снять метки после анализа схемы на корректность и/или достижимость. Проводится очистка схемы исследуемой системы от меток, оставшихся на ней после проведения анализа свойства достижимости и/или корректности.
- **Анализ Достижимости** – Провести анализ схемы на достижимость. При выборе данной команды производится запуск анализа свойства достижимости. Запуск данного анализа возможен только при наличии откомпилированной модели, т.е. из режима коррекции разметки либо моделирования.
- **Flash анимация** – Провести анимацию схемы при помощи Flash. Требуется наличие внешнего модуля, осуществляющего анимацию. Среда разработки осуществляет только передачу данных и взаимодействие с данным модулем через файлы в формате XML.

При активном окне Таблицы содержит сервисные команды работы с таблицами.

- **Передать в новое окно графиков...** – Передает данные из Таблицы в новое окно Графика.

- **Передать в существующее окно графиков...** – Передает данные из Таблицы в существующее окно Графика.

Команда **Передача данных в окно графиков** активируется путём выбора команд:

- **Передать в новое окно графиков...** меню **Сервис** при активном окне таблицы
 - для создания нового окна графиков и построения графика заданной зависимости (от времени либо от другого параметра, присутствующего в данной таблице);
- **Передать в существующее окно графиков...** меню **Сервис** при активном окне таблицы – для построения графика заданной зависимости в существующем окне графиков, уже созданном при помощи команды **Передать в новое окно графиков...**, (от времени либо от другого параметра, присутствующего в данной таблице).

Построение графиков возможно в режимах **Пауза моделирования** или **Останов моделирования**, причем в последнем предпочтительнее, т.к. при продолжении моделирования и появлении новых данных график строится далее уже не будет.

Б.4 Панель инструментов.

Панель инструментов имеет различный вид при различных режимах работы программы. Оно может содержать от двух до четырёх групп кнопок, причём только первые две не изменяются при изменении режима работы программы. Первая из них содержит кнопки помощи, работы с файлами и поиска; вторая – кнопки переключения режимов работы программы (только одна из них может быть нажатой и всегда одна из них нажата); третья в режиме редактирования схемы содержит кнопки управления режимом редактирования, в режиме моделирования – кнопки управления моделированием (только одна из них может быть нажатой), а в режиме изменения разметки вообще отсутствует; четвёртая группа кнопок доступна только в режиме редактирования схемы и содержит кнопки, управляющие видимостью

окнами, содержащими кнопки для установки элементов (управление видимостью также возможно и из меню **Просмотр** – отмеченные группы элементов видны в данный момент).

При открытии файла осуществляется проверка на соответствие формата выбранного файла текущей версии программы, и в случае несоответствия, выдаётся сообщение об ошибке и открытия файла не происходит.

По умолчанию имя файла модели имеет расширение "*dcn*".

При записи файла осуществляется проверка на наличие файла с таким же именем, и в случае его существования происходит создание резервной копии с таким же именем и расширением "*bak*". В случае невозможности записи файла выдаётся сообщение об ошибке.

Б.5 Поиск элемента

Поиск элемента активируется путём нажатия кнопки **Поиск элемента** на панели инструментов. При этом появляется диалоговое окно с полями, в котором необходимо задать информацию для критерия поиска элемента. Если необходимо исключить какой-либо тип информации из критерия поиска, то для этого достаточно убрать маркер из поля слева от содержащего требуемую информацию поля.

Имеются следующие поля:

- **Номер элемента** – позволяет ввести целое число – номер (нижний индекс в режиме Нумерация элементов как в ДН-сети) требуемого элемента. Имеющаяся проверка не позволяет вам задать номер элемента, не являющийся или какой-либо ошибочную информацию.
- **Тип элемента** – позволяет ввести алфавитно-цифровые символы типа искомого элемента (стоящий перед номером). Возможность установки поля присутствует, только если включен режим *Нумерация элементов как в ДН-сети*.
- **Индекс элемента** – позволяет ввести алфавитно-цифровые символы индекса искомого элемента (*верхний индекс* в режиме *Нумерация элементов как в ДН-*

сети). Возможность установки поля присутствует, только если включен режим Нумерация элементов как в ДН-сети.

- **Название элемента** – позволяет ввести название элемента (в таком виде, как оно отображается на схеме). Название будет восприниматься даже для тех элементов, для которых его отображение отключено на текущий момент.
- **Начать поиск с N** – позволяет ввести целое число – номер (нижний индекс в режиме Нумерация элементов ДН-сети) требуемого элемента, с которого необходимо начать поиск. Поиск проводится, начиная с элемента с заданным номером по порядку установки элементов. Т.е., если сначала был установлен элемент с номером 5, а затем 2 и здесь задан номер 5, то при выполнении остальных условий будет найден элемент 2 (при прямом порядке поиска). После удачного поиска содержимое данного поля изменяется на номер найденного элемента – данное свойство позволяет организовать продолжение поиска, если в схеме имеются одинаковые элементы. Имеющаяся проверка не позволяет вам задать номер элемента, не являющийся максимальным имеющимся или какую-либо ошибочную информацию.
- **Поиск только по выделенному** – ограничивает критерий поиска только выделенными элементами. Его установка возможна только если есть выделенные элементы.
- **Поиск внутри макро** – распространяет критерий поиска на содержимое макроэлементов в схеме, начиная с текущего уровня вложенности и до последнего. Его установка возможна только если в схеме имеются макроэлементы.
- **Поиск в обратном направлении** – изменяет направление поиска на реверсивное (от конца к началу – по времени установки элементов). Его установка возможна только при активном поле *Начать поиск с N*.

Если элемент с заданным номером будет найден, то будет сделана попытка поместить его в центр окна и установить на него курсор мыши. В случае неудачного поиска выдаётся сообщение об ошибке.

Для начала поиска следует нажать на клавишу **OK** при помощи мыши; или нажать на клавиши **Alt-O** ; или нажать на клавишу **Enter** .

Для того чтобы закрыть диалоговое окно и отказаться от поиска необходимо нажать на клавишу **Cancel** при помощи мыши; или нажать на клавишу **Escape** (или **Alt-C**).

Для получения справки необходимо нажать на клавишу **Help** при помощи мыши; или нажать на клавишу **Alt-H** .

Б.6 Редактирование схемы.





Окно редактирования схемы позволяет пользователю осуществить основные действия по вводу топологии и параметров элементов модели, представляемой в виде дискретно-непрерывной сети.

Основными режимами редактирования, реализуемыми программой, являются:

- *режим установки элементов (позиций и переходов) моделируемой сети;*
- *режим задания связей;*
- *режим корректировки схемы;*
- *режим работы с группой выделенных элементов.*

Каждый режим характеризуется своей формой курсора и назначением клавиш мыши. Так в режиме установки элементов курсор отображается в виде элемента, в режиме задания связей в виде карандаша, в режиме корректировки схемы в виде стандартной стрелки, и, наконец, в режиме работы с группой элементов в виде перекрестья (см. табл. Б.1).

Таблица Б.1 – Виды курсоров

Элементы редактора	Указатель элемента	Указатель Кисть	Указатель Карандаш	Указатель Карандаш
Визуальное отображение				

Б.7 Режим установки элементов

Режим установки элементов позволяет пользователю осуществить ввод элементов исследуемой модели. В качестве элементов могут выступать дискретные позиции и переходы, непрерывные позиции и переходы, которые могут быть описаны в виде дифференциальных уравнений, передаточных функций либо формул, и управляемые непрерывные элементы (дискретно-непрерывные переходы).

Перед началом установки элементов рекомендуется с помощью кнопок управления видимостью панелей элементов включить панели, с которыми предполагается вести работу, например, панели ДНС, линейных элементов и математических функций.

Далее необходимо выбрать устанавливаемый элемент либо нажав левую клавишу мыши на требуемом элементе в одной из панелей элементов, либо воспользовавшись пунктом главного меню **Ввод**. При этом выбранный элемент подсвечивается на панели другим цветом, и курсор принимает форму элемента (см. табл. Б.1). Для удобства все имеющиеся элементы библиотеки элементов сгруппированы по функциональному признаку. Все они показаны на рис. Б.6. В таблице Б.2 приведено соответствие пиктограмм, показанных на рис. Б.6 и их значения.

После этого нажатие левой клавиши мыши, в области редактирования схемы, вызывает запрос параметров элементов и устанавливает его. Для перехода к вводу элементов другого типа необходимо либо произвести нажатие левой клавиши мыши на одной из панелей элементов, либо воспользоваться меню.

Нажатие правой клавиши мыши возвращает пользователя в режим корректировки схемы.

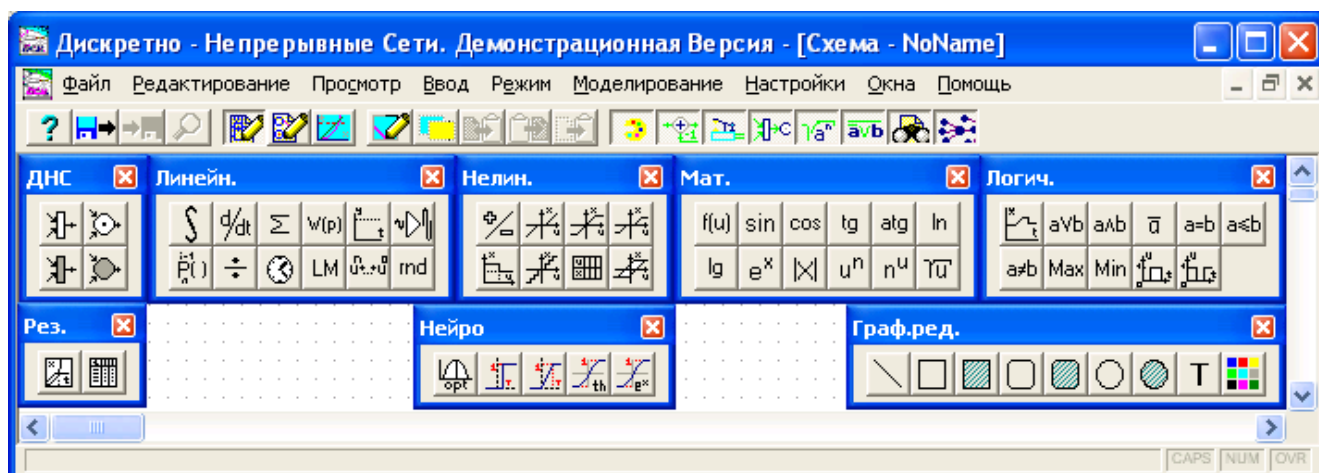













Рисунок Б.6 – Пиктограммы элементов, используемые в среде «Дискретно-непрерывные сети» (DCNET)





Таблица Б.2 – Значение пиктограмм, используемых в интерфейсе

Пиктограмма	Значение
	Помощь
	Открытие файла
	Запись файла
	Поиск элемента
	Переключение в режим корректировки схемы
	Переключение в режим коррекции разметки сети
	Переключение в режим моделирования
	Переключение в режим задания связей
	Переключение в режим работы с элементами
	Копирование в буфер обмена
	Вставка из буфера обмена
	Удаление в буфер обмена
	Шаг моделирования
	Пуск моделирования
	Пауза моделирования
	Останов моделирования
	Параметры моделирования
















Продолжение табл. Б.2

<i>Пиктограмма</i>	<i>Значение</i>
	Группа «Линейные элементы»
	Группа «Нелинейные элементы»
	Группа «Элементы ДН-сетей и сетей Петри»
	Группа «Математические элементы»
	Группа «Логические элеметы»
	Группа «Элементы наблюдения за процессом моделирования»
	Группа «Элементы нейронных сетей»
	Группа «Элементы графического редактора»
Σ	Установка элемента «Сумматор»
\times	Установка элемента «Умножитель»
\div	Установка элемента «Делитель»
	Установка элемента «Логическая формула»
$a \vee b$	Установка элемента «Логическое И»
$a \wedge b$	Установка элемента «Логическое ИЛИ»
\bar{a}	Установка элемента «Логическое НЕ»
$a = b$	Установка элемента «Логическая эквивалентность»
$a \leq b$	Установка элемента «Переключение по \leq »
$a \neq b$	Установка элемента «Переключение по не $=$ »
Min	Установка элемента «Минимум»
Max	Установка элемента «Максимум»
	Установка элемента «Циклограмма 1»
	Установка элемента «Циклограмма 0»
$f(u)$	Установка элемента «Задание формул»
sin	Установка элемента «Синус»
cos	Установка элемента «Косинус»
tg	Установка элемента «Тангенс»
atg	Установка элемента «Арктангенс»

Продолжение табл. Б.2

<i>Пиктограмма</i>	<i>Значение</i>
	Установка элемента «Логарифм натуральный»
	Установка элемента «Логарифм десятичный»
	Установка элемента «Экспонента»
	Установка элемента «Квадратный корень»
	Установка элемента «Абсолютное значение»
	Установка элемента «Показательная функция»
	Установка элемента «Полином»
	Установка элемента «Константа»
	Установка элемента «Датчик случайных чисел»
	Установка элемента «Интегратор»
	Установка элемента «Дифференцирование»
	Установка элемента «Линейная модель»
	Установка элемента «Запаздывание»
	Установка элемента «Передаточная функция»
	Установка элемента «Знаковая функция»
	Установка элемента «Нечувствительность»
	Установка элемента «Ограничение»
	Установка элемента «Нечувствительность с ограничением»
	Установка элемента «Скачок»
	Установка элемента «Квантование»
	Установка элемента «Табличная функция»
	Установка элемента «Люфт»
	Установка элемента «Дискретная позиция»
	Установка элемента «Дискретный переход»
	Установка элемента «Непрерывная позиция»
	Установка элемента «Непрерывный переход»
	Установка элемента «График сигнала»

Продолжение табл. Б.2

<i>Пиктограмма</i>	<i>Значение</i>
	<i>Установка элемента «Таблица сигнала»</i>
	<i>Установка элемента «Оптимизатор»</i>
	<i>Установка элемента «Перцептрон»</i>
	<i>Установка элемента «Линейный порог»</i>
	<i>Установка элемента «Гиперболический сигмоид»</i>
	<i>Установка элемента «Логарифмический сигмоид»</i>
	<i>Установка элемента графического редактора «Линия»</i>
	<i>Установка элемента графического редактора «Прямоугольник»</i>
	<i>Установка элемента графического редактора «Прямоугольник с заливкой»</i>
	<i>Установка элемента графического редактора «Скругленный прямоугольник»</i>
	<i>Установка элемента графического редактора «Скругленный прямоугольник с заливкой»</i>
	<i>Установка элемента графического редактора «Эллипс»</i>
	<i>Установка элемента графического редактора «Эллипс с заливкой»</i>
	<i>Установка элемента графического редактора «Текст»</i>
	<i>Управление настройками графического редактора</i>

Б.8 Режим задания связей

Режим задания связей позволяет ввести информацию о топологии исследуемой модели. Для перехода в режим задания связей необходимо нажать кнопку панели инструментов, изображающую карандаш, либо воспользоваться пунктом *Линия* в подменю *Ввод* главного меню.

При этом курсор меняет свое изображение на изображение не закрашенного карандаша, и работа осуществляется следующим образом. При нажатии на левую клавишу мыши курсор, изменяет свой вид на изображение закрашенного карандаша

и от точки, где было сделано нажатие, за ним начинает тянуться линия. Для того чтобы повернуть линию необходимо снова нажать на левую клавишу мыши, а для того чтобы завершить рисование линии необходимо сделать быстро два нажатия на левую клавишу (*Double Click*). При этом правая клавиша мыши во время проведения линии, т.е. когда курсор имеет форму закрашенного карандаша, осуществляет отмену рисования текущей линии (карандаш становится не закрашенным), а когда линия не проводится, также как и при установке элементов, возвращает пользователя в режим корректировки схемы.

Б.9 Режим работы с группой выделенных элементов

Режим корректировки схемы предназначен для изменения параметров элементов системы, удаления блоков и элементов, а также их объединения в группы. Курсор в этом режиме имеет вид стандартной стрелки. Переход к режиму редактирования схемы, из любого другого режима, осуществляется нажатием правой клавиши мыши либо клавиши *Esc*.

Клавиши мыши в данном режиме имеют следующее назначение:

1. Однократное нажатие левой клавиши мыши, на линии либо элементе приводит к их выделению цветом, либо к снятию выделения, если линия либо элемент были выбраны ранее. Если при этом удерживается клавиша *Ctrl*, то данный элемент добавляется к тому, что было выбрано ранее, в противном случае прежнее выделение снимается. Заметим, что при нажатии клавиши мыши на линии выделяются все линии входящие в данную связь. Для выделения одиночной линии необходимо удерживать нажатой клавишу *Shift*.
2. Двойное нажатие левой клавиши на элементе приводит к выводу диалога редактирования параметров данного элемента.

Выделение можно осуществить при помощи команд **Выделить Все**, **Выделить Линии**, **Выделить Элементы** подменю **Правка**. Эти команды доступны только если есть элементы или связи на схеме. Снять выделение, если

такое имеется возможно при помощи команды **Развыделить Все** подменю **Правка**.

После того как элемент или связь (или их группа) будут выделены, с ними можно осуществить следующие действия.

- **Перемещение.** Для перемещения элемента, связи или группы элементов необходимо нажать на нем левую клавишу мыши и, не отпуская ее, осуществить перемещение выделенного на новое место, после чего отпустить клавишу мыши. Если при этом удерживать клавишу **Ctrl** то выбранные элементы будут скопированы в новое место. Если во время этой операции нажать клавишу **Esc** , то она будет отменена.
- **Отмена.** Для отмены действия, повлекшего изменения в топологии схемы модели или параметрах одного из ее элементов необходимо нажать на клавиши **Alt+BackSpace**, либо воспользоваться командой **Отмена** подменю **Правка**.
- **Копирование в буфер обмена.** Для помещения выделенного блока в буфер обмена необходимо либо нажать клавиши **Ctrl+Insert** , либо воспользоваться командой **Копирование** подменю **Правка**. Заметим, что при копировании в буфер обмена, параметры блоков сохраняются, но не сохраняются их номера.
- **Удаление в буфер обмена.** Для перемещения выделенного блока в буфер обмена необходимо либо нажать клавишу **Delete**, либо воспользоваться командой **Вырезать** подменю **Правка**.
- **Вставка из буфера обмена.** Для осуществления вставки в редактор содержимого буфера необходимо либо нажать клавиши **Shift+Insert** , либо воспользоваться командой **Вставка** меню **Правка**. После этого вставляемые элементы необходимо переместить к месту их нового расположения и нажать левую клавишу мыши для подтверждения операции. Нажатие левой клавиши отменяет действия по вставке.
- **Экспорт в графический файл.** Для осуществления экспорта выделенного фрагмента в графический файл необходимо воспользоваться командой **Копировать в...** меню **Правка** . После этого появится диалоговое окно, в

котором необходимо задать расположение, имя и тип требуемого графического файла, после чего нажать на кнопку **Сохранить**, либо нажать клавишу *Enter*. Нажатие на кнопку **Отмена** либо на клавишу *Esc* отменяет экспорт. Изображение экспортируется в файл с текущим увеличением, сетка и разбиение на страницы не отображается. Сейчас поддерживаются следующие типы файлов: **Enhanced MetaFile (*.EMF)**, **Windows MetaFile (*.WMF)** и **Windows Bitmap (*.BMP)**.

- В данном режиме возможна также перенумерация элементов. Необходимо воспользоваться командой **Перенумерация** элементов через меню **Правка**. Все элементы схемы после выполнения данной команды будут перенумерованы. **Номера** элементам присваиваются последовательно в порядке их установки на схеме.
- **Перемещение блоков.** Для перемещения "**блока**" вам необходимо выполнить следующие действия.
 1. Выделить элементы и связи, которые вы желаете переместить. Это можно сделать как в обычном режиме редактирования, так и в режиме работы с блоками, однако необходимо отметить, что в данном режиме отсутствует возможность развыделения блока.
 2. Активировать режим перемещения блоков нажатием на кнопку **Работа с блоками** панели инструментов.
 3. Установить указатель мыши на выделенный элемент или связь, нажать левую клавишу мыши и потащить этот элемент или связь, все другие выделенные элементы и связи будут перемещаться вместе с ним. При этом курсор примет вид стрелки, направленной вверх, это означает, что текущее выделение рассматривается как "**блок**" и перемещение его может быть отменено нажатием на правую клавишу мыши.
 4. Разместите "**блок**" на выбранное место. Нажмите на левую клавишу мыши вне выделенного элемента или связи для того чтобы зафиксировать "**блок**". Указатель мыши примет обычный вид, показывая, что "**блок**" зафиксирован, и перемещение не может быть отменено.

Указатель мыши в виде стрелки, направленной вверх говорит о том, что имеется незафиксированный "блок", перемещение которого может быть отменено нажатием на правую клавишу мыши. Нажатие же на кнопку **Работа с блоками** панели инструментов приведёт к выходу из этого режима с фиксацией "блока".

Нажатие на правую клавишу мыши, когда курсор имеет обычный вид, приводит к выходу из данного режима.

Б.10 Ввод и редактирование информации о параметрах элемента.

Ввод и редактирование информации о параметрах элемента (Общие параметры) осуществляется с помощью диалогового окна ввода параметров элемента, вид которого приведен на рисунке Б.7.

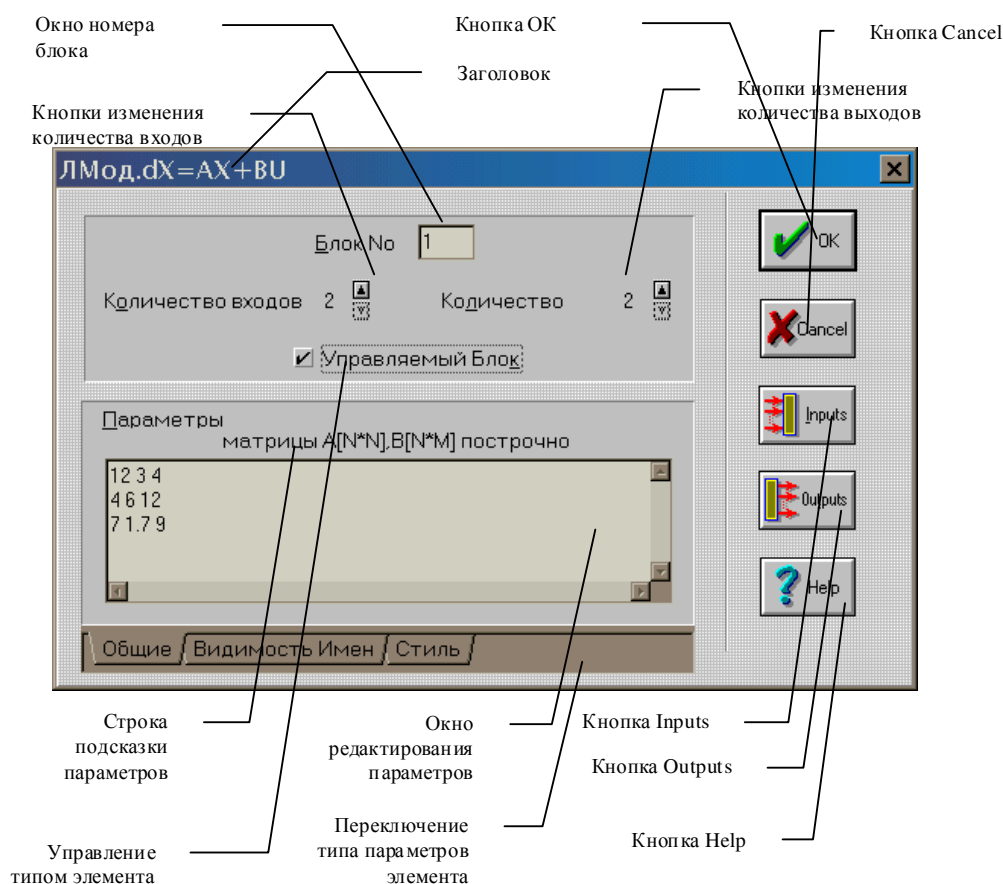


Рисунок Б.7 – Диалоговое окно редактирования параметров элемента (вкладка **Общие параметры**)

Окно ввода параметров элемента имеет следующие элементы:

- **Заголовок** – содержит информацию о наименовании элемента.
- **Окно номера элемента** – для редактирования номера элемента на схеме.
- **Кнопки изменения количества входов** – при нажатии на нижнюю уменьшает количество входов элемента на 1; на верхнюю – увеличивает количество входов на 1 (максимум может быть введено 10 входов для обычного элемента или 11 – для управляемого (11-ый – управление), минимум соответственно – 0 и 1 (управление)).
- **Кнопки изменения количества выходов** – при нажатии на нижнюю уменьшает количество выходов элемента на 1; на верхнюю – увеличивает количество выходов на 1 (максимум может быть введено 10 выходов для обычного элемента или 11 – для управляемого (11-ый – управление), минимум соответственно – 0 и 1 (управление)).
- **Управление типом элемента (управляемый / неуправляемый)** – имеет двойное назначение. При возможности изменения состояния: при включении увеличивает количество (минимальное, максимальное и текущее) входов и выходов и соответствует управляемому элементу; при выключении – обратная процедура; при невозможности изменения состояния – сигнализирует о характере редактируемого элемента: при отмеченном состоянии – элемент дискретный, иначе – непрерывный.
- **Строка подсказки параметров** – содержит краткую информацию о количестве, формате и характере параметров элемента.
- **Окно параметров** – содержит редактируемые параметры.
- **Переключение типа параметров элемента** – позволяет переключаться между редактированием различных типов параметров элемента. Выше приведен вид диалогового окна вкладки **Общие**. Остальные вкладки рассмотрены далее. (Диалоговое окно на вкладке **Видимость Имен** содержит элементы, позволяющие управлять отображением элемента на схеме, а также редактировать имя элемента. Подробное описание данной вкладки см. Ввод и редактирование

информации о параметрах элемента (Видимость Имен); Диалоговое окно на вкладке **Стиль** содержит элементы, позволяющие управлять поведением (стилем) элемента. Подробное описание данной вкладки см. **Ввод и редактирование информации о параметрах элемента (Стиль)**).

В поле "*номер элемента*" задается обязательный параметр – номер элемента на структурной схеме. Вводимые элементы нумеруются в произвольном порядке, однако, эти номера не должны совпадать у разных элементов. Возможна дальнейшая перенумерация элементов по команде **Перенумерация элементов** подменю **Правка**.

Номер элемента при включенном и выключенном режиме **Нумерация элементов как в ДН-сетях** может отличаться. Редактируемый в данный момент номер относится к тому режиму нумерации и типу элементов, которые выбраны в данный момент.

По умолчанию, на экран выдаются последовательно возрастающие номера элементов. Для внешних подмоделей номера элементов могут начинаться с символа "*" (временный номер), чтобы избежать совпадения элементов на набираемой (основной) структурной схеме.

В поле параметров задаются числовые параметры элемента. В качестве параметров могут использоваться любые десятичные константы, в том числе с десятичной точкой и знаком десятичной экспоненты *e* или *E*.

Кроме того, в поле параметров могут появляться ссылки на варьируемые параметры вида ***<номер вар.параметра>** или **<коэффициент>*<номер вар.параметра>**. Появление такой ссылки в поле параметров означает, что вместо числового постоянного параметра должно быть подставлено значение варьируемого параметра (или это значение, умноженное на коэффициент).

Пример:

***2 1.57*3 (V2 1.57*V3)**

Разделителем между параметрами могут быть один или несколько пробелов или конец строки.

Число заданных параметров должно быть равно числу параметров элемента, если это фиксировано для данного типа элемента и контролируется пользователем, если это число не определено при генерации элемента в системе.

Кроме этого, числовые постоянные параметры элемента могут быть заранее подготовлены в текстовом файле и введены в модель заданием в поле параметров имени файла в виде '*имя файла*'.

Для некоторых элементов в поле параметров допускается запись формул.

Ввод и редактирование информации о параметрах элемента (управление видимостью элементов элемента, а также имен) осуществляется с помощью диалогового окна ввода параметров элемента, вид которого приведен на рисунке Б.8.

Диалоговое окно на вкладке **Видимость Имен** содержит элементы, позволяющие управлять отображением элемента на схеме, а также редактировать имя элемента:

- **Номер элемента** – показывает текущий номер элемента в таком виде, как он будет отображаться на схеме, т.е., в зависимости от текущего режима нумерации элементов, либо с разделением на типы и указанием типа, верхнего и нижнего индексов, или только номером элемента.
- **Видимость номера элемента** – позволяет управлять видимостью номера элемента на схеме: если на нем имеется отметка, то номер элемента будет отображаться на схеме, в противном случае информация о номере элемента не выводится.
- **Смещение номера элемента по X** – позволяет управлять горизонтальным положением номера элемента, задавая его смещение по горизонтали относительно нормального положения – сверху над элементом, прижатом к левому краю. Положительное значение соответствует сдвигу его вправо на заданное количество точек с привязкой по левому краю номера. Этот параметр не влияет на отрисовку элемента, если параметр **Видимость номера элемента** выключен.

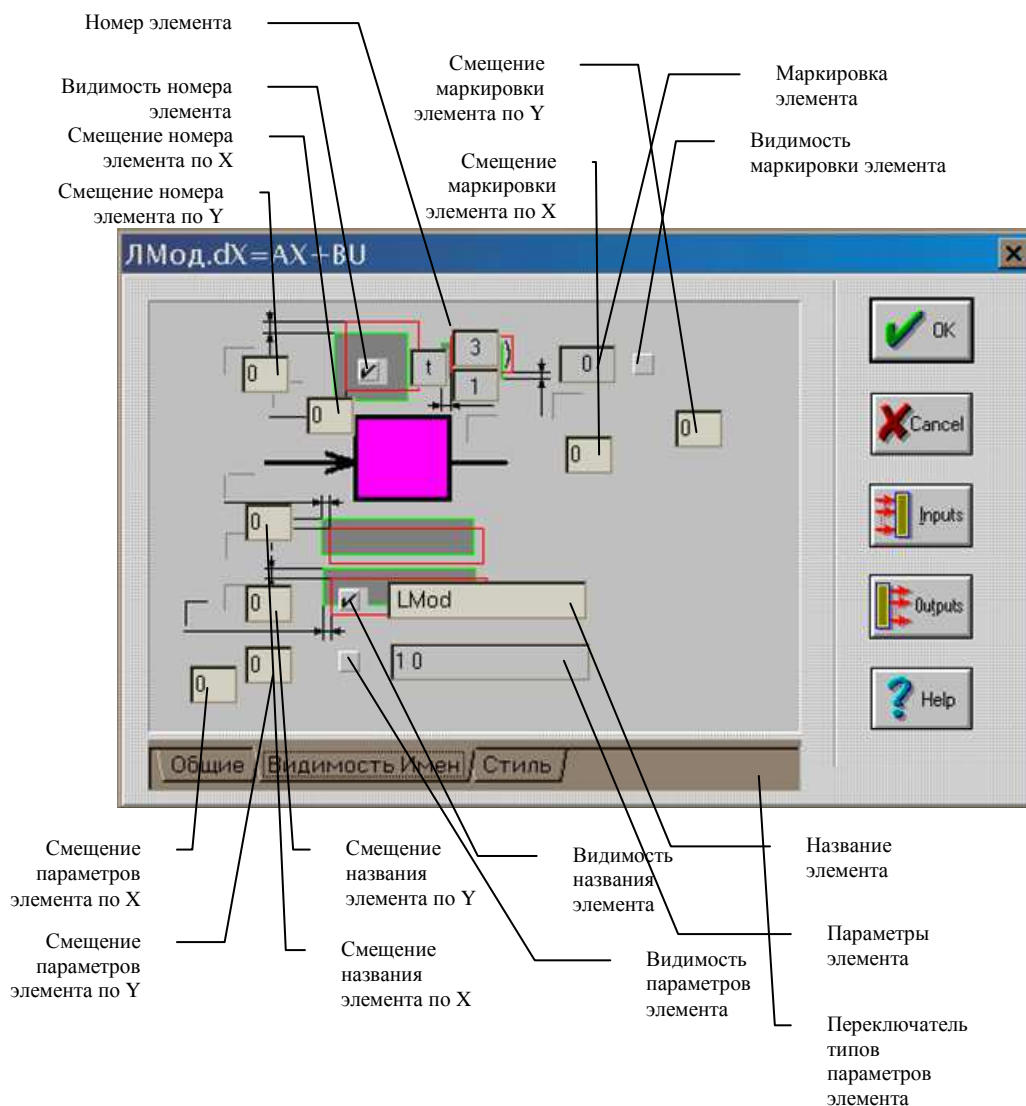


Рисунок Б.8 – Диалоговое окно редактирования параметров элемента (Видимость Имен)

- **Смещение номера элемента по Y** – позволяет управлять вертикальным положением номера элемента, задавая его смещение по вертикали относительно нормального положения – сверху над элементом, прижатом к левому краю. Положительное значение соответствует сдвигу его вверх на заданное количество точек с привязкой по центру типа номера. Этот параметр не влияет на отрисовку элемента если параметр **Видимость номера элемента** выключен.
- **Маркировка элемента** – показывает текущую маркировку элемента в целочисленном виде. Данный параметр имеет значение только для элемента – дискретной позиции.

- **Видимость маркировки элемента** – позволяет управлять видимостью текущей маркировки элемента в целочисленном виде в режиме коррекции разметки сети: если на нем имеется отметка, то текущая маркировка элемента в целочисленном виде будет отображаться на схеме, в противном случае информация о текущей маркировке элемента не выводится (показывается только есть ли маркеры в позиции или нет). Данный параметр имеет значение только для элемента – дискретной позиции.
- **Смещение маркировки элемента по X** – позволяет управлять горизонтальным положением текущей маркировки элемента в целочисленном виде в режиме коррекции разметки сети, задавая ее смещение по горизонтали относительно нормального положения – сверху над элементом, справа от номера элемента. Положительное значение соответствует сдвигу ее вправо на заданное количество точек с привязкой по левому краю. Этот параметр не влияет на отрисовку элемента в режиме коррекции разметки сети, если параметр **Видимость маркировки элемента** выключен. Данный параметр имеет значение только для элемента – дискретной позиции.
- **Смещение маркировки элемента по Y** – позволяет управлять вертикальным положением текущей маркировки элемента в целочисленном виде в режиме коррекции разметки сети, задавая ее смещение по вертикали относительно нормального положения – сверху над элементом, справа от номера элемента. Положительное значение соответствует сдвигу ее вверх на заданное количество точек с привязкой по центру. Этот параметр не влияет на отрисовку элемента в режиме коррекции разметки сети, если параметр **Видимость маркировки элемента** выключен. Данный параметр имеет значение только для элемента – дискретной позиции.
- **Параметры элемента** – показывает текущие параметры элемента, заданные в **Окно параметров** на вкладке **Общие**.
- **Видимость параметров элемента** – позволяет управлять видимостью параметров элемента на схеме: если на нем имеется отметка, то параметры

элемента (если таковые имеются) будет отображаться на схеме, в противном случае информация о параметрах элемента не выводится.

- **Смещение параметров элемента по X** – позволяет управлять горизонтальным положением параметров элемента, задавая их смещение по горизонтали относительно нормального положения – снизу под элементом, под названием элемента, прижатыми к левому краю. Положительное значение соответствует сдвигу их вправо на заданное количество точек с привязкой по левому краю параметров. Этот параметр не влияет на отрисовку элемента если параметр **Видимость параметров элемента** выключен или если параметры у элемента отсутствуют.
- **Смещение параметров элемента по Y** – позволяет управлять вертикальным положением параметров элемента, задавая их смещение по вертикали относительно их нормального положения – снизу под элементом, под названием элемента, прижатыми к левому краю. Положительное значение соответствует сдвигу их вниз на заданное количество точек с привязкой по верхнему краю параметров. Этот параметр не влияет на отрисовку элемента, если параметр **Видимость параметров элемента** выключен или если параметры у элемента отсутствуют.
- **Название элемента** – позволяет редактировать текущее название элемента, отображаемое на схеме, задавая которое можно искать элемент при помощи команды **Поиск элемента**. Лидирующие пробелы могут игнорироваться.
- **Видимость названия элемента** – позволяет управлять видимостью названия элемента на схеме: если на нем имеется отметка, то название элемента будет отображаться на схеме, в противном случае информация о названии элемента не выводится.
- **Смещение названия элемента по X** – позволяет управлять горизонтальным положением названия элемента, задавая его смещение по горизонтали относительно нормального положения – снизу под элементом, прижатым к левому краю. Положительное значение соответствует сдвигу его вправо на заданное количество точек с привязкой по левому краю названия. Этот параметр

не влияет на отрисовку элемента, если параметр **Видимость названия элемента выключен**.

- **Смещение названия элемента по Y** – позволяет управлять вертикальным положением названия элемента, задавая его смещение по вертикали относительно их нормального положения – снизу под элементом, прижатыми к левому краю. Положительное значение соответствует сдвигу их вниз на заданное количество точек с привязкой по верхнему краю названия элемента. Этот параметр не влияет на отрисовку элемента, если параметр **Видимость названия элемента выключен**.
- **Переключение типа параметров элемента** – позволяет переключаться между редактированием различных типов параметров элемента. Выше приведен вид диалогового окна вкладки **Видимость Имен**. Остальные вкладки рассмотрены далее.

Ввод и редактирование информации о параметрах элемента (управление видимостью элементов элемента, а также имен) осуществляется с помощью диалогового окна ввода параметров элемента, вид которого приведен на рисунке Б.9.

Диалоговое окно на вкладке **Стиль** содержит элементы, позволяющие управлять поведением (стилем) элемента. Доступность параметров для изменения зависит от конкретного элемента. Имеются следующие параметры:

- **Непрерывный** – показывает, что данный элемент является непрерывным элементом и будет соответственно отображаться на схеме и моделироваться. Имеет взаимосвязь с параметрами **Дискретный** и **Комплексный** – в один момент времени может быть установлен только один из них.
- **Дискретный** – показывает, что данный элемент является дискретным элементом, и будет соответственно отображаться на схеме и моделироваться. Имеет взаимосвязь с параметрами **Непрерывный** и **Комплексный** – в один момент времени может быть установлен только один из них.

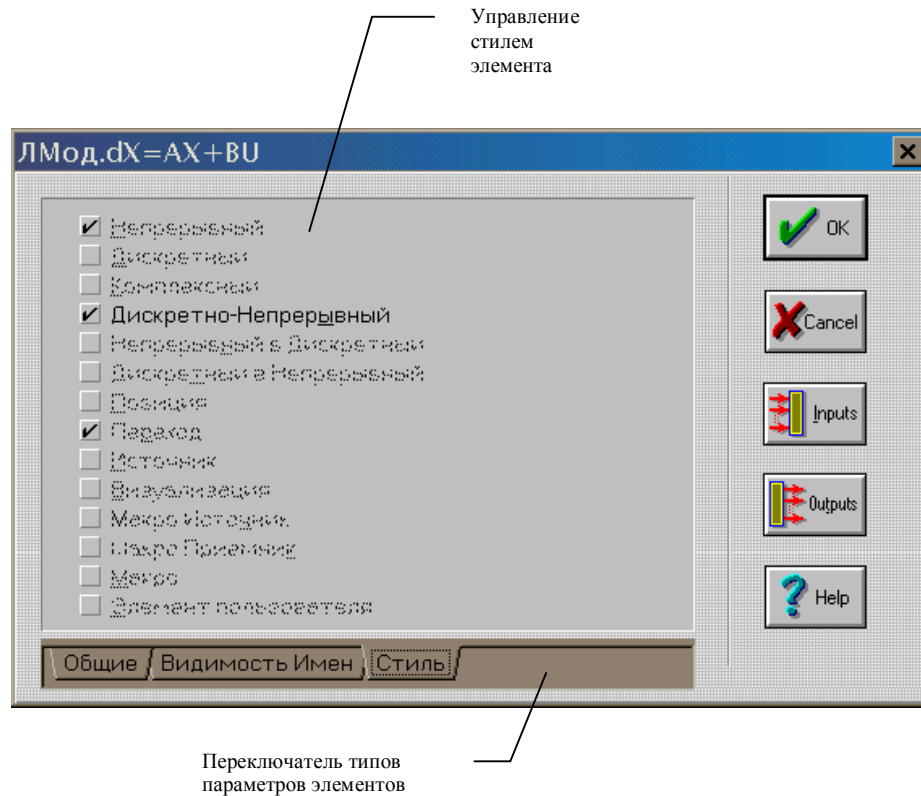


Рисунок Б.9 – Диалоговое окно редактирования параметров элемента (вкладка Стиль)

- **Комплексный** – показывает, что данный элемент является комплексным элементом (смешанным – непрерывным и дискретным) и будет соответственно отображаться на схеме и моделироваться. Имеет взаимосвязь с параметрами **Непрерывный** и **Дискретный** – в один момент времени может быть установлен только один из них.
- **Дискретно-непрерывный** – позволяет управлять типом непрерывного элемента (управляемый/неуправляемый). Если элемент может быть управляемым, то установка или снятие данного флага вызовет соответственно установку или снятие состояния **Управление типом элемента** на вкладке **Общие** и наоборот. Если изменение состояния на вкладке **Общие** параметра **Управление типом элемента** невозможно, то также невозможна установка данного параметра.
- **Непрерывный в дискретный** – показывает, что данный элемент является преобразователем сигнала из непрерывного в дискретный и будет соответственно

отображаться на схеме и моделироваться. Имеет взаимосвязь с параметрами **Дискретный в непрерывный** – в один момент времени может быть установлен только один из них.

- **Дискретный в непрерывный** – показывает, что данный элемент является преобразователем сигнала из дискретного в непрерывный и будет соответственно отображаться на схеме и моделироваться. Имеет взаимосвязь с параметрами **Непрерывный в дискретный** – в один момент времени может быть установлен только один из них.
- **Позиция** – показывает, что данный элемент является позицией и будет соответственно отображаться на схеме и моделироваться. Имеет взаимосвязь с параметром **Переход** – в один момент времени может быть установлен только один из них.
- **Переход** – показывает, что данный элемент является позицией и будет соответственно отображаться на схеме и моделироваться. Имеет взаимосвязь с параметром **Позиция** – в один момент времени может быть установлен только один из них.
- **Источник** – показывает, что данный элемент является источником сигнала и будет соответственно отображаться на схеме и моделироваться.
- **Визуализация** – показывает, что данный элемент является выводом сигнала на отображение (например, на графике) и будет соответственно отображаться на схеме и моделироваться.
- **Макро источник** – показывает, что данный элемент является источником сигнала внутри макроэлемента, на его выходах присутствуют сигналы, поданные на соответствующие входы макроперехода.
- **Макро приемник** – показывает, что данный элемент является приемником сигнала внутри макроэлемента, на его входы подаются сигналы, которые необходимо вывести на соответствующие выходы макроперехода.
- **Макро** – показывает, что данный элемент является макроэлементом, и может быть отредактирован в новом окне схемы (подсистема).

- **Элемент пользователя** – показывает, что данный элемент является элементом, поведение которого полностью определяется пользователем, т.е. данный элемент разрабатывается самим пользователем и может иметь нестандартный алгоритм функционирования (обычно сохраняется в дополнительной библиотеке).
- **Переключение типа параметров элемента** – позволяет переключаться между редактированием различных типов параметров элемента. Выше приведен вид диалогового окна вкладки *Стиль*. Остальные вкладки рассмотрены далее. (Диалоговое окно на вкладке *Общие содержит элементы*, позволяющие редактировать общие параметры элемента. Подробное описание данной вкладки см. Ввод и редактирование информации о параметрах элемента; Диалоговое окно на вкладке *Видимость Имен* содержит элементы, позволяющие управлять отображением элемента на схеме, а также редактировать имя элемента. Подробное описание данной вкладки см. **Ввод и редактирование информации о параметрах элемента (Видимость Имен)**).

Окно также содержит следующие кнопки:

- **Кнопка *Inputs*** – нажатие приводит к вызову окна редактирования параметров входов. Также выбор команды может быть осуществлён путём нажатия на *Alt-I*.
- **Кнопка *Outputs*** – нажатие приводит к вызову окна редактирования параметров выходов. Также выбор команды может быть осуществлён путём нажатия на *Alt-T*.

Ввод и редактирование информации о входах и выходах элемента осуществляется с помощью диалоговых окон ввода параметров входов и выходов элемента, вид которых приведен на рисунках Б.10 и Б.11.

Оба этих окна имеют следующие одинаковые поля по каждому из входов / выходов.

- **Тип** – содержит информацию о типе входа / выхода (непрерывный / дискретный / дискретный ингибиторный).

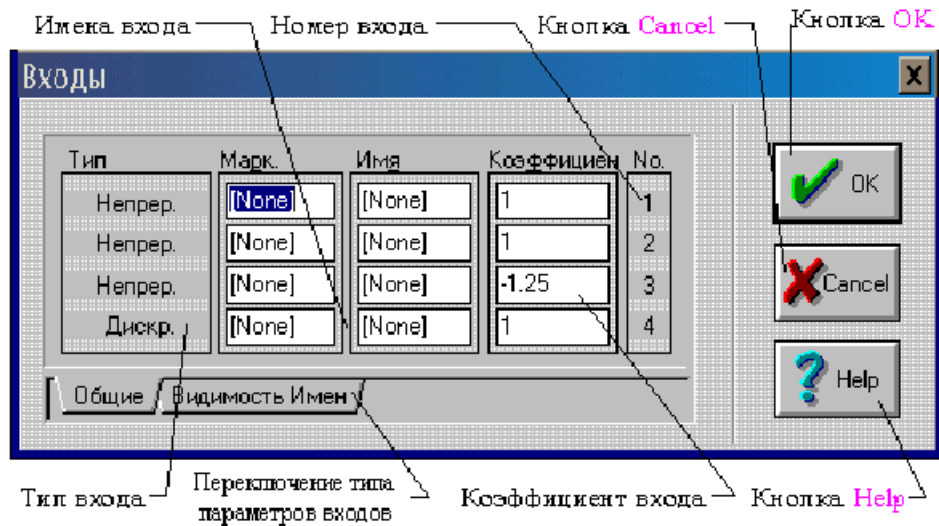


Рисунок Б.10 – Диалоговое окно редактирования параметров входов (вкладка Общие)

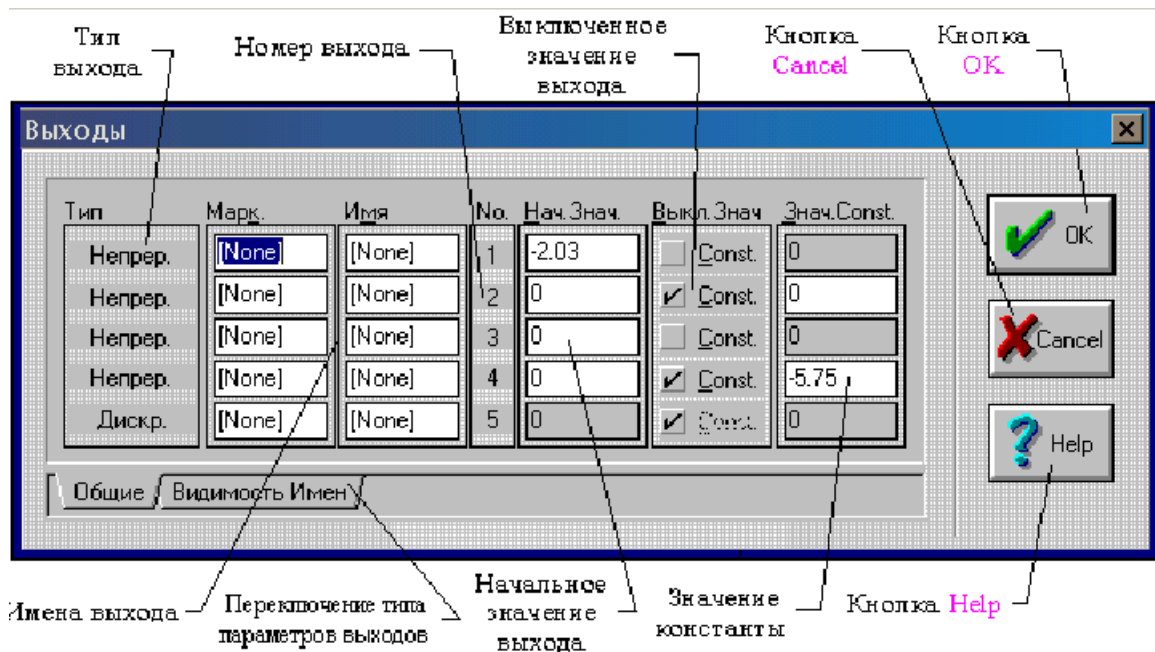


Рисунок Б.11 – Диалоговое окно редактирования параметров выходов (вкладка Общие)

- **Имена** – содержит информацию о именах входа / выхода (сейчас используются только для дополнительного контроля правильности набора схемы: если у соединённых входа и выхода различны имена, то при сборки модели генерируется предупреждение). По умолчанию заполняются значениями [None] / [None] (нет имени).

- **Номер** – содержит информацию о номере входа / выхода в пределах данного элемента (сверху вниз, т.е. верхний вывод элемента на схеме считается первым).
- **Переключение типа параметров входа (выхода)** – позволяет переключаться между редактированием различных типов параметров элемента. Выше приведен вид диалогового окна вкладки **Общие**. Остальные вкладки рассмотрены далее. (Диалоговое окно на вкладке **Видимость Имен** содержит элементы, позволяющие управлять отображением входов / выходов элемента на схеме. Подробное описание данной вкладки см. **Ввод и редактирование информации о входах и выходах элемента (Видимость Имен)**).

Следующие поля различны:

- **Коэффициент** – (для входов) содержит информацию о коэффициенте связи (по умолчанию 1).
- **Начальное значение** – (для выходов) содержит информацию о начальном значении выхода (по умолчанию 0).
- **Выкл. значение** – (для выходов) содержит информацию о поведении выхода при выключении элемента. При отмеченном состоянии – на выходе элемента при выключении поддерживается значение константы, иначе – выход запоминает своё последнее значение (по умолчанию не отмечено – запоминание). Доступен только для выходов управляемых элементов.
- **Значение константы** – (для выходов) содержит информацию о значении поддерживаемой на выходе элемента постоянной величине при выключении элемента (по умолчанию 0). Доступен только для выходов управляемых элементов, для которых установлен параметр "**Выкл. константа**".

Б.11 Режим коррекции разметки сети

Активизация режима коррекции разметки сети осуществляется при нажатии на кнопку *Режим коррекции разметки сети* панели инструментов с помощью мыши либо при выборе соответствующего пункта меню.

При активизации данного режима из режима редактирования схемы происходит проверка схемы модели на наличие в ней ошибок и при обнаружении таковых выдаются соответствующие сообщения, и активизации режима не происходит. Предупреждения также выдаются при несовпадении имён подсоединённых друг к другу входов и выходов, а также их типов (непрерывный или дискретный). При этом возможно проигрывание звука, ассоциированного с критическим событием в системе (при ошибке) или запросом системы (при предупреждении) в зависимости от текущих настроек в *Общих настройках* программы. Необходимо отметить, что одним из необходимых условий для моделирования схемы является подключённость всех входов элементов к выходам, и при обнаружении такой ошибки курсор мыши устанавливается в местонахождения элемента или связи (окно схемы открывается, если оно было закрыто). При удачной активизации режима выдаётся звуковой сигнал, а на схеме появляется разметка сети (с указанием количества маркеров в позиции для дискретных позиций).

Для коррекции разметки необходимо в режиме коррекции разметки сети "*щелкнуть*" по позиции левой клавишей мыши для установки маркера в выбранную позицию, или правой клавишей мыши – для изъятия маркера из позиции.

При двойном "*щелчке*" левой клавишей мыши на дискретной позиции вызывается диалоговое окно (см. рис. Б.12), в котором необходимо задать целое число – количество маркеров в позиции.

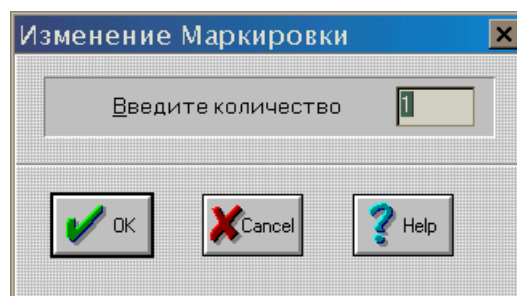


Рисунок Б.12 – Диалоговое окно изменения количества маркеров дискретной позиции

Имеющаяся проверка не позволяет вам задать слишком много маркеров или какую-либо ошибочную информацию.

Б.12 Моделирование

Для того чтобы промоделировать модель вам необходимо открыть имеющийся файл или создать новый в режиме редактирования. Затем запустить режим моделирования, исправив обнаруженные ошибки.

Рекомендуется перед запуском на моделирование сохранить версию модели в файле.

Активизация режима моделирования осуществляется при:

- нажатии на кнопку **Режим моделирования** панели инструментов с помощью мыши;
- введении команд **Запуск моделирования** (с помощью клавиши **F9** или выбора команды **Запуск** меню **Моделирование**) или **Шаг моделирования** (с помощью клавиши **F7** или выбора команды **Шаг** меню **Моделирование**).

В двух последних случаях помимо активизации режима моделирования происходит ещё и выполнение требуемой команды.

При активизации данного режима из режима редактирования схемы происходит проверка схемы модели на наличие в ней ошибок и при обнаружении таковых выдаются соответствующие сообщения и активизации режима не происходит. Предупреждения также выдаются при несовпадении имён подсоединённых друг к другу входов и выходов, а также их типов (непрерывный или дискретный) (Если соответствующие параметры заданы в **Общих настройках** программы). При этом возможно проигрывание звука, ассоциированного с критическим событием в системе (при ошибке) или запросом системы (при предупреждении) в зависимости от текущих настроек в **Общих настройках** программы. Необходимо отметить, что одним из необходимых условий для моделирования схемы является подключённость всех входов элементов к выходам, и при обнаружении такой ошибки курсор мыши устанавливается в

местонахождения элемента или связи (окно схемы открывается, если оно было закрыто). При удачной активизации режима становятся доступными кнопки управлением режима на панели инструментов, и выдаётся звуковой сигнал, а на схеме появляется разметка сети.

В режиме моделирования в состоянии паузы или в состоянии останова можно переключаться в режим редактирования разметки или редактирования модели. При этом если было состояние паузы и при переключении в режим редактирования модели не было сделано никаких изменений в ней, то моделирование может быть продолжено. В противном случае будет зафиксировано состояние ***Останов моделирования*** и моделирование начнётся сначала с запросом параметров моделирования. Состояние ***Останов_моделирования*** можно также активизировать нажатием кнопки **Останов моделирования** на панели инструментов с помощью мыши; нажатием клавиши **F4**; либо выбором команды **Стоп** меню **Моделирование**.

При окончании моделирования по достижении конечной точки заданного временного интервала выдаётся звуковой сигнал, и программа переходит в состояние ожидания, при этом сохраняется последняя разметка сети, и следующий запуск моделирования произойдёт с данными параметрами разметки сети. Для сброса модели в исходное состояние необходимо задать команду **Останов моделирования**.

В процессе моделирования модели (состояния ***Шаг*** или ***Пуск моделирования***) имеется возможность только получения справки, работы с окнами и выдача команд **Останов моделирования** или **Пауза моделирования**. Попытка в этих состояниях закрыть основное окно программы приведёт к выдаче команды **Останов моделирования**, и окно останется открытым.

Параметры моделирования задают ***начальное время моделирования, конечное время, шаг интегрирования, шаг фиксации промежуточных результатов (для выдачи), метод интегрирования (Эйлера или Рунге-Кутты IV)***.

Для варьируемых параметров задают их начальные значения (через пробел). Порядок их следования должен соответствовать использованию в структурной схеме.

Б.13 Анализ свойства достижимости

Анализ исследуемой системы на достижимость проводится путем запуск соответствующей команды из меню **Сервис** при нахождении среды в **режиме коррекции разметки**. При этом открывается окно анализа свойства достижимости, представленное на рис. Б.13. При этом слева отображается таблица, содержащая матрицу инцидентности исследуемой ДН-сети, либо накопленные вектора маркировок. Перед запуском анализа необходимо убедиться в правильности начальной разметки ДН-сети и произвести ее коррекцию в случае необходимости. Тип необходимого анализа выбирается при помощи переключателя **Режим**, обеспечивающего выбор следующих режимов.

- **Аналитический** – в данном режиме проводится анализ свойства достижимости при помощи аналитического метода. При этом проводится редукция-декомпозиция с отображением на схеме исследуемой ДН-сети сокращаемых переходов и позиций с помощью линии, перечеркивающей изображение данной вершины. При этом таблица отображает текущую матрицу инцидентности (рис. Б.14). Результат работы в аналитическом режиме составляет конечную матрицу инцидентности, по которой следует заключение о достижимости системы. Также данный вывод может быть получен если остались неотмеченными вершины, составляющие элементарную ДН-сеть.
- **Маркерный** – анализ свойства достижимости проводится при помощи моделирования в режиме **моделирования**. Возможны два способа работы. В первом из них после нажатия кнопки **Пуск** необходимо задать конечную маркировку на схеме в **режиме коррекции разметки** либо в окне конечной маркировки (рис. Б.15). Запуск анализа производится после повторного нажатия кнопки **Пуск**. В данном случае будет выведено сообщение о достижении конечной маркировки. Если конечную маркировку не задавать и сразу нажать кнопку **Пуск** повторно, то сообщения выдано не будет. В данном режиме таблица отображает накопленные вектора маркировок.

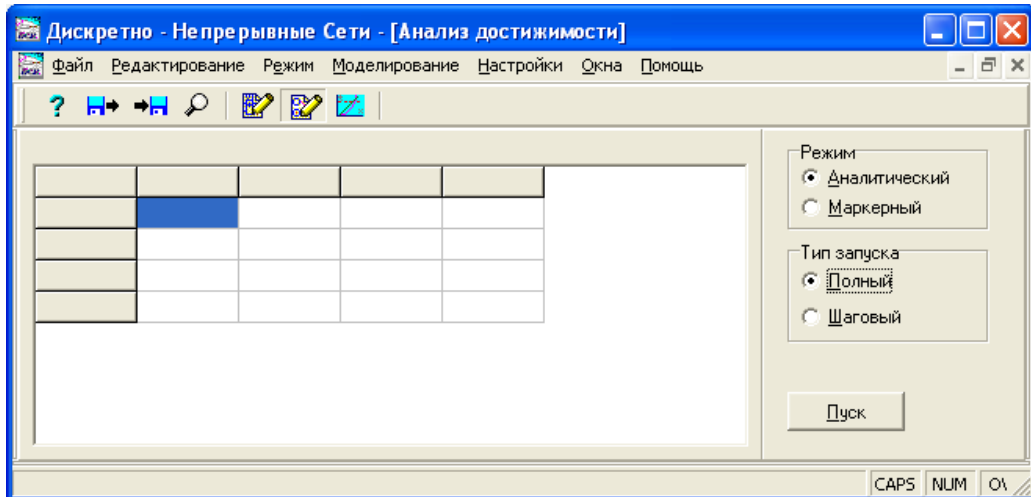


Рисунок Б.13 – Исходное состояние окна анализа свойства достижимости

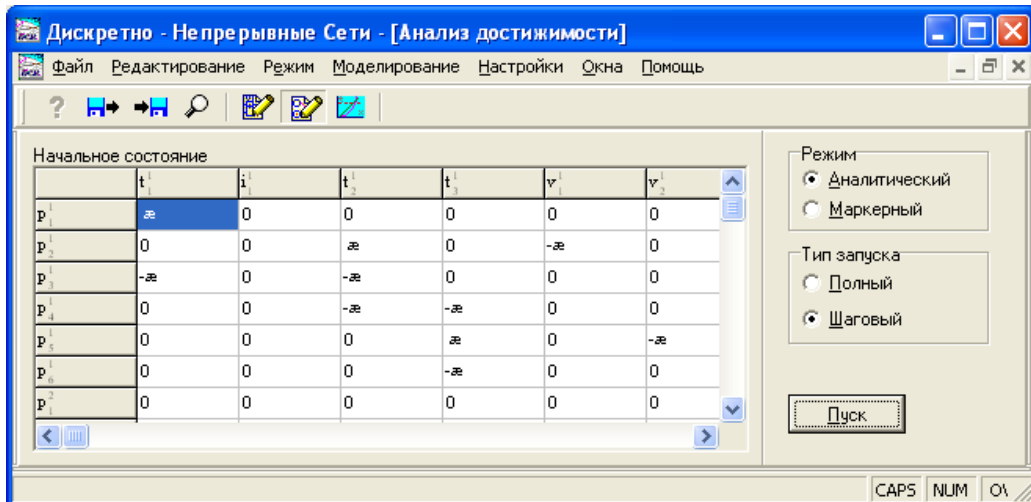


Рисунок Б.14 – Аналитический режим анализа свойства достижимости

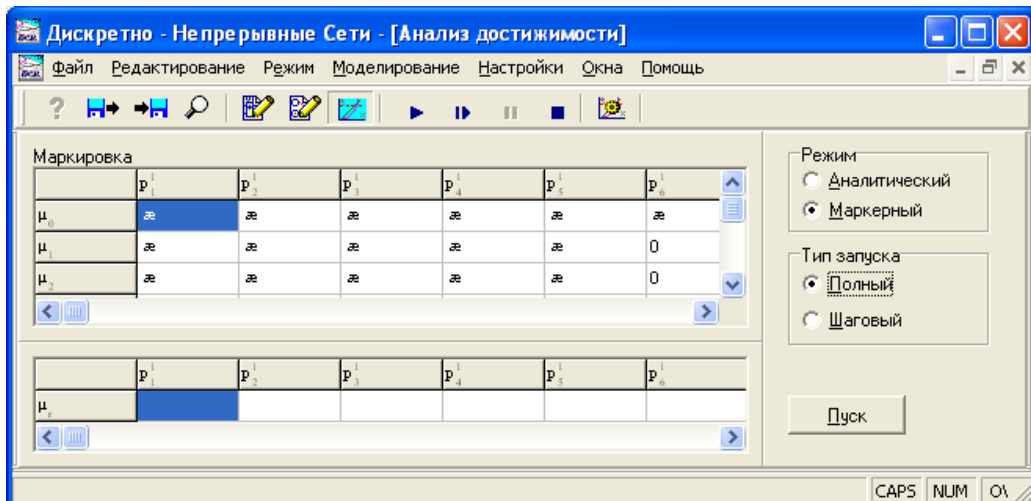


Рисунок Б.15 – Маркерный режим анализа свойства достижимости

Переключатель **Тип запуска** определяет будет анализ проводиться в **Шаговом** или **Полном** режимах. Запуск анализа в шаговом режиме обеспечивается в интерактивном режиме нажатием кнопки **Пуск**. В этом случае на экране дисплея формируется начальное состояние, последующие нажатия кнопки **Пуск** обеспечивают формирование программой следующих итераций.

При выборе режима **Аналитический** тип запуска **Полный** нажатие кнопки **Пуск** реализует все итерации.

Б.14 Опции памяти

Числовые параметры и ограничения

Если ограничения на размерность тех или иных массивов не устраивают пользователя, они могут быть изменены динамически (в пределах доступной оперативной памяти).

Разрешается изменять следующие параметры:

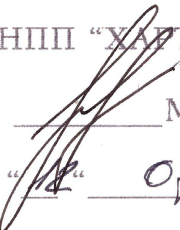
- **максимальное число элементов;**
- **максимальное число входов;**
- **максимальное число выходов;**
- **максимальное число параметров;**
- **максимальное число связей;**
- **максимальное число варьируемых параметров;**
- **максимальное число выдач;**
- **максимальное число выдаваемых переменных;**
- **максимальное число ссылок на варьируемые параметры.**

Невозможно установить параметры меньше, чем размерность имеющейся в памяти модели. При нехватке памяти выдается сообщение об ошибке, а параметры, при обработке которых возникла ошибка и все, следующие за ними, устанавливаются в исходное состояние.

ПРИЛОЖЕНИЕ В
АКТЫ ВНЕДРЕНИЯ РЕЗУЛЬТАТОВ ДИССЕРТАЦИИ

“УТВЕРЖДАЮ”


Генеральный директор
НПП “ХАРТРОН-ЭНКОС”


_____ М. А. Чернышов
“14” 07 1999 г.

“УТВЕРЖДАЮ”

Зам. проректора
по научной работе ОГПУ



_____ Г. В. Кострова
“10” 06 1999 г.


АКТ

приемки-сдачи графической программной среды DCNET

Мы, нижеподписавшиеся, представитель НПП “ХАРТРОН-ЭНКОС” в лице Е. Н. Демидова, с одной стороны, и представитель Одесского политехнического университета В. А. Денисенко, с другой стороны, составили настоящий акт о том, что разработанная в соответствии с внутренними планами университета сотрудниками — В.А. Денисенко, А.Г. Нестерюком, В.В. Ляхиным графическая программная среда DCNET внедрена в НПП “ХАРТРОН-ЭНКОС”. Среда, разработанная на основе теоретических наработок в области дискретно-непрерывных сетей и систем с управляемой структурой, позволяет моделировать сложные технологические объекты управления совместно с алгоритмами непрерывного и логического управления, блокировки, сигнализации и защиты, что дало возможность значительно сократить временные и стоимостные затраты на разработку системы управления конкретным объектом отрасли на этапе эскизного проектирования.

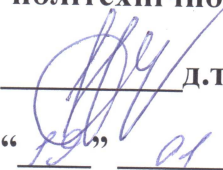
От НПП “ХАРТРОН-ЭНКОС”


_____ Е. Н. Демидов
“14” 06 1999 г.

От ОГПУ


_____ В. А. Денисенко
“8” 06 1999 г.

ЗАТВЕРДЖУЮ

Проректор Одеського національного
політехнічного університету,

 д.т.н., проф. С.А. Нестеренко
 “ 12 ” 2016 р.


АКТ

впровадження в навчальний процес ОНПУ результатів дисертаційної роботи Нестерюка О.Г. на здобуття наукового ступеня кандидата технічних наук

Комісія в складі: голови комісії завідувача кафедри “Комп'ютерні системи” (КС) д.т.н., проф. Ситніков В.С., членів комісії: к.т.н., доц. Ступеня П.В., к.т.н., доц. Стрельцова О.В., старшого викладача Гнатенко В.Ю. підтверджує, що перелічені нижче результати, які отримані ст. викладачем кафедри “Комп'ютерні системи” Нестерюком О.Г. в межах його кандидатської дисертації, було впроваджено в навчальний процес кафедри. Установлено наступне.

У лабораторних роботах по дисциплінах “Проектування спеціалізованих комп'ютерних систем”, “Дослідження складних спеціалізованих комп'ютерних систем”, “Комп'ютерні системи” використовується відповідне програмне забезпечення.

У курсі лекцій по дисципліні “Програмне забезпечення спеціалізованих комп'ютерних систем” використовуються розробки Нестерюка О.Г., зокрема, використовуються прикладна інформаційна технологія моделювання та аналізу властивостей гібридних систем.

У дипломних роботах використано запропонований Нестерюком О.Г. метод визначення досяжності стану гібридної системи.

Голова комісії:
зав. каф. КС,

д.т.н., проф.



В.С. Ситніков

Члени комісії:

к.т.н., доц.



П.В. Ступень

к.т.н., доц.



О.В. Стрельцов

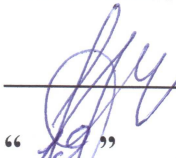

ст. викл.



В.Ю. Гнатенко

ЗАТВЕРДЖУЮ

**Проректор Одеського національного
політехнічного університету,**


д.т.н., проф. **С.А. Нестеренко**
“” _____ **2016 р.**



АКТ

впровадження в навчальний процес ОНПУ результатів дисертаційної роботи
Нестерюка О.Г. на здобуття наукового ступеня кандидата технічних наук

Комісія в складі: голови комісії завідувача кафедри “Комп’ютерні системи” (КС) д.т.н., проф. Ситніков В.С., членів комісії: к.т.н., доц. Ступень П.В., к.т.н., доц. Стрельцова О.В., старшого викладача Гнатенко В.Ю. підтверджує, що перелічені нижче результати, які отримані ст. викладачем кафедри “Комп’ютерні системи” Нестерюком О.Г. в межах його кандидатської дисертації, було впроваджено в навчальний процес кафедри. Установлено наступне.

Модель та інформаційна технологія моделювання та аналізу гібридних систем застосовано у курсі лекцій по дисципліні “Комп’ютерна електроніка” у розділі моделювання аналогових та цифрових компонентів комп’ютерних систем.

Голова комісії:
зав. каф. КС,

д.т.н., проф.

Члени комісії:

к.т.н., доц.

к.т.н., доц.

ст. викл.



В.С. Ситніков



П.В. Ступень



О.В. Стрельцов



В.Ю. Гнатенко



НАУКОВО-ВИРОБНИЧИЙ КОМЕРЦІЙНИЙ ЦЕНТР
ОДИС - W

Вих. № 3/10-1
від 15.04.16 р.

ЗАТВЕРДЖУЮ
Директор
ООО НВКЦ «ОДИС-W»
Казачинський О.М.

АКТ

впровадження практичних результатів дисертаційної роботи
ст. викладача кафедри Комп'ютерних систем
Одеського національного політехнічного університету
Нестерюка Олександра Геннадійовича

Комісія в складі: голови комісії Казачинського Олександра Михайловича та членів комісії Прокопишина Володимира Павловича, Казачинського Василя Михайловича склала даний акт у тому, що ст. викладач кафедри Комп'ютерних систем Одеського національного політехнічного університету Нестерюк О.Г. передал, а комісія прийняла матеріали по забезпеченню методів та базових інформаційних технологій моделювання та аналізу властивості досяжності систем зі структурою, що управляється на базі дискретно-неперервних мереж. В результаті роботи комісії прийнято рішення про доцільність впровадження результатів дисертації Нестерюка О.Г. для використання при розробці та оптимізації системи контролю міського освітлення СПРУТ[®]. В результаті використання запропонованих в дисертації Нестерюка О.Г. методів, інформаційних технологій моделювання та аналізу властивості досяжності дозволило скоротити час розробки та впровадження системи СПРУТ[®] на 30% і покращити якість виробленого обладнання на 6-7%.

Даний акт не є підставою для фінансових розрахунків.

Голова комісії директор, к.т.н. Казачинський О.М.
Члени комісії: технічний директор, к.т.н. Казачинський В.М.
заступник директора, к.т.н. Прокопишін В.П.

