ВВЕДЕНИЕ

Программное обеспечение информационных и управляющих систем представляет собой чрезвычайно сложное изделие. В его создании принимают участие специалисты многих коллективов. Процесс создания программного обеспечения невозможен без чёткой координации и тщательного планирования действий различных специалистов, особенно в связи с выделением необходимых ресурсов для обеспечения его высоких качества и надёжности. Успешное и экономичное решение этой проблемы позволяет избежать нежелательных последствий в процессе эксплуатации сложных систем различного назначения. Опыт создания и применения сложных информационных систем в последние десятилетия выявил множество ситуаций, когда ошибки их функционирования были обусловлены дефектами комплексов программ и приводили к большим ущербам. Из-за ошибок в программах автоматического управления погибло несколько зарубежных и отечественных спутников, происходили срывы и катастрофы в сложных административных, банковских и технологических информационных системах. В ряде работ было показано на примере разработки программ противоракетной обороны США (СОИ), что при существующем уровне развития методов, технологии и средств автоматизации проектирования имеются предельные масштабы объёмов комплексов программ, при которых невозможно обеспечить при допустимых затратах необходимую надёжность их функционирования и безопасность применения [77]. Если в исследованиях по обеспечению надёжности технических систем, начиная с 50-х годов, к 80-м годам прошлого столетия были достигнуты значительные успехи, то работы по исследованию и обеспечению надёжности программных средств только начинали зарождаться. Первые обобщающие работы [15, 31, 35, 64], в которых были сформулированы концепция и основные положения теории надёжности программных

средств для информационных систем, появились в 80-х годах прошлого столетия. Именно в это время были заложены основы методологии и технологии создания высоконадёжных комплексов программ сложных объектов. Из них следовало, что для обеспечения высокой надёжности функционирования и безопасности применения создаваемых сложных комплексов программ, необходима чёткая организация и высокая квалификация многих коллективов специалистов, участвующих в разработке, проектировании и эксплуатации этих комплексов.

Подготовка системных специалистов начинается в ВУЗе. Молодые творцы новых систем, обслуживающий персонал, связанный в будущем с эксплуатацией перспективных образцов различного назначения, должны получать необходимые базовые знания в процессе своего обучения. В настоящее время теория надёжности технических систем изучается практически в каждом учебном заведении инженерного профиля. Надёжности программных систем в них пока уделяется неоправданно мало внимания. Несомненно, в ближайшем времени этот пробел в образовании должен быть устранён. Только при этом условии можно готовить настоящего системного инженера, в котором нуждается практика.

Данная книга нужна для всех, кому необходима теория надёжности программных систем как наука и научная дисциплина. Учёный в ней найдёт проблемы и задачи, решение которых потребует серьёзных и тщательных научных исследований. Руководитель и аспирант найдут тематику диссертационных работ. Инженер, создающий высоконадёжное программное обеспечение, ознакомится с моделями, методами анализа и обеспечения требуемой надёжности. Инженер по эксплуатации систем программного обеспечения найдёт в ней рекомендации, советы по научному обоснованию высокой технологичности программных изделий. Студент и преподаватель получит вспомогательное пособие по изучению теории надёжности программных систем. Она будет полезна и лицам, занимающимся вопросами обеспечения качества, эффективности, безопасности и риска, готовности и живучести сложных информационных систем. Наконец, хочется надеяться, что книга будет способствовать созданию сложных высоконадёжных объектов.

Современное состояние и перспективы в области надёжности программных средств

В настоящее время большое внимание научно-технической общественности привлекает проблема надёжности программных средств. Надёжность, готовность и эксплуатационная технологичность данных средств являются пока не полностью достижимыми целями.

Создание сложного программного обеспечения ныне не базируется на прочном научном фундаменте, а больше является искусством. Поэтому программный продукт не защищён от ошибок человека-создателя, часто не является безопасным в сложных системах из-за риска и неопределённости, которые должны рассматриваться при принятии решения об его практическом использовании.

Причинами этого для сложных новых систем являются: недостаток профессионального опыта разработки систем, сложность их модификации в процессе работы из-за временных ограничений, невозможность проведения испытаний в условиях эксплуатации. Сложность проблемы связана с неопределённостью и быстрым изменением требований, из-за чего можно ожидать возникновение ошибок при проектировании и сопровождении программного обеспечения.

Чрезмерная сложность систем, большой диапазон ограничений при разработке и эксплуатации, частая модификация требований увеличивают неопределённость анализа систем в течение их жизненного цикла. Поэтому научно-технический, проектирующий и эксплуатирующий персонал обычно могут выдвигать лишь аргументы за или против системы вместо того, чтобы произвести систематизированный, а тем более системный, анализ выполнения требований по надёжности в рамках располагаемых ресурсов, стратегии применения и организационных аспектов.

Потенциальное влияние ненадёжных программных средств на выполнение стратегических или тактических операций становится существенным.

Руководители проектов и конструкторы не могут более игнорировать важность проведения анализа надёжности для обоснования решения о целесообразности создания той или иной системы. Именно более точная оценка требований к программным

средствам поможет выявить специфических недостатки в надёжности системы.

Надёжность зависит от процедур и методов, используемых в течение всего жизненного цикла, хотя большинство разработчиков систем сосредотачивают внимание лишь на этапах испытаний и обслуживания. Важность понимания эксплуатационной технологичности связана с использованием систем автоматической обработки данных. Деятельность по обслуживанию имеет две составляющих: непрерывный контроль и подтверждение правильности работы системы, а также оценку возможности усовершенствования системы. Под готовностью системы обычно понимают вероятность того, что она работоспособна в заданный момент времени. Надёжность, готовность и эксплуатационная технологичность программных средств связаны концептуально и функционально. Поэтому их анализ следует производить совместно. Сокращённо эти три свойства обозначают RAM – reliability, availability, maintainability или НГТ. Мы уделяем основное внимание надёжности, поэтому не используем обозначение RAMS, включающее свойство безопасности – security. Этим существенно ограничиваем рассматриваемую проблему.

Существует два основных подхода к анализу проблем RAM. При первом подходе оценивается каждый компонент программного средства на каждом этапе жизненного цикла. При этом показатели RAM программного средства оцениваются «снизу–вверх» по мере введения его в работу и обслуживание. При втором подходе, «сверху–вниз», предусматривается разработка процедурных или технических стратегий, начиная с верхнего уровня таким образом, чтобы обеспечить соответствие требованиям к программному средству. Этот подход обеспечивает более широкий взгляд на программные и процессуальные аспекты RAM, объединяя предупредительные и корректирующие действия для обеспечения надёжности программных средств, а подход «снизу–вверх» сосредотачивает внимание на реализации. Каждый их этих подходов отражает важные особенности RAM, и их учёт позволяет предотвращать концептуальные ошибки в области RAM программных средств. Однако до настоящего времени ещё не определена стратегия объединения этих подходов для многоуровневого анализа программных средств.

Что касается подходов и современных методов оценивания надёжности, то можно указать на наиболее частое использование методологии метрики программного обеспечения, статистических и математических испытаний и моделирования на основе моделей роста надёжности программного обеспечения. Метод метрики программного обеспечения обычно используется для оценивания вероятности наличия дефектов в элементе модуля или модуле. В результате это позволяет обеспечить более эффективные проверки и доработки модулей или сегментов программы.

При оценивании надёжности необходимо учитывать как продукцию (например, число операторов в модуле), так и процесс разработки, реализации и обслуживания программных средств. Одной из основных проблем здесь является неправильное применение методов на основе нереалистических допущений.

В научно-технической литературе отмечается, что для оценивания надёжности программных средств необходимо учитывать некоторые дополнительные характеристики современной техники программного обеспечения. Так, например, возрастающее использование языков программирования четвёртого поколения приведёт к большему удобству сопровождение программных средств. Ряд новых сложных программно-аппаратных систем, таких как системы символической обработки и искусственного интеллекта, архитектура с параллельной обработкой и системы поддержки принятия решений, ещё не проанализированы достаточно полно с точки зрения надёжности. Подобные изменения в методах и стратегиях техники программного обеспечения могут потребовать разработки новых методов и концепций в области оценивания надёжности.

В работе [79] предлагается классификационная схема для оценивания соответствия требованиям по надёжности программных средств. Она служит основой для анализа надёжности программных средств и включает три независимых ключевых измерения программных средств, влияющих на оценку их надёжности. Этими измерениями являются размер, технологический жизненный цикл программных средств (метод) и целевые критерии для понимания условий работы программных средств.

Первое измерение – размер – связано как с программой, так и с процессом. Наиболее общей мерой служит число строк в исходном коде. Другими мерами являются число входов, число выходов и число операций. Могут рассматриваться также тип и область действия интерфейсов, число, размер и взаимосвязь модулей программной системы, количество повторно используемых средств. Эти, связанные с программой, меры предполагается измерять путём проверки исходного кода и оценки операций. Меры, связанные с процессом, разработаны достаточно хорошо и пока обычно не учитываются при анализе надёжности.

К таким мерам относят количество персонала, взаимосвязанного в процессе проектирования, отработки и сопровождения программных средств, так как число каналов и уровни связи влияют на надёжность системы. Приближённую оценку размера системы можно использовать для выбора путей обеспечения надёжности, при этом каждый путь имеет свою совокупность процедур, метрик программирования и деятельности, связанной с процессом. В результате анализа размера систему относят к одной из установленных категорий, – малых, средних или больших проектов. Каждой категории соответствует определённый путь с общими характеристиками программы и процесса. С этими же категориями можно связать показатели надёжности. Однако, пока существуют разногласия в установлении категорий системы по размеру.

Второе измерение – технологический жизненный цикл программных средств – имеет качественный характер и широкую сферу рассмотрения для представления многих элементов категории. Для неё выбирают два индикатора: язык программирования и стратегию обработки (алгоритмическую, символическую и параллельные структуры). Путём разделения технологии на категории и определения конкретных характеристик соответствующих технологий специалисты по программному обеспечению могут выявить, какие стратегии обеспечения надёжности применимы, какую информацию можно использовать для оценивания надёжности проекта, готовность специфичных технологических методов и рекомендуемые модели оценивания надёжности.

Третье измерение – целевые критерии для понимания работы программных средств – представляет собой цели работы системы и важность её характеристик надёжности. Имеет множество элементов, включая объём и стоимости разработки, эксплуатации и отказа; число людей, подверженных действию системы; распределение выгод и затрат от системы, а также приоритеты, даваемые системе её разработчиками, потребителями и государственными или общественными интересами.

Рассмотренные три измерения надёжности обеспечивают основу, связывающую прошлые усилия разработчиков системы в систематизированную стратегию планирования, реализации и оценивания системы. Понимание, достигнутое с помощью такой матрицы из трёх измерений, во многих случаях может помочь исследователям системы и программистам произвести обоснованный выбор среди несметного числа стратегий оценивания и обеспечения надёжности.

Рассмотренный подход применялся к нескольким системам так называемой Стратегической Компьютерной Инициативы (СКИ). Программные средства были их важнейшей составляющей. Однако, не вполне ясно, можно ли разработать требуемые программные средства, и могут ли они быть достаточно надёжными. Программные средства должны обеспечить принятие и координацию сотен и даже тысяч отдельных решений, от которых зависит эффективность системы. Требования к надёжности сложных и взаимосвязанных систем, обрабатывающих огромное количество данных, являются очень высокими.

По приведённой оценке размера одной из систем требуется по крайней мере десять миллионов строк машинного кода. Этот размер существенно больше размера ранее используемых крупнейших систем. Например, для систем МВКА Space Shuttle и Safeguard число кодовых строк составляло около $5 \cdot 10^5$. Сложность создания такой системы усугубляется тем, что она должна работать в реальном времени 24 часа в сутки. Для такой большой системы могут потребоваться новые средства, методы и виды анализа. Стоимость испытаний такой системы может значительно превзойти стоимости конструирования и создания. Предлагаются некоторые пути обеспечения надёжности этой крупноразмерной системы.

В процессе анализа второго измерения отмечается, что для решения проблемы создания программных средств предполагается использовать экспертные системы с базами знаний и другие методы искусственного интеллекта. Однако при этом вопросам надёжности программных средств таких систем часто не уделяется должного внимания. Для этих систем могут потребоваться новые процедуры и методы оценивания надёжности. Важное значение приобретает совершенствование методов макетирования.

В процессе анализа третьего измерения рассматривались влияние операций (всемирных или тактических, многоорганизационных или единичных), их важность для выполнения основных задач, приоритетность которых определяется правилами, доктриной или ситуацией, а также стратегические последствия отказа.

Разрабатывается основа для корреляции главных характеристик системы с требованиями надёжности. Предложенная совокупность трёх измерений (матрица надёжности) позволяет упростить анализ надёжности путём классификации систем по трём измерениям и связи этой классификации с процедурами и оцениванием для различных уровней надёжности. Вместо представления задачи надёжности, как анализа бинарного состояния, или применения известного подхода с рассмотрением метрик, предложенная классификация обеспечивает исследователям путь к оцениванию соответствия системы требованиям по надёжности с учётом большего числа факторов.

В дополнение к организации оценивания традиционных подходов обеспечения и оценивания надёжности программных средств, выполненный анализ основных проблем надёжности может учитывать проблемы, вызванные внедрением новых технологий. Предлагаются некоторые альтернативные решения по отношению к обычной практике работ по надёжности: модификация аппаратной части для преодоления некоторых проблем программных средств, разработка новых усовершенствованных методов оценивания процесса разработки программных средств и уверенность в эффективности количественного или систематизированного качественного анализа деятельности за жизненный цикл. Междисциплинарный анализ может также плодотворно применяться для формирования методик обеспечения и оценивания надёжности. Данные об эффективных мерах надёжности для программных систем с различными характеристиками будут лучше поддерживать усилия проектировщиков, расчётчиков и отработчиков системы.

Таковы взгляды на современное состояние и перспективу развития проблемы надёжности программных систем у ряда зарубежных учёных.

Глава 1

ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММНОЙ ПРОДУКЦИИ КАК ОБЪЕКТА ЭКСПЛУАТАЦИИ

1.1. ПРОГРАММНАЯ ПРОДУКЦИЯ

В 1983 г. программные средства постановлением правительства отнесены к продукции научно-технического назначения и объявлено о создании их индустрии. Были принятые следующие термины-определения [24].

Программа обработки данных (ПОД) (программа) (П) – совокупность конструкций формализованного языка, являющаяся записью алгоритма обработки данных.

Программное средство (ПС) – программа или их взаимосвязанная совокупность, позволяющая реализовать алгоритм обработки средствами вычислительной техники.

Программное обеспечение (ПО) – совокупность ПС, обеспечивающих реализацию целей обработки данных и управления наряду с техническим информационным и другими видами обеспечения.

Программное изделие (ПИ) – ПС, записанное на техническом носителе информации, изготовленное по определённой технологии и укомплектованное эксплуатационной документацией. ПИ должно быть зарегистрировано в Государственном фонде алгоритмов и программ (ГосФАП), снабжено гарантиями поставщика и заказчика, и соответствовать утверждённым ТУ. В Общегосударственном классификаторе продукции (ОКП) выделен специальный класс и ей присваивается определённый код ОКП из данного класса.

Программная продукция (ПП) – ПИ (или их совокупность), разработанное в соответствии с действующей нормативно-

технической документацией, имеющей код ОКП и изготовленное по утверждённой технологии. Таким образом, ПИ став ПП, не перестаёт быть ПС. При этом ПС является кооперативной или индивидуальной научно-технической продукцией.

Макетный образец ПС (МО) – отлаженное ПС, реализующее функции разрабатываемой ПП и предназначенное для отработки и демонстрации принципиальных технических решений.

Опытный образец ПИ (ОО) – ПИ, обладающее всеми функциональными характеристиками разрабатываемой ПП, выполненное по технологии разработчика и предназначенное для проведения испытаний, опытной эксплуатации и постановки на производство изготовителем ПИ.

Эталонный образец ПИ (ЭО) – ПИ, обладающее всеми функциональными характеристиками разрабатываемой ПП, предназначенное для хранения в программных фондах ГосФАП и изготовления ПП по утверждённой технологии.

Серийный образец ПИ (СО) – единичный образец конкретной ПП, предназначенной для применения в соответствии с функциональным назначением.

Товарная программная продукция (ТПП) – ПП, предназначенная для поставки и применения в соответствии с функциональным назначением, имеющая товарный знак изготовителя и соответствующую цену, назначенную в установленном порядке.

Приведённые определения позволяют рассматривать надёжность ПИ как надёжность промышленной продукции.

По сравнению с техническими изделиями, 70…95% стоимости которых составляют затраты на сырьё и обработку материалов, процесс производства наукоёмких ПИ не требует больших относительных расходов материальных ресурсов, но требует расходов ресурсов людских. Тиражирование ПИ и ЭД увеличивает стоимость ПП на 2…5%. Но имеется тенденция уменьшения относительной стоимости разработки ПС из-за: 1) развития технологии изготовления средств; 2) расширения числа типовых технологических операций; 3) использования агрегатирования.

1.2. КЛАССИФИКАЦИЯ ПРОГРАММНЫХ СРЕДСТВ

Не любое ПС имеет смысл изготавливать и поставлять в качестве ПП, что делает целесообразным выделение подклассов ОКП, для которых применимы методы обеспечения надёжности ПИ. Выделяют следующие подклассы ПС: системные ПС, прикладные ПС и прочие ПС (общего хозяйственного назначения).

Системные ПС: операционные системы и средства их расширения; средства управления базами данных; средства создания и преобразования программ; средства интерфейса и управления коммуникациями; средства организации вычислительного процесса; сервисные программы; средства обслуживания ВТ.

Характерной особенностью системных ПС является их широкое распространение на СВТ, что обеспечивает быстрое выявление дефектов и как следствие сравнительно высокую надёжность функционирования ПО данного подкласса. Разработчики системных ПС – специалисты высокой квалификации, оснащённые инструментарием, которое способствует повышению надёжности ПП. Для значительного их количества характерна сильная взаимосвязь со структурой и логикой функционирования СВТ и большой объём (порядка 10^6 более операторов языка). Указанные особенности обуславливают своеобразие задач и методов исследования и обеспечения надёжности ПС, которые требуют специализированных методов и средств.

Прикладные ПС: это ПС для научных исследований; ПС для проектирования; ПС для управления техническими средствами и технологическими процессами; ПС для решения организационно-экономических задач. Они характеризуются множеством разновидностей, меньшим объёмом составляющих их П (порядка 10^4 операторов языка). Являются результатом использования различных технологий программирования. Объём внедрения каждого такого ПС не позволяет рассчитывать на обеспечение достаточного его контроля и, соответственно, уровня надёжности за счёт массового применения. Поэтому на практике уделяют основное внимание вопросам обеспечения надёжности прикладных ПС, тем более что число разработчиков прикладного ПО намного больше, чем системного. Поэтому рассматриваемые методы и средства будут полезны широкому кругу программистов и технологов.

Особое место в классификации ОКП и данного подкласса занимают программы для управления техническими средствами и технологическими процессами. Они ориентированы на применение в составе ВС (специализированных технических СВТ, относящихся к классам научно-технической продукции, СВТ военного и народнохозяйственного назначения). К этому виду ПС обычно предъявляются особые требования в плане надёжности, так как именно в них ошибки часто приводят к отказовым ситуациям с катастрофическими последствиями при функционировании тех сложных технических объектов, которые используют ПС данного подкласса. Следует особо подчеркнуть, что ПС для управления техническими средствами и технологическими процессами рассматриваются в качестве программных комплектующих изделий управляющих вычислительных комплексов (УВК) различных объектов.

Прочие ПС: бухгалтерские, различные сервисные ПС народно-хозяйственного назначения достаточно широкого профиля и другие. К ним предъявляются также определённые требования по качеству и надёжности. Но эти требования значительно менее жёсткие, именно такие, чтобы удовлетворять потребности приобретающих их пользователей.

1.3. ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНЫХ ИЗДЕЛИЙ

Жизненный цикл ПИ состоит из трёх фаз: разработки, производства и использования или эксплуатации. На первых порах развития ПО основное внимание уделялось первой фазе. Это объясняется тем, что в начале развития систем ОДУ практически все стадии жизненного цикла ПС реализовывались разработчиком программ и с его участием. Предполагалось, что функциональных и эксплуатационных свойств ПО определялся исключительно фазой разработки. Поэтому модели жизненного цикла ПС в первую очередь отвечали интересам разработчиков. Методы создания ПС в настоящее время наиболее развиты по сравнению с методами производства и эксплуатации. Научно-техническое обеспечение других двух фаз жизненного цикла начало создаваться позднее.

Концептуальная диаграмма, «замыкающая» между собой начальную и завершающую фазы жизненного цикла через ***стадию сопровождения***, показана на рис. 1.1.

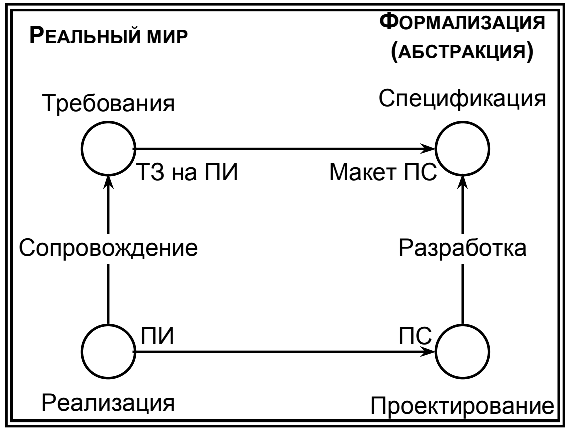


Рис. 1.1

Под ***сопровождением*** П понимают совокупность действий, обеспечивающих работоспособность ПИ, функционирующих у пользователей ПИ в соответствии со спецификацией. В рамках сопровождения могут выполняться работы по модификации ПИ при изменениях требований пользователя или условий эксплуатации, которые не вызывают необходимости полной замены самих ПИ. Важность стадии сопровождения определяется возрастанием масштабов разработки ПО, увеличением его сложности, расширением сферы применения, возможным отчуждением ПП от разработчика и увеличением числа пользователей каждого ПИ.

Стоимость работ по сопровождению ПС составляет примерно 50…60% всех затрат на него в течение жизненного цикла, 30% вложений направляется на разработку новых ПС, остальные 10…20% расходуются на поддержание работоспособности и модернизацию функционирующих ПС.

Важной также является ***стадия фондирования*** программ – выбор, испытания и подготовка к серийному производству ПИ. Определяющим компонентом этой стадии является экспертиза полезности рассматриваемого ПС для определённого контингента

пользователей, подготовка и проведение контрольных и приёмочных испытаний, в результате которых принимается решение о целесообразности постановки продукции на производство. Необоснованность такого решения может приводить к материальному ущербу, на несколько порядков превосходящему затраты на производство ПИ. Структура жизненного цикла ПИ и примерное распределение относительных затрат по его стадиям показаны на рис. 1.2.

Фазы	Стадии	Относит. затраты в %
Разработка	Проектирование Программирование Испытания	10 5 10
Производство	Фондирование Тиражирование Поставка	15 2 2
Использование	Ввод в эксплуатацию Сопровождение Снятие с эксплуатации	5 50 1

Рис. 1.2

Целью фазы разработки является создание макета или опытного образца ПС. Используемые на ней методы связаны с разработкой внутренней структуры ПС, а П выступает в роли предмета труда.

Цель фазы производства – выбор из представленных к фондированию П эталонного образца и изготовление ПИ. На этой фазе моделируется вычислительный процесс, реализуемый П, которая по-прежнему является предметом труда.

Цель фазы использования ПИ – реализация экономического и технического эффекта от его эксплуатации. Методическую основу здесь составляют организационно-управленческие мероприятия. На этой фазе П является уже средством труда при обработке информации.

ПИ на каждой стадии приобретает существенно новые свойства:

– на стадии проектирования создаётся алгоритм обработки данных,

– на стадии программирования он приобретает свойство П,

– на стадии испытаний становится ПС,

– на стадии фондирования П превращается в эталонный образец,

– на стадии тиражирования преобразуется в ПП, на стадии поставки осуществляется доведение его до потребителя.

Новые свойства ПП на стадиях фазы использования обуславливаются превращением ПС из предмета труда в его орудие.

Степень решённости задач по обеспечению работ завершающих стадий жизненного цикла недостаточна. Особенно важными с точки зрения реализации затрат являются стадии фондирования и сопровождения. В рамках первой из них принимается решение о производстве того или иного ПИ, то есть фактически планируется конечная эффективность использования П по всему жизненному циклу. В рамках второй стадии реализуются все свойства ПИ, в том числе и эффективность.

Известно [31], что чем раньше в жизненном цикле будет достигнут заданный уровень надёжности П, тем меньшие затраты потребуются для его достижения. Оценка роста затрат на исправление программных дефектов показывает, что при переходе от стадии к стадии стоимость корректировки в среднем возрастает на порядок. Это является ещё одним обоснованием важности стадии фондирования П с точки зрения оценки требуемого уровня их надёжности при промышленном производстве ПП.

1.4. ОСНОВНЫЕ СТАНДАРТЫ И ГОСТЫ ПО ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

1. Международный стандарт ISO 9126: 1991. ИТ. «Оценка программного продукта. Характеристики качества и руководство по их применению».

Рекомендует 6 основных характеристик качества ПС, каждая из которых детализируется несколькими субхарактеристиками (всего 21). Основные характеристики качества следующие.

Функциональная пригодность – пригодность для применения, точность, защищённость, способность к взаимодействию и

согласованность со стандартами и правилами проектирования. Функциональная пригодность проявляется в корректности и надёжности ПС.

Надёжность – уровень завершённости – отсутствие ошибок, устойчивость к ошибкам, перезапускаемость.

Применимость – понятность, обучаемость, простота использования.

Эффективность – ресурсная экономичность, временная экономичность.

Сопровождаемость – удобство для анализа, изменяемость, стабильность, тестируемость.

Переносимость – адаптируемость, структурированность, замещаемость, внедряемость.

2. ГОСТ 28806-90 – «Оценка качества программных средств. Термины и определения». ГОСТ наиболее близок к ISO 9126: 1991. В нём даны общие термины и определения. Они обязательны для применения во всех видах документации и литературы по вычислительной технике и ПС, входящих в сферу работ по стандартизации и использующих результаты этих работ.

3. ГОСТ 19781-90 – «Обеспечение систем обработки информации программное. Термины и определения».

4. ГОСТ 28195-89 – «Оценка качества программных средств. Общие положения». Введён 1.07.1990 г.

1.5. СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА НАДЁЖНОСТИ ПРОГРАММНОЙ И ТЕХНИЧЕСКОЙ СИСТЕМ

Надёжность технической системы определяется как её свойство выполнять заданные функции, сохраняя во времени значения установленных эксплуатационных показателей в заданных пределах, соответствующих заданным режимам и условиям использования, технического обслуживания, ремонта, хранения и транспортирования [20]. Свойство надёжности проявляется в том, что система должна выполнять предусмотренные для неё функции. Потеря надёжности системы означает её отказ в работе. Из определения следует, что должны быть известны заданное время работы, среда функционирования и эксплуатационные показатели системы.

Приведённое определение надёжности формально приемлемо и для сложной системы «программное обеспечение». Однако в этом определении не отражено различие между видами причин, приводящих к неработоспособности этих систем. Надёжность ПО должна определяться в процессе функционирования как функция от влияния его ошибок. Она в значительной степени отличается от надёжности технической системы. ПО не подвержено износу во времени. Его ненадёжность почти целиком определяется ошибками разработки. Часто сравнивают между собой кривые интенсивность отказа технической и интенсивность ошибки программной систем как показано на рис. 1.3.

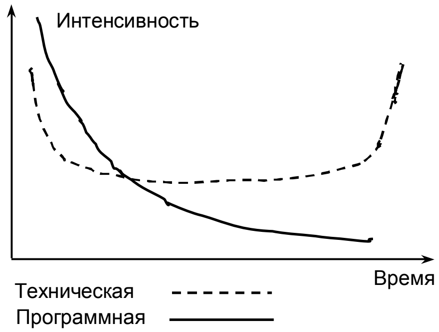


Рис. 1.3

Такое сравнение, на наш взгляд, не вполне справедливо. Интенсивность отказа технической системы условно разделяют на три участка: приработки или детства системы, нормальной эксплуатации и старения или износа. Интенсивность отказа является показателем надёжности невосстанавливаемой системы и характеризует её надёжность до первого отказа. Интенсивность ошибки программной системы изображается всегда монотонно убывающей кривой. Она, в отличие от интенсивности отказа технической системы, является показателем надёжности восстанавливаемой программной системы, как таковой, какой она является на самом деле. Это объясняется тем, что в процессе тестирования и отладки ПО обнаруженные ошибки всегда устраняются, что

учитывают при построении математических моделей надёжности ПО, но при определении интенсивности ошибки она, как правило, определяется как интенсивность отказа аппаратуры. Эта неточность должна быть устранена при определении показателей надёжности ПО, так как, на самом деле, мы имеем дело с потоком ошибок восстанавливаемой программной системы. Потоки восстанавливаемых систем с математической точки зрения описываются всегда интегральными уравнениями восстановления. Поэтому, определяя показатели надёжности ПО в условиях устранения их ошибок, всегда надо начинать с изучения потока ошибок с помощью соответствующего интегрального уравнения. Для редеющего потока ошибок, каким является он на практике, следует из теории надёжности аппаратуры, что интенсивность программной ошибки будет всегда меньше величины параметра потока ошибок. На это будет указано в главе, посвящённой изучению потока ошибок.

Для того, чтобы подчеркнуть различие между отказом и ошибкой, можно отметить, что отказ оборудования не зависит от обрабатываемой в системе информации. А программные ошибки в большей степени зависят от входной информации. Причина появления ошибки в конкретный момент времени заключается в том, что в этот момент была обработана уникальная последовательность входных данных, приведших к осуществлению такой траектории вычислений, на которой и возникла ошибка. Отказы оборудования зависят от времени. Ошибки ПО являются функцией от текущей входной информации и текущего состояния системы. Кроме того, считается, что ошибки ПО имеют, как правило, систематический характер, а отказы оборудования – случайный характер.

По своей природе ПО значительно сложнее оборудования. Входной поток системы ПО в зависимости от области использования и желания пользователя может быть достаточно сложным.

По области применения ПО является более универсальным по сравнению с технической системой.

Оборудование и ПО различаются по природе составных элементов. Блоки ПО являются стандартизованными и более примитивными. К ним относятся операторы на некотором языке, части ранее написанных программных систем. Это значит, что разработчик ПО для обеспечения надёжности должен выполнить более сложную работу по сравнению с разработчиком оборудования. Современные вычислительные системы (ВС) имеют развитое и достаточно сложное ПО. Это порождает огромные трудности по проверке и поддержанию работоспособности таких систем. ВС включают в себя два неотделимых аспекта: технический элементный и функциональный. Первый рассматривается в рамках классической или традиционной теории надёжности. Второй, функциональный, отражает правильность, безошибочность работы ПО систем. Различие в природе этих составляющих обуславливает то, что развитые методы надёжности технического обеспечения, не всегда применимы для ПО.

При анализе надёжности технических систем используется понятие элемент, более точно расчётный элемент, то есть элемент, имеющий самостоятельные или свои показатели надёжности. Число состояний такого элемента ограничено числом небольшой размерности. При анализе функциональной надёжности привлекают понятие структуры систем. Понятие структура указывает на то, что система ПО делится на модули с определёнными связями между ними. Модуль – это также элемент, но имеющий гораздо больше состояний, чем элемент оборудования. Часто модуль означает единицу измерения работы отдельного программиста или группы программистов. Модуль может конструироваться без знания внутренней структуры других модулей ПО. Конкретный облик системы, её структура, определяются набором специфических модулей с определёнными связями с другими модулями – интерфейсами. Спецификации межмодульных интерфейсов и составляют описание структуры системы, определяя её поведение.

С позиций более чёткого различия двух систем можно указать основные отличительные особенности ПО.

Отсутствие физического износа и старения. Исключается возникновение дефектов, имеющих стохастическую природу. Поэтому нет необходимости в проведении планово-предупредительных ремонтов ПО в процессе эксплуатации. Модернизация ПО, целенаправленное внесение изменений в него создаёт, по существу, другое ПО, имеющее свои характеристики надёжности. Поэтому при сопровождении ПО, в отличие от других видов продукции, можно обеспечивать не только поддержание показателей надёжности на некотором уровне, но и увеличение этого уровня.

Более высокий темп морального старения. Требуется создание новых видов ПО для систем обработки данных. Корректировка при модификации ПО в процессе его эксплуатации является одним из факторов снижения надёжности ПО. Например, если в ПО изменено 10…20 операторов языка, то вероятность внесения ошибки составляет приблизительно 0,5, а если изменено 50 операторов, – то она увеличивается до 0,8.

Введение различных видов избыточности в ПО, в частности его резервирование, малоэффективно. Избыточность как традиционное средство повышения надёжности в случае ПО обычно реализуется за счёт контроля входных, промежуточных и окончательных результатов вычислений. Недостаточность такого контроля является одним из основных факторов утраты работоспособности систем ПО (≈ 50% программных ошибок). Ещё одним видом избыточности является использование механизма контрольных точек, когда фиксируется состояние вычислительной среды в определённые моменты времени и выполняются рестарты промежуточных вычислений в случае необходимости.

К другим отличиям ПО по сравнению с ТС можно отнести:

– число способов и элементов контроля у ПО значительно больше, чем у аппаратуры;

– в ПО гораздо проще вносить исправления и дополнения, чем в аппаратуру, но делать это корректно и безошибочно достаточно трудно.

1.6. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ ВОЗДЕЙСТВИЙ НА ПРОГРАММУ

В настоящее время согласно работе [24] принята следующая модель воздействий от различных источников на П со стороны техники (*Т*), информации (данных) (*И*), человека (*Ч*) и программы (*П*): $\langle П \to П; Т \to П; И \to П; Ч \to П \rangle$, которая соответствует следующей схеме вычислительной среды, показанной на рис. 1.4.

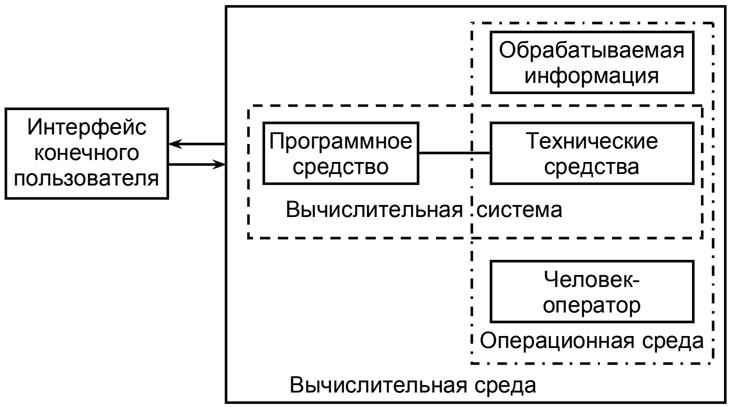


Рис. 1.4

Данная модель позволяет выделить следующие ошибки:

– ***программную*** $(П \rightarrow П)$ – проявление в П ошибки, которая ранее не была обнаружена и возникла при каком-то конкретном сочетании исходных данных и команд, соответствующих спецификации;

– ***аппаратную*** $(Т \rightarrow П; ОС \rightarrow П)$ – возникает при отказе (сбое) ТС и/или при появлении ошибок в ОС (СУБД, табличном процессоре и т.д.), которые привели к искажению результата работы П;

– ***информационную*** $(И \rightarrow П)$ – возникает вследствие ошибки в информации (входных данных (ВД)) и искажает результат работы П;

– ***эргатическую*** $(Ч \rightarrow П)$ – возникает из-за ошибки персонала (ЧО) и искажает результат работы П.

1.7. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Дефект – от латинского «defectus» – изъян, недостаток, от русского – недостаток, изъян, нарушение, отклонение и др. [61]. Под дефектом П в общем случае будем понимать несоответствие функций, заложенных в П, требованиям, оговоренным в спецификации. На наш взгляд, дефекты П нужно подразделять на явные и неявные. Явный дефект при самом полном и тщательном

контроле П может быть всегда выявлен и устранён. Неявный дефект никогда не может быть обнаружен и, следовательно, устранён. Неявные дефекты могут проявлять себя только при таком сочетании компонент вычислительного процесса, когда они проявляются как явные (например, при таком сочетании команд и последовательно получаемых численных результатов, которые реальную траекторию вычислений выводят за поле допуска по величине отклонения, или когда накопленное численное вычислительное отклонение приводит к нарушению некоторого предусматриваемого предикатного условия, а также по другим причинам). В работе [31] отмечается, что «при отладке и тестировании П обычно сначала обнаруживаются ***вторичные ошибки***, то есть последствия и результаты проявления некоторых исходных дефектов, которые следует квалифицировать как ***первичные ошибки*** (по причине вложенных аномалий). Локализация и корректировка первичных ошибок приводит к устранению ошибок, первоначально обнаруживаемых в результатах функционирования П». При этом автор к первичным ошибкам относит:

– технологические ошибки подготовки машинных носителей и документации, ввода П в память и вывода их на внешние средства отображения;

– программные ошибки, допущенные вследствие неправильной записи исходного текста П на языках программирования и ошибок трансляции П в объектный код;

– алгоритмические ошибки, связанные с неполным формированием необходимых условий решения и некорректной постановкой задач;

– системные ошибки, обусловленные отклонением функционирования П в реальной среде и характеристик внешних объектов от предполагающихся при проектировании.

Данные ошибки приводятся в порядке усложнения их обнаружения и увеличения ресурсов на устранение. Приведены следующие данные: при автономной и в начале комплексной отладки доля системных ошибок составляет (≈ 10%), на завершающих этапах комплексной отладки (≈ 35…40%), в процессе сопровождения (≈ 80%) всех ошибок. Среднее число команд, корректируемых при исправлении каждой системной ошибки ≈ 25 строк текста на одну ошибку.

Характер проявления вторичных ошибок П служит основой предполагаемых гипотез для построения математических моделей надёжности П.

Вторичные ошибки подразделяют на три группы [31].

1 ГРУППА:

– наблюдаемость искажений данных П и вычислительного процесса, обусловленных первичными ошибками в П (первичная ошибка либо фиксируется и исправляется после тестирования, либо вообще не обнаруживается);

– множество тестов равномерно покрывает множество реальных исходных данных (случайно относительно области изменения исходных данных П и первичных ошибок в ней). Типы команд перемешаны. Поэтому время между ошибками определяется средним временем выполнения команды на данной ЭВМ и средним количеством команд, исполняемых между ошибками;

– маршрут следования П вычислений случаен. Интенсивность проявления ошибки при реальном функционировании П зависит от среднего быстродействия ЭВМ и практически не зависит от конкретного распределения типов команд между ошибками на маршрутах обработки данных;

– квалификация специалистов и загруженность постоянны на интервале тестирования, исследования.

2 ГРУППА (ОСНОВНАЯ):

– интервалы между ошибками статистически независимы;

– интенсивность проявления ошибок остаётся постоянной пока не произведено исправление первичной ошибки и не изменена программа по другой причине. Интервалы между ошибками распределены экспоненциально;

– интенсивность обнаружения вторичных ошибок пропорциональна суммарному числу первичных ошибок, имеющихся в данный момент в П;

– каждая обнаруженная ошибка подлежит исправлению. Поэтому частота исправления ошибок пропорциональна частоте их обнаружения, однако некоторые исправления в свою очередь содержат ошибки. Некоторые ошибки являются связанными, и при обнаружении одной следует исправление нескольких первичных ошибок. Поэтому частота обнаружения и частота их исправления не равны, а связаны некоторым коэффициентом пропорциональности.

3 ГРУППА детализирует использование ресурсов на корректировку П и повышение их качества. Эти гипотезы сильно связаны с технологией и уровнем автоматизации производства П. Использование их ограничено и далее не используется.

На основе этих допущений строят экспоненциальную модель распределения ошибок в П (модель Дж. Мусы).

В заключение рассмотрения дефектов и ошибок ещё раз акцентируем внимание на том, что, по нашему мнению, дефект это не ошибка, которую можно обнаружить, а изъян, который себя может проявлять только при стечении определённых обстоятельств в процессе функционирования П (потенциальная ошибка). Для оправдания этого положения можно было бы привести несколько примеров из других областей науки и техники. Но для окончательного решения данный вопрос оставляем на изучение читателю.

Программная ошибка – несоответствие результата реализации П предъявляемым к ней требованиям по точности или времени выполнения. Следует различать понятие ошибки и отказа. На наш взгляд, программная ошибка – событие, связанное с работой П, а отказ – событие, связанное с работой технической, организационной и т.д. системы, в которой используется П.

Отказ – событие, после наступления которого система теряет свою работоспособность из-за ошибки ПО, используемого в системе.

Итак, не всякий дефект П приводит к программной ошибке, и не всякая программная ошибка приводит к отказу системы. Программная ошибка должна определяться требованиями, предъявляемыми к П. Отказ любой системы должен определяться требованиями, предъявляемыми к системе либо со стороны технического, либо со стороны программного обеспечения. Говорить об отказе ПО имеет смысл только в некоторых вырожденных случаях, когда только ПО составляет саму систему. Более корректно надо представлять в исследованиях систем с ПО связь «дефект –

ошибка – отказ системы с ПО». Это устранит недоразумения при формулировке отказа в ПО и использующих его системах.

В соответствии с данным замечанием, на наш взгляд, следует по-разному формулировать понятие надёжности ПО и понятие надежности системы.

Прогон программы – набор действий, включающий в себя ввод в П любой возможной комбинации пространства входных данных, выполнение программы при данном наборе, и получение результата при данном наборе данных в виде приемлемого или ошибочного результата.

Надёжность системы (безотказность) – свойство системы сохранять свою работоспособность в определённых условиях эксплуатации в течение заданного времени.

Надёжность П (безошибочность) – свойство П соответствовать предъявленным к ней требованиям в процессе её функционирования в течение заданного времени в условиях определённой среды.

К приведённым определениям надёжности следует добавить, что часто используемое в литературе для неё понятие, отождествляемое с показателями надёжности, например вероятностью, не является правомочным. Надёжность – объективное свойство, а показатель – мера этого свойства. Свойство может долго сохраняться, а мера может изменяться неоднократно, кроме того, она в отличие от свойства не может быть единственной.

В международном стандарте ISO 9126: 1991 отмечается, что функциональная пригодность ПС проявляется в их корректности и надёжности. Приведём полную формулировку корректности ПС, данную в этом стандарте:

«Корректная (правильная) П может рассматриваться статически вне её исполнения во времени. Она не определена вне области изменения исходных данных, заданных в спецификации, не зависит от динамики функционирования в реальном времени. Степень некорректности П определяется вероятностью попадания реальных исходных данных в область, которая задана требованиями спецификации и технического задания, однако не была проверена при тестировании и испытаниях». Далее «… надёжная П должна обеспечивать достаточно низкую вероятность ошибки в процессе функционирования в реальном времени. Является понятием динамическим, проявляющемся во времени и существенно отличается от понятия корректности П».

Устойчивость функционирования – способность к безошибочному функционированию П во времени при наличии воздействий $(П \to П); (Т \to П); (И \to П); (Ч \to П)$. Различают два вида устойчивости: 1) т о л е р а н т н а я   у с т о й ч и в о с т ь – способность П продолжать свою работу и обеспечивать приемлемые результаты при возникновении технологических ошибок. Под технологическими ошибками подразумеваются ошибки, обусловленные четырьмя указанными воздействиями; 2) к о н с е р в а т и в н а я   у с т о й ч и в о с т ь – способность П перейти в состояние «защитной ошибки» при тех же четырёх указанных воздействиях.

Итак, в теории надёжности ПО в отличие от теории надёжности техники выделяют понятие статическое – «корректность» и понятия динамические – «надёжность» и «устойчивость». В теории надёжности техники статического понятия не существует, так как её понятия всегда связаны со временем.

1.8. РАСКРЫТИЕ ПОНЯТИЯ КОРРЕКТНОСТИ ПРОГРАММ

Приведённое в предыдущем разделе понятие корректности дано в сжатом виде. В связи с тем, что это понятие вводится в теории надёжности впервые и является очень важным для количественного оценивания рассмотрим его более детально [26].

Под ***корректностью*** П понимают её соответствие некоторому эталону или совокупности формализованных эталонных правил и характеристик. Наиболее полным эталоном корректности П является *программная спецификация*. Особенностью её является задание требований поведения П для допустимого набора входных данных. Поэтому корректная П может неправильно работать или даже сбиваться на недопустимых наборах входных данных. Свойством устойчивости к недопустимым наборам входных данных обладает надёжная П – в этом заключается разница между надёжной и корректной П.

Требования к корректности делятся в зависимости от двух видов критериев качества:

– для функциональных критериев они определяются предметной областью и функциями выполняемой программы;

– для конструктивных критериев они определяются общими для всех программ свойствами.

В зависимости от проверяемых компонентов П различают следующие виды корректности, представленные на рис. 1.5.

Рис. 1.5

1. *Корректность текстов программ* имеет только конструктивную составляющую; благодаря жёстким правилам языков программирования синтаксическая и семантическая корректность П на этапе её трансляции и после трансляции П является корректной с этой точки зрения.

2. *Корректность программных модулей* имеет и конструктивную и функциональную составляющие:

– конструктивная составляющая определяется правилами построения структуры модулей, задаваемых в технологии и языке программирования,

– функциональная составляющая корректности модулей зависит от предметной области и функциональных спецификаций П.

Функциональная составляющая может проверяться в различных условиях:

детерминированная – для фиксированных наборов входных данных должны быть получены конкретные значения результатов;

стохастическая – входные данные задаются случайными величинами с известными законами распределения и результаты также должны быть случайными величинами с требуемыми законами распределения и заданными корреляционными связями между входными и выходными данными;

динамическая – характерна для систем реального времени и определяется согласованием во времени порядка поступления входных данных и порядка выдачи результатов выполнения П.

В общем случае функциональные спецификации П определяют и функциональные требования к П, и характеристики, с которыми они должны обеспечиваться, как это показано на рис. 1.6.

Рис. 1.6

3. *Корректность данных* имеет конструктивную и функциональную составляющие.

Структурная корректность данных относится к конструктивной составляющей и предполагает правильность построения структурированных данных в программе: массивов, стеков, очередей и т.п. Функциональная корректность данных определяется диапазонами изменения из значений и соответствием типов полей структур типам значений данных.

4. *Корректность комплексов программ*. Также имеет конструктивную и функциональную составляющие: конструктивная составляющая определяется корректностью структуры межмодульных связей по управлению и данным, определяемым в интерфейсных требованиях к П; функциональная корректность комплекса программ определяется так же, как и функциональная корректность модулей.

1.9. ПОКАЗАТЕЛИ НАДЁЖНОСТИ ПРОГРАММНЫХ СРЕДСТВ

В соответствии с особенностями разработки, производства и эксплуатации ПП, обуславливающими наличие и проявление в ней дефектов и ошибок, как событий, имеющих стохастическую природу, показателями надёжности функционирования ПС могут все показатели, имеющие вероятностную основу. В качестве аргумента интервальных показателей могут использоваться время и число прогонов.

Следует отметить, что иногда в литературе по надёжности ПС интенсивность ошибки называют ф у н к ц и е й р и с к а [26].

Данное утверждение, кроме того, поддерживается требованием согласованности значений показателей сложных систем, включающих в свои структуры аппаратные, программные и эргатические компоненты. Это обуславливается необходимостью получения единых системных показателей как в процессе анализа сложных трёхкомпонентных систем, так и при распределении требований по надёжности между указанными компонентами в процессе решения задачи синтеза. Задача анализа обычно называется прямой задачей, а задача синтеза – обратной задачей. Традиционно задача синтеза может формулироваться в двух вариантах. При первом варианте требуется обеспечить систему с заданным показателем качества или надёжности при минимальных затратах располагаемых ресурсов. При втором варианте требуется достичь максимального значения выбранного показателя качества или надёжности при заданных значениях ресурсов. При необходимости указанные задачи могут быть сформулированы однокритериальными или многокритериальными.

Что касается показателей, зависящих от аргументов – времени и прогонов – то они могут быть обычными, как это принято в теории надежности ТС.

В интересах исследований систем могут использоваться и показатели в виде различных коэффициентов.

Выбор показателей вероятностного характера обуславливается также и тем, что при исследовании эффективности операций, живучести, безопасности и т.п. показатели надёжности ПС могут достаточно просто выступать в качестве аргументов этих более сложных свойств.

Изложенные соображения по показателям исключают необходимость как их детальной трактовки, так и математического представления в силу широкой известности в научно-технической литературе.

Следует отдельно рассмотреть только представление одного показателя, связанного с прогоном П. Пусть E_i – один из

входных наборов данных $E_i \in E$, где E – пространство входных данных, при котором получен правильный результат или ошибка. Если $\left|F'(E_i) - F(E_i) \le \Delta_i\right|$, где $F'(E_i)$ – полученный на выходе результат, $F(E_i)$ – требуемый результат, а Δ_i – допустимое отклонение, то решение при данном прогоне безошибочное. Если же $\left|F'(E_i) - F(E_i) > \Delta_i\right|$, то при решении произошла ошибка. Введём индикатор ошибки

$$\varphi_i = \begin{cases} 0, \text{ если ошибка отсутствует;} \\ 1, \text{ если произошла ошибка.} \end{cases}$$

Тогда статистическая оценка вероятности наступления ошибки П при её m прогонах будет равна:

$$Q(m) = \frac{\sum_{i=1}^{m} \varphi_i}{m} .$$

Пусть δ – приемлемая величина погрешности оценки $Q(m)$ для вероятности ошибки П Q. Тогда требуемое количество прогонов П m должно быть пропорционально значению $(Q\delta^2)^{-1}$, где заданная вероятность ошибки П. Это означает, что если, например, требуемая относительная ошибка оценки $\delta = 0{,}1 = 10\%$, и желаемое значение $Q = 0{,}001$, тогда количество независимых прогонов П m должно быть не меньше, чем $m \approx 10^3 \cdot 10^2 = 10^5$, что, конечно, на практике реализовать достаточно нелегко. Для устранения данной неудовлетворённости рекомендуют, например, использование метода последовательного анализа А. Вальда.

И последнее, что касается показателей надёжности ПС, на наш взгляд, необоснованно забыто использование в анализе их надёжности широко применяемого в теории надёжности ТС показателя – параметр (интенсивность) потока отказов, в нашем случае – потока ошибок. Этот вопрос будет рассмотрен в следующем разделе.

В работе [2] предложен показатель надёжности – средняя тяжесть ошибок (СТО). Он характеризует тяжесть ошибок ПО и его возможные последствия для того объекта, на котором оно используется. По нашему мнению, это показатель не надёжности

ПО, а показатель безопасности (эффективности действия – качества функционирования) объекта при появлении возможных ошибок ПО. Авторы отмечают, что «значение этого показателя субъективно и может быть различным для одного и того же программного продукта в зависимости от области его применения. Это объясняется тем, что при использовании конкретного ПО, например, для выполнения студенческих расчётов и для выполнения конструкторских расчётов в космической промышленности последствия ошибок программы – несопоставимы. В ряде случаев, если к ПО предъявляются жёсткие требования, лучше оценивать максимальную тяжесть ошибок ПО. Таким образом, оценивая вероятность сбоя ПО и СТО ПО, получаем многостороннюю оценку надёжности ПО». Эта цитируемая фраза из их работы подтверждает наше замечание о предложенном показателе качества объекта.

Приведём формализованное представление показателя СТО, данное его авторами: $B = \frac{1}{Q}\sum_{i=1}^{m}(b_i p_i z_i)$, в котором Q – вероятность ошибки, b_i – функция принадлежности тяжести последствий ошибки, возникшей при i-м наборе входных данных, к максимально тяжёлым последствиям; p_i – вероятность ввода i-го набора входных данных при эксплуатации ПО; z_i – индикатор ошибки, равный единице, если при i-м наборе входных данных была зафиксирована ошибка, и нулю – в противном случае, m – общее число наборов входных данных. Значения показателя СТО лежат на интервале [0, 1]. Чем ближе значение СТО к 1, тем тяжелее последствия ошибок ПО, и тем менее надёжна П, а близость СТО к 0 показывает незначительность последствий ошибок П – по мнению авторов.

Глава 2

ПОТОКИ СЛУЧАЙНЫХ СОБЫТИЙ И ИХ ХАРАКТЕРИСТИКИ

2.1. КАЧЕСТВЕННОЕ СРАВНЕНИЕ ПОТОКОВ СОБЫТИЙ ТЕХНИЧЕСКОЙ И ПРОГРАММНОЙ СИСТЕМ

Программные системы во времени, как правило, характеризуются последовательностью случайных событий, называемых потоком ошибок. С позиции теории надёжности ТС программы являются восстанавливаемыми объектами. Это объясняется тем, что после обнаружения и устранения любой ошибки в них, они вновь представляются работоспособными системами. Значит работоспособность их, после наступления ошибки, трактуемой в теории надёжности ТС как отказ, вновь возобновляется. Это говорит о том, что программная система после устранения ошибки как бы снова восстанавливается и приобретает снова свойство работоспособной системы. Таким образом, случайный процесс интервалов работоспособности П приобретает свойства процесса восстановления. Отличие этого процесса от обычного процесса восстановления заключается только в том, какие свойства приобретает этот процесс по сравнению с обычным процессом восстановления. В условиях эксплуатации П эти свойства характеризуются, как правило, неполнотой восстановления. Обычно, устранение ошибки в П приводит к уменьшению числа оставшихся в ней ошибок, по крайней мере на единицу. Это значит, что мы имеем случай с редеющим потоком ошибок. Но данное обстоятельство не позволяет утверждать, что ошибки в П в будущем вообще исчезнут. Свойства надёжности П во времени улучшаются, это эквивалентно в ТС замене отказавшего изделия в момент отказа изделием, более надёжным к данному времени, чем отказавшее. Идёт процесс омолаживания П при её эксплуатации. В теории надёжности ТС омолаживания практически не происходит никогда. В самом лучшем случае отказавшее изделие может быть заменено новым изделием, не выработавшим к моменту замены никакой величины своего ресурса надёжности. В самом худшем случае замена отказавшего изделия может производиться изделием, выработавшим такой же ресурс, как и отказавшее изделие. Таким образом, параметр потока отказов ТС может быть только параметром потока с полным или частичным восстановлением работоспособности изделий. В отличие от него, параметр потока ошибок П является параметром потока с заменой П при возникновении в ней ошибок программой более высокой надёжности. Таким образом, параметр потока отказов ТС является параметром потока «стареющей системы», а параметр потока ошибок ПС является параметром потока «молодеющей системы». В этом их основное отличие. Это наводит на мысль о том, что в рассматриваемых потоках определяющей величиной является степень обновления наблюдаемых событий. В технических системах наибольший эффект достигается при замене отказавших систем новыми, не израсходовавшими своего ресурса надёжности. В программных системах наибольший эффект достигается при замене ошибающихся систем старыми, имеющими достаточно большую временную наработку.

2.2. ПАРАМЕТР ПОТОКА ПРИ ПОЛНОМ ОБНОВЛЕНИИ СИСТЕМ

В данном разделе рассматривается хорошо известный из теории надёжности и теории восстановления параметр потока отказов (плотность восстановления). Пусть $M(t)$ обозначает среднее число событий отказов-восстановлений некоторой системы в течение времени t. Параметром потока этих событий называется величина $\omega(t)=\dfrac{dM(t)}{dt}$. Если поток ординарный, то параметр потока совпадает с интенсивностью потока. В теории восстановления параметр потока обозначается $h(t)$. Мы будем придерживаться принятого в теории надёжности обозначения $\omega(t)$ [60].

Данный параметр потока при условии мгновенной замены отказавших систем удовлетворяет следующему интегральному уравнению Вольтерра второго рода с разностным ядром:

$$\omega(t)=a(t)+\int_0^t \omega(\tau)a(t-\tau)d\tau\,, \tag{2.1}$$

в котором $a(t)$ – плотность вероятности системы до отказа. Уравнение (2.1) решается в преобразовании Лапласа или численно. Решение в преобразовании Лапласа имеет вид:

$$\omega^{\circ}(s)=\frac{a^{\circ}(s)}{1-a^{\circ}(s)}, \tag{2.2}$$

где $\omega^{\circ}(s)$, $a^{\circ}(s)$ – изображения $\omega(t)$, $a(t)$, а s – переменная Лапласа.

Свойствами параметра являются [60]:

1) $\lim\limits_{t\to\infty}\omega(t)=\dfrac{1}{T},\ T=\int_0^\infty za(z)dz$; 2) $\omega(t)\geq(t)$;

3) $\begin{cases}\lambda(t)\geq\omega(t)\geq a(t) \text{ для стареющих систем,}\\ \omega(t)\geq\lambda(t)\geq a(t) \text{ для молодеющих систем.}\end{cases}$

4) Для систем из N независимых элементов $\Omega(t)=\sum_{i=1}^{N}\omega_i(t)$.

5) При экспоненциальном законе надёжности $\omega(t)=\lambda$.

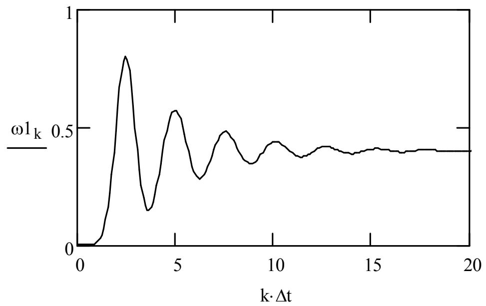
В данных выражениях T – среднее время до отказа, $\lambda(t)$ – интенсивность отказа, $\Omega(t)$ – параметр потока отказов системы, λ – параметр экспоненциального распределения.

Используя метод трапеций можно представить (2.1) для решения в численном виде:

$$\omega(i\Delta t)=\frac{1}{A}a(i\Delta t)+B\sum_{j=1}^{i-1}a((i-j)\Delta t)\omega(j\Delta t),$$

где
$$A=\frac{2-\Delta ta(0)}{2+\Delta ta(0)},\quad B=\frac{2\Delta t}{2-\Delta ta(0)}. \tag{2.3}$$

Пример 2.1. В данном примере представлен расчёт параметра потока при нормальной плотности вероятности времени до отказа некоторого технического изделия. На рис. 2.1 показан график параметра потока отказов в зависимости от времени эксплуатации изделия.



$$\omega_{1_{100}} = 0.435\,; \ \int_0^\infty z a(z)dz = 2.5$$

Рис. 2.1

$$\Delta t := 0.1\,;\ i := 3...1000\,;\ m := 2.5\,;\ \sigma := 0.5\,; A = 1\,; B = 0.1\,;$$

$$a(t) := \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(t-m)^2}{2\sigma^2}}\,;\ A := \frac{2-\Delta t a(0)}{2+\Delta t a(0)}\,;\ B := 2\frac{\Delta t}{2-\Delta t a(0)}\,;\ \omega_{1_0} := a(0)\,;$$

$$\omega_{1_1} := \frac{a(\Delta t)}{A}\,;\ \omega_{1_2} := \frac{a(2\Delta t)}{A} + Ba(\Delta t)\omega_{1_1}\,;\ k := 0...1000\,,$$

$$\omega_{1_i} := \frac{a(i\Delta t)}{A} + B\sum_{j=1}^{i-1}[a[(i-j)\Delta t]\omega_{1_j}]\,.$$

С помощью выражения (2.3) может быть произведен расчёт плотности вероятности $a(t)$ при условии, когда известна функция для параметра потока отказов изделия. Из (2.3) получим следующее выражение для $\omega(t)$:

$$a(i\Delta t) = A\omega(i\Delta t) - AB\sum_{j=1}^{i-1} a((i-j)\Delta t)\omega(j\Delta t)\,. \tag{2.4}$$

Выражение (2.4) может быть применено для определения и показателей надёжности ПС. ▲

Пример 2.2. Пусть в результате испытаний некоторого программного изделия в течение 100 ч получена экспериментальная зависимость параметра его ошибок от времени. Каждая последующая ошибка после её обнаружения была устранена. Экспериментальная кривая параметра была аппроксимирована аналитической функцией $\omega(t)$, которая приведена в примере. На основе (2.4) была вычислена плотность вероятности времени между ошибками ПС $a(t)$.

$$\Delta t := 0.1\,; i := 3...1000\,;\ C_1 := 0.7\,;\ C_2 := 0.3\,;$$

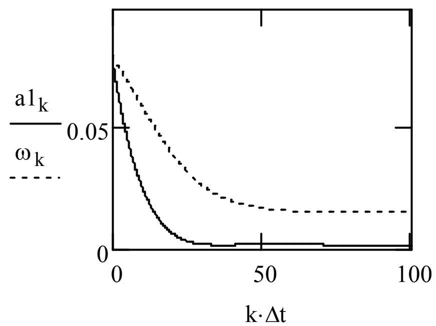
$$\lambda_1 := 0.108\,;\ \lambda_2 := 0.015\,;\ A := \frac{2-\Delta t\omega(0)}{2+\Delta t\omega(0)}\,;\ B := \frac{2\Delta t}{2-\Delta t\omega(0)}\,;$$

$$A = 0.992\,;\ B = 0.1\,;\ \omega(t) := \frac{C_1\lambda_1 e^{-\lambda_1} + C_2\lambda_2 e^{-\lambda_2}}{C_1 e^{-\lambda_1} + C_2 e^{-\lambda_2}}\,;$$

$$a_{1_0} := \omega(0)\,;\ a_{1_1} := A\omega(\Delta t)\,;\ a_{1_2} := A\omega(2\Delta t) - AB\omega(2\Delta t)a_{1_1}\,;$$

$$a_{1_i} := A\omega(i\Delta t) - AB\sum_{j=1}^{i-1}(\omega(j\Delta t)a_{1_{i-j}})\,;\ k := 0...1000\,;\ \omega_k := \omega(k\Delta t)\,.$$

На рис. 2.2 представлены оба показателя: пунктиром – $\omega(t)$, а плотность вероятности $a(t)$ – непрерывной линией.



$$a_{1_{1000}} = 1.376\cdot 10^{-3}$$

Рис.2.2

▲

Можно показать, что свойство 3), справедливое для молодеющих систем, выполняется, а именно, $\omega(t) \geq \lambda(t)$. Здесь $\lambda(t)$ интенсивность проявления ошибки ПС. В данном случае плотность вероятности достаточно близка к экспоненциальной. Другие показатели надёжности ПС могут быть определены по известным для них выражениям. В качестве первого замечания следует указать, что нами применён механизм полного восстановления работоспособности ПС. Сомнение, возникающее в справедливости этого допущения, а именно принимать его или отвергнуть, оставляем для рассмотрения читателю.

В качестве второго замечания укажем на то, что после получения $\omega(t)$ на основе экспериментальных данных об ошибках ПС и выполнения численной процедуры нахождения $a(t)$, необходимо проверить условие нормирования для плотности вероятности. Если оно не выполнено, то это нужно проделать согласно выражению $a(t) = C\int_0^\infty a(z)dz = 1$, из которого нужно найти значение величины C.

2.3. ПАРАМЕТР ПОТОКА ПРИ ОТСУТСТВИИ ОБНОВЛЕНИЯ СИСТЕМЫ

Параметр потока отказов при отсутствии обновления системы после отказа, или параметр потока отказов стареющих систем, впервые был предложен в аналитической форме и подтверждён экспериментально при испытаниях электронных элементов в статье [23].

Аналитическое выражение для параметра потока отказов может быть получено в результате решения следующего интегрального уравнения:

$$\omega(t) = a(t)\left[1 + \int_0^t \frac{\omega(z)}{P(z)}dz\right], \qquad (2.5)$$

где $P(t) = \int_t^\infty a(z)dz$ – вероятность безотказной работы системы в течение времени t. Решением данного уравнения является $\omega(t) = \lambda(t)$, где $\lambda(t)$ – интенсивность отказа системы. Уравнение (2.5) отвечает случайному процессу восстановления с заменой отказавшей системы после её отказа исправной системой, но имеющей такую же наработку, которая наблюдалась у отказав-

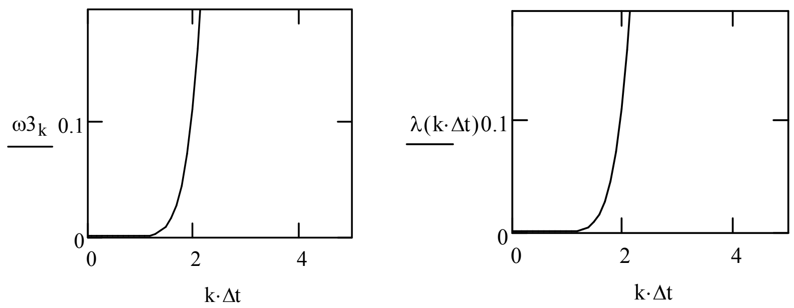
шей системы к моменту её отказа. Иначе говоря, система в процессе её эксплуатации всегда остаётся «старой», но работоспособной. Наблюдаемый поток отказов системы означает, что количество резервных систем для замены бесконечно.

Пример 2.3. Приведены численная реализация уравнения (2.5), расчёт значений параметра потока $\omega(t)$ и интенсивности отказа системы $\lambda(t)$ для нормальной плотности вероятности времени до отказа (рис. 2.3): $\Delta t := 0.1$; $i := 2...1000$; $m := 3$; $\sigma := 0.5$;

$$\lambda(t) := 0.01;\ a(t) := \frac{1}{\sqrt{2\pi}\sigma}e^{\frac{-(t-m)^2}{2\sigma^2}};\ P(t) := \int_t^\infty a(z)dz;\ \lambda(t) := \frac{a(t)}{P(t)};$$

$$\omega_{3_0} := a(0);\ \omega_{3_1} := \frac{a(\Delta t)P(\Delta t)(1 + a(0)\Delta t)}{P(\Delta t) - a(\Delta t)\Delta t};\ k := 0...1000;$$

$$\omega_{3_i} := \frac{a(i\Delta t)\left[1 + \sum_{j=0}^{i-1}\left(\frac{\omega_{3_j}}{P(j\Delta t)}\Delta t\right)\right]P(i\Delta t)}{P(i\Delta t) - a(i\Delta t)\Delta t}.$$



$\omega_{3_3} = 3.715\cdot 10^{-7}$ $\qquad$ $\lambda(3\Delta t) = 3.715\cdot 10^{-7}$

Рис. 2.3

При оценивании показателей надёжности ПС следует иметь в виду, что параметр потока и интенсивность ошибок в данном случае совпадают. Поэтому нет необходимости в решении уравнения (2.5). Однако нужно принимать во внимание тот факт, что здесь замена ПС осуществляется после любой ошибки в ней ПС, имеющей уже наработку в течение времени до наступления ошибки. Поэтому, вероятнее всего, что получаемая оценка надёжности ПС будет оценкой снизу. Для изучения нового потока отказов и его параметра нам потребуются сведения об основном законе надёжности – физическом принципе Н.М. Седякина. Кроме того, основные положения этого принципа будут использоваться и в дальнейшем при изложении некоторых моделей надёжности ПС и человека-оператора.

2.4. ФИЗИЧЕСКИЙ ПРИНЦИП ТЕОРИИ НАДЁЖНОСТИ

Физический принцип теории надёжности был сформулирован профессором Н.М. Седякиным в 1965 г. в его докладе-дискуссии перед учёными академии им. А.Ф. Можайского, а в 1966 г. была опубликована его статья в журнале «Изв. АН СССР. Техническая кибернетика» – «Об одном физическом принципе теории надёжности» [44].

Формулировка принципа: «Надёжность элемента (системы) зависит от величины выработанного им (ей) ресурса в прошлом и не зависит от того, как выработан этот ресурс». В математической форме принцип представлен в виде:

$$P(t/r) = P^{(1)}(t/x_1) = P^{(2)}(t/x_2),$$
$$r = \int_0^{x_1} \lambda(z,\varepsilon_1)dz = \int_0^{x_2} \lambda(z,\varepsilon_2)dz, \qquad (2.6)$$

где x_1 и x_2 удовлетворяют интегральному уравнению – второму выражению в (2.6). Величина $r(t) = \int_0^t \lambda(z,\varepsilon)dz$ означает ресурс, выработанный системой в течение времени t в условиях ε. Второе выражение данной системы уравнений означает, что две системы работали в различных условиях ε_1 и ε_2 в течение времени x_1 и x_2 таким образом, что выработали одинаковый ресурс r. Первое уравнение (2.6) означает, что если в будущем обе системы будут работать в одинаковых условиях ε в течение времени t, то условные вероятности их безотказной работы будут одинаковыми. Это при условии, что они раньше выработали одинаковый ресурс в различных условиях за различное время.

Следствие принципа. Если условия работы системы заданы в виде:

$$\varepsilon = \begin{cases} \varepsilon_1 & \text{при } 0 < z \le t_1, \\ \varepsilon_2 & \text{при } t_1 < z \le t_1 + t_2, \end{cases} \qquad (2.7)$$

то интенсивность её отказа в интервале $(t_1, t_1 + t_2)$ удовлетворяет выражению:

$$\lambda(z, x_2, t_1) = \lambda(z + x_2 - t_1, \varepsilon_2), \qquad (2.8)$$

где $x_2 = x_2(t_1)$ находится в результате решения относительно x_2 уравнения:

$$\int_0^{x_2} \lambda(z,\varepsilon_2)dz = \int_0^{t_1} \lambda(z,\varepsilon_1)dz. \qquad (2.9)$$

За доказательством следствия мы отсылаем к указанной работе.

На основании принципа надёжности и его следствия могут быть решены задачи достаточно большой общности, которые раньше принципиально решить было невозможно. В работе автора книги [51] данный принцип был теоретически обобщён. В обобщении рассмотрен принцип Н.М. Седякина не в абсолютном, а в относительном его понимании. Сущность этого обобщения сводится к принятию во внимание производных от величины ресурса (чувствительности) выработанного системой к величине нагрузки на систему. Физически это означает, что при значительной нагрузке на систему по сравнению с малой нагрузкой может наблюдаться потеря части ресурса, – дефект ресурса, – которая должна быть учтена в математической модели. Для практики расчётов надёжности это сводится к рекомендации учёта не только величин интенсивностей отказов изделий, но и чувствительности интенсивностей в виде дополнительных их приращений. В аналитических моделях учёт чувствительности приводит к коррекции величины ресурса системы и более точной оценке её надёжности. Но этим обобщением в данной работе пользоваться не будем.

Воспользуемся рассмотренным физическим принципом для рассмотрения потока событий с частичным восстановлением систем после их отказа.

2.5. ОБОБЩЁННЫЙ ПАРАМЕТР ПОТОКА

Рассмотрим функционирование ТС в режиме ε, которая после отказа заменяется работоспособной системой, находящейся в режиме ε_p. Запишем выражение для условной вероятности безотказной работы восстановленной (заменённой) системы в течение времени $t-\tau$ в режиме ε при условии, что до замены система находилась в режиме ε_p в течение времени τ и не отказала. Получим:

$$P_в(t-\tau,\varepsilon|\tau,\varepsilon_p) = P(t-\tau+x(\tau),\varepsilon)/P(x(\tau),\varepsilon). \qquad (2.10)$$

В данном выражении $x(\tau)$ означает время работы системы в режиме ε эквивалентное времени её работы τ в режиме ε_p при условии, что израсходованные ими ресурсы надёжности в этих режимах одинаковы. Следует отметить, что данное выражение (2.10) справедливо только в том случае, если надёжность заменяющих систем после постановки их в режим ε не будет зависеть от того, каким образом был израсходован их ресурс, а зависит лишь от величины израсходованного ресурса в прошлом. Это соответствует принципу Н.М. Седякина.

Найдём условную плотность вероятности времени работы системы до отказа, продифференцировав по переменной t выражение (2.10):

$$a_d(t-\tau,\varepsilon|\tau,\varepsilon_p) = a(t-\tau+x(\tau),\varepsilon)/P(x(\tau),\varepsilon). \qquad (2.11)$$

Тогда можно показать, что для параметра потока отказов системы справедливо выражение:

$$\omega(t) = a(t) + \int_0^t \frac{a(t-\tau+x(\tau),\varepsilon)}{P(x(\tau),\varepsilon)}\omega(\tau)d\tau. \qquad (2.12)$$

Рассмотрим частные случаи, следующие из (2.12).

а) Замена отказавших систем производится новыми системами. В этом случае $x(\tau)=0$, $P(x(\tau))=1$. Поэтому из (2.12) следует интегральное уравнение $\omega(t)=a(t)+\int_0^t a(t-\tau)\omega(\tau)d\tau$, рассмотренное в разделе 2.2.

б) Замена отказавших систем производится старыми системами, выработавшими такой же ресурс надёжности, что и основные системы, но работоспособными. В этом случае $x(\tau)=\tau$, $P(x(\tau))=P(\tau)$. Поэтому из (2.12) следует интегральное уравнение $\omega(t)=a(t)\left[1+\int_0^t \frac{\omega(\tau)}{P(\tau)}d\tau\right]$, полученное в разделе 2.3.

<u>Пример 2.4</u>. $\Delta t := 0.1$; $i := 2...300$; $\lambda_a := 0.04$; $\lambda_b := 0.01$;

$$k_a := 4;\ k_b := 0{,}2;\ a(x) := \lambda_a k_a x^{k_a-1}e^{-\lambda_a x^{k_a}};\ b(t) := \lambda_b k_b t^{k_b-1}e^{-\lambda_b t^{k_b}};$$

$$x(t) := \sqrt[k_a]{\frac{\lambda_b}{\lambda_a}t^{k_b}};\ P(x) := \int_x^{1000} a(z)dz;$$

$$\lambda(5) = 1.47\cdot 10^{12};\ \omega_{2_{250}} = 0.814;\ T_a := \int_0^\infty P(z)dz;\ T_b := \left(\frac{1}{\lambda_b}\right)^{\frac{1}{k_b}}\Gamma\left(1+\frac{1}{k_b}\right);$$

$$k := 0...300;\ t := 0.01...300;\ \omega_{2_0} := a(0);\ \omega_{2_1} := \frac{a(\Delta t)(2+a(0)\Delta t)}{2-\lambda(\Delta t)\Delta t};$$

$$x(t) := 0.01...300;\ \omega_{2_2} := \frac{a(2\Delta t)(2+a(0)\Delta t)+\dfrac{a(\Delta t+x(\Delta t))\omega_{2_1}\Delta t}{P(x(\Delta t))}}{2-\lambda(2\Delta t)\Delta t};$$

$$\omega_{2_i} := \frac{a(i\Delta t)(2+a(0)\Delta t)+2\sum\limits_{j=1}^{i-1}\dfrac{a[(i-j)\Delta t+x(j\Delta t)]\omega_{2_j}\Delta t}{P(x(j\Delta t))}}{2-\lambda(x(i\Delta t))\Delta t}.$$

▲

В данном примере приведён расчёт графиков для параметра потока и соответствующих ему плотностей вероятностей основной и резервных систем (рис.2.4). Использованы оба распределения Вейбулла. Это объясняется тем, что рассматривается такой случай, когда интегральное уравнение $\int_0^{x(t)} a(z)dz = \int_0^t b(z)dz$ можно решить аналитически, не усложняя программу вычислений функции $x(t)$. Если бы один из законов был усечённо-нормальным, то пришлось бы усложнить программу вычислений функции $x(t)$.

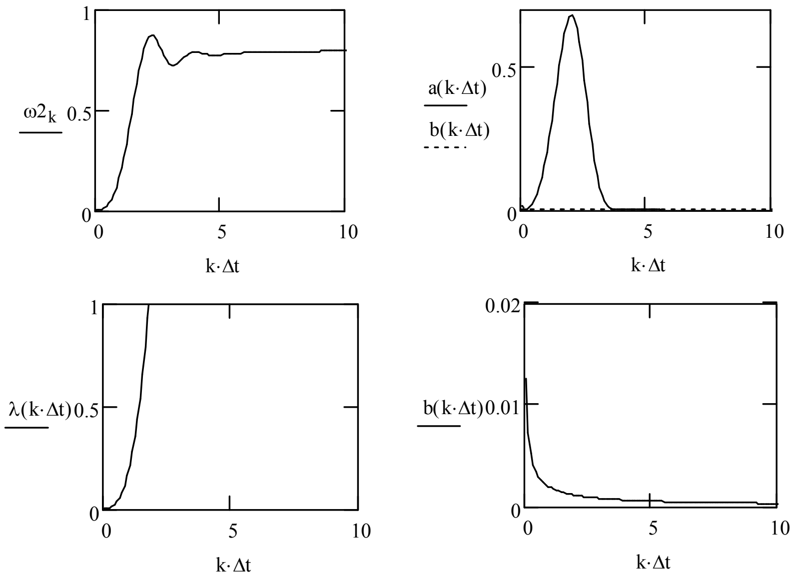


Рис. 2.4

Из примера видно, что учёт плотности $b(t)$ в ядре интегрального уравнения приводит к совершенно иному поведению параметра потока отказов. При мгновенной замене отказавшей системы такой же новой системой график параметра потока был бы более монотонным и стремился бы к пределу $\omega(\infty) = 1/2{,}019 \approx 0{,}49$, а при данном законе $b(t)$ он стремится к $\approx 0{,}814$. Изменяя параметры закона $b(t)$, можно убедиться о других значительных изменениях графика параметра потока. В теории надёжности технических систем применение данной характеристики вполне очевидно [39, 49].

Можно ли применять характеристику данного потока с частичным восстановлением для исследования надёжности ПС? На наш взгляд, ответ на данный вопрос положителен. Параметр потока можно использовать для молодеющих процессов, когда требуется подбирать его значение под конкретные экспериментальные данные об ошибках ПС. При этом подбор может производиться более гибко, чем при управлении одним распределением

вероятностей. Хотя для решения этой задачи могут потребоваться определённая смекалка и достаточное терпение исследователя.

2.6. ПАРАМЕТР ПОТОКА С УЧЁТОМ ПРОФИЛАКТИКИ

Наряду с рассмотренными потоками занимает в исследованиях надёжности технических систем поток с другим способом вмешательства в поведение системы – поток с её периодическим профилактическим обслуживанием для поддержания работоспособности системы в течение продолжительного времени [50]. Приведём основные выражения для параметра такого потока и кратко обсудим его свойства.

Этот вид потока и его параметр рекомендуется рассматривать при совместном оценивании готовности и надёжности ТС, имеющих в своём составе ПС.

Приведём вывод интегрального уравнения для параметра потока отказов ТС с учётом периодического проведения на нём профилактических мероприятий. Условимся, что после ремонта и проведения профилактики ТС полностью восстанавливает свои свойства работоспособности. В принципе, в дальнейшем можно принять во внимание и учесть неполноту восстановления ТС на основе физического принципа надёжности. Но это сильно усложнит математическую модель параметра и возможность его расчёта. Поэтому ограничимся указанным допущением о полноте восстановления ТС.

Примем следующие обозначения: $\omega(t)$ – параметр потока, $a(t)$ – плотность вероятности времени до отказа, $Q(t)$ – вероятность отказа, $U(t)$ – функция распределения времени начала проведения профилактики, $v(t)$ – плотность вероятности длительности профилактики, $g(t)$ – плотность вероятности времени ремонта ТС.

Параметр потока будет равен сумме трёх составляющих, соответствующих событиям:

– произошёл ровно один отказ ТС за время t при условии, что профилактика на нём не была назначена за это время;

– произошло несколько отказов ТС за время t при условии, что первый отказ наступил до момента назначения первой профилактики;

– произошло несколько отказов ТС за время t при условии, что первая профилактика была назначена до момента возникновения первого отказа ТС.

Тогда параметр потока примет вид:

$$\omega(t) = a(t)[1-U(t)] + \int_0^t [1-U(z)]a(z)\omega(t-z)dz + \\ + \int_0^t [1-Q(z)] \int_0^{t-z} v(\theta)\omega(t-z-\theta)d\theta dU(z). \qquad (2.13)$$

Выражение (2.13) получено при условии, что ТС после отказа заменяется работоспособным мгновенно.

Изображение Лапласа (2.13) равно:

$$\omega^\circ(s) = \frac{a^\circ(s)}{1-a^\circ(s)-b^\circ(s)v^\circ(s)}, \quad (2.14)$$

где $a^\circ(s) = \int_0^\infty a(z)[1-U(z)]e^{-sz}dz;\ b^\circ(s) = \int_0^\infty [1-Q(z)]e^{-sz}dU(z)$.

При условии регулярной на практике профилактике имеет смысл рассматривать в качестве $U(t)$ вырожденное распределение, то есть

$$U(t) = \begin{cases} 0, t < T, \\ 1, t \ge T, \end{cases} \qquad (2.15)$$

где T – период между соседними профилактиками.

Из выражения (2.14) при условии длительной эксплуатации ТС получим установившееся значение параметра потока:

$$\omega(\infty, T) = \frac{Q(T)}{\int_0^T P(z)dz + t_{\text{пр}}P(T)}, \qquad (2.16)$$

в котором $t_{\text{пр}}$ – средняя продолжительность одной профилактики, а $P(t) = 1 - Q(t)$.

Рассуждая аналогично, получим $\omega(t), \omega^\circ(s), \omega(\infty, T)$ для случая, когда восстановление ТС после отказа производится не мгновенно, а через случайное время:

$$\omega(t) = a(t)[1-U(t)] + \int_0^t [1-U(z)] \int_0^{t-z} g(\theta)\omega(t-z-\theta)a(z)dz + \\ + \int_0^t [1-Q(z)] \int_0^{t-z} v(\theta)\omega(t-z-\theta)d\theta dU(z);$$

$$\omega^\circ(s) = \frac{a^\circ(s)}{1-a^\circ(s)g^\circ(s)}; \quad \omega(\infty, T) = \frac{Q(T)}{\int_0^T P(z)dz + t_{\text{в}}Q(T) + t_{\text{пр}}P(T)}, \qquad (2.17)$$

где $t_{\text{в}}$ – среднее время восстановления ТС.

Из (2.16), (2.17) при условии, что профилактика не проводится $(T \to \infty)$, следуют известные частные случаи стационарных значений параметра потока:

$$\omega(\infty,\infty) = 1/T_0,\ \omega(\infty,\infty) = 1/(T_0 + t_{\text{в}}),$$

где T_0 – среднее время безотказной работы ТС.

<u>Пример 2.5</u>. Требуется определить оптимальную по коэффициенту готовности продолжительность межпрофилактического периода T_{opt} и среднее число ремонтов n_p ТС за один год, если законом распределения времени до отказа является закон Вейбулла со значениями параметров $\lambda_0 = 1\cdot 10^{-5}$ 1/ч^k, $k = 2{,}5$. Среднее время ремонта ТС после отказа $t_{\text{в}} = 10$ ч, средняя продолжительность профилактики $t_{\text{пр}} = 2$ ч.

Согласно (2.17) для коэффициента готовности ТС найдём следующее выражение

$$K_{\text{г}} = \frac{\int_0^T P(z)dz}{\int_0^T P(z)dz + t_{\text{в}}Q(T) + t_{\text{пр}}P(T)}, \qquad (2.18)$$

где $P(t) = e^{-\lambda_0 t^k}$, $Q(t) = 1 - P(t)$. Величина T_{opt}, максимизирующая (2.18), удовлетворяет уравнению:

$$\frac{t_{\text{в}}}{t_{\text{в}} - t_{\text{пр}}} = \lambda(T_{opt}) \int_0^{T_{opt}} P(z)dz + P(T_{opt}), \qquad (2.19)$$

в котором $\lambda(t)$ – интенсивность отказа ТС.

Результаты расчётов по формулам (2.17), (2.18) приведены на рис. 2.5.

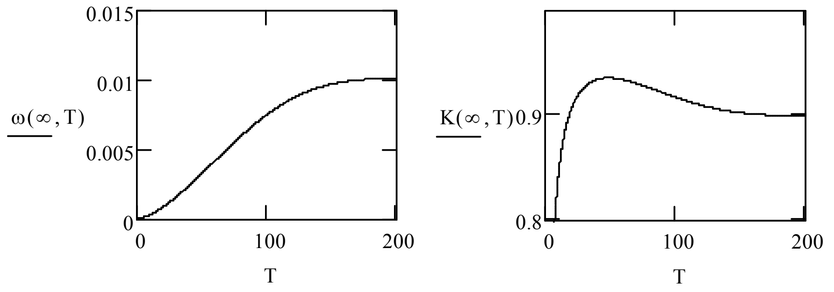


Рис. 2.5

Максимальная величина $K_г = 0{,}95$ достигается при $T_{opt} \approx 50$ ч. Межпрофилактическому периоду $T_{opt} \approx 50$ ч. Соответствует величина параметра потока отказов $\omega(\infty, T_{opt}) \approx$ $\approx 0{,}0035$ ч$^{-1}$. Среднее время безотказной работы ТС без проведения профилактик $T_0 \approx 89$ ч, а с их проведением $T_{0,\,пр} \approx 285$ ч. Среднее ожидаемое число ремонтов ТС в течение года без проведения профилактик $n_p \approx 100$, а с их проведением $n_p \approx 31$. Суммарная полезная наработка ТС за год в среднем увеличивается на полмесяца.

При расчёте показателей надёжности сложных технических систем с программным обеспечением и наличии в контуре управления людей-операторов этот параметр потока может быть успешно использован наряду с математическими моделями указанных двух составляющих системы.

2.7. ПАРАМЕТР ПОТОКА ПРИ НЕСИММЕТРИЧНЫХ СОВМЕСТНЫХ ЗАМЕНАХ СОСТАВНЫХ ЧАСТЕЙ

Этот поток и его параметр рассмотрены В.Ю. Мордвиновым в его статье [36]. Им вводятся понятия *устранимого* и *неустранимого* отказов ТС. В качестве примера рассматривается автомобильная покрышка. В случае порыва она может заменяться на

новую. В случае критичного изнашивания (ведомый отказ) покрышка может навариваться и процесс её эксплуатации продолжается.

При совместных заменах характеристика самостоятельного элемента в паре не меняется, а характеристика ведомого элемента модифицируется за счёт снижения его возраста до нуля при профилактических заменах совместно с самостоятельным элементом. Поэтому значение функции восстановления (среднего числа замен в интервале времени t будет определяться как:

$$M_C(t) = M_\tau^*(t) + M_\eta(t)\,. \qquad (2.20)$$

В данной формуле $M_C(t)$ – суммарное число замен-восстановлений, $M_\tau^*(t)$ – модифицированная функция восстановления, $M_\eta(t)$ – функция восстановления самостоятельного элемента.

Предположим, что наработка до первого отказа самостоятельного элемента $\eta = x$. Тогда если:

$$\begin{aligned} x > t \quad & M_\tau^*(t) = M_\tau(t), \\ x \le t \quad & M_\tau^*(t) = M_\tau(t) + M_\tau^*(t) \end{aligned}\,. \qquad (2.21)$$

Тогда уравнение для функции восстановления примет вид:

$$M_\tau^*(t) = M_\tau(t)P(t) + \int_0^t [M_\tau^*(t-x) + M_\tau(x)]a(x)dx\,, \qquad (2.22)$$

где $P(t)$, $a(t)$ – вероятность безотказной работы и плотность вероятности времени до отказа самостоятельно элемента. Функция $M_\tau^*(t)$ учитывает отказы ведомого элемента при его собственных отказах. Дифференцируя по t (2.22), получим:

$$\omega_\tau^*(t) = \omega_\tau(t)P(t) + \int_0^t \omega_\tau^*(t-x)a(x)dx\,, \qquad (2.23)$$

так как из (2.22) следует $M_\tau^*(0) = M_\tau(0) = 0$. $M_\tau^*(t)$ учитывает отказы ведомого элемента при его собственных отказах.

Формулу (2.23) можно представить в виде изображения Лапласа:

$$\omega_\tau^{*\circ}(s) = \frac{L\{\omega_\tau(t)P(t)\}}{1 - a_\eta^\circ(s)}\,. \qquad (2.24)$$

В (2.24) $L, \circ, s$ обозначают оператор, изображение и переменную Лапласа. Рассмотрим три частных случая для параметра потока $\omega^*_\tau(t)$.

1. Экспоненциальный закон распределения наработки ведомого элемента. Имеем: $a_\tau(t) = \nu\exp(-\nu t)$. Тогда из (2.24) следует:

$$\omega^{*\circ}_\tau(s) = \nu[1/s - a^\circ_\eta(s)] + \omega^{*\circ}_\tau a^\circ_\eta(s) \to \omega^{*\circ}_\tau(s) = \frac{\nu}{s};$$

$$\omega^*_\tau(t) = \omega_\tau(t) = \nu.$$

Таким образом, при любом законе распределения системного элемента совместные замены не оказывают никакого влияния на модифицированный параметр потока отказов при постоянном параметре потока ведомого элемента.

2. Экспоненциальный закон распределения наработки системного элемента. Из (2.24) при значении параметра этого закона λ следует, что

$$\omega^{*\circ}_\tau(s) = (s+\lambda)s^{-1}\omega^\circ_\tau(s+\lambda) \text{ и } \omega^*_\tau(\infty) = \lambda\omega^\circ_\tau(\lambda). \qquad (2.25)$$

Из (2.25) можно получить, что

$$\omega^*_\tau(t) = e^{-\lambda t}\omega_\tau(t) + \lambda\int_0^t e^{-\lambda z}\omega_\tau(z)dz = \omega_\tau(0) + \int_0^t e^{-\lambda z}d\omega_\tau(z). \qquad (2.26)$$

При $\lambda \to 0$ $\omega^*_\tau(t) \to \omega_\tau(t)$. При этом, если производная от $\omega_\tau(t)$ не возрастает быстрее экспоненты, то при $\lambda \to \infty$ $\omega^*_\tau(t) \to 0$, то есть в пределе, теоретически, можно не учитывать отказов ведомого элемента. Из (2.26) следует:

$$[\omega^*_\tau(t)]' = e^{-\lambda t}[\omega_\tau(t)]'. \qquad (2.27)$$

Из этого равенства можно заключить, что параметр потока достигает установившегося значения медленнее, чем параметр модифицированного потока.

3. Экспоненциальные законы распределения наработок системного и ведомого элементов. При данном условии выполняется 1. Поэтому $\omega^*_\tau(t) = \nu$, то есть параметр модифицированного потока равен параметру экспоненциального распределения ведомого элемента.

<u>Пример 2.6</u>. $\Delta t := 0.1$; $i := 3...2000$; $m := 2.5$; $\sigma := 0.5$;

$$\nu := 0.2;\ \sigma_\nu := 0.05;\ m_\nu := 1.5;\ a(t) :+ \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(t-m)^2}{2\sigma^2}};$$

$$b(t) := \nu c(t)\int_t^\infty a(z)dz;\ c(t) := \int_0^t \sigma_\nu m_\nu z^{m_\nu - 1} e^{-\sigma_\nu z^{m_\nu}} dz;\ A := \frac{2}{2 + \Delta t a(0)};$$

$$B := 2\frac{\Delta t}{2 - \Delta t a(0)};\ A = 1;\ B = 0.1;\ \omega_{1_0} := b(0);$$

$$\omega_{1_1} := \frac{b(\Delta t)}{A} + B\omega_{1_0}a(0);\ \omega_{1_2} := \frac{b(\Delta t)}{A} + B(a(\Delta t)\omega_{1_1} + \omega_{1_0}a(2\Delta t));$$

$$\omega_{1_i} := \frac{b(i\Delta t)}{A} + B\sum_{j=0}^{i-1}[a[(i-j)\Delta t]\omega_{1_j}];\ k := 0...2000. \qquad \blacktriangle$$

В данном примере представлен расчёт параметра потока при несимметричных заменах. Первое слагаемое правой части уравнения (2.23) представлено функцией $b(t)$, в которой параметр немодифицированного потока $\omega_\tau(t) = \nu c(t)$, а $c(t)$ представлена в виде функции распределения Вейбулла. На рис. 2.6 показан график параметра модифицированного потока при указанных значениях констант. Установившееся значение параметра равно $\omega_{1_{200}} = 0{,}016$. На рис. 2.7 показан график параметра потока $\omega(t)$, когда ведомый элемент отсутствует. Этот график получен при условии, когда $b(t) = a(t)$, что соответствует параметру потока, рассмотренному в 2.2. Установившееся значение параметра для этого потока равно $\omega_{1_{200}} = 0{,}4$.

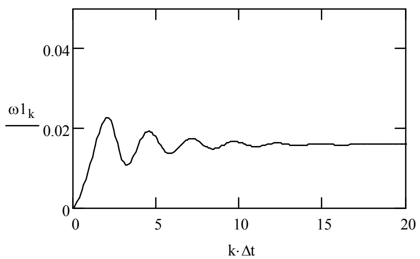


Рис. 2.6

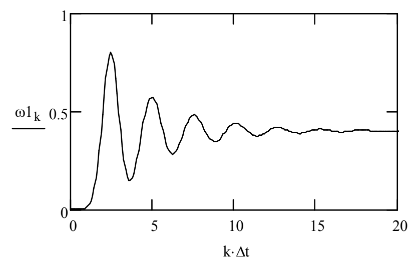


Рис. 2.7

Параметр рассмотренного потока при несимметричных заменах составных частей может успешно использоваться в исследовании надёжности ТС, ПС, а также в системах, включающих все три компонента – технические, программные средства и человека-оператора.

2.8. ЗАМЕЧАНИЯ ОБ УЧЁТЕ ВРЕМЕНИ ВОССТАНОВЛЕНИЯ В ПОТОКАХ И РАСЧЁТЕ ГОТОВНОСТИ СИСТЕМ

В данной главе были рассмотрены модели и параметры различных потоков событий при условии мгновенного возобновления этих событий после отказов или ошибок. В случае необходимости в принятии во внимание конечного, а не мгновенного восстановления в моделях, параметры потоков могут быть скорректированы изменением приведённых интегральных уравнений. Эти новые уравнения будут обладать большей сложностью и расчёт величин параметров потоков также будет более сложным.

При принятии во внимание немгновенного восстановления могут быть получены и такие характеристики систем, как коэффициент готовности, функция готовности и функция (коэффициент) оперативной готовности. Пример подобного расчёта коэффициента готовности был приведён в 2.6. Однако, для всех типов моделей потоков, кроме моделей 2.2 и 2.6, получение указанных характеристик является достаточно трудоёмким процессом.

И, наконец, наибольшую трудность представит использование в моделях физического принципа надёжности Н.М. Седякина, когда необходимо будет учитывать неполноту восстановления работоспособности систем после устранения в них отказов или ошибок. Хотя принципиально это вполне возможно.

К указанным замечаниям следует также добавить, что вычислительные возможности даже современных быстродействующих вычислительных средств для решения отмеченных задач окажутся весьма ограниченными.

Глава 3

МОДЕЛИ ПРИБЛИЖЁННОГО ПРЕДСТАВЛЕНИЯ ЗАКОНОВ РАСПРЕДЕЛЕНИЙ

3.1. ГИПЕРЭКСПОНЕНЦИАЛЬНАЯ АППРОКСИМАЦИЯ

В прикладных задачах теории надёжности, относящихся к самым различным системным исследованиям, часто приходится иметь дело с неоднородными немарковскими процессами с конечным или счётным числом состояний. Экспоненциальные распределения, являясь достаточно грубым представлением реальной действительности, иногда не могут удовлетворить исследователя в силу их простоты.

Описание систем немарковскими случайными процессами требует привлечения более сложного математического аппарата. Известен ряд аналитических методов исследования систем немарковского типа. Одним из эффективных методов, например, является метод «комплексных вероятностей» Кокса [73]. Он предполагает использование фиктивных этапов случайного процесса, представляемых комплексными величинами. При этом вероятности состояний остаются реальными. В книге [6] отмечается, что использование метода «комплексных вероятностей» позволяет исследовать объект, когда плотность вероятности может быть представлена функцией вида

$$p(x) = \sum_i C_i x^{r_i} e^{-\lambda_i x} \qquad (3.1)$$

если только она неотрицательна при $x > 0$ и имеет интеграл, равный единице. Указывается, что любые законы распределения могут быть аппроксимированы суммой экспонент с полиномиальными множителями (3.1). Однако из-за вычислительных трудностей идея метода «комплексных вероятностей» не получила должного внимания в анализе случайных процессов. Здесь мы представим одну из попыток развития упомянутого метода. Для этого представим произвольную, гладкую плотность вероятности суммой экспоненциальных плотностей вероятностей с комплексными параметрами и коэффициентами (метод параллельных фаз) или с комплексными параметрами (метод последовательных фаз) и кратко отметим вычислительные особенности этих представлений.

Метод параллельных фаз. Подобное представление в виде параллельных фаз следует из аппроксимации вещественных функций в классе ортогональных экспоненциальных функций с комплексными параметрами и комплексными постоянными коэффициентами [48]. При этом, когда число членов разложения фиксировано, в качестве критерия наилучшего приближения берётся минимум интеграла от квадрата ошибки, то есть минимизируется величина $\int_0^\infty [a(t)-\overline{a}(t)]^2\,dt$, где $a(t)$, $\overline{a}(t)$ – точное и приближённое выражение плотности распределения. В простейшем случае аппроксимации одной парой экспонент получаем:

$$\overline{a}(t)=(A+jB)(\alpha+j\beta)e^{-(\alpha+j\beta)t}+(A-jB)(\alpha-j\beta)e^{-(\alpha-j\beta)t}. \quad (3.2)$$

Для определения неизвестных A, B, α, β по методу моментов приравняем соответствующие начальные моменты обеих плотностей. Получим систему из четырёх уравнений:

$$\frac{C}{\gamma^i}+\frac{\overline{C}}{\overline{\gamma}^i}=\tilde{\nu}_i,\ i=0,1,2,3,\ C=A+jB, \quad (3.3)$$

$$\overline{C}=A-jB,\ \gamma=\alpha+j\beta,\ \overline{\gamma}=\alpha-j\beta,\ \tilde{\nu}_i=\nu_i/i!,$$

где ν_i – i-й начальный момент плотности распределения $a(t)$. Решив полученную систему уравнений, будем иметь:

$$A=1/2,\ \alpha=(\tilde{\nu}_3-\tilde{\nu}_1\tilde{\nu}_2)/2(\tilde{\nu}_1\tilde{\nu}_3-\tilde{\nu}_2{}^2), \quad (3.4)$$

$$\beta=(6\tilde{\nu}_1\tilde{\nu}_2\tilde{\nu}_3+3\tilde{\nu}_1{}^2\tilde{\nu}_2{}^2-4\tilde{\nu}_1{}^3\tilde{\nu}_3-4\tilde{\nu}_2{}^3-\tilde{\nu}_3{}^2)^{1/2}/(2(\tilde{\nu}_2{}^2-\tilde{\nu}_1\tilde{\nu}_3)),$$

$$B=(\tilde{\nu}_3-3\tilde{\nu}_1\tilde{\nu}_2+2\tilde{\nu}_1{}^3)/(2(6\tilde{\nu}_1\tilde{\nu}_2\tilde{\nu}_3+3\tilde{\nu}^2{}_1\tilde{\nu}_2{}^2-4\tilde{\nu}_1{}^3\tilde{\nu}_3-4\tilde{\nu}_2{}^3-\tilde{\nu}_2{}^3)^{1/2}).$$

Выражения для величин (3.4), полученные для некоторых распределений, приведены в табл. 3.1.

Таблица 3.1а

Характеристики		Распределение		
		Рэлея	Гамма	Вейбулла
Плотность распределения		$a(t)=(t/\sigma^2)\times$ $\times\exp(-t^2/2\sigma^2)$, $0\le t<\infty$	$a(t)=\dfrac{(\mu(\mu t)^{k-1}}{(k-1)!\exp(-\mu t)}$, $0\le t<\infty, k\ge 2$	$a(t)=k\lambda_0 t^{k-1}\times$ $\times\exp(-\lambda_0 t^k)$, $0\le t<\infty,\ k\ge 2$
$\overline{a}(t)$	α	$\sqrt{\pi}/\sqrt{2}\sigma(4-\pi)$	$2\mu/k$	$\lambda_0{}^{1/k}(\Gamma(3/k+1)-$ $-3\Gamma(1/k+1)\Gamma(2/k+$ $+1))/(2\Gamma(1/k+1)\times$ $\times\Gamma(3/k+1)-3\Gamma^2(2/k+1))$
	β	$\dfrac{\sqrt{23\pi-32-4\pi^2}}{\sqrt{2}\sigma(4-\pi)}$	$\dfrac{\mu\sqrt{2(k-2)/(k+1)}}{k}$	$\lambda_0{}^{1/k}R^*/(3\Gamma^2(2/k+1)-$ $-2\Gamma(1/k+1)\Gamma(3/k+1))$
	B	$\dfrac{\sqrt{\pi}(2\pi-5)}{2\sqrt{23\pi-32-4\pi^2}}$	$\dfrac{(2k-1)}{\sqrt{2(k+1)(k-2)}}$	$(\Gamma(3/k+1)-9\Gamma(1/k+1)\times$ $\times\Gamma(2/k+1)+$ $+12\Gamma^3(1/k+1))/2R^*$

Таблица 3.1б

Характеристики		Распределение	
		Равномерное	Нормальное
Плотность распределения		$a(t)=\begin{cases}1/a,\ 0\le t<a,\\ 0,\ t<a, t>a\end{cases}$	$a(t)=(\sqrt{2\pi}\sigma)^{-1}\times$ $\times\exp(-(t-a)^2/2\sigma^2)$, $-\infty<t<\infty,\ 0\le\sigma/a\le\sqrt{2/3}$
$\overline{a}(t)$	α	$3/a$	$2a^3(a^4+3\sigma^4)$
	β	$\sqrt{3}/a$	$\dfrac{\sqrt{2(a^6-3a^4\sigma^2+9a^2\sigma^4-9\sigma^6)}}{(a^4+3\sigma^4)}$
	B	$\sqrt{3}/2$	$\dfrac{a(2a^2-3\sigma^2)}{\sqrt{2(a^6-3a^4\sigma^2+9a^2\sigma^4-9\sigma^6)}}$

$\overline{a}(t)=(A+jB)(\alpha+j\beta)\exp(-(\alpha+j\beta)t)+(A-jB)(\alpha-j\beta)\exp(-(\alpha-j\beta)t)$

Пример 3.1. Гамма-распределение с параметрами $k := 0.4$;

$$\mu := 0.01;\ f(t) := \frac{\mu(\mu_t)^{k-1}}{\Gamma(k)} e^{-\mu_t};\ F(t) := \int_t^{\infty} f(z)dz;\ g(t) := \frac{f(t)}{F(t)};$$

$$\nu_1 := 40.001;\ \nu_2 := 5.6 \cdot 10^3;\ \nu_3 := 1.344 \cdot 10^6.$$

Given $B := 3$; $\alpha := 0.011$; $\beta := 0.015$;

$$\frac{0.5+iB}{\alpha+iB} + \frac{0.5-iB}{\alpha-iB} - \nu_1 = 0;\ \frac{0.5+iB}{(\alpha+iB)^2} + \frac{0.5-iB}{(\alpha-iB)^2} - \frac{\nu_2}{2} = 0;$$

$$\frac{0.5+iB}{(\alpha+iB)^3} + \frac{0.5-iB}{(\alpha-iB)^3} - \frac{\nu_3}{6} = 0;$$

$$Find(\alpha, \beta, B) \to \begin{bmatrix} .49988752249550089982e-1 \\ (-.37785346869987706773e)-1i \\ (-.94507066402572583686e)-1i \end{bmatrix}$$

$$\alpha := 0.05;\ \beta := -0.03781i;\ B := -0{,}0945i;$$

$$P(t) := (0.5+iB)e^{-(\alpha+i\beta)t} + (0.5-iB)e^{-(\alpha-i\beta)t};\ \lambda(t) := a(t)/P(t);$$

$$a(t) := (0.5+iB)(\alpha+i\beta)e^{-(\alpha+i\beta)t} + (0.5-iB)(\alpha-i\beta)e^{-(\alpha-i\beta)t};$$

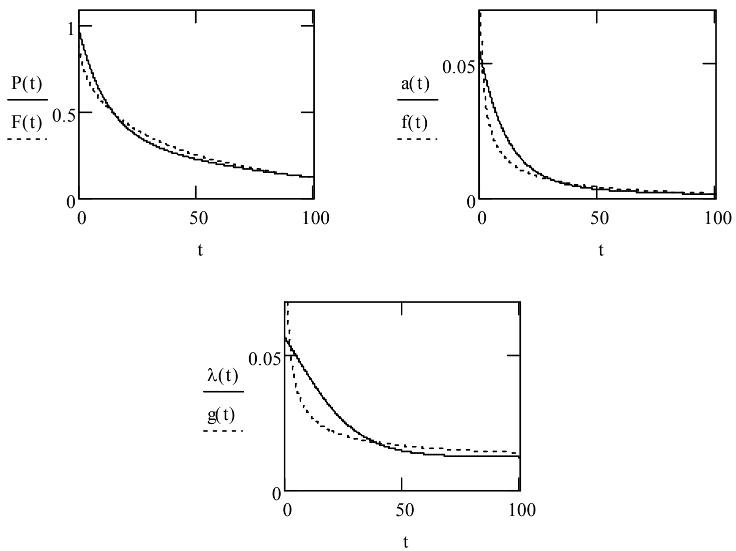


Рис. 3.1

$$a(0) = 0.057;\ \lambda(0) = 0.057;\ \Delta := \lambda(20) - g(20);\ \Delta := 8.618 \cdot 10^{-3};$$

$$\delta\% := \frac{\Delta \cdot 100}{\lambda(20)};\ \delta\% := 28\%;\ C_1 := 0.595;\ C_2 := 0.405;$$

$$a(t) := C_1 \lambda_1 e^{-\lambda_1 t};\ \lambda_1 := 0.088;\ \lambda_2 := 0.012;$$

$$a(t) := 0.052e^{-0.088t} + 0.0048e^{-0.012t};\ P(t) := 0.595e^{-0.088t} + 0.405e^{-0.012t}.$$

▲

Метод последовательных фаз. В простейшем случае аппроксимации плотности распределения с помощью двух последовательных экспоненциальных фаз с комплексно-сопряжёнными параметрами имеем:

$$\overline{a}(t) = \int_0^t (\alpha + j\beta)e^{-(\alpha+j\beta)\tau}(\alpha - j\beta)e^{-(\alpha-j\beta)(t-\tau)}d\tau = \\ = ((\alpha^2+\beta^2)/\beta)e^{-\alpha t}\sin\beta t. \tag{3.5}$$

Приравняв соответствующие два начальных момента плотностей распределений $a(t)$, $\overline{a}(t)$ и определив неизвестные α и β, получим:

$$\alpha = \nu_1/(2\nu_1^2 - \nu_2),\ \beta = (3\nu_1^2 - 2\nu_2)^{1/2}(2\nu_1^2 - \nu_2). \tag{3.6}$$

Значения искомых величин для некоторых распределений приведены в табл. 3.2.

Для приближённого представления показателей надёжности по методу параллельных фаз можно использовать следующие готовые выражения:

$$P(t) = \sqrt{1+4B^2}\, e \sin(\beta t + \mathrm{arctg}\frac{1}{2B});\ P(0) = 1;$$

$$a(t) = \sqrt{(1+4B^2)(\alpha^2+\beta^2)}e^{-\alpha t} \sin\left(\beta t + \mathrm{arctg}\frac{\alpha - 2\beta B}{\beta + 2\alpha B}\right);$$

$$\lambda(t) = \sqrt{\alpha^2+\beta^2}\, \frac{\sin\left(\beta t + \mathrm{arctg}\dfrac{\alpha - 2\beta B}{\beta + 2\alpha B}\right)}{\sin\left(\beta t + \mathrm{arctg}\dfrac{1}{2B}\right)}.$$

Т а б л и ц а 3.2а

Характе-ристики		Распределение		
		Рэлея	**Гамма**	**Вейбулла**
Плотность распределения		$a(t)=(t/\sigma^2)\times$ $\times\exp(-t/2\sigma^2)$, $0\le t<\infty$	$a(t)=\dfrac{\mu(\mu t)^{k-1}}{(k-1)!}\times$ $\times\exp(-\mu t)$, $0\le t<\infty,\ k\ge 2$	$a(t)=k\lambda_0 t^{k-1}\exp(-\lambda_0 t^k)$, $0\le t<\infty,\ k\ge 2$
$\overline{a}(t)$	α	$\sqrt{\pi}/\sqrt{2}\sigma(\pi-2)$	$\mu/(k-1)$	$\dfrac{\lambda_0^{1/k}\Gamma(1/k+1)}{(2\Gamma^2(1/k+1)-\Gamma(2/k+1))}$
	β	$\dfrac{\sqrt{(3/2)\pi-4}}{\sigma(\pi-2)}$	$\dfrac{\mu\sqrt{k-2}}{\sqrt{k}\,(k-1)}$	$\dfrac{\lambda_0^{1/k}\sqrt{3\Gamma^2(1/k+1-2\Gamma(2/k+1}}{(2\Gamma^2(1/k+1)-\Gamma(2/k+1))}$

Т а б л и ц а 3.2б

Характеристики		Распределение	
		Равномерное	**Нормальное**
Плотность распределения		$a(t)=\begin{cases}1/a, & 0\le t\le a,\\ 0, & t<0,\ t>a\end{cases}$	$a(t)=(\sqrt{2\pi}\sigma)^{-1}\exp(-(t-a)^2/2\sigma^2)$, $-\infty<t<\infty,\ \eta=\sigma/a\le 1/2$
$\overline{a}(t)$	α	$3/a$	$(a(1-\eta^2))^{-1}$
	β	$\sqrt{3}/a$	$\sqrt{1-2\eta^2}\,a(1-\eta^2)$

$\overline{a}(t)=((\alpha^2+\beta^2)/\beta)e^{-\alpha t}\sin\beta t$

При $\beta=B=0$ $\lambda(t)=\alpha$; при $\beta=0$, $B\neq 0$ $P(t)=e^{-\alpha t}$; при $\beta\neq 0$, $B=0$ $P(t)=e^{-\alpha t}\cos\beta t$; $\lambda(t)=\sqrt{\alpha^2+\beta^2}\,\dfrac{\cos\left(\beta t+\operatorname{arctg}\dfrac{\beta}{\alpha}\right)}{\cos\beta t}$;

$$a(t)=\sqrt{\alpha^2+\beta^2}\,e^{-\alpha t}\cos\left(\beta t+\operatorname{arctg}\frac{\beta}{\alpha}\right);$$

$$\alpha=0\to P(t)=1;\ \alpha=0,\ \beta=0\to P(t)=\cos\beta t,\ a(t)=\beta\cos\left(\beta t+\frac{\pi}{2}\right);$$

$$\lambda(t)=\beta\frac{\cos\left(\beta t+\dfrac{\pi}{2}\right)}{\cos\beta t}.$$

Однако этими выражениями нужно пользоваться достаточно осторожно, а именно, проверять условие нормировки плотности, неотрицательность значений вероятности, плотности вероятности, интенсивности отказа и т.д. Только при соблюдении этих условий использование показателей в расчётах становится удобным и предпочтительным.

Для приближённого представления показателей надёжности по методу последовательных фаз можно использовать следующие выражения:

$$a(t)=\frac{\alpha^2+\beta^2}{\beta}e^{-\alpha t}\sin\beta t;\ P(t)=\frac{\sqrt{\alpha^2+\beta^2}}{\beta}e^{-\alpha t}\sin\left(\beta t+\operatorname{arctg}\left(\frac{\beta}{\alpha}\right)\right);$$

$$\lambda(t)=\frac{\sqrt{\alpha^2+\beta^2}\sin(\beta t)}{\sin\left(\beta t+\operatorname{arctg}\left(\dfrac{\beta}{\alpha}\right)\right)};\ T=\frac{2\alpha}{\alpha^2+\beta^2}.$$

О корректности представления этими выражениями значений показателей можно не опасаться. Однако, в обоих методах точность вычисления показателей будет определяться числом учитываемых начальных моментов.

Рассмотренными методами можно исследовать различные системы достаточно сложной структуры. В случае предъявления требования повышенной точности аппроксимации плотностей распределений в разложениях следует использовать большее число пар членов с комплексно-сопряжёнными коэффициентами и параметрами. В методе параллельных фаз использование четырёх членов соответствует учёту семи начальных моментов, использование шести членов – одиннадцати начальных моментов и т.д. В методе последовательных фаз использование четырёх попарно-сопряжённых членов соответствует учёту четырёх начальных моментов, использование шести членов – шести начальных моментов и т.д. При этом искомые коэффициенты и параметры определяются численно. Приведённые в табл. 3.1. и 3.2. значения параметров и коэффициентов разложений в этом случае можно брать в качестве первых приближений соответствующих искомых величин, а в качестве первых приближенных значений остальных параметров и коэффициентов разложений – нули.

Точность аппроксимации достаточно высокая даже при использовании только трёх начальных моментов распределений и зависит от требуемой точности определения искомых величин. При их точности порядка 0,01…0,002 точность аппроксимации высока даже для сравнительно «тяжёлых» законов распределений. Для определения параметров и коэффициентов использовались методы наискорейшего спуска и Ньютона.

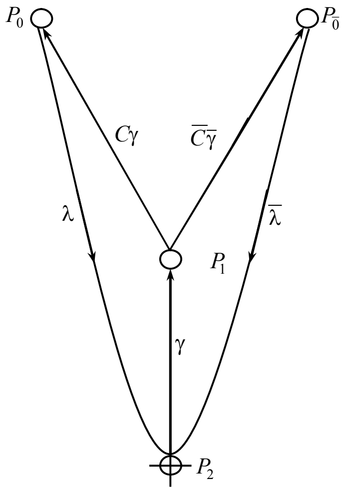


Рис. 3.2

<u>**Пример 3.2.**</u> Требуется определить коэффициент готовности неизбыточной системы программного обеспечения, плотности распределений времени до ошибки и времени её устранения произвольны, но представлены приближённо: первая – по методу параллельных фаз, вторая – по методу последовательных фаз. Тогда, если их обозначить соответственно через $\bar{a}(t)$ и $\bar{g}(t)$, будем иметь: $\bar{a}(t) = C\lambda e^{-\lambda t} + \overline{C}\overline{\lambda} e^{-\bar{\lambda} t}$, $\bar{g}(t) = \gamma\bar{\gamma}(e^{-\bar{\gamma}t} - e^{-\gamma t})/(\gamma - \bar{\gamma})$, где $C = A + jB$, $\overline{C} = A - jB$, $\lambda = \alpha' + j\beta'$, $\bar{\lambda} = \alpha' - j\beta'$, $\bar{\gamma} = \alpha'' - j\beta''$, $\gamma = \alpha'' + j\beta''$. Граф состояний и переходов системы показан на рис. 3.2. Крестиком помечено отказовое состояние системы. Составляя систему дифференциальных уравнений и решая её при начальных условиях: $P_0 = C$, $P_{\bar{0}} = \overline{C}$, $P_1(0) = P_2(0) = 0$, найдём функцию и коэффициент готовности. В частности, коэффициент готовности системы определяется выражением:

$$K_{\text{г}} = ((\overline{C}\lambda + C\bar{\lambda})\gamma\bar{\gamma})/((\overline{C}\lambda + C\bar{\lambda})\gamma\bar{\gamma} + (\gamma + \bar{\gamma})\lambda\bar{\lambda}).$$

Справедливость этого выражения легко проверить. Процесс функционирования системы является простым альтернирующим процессом. Поэтому для функции готовности, представленной в изображении Лапласа, будет справедливо выражение

$$K_{\text{г}}^*(s) = (1 - \bar{a}(s))/(s[1 - \bar{a}^*(s)\bar{g}^*(s)]),$$

в котором $*$ и s обозначают оператор преобразования и переменную Лапласа. Подставляя в это выражение изображения аппроксимированных плотностей $\bar{a}(t)$, $\bar{g}(t)$ и переходя к оригиналу, получим совпадение функций готовности и, следовательно, коэффициентов готовности, найденных указанными в данном примере двумя способами.

3.2. ГИПЕРДЕЛЬТНАЯ АППРОКСИМАЦИЯ

Представим произвольную плотность распределения $a(t)$, сосредоточенную на $[0, \infty)$, приближённо в виде [59]

$$\bar{a}(t) = \sum_{i=1}^{n} C_i \delta(t - T_i), \tag{3.7}$$

где C_i – вероятности, удовлетворяющие условию $\sum_{i=1}^{n} C_i = 1$, а δ – дельта-функция Дирака. Очевидно, что точность подобного представления будет тем выше, чем больше величина n. Для определения неизвестных постоянных C_i, T_i можно воспользоваться методом моментов. Если $n > 2$, то постоянные определяются численно. Когда $n = 2$, эти постоянные можно найти аналитически из следующей системы нелинейных уравнений:

$$\begin{gathered} C_1 + C_2 = 1, C_1T_1 + C_2T_2 = \nu_1, \\ C_1T_1^2 + C_2T_2^2 = \nu_2, C_1T_1^3 + C_2T_2^3 = \nu_3, \end{gathered} \tag{3.8}$$

в которой ν_i – i-й начальный момент случайной величины, распределённой с плотностью $a(t)$. Решая систему (3.8), получим:

$$\begin{gathered} T_{1,2} = \frac{\nu_3 - \nu_2\nu_1 \mp \sqrt{\nu_3^2 - 6\nu_3\nu_2\nu_1 - 3\nu_2^2\nu_1^2 + 4\nu_3\nu_1^3 + 4\nu_2^3}}{2(\nu_2 - \nu_1^2)}, \\ C_{1,2} = \frac{1}{2}\left(1 \pm \frac{3\nu_2\nu_1 - \nu_3 - 2\nu_1^3}{\sqrt{\nu_3^2 - 6\nu_3\nu_2\nu_1 - 3\nu_2^2\nu_1^2 + 4\nu_3\nu_1^3 + 4\nu_2^3}}\right). \end{gathered} \tag{3.9}$$

Для $n = 2$ представление плотности принимает вид $\bar{a}(t) = C_1\delta(t - T_1) + C_2\delta(t - T_2)$, причём значения C_1, C_2, T_1, T_2 находятся из (3.9) при условии, что моменты ν_1, ν_2, ν_3 известны.

В табл. 3.3. приведены значения искомых постоянных величин для некоторых распределений.

Т а б л и ц а 3.3а

Характеристики		Распределение		
		Экспоненциальное	Гамма	Нормальное
Плотность распределения		$a(t)=\lambda\exp(-\lambda t)$, $t\ge 0$	$a(t)=\lambda(\lambda t)^{k-1}\times$ $\times\exp(-\lambda t)/(k-1)!$, $t\ge 0$	$a(t)=\dfrac{\exp(-(t-a)^2/2\sigma^2)}{\sqrt{2\pi}\sigma}$, $-\infty<t<\infty$
$\overline{a}(t)$	C_1	$\dfrac{1}{2}\left(1+\dfrac{\sqrt{2}}{2}\right)$	$\dfrac{1+\sqrt{k+1}}{2\sqrt{k+1}}$	$\dfrac{1}{2}$
	C_2	$\dfrac{1}{2}\left(1-\dfrac{\sqrt{2}}{2}\right)$	$\dfrac{1-\sqrt{k+1}}{2\sqrt{k+1}}$	$\dfrac{1}{2}$
	T_1	$\dfrac{1}{\lambda}\left(1-\dfrac{\sqrt{2}}{2}\right)$	$\dfrac{k+1-\sqrt{k+1}}{\lambda}$	$a-\sigma$
	T_2	$\dfrac{1}{\lambda}\left(1+\dfrac{\sqrt{2}}{2}\right)$	$\dfrac{k+1+\sqrt{k+1}}{\lambda}$	$a+\sigma$

Т а б л и ц а 3.3б

Характеристики		Распределение	
		Равномерное	Рэлея
Плотность распределения		$a(t)=\begin{cases}\dfrac{1}{a}, 0\le t\le a,\\ 0, t<0, t>a\end{cases}$	$a(t)=(t/\sigma^2)\times\exp(-t^2/2\sigma^2)$, $t>0$
$\overline{a}(t)$	C_1	$\dfrac{1}{2}$	$\dfrac{1}{2}\left(1+\dfrac{\sqrt{\pi}(3-\pi)}{\sqrt{6\pi^2-39\pi+64}}\right)$
	C_2	$\dfrac{1}{2}$	$\dfrac{1}{2}\left(1-\dfrac{\sqrt{\pi}(3-\pi)}{\sqrt{6\pi^2-39\pi+64}}\right)$
	T_1	$\dfrac{a}{2}\left(1-\dfrac{1}{\sqrt{3}}\right)$	$\dfrac{\sqrt{\pi}-\sqrt{6\pi^2-39\pi+64}}{\sqrt{2}(4-\pi)}\sigma$
	T_2	$\dfrac{a}{2}\left(1+\dfrac{1}{\sqrt{3}}\right)$	$\dfrac{\sqrt{\pi}+\sqrt{6\pi^2-39\pi+64}}{\sqrt{2}(4-\pi)}\sigma$

$\overline{a}(t)=C_1\delta(t-T_1)+C_2\delta(t-T_2)$

Пример 3.3. Представить аппроксимацию гипердельтным распределением с $n=2$ экспоненциальной функции распределения с параметром $\lambda=10^{-2}$ ч$^{-1}$. Имеем: $\overline{a}(t)=C_1\delta(t-T_1)+$ $+C_2\delta(t-T_2)$, $C_1\approx 0{,}854$; $C_2\approx 0{,}146$; $T_1\approx 58{,}6$ ч; $T_2\approx 341{,}4$ ч. Экспоненциальная функция распределения и её аппроксимация

$$Fa(t)\approx C_1\overline{1}(t-T_1)+C_2\overline{1}(t-T_2)=\begin{cases}0, t<T_1;\\ C_1, T_1\le t<T_2;\\ 1, t\ge T_2.\end{cases}$$

показаны на рис. 3.3.

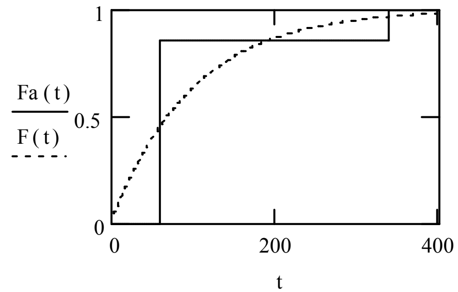


Рис. 3.3

Пример 3.4. Требуется в результате моделирования работы вычислительной сети, топологическая схема которой приведена на рис. 3.3, определить начальные моменты распределения длительности решения задачи в сети. Предполагалось, что распределение времени между задачами на входе сети подчинено гамма-распределению с параметрами $\mu=0{,}5$, $k=0{,}8$, а время обслуживания в каждом из узлов распределено по такому же закону с параметрами $\mu=1$, $k=1{,}2$. На рис. 3.4 показаны вероятности распределения заявок между узлами сети.

На рис. 3.4 буква И означает исток, а буква С – сток сети. Прямоугольниками со штрихами обозначены очереди заявок на входах узлов 1, 2, 3. В табл. 3.4. приведены начальные моменты распределения длительности пребывания заявки в сети, полученные моделированием по двум вариантам. В первом варианте использовался программный датчик, выдающий реализации случайных чисел, подчинённых гамма-распределениям.

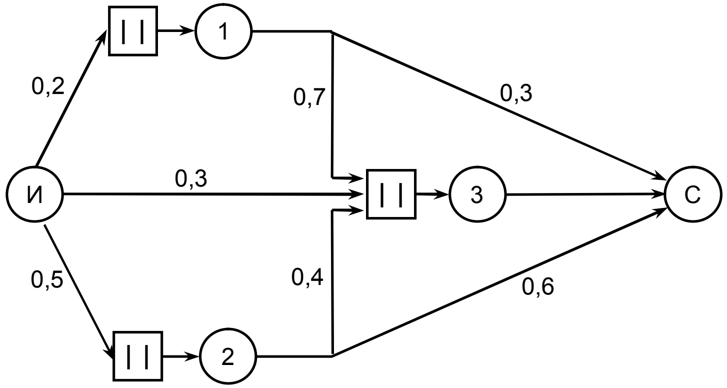


Рис. 3.4.

Во втором варианте использовался датчик случайных чисел, основанный на гипердельтном разложении распределений. Результаты моделирования при $n = 3$ получены для $2{\cdot}10^4$ испытаний. Существенным является тот факт, что время моделирования сети при использовании гипердельтного датчика уменьшилось в четыре раза. ▲

Таблица 3.4

Порядок момента	Вариант 1	Вариант 2
1	1,85 E0	1,94 E0
2	6,92 E0	7,05 E0
3	3,37 E1	4,00 E1
4	1,87 E2	2,14 E2
5	1,13 E3	1,41 E3

Пример 3.5. Требуется найти два первых начальных момента случайного времени жизни дублированной системы с ненагруженным состоянием резерва. При этом возможны две стратегии использования элементов. При первой стратегии резервный элемент его отказа. При второй стратегии оба элемента периодически с весьма малым интервалом времени меняются местами, обеспечивая равномерную выработку ресурсов обоих элементов. Отказ системы наступает при отказе двух элементов. С физической точки зрения применение этих стратегий должно приводить к одинаковому вероятностному результату.

Вероятности исправного функционирования системы при указанных стратегиях равны:

$$P_{c_1}(t) = P(t) + \int_0^t a(z)P(t-z)dz\,;\; P_{c_2}(t) = P^2\left(\frac{t}{2}\right) + 2\int_0^{t/2} a(z)P(t-z)dz\,,$$

где $a(t)$, $P(t)$ – плотность вероятности и вероятность безотказной работы одного элемента.

Соответствующие плотности вероятности времени до отказа системы принимают вид:

$$a_{c_1}(t) = \int_0^t a(z)a(t-z)dz\,;\; a_{c_2}(t) = 2\int_0^{t/2} a(z)a(t-z)dz\,.$$

Заменяя $a(t)$ в данных выражениях её гипердельтным представлением при $n = 2$ и выполняя интегрирование, получим

$$a_{c_1}(t) = a_{c_2}(t) = C_1^2\delta(t-2T_1) + 2C_1C_2\delta(t-T_1-T_2) + C_2^2\delta(t-2T_2)\,,$$

что и следовало ожидать. Первый и второй начальные моменты будут равны:

$$\nu_1 = 2[C_1^2T_1 + C_1C_2(T_1+T_2) + C_2^2T_2]\,;$$
$$\nu_2 = 2[2C_1^2T_1^2 + C_1C_2(T_1+T_2)^2 + 2C_2^2T_2^2]\,.$$

Вероятность исправного функционирования системы

$$P_c(t) = \begin{cases} 1, & t < 2T_1; \\ 1 - C_1^2, & 2T_1 \le t < T_1 + T_2; \\ 1 - C_1^2 - 2C_1C_2, & T_1 + T_2 \le t < 2T_2; \\ 0, & t \ge 2T_2. \end{cases}$$

В частности, при нормальном законе распределения времени работы элемента до отказа $\nu_1 = 2T$, $\nu_2 = 3[2T^2 + \sigma^2]$, где T и σ – параметры данного распределения.

Кроме того, следует заметить, что найти преобразование Лапласа для $a_{c_2}(t)$ непосредственно не представляется возможным. Но так как преобразование Лапласа $a_{c_1}(t)$ равно $(a^*(s))^2$, а обе указанные плотности эквивалентны, можно определить, что

$$L\left\{\int_0^{t/2} a(z)a(t-z)dz\right\} = \frac{1}{2}(a^*(s))^2,$$

где L, s – оператор и переменная преобразования Лапласа. ▲

В порядке сравнения гипердельтной и гиперэкспоненциальной плотностей следует отметить:

– гипердельтная аппроксимация позволяет аппроксимировать плотности вероятностей на $(-\infty, \infty)$, а гиперэкспоненциальная на – $[0, \infty)$;

– параметры гипердельтной аппроксимации всегда являются вещественными, а параметры гиперэкспоненциальной аппроксимации могут быть и комплексно-сопряжёнными. Это обстоятельство несколько ограничивает возможности использования гиперэкспоненциальной аппроксимации;

– гиперэкспоненциальная плотность для сравнительно «тяжёлых» распределений вблизи начала координат может принимать отрицательные значения, чего не наблюдается у гипердельтной аппроксимации.

Отмеченные достоинства гипердельтного представления плотностей по сравнению с гиперэкспоненциальным представлением позволяют использовать его более эффективно при численном моделировании сложных случайных процессов на СВТ. Кроме того, они принципиально позволяют использовать его в анализе разрывных процессов. Напротив, в аналитическом исследовании предпочтение следует отдавать гиперэкспоненциальному представлению плотности вероятности.

3.3. ГИПЕРЭКСПОНЕНЦИАЛЬНАЯ АППРОКСИМАЦИЯ МЕТОДОМ ПРОИЗВОДНЫХ

Несмотря на широкую известность и важность метода моментов в вероятностном анализе и различных прикладных задачах ему присущи некоторые недостатки, снижающие эффективность данного метода. Во-первых, точность аппроксимации наиболее высока в средней и хвостовой части распределения, а в начальной его части оказывается, как правило, сравнительно низкой. Во многих прикладных задачах точность аппроксимации становится удовлетворительной, особенно в тех, где существуют установившиеся значения определяемых вероятностных характеристик. Если же определяются характеристики переходных процессов, данная аппроксимация становится достаточно грубой. Замечено, что в расчётах надёжности функционирования кратковременно действующих систем использование метода моментов может приводить к существенной неточности в оценках показателей, зависящих от времени. Во-вторых, в ряде прикладных задач приходится иметь дело с распределениями, не имеющими моментов, например, с распределениями типа Коши. Подобные распределения получаются при оценивании производительности вычислительных систем, обслуживающих группы абонентов, потоки задач которых сдвинуты друг от друга во времени, а также в некоторых других случаях. При этом полученные распределения, как правило, снова рассматриваются в качестве исходных для решения следующей задачи и т.д. В данных случаях предпочтение следует отдавать методу аппроксимации распределений на основе равенства начальных производных [46].

Пусть известна плотность вероятности $a(t)$, удовлетворяющая условию гладкости. Тогда справедливо её представление в виде ряда:

$$a(t) = a(0) + a'(0)t + \frac{a''(t)}{2!}t^2 + \dots . \tag{3.10}$$

В изображении Лапласа (3.10) примет вид:

$$a^*(s) = \frac{a(0)}{s} + \frac{a'(0)}{s^2} + \frac{a''(0)}{s^3} + \dots , \tag{3.11}$$

где $a(0)$, $a'(0)$, $a''(0)$,... – начальные значения плотности и её производных. После замены переменной $s = 1/z$ из (3.11) получим:

$$\frac{1}{z}a^*\left(\frac{1}{z}\right) = a(0) + a'(0)z + a''(0)z^2 + \dots , \tag{3.12}$$

тогда

$$\lim_{z\to 0}\left(\frac{1}{z}a^*\left(\frac{1}{z}\right)\right) = a(0) = a_0 \text{ ; } \lim_{z\to 0}\left(\frac{1}{z}a^*\left(\frac{1}{z}\right)\right)' = a'(0) = a_1 \text{ ;}$$

$$\frac{1}{2!}\lim_{z\to 0}\left(\frac{1}{z}a^*(z)\right)'' = a''(0) = a_2. \tag{3.13}$$

и т.д.

Таким образом, формально значения плотности и её производных в точке нуль можно найти, используя преобразование Лапласа. Приравняв их соответствующим величинам аппроксимирующей плотности распределения выбранного вида и решив полученную систему уравнений, можно найти искомые параметры аппроксимирующего распределения. Рассмотрим простейшие примеры. В качестве базового аппроксимирующего распределения возьмём гиперэкспоненциальное распределение с параллельными фазами.

<u>**Пример 3.6**</u>. Требуется аппроксимировать плотность распределения времени до отказа дублированной системы с нагруженным состоянием резерва, если безотказность каждого её элемента описывается экспоненциальным распределением с параметром λ. Используя (3.13), найдём $a_0 = 0$, $a_1 = 2\lambda^2$, $a_2 = -6\lambda^3$. Так как базовая плотность вероятности $a(t) = c_1\lambda_1 e^{-\lambda_1 t} + c_2\lambda_2 e^{-\lambda_2 t}$, то соответствующая система уравнений по методу производных имеет вид:

$$\begin{cases} c_1 + c_2 = 1, \\ c_1\lambda_1 + c_2\lambda_2 = a_0, \\ c_1\lambda_1^2 + c_2\lambda_2^2 = -a_1, \\ c_1\lambda_1^3 + c_2\lambda_2^3 = a_2. \end{cases} \tag{3.14}$$

Решив систему (3.14), получим:

$$\begin{aligned} c_{1,2} &= \frac{1}{2}\left(1 \mp \frac{3a_1a_0 + a_2 + 2a_0^3}{\sqrt{a_2^2 + 6a_2a_1a_0 - 3a_1^2a_0^2 + 4a_2a_0^3 - 4a_1^3}}\right), \\ \lambda_{1,2} &= \frac{-(a_2 + a_1a_0) \pm \sqrt{a_2^2 + 6a_2a_1a_0 - 3a_1^2a_0^2 + 4a_2a_0^3 - 4a_1^3}}{2(a_1 + a_0^2)}. \end{aligned} \tag{3.14}$$

После подстановки значений a_0, a_1, a_2 найдём $c_1 = 2$; $c_2 = -1$; $\lambda_1 = \lambda$; $\lambda_2 = 2\lambda$, т.е. аппроксимирующая плотность будет равна $a(t) = 2\lambda e^{-\lambda t}(1 - e^{-\lambda t})$. Нетрудно убедиться, что полученное аппроксимирующее выражение точно совпадает с теоретическим выражением – плотностью распределения времени до отказа рассматриваемой дублированной системы. ▲

<u>**Пример 3.7**</u>. Необходимо аппроксимировать плотность распределения Коши (не имеет моментов). Поступая так же, как и в предыдущем примере, получим:

$$a(x) = \frac{2}{\pi}\cdot\frac{1}{1+x^2}; \quad a_0 = \frac{2}{\pi}; \quad a_1 = 0; \quad a_2 = -\frac{4}{\pi}.$$

В результате решения системы уравнений (3.14) найдём следующие значения:

$$\begin{cases} c_1 = \dfrac{1}{2}\left(1 + \dfrac{\pi^2 - 4}{\pi\sqrt{\pi^2 - 8}}\right); \quad c_2 = \dfrac{1}{2}\left(1 - \dfrac{\pi^2 - 4}{\pi\sqrt{\pi^2 - 8}}\right); \\ \lambda_1 = \dfrac{\pi}{2}\left(1 - \dfrac{\sqrt{\pi^2 - 8}}{\pi}\right); \quad \lambda_2 = \dfrac{\pi}{2}\left(1 + \dfrac{\sqrt{\pi^2 - 8}}{\pi}\right). \end{cases} \tag{3.15}$$

На рис. 3.5. показаны плотность распределения Коши и её гиперэкспоненциальная аппроксимация со значениями параметров (3.15). Отметим, что хорошее приближение начала распределения по методу производных обусловлено ограничением хвостовой части теоретической плотности распределения. Это объясняется меньшим влиянием больших значений случайной величины.

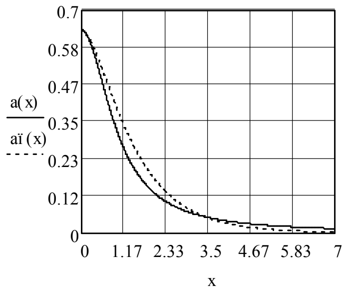


Рис. 3.5

<u>Замечание о возможном численном формализме метода</u>. Использование выражений (3.13) в общем случае затруднительно. Кроме того, измерять значения a_0, a_1, a_2,… по статистическим данным также весьма проблематично. На практике о случайной величине удобнее судить по достаточно просто определяемым начальным моментам, когда они существуют. Поэтому может представить интерес использовать метод производных в том случае, когда известны только значения начальных моментов случайной величины. Формально поставим задачу нахождения начальных значений плотности распределения и её производных по известным моментам. При этом будем предполагать, что исходные распределения удовлетворяют условиям аналитичности. В качестве аппроксимирующего распределения будем брать то же гиперэкспоненциальное распределение. Поступим следующим образом.

Составим систему уравнений, соответствующих равенству известных начальных теоретических и аппроксимирующих моментов. Ограничимся тремя моментами.

$$\begin{cases} c_1 + c_2 = 1; \\ \dfrac{c_1}{\lambda_1} + \dfrac{c_2}{\lambda_2} = \tilde{\nu}_1; \\ \dfrac{c_1}{\lambda_1^2} + \dfrac{c_2}{\lambda_2^2} = \tilde{\nu}_2; \\ \dfrac{c_1}{\lambda_1^3} + \dfrac{c_2}{\lambda_2^3} = \tilde{\nu}_3, \end{cases} \tag{3.16}$$

где $\tilde{\nu}_i = \nu_i / i!$, $i = 1, 2, 3$; ν_i – i-й начальный момент. В результате решения (3.16) найдём значения величин c_1, c_2, λ_1, λ_2, которые в общем случае могут быть комплексно-сопряжёнными. Найденные значения этих величин подставим в систему уравнений (3.14), которую решим относительно искомых неизвестных a_0, a_1, a_2. Получим

$$a_0 = \frac{\tilde{\nu}_3 - 2\tilde{\nu}_1\tilde{\nu}_2 + \tilde{\nu}_1^3}{\tilde{\nu}_1\tilde{\nu}_3 - \tilde{\nu}_2^2};$$

$$a_1 = -\frac{\tilde{\nu}_3^2 - 4\tilde{\nu}_1\tilde{\nu}_2\tilde{\nu}_3 + \tilde{\nu}_1^2\tilde{\nu}_2^2 + 2\tilde{\nu}_1^3\tilde{\nu}_3 - \tilde{\nu}_1^4\tilde{\nu}_2 + \tilde{\nu}_2^3}{(\tilde{\nu}_1\tilde{\nu}_3 - \tilde{\nu}_2^2)^2};$$

$$a_2 = (\tilde{\nu}_3^3 - 6\tilde{\nu}_1\tilde{\nu}_2\tilde{\nu}_3 + 6\tilde{\nu}_1^2\tilde{\nu}_2^2\tilde{\nu}_3 + 3\tilde{\nu}_1^3\tilde{\nu}_3^2 + 2\tilde{\nu}_2^3\tilde{\nu}_3 + 2\tilde{\nu}_1^3\tilde{\nu}_2^3 - \\ - 6\tilde{\nu}_1^4\tilde{\nu}_2\tilde{\nu}_3 - 3\tilde{\nu}_1\tilde{\nu}_2^4 + \tilde{\nu}_1^6\tilde{\nu}_3)/(\tilde{\nu}_1\tilde{\nu}_3 - \tilde{\nu}_2^2)^3.$$

Таким образом, формально в классе гиперэкспоненциальной аппроксимации получим значения параметров, необходимых для метода производных, располагая значениями начальных моментов распределения.

В заключение следует отметить, что вопросы оценивания точности параметров по методу производных применительно к различным распределениям требуют дальнейших исследований. Метод производных может быть успешно применён в анализе нестационарных процессов вычислительных систем и сетей, в задачах исследования надёжности аппаратных и программных средств, в исследовании эффективности систем. Использование в этом методе в качестве базового распределения гипердельтной аппроксимации требует привлечения аппарата обобщённых функций.

Глава 4

МОДЕЛИ НАДЁЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Модели надёжности программ, реализуемых на современных средствах вычислительной техники (СВТ), предназначены для получения численных значений показателей надёжности и представляют собой математические функции от определённых параметров. Рассмотрение этих параметров в качестве аргументов и численное оценивание их значений производится в соответствии с методикой построения моделей надёжности с учётом характеристик процесса разработки конкретно разрабатываемых программ (П). С методической точки зрения модели надёжности подразделяют на три класса: эмпирические, статистические и вероятностные. В рамках каждого класса модели надёжности могут отличаться применимостью на различных фазах и стадиях жизненного цикла П и определяемыми показателями надёжности.

Эмпирические модели являются наиболее простыми моделями. Основаны на анализе накопленной информации о функционировании разработанных П. Например, считалось, что если в П на каждые 1000 операторов приходится 10 ошибок [35], то она пригодна к эксплуатации. По другим данным [24], уровень надёжности П считается приемлемым, если на 1000 операторов приходится одна ошибка, то есть $N = 10^{-3}V$, где N, V – число ошибок и операторов в П. Эмпирические модели для оценивания числа ошибок в П фирмой IBM представлялись в виде:

1) $N = 23M(10) + 2M(1)$, где $M(10)$ – число модулей, требующих 10 и более исправлений, $M(1)$ – число модулей, содержащих менее 10 ошибок;