МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество

АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

ФАКУЛЬТЕТ «АЭРОКОСМИЧЕСКИЕ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

Допущен к защите:

зав. кафедрой «Компьютерные технологии»

д. ф.-м. н., профессор _____ Куралбаев З.К.

«___»_____20 __г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

пояснительная записка

Адаптивные методы и алгоритмы анализа нелинейных электронных схем на ЭВМ

специальность: 6М070400 - Вычислительная техника и программное обеспечение

Выполнил магистрант гр. МВТп-13-1 _____ Ли А.В.

Научный руководитель к. т. н., доцент _____Ахметова М.А.

Нормоконтролер ст. преп., доктор PhD _____Ержан А.А.

Рецензент д. т. н. профессор _____Байбатшаев М.Ш.

АЛМАТЫ, 2015

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

АА - амплитудный анализатор
АБМ - амплитудная балансная модуляция
АВД - анализатор временных диаграмм
АВМ - аналоговая вычислительная машина
АВП - автоматический выбор пределов (измерений)
АВУ - автоматическое вызывное устройство
АД - амплитудный детектор
АДК — аппаратура допускового контроля
АК — автоматическая калибровка
БАП — блок автопоиска
БАРУ — быстродействующая автоматическая регулировка усиления
БГИС — большая гибридная интегральная схема
ВАХ — вольт-амперная характеристика
ВД — видеодетектор
ВЗУ — внешнее запоминающее устройство
ВИМ — время-импульсная модуляция
ВК — высокочастотная коррекция
ВКС — выходная контурная системе
ВКУ — видеоконтрольное устройство; вводно-коммутационное устройство
ГАП — гибкое автоматизированное производство
ГБИС — гигантская большая интегральная схема
ГВЗ — групповое время задержки; групповое время запаздывания
ИБП — Источник бесперебойного питания
ИВИ — измеритель временных интервалов
ИВК — измерительно-вычислительный комплекс
ИВП — источник вторичного питания
ИВЭП — источник вторичного электропитания
ИДН — импульсный делитель напряжения
САПР — видеоконтрольное устройство; вводно-коммутационное устройство
ПДУ — Пульт дистанционного управления
ПЗС — Прибор с зарядовой связью
ПЗУ — Постоянное запоминающее устройство
ПК — видеоконтрольное устройство; вводно-коммутационное устройство
УВЧ — Усилитель высокой частоты
УЗЧ — Усилитель звуковой частоты
УКВ — Ультракороткие волны
УНЧ — Усилитель низкой частоты
ЭВМ — видеоконтрольное устройство; вводно-коммутационное устройство

АҢДАТПА

Диссертациялық жұмыс күрделі желілі емес электр схемаларын есептеу мен талдауға мүмкіндік беретін бейімделме бағдарламалық жасақтамасын жасауға арналған. Диссертациялық жұмыс әзірлеу барысында электр схемаларының математикалық теңдеуін құрауға қажетті әр түрлі математикалық алгоритмдерін талдау; электр тізбектерінің элементтерін алмастыру схемаларын талдау және математикалық теңдеулердің күрделі жүйелерін шешу тәсілдерін талдау жүргізілді. Жоғарыда аталған талдаулар мен тәсілдер арқылы жұмыс істейтін, сонымен қатар электр схемасын ең оңтайлы тәсіл арқылы талдап және есептей алатын бағдарламалық жасақтама жасалды.

АННОТАЦИЯ

Диссертационная работа посвящена созданию адаптивного программного обеспечения, позволяющему рассчитывать и анализировать сложные нелинейные электрические схемы. В ходе выполнения диссертационной работы был проведен анализ различных математических алгоритмов, необходимых для формирования математических уравнений электронных схем; анализ схем замещения элементов электрических цепей и анализ методов решений сложных систем математических уравнений. Разработано программное обеспечение, работающее с помощью перечисленных выше анализов и методов, способное проанализировать и рассчитать электрическую схему наиболее оптимальными методами.

ABSTRACT

Dissertation work is devoted to the creation of the adaptive software allowing to count and analyze difficult nonlinear electric circuits. During performance of dissertation work the analysis of various mathematical algorithms necessary for formation of the mathematical equations of electronic schemes was carried out; analysis of equivalent circuits of elements of electric chains and analysis of methods of decisions of difficult systems of the mathematical equations. The software working with the help of the listed above analyses and methods capable to analyse and calculate an electric circuit by the most optimum methods is developed.

ВВЕДЕНИЕ

Современный этап развития электротехнических и электронных устройств характеризуется усложнением электрических и электронных цепей, большим разнообразием функций, выполняемых ими [1, стр.24].

Проблема обеспечения качества и надежности улучшения технико-экономических показателей сложных устройств и узлов тесно связана с внедрением в производство систем автоматизированного проектирования [1, стр.24]. Особенно остро эта проблема касается наиболее ответственного этапа: оптимального выбора конструкторских решений, тщательной проработкой принципиальной схемы устройства в целом или отдельных его узлов [1, стр.24].

Для решения перечисленных задач необходима разработка методов/ алгоритмов и программы машинного анализа сложных объектов, повышающих эффективность использования электронно-вычислительных средств [1, стр.24]. Существенная нелинейность характеристик электротехнических и электронных устройств делает задачу исследования процессов весьма сложной даже при использовании самых современных средств вычислительной техники [1, стр.24].

Актуальность темы. Перспективный подход к моделированию таких сложных цепей основан на реализации соответствующих принципов адаптации математического обеспечения в подсистеме схемотехнического проектирования [1, стр.24]. По мнению многих ученых данное решение является единственно правильным, т.к. разнообразие электрических схем и ее элементом становится настолько большим, что уже невозможно предугадать способ решения той или иной задачей. Такие программы заслужили огромную популярность в работе самых разнообразных предприятий в рамках САПР. Без умения работы с автоматизированной адаптивной программой уже невозможно представить работу современного инженера на большом предприятии [1, стр.24].

Суть адаптации заключается в ее способности приспосабливаться к особенностям поставленной задачи путем динамической настройки программного обеспечения на основе априорной и апостериорной информации [1, стр.25].

В последние годы много работ посвящено реализации принципов адаптации, предназначенных для систем схемотехнического проектирования [1, стр.25]. Большое внимание стала привлекать разработка структуры адаптивной САПР, обеспечивающей динамическое взаимодействие проектировщика и ЭВМ [1, стр.25]. Создание таких САПР требует единого системно-программного подхода при разработке всех этапов моделирования, от создания математических моделей отдельных элементов электронной цепи до выбора методов формирования и обработки совокупности системы уравнений анализируемой цепи [1, стр.25]. Эффективным способом реализации такого подхода является разработка гибкой модульной иерархической структуры системы, обеспечивающей ее открытость, состоятельность, надежность, что, в

свою очередь, предусматривает глубокое изучение свойств моделей анализируемой схемы, с целью, разработки критериев распознавания класса задач и выбора наиболее эффективных алгоритмов для ее решения [1, стр.25].

Степень научной разработанности проблемы. Созданию адаптивных вычислительных алгоритмов, использующих принцип разбиения сложных задач на ряд более простых, посвящены работы [1-7]. Из этого большого количества работ можно выделить ряд работ, в которых используется идея объединения более простых ее частей на основе уравнений связи [8-11] или идея включения соответствующих источников [6-11]. Кроме того особое место среди адаптивных программ САПР занимает программный продукт PSPICE. Он позволяет рассчитывать электрические цепи различной степени сложности адаптивным путем. Однако, существуют математические методы и алгоритмы, не изученные авторами книг и программного обеспечения, кроме того, эта литература и исходные коды могут показаться новичкам довольно сложными.

Целью данной работы является разработка адаптивного программного обеспечения для анализа и расчета сложных нелинейных электрических и электронных схем, превосходящая по своим хароактеристикам другие бесплатные аналоги.

В связи с вышеизложенным, в диссертационной работе **ставятся задачи** исследования адаптивного подхода по таким направлениям:

1. Разработка способов формирования математических моделей
2. Создание базы алгоритмов считывания данных
3. Получение эквивалентных схем элементов электрической цепи
4. Реализация адаптивных алгоритмов в программах анализа нелинейных электрических схем
5. Накопление базы методов решения полученных математических уравнений и систем
6. Создание блока формирования выходных данных.

Объектом исследования являются нелинейные электрические схемы, для которых необходима разработка оптимальных методов и алгоритмов анализа и расчета программным путем.

Предметом исследования являются адаптивные методы и алгоритмы, наиболее подходящие для анализа и расчета той или иной нелинейной электрической схемы.

Методами исследования являются разделы и законы математики, касающиеся решения нелинейных систем уравнений, разделы и теоремы физики и электроники, инструменты программирования, позволяющие создать систему автоматического проектирования.

Гипотеза исследования заключается предположении о том, что с помощью систем различным математических методов и алгоритмов можно рассчитать нелинейную электрическую схему любой степени сложности с максимально точными и надежными результатами.

Сформированные направления исследований **определили структуру диссертации** следующим образом. Работа состоит из четырех глав и приложения.

В первой главе формируется задача адаптивного моделирования.

Во второй главе формируется совокупность адаптивных алгоритмов моделирования электронных схем. Предложены способы адаптации формирования математических моделей, адаптации схемных моделей и адаптации алгоритмов анализа. Создается база алгоритмов и методов анализа нелинейных электронных цепей для использования программой в расчете цепей.

В третьей главе, описано созданное программное обеспечение для проектирования электронных устройств. Рассказано о принципах работы данного программного обеспечения. Дается справка по эксплуатации программы.

В четвертой главе на примерах расчета электронных схем проводится демонстрация разработанного математического и программного обеспечения.

Научная новизна:

1. База математических алгоритмов и методов пригодная для расчета максимально большого числа разнообразных нелинейных электрических схем.

2. Набор схем замещения, с помощью которого можно упорядочить любую нелинейную электрическую схему.

3. Структура адаптивного программного обеспечения, отвечающая всем современным знаниям о программировании программных продуктов.

На защиту выносятся следующие результаты:

1. Формирование задачи адаптивного моделирования

2. Осуществление адаптивного выбора моделей компонентов.

3. Способы формирования сложных математических уравнений на основе элементов электронной цепи.

4. Адаптивное решение сложных систем уравнений

5. Программа анализа нелинейных электрических и электронных цепей, использующая адаптивный алгоритм.

6. Адаптивный анализ сложной нелинейной электрической схемы с помощью адаптивного программного обеспечения.

7. Рекомендации по использованию предложенного адаптивного метода моделирования для решения проблемы оптимизации вычислений.

Научная значимость в обосновании и надежности полученных результатов созданного адаптивного программного обеспечения в расчетах сложных нелинейных электрических схем.

Практическая значимость заключается реализации адаптивных методов и алгоритмов анализа адаптивного программного обеспечения на основе электрических схем, представленных в работе предприятий в сфере электротехники. Предполагается, что работы с помощью данного программного продукта значительно повысит быстродействие расчета и

анализа нелинейных электронных схем, наряду с получаемыми точными результатами расчета.

Апробация. Основные положения и результаты работы были доложены на конференциях в Алматинском Университете Энергетики и Связи и научном центре IT Алатау. По теме исследования автором подготовлены и опубликованы 4 статьи в сборниках АУЭС, английской международной академии наук и высшего образования, Московского Физико-Технического Института и научного центра IT Алатау. Ниже перечисляются опубликованные работы:

1. Жунусов З.А., Ли А.В. “Анализ сложных нелинейных электронных схем. Адаптивные методы и алгоритмы анализа”. Труды III Международной научно-практической конференции “ИКТ: образование, наука, информация”. – Алматы, 2013. – С. 5-7.

2. Жунусов З.А., Ли А.В. “Адаптивные методы и алгоритмы анализа нелинейных электронных схемна ПК”. Materials digest of the “LXIII International Research and Practice Conference and II stage of the Championship in physics, mathematics and chemistry sciences”. – London, 2013 – С.24-27

3. Жунусов З.А., Ли А.В. “Адаптивные алгоритмы моделирования электронных схем”. Труды 56-й научной конференции МФТИ, всероссийской научной конференции “Актуальные проблемы фундаментальных и прикладных наук в современном информационном обществе”, всероссийской научно-инновационной конференции “Физико-математические науки: актуальные проблемы и их решения”. – Москва: МФТИ, 2013. – С.52-53

4. Ли А.В. “Применение математических методов для адаптивного моделирования электронныз схем”. Сборник научных трудов магистрантов специальностей “Вычислительная техника и программное обеспечение” и “Информационные системы”. – Алматы: АУЭС, 2014. – С.67-81

Структура и объем работы. Диссертация состоит из введения, четырех глав и заключения. Диссертация содержит 101 страниц, 33 рисунка, 1 приложение. Список литературы содержит 92 наименований.

ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ АДАПТИВНОГО МОДЕЛИРОВАНИЯ

1.1 Постановка задачи адаптивного моделирования. Создание структуры программного обеспечения

Внедрение систем автоматизированного проектирования является одним из решающих факторов интенсификации и научного перевооружения производства. За сравнительно короткий период системы автоматизированного проектирования в своем развитии прошли путь от первых простейших машинных программ до мощного инструмента разработчика электронной техники; это стало возможным благодаря значительным усилиям, затраченным на развитие алгоритмического и информационного обеспечения, системных средств и вычислительной техники.

Однако, разработка новых систем автоматизированного проектирования, удовлетворяющих требованиям динамично развивающейся электронной техники, сопряжена с большими затратами труда. Стоимость разработки программного обеспечения составляет значительную часть автоматизированных рабочих мест инженеров-проектировщиков, исследователей, экспериментаторов, причем тенденция неуклонного повышения стоимости программ характерна для современного этапа развития автоматизированного проектирования. Поэтому одним из основных критериев качества разрабатываемого программного обеспечения является его “срок жизни” в условиях быстрого усложнения задач, подлежащих решению. Жизнеспособность программного обеспечения может быть повышена на основе соблюдения принципов, заключенных в идее проблемной адаптации. Проблемная адаптация систем автоматизированного проектирования означает ее способность приспосабливаться к особенностям поставленной задачи путем динамической настройки программного обеспечения на основе априорной и апостериорной информации [8,9].

Термин адаптация использовался вначале для описания способности живых организмов приспосабливаться к условиям обитания и изменениям окружающей среды, а в технических приложениях нашел применение в теории и проектировании систем автоматического управления [9, 10].

Адаптация систем автоматизированного проектирования к кругу решаемых вопросов возможна только при учете специфических особенностей задач и осуществляется путем настройки алгоритмов с помощью критериев адаптации, задаваемых разработчиком или вырабатываемых на основе анализа исходных данных и промежуточных результатов [1, стр.25]. Такая организация позволяет создать системы автоматизированного проектирования, универсальные по отношению к различным классам задач и в то же время специализированные для любого представителя из любого класса. Таким образом, преодолевается существующее противоречие между универсальными и специализированными системами, и открываются новые пути повышения эффективности [1, стр.26].

Пример блок-схемы адаптивной системы схемотехнического проектирования можно найти в [10].

Приведенная блок-схема построена по классическому принципу, широко используемому для проектирования аналоговых линейных схем, которые описываются системами интегро-дифференциальных или разностных уравнений. Однако, для проектирования современных электронных систем, включающих аналоговые, дискретно-аналоговые и цифровые части, целесообразно использовать иерархический подход [11, 12]. При этом для некоторых частей рассчитываемой системы уравнения не формируются, а решение получается алгоритмически на основе логических операций. С учетом приведенных выше рассуждений приведена несколько отличающаяся от предложенной в [12] блок схема (рисунок 1).

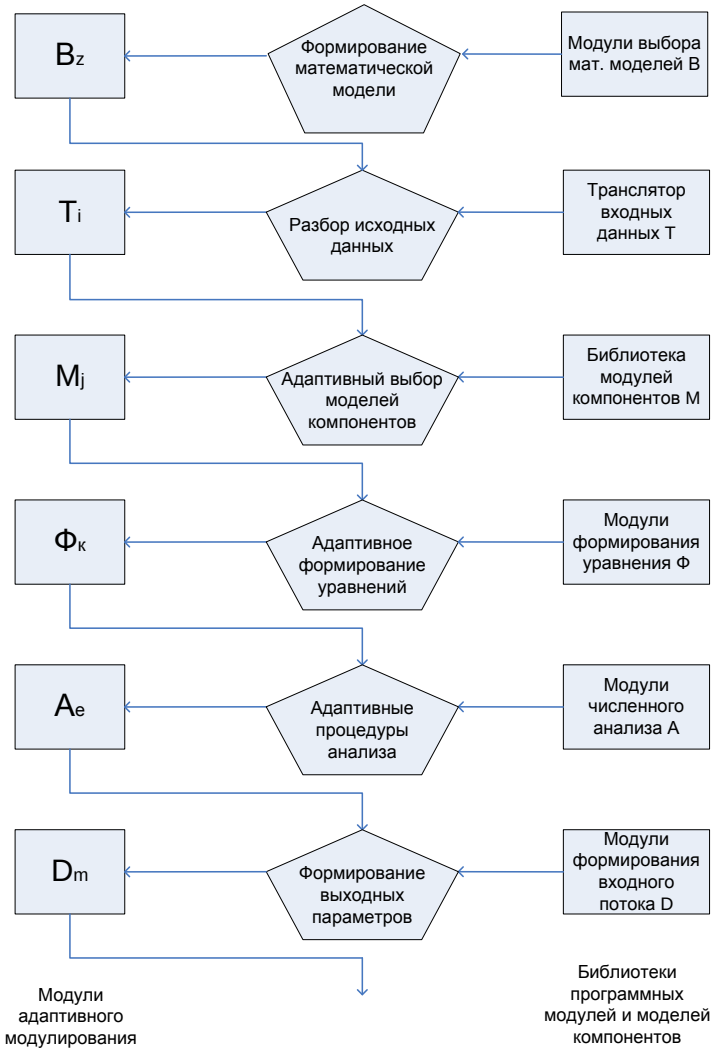

Рисунок 1 - Блоки анализа и принятия решений по критериям адаптации

Отличие состоит в том, что перед блоком формирования уравнений введен дополнительный блок, осуществляющий адаптивный выбор алгоритма решения. Это процедура состоит в анализе предлагаемой задачи, выборе

способа получения решения: на основании уравнений, алгоритмически или комбинаторном методов и подключении необходимых процедур [1, стр.26].

Существует огромное разнообразие электрических и электронных цепей, сложность которых варьируется от самых простых, с несколькими элементами, до самых сложных, с несколькими тысячами элементов электрической цепи. При этом совершенно необязательно, в случае большой сложности цепи, вычислять значения всех элементов цепи. В результате анализа входных данных может выясниться, что для решения необходимо вычислить лишь часть из множества значений. Тем самым представляется возможным в разы сократить время вычисления, а также объем затрачиваемых ресурсов.

За блоком анализа входных данных следует блок адаптивного выбора моделей компонентов. Как было сказано выше, на этой стадии осуществляется выбор наиболее эффективного алгоритма решения задачи в данном конкретном случае. Выбор наилучшего способа решения имеет огромное значение, т.к. при изначально равных по эффективности алгоритмах, при выборе первого мы можем выиграть, а выбрав второй можем потерять огромное количество времени и компьютерных ресурсов.

После того, как сделан выбор наилучшего, в данном случае, алгоритма, составляется система уравнений. Этот модуль, называющийся модулем адаптивного формирования уравнений, также имеет огромное значение. Необходимо выбрать такую систему уравнений, которая с одной стороны полностью отвечала поставленным задачам, а с другой стороны затрачивала бы минимальное количество ресурсов и не была громоздкой.

Сформировав систему уравнений, естественно, необходимо решить ее. Решить систему уравнений необходимо наиболее эффективным способом, за что и отвечает блок адаптивных процедур анализа. Из огромного количества конкурирующих теорий и методов решения уравнений необходимо выбрать наиболее эффективные.

И наконец, завершающим этапом решения задачи адаптивным методом является вывод решений. В блоке формирования выходных параметров формируются и выводятся на экран решения поставленной задачи.

Можно указать два способа организации проблемно-адаптивной системы: альтернативный и императивный.

При альтернативном способе (рисунок 2 а) все модули системы рассчитаны на решение некоторой задачи максимально возможной сложности и организуются в жесткой логической структуре. Адаптация системы к конкретной задаче осуществляется на основе состояний альтернативных ключей, которые идентифицируются в соответствии с критериями адаптации. Таким образом, универсальность системы обеспечивается ее избыточностью, а специализация – ее настройкой, выполняемой как на основе исходных данных, так и в процессе прохождения задачи [13, стр.5].

При императивном способе (рисунок 2 б) в оперативную память ЭВМ загружается информационно-организационная магистральная совокупность программных модулей, обеспечивающих решение типичных задач,

преимущественно встречающихся в практике проектирования. Каждый раз, когда обнаруживается несостоятельность такой совокупности модулей, вызываются соответствующие программные модули из внешней памяти ЭВМ. Таким образом, универсальность специализированной системы обеспечивается за счет резерва программных моделей, ассортимент которых может пополняться в процессе эксплуатации с учетом накапливаемого опыта и расширения круга решаемых задач [13, стр.25].

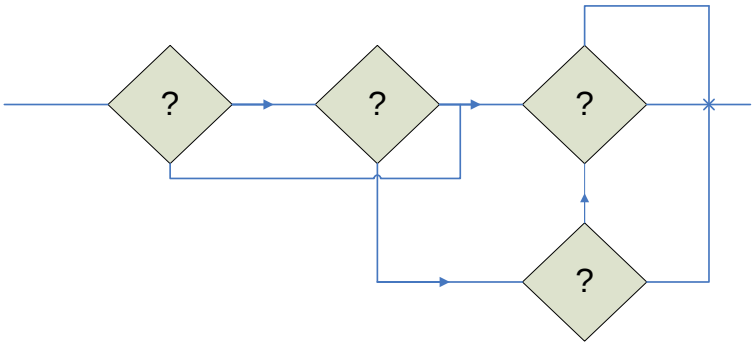

Рисунок 2 (а) - Альтернативный способ

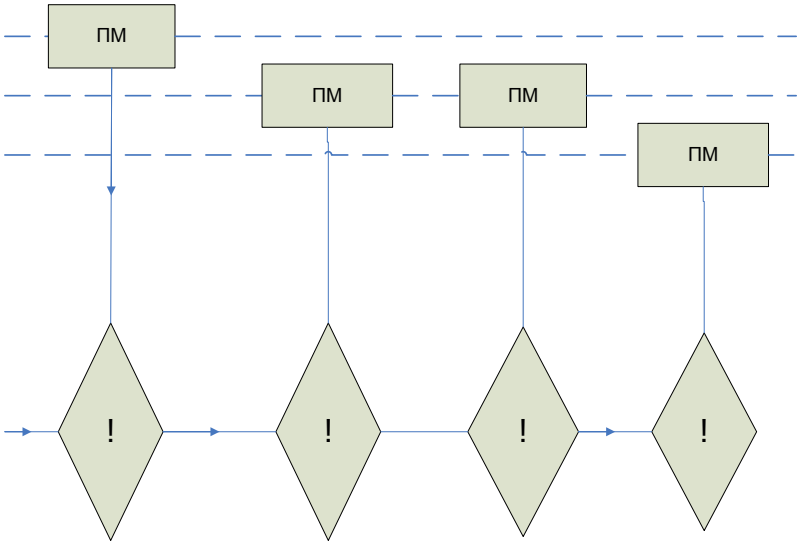

Рисунок 2 (б) - Императивный способ

Каждый из этих способов внутренней адаптации имеет свои положительные стороны, которые наилучшим образом реализуются при комбинированном их применении. Реализация принципов адаптации предъявляет ряд требований к структуре и организации системного, алгоритмического и информационного обеспечения систем автоматизированного проектирования [1, стр.27].

К основным принципам адаптивных систем следует отнести:

1. Диалоговые средства интерфейса с пользователем;
2. Универсальность средства интерфейса с пользователем;
3. Открытость системы;

4. Модульность;
5. Состоятельность системы;
6. Надежность;
7. Эффективность;
8. Самонастраиваемость на круг решаемых задач;
9. Широкие графические возможности.

Широкому применению систем автоматизированного проектирования способствует уровень организации диалога с выдачей диагностических и информационных сообщений, работа в интерактивном режиме с анализом и контролем промежуточных результатов с возможностью прерывания работы, т.е. общительностью программного обеспечения. Одним из важнейших требований является универсальность – применимость системы или ее части для анализа широкого класса электронных схем [1, стр.27].

Открытость системы автоматизированного проектирования определяется готовностью к включению новых алгоритмов и моделей компонентов, а развиваемость – возможностью совершенствования структуры и расширение функциональных возможностей на основе накапливаемого опыта и возникающих потребностей практики [1, стр.27].

Наиболее существенным требованием является состоятельность – способность анализировать критические ситуации и распознавать ошибки при нарушении нормального хода вычислений или недостоверных результатах с целью автоматического устранения подобных ситуаций или прерывании с выдачей соответствующей информации пользователю [1, стр.27]. Повышение эффективности и надежности проектирования связано с принципом параллельности – использованием различных алгоритмов для решения одной и той же задачи на критических этапах моделирования для повышения достоверности результатов, а также распараллеливание алгоритмов для ускорения процесса решения [1, стр.27]. Развитие систем автоматизированного схемотехнического проектирования показало, насколько удовлетворяются эти требования, настолько повышается уровень самоорганизации систем и ее способность к адаптации. Отсюда следует вывод о том, что универсальная система должна быть самонастраивающейся, причем самонастройка должна производиться в процессе решения задач [1, стр.27].

Практическая реализация адаптивного подхода связана с выбором адаптивных соотношений между множеством задач, с одной стороны, и множеством алгоритмов их решения, с другой.

Алгоритмы одного и того же назначения образуют подмножество конкурирующих алгоритмов. Любой из них способен решить одну и ту же задачу, но эффективность решения в каждом конкретном случае зависит от специфики задачи. В системах автоматизированного схемотехнического проектирования подмножества конкурирующих алгоритмов могут образовываться, например, алгоритмами трансляции исходных данных, формирования математической модели схемы в том или ином координатном базисе, алгоритмами численного интегрирования систем нелинейных

дифференциальных уравнений и т.д. Количество таких подмножеств равно количеству последовательных этапов, на которые можно разбить весь процесс проектирования. Таким образом, каждый этап проектирования обслуживается своим подмножеством конкурирующих алгоритмов, и выбор осуществляется на основе критериев проблемной адаптации.

Критерии адаптации по своей природе могут быть империческими, теоретическими или эвристическими и разрабатываются на основе статической обработки экспериментальной информации, теоретического анализа или экспертных оценок. В общем случае для выполнения очередного шага необходимый алгоритм выбирается из подмножества конкурирующих алгоритмов на основе группы критериев, образующих некоторую иерархическую последовательность. Группа критериев адаптации организуется в некоторый адаптор шага, и каждое подмножество конкурирующих алгоритмов снабжается соответствующим адаптором.

Адаптор шага – это процедура, определяющая очередность применения критериев адаптации на подмножестве конкурирующих алгоритмов. Адаптор можно представить в виде матриц, где строки и столбцы соответствуют различным стандартным компонентам систем автоматизированного проектирования и, если элемент матрицы с номерами $a_{ij} = k$, то это соответствует адаптации i-го и j-го видов обеспечения. Альтернативным способом представления адаптора является корневое дерево, внутренние вершины которого соответствуют критериям адаптации, а ветви – путем поиска оптимального алгоритма в соответствующем подмножестве (рисунок 3).

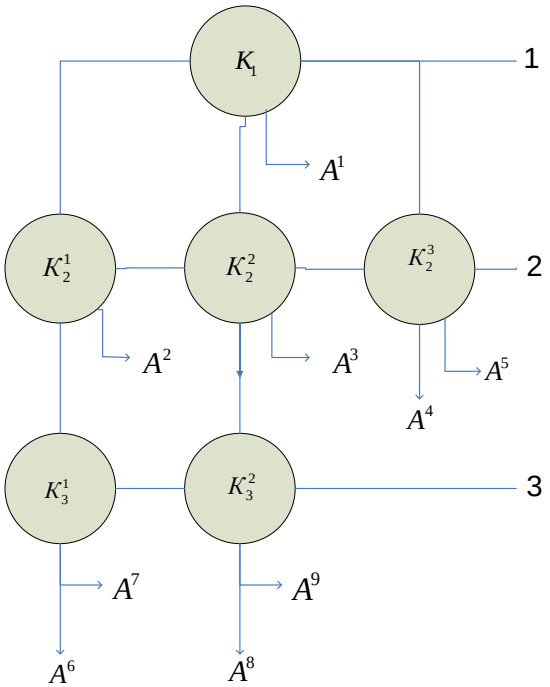

Рисунок 3 - Корневое дерево

Корневая вершина соответствует критерию адаптации первого уровня, а конечные вершины – алгоритмам. При изменении состава алгоритмов в подмножестве вносится изменения и в соответствующий адаптор. Такие структуры позволяют легко вводить новые и модифицировать существующие критерии адаптации [14].

Другим важным понятием в решении задач адаптивным методом является погрешность решения. Пользователь может сам, программно, задавать величину этой погрешности, и тогда, полученная программой величина, находящаяся в пределах этой заданной погрешности будет считаться корректной.

Огромным достоинством описываемого метода является то, что на каждой стадии решения той или иной задачи, проверяется корректность и эффективность пути решения.

Например, имеется электрическая цепь состоящая из семи участков, имеющих произвольные элементы электрической цепи. Пользователь, работающий с данной программой, задал определенную величину погрешности расчета. Первые три участка цепи показали, что вычисленные величины находятся в пределах заданной погрешности, т.е. ход решения выбран корректно. Но после прохождения четвертого участка цепи, мы видим, что найденная величина начала выходить за пределы погрешности. В этот момент программа останавливает ход решения, но откатывается не в начало, а только лишь на предыдущий участок цепи, и проанализировав ход решения, делает поправки, и продолжает расчеты.

Узловым звеном системы схемотехнического проектирования является подсистема моделирования электронных схем: от качества реализации этой системы в существенной степени зависит эффективность всей системы схемотехнического проектирования. Поэтому остановимся на принципах построения адаптивных систем моделирования электронных схем.

1.2 Выводы

1. Поставлена актуальная задача для систем автоматического проектирования – создание полнофункционального программного обеспечения, позволяющего полностью автоматизировать процесс расчета нелинейных электронных схем, тем самым, значительно снизив объем работы инженера.

2. Продемонстрирована схема работы программного обеспечения с разделением на пять последовательных блоков. Разъяснен принцип работы программы анализа нелинейных электрических цепей.

ГЛАВА 2. ФОРМИРОВАНИЕ СОВОКУПНОСТИ АДАПТИВНЫХ АЛГОРИТМОВ МОДЕЛИРОВАНИЯ ЭЛЕКТРОННЫХ СХЕМ

Эффективность использования систем автоматизированного схемотехнического проектирования во многом определяется набором заложенных в системе вычислительных методов, реализующих выполнение основных видов анализа. При огромном многообразии методов трудно и даже невозможно выбрать один метод, идеально приспособленный к особенностям анализируемой электронной схемы. Поэтому при выборе базовых вычислительных алгоритмов целесообразно использовать адаптивные алгоритмы, позволяющие сочетать преимущества различных алгоритмов. Рассмотрим некоторые основные аспекты адаптации на различных этапах моделирования.

Адаптация входного языка направлена на преодоление барьера между системой и пользователем. Входные языки служат для задания исходной информации об объектах и целях проектирования. Во входных языках можно выделить две части: непроцедурную, служащую для описания структур объектов, и процедурную, предназначенную для описания заданий на выполнение проектных операций. Этим частям соответствуют языки описания объектов и языки описания заданий. Разновидности первых: схемные языки, графические языки и языки моделирования.

Схемные языки применяются для описания электрических и электронных схем и содержат данные об элементах схем и их связях друг с другом. Графические языки используются для ввода чертежей, геометрических изображений, деталей и т. п. Задание геометрии деталей осуществляется различными способами: координатным, структурно-символическим (методом типовых графических элементов), аналитическим (математическими уравнениями поверхностей и линий) и рецепторным (мозаичным). Разработаны специальные системы графического программирования.

В качестве примера современного языка проектирования можно указать язык VHDL (VHSIC - hardware description language) - язык описания аппаратуры на базе сверхвысокоскоростных интегральных схем. Этот язык принят в качестве стандарта как инструментальное средство автоматизации проектирования СБИС, ориентированное на методологию нисходящего проектирования. Он является достаточно универсальным, чтобы охватить все аспекты проектирования изделий в области цифровой электроники.

Языки моделирования близки к алгоритмическим языкам и применяются для описания процессов в моделируемом объекте. Анализ алгоритмов расчета электронных схем показывает, что все они состоят из определенного числа алгоритмических операций. Все алгоритмические операции можно разбить на две группы – топологические и вычислительные. К топологическим относятся операции преобразования одной группы топологических переменных в другие, позволяющие модифицировать библиотечные модели, используя для этого любые арифметические выражения и функции, а также составлять и

каталогизировать макромодели в виде эквивалентных схем с произвольным зависимыми источниками и нелинейными элементами. К вычислительным операциям относятся все те алгоритмические операции, в которых, наряду с информацией о топологии, используется информация о математических моделях и видах анализа.

Обеспечить перечисленные свойства входного языка системы схемотехнического проектирования можно, используя модульную структуру программного обеспечения, базу данных задачи, инвариантную к алгоритмическому обеспечению системы. Наиболее сложным вопросом является разработка гибкой базы данных задачи и языка управления данными, так как они служат основной для информационного обеспечения программных модулей и налаживание информационных связей с новыми модулями, пополняющими программное обеспечения программных модулей и налаживания информационных связей с новыми модулями, пополняющими программное обеспечение системы. При этом желательно ограничиться минимальным набором атрибутов, устанавливающих связи, чтобы облегчить привязку новых модулей к базе данных задачи. Для экономии памяти, занимаемой информационной базой задачи, последнюю целесообразно представить в виде одномерного структурированного массива, состоящего из фрагментов, каждый из которых содержит однотипную информацию. В общем случае состав и количество фрагментов соответствует задачи максимальной сложности. В каждом конкретном случае исходный состав фрагментов и их длина уточняются автоматически: фрагменты, не используемые при решении задачи, имеют нулевую длину и остальные фрагменты располагаются вплотную друг к другу. Если в процессе решения задачи потребуется запомнить новую информацию, то длина фрагмента увеличивается до требуемой величины, и он дописывается в конец массива в случае его гибкой структуры или «втискивается» на свое место в случае жесткой структуры массива. Наиболее полная адаптация языка управления базы данных и входного языка реализована в универсальной диалоговой системе анализа электронных схем ДИСАЭС [14].

Адаптивная стратегия наиболее эффективна на этапе формирования и решения уравнений и организации вычислительного процесса. На этом этапе можно несколько условно выделить три основных объекта применения адаптивных методов: адаптация формирования математических моделей, адаптивные схемные модели и адаптация алгоритмов анализа. Такое разделение условно, поскольку часто адаптивная модель тесно связана с методами формирования уравнений и работает только для указанного набора методов. Различные авторы по-разному подходят к организации адаптивных процедур и не классифицируют объекты адаптации подобным образом. Поскольку такая классификация позволяет сравнивать различные методы, она полезна и использована ниже.

2.1 Адаптивные формирования математических моделей электронных схем

При разработке высокоэффективных программ анализа электронных схем особое внимание уделяется методам формирования уравнений, описывающих поведение схемы, и методам их решения. Это связано с тем, что вид уравнений определяет объем хранимой информации о схеме, а метод решения – скорость анализа и достоверность полученных результатов. На формирование уравнения оказывают влияние, по крайней мере, три фактора: компонентный состав и структура цепи, характер решаемой задачи и используемый алгоритм анализа. Если структура описание и характер решаемой задачи участвуют в качестве исходных данных, то отношение между моделью и алгоритмом анализа носят более сложный характер и в значительной мере зависят от свойств и состояния конкретной цепи.

Методы формирования уравнений являются хорошо изученной областью и широко описаны в литературе [15-20]. Метод переменных состояния, где в качестве независимых переменных используются напряжения на емкости и токи через индуктивности (переменные состояния), лег в основу многих программ анализа электронных схем второго поколения (SCEPTRE, AEDNET, АФУС, ПАЭС и др.). Уравнения переменных состояния получаются с помощью сложных совместных преобразований компонентных и топологических уравнений. Преимуществом метода является возможность получения системы дифференциальных уравнений в нормальной форме, которая необходима для применения традиционных методов интегрирования сильно увеличивает требуемое машинное время, так как шаг интегрирования зависит от величины разброса постоянных времени в анализируемой схеме. Если использовать неявные методы интегрирования, которые менее чувствительны к разбросу постоянных времени, то преимущество метода переменных состояний теряется. Применение неявных методов интегрирования требует на каждом шаге интегрирования решения систем нелинейных алгебраических уравнений, которое обычно осуществляется методом Ньютона-Рафсона, что, в свою очередь, порождает процедуру многократного решения системы линейных алгебраических уравнений. Поэтому вычислительные затраты весьма существенны. Неявные методы интегрирования по сравнению с явными позволяют уменьшить требуемое число шагов интегрирования, при этом с увеличением разброса постоянных времени увеличивается выигрыш в количестве шагов интегрирования. При определении динамических характеристик анализируемая схема заменяется на каждом из фиксированных моментов времени последовательностью дискретных схем замещения, получаемых в результате дискретизации компонентных уравнений элементов схемы.

Процесс дискретизации включает в себя алгебраизацию дифференциальных уравнений и линеаризацию нелинейных характеристик и

позволяет перейти к линейным уравнениям в любом из известных координатных базисах (1)

$$A(x_{n+1}^{m})x_{n+1}^{m+1} = B(x_{n+1}^{m}) \qquad (1)$$

где n и m – индексы соответственно временных и ньютоновских итераций решения; x_{n+1}^{m+1} – вектор независимых переменных; $A(x_{n+1}^{m})$ – матрица коэффициентов; $B(x_{n+1}^{m})$ – вектор правых частей уравнений схемы; x_{n+1}^{m+1}, $A()$, $B()$ – дискретизированы на $n+1$ временном шаге и $m+1$ итерации Ньютона. В последнее время широкое распространение получили методы формирования уравнений в виде (1).

При таком формировании полностью отсутствуют присущие методу переменных состояния обычные матричные операции, процедуры исключения переменных; значительно сокращаются затраты на формирование и решение благодаря обработке только ненулевых элементов матрицы A. Широкое применение такой формы представления уравнений электронной схемы стало возможным благодаря развитию неявных методов решения совместных систем алгебро-дифференциальных уравнений [21-27] и созданию эффективных алгоритмов упорядочивания и решения больших разреженных линейных систем уравнений [23-27].

Уравнения электронной схемы в виде (1) могут быть получены различными способами. Наименьший порядок систем уравнений получается при формировании их в однородном координатном базисе, а наибольший – в полном координатном базисе [28, 29]. К недостаткам однородного базиса можно отнести ограничения, накладываемые на характер зависимых источников и нелинейных элементов, что сужает класс анализируемых схем. В полном координатном базисе ограничения, присущие однородному координатному базису, снимаются.

Исключая лишние переменные в полном координатном базисе или добавляя нерегулярные компоненты для однородного координатного базиса, можно перейти к расширенному координатному базису различного вида; такой подход используется во многих зарубежных (SPICE2, ASTAP, NAP-2 и др.), а также отечественных (ДИСАЭС, СПАРС, МАЭС-2, ПАРМ-01, ЕС и др.) программах.

На этапе формирования математической модели проектируемой электронной схемы проблемная адаптация призвана обеспечить ее адекватность к характеру анализируемой схемы и используемому для ее анализа алгоритму. При разнообразии возможных типов математических моделей решение этой задачи было бы чрезвычайно громоздким, если каждый раз начинать формирование на основе выходных данных. Поэтому целесообразно исходить из некоторой модели, служащей промежуточным звеном. Такой фундаментальной моделью может служить модель в узловом координатном базисе.

Адаптация формирования системы уравнений можно рассматривать с двух точек зрения: 1) в процессе адаптации не происходит изменения координатного базиса системы уравнений, а меняются лишь коэффициенты формируемой системы и правая часть; 2) в процессе адаптации происходит переключение на формирование системы уравнений в различных координатных базисах. Если первый подход достаточно прост и широко используется в различных программах, то второй связан со значительными сложностями при программной реализации, не исследован до конца и требует дальнейшей разработки.

Эффективность первого подхода во многом определяется качеством программирования. В этой связи уместно ввести понятие качества программно-алгоритмической связи; этот показатель характеризует, насколько программная реализация сочетается и поддерживает метод и алгоритм моделирования. Многие теоретические методы, описанные в литературе, весьма сложно реализовать программно, что свидетельствует об их низком показателе программно-алгоритмической связи. Очевидно, исчерпывающим подтверждением высокого показателя программно-алгоритмической связи адаптивного метода является высокоэффективная программа. Таким образом, требует дальнейшего развития методы адаптивного формирования математических моделей.

2.2 Адаптивные модели

Сравнительно новым, но уже дающим ощутимые практические результаты, подходом к организации адаптивных форм является адаптация библиотек моделей компонентов и возможность их преобразования, не связанная с особенностями используемых численных методов.

Если в первых программах схемотехнического проектирования модели вводились в качестве исходных данных вместе с анализируемой схемой, составляя с ней единое целое, то дальнейшее развитие привело к созданию специальных библиотек этих моделей.

Организация таких библиотек должна давать возможности не только ввода, замены и вывода моделей, но и их редактирование пользователем при решении конкретных задач.

При идеализации реального объекта адекватная модель позволяет учесть все существенные свойства и связи.

Построение адекватной модели зависит не только от характеристик самого компонента, но и от поставленной задачи и определяет точность решения. Построение адекватной модели зависит не только от характеристик самого компонента, но и от поставленной задачи и определяет точность решения. Поэтому для одного и того же компонента может потребоваться не одна, а несколько моделей, обслуживающих различные задачи проектирования или исследования. Усложнение моделей отрицательно сказывается на общей эффективности моделирования, приводит к снижению предельной размерности решаемых задач и даже может отрицательно сказаться на точности результатов.

Поэтому необходимо, чтобы реальный компонент электронной схемы заменялся моделью, обладающей достаточной точностью и по возможность простой.

Противоречивость этих требований нередко вынуждает поступиться точностью в интересах простоты, однако такой компромисс допустим только при условии, что модель не отражает существенные свойства реального компонента электронной схемы.

Адаптивные модели – это такие модели, структура или параметры которой изменяются в процессе анализа, с математической точки зрения в процессе решения системы уравнений производится изменение их числа, типа или степени обусловленности. Наиболее эффективным оказалось применение адаптивных моделей при анализе динамических процессов, характеризуемых так называемых быстрыми и медленными переменными.

Адаптация модели реактивных компонентов в [29, 30] сводится к замене емкостей (индуктивностей) источниками нулевого тока (источниками нулевого напряжения) при уменьшении тока (напряжения) до заданного уровня.

Описанный механизм позволил снизить жесткость уравнений и привел к использованию явных методов интегрирования. В [31, стр.25] рассмотрена адаптивная асимптотическая модель многозвенной RC-цепи. В [31, стр.25] построена наиболее распространенная модель биполярного транзистора (Эберса-Молла) и разработан вычислительный алгоритм, повышающий эффективность расчета.

Повышение вычислительной эффективности динамической моднли биполярного транзистора (Эберса-Молла) достигается, в частности, путем снижения ее размерности за счет исключения внутренних переменных.

Нормальный режим многих электронных цепей – квазилинейный. Однако активные элементы таких цепей относятся к нелинейным многополюсникам, если не ограничивать диапазон их напряжений и токов, а также скоростей их изменения [32, 33].

В частности, выходное напряжение операционого усилителя зависит от выходного по нелинейному закону; при больших входных напряжениях выходное напряжение практически не изменяется. При больших входных сигналах существенное влияние на режим схем оказывает один из нормируемых параметров операционного усилителя – максимальная скорость нарастания выходного напряжения. Нелинейные параметры операционного усилителя могут привести к нелинейным искажениям выходного сигнала и, следовательно, ограничению динамического диапазона цепи, к появлению паразитных релаксационных колебаний и другим нежелательны явлениям.

2.3 Адаптация алгоритмов численного анализа

Моделирование динамических процессов нелинейных схем - наиболее сложная задача, результаты которой используются в других видах анализа (анализа установившихся режимов при периодических воздействиях,

статистический анализ и др.). Во многих методах моделирования динамических процессов проводится поэтапное упрощение общей задачи: от нелинейных интегродифференциальных уравнений осуществляется переход к нелинейным алгебраическим уравнениям и, наконец, к системам линейных уравнений. Поэтому рассмотрим вначале организацию адаптивных процедур для анализа динамических процессов, затем для статического анализа и, наконец, для решения линейных уравнений.

Адаптивные программы могут строиться на основе алгоритмов решения систем обыкновенных дифференциальных, линейных и нелинейных алгебраических уравнений, организованных альтернативным, императивным или комбинированным способами. В [33] в качестве базового алгоритма численного интегрирования принята формула дифференцирования назад, которая используется для аппроксимации производных как для монотонных, так и для быстро изменяющихся переменных; выбор порядка и шага интегрирования основан на оценке погрешности интегрирования. В [34, стр.124] в качестве базового алгоритма также выбрана формула дифференцирования назад, но адаптивная стратегия обеспечивает выбор порядка формулы интегрирования для каждой переменной и шага интегрирования в зависимости от возникающих ситуаций. Ситуация определяется заданной точностью расчета, устойчивостью расчета, принадлежностью рабочей точки, к области, где нарушаются гладкость функций или их производных, оптимальным расчетом начальных участков, машинными затратами на получение значений переменных в заданных точках временного интервала. Этот подход реализован в системе автоматизированного схемотехнического проектирования ПАРУС.

Как утверждается в [34, стр.124], неявные методы интегрирования становятся излишними, когда в задаче ведется расчет быстрых движений и, следовательно, шаг интегрирования соизмерим с наименьшей постоянной времени анализируемой цепи. В этой работе на примере расчета мультивибратора показано, что благодаря переходу к явному методу интегрирования удается уменьшить время счета на 40%. Более подробно вопросы использования методов интегрирования явного и неявного типа и автоматического перехода с метода на метод (комбинированный метод) рассмотрены в работах [33-35]. Если переход осуществляется путем предварительной настройки на конкретную формулу, то метод называется фиксированным, если очередность определяется на каждом шаге, то метод называется циклическим. В этих методах явные формулы применяются на участках резкого изменения функции, а неявные - на участках с плавным ее изменением. Упрощенный подход заключается в разбиении временного интервала на серии последовательных отрезков, на части этих отрезков интегрирование выполняется по неявным формулам, на остальных - по явным; в этом случае погрешности в каждой серии взаимно компенсируются и, как следствие, порядок точности оказывается выше порядков обоих методов. В [35] предложена следующая схема расчета: один шаг явным, второй шаг неявным

методом Эйлера, утверждается, что погрешность такая же, как в методе трапеций; в [36-38] исследованы двухсторонние одношаговые явные и полуявные вычислительные формулы переменного порядка точности и предложены формулы для оценки ограничений на величину шага интегрирования и критерий адаптации, основанный на соотношении вычислительных затрат. В [36-38] описан один из подходов к выбору критерия перехода от одного метода к другому при решении задач анализа нелинейных электронных схем с учетом ее свойств и предложено два критерия: на основе свойства "поглощения" нелинейного вектора - функции правой части системы дифференциальных уравнений и с использованием так называемого коэффициента эффективности численного алгоритма.

Для расчета динамических режимов на ЭВМ на том или ином этапе [38, 39] осуществляется алгебраизация интегродифференциальных уравнений, т.е. приведение их к системе нелинейных алгебраических уравнений. Нелинейные алгебраические уравнения описывают и стационарные режимы нелинейных цепей при воздействии постоянных источников. Поэтому кратко остановимся на задачах, связанных с этими уравнениями. Для решения систем нелинейных уравнений используются различные методы, которые можно разделить на три класса: ньютоновские итерационные методы, оптимизационные методы, методы продолжения по параметру. Рассмотрим возможности этих методов для организации адаптивных процедур. Пусть требуется решить систему нелинейных уравнений вида (2)

$$f_i(x_1, x_2, \ldots, x_n) = 0, i = 1,2,..n \tag{2}$$

или в векторной форме (3)

$$f(x) = 0 \tag{3}$$

Предполагается, что функции $f(i)$ имеют непрерывные частные производные по всем аргументам $\frac{\partial f_i}{\partial x_i}$, образующие матрицу Якоби $\frac{\partial f}{\partial x}$, которая является невырожденной во всей области определения функции f_i.

Поскольку аналитическое решение таких систем, как правило, получить не удается, для их решения в вычислительной математике разработаны многочисленные численные методы, среди которых широко используется класс так называемых ньютоновских методов [38-50].

Метод Ньютона-Рафсона, основанный на разложении $f_i(x)$ в ряд Тейлора с точностью до членов первого порядка (4)

$$f(x^{(k+1)}) = f(x^{(k)} + \Delta x^{(k)}) \approx f(x^{(k)}) + \left.\frac{\partial f}{\partial x}\right|^{k} \Delta x^{(k)} = 0 \tag{4}$$

Позволяет вычислить на каждой итерации поправку (5)

$$\Delta x^{(k)} = -\left(\left. \frac{\partial f}{\partial x} \right|^{k} \right)^{-1} f(x^{(k)});\ x^{(k+1)} = x^{(K)} + \Delta x^{(k)} \tag{5}$$

Метод Ньютона-Рафсона-.сходится всегда, если только начальное приближение $x^{(0)}$ выбрано достаточно близко к решению и имеет квадратичную скорость сходимости [49, 50]. Высокая скорость сходимости является важным достоинством этого метода. Для уменьшения объема вычислений используют различные модификации метода Ньютона-Рафсона, в одной из модификаций $\frac{\partial f}{\partial x}$ и обратная ей $\left(\frac{\partial f}{\partial x}\right)^{-1}$ вычисляются лишь в точке начального приближения $x^{(0)}$, а затем строится итерационный процесс по формуле (1.3), но уже с постоянной матрицей Якоби. На рисунке 4 6 и 4 в приведены графические интерпретации обычного (б) и модифицированного (в) методов Ньютона-Рафсона при расчете схемы (рисунок 4 а).

Применение оптимизационных методов, в частности, метода спуска для решения системы уравнений сводится к задаче отыскания минимума функции $\Phi(x)$, которая определяется как некоторая норма вектора невязки системы (6); чаще всего берут

$$\Phi(x) = \sum_{i=1}^{n} f_i^{\,2}(x) \tag{6}$$

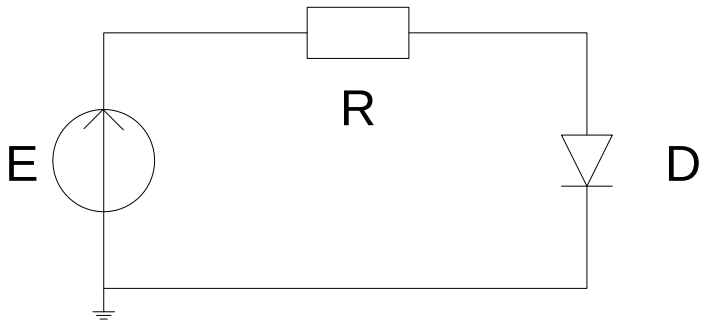

Рисунок 4 (а) - Пример схемы для расчета методом Ньютона-Рафсона

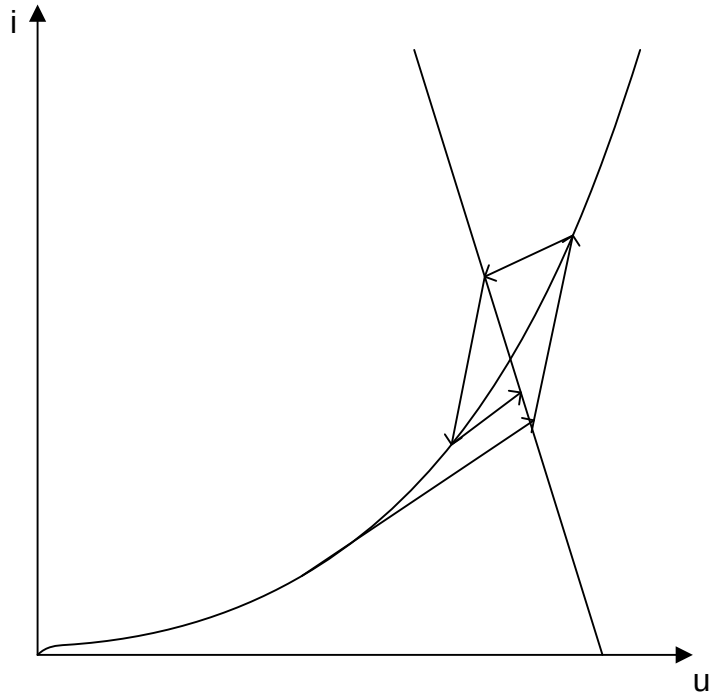

Рисунок 4 (б) - Модифицированный метод Ньютона-Рафсона

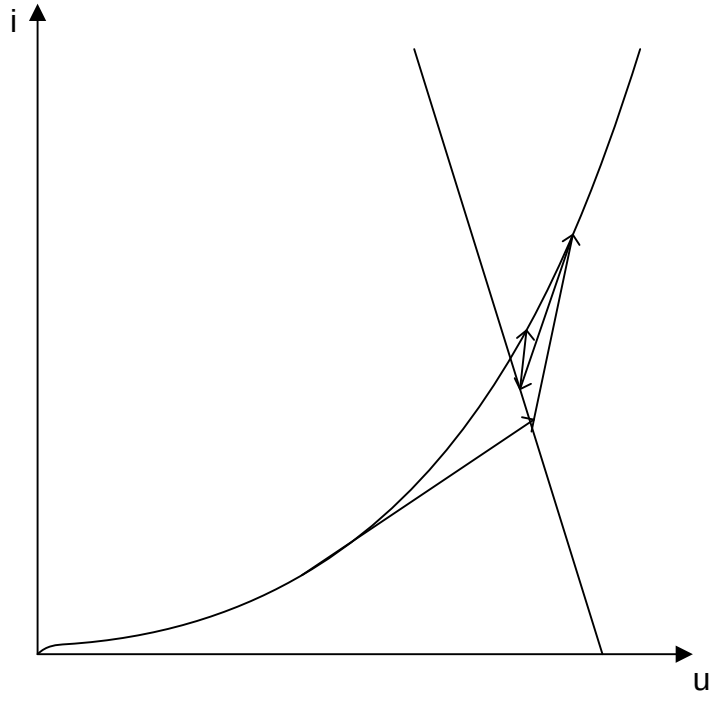

Рисунок 4 (в) - Обычный метод Ньютона-Рафсона

Идея метода спуска заключается в отыскании на каждом шаге итерационного процесса производных $\frac{\partial \Phi}{\partial x} = \nabla \Phi = \left(\frac{\partial \Phi}{\partial x_1}, \frac{\partial \Phi}{\partial x_2}, \ldots, \frac{\partial \Phi}{\partial x_n} \right)$

после строим прямую (7)

$$x = x^{(k)} - \lambda \nabla \Phi(x^{(k)}) \tag{7}$$

проходящую через точку $x^{(k)}$ в направлении вектора $\frac{\partial \Phi}{\partial x}$, ортогонального к поверхности $\Phi(x) = \Phi(x^{(k)})$. Здесь λ - градиентный шаг, определяемый из условия минимума функций $\varphi_k(\lambda) = \Phi(x^{(k)} - \lambda \nabla \Phi(x^{(k)}))$, тогда $x^{(k+1)} = x^{(k)} - \lambda_k \nabla \Phi(x^{(k)})$

Процесс позволяет двигаться в направлении быстрейшего убывания функции $\Phi(x)$ и продолжается до тех пор, пока $\Phi(x^{(k)})$ не станет достаточно малой. Достоинством метода спуска является гарантированная локальная сходимость при произвольном выборе начального приближения для широкого класса нелинейных электронных схем, а недостатком - низкая скорость сходимости и довольно большой объем вычислений на каждом шаге. Как правило, для анализа электронных схем этот метод непосредственно не применяется, а используется в комбинациях с другими методами. Возможно и использование и других оптимизационных методов нелинейного программирования, в частности, не требующих знания производных целевых функций. Однако, эти методы также как и метод спуска, требуют большого объема вычислений и не нашли широкого применения в программах анализа.

Значительное увеличение области сходимости достигается в классе методов продолжения по параметру; суть которых состоит в том, что вместо системы (5) рассматривается система с введенным параметром (8)

$$f(x,t) = 0, 0 \le t \le 1, x \in R^n \tag{8}$$

В зависимости от способа введения параметра система (8) при $t = 0$ превращается в систему с известным или легко определяемым решением, а при $t = 1$ содержит искомое решение. Методы продолжения решения по параметру особенно удобны в тех случаях, когда решение необходимо получить в виде функции непрерывно изменяющегося параметра, например, определения характеристик вход-выход нелинейных схем. Возможны различные подходы к построению $f(x,t)$ и определению решения. В [50, 51] предлагается преобразовать (1.4) в систему нелинейных дифференциальных уравнений путем дифференцирования по параметру $\frac{dx}{dt} = -\left(\frac{\partial f}{\partial x}\right)^{-1} \frac{\partial f}{\partial t}$ и в дальнейшем решить эту систему методом численного интегрирования.

В работах [50-52] система (8) строится путем введения параметра продолжения при векторе вынуждающих сил. Такой подход был назван В.М. Бондаренко [51, 52] методом суммирования конечных приращений. Система решается .для дискретных значений параметра продолжения $0 \le t_0 < t_1 < t_2 < \ldots < t_n = 1$ локально сходящимся методом Ньютона.

Решение систем нелинейных уравнений при нулевом начальном приближении можно осуществить, если учесть тот очевидный факт, что при отсутствии внешних воздействий все переменные состояния схемы равны нулю и их можно считать решением. Для улучшения сходи мости итерационного процесса целесообразно напряжения и токи независимых источников в процессе решения менять от нулевых до номинальной величины. Допустим , что рассматриваемая нелинейная электронная схема содержит только независимые источники напряжения, которая заменяется последовательностью подзадач расчета схем с источником напряжения

$E_k = \lambda_k * E$ где k - номер подзадачи, $0 < \lambda_k < 1$ Очевидно, что здесь

могут возникнуть трудности при выборе λ_k ,аналогичные с выбором шага в методе Ньютона. Известны несколько вариантов этого способа, отличающихся выбором закона изменения коэффициентов λ_k. В [52] задача выбора λ_k решается применительно к задаче одномерной поисковой оптимизации методом дихотомии (деления пополам). Сначала решение ищется при номинальных значениях источников напряжения $\lambda = 1$. Если при $\lambda_1 = 1$процесс расходится, решаются подзадачи при $\lambda = 0{,}5; \lambda = 0{,}25$ и т.д.,т.е. до тех пор, пока итерационный процесс не сойдется, после чего λ увеличивается аналогичным образом. При повторной расходимости в какой-либо i-й подзадаче, когда $\lambda = \lambda_{\partial \grave{a} \grave{n}}$, λ опять выбирается по закону дихотомии $\lambda_{k+1} = \lambda_k + (\lambda_{\eth \grave{a} \grave{n}} - \lambda_k)/2^i$ (где k - номер последней сходящейся подзадачи, $i = 1,2,...$) до сходимости. В работе [53, стр.88] для $\lambda_0 = 0$ вычисляется вектор невязки и дальнейший выбор осуществляется автоматически, обеспечивая уменьшение (или минимизацию) вектора невязки на каждой итерации решения системы уравнений.

В работе [53, стр.88] предлагается принять $\lambda_k = \lambda t_k$ t-формальное время и для каждой подзадачи выполнять только одну ньютоновскую итерацию, а полный цикл итераций производить лишь при λ_k т.е. при полном E. Подход, развитый в работе [54, стр.99] для частного класса цепей - триггеров, заключается в увеличении напряжения питания от 0 до E сначала на одном плече, а после получения решения - на другом; таким образом, можно получить приемлемое решение. Эффективность вычислительного процесса существенно зависит от выбора шага по параметру продолжения в этой связи, теорема из [54, стр.99] дает некоторую оценку эффективности, но она включает ряд параметров, которые трудно контролировать, поэтому метод весьма сложен для практической реализации. В [54, стр.99] предложено величину шага выбирать из условия количества итераций, очевидно это наиболее простой и эффективный метод контроля шага λ .

Таким образом, адаптивные процедуры решения нелинейных уравнений можно разделить на два класса: алгоритмические и параметрические. Алгоритмические методы предполагают изменение самого способа отыскания решений, примером может служить переход от итерационных методов к оптимизационным. Типичными примерами параметрических методов являются

методы продолжения по параметру. Если последние широко используются в адаптивных процедурах, то системы, в которых используется алгоритмическая адаптация, нельзя считать распространенными.

Для решения систем линейных уравнений служат многочисленные методы [55-57], причем, хотя переход с метода на метод хорошо теоретически проработан [57] и реализован в специальных программах, реализация такого перехода для программ анализа схем используется редко. Следует отметить целесообразность перехода к итерационному методу решения систем линейных уравнений в том случае, если прямые методы (Гаусса, LU -разложения и т.д.) не дают решения.

Общим недостатком рассмотренных адаптивных методов является их низкий показатель программно-алгоритмической связи, что можно пояснить следующим образом. При разработке эффективных методов стеемятся к экономии оперативной памяти и времени работы процессора, не учитывая следующие обстоятельства. Общее время работы программы определяется не столько загрузкой центрального процессора, сколько количеством обращений к внешним запоминающим устройствам. Совершенно очевидно, что нецелесообразно хранить всю исходную или некую промежуточную информацию о большой интегральной схеме в оперативной памяти ЭВМ; ее следует хранить на внешних запоминающих устройствах. Поэтому время расчета определяется числом обращений к внешним устройствам, определяемым числом формирований уравнений. Эта особенность весьма актуальна для мини-ЭВМ и особенно остро стоит при использовании персональных компьютеров. В этой связи весьма актуальна проблема адаптивной организации вычислительного процесса, обеспечивающей минимизацию числа формирований уравнений и объема информации, требуемой для их формирования. Один из возможных подходов к решению подобной задачи описан в следующем разделе.

2.4 Создание базы алгоритмов и методов анализа нелинейных электронных цепей

Точных методов, пригодных для решения нелинейных алгебраических уравнений, не существует.

Применяемые в настоящее время алгоритмы решения нелинейных уравнений состояния основываются на двух методиках. Одна из них связана с применением для построения итерационного процесса методов решения линейных алгебраических уравнений. При этом считается, что на каждом шаге итерации система линейна и может быть решена, например, методом Гаусса или с использованием обратной матрицы. Эти методы получили название методов внешней итерации и, по сути, являются методами многократного решения систем линейных уравнений.

Другая методика связана с применением специализированных методов решения нелинейных систем уравнений (итерационные методы решения - метод простой итерации, метод Зейделя, метод Ньютона и др.).

2.4.1 Применение метода внешней итерации

Если считать систему уравнений линейной на данном шаге, то можно использовать метод Гаусса на каждом шаге итерационного процесса.

Например, при использовании уравнений узловых напряжений задаются начальным приближением напряжений узлов и определяют задающие токи узлов. После этого система уравнений может быть решена по методу Гаусса и может быть получено первое приближение напряжений узлов. Далее переходят к следующему шагу, т.е. определяют задающие узловые токи при значениях напряжений узлов, равных их первым приближениям. Затем находят следующее приближение узловых напряжений и так далее, пока итерационный процесс не сойдется. Таким образом, каждый шаг итерационного процесса состоит из определения задающих токов узлов по приближениям узловых напряжений и решения линейной системы уравнений узловых напряжений.

Следует отметить, что для эффективного решения уравнений установившегося режима по Гауссу необходимо учитывать слабую заполненность матрицы узловых проводимостей [58].

Обратная матрица позволяет более рационально организовать процесс решения нелинейных уравнений. Так как при решении уравнений установившегося режима матрица коэффициентов остается неизменной (так как пассивные элементы считаются линейными), а изменяются только правые части, поэтому обратная матрица, получаемая для такой системы, также остается неизменной и может быть использована на каждом шаге итераций. Если число итераций велико (>4), то такой процесс решения становится более эффективным, чем решение по методу Гаусса при хорошо заполненной матрице коэффициентов. Обратная матрица вычисляется один раз, а каждый шаг итерационного процесса при использовании уравнений узловых напряжений заключается в определении задающих токов по приближениям узловых напряжений и умножении на них обратной матрицы.

2.4.2 Применение итерационных методов для решения нелинейных уравнений

Итерационные методы решения линейных уравнений наиболее просто могут быть использованы и для решения нелинейных задач. Именно поэтому метод Зейделя нашел широкое применение в промышленных программах. Алгоритм итерационного процесса достаточно прост. С помощью небольшого изменения алгоритма можно решать и нелинейные задачи. Это изменение состоит в том, что в уравнения в процессе итерации подставляются не только неизвестные, но и изменяющиеся в зависимости от них вынуждающие силы, в

частности, при применении уравнений узловых напряжений - источники тока [58-60].

Рассмотрим алгоритм на примере решения уравнений узловых напряжений по методу Зейделя (9)

$$\begin{aligned}
&\underline{Y}_{11}\underline{U}_1 + \underline{Y}_{12}\underline{U}_2 + \underline{Y}_{13}\underline{U}_3 + \ldots + \underline{Y}_{1k}\underline{U}_k = \underline{I}_{y1} \\
&\underline{Y}_{21}\underline{U}_1 + \underline{Y}_{22}\underline{U}_2 + \underline{Y}_{23}\underline{U}_3 + \ldots + \underline{Y}_{2k}\underline{U}_k = \underline{I}_{y2} \\
&\vdots \\
&\underline{Y}_{k1}\underline{U}_1 + \underline{Y}_{k2}\underline{U}_2 + \underline{Y}_{k3}\underline{U}_3 + \ldots + \underline{Y}_{kk}\underline{U}_k = \underline{I}_{yk}
\end{aligned} \tag{9}$$

Выразим из каждого уравнения соответствующие неизвестные и запишем общее решение на (n +1) шаге (10):

$$\begin{aligned}
&\underline{U}_1^{(n+1)} = \frac{1}{\underline{Y}_{11}}\left(0\cdot\underline{U}_1^{(n)} - \underline{Y}_{12}\underline{U}_2^{(n)} - \underline{Y}_{13}\underline{U}_3^{(n)} - \ldots - \underline{Y}_{1k}\underline{U}_k^{(n)} + \frac{\overset{*}{\underline{S}}_1}{\sqrt{3}\overset{*}{U}_1^{(n)}}\right) \\
&\underline{U}_2^{(n+1)} = \frac{1}{\underline{Y}_{22}}\left(-\underline{Y}_{21}\cdot\underline{U}_1^{(n)} - 0\cdot\underline{U}_2^{(n)} - \underline{Y}_{23}\underline{U}_3^{(n)} - \ldots - \underline{Y}_{2k}\underline{U}_k^{(n)} + \frac{\overset{*}{\underline{S}}_2}{\sqrt{3}\overset{*}{U}_2^{(n)}}\right) \\
&\vdots \\
&\underline{U}_k^{(n+1)} = \frac{1}{\underline{Y}_{kk}}\left(-\underline{Y}_{k1}\cdot\underline{U}_1^{(n)} - \underline{Y}_{k2}\underline{U}_2^{(n)} - \underline{Y}_{k3}\underline{U}_3^{(n)} - \ldots - 0\cdot\underline{U}_k^{(n)} + \frac{\overset{*}{\underline{S}}_k}{\sqrt{3}\overset{*}{U}_k^{(n)}}\right)
\end{aligned} \tag{10}$$

Алгоритм решения аналогичен рассмотренному ранее.

Сходимость метода Зейделя к решению нелинейных уравнений медленная, что связано с особенностями матрицы собственных и взаимных проводимостей узлов. В практических расчетах установившихся режимов электроэнергетических систем методом Зейделя нужно использовать ускоряющие методы, например, применение ускоряющих коэффициентов, использование которых сводится к следующему [61, 62]:

Если обозначить напряжение k - го узла, определенное на $(i+1)$-м шаге по общим итерационным формулам, через $U_{k\text{óñê}}^{k+1}$, то ускоренное $(i+1)$- е приближение значения напряжения k- го узла $\underline{U}_{k\text{óñê}}^{k+1}$ определяется по формуле: $\underline{U}_{k\text{óñê}}^{i+1} = \underline{U}_{k\text{óñê}}^{i} + q\left(\underline{U}_k^{(i+1)} - \underline{U}_{k\text{óñê}}^{(i)}\right) = \underline{U}_{k\text{óñê}}^{(i)} + q\Delta\underline{U}_k^{(i+1)}$

где $\Delta\underline{U}_k^{(i+1)} = \underline{U}_k^{(i+1)} - \underline{U}_{k\text{óñê}}^{(i)}$ - поправка по напряжению k- го узла на $(i+1)$- м шаге; q - ускоряющий коэффициент.

Значение напряжения, вычисленное с ускорением, $\underline{U}_{k\text{óñê}}^{(i+1)}$ принимается в качестве исходного при расчете следующего $(i+2)$- го шага.

В случае $q=1$ выражение (10) сводится к $\underline{U}_{\text{кóñê}}^{(i+1)}=\underline{U}_{k}^{(i+1)}$, т.е. неускоренный итерационный процесс соответствует случаю, когда ускоряющий коэффициент q принимается равным единице.

Основное достоинство метода в том, что он легко программируется и требует малой памяти ЭВМ. Недостаток метода состоит в медленной сходимости. Метод Зейделя особенно медленно сходится, а в ряде случаев и расходится в расчетах установившихся режимов электрических систем с устройствами продольной компенсации, с трехобмоточными трансформаторами или автотрансформаторами с очень малым сопротивлением обмотки среднего напряжения и для электрических систем с дальними линиями электропередачи и сильной неоднородностью параметров. Метод Зейделя также плохо сходится либо расходится в расчетах режимов, близких к предельным, либо при определении режимов, неустойчивых по статической апериодической устойчивости.

2.4.3 Применение метода Ньютона для решения одного уравнения

Метод Ньютона весьма эффективен для решения нелинейных алгебраических и трансцендентных уравнений. Его основное достоинство состоит в том, что при сравнительно несложном алгоритме он обладает быстрой сходимостью. Метод Ньютона универсален и пригоден для решения обширного класса нелинейных уравнений.

Идея метода Ньютона состоит в последовательной замене на каждой итерации нелинейной системы уравнений некоторой линейной, решение которой дает значения неизвестных, более близкие к решению нелинейных систем, чем исходное приближение.

Алгоритм метода состоит в том, что если задаться некоторым нулевым приближением и вычислить функцию для этого приближения, затем рассчитать производную функции в этой точке, то, используя геометрический смысл производной (тангенс угла наклона касательной), можно определить новое значение $x^{(1)}$, то есть новое приближение аргумента. Затем находят значение функции в точке $x^{(1)}$, значение производной функции в этой точке и новое приближение $x^{(2)}$ и т.д. Следует отметить, что контроль сходимости по величине разности значений переменных на соседних интервалах может привести к неверным результатам. Считают, что итерационный процесс сходится, если функция $W(x)$становится близкой к нулю, то есть $\left|W\left(x^{(i)}\right)\right|\le\varepsilon$ [62-64].

Допустим, что имеется функция $W(x)=0$. Зададимся произвольным значением аргумента $x^{(0)}$ и определим значение функции $W\left(x^{(0)}\right)$ (11).

Определим производную функции: $\dfrac{dW\left(x^{(0)}\right)}{dx}=tg\alpha$

Можно записать: $\frac{dW\left(x^{(0)}\right)}{dx} = \frac{W\left(x^{(0)}\right)}{\left(x^{(0)} - x^{(1)}\right)}$

Тогда

$$x^{(1)} = x^{(0)} - \frac{W\left(x^{(0)}\right)}{\frac{dW\left(x^{(0)}\right)}{dx}} \tag{11}$$

Это выражение позволяет определить новое значение аргумента, которое является очередным приближением искомого корня, если итерационный процесс продолжается.

Данное выражение можно несколько преобразовать (12)

$$x^{(1)} = x^{(0)} + \Delta x^{(1)} \tag{12}$$

где $\Delta x^{(1)} = -\frac{W\left(x^{(0)}\right)}{\frac{dW\left(x^{(0)}\right)}{dx}}$

В общем случае для $(i+1)-$го шага итерации можно записать (13):

$$x^{(i+1)} = x^{(i)} + \Delta x^{(i+1)} \tag{13}$$

где $\Delta x^{(i+1)} = -\frac{W\left(x^{(i)}\right)}{\frac{dW\left(x^{(i)}\right)}{dx}}$

Итерационный процесс ведется до тех пор, пока значение функции W не будет удовлетворять точности ε: $\left|W\left(x^{(i)}\right)\right| \le \varepsilon$

В отличие от ранее рассмотренных итерационных методов здесь для контроля не применяется разность, так как заранее известно, что в точке решения функция обращается в нуль. Поэтому е отражает допустимую величину отличия функции от нуля, то есть величину погрешности определения точного решения.

2.4.4 Применение метода Ньютона для решения систем уравнений любого порядка

Рассмотрим алгоритм применения метода Ньютона [61-65] на примере решения системы нелинейных уравнений третьего порядка (14):

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases} \tag{14}$$

где b_1, b_2, b_3 - функции нескольких переменных (15):

$$
\begin{aligned}
b_1 &= f(x_1, x_2, x_3) \\
b_2 &= f(x_1, x_2, x_3) \\
b_3 &= f(x_1, x_2, x_3)
\end{aligned}
\tag{15}
$$

Приведем функции к виду (16):

$$
\begin{aligned}
W_1(x_1, x_2, x_3) &= 0 \\
W_2(x_1, x_2, x_3) &= 0 \\
W_3(x_1, x_2, x_3) &= 0
\end{aligned}
\tag{16}
$$

Если считать, что система уравнений матричная, то можно записать (17):

$$
W(x) = 0 \tag{17}
$$

где $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

Будем считать, что заданы значения нулевых приближений $x_1^{(0)}, x_2^{(0)}, x_3^{(0)}$

Используя описанный выше метод касательных, заменим каждую из нелинейных функций линейной, пользуясь при этом разложением в ряд Тейлора, то есть представим уравнения в следующем виде (18):

$$
\begin{aligned}
&W_1\left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}\right) + \frac{\partial W_1\left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}\right)}{\partial x_1} \underbrace{\left(x_1^{(1)} - x_1^{(0)}\right)}_{\Delta x_1^{(0)1}} \\
&+ \frac{\partial W_1\left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}\right)}{\partial x_2} \underbrace{\left(x_2^{(1)} - x_2^{(0)}\right)}_{\Delta x_2^{(0)1}} + \frac{\partial W_1\left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}\right)}{\partial x_3} \underbrace{\left(x_3^{(1)} - x_3^{(0)}\right)}_{\Delta x_3^{(0)1}} = 0 \\
&W_2\left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}\right) + \frac{\partial W_2\left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}\right)}{\partial x_1} \left(x_1^{(1)} - x_1^{(0)}\right) \\
&+ \frac{\partial W_2\left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}\right)}{\partial x_2} \left(x_2^{(1)} - x_2^{(0)}\right) + \frac{\partial W_2\left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}\right)}{\partial x_3} \left(x_3^{(1)} - x_3^{(0)}\right) = 0 \\
&W_3\left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}\right) + \frac{\partial W_3\left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}\right)}{\partial x_1} \left(x_1^{(1)} - x_1^{(0)}\right) \\
&+ \frac{\partial W_3\left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}\right)}{\partial x_2} \left(x_2^{(1)} - x_2^{(0)}\right) + \frac{\partial W_3\left(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}\right)}{\partial x_3} \left(x_3^{(1)} - x_3^{(0)}\right) = 0
\end{aligned}
\tag{18}
$$

Суть метода состоит в том, что нелинейная система уравнений заменяется линейной с использованием частных производных. Полученная система решается относительно приращений аргумента $\Delta x_1, \Delta x_2, \Delta x_3$.

В матричном виде (19):

$$W_1\left(X^{(0)}\right)+\frac{\partial W\left(X^{(0)}\right)}{\partial X}\underbrace{\left(X^{(1)}-X^{(0)}\right)}_{\Delta X}=0 \tag{19}$$

В этом выражении (20)

$$\frac{\partial W\left(X\right)}{\partial X}=\begin{bmatrix} \frac{\partial w_1}{\partial x_1} & \frac{\partial w_1}{\partial x_2} & \frac{\partial w_1}{\partial x_3} \\ \frac{\partial w_2}{\partial x_1} & \frac{\partial w_2}{\partial x_2} & \frac{\partial w_2}{\partial x_3} \\ \frac{\partial w_3}{\partial x_1} & \frac{\partial w_3}{\partial x_2} & \frac{\partial w_3}{\partial x_3} \end{bmatrix} \tag{20}$$

- матрица частных производных функции W_i на каждом шаге итерации - матрица Якоби.

Матрица $\Delta X^1()$ получила название матрицы поправок. $\Delta X^1=\begin{bmatrix} x_1^{(1)}-x_1^{(0)} \\ x_2^{(1)}-x_2^{(0)} \\ x_3^{(1)}-x_3^{(0)} \end{bmatrix}$

Если начальные приближения $x_1^{(0)}, x_2^{(0)}, x_3^{(0)}$ заданы, то матричное выражение можно записать в виде (21):

$$\frac{dW(X^0)}{dX}\Delta X^{(1)}=-W(X^{(0)}) \tag{21}$$

Учитывая, что вектор $X^{(0)}$ - задан, получим, что коэффициенты матрицы Якоби оказываются числовыми значениями, и последнее выражение представляет собой обыкновенную систему уравнений, которая разрешается относительно $\Delta X^{(1)}$

Если матрица Якоби не вырождена (то есть ее определитель не равен нулю), то решение может быть получено, в том числе, и по методу Гаусса. После этого можно найти первое приближение: $X^{(1)}=X^{(0)}+\Delta X^{(1)}$

Таким образом, каждый шаг итерационного процесса представляет собой решение линейной системы уравнений $W\left(X^{(i)}\right)+\frac{\partial W\left(X^{(i)}\right)}{\partial X}\left(X^{(i+1)}-X^{(i)}\right)=0$

или $\dfrac{\partial W\left(X^{(i)}\right)}{\partial X}\Delta X^{(i+1)} = -W\left(X^{(i)}\right)$

Далее определяются следующие приближения неизвестных. Решение ведется до тех пор, пока не выполнится условие (22):

$$\left|W\left(x^{(i)}\right)\right| \le \varepsilon \tag{22}$$

Метод Ньютона использует на каждом шаге решение линейной системы уравнений, получаемой с использованием частных производных в процессе линеаризации, что создает более целенаправленное движение к получению решения и сокращает количество итераций, необходимое для получения решения с заданной точностью.

2.4.5 Применение метода Ньютона для решения уравнений узловых напряжений

Пусть имеется сеть из трех узлов. Запишем для такой сети уравнения узловых напряжений, считая узел 3 балансным (23)

$$\begin{aligned} \underline{Y}_{11}\underline{U}_1 + \underline{Y}_{12}\underline{U}_2 &= \frac{\overset{*}{\underline{S}}_1}{\underline{\underline{U}}_1^{*}} - \underline{Y}_{13}\underline{U}_3 \\ \underline{Y}_{21}\underline{U}_1 + \underline{Y}_{22}\underline{U}_2 &= \frac{\overset{*}{\underline{S}}_2}{\underline{\underline{U}}_2^{*}} - \underline{Y}_{23}\underline{U}_3 \end{aligned} \tag{23}$$

Будем считать, что мощности имеют постоянную величину, и вследствие этого система является нелинейной (24). Придадим уравнениям вид функций, зависящих от напряжений U_1, U_2 [63, 64, 65]:

$$\begin{cases} W_1(\underline{U}_1, \underline{U}_2) = \underline{Y}_{11}\underline{U}_1 + \underline{Y}_{12}\underline{U}_2 - \dfrac{\overset{*}{\underline{S}}_1}{\underline{\underline{U}}_1^{*}} + \underline{Y}_{13}\underline{U}_3 \\ W_2(\underline{U}_1, \underline{U}_2) = \underline{Y}_{21}\underline{U}_1 + \underline{Y}_{22}\underline{U}_2 - \dfrac{\overset{*}{\underline{S}}_2}{\underline{\underline{U}}_2^{*}} + \underline{Y}_{23}\underline{U}_3 \end{cases} \tag{24}$$

или на i - ом шаге итераций (25):

$$W_1(\underline{U}^{(i)}) = \begin{bmatrix} \underline{Y}_{11}\underline{U}_1^{(i)} + \underline{Y}_{12}\underline{U}_2^{(i)} - \dfrac{\overset{*}{\underline{S}}_1}{\overset{*}{\underline{U}}_1^{(i)}} + \underline{Y}_{13}\underline{U}_3 \\ \underline{Y}_{21}\underline{U}_1^{(i)} + \underline{Y}_{22}\underline{U}_2^{(i)} - \dfrac{\overset{*}{\underline{S}}_2}{\overset{*}{\underline{U}}_2^{(i)}} + \underline{Y}_{23}\underline{U}_3 \end{bmatrix} \tag{25}$$

Составим матрицу Якоби (26):

$$\frac{\partial W(\underline{U}^{(i)})}{\partial \underline{U}} = \begin{bmatrix} \underline{Y}_{11} + \dfrac{\overset{*}{\underline{S}}_1}{\left(\overset{*}{\underline{U}}_1^{(i)}\right)^2} & \underline{Y}_{12} \\ \underline{Y}_{21} & \underline{Y}_{22} + \dfrac{\overset{*}{\underline{S}}_2}{\left(\overset{*}{\underline{U}}_2^{(i)}\right)^2} \end{bmatrix} \tag{26}$$

Как видно из полученной матрицы Якоби, в этой матрице диагональные элементы зависят от искомых переменных. Этим она отличается от матрицы коэффициентов линейных уравнений. Анализ этой матрицы показывает, что структура ее, за исключением диагональных элементов, остается такой же, как и структура матрицы коэффициентов линейной системы уравнений узловых напряжений. Это позволяет использовать в методе Ньютона все способы оптимизации решения, применяемые для расчета режимов сложных энергосистем (учет разряженности, симметрия матрицы коэффициентов и др.). С другой стороны, для решения линеаризованных уравнений методом Ньютона не эффективно использование обратной матрицы, так как из-за зависимости диагональных элементов от значений переменных на каждом шаге итерации необходимо было бы обращать матрицу коэффициентов, что значительно уступает по эффективности методу Гаусса с обратным ходом.

Систему нелинейных уравнений можно записать в следующем виде (27):

$$\frac{\partial W(U^{(i)})}{\partial U}\Delta U^{(i+1)} = -W(U^{(i)}) \tag{27}$$

Тогда фактическое решение может быть записано следующим образом (28):

$$\begin{bmatrix} \underline{U}_1^{(i+1)} \\ \underline{U}_2^{(i+1)} \end{bmatrix} = \begin{bmatrix} \underline{U}_1^{(i)} + \Delta\underline{U}_1^{(i+1)} \\ \underline{U}_2^{(i)} + \Delta\underline{U}_2^{(i+1)} \end{bmatrix} \tag{28}$$

Итерационный процесс прекращается при выполнении условия (29):

$$\left|W\left(x^{(i)}\right)\right| \le \varepsilon \tag{29}$$

2.5 Выводы

1. Развита блок схема адаптивной системы схемотехнического проектирования, основные компоненты которой: блок выбора моделей компонентов, блок формирования уравнений и блок адаптивных процедур анализа. Обобщены и классифицированы методы адаптации.

2. На основании обзора адаптивных процедур моделирования электронных цепей сделан вывод об актуальности и необходимости развития методов, повышающих эффективность вычислений не только за счет снижения количества вычислений, но и за счет рациональной организации блока формирования системы уравнений.

3. Предложена адаптивная модель двухполюсного элемента, пригодная для параметрической адаптации к различным режимам работы и компонентам схемы.

Предложенная модель универсальна, поскольку позволяет замещать нелинейные, реактивные, ключевые элементы и другие компоненты схем.

4. На основании адаптивной схемной модели предложен адаптивный метод моделирования электронных схем, позволяющий повысить эффективность вычислений и формирования уравнений и названный методом последовательного частичного A.U -разложения.

ГЛАВА 3. ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. ПРИНЦИПЫ РАБОТЫ, СПРАВКА ПО ЭКСПЛУАТАЦИИ ПРОГРАММЫ

Несмотря на многочисленное использование цифровой электроники во многих прикладных областях, аналоговые электронные устройства, как и раньше, остаются востребованными, а в некоторых случаях и необходимыми. От специалистов в сфере электроники требуются профессиональные знания процессов, протекающих при работе аналоговых электронных схем, потому что именно эти процессы лежат в основе работы электронных устройств любого типа, в том числе и цифровых. Построение математических моделей электронных компонентов в сочетании с развитием вычислительной техники и созданием программ, выполняющих моделирование работы электронной схемы в целом, позволяет анализировать процессы, происходящие в аналоговых схемах максимально эффективно. В таких программах визуальная среда проектирования, в которой несложными средствами создается принципиальная схема электронного устройства, органично сочетается с достаточно сложным математическим аппаратом, используемым в процессе моделирования и анализа работы этой схемы.

Особую актуальность приобретают программные продукты, не требующие платы за покупку самого ПО и лицензии к нему. Они называются свободным или открытым программным обеспечением.

Такие программы не просто более удобны или доступны, зачастую они бывают единственным возможным источником обучения студентов или молодых специалистов. Полнофункциональное программное обеспечение обеспечивает инженера инструментом изучения электроники, с помощью которого, он сможет учиться анализировать алгоритмы, используемые в программе модели, и вообще всю работу программы. Разобравшись, как реализуется в программе та или иная операция, каждый может усовершенствовать их, наконец, создать свою, лучшую программу. Конечно, чтобы использовать эту возможность, кроме изучаемого предмета, - электроники, необходимо дополнительно хорошо освоить технологию программирования, с помощью которой и создаются любые программы [66, 67].

Разработанное программное обеспечение поможет студенту или инженеру разобраться в дискретных и нелинейных элементах электрической цепи, освоить источники тока, наглядно увидеть работу диодов и многое другое.

Надо сказать, что уже создано немало программ для работы с электронными схемами. Они также имеют в своем активе множество адаптивных алгоритмов. Среди них особое место занимает программное обеспечение Spice.

3.1 Основы работы с программой

Существует достаточно много программ или сред компьютерной разработки электроники; ничего плохого о них сказать нельзя, они развиваются и совершенствуются по мере развития и совершенствования самих компьютеров. Разработчики программного обеспечения стараются включить в состав своих пакетов и такие средства, как программы разводки печатных плат и подготовки их к производству. В последнее время в состав компонентов стараются добавить специализированные микросхемы, как, например, стабилизаторы напряжения или микроконтроллеры. Симуляторы электрических цепей, лежащие в основе таких программ, тоже совершенствуются, а проблемы, существующие в их работе, преодолеваются с применением разных математических и программных решений [68, 69].

Однако во всем многообразии программ для изучения радиоэлектроники с применением компьютеров есть один элемент, на который все чаще обращают внимание многие учебные заведения. Это стоимость программ. Даже с теми скидками, которые предоставляются учебным заведениям, использование компьютеров, а как их не использовать? – использование компьютеров становится слишком дорогостоящим [68, 69].

Но и здесь есть неплохой выход – применение свободно распространяемого программного обеспечения, включая и операционную систему. Все разговоры о трудностях перехода с Windows, а я не думаю, что в учебных заведениях используют другие операционные системы, на любой из дистрибутивов Linux, скорее, надуманы. Программа Ecaasys существует в версиях для всех общеупотребительных операционных систем: Windows, Linux, MacOS. Программа может использоваться и распространяться бесплатно в некоммерческих целях. Все аргументы, если бесплатное, значит плохое, имеют сомнительное происхождение. Достаточно напомнить, сколько неприятностей принесло применение Windows покупателям этой операционной системы [68, 69].

Ecaasys достаточно неприхотливая программа по нынешним временам. Если на компьютер устанавливается Windows 2000, значит, и программа будет работать. Требования к компьютеру в основном определялись тем фактом, что сам проект появился не так давно, когда основная масса компьютеров использовала операционную систему Windows XP, с теми требованиями, которые к компьютеру предъявляла эта операционная система. Хотя симуляция некоторых схем может с трудом проходить и на вполне современном компьютере, но это уже другой вопрос. Таким образом, программу следует установить и опробовать, если вы собираетесь с ней работать. Для симуляции цифровых устройств вам понадобится дополнительно установить еще несколько свободно распространяемых программ. Одной из особенностей программы, которую следует отметить, является наличие в составе базовых компонентов множества таких, которые ориентированы на применение в современных радиотехнических устройствах. Подобные компоненты редко

встречаются в других программах общего применения. Но это никак не влияет на возможности использовать программу в изучении любых других электрических цепей [68, 69].

Как многие проекты свободного программного обеспечения, Ecaasys постоянно развивается, и каждая новая версия доступна для свободного использования, достаточно иметь /доступ к Интернету, чтобы получить новую версию. И подобно всем программам с открытым кодом, Ecaasys для серьезных пользователей, умеющих программировать, дает возможность полностью переделать всю программу под свои нужды или интересы [68, 69].

В отличие от многих аналогов программа полностью русифицирована при участии ее создателей, что избавляет от случайных ошибок или появления проблем, связанных с пользовательской русификации программы. В разделе «Справка» есть прекрасный раздел быстрого начала работы с Ecaasys. Опытным пользователем его будет достаточно. На сайте проекта много документов, статей и примеров, восполняющих то, чего нет в справочной системе. И, наконец, эта книга для начинающих работать с Ecaasys должны помочь всем, кого интересуют программы этого направления, познакомиться с программой [68, 69].

3.1.1 Общие сведения

При первом запуске Ecaasys создает папку ".Ecaasys" в вашем домашнем каталоге (если вы работаете в Windows-версии, то эта папка создается в директории “Document and Settings\Ваш_Логин” системного диска). Каждый файл проекта сохраняется в этой папке или в одной из ее подпапок. После загрузки Ecaasys показывается главное окно, которое выглядит примерно как на рисунок 1.

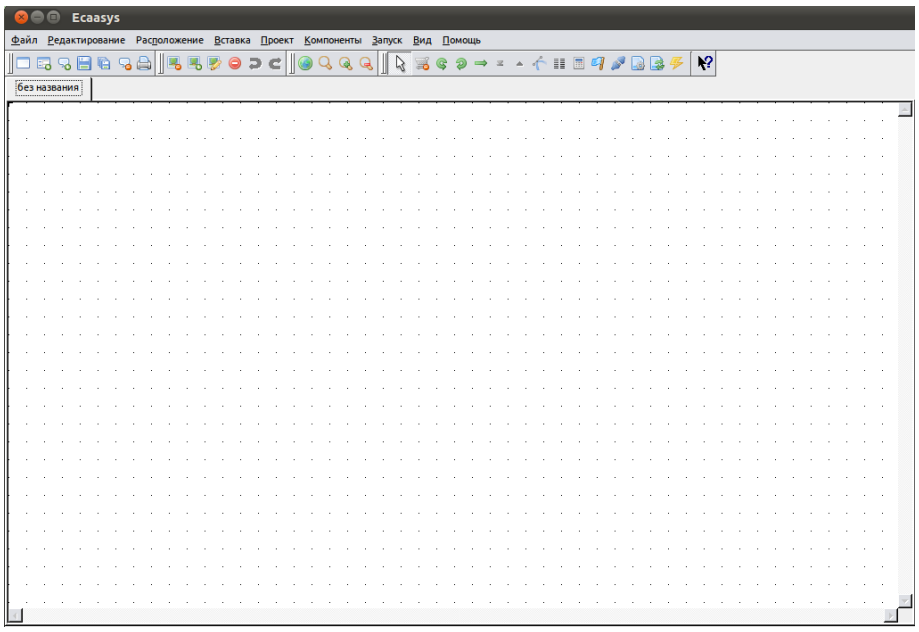

Рисунок 5 - Главное окно Ecaasys

С правой стороны расположена рабочая область, в которой содержатся схемы, документы показа данных и т.д. С помощью вкладок над этой областью можно быстро переключиться на любой документ, открытый в данный момент. С левой стороны главного окна Ecaasys находится информационно-командная область, содержание которой зависит от состояния вкладок, расположенных с левой стороны: «Компоненты», «Содержание» и «Проекты». В верхней части этой области находятся управляющие клавиши «Создать», «Открыть» и «Удалить», выполняющие соответствующие действия с проектами.

3.1.2 Создание проекта

После запуска Ecaasys активируется вкладка «Проекты». Если вы запустили программу в первый раз, эта область будет пуста, поскольку еще не создано ни одного проекта (На рисунке 5 в этой области видны четыре проекта с именами: 1, 2, 3 и 4). Работа начинается с создания проекта и присвоения ему имени. Нажатие на вкладку «Создать» вызывает диалоговое окно рисунок 6. Введите имя для вашего первого проекта, например, «firstProject» и нажмите кнопку «Создать». Ecaasys создает в вашей домашней папке папку проекта с соответствующим названием «firstProject_prj».

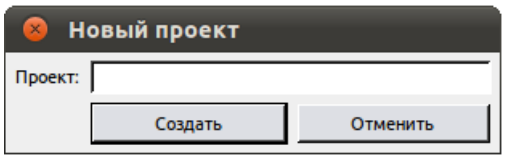

Рисунок 6 - Диалоговое окно создания проекта

Каждый файл схемы, диаграммы или описания, принадлежащий этому новому проекту, будет сохранен в этой папке. Новый проект немедленно открывается (это можно прочитать в заголовке окна программы), автоматически генерируя первый документ-схему без названия. Вкладки информационно-командной области при этом переключаются в режим «Содержание», отображая в виде древообразной схемы различные аспекты содержимого проекта.

Здесь есть, например, такие «ветви» как VHDL и Verilog - языки программирования, широко используемые для описания электронных схем. Программа позволяет сохранить модель разработанной схемы в виде файла с описанием на одном из указанных языков или выполнить обратное преобразование, прочитав файл с таким описанием, отобразить его в виде схемы в окне программы. Языки VHDL и Verilog - стали стандартами описания электронных схем, поэтому в программу включена их поддержка. Однако при создании исследуемой схемы и выполнении операций моделирования от пользователя не требуется знание указанных языков программирования. Вся

работа основана на визуальном проектировании и автоматическом анализе схемы.

Начать работу по созданию схемы рекомендуется с операции сохранения документа, во время которой ему присваивается конкретное имя. Хотя создание схемы можно начинать и в документе без названия, при первой попытке выполнения моделирования, программа потребует ввести имя для обрабатываемого документа.

Поэтому лучше это имя задать сразу. Для выполнения этой операции можно нажать пиктограмму дискеты на панели инструментов (или в меню «Файл» выбрать пункт «Сохранить» или «Сохранить как...», наконец, можно воспользоваться комбинацией клавиш Ctrl+S). В появившемся диалоговом окне (рисунок 7) документу присваивается имя, например, «MyfirstSchem», а нажатие кнопки «Сохранить» приводит к созданию в папке проекта файла документа MyfirstSchem.sch.

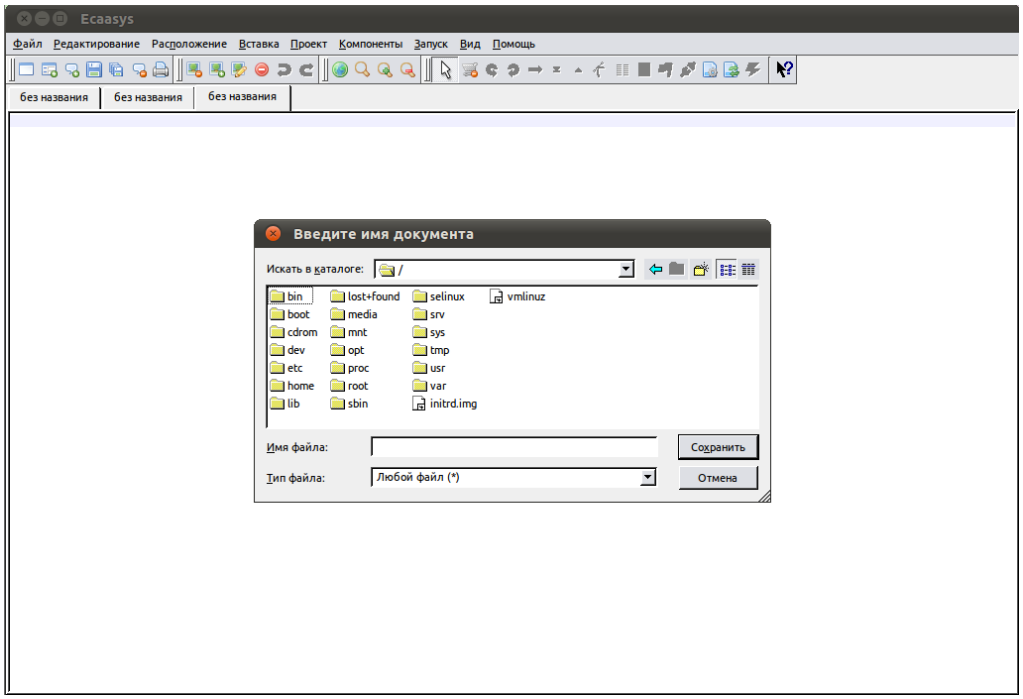

Рисунок 7 - Диалог сохранения документа

После этого имя документа-схемы появляется на вкладке рабочей области и в ветви «Схемы».

Все созданные в одном проекте документы, можно увидеть в окне программы в ветви «Схемы» на вкладке «Содержание» с левой стороны и в рабочей области, где активен один из документов, остальные показаны вкладками. Каждый проект может содержать как один документ, так и многие десятки.

Заметим, что документ «без названия» не отображается в ветви «Схемы», он как бы «не существует».

3.1.3 Компоненты Ecaasys

Для создания модели электронной схемы нужно перейти на вкладку «Компоненты». В верхней части информационно-командной области при этом появляется выпадающее меню, по умолчанию отображающее первую строку: «Дискретные компоненты».

Ниже отображены сами компоненты соответствующего класса.

Кроме «Дискретных компонентов», в число которых входят все пассивные элементы электрических цепей и некоторые специальные компоненты, например, «Реле», «Гиратор», и т.д., в меню компонентов девять пунктов:

- Источники
- Измерители
- Линии передачи данных
- Нелинейные компоненты
- Цифровые компоненты
- Файловые компоненты
- Виды моделирования
- Диаграммы
- Рисунки

Активные элементы электрических цепей, такие как диоды и транзисторы, находятся в разделе «Нелинейные компоненты». В разделе «Источники» размещены источники постоянного и переменного тока и напряжения, а раздел «Диаграммы» позволяет выбрать способ отображения результатов моделирования.

Программа «Ecaasys» построена таким образом, что может выполнять различные виды моделирования электронных схем - на постоянном и переменном токе, с разверткой параметра, моделирование процессов во времени и т.д. Для выполнения того или другого вида моделирования необходимо явно указать этот вид моделирования на принципиальной схеме. Такое указание выполняется перетаскиванием кубика соответствующего вида моделирования из раздела «Виды моделирования» на рабочую область. Если этого не сделать и не указать вид моделирования, то при попытке выполнения моделирования будет сгенерирована ошибка. Одновременное же выполнение нескольких видов моделирования для одной и той же схемы не только допускается, но и используется чаще всего.

3.1.4 Создание схем

Рисование принципиальной схемы в рабочем окне заключается в перетаскивании соответствующих компонентов из левого фрейма в рабочую область и соединении их между собой. Для выделения компонента достаточно однократно щелкнуть на нем левой клавишей мышки. Держать левую клавишу нажатой во время перетаскивания не обязательно. После этого при перенесении мышью указателя в рабочую область он сопровождается штриховым

изображением выделенного компонента, а следующий щелчок левой клавишей мышки вставляет компонент в рабочую область. Если до вставки компонента выполнить нажатие правой клавиши мышки, то происходит поворот компонента на 90°. Отметим, что, вставив компонент в рабочую область, мы не отменяем выделение выбранного компонента. Однократно выбрав нужный компонент, мы можем продолжать вставлять в рабочую область столько его копий, сколько необходимо для построения схемы. Снять выделение компонента можно клавишей «Esc» или выбором нового компонента в информационно-командной области.

Если же выполнить перетаскивание компонента с нажатой левой клавишей мышки, то при отпускании клавиши в рабочей области выделение компонента сразу отменяется, таким способом можно пользоваться для вставки в схему единственного компонента соответствующего типа.

Количество компонентов разных типов в программе Ecaasys очень велико, более сотни, это одно из важных преимуществ программы перед аналогичными по назначению продуктами других разработчиков. Благодаря этому, программа Ecaasys позволяет моделировать такие процессы, которые в других программах остаются, что называется «за кадром». Так, например, здесь наряду с обычным набором источников тока и напряжения, есть модели источников шумовых сигналов, в качестве компонентов схем могут использоваться линии связи различных видов и т.д.

Соединение компонентов производится при переключении в режим рисования соединений. Такое переключение выполняется нажатием клавиш Ctrl+E или выбором на панели инструментов значка, символизирующего проводник, рисунок 8.

Рисунок 8 - Пиктограмма проводника на панели инструментов

Отметим, что переключение в режим рисования соединительного проводника может происходить автоматически при задании опции «Начать проводку при нажатии кнопки на открытом узле» в меню «Файл -> Настройки программы».

Начало и конец проводника отмечаются нажатием левой клавиши мышки. Трассировка проводника осуществляется автоматически, линия, показывающая путь прохождения будущего проводника следует за курсором после первого нажатия и отображается пунктиром до тех пор, пока пользователь повторно не нажмет левую клавишу. Во время построения соединения линия имеет вид ломанной, состоящей из двух прямых участков, соединенных под прямым углом. Щелчок правой клавишей мышки изменяет вид ломанной на зеркально противоположный. Иногда возникает потребность построить соединение элементов с помощью ломанной линии, имеющей более сложную структуру,

чем два отрезка, соединенных под прямым углом. Для этого при проведении трассы проводника необходимо выполнить несколько «промежуточных остановок» при проведении проводника от одного компонента к другому, завершая рисование соединения нажатием левой клавиши мышки и, повторно выполнив нажатие, начиная новое соединение из той же точки. Никаких соединительных узлов на проводнике при таком изменении траектории его прохождения не образуется. Для проведения соединений «от» или «в» произвольную точку на уже существующем проводнике (точку без узла), необходимо переключиться в режим рисования проводника явно.

Для удаления отдельных компонентов или проводников их необходимо предварительно выделить щелчком левой клавиши мышки. Нажатие после этого клавиши «Del» приведет к удалению компонента или проводника. Более тонкое управление удалением участков проводников можно выполнить, переключив курсор в режим удаления. Сделать это можно щелкнув левой клавишей мышки пиктограмму красного креста на панели управления, или щелкнув на рабочем поле левой клавишей мышки при нажатой клавише «Del».

В режиме удаления при наведении курсора на проводник и щелчке левой клавишей мышки удаляется только прямолинейный участок этого проводника, а не весь проводник.

При перенесении компонентов в рабочую область из информационно-командной области их свойства соответствуют идеальным компонентам. Свойства любого компонента можно изменить в диалоговом окне, которое открывается при двойном щелчке на компоненте.

3.2 Выводы

1. Описываются основы созданного программного обеспечения, даются основные понятия о программе Ecaasys. Изложены принципы работы программы, а также дается справка по эксплуатации программного обеспечения.
2. Рассказано об основных элементах программы
3. В работе подробным образом описывается создание электрической схемы с помощью данного программного обеспечения, что значительно облегчит работу начинающему пользователю.

ГЛАВА 4. ОПИСАНИЕ СТРУКТУРЫ СОЗДАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ДЕМОНСТРАЦИЯ РАЗРАБОТАННОГО МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ПРИМЕРАХ

Приложение Ecaasys основано на открытом исходном коде и, подобно аналогичным редакторам, обладает всеми необходимыми для модификации схем средствами. Для удобства имеется поддержка «горячих клавиш» и выпадающие меню. При работе со сложными схемами предлагается использование подсхем, позволяющих отделить часть основной схемы в виде блоков. Кроме того, программное обеспечение имеет собственный текстовый редактор, приложения для расчета фильтров и согласованных цепей, калькуляторы линий и синтеза аттенюаторов. Присутствует возможность оформления чертежа путем добавления рамки и штампа. К схемам можно подключать пользовательские уравнения, причем в формулах могут быть использованы все доступные функции данного софта или их сочетания.

Ecaasys имеет обширную базу современных компонентов, разбитых на группы: дискретные (резисторы, конденсаторы и т.д.), нелинейные (в основном транзисторы и диоды), цифровые (базовые цифровые устройства и логические вентили) и некоторые другие (источники, измерители, линии передачи данных). Особыми группами представлены виды моделирования, рисунки и диаграммы. Каждый элемент имеет собственное диалоговое окно свойств, которое может быть подвергнуто редактированию. Если в библиотеке нет необходимого компонента, то его можно добавить туда при наличии SPICE-модели.

4.1 Моделирование на постоянном токе

В электротехнике рассматривается устройство и принцип действия основных электротехнических устройств, используемых в быту и промышленности. Чтобы электротехническое устройство работало, должна быть создана электрическая цепь, задача которой передать электрическую энергию этому устройству и обеспечить ему требуемый режим работы.

Электрической цепью называется совокупность устройств и объектов, образующих путь для электрического тока, электромагнитные процессы в которых могут быть описаны с помощью понятий об электрическом токе, ЭДС (электродвижущая сила) и электрическом напряжении.

Для анализа и расчета электрическая цепь графически представляется в виде электрической схемы, содержащей условные обозначения ее элементов и способы их соединения.

4.1.1 Простое моделирование

После того, как схема нарисована, можно выполнять моделирование ее работы. Программа Ecaasys имеет в своем арсенале восемь видов моделирования. Если учесть, что могут использоваться и комбинации этих

видов, число вариантов моделирования возрастает до десятков и, по задумке авторов программы, должно удовлетворить самым изощренным запросам разработчика [70-74].

Рассмотрим выполнение самого простого моделирования: моделирования на постоянном токе. Этот вид моделирования присутствует практически в любой схеме, использующей источники питания. Для примера выберем простейшую схему - делителя напряжения. Перенесем в рабочую область два резистора из раздела «Дискретные компоненты», затем раскроем меню в верхней части левого фрейма и, выбрав пункт «Источники», добавим источник постоянного напряжения. Изменить их можно через диалоговое окно свойств компонента, которое открывается при двойном щелчке на компоненте левой клавишей мышки. Это же окно можно вызвать, выбрав пункт «Изменить свойства» в контекстном меню, появляющемся при нажатии на компоненте правой клавиши мышки. Наконец, можно отредактировать номинальное значение компонента, щелкнув левой клавишей мышки в поле отображения значения на схеме.

Для задания величины параметра компонента можно использовать три типа записи:

- Стандартную
- Научную
- Инженерную

При использовании условных обозначений также «работают» инженерные приставки. Т.е. можно указать для сопротивления величину номинала 1m без указания единиц измерения, а можно 1 mOhm, и то и другое определение дадут одинаковый результат. Обратите внимание на совпадение обозначения длины в метрах и приставки «милли». Дело в том, что длину нельзя задать в метрах, могут использоваться только производные от метра единицы, например, - mm - миллиметр. На самом деле, при разработке микрополосковых линий в СВЧ устройствах их размеры далеки от метровых, поэтому длина всегда измеряется достаточно малыми величинами.

В программе нет жестких требований по обязательному применению только одного способа задания номинальных значений для всех элементов схемы. В принципе, использование для разных элементов одной и той же схемы разных способов задания параметров никак не отразится на результате моделирования. Единообразие в этом случае - это скорее требование правильной организации труда разработчика. Оно поможет легче работать со схемой.

Все виды аналогового моделирования, в том числе и моделирование на постоянном токе, требуют наличия в схеме точки для отсчета потенциала - точки заземления или просто «Земли». Элемент «Земля» присутствует среди дискретных компонентов, он отображается и на панели инструментов в верхней части окна программы, кроме того, для активации этого элемента можно использовать комбинацию клавиш Ctrl+G. Как показано на рисунке 9, мы

добавили на схему этот элемент и подключили его к отрицательному выводу источника питания.

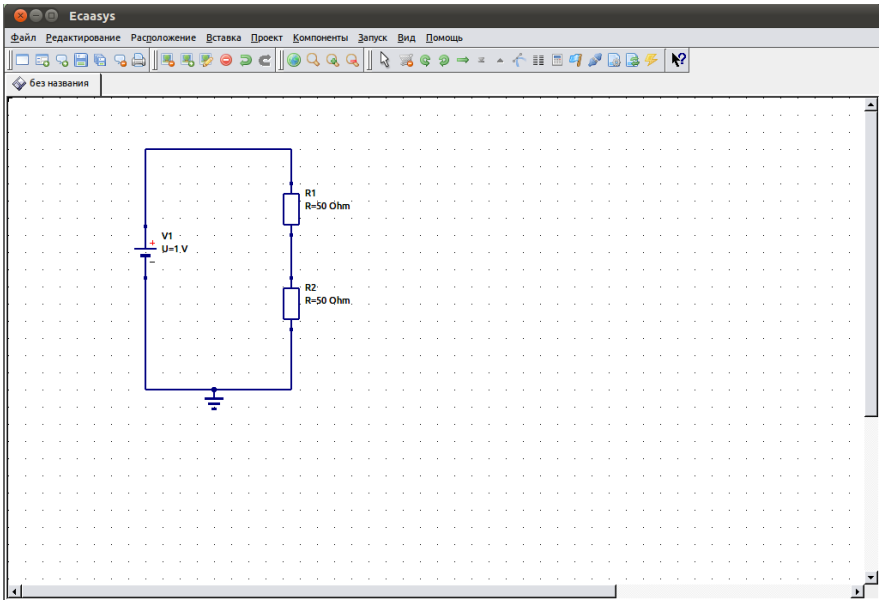

Рисунок 9 - Исследование делителя напряжения на постоянном токе

Задача моделирования на постоянном токе сводится к тому, что мы хотим определить потенциалы в определенных точках схемы и токи в ее ветвях. Для указания программе точек, для которых выполняется расчет потенциала, используется элемент «Метка». Для активизации этого элемента можно также использовать комбинацию клавиш Ctrl+L.

После установки метки в конкретную точку схемы, открывается диалоговое окно, в котором задается имя для создаваемой метки.

Заключительным действием перед запуском процесса моделирования является перенесение на рабочую область кубика с указанием вида моделирования. Вид моделирования выбирается из левого фрейма при переходе на закладке «Компоненты» в меню «Виды моделирования».

Моделирование завершится ошибкой, если на рабочем поле не будет задан вид моделирования (задается он перенесением на рабочее поле соответствующего кубика).

Процесс моделирования запускается нажатием левой клавишей мышки на пиктограмме шестеренки на панели инструментов, рисунок 10, или нажатием функциональной клавиши F2.

Рисунок 10 - Пиктограмма запуска моделирования

В процессе моделирования отображается окно рисунок 11, в которое выводится информация о исходных данных для расчета и этапах обработки схемы в виде горизонтальной бегущей полоски - прогресс-бара.

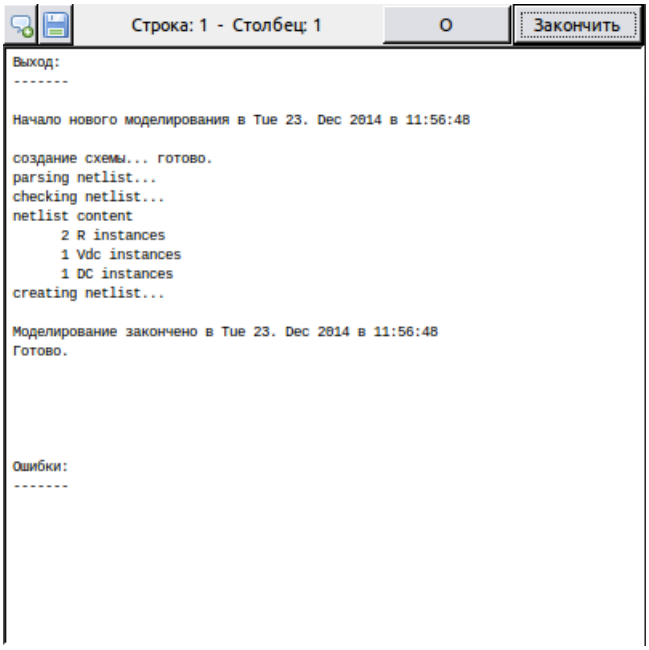

Рисунок 11 - Окно обработки данных

Настройки программы по умолчанию таковы, что после завершения расчета автоматически открывается окно отображения диаграмм, а окно, показанное на рисунок 11, пропадает. Для простых схем, подобных той, что мы выбрали для примера, время расчета не велико и мы видим только промелькнувшее на экране окно обработки. Изменить поведение программы можно, сняв галочку «Открыть просмотр данных после моделирования» в диалоговом окне «Файл - Настройки документа».

В этом случае окно рисунок 11 останется на экране и мы сможем просмотреть этапы моделирования, выполняемые программой. Конечно, специфическая информация, выводимая в это окно будет больше понятна программистам, но иногда она может быть полезна при анализе ошибок, допущенных в схеме. Впрочем, программа всегда записывает служебный лог-файл с информацией о результатах последнего моделирования. Вызвать встроенный редактор с отображением этого файла можно нажатием функциональной клавиши F5.

В окне отображения диаграмм после окончания моделирования ничего не отображается. Это может вызвать недоумение - проведено моделирование и где же результаты? Дело в том, что программа, получив в ходе моделирования некие расчетные величины, может представить их по-разному. Для того, чтобы увидеть эти результаты, нужно указать программе, как она должна их отобразить. В открывающемся по- умолчанию после выполнения

моделирования меню «Диаграммы» в левом фрейме находятся 11 типов диаграмм. Выберем табличное представление и перенесем диаграмму этого типа в рабочую область на окно просмотра данных. Окно просмотра данных и файл с именем MyfirstSchem.dpl создаются автоматически в процессе моделирования. Программа построена так, что результаты моделирования отделяются от исходной схемы и даже хранятся в отдельном файле. Но мы можем, если захотим, вставить диаграмму и на рабочую область в поле отображения схемы.

Как только мы сделаем щелчок левой клавишей мышки на рабочей области, открывается диалоговое окно свойств создаваемой диаграммы, рисунок 12.

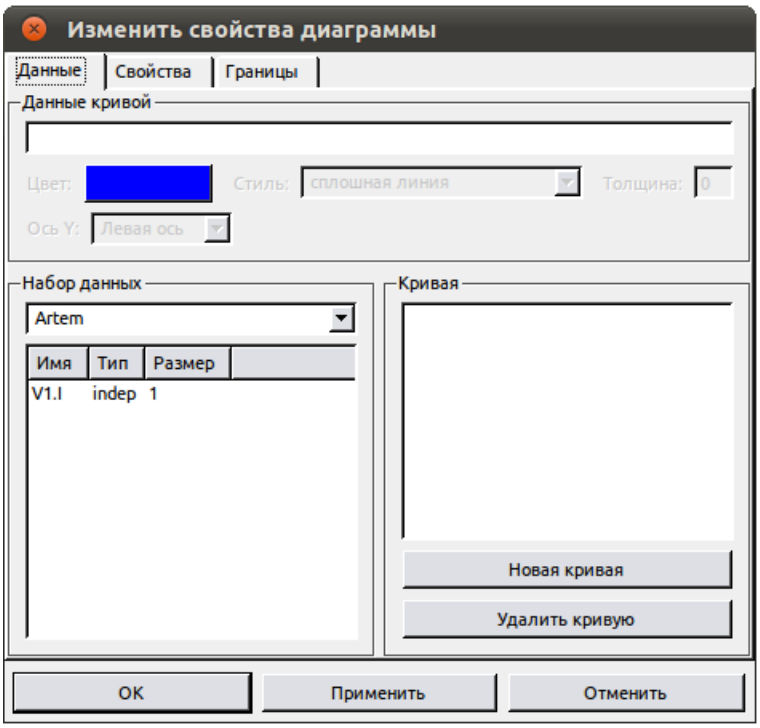

Рисунок 12 - Диалоговое окно свойств диаграммы

В этом окне необходимо выбрать переменную, значение которой было рассчитано программой и которую мы хотим отобразить на диаграмме. Все рассчитанные переменные отображаются в окне «Набор данных». Двойной щелчок на любой из переменных приводит к тому, что эта переменная появляется в окне «Кривая» и, после нажатия клавиши «ОК», будет отображена на диаграмме. Мы можем перенести в окно «Кривая» одну из рассчитанных переменных, несколько переменных или все, но отображены на диаграмме могут быть только те переменные, которые видны в окне «Набор данных».

Создавая схему электронной цепи, мы формируем и тот набор переменных, который будет рассчитан программой. Так, например, в наборе данных, показанном на рисунок 12 можно видеть потенциал метки «А». Появление этой переменной в наборе данных связано с размещением метки на

схеме. Автоматически программа ничего не рассчитывает, кроме тока, протекающего через источник. Соответствующую переменную можно видеть в наборе данных на рисунок 12. У расчета программой тока источника есть одна особенность, - считается, что ток внутри источника движется от вывода с более высоким потенциалом к выводу с более низким потенциалом. Во внешней по отношению к источнику цепи ток протекает тоже от «плюса» к «минусу», поэтому токи внутри и вне источника оказываются противоположно направленными (на самом деле внутри источника ток движется от «минуса» к «плюсу»). Эту особенность необходимо учитывать при использовании рассчитанного значения тока источника для определения тока во внешней цепи, помножив это значение на (-1).

Для расчета тока в отдельных ветвях схемы или падений напряжения на отдельных элементах, а не относительно нулевого потенциала, можно применить измерительные приборы: амперметр или вольтметр, подключив их к собранной схеме. В этом случае в наборе данных после расчета будут доступны переменные с показаниями приборов [74-79].

Другим, более универсальным способом добавления расчетных переменных, является математическая запись выражений в виде формулы, размещенной в рабочей области. Работа с формулами будет рассмотрена ниже. При выполнении же анализа на постоянном токе часто вполне достаточно знать потенциалы в той или иной точке схемы, а для этого - просто расставить метки в требуемых местах.. Отрицательное значение тока источника связано с упомянутой выше особенностью программы.

4.1.2 Моделирование на постоянном токе с разверткой параметра.

Моделирование с разверткой параметра - один из самых часто используемых приемов моделирования. Построим простую цепь с биполярным транзистором, включенным по схеме с ОЭ, рисунок 13. Выполним анализ зависимости коэффициента усиления по току биполярного транзистора от величины тока коллектора. Так как ток коллектора зависит от тока базы транзистора, то очевидно, что параметром, развертку которого необходимо выполнить, будет ток базы. С учетом этого строим принципиальную схему.

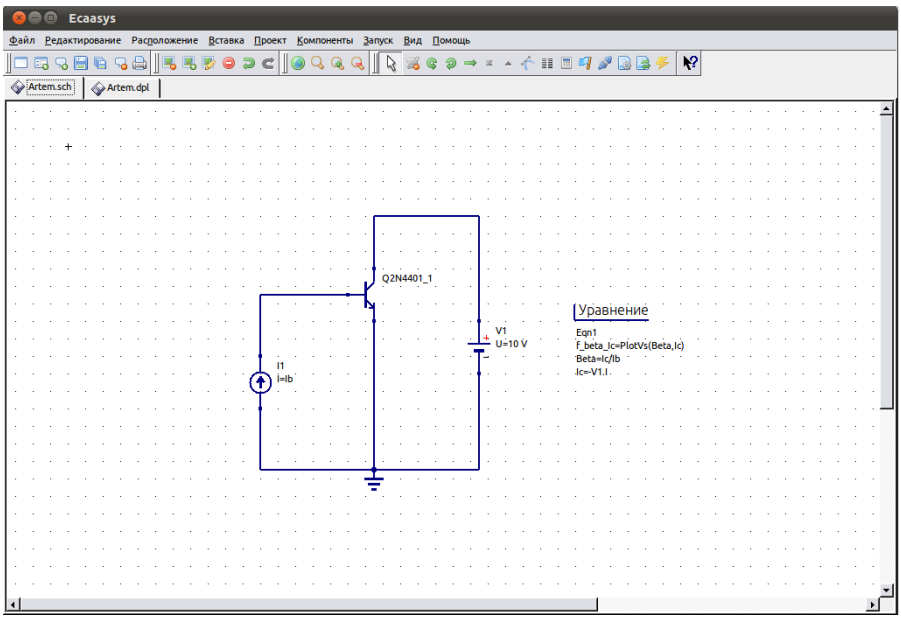

Рисунок 13 - Цепь с нелинейным элементом для выполнения моделирования с разверткой параметра

Для питания базы используем источник постоянного тока, величину этого тока установив в общем виде - Ib (вместо значения по-умолчанию 1мА). Используем модель реального транзистора Q2N4401_1 из библиотеки компонентов (пункт меню «Инструменты/Библиотека компонентов»), для того, чтобы зависимость была более наглядной. Для питания коллекторной цепи транзистора установим источник постоянного напряжения 10 В. После соединения элементов схемы, перенесем на рабочую область кубики, символизирующие моделирование на постоянном токе и развертку параметра. Обратите внимание, что каждый кубик имеет имя. Каждый вид моделирования выполняется, вообще говоря, независимо от других. Чтобы согласовать работу двух используемых видов моделирования, необходимо задать ряд параметров в свойствах моделирования. Вызов окна свойств происходит при двойном щелчке левой клавишей мышки по кубику моделирования или вызовом правой кнопкой мышки контекстного меню и выбором в нем пункта «Изменить свойства». Перечислим изменения, которые необходимо задать в свойствах моделирования с разверткой параметра:

1. Укажем имя моделирования на постоянном токе
2. Зададим параметр развертки
3. Установим логарифмический тип развертки параметра (можно задать линейную развертку или использовать список значений);
4. Определим начальное и конечное значение тока базы, исходя из разумного диапазона значений (ясно, что необходимо ориентироваться на справочные данные транзистора). Так, выбраны, соответственно, 10 наноампер и 10 миллиампер;

Завершается настройка свойств моделирования с разверткой параметра изменением (если необходимо) в нижнем поле формы общего количества

расчетных точек. По-умолчанию в этой строке всего 20 точек. В случае сложной кривой зависимость, построенная по такому количеству точек может быть недостаточно гладкой. В примере выбрано 102 точки, т.к. при этом получается ровно по 17 точек на декаду развертки.

Настройка моделирования завершена, но выполнять его еще рано. Как уже говорилось, программа делает расчет только заданных величин. Нам необходимо построить зависимость коэффициента усиления транзистора по току от величины тока коллектора, но ни та, ни другая переменная не будут рассчитаны, т.к. неизвестно, как их считать. Универсальным способом определения переменных является введение в схему уравнений или блока вычислений, определяющих эти переменные. Для того, чтобы вставить в схему блок вычислений достаточно щелкнуть левой клавишей мышки на пиктограмме уравнения, расположенной на панели управления, рисунок 14.

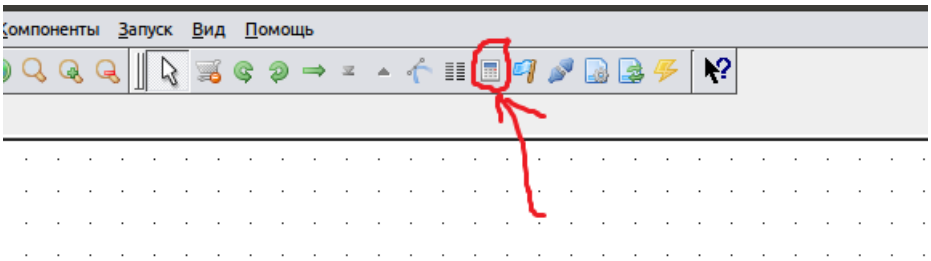

Рисунок 14 - Пиктограмма уравнения на панели управления

Это приводит к переключению программы в режим ввода уравнений в блоке вычислений, что сопровождается изменением формы курсора, - он отображается теперь пунктирным уголком. В режиме ввода уравнений щелчок левой клавишей мышки на свободном месте рабочей области вставляет в схему «пустое» уравнение вида: y = 1.

Каждый блок вычислений может содержать несколько уравнений. Двойной щелчок левой клавишей мышки, либо выбор из контекстного меню, вызываемого правой клавишей мышки, строки «Изменить свойства», открывает окно свойств блока вычислений рисунок 15.

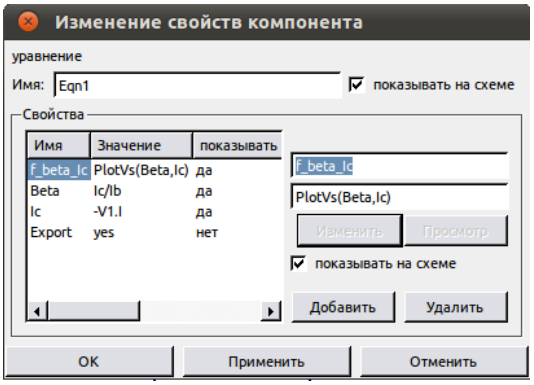

Рисунок 15 - Окно свойств блока вычислений

В этом окне в двух полях справа мы можем изменить имя функции (в верхнем поле, имя функции по умолчанию y) и записать формулу (в нижнем поле, по умолчанию здесь находится единица), по которой выполняется расчет. Знак равенства между правой частью уравнения (верхнее поле) и его левой частью (нижнее поле) не ставится, но подразумевается. Нажатие клавиши «Применить» записывает сделанные изменения и они становятся видны слева в окне свойств блока вычислений. Снова изменив имя функции, и задав новую формулу для ее расчета, с помощью клавиши «Добавить» мы можем увеличивать число уравнений в блоке вычислений. И, наоборот, выбрав в списке слева какую-то строку с уже введенным уравнением, с помощью клавиши «Удалить», можем удалить его. После определения необходимых уравнений в свойствах блока вычислений и нажатия клавиши ОК, все они отображаются в рабочей области ниже значка уравнения в виде набора равенств. В примере, показанном на рисунке 15, блок вычислений содержит три уравнения:

1. Ток коллектора задан равным току источника питания коллекторной цепи с обратным знаком (30):

$$I_c = -V1.I \tag{30}$$

2. Коэффициент усиления транзистора по току определен как отношение тока коллектора к току базы (31):

$$Beta = I_c / I_b \tag{31}$$

3. Функциональная зависимость коэффициента усиления по току от величины тока коллектора (32) определена с помощью встроенной функции $PlotVs()$:

$$f\_beta\_Ic = PlotVs(Beta, I_c) \tag{32}$$

При записи выражений в поле формулы используются обычные символы математических операций:

- сложение;
- вычитание;
- деление;
- умножение;
- остаток от деления;
- возведение в степень.

Программа Ecaasys содержит большое число встроенных функций, как элементарных математических (тригонометрические, логарифмические и т.п.), так и специальных (функции Бесселя, преобразования Фурье, преобразования

параметров четырехполюсника и т.д.). С достаточно полным списком функций и краткой справкой по ним можно ознакомиться в справочной системе программы в разделе «Краткое описание математических функций». Надо отметить, что функции в этом описании перечисляются в случайном порядке, поиск по имени или назначению функции в справочной системе не организован, поэтому пользоваться такой справкой неудобно. В Приложении 2 к данному пособию приведен список функций Ecaasys, сгруппированных по функциональному признаку.

После задания функциональных зависимостей для рассматриваемого примера было выполнено моделирование. Для отображения результатов моделирования выбран декартов график. Окно свойств диаграммы при построении графика показано на рисунок 15. В поле «Набор данных» этого окна отображаются, в том числе, и все функции, определенные в блоке вычислений. Т.к. нам необходимо построить только одну функциональную зависимость, выбираем именно ее (выбор выполняется двойным щелчком левой клавишей мышки по имени выбранной функции). Обратите внимание, что для функциональных зависимостей в колонке «Тип» стоят символы «dep» - сокращение от «dependent», а в следующей колонке показано от какого аргумента зависит эта функция. Независимые переменные отмечены символами «indep» - independent, это просто наборы значений. Таких переменных в нашем примере всего две: ток базы Ib и Versus.0001. Набор значений тока базы получен в результате выполнения моделирования с разверткой параметра - 102 точки, в которых ток базы изменяется по логарифмическому закону от 10 наноампер до 10 миллиампер. Набор «Выборки из..» Versus.0001 создан автоматически функцией $PlotVs()$ и содержит 102 значения тока коллектора, при заданных в наборе I_b токах базы.

Откроем закладку «Свойства» в окне свойств диаграммы, рисунок 16. По умолчанию ось абсцисс графика будет подписана именем Versus.0001, введем в поле «Метка оси X» другое значение - I_c (Можно использовать надписи и на русском языке). Кроме того, установим флажок «Логарифмическая разметка оси X», чтобы более подробно увидеть область наибольших изменений коэффициента усиления по току.

Результирующая зависимость показана на рисунок 16. Линии сетки отображаются на графике, если установлен флажок «Показывать сетку». На вкладке «Границы» можно, отменив автоматическое определение границ графика и шага по осям, назначить эти величины вручную.

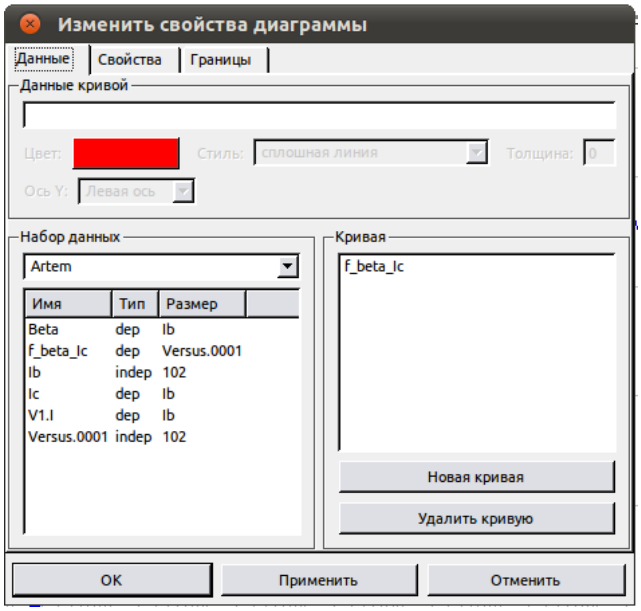

Рисунок 16 - Окно свойств диаграммы для рассматриваемого примера

Для просмотра значений функции в различных точках на ее графике применяются маркеры. Установить маркер на график, можно переключившись в режим установки маркера комбинацией клавиш Ctrl+B или щелчком левой клавишей мышки по символу маркера на панели управления, рисунок 17. Курсор превращается в треугольник. После этого, щелчок левой клавишей мышки на любой токе кривой формирует на графике сноску с изображением значений функции и аргумента, рисунок 18 и рисунок 19.

Выделив щелчком левой клавиши мышки любой из маркеров, точку привязки к графику маркера можно перемещать вдоль кривой клавишами управления движением курсора - стрелками «вправо», «влево».

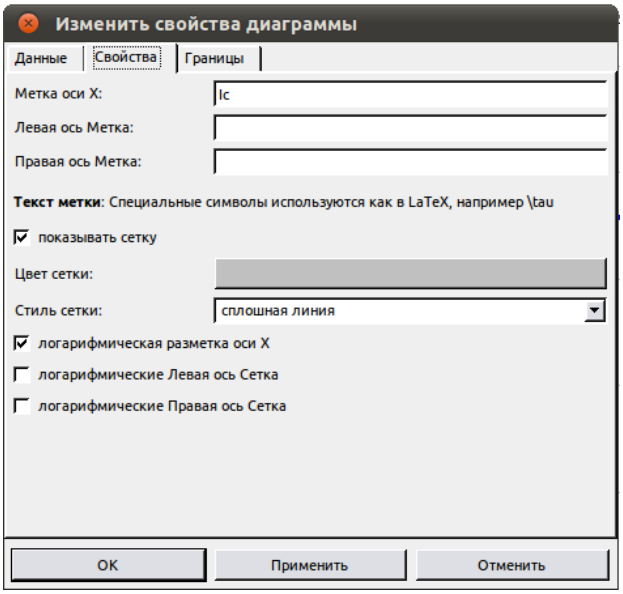

Рисунок 17 - Окно свойств графика функции

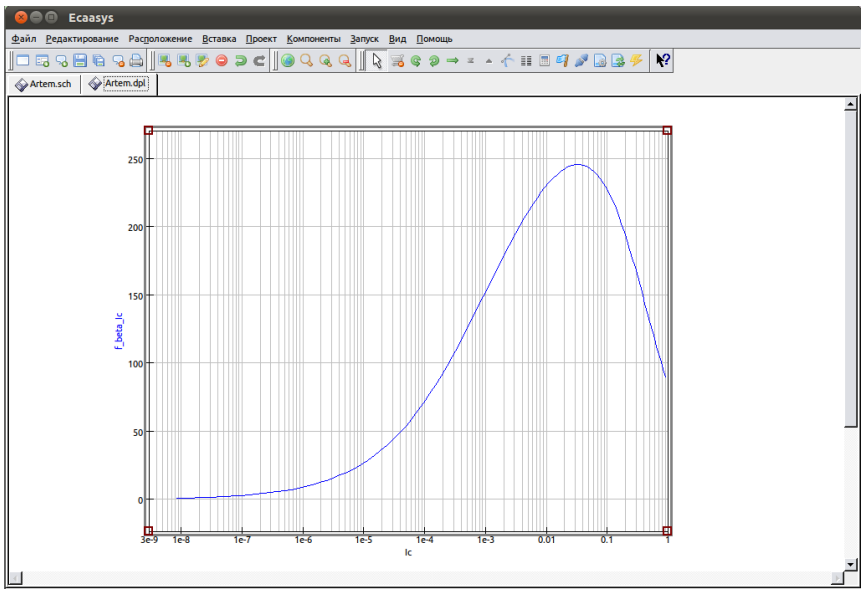

Рисунок 18 - График зависимости коэффициента усиления транзистора по току от величины тока коллектора

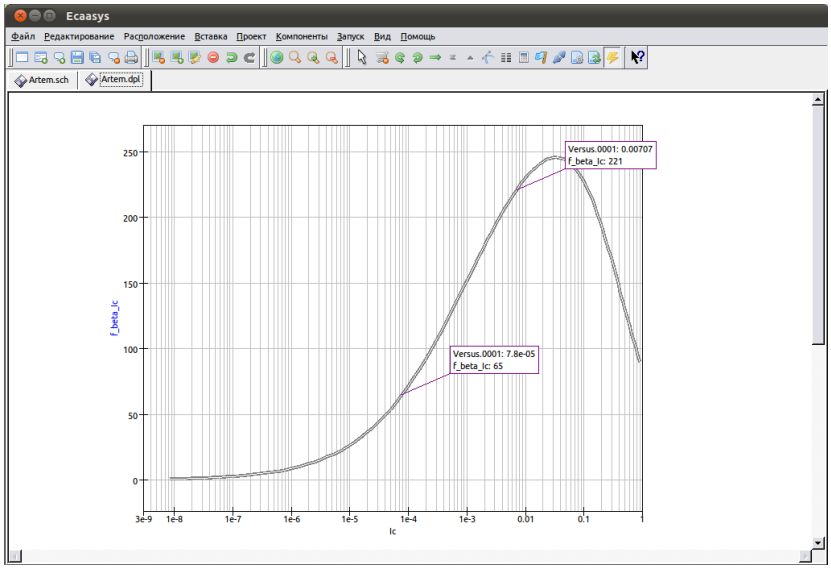

Рисунок 19 - Установка маркеров на график функции

Используя моделирование с разверткой параметра с помощью программы Ecaasys достаточно легко построить и график функции, зависящей от двух (или более) переменных. Для примера рассмотрим, как можно построить семейство выходных характеристик биполярного транзистора. Внесем в схему, показанную на рисунке небольшие изменения. Напряжение источника питания коллекторной цепи зададим в общем виде.

На рабочей области добавим еще один блок моделирования с разверткой параметра. Первый блок моделирования будет изменять напряжение питания коллекторной цепи (Обратите внимание, что в этом блоке в качестве «цели» моделирования указано имя блока моделирования на постоянном токе - DC1).

Эти изменения происходят линейно, от 0 до 4 вольт. Задав общее количество точек - 81, мы обеспечим шаг изменения напряжения 0,05 В.

Второй блок моделирования с разверткой параметра будет изменять величину базового тока для первого блока. Поэтому в свойствах моделирования для второго блока в качестве «цели» моделирования мы должны указать имя первого блока - SW1. Вполне достаточно получить на графике пять выходных характеристик. Значения токов базы, при которых строится выходная характеристика, установим линейно изменяющимися от 100 до 900 мкА с шагом 200 мкА, это даст необходимые пять значений тока базы. В блоке вычислений нам понадобится только одно уравнение - определяющее ток коллектора, т.к. необходимо построить зависимость именно этого тока от напряжения питания коллекторной цепи и тока базы.

Моделирование выполняется следующим образом:

1. Блок развертки параметра SW2 устанавливает первое значение тока базы - 100 мкА для блока SW1;
2. Блок SW1 для заданного тока базы рассчитывает 81 значение постоянного тока коллектора (в соответствии с уравнением) для различных напряжений источника питания;
3. Блок SW2 изменяет значение тока базы;
4. Блок SW1 повторяет расчет и т.д.

4.1.3 Снятие входных и проходных характеристик транзистора.

Методика работы по снятию статических характеристик для биполярного и полевого транзисторов в целом едина (с учетом того, что ток в цепи затвора полевого транзистора не протекает), поэтому будут рассмотрены особенности снятия характеристик только для биполярного транзистора. Что касается расчета такого важного для полевых транзисторов параметра, как крутизна, отметим наличие в программе встроенной функции дифференцирования $diff(y, x)$, аналогичной математической записи [80-87]:

Данная функция позволяет определять крутизну характеристики в рабочей точке, а также выполнять анализ ее изменений при изменении режима работы транзистора.

Для снятия входной $I_b = f(U_á)$ и проходной $I_k = f(U_á)$ статических характеристик биполярного транзистора можно использовать схему с двумя отдельными источниками питания для коллекторной и базовой цепей. Характеристики снимаются при фиксированном напряжении на коллекторе и задании моделирования с разверткой параметра для тока базы.

Величина тока базы в свойствах источника должна быть задана в общем виде (буквенно). Верхний и нижний предел изменения тока, а также количество рассчитываемых точек определяются в свойствах блока моделирования с разверткой параметра.

Для определения напряжения в базе транзистора необходимо установить метку в проводник, присоединенный к базе транзистора, либо подключить

между базовым выводом и общей точкой вольтметр. При выполнении моделирования изменяется базовый ток, выступающий в качестве аргумента. Изменения тока коллектора и напряжения базы при этом оказываются его функциями. В то же время, для входной характеристики требуется получить обратную зависимость, в которой ток базы - функция от приложенного напряжения $U_б$. Для решения этой задачи необходимо ввести в моделирование расчет уравнения (вставить на рабочее поле блок вычислений), в котором определить новую функцию, назвав ее, например, Ib_of_Ub, и определить эту функцию через встроенную функцию PlotVs(y,x). Где в качестве «y» задать ток базы, а в качестве «x» - напряжение в цепи базы. Если ток базы в свойствах источника был задан как I1, а для измерения напряжения в цепи базы установлен вольтметр Pr1, то функция запишется PlotVs(I1, Pr1.V). Подобную функцию нужно задать и для построения проходной характеристики, т.к. при выполнении моделирования нет прямой зависимости тока коллектора от приложенного к базе напряжения.

В названии функции могут использоваться только латинские буквы, цифры и знак подчеркивания. Если специальные символы (скобки, проценты и т.п.) ввести в поле имени функции нельзя в принципе, то символы кириллицы написать можно, но вычисляться такая функция не будет.

В реальных схемах включения транзистора применяется один источник питания, а для задания нужного тока в цепи базы используются различные схемы смещения, например, фиксированным током базы, рисунок 20. Если статические характеристики транзистора были сняты в схеме с двумя источниками питания и на этих характеристиках была выбрана рабочая точка, сохранить ее положение в показанной схеме нетрудно. Очевидно, что при появлении в коллекторной цепи нагрузки (сопротивления в цепи коллектора), напряжение источника питания необходимо увеличить на величину падения напряжения на этом сопротивлении. Так как ток коллектора в рабочей точке известен, рассчитать величину падения не составляет труда. На сопротивлении в цепи базы происходит полное падение напряжения источника питания, за исключением напряжения базы в выбранной рабочей точке. Ток базы в рабочей точке также известен, поэтому вычисления сводятся к применению закона Ома [88-92].

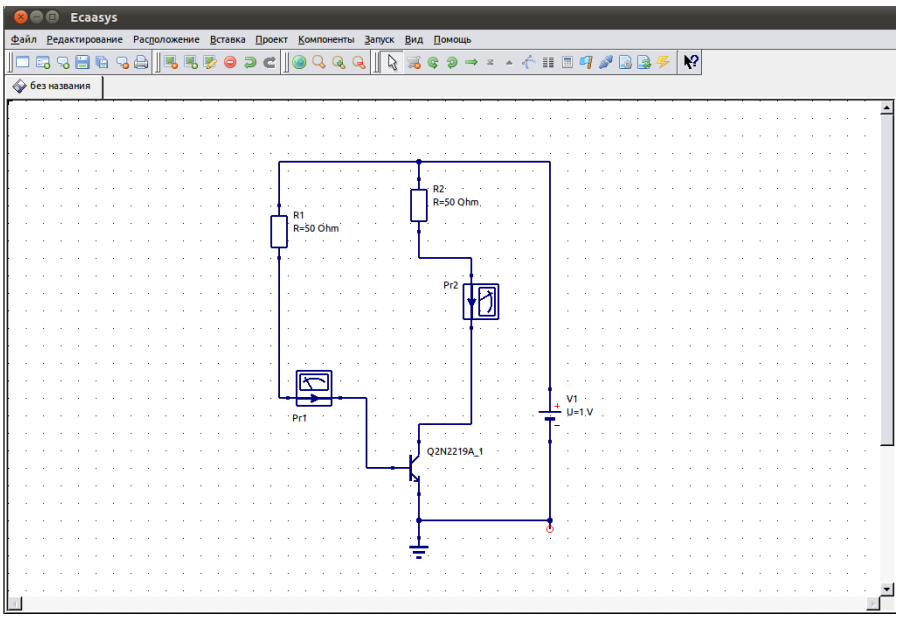

Рисунок 20 - Снятие статических характеристик в схеме смещения фиксированным током базы

В схеме на рисунке 20 для контроля тока в цепи базы и коллектора использованы амперметры, а для расчета напряжений коллектора и базы в схему вставлены метки U_b и U_c.

В схеме смещения с коллекторной стабилизацией (рисунок 21) к базовому резистору прикладывается разность напряжений коллектора и базы транзистора в рабочей точке. В остальном расчет величины сопротивлений повторяется. Напряжение базы и коллектора транзистора, как и в предыдущей схеме, можно контролировать, установив метки или с помощью вольтметров.

В схеме смещения с эмиттерной стабилизацией (рисунок 22) добавляется относительно небольшой по величине резистор в цепи эмиттера. Падение напряжения на этом сопротивлении повышает потенциал эмиттера, а, следовательно, и базы, поэтому для контроля напряжения базы и коллектора здесь необходимо использовать дополнительные вольтметры (рисунке 20). Если применить метки, то из рассчитанного для них напряжения придется вычесть падение напряжения на эмиттерном резисторе. На рисунке все сопротивления, а также напряжение источника питания, заданы в общем виде, т.к. рисунок служит иллюстрацией включения измерительных приборов. При построении исследуемой модели необходимо задание конкретных значений элементов и напряжений источников питания, иначе процесс моделирования завершится ошибкой. Кроме того, на рисунке не показан кубик моделирования на постоянном токе, без помещения на рабочее поле этого кубика моделирование не выполняется.

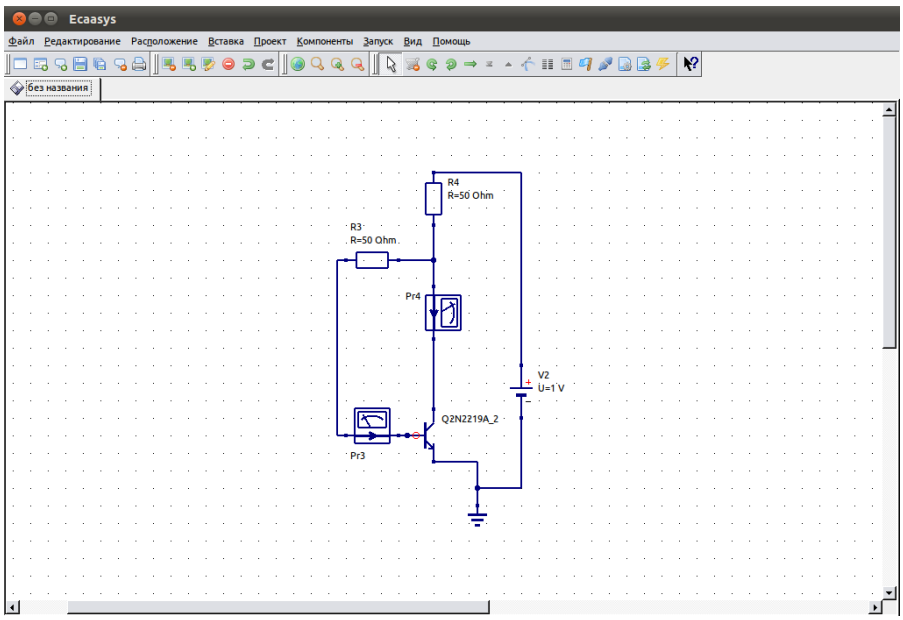

Рисунок 21 - Включение измерительных приборов в схеме смещения с коллекторной стабилизацией

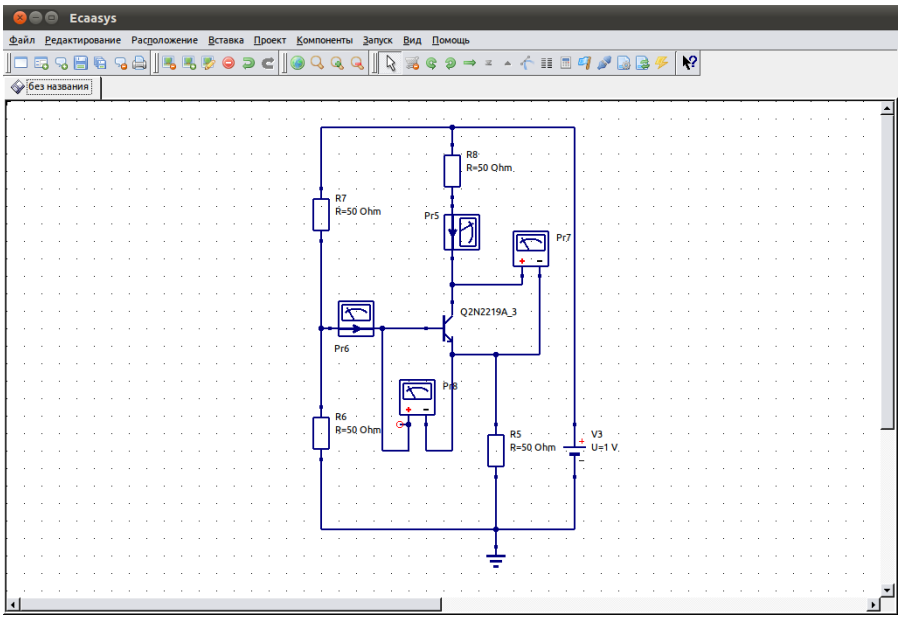

Рисунок 22 - Выполнение измерений в схеме с базовым делителем и эмиттерной стабилизацией режима

Отметим, что в качестве параметра, значение которого можно изменять во время моделирования, можно использовать любое свойство компонента, например, для транзистора можно изменять его температуру, моделируя работу схемы в разных условиях. Пример моделирования, где в качестве параметра задана температура транзистора, показан на рисунке 22. Здесь в свойствах транзистора его температура вместо установленного по умолчанию значения 26,85 °С задана в виде параметра - Tr (рисунок 23), а в свойствах моделирования с разверткой параметра для переменной Tr определены два

значения: 10 и 100 °С. Величины сопротивлений в цепи смещения подобраны таким образом, чтобы режим работы транзисторов при 10 °С был одинаковым.

Выполнив моделирование можно определить эффективность температурной стабилизации для разных схем.

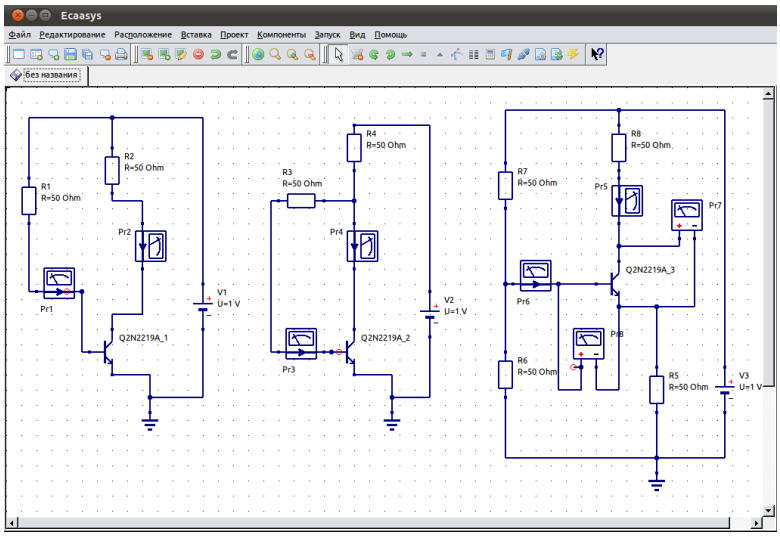

Рисунок 23 - Анализ эффективности температурной стабилизации различных схем смещения

Хорошо видно, что схема с эмиттерной термостабилизацией (показания прибора Pr3) оказывается наиболее эффективной - десятикратное увеличение температуры приводит к увеличению коллекторного тока транзистора менее чем на 20 %.

4.2 Моделирование на переменном токе

Большинство потребителей электрической энергии работает на переменном токе. В настоящее время почти вся электрическая энергия вырабатывается в виде энергии переменного тока. Это объясняется преимуществом производства и распределения этой энергии. Переменный ток получают на электростанциях, преобразуя с помощью генераторов механическую энергию в электрическую. Основное преимущество переменного тока по сравнению с постоянным заключается в возможности с помощью трансформаторов повышать или понижать напряжение, с минимальными потерями передавать электрическую энергию на большие расстояния, в трехфазных источниках питания получать сразу два напряжения: линейное и фазное. Кроме того, генераторы и двигатели переменного тока более просты по устройству, надежней в работе и проще в эксплуатации по сравнению с машинами постоянного тока.

В электрических цепях переменного тока наиболее часто используют синусоидальную форму, характеризующуюся тем, что все токи и напряжения являются синусоидальными функциями времени. В генераторах переменного

тока получают ЭДС, изменяющуюся во времени по закону синуса, и тем самым обеспечивают наиболее выгодный эксплуатационный режим работы электрических установок. Кроме того, синусоидальная форма тока и напряжения позволяет производить точный расчет электрических цепей с использованием метода комплексных чисел и приближенный расчет на основе метода векторных диаграмм. При этом для расчета используются законы Ома и Кирхгофа, но записанные в векторной или комплексной форме.

Анализ работы электронных схем на переменном токе в программе Ecaasys позволяет выяснить, как ведет себя схема при изменении частоты. Другими словами, использование моделирования на переменном токе сразу предполагает изучение работы схемы не на определенной, заданной частоте, а в некоторой полосе частот, хотя, при необходимости, как частный случай, можно выполнить моделирование и для переменного тока одной, определенной частоты.

4.2.1 Моделирование RC-цепи

Рассмотрим, как выполняется моделирование работы на переменном токе простой RC- цепи, представляющей собой фильтр высоких частот (ФВЧ). Схема цепи представлена на рисунок 24. Соединив между собой элементы цепи и источник переменного напряжения, добавим на схему две метки, одну - сразу после источника напряжения, другую - между сопротивлением и емкостью. Потенциал первой метки - это амплитудное значение переменного напряжения источника, потенциал второй метки - амплитудное значение переменного напряжения с выхода ФВЧ. Добавим также на рабочее поле кубик моделирования на переменном токе, указав в его свойствах диапазон частот от 0 до 1000 Гц, линейное изменение частоты и общее количество рассчитываемых точек.

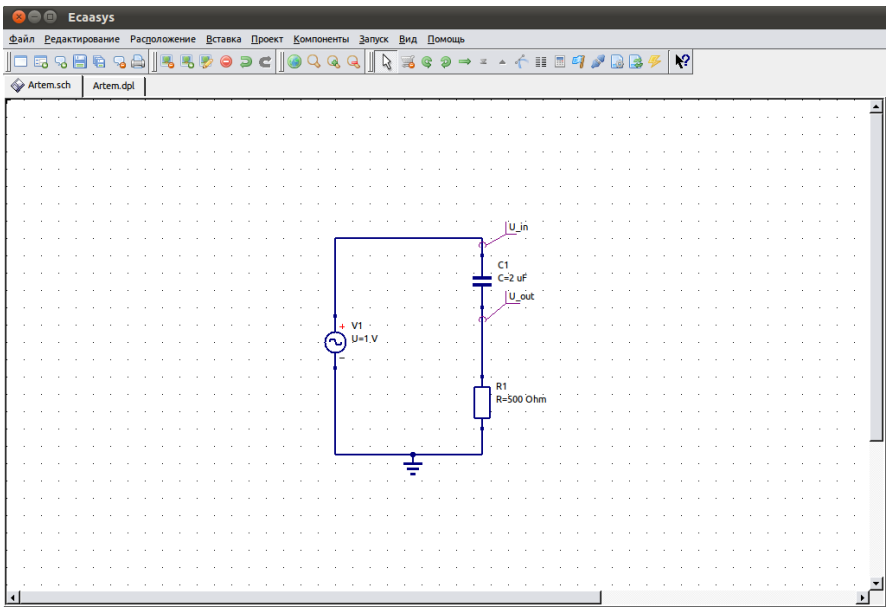

Рисунок 24 - Схема анализа работы ФВЧ на переменном токе

Для того, чтобы получить полное представление о поведении электронной схемы в частотной области, кроме амплитудно-частотной характеристики, необходимо знать как изменяется с частотой фаза электрического сигнала. Программа Ecaasys позволяет получить информацию о фазе сигнала при построении табличной диаграммы (фаза сигнала в определенной точке схемы рассчитывается относительно фазы колебаний напряжения источника питания). На рисунке 25 показана часть табличной диаграммы, полученной при анализе схемы риеуrt 24. Выходное напряжение, имеющее комплексное значение, в табличной диаграмме представлено в экспоненциальной форме с указанием амплитуды и фазы в градусах.

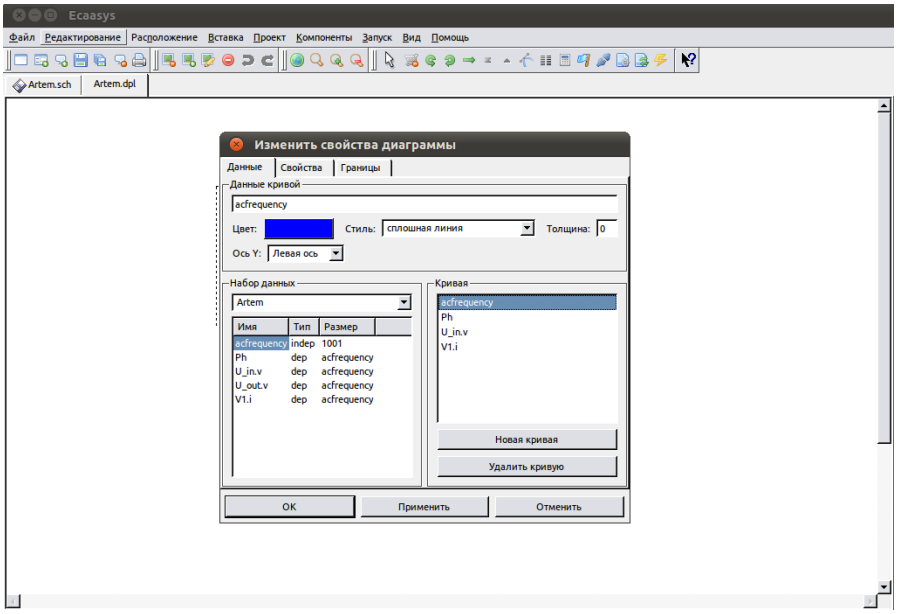

Рисунок 25 - Задание формы представления комплексного числа

Построить график изменения фазы сигнала можно с помощью встроенных функций $phase(x)$ и $angle(x)$, где x- комплексное число. Первая из этих функций определяет фазу в градусах, а вторая в радианах. Добавив в схему рисунке 25 уравнение для расчета фазы и несколько изменив параметры элементов, рисунок 26, мы можем построить характерный для RC-цепи график изменения фазы с частотой, рисунок 26 (в свойствах графика установлен логарифмический масштаб по оси абсцисс).

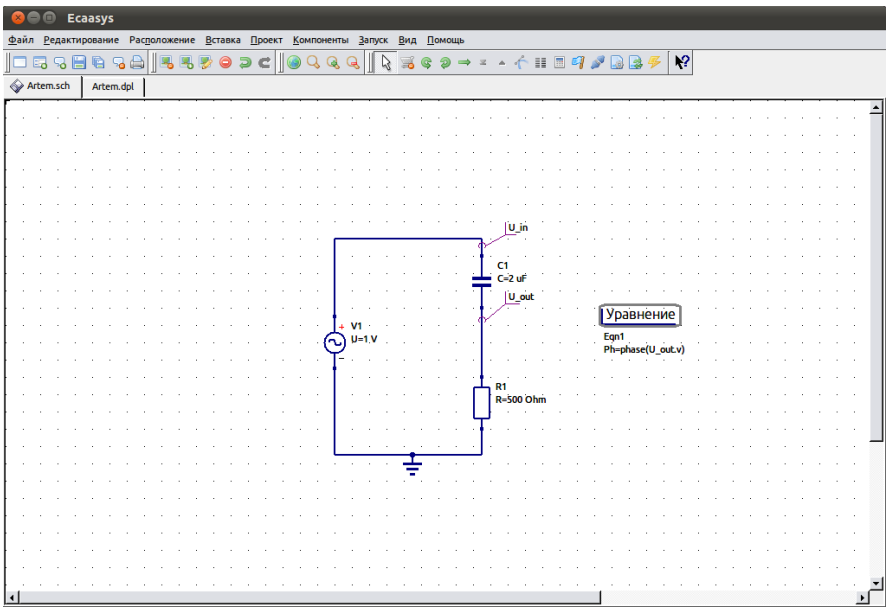

Рисунок 26 - Расчет зависимости фазы на нагрузочном резисторе от частоты

При построении графиков комплексных величин и добавлении на график маркеров, отображающих значение величины в конкретной точке, в свойствах маркера также можно выбрать форму представления выводимого значения: в виде комплексного числа или амплитудного значения и относительной фазы.

4.2.2 Моделирование на переменном токе с разверткой параметра.

Рассмотрим более сложный пример. Предположим, нам необходимо проанализировать усилительные характеристики биполярного транзистора в некотором частотном диапазоне и выбрать для него оптимальный режим работы по постоянному току. Источник постоянного тока I1 создает необходимое смещение в цепи базы, ток этого источника задан в общем виде. Из библиотеки компонентов выбран другой транзистор - BC109C.

В блоке вычислений переменная составляющая тока коллектора определена через переменную составляющую тока (малый индекс), протекающего через источник напряжения в коллекторной цепи (33):

$$I_c = -V1.I \tag{33}$$

Коэффициент усиления по переменному току определяется как отношение амплитуд переменных составляющих тока в цепи коллектора и базы (34):

$$Beta = I_c / I_b \tag{34}$$

Переменная составляющая тока в цепи базы транзистора задается равной амплитуде тока источника $I2$ (35):

$$I_b - I2.I \tag{35}$$

Выполняется моделирование работы транзистора на переменном токе, поэтому на рабочую область помещен соответствующий кубик. Верхняя частота моделирования задана 10 МГц. В то же время, очевидно, что для нормальной работы транзистора должен быть задан (а при моделировании - рассчитан) определенный режим по постоянному току, поэтому перед каждым моделированием на переменном токе, необходимо выполнить моделирование на постоянном токе. Для этого на рабочую область перенесен кубик моделирования на постоянном токе. Наконец, в свойствах третьего блока моделирования, расположенного на рабочей области, - блока моделирования с разверткой параметра, в качестве моделируемого блока указано AC1 - имя блока моделирования на переменном токе, а в качестве изменяемого параметра задан I_b - постоянный ток смещения (в виде списка значений).

При одновременном моделировании на постоянном и переменном токе, вывод результатов на табличную диаграмму допускается только для одного вида моделирования. Если необходимо отобразить результаты для того и другого режима, нужно построить две разных диаграммы. При попытке вывести рассчитанные напряжения и токи для двух режимов на одну диаграмму табличного вида в колонках для переменных напряжений и токов возникает ошибка. Это связано с разными форматами вывода результатов моделирования для постоянного и переменного токов. Для переменного тока в табличной диаграмме отображается относительная фаза, а для постоянного тока этот параметр отсутствует.

Результаты моделирования показаны на графике, (в свойствах графика установлен режим логарифмической разметки оси абсцисс (ось X)). На графике хорошо видно, что коэффициент усиления транзистора по току мало меняется в полосе частот 1 кГц - 1 МГц. Увеличение тока смещения в цепи базы позволяет добиться несколько большего усиления при ухудшении равномерности коэффициента усиления в полосе частот. Больший ток смещения приводит, кроме того, и к неэффективному рассеиванию энергии источника. Оптимальным поэтому можно считать режим с базовым током 50 мкА.

Отметим, что в данном случае на маркере для коэффициента усиления на переменном токе Beta, выбран формат комплексного числа.

В следующем примере определим граничную частоту усиления транзистора, т.е. частоту, на которой усиление транзистора становится равным единице. Внесем небольшие изменения в схему на рисунок 26. Во-первых, уберем блок моделирования с разверткой параметра, вместо этого зададим рабочую точку, установив $I_1 = I_b = 50$ мкА. Расширим диапазон моделирования на переменном токе до 500 МГц. Добавим еще один блок вычислений и с помощью встроенной функции dB(x) запишем коэффициент усиления транзистора по переменному току в децибелах (36):

$$Beta\_dB = dB(I_c / 1e-6) \qquad (36)$$

Где I_c - переменный ток в цепи коллектора, 1 мкА (1e-6) - амплитуда переменного тока, поступающего в базу. Их отношение дает коэффициент усиления транзистора по переменному току. Это отношение будет изменяться с частотой, т.к. переменный ток коллектора - функция частоты. Функция $dB(x)$ выполняет преобразование отношения токов по формуле (37):

$$dB(x) = 20\lg() \tag{37}$$

И преобразует зависимость переменного тока коллектора от частоты к логарифмическому виду. Используя еще одну встроенную функцию программы Ecaasys - $xvalue(f, y)$, определим граничную частоту (38):

$$Ft = xvalue(Beta\_db, 0) \tag{10}$$

Функция $xvalue(f, y)$ ищет значение аргумента функции (аргументом функции Beta_db является частота), при котором ее значение наиболее близко к указанному - y. В данном случае будет определяться частота, на которой коэффициент усиления равен единице (значение отношения амплитуд сигналов в децибелах равно нулю).

Часто требуется определить частоту, на которой уровень выходного сигнала снижается не до уровня входного сигнала, а в $V2$ раз (до уровня 0,7 от максимума). В этом случае вместо нуля в функции $xvalue(f, y)$ нужно задать y равным минус три децибела (39):

$$F = xvalue(Beta\_db, -3) \tag{39}$$

Таким образом, для транзистора BC109C граничная частота составила 345 МГц.

Распространенной ошибкой при определении значений функции в логарифмических единицах децибелах является задание диапазона частот (в свойствах моделирования на переменном токе) от 0 Гц. Логарифмическая функция не может иметь аргументом 0, поэтому при попытке построении графика возникнет ошибка определения функции.

4.2.3 Схемы генераторов колебаний

В программе Ecaasys можно получить осциллограмму напряжения и для схемы содержащей обратные связи, например для автоколебательного генератора. Типичным примером такого генератора служит симметричный мультивибратор (рисунок 27).

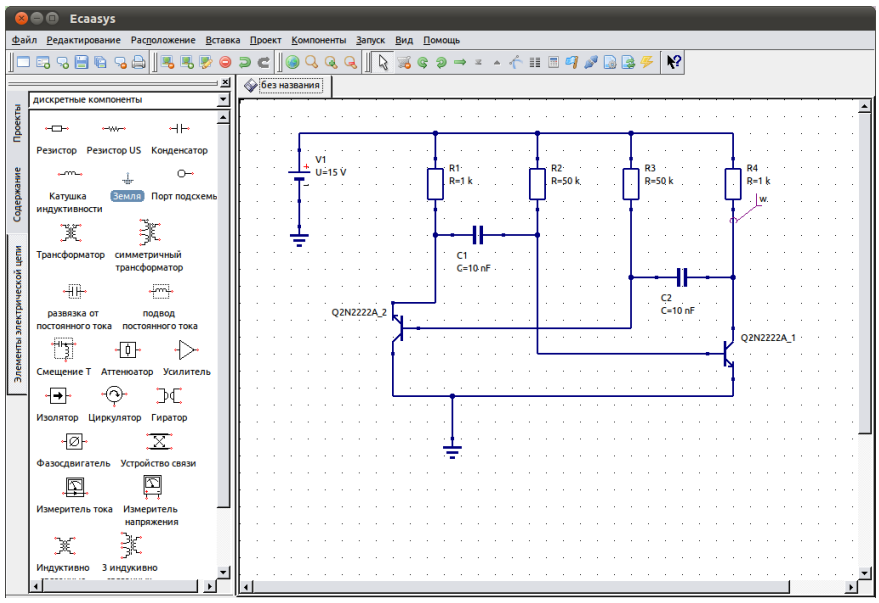

Рисунок 27 - Моделирование работы мультивибратора в программе Ecaasys

Расчет напряжения в точке, отмеченной меткой «w» выполняется на основе решения системы дифференциальных уравнений. Так как в случае автогенератора (и любой другой электронной схемы, в которой происходят сложные взаимосвязанные процессы) система полученных программой уравнений оказывается достаточно сложной, она не всегда может быть решена (рисунок 28).

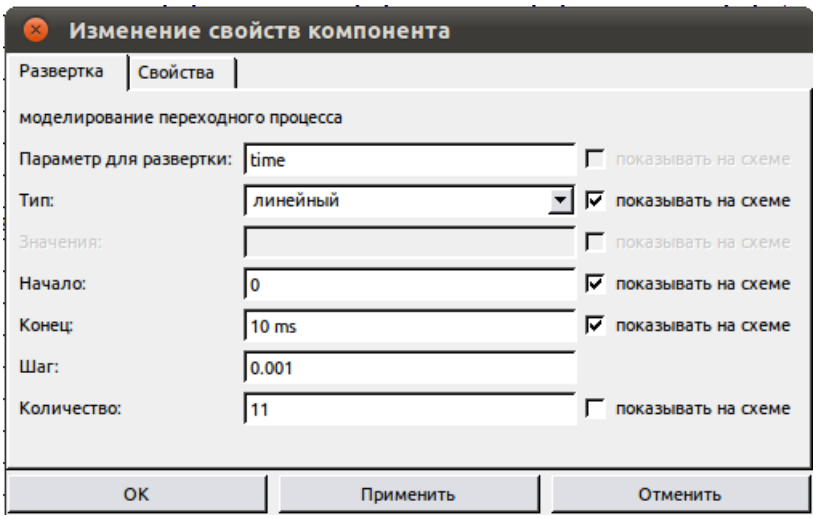

Рисунок 28 - Количество точек, на которые разбивается временной интервал может оказать решающее влияние на результат моделирования

Подобные трудности возникают не только в программе Ecaasys, но и во всех других программах моделирования работы электронных схем. На то успешным или нет будет результат моделирования влияют многие параметры расчетного режима. Так, например, использование транзистора из библиотеки компонентов предпочтительнее идеальной модели транзистора. Значительное

влияние на результат расчета оказывает выбор количества точек, на которые разбивается временной интервал, рисунок 29. Такие параметры, как метод интегрирования, начальный шаг, порядок дифференциального уравнения, максимальная точность и ряд других также могут повлиять на результат расчета, их значения можно изменить в окне свойств моделирования переходного процесса на вкладке «Свойства» (рисунок 29).

Предусмотрена возможность выбора из четырех методов интегрирования: Эйлера, Гира, метода трапеций и Адамса-Молтона. По-умолчанию применяется метод трапеций, но для схемы мультивибратора, например, лучшие результаты дает метод Адамса-Молтона. На рисунке 29 показана осциллограмма, полученная для схемы рисунке 27.

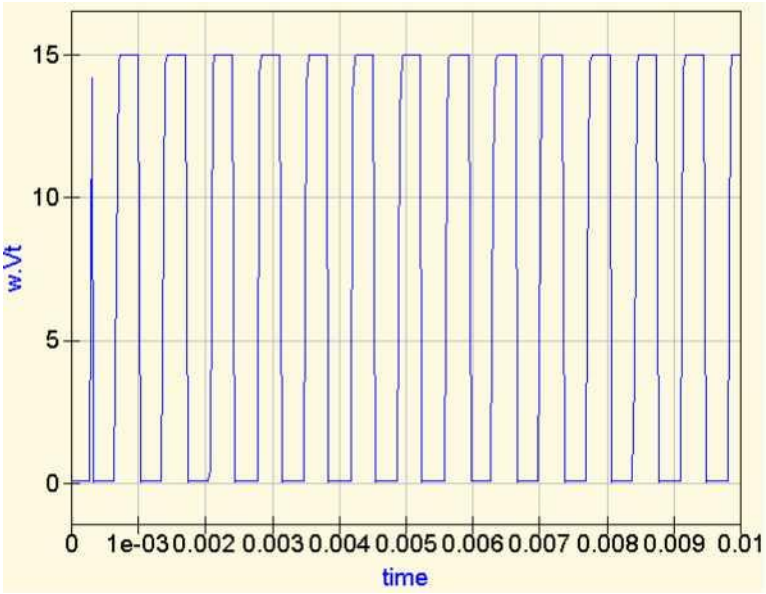

Рисунок 29 - Осциллограмма напряжения на выходе мультивибратора, полученная в результате моделирования

Другой пример построения генератора показан на риснке 30, это схема генератора Вина на операционном усилителе. На рисунке 31 изображена построенная Ecaasys осциллограмма его работы.

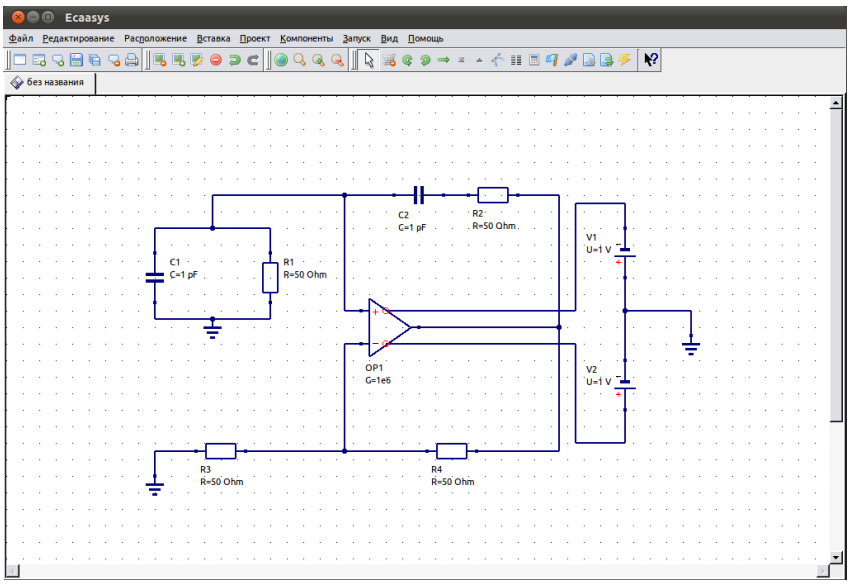

Рисунок 30 - Схема генератора Вина на операционном усилителе

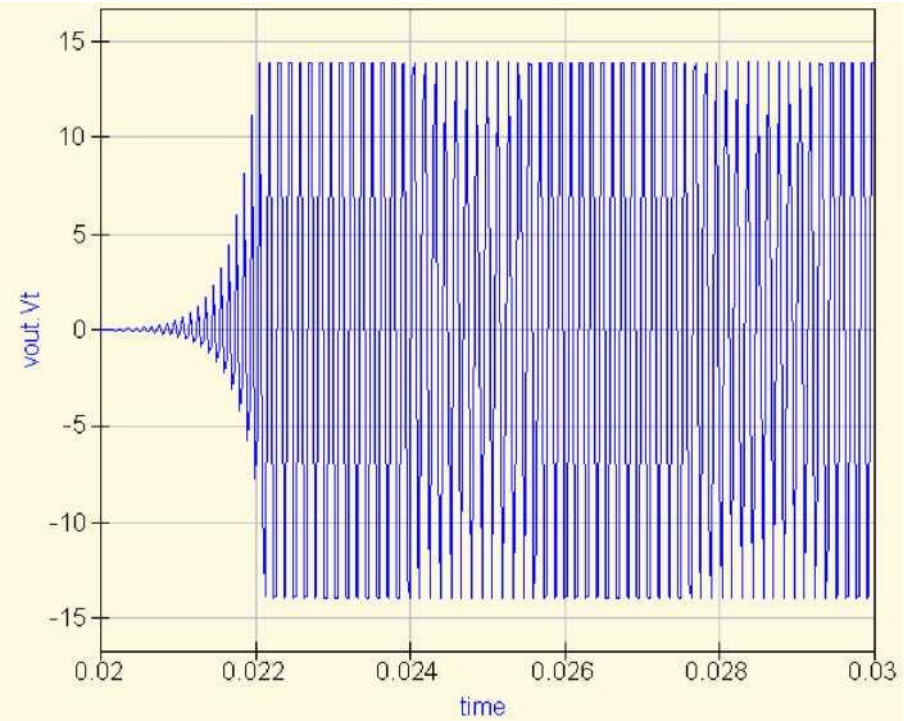

Рисунок 31 - Осциллограмма работы генератора Вина

4.2.4 Гармонический анализ

Возможность исследования частотного спектра несинусоидального сигнала имеет важное значение при изучении электронных схем. В программе Ecaasys для получения представления о спектральном составе сигнала сложной формы можно выполнить моделирование гармонического баланса. Для выполнения моделирования этого типа достаточно перенести из вкладки «Виды моделирования» на рабочую область кубик «Моделирование гармонического

баланса». В свойствах моделирования можно указать предполагаемую частоту основной гармоники и количество рассчитываемых гармоник. По-умолчанию в качестве частоты основной гармоники установлена частота 1 ГГц и это значение можно не менять, т.к. на результат расчета значение частоты, указанное в свойствах моделирования не влияет. Пример схемы ограничителя напряжения на диоде, вырабатывающей негармонический сигнал, показан на рисунке 32.

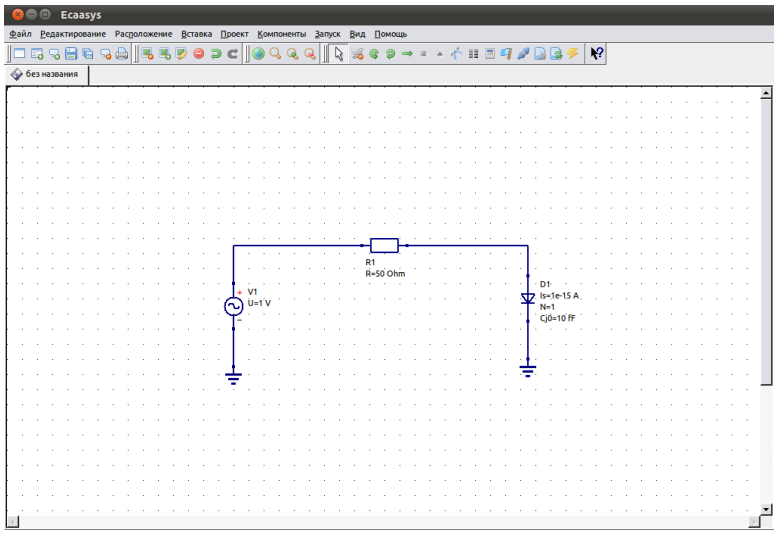

Рисунок 32 - Моделирование спектрального состава сигнала на выходе диодного ограничителя напряжения

На рисунке показано одновременное выполнение моделирования переходного процесса. Моделирование гармонического баланса может выполняться независимо и не требует непременного использования других видов моделирования. Однако, для получения не только спектрального состава, но и осциллограммы сигнала удобно совмещать эти два вида моделирования. Рисунок 33 демонстрирует полученные в результате моделирования диаграммы. Для получения диаграммы из расчитанных во время моделирования данных выбран параметр s.Vb - напряжение гармонического баланса на выходе ограничителя, обозначенном меткой s. На верхнем графике на рисунке 33 отображена осциллограмма синусоидального сигнала, ограниченного в положительной полуволне. На нижнем показан его спектральный состав. В свойствах первой диаграммы выбран стиль «Сплошная линия», а второй - «Стрелочки». Видно, что кроме основной гармоники с частотой 1 кГц сигнал содержит достаточно большую по величине постоянную составляющую, а также вторую, третью и пятую гармоники. При этом существенными можно считать только вторую и третью гармоники, сдвинутые по фазе относительно основного сигнала на 180 градусов.

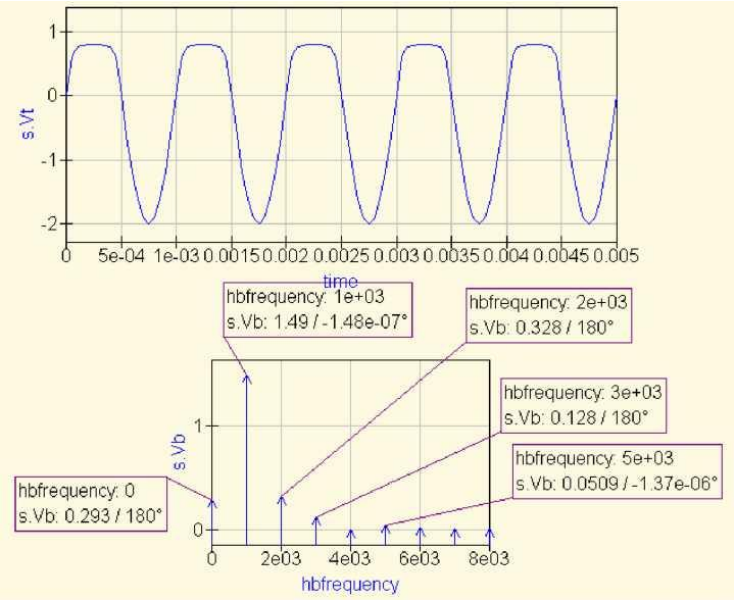

Рисунок 33 - Осциллограмма напряжения на выходе ограничителя и диаграмма, отражающая спектр сигнала

Моделирование гармонического баланса появилось только в версии 0.0.11 программы. На сегодняшний день работа этого вида моделирования еще очень несовершенна. Так, при попытке моделирования гармонического баланса для схем генераторов, рассмотренных в предыдущем разделе (рисунки 32, 33), возникают ошибки работы программы.

Ранее в программе Ecaasys была предусмотрена возможность выполнять частотный анализ другими способами, - с помощью встроенных функций, выполняющих частотные преобразования.

Наиболее простой способ получения спектра сложного сигнала - использование функции $Time2Freq(v(t),time)$. Данная функция выполняет дискретное преобразование Фурье (ДПФ) для функции $v(t)$ в диапазоне изменения time, создавая «изображение» изменяющейся во времени функции в частотной области. Начальное и конечное значение переменной time в случае выполнения моделирования переходных процессов задаются в свойствах самого моделирования. Если же необходимо сузить интервал, на котором будет выполняться ДПФ, то это можно сделать, указав в квадратных скобках номер начального и конечного элементов вектора time, разделенных двоеточием (вектор time создается автоматически при выполнении моделирования переходных процессов). Например, если в свойствах моделирования было указано количество расчетных точек 1000, то функция $Time2Freq(v(t),time[750:1000])$ рассчитает ДПФ на последней четверти временного интервала. Необходимый интервал можно также задавать указанием только одной (правой или левой) его границы, например, $time[750:]$ означает: «начиная с 750-го элемента и до конца», $time[:500]$ «сначала и до

500-го элемента». На рисунке 33 показана диаграмма спектрального состава сигнала, создаваемого симметричным мультивибратором (рисунок 32).

4.3 Выводы

1. Создана и продемонстрирована предельно понятная структура файлов.
2. Компиляция программная обеспечения легка и доступна на каждом персональном компьютере с помощью make-файлов.
3. Исходные коды программного обеспечения полностью свободны и документированы.
4. Продемонстрирована работа программного обеспечения на схемах, содержащих нелинейные элементы.

ЗАКЛЮЧЕНИЕ

В работе поставлена и обоснована задача создания программы расчета и анализа нелинейных электрических цепей. Рассказывается о важности решения этой задачи в условиях современного развития науки и увеличения требований рынка.

Создана новая модель адаптивного программного обеспечения, отличием которой, является блок определения алгоритма анализа и расчета электрической цепи, за счет чего, программное обеспечение становится еще более гибким и адаптивным.

Сформирована совокупность адаптивных алгоритмов моделирования электронных схем. Изложен набор заложенных в системе вычислительных методов, реализующих выполнение основных видов анализа. Используется адаптивные алгоритмы**,** позволяющие сочетать преимущества различных алгоритмов.

Продемонстрированы методы формирования уравнений, описывающие поведение схемы.

Приведен сравнительно новый, но уже дающий ощутимые практические результаты, подход к организации адаптивных форм являются библиотек моделей компонентов и возможность их преобразования, не связанная с особенностями используемых численных методов.

Разобрана наиболее сложная задача, результаты которой используются в других видах анализа (анализа установившихся режимов при периодических воздействиях, статистический анализ и др.) - моделирование динамических процессов нелинейных схем.

Создана база алгоритмов и методов анализа нелинейных электронных цепей программного обеспечения Ecaasys, основанная на применении двух методиках. Первая из них связана с применением для построения итерационного процесса методов решения линейных алгебраических уравнений. При этом считается, что на каждом шаге итерации система линейна и может быть решена, например, методом Гаусса или с использованием обратной матрицы. Эти методы получили название методов внешней итерации и, по сути, являются методами многократного решения систем линейных уравнений. Вторая методика связана с применением специализированных методов решения нелинейных систем уравнений (итерационные методы решения - метод простой итерации, метод Зейделя, метод Ньютона и др.).

Подробно описывается применение метода внешней итерации. Объясняется применение итерационных методов для решения нелинейных уравнений, например, метод Зейделя.

Приводится описание использования метода Ньютона для решения одного уравнения, для решения систем уравнений любого порядка и для решения уравнений узловых напряжений.

Описывается основы созданного программного обеспечения, даются основные понятия о программе Ecaasys. Изложены принципы работы

программы, а также дается справка по эксплуатации программного обеспечения.

В работе подробным образом описывается создание электрической схемы с помощью данного программного обеспечения, что значительно облегчит работу начинающему пользователю.

Дается подробное описание структуры созданного программного обеспечения, на примерах демонстрируется разработанное математическое и программное обеспечение.

С помощью примеров показано моделирование на постоянном токе, в том числе, простое моделирование, моделирование на постоянном токе с разверткой параметра и снятие входных и проходных характеристик транзистора.

Кроме того, подробно разобраны примеры моделирования на переменном токе, в том числе, моделирование RC-цепи, моделирование на переменном токе с разверткой параметра, схемы генераторов колебаний и гармонический анализ.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Жунусов З.А., Ли А.В. “Адаптивные методы и алгоритмы анализа нелинейных электронных схемна ПК”. Materials digest of the “LXIII International Research and Practice Conference and II stage of the Championship in physics, mathematics and chemistry sciences”. – London, 2013 – С.24-27

2. Жунусов З.А., Ли А.В. “Анализ сложных нелинейных электронных схем. Адаптивные методы и алгоритмы анализа”. Труды III Международной научно-практической конференции “ИКТ: образование, наука, информация”. – Алматы, 2013. – С. 5-7.

3. Жунусов З.А., Ли А.В. “Адаптивные алгоритмы моделирования электронных схем”. Труды 56-й научной конференции МФТИ, всероссийской научной конференции “Актуальные проблемы фундаментальных и прикладных наук в современном информационном обществе”, всероссийской научно-инновационной конференции “Физико-математические науки: актуальные проблемы и их решения”. – Москва: МФТИ, 2013. – С.52-53

4. Ли А.В. “Применение математических методов для адаптивного моделирования электронныз схем”. Сборник научных трудов магистрантов специальностей “Вычислительная техника и программное обеспечение” и “Информационные системы”. – Алматы: АУЭС, 2014. – С.67-81

5. Сигорский В.П., Витязь О.А. Покоординатная адаптация алгоритмов анализа динамического режима электронных схем //Электронное моделирование. -1980, -№4. -С. 22-26

6. Сигорский В.П., Витязь О.А, Проблемная адаптация систем автоматизированного проектирования. -Киев: Знание УССР, 1986. -20с.

7. Сигорский В.П., Коляда Ю.В., Колодницкий Й.М. Проблемная адаптация численного анализа электронных схем. 1: Формализация и рекуррентность вычислительных формул //Радиоэлектроника. -1986. -С. 18-23 (Изв. высш. учеб. заведений).

8. Сигорский В.П., Коляда Ю.В., Колодницкий Н.М. Проблемная адаптация численного анализа электронных схем. П: Двусторонние алгоритмы //Радиоэлектроника. -1987. №6. -С. 42-47 (Изв. высш. учеб. заведений).

9. Ильин В.Н., Жигалов И.Е., Ланцов В.Н. Методы автоматизированного схемотехнического проектирования нелинейных радиотехнических цепей //Радиоэлектроника. -1985, - №6. -С. 7-17 (Изв. высш. учеб. заведений).

10. Ильин В.Н., Коган В.Л. Расчет электронных схем методом формирования уравнений многополюсников на основе алгоритма Гаусса. - В кн.: Машинные методы проектирования электронных схем. М., МДЦНТП, 1975, с. 123-127.

11. Ильин В.Н., Коган В.Л. Разработка и применение программ автоматизации схемотехнического проектирования. -М.: Радио и связь 1984.-368с.

12. Ильин В.Н., Усков В.Л. Программа схемотехнического анализа электронных схем на мини-ЭВМ (СПР0С2-СМ)// Радиоэлектроника. -1987., № 1. -С. 56-62. (Изв. высш. учеб. заведений).

13. Ильин В.Н., Усков В.Л. Программа анализа электронных схем СПРОС-2 //Радиоэлектроника. -1987. - №6. -С. 75-77 (Изв. высш. заведений).

14. Анисимов Б.В., Белов Б.И., Норенков И.П. Машинный расчет элементов ЭВМ. -М.: высшая школа. 1976. -336с.

15. Агаханян Т.М. Интегральные микросхемы. -М.: Энергоатомиздат. 1983. -4б4с.

16. Арайс Е.А. Анализ цепей с адаптирующими моделями //Радиоэлектроника. -1979. -№ 6. - С. 20-26.

17. Арайс Е.А., Арайс Л.А., Дмитриев В. М. Алгоритмы и программы анализа сложных цепей и систем. - Томск: ТГУ, 1976. - 170с.

18. Арайс Е.А., Дмитриев В.М. Моделирование неоднородных цепей и систем на ЭВМ. - М.: Радио и связь, 1982. - 160 с.

19. Ионкин П.А., Миронов В.Г., Елисеев В.Н, Расчет нелинейных резистивных цепей на основе кусочно-линейной аппроксимации характеристик. I В кн.: Тезисы докладов на 5-ой Всесоюзной конференции по теории и методам расчета нелинейных электрических цепей и систем, Ташкент, 1975, вып. 1. - С. 95-96.

20. Ионкин П.А., Миронов В.Г., Елисеев В.Н. Расчет переходных процессов в цепях с нелинейными элементами, имешими кусочнолинейные характеристики. - В кн.: Тезисы докладов на 5-ой Всесоюзной конференции по теории и методам расчета нелинейных электрических цепей и систем, Ташкент, 1975, вып. 1.-С. 96.

21. Ионкин П.А., Миронов В.Г., Миронов Л.А. Расчет кусочно-линейньгх цепей в полном координатном баэисе. -Тр./Моск. энерг. ин-т, 1977, вып. 319. – С. 6-9.

22. Ионкин П.А., Пуньков И.М. Алгориты упорядочения пестренных узловых уравнений электронных цепе?.-Тр./Моск. энерг. ин-т,1981, вып. 556.-С.3-7.

23. Архангельский А.Я. Смешанное моделирование БИС. - Киев: Знание. - 1985. - 24с.

24. Архангельский А.Я. Моделирование больших электронных схем методами структурной и событийной декомпозиции. - М.: МИФИ, 1986. – 20 с.

25. Архангельский А.Я. Модели полупроводниковых приборов для машинного расчета электронных схем. - Ч.1. - М.: МИФИ, 1978. - 109с.

26. Архангельский А.Я., Меликян В.Ш. Смешанное схемотехническое и функционально-логическое моделирование аналого-цифровых схем //Электронное моделирование. - 1984. - №- 5. - С. 35-39.

27. Миронов В.Г. Уравнения нелинейных резистивных цепей. -Тр./Моск. энерг. ин-т, 1978. Вып. 348. -С. 19-24.

28. Миронов В.Г. Модели операционного усилителя. -Тр./Моск. энерг. ин-т, 1979. Вып. 415. -С. 7-16.

29. Миронов В.Г., Пуньков И.М. Моделирование и оптимизация частотно-избирательных схем. -М.: Моск. энерг. ин-т. 1987. -104с.

30. Миронов В.Г. Методы кусочно-линейного анализа нелинейных цепей //Теоретическая электротехника, -1983. -Вып. 35. -С. 162-168.

31. Миронов В.Г., Пуньков И.М., Жунусов З.А. Моделирование нелинейных схем на мини-ЭВМ //Теоретическая электротехника, -1988, - Вып. 45. -С. 96-102.

32. Миронов В.Г., Пуньков И.М., Жунусов З.А. Адаптивное моделирование нелинейных схем на ЭВМ. Доклады семинара "САПР Электронных схем", Чебоксары, 1989г. С. 26-37.

33. Миронов В.Г., Пуньков И.М., Жунусов З.А. Повышение выделительной эффективности анализа электронных схем. -Тр./Моск. институт, 1989г.

34. Баталов Б.В., Егоров Ю.Б., Русаков С.Г. Основы математического моделирования больших интегральных схем на ЭВМ. - М.: Радио и связь, 1982. - 168с.

35. Бахвалов И.С. Численные методы (анализ, алгебра, обыкновенные дифференциальные уравнения). -М.: Наука, 1973. -632с.

36. Бахов В.А. Адаптивный метод электрического расчета больших схем, состоящих из пороговых элементов //Радиоэлектроника.-1980 - № 2. -С. 50-54 (Изв. высш. учеб. заведений).

37. Бахов В.А., Ильин В.И., Фролкин В.Т. Алгоритм расчета нелинейных схем методом подсхем с использованием итераций по Ньютону //Радиоэлектроника. -1974. - № 6. - С.5-15 (Изв. высш. учеб. заведений).

38. Беллман Р. Процессы регулирования с адаптацией. -М.: Наука, 1964. - 359с.

39. Благитко Б.Я., Конник С.И. Сравнительная характеристика различных модификаций метода продолжения по параметру //Теоретическая электротехника. -1980. -Вып. 29. -С. 147-153.

40. Петренко А.И. Состояние и перспективы схемотехнического моделирования электронных схем на ЭВМ //Автоматизация проектирования в электронике: Респуб. межвед. научн.-техн. сб. -1980. –Вып. 26. –С. 15-22.

41. Петренко А.И., Слюсар П.Б. Комбинированный алгоритм анализа нелинейных электронных схем, основанный на оценке локальной жесткости задачи //Проблемы нелинейной электроники, -Ч.2. - Киев: Наукова думка, 1984. -С. 117-119.

42. Петренко А.И., Гумен Н.Б. Алгоритм и программа расчета статического режима нелинейных схем //Проблемы нелинейной электротехники. Ч.2. -Киев: Наукова думка. -I984. -С. 31-36.

43. Петренко А.И., Слюсар П.Б. Использование явных методов интегрирования при моделировании электронных схем в расширенном однородном координатном базисе //Автоматизация проектирования в электронике: Респ. межвед. науч.-техн. сб.-1985.-Вып. 32.-С. 50-52.

44. Петренко А.И., Слюсар П.В. Моделирование электронных схем в расширенном однородном координатном базисе с использованием адаптивных явно-неявных процедур численного интегрирования //Радиоэлектроника. -1988. -№9. -С. 73-74. (Изв. высш. учеб. заведений).

45. Петренко А.И., Власов А.И., Тимченко А.П. Моделирование электронных схем в расширенном однородном координатном базисе. – В сб. "Машинные методы проектирования электронных схем". М.: "Знание" 1975. -С. 47-56.

46. Петренко А.И., Слюсар П.Б. Автоматическое переключение явных и неявных методов интегрирования при решении систем обыкновенных дифференциальных уравнений //Радиоэлектроника. -1986. №1. -49-51 (изв. высш. учеб. заведений).

47. Блажкевич Б.И. Физические основы алгоритмов анализа электронных цепей. -Киев, Наукова думка, 1979.-296с.

48. Бондаренко В.М. Вопросы анализа нелинейных цепей.- Киев, Наукова думка, 1967.-160с.

49. Бондаренко В.М. Методы и алгоритмы анализа статических и динамических режимов нелинейных цепей. -К., 1974. -(Препринт/ АН УССР.ИЗД; № 66).

50. Бондаренко В.М., Пфенинг В.В. Исследование и разработка алгоритмов и программ машинного проектирования электронных схем. -К., 1973. -(Препринт/ АН УССР.ИЗД; № 57).

51. Витязь О.А., Колядо D.В. Расчет динамического режима нелинейных электронных схем рекуррентными полуявными алгоритмами типа Рунге-Кутта // Автоматизация проектирования в электронике: Респуб. межвед. научн.-техн. сб. -1979. -Вып. 20. С.101-103.

52. Воеводин В.В., Кузнецов D.А. Матрицы и вычисления. -М.: Наука, 1984. -320с.

53. Гантмахер Ф.Р. Теория матриц. -М.: Наука, 1967. -576с.

54. Гаврилов Л.П. Блочный принцип моделирования нелинейных схем во временной области //Радиоэлектроника. -1980. - №6. -С. 89-91 (Изв. высш. учеб. заведений).

55. Пухов Г.Е. Методы анализа и синтеза квазианалоговых электронных цепей. -К.: Наукова думка, 1967. -568с.

56. Пухов Г.Е. Дифференциальные преобразования функций и уравнений. -К. : Наукова думка, 1980. -419с.

57. Пухов Г.Е. О применении преобразований Тейлора к решению дифференциальных уравнений //Электронное моделирование.- 1976. Вып. 11. -С. 18-23.

58. Герра А. Эффективные алгоритмы анализа нелинейных электрических и электронных схем на ЭВМ. Дисс. на соиск. уч. степени к.т.н., М.,1984.

59. Глориозов Е.Л., Ссорин В.Г., Сыпчук П.П. Введение в автоматизацию схемотехнического проектирования. -М.: Сов. радио, 1976. -224с.

60. Демирчян К.С., Бутырин П.А. Моделирование и машинный расчет электрических цепей. - М.: высшая школа, 1988. -335с.

61. Дмитриев-Здоров В.Б., Дудка В.Б. Алгоритм решения системы нелинейных алгебраических уравнений при анализе электрических цепей с помощью адаптивных моделей //Радиоэлектроника.- 1988 - № б.-С.43-48.

62. Дмитриев-Здоров В.Б., Попов В.П. Адаптивный алгоритм анализа линейных цепей во временной области //Радиоэлектроника.- 1983.-№ 6.-С. 5-8.

63. Дмитриев-Здоров В.Б., Пономарев А.М., Попов В.П. Применение перестраиваемых моделей для анализа электрических цепей во временной области //Автоматизация проектирования в электронике: Респ. межвед. науч.-техн. сб. -1985, -Вып. 32. –С. 37-42.

64. Деккер К., Вервер Я., Устойчивость методов Рунге-Кутта для жестких нелинейных дифференциальных равнений. - М.: Мир. 1988. - 334с.

65. Джордж А., Лю Дж. Численное решение больших разреженных уравнений. - М.: Мир, 1984. - 333с.

66. Захария А.И., Мандзий Б.А., Подольский М.Р. Моделирование нелинейных процессов в электрических цепях с учетом внешних условий //Проблемы нелинейной электротехники: Ч.2. -Киев: Наукова думка. -1984. -С. 13-15.

67. Захария А.И., Саноцкий Ю.В. Применение генерации и адаптации алгоритмов для определения статического режима радиоэлектронных схем //Проектирование систем автоматики и вычислительной техники. -Воронеж :ВПИ. -1978, -С. 8-11.

68. Жигалов И.Е., Ланцов В.Н., Меркутов А.С. Комплекс программ анализа с адаптацией методов моделирования нелинейных радиотехнических устройств //Автоматизация проектирования в электронике: Респ. межвед. науч.-техн. сб. -1987. -Вып.35. -С. 8-11.

69. Жумик В.В., Стахив П.Г. Оценка устойчивости диакоптических методов расчета динамических режимов электронных цепей Теоретическая электротехника. -1987. -Вып. 43. -С. 123-126.

70. Калахан Д.А. Методы машинного расчета электронных схем. -М.: Мир, 1970. -344с.

71. Коваль Н.В., Семагина Э.П. Об устойчивости алгоритмов решения систем обыкновенных дифференциальных уравнений методом дифференциального преобразования //Теоретическая электротехника. -1985. -Вып. 41. -С. 108-118.

72. Ракитский Ю.В., Устинов С.М., Черноруцкий Н.Г. Численные методы решения жестких систем. -М.: Наука, 1979. -208с.

73. Рендзиняк С.И., Стахив П.Г. Об устойчивости вычислительного процесса при расчете переходных режимов по частям //Теоретическая электротехника. -1986. - Вып. 41. - С. №423.

74. Рендзиняк С.И. Исследование эффективности метода подсхем при расчете переходных процессор в некоторых сложных радиоэлектронных схемах //Теоретическая электротехника. -1985. -Вып. 41. -С. 139-143.

75. Синицкий Л.А. Элементы качественной теории нелинейных электрических цепей.-Львов: Вища школа, 1975. -153с.

76. Синицкий Л.А. Методы аналитической механики в теории электрических цепей.-Львов: Вища школа, 1978. -138с.

77. Синицкий Л.А. О комбинированных методах численного интегрирования уравнений электронных цепей //Теоретическая электротехника . -1984. -Вып. 37. -С. 65-73.

78. Синицкий Л.А. Адаптивные алгоритмы анализа электронных схем. -Киев: Знание. -1988 -16с.

79. Колесин И.Д. Обнаружение быстрых компонент //Вопросы механики и процессов управления. -1982. -№ 5. -С. 205-2I0.

80. Колосов С.П., Сидоров Ю.А. Нелинейные двухполюсники и четырехполюсники. -М.: Высшая школа, 1981. -224с.

81. Коляда Ю.В., Сигорский В.П. Полуявные алгоритмы численного интегрирования "жестких" уравнений //Радиоэлектроника.

-1975.- №3. -с. 83-84 (Изв. высш. учеб. заведений).

82. Нагорный Л.Я. Декомпозиция и распараллеливание алгоритмов решения систем нелинейных уравнений большей размерности. - Киев: Знание. -1981. -26с.

83. Норенков И.П. Комбинированные методы моделирования и анализа в системах автоматизированного проектирования //Приборостроение. -1983. -№ 9. -С. 77-82 (Изв. высш. учеб. заведений).

84. Норенков И.П., Евстифеев Ю.А., Маничев В.Б. Адаптизный метод ускоренного анализа многопериодных электронных схем //Радиоэлектроника . -1987. -№6. -С. 47-51.

85. Норенков И.П., Жук Д.М., Макичев В.Г., Трудокожин А. Анализ электронных схем при совместном применении явных и неявных методов интегрирования //Радиоэлектроника. -1979. -№6. -С. 27-31 (Изв. высш. учеб. заведений).

86. Ортега Дж., Пул У. Введение в численные методы решения дифференциальных уравнений. - М.: Наука. I986.

87. Павлюк Н.И., Синицкий Л.А., Икигельский Я.А. Исследование одного адаптивного алгоритма численного моделирования динамических систем //Теоретическая электротехника. -1987, -Вып. 43. - С. 86-93.

88. Пуньков И.М., Жунусов З.А. Анализ нелинейных цепей на СМ ЭВМ. В кн.: Автоматизация электроприводов и оптимизация режимов электропотребления, Красноярск, 1985г., с.52-53.

89. Попов Н.П., Дмитриев-Здоров В.Б., Адаптируемые модели для автоматизированного анализа электрических цепей: (Обзор) //Радиоэлектроника. -I986. -№ 6. -С. 32-38 (Изв. высш. учеб. зазедений).

90. Стахив П.Г., Рендзиняк С.И. Алгоритм расчета переходных процессов в радиоэлектронных схемах при делении их на части //Теоретическая электротехника. -1980. -Вып. 29. -С. 143-146.

91. Стахив П.Г., Рендзиняк С.И. Алгоритм расчета переходных процессов в радиоэлектронных схемах при делении их на части с использованием резистивных моделей реактивных компонентов //Автоматизация проектирования в электронике: Респ. межвед. науч.-техн. сб. -1985. -Вып. 31. -С. 13-16.

92. Тимовский А.К. Критерии качества программ анализа электронных схем //Автоматизация проектирования в электронике: Респ. межвед. науч.-техн. с6. -I986. -Вып. 34. -С. 3-13.

ПРИЛОЖЕНИЕ А

Папка Ecaasys/ecaasys – покрывает практически все исходные коды и компоненты проекта. В частности, она содержит набор иконок элементов электрической цепи, панелей управления Ecaasys и др.; исходные коды построения и функционирования элементов электрической цепи; диалоги ecaasys и многое другое.

Папка Ecaasys/ecaasys состоит из 6 папок и 80 файлов, объем на диске - 11,0 МБ.

Структура: Смотрите файлы *about ecaasys_structure.docx*, *about ecaasys_structure.rtf* в корневой папке (Ecaasys)

Описание:

- *Папка Ecaasys/ecaasys/bitmaps:* расширение – папка (509 КБ)

Изображение: 

Описание: В этой папке содержатся иконки, являющиеся изображениями элементов электрической цепи, а также иконки элементов меню. Среди них: резисторы, емкости, усилители, заземление и др. Среди иконок меню: сохранить, отменить, вырезать и др.

Исходный код: нет

Участие в программе: нет

Основная процедура: нет

- *Папка Ecaasys/Папка ecaasys/components:* расширение - папка (1,09 МБ)

Изображение: 

Описание: Эта папка сочетается с папкой bitmaps. В bitmaps содержатся иконки элементов электрической цепи, а в components их исходные коды. Т.е. в папке находятся файлы C++ отвечающие за работу резисторов, емкостей, усилителей и других компонентов.

Исходный код: нет

Участие в программе: нет

Основная процедура: нет

- *Папка Ecaasys/Папка ecaasys/diagrams:* расширение - папка (259 КБ)

Изображение: 

Описание: В папке diagrams находятся файлы C++, отвечающие за работу различных видов электрических схем. Они разделены по типам этих схем. Среди них: диаграммы кривых, 3D-диаграммы и др. с их графиками и маркерами.

Исходный код: нет

Участие в программе: нет

Основная процедура: нет

- *Папка Ecaasys/Папка ecaasys/dialogs:* расширение - папка (219 КБ)

Изображение: 

Описание: В этой папке содержатся файлы C++, отвечающие за взаимодействия интерфейса программы с пользователем, с помощью диалогов. Среди них: диалог сохранения проекта, удаления проекта, создания нового проекта, изменения определенных настроек, выводе информационных сообщений и др.

Исходный код: нет

Участие в программе: нет

Основная процедура: нет

- *Папка Ecaasys/Папка ecaasys/octave:* расширение - папка (18,4 КБ)

Изображение: 

Описание: Данная папка отвечает за работу октавы GNU. Октава GNU - интерпретируемый язык высокого уровня, прежде всего предназначенный для числовых вычислений. Это обеспечивает возможности числового решения линейных и нелинейных проблем, и для того, чтобы выполнить другие числовые эксперименты. Это также обеспечивает обширные графические возможности визуализации данных и манипулирования. Октава обычно используется через ее интерактивный интерфейс командной строки, но она может также использоваться, чтобы записать неинтерактивные программы. Язык Октавы довольно подобен Matlab так, чтобы большинство программ было легко переносимо. Эта папка появилась только в в последней 16-ой версии Ecaasys. Имеет очень простую структуру, состоящую из установочных MAKE-файлов, а также файлов приема, обработки и отображения числовых данных.

Исходный код: нет

Участие в программе: нет

Основная процедура: нет

- *Папка Ecaasys/Папка ecaasys/paintings:* расширение - папка (146 КБ)

Изображение: 

Описание: В папке paintings находятся файлы C++, отвечающие за работу элементов рисования, используемых в электрических схемах. Среди них: прямоугольники, линии, овалы, круга и др.

Исходный код: нет

Участие в программе: нет

Основная процедура: нет

- *Папка Ecaasys/ecaasys/ChangeLog:* расширение – файл Linux (63,8 КБ)

Изображение: 

Описание: Обычный файл Linux, содержащий информацию об изменениях, произведенных за всю история существования программного продукта. Под изменениями понимается не работа программы, а изменения самого софта его разработчиками.

Исходный код:

```
============
Unpack the distribution tarball:
   $ tar xvzf ecaasys-<version>.tar.gz              (using GNU tar)
   $ gzip -cd ecaasys-<version>.tar.gz | tar xvf -   (using another tar)
Change into the source directory:
   $ cd ecaasys-<version>
Configure the source package for your system:
   $ ./configure
Now compile the package:
   $ make
Install Ecaasys:
   $ make install
You must have root privileges if you want to install the package in the
standard location (/usr/local) or in any location that is only writable
by root.
For further information on installing the package, please consult the
file INSTALL included in this distribution.
Please note:  Users of the FreeBSD OS may use a GNU make (probably gmake)
to compile and install the package.
Getting the latest SVN snapshot
==============================
You can always get the latest Ecaasys version from our SVN repository.
Please use an official release if you want to work with Ecaasys.  The SVN
version might not even compile.
   $ svn co https://ecaasys.svn.sourceforge.net/svnroot/ecaasys/trunk/ecaasys
Press 'Enter' when asked for a password.  Run `sh autogen.sh' and
`configure' with the appropriate options.  Maintainance currently
requires Autoconf version 2.57 and GNU automake 1.7.0.
```

Участие в программе: да

- *Папка Ecaasys/ecaasys/element.cpp:* расширение – .cpp (1,28 КБ)

Изображение: 

Описание: Файл C++, функционирующий во время работы с тем или иным элементом электрической схемы. Он отвечает за форму, размеры, цвет, стиль и свойства определенного компонента.

Исходный код:

```
#ifdef HAVE_CONFIG_H
# include <config.h>
#endif
#include "ecaasyshelp.h"
#include "htmldatafetcher.h"
#include <qpushbutton.h>
#include <qaction.h>
#include <qpixmap.h>
#include <qfile.h>
#include <qtextstream.h>
#include <qpopupmenu.h>
#include <qmenubar.h>
```

```
#include <qapplication.h>
#include <qlistview.h>
EcaasysHelp::EcaasysHelp(const QString& page)
{
  currentIndex = 0;
  dataFetcher = new HtmlDataFetcher();
  links = dataFetcher->fetchLinksToFiles(EcaasysHelpDir.filePath("index.html"));
  // set application icon
  setIcon (QPixmap(EcaasysSettings.BitmapDir + "big.ecaasys.xpm"));
  setCaption(tr("Ecaasys Help System"));
  textBrowser = new TextBrowser(this);
  textBrowser->setMinimumSize(400,200);
  setCentralWidget(textBrowser);
  setupActions();
  createSidebar();
  textBrowser->setSource(EcaasysHelpDir.filePath(links[0]));
  // ....................................
  if(!page.isEmpty())
    textBrowser->setSource(EcaasysHelpDir.filePath(page));
}
EcaasysHelp::~EcaasysHelp()
{}
void EcaasysHelp::setupActions()
{
  QToolBar *toolbar = new QToolBar(this,"main_toolbar");
  QMenuBar *bar = menuBar();
  statusBar();
  const QKeySequence ks = QKeySequence();
  QAction *quitAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "quit.png")),
                          tr("&Quit"), CTRL+Key_Q, this);
  QAction *backAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "back.png")),
                          tr("&Back"), ALT+Key_Left, this);
  QAction *forwardAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "forward.png")),
                            tr("&Forward"), ALT+Key_Right, this);
  QAction *homeAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "home.png")),
                          tr("&Home"),CTRL+Key_H,this);
  previousAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir +
"previous.png")),tr("&Previous"),
                          ks, this);
  nextAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "next.png")),
                      tr("&Next"), ks, this);
  viewBrowseDock = new QAction(tr("&Table of Contents"), 0, this);
  viewBrowseDock->setToggleAction(true);
  viewBrowseDock->setOn(true);
  viewBrowseDock->setStatusTip(tr("Enables/disables the table of contents"));
  viewBrowseDock->setWhatsThis(tr("Table of Contents\n\nEnables/disables the table of contents"));
  connect(quitAction,SIGNAL(activated()),qApp,SLOT(quit()));
  connect(backAction,SIGNAL(activated()),textBrowser,SLOT(backward()));
  connect(textBrowser,SIGNAL(backwardAvailable(bool)),backAction,SLOT(setEnabled(bool)));
  connect(forwardAction,SIGNAL(activated()),textBrowser,SLOT(forward()));
  connect(textBrowser,SIGNAL(forwardAvailable(bool)),forwardAction,SLOT(setEnabled(bool)));
  connect(homeAction,SIGNAL(activated()),textBrowser,SLOT(home()));
  connect(textBrowser,SIGNAL(sourceChanged(const QString &)),this,SLOT(slotSourceChanged(const
QString&)));
  connect(previousAction,SIGNAL(activated()),this,SLOT(previousLink()));
  connect(nextAction,SIGNAL(activated()),this,SLOT(nextLink()));
  connect(viewBrowseDock, SIGNAL(toggled(bool)), SLOT(slotToggleSidebar(bool)));
  backAction->addTo(toolbar);
  forwardAction->addTo(toolbar);
  toolbar->addSeparator();
  homeAction->addTo(toolbar);
  previousAction->addTo(toolbar);
```

```
  nextAction->addTo(toolbar);
  toolbar->addSeparator();
  quitAction->addTo(toolbar);
  QPopupMenu *fileMenu = new QPopupMenu(this);
  quitAction->addTo(fileMenu);
  QPopupMenu *viewMenu = new QPopupMenu(this);
  backAction->addTo(viewMenu);
  forwardAction->addTo(viewMenu);
  homeAction->addTo(viewMenu);
  previousAction->addTo(viewMenu);
  nextAction->addTo(viewMenu);
  viewMenu->insertSeparator();
  viewBrowseDock->addTo(viewMenu);
  QPopupMenu *helpMenu = new QPopupMenu(this);
  helpMenu->insertItem(tr("&About Qt"),qApp,SLOT(aboutQt()));
  bar->insertItem(tr("&File"), fileMenu );
  bar->insertItem(tr("&View"),viewMenu);
  bar->insertSeparator();
  bar->insertItem(tr("&Help"),helpMenu);
}
void EcaasysHelp::createSidebar()
{
  dock = new QDockWindow(QDockWindow::InDock,this);
  dock->setResizeEnabled(true);
  dock->setCloseMode(QDockWindow::Always);
  connect(dock,SIGNAL(visibilityChanged(bool)),this,SLOT(slotToggleSidebarAction(bool)));
  chaptersView = new QListView(dock,"chapters_view");
  chaptersView->setRootIsDecorated(false);
  chaptersView->addColumn(tr("Contents"));
  chaptersView->setSorting(-1);
  chaptersView->setSelectionMode(QListView::Single);
  dock->setWidget(chaptersView);
  moveDockWindow(dock,QDockWindow::Left);
  QStringList l = dataFetcher->fetchChapterTexts(EcaasysHelpDir.filePath("index.html"));
  for(int i=l.count()-1;i>=0;i--)
    chaptersView->insertItem(new QListViewItem(chaptersView,l[i],QString::number(i+1)));
  QListViewItem *curItem = new QListViewItem(chaptersView,tr("Home"),QString::number(0));
  chaptersView->insertItem(curItem);
  chaptersView->setSelected(curItem,true);
  connect(chaptersView,SIGNAL(selectionChanged()),this,SLOT(displaySelectedChapter()));
}
void EcaasysHelp::displaySelectedChapter()
{
  if(chaptersView->selectedItem() == 0)
      return;
  uint y = chaptersView->selectedItem()->text(1).toUInt();
  Q_ASSERT(y < links.count());
  textBrowser->setSource(EcaasysHelpDir.filePath(links[y]));
  cachedSelectedText = chaptersView->selectedItem()->text(1);
}
//This slot updates next and previous actions i.e enabling/disabling
void EcaasysHelp::slotSourceChanged(const QString& _str)
{
  QString str(_str);
  // Remove '#*' chars in link since we don't check '#top,etc' while tracking previous actions
  int hashPos = str.findRev('#');
  if(hashPos != -1)
    str = str.left(hashPos);
  // Don't do anything if accessing same page
  if(str == currentSource)
    return;
  bool found = false;
```

```
  for(unsigned int i=0;i < links.count(); i++)
  {
    if(str.endsWith(links[i]))
    {
      currentIndex = i;
      previousAction->setEnabled(bool(i!=0));
      nextAction->setEnabled(bool(i+1 != links.count()));
      QString temp = cachedSelectedText;
      if(chaptersView->selectedItem())
        temp = chaptersView->selectedItem()->text(1);
      if(temp.toUInt() != i)
      {
        QListViewItem *item = chaptersView->findItem(QString::number(i),1);
        if(item != 0l)
        {
          chaptersView->blockSignals(true);
          chaptersView->setSelected(item,true);
          chaptersView->blockSignals(false);
        }
      }
      found = true;
      break;
    }
  }
  if(found == false) // some error
  {
    textBrowser->setSource(EcaasysHelpDir.filePath(links[0]));
    currentSource = EcaasysHelpDir.filePath(links[0]);
    qDebug("EcaasysHelp::slotSourceChanged():  Link mismatch \n Link: %s",str.ascii());
  }
  else
    currentSource = str;
}
void EcaasysHelp::previousLink()
{
  if(currentIndex > 0)
    --currentIndex;
  textBrowser->setSource(EcaasysHelpDir.filePath(links[currentIndex]));
}
void EcaasysHelp::nextLink()
{
  ++currentIndex;
  if(currentIndex >= links.count())
    currentIndex = links.count();
  textBrowser->setSource(EcaasysHelpDir.filePath(links[currentIndex]));
}
void EcaasysHelp::slotToggleSidebar(bool b)
{
  dock->setShown(b);
}
void EcaasysHelp::slotToggleSidebarAction(bool b)
{
  viewBrowseDock->blockSignals(true);
  viewBrowseDock->setOn(b);
  viewBrowseDock->blockSignals(false);
}
```

Участие в программе: отвечает за реакцию на нажатие 
Основная процедура: ecaasysApp()

- *Папка Ecaasys/ecaasys/element.h:* расширение – .h (3,96 КБ)

Изображение: 

Описание: Как и предыдущий файл, имеет расширении C++ и является заголовочных для element.cpp. Без element.h компиляция element.cpp – невозможна. Содержит объявления все конструкторов, функций, процедур, содержащихся в соответствующем файле с расширением .cpp.

Исходный код:

```
/*****************************************************************************
 *                                                                           *
 *   This program is free software; you can redistribute it and/or modify  *
 *   it under the terms of the GNU General Public License as published by  *
 *   the Free Software Foundation; either version 2 of the License, or     *
 *   (at your option) any later version.                                   *
 *                                                                           *
 *****************************************************************************/
#ifdef HAVE_CONFIG_H
# include <config.h>
#endif
#include "ecaasyshelp.h"
#include "htmldatafetcher.h"
#include <qpushbutton.h>
#include <qaction.h>
#include <qpixmap.h>
#include <qfile.h>
#include <qtextstream.h>
#include <qpopupmenu.h>
#include <qmenubar.h>
#include <qapplication.h>
#include <qlistview.h>
EcaasysHelp::EcaasysHelp(const QString& page)
{
  currentIndex = 0;
  dataFetcher = new HtmlDataFetcher();
  links = dataFetcher->fetchLinksToFiles(EcaasysHelpDir.filePath("index.html"));
  // set application icon
  setIcon (QPixmap(EcaasysSettings.BitmapDir + "big.ecaasys.xpm"));
  setCaption(tr("Ecaasys Help System"));
  textBrowser = new TextBrowser(this);
  textBrowser->setMinimumSize(400,200);
  setCentralWidget(textBrowser);
  setupActions();
  createSidebar();
  textBrowser->setSource(EcaasysHelpDir.filePath(links[0]));
  // ....................................
  if(!page.isEmpty())
    textBrowser->setSource(EcaasysHelpDir.filePath(page));
}
EcaasysHelp::~EcaasysHelp()
{}
void EcaasysHelp::setupActions()
{
  QToolBar *toolbar = new QToolBar(this,"main_toolbar");
  QMenuBar *bar = menuBar();
  statusBar();
  const QKeySequence ks = QKeySequence();
  QAction *quitAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "quit.png")),
                          tr("&Quit"), CTRL+Key_Q, this);
  QAction *backAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "back.png")),
                          tr("&Back"), ALT+Key_Left, this);
```

```
        QAction *forwardAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "forward.png")),
                                tr("&Forward"), ALT+Key_Right, this);
        QAction *homeAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "home.png")),
                              tr("&Home"),CTRL+Key_H,this);
        previousAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir +
"previous.png")),tr("&Previous"),
                              ks, this);
        nextAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "next.png")),
                        tr("&Next"), ks, this);
        viewBrowseDock = new QAction(tr("&Table of Contents"), 0, this);
        viewBrowseDock->setToggleAction(true);
        viewBrowseDock->setOn(true);
        viewBrowseDock->setStatusTip(tr("Enables/disables the table of contents"));
        viewBrowseDock->setWhatsThis(tr("Table of Contents\n\nEnables/disables the table of contents"));
        connect(quitAction,SIGNAL(activated()),qApp,SLOT(quit()));
        connect(backAction,SIGNAL(activated()),textBrowser,SLOT(backward()));
        connect(textBrowser,SIGNAL(backwardAvailable(bool)),backAction,SLOT(setEnabled(bool)));
        connect(forwardAction,SIGNAL(activated()),textBrowser,SLOT(forward()));
        connect(textBrowser,SIGNAL(forwardAvailable(bool)),forwardAction,SLOT(setEnabled(bool)));
        connect(homeAction,SIGNAL(activated()),textBrowser,SLOT(home()));
        connect(textBrowser,SIGNAL(sourceChanged(const QString &)),this,SLOT(slotSourceChanged(const
QString&)));
        connect(previousAction,SIGNAL(activated()),this,SLOT(previousLink()));
        connect(nextAction,SIGNAL(activated()),this,SLOT(nextLink()));
        connect(viewBrowseDock, SIGNAL(toggled(bool)), SLOT(slotToggleSidebar(bool)));
        backAction->addTo(toolbar);
        forwardAction->addTo(toolbar);
        toolbar->addSeparator();
        homeAction->addTo(toolbar);
        previousAction->addTo(toolbar);
        nextAction->addTo(toolbar);
        toolbar->addSeparator();
        quitAction->addTo(toolbar);
        QPopupMenu *fileMenu = new QPopupMenu(this);
        quitAction->addTo(fileMenu);
        QPopupMenu *viewMenu = new QPopupMenu(this);
        backAction->addTo(viewMenu);
        forwardAction->addTo(viewMenu);
        homeAction->addTo(viewMenu);
        previousAction->addTo(viewMenu);
        nextAction->addTo(viewMenu);
        viewMenu->insertSeparator();
        viewBrowseDock->addTo(viewMenu);
        QPopupMenu *helpMenu = new QPopupMenu(this);
        helpMenu->insertItem(tr("&About Qt"),qApp,SLOT(aboutQt()));
        bar->insertItem(tr("&File"), fileMenu );
        bar->insertItem(tr("&View"),viewMenu);
        bar->insertSeparator();
        bar->insertItem(tr("&Help"),helpMenu);
       }
       void EcaasysHelp::createSidebar()
       {
        dock = new QDockWindow(QDockWindow::InDock,this);
        dock->setResizeEnabled(true);
        dock->setCloseMode(QDockWindow::Always);
        connect(dock,SIGNAL(visibilityChanged(bool)),this,SLOT(slotToggleSidebarAction(bool)));
        chaptersView = new QListView(dock,"chapters_view");
        chaptersView->setRootIsDecorated(false);
        chaptersView->addColumn(tr("Contents"));
        chaptersView->setSorting(-1);
        chaptersView->setSelectionMode(QListView::Single);
        dock->setWidget(chaptersView);
```

```
  moveDockWindow(dock,QDockWindow::Left);
  QStringList l = dataFetcher->fetchChapterTexts(EcaasysHelpDir.filePath("index.html"));
  for(int i=l.count()-1;i>=0;i--)
    chaptersView->insertItem(new QListViewItem(chaptersView,l[i],QString::number(i+1)));
  QListViewItem *curItem = new QListViewItem(chaptersView,tr("Home"),QString::number(0));
  chaptersView->insertItem(curItem);
  chaptersView->setSelected(curItem,true);
  connect(chaptersView,SIGNAL(selectionChanged()),this,SLOT(displaySelectedChapter()));
}
void EcaasysHelp::displaySelectedChapter()
{
  if(chaptersView->selectedItem() == 0)
      return;
  uint y = chaptersView->selectedItem()->text(1).toUInt();
  Q_ASSERT(y < links.count());
  textBrowser->setSource(EcaasysHelpDir.filePath(links[y]));
  cachedSelectedText = chaptersView->selectedItem()->text(1);
}
//This slot updates next and previous actions i.e enabling/disabling
void EcaasysHelp::slotSourceChanged(const QString& _str)
{
  QString str(_str);
  // Remove '#*' chars in link since we don't check '#top,etc' while tracking previous actions
  int hashPos = str.findRev('#');
  if(hashPos != -1)
    str = str.left(hashPos);
  // Don't do anything if accessing same page
  if(str == currentSource)
    return;
  bool found = false;
  for(unsigned int i=0;i < links.count(); i++)
  {
    if(str.endsWith(links[i]))
    {
      currentIndex = i;
      previousAction->setEnabled(bool(i!=0));
      nextAction->setEnabled(bool(i+1 != links.count()));
      QString temp = cachedSelectedText;
      if(chaptersView->selectedItem())
         temp = chaptersView->selectedItem()->text(1);
      if(temp.toUInt() != i)
      {
        QListViewItem *item = chaptersView->findItem(QString::number(i),1);
        if(item != 0l)
        {
          chaptersView->blockSignals(true);
          chaptersView->setSelected(item,true);
          chaptersView->blockSignals(false);
        }
      }
      found = true;
      break;
    }
  }
  if(found == false) // some error
  {
    textBrowser->setSource(EcaasysHelpDir.filePath(links[0]));
    currentSource = EcaasysHelpDir.filePath(links[0]);
    qDebug("EcaasysHelp::slotSourceChanged():  Link mismatch \n Link: %s",str.ascii());
  }
  else
    currentSource = str;
```

```
}
void EcaasysHelp::previousLink()
{
  if(currentIndex > 0)
    --currentIndex;
  textBrowser->setSource(EcaasysHelpDir.filePath(links[currentIndex]));
}
void EcaasysHelp::nextLink()
{
  ++currentIndex;
  if(currentIndex >= links.count())
    currentIndex = links.count();
  textBrowser->setSource(EcaasysHelpDir.filePath(links[currentIndex]));
}
void EcaasysHelp::slotToggleSidebar(bool b)
{
  dock->setShown(b);
}
void EcaasysHelp::slotToggleSidebarAction(bool b)
{
  viewBrowseDock->blockSignals(true);
  viewBrowseDock->setOn(b);
  viewBrowseDock->blockSignals(false);
}
```

Участие в программе: да

- *Папка Ecaasys/ecaasys/main.cpp:* расширение – .cpp (18,7 КБ)

Изображение: 

Описание: Файл C++, отвечающий за все операции производимые с помощью элементов этой папки, среди которых, как системные, например, ввод и сохранение настроек проекта, проверка версионности системы, конвертация типов данных (Unicod, ASCII); так и непосредственно запуск работы папок диаграмм, диалогов, компонентов, подключение библиотек. Таким образом ecaasys.cpp собирает воедино папку ecaasys и организует ее работу.

Исходный код:

```
/*****************************************************************************
 *                                                                           *
 *   This program is free software; you can redistribute it and/or modify  *
 *   it under the terms of the GNU General Public License as published by  *
 *   the Free Software Foundation; either version 2 of the License, or     *
 *   (at your option) any later version.                                   *
 *                                                                           *
 *****************************************************************************/
#ifdef HAVE_CONFIG_H
# include <config.h>
#endif
#include "ecaasyshelp.h"
#include "htmldatafetcher.h"
#include <qpushbutton.h>
#include <qaction.h>
#include <qpixmap.h>
#include <qfile.h>
#include <qtextstream.h>
#include <qpopupmenu.h>
#include <qmenubar.h>
#include <qapplication.h>
```

```
#include <qlistview.h>
EcaasysHelp::EcaasysHelp(const QString& page)
{
  currentIndex = 0;
  dataFetcher = new HtmlDataFetcher();
  links = dataFetcher->fetchLinksToFiles(EcaasysHelpDir.filePath("index.html"));
  // set application icon
  setIcon (QPixmap(EcaasysSettings.BitmapDir + "big.ecaasys.xpm"));
  setCaption(tr("Ecaasys Help System"));
  textBrowser = new TextBrowser(this);
  textBrowser->setMinimumSize(400,200);
  setCentralWidget(textBrowser);
  setupActions();
  createSidebar();
  textBrowser->setSource(EcaasysHelpDir.filePath(links[0]));
  // ....................................
  if(!page.isEmpty())
    textBrowser->setSource(EcaasysHelpDir.filePath(page));
}
EcaasysHelp::~EcaasysHelp()
{}
void EcaasysHelp::setupActions()
{
  QToolBar *toolbar = new QToolBar(this,"main_toolbar");
  QMenuBar *bar = menuBar();
  statusBar();
  const QKeySequence ks = QKeySequence();
  QAction *quitAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "quit.png")),
                          tr("&Quit"), CTRL+Key_Q, this);
  QAction *backAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "back.png")),
                          tr("&Back"), ALT+Key_Left, this);
  QAction *forwardAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "forward.png")),
                            tr("&Forward"), ALT+Key_Right, this);
  QAction *homeAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "home.png")),
                          tr("&Home"),CTRL+Key_H,this);
  previousAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir +
"previous.png")),tr("&Previous"),
                          ks, this);
  nextAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "next.png")),
                      tr("&Next"), ks, this);
  viewBrowseDock = new QAction(tr("&Table of Contents"), 0, this);
  viewBrowseDock->setToggleAction(true);
  viewBrowseDock->setOn(true);
  viewBrowseDock->setStatusTip(tr("Enables/disables the table of contents"));
  viewBrowseDock->setWhatsThis(tr("Table of Contents\n\nEnables/disables the table of contents"));
  connect(quitAction,SIGNAL(activated()),qApp,SLOT(quit()));
  connect(backAction,SIGNAL(activated()),textBrowser,SLOT(backward()));
  connect(textBrowser,SIGNAL(backwardAvailable(bool)),backAction,SLOT(setEnabled(bool)));
  connect(forwardAction,SIGNAL(activated()),textBrowser,SLOT(forward()));
  connect(textBrowser,SIGNAL(forwardAvailable(bool)),forwardAction,SLOT(setEnabled(bool)));
  connect(homeAction,SIGNAL(activated()),textBrowser,SLOT(home()));
  connect(textBrowser,SIGNAL(sourceChanged(const QString &)),this,SLOT(slotSourceChanged(const
QString&)));
  connect(previousAction,SIGNAL(activated()),this,SLOT(previousLink()));
  connect(nextAction,SIGNAL(activated()),this,SLOT(nextLink()));
  connect(viewBrowseDock, SIGNAL(toggled(bool)), SLOT(slotToggleSidebar(bool)));
  backAction->addTo(toolbar);
  forwardAction->addTo(toolbar);
  toolbar->addSeparator();
  homeAction->addTo(toolbar);
  previousAction->addTo(toolbar);
  nextAction->addTo(toolbar);
```

```
  toolbar->addSeparator();
  quitAction->addTo(toolbar);
  QPopupMenu *fileMenu = new QPopupMenu(this);
  quitAction->addTo(fileMenu);
  QPopupMenu *viewMenu = new QPopupMenu(this);
  backAction->addTo(viewMenu);
  forwardAction->addTo(viewMenu);
  homeAction->addTo(viewMenu);
  previousAction->addTo(viewMenu);
  nextAction->addTo(viewMenu);
  viewMenu->insertSeparator();
  viewBrowseDock->addTo(viewMenu);
  QPopupMenu *helpMenu = new QPopupMenu(this);
  helpMenu->insertItem(tr("&About Qt"),qApp,SLOT(aboutQt()));
  bar->insertItem(tr("&File"), fileMenu );
  bar->insertItem(tr("&View"),viewMenu);
  bar->insertSeparator();
  bar->insertItem(tr("&Help"),helpMenu);
}
void EcaasysHelp::createSidebar()
{
  dock = new QDockWindow(QDockWindow::InDock,this);
  dock->setResizeEnabled(true);
  dock->setCloseMode(QDockWindow::Always);
  connect(dock,SIGNAL(visibilityChanged(bool)),this,SLOT(slotToggleSidebarAction(bool)));
  chaptersView = new QListView(dock,"chapters_view");
  chaptersView->setRootIsDecorated(false);
  chaptersView->addColumn(tr("Contents"));
  chaptersView->setSorting(-1);
  chaptersView->setSelectionMode(QListView::Single);
  dock->setWidget(chaptersView);
  moveDockWindow(dock,QDockWindow::Left);
  QStringList l = dataFetcher->fetchChapterTexts(EcaasysHelpDir.filePath("index.html"));
  for(int i=l.count()-1;i>=0;i--)
    chaptersView->insertItem(new QListViewItem(chaptersView,l[i],QString::number(i+1)));
  QListViewItem *curItem = new QListViewItem(chaptersView,tr("Home"),QString::number(0));
  chaptersView->insertItem(curItem);
  chaptersView->setSelected(curItem,true);
  connect(chaptersView,SIGNAL(selectionChanged()),this,SLOT(displaySelectedChapter()));
}
void EcaasysHelp::displaySelectedChapter()
{
  if(chaptersView->selectedItem() == 0)
      return;
  uint y = chaptersView->selectedItem()->text(1).toUInt();
  Q_ASSERT(y < links.count());
  textBrowser->setSource(EcaasysHelpDir.filePath(links[y]));
  cachedSelectedText = chaptersView->selectedItem()->text(1);
}
//This slot updates next and previous actions i.e enabling/disabling
void EcaasysHelp::slotSourceChanged(const QString& _str)
{
  QString str(_str);
  // Remove '#*' chars in link since we don't check '#top,etc' while tracking previous actions
  int hashPos = str.findRev('#');
  if(hashPos != -1)
    str = str.left(hashPos);
  // Don't do anything if accessing same page
  if(str == currentSource)
    return;
  bool found = false;
  for(unsigned int i=0;i < links.count(); i++)
```

```
  {
    if(str.endsWith(links[i]))
    {
      currentIndex = i;
      previousAction->setEnabled(bool(i!=0));
      nextAction->setEnabled(bool(i+1 != links.count()));
      QString temp = cachedSelectedText;
      if(chaptersView->selectedItem())
        temp = chaptersView->selectedItem()->text(1);
      if(temp.toUInt() != i)
      {
        QListViewItem *item = chaptersView->findItem(QString::number(i),1);
        if(item != 0l)
        {
          chaptersView->blockSignals(true);
          chaptersView->setSelected(item,true);
          chaptersView->blockSignals(false);
        }
      }
      found = true;
      break;
    }
  }
  if(found == false) // some error
  {
    textBrowser->setSource(EcaasysHelpDir.filePath(links[0]));
    currentSource = EcaasysHelpDir.filePath(links[0]);
    qDebug("EcaasysHelp::slotSourceChanged():  Link mismatch \n Link: %s",str.ascii());
  }
  else
    currentSource = str;
}
void EcaasysHelp::previousLink()
{
  if(currentIndex > 0)
    --currentIndex;
  textBrowser->setSource(EcaasysHelpDir.filePath(links[currentIndex]));
}
void EcaasysHelp::nextLink()
{
  ++currentIndex;
  if(currentIndex >= links.count())
    currentIndex = links.count();
  textBrowser->setSource(EcaasysHelpDir.filePath(links[currentIndex]));
}
void EcaasysHelp::slotToggleSidebar(bool b)
{
  dock->setShown(b);
}
void EcaasysHelp::slotToggleSidebarAction(bool b)
{
  viewBrowseDock->blockSignals(true);
  viewBrowseDock->setOn(b);
  viewBrowseDock->blockSignals(false);
}
```

Участие в программе: отвечает за первоначальные насройки интерфейса Ecaasys

Основная процедура: ecaasysApp()

- *Папка Ecaasys/ecaasys/main.h:* расширение – .h (3,04 КБ)

Изображение: 

Описание: Как и предыдущий файл, имеет расширении C++ и является заголовочных для main.cpp. Без main.h компиляция main.cpp – невозможна. Содержит объявления все конструкторов, функций, процедур, содержащихся в соответствующем файле с расширением .cpp.

Исходный код:

```
/*****************************************************************************
 *                                                                           *
 *   This program is free software; you can redistribute it and/or modify  *
 *   it under the terms of the GNU General Public License as published by  *
 *   the Free Software Foundation; either version 2 of the License, or     *
 *   (at your option) any later version.                                   *
 *                                                                           *
 *****************************************************************************/
#ifdef HAVE_CONFIG_H
# include <config.h>
#endif
#include "ecaasyshelp.h"
#include "htmldatafetcher.h"
#include <qpushbutton.h>
#include <qaction.h>
#include <qpixmap.h>
#include <qfile.h>
#include <qtextstream.h>
#include <qpopupmenu.h>
#include <qmenubar.h>
#include <qapplication.h>
#include <qlistview.h>
EcaasysHelp::EcaasysHelp(const QString& page)
{
  currentIndex = 0;
  dataFetcher = new HtmlDataFetcher();
  links = dataFetcher->fetchLinksToFiles(EcaasysHelpDir.filePath("index.html"));
  // set application icon
  setIcon (QPixmap(EcaasysSettings.BitmapDir + "big.ecaasys.xpm"));
  setCaption(tr("Ecaasys Help System"));
  textBrowser = new TextBrowser(this);
  textBrowser->setMinimumSize(400,200);
  setCentralWidget(textBrowser);
  setupActions();
  createSidebar();
  textBrowser->setSource(EcaasysHelpDir.filePath(links[0]));
  // ....................................
  if(!page.isEmpty())
    textBrowser->setSource(EcaasysHelpDir.filePath(page));
}
EcaasysHelp::~EcaasysHelp()
{}
void EcaasysHelp::setupActions()
{
  QToolBar *toolbar = new QToolBar(this,"main_toolbar");
  QMenuBar *bar = menuBar();
  statusBar();
  const QKeySequence ks = QKeySequence();
  QAction *quitAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "quit.png")),
                          tr("&Quit"), CTRL+Key_Q, this);
  QAction *backAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "back.png")),
                          tr("&Back"), ALT+Key_Left, this);
```

```
        QAction *forwardAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "forward.png")),
                                tr("&Forward"), ALT+Key_Right, this);
        QAction *homeAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "home.png")),
                              tr("&Home"),CTRL+Key_H,this);
        previousAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir +
"previous.png")),tr("&Previous"),
                              ks, this);
        nextAction = new QAction(QIconSet(QPixmap(EcaasysSettings.BitmapDir + "next.png")),
                        tr("&Next"), ks, this);
        viewBrowseDock = new QAction(tr("&Table of Contents"), 0, this);
        viewBrowseDock->setToggleAction(true);
        viewBrowseDock->setOn(true);
        viewBrowseDock->setStatusTip(tr("Enables/disables the table of contents"));
        viewBrowseDock->setWhatsThis(tr("Table of Contents\n\nEnables/disables the table of contents"));
        connect(quitAction,SIGNAL(activated()),qApp,SLOT(quit()));
        connect(backAction,SIGNAL(activated()),textBrowser,SLOT(backward()));
        connect(textBrowser,SIGNAL(backwardAvailable(bool)),backAction,SLOT(setEnabled(bool)));
        connect(forwardAction,SIGNAL(activated()),textBrowser,SLOT(forward()));
        connect(textBrowser,SIGNAL(forwardAvailable(bool)),forwardAction,SLOT(setEnabled(bool)));
        connect(homeAction,SIGNAL(activated()),textBrowser,SLOT(home()));
        connect(textBrowser,SIGNAL(sourceChanged(const QString &)),this,SLOT(slotSourceChanged(const
QString&)));
        connect(previousAction,SIGNAL(activated()),this,SLOT(previousLink()));
        connect(nextAction,SIGNAL(activated()),this,SLOT(nextLink()));
        connect(viewBrowseDock, SIGNAL(toggled(bool)), SLOT(slotToggleSidebar(bool)));
        backAction->addTo(toolbar);
        forwardAction->addTo(toolbar);
        toolbar->addSeparator();
        homeAction->addTo(toolbar);
        previousAction->addTo(toolbar);
        nextAction->addTo(toolbar);
        toolbar->addSeparator();
        quitAction->addTo(toolbar);
        QPopupMenu *fileMenu = new QPopupMenu(this);
        quitAction->addTo(fileMenu);
        QPopupMenu *viewMenu = new QPopupMenu(this);
        backAction->addTo(viewMenu);
        forwardAction->addTo(viewMenu);
        homeAction->addTo(viewMenu);
        previousAction->addTo(viewMenu);
        nextAction->addTo(viewMenu);
        viewMenu->insertSeparator();
        viewBrowseDock->addTo(viewMenu);
        QPopupMenu *helpMenu = new QPopupMenu(this);
        helpMenu->insertItem(tr("&About Qt"),qApp,SLOT(aboutQt()));
        bar->insertItem(tr("&File"), fileMenu );
        bar->insertItem(tr("&View"),viewMenu);
        bar->insertSeparator();
        bar->insertItem(tr("&Help"),helpMenu);
       }
       void EcaasysHelp::createSidebar()
       {
        dock = new QDockWindow(QDockWindow::InDock,this);
        dock->setResizeEnabled(true);
        dock->setCloseMode(QDockWindow::Always);
        connect(dock,SIGNAL(visibilityChanged(bool)),this,SLOT(slotToggleSidebarAction(bool)));
        chaptersView = new QListView(dock,"chapters_view");
        chaptersView->setRootIsDecorated(false);
        chaptersView->addColumn(tr("Contents"));
        chaptersView->setSorting(-1);
        chaptersView->setSelectionMode(QListView::Single);
        dock->setWidget(chaptersView);
```

```
  moveDockWindow(dock,QDockWindow::Left);
  QStringList l = dataFetcher->fetchChapterTexts(EcaasysHelpDir.filePath("index.html"));
  for(int i=l.count()-1;i>=0;i--)
    chaptersView->insertItem(new QListViewItem(chaptersView,l[i],QString::number(i+1)));
  QListViewItem *curItem = new QListViewItem(chaptersView,tr("Home"),QString::number(0));
  chaptersView->insertItem(curItem);
  chaptersView->setSelected(curItem,true);
  connect(chaptersView,SIGNAL(selectionChanged()),this,SLOT(displaySelectedChapter()));
}
void EcaasysHelp::displaySelectedChapter()
{
  if(chaptersView->selectedItem() == 0)
      return;
  uint y = chaptersView->selectedItem()->text(1).toUInt();
  Q_ASSERT(y < links.count());
  textBrowser->setSource(EcaasysHelpDir.filePath(links[y]));
  cachedSelectedText = chaptersView->selectedItem()->text(1);
}
//This slot updates next and previous actions i.e enabling/disabling
void EcaasysHelp::slotSourceChanged(const QString& _str)
{
  QString str(_str);
  // Remove '#*' chars in link since we don't check '#top,etc' while tracking previous actions
  int hashPos = str.findRev('#');
  if(hashPos != -1)
    str = str.left(hashPos);
  // Don't do anything if accessing same page
  if(str == currentSource)
    return;
  bool found = false;
  for(unsigned int i=0;i < links.count(); i++)
  {
    if(str.endsWith(links[i]))
    {
      currentIndex = i;
      previousAction->setEnabled(bool(i!=0));
      nextAction->setEnabled(bool(i+1 != links.count()));
      QString temp = cachedSelectedText;
      if(chaptersView->selectedItem())
         temp = chaptersView->selectedItem()->text(1);
      if(temp.toUInt() != i)
      {
        QListViewItem *item = chaptersView->findItem(QString::number(i),1);
        if(item != 0l)
        {
          chaptersView->blockSignals(true);
          chaptersView->setSelected(item,true);
          chaptersView->blockSignals(false);
        }
      }
      found = true;
      break;
    }
  }
  if(found == false) // some error
  {
    textBrowser->setSource(EcaasysHelpDir.filePath(links[0]));
    currentSource = EcaasysHelpDir.filePath(links[0]);
    qDebug("EcaasysHelp::slotSourceChanged():  Link mismatch \n Link: %s",str.ascii());
  }
  else
    currentSource = str;
```

```
}
void EcaasysHelp::previousLink()
{
  if(currentIndex > 0)
    --currentIndex;
  textBrowser->setSource(EcaasysHelpDir.filePath(links[currentIndex]));
}
void EcaasysHelp::nextLink()
{
  ++currentIndex;
  if(currentIndex >= links.count())
    currentIndex = links.count();
  textBrowser->setSource(EcaasysHelpDir.filePath(links[currentIndex]));
}
void EcaasysHelp::slotToggleSidebar(bool b)
{
  dock->setShown(b);
}
void EcaasysHelp::slotToggleSidebarAction(bool b)
{
  viewBrowseDock->blockSignals(true);
  viewBrowseDock->setOn(b);
  viewBrowseDock->blockSignals(false);
}
```

Участие в программе: да

- *Папка Ecaasys/ecaasys/Makefile.am:* расширение – am (3,94 КБ)

Изображение: 

Описание: Файл Makefile.am – выходной файл процесса установки с необходимыми для нее параметрами, не требующий вмешательства программиста, т.к. система производит его сама, исходя из установочных файлов. Является обязательным компонентом каждой директории.

Исходный код:

```
SUBDIRS = ecaasys ecaasys-edit ecaasys-help ecaasys-transcalc ecaasys-filter ecaasys-lib \
   ecaasys-attenuator contrib $(RELEASEDIRS)
EXTRA_DIST = autogen.sh depcomp PLATFORMS RELEASE Info.plist
# MacOSX specific installation of applications
if COND_MACOSX
app_PROGS = $(top_builddir)/ecaasys/ecaasys \
  $(top_builddir)/ecaasys-attenuator/ecaasysattenuator \
  $(top_builddir)/ecaasys-filter/ecaasysfilter \
  $(top_builddir)/ecaasys-help/ecaasyshelp \
  $(top_builddir)/ecaasys-lib/ecaasyslib \
  $(top_builddir)/ecaasys-edit/ecaasysedit \
  $(top_builddir)/ecaasys-transcalc/ecaasystrans
install-exec-hook: mac-install-apps mac-install-framework
mac-install-apps:
   @echo "Creating MacOSX applications...";
   @list='$(app_PROGS)'; for file in $$list; do \
   app=`basename $$file` && \
   $(mkinstalldirs) $(bindir)/$$app.app/Contents && \
   $(mkinstalldirs) $(bindir)/$$app.app/Contents/Resources && \
   $(mkinstalldirs) $(bindir)/$$app.app/Contents/MacOS && \
   $(install_sh_PROGRAM) $$file $(bindir)/$$app.app/Contents/MacOS/ && \
   strip $(bindir)/$$app.app/Contents/MacOS/$$app && \
   case $$app in \
     ecaasys)                        desc="Ecaasys";; \
```

```
    ecaasysattenuator) desc="Ecaasys Attenuator";; \
    ecaasysfilter)     desc="Ecaasys Filter";; \
    ecaasyshelp)       desc="Ecaasys Help";; \
    ecaasyslib)        desc="Ecaasys Library";; \
    ecaasysedit)       desc="Ecaasys Editor";; \
    ecaasystrans)      desc="Ecaasys Transcalc";; \
  esac && \
  cat $(srcdir)/Info.plist | \
    sed -e "s/@version@/$(PACKAGE_VERSION)/g" | \
    sed -e "s/@name@/$$desc/g" | \
    sed -e "s/@exec@/$$app/g" > \
    $(bindir)/$$app.app/Contents/Info.plist && \
  $(install_sh_DATA) $(srcdir)/ecaasys/bitmaps/$$app.icns \
    $(bindir)/$$app.app/Contents/Resources/application.icns && \
  echo "#!/bin/sh\n$(bindir)/$$app.app/Contents/MacOS/$$app \$$*\n" > \
    $(bindir)/$$app && \
  chmod +x $(bindir)/$$app; \
  done
mac-install-framework:
  @echo "Copying and setting up MacOSX/Qt framework...";
  @if test -e "$(QTDIR)/lib/libqt.3.dylib"; then \
    qtlib="$(QTDIR)/lib/libqt.3.dylib"; fi && \
  if test -e "$(QTDIR)/lib/libqt-mt.3.dylib"; then \
    qtlib="$(QTDIR)/lib/libqt-mt.3.dylib"; fi && \
  if ! test -z "$$qtlib"; then \
    $(mkinstalldirs) "$(bindir)/ecaasys.app/Contents/Frameworks"; \
    cp "$$qtlib" "$(bindir)/ecaasys.app/Contents/Frameworks/"; \
    qtlib=`basename $$qtlib`; \
    install_name_tool -id "@executable_path/../Frameworks/$$qtlib" \
      "$(bindir)/ecaasys.app/Contents/Frameworks/$$qtlib"; \
    list='$(app_PROGS)'; for file in $$list; do \
    app=`basename $$file` && \
    install_name_tool -change $$qtlib \
      "@executable_path/../../../ecaasys.app/Contents/Frameworks/$$qtlib" \
      "$(bindir)/$$app.app/Contents/MacOS/$$app"; \
    done; \
  fi
uninstall-hook:
  @list='$(app_PROGS)'; for file in $$list; do \
  app=`basename $$file` && \
  rm -rf $(bindir)/$$app.app; \
  done
else
install-exec-hook:
uninstall-hook:
endif
CLEANFILES = *~ *.rej *.orig
MAINTAINERCLEANFILES = aclocal.m4 config.h.in configure Makefile.in \
      stamp-h.in stamp-h[0-9].in
DISTCLEANFILES = config.cache config.log
```

Участие в программе: да