На правах рукописи

САРАМУД МИХАИЛ ВЛАДИМИРОВИЧ

# МОДЕЛЬНО – АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ АНАЛИЗА ОТКАЗОУСТОЙЧИВОСТИ ПРОГРАММНЫХ КОМПЛЕКСОВ ВСТРАИВАЕМЫХ СИСТЕМ УПРАВЛЕНИЯ РЕАЛЬНОГО ВРЕМЕНИ

05.13.01 – Системный анализ, управление и обработка информации
(космические и информационные технологии)

**АВТОРЕФЕРАТ**
диссертации на соискание ученой степени
кандидата технических наук

Красноярск – 2018

Работа выполнена в ФГБОУ ВО «Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева» (СибГУ им. М.Ф. Решетнева) на кафедре системного анализа и исследования операций

Научный руководитель: доктор технических наук, профессор
**Ковалев Игорь Владимирович**

Официальные оппоненты: **Дулесов Александр Сергеевич**
доктор технических наук, доцент
ФГБОУ ВО «Хакасский государственный университет им. Н.Ф. Катанова», г. Абакан, профессор кафедры информационных технологий и систем

**Шеенок Дмитрий Александрович**
кандидат технических наук
ООО "Сибирские интеграционные системы", г. Красноярск, старший аналитик

Ведущая организация: ФГБОУ ВО «Томский государственный университет систем управления и радиоэлектроники»

Защита состоится «14» сентября 2018 г. в __ часов на заседании диссертационного совета Д 212.249.05, созданного на базе Сибирского государственного университета науки и технологий имени академика М.Ф. Решетнева по адресу: 660037 г. Красноярск, проспект имени газеты «Красноярский рабочий», 31.

С диссертацией можно ознакомиться в библиотеке Сибирского государственного университета науки и технологий имени академика М.Ф. Решетнева и на сайте https://www.sibsau.ru

Автореферат разослан «___» _____ 2018 г.

Ученый секретарь
диссертационного совета,
кандидат технических наук, доцент — Панфилов Илья Александрович

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность работы.** В последние годы активно развиваются отрасли, требующие надежных, отказоустойчивых систем управления реального времени. К их числу относятся высокотехнологичные производства, использующие композитные и опасные материалы, автономные беспилотные объекты - от мультироторных систем до автомобилей с функцией автопилота и моторизованных кресел с голосовым управлением для людей с ограниченными возможностями. Программное обеспечение (ПО) является неотъемлемой частью современных систем управления, однако, лишь простое ПО может быть гарантированно создано без ошибок. Современное ПО систем управления используется для решения все более сложных задач, лавинообразно возрастает объем обрабатываемой информации. С возрастанием сложности ПО растет и вероятность возникновения ошибок в нем.

Традиционно, методы повышения надежности встраиваемых систем управления (СУ) были основаны на повышении надежности аппаратных средств, для чего существуют классические инструменты и подходы, обеспечивающие требуемый уровень надежного функционирования СУ, включая применение дублирования компонентов СУ, что не имеет смысла для ПО, поскольку в этом случае будут дублироваться и ошибки в программах. В настоящее время актуальна разработка и применение методов программной отказоустойчивости, что связано с усложнением и укрупнением программных систем, увеличением количества функциональных модулей СУ, что приводит к росту вероятности отказа системы из-за отказа одного из модулей. Рост же надежности современных аппаратных средств СУ обеспечивается благодаря развитию технологических процессов производства радиоэлектронной аппаратуры, а благодаря снижению стоимости упрощается дублирование аппаратных компонентов.

Особенно критична надежность ПО в системах реального времени. Если в обычной системе в случае возникновения сбоя есть время, чтобы его обработать, восстановить состояние системы до сбоя и повторить вычисления, то в случае работы в режиме реального времени нет возможности восстановления и повторного исполнения задачи, а по истечении времени на выполнение данная задача будет завершена принудительно.

Все более актуальным становится вопрос научно-обоснованной разработки отказоустойчивых систем управления. Важность повышения надежности программного обеспечения обусловлена тем, что оно выполняет основные функции системного управления обработкой данных, и его отказы в работе могут оказать существенное влияние на функционирование систем обработки данных и управления в целом.

Создание отказоустойчивого ПО невозможно без средств анализа надежности. Если не проводить анализ надежности на ранних этапах разработки, возможна ситуация, когда уже разработанная программная система при функционировании не обеспечивает заданных параметров надежности СУ, что приводит к необходимости повторной разработки ПО с использованием других моделей и алгоритмов. Это обстоятельство создает риск существенного удорожания и увеличения времени разработки. Возможность осуществить анализ надежности ПО на более ранних этапах проектирования системы делает ее разработку более предсказуемой и эффективной.

Таким образом, создание модельно-алгоритмического обеспечения анализа отказоустойчивости программных комплексов встраиваемых систем управления реального времени является актуальным.

**Целью настоящего диссертационного исследования является** повышение отказоустойчивости разрабатываемого программного обеспечения за счет получения оценок надежности программных модулей на этапе проектирования и алгоритмов принятия решения в мультиверсионных системах.

Для достижения поставленной цели решались следующие задачи.

- анализ существующих моделей отказоустойчивых систем управления, механизмов обеспечения надежности СУ с применением программной избыточности;
- разработка типовой структуры мультиверсионной СУ на основе операционной системы реального времени (ОСРВ);
- разработка комбинированного селективного алгоритма выбора мультиверсионной модели повышения отказоустойчивости для реализации программного комплекса СУ с требуемыми характеристиками надежности. Алгоритм основан на модифицированной модели деревьев сбоев и впервые предложенной модели расчета величины корректности;
- разработка новых и модификация существующих алгоритмов принятия решения в мультиверсионных системах, позволяющих повысить отказоустойчивость;
- реализация имитационной среды моделирования (анализа надежностных характеристик) мультиверсионного программного комплекса, для возможности тестирования и получения характеристик исследуемых моделей, предложенных модифицированных алгоритмов голосования, возможности их сравнения в одинаковых условиях с классическими алгоритмами;
- реализация мультиверсионной среды исполнения реального времени для валидации предложенных моделей и алгоритмов;
- разработка модели прогнозирования надежности на основе результатов тестирования программных модулей.

**Область исследования.** Работа выполнена в соответствии со следующими пунктами паспорта специальности 05.13.01:

- разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений и обработки информации;
- методы и алгоритмы прогнозирования и оценки эффективности, качества и надежности сложных систем.

**Методы исследования.**

При выполнении работы использовались методы и подходы теории вероятностей, имитационного моделирования, методы анализа и проектирования информационных систем, системного анализа, теории надежности, мультиверсионного проектирования программного обеспечения отказоустойчивых систем управления, методы теории графов.

**Научная новизна работы.**

1. Впервые сформирована типовая структура мультиверсионной системы управления реального времени, в том числе ее программная составляющая, которая позволяет: разработать имитационную среду моделирования, в которой тестируются отдельные программные модули, определяются их характеристики и общие

характеристики системы при их применении; реализовать мультиверсионную среду исполнения реального времени на основе ОСРВ, применимую на большинстве современных одноплатных компьютеров.

2. Предложен и реализован комбинированный селективный алгоритм оценки эффективности мультиверсионных моделей, впервые позволивший получить верхнюю и нижнюю границы надежности. Нижняя граница рассчитывается с применением модели «деревьев сбоев», верхняя – с применением модели корректности мультиверсионной системы. Оригинальность данных подходов заключается в расчете характеристик надежности разрабатываемой системы, с учетом характеристик составляющих ее программных и аппаратных модулей, и структуры системы (используя знания о структуре системы, в зависимости от применяемых в ней мультиверсионных моделей).

3. Предложены и реализованы модификации алгоритмов голосования согласованным большинством и нечеткого голосования согласованным большинством. Для повышения устойчивости алгоритмов к межверсионным ошибкам программных модулей, по сравнению с невзвешенными аналогами, введены веса версий и элемент забывания.

4. Разработан новый алгоритм предсказания времени наработки на отказ на основе данных автоматизированного тестирования. Алгоритм основан на экспоненциальном распределении времени нахождения ошибок в программной системе и на основании данных об уже выявленных ошибках позволяет предсказать время работы системы до проявления следующей ошибки, что и является временем наработки на отказ. Алгоритм позволяет узнать предполагаемый диапазон значения времени наработки на отказ и вероятность попадания в предсказанный диапазон.

**Значение для теории** состоит в разработке новых взвешенных алгоритмов голосования с забыванием, методик предсказания времени наработки на отказ, новых методов оценки надежности мультиверсионных систем. Результаты, полученные при выполнении диссертационной работы, создают теоретическую основу для разработки новых технологий проектирования мультиверсионного программного обеспечения сложных систем управления. Благодаря предложенным методам и алгоритмам обеспечивается повышение эффективности процессов обработки информации и управления.

**Практическая ценность.**

Предложенная в диссертации технология разработки отказоустойчивых программных комплексов систем управления позволяет автоматизировать процесс решения задачи компоновки состава мультиверсионного программного комплекса, выбрать алгоритмы принятия решения, сформировать требования к надежности составляющих систему программных блоков. Применения технологии позволяет оценить характеристики системы на ранних этапах разработки, что не только позволит выбрать оптимальные алгоритмы и программные модули, но и существенно уменьшит риск реализации системы, которая на практике не будет удовлетворять поставленным требованиям надежности.

Типовая структура отказоустойчивого программного комплекса позволит создавать мультиверсионную среду исполнения на базе ОСРВ FreeRTOS, применимую на большинстве современных одноплатных компьютеров и платах разработчиков.

Применение модифицированных алгоритмов голосования не только увеличивает устойчивость мультиверсионных систем к межверсионным ошибкам, но и способствует увеличению надежности системы в целом.

Применение методики предсказания времени наработки на отказ существенно сокращает время на этапе тестирования, поскольку дает возможность тестировать систему не в масштабе реального времени, а настолько быстрее, насколько производительна тестовая платформа, что позволяет гарантировать время наработки на отказ в месяцах, затрачивая на тестирование на порядок меньшие промежутки времени.

Реализованная имитационная среда моделирования мультиверсионных программных комплексов позволяет тестировать и получать характеристики исследуемых моделей, предложенных модифицированных алгоритмов голосования, а также дает возможность их сравнения в одинаковых условиях с классическими алгоритмами. Оригинальность данной методики заключается в сравнении моделей и алгоритмов с помощью анализа их реальной работы в рамках имитационной среды моделирования, которая симулирует выходы версий с заданными характеристиками и собирает статистику работы всех алгоритмов на каждой итерации.

**Достоверность полученных результатов** подтверждается результатами моделирования в имитационной среде и реализованной мультиверсионной средой исполнения реального времени, проведенными тестами системы на реальной задаче. Непротиворечивостью использованных моделей, методов и численных результатов моделирования и расчетов.

**Реализация результатов работы.** В диссертационной работе были разработаны шесть программных систем, четыре из них прошли регистрацию в Роспатенте.

Диссертационная работа выполнена в рамках проектов:

РФФИ № 16-47-242143 «Мультиверсионная среда исполнения алгоритмов управления автономными беспилотными объектами»;

Госзадание № 2.2867.2017/ПЧ «Технология организации жизненного цикла кроссплатформенного программного обеспечения бортовой аппаратуры беспилотных летательных объектов».

**Апробация работы.** Основные положения и результаты работы прошли всестороннюю апробацию на международных конференциях.

1. Международная научно-практическая конференция «Современное состояние науки и техники» ССНиТ, Сочи, 2016 г.
2. XIX Международная научно-практическая конференция, посвященная 55-летию Сибирского государственного аэрокосмического университета имени академика М. Ф. Решетнева “Решетнёвские чтения”, Красноярск, СибГАУ, 2015 г.
3. XVIII Международная научная конференция, посвященная 90-летию со дня рождения генерального конструктора ракетно-космических систем академика М. Ф. Решетнева “Решетнёвские чтения”, Красноярск, СибГАУ, 2014 г.
4. Международная научно-практическая конференция «Теоретические и прикладные проблемы науки и образования в 21 веке», Тамбов, 2012 г.
5. Международная научная конференция «Современные проблемы математического моделирования, обработки изображений и параллельных вычислений 2017», пос. Дивноморское, Геленджик, 2017 г.

6. XIII Международная научно-практическая конференция, посвященная дню космонавтики «АКТУАЛЬНЫЕ ПРОБЛЕМЫ АВИАЦИИ И КОСМОНАВТИКИ», Красноярск, 2017 г.

7. XIII Международная научно-техническая конференция «Динамика технических систем» (ДТС-2017), Ростов-на-Дону, 2017 г.

8. European Conference on Electrical Engineering and Computer Science (EECS 2017), Берн, Швейцария, 2017 г.

9. INTE 2017 International Conference on New Horizons in Education, Берлин, Германия, 2017 г.

**Публикации.** По теме диссертации опубликовано 19 печатных работ, из них пять статей в журналах перечня ВАК РФ и шесть в изданиях, индексируемых в международных базах цитирования Web of Science и/или Scopus. Получено четыре свидетельства о регистрации программных систем в Роспатенте.

**Структура и объем работы.** Диссертация состоит из введения, пяти глав, заключения, списка литературы из 152 наименований.

## ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

**Во введении** представлена общая характеристика проблемы, обоснована актуальность выбранной темы, определены цель и задачи исследования. Сформулированы основные положения, выносимые на защиту, научная новизна и практическая значимость полученных результатов.

**В первой главе** на основании трудов различных исследователей рассмотрены теоретические и практические аспекты проектирования отказоустойчивых систем управления, в том числе программной составляющей критичных по надежности систем управления, надежностных характеристик программных модулей, механизмов обеспечения надежности программ с помощью введения избыточности. Изучена как классическая теория (Avizienis A.A., Lyu M.R., Boehm B.W.), в работах этих авторов содержатся фундаментальные основы программной надежности, так и последние достижения в области решения прикладных задач: работы Липаева В.В., посвящённые надежности функционирования программ реального времени и различным методам, технологиям и инструментальным средствам разработки сложных комплексов программ реального времени; работы Ковалева И.В., посвящённые различным мультиверсионным подходам повышения отказоустойчивости ПО и методам оценки его надежностных характеристик; работы Царева Р.Ю., посвящённые методам и инструментам формирования оптимального состава программных комплексов; работы Jie Xu, в которых описан принцип t/(n-1) диагностируемости и предложен t/(n-1) алгоритм принятия решения в мультиверсионных системах, основанный на данном принципе. Однако, эти работы не могут без модификаций быть применены для решения задачи получения достоверных оценок надежности создаваемых встраиваемых СУ реального времени на ранних этапах разработки, не существует реализованных программных инструментов исследования влияния характеристик отдельных модулей, их связей и структуры системы на ее общие характеристики надежности. Существующие алгоритмы принятия решения в мультиверсионных системах обладают низкой устойчивостью к межверсионным ошибкам.

Проанализирован жизненный цикл разработки программной составляющей критичных по надежности систем управления, определены действия на каждом этапе, специфичные для создания отказоустойчивых систем.

Рассмотрены различные модели надежности ПО: модель Шумана, Джелинского – Моранды, Вейса, Нельсона, модели, предложенные Коркорэном и др., позволяющие изучить характеристики надежности, атрибуты, средства, нарушения надежности, сбои и отказы.

Проведено исследование методов повышения надежности программных систем, в том числе анализ моноверсионных моделей и мультиверсионных, среди которых на текущий момент наиболее распространены: модель N-версионного программирования (NVP), модель N-версионного программирования с самопроверкой (NSCP), модель восстанавливающихся блоков (RB), модель согласованных восстанавливающихся блоков (CRB) и модель t/(n-1)-версионного программирования.

На основании сравнительного анализа сделан вывод о том, что наиболее эффективными являются именно мультиверсионные модели, однако среди них нет однозначно лучшей, выбор конкретной зависит от требований к системе и характеристик составляющих ее модулей, как программных, так и аппаратных. Возникает задача обоснованного выбора модели повышения отказоустойчивости.

Для решения задачи выбора, на этапе принятия решения, необходимо сравнение различных моделей в одинаковых условиях, с обязательной возможностью исследования в рамках работающей среды исполнения отказоустойчивого ПО. Для создания имитационной среды, моделирующей работу программного комплекса, необходимо сформировать типовую структуру системы управления, включающую ее программную составляющую, аппаратные и программные интерфейсы.

**Во второй главе** рассмотрена модель системы управления и ее программной составляющей. Проведен анализ необходимости применения ОС на целевых аппаратных платформах, сделан вывод о предпочтительности использования в качестве основы среды исполнения существующей операционной системы реального времени (ОСРВ) с открытым исходным кодом и возможностью его модификации. Это решение обеспечит систему многозадачностью, временной базой, механизмом обмена данными между задачами (очереди), механизмами синхронизации, работой в реальном времени. Недостатки применения ОСРВ менее существенны и заключены, в основном, в увеличении требований к объему ОЗУ.

Рассмотрены существующие ОСРВ: FreeRTOS, KeilRTX, UC/OS, QNX. В результате анализа сделан выбор в пользу ОСРВ FreeRTOS, поскольку она обладает всем необходимым функционалом, портирована на большинство аппаратных платформ, распространяется по лицензии, разрешающей модификацию исходного кода и дальнейшее использование, в том числе и в коммерческих целях.

Проанализирована логика работы выбранной в качестве основы ОСРВ, сформированы необходимые модули и программные интерфейсы (API) для обеспечения работы как моноверсионного, так и мультиверсионного прикладного ПО.

В результате сформирована блочно-модульная структура типового программного комплекса системы управления (рисунок 1).

Определено, что для реализации требуемой функциональности среды исполнения мультиверсионного ПО необходимо разработать блок принятия решения (БПР), который должен обеспечить все версии необходимыми входными данными, в случае ограничения на время ответа версий, прервать выполнение версий, не успевших дать ответ в установленный срок, непосредственно принять решение о правильном выходе на основе коллекции выходов версий и т.д.

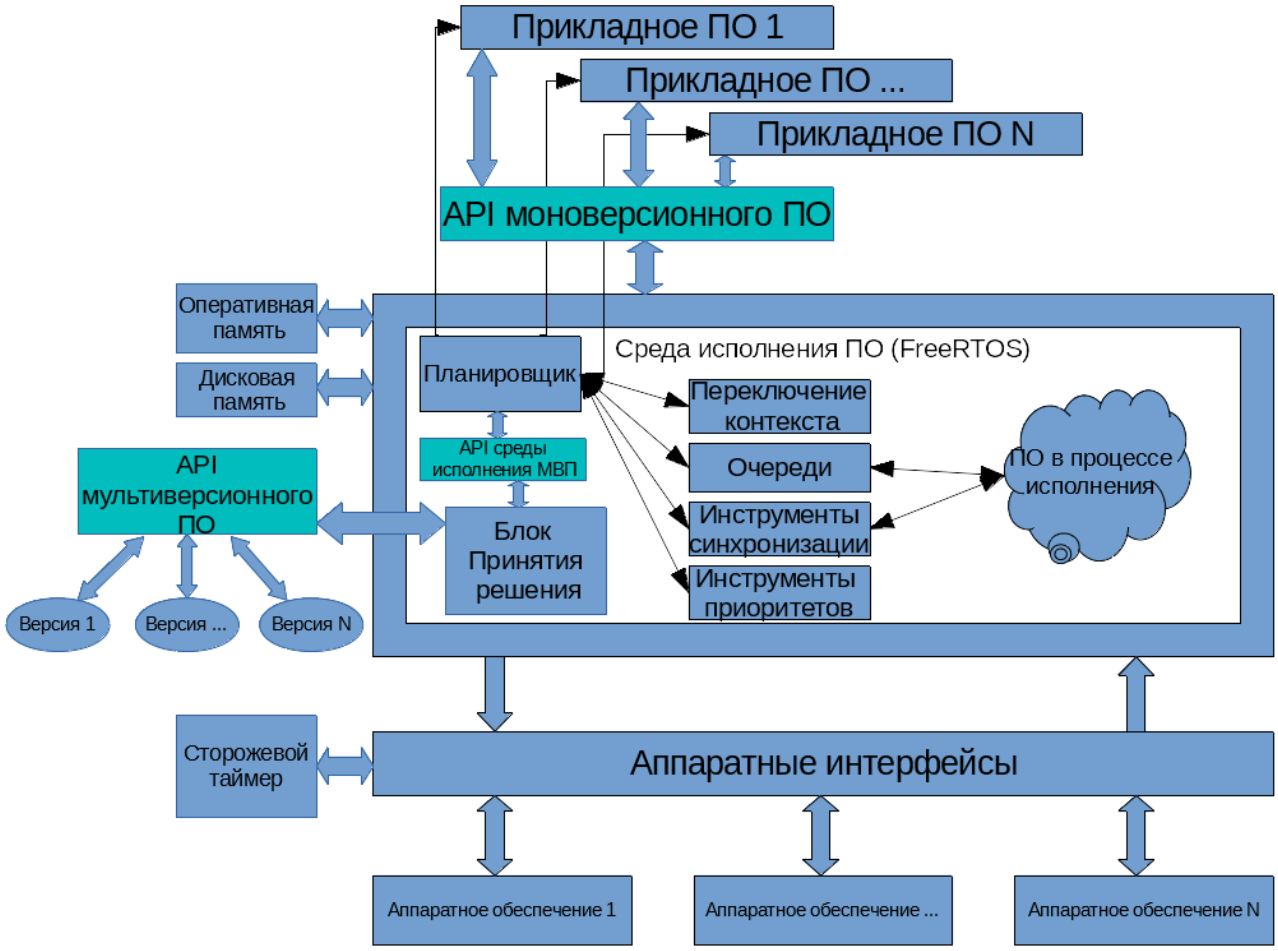


Рисунок 1 – Типовая структура системы управления и ее программного комплекса

Формализованы три программных интерфейса для обеспечения требуемой функциональности системы. Сформированы необходимые модификации планировщика для корректного исполнения мультиверсий.

В результате возникает задача выбора применяемых в блоке принятия решения моделей повышения отказоустойчивости ПО.

**В третьей главе** предложен комбинированный селективный алгоритм для решения данной задачи. Для оценки нижней границы надежности предложен модифицированный алгоритм расчета «деревьев сбоев», который ранее применялся для аппаратной надежности. Алгоритм расширен для возможности учитывать и аппаратные и программные сбои.

«Дерево сбоев» состоит из нежелательного события (сбоя системы или подсистемы), связанного с основными событиями через логические затворы. Нижнее событие разделяется на составляющие его причины, связанные логическими затворами «И» и «ИЛИ», которые далее разрешаются до тех пор, пока не будут

определены основные события. Основные события выражают базовые причины сбоя и представляют собой предел разрешения «дерева сбоев».

При анализе программного обеспечения возможными причинами могут быть комбинации ошибок программного обеспечения, входов и режимов работы. При анализе аппаратных систем основные события обычно связаны с отказами физических компонентов. В программной системе основные события представляют собой программные модули, которые могут привести к неправильному результату или принять неверный ввод.

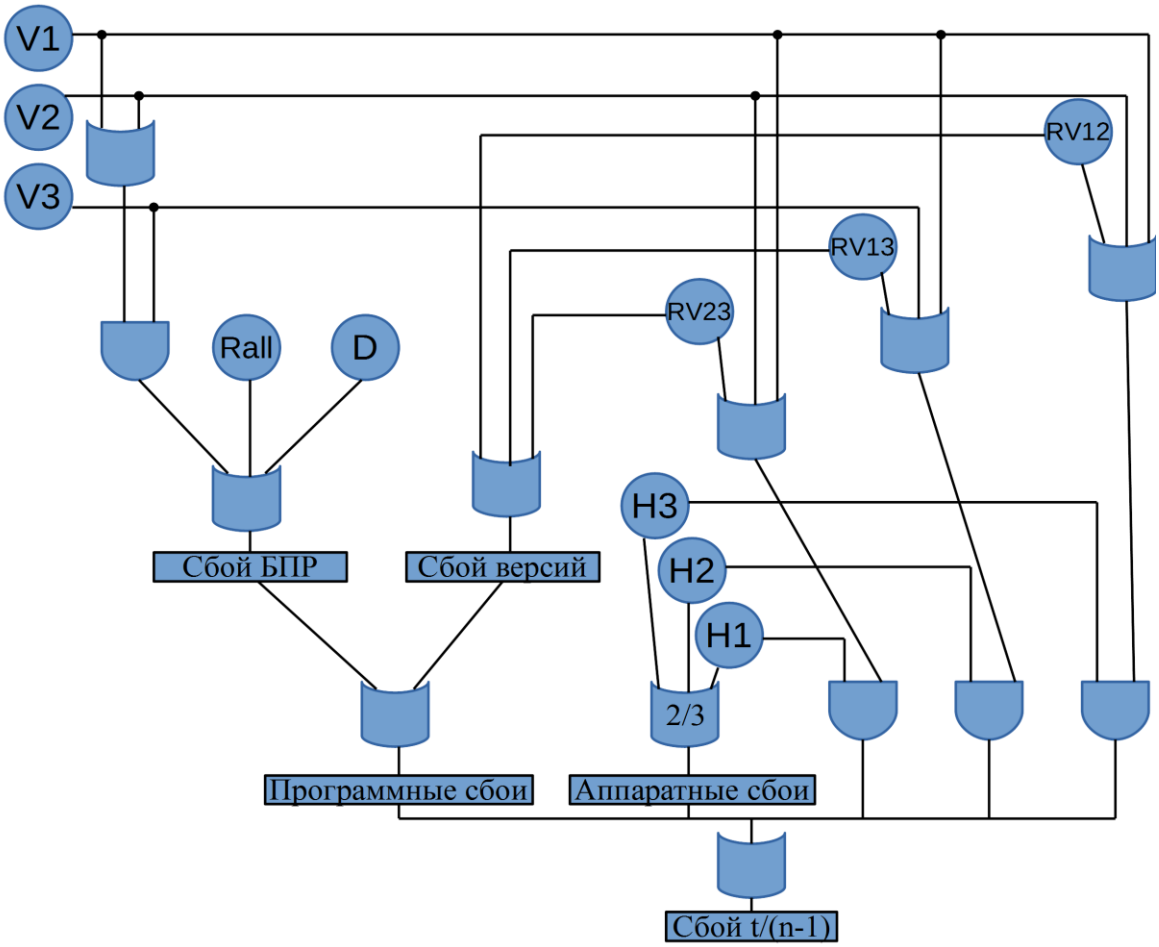


Рисунок 2 – «Дерево сбоев» для t/(n-1) алгоритма

Каждая из моделей «дерева сбоев» использует следующие обозначения для основных событий:

V# - несвязанная ошибка версии (где # - целое число от 1 до n (n - число версий));

D - независимая ошибка в решении (приемочный тест, мажоритарный избиратель, компаратор, арбитр);

RV## - связанная ошибка, которая возникает в двух разных версиях, в результате чего оба результата приводят к совпадающему ошибочному результату;

RALL - связанная ошибка, влияющая на все версии, а также на решение, вызванное несовершенными спецификациями;

H# - аппаратная ошибка, которая влияет на корректное исполнение ПО;

$P_i$ – вероятность события i, $Q_i = (1 - P_i)$.

Для примера рассмотрим «дерево сбоев» для t/(n-1) алгоритма при трех мультиверсиях.

Для расчета нам необходимо определить все возможные «сечения» - сочетания событий, которые приведут к активации нижнего события – сбоя системы.

Дерево сбоев t/(n-1) алгоритма имеет 20 минимальных сечений, из которых 5 имеют один элемент (RV12, RV13, RV23, RALL и D). Остальные сечения перечисляют 15 способов, с помощью которых программное обеспечение или аппаратные компоненты объединяются, чтобы привести к сочетанию, которое позволит пройти сбою. Модель «дерева сбоев» t/(n-1) алгоритма представляется суммой сечений:

$$T_s = P_{RV} + Q_{RV}P_{RV} + Q_{RV^2}P_{RV} + Q_{RV^3}P_D + Q_{RV^3}P_{RALL}Q_D + P_{V1}P_{V2}Q_{RV12}Q_DP_{V3} + P_{V1}Q_{V2}Q_DP_{V3} +$$
$$+ Q_{V1}P_{V2}Q_DP_{V3} + Q_VP_VQ_{RV^3}Q_{RALL}Q_DP_{H^2}Q_H + Q_{RV^3}Q_{RALL}Q_DP_{H^2}Q_H + Q_VP_VQ_{RV^3}Q_{RALL}Q_DP_{H^2}Q_H +$$
$$+ Q_{RV^3}Q_{RALL}Q_DP_{H^2}Q_H(1-P_V) + Q_VP_VQ_{RV^3}Q_{RALL}Q_DP_{H^2}Q_H + Q_{RV^3}Q_{RALL}Q_DP_{H^2}Q_H(1-P_V) +$$
$$+ P_VQ_VQ_{RV^3}Q_{RALL}Q_DP_HQ_H(1-P_H) + P_VQ_{RV^3}Q_{RALL}Q_DP_HQ_H(1-P_H) + P_VQ_VQ_{RV^3}Q_{RALL}Q_DQ_{H^2}P_H +$$
$$+ P_VQ_{RV^3}Q_{RALL}Q_DQ_{H^2}P_H + P_VQ_VQ_{RV^3}Q_{RALL}Q_DQ_{H^2}P_H + P_VQ_VQ_{RV^3}Q_{RALL}Q_DQ_{H^2}P_H.$$

Для остальных моделей сечения рассчитываются аналогично.

Полученная методика реализована программно и обеспечивает моделирование СУ с программной избыточностью, позволяя задавать все надежностные характеристики: вероятности сбоев каждой версии, вероятности межверсионных ошибок, вероятность ошибки блока принятия решения, вероятность аппаратных сбоев, количество версий для NVP, вероятность возникновения связанной ошибки, влияющей на все версии, а также на решение, вызванное несовершенными спецификациями, а также минимальную надежность системы, которую необходимо достигнуть при обратном расчете.

Алгоритм позволяет изучить поведение системы (изменение вероятности прохождения сбоя) при изменениях входных параметров, полученные результаты представлены в таблице 1.

Таблица 1 – Результаты прямого расчета деревьев сбоев при различной вероятности сбоев версий Pvx

| Вероятность сбоев версий Pvx | 0,001 | 0,010 | 0,050 | 0,100 |
|---|---|---|---|---|
| Вероятность сбоя для N-версионного программирования | 0,005 | 0,005 | 0,013 | 0,033 |
| Вероятность сбоя для t/(n-1) – алгоритма | 0,006 | 0,007 | 0,011 | 0,025 |
| Вероятность сбоя для самовосстанавливающихся блоков | 0,012 | 0,012 | 0,014 | 0,022 |
| Вероятность сбоя для N-версионного программирования с самопроверкой | 0,008 | 0,009 | 0,018 | 0,044 |

Проанализировав результаты, представленные в таблице 1, можно проследить влияние вероятности одиночных сбоев версий на надежность системы в целом. Также данные показывают, что надежность системы даже с учетом не абсолютно надежных компонентов (аппаратного обеспечения, БПР и т.д.) существенно выше, чем надежность каждой версии.

Рассмотрим пример работы обратного расчета «деревьев сбоев», когда мы фиксируем все характеристики системы, кроме искомых, в нашем случае это

вероятности одиночного сбоя версий, для простоты расчеты примем их равными, поскольку в противном случае мы получим не конкретный ответ, а бесконечное количество сочетаний вероятностей.

Таблица 2 – Результаты обратного расчета «деревьев сбоев» для различной заданной надежности системы

| Надежности системы | 0,900 | 0,950 | 0,980 |
|---|---|---|---|
| N-версионное программирование | 0,819 | 0,882 | 0,938 |
| t/(n-1) - алгоритм | 0,769 | 0,855 | 0,928 |
| Самовосстанавливающиеся блоки | 0,702 | 0,805 | 0,911 |
| N-версионное программирование с самопроверкой | 0,846 | 0,903 | 0,954 |

Из результатов, представленных в таблице 2, можно сделать вывод о том, что рассмотренные модели позволяют получить из ненадежных программных модулей систему, функционирующую с более высоким уровнем надежности, чем любая из версий.

Результаты работы данного инструмента позволяют оптимально скомпоновать разрабатываемую систему, так как на значение вероятности прохождения сбоя влияет не только характеристика каждого элемента системы, но и ее структура, поскольку от структуры зависят «срезы» - сочетания некорректной работы компонентов, достаточные для прохождения сбоя.

Для расчета верхней границы надежности предложена методика расчета параметра **корректности**.

Корректность - это сумма вероятности всех сочетаний возможных выходов мультиверсий, успешной работы аппаратного обеспечения и вариантов работы блока принятия решения, которые дадут корректный выход.

Общую формулу корректности можно выразить следующим образом: $S_n = D_X^+ \cdot H_{D_x}^+ \sum_{i=m}^{n} P_i^+ \cdot Q_1^+$, где $P_i{}^+$ - вероятность сочетания корректных выходов версий, достаточного для правильной работы блока принятия решений; $D_X^+$ - вероятность корректного выхода блока принятия решения при X входах; $H_{D_x}^+$ - вероятность корректной работы аппаратной платформы, на которой исполняется программа блока принятия решения.

Для исследования зависимости величины корректности от основных вероятностных параметров надежности и применяемой схемой повышения отказоустойчивости реализуем программную среду, рассчитывающую корректность. Для этого выразим расчетные формулы для каждой из предложенных схем. Для N-версионного программирования при N=3:

$$S_{NVP(3)} = D_{NVP(3)}^+ \cdot H_{D_3}^+ \cdot P_{V1} \cdot P_{V2} \cdot P_{V3} + D_{NVP(3)}^+ \cdot H_{D_3}^+ \cdot P_{V1} \cdot P_{V2} \cdot (1 - P_{V3}) +$$

$$+ D_{NVP(3)}^+ \cdot H_{D_3}^+ \cdot P_{V1} \cdot (1 - P_{V2}) \cdot P_{V3} + D_{NVP(3)}^+ \cdot H_{D_3}^+ \cdot (1 - P_{V1}) \cdot P_{V2} \cdot P_{V3}.$$

Для N-версионного программирования при N=4:

$$S_{NVP(4)} = D^{+}_{NVP(4)} \cdot H^{+}_{D_4} \cdot P_{V1} \cdot P_{V2} \cdot P_{V3} \cdot P_{V4} + D^{+}_{NVP(4)} \cdot H^{+}_{D_4} \cdot P_{V1} \cdot P_{V2} \cdot P_{V3} \cdot (1-P_{V4}) +$$
$$+ D^{+}_{NVP(4)} \cdot H^{+}_{D_4} \cdot P_{V1} \cdot P_{V2} \cdot (1-P_{V3}) \cdot P_{V4} + D^{+}_{NVP(4)} \cdot H^{+}_{D_4} \cdot P_{V1} \cdot (1-P_{V2}) \cdot P_{V3} \cdot P_{V4} +$$
$$+ D^{+}_{NVP(4)} \cdot H^{+}_{D_4} \cdot (1-P_{V1}) \cdot P_{V2} \cdot P_{V3} \cdot P_{V4}.$$

Для N-версионного программирования при N=5:

$$S_{NVP(5)} = D^{+}_{NVP(5)} \cdot H^{+}_{D5} \cdot P_{V1} \cdot P_{V2} \cdot P_{V3} \cdot P_{V4} \cdot P_{V5} + D^{+}_{NVP(5)} \cdot H^{+}_{D5} \cdot P_{V1} \cdot P_{V2} \cdot P_{V3} \cdot P_{V4} \cdot (1-P_{V5}) +$$
$$+ D^{+}_{NVP(5)} \cdot H^{+}_{D5} \cdot P_{V1} \cdot P_{V2} \cdot P_{V3} \cdot (1-P_{V4}) \cdot P_{V5} + D^{+}_{NVP(5)} \cdot H^{+}_{D_5} \cdot P_{V1} \cdot P_{V2} \cdot (1-P_{V3}) \cdot P_{V4} \cdot P_{V5} +$$
$$+ D^{+}_{NVP(5)} \cdot H^{+}_{D_5} \cdot P_{V1} \cdot (1-P_{V2}) \cdot P_{V3} \cdot P_{V4} \cdot P_{V5} + D^{+}_{NVP(5)} \cdot H^{+}_{D_5} \cdot (1-P_{V1}) \cdot P_{V2} \cdot P_{V3} \cdot P_{V4} \cdot P_{V5} +$$
$$+ D^{+}_{NVP(5)} \cdot H^{+}_{D5} \cdot P_{V1} \cdot P_{V2} \cdot P_{V3} \cdot (1-P_{V4}) \cdot (1-P_{V5}) + D^{+}_{NVP(5)} \cdot H^{+}_{D5} \cdot P_{V1} \cdot P_{V2} \cdot (1-P_{V3}) \cdot P_{V4} \cdot (1-P_{V5}) +$$
$$+ D^{+}_{NVP(5)} \cdot H^{+}_{D5} \cdot P_{V1} \cdot (1-P_{V2}) \cdot P_{V3} \cdot P_{V4} \cdot (1-P_{V5}) + D^{+}_{NVP(5)} \cdot H^{+}_{D5} \cdot (1-P_{V1}) \cdot P_{V2} \cdot P_{V3} \cdot P_{V4} \cdot (1-P_{V5}) +$$
$$+ D^{+}_{NVP(5)} \cdot H^{+}_{D5} \cdot P_{V1} \cdot P_{V2} \cdot (1-P_{V3}) \cdot (1-P_{V4}) \cdot P_{V5} + D^{+}_{NVP(5)} \cdot H^{+}_{D5} \cdot P_{V1} \cdot (1-P_{V2}) \cdot P_{V3} \cdot (1-P_{V4}) \cdot P_{V5} +$$
$$+ D^{+}_{NVP(5)} \cdot H^{+}_{D5} \cdot (1-P_{V1}) \cdot P_{V2} \cdot P_{V3} \cdot (1-P_{V4}) \cdot P_{V5} + D^{+}_{NVP(5)} \cdot H^{+}_{D_5} \cdot P_{V1} \cdot (1-P_{V2}) \cdot (1-P_{V3}) \cdot P_{V4} \cdot P_{V5} +$$
$$+ D^{+}_{NVP(5)} \cdot H^{+}_{D_5} \cdot (1-P_{V1}) \cdot P_{V2} \cdot (1-P_{V3}) \cdot P_{V4} \cdot P_{V5} + D^{+}_{NVP(5)} \cdot H^{+}_{D_5} \cdot (1-P_{V1}) \cdot (1-P_{V2}) \cdot P_{V3} \cdot P_{V4} \cdot P_{V5}.$$

Представленный расчет сечений аналитически подтверждает, что при увеличении числа версий корректность будет повышаться, поскольку увеличивается число сочетаний, которые обеспечивают корректный ответ системы, так как для успешного голосования абсолютным большинством достаточно $\lfloor N/2 \rfloor + 1$ корректно сработавших версий, где для корректного голосования подходят случаи как 5 корректно сработавших версий, 4 из 5, и даже любые 3 из 5. Это подтверждает достоверность модели.

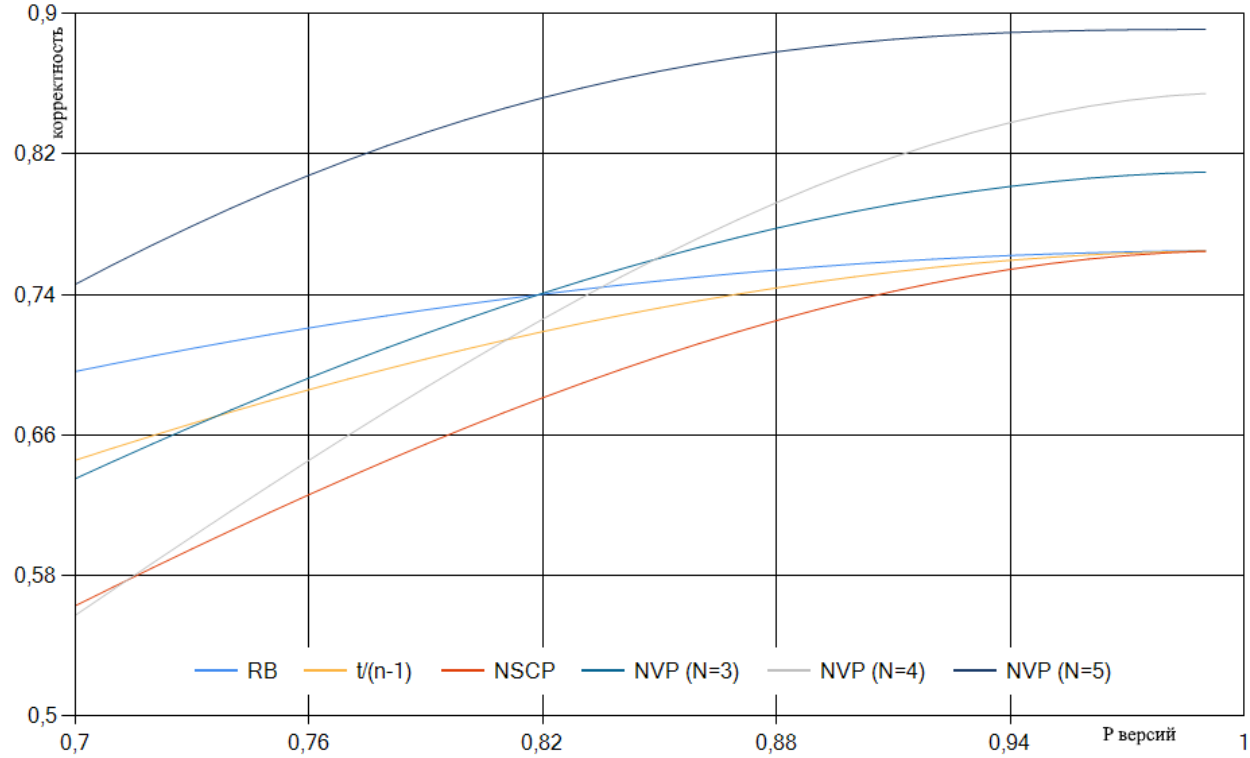


Рисунок 3 –графики корректности при надежности версий от 0.7 до 0.99.

Для практической проверки предложенной методики и анализа результатов моделирования была реализована программная среда расчета величины корректности для различных мультиверсионных моделей, реализующая прямой и обратный расчет и построение графиков.

Пересечение графиков корректности на рисунке 3 показывает, что при различных характеристиках модулей меняется оптимальная модель, это означает, что оптимальный выбор будет зависеть от множества характеристик моделируемой системы.

Предложенный селективный алгоритм, позволяет изучить влияние входных параметров надежности на ее корректность и вероятность прохождения сбоя, что является важным инструментом для разработчика отказоусточивых программных систем, поскольку позволяет осознанно выбрать методологию и определить требования к элементам разрабатываемой системы еще на этапе проектирования, что позволит упростить разработку системы с требуемым уровнем надежности и снизить затраты на ее реализацию.

В большинстве случаев оптимальной оказывается модель мультиверсионного программирования, и возникает задача выбора алгоритма принятия правильного решения в ней.

**В четвертой главе** исследованы базовые алгоритмы голосования, сделан вывод о недостаточной устойчивости существующих алгоритмов к межверсионным ошибкам. Предложены модификации базовых алгоритмов голосования согласованным большинством и нечеткого голосования согласованным большинством, необходимые для повышения надежности блока принятия решения. Модификации заключаются во введении динамической оценки надежности каждой версии или ее «веса», имеющей элемент забывания.

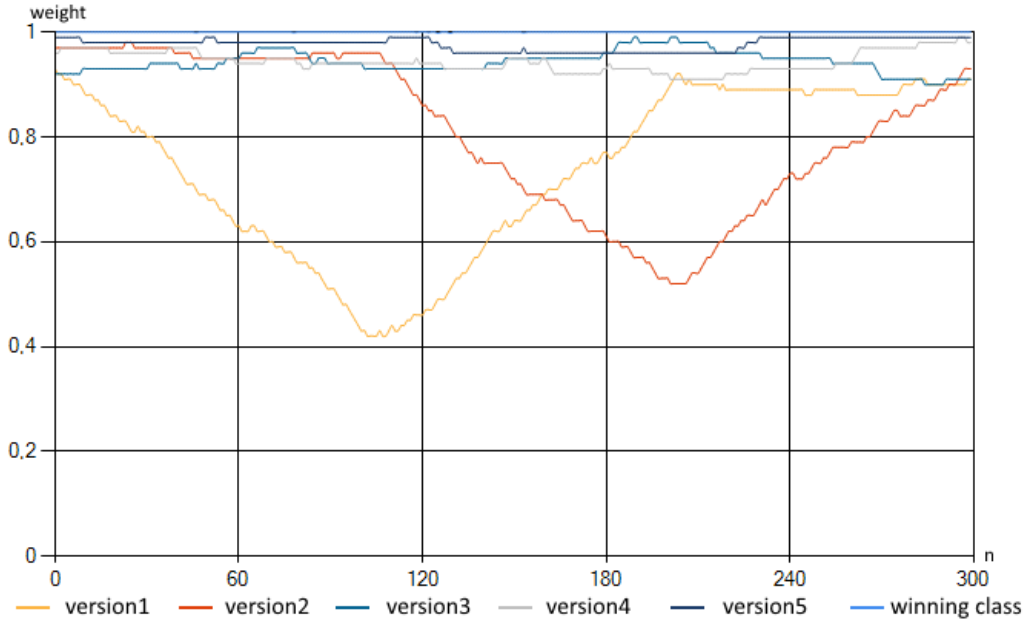


Рисунок 4 - Результаты работы имитационной среды (проявлен сбой в первой версии первого набора данных и во второй версии второго набора данных с вероятностью 0,5).

Модифицированные алгоритмы были реализованы в программном инструменте – имитационной среде, с целью их проверки и сравнения эффективности.

По результатам моделирования формируется суммарное количество ошибок за все итерации для каждого алгоритма, количество допущенных «неточностей» для нечетких алгоритмов и сумма возникших межверсионных ошибок. Если по результатам моделирования существует однозначно превосходящий другие алгоритм, система выводит сообщение о выбранном оптимальном алгоритме.

Особый интерес представляет нетривиальная модель t/(n-1)-версионного программирования, предложенная Цзе Сюй (Jie Xu) основанная на t/(n-1) диагностируемости. Суть алгоритма не в голосовании всех выходов версий, а лишь в сравнении некоторых из них, достаточных для принятия решения. Рассмотрим на примере системы с количеством версий n=5 и максимальным количеством ошибок t=2, то есть рассмотрим 2/(5-1) вариант t/(n-1)-версионного программирования. Сравниваются попарно выходы четырех версий – 1 со 2, 2 с 3, 3 с 4, получаем три результата сравнений $\omega_{12}$, $\omega_{23}$, $\omega_{34}$, равные 0, если выходы совпадают и 1, если различаются. На основании только этих трех результатов сравнения алгоритм принимает решение о переключении выхода между выходами 1, 4 и 5 версий, то есть версии 2 и 3 используются только для сравнения, значения их выходов никогда не используется в качестве выхода системы.

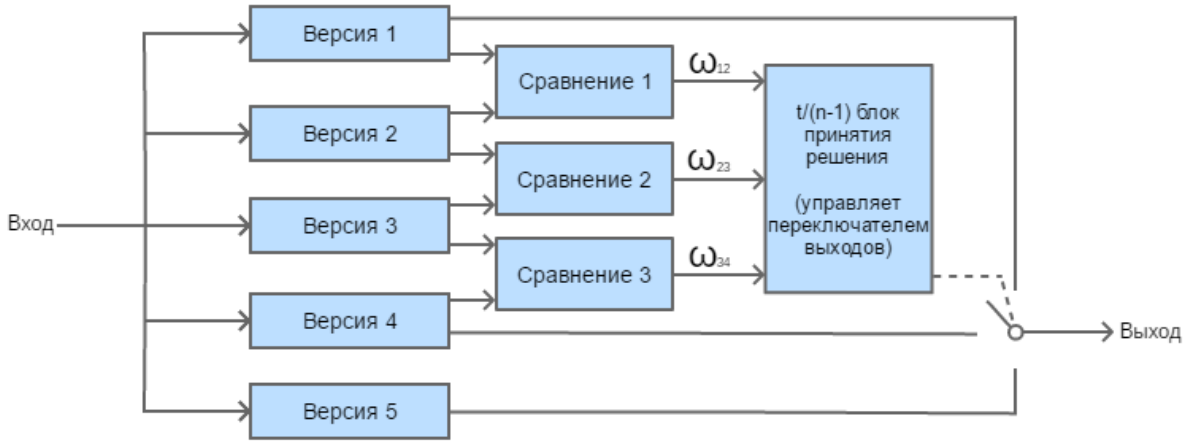


Рисунок 5 - Архитектура t/(n-1) алгоритма при n=5 и t=2

В работе впервые предложена нечеткая модификация t/(n-1) алгоритма, она заключается во введении нечеткой логики в компараторы. Поскольку компараторы имеют булевый выход, то они возвращают 0, если значения отличаются друг от друга на величину, не более заданного допуска, и 1, иначе. Разница значений, если она не превосходит допуск, не учитывается.

Рассмотрим результаты исполнения программы с различными входными параметрами. Сравним показатели надежности t/(n-1) алгоритма по сравнению с взвешенными модификациями алгоритма голосования согласованным большинством (его четкой и не четкой версии) и медианным голосованием.

Таблица 3 – Результаты моделирования при различной надежности версий

| Заданная надежность версий | | 0,65 | 0,70 | 0,75 | 0,80 | 0,85 | 0,90 | 0,95 |
|---|---|---|---|---|---|---|---|---|
| Четкое голосование согласованным большинством | Кол-во ошибок | 22 | 12 | 9 | 1 | 1 | 0 | 0 |
| Нечеткое голосование согласованным большинством | Кол-во ошибок | 24 | 14 | 7 | 3 | 1 | 0 | 0 |
| | Из них сбоев | 14 | 10 | 3 | 2 | 1 | 0 | 0 |
| | неточностей | 10 | 4 | 4 | 1 | 0 | 0 | 0 |
| t/(n-1) алгоритм | Кол-во ошибок | 39 | 24 | 17 | 10 | 3 | 1 | 0 |
| Нечеткая модификация t/(n-1) алгоритма | Кол-во ошибок | 44 | 35 | 34 | 21 | 16 | 4 | 3 |
| | Из них сбоев | 19 | 14 | 14 | 2 | 2 | 0 | 0 |
| | неточностей | 25 | 21 | 20 | 19 | 14 | 4 | 3 |
| Медианное голосование | Кол-во ошибок | 73 | 45 | 29 | 16 | 8 | 2 | 0 |
| Межверсионных ошибок | | 134 | 94 | 66 | 41 | 26 | 10 | 1 |

Результаты моделирования показали, что применение t/(n-1) алгоритма, который представляет интерес, как альтернатива алгоритмам голосования, оправдано лишь в системах с существенными ограничениями по вычислительным ресурсам и применением достаточно надежных версий, поскольку алгоритм показывает худшую устойчивость к ненадежным версиям и различным типам ошибок.

**В пятой главе**

Для практической проверки разработана имитационная среда, реализующая все рассмотренные мультиверсионные модели и модифицированные алгоритмы голосования в них. Результаты моделирования показывают адекватность подхода и позволяют сравнить основные мультиверсионные модели по количеству допущенных ошибок в одинаковых условиях, а также позволяют сравнить различные алгоритмы принятия решения в мультиверсионных системах.

Реализована мультиверсионная среда исполнения реального времени для проверки работоспособности предложенной во второй главе типовой структуры. Среда реализована в ОСРВ FreeRTOS v10, в ней реализован блок принятия решения, выделенные очереди, функции, реализующие предложенные модели и алгоритмы, которые запускаются в виде отдельных потоков.

Система использовалась для решения прикладной задачи – повышения эффективности распознавания голосовых команд для передачи управляющих воздействий на подсистемы автономных беспилотных объектов. Объектами применения разработанного программно-аппаратного комплекса на базе FreeRTOS служат такие автономные беспилотные объекты, как моторизованные и мультироторные системы, а также вспомогательные устройства для работ в открытом космосе. Использовались выходы 3 готовых алгоритмов распознавания, на основе которых система выбирала правильный ответ. Результаты работы программной

системы показывают эффективность мультиверсионного подхода, поскольку качество работы системы превышает качество любого из используемых алгоритмов (процент корректно распознанных команд для первого алгоритма составил 87, для второго алгоритма - 81, для третьего алгоритма - 86, а система корректно распознала 96% команд). Исполнение происходило в режиме реального времени, результаты подтверждают выполнение требований по времени реакции системы на событие. Каждая итерация мультиверсионного голосования уложилась в 1 мс.

В рамках реализации прикладных задач, на этапе тестирования возникает ряд проблем, связанных с невозможностью гарантировано выявить 100% ошибок в сложном программной комплексе.

На основе модели Джелинского – Моранды и полученном экспоненциальном распределении времени обнаружения ошибок в тестируемом ПО предложена деэвтрофикационная модель предсказания времени наработки на отказ (MTTF). Модель реализована в виде программного инструмента, проведено исследование экспериментальных данных о результатах тестирования программного обеспечения (397 точек):

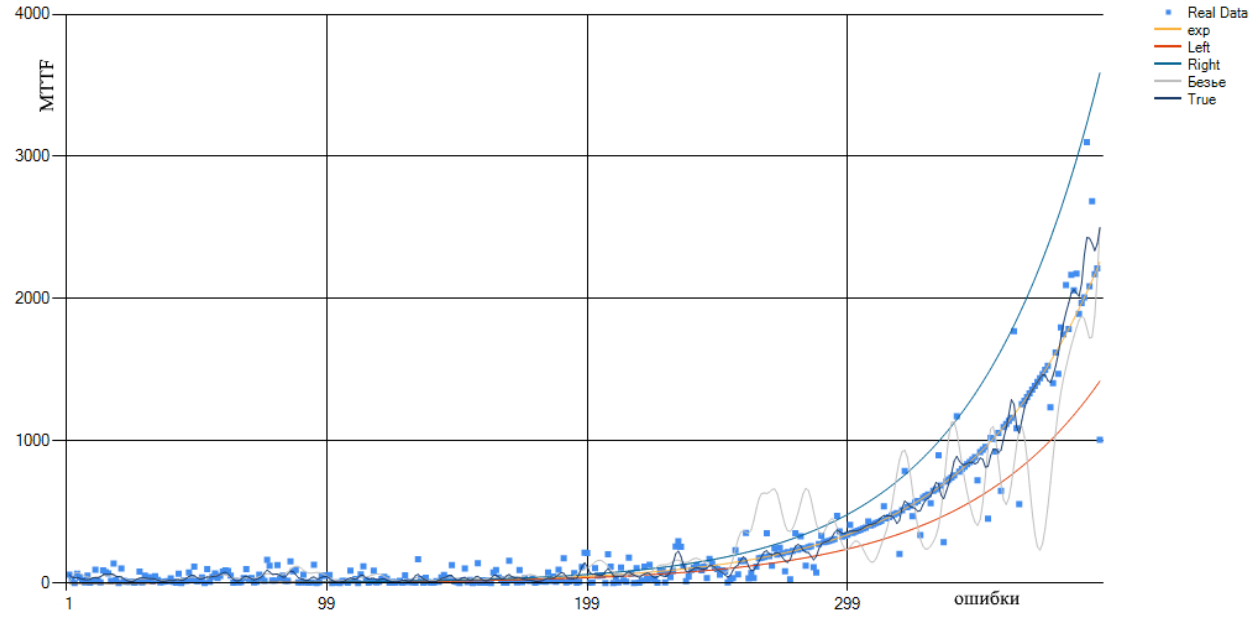


Рисунок 6 - Деэвтрофикационная модель предсказания MTTF.

Данная модель и реализованный инструмент позволяют предсказать среднее время наработки на отказ, его разброс и вероятность попадания в предсказанный диапазон. Также данный инструмент позволяет существенно ускорить тестирование, поскольку дает возможность тестировать систему не в масштабе реального времени, а настолько быстрее, насколько позволит производительность тестовой платформы, что позволит гарантировать время наработки на отказ в месяцах, затрачивая на тестирования на порядки меньшие промежутки времени.

В **заключении** сформулированы основные выводы и результаты, полученные в диссертационной работе.

# ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ

1. Сформирована типовая структура системы управления, в том числе ее программной составляющей, которая позволяет: разработать имитационную среду моделирования, в которой можно протестировать отдельные программные модули, получить их характеристики и характеристики системы при их применении; реализовать мультиверсионную среду исполнения реального времени на основе ОСРВ, применимую на большинстве современных одноплатных компьютеров и платах разработчиков.

2. Предложен комбинированный селективный алгоритм для выбора оптимальной модели повышения отказоустойчивости, основанной на программной избыточности, использующей следующие модели:

2.1 Модифицирована и реализована модель деревьев сбоев для возможности расчета характеристик системы с учетом и аппаратных и программных модулей. Полученный инструмент является полезным для разработчика на практике, поскольку позволяет получить обоснованную компоновку системы, а также изучить влияние структуры системы и характеристики ее модулей на общие характеристики надежности системы.

2.2 Расчетный параметр корректности впервые предложен и представляет собой сумму всех срезов, по аналогии с деревьями сбоев, однако учитывающий, в отличии от них, все возможные сочетания работы узлов, которые приведут к корректному выходу. Реализован программный инструмент, автоматизирующий расчет параметра, как прямой, позволяющий получить характеристики системы, так и обратный, позволяющий определить минимальные требования к надежности модулей.

3. Предложены новые взвешенные модификации с элементом забывания алгоритмов голосования, которые повышают устойчивость к межверсионным ошибкам и повышающие надежность в целом. Реализована программная среда, реализующая предложенные алгоритмы, которая позволила доказать их эффективность по сравнению с базовыми алгоритмами (на 0,08 сократилась вероятность прохождения ошибки).

4. Разработана имитационная среда моделирования, реализующая все распространенные модели повышения отказоустойчивости и нетривиальный алгоритм t/(n-1)-вариантного программирования, которая позволяет сравнить все модели в одинаковых условиях – на одинаковых выходах симуляций версий, работающих в соответствии с заданными параметрами надежности.

5. Реализована мультиверсионная среда исполнения реального времени на основе ОСРВ FreeRTOS v10, для проверки предложенной технологии. Результаты работы практически реализованной системы, решающей реальную задачу, показывают применимость предложенной технологии и эффективность моделей и алгоритмов.

6. Впервые предложена и реализована деэвтрофикационная модель предсказания времени наработки на отказ. Модель позволяет предсказать параметры надежности программного модуля по результатам его испытаний и существенно (на порядки, в зависимости от производительности тестовой платформы) ускоряет этап тестирования.

Таким образом, цель данного исследования достигнута путем создания моделей, алгоритмов и программных инструментов, применяемых на различных этапах жизненного цикла создания отказоустойчивых программных комплексов встраиваемых систем управления реального времени.

**Публикации в изданиях, рекомендованных ВАК РФ:**

1. Ковалев И.В., Лосев В.В., Сарамуд М.В., Ковалев Д.И., Петросян М.О. К вопросу реализации мультиверсионной среды исполнения бортового программного обеспечения автономных беспилотных объектов средствами операционной системы реального времени // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. 2017. Т. 18. № 1. С. 58-61
2. Kovalev I.V., Losev V.V., Saramud M.V., Kovalev D.I., Petrosyan M.O., Brezitskaya V.V. To the question on implementation of multi-version execution environment software of onboard autonomous unmanned objects by means of real-time operating system // Сибирский журнал науки и технологий. 2017. Т. 18. № 4. С. 744-747.
3. Ковалев Д.И., Сарамуд М.В., Карасева М.В., Нургалеева Ю.А. Развитие методов эквивалентного преобразования ГЕРТ-сетей для анализа мультиверсионного программного обеспечения // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. 2014. № 1 (53). С. 11-16.
4. Ковалев И.В., Зеленков П.В., Сарамуд М.В., Сидорова Г.А., Брезицкая В.В. Модели ГЕРТ-сетей для различных способов применения методологии мультиверсий // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. 2013. № 1 (47). С. 41-47.
5. Калинин А.О., Посконин М.В., Сарамуд М.В., Лосев В.В., Ковалев И.В. Методика расчета временных характеристик элементов автоматизированной системы управления на примере замкнутого контура регулирования давления на участке трубопровода под управлением контроллера "ОВЕН ПЛК100 220" // Сибирский журнал науки и технологий. 2017. Т. 18. № 2. С. 387-395.

**Публикации в изданиях, индексируемых в международных базах**

6. Saramud M.V., Zelenkov P.V., Kovalev I.V., Kovalev D.I., Kartsan I.N. Development of methods for equivalent transformation of gert networks for application in multi-version software // IOP Conference Series: Materials Science and Engineering. 12. Series: "XII International Scientific and Research Conference "Topical Issues in Aeronautics and Astronautics"" 2016. С. 012015. **(Web of Science, Scopus)**.
7. Kovalev I.V., Zelenkov P.V., Losev V.V., Kovalev D.I., Ivleva N.V., Saramud M.V. Multiversion environment creation for control algorithm execution by autonomous unmanned objects // IOP Conference Series: Materials Science and Engineering. 5. Series: "V International Workshop on Mathematical Models and their Applications 2016" 2017. С. 012025. **(Web of Science, Scopus)**.
8. Kovalev I., Losev V., Saramud M., Petrosyan M. Model implementation of the simulation environment of voting algorithms, as a dynamic system for increasing the reliability of the control complex of autonomous unmanned objects // (2017) MATEC Web of Conferences, 132, статья № 04011 **(Scopus)**.

9. Kovalev I., Voroshilova A., Losev V., Saramud M., Chuvashova M., Medvedev A. Comparative Tests of Decision Making Algorithms for a Multiversion Execution Environment of the Fault Tolerance Software // European Conference on Electrical Engineering and Computer Science (EECS 2017) **(Scopus)**.

10. Saramud M.V., Kovalev I.V., Losev V.V., Kuznetsov P.A. Software interfaces and decision block for the execution environment of multi-version software in real-time operating systems // International Journal on Information Technologies & Security, No 1 (vol. 10), 2018, pp. 25-34. **(Web of Science)**

11. Kovalev I., Losev V., Saramud M., Kuznetsov P., Petrosyan M. To the Question of the Organization of a Learning Environment for Developers of Cross-Platform OnBoard Software for Unmanned Aerial Vehicles // TOJET: The Turkish Online Journal of Educational Technology – December 2017, Special Issue for INTE 2017, pp. 700-705 **(Scopus)**.

**Публикации в других изданиях:**

12. Сарамуд М.В., Зеленков П.В., Ковалев И.В., Ковалев Д.И., Брезицкая В.В., Характеристика надежности программных модулей // Решетневские чтения. 2015. Т. 2. № 19. С. 87-88.

13. Зеленков П.В., Ковалев Д.И., Соловьев Е.В., Сарамуд М.В., Прохорович Г.А., Программная реализация модифицированного алгоритма муравьиной колонии для n-версионного программного обеспечения // Решетневские чтения. 2014. Т. 2. № 18. С. 43-45.

14. Сарамуд М.В., Зеленков П.В., Брезицкая В.В., Ковалев И.В., Ковалев Д.И., К вопросу надежности программного обеспечения отказоустойчивых систем // Материалы Международной научно-практическая конференции «Современное состояние науки и техники», 2016 г. С. 149-151.

15. Сарамуд М.В., Ковалев И.В., Лосев В.В., Посконин М.В., Калинин А.О., К вопросу классификации автономных беспилотных объектов // Актуальные проблемы авиации и космонавтики – 2017. Том 2, С. 64-66.

**Зарегистрированные программные системы**

16. Сарамуд М.В., Ковалев И.В., Лосев В.В., Брезицкая В.В., Петросян М.О. Имитационная среда исполнения мультиверсионного программного обеспечения с алгоритмами взвешенного голосования с забыванием и t/(n-1) алгоритмом принятия решения. Свидетельство о государственной регистрации программы для ЭВМ №2018611721 от 06.02.2018

17. Сарамуд М.В., Ковалев И.В., Лосев В.В., Бурдина Е.В., Петросян М.О., Кузнецов П.А. Обучающая среда для разработчиков кроссплатформенного бортового программного обеспечения беспилотных летательных объектов. Свидетельство о государственной регистрации программы для ЭВМ №2018611722 от 06.02.2018

18. Сарамуд М.В., Ковалев И.В., Лосев В.В., Брезицкая В.В., Петросян М.О., Кузнецов П.А. Имитационная среда исполнения различных отказоустойчивых схем программного обеспечения, основанных на программной избыточности, с имитацией работы сбойных версий. Свидетельство о государственной регистрации программы для ЭВМ №2018613315 от 07.03.2018

19. Середин А.И., Белорусов А.И., Кабайкин К.С., Сарамуд М.В., Штарик А.В. Система вычисления оптимального состава мультиверсионного программного

обеспечения «Oprocom». Свидетельство о государственной регистрации программы для ЭВМ №2010614929 от 15.05.2010

Сарамуд Михаил Владимирович

Модельно-алгоритмическое обеспечение анализа отказоустойчивости программных комплексов встраиваемых систем управления реального времени

Автореферат
Подписано к печати 06.06.2018. Формат 60x84/16
Уч. изд. л. 1.0 Тираж 100 экз. Заказ № _____
Отпечатано в отделе копировальной и множительной техники
СибГУ им. М.Ф. Решетнева.
660037, г. Красноярск, пр. им. газ. «Красноярский рабочий», 31