



Желтов С.Ю., Каляев И.А., Косьянчук В.В.,
Мельник Э.В., Зыбин Е.Ю.

РЕКОНФИГУРАЦИЯ СИСТЕМ УПРАВЛЕНИЯ ВОЗДУШНЫХ СУДОВ

Москва
2021

УДК 629.7.067
ББК 39.56
Р36

Желтов С.Ю., Каляев И.А., Косьянчук В.В., Мельник Э.В., Зыбин Е.Ю. **Рекон-
фигурация систем управления воздушных судов.** – М., РАН, 2021. – 204 с., 91 илл.

СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ	6
ВВЕДЕНИЕ	7
Глава 1. ОБЩАЯ ХАРАКТЕРИСТИКА ЗАДАЧИ РЕКОНФИГУРАЦИИ СИСТЕМ УПРАВЛЕНИЯ ВОЗДУШНЫХ СУДОВ	10
1.1. Архитектура систем управления воздушных судов	10
1.2. Характеристика исполнительных подсистем систем управления	14
1.3. Характеристика вычислительных подсистем систем управления	19
Глава 2. РЕКОНФИГУРАЦИЯ ИСПОЛНИТЕЛЬНЫХ ПОДСИСТЕМ СИСТЕМ УПРАВЛЕНИЯ ВОЗДУШНЫХ СУДОВ	24
2.1. Способы и методы реконфигурации исполнительных подсистем	24
2.2. Точное аналитическое решение задачи реконфигурации исполнительной подсистемы	27
2.2.1. Постановка задачи реконфигурации исполнительной подсистемы ...	27
2.2.2. Аналитическое решение линейных матричных уравнений методом канонизации	30
2.2.3. Решение задачи реконфигурации методом канонизации	34
2.2.4. Пример решения задачи реконфигурации методом канонизации	36
2.2.5. Резюме	60
2.3. Оптимальное аналитическое решение задачи реконфигурации исполнительной подсистемы	61
2.3.1. Постановка задачи оптимальной реконфигурации	61
2.3.2. Оптимальное решение линейных матричных уравнений	62
2.3.3. Оптимальное решение задачи реконфигурации	67
2.3.4. Пример оптимального решения задачи реконфигурации	68
2.3.5. Резюме	83
2.4. Робастное аналитическое решение задачи реконфигурации исполнительной подсистемы	83
2.4.1. Постановка задачи робастной реконфигурации	83
2.4.2. Робастное решение задачи реконфигурации	86
2.4.3. Пример робастного решения задачи реконфигурации	86
2.4.4. Резюме	88

2.5. Субоптимальное решение задачи реконфигурации исполнительной подсистемы с учетом ограничений на управление	91
2.5.1. Постановка задачи реконфигурации с учетом ограничений на управление	91
2.5.2. Субоптимальное решение линейных матричных уравнений	93
2.5.3. Решение задачи реконфигурации с учетом ограничений на управление	98
2.5.4. Пример решения задачи реконфигурации с учетом ограничений на управление	99
2.5.5. Резюме	111
2.6. Обобщенный подход к реконфигурации исполнительной подсистемы системы управления	112
 Глава 3. РЕКОНФИГУРАЦИЯ ВЫЧИСЛИТЕЛЬНЫХ ПОДСИСТЕМ СИСТЕМ УПРАВЛЕНИЯ ВОЗДУШНЫХ СУДОВ	 116
3.1. Принципы организации вычислительного процесса в системах управления воздушных судов	116
3.2. Способы повышения надежности вычислительных подсистем	123
3.2.1. Структурное резервирование	123
3.2.2. Резервирование производительности	127
3.2.3. Выравнивание вычислительной нагрузки	133
3.2.4. Резюме	139
3.3. Методы и алгоритмы диспетчирования вычислительных подсистем	140
3.3.1. Способы организации диспетчера	140
3.3.2. Децентрализованное диспетчирование	143
3.3.3. Алгоритмы работы локального диспетчера	151
3.3.4. Резюме	155
3.4. Методы и алгоритмы реконфигурации вычислительных подсистем	156
3.4.1. Постановка задачи реконфигурации вычислительной подсистемы	156
3.4.2. Формализация задачи размещения подзадач по процессорным узлам	159
3.4.3. Алгоритмы размещения подзадач при реконфигурации	162
3.4.4. Резюме	169

3.5. Реконфигурация вычислительных подсистем систем управления с интегрированной модульной архитектурой	169
3.5.1. Особенности функционирования вычислительных подсистем с интегрированной модульной архитектурой	169
3.5.2. Способы «преодоления» ограничений стандартов ARINC 653 и ARINC 664 при выполнении процедуры реконфигурации	173
3.5.3. Алгоритм реконфигурации вычислительной подсистемы с учетом ограничений стандартов ARINC 653 и ARINC 664	175
3.5.4. Реализация реконфигурации процессорных узлов с помощью планировщика расписаний	179
3.5.5. Организация работы с контекстными данными при реализации реконфигурации за счет смены расписаний	183
3.5.6. Организация программных средств локального диспетчера процессорного узла вычислительной подсистемы	184
3.5.7. Резюме	190
ЗАКЛЮЧЕНИЕ	191
СПИСОК ЛИТЕРАТУРЫ	196

СПИСОК СОКРАЩЕНИЙ

БРУ	–	боковая ручка управления
ВП	–	вычислительная подсистема
ВС	–	воздушное судно
ГО	–	горизонтальное оперение
ГС	–	гидросистема
ИИ	–	источник информации
ИМА	–	интегрированная модульная авионика
ИП	–	исполнительная подсистема
ИУ	–	исполнительное устройство
КС	–	коммуникационная сеть
ЛД	–	локальный диспетчер
ОВТ	–	отклоняемый вектор тяги
ОС	–	операционная система
ОСРВ	–	операционная система реального времени
ПГО	–	переднее горизонтальное оперение
ПУ	–	процессорный узел
РВ	–	руль высоты
РН	–	руль направления
РУС	–	ручка управления самолетом
САУ	–	система автоматического управления
СУ	–	система управления
ФМ	–	функциональный модуль
ЭГРП	–	электрогидравлический рулевой привод
ЭДСУ	–	электродистанционная система управления
ЭС	–	электрическая система
ACE	–	actuator control electronics
EBHA	–	electro-backup hydrostatic actuator
EHA	–	electrohydraulic actuator
MACE	–	motor actuator control electronics
PFCU	–	primary flight control unit
PRIM	–	primary flight control and guidance computer
SEC	–	secondary flights control computer
SFCC	–	slats and flaps control computer

ВВЕДЕНИЕ

Повышение безопасности полетов является ключевым требованием к вновь создаваемым и находящимся в эксплуатации ВС. Наиболее критичными отказами с точки зрения безопасности полетов являются отказы, приводящие к изменению стереотипа управления ВС. К таким отказам относятся отказы в СУ ВС. По данным Росавиации, потеря управления в полете явилась причиной 21% катастроф и находится на первом месте по опасности последствий отказов.

Ключевой тенденцией современности в концепции построения СУ ВС является переход на ЭДСУ, в которой полностью исключаются механические связи между органами управления и поверхностями управления. Это, с одной стороны, расширяет возможности СУ по обеспечению заданного качества управления ВС, а с другой – предъявляет повышенные требования к надёжности ее элементов. Поэтому традиционно для обеспечения требований к надежности СУ ВС используется 3- и 4-кратное резервирование каналов управления.

Однако резервирование не всегда позволяет обеспечить отказоустойчивость СУ ВС. Например, при отказе общих для всех каналов элементов резервирование не дает желаемых результатов. Кроме того, существуют нерасчетные случаи повреждения всех каналов управления одной из поверхностей управления. Например, в результате пожара в районе одной из поверхностей управления могут быть повреждены все линии связи ЭДСУ с приводом, в результате поверхность не перемещается в соответствии с законом управления и, как следствие, появляются нерасчетные силы и моменты, воздействующие на ВС. Вследствие этого стереотип управления ВС изменяется, может быть потеряна устойчивость и управляемость.

Ключевыми и наиболее сложными с точки зрения задачи реконфигурации являются две подсистемы СУ ВС: ИП и ВП. ВП СУ формирует заданные сигналы управления в соответствии с реализуемым законом управления, а ИП СУ позволяет сформировать необходимые ВС силы и моменты, определяемые законом управления за счет отклонения поверхностей управления. В этой связи в монографии рассматриваются новые возможности по реализации реконфигурации СУ ВС, реализуемые в исполнительной и вычислительной подсистемах.

Реконфигурация ИП СУ ВС осуществляется за счет функционального резервирования – нового резерва СУ, который может быть использован для сохранения устойчивости и управляемости ВС в условиях отказов ИП, позволяющий возложить на штатные поверхности управления дополнительные, не свойственные им в штатном режиме работы функции. Использование функционального резервирования позволяет реализовывать реконфигурацию ИП СУ за счет перераспределения функций с неработоспособных поверхностей управления (основных) ВС на работоспособные (функционально резервные).

Большинство известных способов и методов реконфигурации ИП СУ обладают существенными недостатками, не позволяющими их реализовать на практике ввиду высокой степени критичности СУ ВС. Как правило, решение ищется в единственном численном виде, что требует пересчет законов реконфигурации в режиме реального времени полета, а учет конструктивных ограничений на отклонение поверхностей управления осуществляется сложными нелинейными методами, не позволяющими обеспечить детерминированность трудоемких вычислений.

В монографии формулируется новый способ реконфигурации ИП СУ ВС аналитическими методами, позволяющий получать простые аналитические законы реконфигурации, коэффициенты которых непосредственно настраиваются по режимам полета. Описываются оптимальные решения, минимизирующие как норму ошибки реконфигурации, так и норму самого решения, одновременно обеспечивая при этом максимальное сохранение стереотипа управления, минимальное изменение собственной динамики ВС с замкнутой СУ и минимально возможные компенсирующие отклонения работоспособных поверхностей управления, задействованных при реконфигурации. Оптимальные методы положены в основу получения робастных законов реконфигурации, сохраняющих удовлетворительные характеристики управления ВС при наличии параметрических или структурных неопределенностей. Приведена оригинальная методика реконфигурации ИП СУ ВС с учетом ограничений на отклонения поверхностей управления, позволяющая получать реализуемые на практике законы реконфигурации.

В основу методов реконфигурации ИП СУ ВС положены разработанные авторами специальные процедуры решения линейных матричных уравнений, позволяющие описывать множество всех возможных решений в аналитическом виде, получать оптимальные и субоптимальные решения поставленных задач. Все задачи реконфигурации ИП СУ иллюстрируются примерами решения с использованием символьных и числовых математических моделей реальных ВС.

Реконфигурация ВП СУ ВС предполагает перерасмещение функциональных программ на ПУ подсистемы, в результате чего изменяется номенклатура функций, выполняемых каждым из узлов. Реконфигурация ВП выполняется с целью повышения надежности за счет реализации устойчивости к отказам: программы перерасмещаются с отказавших ПУ на исправные. Это позволяет восстановить выполнение необходимых функций.

Большинство известных методов реконфигурации основаны на использовании структурного резерва, предполагающего введение в состав ВП дополнительных ПУ, что ухудшает массогабаритные и энергетические характеристики СУ ВС.

В монографии предложен способ резервирования производительности, основанный на использовании резерва производительности уже имеющихся ПУ и позволяющий уменьшить объем дополнительных аппаратных средств

при выполнении требований к надежности. Для данного способа резервирования разработаны методы и алгоритмы диспетчирования, размещения вычислительной нагрузки и восстановления вычислительного процесса, позволяющие проектировать ВП СУ ВС с учетом требований стандартов построения информационно-управляющих систем ИМА.

В целом применение излагаемых в книге подходов может составить основу построения реконфигурируемых СУ ВС.

Глава 1

ОБЩАЯ ХАРАКТЕРИСТИКА ЗАДАЧИ РЕКОНФИГУРАЦИИ СИСТЕМ УПРАВЛЕНИЯ ВОЗДУШНЫХ СУДОВ

1.1. Архитектура систем управления воздушных судов

Усложнение СУ ВС, главным образом из-за расширения их функций, увеличения числа поверхностей управления и их приводов, а также ужесточение требований к динамике СУ, точности отработки их сигналов и надежности управления обусловили необходимость широкомасштабного перехода к цифровым ЭДСУ.

Под ЭДСУ обычно понимается СУ, в которой механическая проводка управления заменена электрической или волоконно-оптической связью. ЭДСУ обеспечивает: ручное управление ВС; устойчивость и управляемость ВС; поддержание заданных динамических свойств объекта управления и т.д.

На рисунке 1.1 представлена общая схема управления самолетом с использованием ЭДСУ.



Рис. 1.1. Схема управления самолетом с использованием ЭДСУ

В качестве примера типовых ЭДСУ рассмотрим СУ самолетов SSJ-100 и А-380.

На рисунке 1.2 изображена структурная схема ЭДСУ самолета SSJ-100 [99]. ЭДСУ состоит из трех блоков вычислителей «верхнего» уровня (Primary Flight Control Unit – PFCU), каждый из которых содержит основной и контрольный канал, пятнадцать вычислителей «нижнего» уровня (Actuator Control Electronics – ACE) и шесть вычислителей «нижнего» уровня (Motor Actuator Control Electronics – MACE). Элероны и рули высоты приводятся в действие двумя электрогидроприводами каждый, работающими от разных ГС. Один из приводов всегда находится в активном режиме, а второй – в пассивном. Руль направления управляется тремя приводами, являющимися

активными и работающими от трёх гидросистем. Каждый привод элерона, руля высоты и направления управляется своим «персональным» вычислителем ACE. Интерцепторы имеют по одному гидроприводу на каждую секцию и по одному ACE на одну пару интерцепторов.

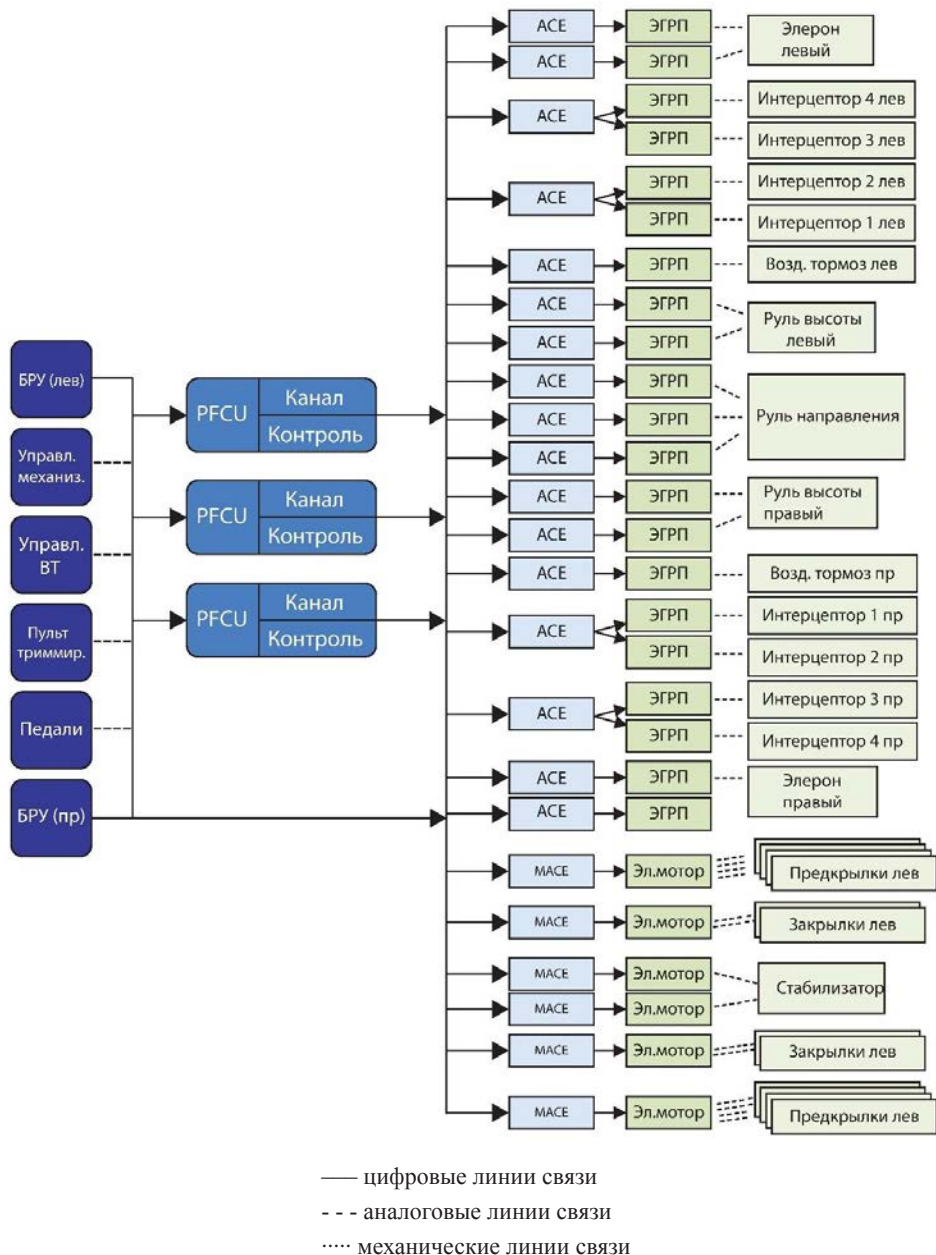


Рис. 1.2. Структурная схема ЭДСУ самолета SSJ-100

Вычислители получают управляющие сигналы от органов управления (БРУ, педали, рукоятка воздушных тормозов), передают их своим приводам и контролируют их положение по сигналам датчиков положения своего и соседнего привода на каждой поверхности.

Три вычислителя «верхнего» уровня PFCU обеспечивают работу ЭДСУ в нормальном режиме «Normal Mode». Каждый PFCU управляет всеми вычислителями «нижнего» уровня ACE, что обеспечивает полное 3-х кратное резервирование ЭДСУ. Отказ одного или двух PFCU приводит к деградации (упрощению) режима «Normal», а после отказа всех 3-х блоков система просто переходит в режим резервного (прямого) управления «Direct Mode».

В качестве поверхностей управления используются: стабилизатор, левый и правый элероны, левый и правый рули высоты, интерцепторы, предкрылки и закрылки.

СУ самолета A380 спроектирована с использованием ЭДСУ всеми поверхностями без механического резерва [39]. Основной ее инновацией является переход на две гидравлические системы и широкое применение электрогидростатических (ЕНА – electrohydraulic actuator) и комбинированных приводов (ЕВНА – electro-backup hydrostatic actuators) для управления самолетом. Схема ЭДСУ самолета представлена на рисунке 1.3. Силовое питание ЭДСУ включает две независимые между собой ГС и четыре ЭС переменного тока. Имеет место разнородное резервирование силового питания (ГС и ЭС). В работоспособном (безотказном) состоянии силовая система управления работает только от ГС. При отказе одной или двух ГС силовое питание ССУ осуществляется от ЭС. При отказе одной из ГС (вероятность отказа $\sim 10^{-6}$ может быть потеряно 4 (из 16) секции

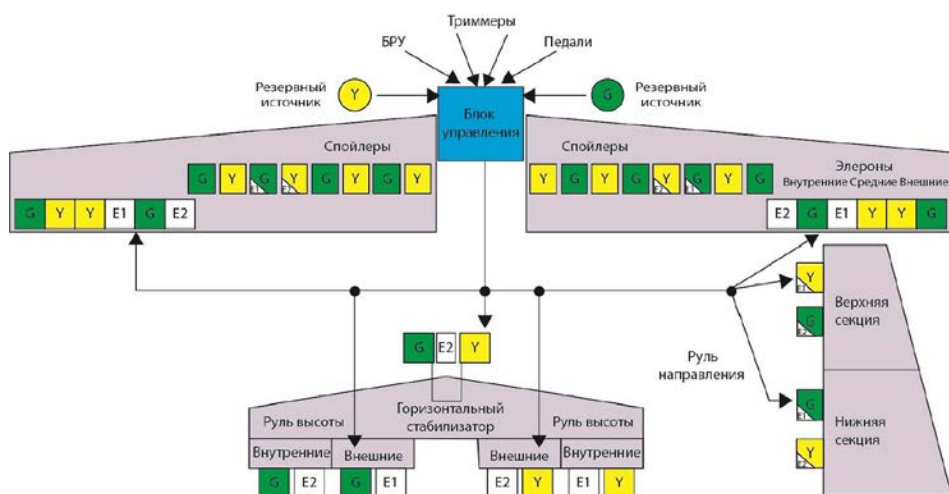


Рис. 1.3. Схема СУ самолета Airbus A380

спойлеров при сохранении всех остальных органов управления. Отказ двух ГС практически невероятен, но даже при этом будут потеряны только 8 (из 16) спойлеров и крайние секции элеронов. Так как надежность ГС на порядок выше, чем надежность ЭС, ЭС в штатном режиме не используется для силового питания.

В штатном (безотказном) режиме все секции органов управления отключаются ЭГРП, получающими энергию от ГС. При отказе ГС подключаются приводы ЕНА или ЕВНА, работающие от ЭС. На средних и корневых элеронах после отказа ГС активизируются приводы ЕНА. Они работают от электромоторов, а передача энергии от мотора к органу управления осуществляется местной для данного привода гидравликой (гидравлический редуктор для каждого привода).

На спойлерах и рулях направления установлены приводы с комбинированным энергопитанием – ЕВНА. В штатном режиме эти приводы работают от централизованной ГС, а при отказе ГС работают как ЕНА.

На секциях элеронов, руля высоты и руля направления установлено по два привода – один из них активен, а другой находится в «горячем» резерве.

Анализ ЭДСУ современных ВС позволяет представить обобщенную схему СУ ВС в виде, изображенном на рисунке 1.4, в структуре которой можно выделить следующие подсистемы:

- управляющую;
- исполнительную;
- вычислительную;
- измерительную.



Рис. 1.4. Обобщенная структурная схема СУ ВС

Управляющая подсистема предназначена для формирования командных сигналов управления пилота. Управляющая подсистема преобразует команды управления летчика в управляющие сигналы, передаваемые в систему управления. Обычно она включает: боковые ручки управления самолетом, штурвалы, ручки управления некоторыми системами и т.д.

Исполнительная подсистема предназначена для отклонения поверхностей управления ВС в соответствии с информацией об управляющих воздействиях и состоянии ВС с целью обеспечения желаемой динамики

полета. Обобщенно она включает: привода поверхностей управления, непосредственно сами поверхности управления, а также силовые и информационные связи между этими элементами.

Вычислительная подсистема предназначена для формирования управляющих воздействий в соответствии с заданными законами управления. Она включает вычислители различных уровней и соответствующую коммутационную аппаратуру.

Измерительная подсистема предназначена для сбора информации о состоянии ВС. Обычно измерительная система включает набор датчиков и наблюдатели, позволяющие восстановить недоступные для прямого измерения переменные вектора состояния.

Управляющая подсистема, как и измерительная подсистема СУ ВС, многократно резервирована, и отдельные отказы обычно не приводят к катастрофическим последствиям. В этой связи в данной работе рассматривать эти подсистемы не будем. А вот исполнительная и вычислительная подсистемы существенно влияют как на отказоустойчивость, так и на надежность СУ ВС. Исходя из этого проанализируем, за счет каких ресурсов можно обеспечить отказоустойчивость и каким способом осуществлять реконфигурацию этих подсистем.

1.2. Характеристика исполнительных подсистем систем управления

Статистика показывает, что около 20% потерь ВС происходят по причине потери управляемости ЛА в полете, обусловленных повреждениями и отказами СУ [2]. Наиболее опасными являются отказы и повреждения, приводящие к изменению логики работы поверхностей управления ВС и, следовательно, стереотипа управления летчика. Например, заклинивание поверхности управления, пропадание сигналов управления с привода поверхности управления, например, в результате пожара в районе поверхности управления и т.д. В таких ситуациях летчик, неизбежно сталкиваясь с внезапным и существенным изменением пилотажных характеристик ВС, из-за свойственного ему запаздывания не всегда способен компенсировать последствия этих изменений. При возникновении таких отказов реакция ВС на управляющие воздействия становится неадекватной сформированному у летчика в процессе обучения и летной практики стереотипу, что в совокупности с острым дефицитом времени на принятие решения приводит к авиационным катастрофам.

Для компенсации отказов ИП СУ ВС возможно использование как традиционных подходов, основанных на аппаратном резервировании, так и нетрадиционных – основанных на функциональном резервировании.

Аппаратное резервирование в настоящее время является основным методом повышения надежности и отказоустойчивости СУ ВС. Вероятность

отказа СУ, приводящего к катастрофическим последствиям, не должна превышать величину $10^{-7} \dots 10^{-9}$ за час полета. Этому уровню соответствуют четырехканальные (трехкратно резервированные) системы постоянного резервирования с выключением отказавших каналов. Они обеспечивают при использовании метода контроля по «большинству» (так называемый метод кворум элементов) работоспособность системы при наличии двух аналогичных по характеру отказов.

Теоретические и экспериментальные исследования, а также опыт, накопленный в процессе эксплуатации различных СУ ВС в нашей стране и за рубежом, позволяют сегодня выявить ряд общих недостатков, присущих резервированным системам.

Во-первых, главным и существенным недостатком резервированных систем является увеличение суммарного потока отказов и соответственно уменьшение наработки на отказ, что приводит к увеличению эксплуатационных затрат, ухудшению показателей готовности и регулярности вылета ВС. Существенный вклад в столь низкий уровень надежности (низкую отказоустойчивость) вносят и элементы резервированных каналов СУ, поскольку резервирование увеличивает собственно количество элементов, подверженных отказам. Таким образом, резервирование, как способ повышения безопасности полетов, вступает в противоречие с требованиями по безопасности и эксплуатационной технологичности.

Во-вторых, в резервированной системе можно выделить так называемые «общие точки» и «общие внешние факторы». В качестве «общих точек» могут служить устройства контроля и/или устройства диагностирования, объединяющие:

- все подканалы данного канала управления;
- силовые приводы (выходное звено гидропривода);
- устройство суммирования и др.

Отказы таких единичных элементов и устройств также могут приводить к потере работоспособности всей СУ ВС.

В качестве примера «общих внешних факторов» можно указать перегорание из-за местного пожара электрического кабеля, в котором, например, проходят все кабели резервных подканалов управления приводом какой-либо поверхности управления ВС.

В-третьих, при аппаратном резервировании из-за наличия избыточных элементов ухудшается ряд важных показателей СУ:

- возрастают масса, габаритные размеры и стоимость;
- увеличивается потребление энергии;
- растут трудоемкость и стоимость технического обслуживания;
- растет время отыскания неисправности;
- увеличивается время и затраты на испытания.

В-четвертых, аппаратное резервирование становится не эффективным, когда появляются (или имеются) ошибки в программном обеспечении. Та-

кие ошибки могут вызвать потерю работоспособности одновременно во всех резервированных каналах СУ.

В-пятых, на некоторых образцах авиационной техники предусматривается резервная механическая проводка, подключаемая в случае отказа СУ (так называемое резервирование замещением с облегченным резервом). Однако использование такого резерва возможно только на статически устойчивых ВС (либо на статически устойчивых режимах полета). В противном случае подобное резервирование не даст положительного эффекта, тем более в случае активного отказа СУ (заклинивание поверхности управления в каком-либо промежуточном положении, ее самопроизвольный «увод» из заданного положения и т.д.).

Таким образом, можно констатировать, что весьма актуальной является задача разработки принципиально новых подходов к построению систем управления перспективных ВС и в том числе их алгоритмического и программного обеспечения. Такие СУ должны удовлетворять требованиям безопасности полетов, быть в максимальной степени эксплуатационно-технологичными, и самое главное – обеспечивать максимальную эффективность использования всех имеющихся ресурсов в интересах обеспечения отказоустойчивости СУ. В этой связи рассмотрим новый ресурс обеспечения отказоустойчивости ИП СУ – функциональное резервирование поверхностей управления ВС.

Под функциональным резервированием понимается способность СУ ВС использовать не свойственные ей в штатном режиме резервы, которые в штатном режиме выполняют одни функции, а в аварийном – другие. Для этого предполагается реализовать перераспределение функций отказавших или поврежденных каналов управления между оставшимися работоспособными. Ясно, что успех в этом направлении позволит в какой-то мере «заменить» резервирование, основанное на аппаратной избыточности, резервированием, основанным на функциональной избыточности (функциональным резервированием), которое, в свою очередь, способно обеспечить эффективное управление ВС в условиях отказов и повреждений. Этот подход в литературе получил название функциональной реконфигурации СУ [1, 16, 83]. Его основу составляют два важных момента.

Во-первых, это интенсивное развитие компьютерных технологий, способных обеспечить функционирование в реальном масштабе времени высокоэффективных и весьма сложных алгоритмов управления многосвязными динамическими системами.

Во-вторых, усиливающаяся тенденция роста количества поверхностей управления в современных и перспективных аэродинамических компонентах ВС, включая появление нетрадиционных каналов управления.

На рисунке 1.5 представлена схема современного маневренного самолета с различными поверхностями управления [25, 40].

Для управления самолетом, изображенным на рисунке 1.5, используются:

- две консоли ПГО;
- две консоли заднего ГО или стабилизатора;
- две секции флаперонов;
- две секции РН;
- две секции поворотных носков крыла.

Дополнительно используются два поворотных сопла (левое и правое) для управления ОВТ двигателей для создания управляющих моментов по тангажу (синфазное отклонение сопел), крену и рысканью (дифференциальное отклонение сопел).



Рис. 1.5. Поверхности управления маневренного самолета

На рисунке 1.6 представлены различные поверхности управления современного типового неманевренного самолета: элероны, элерон-интерцепторы, интерцепторы (спойлеры), предкрылки, руль направления, рули высоты и горизонтальный стабилизатор.

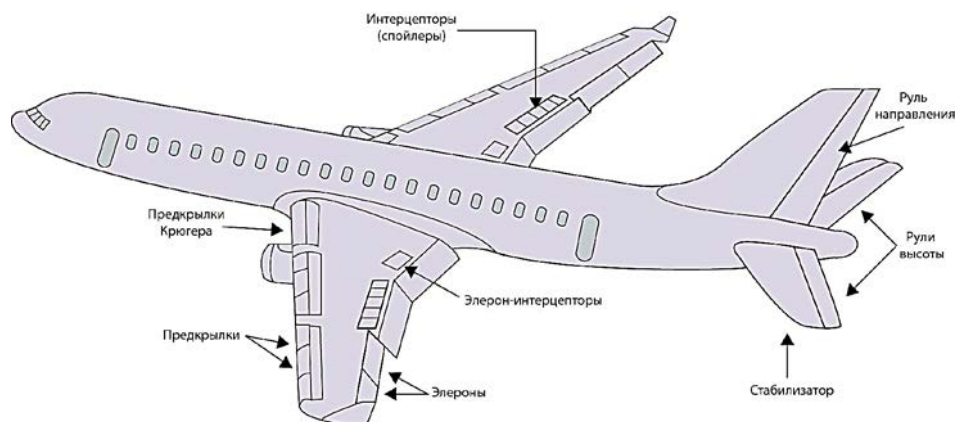


Рис. 1.6. Поверхности управления неманевренного самолета

Анализ приведенных выше аэродинамических компоновок современных самолетов показывает, что уже сейчас они имеют заведомо функционально избыточный набор поверхностей управления, позволяющих создавать управляющие (не только аэродинамические) силы и моменты одновременно по разным осям как в соответствии с их основными предназначением, так и выполняя не свойственные им в штатном режиме функции [26, 27, 55]. Так, например, при отказе стабилизатора компенсировать отказ можно за счет реконфигурации СУ самолета путем соответствующего отклонения элеронов, рулей высоты и интерцепторов (рисунок 1.7).

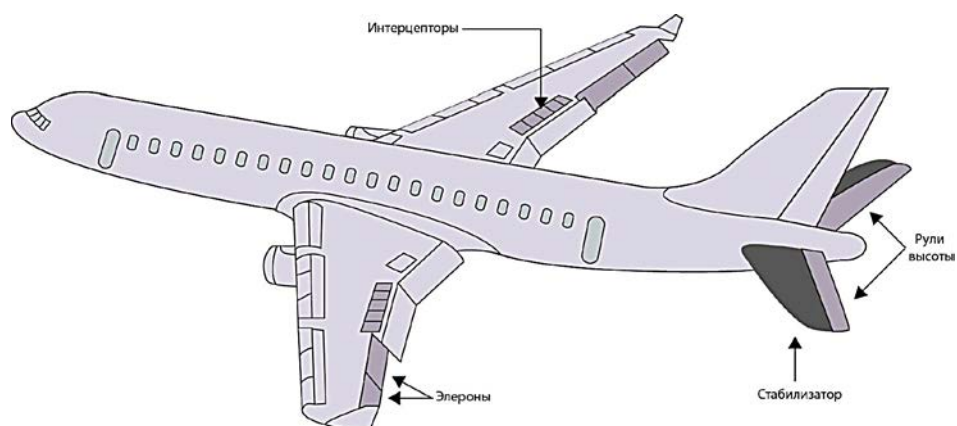


Рис. 1.7. Пример реконфигурации управления при отказе стабилизатора

В перспективных самолетах ожидается еще большее увеличение функциональной избыточности поверхностей управления как за счет введения новых поверхностей в аэродинамическую компоновку, так и за счет использования

независимых приводов для каждой из консолей существующих поверхностей управления. В таблице 1.1 перечислены варианты использования поверхностей управления перспективного самолета в различных каналах управления.

Табл. 1.1. Использование поверхностей управления перспективного самолета в различных каналах управления

Поверхности управления	Канал управления		
	крен	тангаж	курс
Горизонтальный стабилизатор	*	*	
Элевоны	*	*	*
Элероны	*	*	*
Флапероны	*	*	
Предкрылки	*	*	
Закрылки	*	*	*
Рули направления	*		*
Интерцепторы	*	*	*
Тормозные щитки	*	*	
Поворотные сопла	*	*	

Таким образом, можно утверждать, что функциональная реконфигурация ИП СУ за счет перераспределения функций с отказавших поверхностей управления (основных) на работоспособные (функционально резервные) является перспективным способом обеспечения отказоустойчивости СУ ВС. Именно разработке методов функциональной реконфигурации ИП СУ ВС будет посвящена 2-я глава монографии.

1.3. Характеристика вычислительных подсистем систем управления

Рассмотрим особенности построения ВП СУ ВС. На разных этапах развития авиации на ВС применялись СУ, построенные на основе различных архитектурных решений. Это было обусловлено номенклатурой реализуемых функций, имеющейся в наличии элементной базой, существующими методами и подходами к построению управляющих систем. Можно выделить несколько основных видов архитектур вычислительной подсистемы СУ ВС.[3, 48, 24]:

- конфедеративная архитектура;
- федеративная архитектура;
- архитектура на базе концепции ИМА.

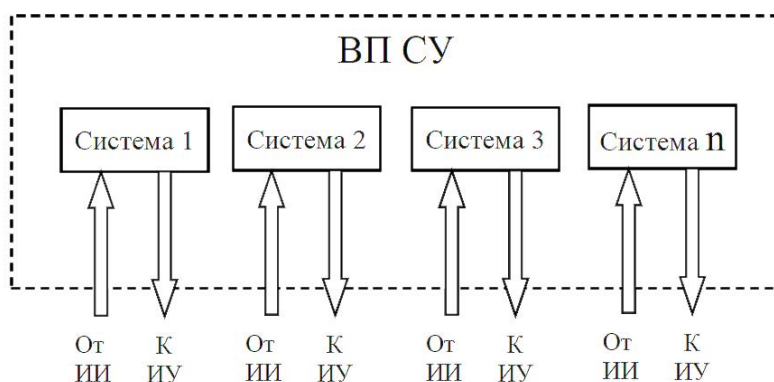
Конфедеративная архитектура (рис. 1.8) подразумевает наличие в составе СУ ВС множества независимых (не связанных друг с другом) систем, каждая из которых реализует определенную функцию [48]). К недостаткам такой архитектуры следует отнести:

- практически полное отсутствие унификации оборудования, т.к. каждая из подсистем разрабатывается и создается отдельным производителем. Вследствие этого увеличивается время разработки системы в целом, снижается ее ремонтпригодность, затрудняется модернизация, увеличивается номенклатура комплектующих;

- отсутствие возможности разделения ресурсов между отдельными подсистемами, поскольку каждая из них комплектуется своими уникальными датчиками, вычислителями и другими устройствами. Вследствие этого ухудшаются массогабаритные характеристики и увеличивается стоимость;

- необходимость использования в каждой системе отдельного дополнительного оборудования для обеспечения ее отказоустойчивости, что ухудшает массогабаритные характеристики, значительно увеличиваются энергопотребление и стоимость;

- расширение номенклатуры функций, решаемых системой, увеличивает число входящих в ее состав отдельных систем, что также негативно отражается на массогабаритных характеристиках.

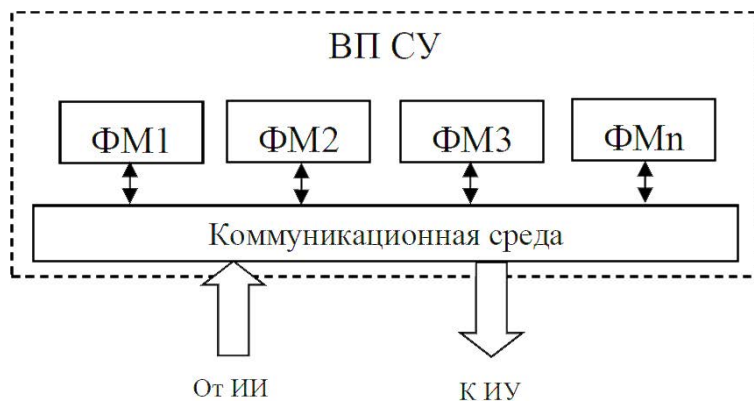


ИИ – источники информации, ИУ – исполнительные устройства

Рис. 1.8. Схема конфедеративной архитектуры

Федеративная архитектура в СУ ВС появилась в начале 80-х годов XX века вследствие расширения номенклатуры реализуемых функций. Ее главным отличием является распределение функций между взаимодействующими подсистемами, связанными между собой посредством мультиплексных каналов и других линий передач информации (ри. 1.9) [102, 103].

При этом, как и в конфедеративной архитектуре, каждая из подсистем представляет собой отдельный функциональный модуль (ФМ), ориентированный на выполнение одной из функций ВС.



ИИ – источники информации, ИУ – исполнительные устройства,
ФМ – функциональный модуль

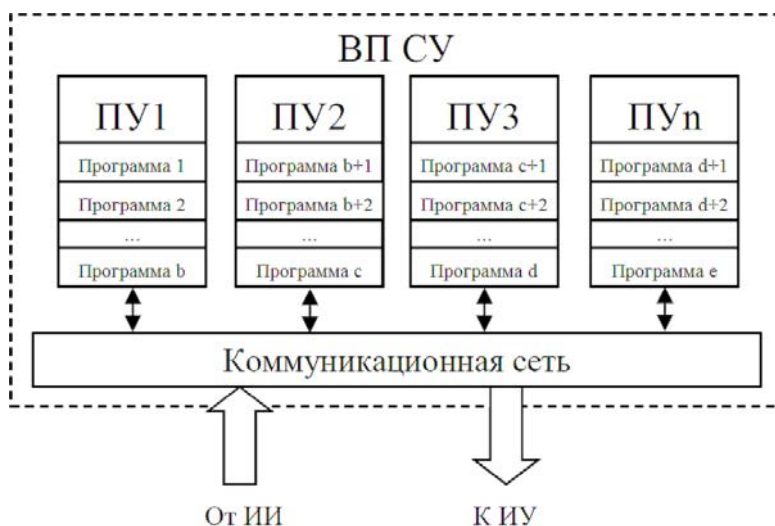
Рис. 1.9. Схема федеративной архитектуры

По сравнению с конфедеративной архитектурой федеративная архитектура позволила за счет комплексной обработки бортовой информации и применения мультиплексированных линий связи улучшить массогабаритные характеристики. Однако при этом системы, построенные на базе данной архитектуры, обладают следующими недостатками [48]:

- отсутствие унификации различных блоков, в частности ФМ и коммуникационного оборудования;
- программное обеспечение неразрывно связано с конструкцией ФМ и поэтому не может быть вторично использовано в последующих модификациях СУ или при наращивании ее дополнительных функций без повторной верификации и валидации;
- для обеспечения устойчивости к отказам необходимо осуществлять резервирование каждого ФМ по отдельности, что требует больших дополнительных аппаратных затрат.

С начала 2000-х годов начала развиваться принципиально новая концепция построения архитектуры СУ ВС, так называемая концепция ИМА. Данная концепция предполагает, что различные функции СУ ВС выделяются в логические разделы, которые могут физически реализовываться на любом унифицированном вычислительном устройстве (ПУ), входящем в состав ВП СУ ВС (рис. 1.10). При этом все вычислительные устройства (ПУ) соединяются посредством коммуникационной среды на базе высокоскоростных бортовых сетевых интерфейсов (КС) и являются высокоинтегрированными устройствами с общим программным уровнем, типичным для спецификации ARINC 653 APEX. Функции СУ ВС в этом случае выполняют программные приложения, разделяющие общие вычислительные ресурсы [47, 48].

Архитектура ВП СУ ВС на базе концепции ИМА является открытой, ориентированной на широкое применение унифицированных решений, аппаратуры и программ. Данная архитектура позволяет реализовать несколько функций на одних и тех же аппаратных средствах, выполнять отдельную разработку аппаратных и программных средств. Все это позволяет уменьшить время и снизить стоимость разработки СУ ВС.



ПУ – процессорный узел

Рис. 1.10. Схема с архитектурой ИМА

В целом к преимуществам организации вычислительного процесса на базе концепции ИМА следует отнести:

- высокий уровень стандартизации и унификации в рамках открытой архитектуры вычислительного устройства, что позволяет легко наращивать вычислительные ресурсы, одновременно использовать изделия различных производителей;
- широкие возможности совместного использования аппаратных средств при реализации различных функций СУ ВС, что позволяет существенно улучшить массогабаритные характеристики.

Примером типовой архитектуры ИМА можно считать архитектуру ВП СУ самолета А380. Информационно-вычислительная часть СУ самолета А380 построена как комплексная СУ, т.е. она интегрирует функции собственно ЭСДУ и системы автоматического управления (САУ) или автопилота. С точки зрения организации вычислительного процесса, использования вычислительных ресурсов и формирования информационных потоков архитектура А380 интересна использованием принципов ИМА и сетевых технологий (рис. 1.11).

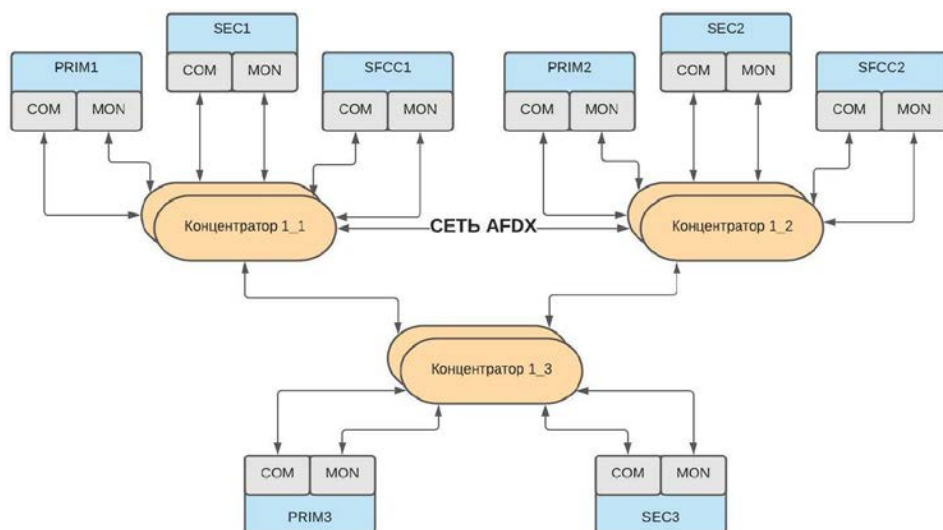


Рис. 1.11. Вычислительная подсистема СУ самолета Airbus A380

Основной режим управления реализуется через три основных вычислителя PRIM (PRIMary Flight Control and Guidance Computer), которые реализуют функции как углового, так и траекторного управления. В качестве аварийной СУ на самолете A380 используется резервная система, включающая прямую связь от ручки управления, дополненную сигналами демпфирования тангажа и рыскания и сигналами триммирования. Эта система реализована с помощью трех вычислителей SEC (SECondary Flights Control Computer). Два вычислителя SFCC (Slats & Flaps Control Computer) управляют механизацией крыла.

Ключевым преимуществом архитектуры ВП СУ ВС на базе концепции ИМА является возможность повышения ее отказоустойчивости за счет перераспределения функций между отдельными ПУ в случае возникновения отказов. Такую процедуру перераспределения функций в дальнейшем будем называть реконфигурацией ВП СУ ВС. Именно разработке методов реконфигурации ВП СУ ВС с архитектурой ИМА будет посвящена 3-я глава монографии.

Глава 2

РЕКОНФИГУРАЦИЯ ИСПОЛНИТЕЛЬНЫХ ПОДСИСТЕМ СИСТЕМ УПРАВЛЕНИЯ ВОЗДУШНЫХ СУДОВ

2.1. Способы и методы реконфигурации исполнительных подсистем

СУ с реконфигурацией реагируют на отказы системных компонентов путем изменения управляющих воздействий так, чтобы показатели устойчивости и управляемости ВС в целом оставались приемлемыми. В таких СУ компенсируется любой отказ заранее рассчитанным законом управления или синтезированным в режиме реального времени новым законом управления. При этом при реконфигурации пересчитываются не только параметры законов управления, но и структура реконфигурируемой СУ. Поэтому соответствующие системы иногда называют реструктурируемыми СУ [97, 113], подчеркивая этим то, что в СУ меняется структура управления.

Рассмотрим известные способы и методы, позволяющие обеспечить реконфигурацию управления ВС.

В целом способы реконфигурации ИП СУ можно разделить на две большие группы: пассивные и активные. В пассивных СУ с реконфигурацией отклонения параметров от их истинных значений или отклонения приводов от их ожидаемых позиций могут эффективно компенсироваться заранее рассчитанными жесткими отказоустойчивыми контроллерами в обратной связи [52, 54, 83]. Этот способ не нуждается в схемах диагностики отказов и настройки законов реконфигурации по режимам полета. Однако если эти отклонения становятся чрезмерно большими и превышают допустимые пределы, то свойство отказоустойчивости СУ теряется. Поэтому более предпочтительными являются активные отказоустойчивые СУ, включающие в свой состав алгоритмы диагностирования отказов.

Классификация широко известных подходов и методов реконфигурации ИП СУ приведена на рисунке 2.1 и в таблице 2.1 [73, 97].

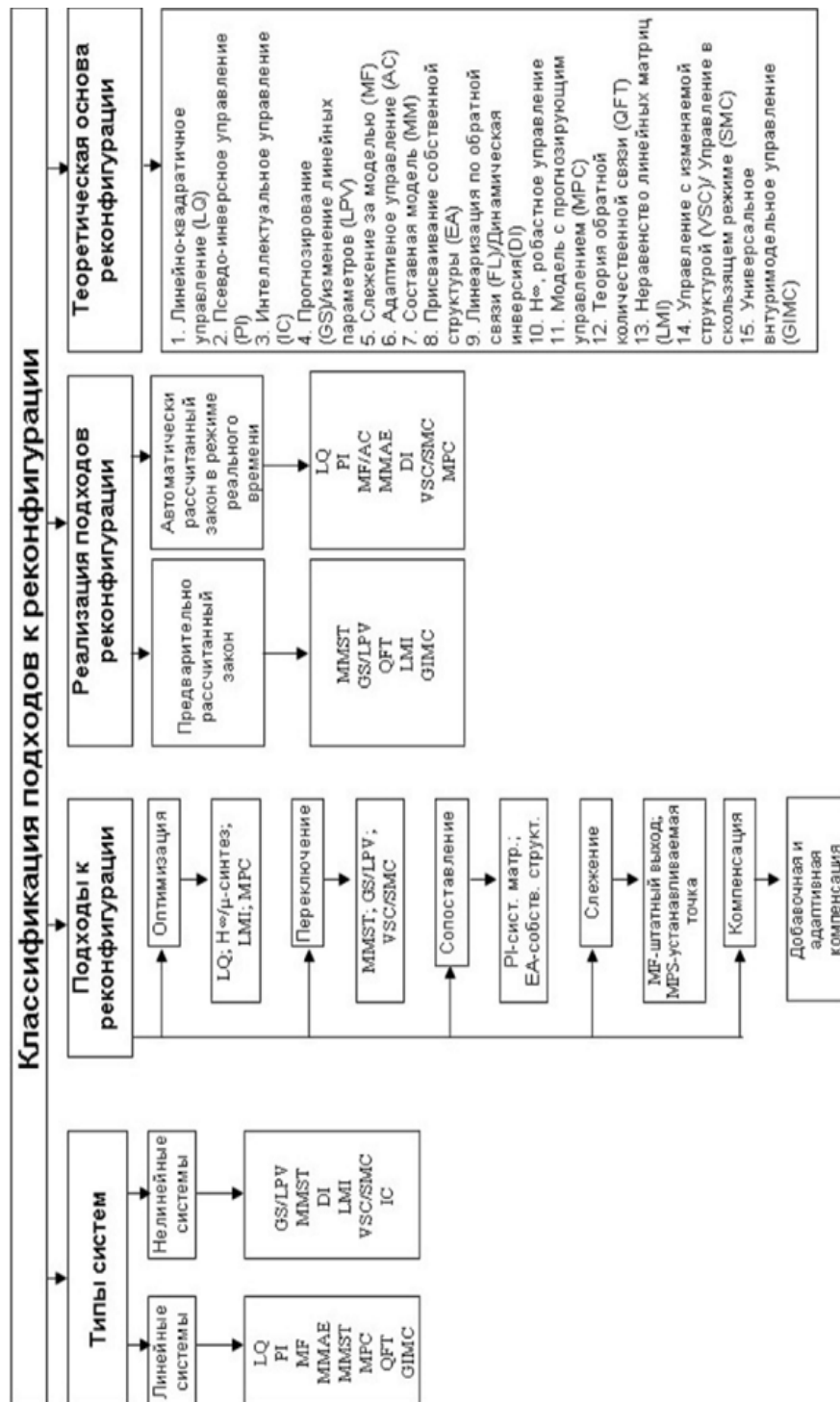


Рис. 2.1. Классификация подходов и методов реконфигурации управления

Табл. 2.1. Методы реконфигурации управления

Разработанные подходы	Источник
Псевдоинверсное управление (PI)	[56, 70, 110]
Линейно-квадратичное управление (LQ)	[80, 109]
Интеллектуальное управление (IC)	[63, 64, 72, 81]
Прогнозирование (GS) / изменение линейных параметров (LPV)	[66, 95]
Слежение за моделью (MF)	[76, 112]
Метод адаптивной составной модели (MMAE)	[84, 100]
Составная модель с переключением и настройкой (MMST)	[61]
Интегрированное диагностирование и управление (EA)	[70, 87]
Управление собственными значениями (EA)	[77, 112, 113]
Адаптивное управление с эталонной моделью (MRAC)	[59, 101]
Робастное управление	[74, 108]
Модель с прогнозирующим управлением (MPC)	[74, 82]
Теория обратной количественной связи (QFT)	[88]
Матричные линейные неравенства (LMI)	[79]
Управление с изменяемой структурой / Управление в скользящем режиме (VSC/SMC)	[79, 116]
Распределенное управление (AC)	[58, 71]
Универсальное внутримодельное управление (GIMC)	[107]
Общеархитектурный и прочие	[91]

В целом все известные способы реконфигурации в современной теории отказоустойчивого управления можно разделить на две большие группы: не изменяющие и изменяющие штатную структуру СУ. При неизменной структуре СУ предполагается полное изменение штатных настроек СУ, что для современных сложных систем практически не реализуемо. Изменение структуры СУ при реконфигурации предполагает введение в контур управления дополнительных модулей, обеспечивающих компенсацию последствий отказов СУ.

Анализ показывает, что основными недостатками широко известных методов реконфигурации ИП СУ ВС являются:

- решение численное, требующее пересчета значений коэффициентов при изменении условий полета;
- решение единственное, что не позволяет учесть дополнительные факторы, условия и требования;
- реализация алгоритмов в бортовом вычислителе требует выполнения трудоемких матричных операций;
- сложность настройки коэффициентов матрицы реконфигурации по режимам полета;

– учет конструктивных ограничений на отклонение поверхностей управления осуществляется сложными нелинейными методами, не позволяющими обеспечить детерминированность трудоемких вычислений.

На наш взгляд, именно эти недостатки не позволяют реализовать концепцию реконфигурации в СУ современных ВС на практике. В этой связи в монографии формулируется новый способ реконфигурации ИП СУ ВС аналитическими методами, позволяющий получать простые аналитические законы реконфигурации, коэффициенты которых непосредственно настраиваются по режимам полета. Описываются оптимальные решения, минимизирующие как норму ошибки реконфигурации, так и норму самого решения, одновременно обеспечивая при этом максимальное сохранение стереотипа управления, минимальное изменение собственной динамики ВС с замкнутой СУ и минимально возможные компенсирующие отклонения работоспособных поверхностей управления, задействованных при реконфигурации. Оптимальные методы положены в основу получения робастных законов реконфигурации, сохраняющих удовлетворительные характеристики управления ВС при наличии параметрических или структурных неопределенностей. Приведена оригинальная методика реконфигурации ИП СУ ВС с учетом ограничений на отклонения поверхностей управления, позволяющая получать реализуемые на практике законы реконфигурации.

В основу методов реконфигурации ИП СУ ВС положены разработанные авторами специальные процедуры решения линейных матричных уравнений, позволяющие описывать множество всех возможных решений в аналитическом виде, получать оптимальные и субоптимальные решения поставленных задач. Все задачи реконфигурации ИП СУ иллюстрируются примерами решения с использованием символьных и числовых математических моделей реальных ВС.

2.2. Точное аналитическое решение задачи реконфигурации исполнительной подсистемы

2.2.1. Постановка задачи реконфигурации исполнительной подсистемы

Сформулируем задачу реконфигурации ИП СУ ВС следующим образом. Представим динамику полета ВС с работоспособной СУ в виде непрерывной нелинейной нестационарной модели с линейно входящим управлением [42]

$$\dot{x} = f(x, a, t) + B(x, a, t)u, \quad (2.1)$$

где x – вектор состояния ВС размерности n ; u – вектор управления размерности m ; $f(x, a, t)$ – векторная функция векторных аргументов x , a , и времени t , $B(x, a, t)$ – матричная функция эффективности управления.

Без потери общности в целях вывода расчетных соотношений перейдем к линейной нестационарной модели вида

$$\dot{x} = A(x, a, t)x + B(x, a, t)u, \quad (2.2)$$

где $A(x, a, t)$ – матричная функция собственной динамики ВС размера $n \times n$ (в дальнейшем покажем, что этот переход не является обязательным). Предполагается, что зависимости элементов этих матриц от параметров ВС известны. Помимо этого примем, что на интервале времени переходных процессов матричные функции $A(x, a, t)$ и $B(x, a, t)$ можно заменить постоянными матрицами A и B , а после получения результата обобщим подход на нестационарные и нелинейные объекты управления вида (2.1).

Тогда стационарная линейная модель динамики полета ВС с работоспособной СУ в пространстве состояний будет иметь вид

$$\dot{x} = Ax + Bu, \quad (2.3)$$

где A, B – матрицы собственной динамики и эффективности управления ВС.

В замкнутом виде, не конкретизируя структуру штатной СУ, замкнутый контур управления «воздушное судно – работоспособная СУ» можно представить схемой, изображенной на рисунке 2.2.

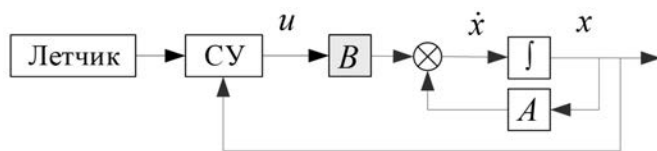


Рис. 2.2. Схема замкнутого контура управления без отказов

Введем в рассмотрение модель ВС с отказом ИП СУ

$$\dot{x}_f = Ax_f + Bu_f, \quad (2.4)$$

где x_f, u_f – векторы состояния и управления ВС с отказом в СУ.

Будем говорить о потере эффективности поверхностей управления. В этом случае можно записать

$$u_f = Fu, \quad (2.5)$$

где F – матрица отказов ИП

$$F = \text{diag}(f_1 \quad \dots \quad f_k \quad \dots \quad f_n), \quad (2.6)$$

коэффициенты которой характеризуют отказ k -го канала: $f_k = 1$ – отказа нет, $0 \leq f_k < 1$ – отказ есть. При $f_k = 0$ сигнал управления $u_k = 0$, что соответствует заклиниванию поверхности управления в балансировочном (нулевом) положении. В общем случае $0 \leq f_k < 1$, что соответствует потере эффективности k -го органа управления при сохранении его работоспособности.

Подставим (2.5) в (2.4)

$$\dot{x}_f = Ax_f + BFu$$

и запишем модель ВС с отказом ИП СУ в виде

$$\dot{x}_f = Ax_f + B_f u, \quad (2.7)$$

где

$$B_f = BF \quad (2.8)$$

– матрица эффективности управления ВС с отказом одного или нескольких каналов управления СУ.

Представим модель (2.7) с учетом (2.8) в виде схемы, изображенной на рисунке 2.3.

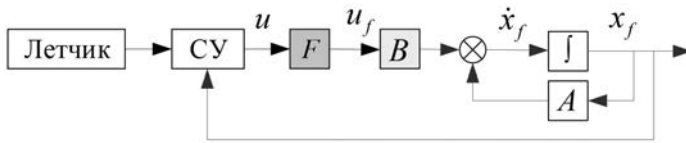


Рис. 2.3. Схема замкнутого контура управления с отказом в СУ

Введем управление u_p , обеспечивающее реконфигурацию в случае отказов в СУ. В этом случае модель замкнутого контура управления с отказом в СУ и реконфигурацией примет вид

$$\dot{x}_p = Ax_p + B_f u_p, \quad (2.9)$$

где управление u_p будем искать в виде суммы штатного u и компенсирующего u_k

$$u_p = u + u_k, \quad (2.10)$$

которое также зависит от штатного

$$u_k = Hu.$$

Тогда реконфигурированный сигнал управления будет определяться выражением

$$u_p = (I + H)u. \quad (2.11)$$

Именно за счет матрицы реконфигурации H и достигается перераспределение управления с отказавших каналов управления на работоспособные.

Подставим (2.11) в (2.9), тогда модель замкнутого контура управления с отказом в СУ и реконфигурацией примет вид

$$\dot{x}_p = Ax_p + B_f (I + H)u = Ax_p + B_p u, \quad (2.12)$$

где $B_p = B_f (I + H)$ – матрица эффективности управления ВС с отказом и реконфигурацией (рисунок 2.4).

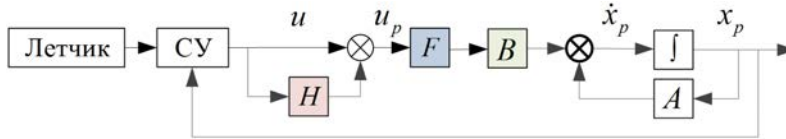


Рис. 2.4. Модель замкнутого контура управления с отказом в СУ и реконфигурацией

Целью реконфигурации управления при отказе ИП является определение матрицы реконфигурации H , обеспечивающей полное совпадение параметров моделей ВС с работоспособной СУ (2.3) и модели с отказом в СУ и реконфигурацией (2.12).

Приравнявая (2.3) и (2.12) при условии совпадения момента отказа с моментом подачи компенсирующего управления, когда $x_p(t_0) = x(t_0)$, получим

$$B_p = B_f(I + H) = B_f + B_f H = B. \quad (2.13)$$

Преобразуем (2.13) к виду

$$\underbrace{B_f}_{A^*} \underbrace{H}_{X^*} = \underbrace{\Delta B_f}_{B^*}, \quad (2.14)$$

где $\Delta B_f = B - B_f$. В результате задача реконфигурации ИП СУ принимает вид линейного матричного уравнения (2.14) относительно неизвестной матрицы реконфигурации H .

Для решения полученного уравнения сформулируем специальную процедуру построения множеств эквивалентных решений линейных матричных уравнений в аналитическом виде.

2.2.2. Аналитическое решение линейных матричных уравнений методом канонизации

Существует мнение [8], что три четверти всех расчетных математических задач приходится на решение линейных матричных уравнений. При этом можно утверждать, что все известные способы решения таких уравнений представляют собой модификации одного из двух способов, основанных либо на вычислении значений определителей, либо на эквивалентных (в частности, ортогональных) преобразованиях матриц коэффициентов решаемого уравнения.

Типичным способом первой группы является правило Крамера. Основное преимущество этой группы способов заключается в возможности аналитического исследования решения. Однако с точки зрения вычислительных затрат данные способы далеко не самые экономичные и практическое применение находят только при решении вручную матричных уравнений невысокого размера [9].

Способы второй группы характеризуются более высокой вычислительной эффективностью, благодаря чему они доминируют в процедурах машинного решения уравнений. К таким способам относятся приведенные в [46] программы, реализующие метод вращения Якоби и модификации QR-алгоритма. В то же время эти способы не применимы для аналитических исследований свойств получаемого решения.

В данной работе предлагается метод решения линейных матричных уравнений, сочетающий достоинства каждого из указанных подходов. С одной стороны, он относится к способам второй группы, так как опирается на алгоритм Гаусса. С другой стороны, он является аналитическим, т.е. позволяет проводить исследования получаемого решения в символьном виде, зачастую даже в более компактном представлении, нежели это осуществляется в способах первой группы. При этом предлагаемый метод достаточно прост как в применении, так и в переложении на язык компьютерных программ.

Рассмотрим произвольную матрицу A размера $m \times n$ ранга r . Одно из центральных мест в дальнейших построениях будут занимать, так называемые, матричные делители нуля и единицы матрицы A .

Определение 1. Прямоугольная матрица \bar{A}^L , содержащая $w = m - r > 0$ строк и m столбцов, одновременно удовлетворяющая тождеству и двум неравенствам

$$\bar{A}^L A = 0_{w \times n}, \quad A \neq 0_{m \times n}, \quad \bar{A}^L \neq 0_{w \times m}, \quad (2.15)$$

называется левым делителем нуля полного ранга матрицы A .

Аналогично с учетом симметрии вводится правый делитель нуля \bar{A}^R .

Определение 2. Прямоугольная матрица \bar{A}^R , содержащая n строк и $q = n - r > 0$ столбцов, одновременно удовлетворяющая тождеству и двум неравенствам

$$A \bar{A}^R = 0_{m \times q}, \quad A \neq 0_{m \times n}, \quad \bar{A}^R \neq 0_{n \times q}, \quad (2.16)$$

называется правым делителем нуля полного ранга матрицы A .

Если подобрать матрицы, удовлетворяющие (2.16) или (2.15), невозможно, то у матрицы A отсутствуют соответствующие делители нуля. Этот факт отражается записью $\bar{A}^L = 0_{0 \times m}$ или $\bar{A}^R = 0_{n \times 0}$, полагая, что делители нуля представляют собой матрицы с нулевым числом столбцов.

Определение 3. Прямоугольные матрицы \tilde{A}^L и \tilde{A}^R ранга r , одновременное умножение на которые слева и справа по формуле

$$\tilde{A}^L_{r \times m} A_{m \times n} \tilde{A}^R_{n \times r} = I_r$$

приводит матрицу A ранга r к единичной матрице I_r , называются соответственно левым и правым делителями единицы.

Определение 4. Матрица размера $n \times m$, образованная для прямоугольной матрицы A размера $n \times m$ и ранга r произведением ее правого и левого делителей единицы по формуле

$$\tilde{A}_{n \times m} = \tilde{A}_{n \times r}^R \tilde{A}_{r \times m}^L,$$

называется сводным делителем единицы (канонизатором) матрицы A или обобщенно-обратной матрицей, так как она удовлетворяет условиям регулярности

$$\tilde{A} \tilde{A} \tilde{A} = \tilde{A}, \quad A \tilde{A} A = A. \quad (2.17)$$

В соответствии с [38] выбором не обязательно единственных матриц преобразования базисов T_L и T_R любая матрица A может быть представлена в виде канонического разложения

$$A = T_L^{-1} E_A T_R^{-1} = T_L^{-1} \begin{bmatrix} I_r & 0_{r \times (n-r)} \\ 0_{(m-r) \times r} & 0_{(m-r) \times (n-r)} \end{bmatrix} T_R^{-1}, \quad (2.18)$$

где I_r – единичная матрица размера $r \times r$, а центральная матрица в правой части называется матрицей в канонических базисах или просто каноническим базисом.

Каноническое разложение (2.18) фактически формализует прямые и обратные эквивалентные преобразования строк и столбцов матрицы, которые легко могут быть определены аналитически. Изложим эффективный способ получения канонического разложения в аналитическом виде. Для этого дополним матрицу A двумя единичными матрицами слева и снизу так, что получается конструкция типа планшета

$$\frac{I_m}{I_n} \left| \frac{A_{m \times n}}{I_n} \right., \quad (2.19)$$

формально не являющаяся матрицей.

Далее выполняются элементарные преобразования строк и столбцов матрицы A с одновременным соответствующим преобразованием прилегающих единичных матриц. Цель преобразований заключается в приведении матрицы A к записи в каноническом базисе вида

$$\frac{T_L}{T_R} \left| \frac{E_A}{T_R} \right. = \frac{\begin{bmatrix} \tilde{A}_{r \times m}^L \\ \bar{A}_{(m-r) \times m}^L \end{bmatrix}}{\begin{bmatrix} \tilde{A}_{n \times r}^R \\ \bar{A}_{n \times (n-r)}^R \end{bmatrix}} \left| \begin{bmatrix} I_r & 0_{r \times (n-r)} \\ 0_{(m-r) \times r} & 0_{(m-r) \times (n-r)} \end{bmatrix} \right. \quad (2.20)$$

Тогда блоки первоначально единичных матриц напротив нулевых блоков канонической матрицы E_A , стоящей в правом верхнем углу выражения

(2.20), будут содержать делители нуля полного ранга. Левее матрицы E_A , будет расположен левый делитель, а ниже – правый делитель. Отсутствие нулевых строк и/или нулевых столбцов у матрицы E_A говорит о том, что матрица A не имеет левых и/или правых делителей нуля. Матрицы, стоящие слева и внизу конструкции (2.20) напротив единичного блока матрицы E_A содержат соответствующие делители единицы $\tilde{A}_{r \times m}^L$ и $\tilde{A}_{n \times r}^R$ максимального ранга, равного рангу r исходной матрицы A . При этом всегда обратимые блочные матрицы

$$T_L = \begin{bmatrix} \tilde{A}^L \\ \bar{A}^L \end{bmatrix}, T_R = \begin{bmatrix} \tilde{A}^R & \bar{A}^R \end{bmatrix}$$

называются левыми и правыми канонизаторами соответственно.

Таким образом, любой матрице A можно сопоставить в общем случае неединственную тройку матриц, включающую левый \bar{A}^L и правый \bar{A}^R делители нуля, а также сводный канонизатор \tilde{A} , т.е.

$$A \rightarrow (\bar{A}^L, \tilde{A}, \bar{A}^R). \quad (2.21)$$

Представление (2.21) называется канонизацией матрицы A . Оно значительно информативнее ранговых критериев (типа Кронекера-Капели), использующих операции вычисления определителей, так как полностью характеризует «внутреннюю структуру» матрицы A :

- все линейно зависимые комбинации строк;
- все линейно зависимые комбинации столбцов;
- совокупность линейно независимых комбинаций строк и столбцов.

На основании вышеприведенного сформулируем далее теорему об аналитическом решении матричного уравнения методом канонизации [5, 6, 19].

Теорема 2.1. Линейное матричное уравнение вида

$$AX = B, \quad (2.22)$$

где A и B – заданные числовые или символьные матрицы параметров (коэффициентов), X – матрица с неизвестными элементами, разрешимо тогда и только тогда, когда выполняется условие разрешимости

$$\bar{A}^L B = 0, \quad (2.23)$$

при этом все множество решений описывается формулой

$$X = \begin{bmatrix} \tilde{A}^R & \bar{A}^R \end{bmatrix} \begin{bmatrix} \tilde{A}^L B \\ \Psi \end{bmatrix} = \tilde{A} B + \bar{A}^R \Psi, \quad (2.24)$$

где Ψ – произвольная матрица подходящего размера. □

Доказательство. Запишем каноническое разложение матрицы A :

$$A = \begin{bmatrix} \tilde{A}^L \\ \bar{A}^L \end{bmatrix}^{-1} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{A}^R & \bar{A}^R \end{bmatrix}^{-1},$$

подстановка которого в искомое уравнение (2.22) приведет его к виду

$$\begin{bmatrix} \tilde{A}^L \\ \bar{A}^L \end{bmatrix}^{-1} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{A}^R & \bar{A}^R \end{bmatrix}^{-1} X = B. \quad (2.25)$$

Умножим слева на $\begin{bmatrix} \tilde{A}^L \\ \bar{A}^L \end{bmatrix}$ и введем обозначение

$$\begin{bmatrix} \tilde{A}^R & \bar{A}^R \end{bmatrix}^{-1} X = \begin{bmatrix} \Omega \\ \Psi \end{bmatrix}. \quad (2.26)$$

Тогда (2.25) примет вид уравнения

$$\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Omega \\ \Psi \end{bmatrix} = \begin{bmatrix} \tilde{A}^L \\ \bar{A}^L \end{bmatrix} B, \quad (2.27)$$

из которого можно в явном виде записать условие (2.23) и выражение для неизвестной $\Omega = \tilde{A}^L B$. На матрицу Ψ ограничений не накладывается, тогда (2.26) примет вид

$$\begin{bmatrix} \tilde{A}^R & \bar{A}^R \end{bmatrix}^{-1} X = \begin{bmatrix} \tilde{A}^L B \\ \Psi \end{bmatrix}, \quad (2.28)$$

а искомая формула (2.24) получается при умножении выражения (2.28) слева на $\begin{bmatrix} \tilde{A}^R & \bar{A}^R \end{bmatrix}$.

Приведенный метод решения линейных матричных уравнений называется методом канонизации. Его применение подразумевает две неперенные составляющие: проверку условия разрешимости уравнения (2.23) и в случае его выполнения построение всего множества решений (2.24), представляющего собой сумму двух слагаемых:

- частного решения $\tilde{A}B$ неоднородного уравнения (2.22);
- общего решения $\bar{A}^R \Psi$ однородного уравнения $AX = 0$ с минимальной параметризацией, удовлетворяющего исходному уравнению (2.22) при нулевом значении правой части $B = 0$.

2.2.3. Решение задачи реконфигурации методом канонизации

Воспользуемся результатами теоремы 2.1 и сформулируем теорему о решении задачи реконфигурации системы управления методом канонизации.

Теорема 2.2. Задача реконфигурации вида

$$B_f H = \Delta B_f, \quad (2.29)$$

разрешима тогда и только тогда, когда выполняется условие реконфигурируемости

$$\bar{B}_f^L \Delta B = \bar{B}_f^L B = 0, \quad (2.30)$$

при этом все множество решений описывается формулой

$$H = \tilde{B}_f \Delta B + \bar{B}_f^R \Psi, \quad (2.31)$$

где Ψ – произвольная матрица, \bar{B}_f^L , \bar{B}_f^R – левый и правый делители нуля полного ранга, удовлетворяющие тождествам

$$\bar{B}_f^L B_f = 0, \quad B_f \bar{B}_f^R = 0, \quad (2.32)$$

$\tilde{B}_f = \tilde{B}_f^R \tilde{B}_f^L$ – обобщенно-обратная матрица, а каноническое разложение матрицы B_f имеет вид

$$B_f = \begin{bmatrix} \tilde{B}_f^L \\ \bar{B}_f^L \end{bmatrix}^{-1} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{B}_f^R & \bar{B}_f^R \end{bmatrix}^{-1}. \quad (2.33)$$

Доказательство теоремы 2.2 вытекает из доказательства теоремы 2.1. Здесь только заметим, что условие разрешимости (2.30) обусловлено тождеством по определению левого делителя нуля $\bar{B}_f^L B_f = 0$.

Использование (2.31) для расчета матрицы реконфигурации позволяет обеспечить компенсацию последствий отказа. Аналитическое представление (2.33), полученное исключительно с применением эквивалентных преобразований матрицы B_f , допускает обобщение результата на модели вида (2.1), где в качестве коэффициентов матричных функций могут быть задействованы символьные переменные, предполагающие различные формы записи и учитывающие зависимости коэффициентов от параметров x, a, t

$$H(x, a, t) = \widetilde{B_f(x, a, t)} (B(x, a, t) - B_f(x, a, t)) + \overline{B_f(x, a, t)}^R \Psi. \quad (2.34)$$

Заметим, что векторная функция $f(x, a, t)$ в расчетных соотношениях не участвует. Соотношение (2.34) имеет аналитический вид, и на его основе могут быть синтезированы упрощенные схемы реализации предложенного алгоритма, позволяющие реализовать реконфигурацию в виде классических структурных схем. Наиболее просто это можно показать на примере. В этой связи рассмотрим пример решения задачи реконфигурации на основе метода канонизации.

2.2.4. Пример решения задачи реконфигурации методом канонизации

Рассмотрим возможность применения описанного подхода к решению задачи реконфигурации ИП СУ самолета, модель динамики полета которого описывается следующей системой дифференциальных уравнений [7, 16, 25]

$$\begin{aligned}
 \frac{d\omega_x}{dt} &= \bar{M}_x^{\omega_x} \omega_x + \bar{M}_x^{\omega_y} \omega_y + \bar{M}_x^{\beta} \beta + \\
 &\quad - \bar{M}_x^{\varphi_{\text{пр}}} \varphi_{\text{пр}} + \bar{M}_x^{\varphi_{\text{л}}} \varphi_{\text{л}} - \bar{M}_x^{\delta_{\text{ф.пр}}} \delta_{\text{ф.пр}} + \bar{M}_x^{\delta_{\text{ф.л}}} \delta_{\text{ф.л}} - \bar{M}_x^{\delta_{\text{н.пр}}} \delta_{\text{н.пр}} - \bar{M}_x^{\delta_{\text{н.л}}} \delta_{\text{н.л}}, \\
 \frac{d\omega_y}{dt} &= \bar{M}_y^{\omega_x} \omega_x + \bar{M}_y^{\omega_y} \omega_y + \bar{M}_y^{\beta} \beta + \\
 &\quad + \bar{M}_y^{\varphi_{\text{пр}}} \varphi_{\text{пр}} - \bar{M}_y^{\varphi_{\text{л}}} \varphi_{\text{л}} + \bar{M}_y^{\delta_{\text{ф.пр}}} \delta_{\text{ф.пр}} - \bar{M}_y^{\delta_{\text{ф.л}}} \delta_{\text{ф.л}} - \bar{M}_y^{\delta_{\text{н.пр}}} \delta_{\text{н.пр}} - \bar{M}_y^{\delta_{\text{н.л}}} \delta_{\text{н.л}}, \\
 \frac{d\omega_z}{dt} &= \bar{M}_z^{\omega_z} \omega_z + \bar{M}_z^{\alpha} \alpha - \bar{M}_z^{\varphi} \left(\frac{\varphi_{\text{пр}} + \varphi_{\text{л}}}{2} \right) - \bar{M}_z^{\delta_{\text{ф}}} \left(\frac{\delta_{\text{ф.пр}} + \delta_{\text{ф.л}}}{2} \right) + \bar{M}_z^{\delta_{\text{н.г.о}}} \delta_{\text{н.г.о}}, \\
 \frac{d\alpha}{dt} &= \omega_z - \bar{Y}^{\alpha} \alpha - \left(\bar{Y}^{\varphi} \left(\frac{\varphi_{\text{пр}} + \varphi_{\text{л}}}{2} \right) + \bar{Y}^{\delta_{\text{ф}}} \left(\frac{\delta_{\text{ф.пр}} + \delta_{\text{ф.л}}}{2} \right) + \bar{Y}^{\delta_{\text{н.г.о}}} \delta_{\text{н.г.о}} \right), \\
 \frac{d\beta}{dt} &= \alpha_0 \sin \omega_x + \omega_y + \bar{Z}^{\beta} \beta + \frac{g}{V} \sin \gamma \cos \vartheta - \bar{Z}^{\delta_{\text{н.пр}}} \delta_{\text{н.пр}} - \bar{Z}^{\delta_{\text{н.л}}} \delta_{\text{н.л}}, \\
 \frac{d\gamma}{dt} &= \omega_x - \alpha_0 \omega_y, \\
 \frac{d\vartheta}{dt} &= \omega_y \sin \gamma + \omega_z \cos \gamma, \quad \vartheta(0) = \alpha_0,
 \end{aligned} \tag{2.35}$$

где $\omega_x, \omega_y, \omega_z$ – угловые скорости крена, рыскания и тангажа; $\alpha, \vartheta, \gamma, \beta$ – углы атаки, тангажа, крена и скольжения; V – скорость полета; g – ускорение свободного падения; α_0 – балансирующее значение угла атаки; $\varphi_{\text{пр}}, \varphi_{\text{л}}, \delta_{\text{ф.пр}}, \delta_{\text{ф.л}}, \delta_{\text{н.пр}}, \delta_{\text{н.л}}, \delta_{\text{н.г.о}}$ – углы отклонения от балансирующего положения правых и левых стабилизаторов, флаперонов, рулей направления и переднего горизонтального оперения соответственно. Рассматриваемая модель описывает детерминированную непрерывную систему с линейно входящим управлением.

Выделим векторы состояния x и управления u

$$\tilde{o} = [\omega_x \omega_y \omega_z \alpha \beta \gamma \vartheta]^T, u = [\varphi_{\text{пр}} \varphi_{\text{л}} \delta_{\text{ф.пр}} \delta_{\text{ф.л}} \delta_{\text{н.пр}} \delta_{\text{н.л}} \delta_{\text{про}}]^T, \quad (2.36)$$

тогда модель динамики полета самолета будет описываться выражением

$$\dot{x} = A(x)x + Bu, \quad (2.37)$$

где матрица эффективности управления имеет вид

$$B = \begin{bmatrix} -\bar{M}_x^{\varphi_{\text{пр}}} & \bar{M}_x^{\varphi_{\text{л}}} & -\bar{M}_x^{\delta_{\text{ф.пр}}} & \bar{M}_x^{\delta_{\text{ф.л}}} & -\bar{M}_x^{\delta_{\text{н.пр}}} & -\bar{M}_x^{\delta_{\text{н.л}}} & 0 \\ \bar{M}_y^{\varphi_{\text{пр}}} & -\bar{M}_y^{\varphi_{\text{л}}} & \bar{M}_y^{\delta_{\text{ф.пр}}} & -\bar{M}_y^{\delta_{\text{ф.л}}} & -\bar{M}_y^{\delta_{\text{н.пр}}} & -\bar{M}_y^{\delta_{\text{н.л}}} & 0 \\ -\frac{1}{2}\bar{M}_z^{\varphi_{\text{пр}}} & -\frac{1}{2}\bar{M}_z^{\varphi_{\text{л}}} & -\frac{1}{2}\bar{M}_z^{\delta_{\text{ф.пр}}} & -\frac{1}{2}\bar{M}_z^{\delta_{\text{ф.л}}} & 0 & 0 & \bar{M}_z^{\delta_{\text{про}}} \\ -\frac{1}{2}\bar{Y}^{\varphi_{\text{пр}}} & -\frac{1}{2}\bar{Y}^{\varphi_{\text{л}}} & -\frac{1}{2}\bar{Y}^{\delta_{\text{ф.пр}}} & -\frac{1}{2}\bar{Y}^{\delta_{\text{ф.л}}} & 0 & 0 & -\bar{Y}^{\delta_{\text{про}}} \\ 0 & 0 & 0 & 0 & -\bar{Z}^{\delta_{\text{н.пр}}} & -\bar{Z}^{\delta_{\text{н.л}}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (2.38)$$

Так как коэффициенты имеют одинаковые значения в парных каналах, количество различающихся параметров может быть сокращено

$$B = \begin{bmatrix} -\bar{M}_x^{\varphi} & \bar{M}_x^{\varphi} & -\bar{M}_x^{\delta_{\text{ф}}} & \bar{M}_x^{\delta_{\text{ф}}} & \bar{M}_x^{\delta_{\text{н}}} & \bar{M}_x^{\delta_{\text{н}}} & 0 \\ -\bar{M}_y^{\varphi} & \bar{M}_y^{\varphi} & -\bar{M}_y^{\delta_{\text{ф}}} & \bar{M}_y^{\delta_{\text{ф}}} & \bar{M}_y^{\delta_{\text{н}}} & \bar{M}_y^{\delta_{\text{н}}} & 0 \\ \frac{1}{2}\bar{M}_z^{\varphi} & \frac{1}{2}\bar{M}_z^{\varphi} & \frac{1}{2}\bar{M}_z^{\delta_{\text{ф}}} & \frac{1}{2}\bar{M}_z^{\delta_{\text{ф}}} & 0 & 0 & \bar{M}_z^{\delta_{\text{про}}} \\ -\frac{1}{2}\bar{Y}^{\varphi} & -\frac{1}{2}\bar{Y}^{\varphi} & -\frac{1}{2}\bar{Y}^{\delta_{\text{ф}}} & -\frac{1}{2}\bar{Y}^{\delta_{\text{ф}}} & 0 & 0 & -\bar{Y}^{\delta_{\text{про}}} \\ 0 & 0 & 0 & 0 & \bar{Z}^{\delta_{\text{н}}} & \bar{Z}^{\delta_{\text{н}}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (2.39)$$

Заметим, что коэффициенты матрицы эффективности управления представлены в символьном (аналитическом) виде и в общем случае могут изменять свои значения в зависимости, например, от режима полета и времени.

Более того, при переходе на сверхзвуковой режим полета помимо параметров изменяется структура матрицы эффективности управления

$$B = \begin{bmatrix} -\bar{M}_x^\varphi & \bar{M}_x^\varphi & -\bar{M}_x^{\delta_\Phi} & \bar{M}_x^{\delta_\Phi} & \bar{M}_x^{\delta_n} & \bar{M}_x^{\delta_n} & 0 \\ -\bar{M}_y^\varphi & \bar{M}_y^\varphi & -\bar{M}_y^{\delta_\Phi} & \bar{M}_y^{\delta_\Phi} & \bar{M}_y^{\delta_n} & \bar{M}_y^{\delta_n} & 0 \\ \frac{1}{2}\bar{M}_z^\varphi & \frac{1}{2}\bar{M}_z^\varphi & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2}\bar{Y}^\varphi & -\frac{1}{2}\bar{Y}^\varphi & -\frac{1}{2}\bar{Y}^{\delta_\Phi} & -\frac{1}{2}\bar{Y}^{\delta_\Phi} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \bar{Z}^{\delta_n} & \bar{Z}^{\delta_n} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (2.40)$$

так как обнуляются коэффициенты эффективности управления ПГО и флаперонов при управлении в канале угловой скорости тангажа, непосредственное управление которой становится возможным только с помощью отклонения стабилизатора.

Рассмотрим для примера отказ правого стабилизатора на дозвуковом режиме полета, когда матрица отказов (2.6) имеет вид $F = (0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1)$. Запишем с учетом (2.39), (2.8) матрицу эффективности управления с отказавшим приводом правого стабилизатора

$$B_f = \begin{bmatrix} 0 & \bar{M}_x^\varphi & -\bar{M}_x^{\delta_\Phi} & \bar{M}_x^{\delta_\Phi} & \bar{M}_x^{\delta_n} & \bar{M}_x^{\delta_n} & 0 \\ 0 & \bar{M}_y^\varphi & -\bar{M}_y^{\delta_\Phi} & \bar{M}_y^{\delta_\Phi} & \bar{M}_y^{\delta_n} & \bar{M}_y^{\delta_n} & 0 \\ 0 & \frac{1}{2}\bar{M}_z^\varphi & \frac{1}{2}\bar{M}_z^{\delta_\Phi} & \frac{1}{2}\bar{M}_z^{\delta_\Phi} & 0 & 0 & \bar{M}_z^{\delta_{\text{про}}} \\ 0 & -\frac{1}{2}\bar{Y}^\varphi & -\frac{1}{2}\bar{Y}^{\delta_\Phi} & -\frac{1}{2}\bar{Y}^{\delta_\Phi} & 0 & 0 & -\bar{Y}^{\delta_{\text{про}}} \\ 0 & 0 & 0 & 0 & \bar{Z}^{\delta_n} & \bar{Z}^{\delta_n} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.41)$$

и определим делители нуля матрицы (2.41), удовлетворяющие тождествам (2.32)

$$\bar{B}_f^R = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \quad (2.42)$$

$$\bar{B}_f^R = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T. \quad (2.43)$$

Левый делитель нуля (2.42) матрицы B_f совпадает с левым делителем нуля матрицы B , поэтому задача реконфигурации в данном случае в соответствии с (2.30) разрешима.

Определим далее в соответствии с (2.31) все множество матриц реконфигурации, позволяющих компенсировать отказ правого стабилизатора

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{h_{\varphi}^{\text{про}}}{h_{\Phi}^{\text{про}}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{h_{\varphi}^{\text{про}}}{h_{\Phi}^{\text{про}}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{h_{\varphi}^{\Phi}}{h_{\Phi}^{\text{про}}} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ -1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mu_{11} & \mu_{12} & \mu_{13} & \mu_{14} & \mu_{15} & \mu_{16} & \mu_{17} \\ \mu_{21} & \mu_{22} & \mu_{23} & \mu_{24} & \mu_{25} & \mu_{26} & \mu_{27} \end{bmatrix},$$

где μ_* – произвольные элементы, а коэффициенты h_*^* вычисляются на основе следующих аналитических соотношений:

$$\begin{aligned} h_{\varphi}^{\text{про}} &= \bar{M}_z^{\delta_{\text{про}}} \bar{Y}^{\varphi} - \bar{M}_z^{\varphi} \bar{Y}^{\delta_{\text{про}}}, \\ h_{\Phi}^{\text{про}} &= \bar{M}_z^{\delta_{\text{про}}} \bar{Y}^{\delta_{\Phi}} - \bar{M}_z^{\delta_{\Phi}} \bar{Y}^{\delta_{\text{про}}}, \\ h_{\varphi}^{\Phi} &= \bar{M}_z^{\varphi} \bar{Y}^{\delta_{\Phi}} - \bar{M}_z^{\delta_{\Phi}} \bar{Y}^{\varphi}. \end{aligned} \quad (2.44)$$

Представим полученное множество в виде матрицы

$$H = \begin{bmatrix} \mu_{21} & \mu_{22} & \mu_{23} & \mu_{24} & \mu_{25} & \mu_{26} & \mu_{27} \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{h_{\varphi}^{\text{про}}}{h_{\Phi}^{\text{про}}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{h_{\varphi}^{\text{про}}}{h_{\Phi}^{\text{про}}} & 0 & 0 & 0 & 0 & 0 & 0 \\ -\mu_{11} & -\mu_{12} & -\mu_{13} & -\mu_{14} & -\mu_{15} & -\mu_{16} & -\mu_{17} \\ \mu_{11} & \mu_{12} & \mu_{13} & \mu_{14} & \mu_{15} & \mu_{16} & \mu_{17} \\ \frac{h_{\varphi}^{\Phi}}{h_{\Phi}^{\text{про}}} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (2.45)$$

Так как в (2.45) присутствуют произвольные элементы μ , то задача реконфигурации имеет множество решений. Выбор конкретного представителя матрицы реконфигурации может осуществляться на основе дополнительных требований к системе (оптимальность, робастность и т.д.).

Потребуем минимального количества элементов в матрице реконфигурации (2.45). Это условие обеспечивает простоту реализации системы реконфигурации и максимальную надежность ее функционирования. Очевидно, что такое представление соответствует виду

$$H_{\min} = [h \quad 0] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{h_{\varphi}^{\text{про}}}{h_{\phi}^{\text{про}}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{h_{\varphi}^{\text{про}}}{h_{\phi}^{\text{про}}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{h_{\varphi}^{\phi}}{h_{\phi}^{\text{про}}} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (2.46)$$

получаемому при обнулении всех произвольных элементов, кроме $\mu_{21} = -1$.

Для перехода от матричного описания алгоритма реконфигурации к его представлению в виде структурной схемы определим вектор параметров, подаваемых непосредственно на приводы поверхностей управления с выхода блока реконфигурации

$$u^p = \left[\varphi_{\text{пр}}^p \quad \varphi_{\text{л}}^p \quad \delta_{\text{ф.пр}}^p \quad \delta_{\text{ф.л}}^p \quad \delta_{\text{н.пр}}^p \quad \delta_{\text{н.л}}^p \quad \delta_{\text{пто}}^p \right]^T. \quad (2.47)$$

С учетом $u^p = (I + H)_u$, (2.36) и (2.47) приходим к упрощенной схеме, реализующей алгоритм реконфигурации в виде классической структурной схемы, показанной на рисунке 2.5. Данная схема достаточно просто реализуется в вычислителе СУ. Аналитическое представление коэффициентов реконфигурации (2.44) позволяет осуществлять настройку схемы по режимам полета в явном виде. Анализ схемы показывает, что сигнал на отклонение отказавшего органа управления $\varphi_{\text{пр}}$ подается на флапероны, левый стабилизатор и ПГО. Тем самым обеспечивается перераспределение управления по оставшимся работоспособным поверхностям управления. Сигналы, поступающие на работоспособные поверхности, изменению не подвергаются.

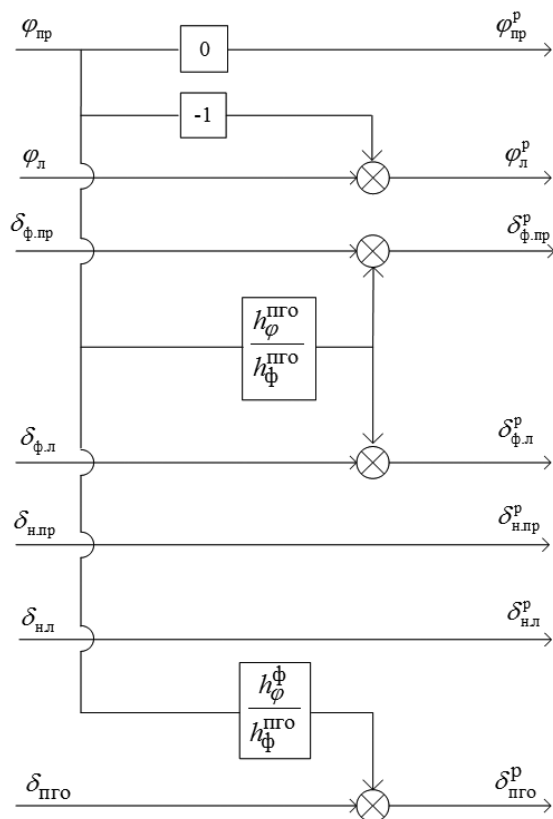


Рис. 2.5. Структурная схема реконфигурации системы управления при отказе правой секции стабилизатора на дозвуковых режимах полета

Для проверки работоспособности полученного закона реконфигурации осуществим численное моделирование компенсации последствий отказов с использованием модели пространственного движения самолета вида (2.35), дополненной штатной нелинейной моделью системы управления.

Матрица эффективности управления на рассматриваемом режиме полета содержит следующие элементы

$$B = \begin{bmatrix} -0.5 & 0.5 & -0.378 & 0.378 & -0.0029 & -0.0029 & 0 \\ 0.0008 & -0.0008 & 0.0014 & -0.0014 & -0.0059 & -0.0059 & 0 \\ -0.1702 & -0.1702 & -0.0009 & -0.0009 & 0 & 0 & 0.0579 \\ -0.0031 & -0.0031 & -0.0016 & -0.0016 & 0 & 0 & -0.0003 \\ 0 & 0 & 0 & 0 & -0.0003 & -0.0003 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Вычисленные в соответствии с формулами (2.44) коэффициенты матрицы реконфигурации составили $h_{\varphi}^{\text{про}} = 1.5134$, $h_{\psi}^{\text{про}} = 0.6126$, $h_{\varphi}^{\Phi} = -3.5524$. При этом согласно (2.46) матрица реконфигурации с минимальным числом ненулевых элементов имеет вид

$$H_{\min} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2.47 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2.47 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -5.8 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Графики переходных процессов при моделировании отказа привода правого стабилизатора и отклонении РУС по тангажу на -5° приведены на рисунках 2.6, 2.7. Из графиков видно, что, если не принимать меры по компенсации последствий отказа, помимо изменения параметров продольного движения появляются значительные отклонения параметров в боковом канале. Результаты моделирования доказывают, что реконфигурация управления в случае отказа привода стабилизатора сопровождается полной компенсацией последствий отказа. Кроме того, убеждаемся, что компенсация последствий отказа осуществляется за счет отклонения работоспособного стабилизатора, флаперонов и переднего горизонтального оперения.

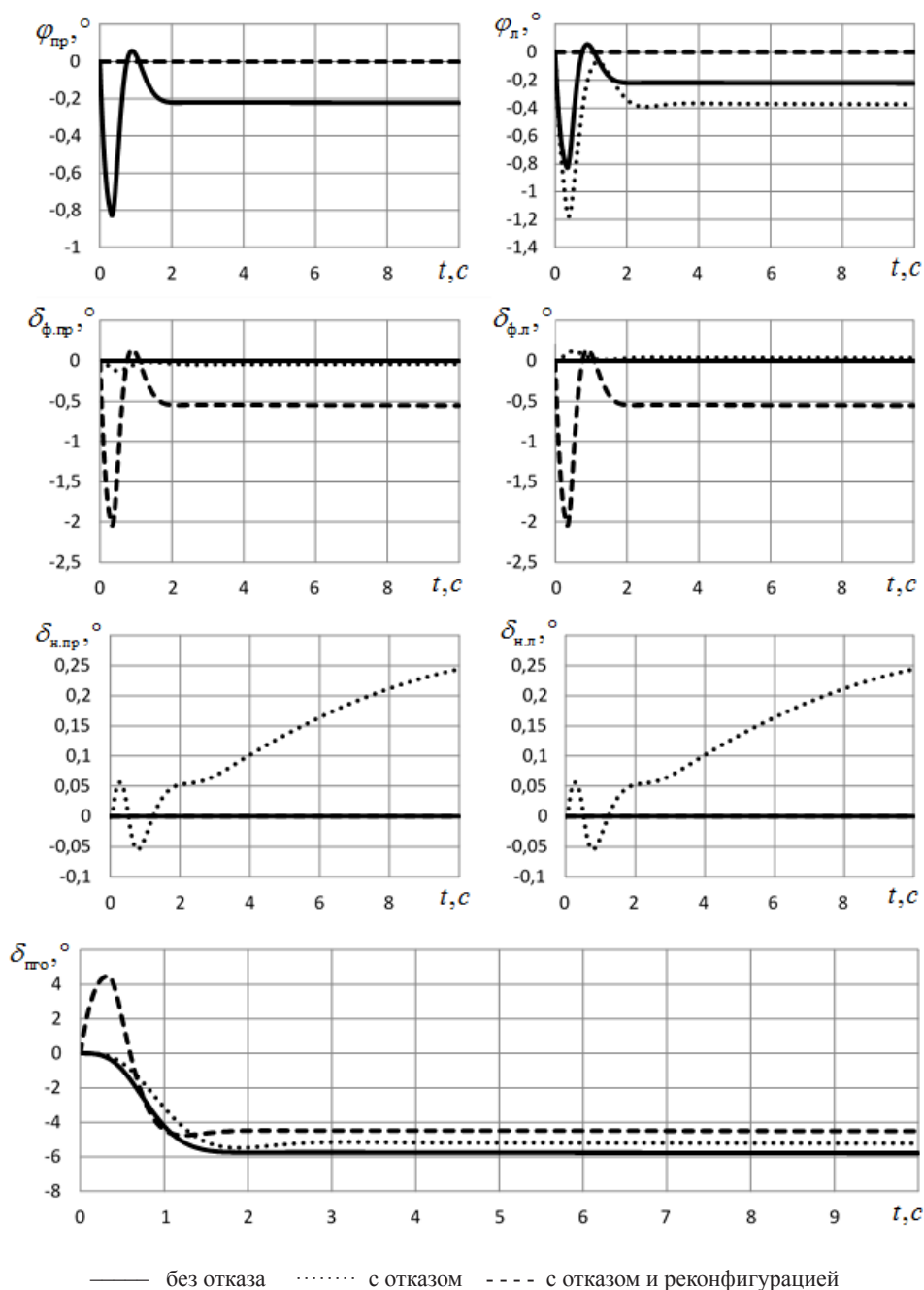


Рис. 2.6. Отклонение поверхностей управления при отказе правой секции стабилизатора и отклонении РУС по тангажу на -5°

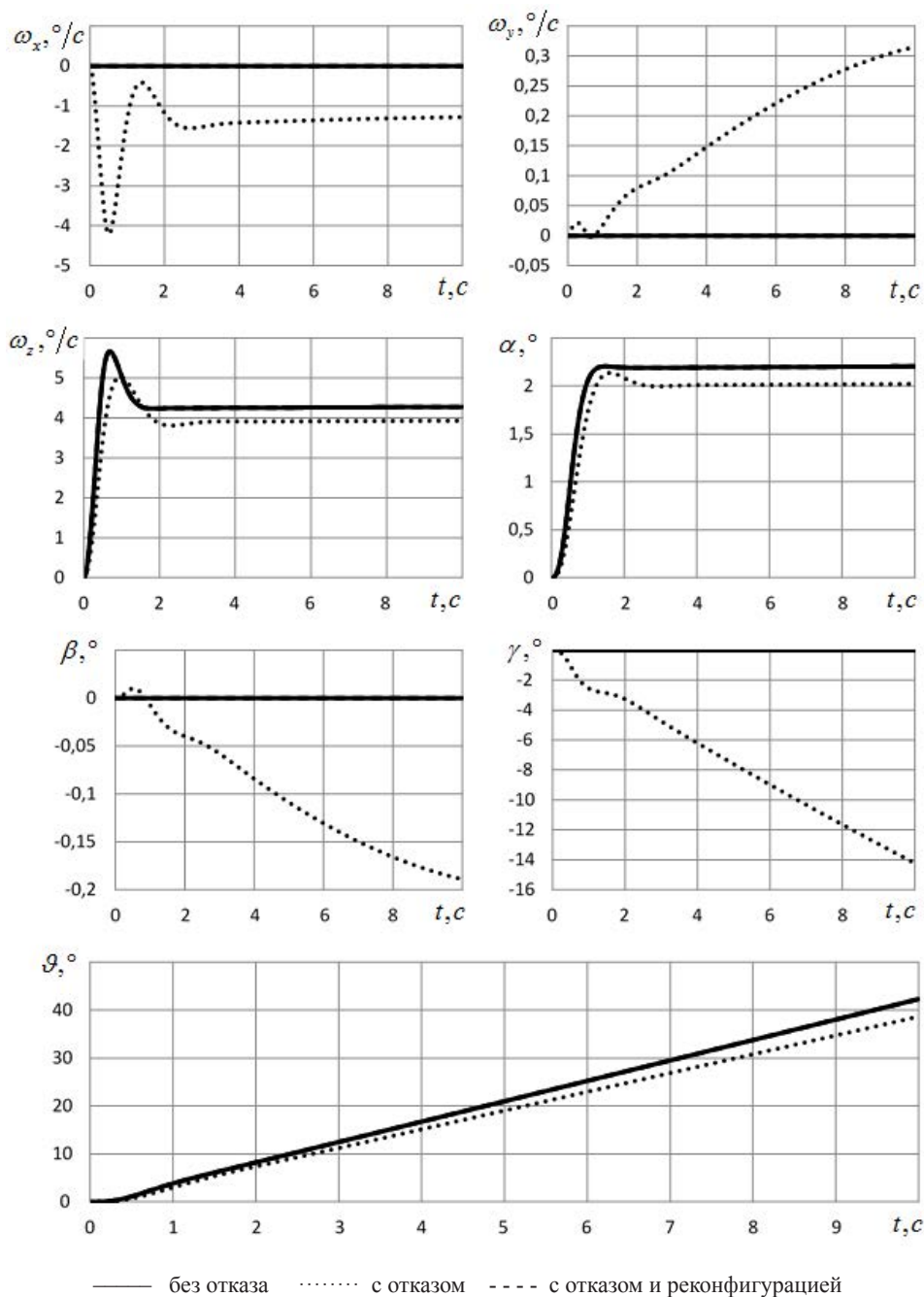


Рис. 2.7. Изменение аэродинамических параметров при отказе правой секции стабилизатора и отклонении РУС по тангажу на -5°

Аналогичным образом решим задачу реконфигурации СУ для компенсации отказов во всех исследуемых каналах управления на дозвуковом режиме полета.

Так как левый делитель нуля матрицы эффективности управления B_f для всех отказов на дозвуковом режиме полета остается постоянным (2.42), то задача реконфигурации управления разрешима в соответствии с (2.30) для всех рассматриваемых отказов.

Правый делитель нуля, за счет которого может быть расширено множество законов реконфигурации в выражении (2.31), имеет аналогичный (2.43) вид для всех отказов, кроме отказов рулей направления, при которых он зависит от параметров

$$\bar{B}_f^R = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -h_{\phi}^{\text{про}} & -h_{\phi}^{\text{про}} & h_{\varphi}^{\text{про}} & h_{\varphi}^{\text{про}} & 0 & 0 & h_{\varphi}^{\phi} \end{bmatrix}^T.$$

Полученные в соответствии с (2.31) коэффициенты матриц реконфигурации с минимальным числом ненулевых элементов приведены в таблице 2.2.

Из приведенных таблиц видно, что все коэффициенты матрицы реконфигурации определяются различными комбинациями отношений всего трех различных элементов в соответствии с аналитическими зависимостями (2.44).

На рисунках 2.8–2.11 приведены результаты численного моделирования работы алгоритмов реконфигурации при отказах руля направления и переднего горизонтального управления.

Графики показывают, что реконфигурация на дозвуковом режиме полета позволяет полностью компенсировать последствия отказов. Поведение самолета с работоспособной СУ полностью совпадает с поведением ВС с отказом в СУ и реконфигурацией.

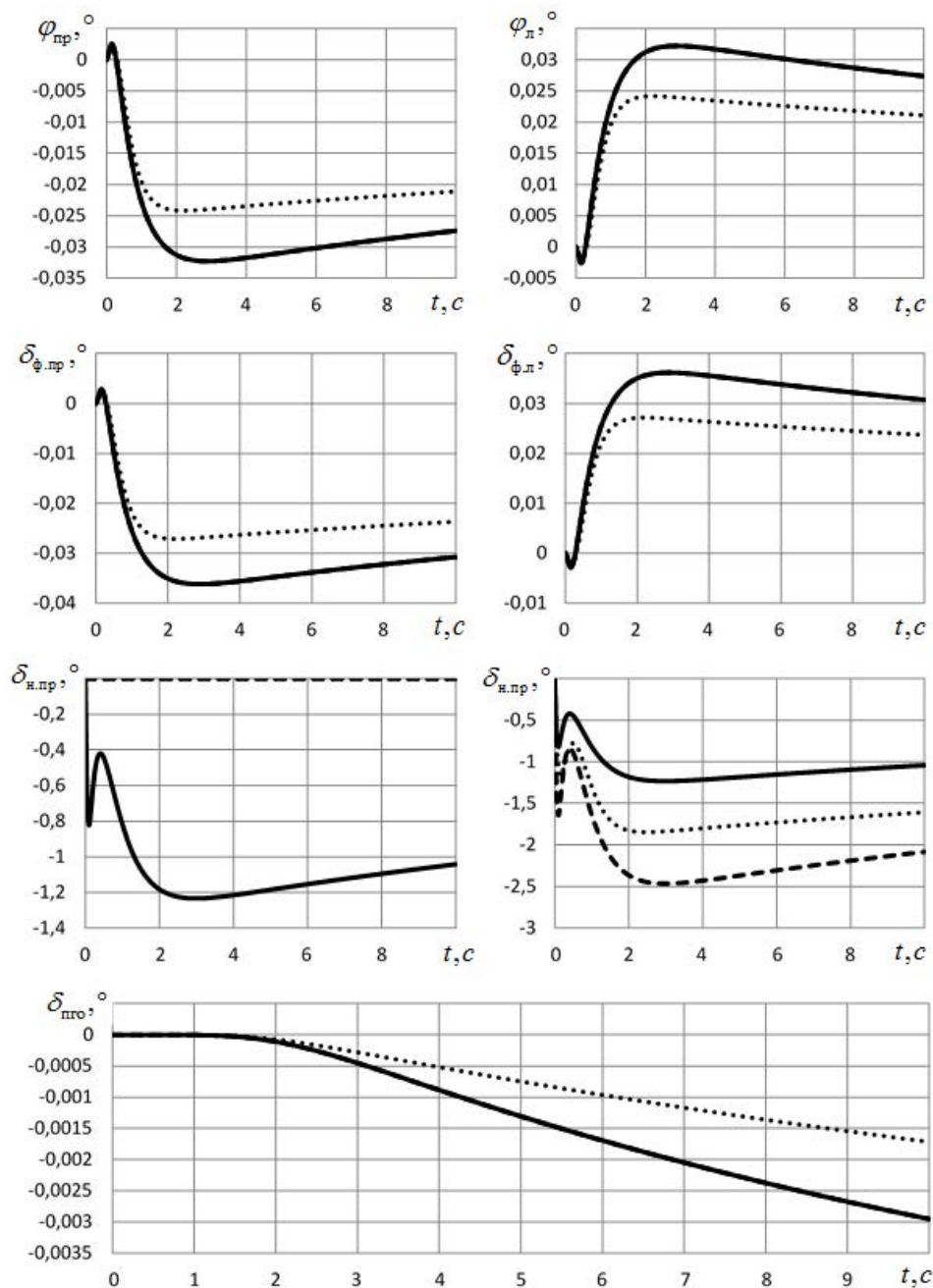
Табл. 2.2. Коэффициенты матриц реконфигурации для дозвуковых режимов полета

Канал	Отказ						
	$\varphi_{\text{пр}}$	$\varphi_{\text{л}}$	$\delta_{\text{ф.пр}}$	$\delta_{\text{ф.л}}$	$\delta_{\text{н.пр}}$	$\delta_{\text{н.л}}$	$\delta_{\text{про}}$
$\varphi_{\text{пр}}$	0	-1	$\frac{h_{\phi}^{\text{про}}}{h_{\varphi}^{\text{про}}}$	$\frac{h_{\phi}^{\text{про}}}{h_{\varphi}^{\text{про}}}$	0	0	$\frac{h_{\phi}^{\text{про}}}{h_{\varphi}^{\phi}}$
$\varphi_{\text{л}}$	-1	0	$\frac{h_{\phi}^{\text{про}}}{h_{\varphi}^{\text{про}}}$	$\frac{h_{\phi}^{\text{про}}}{h_{\varphi}^{\text{про}}}$	0	0	$\frac{h_{\phi}^{\text{про}}}{h_{\varphi}^{\phi}}$
$\delta_{\text{ф.пр}}$	$\frac{h_{\varphi}^{\text{про}}}{h_{\phi}^{\text{про}}}$	$\frac{h_{\varphi}^{\text{про}}}{h_{\phi}^{\text{про}}}$	0	-1	0	0	$-\frac{h_{\varphi}^{\text{про}}}{h_{\varphi}^{\phi}}$
$\delta_{\text{ф.л}}$	$\frac{h_{\varphi}^{\text{про}}}{h_{\phi}^{\text{про}}}$	$\frac{h_{\varphi}^{\text{про}}}{h_{\phi}^{\text{про}}}$	-1	0	0	0	$-\frac{h_{\varphi}^{\text{про}}}{h_{\varphi}^{\phi}}$

Канал	Отказ						
	$\varphi_{\text{пр}}$	$\varphi_{\text{л}}$	$\delta_{\text{ф.пр}}$	$\delta_{\text{ф.л}}$	$\delta_{\text{н.пр}}$	$\delta_{\text{н.л}}$	$\delta_{\text{пто}}$
$\delta_{\text{н.пр}}$	0	0	0	0	0	1	0
$\delta_{\text{н.л}}$	0	0	0	0	1	0	0
$\delta_{\text{пто}}$	$\frac{h_{\varphi}^{\text{ф}}}{h_{\text{ф}}^{\text{пто}}}$	$\frac{h_{\varphi}^{\text{ф}}}{h_{\text{ф}}^{\text{пто}}}$	$-\frac{h_{\varphi}^{\text{ф}}}{h_{\varphi}^{\text{пто}}}$	$-\frac{h_{\varphi}^{\text{ф}}}{h_{\varphi}^{\text{пто}}}$	0	0	0

Аналогичным образом определим матрицы реконфигурации при возникновении одиночных отказов во всех исследуемых каналах управления на сверхзвуковых режимах полета. Коэффициенты матриц реконфигурации с минимальным числом ненулевых элементов, приведенные в таблице 2.3, для всех возможных случаев определяются различными комбинациями отношений всего трех различных элементов в соответствии со следующими аналитическими зависимостями

$$\begin{aligned}
 h_{\varphi}^{\text{н}} &= 2 \left(\bar{\mathbf{M}}_x^{\varphi} \bar{\mathbf{M}}_y^{\delta_{\text{н}}} - \bar{\mathbf{M}}_x^{\delta_{\text{н}}} \bar{\mathbf{M}}_y^{\varphi} \right), \\
 h_{\text{ф}}^{\text{н}} &= \bar{\mathbf{M}}_x^{\delta_{\text{ф}}} \bar{\mathbf{M}}_y^{\delta_{\text{н}}} - \bar{\mathbf{M}}_x^{\delta_{\text{н}}} \bar{\mathbf{M}}_y^{\delta_{\text{ф}}}, \\
 h_{\text{ф}}^{\varphi} &= 2 \left(\bar{\mathbf{M}}_x^{\delta_{\text{ф}}} \bar{\mathbf{M}}_y^{\varphi} - \bar{\mathbf{M}}_x^{\varphi} \bar{\mathbf{M}}_y^{\delta_{\text{ф}}} \right).
 \end{aligned}
 \tag{2.48}$$



— без отказа с отказом ---- с отказом и реконфигурацией

Рис. 2.8. Отклонение поверхностей управления при отказе правого руля направления и отклонении педалей на 35 мм

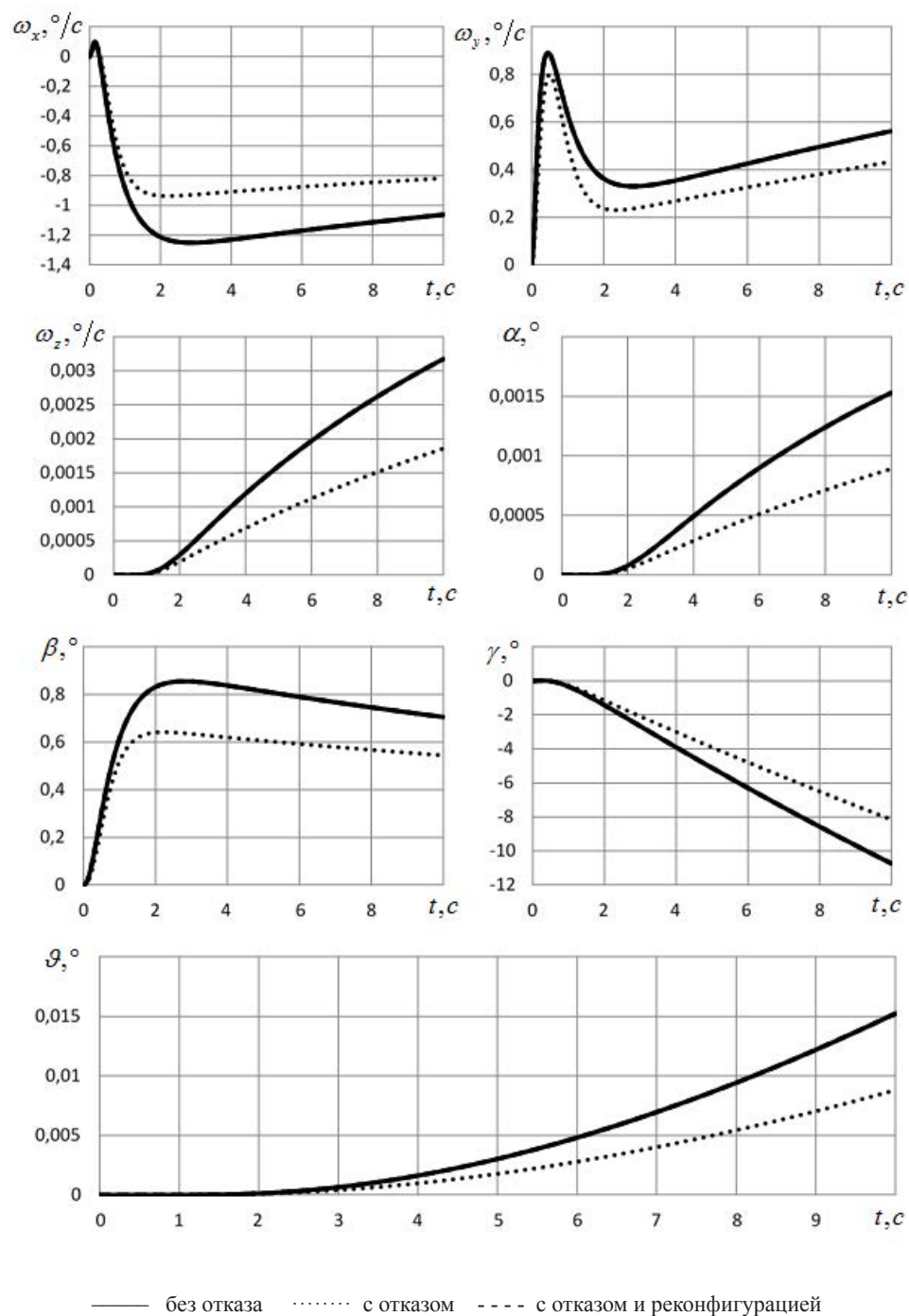


Рис. 2.9. Изменение аэродинамических параметров при отказе правого руля направления и отклонении педалей на 35 мм

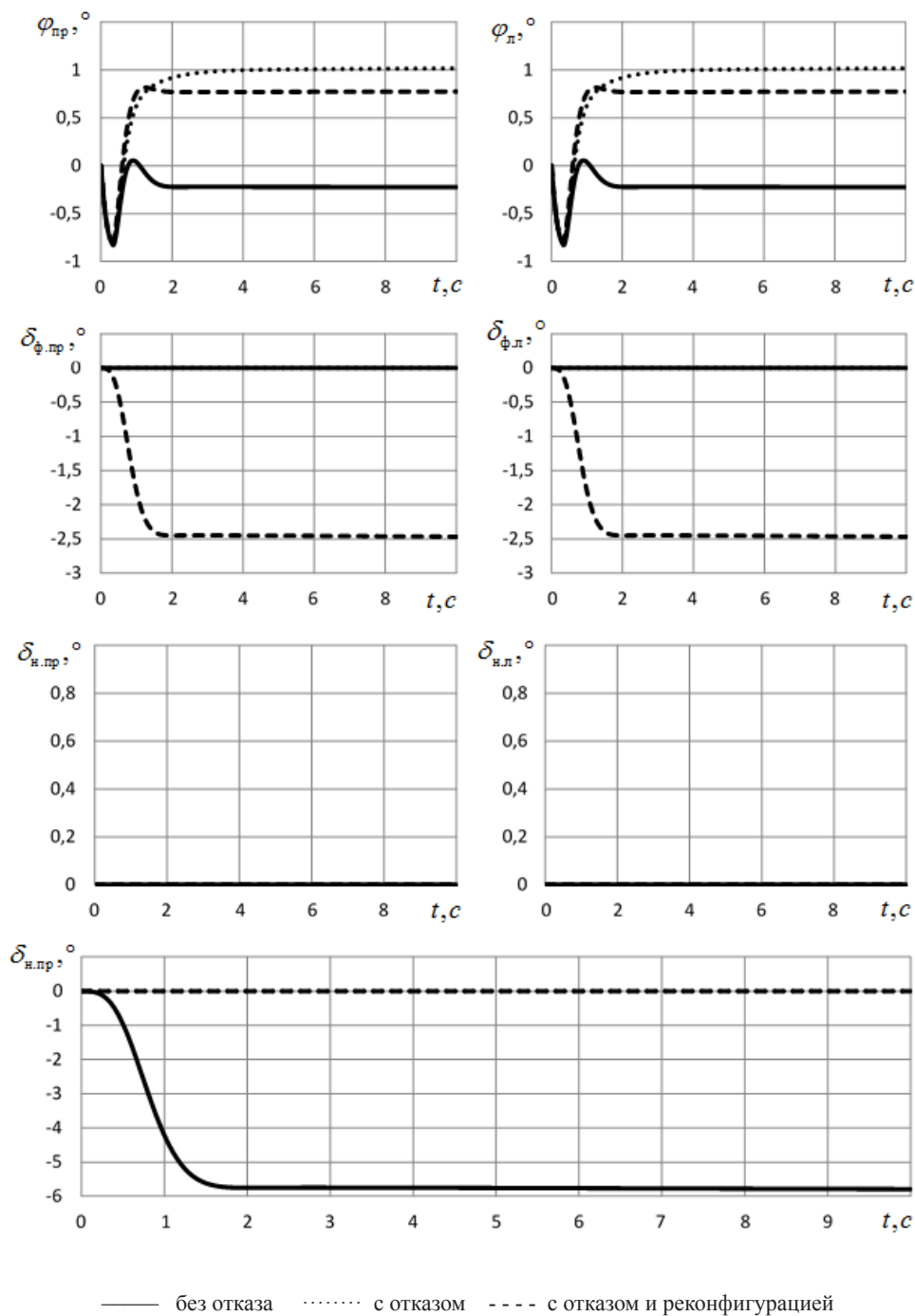


Рис. 2.10. Отклонение поверхностей управления при отказе ПГО и отклонении РУС по тангажу на -5°

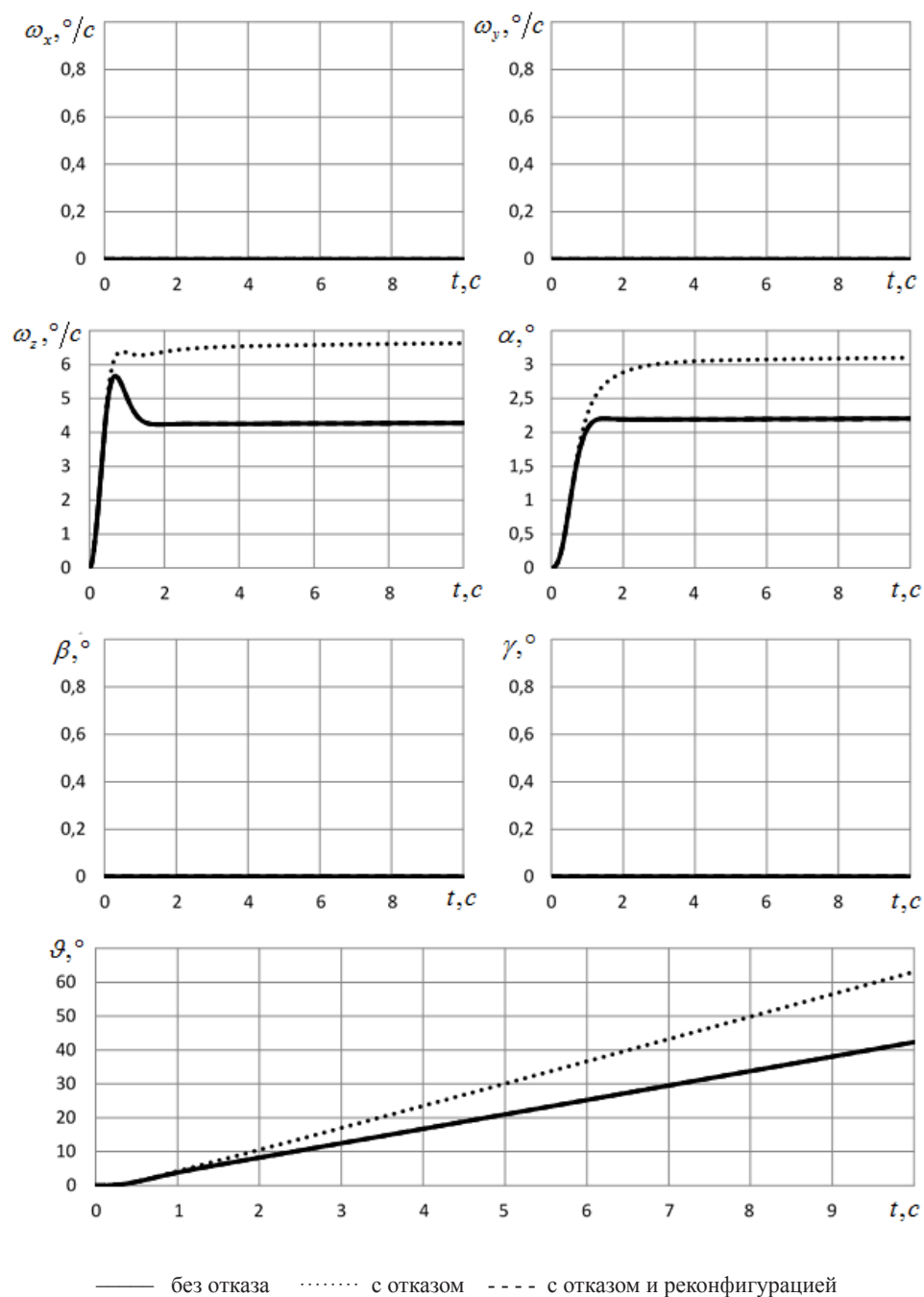


Рис. 2.11. Изменение аэродинамических параметров при отказе ПГО и отклонении РУС по тангажу на -5°

Табл. 2.3. Коэффициенты матриц реконфигурации для сверхзвуковых режимов полета

Канал	Отказ						
	$\varphi_{пр}$	$\varphi_{л}$	$\delta_{ф.пр}$	$\delta_{ф.л}$	$\delta_{н.пр}$	$\delta_{н.л}$	$\delta_{пто}$
$\varphi_{пр}$	0	1	0	0	0	0	0
$\varphi_{л}$	1	0	0	0	0	0	0
$\delta_{ф.пр}$	$\frac{h_{\varphi}^н}{h_{\varphi}^н}$	$-\frac{h_{\varphi}^{пто}}{h_{\varphi}^{пто}}$	0	-1	0	0	0
$\delta_{ф.л}$	0	0	-1	0	0	0	0
$\delta_{н.пр}$	0	0	0	0	0	1	0
$\delta_{н.л}$	$-\frac{h_{\varphi}^{\varphi}}{h_{\varphi}^н}$	$-\frac{h_{\varphi}^{\varphi}}{h_{\varphi}^н}$	0	0	1	0	0
$\delta_{пто}$	0	0	0	0	0	0	0

На рисунках 2.12–2.17 приведены результаты численного моделирования работы алгоритмов реконфигурации при отказах руля направления и переднего горизонтального управления.

Поведение самолета с работоспособной СУ совпадает с поведением ВС с отказом в СУ и реконфигурацией, кроме видимых незначительных расхождений на рисунках 2.16, 2.17, соответствующих отказу правой секции стабилизатора. Проанализируем далее причины этих расхождений.

На сверхзвуковых режимах полета при всех рассматриваемых отказах, кроме отказа стабилизатора, делители нуля матрицы B_f не изменяются и совпадают с делителями нуля B :

$$\bar{B}_f^L = \bar{B}^L = \begin{bmatrix} 0 & 0 & \bar{Y}^{\varphi} & \bar{M}_z^{\varphi} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \bar{B}_f^R = \bar{B}^R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T.$$

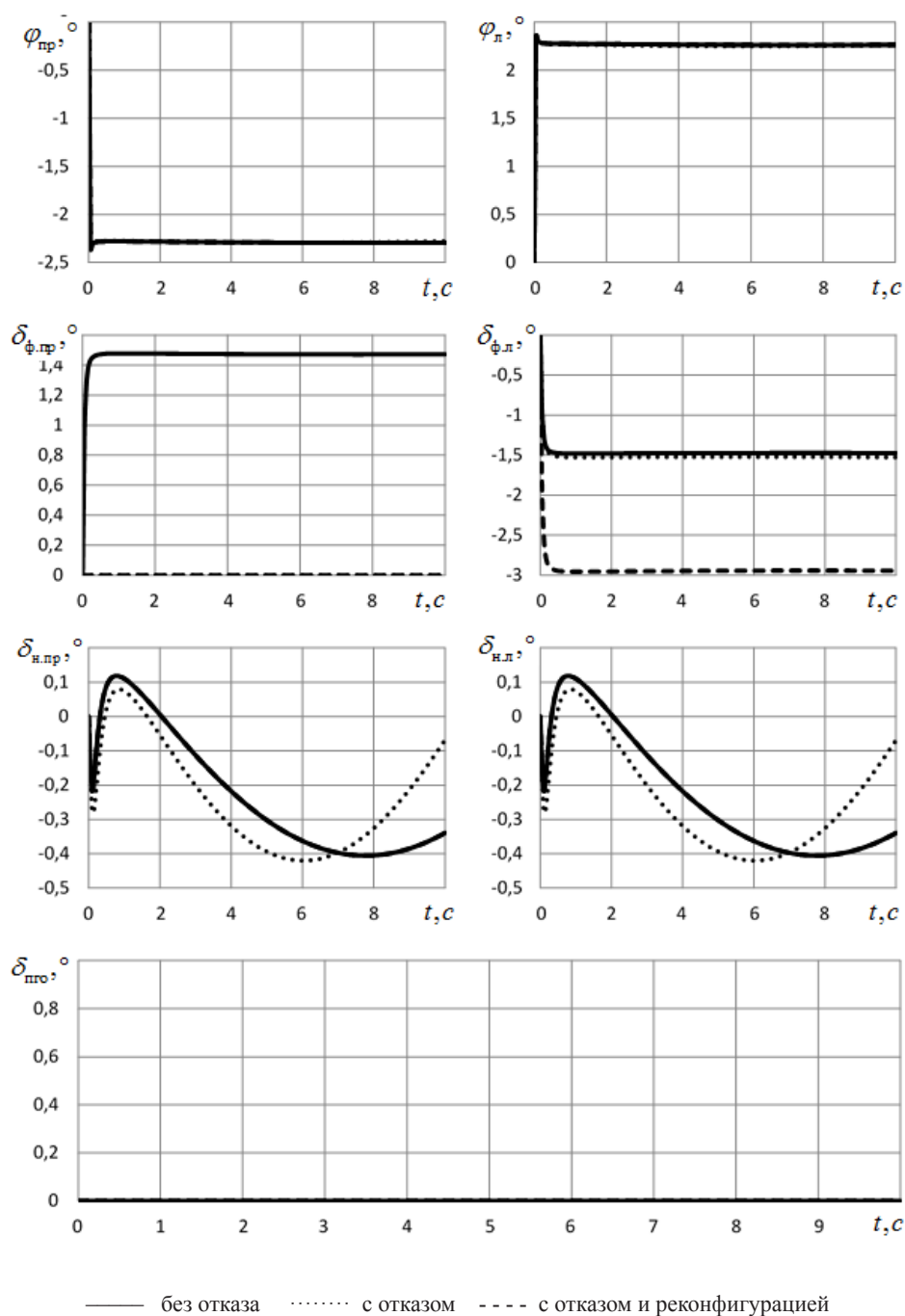


Рис. 2.12. Отклонение поверхностей управления при отказе правого флаперона и отклонении РУС по крену на -5°

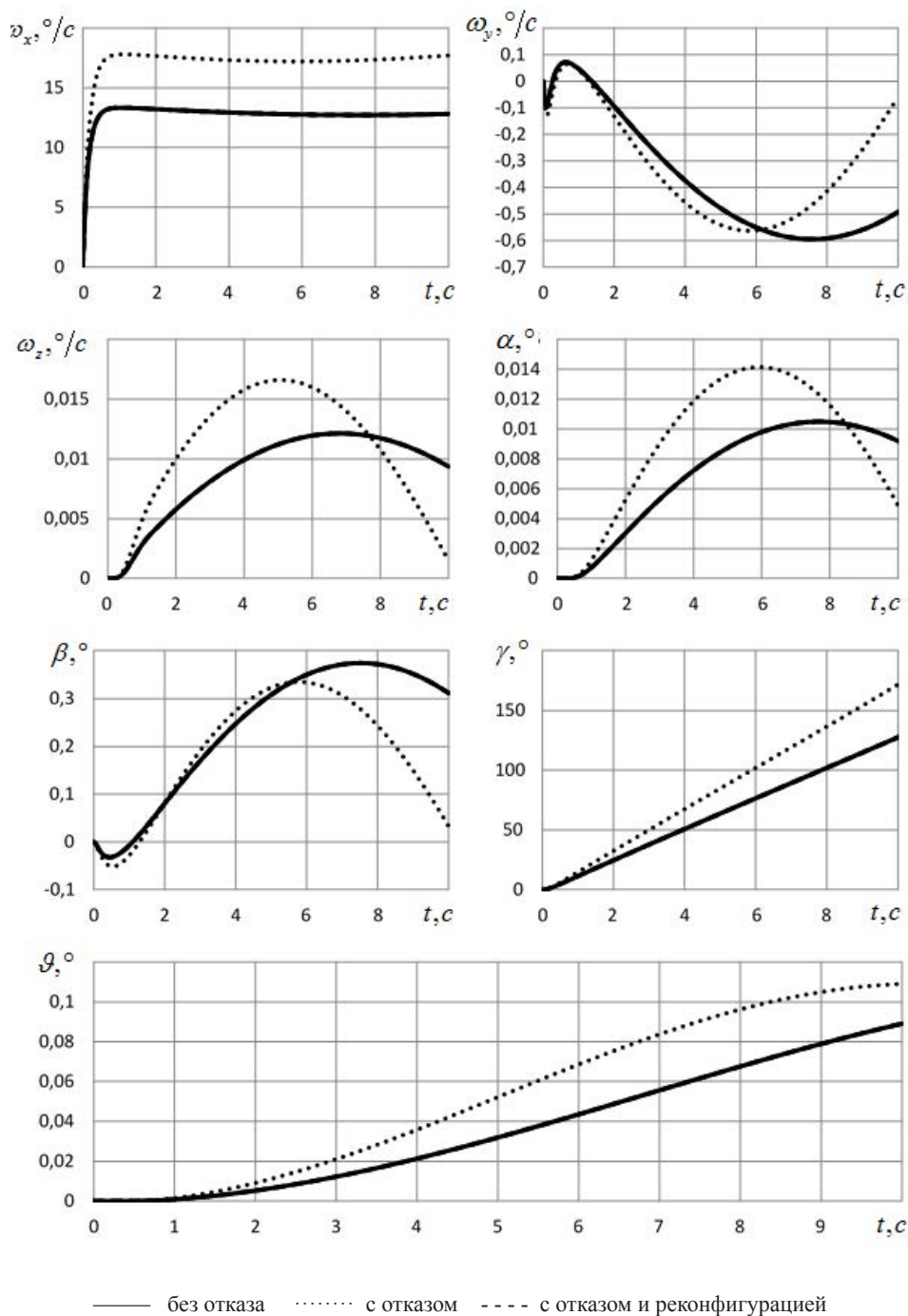


Рис. 2.13. Изменение аэродинамических параметров при отказе правого флаперона и отклонении РУС по крену на -5°

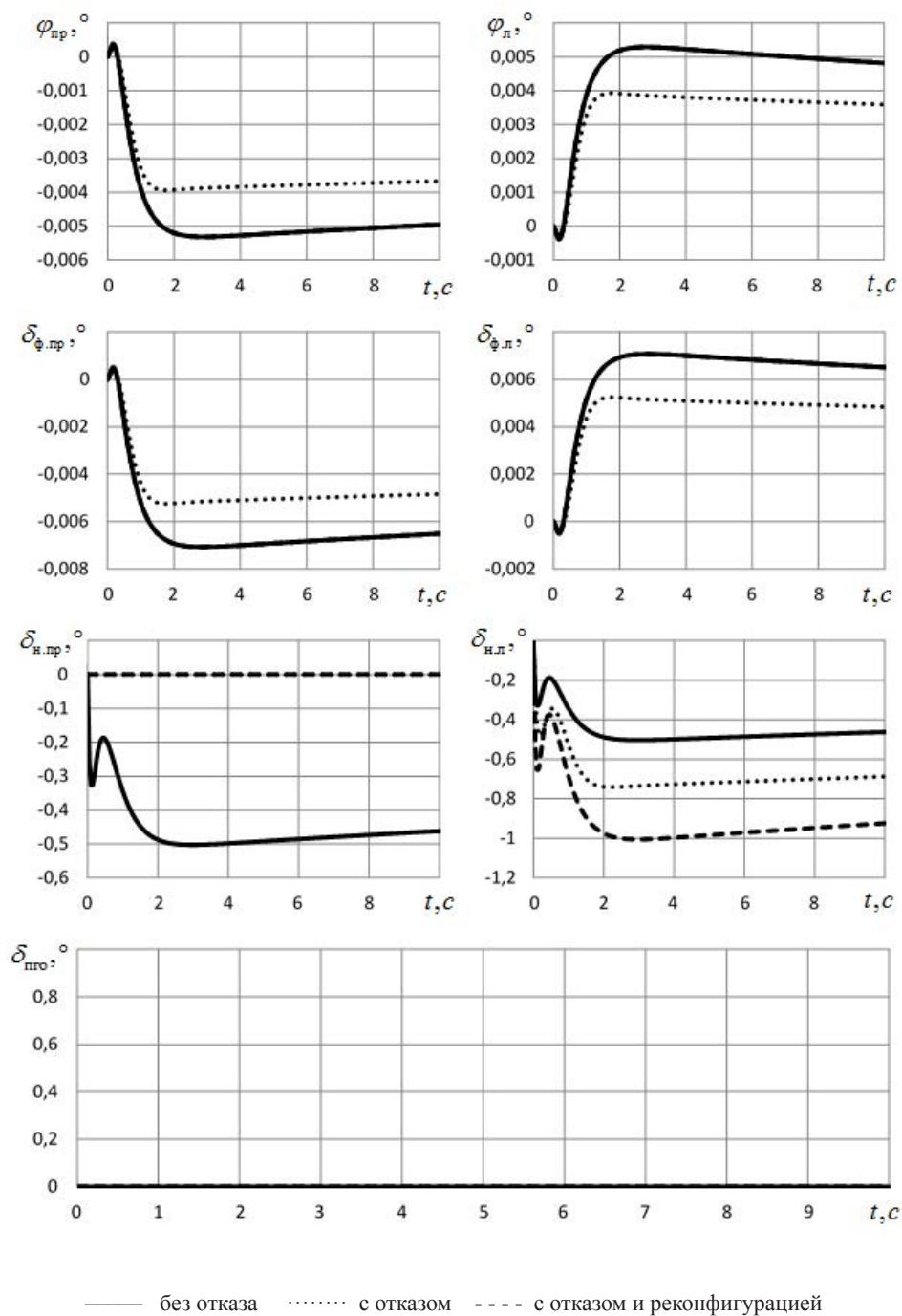


Рис. 2.14. Отклонение поверхностей управления при отказе правого руля направления и отклонении педалей на 35 мм

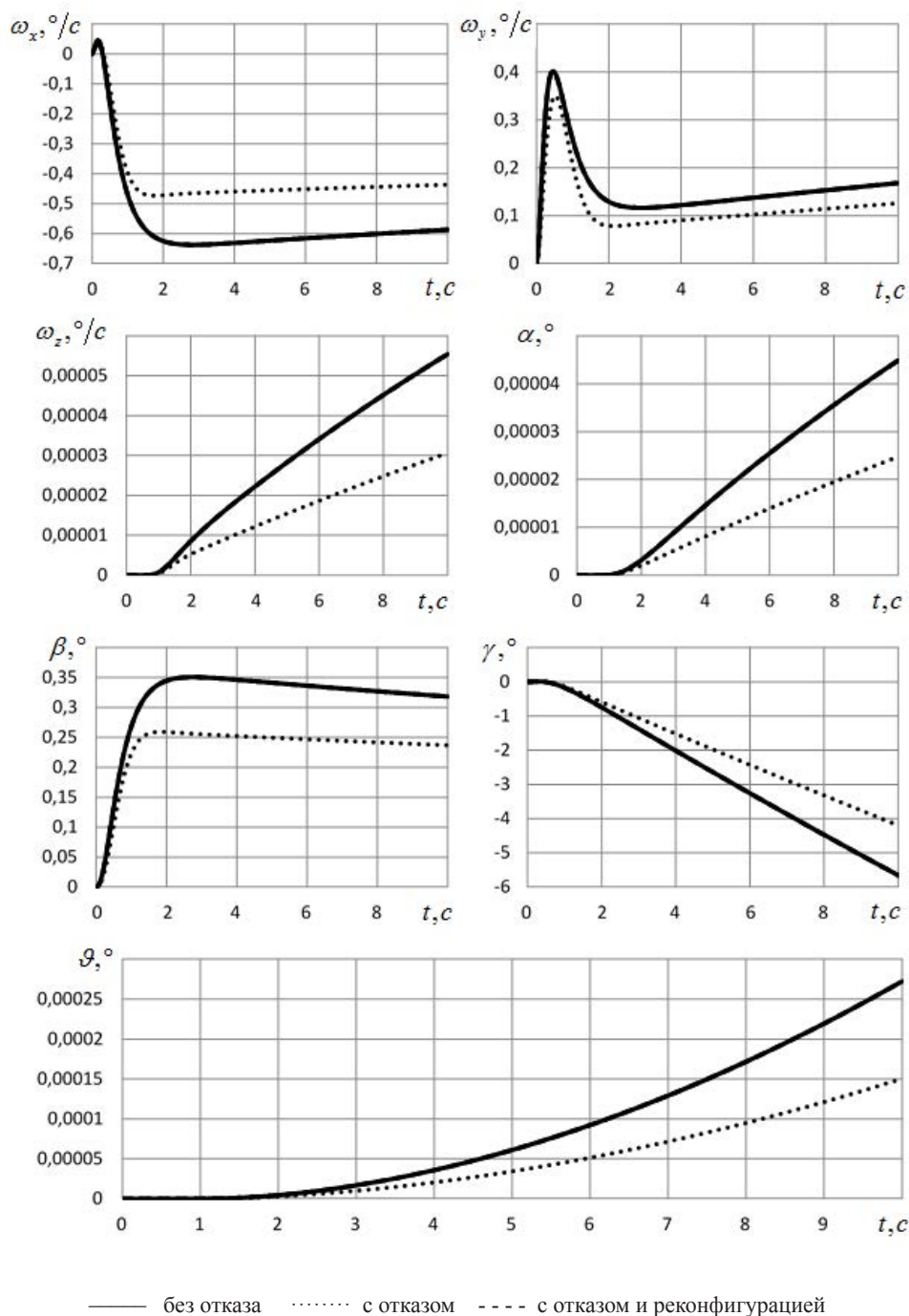


Рис. 2.15. Изменение аэродинамических параметров при отказе правого руля направления и отклонении педалей на 35 мм

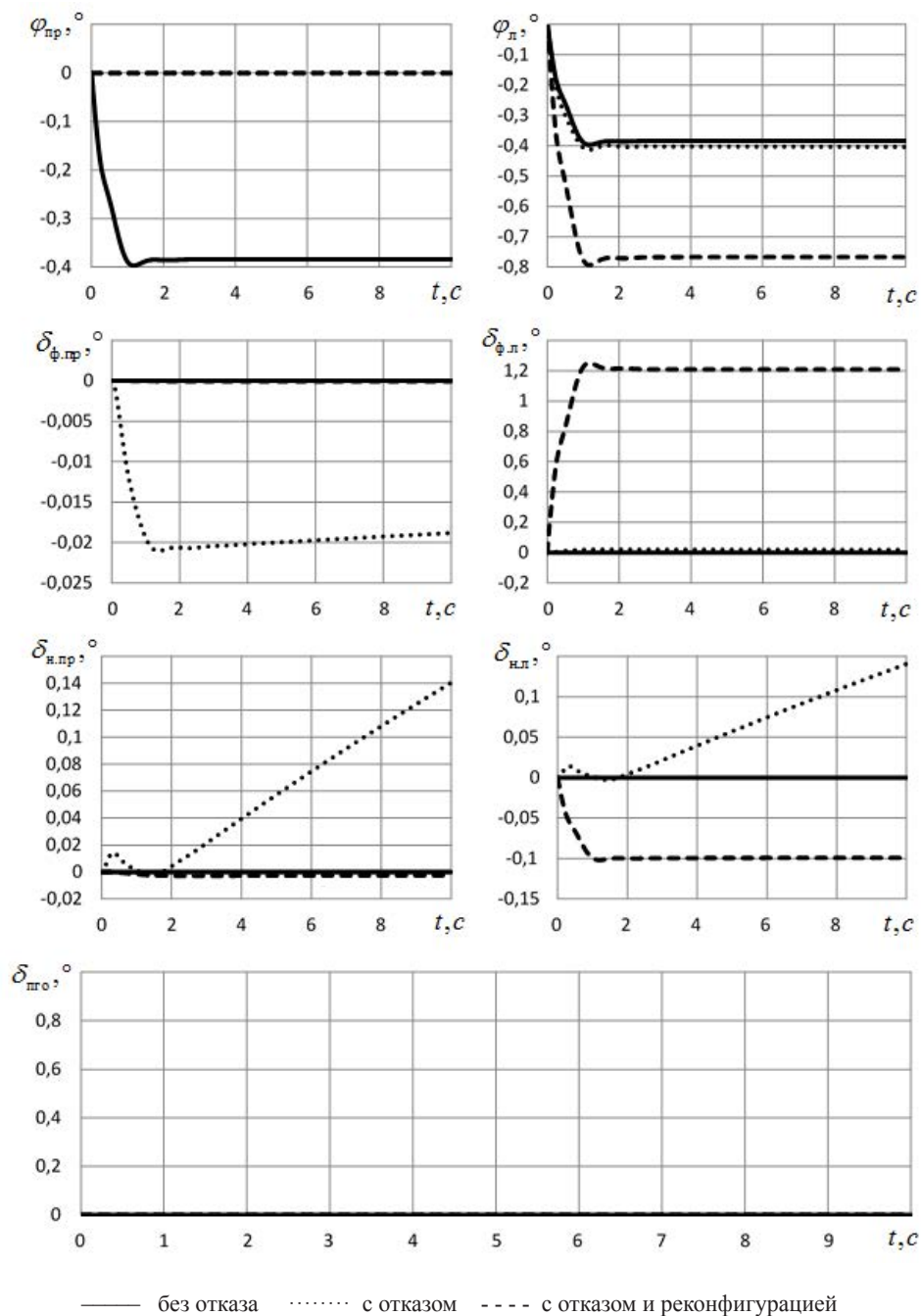


Рис. 2.16. Отклонение поверхностей управления при отказе правой секции стабилизатора и отклонении РУС по тангажу на -5°

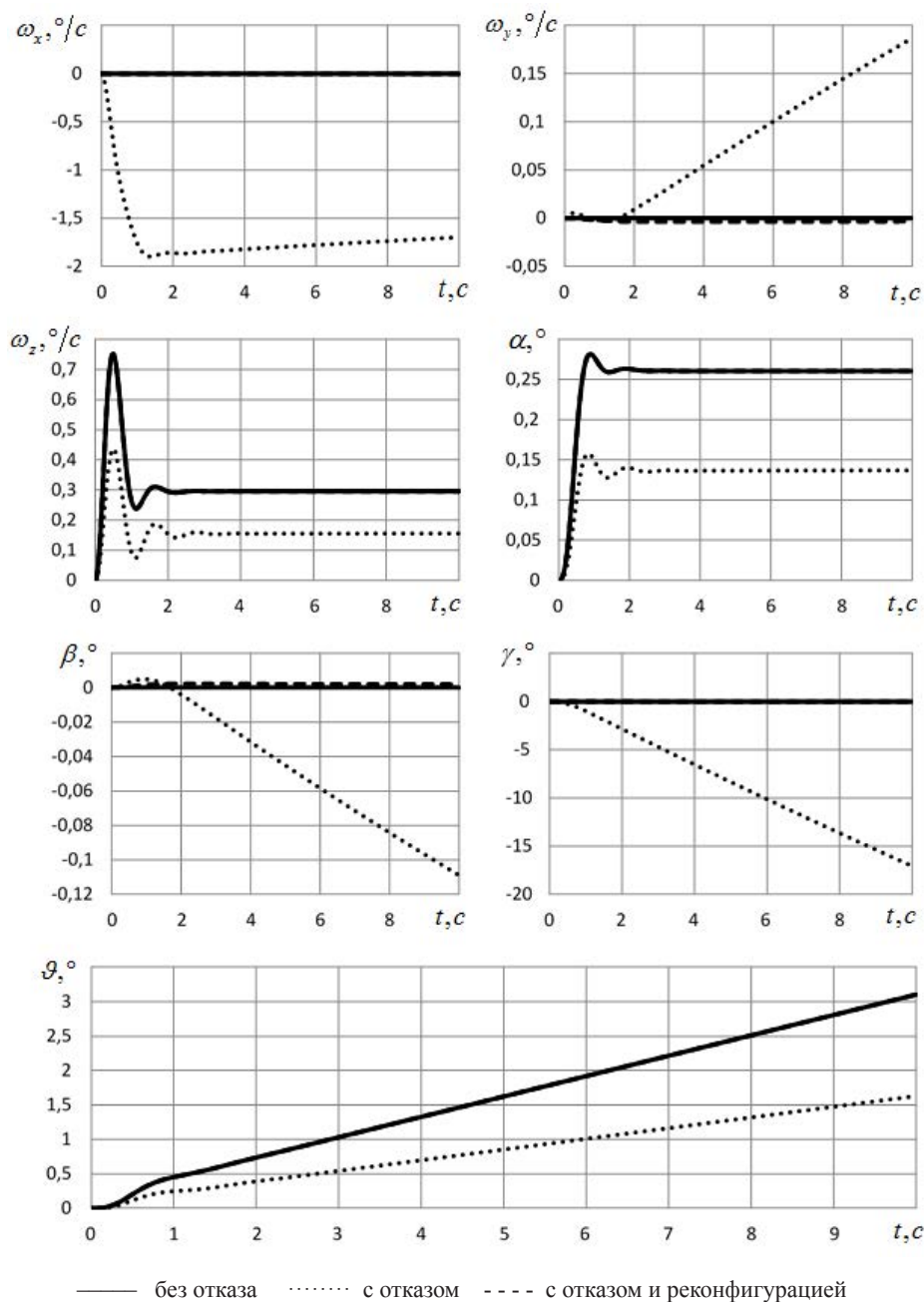


Рис. 2.17. Изменение аэродинамических параметров при отказе правой секции стабилизатора и отклонении РУС по тангажу на -5°

Поэтому при данных отказах задача реконфигурации является так же, как и на дозвуковых режимах, разрешимой, а множество общих для всех отказов решений имеет одинаковую структуру.

При отказе правой или левой секций стабилизатора матрица B_f с учетом (2.40) имеет по сравнению с (2.41) более разреженный вид

$$B_f = \begin{bmatrix} 0 & \bar{M}_x^\varphi & -\bar{M}_x^{\delta_\Phi} & \bar{M}_x^{\delta_\Phi} & \bar{M}_x^{\delta_n} & \bar{M}_x^{\delta_n} & 0 \\ 0 & \bar{M}_y^\varphi & -\bar{M}_y^{\delta_\Phi} & \bar{M}_y^{\delta_\Phi} & \bar{M}_y^{\delta_n} & \bar{M}_y^{\delta_n} & 0 \\ 0 & \frac{1}{2}\bar{M}_z^\varphi & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{2}\bar{Y}^\varphi & -\frac{1}{2}\bar{Y}^{\delta_\Phi} & -\frac{1}{2}\bar{Y}^{\delta_\Phi} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \bar{Z}^{\delta_n} & \bar{Z}^{\delta_n} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

что приводит к расширению размерности как левого

$$\bar{B}_f^L = \begin{bmatrix} 0 & 0 & \bar{Y}^\varphi & \bar{M}_z^\varphi & 0 & 0 & 0 \\ \bar{M}_y^{\delta_\Phi} \cdot \bar{Z}^{\delta_n} & -\bar{M}_x^{\delta_\Phi} \cdot \bar{Z}^{\delta_n} & \frac{\bar{Z}^{\delta_n}}{\bar{M}_z^\varphi} h_\Phi^\varphi & 0 & h_\Phi^n & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

так и правого

$$\bar{B}_f^R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T,$$

делителей нуля. При этом матрица реконфигурации с минимальным количеством ненулевых элементов имеет вид

$$H_{\min} = [h \quad 0] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{h_\varphi^n}{h_\Phi^n} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{h_\Phi^\varphi}{h_\Phi^{pn}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (2.49)$$

схема реализации которой представлена на рисунке 2.18.

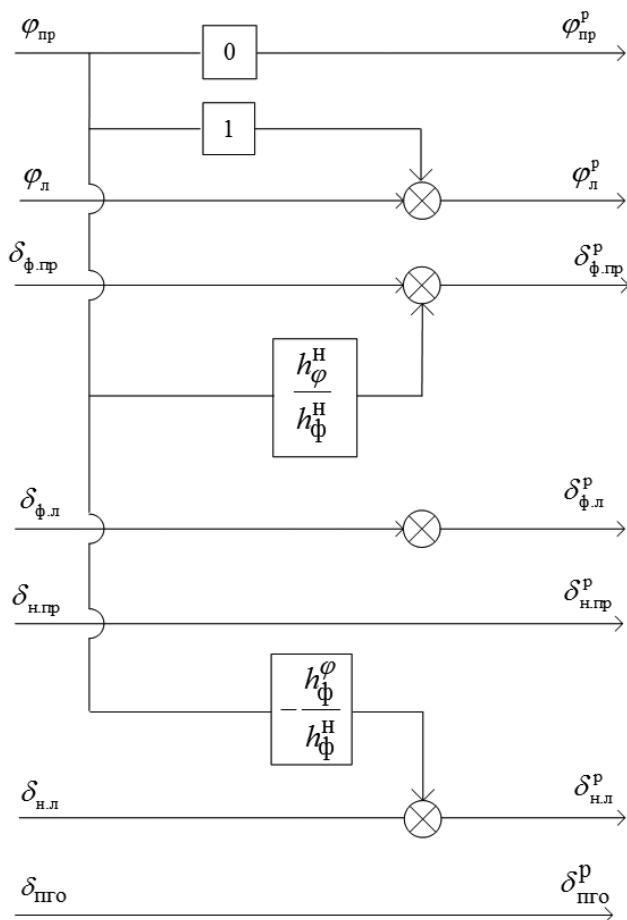


Рис. 2.18. Структурная схема реконфигурации системы управления при отказе правой секции стабилизатора на сверхзвуковых режимах полета

Принципиальное отличие схемы, представленной на рисунке 2.18, от схемы на рисунке 2.5, состоит в том, что на сверхзвуком режиме полета для реконфигурации не задействовано ПГО. Этим объясняется использование для реконфигурации каналов бокового движения.

Заметим также, что, в отличие от дозвукового режима, в данном случае условие разрешимости задачи реконфигурации (2.30) не выполняется

$$\bar{B}_f^L B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \bar{Z}^{\delta_n} \frac{h_{\phi}^{\phi}}{h_{\phi}^H} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (2.50)$$

а ошибка реконфигурации имеет следующий аналитический вид

$$B_f H - B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\bar{Z}^{\delta_n} \frac{h_\phi^\varphi}{h_\phi^n} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Невыполнение условия разрешимости задачи реконфигурации (2.30) при отказе одной из секций стабилизатора на сверхзвуковых режимах показывает, что реконфигурация в канале угла скольжения в принципе не осуществима.

При этом использование полученного закона обеспечит сохранение всех коэффициентов матрицы эффективности управления ВС, кроме одного, характеризующего непосредственное влияние отклонения работоспособной секции стабилизатора на скольжение самолета

$$B_p^{\text{нп}} = \begin{bmatrix} -\bar{M}_x^\varphi & \bar{M}_x^\varphi & -\bar{M}_x^{\delta_\phi} & \bar{M}_x^{\delta_\phi} & \bar{M}_x^{\delta_n} & \bar{M}_x^{\delta_n} & 0 \\ -\bar{M}_y^\varphi & \bar{M}_y^\varphi & -\bar{M}_y^{\delta_\phi} & \bar{M}_y^{\delta_\phi} & \bar{M}_y^{\delta_n} & \bar{M}_y^{\delta_n} & 0 \\ \frac{1}{2}\bar{M}_z^\varphi & \frac{1}{2}\bar{M}_z^\varphi & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2}\bar{Y}^\varphi & -\frac{1}{2}\bar{Y}^\varphi & -\frac{1}{2}\bar{Y}^{\delta_\phi} & -\frac{1}{2}\bar{Y}^{\delta_\phi} & 0 & 0 & 0 \\ -2\bar{Z}^{\delta_n} \frac{\bar{M}_x^{\delta_\phi} \cdot \bar{M}_y^\varphi - \bar{M}_x^\varphi \cdot \bar{M}_y^{\delta_\phi}}{\bar{M}_x^{\delta_\phi} \cdot \bar{M}_y^{\delta_n} - \bar{M}_x^{\delta_n} \cdot \bar{M}_y^{\delta_\phi}} & 0 & 0 & 0 & \bar{Z}^{\delta_n} & \bar{Z}^{\delta_n} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Именно этот элемент, появляющийся после реконфигурации в матрице эффективности управления, определяет незначительные ошибки в изменениях угла скольжения и угловой скорости рысканья, видимые на рисунках 2.16, 2.17.

2.2.5. Резюме

Изложенный аналитический подход не требует реализации в бортовых вычислителях сложных матричных операций при решении задачи реконфигурации и может быть реализован в виде классической структурной схемы

с настраиваемыми коэффициентами усиления. Простейшие в реализации законы реконфигурации, имеют минимальное количество ненулевых элементов. Аналитические зависимости коэффициентов законов реконфигурации от параметров полета допускают их настройку по режимам полета без непрерывного вычисления матриц реконфигурации, т.е. практически решается проблема нестационарности управления ВС. Применение данного подхода позволяет парировать последствия отказов СУ без какого-либо изменения реакции на управляющее воздействие и ненулевые начальные условия.

Метод канонизации является наиболее общим и позволяет описать в аналитическом виде все множество реализаций законов реконфигурации, среди которых заранее можно выбрать самые надежные варианты, т.е. более простые в реализации схемы с минимальным числом элементов. Кроме того, он не требует учета логики функционирования штатной СУ и может работать совместно с СУ любой сложности. Использование такого подхода при синтезе резервных законов управления способно существенно упростить их реализацию в бортовых вычислителях.

2.3. Оптимальное аналитическое решение задачи реконфигурации исполнительной подсистемы

2.3.1. Постановка задачи оптимальной реконфигурации

Основным недостатком рассмотренного выше метода является необходимость удовлетворения условия реконфигурируемости (2.30). При этом при возникновении отказов, приводящих к несовместности уравнения реконфигурации (2.29) и нарушению условия (2.30), не все законы из множества (2.31) способны гарантированно предотвратить потерю устойчивости и управляемости ВС. Анализ этого множества не позволяет в явном виде оценить оптимальность получаемых решений, с точки зрения как возможных ошибок реконфигурации, так и затрачиваемой на реконфигурацию мощности управления, характеризующей амплитуды компенсирующих отклонений работоспособных поверхностей управления.

Поэтому в случае невозможности получения «точного» решения задачи реконфигурации ИП СУ необходимо перейти к задаче определения оптимального решения, обеспечивающего не строгое равенство $B_f H - \Delta B = 0$, а некоторую ошибку реконфигурации

$$\Delta = B_f H - \Delta B, \quad (2.51)$$

удовлетворяющую условию минимальности некоторой нормы вида

$$\|\Delta\|_p = \left(\sum_{q=1}^{\min\{m,n\}} \sigma_q^p \right)^{1/p} = \min, \quad (2.52)$$

где σ_q – сингулярные числа получаемой матрицы ошибки.

В качестве такой нормы могут быть использованы следовая норма ($p = 1$), норма Фробениуса ($p = 2$), спектральная норма ($p = \infty$) и т.д., по сути, не меняющие главное – оценку степени близости ошибки (2.52) к нулю. Далее для определенности будем использовать матричную норму Фробениуса, для которой справедливо следующее равенство

$$\|\Delta\|_2 = \|\Delta\| = \sqrt{\sum_{q=1}^{\min\{m,n\}} \sigma_q^2} = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |\delta_{ij}|^2}, \quad (2.53)$$

где δ_{ij} – элементы матрицы ошибки реконфигурации Δ , а вывод формул будем производить с использованием только ортогональных преобразований, что позволит распространить полученные результаты на любые нормы вида (2.52) вследствие их инвариантности к таким преобразованиям.

В работах [15, 67, 90] показано, что такая постановка задачи соответствует максимальному сохранению стереотипа управления, минимальному отклонению собственных значений и нормы разности матриц собственной динамики ВС с замкнутой СУ. Для решения этой задачи разработаем специальную процедуру получения оптимального решения линейных матричных уравнений в аналитическом виде.

2.3.2. Оптимальное решение линейных матричных уравнений

Предположим, что уравнение (2.22) является несовместным, так как не выполняется условие разрешимости (2.23). В этом случае уравнение не имеет точного решения, и мы будем искать приближенное решение \hat{X} с точностью до некоторой величины $\Delta = A\hat{X} - B$.

Запишем сингулярное разложение матрицы A вида

$$A = L^T \hat{\Sigma} R^T = \begin{bmatrix} L_\Sigma^T & L_0^T \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} R_\Sigma^T \\ R_0^T \end{bmatrix}, \quad (2.54)$$

где L, R – матрицы левых и правых сингулярных векторов, удовлетворяющие условиям ортогональности

$$LL^T = \begin{bmatrix} L_\Sigma \\ L_0 \end{bmatrix} \begin{bmatrix} L_\Sigma^T & L_0^T \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}, \quad (2.55)$$

$$R^T R = \begin{bmatrix} R_\Sigma^T \\ R_0^T \end{bmatrix} \begin{bmatrix} R_\Sigma & R_0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}, \quad (2.56)$$

$\hat{\Sigma}$ – диагональная матрица сингулярных чисел (СЧ) с ранжированными по убыванию положительными числами; Σ – диагональная матрица ненулевых СЧ; L_{\max}, R_{\max} – блоки левых и правых матриц сингулярных векторов, соответствующих максимальным СЧ; l_{\min}, r_{\min} – левые и правые сингулярные вектора, соответствующие ненулевым СЧ.

Сформулируем и докажем теорему о численном оптимальном решении линейного матричного уравнения.

Теорема 2.3. Линейное матричное уравнение вида

$$AX = B, \quad (2.57)$$

разрешимо с точностью до $\|\Delta = A\hat{X} - B\| \leq \varepsilon$ тогда и только тогда, когда выполняется приближенное условие разрешимости

$$\|L_0 B\| = \|\Delta\| \leq \varepsilon, \quad (2.58)$$

при этом все множество решений, минимизирующих норму ошибки $\varepsilon_{\min} = \|\Delta\| = \min$, описывается формулой

$$\hat{X} = \begin{bmatrix} R_\Sigma & R_0 \end{bmatrix} \begin{bmatrix} \Sigma^{-1} L_\Sigma B \\ \Psi \end{bmatrix} = R_\Sigma \Sigma^{-1} L_\Sigma B + R_0 \Psi = A^+ B + R_0 \Psi, \quad (2.59)$$

где Ψ – матрица с произвольными элементами, а единственное оптимальное решение с минимальной нормой $\|\hat{X}_{\min}\| = \min$ определяется при $\Psi = 0$ в виде

$$\hat{X}_{\min} = \begin{bmatrix} R_\Sigma & R_0 \end{bmatrix} \begin{bmatrix} \Sigma^{-1} L_\Sigma B \\ 0 \end{bmatrix} = R_\Sigma \Sigma^{-1} L_\Sigma B = A^+ B. \quad \square$$

Доказательство. Подставим (2.54) в исходное уравнение (2.57)

$$\begin{bmatrix} L_\Sigma^T & L_0^T \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} R_\Sigma^T \\ R_0^T \end{bmatrix} X = B \quad (2.60)$$

и умножим (2.60) слева на матрицу левых сингулярных векторов L :

$$\begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} R_\Sigma^T \\ R_0^T \end{bmatrix} X = \begin{bmatrix} L_\Sigma \\ L_0 \end{bmatrix} B.$$

Введем обозначения

$$\begin{bmatrix} Y_\Sigma^T \\ Y_0^T \end{bmatrix} = \begin{bmatrix} R_\Sigma^T \\ R_0^T \end{bmatrix} X, \quad (2.61)$$

$$\begin{bmatrix} \Delta_\Sigma \\ \Delta_0 \end{bmatrix} = \begin{bmatrix} L_\Sigma \\ L_0 \end{bmatrix} B \quad (2.62)$$

и запишем эквивалентное (2.57) уравнение

$$\begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Y_\Sigma^T \\ Y_0^T \end{bmatrix} = \begin{bmatrix} \Delta_\Sigma \\ \Delta_0 \end{bmatrix}. \quad (2.63)$$

Из (2.63) видно, что Y_0^T может всегда принимать произвольные значения $Y_0^T = \Psi$, а все множество решений уравнения (2.63) записывается в виде

$$\begin{bmatrix} Y_\Sigma^T \\ Y_0^T \end{bmatrix} = \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta_\Sigma \\ \Psi \end{bmatrix}. \quad (2.64)$$

При этом ошибка решения (2.64) после подстановки его в (2.63) всегда будет равна

$$\Delta = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta_\Sigma \\ \Psi \end{bmatrix} - \begin{bmatrix} \Delta_\Sigma \\ \Delta_0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\Delta_0 \end{bmatrix}. \quad (2.65)$$

Подставим множество решений (2.64) эквивалентного уравнения (2.63) в выражение (2.61)

$$\begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta_\Sigma \\ \Psi \end{bmatrix} = \begin{bmatrix} R_\Sigma^T \\ R_0^T \end{bmatrix} X$$

и запишем решение исходного уравнения

$$X = \begin{bmatrix} R_\Sigma & R_0 \end{bmatrix} \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \Delta_\Sigma \\ \Psi \end{bmatrix}.$$

Далее с учетом (2.62) получим множество решений уравнения (2.57) методом наименьших квадратов

$$X = \begin{bmatrix} R_\Sigma & R_0 \end{bmatrix} \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} L_\Sigma B \\ \Psi \end{bmatrix} = R_\Sigma \Sigma^{-1} L_\Sigma B + R_0 \Psi = A^+ B + R_0 \Psi, \quad (2.66)$$

обеспечивающее в виду свойства инвариантности нормы к ортогональным преобразованиям минимально возможную норму ошибки решения

$$\|\Delta\| = \|\Delta_0\| = \|L_0 B\| = \min. \quad (2.67)$$

При этом минимальная норма решения

$$\|X\| = \|\Sigma^{-1} L_\Sigma B\| = \min \quad (2.68)$$

обеспечивается при $\Psi = 0$ с использованием псевдообратной матрицы

$$X = A^+ B, \quad (2.69)$$

вычисляемой по формуле

$$A^+ = R_\Sigma \Sigma^{-1} L_\Sigma. \quad (2.70)$$

Матрица (2.70) дополнительно к условиям регулярности вида (2.17)

$$A^+ AA^+ = A^+, \quad (2.71)$$

$$AA^+ A = A \quad (2.72)$$

одновременно удовлетворяет условиям симметричности

$$(AA^+)^T = AA^+, \quad (2.73)$$

$$(A^+A)^T = A^+A, \quad (2.74)$$

что можно проверить с учетом свойств ортогональности (2.55), (2.56) и вытекающих из (2.54) тождеств

$$A = L_\Sigma^T \Sigma R_\Sigma^T,$$

$$\Sigma = L_\Sigma A R_\Sigma$$

простыми подстановками:

$$\begin{aligned} \underbrace{R_\Sigma \Sigma^{-1} L_\Sigma}_{A^+} \underbrace{A R_\Sigma \Sigma^{-1} L_\Sigma}_{A^+} &= \underbrace{R_\Sigma \Sigma^{-1} L_\Sigma}_{A^+}, \\ \underbrace{L_\Sigma^T \Sigma R_\Sigma^T}_A \underbrace{R_\Sigma \Sigma^{-1} L_\Sigma}_{A^+} \underbrace{L_\Sigma^T \Sigma R_\Sigma^T}_A &= \underbrace{L_\Sigma^T \Sigma R_\Sigma^T}_A, \\ \left(\underbrace{L_\Sigma^T \Sigma R_\Sigma^T}_A \underbrace{R_\Sigma \Sigma^{-1} L_\Sigma}_{A^+} \right)^T &= L_\Sigma^T \Sigma^{-T} \underbrace{R_\Sigma^T R_\Sigma^T}_I \Sigma^T L_\Sigma = \underbrace{L_\Sigma^T \Sigma R_\Sigma^T}_A \underbrace{R_\Sigma \Sigma^{-1} L_\Sigma}_{A^+}, \\ \left(\underbrace{\Sigma^{-1} L_\Sigma}_{A^+} \underbrace{L_\Sigma^T \Sigma R_\Sigma^T}_A \right)^T &= R_\Sigma \Sigma^T \underbrace{L_\Sigma L_\Sigma^T}_I \Sigma^{-T} R_\Sigma^T = \underbrace{R_\Sigma \Sigma^{-1} L_\Sigma}_{A^+} \underbrace{L_\Sigma^T \Sigma R_\Sigma^T}_A. \end{aligned}$$

Заметим, что сингулярное разложение (2.54), используемое для определения псевдообратной матрицы (2.70), по определению имеет численный вид, поэтому формула (2.59) для получения аналитического решения в явном виде не применима. Однако заметим, что вычисление левых сингулярных векторов, соответствующих нулевым сингулярным числам, для проверки условия разрешимости (2.58) можно производить по следующей аналитической формуле

$$L_0 = \left(\bar{A}^L \bar{A}^{L^T} \right)^{-\frac{1}{2}} \bar{A}^L. \quad (2.75)$$

Матрица правых сингулярных векторов R_0 в (2.59) может быть вычислена по аналогичной дуальной формуле, однако по постановке задачи вследствие равенства по определению $AR_0 = 0$ может быть без потери общности заменена на $R_0 = \bar{A}^R$.

Последним шагом получения аналитического решения является определение псевдообратной матрицы A^+ в выражении (2.59), в основу которого положим следующую теорему.

Теорема 2.4. Аналитическая формула псевдообратной матрицы имеет вид

$$A^+ = \overline{A}^R{}^{LT} \left(\overline{A}^L{}^{RT} \overline{A} \overline{A}^R{}^{LT} \right)^{-1} \overline{A}^L{}^{RT}, \quad (2.76)$$

где \overline{A}^L , \overline{A}^R – левый и правый делитель нуля полного ранга матрицы A , удовлетворяющие условиям $\overline{A}^L A = 0$, $A \overline{A}^R = 0$; $\overline{A}^L{}^R$, $\overline{A}^R{}^L$ – правый и левый делитель нуля полного ранга матриц \overline{A}^L , \overline{A}^R , удовлетворяющие условиям

$$\overline{A}^L \overline{A}^L{}^R = 0, \quad (2.77)$$

$$\overline{A}^R{}^L \overline{A}^R = 0. \quad (2.78)$$

Доказательство. Представим сингулярное разложение (2.54) в виде

$$\begin{bmatrix} L_\Sigma \\ L_0 \end{bmatrix} A \begin{bmatrix} R_\Sigma & R_0 \end{bmatrix} = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix}. \quad (2.79)$$

Из (2.79) можно записать все множества левых и правых делителей нуля полного ранга матрицы A

$$\overline{A}^L = \Phi L_0, \quad (2.80)$$

$$\overline{A}^R = R_0 \Xi, \quad (2.81)$$

порождаемые соответствующими произвольными обратимыми матрицами подходящих размеров Φ , Ξ .

Подставим (2.80), (2.81) в выражения (2.77), (2.78) и запишем с учетом обратимости Ξ , Φ следующие однородные уравнения

$$L_0 \overline{A}^L{}^R = 0,$$

$$\overline{A}^R{}^L R_0 = 0,$$

множества решений которых с учетом тождеств

$$L_0 L_\Sigma^T = 0,$$

$$R_\Sigma^T R_0 = 0,$$

следующих из условий ортогональности (2.55), (2.56),

$$\overline{A}^L{}^R = \overline{L}_0^R \Omega,$$

$$\overline{A}^R{}^L = \Upsilon \overline{R}_0^L$$

могут быть определены в соответствии с (2.24) по формулам

$$\overline{\overline{A}}^L{}^R = \overline{\overline{L}}_0^R \Omega = L_\Sigma^T \Omega, \quad (2.82)$$

$$\overline{\overline{A}}^R{}^L = \Upsilon \overline{\overline{R}}_0^L = \Upsilon R_\Sigma^T, \quad (2.83)$$

где Ω , Υ – произвольные обратимые матрицы подходящих размеров.

Подставим (2.83), (2.82) в аналитическую формулу для псевдообратной (2.76) и получим выражение

$$A^+ = R_\Sigma \Upsilon^T (\Omega^T L_\Sigma A R_\Sigma \Upsilon^T)^{-1} \Omega^T L_\Sigma = R_\Sigma \Upsilon^T (\Omega^T \Sigma \Upsilon^T)^{-1} \Omega^T L_\Sigma = R_\Sigma \Sigma^{-1} L_\Sigma, \quad (2.84)$$

полностью совпадающее с численной формулой (2.70). \square

На основании формул (2.75) и (2.76) сформулируем далее теорему об оптимальном аналитическом решении линейного матричного уравнения [17].

Теорема 2.5. Линейное матричное уравнение вида

$$AX = B$$

разрешимо с точностью до $\|\Delta = A\hat{X} - B\| \leq \varepsilon$ тогда и только тогда, когда выполняется приближенное условие разрешимости

$$\left\| \left(\overline{\overline{A}}^L \overline{\overline{A}}^{LT} \right)^{-\frac{1}{2}} \overline{\overline{A}}^L B \right\| \leq \varepsilon,$$

при этом все множество решений, минимизирующих норму ошибки $\|\Delta\| = \min$, описывается формулой

$$\hat{X} = \overline{\overline{A}}^R{}^{LT} \left(\overline{\overline{A}}^L{}^{RT} A \overline{\overline{A}}^R{}^{LT} \right)^{-1} \overline{\overline{A}}^L{}^{RT} B + R_0 \Psi = A^+ B + R_0 \Psi,$$

где Ψ – матрица с произвольными элементами, а единственное оптимальное решение с минимальной нормой $\|\hat{X}_{\min}\| = \min$ определяется при $\Psi = 0$ в виде

$$\hat{X}_{\min} = \overline{\overline{A}}^R{}^{LT} \left(\overline{\overline{A}}^L{}^{RT} A \overline{\overline{A}}^R{}^{LT} \right)^{-1} \overline{\overline{A}}^L{}^{RT} B = A^+ B. \quad (2.85)$$

2.3.3. Оптимальное решение задачи реконфигурации

Воспользуемся результатами теоремы 2.5 и сформулируем теорему об оптимальном аналитическом решении задачи реконфигурации ИП СУ ВС [17].

Теорема 2.6. Уравнение задачи реконфигурации вида (2.29)

$$B_f H = \Delta B_f \quad (2.86)$$

разрешимо с точностью до $\|\Delta = B_f \hat{H} - \Delta B_f\| \leq \varepsilon_{\text{доп}}$ тогда и только тогда, когда выполняется приближенное условие разрешимости

$$\left\| \left(\overline{\overline{B}}_f^L \overline{\overline{B}}_f^{LT} \right)^{0.5} \overline{\overline{B}}_f^L B_f \right\| = \|\Delta\| \leq \varepsilon_{\text{доп}}, \quad (2.87)$$

при этом все множество оптимальных решений, минимизирующих норму ошибки реконфигурации

$$\varepsilon_{\min} = \|\Delta\| = \left\| \left(\bar{B}_f^L \bar{B}_f^{LT} \right)^{0.5} \bar{B}_f^L B_f \right\| \rightarrow \min \quad (2.88)$$

описывается формулой

$$\hat{H} = B_f^+ \Delta B_f + \bar{B}_f^R \Theta = \bar{B}_f^R{}^{LT} \left(\bar{B}_f^L{}^{RT} B_f \bar{B}_f^R{}^{LT} \right)^{-1} \bar{B}_f^L{}^{RT} \Delta B_f + \bar{B}_f^R \Theta, \quad (2.89)$$

где \bar{B}_f^L , \bar{B}_f^R – левый и правый делителя нуля полного ранга матрицы B_f , удовлетворяющие условиям $\bar{B}_f^L B_f = 0$, $B_f \bar{B}_f^R = 0$; $\bar{B}_f^L{}^R$, $\bar{B}_f^R{}^L$ – правый и левый делителя нуля полного ранга матриц \bar{B}_f^L , \bar{B}_f^R , удовлетворяющие условиям $\bar{B}_f^L \bar{B}_f^L{}^R = 0$, $\bar{B}_f^R{}^L \bar{B}_f^R = 0$, Θ – матрица с произвольными элементами, а единственное оптимальное решение с минимальной мощностью реконфигурации $\mu_{\min} = \|\hat{H}_{\min}\| = \min$ определяется при $\Theta = 0$ в виде

$$\hat{H}_{\min} = B_f^+ \Delta B_f. \quad (2.90)$$

Прагматический смысл полученного результата заключается в следующем. При выполнении приближенного условия разрешимости (2.87) гарантируется, что задача реконфигурации управления на основе соотношений (2.89) будет решена с априорно заданной точностью, не превышающей $\varepsilon_{\text{доп}}$. Это позволяет сохранить все достоинства аналитического решения и использовать полученное множество оптимальных законов управления для решения дополнительных задач (минимизации амплитуд отклонений поверхностей управления, настройки коэффициентов усиления по режимам полета, упрощения схемной реализации СУ ВС и т.д.). Поясним это на примере.

2.3.4. Пример оптимального решения задачи реконфигурации

Пусть линейная модель движения ВС имеет вид выражения (2.3), где $x = [\Delta V \ \Delta \alpha \ \Delta \Theta \ \Delta \omega_z \ \Delta \beta \ \Delta \omega_x \ \Delta \omega_y \ \Delta \gamma]$ – вектор состояния; ω_x , ω_y , ω_z – угловые скорости крена, рыскания и тангажа; α , Θ , γ , β – углы атаки, наклона траектории, крена и скольжения; V – скорость полета; $[\delta_{в.л} \ \delta_{в.пр} \ \varphi \ \delta_{н} \ \delta_{э.л} \ \delta_{э.пр} \ \delta_{и.л} \ \delta_{и.пр}]$ – вектор управления; $\delta_{в.л}$, $\delta_{в.пр}$ – углы отклонения левого и правого руля высоты; φ – угол отклонения стабилизатора; $\delta_{н}$ – угол отклонения руля направления; $\delta_{э.л}$, $\delta_{э.пр}$ – углы отклонения левого и правого элерона; $\delta_{и.л}$, $\delta_{и.пр}$ – углы отклонения левой и правой секции интерцепторов. Аэродинамическими органами управления по тангажу являются управляемый стабилизатор и рули высоты, по крену – элероны и интерцепторы на крыле, по рысканию – руль направления.

В общем виде матрица эффективности управления имеет символьный вид

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & b_{17} & b_{17} \\ b_{21} & b_{21} & 0 & 0 & b_{25} & b_{25} & b_{27} & b_{27} \\ -b_{21} & -b_{21} & 0 & 0 & -b_{25} & -b_{25} & -b_{27} & -b_{27} \\ b_{41} & b_{41} & b_{43} & 0 & b_{45} & b_{45} & b_{47} & b_{47} \\ 0 & 0 & 0 & b_{54} & 0 & 0 & b_{57} & -b_{57} \\ b_{61} & -b_{61} & 0 & b_{64} & b_{65} & -b_{65} & b_{67} & -b_{67} \\ b_{71} & -b_{71} & 0 & b_{74} & b_{75} & -b_{75} & b_{77} & -b_{77} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Исследования осуществлялись для 8 режимов полета, параметры которых приведены в таблице 2.4.

Таб. 2.4. Параметры исследуемых режимов полета

Параметры полета		Номер режима							
		1	2	3	4	5	6	7	8
H	м	451,1			1066,8		1524	2926,1	7010,4
	ft	1480			3500		5000	9600	23000
V_{CAS}	км/ч	249,8	286,8	296	259	296	370	421,8	481
	kts	135	155	160	140	160	200	228	260

Расположение исследуемых режимов полета на карте эксплуатационных высот и скоростей показано на рисунке 2.19.

Числовые значения коэффициентов матрицы эффективности управления для всех исследуемых режимов полета приведены в таблице 2.5.

Исследование проводилось для всех возможных сценариев, определяемых различным сочетанием управляющих воздействий и возникающих отказов, приведенных в таблице 2.6.

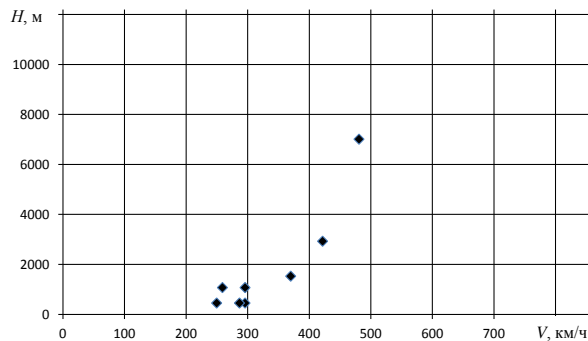


Рис. 2.19. Режимы полета на карте эксплуатационных высот и скоростей

Табл. 2.5. Коэффициенты матрицы B по режимам полета

b_*	Режим							
	1	2	3	4	5	6	7	8
b_{17}	0,0052	0,0237	0,0119	0,0187	0,0118	0,0170	0,0306	0,0438
b_{21}	0,0178	0,0155	0,0150	0,0093	0,0148	-0,0021	-0,0026	-0,0024
b_{23}	-0,0084	-0,0094	-0,0102	-0,0091	-0,0099	0	0	0
b_{25}	-0,001	-0,0011	-0,0012	-0,001	-0,0011	-0,0009	-0,0012	-0,001
b_{27}	-0,0070	-0,0075	-0,0065	-0,0082	-0,0063	-0,0022	-0,0031	-0,0030
b_{41}	-0,3378	-0,5089	-0,5367	-0,3579	0,5366	-0,7353	-1,0166	-1,2216
b_{43}	-1,3433	-1,5634	-1,8603	-1,3313	-1,8567	-2,5461	-2,8291	-4,0923
b_{45}	-0,0501	-0,0634	-0,0645	-0,0544	-0,0644	-0,0814	-0,1278	-0,1698
b_{47}	0,1271	0,1384	0,1337	0,1442	0,1337	0,0856	0,1230	0,1620
b_{54}	-0,0054	-0,0060	-0,0063	-0,0052	-0,0061	-0,0013	-0,0015	-0,0011
b_{57}	-0,0022	-0,0020	-0,0020	-0,0023	-0,0020	-0,0001	-0,0001	-0,0001
b_{61}	0,2052	0,2693	0,2867	0,2198	0,2863	0,3895	0,5794	0,7527
b_{64}	-0,1973	-0,2659	-0,2915	-0,2027	-0,2910	-0,3929	-0,5604	-0,5505
b_{65}	0,5419	0,7736	0,8173	0,5463	0,8156	0,7942	1,1788	1,3880
b_{67}	-1,6357	-2,0488	-3,6434	-2,8683	-3,6385	-2,2845	-3,6055	-5,3949
b_{71}	-0,0092	-0,0121	-0,0129	-0,0099	-0,0129	-0,0156	-0,0261	-0,0339
b_{74}	-3,1463	-4,0119	-4,3539	-3,2558	-4,3476	-4,3085	-6,9865	-8,4561
b_{75}	-0,0579	-0,0435	-0,0612	-0,0507	-0,0614	-0,0287	-0,0381	-0,0144
b_{77}	0,2973	0,3142	0,2972	0,3441	0,2972	0,3194	0,5032	0,6434

Табл. 2.6. Виды отказов приводов

Обозначение	Наименование
Elv_L	Отказ привода левого руля высоты
Elv_LR	Отказ приводов всех рулей высоты
Stab	Отказ привода стабилизатора
Rud	Отказ привода руля направления
Ail_L	Отказ привода левого элерона
Ail_LR	Отказ приводов всех элеронов
Int_L	Отказ приводов левой секции интерцепторов
Int_LR	Отказ приводов всех интерцепторов

В таблице 2.7 приведены нормы условий разрешимости задачи реконфигурации для всех исследуемых отказов и режимов полета, вычисленные

по формуле (2.87). Данные условия априорно позволяют оценить точность решения задачи реконфигурации без учета нелинейных свойств модели и ограничений на отклонения поверхностей управления.

Табл. 2.7. Нормы условий разрешимости задачи реконфигурации

Режим	Вид отказа							
	Elv_L	Elv_LR	Stab	Rud	Ail_L	Ail_LR	Int_L	Int_LR
1	7×10^{-16}	3×10^{-16}	5×10^{-16}	2×10^{-14}	1×10^{-15}	4×10^{-16}	1×10^{-14}	0,0074
2	4×10^{-16}	2×10^{-16}	3×10^{-16}	1×10^{-14}	8×10^{-16}	6×10^{-16}	2×10^{-15}	0,0335
3	7×10^{-16}	3×10^{-16}	3×10^{-16}	7×10^{-14}	3×10^{-15}	7×10^{-16}	2×10^{-14}	0,0168
4	3×10^{-16}	3×10^{-17}	1×10^{-16}	6×10^{-14}	5×10^{-16}	2×10^{-16}	1×10^{-14}	0,0264
5	1×10^{-15}	6×10^{-16}	2×10^{-16}	7×10^{-14}	1×10^{-15}	7×10^{-16}	7×10^{-15}	0,0167
6	5×10^{-16}	3×10^{-16}	7×10^{-16}	1×10^{-12}	1×10^{-16}	1×10^{-15}	4×10^{-14}	0,0240
7	1×10^{-15}	3×10^{-16}	9×10^{-16}	6×10^{-14}	5×10^{-16}	1×10^{-15}	2×10^{-15}	0,0433
8	1×10^{-15}	6×10^{-17}	1×10^{-15}	8×10^{-14}	6×10^{-16}	6×10^{-15}	3×10^{-15}	0,0619

Из таблицы видно, что исследуемый самолет обладает достаточной функциональной избыточностью для точной компенсации всех исследуемых отказов, за исключением отказа всех секций интерцепторов, причину которого проанализируем.

Значения ненулевых коэффициентов матрицы реконфигурации для всех исследуемых отказов и режимов полета приведены в таблицах 2.8–2.15, вычисленных по формуле (2.89) при $\Theta = 0$.

Табл. 2.8. Коэффициенты матрицы реконфигурации ($H = 451,1$ м, $V = 249,8$ км/ч)

Ка- нал	Вид отказа										
	Elv_L	Elv_LR		Stab	Rud	Ail_L	Ail_LR		Int_L	Int_LR	
$\delta_{в.л}$	0	0	0	0,38	-136,3	0,276	1,219	-1,253	111	55,3	-55,7
$\delta_{в.пр}$	0,997	0	0	0,38	136,3	-0,336	-1,253	1,219	-111	-55,7	55,3
φ	0,004	1,333	1,333	0	0	0,081	0,046	0,046	0	0,003	0,003
δ_n	-0,007	-0,004	0,004	0	0	0,002	0,009	-0,009	0,815	0,407	-0,407
$\delta_{э.л}$	0,364	-14,3	-14,7	10,9	55,1	0	0	0	-45	-22,5	22,5
$\delta_{э.пр}$	-0,444	-14,7	-14,3	10,9	-55,1	-0,752	0	0	45	22,5	-22,5
$\delta_{и.л}$	0,009	0,005	-0,005	0	1,23	-0,003	-0,011	0,011	0	0	0
$\delta_{и.пр}$	-0,009	-0,005	0,005	0	-1,23	0,003	0,011	-0,011	1	0	0

Табл. 2.9. Коэффициенты матрицы реконфигурации ($H = 451,1$ м, $V = 286,8$ км/ч)

Ка- нал	Вид отказа										
	Elv_L	Elv_LR		Stab	Rud	Ail_L	Ail_LR		Int_L	Int_LR	
$\delta_{в.л}$	0	0	0	0,38	-136,3	0,276	1,219	-1,253	111	55,3	-55,7
$\delta_{в.пр}$	0,997	0	0	0,38	136,3	-0,336	-1,253	1,219	-111	-55,7	55,3
φ	0,004	1,333	1,333	0	0	0,081	0,046	0,046	0	0,003	0,003
δ_n	-0,007	-0,004	0,004	0	0	0,002	0,009	-0,009	0,815	0,407	-0,407
$\delta_{э.л}$	0,364	-14,3	-14,7	10,9	55,1	0	0	0	-45	-22,5	22,5
$\delta_{э.пр}$	-0,444	-14,7	-14,3	10,9	-55,1	-0,752	0	0	45	22,5	-22,5
$\delta_{и.л}$	0,009	0,005	-0,005	0	1,23	-0,003	-0,011	0,011	0	0	0
$\delta_{и.пр}$	-0,009	-0,005	0,005	0	-1,23	0,003	0,011	-0,011	1	0	0

Табл. 2.10. Коэффициенты матрицы реконфигурации ($H = 451,1$ м, $V = 296$ км/ч)

Ка- нал	Вид отказа										
	Elv_L	Elv_LR		Stab	Rud	Ail_L	Ail_LR		Int_L	Int_LR	
$\delta_{в.л}$	0	0	0	0,49	-259,8	0,253	1,302	-1,349	164	82,3	-82,7
$\delta_{в.пр}$	0,995	0	0	0,49	259,8	-0,337	-1,349	1,302	-164	-82,7	82,3
φ	0,005	1,024	1,024	0	0	0,086	0,048	0,048	0	0,043	0,043
δ_n	-0,004	-0,002	0,002	0	0	0,001	0,005	-0,005	0,635	0,317	-0,317
$\delta_{э.л}$	0,323	-10,4	-10,8	10,36	97,99	0	0	0	-62,2	-31,1	31,1
$\delta_{э.пр}$	-0,430	-10,8	-10,4	10,36	-97,99	-0,777	0	0	62,2	31,1	-31,1
$\delta_{и.л}$	0,006	0,003	-0,003	0	1,575	-0,002	-0,008	0,008	0	0	0
$\delta_{и.пр}$	-0,006	-0,003	0,003	0	-1,575	0,002	0,008	-0,008	1	0	0

Табл. 2.11. Коэффициенты матрицы реконфигурации ($H = 1066,8$ м, $V = 259$ км/ч)

Ка- нал	Вид отказа										
	Elv_L	Elv_LR		Stab	Rud	Ail_L	Ail_LR		Int_L	Int_LR	
$\delta_{в.л}$	0	0	0	0,484	-164,3	0,27	1,12	-1,17	145,38	72,3	-73,1
$\delta_{в.пр}$	0,993	0	0	0,484	164,3	-0,37	-1,17	1,12	-145,38	-73,1	72,3
φ	0,007	1,033	1,033	0	0	0,1	0,055	0,055	0	0,1	0,1
δ_n	-0,006	-0,003	0,003	0	0	0,002	0,007	-0,007	0,885	0,442	-0,442
$\delta_{э.л}$	0,373	-9,131	-9,569	9,052	71,9	0	0	0	-63,578	-31,76	31,82
$\delta_{э.пр}$	-0,499	-9,569	-9,131	9,052	-71,9	-0,72	0	0	63,578	31,82	-31,76

Окончание таблицы 2.11

Ка- нал	Вид отказа										
	Elv_L	Elv_LR		Stab	Rud	Ail_L	Ail_LR		Int_L	Int_LR	
$\delta_{и.л}$	0,007	0,003	-0,003	0	1,13	-0,002	-0,008	0,008	0	0	0
$\delta_{и.пр}$	-0,007	-0,003	0,003	0	-1,13	0,002	0,008	-0,008	1	0	0

Табл. 2.12. Коэффициенты матрицы реконфигурации ($H = 1066,8$ м, $V = 296$ км/ч)

Ка- нал	Вид отказа										
	Elv_L	Elv_LR		Stab	Rud	Ail_L	Ail_LR		Int_L	Int_LR	
$\delta_{в.л}$	0	0	0	1,936	-256,6	0,238	1,3	-1,36	168,25	83,8	-84,4
$\delta_{в.пр}$	0,991	0	0	1,936	256,6	-0,351	-1,36	1,3	-168,25	-84,4	83,8
φ	0,002	0,258	0,258	0	0	0,029	0,016	0,016	0	-0,242	-0,242
$\delta_{и}$	-0,004	-0,002	0,002	0	0	0,001	0,005	-0,005	0,656	0,328	-0,328
$\delta_{э.л}$	0,303	-7,701	-8,078	30,545	96,7	0	0	0	-63,41	-31,69	31,72
$\delta_{э.пр}$	-0,447	-8,078	-7,701	30,545	-96,7	-0,778	0	0	63,41	31,72	-31,69
$\delta_{и.л}$	0,006	0,003	-0,003	0	1,525	-0,002	-0,008	0,008	0	0	0
$\delta_{и.пр}$	-0,006	-0,003	0,003	0	-1,525	0,002	0,008	-0,008	1	0	0

Табл. 2.13. Коэффициенты матрицы реконфигурации ($H = 1524$ м, $V = 370$ км/ч)

Ка- нал	Вид отказа										
	Elv_L	Elv_LR		Stab	Rud	Ail_L	Ail_LR		Int_L	Int_LR	
$\delta_{в.л}$	0	0	0	2,33	2427,4	0,732	1,250	-0,821	-373,44	-186,3	187,2
$\delta_{в.пр}$	0,683	0	0	2,33	-2427,4	-0,034	-0,821	1,250	373,44	187,2	-186,3
φ	0,068	0,214	0,214	0	0	-0,150	-0,092	-0,092	0	-0,3	-0,3
$\delta_{и}$	0	0	0	0	0	0	0	0	0,154	0,08	-0,08
$\delta_{э.л}$	0,777	1,41	0,93	-5,45	-1171	0	0	0	180,31	90,37	-89,94
$\delta_{э.пр}$	-0,036	0,93	1,41	-5,45	1171	-0,630	0	0	-180,31	-89,94	90,37
$\delta_{и.л}$	-0,002	-0,001	0,001	0	6,5	0,001	0,003	-0,003	0	0	0
$\delta_{и.пр}$	0,002	0,001	-0,001	0	-6,5	-0,001	-0,003	0,003	1	0	0

Табл. 2.14. Коэффициенты матрицы реконфигурации ($H = 2926,1$ м, $V = 421,8$ км/ч)

Ка- нал	Вид отказа										
	Elv_L	Elv_LR		Stab	Rud	Ail_L	Ail_LR		Int_L	Int_LR	
$\delta_{в.л}$	0	0	0	1,912	868,320	0,767	1,305	-0,844	-115,8	-57,4	58,4
$\delta_{в.пр}$	0,655	0	0	1,912	-868,320	-0,011	-0,844	1,305	115,8	58,4	-57,4
φ	0,090	0,261	0,261	0	0	-0,198	-0,121	-0,121	0	-0,41	-0,41
$\delta_{и}$	0,001	0,001	-0,001	0	0	0	-0,001	0,001	0,133	0,067	-0,067

Ка- нал	Вид отказа										
	Elv_L	Elv_LR		Stab	Rud	Ail_L	Ail_LR		Int_L	Int_LR	
$\delta_{э.л}$	0,759	1,316	0,851	-4,143	-404,1	0	0	0	53,9	27,21	-26,7
$\delta_{э.пр}$	-0,011	0,851	1,316	-4,143	404,1	-0,638	0	0	-53,9	-26,7	27,21
$\delta_{и.л}$	-0,007	-0,004	0,004	0	7,500	0,003	0,009	-0,009	0	0	0
$\delta_{и.пр}$	0,007	0,004	-0,004	0	-7,500	-0,003	-0,009	0,009	1	0	0

Табл. 2.15. Коэффициенты матрицы реконфигурации (H=7010,4 м, V=481 км/ч)

Ка- нал	Вид отказа										
	Elv_L	Elv_LR		Stab	Rud	Ail_L	Ail_LR		Int_L	Int_LR	
$\delta_{в.л}$	0	0	0	2,513	285	0,719	1,276	-0,860	-51,998	-25,5	26,5
$\delta_{в.пр}$	0,697	0	0	2,513	-285	-0,032	-0,860	1,276	51,998	26,5	-25,5
φ	0,060	0,199	0,199	0	0	-0,137	-0,083	-0,083	0	-0,37	-0,37
$\delta_{и}$	0,003	0,002	-0,002	0	0	-0,001	-0,004	0,004	0,182	0,1	-0,1
$\delta_{э.л}$	0,761	1,434	0,966	-6,032	-133,9	0	0	0	24,347	12,422	-11,93
$\delta_{э.пр}$	-0,034	0,966	1,434	-6,032	133,9	-0,648	0	0	-24,347	-11,93	12,422
$\delta_{и.л}$	-0,016	-0,010	0,010	0	5,500	0,007	0,021	-0,021	0	0	0
$\delta_{и.пр}$	0,016	0,010	-0,010	0	-5,500	-0,007	-0,021	0,021	1	0	0

Средневзвешенные значения норм матриц реконфигурации по всем рассматриваемым отказам для каждого режима полета приведены в таблице 2.16.

Табл. 2.16. Нормы матриц реконфигурации

Вид отказа	Режим							
	1	2	3	4	5	6	7	8
Elv_L	1,15	1,12	1,13	1,17	1,13	1,04	1,01	1,03
Elv_LR	29,06	25,9	21,25	18,76	15,79	2,4	2,25	2,46
Stab	15,39	13,31	14,66	12,82	43,28	8,38	6,45	9,24
Rud	207,9	1139,51	392,72	253,67	387,78	3812	1354,49	446,66
Ail_L	0,87	0,9	0,89	0,86	0,89	0,98	1,02	0,98
Ail_LR	2,47	2,83	2,65	2,29	2,65	2,12	2,2	2,18
Int_L	169,4	759,67	249,34	224,4	254,28	586,46	180,6	81,2
Int_LR	119,78	537,17	176,31	158,68	179,8	414,69	127,7	57,43

Анализ полученных значений позволяет теоретически оценить взаимную эффективность поверхностей управления для компенсации различных видов и типов отказов. Затемнение значений ячеек таблиц характеризует критичность соответствующего отказа вследствие недостаточной эффективности работоспособных поверхностей управления и ограничений на их

максимальные отклонения. Так, априорно можно утверждать о невозможности компенсации отказов руля направления и интерцепторов ни на одном из режимов полета, возможности частичной компенсации отказов двух рулей высоты и стабилизатора на некоторых режимах полета и почти полной компенсации отказов одного руля высоты и элеронов на всех режимах полета.

Примеры результатов численного моделирования, характеризующих работоспособность и эффективность предлагаемых алгоритмов реконфигурации при отказах рулей высоты, приведены на рисунках 2.20–2.23.

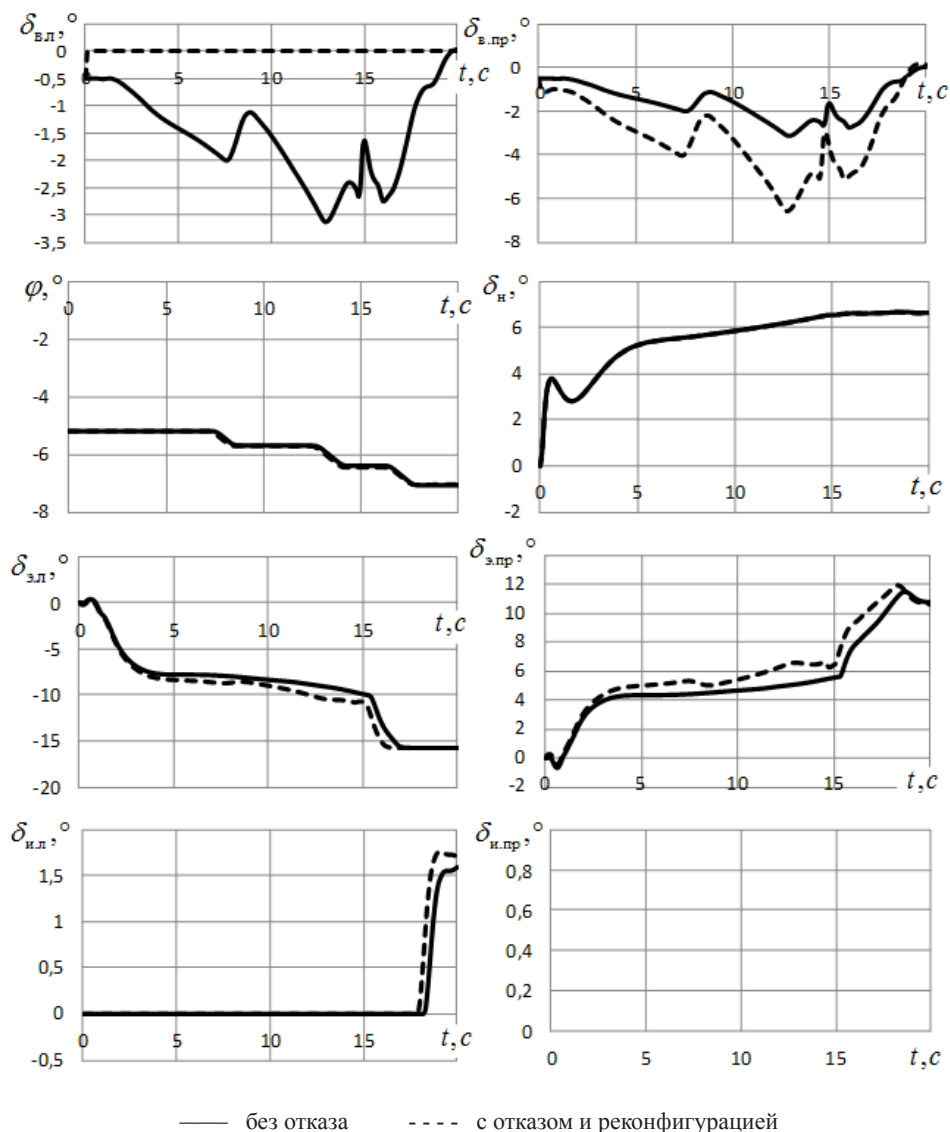


Рис. 2.20. Отклонение поверхностей управления при отказе левого руля высоты и отклонении педалей на 1/3 хода ($H = 451.1$ м, $V = 249.8$ км/ч)

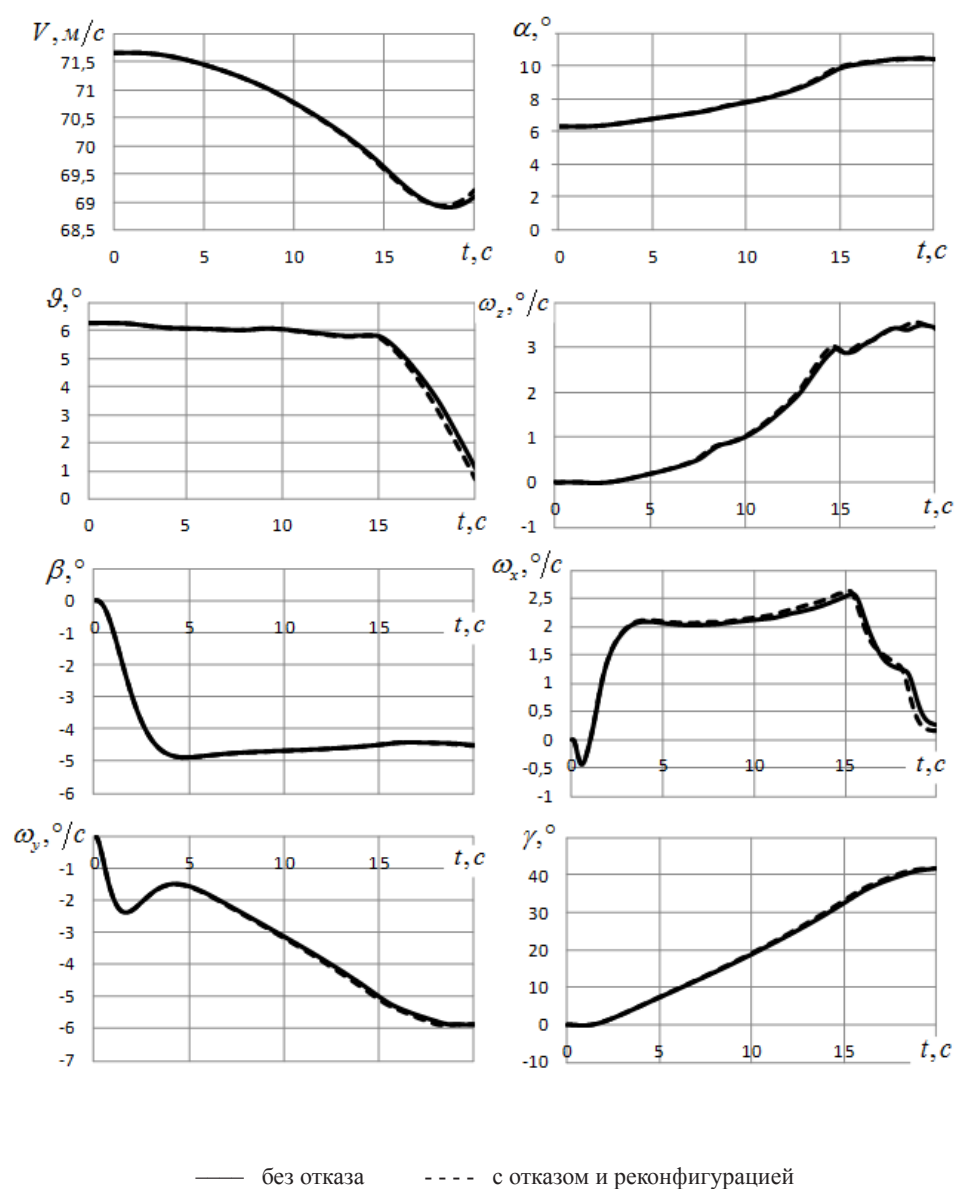
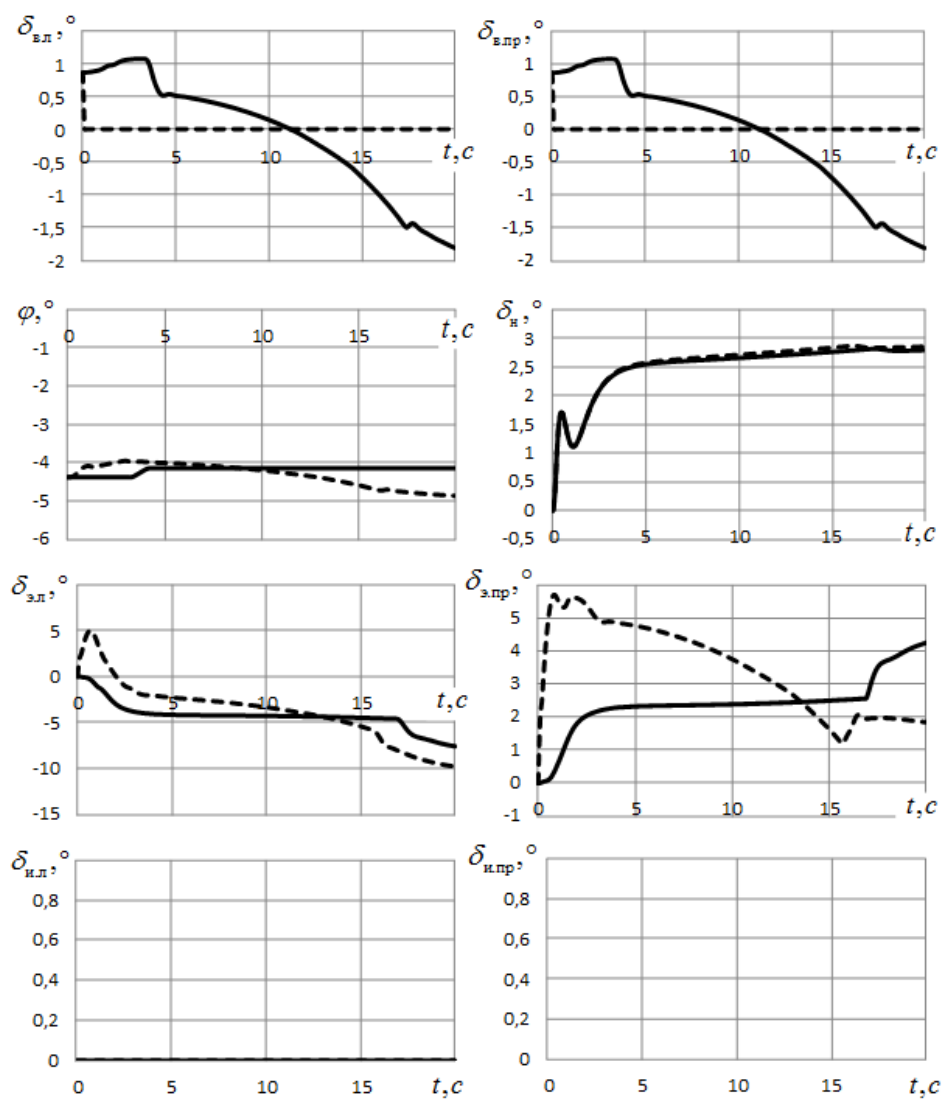


Рис. 2.21. Изменение аэродинамических параметров при отказе левого руля высоты и отклонении педалей на 1/3 хода ($H = 451.1$ м, $V = 249.8$ км/ч)



— без отказа - - - - с отказом и реконфигурацией

Рис. 2.22. Отклонение поверхностей управления при отказе всех рулей высоты отклонении педалей на 1/3 хода ($H = 1524$ м, $V = 370$ км/ч)

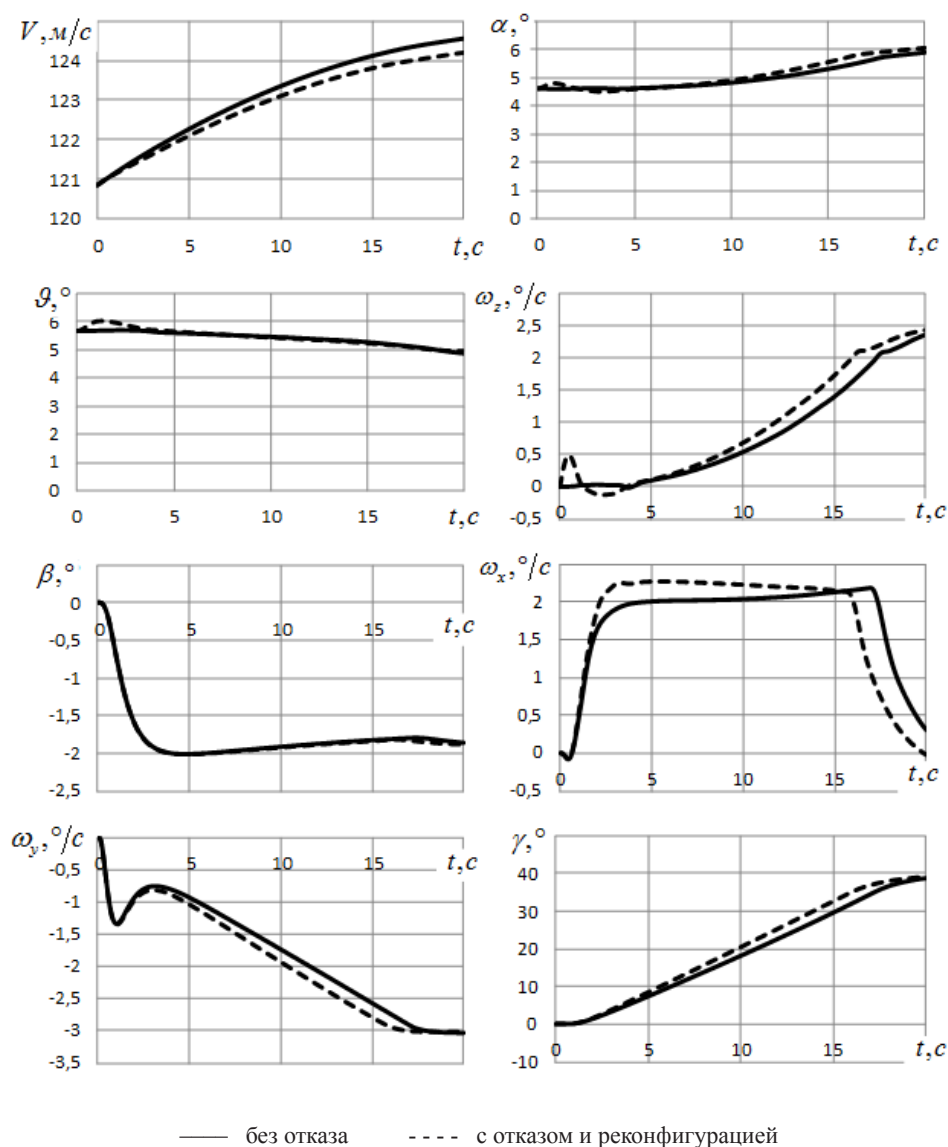


Рис. 2.23. Изменение аэродинамических параметров при отказе всех рулей высоты и отклонении педалей на 1/3 хода ($H = 1524$ м, $V = 370$ км/ч)

Проанализируем причины неразрешимости задачи компенсации отказа двух секций интерцепторов, когда $F = \text{diag}([1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0])$, а матрица эффективности управления ВС с отказом СУ имеет вид

$$B_f = BF = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{21} & b_{21} & 0 & 0 & b_{25} & b_{25} & 0 & 0 \\ -b_{21} & -b_{21} & 0 & 0 & -b_{25} & -b_{25} & 0 & 0 \\ b_{41} & b_{41} & b_{43} & 0 & b_{45} & b_{45} & 0 & 0 \\ 0 & 0 & 0 & b_{54} & 0 & 0 & 0 & 0 \\ b_{61} & -b_{61} & 0 & b_{64} & b_{65} & -b_{65} & 0 & 0 \\ b_{71} & -b_{71} & 0 & b_{74} & b_{75} & -b_{75} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (2.91)$$

В соответствии с теоремой 2.6 последовательно определим в аналитическом виде соответствующие делители нуля матрицы B_f

$$\bar{B}_f^L = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \bar{B}_f^{LR} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\bar{B}_f^R = \begin{bmatrix} -\frac{b_{25}}{b_{21}} & 0 & 0 \\ -\frac{b_{25}}{b_{21}} & 0 & 0 \\ -2\frac{p_2}{b_{21}b_{43}} & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \bar{B}_f^{RL} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2\frac{p_2}{b_{25}b_{43}} & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{b_{21}}{b_{25}} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{b_{21}}{b_{25}} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix},$$

обратимую матрицу

$$\overline{\overline{B}}_f^{L RT} B_f \overline{\overline{B}}_f^{R LT} = \begin{bmatrix} 0 & 4p_2 \frac{b_{21}}{b_{25}b_{43}} & 0 & -2\frac{p_8}{b_{25}} & -2\frac{p_8}{b_{25}} \\ 0 & b_{43} - 2p_2 \frac{b_{41}}{b_{25}b_{43}} & 0 & \frac{b_{21}b_{41}}{b_{25}} + b_{45} & \frac{b_{21}b_{41}}{b_{25}} + b_{45} \\ 0 & 0 & b_{54} & 0 & 0 \\ -2b_{61} & -2p_2 \frac{b_{61}}{b_{25}b_{43}} & b_{64} & \frac{b_{21}b_{61}}{b_{25}} + b_{65} & \frac{b_{21}b_{61}}{b_{25}} - b_{65} \\ -2b_{71} & -2p_2 \frac{b_{71}}{b_{25}b_{43}} & b_{74} & \frac{b_{21}b_{71}}{b_{25}} + b_{75} & \frac{b_{21}b_{71}}{b_{25}} - b_{75} \end{bmatrix},$$

псевдообратную матрицу

$$B_f^+ = \overline{\overline{B}}_f^{R LT} \left(\overline{\overline{B}}_f^{L RT} B_f \overline{\overline{B}}_f^{R LT} \right)^{-1} \overline{\overline{B}}_f^{L RT} = \begin{bmatrix} 0 & p_4 & -p_4 & -b_{25} \frac{p_2}{p_9} & -\frac{p_5}{2p_8} & \frac{b_{75}}{p_1} & -\frac{b_{65}}{p_1} & 0 \\ 0 & p_4 & -p_4 & -b_{25} \frac{p_2}{p_9} & \frac{p_5}{2p_8} & -\frac{b_{75}}{p_1} & \frac{b_{65}}{p_1} & 0 \\ 0 & -p_7 & p_7 & b_{43} \frac{p_8}{p_9} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{b_{54}} & 0 & 0 & 0 \\ 0 & p_3 & -p_3 & b_{21} \frac{p_2}{p_9} & -\frac{p_6}{2p_8} & -\frac{b_{71}}{p_1} & \frac{b_{61}}{p_1} & 0 \\ 0 & p_3 & -p_3 & b_{21} \frac{p_2}{p_9} & \frac{p_6}{2p_8} & \frac{b_{71}}{p_1} & -\frac{b_{61}}{p_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

а также оптимальное решение задачи реконфигурации (2.90)

$$\hat{H}_{\min} = B_f^+ \Delta B_f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & p_{10} & p_{11} \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{11} & p_{10} \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{14} & p_{14} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{b_{57}}{b_{54}} & -\frac{b_{57}}{b_{54}} \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{12} & p_{13} \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{13} & p_{12} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\begin{aligned}
 \text{где } p_1 &= 2(b_{61}b_{75} - b_{65}b_{71}), \quad p_2 = b_{21}b_{45} - b_{25}b_{41}, \quad p_3 = \frac{2b_{25}b_{41}^2 - 2b_{21}b_{45}b_{41} + b_{25}b_{43}^2}{4p_9}, \\
 p_4 &= \frac{b_{21}b_{43}^2 + 2b_{21}b_{45}^2 - 2b_{25}b_{41}b_{45}}{4p_9}, \quad p_5 = \frac{b_{64}b_{75} - b_{65}b_{74}}{b_{54}}, \quad p_6 = \frac{b_{61}b_{74} - b_{64}b_{71}}{b_{54}}, \\
 p_7 &= \frac{b_{43}(b_{21}b_{41} + b_{25}b_{45})}{2p_9}, \quad p_8 = b_{21}^2 + b_{25}^2, \quad p_9 = b_{21}^2b_{43}^2 + 2b_{21}^2b_{45}^2 - 4b_{21}b_{25}b_{41}b_{45} + 2b_{25}^2b_{41}^2 + b_{25}^2b_{43}^2, \\
 p_{10} &= \frac{b_{67}b_{75} - b_{65}b_{77} - b_{57}p_5}{p_1} + \frac{2b_{27}p_4 - b_{25}b_{47}p_2}{p_9}, \quad p_{11} = \frac{b_{65}b_{77} + b_{57}p_5 - b_{67}b_{75}}{p_1} + \frac{2b_{27}p_4 - b_{25}b_{47}p_2}{p_9}, \\
 p_{12} &= \frac{b_{61}b_{77} - b_{67}b_{71} - b_{57}p_6}{p_1} + \frac{2b_{27}p_3 + b_{21}b_{47}p_2}{p_9}, \quad p_{13} = \frac{b_{67}b_{71} + b_{57}p_6 - b_{61}b_{77}}{p_1} + \frac{2b_{27}p_3 + b_{21}b_{47}p_2}{p_9}, \\
 p_{14} &= \frac{b_{43}(b_{47}b_{21}^2 - b_{27}b_{41}b_{21} + b_{47}b_{25}^2 - b_{27}b_{45}b_{25})}{p_9}.
 \end{aligned}$$

Тогда все множество оптимальных решений задачи реконфигурации может быть записано в соответствии с (2.89) по формуле

$$\hat{H} = \hat{H}_{\min} + \bar{B}_f^R \Omega, \quad (2.92)$$

где Ω – матрица с произвольными элементами.

Заметив, что левый делитель нуля уже удовлетворяет требованию ортогональности, проверим приближенное условие разрешимости (2.87)

$$\left\| \left(\bar{B}_f^L \bar{B}_f^{LT} \right)^{0.5} \bar{B}_f^L B_f = \bar{B}_f^L B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & b_{17} & b_{17} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \right\| = \varepsilon_{\min}, \quad (2.93)$$

из которого с учетом (2.88) следует, что задача реконфигурации в данном случае разрешима с точностью до $\varepsilon_{\min} = \sqrt{2}|b_{17}|$.

Подставим полученное множество решений в (2.51) и получим следующую ошибку реконфигурации

$$\Delta = B_f \hat{H} - \Delta B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -b_{17} & -b_{17} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

норма которой в точности соответствует норме приближенного условия разрешимости (2.93) $\|\Delta\|_2 = \sqrt{2}|b_{17}|$.

Так, например, на одном из режимов полета ($H = 451.1$ м, $V = 286.8$ м/с) матрица эффективности управления в соответствии с таблицей 2.5 имеет следующее числовое значение

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0.0237 & 0.0237 \\ 0.0155 & 0.0155 & -0.0094 & 0 & -0.0011 & -0.0011 & -0.0075 & -0.0075 \\ -0.0155 & -0.0155 & 0.0094 & 0 & 0.0011 & 0.0011 & 0.0075 & 0.0075 \\ -0.5089 & -0.5089 & -1.5634 & 0 & -0.0634 & -0.0634 & 0.1384 & 0.1384 \\ 0 & 0 & 0 & -0.0060 & 0 & 0 & -0.002 & 0.0020 \\ 0.2693 & -0.2693 & 0 & -0.2659 & 0.7736 & -0.7736 & -2.05 & 2.0488 \\ -0.0121 & 0.0121 & 0 & -4.0119 & -0.0435 & 0.0435 & 0.314 & -0.3142 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Тогда все множество оптимальных решений задачи реконфигурации с точностью до $\varepsilon_{\min} = \sqrt{2}|b_{17}| = 0.0335$ имеет вид (2.92), где

$$\hat{H}_{\min} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 253 & -253.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & -253.5 & 253 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.056 & 0.056 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.33 & 0.33 \\ 0 & 0 & 0 & 0 & 0 & 0 & -89.42 & 89.44 \\ 0 & 0 & 0 & 0 & 0 & 0 & 89.44 & -89.42 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \bar{B}_f^R = \begin{bmatrix} 0.0387 & 0 & 0 \\ 0.0387 & 0 & 0 \\ -0.1063 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.94)$$

При $\Omega = 0$ приходим к оптимальному решению, обеспечивающему минимизацию нормы Фробениуса $\|\hat{H}_{\min}\|_2 = 537.168$.

Полученное множество решений может быть использовано и для удовлетворения других требований. Минимизируем, например, L_0 -норму решения, обеспечив минимальное количество ненулевых элементов в матрице реконфигурации H , приняв в матрице свободных переменных Ω все элементы равными нулю, кроме $\omega_{17} = \omega_{18} = 1.5194$. Тогда получим следующее решение задачи реконфигурации

$$H_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 253.1 & -253.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & -253.5 & 253.1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.33 & 0.33 \\ 0 & 0 & 0 & 0 & 0 & 0 & -88.89 & 89.97 \\ 0 & 0 & 0 & 0 & 0 & 0 & 89.97 & -88.89 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Данное решение содержит всего 10 нетривиальных переменных с 4 уникальными значениями и также обеспечивает оптимальное решение задачи реконфигурации в смысле минимума нормы ошибки вида (2.51). При этом произошло незначительное увеличение нормы самого решения $\|H_0\|_2 = 537.169$ с относительной ошибкой $(\|H_0\|_2 - \|\hat{H}_{\min}\|_2) / \|\hat{H}_{\min}\|_2 = 1.86 \cdot 10^{-6}$. Одновременно с этим упрощение оптимального закона управления не изменило его L^2 -норму, совпадающую с максимальным сингулярным числом решения $\|H_0\|_{L^2} = \|\hat{H}_{\min}\|_{L^2} = 537.1674$.

2.3.5. Резюме

Оптимальное решение задачи реконфигурации ИП СУ ВС позволяет в случае неразрешимости задачи по критерию получения точного решения гарантировать приближение решения с заранее заданной максимальной точностью, определяемой с помощью приближенного условия реконфигурируемости. При этом аналитическое представление всего множества решений позволяет сохранить все достоинства решения, полученного методом канонизации, в том числе выполнять настройку коэффициентов законов реконфигурации ИП СУ в условиях нестационарности и получать простые в реализации схемы, позволяющие парировать последствия отказов.

2.4. Робастное аналитическое решение задачи реконфигурации исполнительной подсистемы

2.4.1. Постановка задачи робастной реконфигурации

Приведенные выше методы не предполагают наличия неопределенностей в постановке задачи реконфигурация. Однако в некоторых случаях может отсутствовать достоверная оперативная информация как о текущих параметрах модели ВС, так и о возникших в его СУ отказах [69, 104, 105, 111]. При этом основными требованиями к отказоустойчивой СУ является сохранение (восстановление) своей работоспособности при отказах, в том числе за счет перехода в режим работы, не представляющий опасности для системы и человека [83]. Поэтому все СУ ВС для предотвращения катастрофической ситуации должны иметь возможность работать с некоторой неопределенностью. Правильно синтезированные законы реконфигурации должны уменьшать чувствительность объекта к различным ошибкам и резким изменениям в системе [36, 52].

Регуляторы, которые сохраняют удовлетворительные характеристики при наличии параметрической или структурной неопределенности без изменения своей структуры и параметров, называются робастными [97, 106]. Робастные (грубые) регуляторы – это регуляторы, обеспечивающие приемлемое (в смысле некоторого критерия) качество при наличии параметрических,

сигнальных, функциональных или структурных неопределенностей объекта управления. При этом, как правило, в ходе нормального функционирования объекта коэффициенты регулятора не подстраиваются, а малая чувствительность (грубость, робастность) к различного рода вариациям математической модели объекта достигается за счет специальным образом выбранной структуры регулятора (алгоритма управления). Таким образом, робастные регуляторы относятся к классу ненастраивающихся систем управления [37].

Существует два основных подхода к синтезу робастного управления [98]. В первом случае выбирают некоторые базовые значения параметров модели, а область параметров, к изменению которых управление должно быть робастным, задают в виде аддитивных или мультипликативных неопределенностей, зависящих, как правило, от нормы некоторой матрицы.

Во втором случае область параметров, к изменению которых необходимо обеспечить робастность, задается множеством реализаций моделей объекта и/или отказов. А робастное решение находится в виде пересечения множеств решений каждой задачи, соответствующей конкретной реализации моделей.

Первый подход является наиболее общим и может быть применим для любых динамических моделей. Однако это достоинство становится недостатком при решении задач, в которых известны физические принципы построения модели. В таких задачах возможные неопределенности имеют структурированный характер и представление их в виде неопределенностей общего вида не может привести к оптимальным результатам. Законы управления, полученные на основе общего подхода, будут обладать излишней консервативностью, компенсируя неопределенности максимальной степени.

При наличии априорной информации о системе степень неопределенности можно существенно понизить. Математическая модель ВС является нелинейной. Ее можно линеаризовать при малых отклонениях параметров полета от стационарных для заданных значений высоты и скорости. При исследовании пилотажных характеристик ВС характерные значения высоты и скорости определяют режим полета, который характеризуется конкретными значениями параметров модели. Таким образом, коэффициенты линеаризованной модели зависят от данных параметров полета (режима полета), которые всегда имеют значения, ограниченные областью допустимых условий полета. Кроме того, данные параметры полета не являются полностью независимыми: ВС из-за плотности воздуха не способен осуществлять полет на малых скоростях и больших высотах или на больших скоростях и малых высотах.

Поэтому для робастной реконфигурации управления ВС целесообразно использовать второй подход, в котором задается конечное множество моделей динамики полета на различных режимах и/или множество отказов, для которых априорно нужно обеспечить отказоустойчивость. При этом ставится задача поиска такого компенсирующего управления или его множества,

которое удовлетворяет заданному критерию для всех заданных режимов полета (робастность к режимам) и/или для всех заданных отказов (робастность к отказам). Такой подход к синтезу робастного управления называется мультимодельным [51].

В мультимодельной постановке задача реконфигурации СУ, робастной к изменению режимов полета ВС, заключается в решении задач оптимальной реконфигурации (2.86) для каждого из q исследуемых режимов

$$\begin{aligned} B_1 F H &= B_1 (I - F), \\ &\vdots \\ B_q F H &= B_q (I - F), \end{aligned} \quad (2.95)$$

так, чтобы ошибки реконфигурации по всем режимам

$$\Delta_i = B_i \hat{F} H - B_i (I - F)$$

имели минимально возможную совокупную норму

$$\mathcal{E}_{\text{рек}} = \sqrt{\sum_{i=1}^q \|\Delta_i\|^2} \rightarrow \min.$$

Запишем уравнения (2.95) в блочно-матричном виде

$$\begin{bmatrix} B_1 \\ \vdots \\ B_q \end{bmatrix} F H = \begin{bmatrix} B_1 \\ \vdots \\ B_q \end{bmatrix} (I - F),$$

тогда задача робастной к режимам полета реконфигурации СУ ВС будет заключаться в решении матричного уравнения вида

$$B_f H = \Delta B_f \quad (2.96)$$

где

$$B_f = \begin{bmatrix} B_1 \\ \vdots \\ B_q \end{bmatrix} F, \quad \Delta B_f = \begin{bmatrix} B_1 \\ \vdots \\ B_q \end{bmatrix} (I - F), \quad \Delta = \begin{bmatrix} \Delta_1 \\ \vdots \\ \Delta_q \end{bmatrix}.$$

Аналогичный мультимодельный подход может быть применен для решения задач реконфигурации управления, робастной к множествам отказов, то есть к синтезу пассивных законов реконфигурации. Тогда задача робастной к отказам реконфигурации СУ ВС заключается в решении задач оптимальной реконфигурации (2.86) для каждого из s исследуемых отказов

$$\begin{aligned} B F_1 H &= B (I - F_1), \\ &\vdots \\ B F_s H &= B (I - F_s), \end{aligned} \quad (2.97)$$

так, чтобы совокупная норма ошибок при всех отказах была минимальной.

В этом случае уравнение реконфигурации СУ ВС будет иметь вид (2.96), где

$$\mathbf{B}_f = \begin{bmatrix} \mathbf{B}\mathbf{F}_1 \\ \vdots \\ \mathbf{B}\mathbf{F}_s \end{bmatrix}, \quad \Delta\mathbf{B}_f = \begin{bmatrix} \mathbf{B}(\mathbf{I} - \mathbf{F}_1) \\ \vdots \\ \mathbf{B}(\mathbf{I} - \mathbf{F}_s) \end{bmatrix}, \quad \Delta = \begin{bmatrix} \Delta_1 \\ \vdots \\ \Delta_s \end{bmatrix}.$$

2.4.2. Робастное решение задачи реконфигурации

С учетом вышеприведенного на основе теоремы 2.6 сформулируем теорему о робастном решении задачи реконфигурации СУ ВС при отказах ИП [18].

Теорема 2.7. Задача робастной реконфигурации системы управления вида

$$\mathbf{B}_f \mathbf{H} = \Delta\mathbf{B}_f$$

разрешима с точностью до $\varepsilon_{\text{доп}}$ тогда и только тогда, когда выполняется приближенное условие разрешимости

$$\|\Delta = \mathbf{B}_f \hat{\mathbf{H}} - \Delta\mathbf{B}_f\| = \left\| \left(\bar{\mathbf{B}}_f^L \bar{\mathbf{B}}_f^{LT} \right)^{0.5} \bar{\mathbf{B}}_f^L \Delta\mathbf{B}_f \right\| \leq \varepsilon,$$

при этом все множество робастных решений, минимизирующих норму обобщённой невязки

$$\varepsilon_{\text{рек}} = \|\Delta\| = \left\| \left(\bar{\mathbf{B}}_f^L \bar{\mathbf{B}}_f^{LT} \right)^{0.5} \bar{\mathbf{B}}_f^L \Delta\mathbf{B}_f \right\| \rightarrow \min,$$

описывается формулой

$$\hat{\mathbf{H}} = \mathbf{B}_f^+ \Delta\mathbf{B}_f + \bar{\mathbf{B}}_f^R \Theta = \bar{\mathbf{B}}_f^{LT} \left(\bar{\mathbf{B}}_f^{RT} \mathbf{B}_f \bar{\mathbf{B}}_f^{LT} \right)^{-1} \bar{\mathbf{B}}_f^{RT} \Delta\mathbf{B}_f + \bar{\mathbf{B}}_f^R \Theta, \quad (2.98)$$

где $\bar{\mathbf{B}}_f^L$, $\bar{\mathbf{B}}_f^R$ – левый и правый делителя нуля полного ранга матрицы \mathbf{B}_f , удовлетворяющие условиям $\bar{\mathbf{B}}_f^L \mathbf{B}_f = 0$, $\mathbf{B}_f \bar{\mathbf{B}}_f^R = 0$; $\bar{\mathbf{B}}_f^{LR}$, $\bar{\mathbf{B}}_f^{RL}$ – правый и левый делителя нуля полного ранга матриц $\bar{\mathbf{B}}_f^L$, $\bar{\mathbf{B}}_f^R$, удовлетворяющие условиям $\bar{\mathbf{B}}_f^L \bar{\mathbf{B}}_f^{LR} = 0$, $\bar{\mathbf{B}}_f^{RL} \bar{\mathbf{B}}_f^R = 0$, Θ – матрица с произвольными элементами, а единственное решение с минимальной мощностью реконфигурации $\mu_{\min} = \|\hat{\mathbf{H}}_{\min}\| = \min$ определяется при $\Theta = 0$ в виде

$$\hat{\mathbf{H}}_{\min} = \mathbf{B}_f^+ \Delta\mathbf{B}_f.$$

2.4.3. Пример робастного решения задачи реконфигурации

Поставим задачу реконфигурации управления СУ, робастной ко всем исследуемым режимам полета самолета, приведенным в предыдущем примере в таблице 2.4. Тогда матрица эффективности управления для всех режимов

полета будет иметь вид $\mathbf{B} = [\mathbf{B}_1^T \quad \dots \quad \mathbf{B}_8^T]^T$, где \mathbf{B}_* – матрицы эффективности управления, соответствующие каждому режиму полета.

Значения коэффициентов ненулевых столбцов матриц робастной реконфигурации $\mathbf{h}(k) = [h_1^T \quad h_2^T \quad \dots \quad h_8^T]^T$, вычисленных по формуле (2.98) для различных отказов, описанных в таблице 2.6, приведены в таблице 2.17.

Значения норм совокупной ошибки и ошибок для каждого режима полета приведены в таблице 2.18.

Анализ таблицы показывает, что за исключением случая отказа руля направления полученные законы управления обеспечивают сохранение текущих характеристик режимов в приемлемом диапазоне.

Графики изменения вектора состояния ВС с работоспособной и реконфигурированной СУ при подаче управляющего сигнала 10° в отказавший канал управления для наихудшего сценария отказа стабилизатора на 7–м режиме полета с максимальной нормой ошибки (0,3854) приведены для примера на рисунке 2.24.

Табл. 2.17. Коэффициенты матриц робастной реконфигурации

	Отказ										
	Elv_L	Elv_LR		Stab	Rud	Ail_L	Ail_LR		Int_L	Int_LR	
h	0	0	0	1,084	2,072	0,1	0,636	-0,542	0,209	0,744	-0,542
h_1	-0,286	0	0	1,084	-2,072	0,086	-0,542	0,636	0,191	-0,542	0,744
h_2	0,201	0,156	0,156	0	0	0,022	0,011	0,011	-0,212	-0,107	-0,107
h_3	0,001	0,001	-0,001	0	0	-0,0001	-0,006	0,006	-0,001	-0,05	0,05
h_4	2,774	2,271	1,757	3,244	-6,09	0	0	0	-0,207	-2,209	2,031
h_5	2,406	1,757	2,271	3,244	6,09	-0,989	0	0	-0,147	2,031	-2,209
h_6	0,132	0,107	0,09	-0,723	-1,39	-0,005	-0,063	0,06	0	0	0
h_7	0,121	0,09	0,107	-0,723	1,39	-0,003	0,06	-0,063	-0,986	0	0

Табл. 2.18. Нормы ошибок робастной реконфигурации

Ошибка	Отказ							
	Elv_L	Elv_LR	Stab	Rud	Ail_L	Ail_LR	Int_L	Int_LR
$\ \Delta_1\ $	0,0851	0,1106	0,1248	3,1528	0,0103	0,1449	0,0923	0,5679
$\ \Delta_2\ $	0,0926	0,1514	0,1689	4,3135	0,0079	0,2875	0,1383	1,2381
$\ \Delta_3\ $	0,0701	0,1181	0,1067	4,5305	0,0142	0,0678	0,0749	0,8294
$\ \Delta_4\ $	0,0622	0,0723	0,0563	3,7648	0,0131	0,0981	0,1380	1,1998
$\ \Delta_5\ $	0,0695	0,1173	0,1042	4,5287	0,0141	0,0674	0,0751	0,8320
$\ \Delta_6\ $	0,0369	0,0462	0,3007	3,9336	0,0299	0,0763	0,1093	0,7981

Ошибка	Отказ							
	Elv_L	Elv_LR	Stab	Rud	Ail_L	Ail_LR	Int_L	Int_LR
$\ \Delta_7\ $	0,1108	0,1281	0,3854	6,1020	0,0048	0,0710	0,0623	0,8765
$\ \Delta_8\ $	0,1303	0,2102	0,1249	6,9290	0,0221	0,2459	0,1452	0,7786
$\ \Delta\ $	0,2451	0,3617	0,5693	13,5821	0,0464	0,4402	0,3080	2,5858

Минимальная степень робастности для исследуемой области режимов полета обеспечивается при отказе руля направления. При этом на некоторых режимах стереотип управления качественно сохраняется, как это показано на рисунке 2.25 для наиболее оптимистичного сценария на 4-м режиме полета с минимальной нормой ошибки (3,7648). Однако во всех подобных случаях существенно снижается эффективность управления, что приводит к необходимости разделения области исследуемых режимов и пересчета матриц реконфигурации для каждой области отдельно. Тем не менее применение рассмотренного подхода позволяет синтезировать реконфигурированное управление, робастное к изменению условий полета и обеспечивающее приемлемое качество управления во всем диапазоне высот и скоростей.

2.4.4. Резюме

Необходимость обеспечения продолжения полета в любой ситуации возможно с дополнительными ограничениями – основная цель реконфигурации СУ ВС, поэтому для предотвращения катастрофической ситуации все методы реконфигурации должны иметь возможность работать с некоторой неопределенностью. В монографии наряду с оптимальными решениями задач реконфигурации получены также робастные (грубые) решения, обеспечивающие минимальную чувствительность законов реконфигурации ИП СУ к изменениям режима полета или параметров отказов и сохраняющие удовлетворительные характеристики управления ВС при наличии параметрических или структурных неопределенностей.

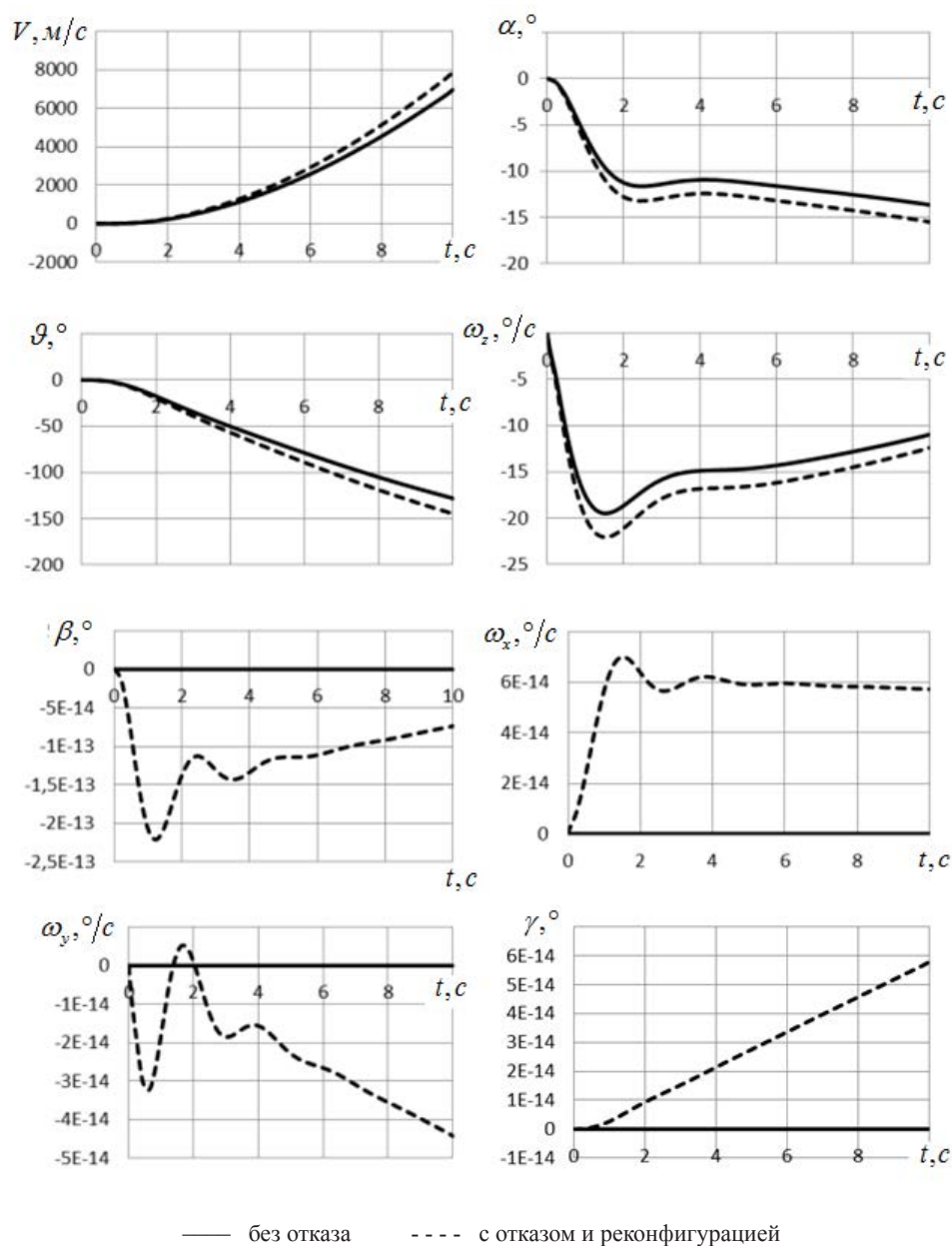


Рис. 2.24. Отклонения состояний при отказе стабилизатора
($H = 2926.1$ м, $V = 421.8$ км/ч)

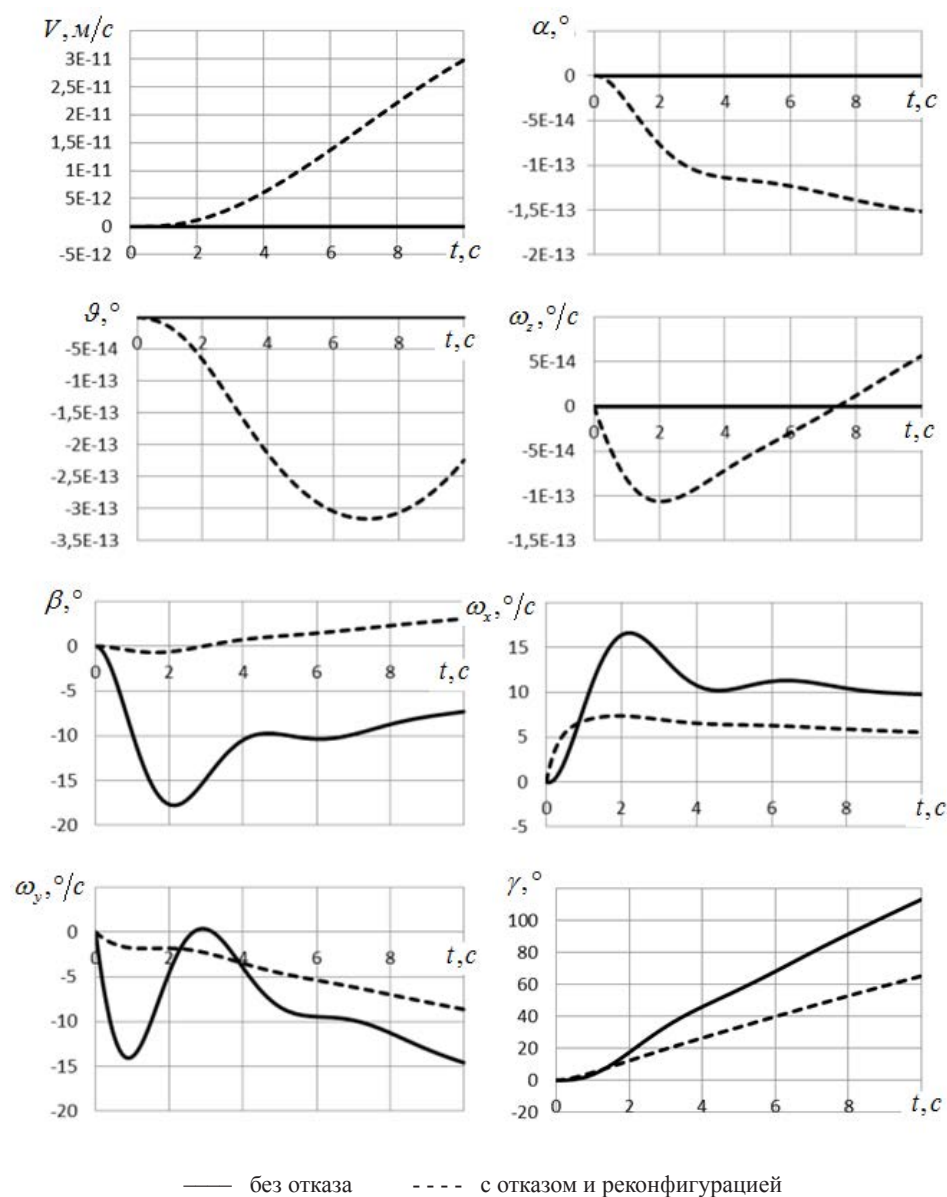


Рис.2.25. Отклонения состояний при отказе руля направления
 ($H = 1066.8$ м, $V = 259$ км/ч)

2.5. Субоптимальное решение задачи реконфигурации исполнительной подсистемы с учетом ограничений на управление

2.5.1. Постановка задачи реконфигурации с учетом ограничений на управление

При перераспределении сигналов с отказавших каналов управления на работоспособные в результате реконфигурации СУ ВС отклонения работоспособных поверхностей управления всегда превышают штатные, и они могут выйти на конструктивные упоры. Поэтому при решении практических задач реконфигурации СУ ВС всегда необходимо учитывать физические ограничения на амплитуды отклонений его поверхностей управления. Законы реконфигурации, рассчитанные без учета таких ограничений, могут привести к полной потере управляемости и устойчивости ВС.

В рассмотренных выше примерах решения задач оптимальной и робастной реконфигурации в некоторых случаях прослеживалось, что для обеспечения минимальной ошибки (максимальной точности) реконфигурации требовались существенные отклонения поверхностей управления. Это зачастую приводило к большим значениям коэффициентов усиления законов реконфигурации и как следствие, выходу поверхностей управления на конструктивные ограничения с последующей потерей устойчивости и управляемости ВС [15, 29, 67, 85, 92, 93]. Несмотря на то что, как правило, требования к ошибкам реконфигурации можно было бы и смягчить, а полученное решение привести к виду, при котором отклонения поверхностей управления по постановке задачи минимальны при рассмотренных выше постановках задачи, добиться этого не удавалось. В таких случаях на реконфигурацию затрачивается чрезмерная мощность управления в работоспособных каналах, которой не хватает для сохранения устойчивости и управляемости ВС [15, 67].

Так, в рассмотренном выше примере решения задачи робастной реконфигурации при возникновении отказа руля направления матрица реконфигурации содержит коэффициенты перераспределения управления на элероны, модуль которых равен 6.09 (таблица 2.17). Это означает, что для создания эквивалентного отклонению руля направления на 1° момента по рысканью необходимо отклонить элероны более чем на 6° . С учетом конструктивных ограничений на отклонение элеронов, например, в 27° , при реконфигурации они выйдут на упоры для создания момента, эквивалентному всего 5° отклонению руля направления (без учета сигналов штатного управления элеронами). В примере аналитической оптимальной реконфигурации (2.94) ситуация еще критичнее – компенсирующие сигналы отказавших интерцепторов необходимо усиливать более чем 14 раз в каналах элеронов и более чем в 30 раз в каналах рулей высоты. То есть на реконфигурацию должна затрачиваться

чрезмерная мощность работоспособных каналов управления, которой не хватает для сохранения устойчивости и управляемости ВС вследствие наличия физических ограничений на отклонение поверхностей управления.

Существует два принципиально различных подхода (и их комбинации) к учету физических ограничений на отклонение поверхностей управления [62, 73, 75, 96, 115]. Первый подход предполагает изменение и ограничение командных сигналов управления [60, 65, 114]. Данный подход здесь не рассматривается, так как он подразумевает изменение штатных алгоритмов работы СУ ВС.

Во втором подходе ограничения учитываются непосредственно в процессе синтеза закона реконфигурации [86, 89]. В [85] описана реализация метода псевдообращения, обеспечивающего ограничение отклонений поверхностей управления на самолете CRCA (Control Reconfigurable Combat Aircraft) фирмы Grumman. В [67] предложен подход, позволяющий оценить предельно допустимое значение нормы решения, при котором система остается устойчивой. В [92] метод псевдообращения модифицирован с использованием весовой диагональной матрицы, элементами которой являются физические ограничения на отклонения поверхностей управления. Этот подход назван в [92] смешиванием управления. Основной недостаток данного подхода состоит в том, что за счет весовой матрицы искусственно снижается эффективность каналов управления, задействованных при реконфигурации. При этом не гарантируется минимальность ошибки реконфигурации и сохранение работы поверхностей управления в линейной зоне на всех режимах. В [1] для учета ограничений используется итерационный алгоритм сужения множества всех возможных решений за счет перераспределения управления с «перегруженного» канала при выходе на упоры поверхностей управления на «недозагруженные», сходимость которого не всегда позволяет получить требуемое решение. В работе [93] для перераспределения управления с учетом ограничений используется понятие виртуального привода. При этом многие практические результаты были получены для односвязных систем с одним входом и одним выходом. Для сложных многосвязных систем до сих пор общепринятых универсальных подходов не существует [115].

Основной отличительной особенностью всех рассмотренных подходов является попытка решить уравнение реконфигурации с максимальной точностью (минимальной ошибкой). В данной работе проблема выхода поверхностей управления на физические ограничения решается принципиально другим способом. Предлагается учитывать ограничения по значениям элементов матрицы реконфигурации или по норме матрицы реконфигурации, характеризующей мощность компенсирующего управления [29].

Использование модифицированного метода оптимального решения задачи реконфигурации позволяет получать приближенные субоптимальные решения с заранее известными при заданной степени приближения минимально возможными мощностью (нормой матрицы реконфигурации) и

ошибкой (нормой матрицы ошибок) реконфигурации. Это позволяет согласованно понижать мощность и повышать ошибку реконфигурации за несколько шагов вплоть до получения практически допустимого решения. За счет увеличения ошибки решения задачи на каждом шаге появляется дополнительная свобода в уменьшении мощности реконфигурации. Уменьшение мощности реконфигурации приводит к уменьшению амплитуд отклонений работоспособных поверхностей управления, на которые перераспределяются сигналы с отказавших каналов управления.

Очевидно, что минимальная норма матрицы реконфигурации, соответствующая полностью работоспособной СУ ВС, равна нулю. Любой отказ и его последующая компенсация приводит к появлению внедиагональных ненулевых элементов в матрице реконфигурации и неизбежному повышению ее нормы. Это, в свою очередь, приводят к снижению запасов управления по работоспособным каналам и степени управляемости ВС.

С физической точки зрения не всегда существует необходимость обеспечивать максимальную точность решения задачи реконфигурации на практике. Решение с меньшей точностью может достаточно эффективно компенсировать отказ, но при этом появляется дополнительная свобода в уменьшении мощности реконфигурации. В результате такая постановка задачи может существенно расширить область компенсируемых отказов на практике.

2.5.2. Субоптимальное решение линейных матричных уравнений

Как показано выше, оптимальное решение (2.85) минимизирует норму как ошибки решения (2.67) уравнения (2.57)

$$\varepsilon_0 = \|\Delta\| = \|L_0 B\| \leq \varepsilon_{\text{доп}}, \quad (2.99)$$

так и самого решения (2.68)

$$\mu_0 = \|X\|_{\min} = \left\| \Sigma_{\max}^{-1} L_{\Sigma} B \right\|, \quad (2.100)$$

которое существует всегда и может быть записано в аналитическом виде. Однако на практике такие требования в явном виде предъявляются далеко не всегда. Более того, как будет показано ниже, чрезмерное ужесточение требований по обеспечению минимальной ошибки решения может привести к получению такого решения, которое нереализуемо на практике ввиду слишком больших значений элементов матрицы X , обуславливающих превышение допустимой нормы $\mu_0 > \mu_{\text{доп}}$.

Покажем далее, что смягчение требования к точности решения уравнения (2.57)

$$\varepsilon_{\text{доп}} \geq \varepsilon_1 > \varepsilon_0 \quad (2.101)$$

позволяет находить решения с ограниченной по постановке задачи нормой $\|X\| \leq \mu_{\text{доп}}$. Представим для этого сингулярное разложение матрицы A (2.54) в следующем виде

$$A = L^T \Sigma R^T = \begin{bmatrix} L_{\max}^T & l_{\min}^T & L_0^T \end{bmatrix} \begin{bmatrix} \Sigma_{\max} & 0 & 0 \\ 0 & \sigma_{\min} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} R_{\max}^T \\ r_{\min}^T \\ R_0^T \end{bmatrix}, \quad (2.102)$$

где Σ_{\max} – диагональная матрица максимальных СЧ; σ_{\min} – минимальное ненулевое СЧ; L_{\max}, R_{\max} – блоки левых и правых матриц сингулярных векторов, соответствующих максимальным СЧ; l_{\min}, r_{\min} – левые и правые сингулярные вектора, соответствующие минимальному ненулевому СЧ.

Подставим (2.102) в исходное уравнение (2.57)

$$\begin{bmatrix} L_{\max}^T & l_{\min}^T & L_0^T \end{bmatrix} \begin{bmatrix} \Sigma_{\max} & 0 & 0 \\ 0 & \sigma_{\min} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} R_{\max}^T \\ r_{\min}^T \\ R_0^T \end{bmatrix} X = B$$

и умножим (2.60) слева на матрицу левых сингулярных векторов L :

$$\begin{bmatrix} \Sigma_{\max} & 0 & 0 \\ 0 & \sigma_{\min} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} R_{\max}^T \\ r_{\min}^T \\ R_0^T \end{bmatrix} X = \begin{bmatrix} L_{\max} \\ l_{\min} \\ L_0 \end{bmatrix} B.$$

Введем обозначения

$$\begin{bmatrix} Y_{\max}^T \\ y_{\min}^T \\ Y_0^T \end{bmatrix} = \begin{bmatrix} R_{\max}^T \\ r_{\min}^T \\ R_0^T \end{bmatrix} X, \quad (2.103)$$

$$\begin{bmatrix} \Delta_{\max} \\ \delta_{\min} \\ \Delta_0 \end{bmatrix} = \begin{bmatrix} L_{\max} \\ l_{\min} \\ L_0 \end{bmatrix} B \quad (2.104)$$

и запишем эквивалентное уравнение

$$\begin{bmatrix} \Sigma_{\max} & 0 & 0 \\ 0 & \sigma_{\min} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Y_{\max}^T \\ y_{\min}^T \\ Y_0^T \end{bmatrix} = \begin{bmatrix} \Delta_{\max} \\ \delta_{\min} \\ \Delta_0 \end{bmatrix}. \quad (2.105)$$

Из (2.63) видно, что Y_0^T может всегда принимать произвольные значения $Y_0^T = \Psi$. Тогда если множество решений эквивалентного уравнения (2.105) записать в виде

$$\begin{bmatrix} Y_{\max}^T \\ y_{\min}^T \\ Y_0^T \end{bmatrix} = \begin{bmatrix} \Sigma_{\max}^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \Delta_{\max} \\ 0 \\ \Psi \end{bmatrix}, \quad (2.106)$$

то ошибка решения (2.106) после подстановки его в (2.105) будет удовлетворять выражению

$$\Delta = \begin{bmatrix} I & 0 & 0 \\ 0 & \delta_{\min} & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \Delta_{\max} \\ 0 \\ \Psi \end{bmatrix} - \begin{bmatrix} \Delta_{\max} \\ \delta_{\min} \\ \Delta_0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\delta_{\min} \\ -\Delta_0 \end{bmatrix},$$

а субоптимальное решение с минимально возможной нормой для данной ошибки, аналогично оптимальному случаю, определяется при $\Psi = 0$ также в единственном виде

$$\begin{bmatrix} Y_{\max}^T \\ y_{\min}^T \\ Y_0^T \end{bmatrix} = \begin{bmatrix} \Sigma_{\max}^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \Delta_{\max} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \Sigma_{\max}^{-1} \Delta_{\max} \\ 0 \\ 0 \end{bmatrix} \quad (2.107)$$

со значением нормы

$$\left\| \begin{bmatrix} Y_{\max}^T \\ y_{\min}^T \\ Y_0^T \end{bmatrix} \right\| = \left\| \begin{bmatrix} \Sigma_{\max}^{-1} \Delta_{\max} \\ 0 \\ 0 \end{bmatrix} \right\|. \quad (2.108)$$

После подстановки множества решений (2.106) в выражение (2.103)

$$\begin{bmatrix} \Sigma_{\max}^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \Delta_{\max} \\ 0 \\ \Psi \end{bmatrix} = \begin{bmatrix} R_{\max}^T \\ r_{\min}^T \\ R_0^T \end{bmatrix} X$$

можно записать с учетом (2.104) решение исходного уравнения (2.57) для степени приближения $\xi = 1$

$$X = \begin{bmatrix} R_{\max} & r_{\min} & R_0 \end{bmatrix} \begin{bmatrix} \Sigma_{\max}^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} L_{\max} B \\ 0 \\ 0 \end{bmatrix} = R_{\max} \Sigma_{\max}^{-1} L_{\max} B,$$

значение нормы которого также совпадает с (2.108)

$$\mu_\xi = \mu_1 = \|X\| = \left\| \begin{bmatrix} \Sigma_{\max}^{-1} \Delta_{\max} \end{bmatrix} \right\| = \left\| \begin{bmatrix} \Sigma_{\max}^{-1} L_{\max} \Delta B \end{bmatrix} \right\|. \quad (2.109)$$

Условие разрешимости исходного уравнения при этом имеет вид

$$\varepsilon_\xi = \varepsilon_1 = \left\| \begin{bmatrix} \delta_{\min} \\ \Delta_0 \end{bmatrix} \right\| = \left\| \begin{bmatrix} l_{\min} B \\ L_0 B \end{bmatrix} \right\| \leq \varepsilon_{\text{доп}}. \quad (2.110)$$

Сравним полученные решения. Найдем коэффициенты эффективности приближения решения по нормам решения ρ_1^μ и ошибки ρ_1^ε как отношения норм (2.100), (2.109) и (2.99), (2.110)

$$\rho_1^\mu = \frac{\mu_0}{\mu_1} = \frac{\left\| \begin{bmatrix} \Sigma_{\max}^{-1} L_{\max} B \\ \sigma_{\min}^{-1} l_{\min} B \end{bmatrix} \right\|}{\left\| \begin{bmatrix} \Sigma_{\max}^{-1} L_{\max} B \end{bmatrix} \right\|} = \frac{\sqrt{\left\| \Sigma_{\max}^{-1} L_{\max} B \right\|^2 + \left\| \sigma_{\min}^{-1} l_{\min} B \right\|^2}}{\left\| \begin{bmatrix} \Sigma_{\max}^{-1} L_{\max} B \end{bmatrix} \right\|}, \quad (2.111)$$

$$\rho_1^\varepsilon = \frac{\varepsilon_0}{\varepsilon_1} = \frac{\left\| L_0 B \right\|}{\left\| \begin{bmatrix} l_{\min} B \\ L_0 B \end{bmatrix} \right\|} = \frac{\left\| L_0 B \right\|}{\sqrt{\left\| L_0 B \right\|^2 + \left\| l_{\min} B \right\|^2}}. \quad (2.112)$$

Из (2.111), (2.112) видно, что при увеличении степени приближения норма решения всегда уменьшается ($\rho_\xi^\mu > 1$), а норма ошибки решения – увеличивается или не изменяется ($\rho_\xi^\varepsilon \leq 1$). Поэтому, если и в первом приближении норма решения μ вновь превышает допустимую, но ошибка все еще находится в допустимой области, необходимо повторить аналогичные расчеты для всех возможных степеней приближений $\xi = 2, 3, \dots, r-1$ с увеличением влияния следующих минимальных СЧ в матрице Σ_{\max} :

$$\mu_\xi = \left\| \Sigma_{1:r-\xi}^{-1} L_{1:r-\xi, 1:m} B \right\|, \quad (2.113)$$

$$\varepsilon_\xi = \left\| L_{r-\xi+1:m, 1:m} B \right\|, \quad (2.114)$$

вплоть до получения решения

$$X_\xi = R_{1:m, 1:r-\xi} \Sigma_{1:r-\xi}^{-1} L_{1:r-\xi, 1:m} B, \quad (2.115)$$

удовлетворяющего практическим требованиям по допустимым нормам $\mu_\xi \leq \mu_{\text{доп}}$, $\varepsilon_\xi \leq \varepsilon_{\text{доп}}$. Невозможность одновременного удовлетворения обоим требованиям говорит о неразрешимости уравнения с заданной точностью.

Количественный анализ полученных значений удобно проводить в табличном виде, как это показано в таблице 2.19.

Табл. 2.19. Пример анализа эффективности приближения решений

ξ	0	1	...	$\rho - 1$
μ_ξ	μ_0	μ_1	...	$\mu_{\rho-1}$
ε_ξ	ε_0	ε_1	...	$\varepsilon_{\rho-1}$
ρ_ξ^μ	—	$\frac{\mu_1}{\mu_0}$...	$\frac{\mu_{\rho-1}}{\mu_{\rho-2}}$
ρ_ξ^ε	—	$\frac{\varepsilon_1}{\varepsilon_0}$...	$\frac{\varepsilon_{\rho-1}}{\varepsilon_{\rho-2}}$

Для визуального анализа результатов можно составить приближенный график взаимозависимости норм решения μ и ошибки решения ε для каждой степени приближения ξ , который будет иметь качественный вид, показанный на рисунке 2.26.

Итоговое решение выбирается на основе оптимального сочетания «выигрыша» в норме решения и «проигрыша» в его точности с учетом заданных ограничений $\varepsilon_{\text{зад}}$ и $\mu_{\text{зад}}$.

В рассматриваемом методе на каждом шаге приближения обеспечивается субоптимальность решения – минимизация норм решения и ошибки для данной степени приближения. С математической точки зрения аналогичный подход применяется в методе главных компонент, который используется для аппроксимации матриц исходных данных за счет снижения размерности задач с потерей наименьшего количества значимой информации. Предлагаемый метод основан на решении обратной задачи – вычислении приближенной обобщенно-обратной матрицы с заданной степенью точности.

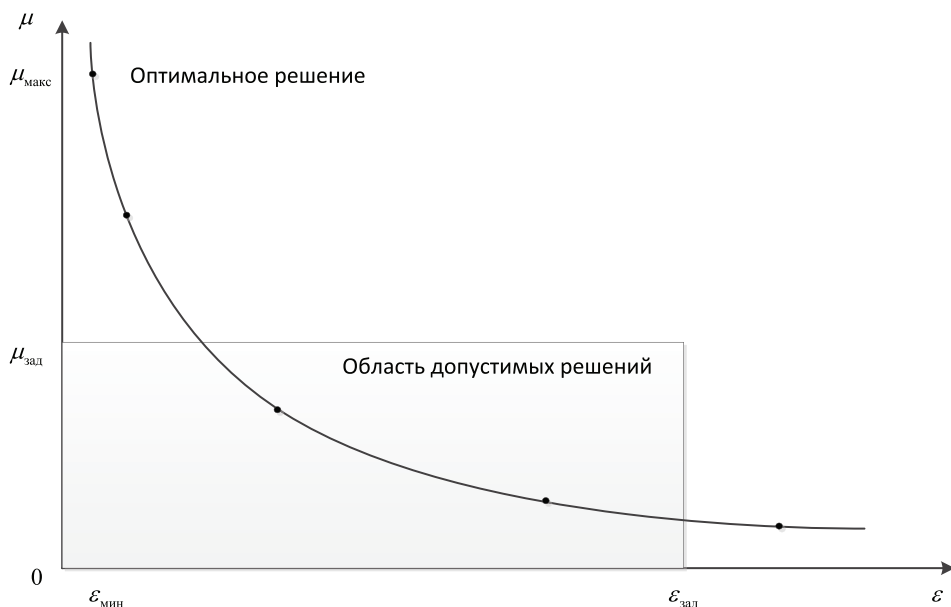


Рис. 2.26. График норм решения и ошибки

2.5.3. Решение задачи реконфигурации с учетом ограничений на управление

Сформулируем на основе вышеприведенных рассуждений методику реконфигурации ИП СУ ВС с учетом ограничений на управление.

1. Определить матрицу эффективности управления $B_{m \times n}$ ВС с работоспособной СУ для текущего режима полета.
2. Сформировать матрицу отказов F ИП СУ.
3. Сформировать в соответствии с (2.8) линейаризованную матрицу эффективности управления B_f ВС с отказавшей ИП СУ.
4. Вычислить в соответствии с (2.54) сингулярное разложение матрицы $B_f = L^T \Sigma R^T$ и определить ее ранг ρ .

$$B_f = \begin{bmatrix} l_1^T & l_2^T & \dots & l_\xi^T & \dots & l_\rho^T & L_0^T \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \dots & 0 & \dots & 0 & 0 \\ 0 & \sigma_2 & \dots & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \sigma_\xi & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & \dots & \sigma_\rho & 0 \\ 0 & 0 & \dots & 0 & \dots & 0 & 0 \end{bmatrix} \begin{bmatrix} r_1^T \\ r_2^T \\ \vdots \\ r_\xi^T \\ \vdots \\ r_\rho^T \\ R_0^T \end{bmatrix}, \quad (2.116)$$

где L, R – ортогональные матрицы левых l_ξ и r_ξ правых сингулярных векторов; матрица сингулярных чисел Σ имеет диагональный вид с ранжированными по убыванию сингулярными числами σ_ξ соответствующими степени приближения ξ .

5. Определить границы допустимых решений путем определения значений $\mu_{\text{зад}}$ и $\varepsilon_{\text{зад}}$. Точное определение границ допустимых решений $\mu_{\text{доп}}$ и $\varepsilon_{\text{доп}}$ является отдельной сложной задачей, так как они зависят от многих факторов: штатных сигналов управления, ограничений на отклонения поверхностей управления, динамических характеристик приводов, параметров отказов, эффективностей поверхностей управления и т.д. [67, 73, 75, 93, 117]. Анализ результатов модельных экспериментов показывает, что для сохранения устойчивости и управляемости ВС после реконфигурации мощность реконфигурации не должна превышать нормы матрицы эффективности управления ВС с работоспособной СУ

$$\mu_{\text{зад}} \leq \|B\|, \quad (2.117)$$

$$\varepsilon_{\text{зад}} \leq \|B\|. \quad (2.118)$$

В этом случае при отсутствии резких маневров ВС в большинстве случаев оставшегося запаса по управлению оказывается достаточно для компенсации отказов и сохранения характеристик устойчивости и управляемости ВС.

6. Рассчитать мощность μ_ξ и ошибку ε_ξ решений для всех степеней приближения $\xi = 0, 1, \dots, \rho - 1$ по формулам

$$\mu_\xi = \|H_\xi\| = \|\Sigma_{1:\rho-\xi}^{-1} L_{1:\rho-\xi, 1:n} \Delta B\|, \quad (2.119)$$

$$\varepsilon_\xi = \|L_{\rho-\xi+1:n, 1:n} B\|. \quad (2.120)$$

7. Провести анализ эффективности приближения решений по формулам

$$\rho_\xi^\mu = \frac{\mu_{\xi-1}}{\mu_\xi}, \quad (2.121)$$

$$\rho_\xi^\varepsilon = \frac{\varepsilon_{\xi-1}}{\varepsilon_\xi}, \quad (2.122)$$

значения которых фиксируются в таблице 2.19.

Для визуального анализа составляется приближенный график взаимозависимости мощности μ и ошибки ε реконфигурации для каждой степени приближения ξ , который будет иметь качественный вид, показанный на рисунке 2.26. Итоговое решение выбирается на основе оптимального сочетания «выигрыша» в мощности и «проигрыша» в точности реконфигурации с учетом заданных ограничений $\varepsilon_{\text{зад}}$ и $\mu_{\text{зад}}$. В большинстве случаев оно соответствует максимально допустимой мощности реконфигурации.

8. В результате анализа выбрать необходимую степень приближения с допустимыми мощностью и ошибкой и записать окончательное решение задачи реконфигурации в соответствии с (2.90) по формуле

$$H_{\hat{\xi}} = R_{1:n, 1:\rho-\hat{\xi}} \Sigma_{1:\rho-\hat{\xi}}^{-1} L_{1:\rho-\hat{\xi}, 1:n} \Delta B. \quad (2.123)$$

9. Все отказы, при компенсации которых привода системы управления выходят на ограничения с потерей устойчивости и управляемости, отнести к некомпенсируемым.

2.5.4. Пример решения задачи реконфигурации с учетом ограничений на управление

Для проверки работоспособности разработанной методики рассмотрим далее решение задачи реконфигурации ИП СУ самолета с учетом ограничений на управление. Анализ законов реконфигурации будем проводить на полной нелинейной модели самолета с учетом моделей приводов на этапе захода на посадку в автоматическом режиме для следующих сценариев:

- отказа привода стабилизатора
- отказа привода левого руля высоты
- одновременного отказа приводов руля направления, левого и правого элеронов.

После начала моделирования включается автопилот в режим захвата курсового и глассадного радиомаяка и автомат тяги в режиме стабилизации скорости. На 10-й секунде моделируется отказ приводов, и одновременно вычисляется матрица реконфигурации H , исходя из предположения, что время наступления отказа совпадает с временем его обнаружения.

Линеаризованная матрица эффективности управления самолета с работоспособной СУ в соответствии с таблицей 2.5 при $H = 451,1$ м, $V = 249,8$ км/ч имеет вид

$$B = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0052 & 0.0052 \\ 0.0178 & 0.0178 & -0.0084 & 0.0 & -0.0010 & -0.0010 & -0.0070 & -0.0070 \\ -0.0178 & -0.0178 & 0.0084 & 0.0 & 0.0010 & 0.0010 & 0.0070 & 0.0070 \\ -0.3378 & -0.3378 & -1.3433 & 0.0 & -0.0501 & -0.0501 & 0.1271 & 0.1271 \\ 0.0 & 0.0 & 0.0 & -0.0054 & 0.0 & 0.0 & -0.0022 & 0.0022 \\ 0.2052 & -0.2052 & 0.0 & -0.1973 & 0.5419 & -0.5419 & -1.6357 & 1.6357 \\ -0.0092 & 0.0092 & 0.0 & -3.1463 & -0.0579 & 0.0579 & 0.2973 & -0.2973 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}.$$

Вычислим норму матрицы эффективности управления ВС с работоспособной СУ и определим в соответствии с (2.117), (2.118) допустимые на практике значения мощности и ошибки реконфигурации

$$\mu_{\text{зад}} \leq \|B\| = 4,27, \quad (2.124)$$

$$\varepsilon_{\text{зад}} \leq \|B\| = 4,27. \quad (2.125)$$

Рассмотрим для примера отказ канала левого руля высоты, при котором матрица отказов имеет следующий диагональный вид

$$F = \text{diag} (0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1),$$

которому соответствует согласно (2.8) следующая матрица эффективности управления самолета с отказавшей СУ

$$B_f = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0052 & 0.0052 \\ 0.0 & 0.0178 & -0.0084 & 0.0 & -0.0010 & -0.0010 & -0.0070 & -0.0070 \\ 0.0 & -0.0178 & 0.0084 & 0.0 & 0.0010 & 0.0010 & 0.0070 & 0.0070 \\ 0.0 & -0.3378 & -1.3433 & 0.0 & -0.0501 & -0.0501 & 0.1271 & 0.1271 \\ 0.0 & 0.0 & 0.0 & -0.0054 & 0.0 & 0.0 & -0.0022 & 0.0022 \\ 0.0 & -0.2052 & 0.0 & -0.1973 & 0.5419 & -0.5419 & -1.6357 & 1.6357 \\ 0.0 & 0.0092 & 0.0 & -3.1463 & -0.0579 & 0.0579 & 0.2973 & -0.2973 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}.$$

Вычислим сингулярное разложение матрицы B_f (2.116), определим условие разрешимости задачи оптимальной реконфигурации (2.120) при $\xi = 0$

$$L_0 B = \begin{bmatrix} 1.43 \cdot 10^{-17} & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

и оценим ее норму $\varepsilon_0 = \|L_0 B\| = 1.43 \cdot 10^{-16}$. Так как норма условия разрешимости практически равна нулю (машинному нулю для чисел формата двойной точности), то можно сделать вывод о том, что задача оптимальной реконфигурации имеет решение, удовлетворяющее требованию допустимости (2.125).

Определим в соответствии с (2.123) при $\xi = 0$ матрицу оптимальной реконфигурации

$$H_0 = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.997 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.004 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ -0.007 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.364 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ -0.444 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.009 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ -0.009 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

и вычислим ее норму

$$\mu(0) = \|H(0)\| = 1,151.$$

Значение нормы находится в соответствии с (2.124) в допустимых пределах, следовательно эффективности работоспособных поверхностей управления достаточно для осуществления реконфигурации, полностью восстанавливающей эффективность управления путем компенсации отказа канала левого руля высоты.

Проанализируем далее в соответствии с (2.119)–(2.122) эффективность приближений решения задачи реконфигурации с учетом ограничений на управление (таблица 2.20, рисунок 2.27).

Табл. 2.20. Анализ эффективности приближения решений при отказе канала левого руля высоты

ξ	0	1	2	3	4	5
μ_ξ	1.151	1.012	0.875	0.257	0.081	0.009
ε_ξ	$1.43 \cdot 10^{-16}$	$5.85 \cdot 10^{-4}$	$3.297 \cdot 10^{-3}$	$2.611 \cdot 10^{-2}$	0.342	0.395
ρ_ξ^μ	–	1.138	1.156	3.401	3.165	8.781
ρ_ξ^ε	–	$5.62 \cdot 10^{-13}$	$1.77 \cdot 10^{-1}$	$1.27 \cdot 10^{-1}$	$7.63 \cdot 10^{-2}$	$8.662 \cdot 10^{-1}$

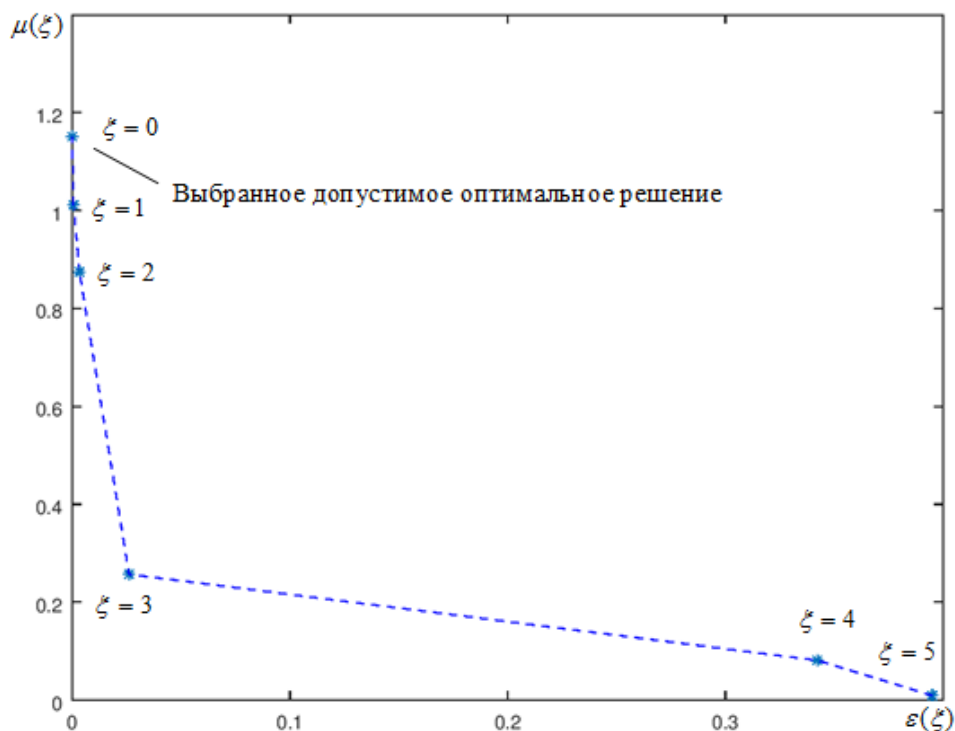


Рис. 2.27. Зависимость от степени приближения ξ мощности μ и ошибки ε реконфигурации при отказе канала левого руля высоты

Анализ показывает, что все решения данной задачи являются допустимыми. При этом ошибка реконфигурации при первом приближении ($\xi = 1$) уменьшается всего в 1,138 раза, однако это происходит за счет существенной потери точности решения задачи реконфигурации в 13 порядков. Следовательно, целесообразно выбрать оптимальное решение.

Графики переходных процессов при отклонении штурвала по тангажу приведены на рисунке 2.28.

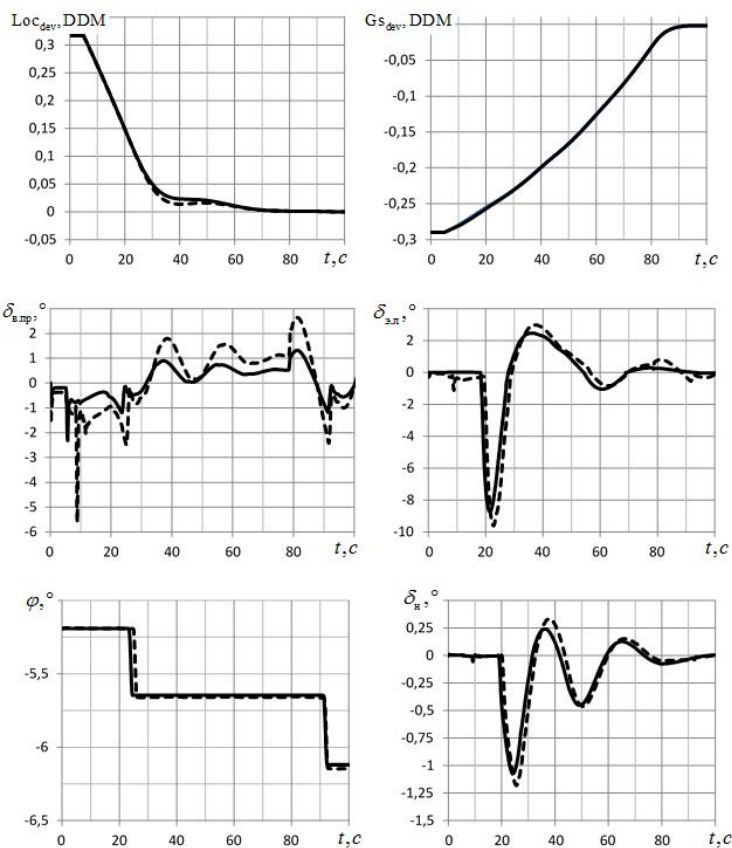
Результаты моделирования показывают, что автопилот успевает за счет дополнительного отклонения работоспособного руля высоты компенсировать данный отказ, и точности захвата курса и глиссады практически совпадают. Однако переходные процессы отклонения поверхностей управления с реконфигурацией являются более плавными и с меньшей амплитудой. Отказ канала левого руля высоты СУ компенсирует дополнительным отклонением в основном работоспособного правого руля высоты, правого и левого элерона. При этом все работоспособные поверхности управления на упоры не выходят, и штатное функционирование СУ не нарушается. Результаты с компенсацией отказа за счет реконфигурации и без отказа практически совпадают.

При отказе канала стабилизатора матрица отказов имеет следующий диагональный вид

$$F = \text{diag} (1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1),$$

которому соответствует согласно (2.8) следующая матрица эффективности управления самолета с отказавшей СУ

$$B_f = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0052 & 0.0052 \\ 0.0178 & 0.0178 & 0.0 & 0.0 & -0.0010 & -0.0010 & -0.0070 & -0.0070 \\ -0.0178 & -0.0178 & 0.0 & 0.0 & 0.0010 & 0.0010 & 0.0070 & 0.0070 \\ -0.3378 & -0.3378 & 0.0 & 0.0 & -0.0501 & -0.0501 & 0.1271 & 0.1271 \\ 0.0 & 0.0 & 0.0 & -0.0054 & 0.0 & 0.0 & -0.0022 & 0.0022 \\ 0.2052 & -0.2052 & 0.0 & -0.1973 & 0.5419 & -0.5419 & -1.6357 & 1.6357 \\ -0.0092 & 0.0092 & 0.0 & -3.1463 & -0.0579 & 0.0579 & 0.2973 & -0.2973 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}.$$



— без отказа - - - с отказом и реконфигурацией

Рис. 2.28. Результаты моделирования при отказе левого руля высоты

Вычислим сингулярное разложение матрицы B_f (2.116), определим условие разрешимости задачи оптимальной реконфигурации

$$L_0 B = \begin{bmatrix} 0.0 & 0.0 & -3.3 \cdot 10^{-17} & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix},$$

и оценим норму ошибки оптимального решения (2.120) ($\zeta = 0$):

$$\varepsilon = \|L_0 B\| = 3,3 \cdot 10^{-17}.$$

Норма условия разрешимости практически равна нулю (машинному нулю для чисел формата двойной точности), следовательно, задача оптимальной реконфигурации имеет решение, удовлетворяющее требованию допустимости (2.125).

Определим в соответствии с (2.123) при $\zeta = 0$ матрицу оптимальной реконфигурации

$$H_0 = \begin{bmatrix} 0.0 & 0.0 & 0.38 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.38 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 10.9 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 10.9 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \quad (2.126)$$

и вычислим ее норму

$$\mu_0 = \|H_0\| = 15,325.$$

Значение нормы в соответствии с (2.124) превышает допустимый предел. Эффективности элеронов недостаточно для осуществления реконфигурации, полностью восстанавливающей управляемость ВС путем компенсации отказа канала стабилизатора.

Проанализируем далее в соответствии с (2.119)–(2.122) эффективность приближений решения задачи реконфигурации с учетом ограничений на управление (таблица 2.21, рисунок 2.29).

Анализ показывает, что не все решения данной задачи являются допустимыми. Первым допустимым по мощности ($\mu = 2,591$) является решение, полученное с использованием 3-го приближения ($\varepsilon = 3$). При этом приближении обеспечивается также допустимая ошибка реконфигурации ($\varepsilon_3 = 0,109$).

Табл. 2.21. Анализ эффективности приближения решений при отказе канала стабилизатора

ξ	0	1	2	3	4	5
μ_{ξ}	15,325	15,325	12,681	2,591	$4,66 \cdot 10^{-6}$	$2,49 \cdot 10^{-7}$
ε_{ξ}	$3,3 \cdot 10^{-17}$	$1,15 \cdot 10^{-5}$	$5,64 \cdot 10^{-2}$	0,109	1,3433	1,3433
ρ_{ξ}^u	—	1	1,2	4,89	$5,57 \cdot 10^5$	18,7
ρ_{ξ}^{ε}	—	$2,87 \cdot 10^{-12}$	$2,04 \cdot 10^{-4}$	$5,17 \cdot 10^{-1}$	$8,136 \cdot 10^{-2}$	1

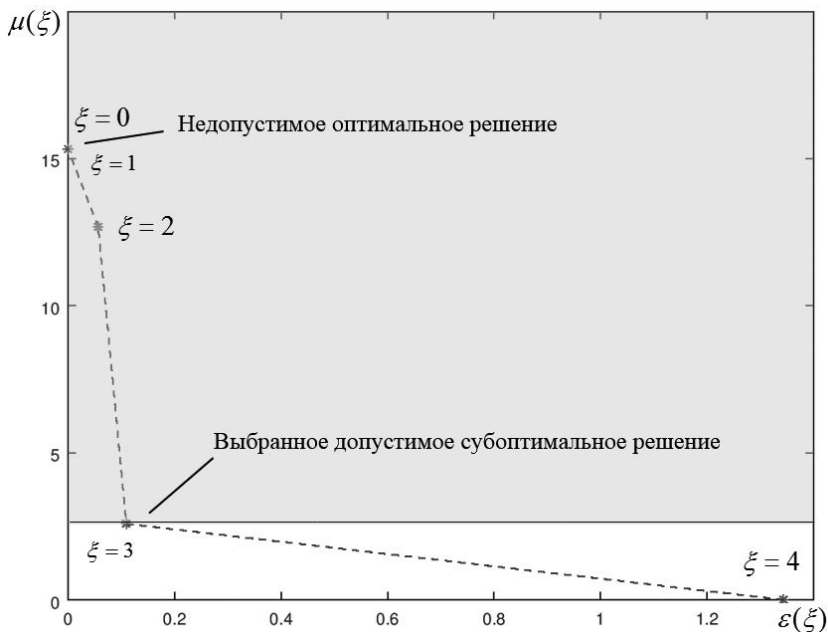


Рис. 2.29. График мощность (μ) – погрешность (ε) приближений решений задачи реконфигурации при отказе канала стабилизатора

Тогда окончательно вычислим по формуле (2.123) при $\xi = 3$ матрицу реконфигурации, описывающей закон компенсации отказа канала стабилизатора:

$$H_3 = \begin{bmatrix} 0.0 & 0.0 & 1.699 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.699 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.00001 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.25006 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.24961 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & -0.6394 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & -0.6396 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}. \quad (2.127)$$

Графики переходных процессов при отклонении штурвала по тангажу для вариантов моделирования без отказа и с отказом и реконфигурацией приведены на рисунке 2.30.

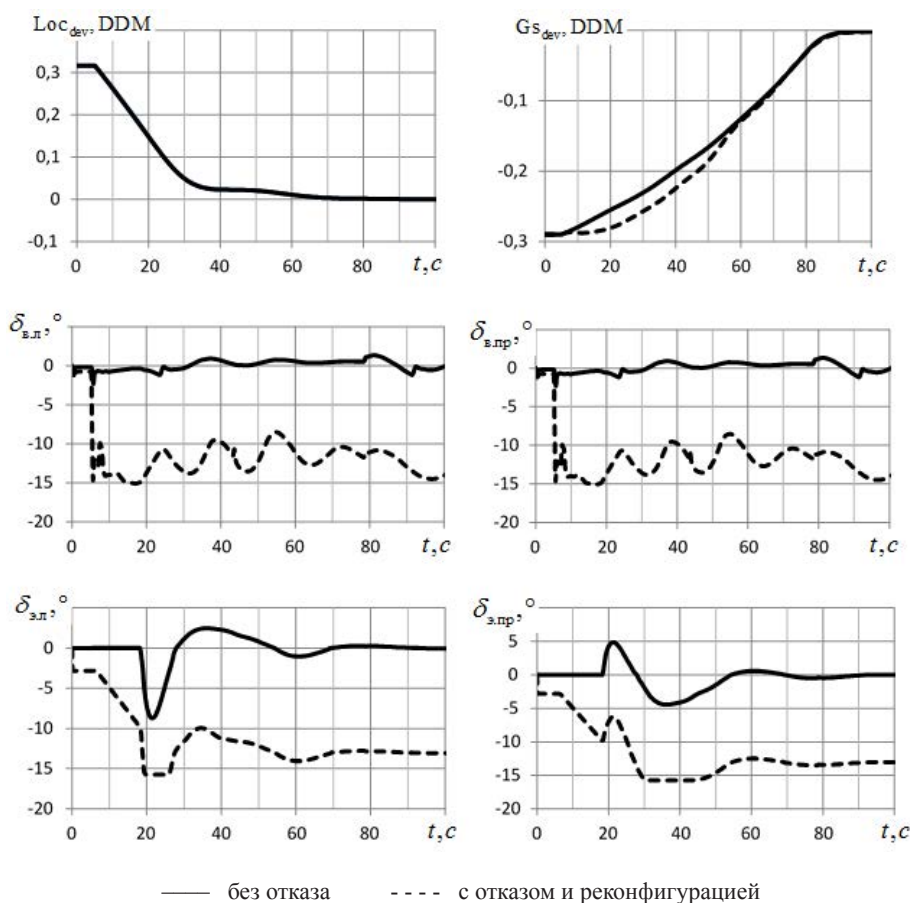


Рис. 2.30. Отказ стабилизатора

Результаты моделирования показывают, что отказ канала стабилизатора вследствие недостаточной эффективности элеронов СУ компенсирует дополнительным синфазным отклонением в основном руле высоты и интерцепторах. При этом все работоспособные поверхности управления на упоры не выходят, и штатное функционирование СУ не нарушается. Результаты с компенсацией отказа за счет реконфигурации и без отказа практически совпадают. Реконфигурация управления позволяет понизить амплитуду отклонения рулей высоты на 7%. При этом отклонения всех поверхностей управления находятся в допустимой области, и штатное функционирование СУ не нарушается. Результаты с компенсацией отказа за счет реконфигурации и без отказа практически совпадают.

Оптимальное решение (2.126) в теории обеспечивает практически полную компенсацию отказа стабилизатора $\varepsilon_0 = 3,3 \cdot 10^{-17}$, однако такая точность обеспечивается за счет трехкратного превышения ограничений на амплитуды отклонения элеронов из-за больших значений коэффициентов закона реконфигурации $h_{эл}^n \approx h_{эл}^n \approx 10.8$. На практике такое решение вследствие недостаточной эффективности поверхностей управления приводит к неустойчивости и неуправляемости самолета и еще больше усугубляет критическую ситуацию.

Закон реконфигурации, рассчитанный с учетом ограничений на управление (2.127), снижает чрезмерную нагрузку на элероны $h_{эл}^n \approx h_{эл}^n \approx 0.25$ и компенсирует отказ стабилизатора дополнительным синфазным отклонением рулей высоты и интерцепторов, которые имеют большую эффективность в канале тангажа. Отклонения элеронов уменьшаются по сравнению с оптимальным решением более чем в 40 раз.

При одновременных отказах каналов руля направления, левого и правого элеронов матрица отказов имеет следующий диагональный вид

$$F = \text{diag} (1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1),$$

которому соответствует согласно (2.8) следующая матрица эффективности управления самолета с отказавшей СУ

$$B_f = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0052 & 0.0052 \\ 0.0178 & 0.0178 & -0.0084 & 0.0 & 0.0 & 0.0 & -0.0070 & -0.0070 \\ -0.0178 & -0.0178 & 0.0084 & 0.0 & 0.0 & 0.0 & 0.0070 & 0.0070 \\ -0.3378 & -0.3378 & -1.3433 & 0.0 & 0.0 & 0.0 & 0.1271 & 0.1271 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -0.0022 & 0.0022 \\ 0.2052 & -0.2052 & 0.0 & 0.0 & 0.0 & 0.0 & -1.6357 & 1.6357 \\ -0.0092 & 0.0092 & 0.0 & 0.0 & 0.0 & 0.0 & 0.2973 & -0.2973 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

Вычислим сингулярное разложение матрицы B_f (2.116), определим условие разрешимости задачи оптимальной реконфигурации

$$L_0 B = \begin{bmatrix} 0.0 & 0.0 & 0.0 & -0.0354 & -0.000321 & 0.000322 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.00836 & 0.000076 & -0.000076 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

и оценим норму ошибки оптимального решения (2.120) ($\xi = 0$):

$$\varepsilon_0 = \|L_0 B\| = 0,0364.$$

Норма условия разрешимости удовлетворяет требованию допустимости (2.125), однако она не равна нулю, следовательно задача оптимальной реконфигурации не имеет точного решения.

Определим в соответствии с (2.123) при $\xi = 0$ матрицу оптимального компенсирующего управления

$$H_0 = \begin{bmatrix} 0.0 & 0.0 & 0.0 & -56.62262 & 0.70476 & -0.73873 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 56.62262 & -0.74017 & 0.70328 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.04620 & 0.04614 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -7.04305 & -0.07501 & 0.07523 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 7.04305 & 0.07501 & -0.07523 & 0.0 & 0.0 \end{bmatrix}$$

и вычислим ее норму

$$\mu_0 = \|H_0\| = 80,707.$$

Значение нормы в соответствии с (2.124) существенно превышает допустимый предел. Эффективности рулей высоты недостаточно для осуществления реконфигурации, полностью восстанавливающей управляемость ВС.

Проанализируем далее в соответствии с (2.119)–(2.122) эффективность приближений решения задачи реконфигурации с учетом ограничений на управление (таблица 2.22, рисунок 2.31).

Табл. 2.22. Анализ эффективности приближения решений при отказах каналов руля направления, левого и правого элеронов

ξ	0	1	2	3	4
μ_ξ	80,7	80,7	0,366	0,362	0,359
ε_ξ	0,0364	0,0364	3,132085	3,132086	3,132886
ρ_ξ^μ	–	1	220,4	1,01	1,0094
ρ_ξ^ε	–	0,999997	0,0116	1	0,9997

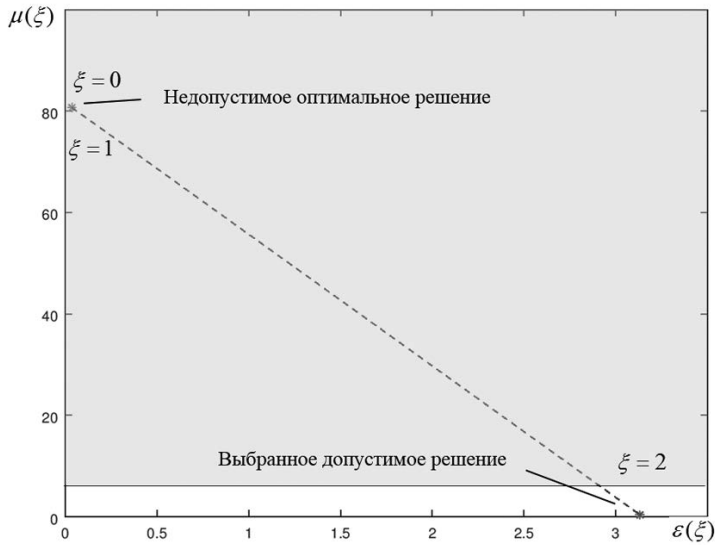


Рис. 2.31. График мощность (μ) – погрешность (ε) решений при отказах каналов руля направления, левого и правого элеронов

Анализ показывает, что не все решения данной задачи являются допустимыми. Первым допустимым по мощности ($\mu_{\xi} = 0,366$) является решение, полученное с использованием 2-го приближения ($\xi = 2$), как показано на рисунке 2.32.

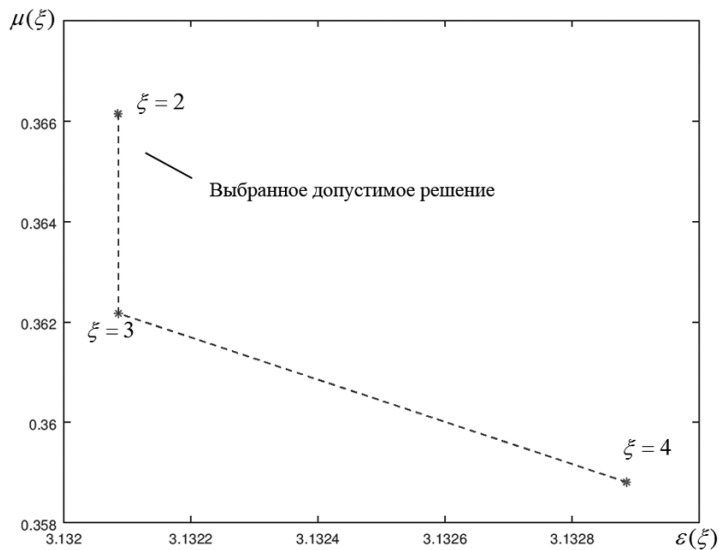


Рис. 2.32. График мощность (μ) – погрешность (ε) приближений решений задачи реконфигурации для 2–4 приближений при отказах каналов руля направления, левого и правого элеронов

При этом приближении обеспечивается также допустимая ошибка реконфигурации ($\varepsilon_2 = 3,132085$). Тогда окончательно вычислим по формуле (2.123) при $\xi = 2$ матрицу реконфигурации, описывающей закон компенсации отказа каналов руля направления, левого и правого элеронов

$$H_2 = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.01314 & 0.00461 & -0.03486 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & -0.01314 & -0.03484 & 0.00460 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 & 0.04611 & 0.04605 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -0.10733 & -0.15465 & 0.16753 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.10733 & 0.16749 & -0.15468 & 0.0 & 1.0 \end{bmatrix}$$

и определим ошибку реконфигурации

$$B - B_f H_2 = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 & -0.00007 & -0.00007 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -0.00587 & -0.00007 & 0.00007 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -0.55383 & 0.00687 & 0.00687 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -3.08224 & 0.03824 & -0.03816 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}.$$

Анализ показывает, что максимальная ошибка возникает в канале ω_y (-3.08224), и она сопоставима с соответствующим значением элемента в матрице эффективности управления ВС с работоспособной СУ (-3.14630). Результаты моделирования приведены на рисунке 2.33.

Реконфигурация приводит к существенному ухудшению эффективности управления по ω_y , однако управляемость по курсу сохраняется – остается возможность управления, в том числе за счет согласованного отклонения рулей высоты и интерцепторов. При этом, так как на этапе автоматического захода на посадку сигналы на отклонения поверхностей управления существенно ограничены, а руль направления практически не используется, то реконфигурация позволяет практически полностью компенсировать последствия таких критических отказов. Без использования методики учета ограничений самолет становится неустойчивым и неуправляемым сразу после включения реконфигурации. До настоящего момента такие отказы считались некомпенсируемыми.

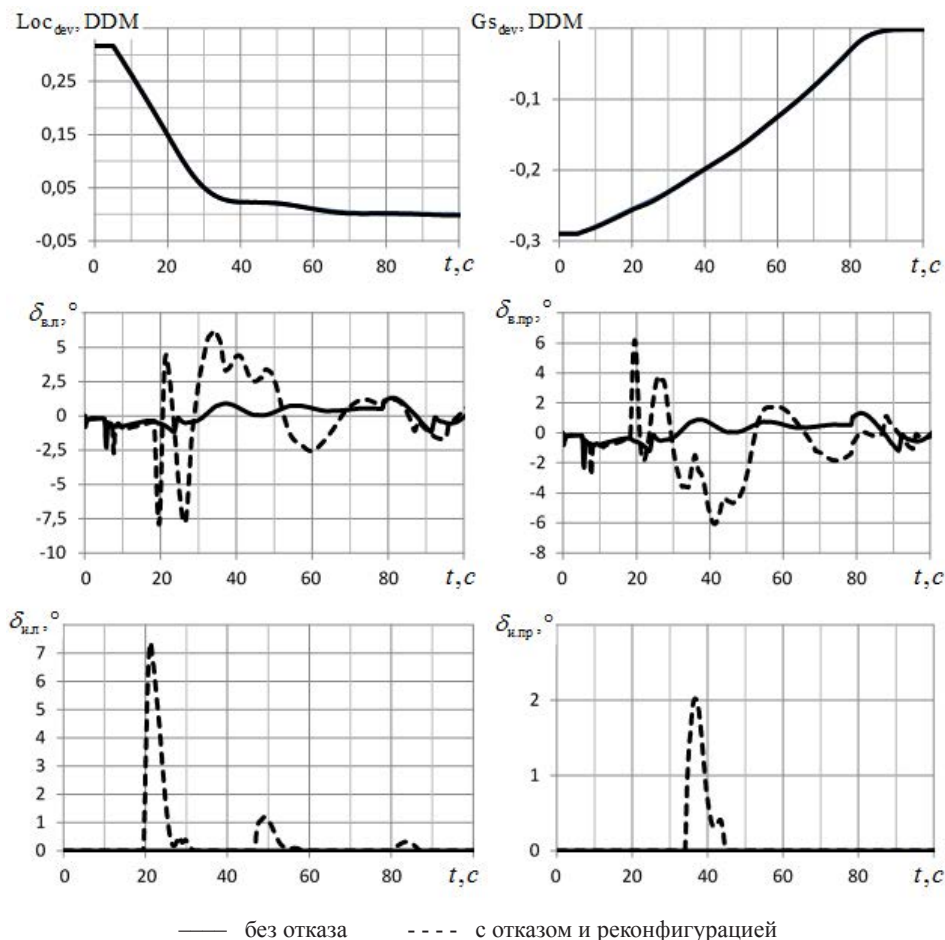


Рис. 2.33. Отказ руля направления, левого и правого элерона

2.5.5. Резюме

В результате проведенных исследований показано, что физические ограничения на амплитуды отклонения поверхностей управления ВС при реконфигурации ИП СУ можно учитывать за счет согласованного управления мощностью (нормой матрицы) реконфигурации, используемой для перераспределения управления, и точностью решения задачи.

Предлагается понижать точность и мощность решения задачи реконфигурации дискретно в несколько шагов. При этом на каждом шаге приближения обеспечивается субоптимальность решения с заранее известными минимально возможными мощностью и ошибкой для заданной степени приближения. Снижение мощности реконфигурации уменьшает максимальные отклонения поверхностей управления, обеспечивая их работу в допустимой области. Платой за это является повышение ошибки решения задачи реконфигурации.

На модельных примерах реконфигурации ИП СУ самолета показано, что оптимальное решение задачи реконфигурации может приводить к значительному выходу поверхностей управления за физические ограничения и потере управляемости, а применение субоптимального решения, рассчитанного с учетом ограничений на управление, снижает в несколько раз отклонения поверхностей управления и обеспечивает эффективное решение задачи в допустимой области мощности и ошибки реконфигурации.

Развитие подхода предполагает точное определение допустимых областей решений, непрерывное управление мощностью и ошибкой реконфигурации, распространение всех полученных результатов на задачи мультимодельной робастной реконфигурации.

2.6. Обобщенный подход к реконфигурации исполнительный подсистемы системы управления

Итак, обобщая изложенный материал, отметим, что наиболее предпочтительным подходом является реконфигурация на основе канонизации. При выполнении условий разрешимости используется аналитический алгоритм определения всех решений задачи реконфигурации управления при отказах ИП, полученный на основе канонического подхода к решению матричных уравнений.

Однако не все случаи отказов ИП СУ удастся парировать с использованием этого подхода. При невыполнении условия разрешимости задачи полученное аналитическое множество решений не может гарантировать обеспечения приемлемого качества управления. Исходя из этого, область применения канонического подхода была расширена на решение более широкого круга задач реконфигурации.

При невыполнении условия разрешимости задачи реконфигурации ИП СУ для поиска численных решений, оптимальных в смысле минимума нормы ошибки, составляющие канонического разложения предлагается определять с помощью сингулярного разложения матрицы эффективности управления. Показано, что в частном случае предлагаемый алгоритм формирует псевдообратное решение.

Для получения аналитического решения оптимальной реконфигурации ИП СУ предлагается алгоритм определения псевдообратной матрицы эффективности управления, основанный на скелетном разложении, являющимся частным случаем канонического. Приведена методика получения субоптимального решения задачи реконфигурации с учетом ограничений на амплитуды отклонения поверхностей управления ВС. Гарантируется, что задача реконфигурации ИП СУ, основанная на полученных соотношениях, будет решена с априорно заданной точностью при выполнении соответствующих приближенных условий разрешимости.

Полученный алгоритм оптимальной реконфигурации лег в основу алгоритма мультимодельной робастной реконфигурации. В такой постановке обеспечивается минимальность ошибки реконфигурации ИП СУ для нескольких априорно заданных режимов полета или наборов отказов.

В таблице 2.23 приведена логика работы гибридной отказобезопасной СУ ВС, предусматривающая применение всех рассмотренных методов реконфигурации в зависимости от точности (ошибок) решения задач идентификации модели ВС, диагностирования отказов ИП СУ и собственно реконфигурации управления.

Пассивные режимы реконфигурации основаны на построении алгоритма, робастного к отказам ИП СУ. Такие режимы реконфигурации определяются требованиями по обеспечению отказобезопасности СУ при невозможности получения достоверной информации о возникающих отказах. Активные режимы способны идентифицировать отказы, и так же, как и пассивные, различаются степенью робастности законов реконфигурации управления, определяемой оперативной доступностью достоверной информации о параметрах модели ВС и отказах.

Табл. 2.23. Логика работы отказобезопасной системы управления

Режим / характеристика	Пассивный		Активный	
	к отказам и режимам	к отказам	к режимам	не обязательна
Робастность				
Точность	мин	оптимальная	оптимальная	макс
Ошибка диагностирования	$\varepsilon_{\text{диаг}} \geq \varepsilon_{\text{диаг}}^{\text{зад}}$	$\varepsilon_{\text{диаг}} \geq \varepsilon_{\text{диаг}}^{\text{зад}}$	$\varepsilon_{\text{диаг}} < \varepsilon_{\text{диаг}}^{\text{зад}}$	$\varepsilon_{\text{диаг}} < \varepsilon_{\text{диаг}}^{\text{зад}}$
Ошибка идентификации	$\varepsilon_{\text{идент}} \geq \varepsilon_{\text{идент}}^{\text{зад}}$	$\varepsilon_{\text{идент}} < \varepsilon_{\text{идент}}^{\text{зад}}$	$\varepsilon_{\text{идент}} \geq \varepsilon_{\text{идент}}^{\text{зад}}$	$\varepsilon_{\text{идент}} < \varepsilon_{\text{идент}}^{\text{зад}}$
Ошибка реконфигурации	$\varepsilon_{\text{рек}} \geq \varepsilon_{\text{рек}}^{\text{зад}}$	$\varepsilon_{\text{рек}} < \varepsilon_{\text{рек}}^{\text{зад}}$		

Режим с максимальной точностью предполагает оперативный расчет оптимальных коэффициентов законов реконфигурации по заранее определенным алгоритмам при любых изменениях в системе, в том числе и при возникновении отказов. Данный режим включается при доступности достоверных данных как о параметрах модели ВС, так и об отказах ИП СУ.

Режим реконфигурации, робастный к изменению режимов полета ВС, включается при отсутствии достоверных данных о параметрах модели ВС и наличии достоверной информации о параметрах отказов. Для построения таких решений должна быть определена область режимов полетов, для которых необходимо обеспечить робастность.

Режим реконфигурации, робастный к отказам приводов, включается при доступности достоверных данных о параметрах модели ВС и отсутствии достоверных данных об отказах ИП ВС. В данном режиме должна быть определена область отказов, для которых необходимо обеспечить робастность.

Режим реконфигурации, робастный как к изменению режимов полета ВС, так и к возникающим отказам, включается при отсутствии достоверных данных о параметрах модели ВС и отказах ИП СУ, а также при невозможности решения задачи реконфигурации управления с заданной точностью. В зависимости от ситуации предусматривается принудительное включение любого из режимов реконфигурации.

В соответствии с приведенной логикой сформируем обобщенный алгоритм реконфигурации ИП СУ ВС, блок-схема которого приведена на рисунке 2.34.

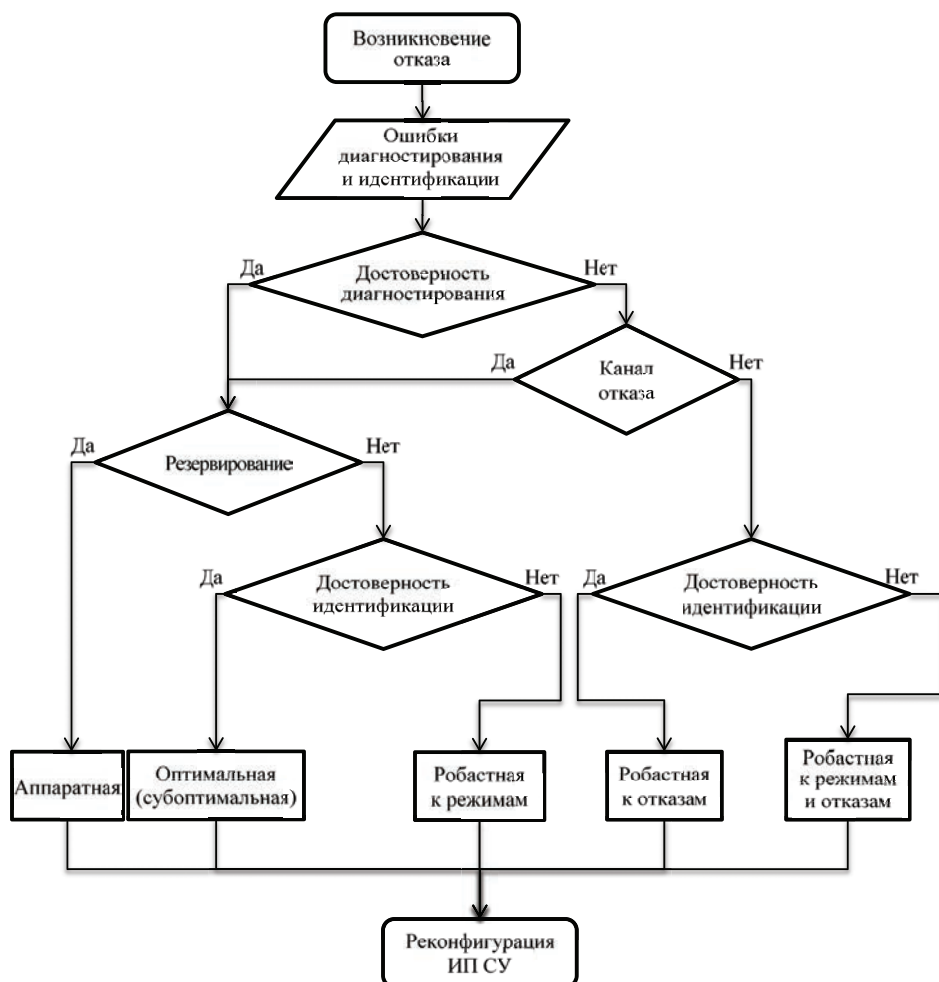


Рис. 2.34. Блок-схема алгоритма реконфигурации ИП СУ ВС

Он включает в себя следующие этапы.

1. Проверка признака достоверности диагностирования отказов. В случае полностью или частично работоспособной системы диагностирования

система реконфигурации переходит в активный режим, и осуществляется проверка возможности использования аппаратного резервирования. При полном отсутствии информации об отказах СУ переходит в режим функциональной реконфигурации и проверяется работоспособность системы идентификации параметров модели ВС.

2. Проверка признака достоверности идентификации параметров модели ВС. В зависимости от условий, приведенных в таблице 2.23, СУ переходит к синтезу соответствующего алгоритма реконфигурации.

3. Синтез алгоритма реконфигурации ИП СУ. При решении задачи синтеза алгоритма во всех случаях осуществляется проверка условия реконфигурируемости управления и анализ возможного множества решений.

В результате совместного применения разработанных методов реконфигурации повышается отказоустойчивость ИП СУ ВС за счет согласованного переключения активных и пассивных режимов реконфигурации в зависимости от требуемой степени точности и робастности решений, определяемой условиями разрешимости задачи реконфигурации и достоверности информации о параметрах модели ВС в работоспособном и неработоспособном состояниях ИП СУ.

Глава 3

РЕКОНФИГУРАЦИЯ ВЫЧИСЛИТЕЛЬНЫХ ПОДСИСТЕМ СИСТЕМ УПРАВЛЕНИЯ ВОЗДУШНЫХ СУДОВ

3.1. Принципы организации вычислительного процесса в системах управления воздушных судов

Прежде чем приступить к разработке методов реконфигурации ВП СУ ВС, рассмотрим общие принципы организации вычислительного процесса с использованием ИМА.

Заметим, что ВП СУ, построенную на базе ИМА, пример которой приведен на рисунке 3.1, формально можно отнести к классу сетевых информационно-управляющих систем [20, 49]. В общем виде будем считать, что в состав ВП СУ входит N одинаковых ПУ, каждый из которых подключен к коммуникационной сети (КС). При этом, положим, что каждый из ПУ имеет производительность $S_{\text{ПУ}}$ (операций в секунду), вероятность безотказной работы $P_{\text{ПУ}}$, а КС имеет пропускную способность Y .



Рис. 3.1. ВП СУ с распределенной сетевой организацией

Допустим, что ВП предназначена для решения некоторой задачи управления (в случае ВП СУ ВС под задачей управления понимается множество программ, необходимых для реализации заданных функций ЭДСУ, причем каждую отдельную программу можно рассматривать как подзадачу), которая в общем виде может быть представлена в виде некоторого графа $G(Q, X)$, как это показано на рисунке 3.2. При этом множество Q вершин графа $G(Q, X)$ содержит три подмножества вершин различных типов:

- подмножество $Q_{\text{вх}} \in Q$ входных вершин, соответствующих источникам данных, т.е. сенсорным устройствам (вершинам, принадлежащим множеству $Q_{\text{вх}}$, инцидентны только исходящие дуги);
- подмножество $Q_{\text{вых}} \in Q$ выходных вершин, соответствующих приемникам информации, т.е. исполнительным устройствам (вершинам множества $Q_{\text{вых}}$ инцидентны только входящие дуги);
- подмножество операционных вершин $Q_{\text{оп}} \in Q$, определяющее ее вычислительные операции (подзадачи), которые должны быть выполнены при ре-

ализации алгоритма управления. Вершинам множества $Q_{оп}$ одновременно инцидентны как входящие, так и исходящие дуги.

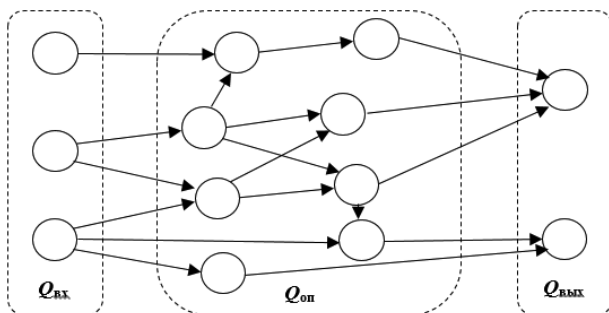


Рис.3.2. Представление задачи в виде графа

Будем считать, что каждой вершине $q_i \in Q_{оп}$ графа $G(Q, X)$ соответствует некоторая подзадача, вычислительная сложность которой определяется величиной D_i (т.е. количеством элементарных вычислительных операций, необходимых для решения этой подзадачи на ПУ).

Дуги множества X графа $G(Q, X)$ определяют пересылки данных между подзадачами, соответствующими вершинам множества Q . При этом если две вершины q_i и q_j графа $G(Q, X)$ соединены дугой $x(q_i, q_j)$, то это означает, что выходные данные, полученные в результате реализации вершины (подзадачи) q_i , являются входными данными, необходимыми для реализации подзадачи q_j . Будем считать, что каждой дуге $x(q_i, q_j) \in X$ приписана величина W_{ij} , определяющая количество (объем) данных, передаваемых от вершины (подзадачи) q_i к вершине (подзадаче) q_j .

Для дальнейших построений граф $G(Q, X)$ удобно представить в ярусно-параллельной форме, показанной на рисунке 3.3.

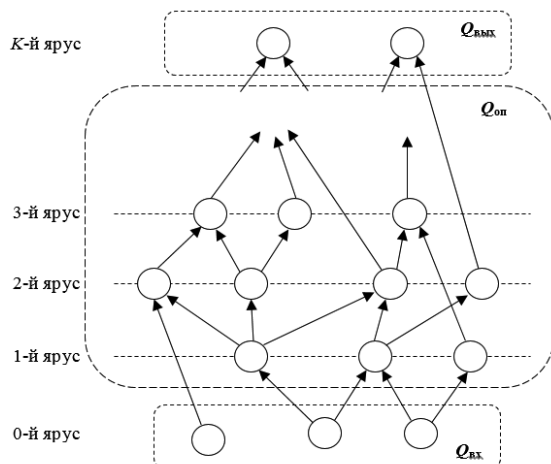


Рис. 3.3. Ярусно-параллельная форма графа-задачи $G(Q, X)$

При таком представлении графа $G(Q, X)$ на каждом ярусе располагаются информационно-независимые вершины, т.е. вершины, соответствующие подзадачам, которые могут решаться параллельно (одновременно) на различных ПУ. На нулевом ярусе располагаются входные вершины, принадлежащие множеству $Q_{\text{вх}}$, на последнем K -м ярусе – выходные вершины, принадлежащие множеству $Q_{\text{вых}}$, а на j -м ярусе ($j = 1, \dots, K - 1$) вершины множества $Q_{\text{оп}}$, удаленные от вершин нулевого уровня не более чем j дугами. В этом случае каждой подзадаче можно назначить максимально допустимые времена ее выполнения и передачи результатов, в сумме составляющие время ее решения

$$T_{\text{реш } i} = T_{\text{В } i} + T_{\text{ПД } i},$$

где $T_{\text{В } i}$ – время решения i -й подзадачи;

$T_{\text{ПД } i}$ – время передачи данных i -й подзадачи.

Эффективность работы ВП СУ при решении сформулированной выше задачи управления в значительной степени будет зависеть от организации вычислительного процесса в ней. Рассмотрим различные способы организации вычислительного процесса в ВП СУ ВС более подробно.

В простейшем случае решение всей задачи $G(Q, X)$ можно поручить одному ПУ, входящему в состав ВП СУ, в то время как остальные $(N-1)$ ПУ будут находиться в резерве. При этом время решения задачи $G(Q, X)$ будет составлять

$$T_{\text{реш}} = \frac{\sum_{j=1}^{M_{\text{оп}}} D_j}{S_{\text{ПУ}}}, \quad (3.1)$$

где $M_{\text{оп}} = |Q_{\text{оп}}|$ – число операционных вершин в графе $G(Q, X)$;

D_j – вычислительная сложность j -й операционной вершины.

Быстродействие ВП СУ можно существенно повысить, если каким-то образом распараллелить процедуру решения задачи, представленной графом $G(Q, X)$, между всеми N процессорными узлами. Для этого граф $G(Q, X)$ должен быть каким-то образом предварительно разрезан на N подграфов, решение каждого из которых может осуществляться с помощью одного из ПУ. В зависимости от того, как будет осуществлено это разрезание, будут зависеть и способы организации вычислительного процесса в ВП СУ. Рассмотрим этот вопрос подробнее.

Разрезание графа $G(Q, X)$ на подграфы можно осуществлять различными способами, однако все эти способы сводятся к двум основным: способу продольного разрезания и способу поперечного разрезания.

Способ продольного разрезания предполагает разрезание графа $G(Q, X)$ на непересекающиеся подграфы $G_i(Q, X_i)$ ($i = 1, N$) таким образом, чтобы в подграф $G_i(Q, X_i)$ входили вершины, принадлежащие различным ярусам графа (рисунок 3.4).

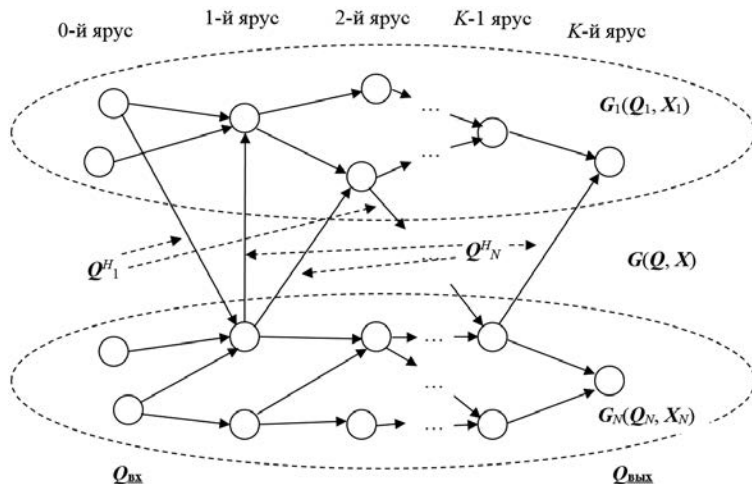


Рис. 3.4. Способ продольного разрезания графа $G(Q, X)$

Каждый из полученных таким образом подграфов $G(Q, X)$ закрепляется за одним из ПУ, а их решение может осуществляться параллельно (одновременно) на всех ПУ, входящих в состав ВП СУ. Такой способ организации вычислительного процесса принято называть параллельным [10]. Временная диаграмма решения задачи $G(Q, X)$ при параллельном способе организации вычислительного процесса представлена на рисунке 3.5.

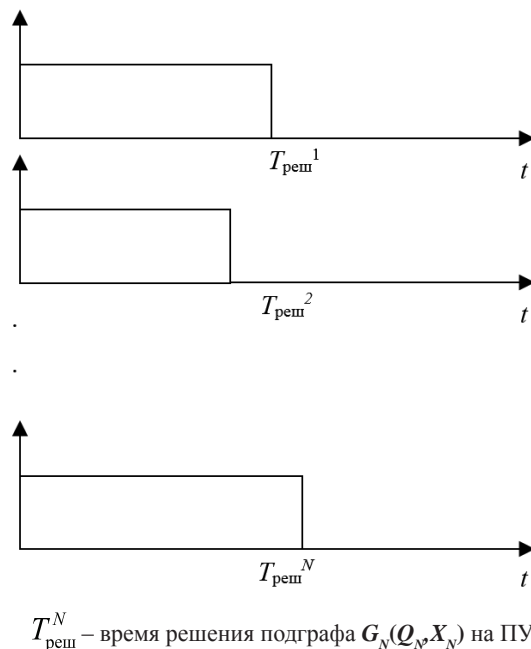


Рис. 3.5. Временная диаграмма параллельного решения задачи управления

При этом общее время решения задачи управления будет составлять

$$T_{\text{реш}} = \max_{i=1, N} T_{\text{реш}}^i \quad (3.2)$$

где $T_{\text{реш}}^i = T_{\text{В}}^i + T_{\text{п}}^i$,

$T_{\text{В}}^i = \sum_{j=1}^{M_i} \frac{D_j^i}{S_{\text{ПВ}}}$ – время решения всех подзадач, приписанных вершинам гра-

фа $G(Q, X)$, $M_i = |Q_i|$;

$T_{\text{п}}^i = \sum_{j=1}^{R_i} \frac{W_j^i}{Y_{\text{КС}}}$ – время передачи данных, соответствующих дугам, исходя-

щим из графа $G_i(Q, X)$ (т.е. время передачи межпроцессорных данных, формируемых в процессе решения задачи $G_i(Q, X_i)$);

$R_i = |Q_i^n|$, где Q_i^n – множество дуг, исходящих из графа $G(Q, X)$ (см. рисунок 3.4).

Из выражения (3.2) следует, что в целях минимизации времени $T_{\text{реш}}$ граф $G(Q, X)$ следует разрезать на подграфы таким образом, чтобы

$$T_{\text{реш}}^i \approx T_{\text{реш}}^j \quad (i = \overline{1, N}; j = \overline{1, N}).$$

Необходимо отметить, что при продольном разрезании графа $G(Q, X)$ на подграфы возникает проблема синхронизации передачи данных между отдельными ПУ. Действительно, если два подграфа $G_i(Q, X_i)$ и $G_l(Q, X_l)$ соединены дугой $x(q_j^i, q_f^l)$, то это означает, что данные, полученные в результате решения подзадачи $q_j^i \subseteq Q_i$, используются при решении подзадачи $q_f^l \subseteq Q_l$. Поэтому ПУ_{*l*}, реализующий подграф $G_l(Q, X_l)$, должен передать эти данные ПУ_{*i*}, реализующему подграф $G_i(Q, X_i)$. Поскольку процесс решения подграфов $G_i(Q, X_i)$ на ПУ_{*i*} и $G_l(Q, X_l)$ на ПУ_{*l*} происходит асинхронно, то возникает проблема синхронизации их работы при передаче и приеме межпроцессорных данных. При этом могут появляться дополнительные временные задержки, когда, например, ПУ_{*l*} уже готов решать подзадачу q_f^l , а ПУ_{*i*} еще не решил подзадачу q_j^i и не готов передать результат ее решения на ПУ_{*l*}.

При большом числе N проблема межпроцессорной синхронизации при передаче данных может оказаться достаточно сложной и приводить к существенному увеличению общего времени решения $G(Q, X)$.

Второй способ – это способ поперечного разрезания графа $G(Q, X)$ на такие непересекающиеся подграфы $G_i(Q, X_i)$ ($i = \overline{1, N}$), каждый из которых включает один или несколько последовательных ярусов графа $G(Q, X)$ (рисунок 3.6).

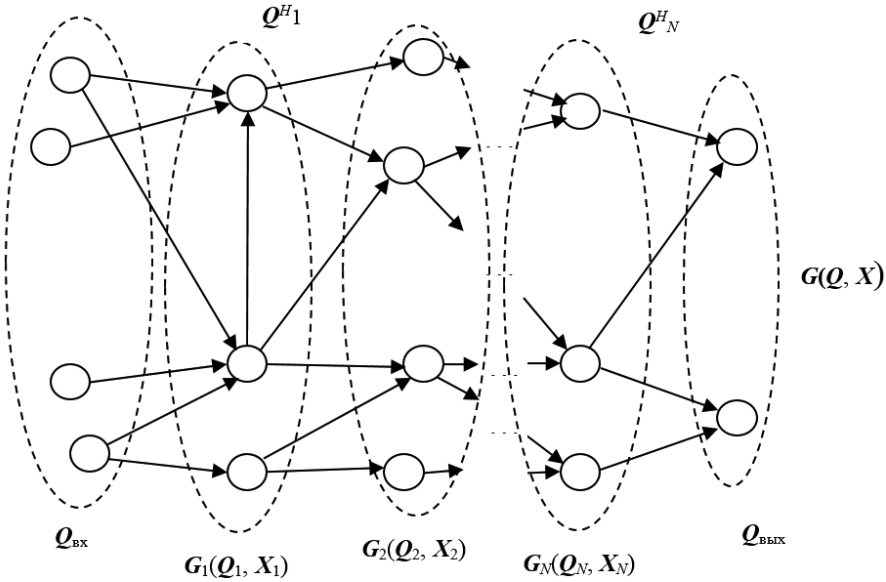


Рис. 3.6. Способ поперечного разрезания графа $G(Q, X)$

Для решения задачи $G(Q, X)$ при таком способе разрезания все N ПУ должны быть соединены в последовательную цепочку (рисунок 3.7), причем ПУ _{i} поручается реализация подграфа $G_i(Q_i, X_i)$ ($i = 1, N$).

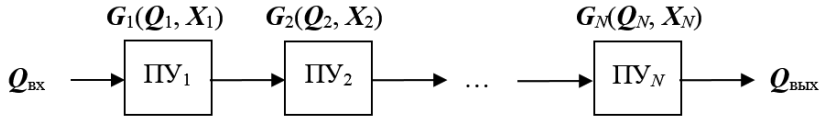


Рис. 3.7. Конвейерный способ организации вычислительного процесса

Если на вход такой цепочки процессоров подать исходные данные для решения задачи $G(Q, X)$, то через время

$$T_{\text{реш}} = \sum_{i=1}^N (T_{\text{в}}^i + T_{\text{п}}^i)$$

где $T_{\text{в}}^i = \sum_{j=1}^{M_i} \frac{D_j^i}{S_{\text{пу}}}$ – время решения вычислительных подзадач $G_i(Q_i, X_i)$, $M_i = |Q_i|$;

$T_{\text{п}}^i = \sum_{j=1}^{R_i} \frac{W_j^i}{Y_{\text{кв}}}$ – время передачи данных, соответствующих исходящим дугам подграфа $G_i(Q_i, X_i)$, $R_i = |Q_i^H|$ (см. рисунок 3.6), на выходе цепочки будет сформировано ее решение.

Время $T_{\text{реш}}$ можно сократить приблизительно в N раз (т.е. повысить быстродействие ВП СУ в N раз), если организовать в такой цепочке процессоров конвейерную обработку, суть которой заключается в следующем.

После того как ПУ₁ решил задачу, представленную подграфом $G_1(Q_1, X_1)$, и передал результирующие данные на ПУ₂, он может приступить к обработке следующего множества входных данных, выдаваемых информационной подсистемой. Аналогично, ПУ₂ по завершении процедуры обработки очередных данных, поступивших от ПУ₁, по алгоритму $G_2(Q_2, X_2)$ и после передачи результирующих данных на ПУ₃ может приступить к обработке следующего множества данных, полученных ПУ₁ в результате обработки следующих входных данных. Аналогичным образом должны работать все ПУ, входящие в состав ВП СУ. В результате такой организации процесса вычислений первый вектор результирующих данных (команд управления) появится на выходе цепочки из N ПУ, как и в рассмотренном выше случае, через время

$$T_{\text{реш}} = \sum_{i=1}^N T_{\text{реш}}^i = \sum_{i=1}^N (T_{\text{В}}^i + T_{\text{П}}^i),$$

где $T_{\text{реш}}^i$ – время решения подграфа $G_i(Q_i, X_i)$ ($i = \overline{1, N}$) от момента начала работы ВП СУ. Зато все последующие команды управления будут уже появляться на выходе такой цепочки ПУ через время

$$T_{\text{реш}} = \max_{i=1, N} T_{\text{реш}}^i = T_{\text{реш}}^M,$$

т.е. через время, определяемое временем решения наиболее трудоемкого подграфа $G_i(Q_i, X_i)$ [10].

На рисунке 3.8 показана временная диаграмма работы ВП СУ при конвейерном способе организации вычислений.

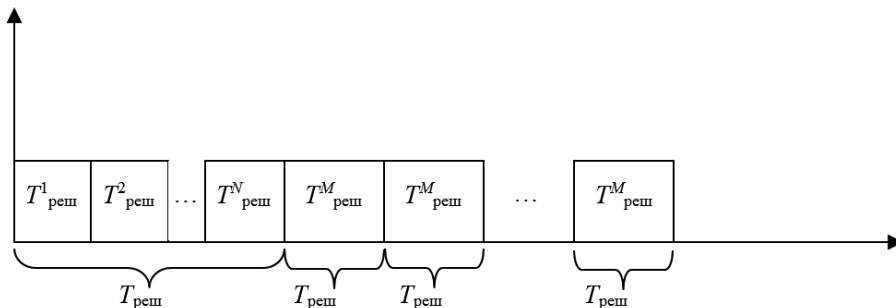


Рис. 3.8. Диаграмма работы ВП СУ при конвейерном способе организации вычислений

Очевидно, что для уменьшения времени $T_{\text{реш}}$ необходимо разрезать граф $G(Q, X)$ таким образом, чтобы

$$T_{\text{реш}}^i \approx T_{\text{реш}}^j, \quad (i=1, N; j=1, N),$$

т.е. время решения всех подграфов $G_i(Q_i, X_i)$ было приблизительно одинаково.

К преимуществам конвейерного способа организации вычислений в ВП СУ ВС следует отнести отсутствие проблем с синхронизацией передачи данных между отдельными ПУ. Действительно, пакет данных, получаемых в результате решения подграфа $G_i(Q_i, X_i)$ на ПУ_{*i*}, должен доставляться на ПУ_{*i+1*} к моменту начала следующего шага решения, т.е. через время $T_{\text{реш}}^M$. Поэтому обмен данными между всеми ПУ может в данном случае осуществляться строго синхронно в заданные моменты времени с частотой $1/T_{\text{реш}}^M$. Это обстоятельство, безусловно, упрощает организацию вычислительного процесса в ВП СУ.

Поскольку среди представленных способов организации вычислительного процесса невозможно однозначно выделить лучший, при проектировании конкретной ВП СУ следует принимать во внимание свойства графа задачи управления, ПУ и коммуникационной среды. Например, при недостаточной пропускной способности коммуникационной среды предпочтительным будет параллельный способ, т.к. позволяет уменьшить количество передаваемых данных между ПУ. Но параллельный способ сложнее применить при низкой производительности ПУ, т.к. при таком способе сложнее равномерно разместить вычислительную нагрузку, кроме того, необходимо, чтобы ПУ был способен выполнить цепочку подзадач от начальной до конечной вершины графа задачи управления за время, меньшее максимально допустимого.

3.2. Способы повышения надежности вычислительных подсистем

Одним из важнейших требований, предъявляемых к ВП СУ ВС, является их высокая надежность и отказоустойчивость. Как уже указано выше, архитектура ИМА открывает дополнительные возможности повышения надежности ВП СУ ВС за счет реконфигурации при возникновении отказов. Рассмотрим в общем виде способы повышения надежности ВП с сетевой архитектурой, к которым относится архитектура ИМА.

3.2.1. Структурное резервирование

В качестве характеристик надежности технических систем, как правило, используют такие параметры, как вероятность безотказной работы P и гамма-процентную наработку на отказ t_γ . С учетом того, что эти параметры взаимозависимы, поскольку t_γ представляет собой время от начала работы системы до момента, когда P станет меньше минимально допустимого значения, для оценки надежности технической системы достаточно получить зависимость для расчета P , на основе которой можно определить и t_γ .

В общем случае, если в составе ВП СУ имеются H ПУ, которые задействованы в решении задачи $G(Q, X)$, то вероятность безотказной работы ВП будет равна

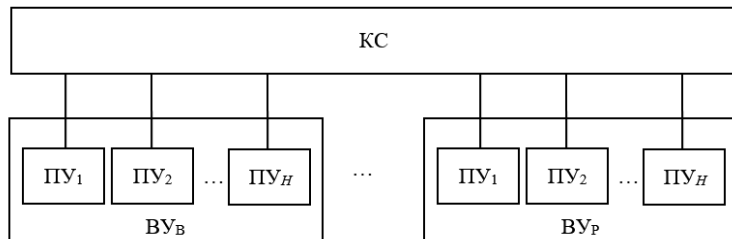
$$P_B = \prod_{i=1}^H P_{\text{ПУ}_i},$$

где $P_{\text{ПУ}}$ – вероятность безотказной работы отдельного ПУ.

Наиболее распространенным способом повышения надежности технических систем является способ структурного резервирования [45], суть которого заключается в подключении к основной аппаратуре системы дополнительной резервной таким образом, чтобы при отказе основной аппаратуры резервная продолжала выполнять ее функции.

Очевидно, что вероятность безотказной работы ВП СУ можно повысить за счет введения в ее состав дополнительных резервных ПУ, непосредственно не участвующих в решении задачи управления, но которые могут заменить в случае необходимости вышедшие из строя (отказавшие) основные ПУ, задействованные в решении задачи управления $G(Q, X)$. При этом можно предложить несколько подходов.

Первый подход заключается в использовании способа общего структурного резервирования, когда резервируется все вычислительное устройство ВП СУ (рисунок 3.9).



$\text{ВУ}_в$ – основное вычислительное устройство, задействованное в решении задачи управления;
 $\text{ВУ}_р$ – резервное вычислительное устройство

Рис. 3.9. Общее резервирование вычислительных устройств ВП СУ

Вероятность безотказной работы такой системы будет составлять

$$P_B = 1 - (1 - P_{\text{ПУ}}^H)^{B+1},$$

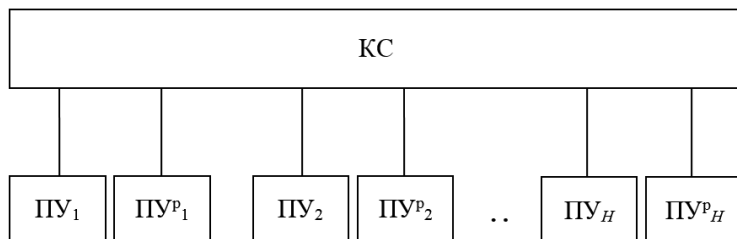
где B – число резервных вычислительных устройств.

При этом общее число ПУ, входящих в состав ВП СУ, будет равно

$$N = H + B \cdot H = H(B + 1).$$

Очевидно, что данный способ резервирования требует больших дополнительных аппаратных затрат и потому является малоэффективным.

Второй подход заключается в поэлементном резервировании каждого отдельного ПУ, входящего в состав ВП СУ (рисунок 3.10).



$ПУ_i (i = 1, H)$ – основные ПУ
 $ПУ_j^p$ – резервные ПУ

Рис. 3.10. Резервирование отдельных ПУ

Общее число ПУ в такой системе будет таким же, как и в предыдущем случае, т.е.

$$N = H(B + 1),$$

где B – число резервных ПУ для каждого из основных ПУ, задействованных в решении задачи.

Однако при этом вероятность безотказной работы ВП СУ по сравнению с предыдущим подходом увеличивается и в данном случае будет составлять

$$P_B = (1 - (1 - P_{ПУ})^{B+1})^H,$$

где B – число резервных ПУ для каждого из основных ПУ; H – число основных ПУ.

Можно использовать третий подход к решению проблемы повышения вероятности безотказной работы ВП СУ путем так называемого скользящего резервирования ПУ [45], суть которого заключается в следующем. Поскольку мы приняли, что все ПУ в составе ВП СУ одинаковые, то нет необходимости создавать поэлементный резерв для каждого отдельного ПУ. Достаточно иметь некоторый общий (скользящий) резерв, т.е. набор резервных ПУ, каждый из которых можно использовать для замены любого отказавшего основного ПУ (рисунок 3.11).

В этом случае вероятность безотказной работы ВП СУ будет определяться выражением [45]

$$P_B = \sum_{k=0}^B C_{H+B}^k P_{ПУ}^{H+B-k} (1 - P_{ПУ})^k, \quad (3.3)$$

где B – число резервных ПУ; H – число основных ПУ.

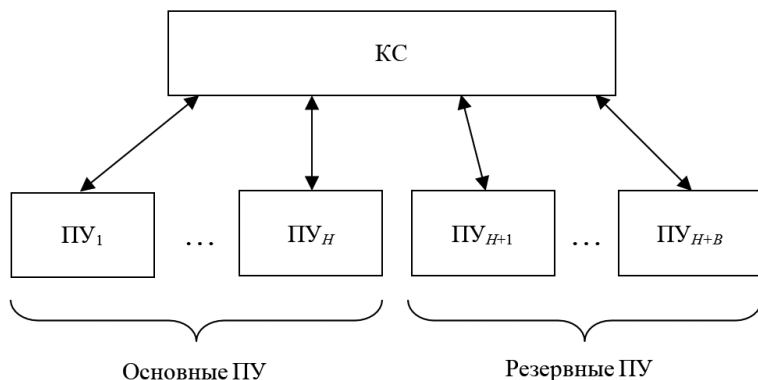


Рис. 3.11. Повышение отказоустойчивости с помощью скользящего резерва ПУ

При этом общее число ПУ в системе будет равно

$$N = H + B.$$

Очевидно, что последний способ повышения вероятности безотказной работы ВП СУ наиболее эффективен, поскольку при его использовании обеспечивается та же вероятность безотказной работы системы по сравнению с предыдущими способами, но при существенно меньших аппаратурных и, соответственно, стоимостных затратах. Продемонстрируем это на примере. Допустим, что число H основных ПУ, задействованных в решении задачи управления, равно 2, а вероятность безотказной работы отдельного ПУ равна 0,7, т.е. $P_{\text{ПУ}} = 0,7$. На рисунке 3.12 для рассмотренных выше различных способов резервирования приведены графики зависимости вероятности безотказной работы ВП СУ от числа B резервных ПУ.

Как видно из графиков, способ скользящего резервирования при одинаковой вероятности безотказной работы системы требует наличия существенно меньшего числа резервных ПУ по сравнению с другими способами резервирования и, соответственно, он эффективнее с точки зрения стоимости.

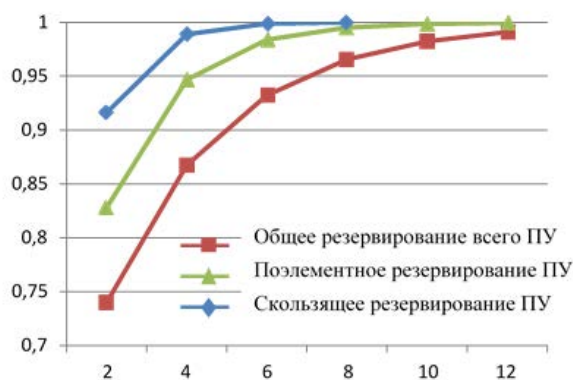


Рис. 3.12. Зависимости вероятности безотказной работы ВП СУ для различных способов структурного резервирования

Обобщая вышеизложенное, можно сделать вывод, что основными преимуществами применения способа структурного резервирования для повышения надежности ВП СУ являются:

- возможность достижения заданных показателей надежности ВП СУ только путем введения в ее состав дополнительных ПУ;
- относительная простота реализации процедуры замены отказавшего ПУ на резервный (просто путем перезапуска всех подзадач, решавшихся на отказавшем ПУ, на одном из резервных ПУ).

В то же время этот способ имеет и серьезные недостатки, затрудняющие, а порой и делающие невозможным его применение при создании бортовых ВП СУ ВС с архитектурой ИМА:

– введение в состав ВП СУ дополнительных ПУ, которые большую часть времени могут простаивать (т.е. не быть задействованы в полезном вычислительном процессе до момента выхода из строя одного из основных ПУ), что приводит к увеличению массогабаритных характеристик ВП СУ и ее энергопотребления и в то же время никак не сказывается на быстродействии ВП;

– различная, в общем случае, вычислительная нагрузка, приходящаяся на основные и резервные ПУ (в зависимости от типа и важности решаемых задач резервные ПУ могут не выполнять никаких операций, могут выполнять накопление данных, необходимых для запуска подзадач, могут выполнять параллельно отдельные подзадачи), приводит к температурному дисбалансу, что, как это будет показано ниже, ведет к снижению вероятности безотказной работы ВП СУ;

– все резервные ПУ выполняют одинаковый набор действий (каждый из них должен иметь возможность заменить любой из основных ПУ), в связи с этим объем решаемых задач обеспечения отказоустойчивости ограничивается производительностью резервных ПУ;

– свободные вычислительные ресурсы основных ПУ (если таковые имеются) никак не используются в процессе решения задачи управления.

Можно предложить иной подход к проблеме повышения надежности ВП СУ, лишенный указанных выше недостатков.

3.2.2. Резервирование производительности

Поскольку мы приняли, что все ПУ, входящие в состав ВП СУ, одинаковые, то это обстоятельство открывает возможности перераспределения (перераспределения) подзадач между отдельными ПУ в случае отказов некоторых из них. Учитывая это соображение, можно предложить способ резервирования, основанный на идее использования «свободной» производительности ПУ. Суть этой идеи заключается в следующем. Допустим, что каждый из ПУ имеет некоторый резерв производительности, как показано на рисунке 3.13. при этом под резервом производительности ПУ понимается возможность решения с его помощью за отведенное время $T_{ц}^3$ большего числа подзадач,

чем это было ему задано при исходном распределении множества подзадач Q по ПУ. Тогда в случае выхода из строя некоторого ПУ подзадачи, закрепленные за ним, можно перераспределить между другими основными ПУ, имеющими резерв производительности, не выходя при этом за ограничения по времени решения общей задачи (рисунок 3.13).

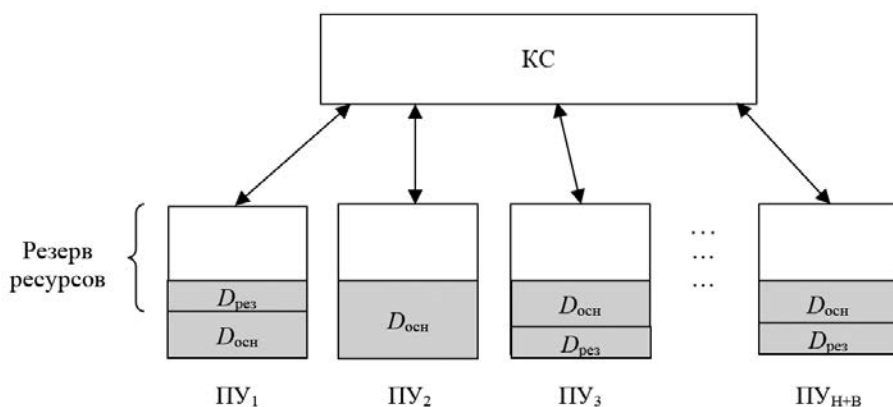


Рис. 3.13. Распределение нагрузки в ВП СУ с резервированием производительности

Предлагаемый подход имеет следующие преимущества:

- для достижения заданного уровня надежности не нужно вводить в состав ВП СУ дополнительные ПУ, а достаточно создать резерв производительности на уже имеющихся ПУ, что достигается либо путем его «закладки» на этапе создания ВП СУ, либо просто заменой процессоров на более производительные;
- в отличие от способа скользящего структурного резервирования данный подход позволяет максимально задействовать вычислительные ресурсы всех ПУ, входящих в состав ВП СУ, для обеспечения надежности ее функционирования.

Однако данный подход имеет один недостаток, заключающийся в значительном усложнении процедуры восстановления вычислительного процесса в ВП СУ, поскольку при этом, в случае возникновения отказа, требуется каким-то образом осуществить перераспределение подзадач общей задачи на работоспособные ПУ, а не просто перенести их с отказавшего ПУ на резервное, как это осуществляется при использовании способа скользящего структурного резервирования.

Рассмотрим основные принципы применения способа резервирования производительности для повышения надежности ВП СУ. Для наглядности и простоты дальнейших рассуждений рассмотрим сначала частные случаи.

Допустим, что в состав вычислительного устройства ВП СУ входят два основных ПУ (т.е. $H = 2$) с производительностью $S_{\text{ПУ}}$, которые обеспечивают решение всех подзадач задачи управления $G(Q, X)$ за время (см. формулу (3.1))

$$T_B = \frac{\sum_{i=1}^{M_{оп}} D_i}{2S_{ПУ}},$$

где $M_{оп}$ – число операционных вершин в графе $G(Q, X)$, D_i , $i = (1, M)$ – вычислительная сложность подзадачи, приписанной i -й операционной вершине графа.

При этом вероятность безотказной работы такого вычислительного устройства (ВУ), состоящего из двух ПУ, будет равна

$$P_B = P_{ПУ}^2.$$

Заменим в устройстве ПУ с производительностью $S_{ПУ}$ на ПУ с производительностью $S_{ПУ}^*$ (рисунок 3.14).

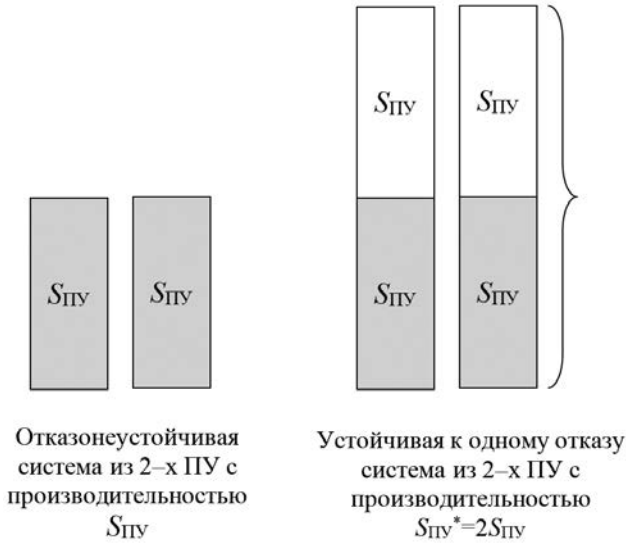


Рис. 3.14. Повышение отказоустойчивости за счет увеличения производительности ПУ

В этом случае при выходе из строя одного из ПУ второй сможет обеспечить решение задачи $G(Q, X)$ за то же время T_B , поскольку его производительности для этого достаточно. Поэтому вероятность безотказной работы двух ПУ с производительностью $2S_{ПУ}$ будет уже равна

$$P_B = 1 - (1 - P_{ПУ})^2 = 1 - (1 - 2P_{ПУ} + P_{ПУ}^2) = 2P_{ПУ} - P_{ПУ}^2,$$

что на величину $2(P_{ПУ} - P_{ПУ}^2)$ больше, чем в случае, когда производительность ПУ равна $S_{ПУ}$.

Рассмотрим теперь случай, когда в решении задачи $G(Q, X)$ задействовано три ПУ с производительностью $S_{ПУ}$. При этом время решения подзадач графа $G(Q, X)$ на таком вычислительном устройстве приблизительно составит (см. формулу (3.1))

$$T_B = \frac{\sum_{i=1}^{M_{оп}} D_i}{3S_{ПУ}}$$

а вероятность безотказной работы ВУ будет равна

$$P_B = P_{ПУ}^3.$$

Но если мы заменим в таком устройстве ПУ с производительностью $S_{ПУ}$ на ПУ с производительностью $S_{ПУ}^* = 2S_{ПУ}$, то при сохранении того же времени вычислений T_B (для самого худшего случая) вероятность безотказной работы устройства повышается и будет уже в данном случае составлять

$$P_B = 1 - (1 - P_{ПУ})^3 - 3 \cdot P_{ПУ}(1 - P_{ПУ})^2 = 3 \cdot P_{ПУ}^2 - 2 \cdot P_{ПУ}^3.$$

В случае если задача должна быть решена вычислительным устройством за время T_B , то система, состоящая из трех ПУ с производительностью $S_{ПУ}^* = 2S_{ПУ}$, сможет парировать один отказ, т.е. будет сохранять работоспособность при выходе из строя любого ПУ из трех (рисунок 3.15).

Заметим, что для обеспечения парирования одного отказа производительность ПУ не обязательно нужно повышать в два раза. Достаточно повысить производительность ПУ до величины

$$S_{ПУ}^* = \frac{3}{2}S_{ПУ},$$

как показано на рисунке 3.16.

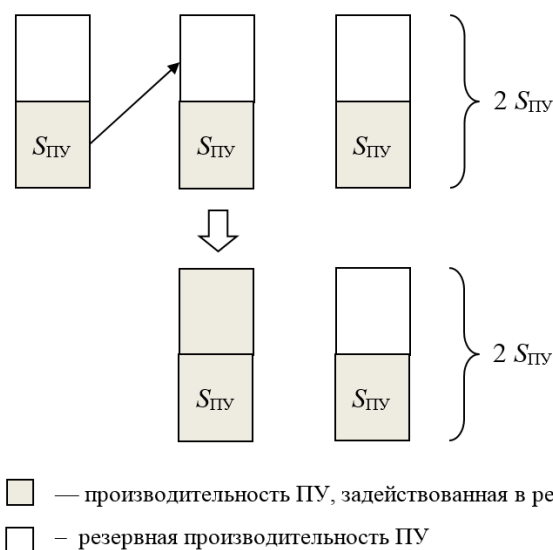


Рис. 3.15. Перераспределение задач в случае отказа одного ПУ

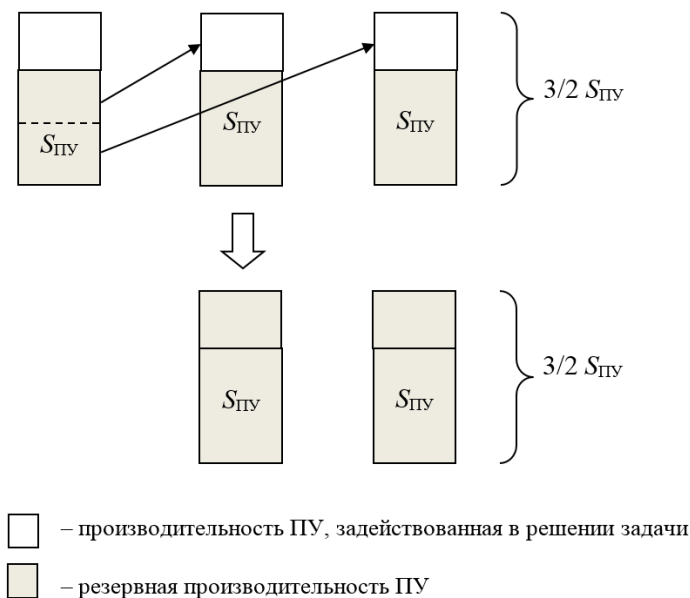


Рис. 3.16. Перераспределение задач в случае отказа одного ПУ

При этом вычислительное устройство, состоящее из трех таких ПУ, обеспечит парирование отказа одного ПУ, поскольку два оставшихся смогут обеспечить решение своих подзадач, а также подзадач вышедшего из строя ПУ за то же время T_B . При этом вероятность безотказной работы такой системы составит

$$P_B = P_{ПУ}^3 + 3P_{ПУ}^2(1 - P_{ПУ}) = P_{ПУ}^2(3 - 2 \cdot P_{ПУ}).$$

Если продолжить рассуждения далее, то можно показать, что вычислительное устройство, содержащее H основных ПУ с производительностью

$$S_{ПУ}^* = S_{ПУ} + \frac{1}{H-1} S_{ПУ} = \frac{H}{H-1} S_{ПУ}$$

каждый, будет парировать один отказ и иметь вероятность безотказной работы

$$P_B = P_{ПУ}^H + H P_{ПУ}^{H-1} \cdot (1 - P_{ПУ}).$$

Для того чтобы парировать B отказов, производительность всех H основных ПУ необходимо повысить до величины

$$S_{ПУ}^* = S_{ПУ} + \frac{B}{H-B} S_{ПУ} = \frac{H}{H-B} S_{ПУ},$$

при этом вероятность безотказной работы системы будет равна

$$P_B = \sum_{k=0}^B C_H^k P_{ПУ}^{H-k} (1 - P_{ПУ})^k.$$

Необходимо отметить, что приведенные выше рассуждения несколько идеализированы, поскольку предполагается, что задачи, решаемые на отдельных ПУ, можно «дробить» на отдельные подзадачи с произвольным «квантом». Однако несмотря на это допущение, сам подход к повышению надежности за счет резервной (избыточной) производительности ПУ применим и в общем случае.

Обобщая вышесказанное, можно сделать вывод, что способ, основанный на резервировании производительности, имеет следующие особенности:

- более сложную по сравнению со способом скользящего структурного резервирования реализацию (в случае отказа необходимо обеспечить размещение подзадач на исправных ПУ с учетом уже выполняющихся подзадач и имеющихся резервных ресурсов);

- возможность задействовать все свободные ресурсы всех ПУ (т.е. ресурсы, не используемые для решения подзадач задачи управления) в целях обеспечения устойчивости к отказам;

- возможность выровнять вычислительную нагрузку, приходящуюся на отдельные ПУ, за счет оптимального размещения подзадач по работоспособным ПУ.

Можно заметить, что с целью повышения надежности ВП СУ можно одновременно применять как способ резервирования производительности, так и способ скользящего структурного резервирования. Действительно, если в составе ВП СУ имеется N ПУ, каждый из которых имеет некоторый резерв производительности, то в этом случае часть ПУ можно полностью освободить от вычислительной нагрузки, перераспределив их подзадачи на резервы других ПУ, тем самым создав полностью свободные резервные ПУ, как это имеет место при использовании способа скользящего структурного резервирования (рисунок 3.17).

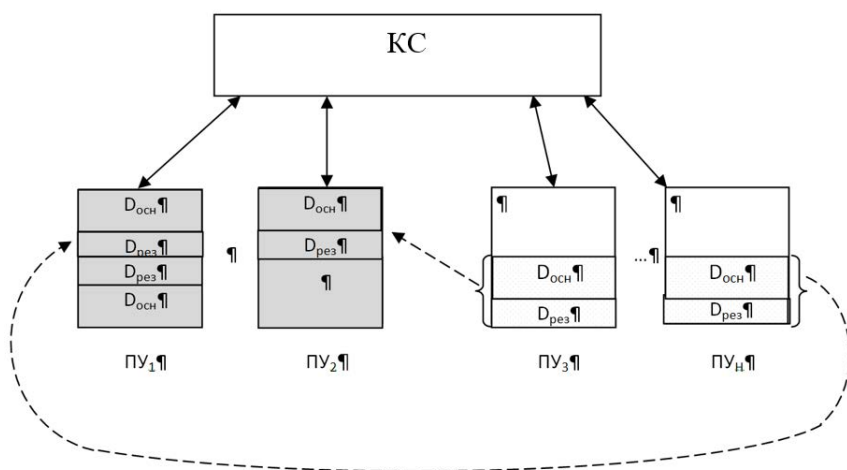


Рис. 3.17. Совмещение способов резервирования производительности и скользящего структурного резервирования

3.2.3. Выравнивание вычислительной нагрузки

Выше мы считали, что вероятность безотказной работы всех ПУ, входящих в состав ВП СУ, одинакова. В действительности это будет не так, поскольку величина данного параметра у различных ПУ будет различной вследствие отличий в условиях функционирования. Одним из факторов, определяющих различные значения вероятности безотказной работы отдельных ПУ, является их вычислительная нагрузка, от которой, в свою очередь, зависит их температура нагрева ПУ. Рассмотрим эту проблему подробнее.

В процессе своей работы ПУ выделяют тепло, причем увеличение вычислительной нагрузки и, как следствие, интенсивности работы, которое может быть выражено в числе переключений логических элементов, приводит к увеличению тепловыделения ПУ (вследствие разницы в нагрузке температура микропроцессоров и других элементов современных ПУ может отличаться на 30–40°C). Учитывая, что потребляемая мощность ПУ растет пропорционально количеству выполняемых операций в единицу времени (т.е. нагрузке), можно сделать вывод о пропорциональном увеличении выделения тепла при увеличении нагрузки ПУ. Будем считать, что при нулевой нагрузке (т.е. когда ПУ не выполняет никаких действий по решению задачи) температура ПУ равна T_0 , тогда при нагрузке D его температура T составит

$$T = T_0 + \Delta T = T_0 + k_D \cdot D,$$

где k_D – коэффициент пропорциональности между увеличением нагрузки и температуры.

В свою очередь, увеличение температуры ПУ ведет уже к непропорциональному увеличению интенсивности его отказов, что отражено в работах [13, 31, 43]. Как показывает анализ данных работ, зависимость увеличения интенсивности отказов ПУ от температуры является степенной, причем упрощенно можно считать, что при увеличении температуры на 10°C интенсивность отказов удваивается, т.е.

$$\lambda = \lambda_0 \cdot 2^{\frac{\Delta T}{10}},$$

где λ_0 – интенсивность отказов при некоторой заданной температуре T_0 (обычно 40°C).

Для примера на рисунке 3.18 показан график зависимости интенсивности отказов ПУ от разности температур ($\lambda_0 = 0,00001 \text{ ч}^{-1}$).

Вероятность безотказной работы ПУ при постоянной интенсивности отказов рассчитывается по формуле

$$P_{\text{ПУ}} = e^{-\lambda t},$$

где t – время работы.

Соответственно, если формулу переписать с учетом разности температур, то получим

$$P_{\text{ПУ}} = e^{-\lambda_0 \cdot 2^{\frac{\Delta T}{10}} \cdot t} = P_{\text{ПУ}0}^{2^{\frac{\Delta T}{10}}}$$

где $P_{\text{ПУ}0}$ – вероятность безотказной работы ПУ при «базовой» температуре.

Введем обозначение $k(\Delta T) = 2^{\frac{\Delta T}{10}}$, тогда $P_{\text{ПУ}} = P_{\text{ПУ}0}^{k(\Delta T)}$. График зависимости вероятности безотказной работы ПУ от разности температур приведен на рисунке 3.19. Данный график показывает, насколько будут отличаться вероятности безотказной работы ПУ, работавшего при «базовой» температуре, и ПУ, работавшего при температуре, отличающейся на (ΔT) .

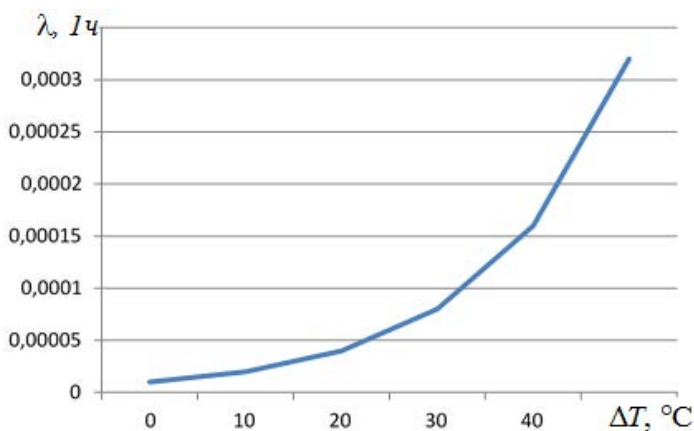


Рис. 3.18. График зависимости интенсивности отказов ПУ от разности температур ($\Delta T = T - T_0$)

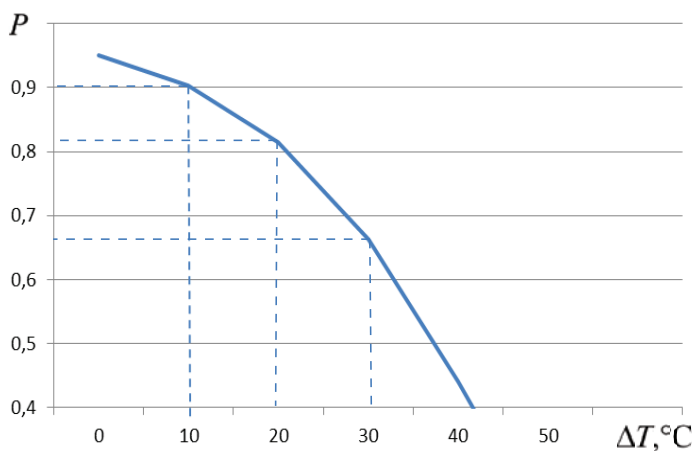


Рис. 3.19. График зависимости вероятности безотказной работы ПУ от разности температур

Обратим внимание на тот факт, что равное отклонение температуры от некоторого значения в сторону увеличения или уменьшения приводит к неодинаковому изменению вероятности безотказной работы, причем увеличение температуры приводит к большему изменению, чем уменьшение ($P(20^\circ\text{C}) - P(10^\circ\text{C}) < P(30^\circ\text{C}) - P(20^\circ\text{C})$). Поскольку температура ПУ зависит в том числе и от нагрузки, то логично предположить, что при равной суммарной нагрузке ВП с равномерно и неравномерно нагруженными ПУ будут иметь разную вероятность безотказной работы. При этом чем более равномерно вычислительная нагрузка будет распределена между всеми ПУ, входящими в состав ВП, тем выше будет вероятность ее безотказной работы.

Допустим, что в состав ВП СУ входит N ПУ, причем должно обеспечиваться парирование B отказов. Рассмотрим два случая распределения нагрузки:

– равномерное распределение (рисунок 3.20, а), т.е. суммарное число вычислительных операций D , выполняемых каждым из N ПУ, одинаково и равно

$$D = \frac{D_{\text{общ}}}{N}, \quad T = T_0 + k_D D,$$

где $D_{\text{общ}}$ – суммарное число вычислительных операций решаемой задачи;

– неравномерное распределение (рисунок 3.20, б), предполагающее наличие трех групп ПУ по нагрузке – с повышенной ($D_+ - N_+$ ПУ в группе), со средней ($D_p - N_p$ ПУ в группе) и с пониженной нагрузкой ($D_- - N_-$ ПУ в группе) ($N_+ + N_p + N_- = N$, $N_+ > 0$, $N_- > 0$). При этом отдельные ПУ в группах имеют следующую нагрузку и температуру:

$$D_- = \frac{D_{\text{общ}}}{N} - \frac{\Delta D_{\pm}}{N_-}, \quad T_- = T_0 + k_D D_-,$$

$$D_p = \frac{D_{\text{общ}}}{N}, \quad T_p = T_0 + k_D D,$$

$$D_+ = \frac{D_{\text{общ}}}{N} + \frac{\Delta D_{\pm}}{N_+}, \quad T_+ = T_0 + k_D D_+,$$

где D_- – нагрузка на один ПУ в группе с пониженной нагрузкой;

D_p – нагрузка на один ПУ в группе со средней нагрузкой;

D_+ – нагрузка на один ПУ в группе с повышенной нагрузкой;

ΔD_{\pm} – разница в нагрузке между ПУ с повышенной (пониженной) нагрузкой и таким же числом ПУ со средней нагрузкой.

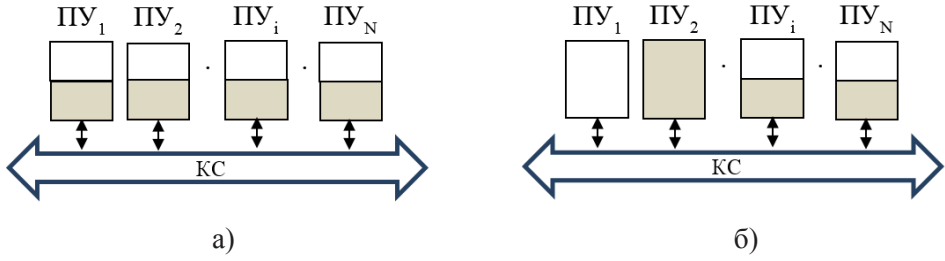


Рис. 3.20. ВП СУ с равномерно и неравномерно нагруженными ПУ

Если положить, что $N_+ = N_- = 0$ и исключить выражения для расчета D_+ и D_- , то формулы расчета нагрузки и температуры отдельных ПУ для ВП с неравномерной загрузкой ПУ сводятся к аналогичным формулам для ВП с равномерной загрузкой. Поэтому если записать формулу для расчета вероятности безотказной работы для второго случая, то она будет пригодна и для первого. Данную формулу можно вывести на основе формулы (3.3) с учетом того факта, что в состав ВП входят ПУ с тремя значениями вероятности безотказной работы, т.е. она должна представлять собой сумму вероятностей перехода во все работоспособные конфигурации:

$$P_B = \sum_{i=0}^{N_p} \sum_{j=0}^{N_+} \sum_{k=0}^{N_-} f(P_{\text{ПУ}}, \Delta T, N_p, N_+, N_-, B), \quad (3.4)$$

где $f(P_{\text{ПУ}}, \Delta T, N_p, N_+, N_-, B) =$

$$= \begin{cases} C_{N_p}^{Ni} P_{\text{ПУ}}^{N_p-i} (1 - P_{\text{ПУ}})^i C_{N_+}^j P_+^{N_+-j} (1 - P_+)^j C_{N_-}^k P_-^{N_--k} (1 - P_-)^k & \text{при } i + j + k \leq B \\ 0 & \text{при } i + j + k > B; \end{cases}$$

$$P_+ = (P_{\text{ПУ}})^{k(\Delta T)};$$

$$P_- = (P_{\text{ПУ}})^{k(-\Delta T)}.$$

В представленной формуле обозначено:

$C_{N_p}^i P_{\text{ПУ}}^{N_p-i} (1 - P_{\text{ПУ}})^i$ – выражение для учета вклада в величину вероятности безотказной работы ПУ со средней нагрузкой;

$C_{N_+}^j P_+^{N_+-j} (1 - P_+)^j$ – выражение для учета вклада в величину вероятности безотказной работы ПУ с повышенной нагрузкой;

$C_{N_-}^k P_-^{N_--k} (1 - P_-)^k$ – выражение для учета вклада в величину вероятности безотказной работы ПУ с пониженной нагрузкой.

Очевидно, что при $N_+ = 0$ и $N_- = 0$ формула (3.4) сводится к формуле (3.3). Также очевидно, что данная формула может быть модифицирована для любого числа групп устройств с различной вероятностью безотказной работы (с различной температурой/нагрузкой применительно для рассматриваемо-

го случая). Для этого необходимо для каждой i -й группы определить вероятность безотказной работы P_i , добавить выражение для учета вклада в величину P данной группы $\tilde{N}_i^{k_i} P_i^{N_i - k_i} (1 - P_i)^{k_i}$ и соответствующую сумму $\sum_{k_i=0}^{N_i}$.

При сравнении вероятности безотказной работы при равномерном и неравномерном распределении нагрузки между ПУ необходимо учитывать ряд параметров:

- N_+ и N_- (будем полагать, что $N_+ = N_- = N_{\pm}$);
- $\Delta T_{\pm} = T_+ - T_0 = T_0 - T_-$ (так как $N_+ = N_-$);
- N ;
- $P_{\text{пу}}$

Определим характер зависимости получаемого эффекта от выравнивания нагрузки от перечисленных параметров (получение такой зависимости позволяет определить направление дальнейших исследований). Для этого выполним расчет разности вероятностей безотказной работы для ВП СУ с выравниванием нагрузки между ПУ и без него в широком диапазоне значений указанных факторов (диапазоны значений выбирались с учетом значений, актуальных для существующих и перспективных конфигураций ВП СУ):

- $N_{\pm} \in [2, 8]$;
- $\Delta T_{\pm} \in [5, 25^\circ\text{C}]$;
- $N \in [2, 150]$;
- $P_{\text{пу}} = 0,8, 0,9, 0,99$ (данные значения приблизительно соответствуют времени работы ПУ 1000, 10000 и 20000 часов при $\lambda = 0,00001$ 1/час).

Проведенные исследования показали, что:

- в большинстве случаев равномерное распределение нагрузки позволяет повысить вероятность безотказной работы ВП СУ (исключение – ВП, в которых допустимое число отказов B близко к N ($N = 2, B = 1$; $N = 5, B = 4$; $N = 11, B = 10$ и т.п.));
- в общем виде зависимость разности вероятностей безотказной работы ВП СУ без выравнивания и с выравниванием нагрузки ($\Delta P = P_{\text{нр}} - P_{\text{р}}$, где $P_{\text{нр}}$ и $P_{\text{р}}$ вероятности безотказной работы без выравнивания и с выравниванием нагрузки) имеет вид, представленный на рисунке 3.21. (На рисунке 3.22 представлен вид графиков зависимости $P_{\text{нр}}$ и $P_{\text{р}}$ от числа ПУ в ВП при постоянном числе парируемых отказов, поясняющий вид зависимости разности данных величин. Из рисунка видно, что графики «выходят» из одной точки и «сходятся» также в одной точке, соответственно, максимальная разница между ними имеет место при средних величинах N);
- при увеличении ΔT_{\pm} и N_{\pm} увеличивается эффект от применения выравнивания нагрузки (рисунок 3.21);

– при уменьшении вероятности безотказной работы ПУ и числа допустимых отказов величина максимального эффекта от выравнивания нагрузки увеличивается, при этом чем больше $P_{\text{пу}}$ и B , тем большему N соответствует максимум разности вероятностей безотказной работы ВП СУ с выравниванием и без выравнивания нагрузки (рисунки 3.23 и 3.24). Таким образом, выравнивание нагрузки можно эффективно применять при использовании в составе ВП СУ процессорных узлов с относительно высокой интенсивностью отказов.

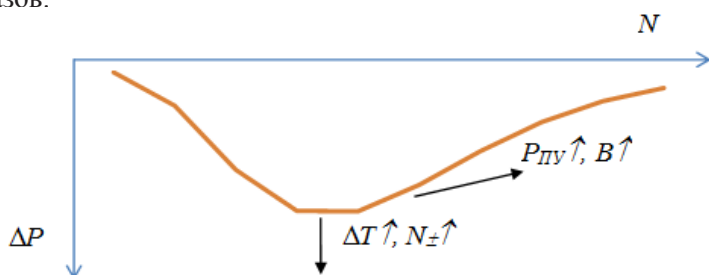


Рис. 3.21. Вид зависимости разности вероятностей безотказной работы ВП СУ с выравниванием и без выравнивания нагрузки при различном числе ПУ

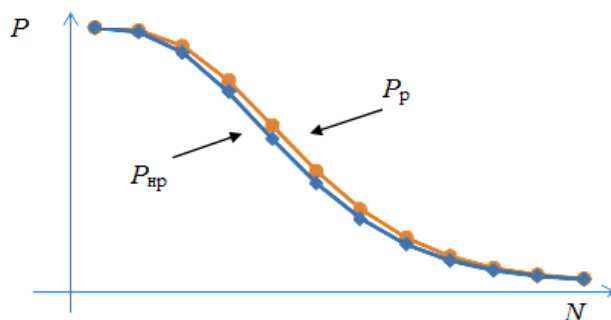


Рис. 3.22. Вид графиков вероятностей безотказной работы ВП СУ с выравниванием и без выравнивания нагрузки при различном числе ПУ

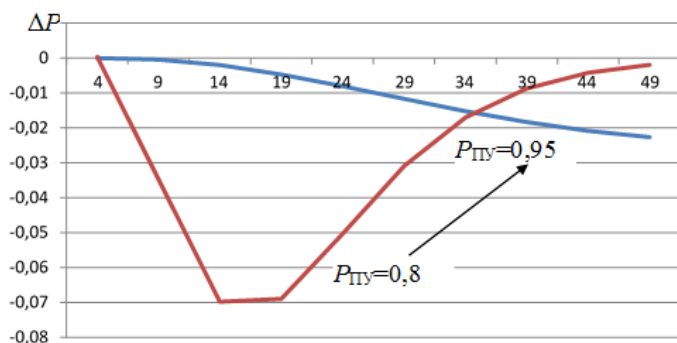


Рис. 3.23. Пример, демонстрирующий влияние $P_{\text{пу}}$ на разность вероятностей безотказной работы ВП СУ с выравниванием и без выравнивания нагрузки ($\Delta T_{\pm} = 15^{\circ}\text{C}$, $B = 3$, $N_{\pm} = 2$)

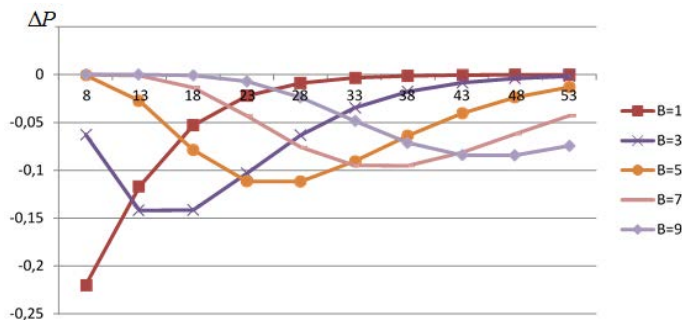


Рис. 3.24. Пример, демонстрирующий влияние $P_{пу}$ на разность вероятностей безотказной работы ВУ ВП СУ с выравниванием и без выравнивания нагрузки ($\Delta T_{\pm} = 15^{\circ}\text{C}$, $P_{пу} = 0,8$, $N_{\pm} = 4$)

Проведенные исследования позволяют сделать вывод, что использование способа выравнивания нагрузки позволяет повысить надежность характеристики ВП СУ, причем эффект от его применения возрастает при увеличении числа ПУ с отклонением нагрузки от среднего значения и увеличении отклонения от усредненной нагрузки D_p .

3.2.4. Резюме

В результате проведенных исследований показано, что современные ВП СУ ВС строятся с использованием ИМА. Предложен новый способ обеспечения надежности таких ВП – способ резервирования производительности, позволяющий обеспечить требуемые характеристики надежности без существенного увеличения аппаратных затрат (без введения в состав ВП СУ дополнительных ПУ), а только за счет создания необходимого резерва производительности в имеющихся ПУ.

Способ резервирования производительности позволяет использовать все ресурсы работоспособных ПУ для улучшения характеристик ВП СУ, поскольку не требуется выделять специальные резервные ПУ, которые в процессе работы СУ ВС не участвуют в решении задачи управления.

Получены выражения, определяющие резерв производительности отдельных ПУ, необходимый для достижения заданного уровня отказоустойчивости ВП СУ.

Также получены выражения, определяющие зависимость вероятности безотказной работы отдельного ПУ, входящего в состав ВП СУ, от температуры его нагрева, которая, в свою очередь, зависит от вычислительной нагрузки данного ПУ. Показано, что чем более равномерно вычислительная нагрузка будет распределена между всеми ПУ, входящими в состав ВП, тем выше будет вероятность ее безотказной работы.

Исследование метода повышения надежности ВП СУ за счет выравнивания вычислительной нагрузки отдельных ПУ, показало, что использование данного метода позволяет повысить надежность характеристики ВП.

3.3. Методы и алгоритмы диспетчирования вычислительных подсистем

3.3.1. Способы организации диспетчера

В предыдущем подразделе было показано, что при создании высоконадежных ВП СУ целесообразно использовать способ резервирования производительности, так как данный способ резервирования позволяет сократить аппаратные расходы на обеспечение требуемого уровня надежности по сравнению со способом структурного резервирования. При этом в качестве резерва производительности могут быть использованы как ресурсы резервных (незагруженных) ПУ, так и свободные ресурсы «основных» ПУ, задействованных в решении задачи управления. Данный способ обладает рядом преимуществ, поскольку, во-первых, позволяет полностью использовать все имеющиеся ресурсы отдельных ПУ для решения задачи управления, во-вторых, позволяет использовать все избыточные ресурсы для повышения надежности ВП, а в-третьих, позволяет дополнительно повысить показатели надежности ВП за счет возможности выравнивания нагрузки ПУ при реконфигурации ВП в случае возникновения отказов. Однако при использовании данного подхода возникает вопрос: кто и каким образом должен обнаруживать отказы отдельных ПУ и перераспределять подзадачи между работоспособными ПУ в случае их возникновения? Очевидно, что для обеспечения этого процесса в составе системы должен находиться некоторый диспетчер, который может быть организован разными способами [11, 12]. Рассмотрим эти способы более подробно.

3.3.1.1. Организация диспетчера с помощью выделенного процессорного узла

При таком способе среди всего множества устройств, входящих в состав ВП СУ, необходимо выделить один специальный ПУ, на который возлагаются функции диспетчера всей системы в целом (рисунок 3.25) [14, 28]. Данный ПУ должен выполнять следующие функции.

1. Тестирование всех ПУ, входящих в состав ВП СУ, и определение их работоспособности. В простейшем случае это тестирование может заключаться в получении определенного «отклика» на запрос от всех ПУ.

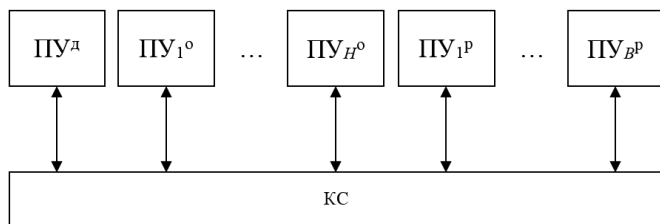
2. В случае если хотя бы один из основных $ПУ_i^o$ ($i = 1, H$), задействованных в решении задачи $G(Q, X)$, не «откликается», то определение рабочей конфигурации ВП СУ, обеспечивающей решение задачи $G(Q, X)$ за заданное время.

3. Распределение (перераспределение) подзадач, соответствующих вершинам графа $G(Q, X)$, по работоспособным ПУ, входящим в новую рабочую конфигурацию ВП СУ.

4. Запуск задачи $G(Q, X)$ на решение на новой конфигурации ВП СУ.

Данный подход – наиболее очевидный и простой с точки зрения его организации, но в то же время он обладает рядом существенных недостатков.

Во-первых, он требует введения в состав системы дополнительного ПУ, реализующего функции диспетчера.



$ПУ^д$ – ПУ, реализующее функции диспетчера
 $ПУ_i^о$ ($i = 1, H$) – основные ПУ, задействованные в решении задачи управления
 $ПУ_i^р$ ($i = 1, B$) – резервные ПУ

Рис. 3.25. Реализация диспетчера с помощью выделенного ПУ

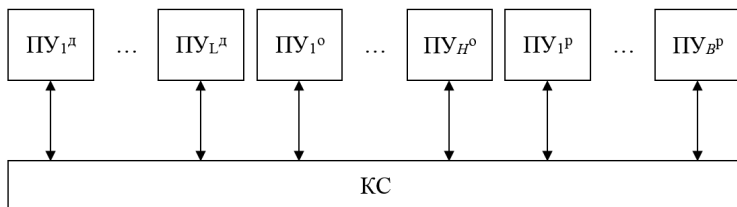
Во-вторых, отказ такого выделенного диспетчера приводит к выходу из строя ВП СУ в целом. Иными словами, реализация диспетчера на выделенном ПУ резко снижает надежность всей ВП СУ. Действительно, вероятность безотказной работы всей системы в целом в данном случае будет составлять

$$P_B^* = P_{ПУ} P_B,$$

где $P_{ПУ}$ – вероятность безотказной работы ПУ, выделенного под диспетчер;
 P_B – вероятность безотказной работы ВП СУ без выделенного ПУ.

Поскольку $P_{ПУ} < 1$, то очевидно, что $P_B^* < P_B$, т.е. вероятность безотказной работы всей системы в целом уменьшается.

Для того чтобы повысить надежность такой системы, следует, очевидно, осуществить резервирование диспетчера (рисунок 3.26), т.е. ввести в состав системы несколько дополнительных резервных ПУ, способных выполнять функции диспетчера в случае отказа основного диспетчера.



$ПУ_i^д$ ($i = 1, L$) – ПУ, реализующее функции диспетчера

Рис. 3.26. Резервирование ПУ, реализующего функции диспетчера

При этом вероятность безотказной работы такой системы будет составлять

$$P_B^* = (1 - (1 - P_{ПУ})^{L+1}) P_B,$$

где L – число резервных ПУ, выполняющих функции диспетчера.

Из последней формулы можно определить количество L резервных диспетчеров, необходимое для обеспечения заданной P_B^3 вероятности безотказной работы всей системы в целом, приняв $P_B^* = P_B^3$:

$$L = \log_{(1-P_{\text{ПВ}})} \left(1 - \frac{P_B^3}{P_B} \right) - 1.$$

При этом общее количество ПУ в ВП, соответственно, будет составлять:

$$N + (H + B + L + 1), \quad (3.5)$$

где H – число основных ПУ, задействованных в решении задачи управления; B – число ПУ, резервирующих основные ПУ; L – число ПУ, резервирующих диспетчер.

Однако даже при наличии резервных ПУ, способных заменить основного диспетчера, остается проблема диагностирования его работоспособности и переключения функции диспетчирования на один из резервных ПУ. Возникает вопрос, кто и каким образом должен это сделать? Получается, что в системе должен быть еще один диспетчер верхнего уровня, который должен следить за основным диспетчером и в случае его отказа либо включать в работу резервный диспетчер, либо перераспределять граф задачи $G(Q, X)$ на работоспособные ПУ. А кто будет отслеживать работу этого диспетчера верхнего уровня?

Таким образом, наличие центрального диспетчера, реализованного тем или иным способом, существенно снижает надежность ВП СУ и затрудняет процесс восстановления вычислений в ней. Этих недостатков можно избежать путем использования способа распределенной организации диспетчера ВП СУ.

3.3.1.2. Организация распределенного диспетчера

Основная идея данного подхода заключается в том, что все функции диспетчера реализуются не на каком-либо отдельном ПУ, а распределяются на все ПУ, входящие в состав системы, причем на каждом отдельном ПУ реализуется свой локальный диспетчер (ЛД) (рисунок 3.27).

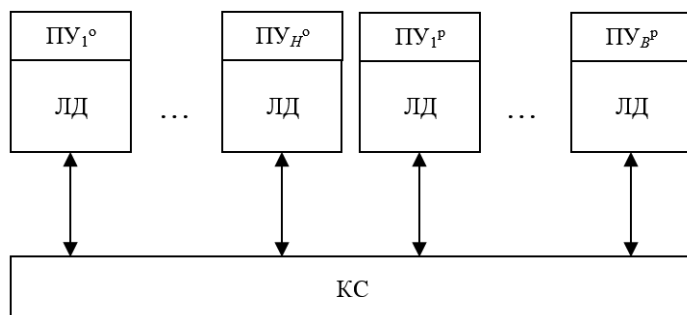


Рис. 3.27. ВП СУ с распределенным диспетчером

При этом каждый локальный диспетчер, реализуемый на отдельном ПУ, должен выполнять следующие основные функции [14, 28].

1. Периодически тестировать (диагностировать) работоспособность «своего» ПУ.

2. Периодически посылать ЛД других ПУ «диагностическое» сообщение, подтверждающее работоспособность «его» ПУ.

3. Принимать от ЛД других ПУ «диагностические» сообщения о работоспособности «их» ПУ.

4. Участвовать наравне с локальными диспетчерами других ПУ в процедуре перераспределения подзадач графа $G(Q, X)$ между работоспособными ПУ в случае выхода из строя какого-либо ПУ, задействованного в решении задачи.

Очевидно, что данный подход имеет ряд преимуществ по сравнению с рассмотренными выше [21, 22, 32–35].

1. Поскольку в системе нет отдельного ПУ, реализующего функции центрального диспетчера, и все ПУ равны, то выход из строя любого из них не приводит к катастрофическим последствиям. Локальные диспетчеры работоспособных ПУ обеспечат перераспределение задачи $G(Q, X)$ между ними самостоятельно.

2. Вероятность безотказной работы ВП СУ практически не уменьшается, поскольку в ее состав не вводится никакого дополнительного оборудования.

3. Способ практически не требует каких-либо дополнительных аппаратурных затрат на реализацию ВП СУ, поскольку число ПУ в системе не увеличивается. Безусловно, производительность отдельных ПУ должна быть увеличена с тем, чтобы обеспечить возможность реализации дополнительных процедур локального диспетчера на ПУ без увеличения времени решения задачи управления $G(Q, X)$.

Однако использование распределенного диспетчера приводит к существенному усложнению самой процедуры диспетчирования работы ВП СУ. Действительно, необходимо каким-то образом с помощью множества локальных диспетчеров отслеживать работоспособность отдельных ПУ и перераспределять задачу управления на работоспособные ПУ с целью восстановления вычислительного процесса в случае возникновения отказов. Иными словами, должны быть разработаны методы децентрализованного диспетчирования работы ВП СУ с помощью множества распределенных локальных диспетчеров. Именно этой проблеме и будет посвящена последующая часть настоящего подраздела.

3.3.2. Децентрализованное диспетчирование

Рассмотрим особенности функционирования ВП СУ с распределенным диспетчером при различных способах организации вычислительного процесса. Предварительно уточним ряд определений, которые будут использоваться в дальнейшем.

Под временем $T_{\text{реш}}$ решения задачи управления (временем реакции) будем понимать период времени от момента начала решения задачи

управления $G(Q, X)$ в ВП (считывания исходных данных с источников информации) до момента завершения решения (выдачи команд управления на исполнительные подсистемы).

Под временем $T_{\text{ц}}$ цикла управления будем понимать промежуток времени между двумя последовательными командами управления, выдаваемыми ВП СУ.

Под временем $T_{\text{ВВП}}$ восстановления вычислительного процесса будем понимать период времени от момента обнаружения неисправности в ВП СУ до момента запуска нового решения задачи $G(Q, X)$ после ее устранения.

Под временем $T_{\text{п}}$ потери управления будем понимать период времени, когда задача управления либо вообще не решается, либо решается неверно.

Под временем $T_{\text{ку}}$ формирования команд управления будем понимать период времени от начала очередного цикла решения задачи управления до момента выдачи команд управления на исполнительные подсистемы МО.

В зависимости от способа организации вычислительного процесса в СУ ВС эти времена будут различными. Рассмотрим этот вопрос подробнее.

3.3.2.1. Диспетчирование при однопроцессорном способе организации вычислительного процесса

Как показано в подразделе 3.1 в простейшем случае вся задача управления $G(Q, X)$ может решаться на одном ПУ, в то время как все остальные ПУ, входящие в состав ВП СУ, будут находиться в резерве. Будем считать, что на каждом ПУ физически реализован свой локальный диспетчер, участвующий в процедуре обнаружения неисправностей и восстановления вычислительного процесса в СУ ВС в случае отказа одного из основных ПУ, задействованных в решении задачи управления. При этом процедура восстановления вычислительного процесса (реконфигурации ВП СУ ВС) будет заключаться просто в передаче функции основного (решающего) ПУ на один из резервных.

Временная диаграмма работы ВП при такой организации вычислительного процесса показана на рисунке 3.28.

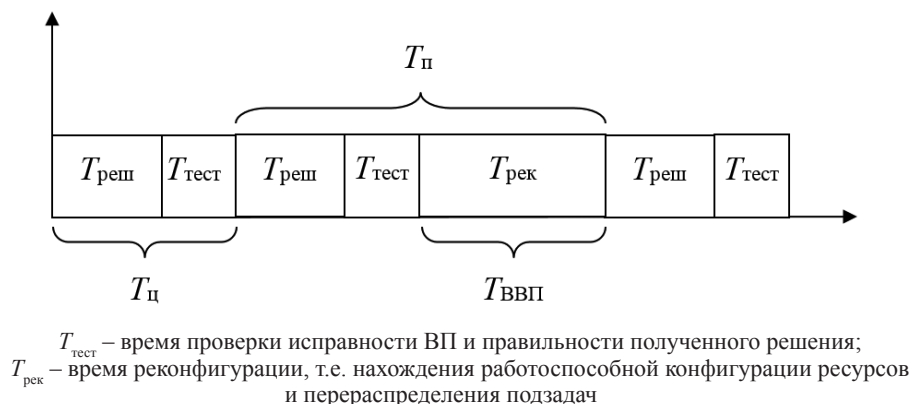


Рис. 3.28. Временная диаграмма работы ВП СУ при однопроцессорном способе организации вычислительного процесса

В целом такая система должна работать следующим образом. Основной решает задачу управления $G(Q, X)$, затрачивая на это время

$$T_{\text{реш}} = \frac{\sum_{i=1}^{M_{\text{оп}}} D_i}{S_{\text{ПУ}}},$$

где $M_{\text{оп}}$ – число операционных вершин в графе $G(Q, X)$;

D_i – вычислительная сложность i -й операционной вершины.

По завершении времени $T_{\text{реш}}$ локальный диспетчер ЛД_{*i*} ($i = 1, N$) осуществляет тестирование «своего» ПУ и передает сообщения о его работоспособности всем остальным ЛД. Допустим, что эта процедура занимает время $T_{\text{тест}}$. Если локальный диспетчер основного (решающего) ПУ подтвердил его работоспособность, то полученные в результате решения задачи управления команды управления выдаются на исполнительные устройства МО. После этого цикла решения повторяется снова и т.д.

Если в очередном цикле решения по завершении процедуры тестирования обнаружилось, что основной ПУ₁ вышел из строя (от его ЛД не пришло сообщение о работоспособности), то локальные диспетчеры работоспособных ПУ запускают процедуру восстановления вычислительного процесса (реконфигурации). При этом один из резервных ПУ (например, ПУ с наименьшим номером) принимает на себя функции основного ПУ. Будем считать, что процедура реконфигурации занимает время $T_{\text{рек}}$. Далее осуществляется перезапуск задачи управления на новом основном ПУ, после чего цикл управления восстанавливается и решение задачи управления вновь осуществляется в циклическом режиме. При таком способе организации вычислительного процесса в СУ ВС имеем следующие зависимости:

$$T_{\text{реш}} = \frac{\sum_{i=1}^{M_{\text{оп}}} D_i}{S_{\text{ПУ}}},$$

$$T_{\text{ц}} = T_{\text{реш}} + T_{\text{тест}}, \quad (3.6)$$

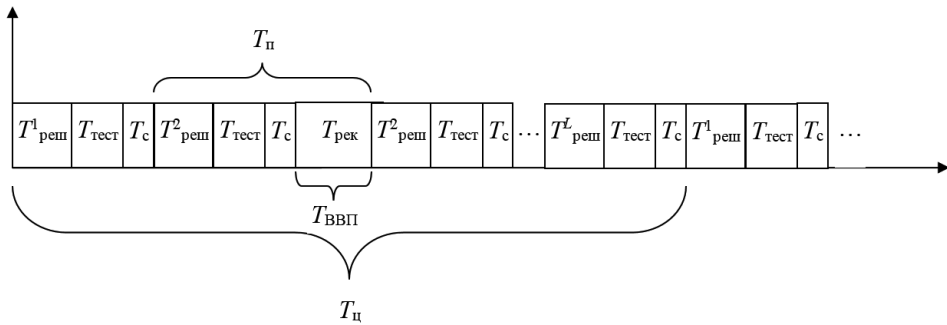
$$T_{\text{ввп}} = T_{\text{рек}},$$

$$T_{\text{п}} = T_{\text{реш}} + T_{\text{тест}} + T_{\text{рек}}.$$

Отметим, что при использовании описанного выше способа организации вычислений и диспетчирования в ВП СУ происходит потеря одного цикла управления. Действительно, тот цикл управления, в котором была обнаружена неисправность основного ПУ, следует считать неверным (т.е. потерянным). Именно поэтому время $T_{\text{п}}$ в данном случае определяется формулой (3.6).

Если объект управления достаточно инерционный, то потеря одного цикла управления не приведет к каким-либо катастрофическим последствиям и поэтому может быть «безболезненной».

Однако в ряде случаев потеря цикла управления недопустима. Это означает, что тестирование работоспособности ПУ должно осуществляться не только после завершения решения задачи управления, но и в процессе решения. Для этого время $T_{\text{реш}}$ должно быть разбито на участки $T_{\text{реш}}^1, T_{\text{реш}}^2, \dots, T_{\text{реш}}^L$, по завершении каждого из которых локальный диспетчер ЛД_{*i*} ($i = 1, N$) должен тестировать работоспособность своих ПУ. При этом локальный диспетчер должен также пересылать всем резервным ПУ промежуточные данные решения задачи $G(Q, X)$ в этих контрольных точках. Последнее необходимо для восстановления вычислительного процесса с этой контрольной точки в случае отказа основного ПУ. В случае обнаружения неисправности основного ПУ локальные диспетчеры резервных ПУ должны выполнить процедуру реконфигурации системы и перезапустить решения задачи $G(Q, X)$ на новом основном ПУ с последней сохраненной контрольной точки, в результате чего цикл управления не будет пропущен. Временная диаграмма работы ВП СУ в однопроцессорном режиме без потери цикла управления показана на рисунке 3.29.



T_c – время сохранения промежуточных данных во всех ПУ, входящих в ВП СУ

Рис. 3.29 – Временная диаграмма работы ВП СУ в однопроцессорном режиме без потери цикла управления

В этом случае

$$T_{\text{реш}} = \sum_{i=1}^L (T_{\text{реш}}^i + T_{\text{тест}} + T_c) = \sum_{j=1}^{M_{\text{оп}}} \frac{D_j}{S_{\text{ПУ}}} + L \cdot (T_{\text{тест}} + T_c),$$

где L – число контрольных точек, $M_{\text{оп}} = |Q_{\text{оп}}|$;

$$T_{\text{ц}} = \sum_{i=1}^L (T_{\text{реш}}^i + T_{\text{тест}} + T_c) = T_{\text{реш}} + L(T_{\text{тест}} + T_c),$$

$$T_{\text{ввп}} = T_{\text{рек}}, \quad (3.7)$$

$$T_{\text{п}} = T_{\text{реш}}^i + T_{\text{тест}} + T_c + T_{\text{рек}}.$$

3.3.2.2. Диспетчирование при параллельном способе организации вычислительного процесса

При использовании рассмотренного выше способа решения задачи управления на одном ПУ время $T_{\text{ц}}$ цикла управления может оказаться недопустимо велико вследствие большого значения времени $T_{\text{реш}}$ (см. формулы (3.5) и (3.7)). Очевидно, что время решения $T_{\text{реш}}$ можно уменьшить путем распараллеливания решения задачи управления $G(Q, X)$ на нескольких ПУ, входящих в состав ВП СУ. Как показано в подразделе 3.1, такое распараллеливание можно осуществить двумя способами: параллельным и конвейерным.

При параллельном способе организации вычислительного процесса в СУ ВС граф $G(Q, X)$ задачи управления разрезается на H продольных подграфов, реализация каждого из которых поручается одному из основных ПУ, входящих в состав ВП СУ (рисунок 3.4). Временная диаграмма процесса решения задачи управления и его восстановления в случае возникновения отказа при использовании параллельного способа решения аналогична той, что показана на рисунке 3.28. Разница заключается лишь в том, что в данном случае

$$T_{\text{реш}} = \frac{\sum_{i=1}^{M_{\text{оп}}} D_i}{H \cdot S_{\text{ПУ}}},$$

где H – число ПУ, на которых распараллеливается решение задачи $G(Q, X)$.

По завершении очередного решения задачи управления на ПУ их локальные диспетчеры осуществляют процедуру тестирования своих ПУ и передают сообщения об их работоспособности всем остальным ЛД_у ($i = 1, N$). Если процедура тестирования не выявила отказавших ПУ, то процесс решения задачи управления повторяется заново. В противном случае локальные диспетчеры ПУ запускают процедуру восстановления вычислительного процесса, которая включает процедуру реконфигурации и запуска решения подзадач на новых ПУ. После этого цикл управления восстанавливается и повторяется периодически до момента обнаружения очередного отказа.

При этом

$$T_{\text{ц}} = T_{\text{реш}} + T_{\text{тест}} = \frac{\sum_{i=1}^{M_{\text{оп}}} D_i}{H \cdot S_{\text{ПУ}}} + T_{\text{тест}},$$

$$T_{\text{ввп}} = T_{\text{рек}}, \quad (3.8)$$

$$T_{\text{п}} = T_{\text{реш}} + T_{\text{тест}} + T_{\text{рек}}.$$

Выше рассмотрен вариант восстановления вычислительного процесса в СУ ВС с потерей одного цикла управления. Для того чтобы избежать потери цикла, время $T_{\text{реш}}$ нужно разбить на контрольные «точки», в которых должны осуществляться тестирование и сохранение промежуточных данных. В отличие от рассмотренного выше однопроцессорного способа организации диспетчера ВП здесь каждый основной ПУ, задействованный в решении за-

дачи, должен сообщать полученные промежуточные данные всем остальным ПУ, которые эти данные должны сохранять в своей памяти. Это обеспечивает возможность любому из ПУ «подхватить» выполнение подграфа $G(Q, X)$ в случае выхода из строя ПУ, ответственного за его решение.

Временная диаграмма решения задачи управления в ВП СУ и восстановления вычислительного процесса без потери цикла управления при параллельном способе вычислений аналогична показанной на рисунке 3.28. Отличие опять-таки будет заключаться только в том, что

$$T_{\text{реш}} = \sum_{i=1}^L (T_{\text{реш}}^i + T_{\text{тест}} + T_{\text{с}}) = \sum_{j=1}^{M_{\text{оп}}} \frac{D_j}{H \cdot S_{\text{ПУ}}} + L \cdot (T_{\text{тест}} + T_{\text{с}}),$$

где L – число точек контроля, $M_{\text{оп}} = |Q_{\text{оп}}|$.

$$T_{\text{ц}} = \sum_{j=1}^L (T_{\text{реш}}^L + T_{\text{тест}} + T_{\text{с}}) = T_{\text{реш}} + L(T_{\text{тест}} + T_{\text{с}}),$$

$$T_{\text{ввп}} = T_{\text{рек}},$$

$$T_{\text{п}} = T_{\text{реш}} + T_{\text{тест}} + T_{\text{рек}}.$$

3.3.2.3. Диспетчирование при конвейерном способе организации вычислений

При конвейерном способе организации вычислений граф $G(Q, X)$ разрезается на поперечные подграфы $G(Q, X)$ ($j = 1, 2, \dots, H$) (рисунок 3.6), реализация каждого из которых поручается отдельному ПУ. При этом с помощью коммутационной сети ПУ соединяются в конвейерную цепочку, каждая ступень которой (ПУ) получает данные от предыдущей ступени, обрабатывает их согласно графу $G_j(Q_j, X_j) \subseteq G(Q, X)$ и передает результаты на следующую ступень конвейера. При этом время $T_{\text{реш}}$ задачи управления будет таким же, как и при однопроцессорной обработке, т.е.

$$T_{\text{реш}} = \frac{\sum_{i=1}^{M_{\text{оп}}} D_i}{S_{\text{ПУ}}},$$

но время $T_{\text{ц}}$ цикла управления будет равно

$$T_{\text{ц}} = T_{\text{ку}} + T_{\text{тест}},$$

где

$$T_{\text{ку}} = \max_{j=1, H} \left(\frac{\sum_{i=1}^{M_j} D_i^j}{S_{\text{ПУ}}} \right) \approx \frac{\sum_{i=1}^{M_{\text{оп}}} D_i}{H \cdot S_{\text{ПУ}}},$$

H – число основных ПУ, задействованных в решении задачи; D_i^j – вычислительная сложность i -й вершины j -го подграфа $G_j(Q_j, X_j) \subseteq G(Q, X)$;

$M_j = |Q_j|$, т.е. время цикла управления в данном случае будет таким же, как и при параллельном способе организации вычислительного процесса в ВП СУ ВС (см. формулу (3.8)).

Рассмотрим принципы восстановления вычислительного процесса в такой ВП СУ в случае возникновения отказа того или иного ПУ.

Простейший вариант восстановления вычислительного процесса при конвейерной организации вычислений заключается в следующем: при обнаружении отказа необходимо провести реконфигурацию системы, т.е. перераспределить подграфы $G_j(Q_j, X_j)$ ($j = 1, H$) на работоспособные ПУ, после чего запустить цикл управления заново. Временная диаграмма работы ВП СУ, отвечающая такому подходу, показана на рисунке 3.30.

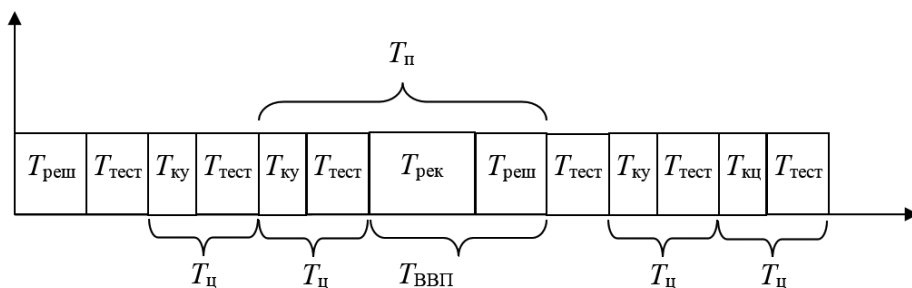


Рис. 3.30. Временная диаграмма работы ВП СУ с конвейерной организацией при восстановлении вычислительного процесса

В момент запуска системы в течение времени $T_{\text{реш}}$ происходит заполнение конвейера. После заполнения конвейера и формирования первого вектора команд управления локальные диспетчеры ЛД i ($i = 1, N$) осуществляют процедуру тестирования «своих» ПУ и оповещения всех остальных ПУ о их работоспособности.

Следующий вектор команд управления формируется через время $T_{\text{ку}}$, после чего вновь осуществляется процедура тестирования. Далее цикл управления осуществляется с периодом

$$T_{\text{ц}} = T_{\text{ку}} + T_{\text{тест}}$$

до того момента, пока при очередной процедуре тестирования не обнаруживается отказ одного из основных ПУ, задействованных в решении задачи.

После этого локальные диспетчеры всех ПУ переходят в режим восстановления рабочей конфигурации ВП СУ (реконфигурации) и перезапуска программы решения задачи управления на новой конфигурации ВП СУ. Далее вновь осуществляется процесс заполнения конвейера, после чего цикл управления восстанавливается. При такой организации

$$\begin{aligned} T_{\text{ц}} &= T_{\text{ку}} + T_{\text{тест}}, \\ T_{\text{ВВП}} &= T_{\text{рек}} + T_{\text{реш}}, \\ T_{\text{П}} &= T_{\text{ку}} + T_{\text{тест}} + T_{\text{рек}} + T_{\text{реш}} \end{aligned}$$

где $T_{\text{реш}} = \frac{\sum_{i=1}^{M_{\text{оп}}} D_i}{S_{\text{ПУ}}}$; $T_{\text{ку}} = \frac{\sum_{i=1}^{M_{\text{оп}}} D_i}{H \cdot S_{\text{ПУ}}}$, т.е. время восстановления вычислительного

процесса и время потери управления оказываются достаточно велики.

Кроме того, рассмотренный выше способ приводит к потере одного цикла управления, что не всегда допустимо.

Поэтому можно предложить иной подход к организации процесса восстановления вычислительного процесса в ВП СУ ВС конвейерного типа, лишенный указанных недостатков. Суть этого подхода заключается в следующем. При отказе какого-либо основного ПУ⁰_р, задействованного в решении задачи управления, теряются данные, полученные в результате решения на данном ПУ подграфа $G_j(Q_j, X_j) \subseteq G(Q, X)$, в то время как данные, полученные в результате решения всех остальных подграфов $G_1(Q_1, X_1), G_2(Q_2, X_2), \dots, G_{j-1}(Q_{j-1}, X_{j-1}), G_{j+1}(Q_{j+1}, X_{j+1}), \dots, G_H(Q_H, X_H)$ на всех остальных ПУ, годны для дальнейшего решения задачи. Поэтому если просто заменить отказавший ПУ⁰_р на резервный или передать его функции на другой основной ПУ, обладающий резервом производительности, и повторить на этом ПУ заново расчет подзадачи $G_j(Q_j, X_j)$ с входными данными, полученными в результате решения на ПУ⁰_{р-1} подграфа $G_{j-1}(Q_{j-1}, X_{j-1})$ в предыдущем цикле, то конвейерный вычислительный процесс будет восстановлен без потери цикла управления и без необходимости осуществления изначального заполнения всего конвейера.

Временная диаграмма работы ВП СУ, соответствующая такому процессу, показана на рисунке 3.31.

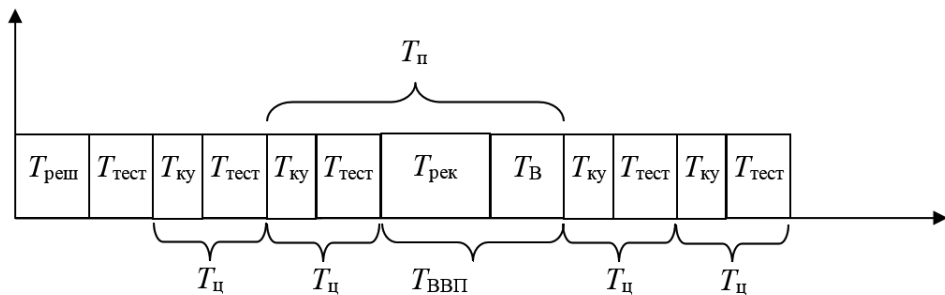


Рис. 3.31. Временная диаграмма восстановления вычислительного процесса в конвейерной ВП СУ ВС без потери цикла управления

При такой организации процедуры восстановления вычислительного процесса

$$\begin{aligned} T_{\text{ц}} &= T_{\text{ку}} + T_{\text{в}}, \\ T_{\text{ВВП}} &= T_{\text{рек}} + T_{\text{в}}, \\ T_{\text{П}} &= T_{\text{ку}} + T_{\text{тест}} + T_{\text{рек}} + T_{\text{в}}, \end{aligned}$$

где $T_{\text{в}} \approx T_{\text{ку}}$ – время восстановления данных, потерянных в результате отказа ПУ.

В таблице 3.1 сведены величины $T_{\text{ц}}$, $T_{\text{ввп}}$, и $T_{\text{п}}$ для различных рассмотренных выше способов восстановления вычислительного процесса в ВП СУ ВС. В данной таблице

$$T_{\text{реш}}^0 = \frac{\sum_{i=1}^{M_{\text{оп}}} D_i}{S_{\text{ПУ}}}, \quad T_{\text{реш}}^1 = \frac{\sum_{i=1}^{M_{\text{оп}}} D_i}{H \cdot S_{\text{ПУ}}},$$

$$T_{\text{реш}}^{0j} = \frac{\sum_{i=1}^{M_{\text{оп}}} D_i}{L \cdot S_{\text{ПУ}}}, \quad T_{\text{реш}}^{1j} = \frac{\sum_{i=1}^{M_{\text{оп}}} D_i}{L \cdot H \cdot S_{\text{ПУ}}},$$

где H – число основных ПУ, задействованных в решении задачи $G(Q, X)$; L – число «точек контроля» в цикле управления.

Табл. 3.1. Величины $T_{\text{ц}}$, $T_{\text{ввп}}$, и $T_{\text{п}}$ для различных способов восстановления вычислительного процесса

Способ восстановления выч. процесса	Временные параметры	Способ организации вычислительного процесса		
		однопроцессорный	параллельный	конвейерный
С потерей цикла управления	$T_{\text{ц}}$	$T_{\text{реш}}^0 + T_{\text{тест}}$	$T_{\text{реш}}^1 + T_{\text{тест}}$	$T_{\text{реш}}^1 + T_{\text{тест}}$
	$T_{\text{ввп}}$	$T_{\text{рек}}$	$T_{\text{рек}}$	$T_{\text{рек}} + T_{\text{реш}}^0$
	$T_{\text{п}}$	$T_{\text{реш}}^0 + T_{\text{тест}} + T_{\text{рек}}$	$T_{\text{реш}} + T_{\text{тест}} + T_{\text{рек}}$	$T_{\text{реш}}^1 + T_{\text{тест}} + T_{\text{рек}} + T_{\text{реш}}^0$
Без потери цикла управления	$T_{\text{ц}}$	$T_{\text{реш}}^0 + L(T_{\text{тест}} + T_{\text{с}})$	$T_{\text{реш}}^1 + L(T_{\text{тест}} + T_{\text{с}})$	$T_{\text{реш}}^1 + T_{\text{тест}}$
	$T_{\text{ввп}}$	$T_{\text{рек}}$	$T_{\text{рек}}$	$T_{\text{рек}} + T_{\text{реш}}^1$
	$T_{\text{п}}$	$T_{\text{реш}}^{0j} + T_{\text{тест}} + T_{\text{с}} + T_{\text{рек}}$	$T_{\text{реш}}^{1j} + T_{\text{тест}} + T_{\text{рек}}$	$T_{\text{реш}}^1 + T_{\text{тест}} + T_{\text{рек}} + T_{\text{реш}}^1$

С помощью таблицы 3.1 можно осуществить выбор того или иного способа организации вычислительного процесса и его восстановления в случае возникновения отказа в зависимости от заданных значений времени $T_{\text{ц}}$ цикла управления, времени $T_{\text{ввп}}$ восстановления вычислительного процесса и времени $T_{\text{п}}$ потери управления.

3.3.3. Алгоритмы работы локального диспетчера

Анализируя приведенные в предыдущем подразделе временные диаграммы работы ВП СУ с распределенным диспетчером, можно выделить

следующие основные режимы функционирования локального диспетчера ЛД_{*j*} ($j = 1, 2, \dots, N$).

1. Режим тестирования.
2. Режим реконфигурации.
3. Режим решения задачи управления.

В режиме тестирования ЛД_{*j*} должен выполнять следующие функции.

1. Проводить диагностирование работоспособности «своего» ПУ.
2. Пересылать диагностическое сообщение о работоспособности «своего» ПУ всем остальным ПУ, входящим в состав ВП СУ.

3. Принимать сообщения от локальных диспетчеров других ПУ, подтверждающие их работоспособность.

4. В случае если хотя бы один из основных ПУ, задействованных в решении задачи, не подтвердил свою работоспособность, переходить в режим реконфигурации.

5. Если же все основные ПУ подтвердили свою работоспособность, то переходить в режим решения задачи управления.

В режиме реконфигурации ЛД_{*j*} ($j = 1, 2, \dots, N$) должен выполнять следующие основные функции.

1. Принимать участие в процедуре перераспределения (размещения) операционных вершин графа $G(Q, X)$ по работоспособным ПУ.

2. По завершении процедуры перераспределения (размещения) переходить в режим решения задачи управления.

В режиме решения задачи управления ЛД_{*j*} ($j = 1, 2, \dots, N$) должен выполнять следующие функции:

1. Получать от других ПУ или информационных подсистем исходные данные, необходимые для решения множества подзадач $Q_i \subseteq Q_{\text{оп}}$, закрепленных за ПУ_{*i*}.

2. Решать подзадачи множества $Q_i \subseteq Q_{\text{оп}}$.

3. Передавать данные, полученные в результате решения подзадач множества Q_i , на другие ПУ или на исполнительные устройства в соответствии с топологией связей вершин графа $G(Q, X)$.

4. Переходить в режим тестирования.

Рассмотрим алгоритмы работы локального диспетчера в перечисленных режимах более подробно. При этом будем считать, что ЛД_{*j*} ($j = 1, 2, \dots, N$) имеет в своем распоряжении:

1. Программы решения всех подзадач $q_i \in Q_{\text{оп}}$ графа $G(Q, X)$.
2. Программу диагностирования работоспособности ПУ_{*i*}.
3. Программу реализации алгоритма перераспределения (размещения) подзадач графа $G(Q, X)$ по работоспособности ПУ.

4. Таблицу T_1 размерами $M \times M$, где $M = |Q|$, содержащую информацию об инцидентности вершин графа $G(Q, X)$ или, иными словами, информацию о топологии графа $G(Q, X)$. При этом если вершины q_i и q_j графа $G(Q, X)$ соединены дугой $x(q_i, q_j) \subseteq X$, то на пересечении i -й строки и j -го столбца таблицы T_1 стоит единица, а в противном случае нуль.

5. Таблицу T_2 размерами $N \times M_{\text{оп}}$, где $M_{\text{оп}}$ – число операционных вершин в графе $G(Q, X)$. Каждая строка таблицы T_2 соответствует ПУ, входящему в состав ВП СУ, а столбец – операционной вершине q графа $G(Q, X)$. Будем считать, что если на пересечении i -й строки и j -го столбца таблицы T_2 стоит единица, то это означает, что подзадача q_j закреплена за ПУ _{i} , а если нуль, то не закреплена за данным ПУ.

6. Таблицу T_3 размерами $N \times 1$. Будем считать, что если i -й элемент этой таблицы равен 1, то ПУ _{i} работоспособен, а если нуль, то неработоспособен.

7. Список O номеров основных ПУ, задействованных в текущий момент времени в решении задачи управления $G(Q, X)$.

8. Список R резервных ПУ, не задействованных в текущий момент времени в решении задачи управления.

9. Список L номеров отказавших ПУ.

Отметим, что таблица T_1 определяется графом $G(Q, X)$ решаемой в ВП задачи управления, поэтому заполняется до момента начала ее решения и не изменяется в процессе решения.

Таблица T_2 также изначально заполняется до начала функционирования ВП СУ на основе результатов начального распределения (размещения) вершин графа $G(Q, X)$ по основным ПУ.

Все элементы таблицы T_3 до начала функционирования ВП СУ равны единице, т.е. считаем, что изначально все ПУ работоспособны.

Отметим, что в отличие от таблицы T_1 , которая не изменяется в процессе работы ВП СУ, таблицы T_2 и T_3 могут динамически изменяться в процессе функционирования ВП СУ в случае выхода из строя ряда ПУ и перераспределения подзадач графа $G(Q, X)$ по работоспособным ПУ.

Рассмотрим алгоритмы работы локального диспетчера ЛД _{i} ($i = \overline{1, N}$) в различных режимах.

Прежде чем перейти к описанию алгоритма функционирования ЛД _{i} ($i = \overline{1, N}$) в режиме тестирования, обоснуем ряд соображений, лежащих в его основе. Будем считать, что в случае штатного выполнения программы тестирования ПУ _{i} его локальный диспетчер ЛД _{i} формирует сообщение о работоспособности ПУ _{j} , которое передается всем остальным ЛД _{j} ($j = \overline{1, N}; j \neq i$). Если же программа тестирования ПУ _{i} выполнена не штатно либо ПУ _{i} полностью вышел из строя, то сообщение о его работоспособности вообще не формируется и, соответственно, не передается другим ЛД либо формируется и передается в недопустимом виде. Поэтому если в течение определенного промежутка времени, выделяемого на тестирование отдельных ПУ, от ЛД _{j} не поступило никакого сообщения (либо сообщение пришло в нестандартном виде), то это означает, что произошел отказ ПУ _{j} .

Исходя из этих соображений укрупненный алгоритм работы ЛД _{i} ($i = \overline{1, N}$) в режиме тестирования можно представить в следующем виде.

Алгоритм 3.1.

1. Обнуление списка L .
2. Запуск программы диагностирования работоспособности ПУ _{i} .
3. В случае если программа диагностирования выполнена на ПУ _{i} штатно, то передача сообщения о работоспособности ПУ _{i} всем остальным ЛД _{j} ($j = \overline{1, N}; j \neq i$). В противном случае сообщение не передается.
4. Прием сообщений о работоспособности ПУ _{j} от ЛД _{j} ($j = \overline{1, N}; j \neq i$).
5. Запись в список L номеров ПУ, не приславших сообщение о своей работоспособности в течение времени тестирования либо приславших сообщения нестандартного вида.
6. Если список L пустой или не изменился (а это означает, что за предыдущий цикл работы ВП СУ не появилось новых отказавших ПУ), то перейти к п. 8, иначе к п. 7.
7. Переход в режим реконфигурации ВП СУ.
8. Переход в режим решения задачи управления.

Далее рассмотрим алгоритм работы ЛД _{i} ($i = \overline{1, N}$) в режиме решения задачи управления. Также, прежде чем перейти непосредственно к описанию данного алгоритма, сформулируем ряд соображений, лежащих в его основе.

Номера подзадач множества $Q_i \subseteq Q_{\text{оп}}$, закрепленных за ПУ _{i} , определяются ненулевыми элементами i -й строки таблицы T_2 . Поэтому, для того чтобы решить очередную подзадачу q_j , закрепленную за ПУ _{i} , его локальный диспетчер ЛД _{i} должен сначала найти очередной ненулевой элемент в i -й строки таблицы T_2 , затем по номеру j -го столбца элемента определить номер подзадачи q_j , и в случае если исходные данные, необходимые для ее решения, уже имеются (поступили от других ПУ либо от информационных модулей), то запустить программу ее решения. Результаты решения подзадачи $q_i \in Q_{\text{оп}}$ должны быть переданы тем ПУ, которые решают смежные с ней подзадачи (т.е. те подзадачи, которые инцидентны вершине q_j). Номера этих ПУ могут быть определены с помощью таблиц T_1 и T_2 . Для этого ЛД _{i} должен считать j -ю строку таблицы T_1 и определить номера всех вершин графа $G(Q, X)$, которые инцидентны вершине q_j . Далее по номеру l вершины q_l , инцидентной вершине q_j , необходимо считать столбец таблицы T_2 и найти в нем ненулевой элемент. Номер строки этого элемента и будет определять номер ПУ, за которым закреплена подзадачи q_l , смежная подзадаче q_j .

Исходя из этих соображений, укрупненный алгоритм работы локального диспетчера ЛД _{i} ($i = \overline{1, N}$) в режиме решения задачи управления можно представить в следующем виде.

Алгоритм 3.2.

1. Перезапись i -й строки таблицы T_2 в вектор $V = \langle v_j; j = \overline{1, M_{\text{оп}}} \rangle$.
2. Если $V = \emptyset$ (т.е. все элементы вектора V нулевые), то переход к п. 12, иначе к п. 3.
3. $j = 1$.

4. Если $v_j = 0$ (что означает, что подзадача q_j не закреплена за ПУ_{*j*}), то переход к п. 6, иначе к п. 5.

5. Если исходные данные для решения подзадачи q_j готовы (поступили от других ПУ), то переход к п. 7, иначе к п. 6.

6. $j = j + 1$, если $j \leq M_{\text{оп}}$, то переход к п. 3, иначе к п. 2.

7. Запуск программы решения подзадачи q_j .

8. Если решение подзадачи q_j завершено, то в таблице T_1 считывается j -я строка и определяются номера подмножества вершин Q_j , инцидентных вершине q_j .

9. По номерам вершин, входящих в подмножество Q_j , с помощью таблицы T_2 определяются номера ПУ, за которыми данные вершины (подзадачи) закреплены. Для этого для каждого $q_i \in Q_j$ в таблице T_2 считывается i -й столбец и в нем находится элемент, равный единице. Номер строки этого элемента определяет номер ПУ, за которым закреплена подзадача q_j .

10. Пересылка результатов решения подзадачи q_j всем ПУ, за которыми закреплены подзадачи подмножества Q_j .

11. $v_j = 0$, переход к п. 2.

12. Переход в режим тестирования.

Наибольшую сложность представляет работа ЛД_{*i*} ($i = \overline{1, N}$) в режиме реконфигурации при обнаружении отказавших ПУ. Алгоритмы функционирования ЛД_{*i*} ($i = 1, 2, \dots, N$) в этом режиме зависят от ряда факторов. Поэтому подробному рассмотрению методов и алгоритмов реконфигурации ВП СУ будет посвящен следующий раздел.

3.3.4. Резюме

В результате проведенных исследований показано, что для организации отказоустойчивого функционирования ВП СУ с использованием способа резервирования производительности ПУ и реконфигурации при возникновении отказов необходимо обеспечить выполнение процедуры перераспределения подзадач задачи управления на исправных ПУ. Данная процедура не тривиальна и требует применения специального ПУ, выполняющего роль диспетчера, что ухудшает не только массогабаритные, но и надежность характеристики ВП СУ.

Предложен способ распределенной организации диспетчера с помощью множества ЛД, физически реализуемых на отдельных ПУ ВП СУ и совместно участвующих в процессе диспетчирования ресурсов ВП на принципах мультиагентного взаимодействия. Данный подход лишен перечисленных выше недостатков, поскольку не требует ввода в состав ВП СУ дополнительных устройств.

Проведен анализ особенностей функционирования ВП с распределенным диспетчером при различных способах организации вычислительного процесса в СУ ВС и получены оценочные выражения, определяющие временные затраты на реализацию основных режимов диспетчирования (те-

стирование, цикл управления, восстановление вычислительного процесса). Для каждого из режимов разработан алгоритм работы ЛД, реализуемый на отдельном ПУ.

3.4. Методы и алгоритмы реконфигурации вычислительных подсистем

3.4.1. Постановка задачи реконфигурации вычислительной подсистемы

Как показано в предыдущем подразделе, наиболее сложным этапом работы ВП СУ при возникновении отказа является этап реконфигурации, т.е. перераспределения подзадач графа $G(Q, X)$ по работоспособности ПУ. От времени выполнения процедуры реконфигурации зависит время $T_{\text{ввп}}$ восстановления вычислительного процесса в ВП СУ ВС. Поскольку, как правило, на время $T_{\text{ввп}}$ наложены ограничения, то время выполнения процедуры реконфигурации также должно быть ограничено, т.е. $T_{\text{рек}} \leq T_{\text{ввп}}^3$, где $T_{\text{ввп}}^3$ – заданное время восстановления вычислительного процесса в ВП СУ ВС.

В общем виде задачу реконфигурации ВП СУ можно представить в следующем виде. Допустим, что в состав ВП СУ входит H основных ПУ, задействованных в решении задачи $G(Q, X)$, и B резервных ПУ. Будем считать, что задача $G(Q, X)$ каким-то образом была изначально распараллелена по основным ПУ, причем ПУ $_i^0$ ($i = 1, 2, \dots, H$) было «поручено» решение множества подзадач $Q_{\text{оп}}^i \subseteq Q_{\text{оп}}$. Пусть произошел отказ L основных ПУ, задействованных в решении задачи управления $G(Q, X)$. Необходимо таким образом перераспределить (разместить) подзадачи множества $Q_{\text{оп}} \subseteq Q$ по работоспособным ПУ (основным и резервным), чтобы время $T_{\text{ц}}$ цикла управления не превышало заданной величины $T_{\text{ц}}^3$, а вычислительная загрузка всех ПУ, задействованных в решении задачи управления, была приблизительно одинаковая, т.е.

$$D_i^{\text{пу}} \approx D_j^{\text{пу}} \approx \frac{\sum_{l=1}^{M_{\text{оп}}} D_l}{H} \quad (l = 1, \dots, H, j = 1, \dots, H, i \neq j),$$

где $D_i^{\text{пу}}$ – вычислительная нагрузка i -го ПУ ($i = 1, \dots, N$);

D_l – вычислительная сложность подзадачи $q_l \in Q_{\text{оп}}$ графа $G(Q, X)$;

M_0 – число операционных вершин в графе $G(Q, X)$;

H – число ПУ, задействованных в решении задачи.

Результатом такого перераспределения (размещения) должна стать модифицированная таблица T_2 , содержащая информацию о новом закреплении вершин графа $G(Q, X)$ за ПУ, входящими в состав ВП СУ, которая будет использована при последующей реализации процедуры решения задачи управления с использованием ВП СУ.

Следует отметить, что проблеме оптимального распределения (размещения) вершин графа задачи по процессорам многопроцессорной вычислительной системы посвящено достаточно большое число исследований [4, 12, 23, 50, 68]. Однако большинство этих исследований ориентировано на реализацию разрабатываемых методов и алгоритмов с помощью центрального (специально выделенного) диспетчера. Нас же будут интересовать методы и алгоритмы распределения, реализуемые с помощью распределенного диспетчера, т.е. множества локальных диспетчеров ЛД_{*j*} ($j = 1, 2, \dots, H - L$), реализованных на отдельных ПУ.

Алгоритмы работы локальных диспетчеров при решении задачи реконфигурации ВП СУ будут в значительной степени зависеть как от соотношения числа H основных, B резервных и L отказавших ПУ [44], так и от способа организации вычислительного процесса в ВП СУ ВС.

Рассмотрим принципы организации процедуры реконфигурации при различных соотношениях H , B и L .

Сначала рассмотрим простейший случай, когда $B \geq L$, т.е. число вышедших из строя (отказавших) основных ПУ, меньше числа резервных ПУ. Для простоты рассуждений допустим, что $L = 1$, т.е. вышел из строя один основной ПУ, например ПУ⁰_{*i*}.

Будем считать, что предыдущее распределение (размещение) задачи $G(Q, X)$ по основным ПУ (т.е. размещение, имевшее место до момента обнаружения отказа и отраженное в таблице T_2) удовлетворяло заданным требованиям по быстродействию, т.е. $T_{ii} \leq T_{ii}^3$, а также требованию по равномерности вычислительной нагрузки ПУ. В этом случае процедуру распределения не имеет смысла повторять заново, а достаточно просто «перепоручить» множество подзадач $Q_{оп}^i$, закрепленных за отказавшим ПУ⁰_{*i*}, одному из резервных ПУ, например резервному ПУ^p_{*r*} с наименьшим номером.

Учитывая, что локальный диспетчер каждого ПУ, входящего в состав ВП СУ, обладает таблицей T_2 , содержащей информацию о закреплении вершин графа $G(Q, X)$ за ПУ, а также списками O и R , содержащими номера основных и резервных ПУ, алгоритм работы локального диспетчера ЛД_{*j*} при реализации процедуры реконфигурации в данном случае можно записать в следующем виде.

Алгоритм 3.3.

1. Определение номера отказавшего ПУ⁰_{*i*} путем анализа сообщений о работоспособности, поступивших в процессе выполнения процедуры тестирования. Исключение номера ПУ⁰_{*i*} из списка O .

2. Определение резервного ПУ^p_{*r*} с наименьшим номером путем анализа списка R .

3. Удаление номера ПУ^p_{*r*} из списка R и включение его в список O .

4. Модификация таблицы T_2 : перезапись (копирование) j -й строки в r -ю строку таблицы T_2 . Обнуление j -й строки таблицы T_2 .

5. Переход в режим решения задачи управления $\hat{G}(Q, X)$.

В результате выполнения данного алгоритма ПУ^p , переходит в список основных ПУ и на него возлагаются подзадачи множества $\mathcal{Q}_{\text{оп}}^i$, ранее закрепленные за ПУ_i^0 согласно таблице T_2 .

Рассмотрим теперь случай, когда $L > 1$, т.е. одновременно вышли из строя (отказали) несколько основных ПУ. Если при этом $B \geq L$, т.е. число резервных ПУ, не меньше числа отказавших ПУ, то процедуру реконфигурации можно осуществить с помощью описанного выше алгоритма работы локального диспетчера ЛД, с той лишь разницей, что процедуру замещения отказавшего ПУ на резервный необходимо повторить L раз.

Сложнее дело обстоит в случае, когда $L > B$, т.е. число отказавших ПУ, больше числа резервных ПУ. В этом случае простую замену отказавших ПУ на резервные уже осуществить нельзя, а необходимо проводить процедуру полного (или частичного) перераспределения (размещения) подзадачи задачи $G(\mathcal{Q}, X)$ по работоспособности ПУ для того, чтобы обеспечить заданное время $T_{\text{ц}}^3$ цикла управления, а также равномерность вычислительной загрузки ПУ.

Поскольку выше мы приняли, что все ПУ одинаковые, то в простейшем случае можно заранее (до момента начала решения задачи управления в ВП СУ) определить варианты размещения вершин множества $\mathcal{Q}_{\text{оп}} \subseteq \mathcal{Q}$ графа $G(\mathcal{Q}, X)$ по ПУ для всех возможных конфигураций ВП, содержащих соответственно $H - 1, H - 2, \dots, H - L$ ПУ. Эти варианты должны быть сохранены в памяти всех ПУ в виде множества таблиц T_2 . Тогда, в случае возникновения отказа L ПУ, работоспособные ПУ смогут извлечь из своей памяти ту таблицу T_2 , которая соответствует данной конфигурации, и принять ее в качестве рабочей, после чего система будет готова к решению задачи управления $G(\mathcal{Q}, X)$ на данной (текущей) конфигурации.

Однако такой подход возможен только в случае, когда, во-первых, число ПУ в системе относительно невелико, а во-вторых, когда граф $G(\mathcal{Q}, X)$ алгоритма решения задачи управления не изменяется в течение всего периода работы ВП СУ. Если же граф $G(\mathcal{Q}, X)$ каким-то образом может изменяться в зависимости от внешних условий (например, режимов функционирования объекта управления или его текущего состояния), то описанный выше подход становится практически неприменим. В этом случае ВП СУ должна обладать возможностями самостоятельного (автоматического) решения задачи размещения операционных вершин графа $G(\mathcal{Q}, X)$ по работоспособным ПУ. Алгоритмы такого размещения в значительной степени будут зависеть от способа организации вычислительного процесса в ВП СУ ВС. Рассмотрим алгоритмы работы локальных диспетчеров ЛД_{*j*} ($j = 1, 2, \dots, N$) при реализации процедуры размещения вершин графа $G(\mathcal{Q}, X)$ по $N = H + B - L$ работоспособным ПУ при различных способах организации вычислительного процесса в ВП СУ ВС. При этом будем считать, что вычислительная сложность подзадачи, приписанной вершине $q_i \in \mathcal{Q}$, равна D_i , а объем передаваемых данных, соответствующих дуге $x(q_i, q_j)$, равен W_{ij} .

3.4.2. Формализация задачи размещения подзадач по процессорным узлам

Прежде чем приступить к рассмотрению различных алгоритмов размещения, необходимо дать формальную постановку задачи размещения вершин графа $G(Q, X)$ по работоспособным ПУ.

Как показано выше, с целью распараллеливания вычислительного процесса в ВП СУ ВС граф $G(Q, X)$ разрезается на отдельные подграфы $G_1(Q, X_1), \dots, G_N(Q_N, X_N)$, решение каждого из которых закрепляется за отдельным ПУ, т.е. ПУ_{*i*} (*i* = 1, 2, ..., *N*) поручается реализация подзадач, соответствующих вершинам подмножества $Q_i \subseteq Q$.

Иными словами, процедура разрезания графа $G(Q, X)$ на подграфы эквивалентна процедуре размещения подзадач, приписанных вершинам графа $G(Q, X)$ по ПУ, входящим в состав ВП СУ. При этом размещение будет допустимым, если время $T_{ц}$ цикла управления будет меньше заданного $T_{ц}^3$. При использовании параллельного способа организации вычислительного процесса $T_{ц} = T_{реш}$, где $T_{реш}$ – общее время решения задачи управления $G(Q, X)$. Поэтому в данном случае размещение вершин графа по ПУ можно считать допустимым, если $T_{реш} \leq T_{ц}^3$.

Предположим, что все операционные вершины графа $G(Q, X)$ каким-то образом были размещены по *N* ПУ. Тогда время $T_{реш}(q_i^k)$ решения подзадачи, соответствующей вершине $q_i^k \in Q_{оп}$ *k*-го яруса графа $G(Q, X)$ (т.е. абсолютное время, считая от момента начала решения задачи $G(Q, X)$), можно определить с помощью следующего выражения:

$$T_{реш}(q_i^k) = \max_{l=1,d} (T_{реш}(q_l^{k-1}) + T_{пл}(q_l^{k-1}, q_i^k)) + T(q_i^k), \quad (3.9)$$

где $T_{реш}(q_l^{k-1})$ – абсолютное время решения подзадачи, соответствующей вершине q_l^{k-1} , которая расположена на более низком ярусе графа $G(Q, X)$ и инцидентна вершине q_i^k ;

$$T(q_i^k) = \frac{D_i^k}{S_{ПУ}} - \text{время решения подзадачи на ПУ};$$

$T_{пл}(q_l^{k-1}, q_i^k)$ – время передачи данных, приписанных дуге $x(q_l^{k-1}, q_i^k)$, от подзадачи q_l^{k-1} к подзадаче q_i^k , причем

$$T_{пл}(q_l^{k-1}, q_i^k) = \begin{cases} \frac{W_{li}}{Y_{КС}}, & \text{если подзадачи } q_l^{k-1} \text{ и } q_i^k \text{ решаются на разных ПУ,} \\ 0, & \text{если подзадачи } q_l^{k-1} \text{ и } q_i^k \text{ решаются на одном ПУ;} \end{cases}$$

D – число вершин, инцидентных вершине q_j^k .

Последнее выражение означает, что абсолютное время решения подзадачи определяется максимальным абсолютным временем поступления исход-

ных данных от предшествующих подзадач (первое слагаемое) и собственно временем решения подзадачи на ПУ (второе слагаемое).

Тогда общее (абсолютное) время $T_{\text{реш}}$ решения всей задачи, представленной графом $G(Q, X)$, будет равно

$$T_{\text{реш}}(q_i^{K-1}) = \max_{l=1, M_{k-2}} (T_{\text{реш}}(q_l^{K-2}) + T_{\text{пл}}(q_l^{K-2}, q_i^{K-1})) + T(q_i^{K-1}), i = 1, 2, \dots, M_{k-1}, \quad (3.10)$$

где q_l^{K-1} ($l = 1, d_{k-1}$) – вершина (K-1)-го яруса графа $G(Q, X)$, причем

$$T_{\text{реш}}(q_i^{K-1}) = \max_{l=1, M_{k-2}} (T_{\text{реш}}(q_l^{K-2}) + T_{\text{пл}}(q_l^{K-2}, q_i^{K-1})) + T(q_i^{K-1}), i = 1, 2, \dots, M_{k-1}, \quad (3.11)$$

$$T_{\text{реш}}(q_i^{K-2}) = \max_{l=1, M_{k-3}} (T_{\text{реш}}(q_l^{K-3}) + T_{\text{пл}}(q_l^{K-3}, q_i^{K-2})) + T(q_i^{K-2}), i = 1, 2, \dots, M_{k-2},$$

.....

$$T_{\text{реш}}(q_i^j) = \max_{l=1, M_{j-1}} (T_{\text{реш}}(q_l^{j-1}) + T_{\text{пл}}(q_l^{j-1}, q_i^j)) + T(q_i^j), i = 1, 2, \dots, M_j,$$

.....

$$T_{\text{реш}}(q_i^2) = \max_{l=1, M_1} (T_{\text{реш}}(q_l^1) + T_{\text{пл}}(q_l^1, q_i^2)) + T(q_i^2), i = 1, 2, \dots, M_1,$$

$$T_{\text{реш}}(q_i^1) = \max_{l=1, M_0} (T_{\text{пл}}(q_0, q_i^1)) + T(q_i^1), i = 1, 2, \dots, M_0,$$

где $T_{\text{реш}}(q_i^j)$ – абсолютное время решения i -й подзадачи j -го яруса графа $G(Q, X)$;

$T_{\text{пл}}(q_0, q_i^1)$ – время передачи данных от информационных (сенсорных) устройств к i -й подзадаче яруса 1;

$T_{\text{пл}}(q_l^{j-1}, q_i^j)$ – время передачи данных от подзадачи $(j-1)$ -го яруса к подзадаче j -го яруса;

$T_{\text{пл}}(q_l^{K-1}, q_K)$ – время передачи команд управления от i -й подзадачи яруса K-1 к исполнительным устройствам;

M_j – число вершин (подзадач) на j -м ярусе графа $G(Q, X)$;

$K+1$ – число ярусов в графе $G(Q, X)$.

Как показано выше, для того чтобы ВП СУ могла выполнять возложенную на нее функцию (т.е. функцию решения задачи управления за отведенное время), получаемое размещение должно удовлетворять условию

$$T_{\text{реш}} < T_{\text{ц}}^3,$$

где $T_{\text{реш}}$ – время решения задачи управления, получаемое в результате размещения вершин графа $G(Q, X)$ по ПУ;

$T_{\text{ц}}^3$ – заданное время цикла управления.

В свою очередь, среди всех размещений, удовлетворяющих условию $T_{\text{реш}} < T_{\text{ц}}^3$, наилучшим следует считать то размещение, для которого выполняется условие

$$\Delta D = \frac{\sum_{j=1}^N \left| D_j^{\text{пв}} - \frac{\sum_{i=1}^{\text{Mon}} D_i}{N} \right|}{N} = \min, \quad (3.12)$$

где $D_i^{\text{пв}}$ – вычислительная нагрузка i -го ПУ ($i = 1, \dots, N$);

$M_{\text{оп}}$ – общее число операционных вершин в графе $G(Q, X)$, т.е. суммарное отклонение вычислительной нагрузки отдельных ПУ от средней будет минимально.

Будем считать, что размещение оптимально, если

$$T_{\text{реш}} = T_{\text{реш}}^{\min} \quad \text{и} \quad D_i^{\text{пв}} = D_p^{\text{пв}} = \frac{\sum_{i=1}^{\text{Mon}} D_i}{N} \quad (i = 1, \dots, N),$$

где $D_i^{\text{пв}}$ – вычислительная нагрузка i -го ПУ ($i = 1, \dots, N$);

$T_{\text{реш}}^{\min}$ – минимально возможное время решения задачи управления на имеющемся в текущий момент времени работоспособном ресурсе (конфигурации) ВП СУ;

$D_p^{\text{пв}}$ – величина нагрузки отдельного ПУ при равномерном распределении.

Под допустимым размещением будем понимать такое размещение, которое удовлетворяет условию

$$T_{\text{реш}} < T_{\text{ц}}^3 \quad \text{и} \quad \Delta D < \Delta D_{\text{доп}},$$

где $\Delta D_{\text{доп}}$ – допустимое отклонение вычислительной нагрузки ПУ от средней (в качестве $\Delta D_{\text{доп}}$ можно принять нагрузку, приводящую к изменению температуры на 5°C , т.к. при таком отклонении не наблюдается существенного влияния на надежность ВП СУ).

Под качеством решения задачи размещения будем понимать степень приближения полученного допустимого размещения к оптимальному.

С другой стороны, на время $T_{\text{раз}}$ решения самой задачи размещения также накладываются ограничения. Действительно, как показано выше, от времени $T_{\text{раз}}$ будет зависеть время $T_{\text{рек}}$ выполнения процедуры реконфигурации ВП СУ, от которого, в свою очередь, зависит время $T_{\text{ввп}}$ восстановления вычислительного процесса в СУ ВС. Поскольку время $T_{\text{ввп}}$, как правило, должно быть ограничено сверху, то, следовательно, и время $T_{\text{раз}}$ решения задачи размещения также должно быть ограничено, т.е.

$$T_{\text{раз}} < T_{\text{ввп}}^3$$

где $T_{\text{ввп}}^3$ – заданное время восстановления вычислительного процесса в СУ ВС.

Очевидно, что параметры времени $T_{\text{раз}}$ решения задачи размещения и качества этого решения имеют обратную зависимость, т.е. чем больше будет время решения, тем лучше будет его качество и наоборот.

Исходя из вышеизложенного, можно сделать вывод, что выбор того или иного алгоритма решения задачи размещения графа $G(Q, X)$ по работоспособным ПУ при реализации процедуры реконфигурации ВП СУ должен определяться двумя требованиями: с одной стороны, алгоритм должен обеспечивать допустимое размещение, при котором $T_{\text{реш}} < T_{\text{ц}}^3$ и

$$\Delta D = \frac{\sum_{j=1}^N \left| D_j^{\text{нв}} - \frac{\sum_{i=1}^{\text{Mon}} D_i}{N} \right|}{N} < \Delta D_{\text{доп}}, \quad (3.13)$$

где $\Delta D_{\text{доп}}$ – допустимое значение отклонения от средней нагрузки ПУ, а с другой стороны, быть реализован за время $T_{\text{раз}} < T_{\text{впп}}^3$.

3.4.3. Алгоритмы размещения подзадач при реконфигурации

Наиболее очевидный подход к решению задачи размещения вершин графа $G(Q, X)$ по N работоспособным ПУ – это проанализировать все возможные варианты размещения, для каждого варианта рассчитать значение $T_{\text{реш}}$ и ΔD согласно формулам (3.10), (3.11) и (3.12) и выбрать тот вариант, для которого $T_{\text{реш}} < T_{\text{ц}}^3$ и $\Delta D = \min$. Очевидно, что данный подход потенциально обеспечивает наивысшее качество решения задачи размещения, но, с другой стороны, при этом время $T_{\text{раз}}$ может оказаться недопустимым. Как это показано в [20], данное время в случае параллельной реализации полного перебора составит:

$$T_{\text{раз}} = \frac{M_{\text{оп}}^N}{N} M_{\text{оп}} T_{\text{в}} + N T_{\text{п}} = \frac{M_{\text{оп}}^{N+1}}{N} T_{\text{в}} + N T_{\text{п}},$$

где $T_{\text{в}}$ – время вычисления по формулам (3.10) и (3.11) на ПУ;

$T_{\text{ц}}$ – время пересылки ЛД сообщения о выбранном варианте размещения всем остальным ЛД.

Существенно сократить время $T_{\text{раз}}$ решения задачи размещения можно с помощью различного рода итерационных алгоритмов, позволяющих сократить количество перебираемых вариантов размещения за счет последовательного приближения к искомому допустимому решению, например, таких как алгоритм имитации отжига [57, 94] или так называемый венгерский алгоритм [78]. Рассмотрим некоторые примеры итерационных алгоритмов размещения подзадач (вершин) графа $G(Q, X)$ по работоспособным ПУ, отвечающего приведенным выше требованиям.

3.4.3.1. Алгоритм поярусного размещения

Одним из алгоритмов последовательной оптимизации является так называемый «жадный» алгоритм [41]. Несмотря на свою простоту, жадный

алгоритм может достаточно успешно применяться при решении различных оптимизационных задач, в том числе задач размещения.

Суть использования жадного алгоритма при решении задачи размещения заключается в последовательном назначении на ПУ подзадач (вершин) графа $G(Q, X)$ таким образом, чтобы абсолютное время их решения было минимально. При этом поскольку подзадачи, приписанные вершинам графа $G(Q, X)$, информационно связаны друг с другом, то такое размещение целесообразно выполнять поярусно, начиная с вершин первого яруса графа $G(Q, X)$ и т.д. вплоть до $-го$ яруса. Следует отметить, что при использовании такого подхода окончательное размещение подзадач задачи управления по ПУ и, соответственно, значения оптимизируемых параметров $T_{\text{реш}}$ и ΔD не могут быть оценены заранее, а могут быть рассчитаны только по окончании процедуры размещения. Вследствие этого процедуру размещения при использовании метода поярусного размещения имеет смысл разбить на два этапа:

1) этап получение размещения, обеспечивающего выполнение необходимых временных требований, т.е. $T_{\text{реш}} < T_{\text{ц}}^3$,

2) этап модификация полученного размещения с целью удовлетворения условию $\Delta D < \Delta D_{\text{доп}}$.

Рассмотрим принципы реализации этих двух этапов более подробно. При этом будем считать, что каждый локальный диспетчер ЛД_г процессорного узла ПУ_г ($g = 1, 2, \dots, N$) имеет в своем распоряжении K таблиц A_k ($k = 1, 2, \dots, K-1$) размерами $N \times M_k$, где $N = (H + B + L)$ – число работоспособных ПУ; K – число ярусов в графе $G(Q, X)$; M_k – число вершин на k -м ярусе, а также векторов P_k , содержащих M_k элементов каждый. Вначале все эти таблицы и векторы пустые, т.е. все их элементы равны нулю.

Процесс размещения вершин графа $G(Q, X)$ по N работоспособным ПУ разбивается на K шагов. На первом шаге осуществляется размещение вершин первого яруса графа $G(Q, X)$. Для этого каждый локальный диспетчер ЛД_г рассчитывает согласно формуле (3.9) значения

$$T_{\text{реш}}^j(q_i^1) = \max_{l=1, M_0}(T_{\text{но}}^0(q_l^0, q_i^1)) + T(q_i^1) \\ (i = 1, 2, \dots, M_1; j = 1, 2, \dots, N)$$

времени решения всех вершин первого яруса q_i^1 ($i = 1, 2, \dots, M_1$) в случае их реализации на ПУ_г ($g = 1, 2, \dots, N$).

Полученные таким образом значения $T_{\text{реш}}^j(q_i^1)$ ($i = 1, M_1; j = 1, N$) заносятся в таблицу A_1 в качестве элемента a_{ij} . Таким образом, происходит заполнение всей таблицы A_1 .

Далее ЛД_г анализирует f -ю строку своей таблицы A_1 и находит в ней минимальный элемент $a_{fl} = T_{\text{реш}}^f(q_l^1)$. Это означает, что абсолютное время решения задачи q_l^1 на ПУ_г будет наименьшим среди всех вершин первого яруса. Если таковых минимальных элементов в f -й строке таблицы A_1 несколько, то выбирается первый из них. После этого ЛД_г анализирует l -й

столбец таблицы A_1 . Если в этом столбце есть элемент $a_{jl} < a_{fl}$, то это означает, что в случае закрепления подзадачи q_l^1 за ПУ j абсолютное время ее решения будет меньше, чем в случае закрепления этой подзадачи за ПУ f .

Поэтому в этом случае ЛД_{*f*} «пропускает» значение $a_{fl} = T_{\text{реш}}^f(q_l^1)$ и ищет следующий по величине наименьший элемент $a_{fr} = T_{\text{реш}}^f(q_r^1)$ в f -й строке таблицы A_1 . После этого он анализирует r -й столбец таблицы A_1 , и если в этом столбце отсутствуют элементы меньше, чем a_{fr} , то вершина q_r^1 закрепляется для исполнения за ПУ f .

Информацию об этом закреплении ЛД_{*f*} сообщает всем остальным ЛД, на основании чего все они записывают номер f данного ПУ в r -й элемент вектора P_1 а также модифицируют все элементы f -й строки таблицы A_1 следующим образом:

$$a_{fl} = T_{\text{реш}}^f(q_l^1) + T_{\text{реш}}^f(q_r^1),$$

$$l = 1, 2, \dots, M_1, \quad l \neq r,$$

где q_r^1 – вершина, закрепленная за ПУ f .

Последнее означает, что все остальные вершины первого яруса в случае их реализации на ПУ f будут выполняться после подзадачи q_r^1 и, соответственно, время их решения будет добавляться к времени решения вершины q_r^1 .

В случае если несколько ЛД_{*f*} ($f = k, l, \dots, m$) выберут в результате описанной выше процедуры одну и ту же вершину q_l^1 , то она закрепляется за ПУ с меньшим порядковым номером.

После этого процесс размещения вершин первого яруса графа $G(Q, X)$ между ПУ f ($f = 1, 2, \dots, N$) повторяется заново.

ПУ f вновь проводит анализ элементов f -й строки таблицы A_1 за исключением элемента $a_{fr} = T_{\text{реш}}^f(q_r^1)$ и выделяет наименьший элемент $a_{fd} = T_{\text{реш}}^f(q_d^1)$. Далее анализируется d -й столбец и в случае, если в этом столбце присутствует элемент $a_{jd} < a_{fd}$, выбирается следующий по величине минимальный элемент f -й строки таблицы A_1 . Если же элемент $a_{fd} = T_{\text{реш}}^f(q_d^1)$ минимальный в столбце, то вершина q_d^1 закрепляется за ПУ f , о чем ЛД_{*f*} сообщает всем остальным ЛД. При этом все ЛД записывают номер f в d -й элемент вектора P_1 , а также модифицируют f -ю строку таблицы A_1 следующим образом:

$$a_{fl} = T_{\text{реш}}^f(q_l^1) + T_{\text{реш}}^f(q_d^1),$$

$$(l = 1, 2, \dots, M_1), \quad l \neq d, r.$$

Процесс продолжается до тех пор, пока не будут распределены (размещены) все вершины первого яруса графа $G(Q, X)$. Следует отметить, что в случае если $N < M$, то за одним ПУ будет закреплено несколько вершин первого яруса.

Далее ЛД переходят к размещению (распределению) вершин второго яруса графа $G(Q, X)$. С этой целью ЛД_{*f*} рассчитывает значения времени ре-

шения вершин второго яруса $q_s^2 (S=1, M_2)$ на ПУ_{*j*} ($j = 1, N$) с помощью формулы (3.9) следующим образом:

$$T_{\text{реш}}^i(q_s^2) = \max_{l=1, M_1} (T_{\text{реш}}^p(q_l^1) + T_{\text{пд}}(q_l^1, q_s^2)) + T(q_s^2),$$

причем $T_{\text{реш}}^p(q_l^1) = a_{lp}$, где a_{lp} – lp -й элемент таблицы A_1 ; p – номер ПУ, записанный в l -м элементе вектора P_1 (т.е. номер ПУ, за которым закреплена вершина q_l^1), а значения $T_{\text{пд}}(q_l^1, q_s^2)$ и $T(q_s^2)$ определяются как

$$T(q_s^2) = \frac{D(q_s^2)}{S_{\text{ПУ}}},$$

$$T_{\text{пд}}(q_l^1, q_s^2) = \begin{cases} \frac{W(q_l^1, q_s^2)}{Y_{\text{ПУ}}}, & \text{если } j \neq p, \\ 0, & \text{если } j = p, \end{cases}$$

где $T_{\text{пд}}(q_l^1, q_s^2)$ – время пересылки данных от ПУ_{*p*} к ПУ_{*j*}, причем если $j = p$, то $T_{\text{пд}}(q_l^1, q_s^2) = 0$;

$T(q_s^2)$ – время решения подзадачи q_s^2 на ПУ;

$D(q_s^2)$ – вычислительная сложность подзадачи q_s^2 ;

$W(q_l^1, q_s^2)$ – объем данных, соответствующих дуге $x(q_l^1, q_s^2)$.

Полученное таким образом значение $T_{\text{реш}}^i(q_s^2)$ записывается в таблицу A_2 в качестве элемента a_{js} . Аналогичным образом заполняется вся A_2 .

Далее на основе описанных выше принципов осуществляется размещение (распределение) вершин второго уровня между ПУ. Для этого ЛД_{*f*} анализирует f -ю строку таблицы A_2 и выделяет в ней минимальный элемент $a_{fl} = T_{\text{реш}}^f(q_l^2)$, после чего анализирует l -й столбец таблицы A_2 . Если элемент $a_{fl} = T_{\text{реш}}^f(q_l^2)$ самый минимальный и в l -м столбце, то вершина q_l^2 закрепляется за ПУ_{*f*}. ЛД_{*f*} информирует об этом всех остальных ЛД, на основании чего они записывают номер f в l -й элемент вектора P_2 , а также модифицируют таблицу A_2 путем увеличения всех элементов a_{fl} f -й строки на величину $T_{\text{реш}}^f(q_l^2)$ (за исключением самого элемента $T_{\text{реш}}^f(q_l^2)$).

Если же в l -м столбце есть элементы меньше, чем элемент a_{fl} , то ЛД_{*f*} выбирает в f -й строке следующий по величине наименьший элемент и вновь осуществляет описанную выше процедуру анализа на предмет возможности закрепления соответствующей этому элементу вершины графа $G(Q, X)$ за ПУ_{*f*}. Процесс продолжается до тех пор, пока не будут распределены все вершины второго яруса графа $G(Q, X)$.

Аналогичным образом распределяются вершины 3-го, 4-го и т.д. вплоть до $K-1$ -го яруса графа $G(Q, X)$.

По завершении описанной выше процедуры в векторах P_k ($k = 1, 2, \dots, K-1$) будет содержаться полная информация о номерах

ПУ, за которыми закреплены вершины k -го яруса графа $G(Q, X)$. На основании этих данных заполняется таблица T_2 размещения вершин графа $G(Q, X)$ по ПУ.

После того как процедура размещения завершена, с помощью формул (3.10) и (3.11) необходимо рассчитать значение получаемого при таком размещении времени $T_{\text{реш}}$. Если $T_{\text{реш}} > T_{\text{ц}}$, то это означает, что полученное размещение не удовлетворяет требованию обеспечения режима реального времени при решении задачи управления и потому дальнейшая работа ВП СУ невозможна. Если же $T_{\text{реш}} < T_{\text{ц}}^3$, то необходимо рассчитать значение ΔD согласно формуле (3.13). Если $\Delta D < \Delta D_{\text{дон}}$, то это означает, что требуемые условия размещения выполнены и можно переходить в режим решения задачи управления.

Если же $\Delta D > \Delta D_{\text{дон}}$, то необходимо осуществить этап модификации размещения с целью выравнивания нагрузки отдельных ПУ. Данная процедура также может быть выполнена с использованием «жадного» алгоритма. Для уменьшения трудоемкости процедуры можно рассматривать только варианты переноса подзадач с наиболее нагруженных ПУ на наименее нагруженные. Далее из рассмотренных вариантов выбрать тот, который дает наибольшее снижение значения ΔD при сохранении условия $T_{\text{реш}} < T_{\text{ц}}^3$, после чего повторить модификацию до тех пор, пока не будет достигнуто условие $\Delta D < \Delta D_{\text{дон}}$, либо не истечет интервал времени $T_{\text{ввп}}$, отведенный на размещение, т.е. $T_{\text{раз}}$ превысит значение $T_{\text{ввп}}$. Следует отметить, что в последнем случае ВП СУ все равно может переходить в режим решения задачи управления, т.к. основное требование $T_{\text{реш}} < T_{\text{ц}}^3$ по времени решения задачи управления выполняется.

Время $T_{\text{раз}}$ решения задачи размещения при использовании описанного выше алгоритма поярусного размещения можно оценить как:

$$\begin{aligned} T_{\text{раз}} &= T_{\text{э1}} + T_{\text{э2}} = \\ &= \sum_{k=1}^{K-1} (M_k T_B^* + M_k T_A + M_k N T_{\text{п}}) + \frac{M_{\text{он}}}{2} \left(\frac{M_{\text{он}}}{N} T_{\text{г}} + N T_{\text{п}} \right) \approx \\ &\approx M_{\text{он}} \left(T_B^* + T_A + N T_{\text{п}} + \frac{M_{\text{он}}}{2N} T_{\text{г}} + N \frac{T_{\text{п}}}{2} \right) \approx \\ &\approx M_{\text{он}} \left(\frac{1}{M_{\text{он}}} T_B + \frac{M_{\text{он}}}{2N} T_B + T_A + 1,5 T_{\text{п}} \right) \approx \\ &\approx M_{\text{он}} \left(\frac{M_{\text{он}}}{2N} T_B + T_A + 1,5 N T_{\text{п}} \right), \end{aligned}$$

где $T_{\text{э1}}$ – время выполнения этапа 1, т.е. поиск распределения, удовлетворяющего требованию $T_{\text{реш}} < T_{\text{ц}}^3$;

$T_{\text{э2}}$ – время выполнения этапа модификации полученного на этапе 1 размещения с целью выравнивания нагрузки на ПУ. При оценке $T_{\text{э2}}$ можно ис-

ходить из того, что переносится половина из $M_{\text{оп}}$ подзадач (считаем, что все подзадачи находятся на ПУ с повышенной нагрузкой) и, соответственно, выполняется $M_{\text{оп}}/2$ итераций, а в ходе одной итерации рассматривается M/N вариантов размещения (это среднее число подзадач на ПУ), причем на каждый вариант размещения тратится $T_{\text{в}}$ времени.

$T_{\text{в}}^*$ – время вычисления по формуле (3.9);

$T_{\text{в}}$ – время вычисления по формулам (3.11) и (3.12), приблизительно можно считать, что $T_{\text{в}} = M_{\text{оп}} T_{\text{в}}^*$;

$T_{\text{а}}$ – время анализа таблицы A_k и выделения минимального элемента в строке и в столбце;

$T_{\text{п}}$ – время передачи сообщения о закреплении вершины графа за ПУ локальным диспетчером ЛДf, о выбранном варианте размещения всем остальным ЛД;

$T_{\text{оп}}$ – число операционных вершин в графе $G(Q, X)$, $M_{\text{оп}} = \sum_{k=1}^{K-1} M_k$.

3.4.3.2. Алгоритм частичного размещения

Время $T_{\text{раз}}$ решения задачи размещения можно сократить за счет использования следующего подхода, основная идея которого основывается на предположении, что исходное размещение вершин графа $G(Q, X)$ по ПУ, установленное до момента возникновения отказа, допустимо (т.е. обеспечивается выполнение условий $T_{\text{реш}} < T_{\text{ц}}^3$ и $\Delta D < \Delta D_{\text{дон}}$).

Допустим, что в решении задачи управления было задействовано H основных ПУ. Предположим, что обнаружен отказ $L < H$ основных ПУ, причем на этих L отказавших ПУ до момента отказа было размещено множество подзадач

$$Q_{\text{в}} = \bigcup_{i=1}^L Q_{\text{в}}^i$$

где $Q_{\text{в}}^i$ – множество подзадач, размещенных на i -м отказавшем ПУ.

Поскольку начальное размещение вершин графа $G(Q, X)$ по H основным ПУ считаем допустимым, то можно предположить, что размещение вершин множества $Q_{\text{а}} = Q_{\text{оп}}/Q_{\text{в}}$ (где $Q_{\text{оп}}$ – множество операционных вершин графа $G(Q, X)$) по $(H - L)$ оставшимся работоспособным ПУ оптимально, и поэтому не имеет смысла их переразмещать заново. Иными словами, в этом случае задача размещения вершин графа $G(Q, X)$ по работоспособным ПУ сводится к задаче размещения не всего множества вершин $Q_{\text{оп}}$ по $N(H + B - L)$ работоспособным ПУ, а только подмножества $Q_{\text{в}} \subseteq Q_{\text{оп}}$ вершин, закрепленных ранее за отказавшими ПУ.

Для этого представим вершины множества в виде кортежа $Q_{\text{в}} = \langle q_1^{\text{в}}, q_2^{\text{в}}, \dots, q_{M_1}^{\text{в}} \rangle$, причем номер вершины $q_i^{\text{в}}$ в кортеже должен определяться ярусом графа $G(Q, X)$, на котором данная вершина расположена. Иными словами, чем меньше номер яруса графа $G(Q, X)$, на котором расположена вершина $q_i^{\text{в}}$, тем меньше должен быть ее номер i в кортеже. Если же в множество $Q_{\text{в}}$ попадают вершины $q_i^{\text{в}}$ и $q_j^{\text{в}}$, расположенные на одном

ярусе $G(Q, X)$, то последовательность их размещения в кортеже может быть произвольной.

Далее каждый локальный диспетчер ЛД_{*i*} ($i = 1, N$) ПУ берет вершину из кортежа Q_B и находит абсолютное время ее решения $T_{\text{реш}}^i(q_1^B)$ с помощью формулы (3.9) при условии ее закрепления за «своим» ПУ_{*i*}. При этом, поскольку вершины графа $G(Q, X)$, расположенные на более низких ярусах, чем вершина q_1^B , принадлежат множеству Q_A ранее размещенных вершин, то формула (3.9) будет полностью определена. Полученное ЛД_{*i*} значение $T_{\text{реш}}^i(q_1^B)$ сообщается всем остальным ЛД. Среди значений $T_{\text{реш}}^i(q_1^B)$, полученных всеми ЛД_{*i*} ($i = 1, N$), выбирается минимальное, при этом вершина q_1^B закрепляется за тем ПУ_{*j*}, который обеспечивает это минимальное значение $T_{\text{реш}}^i(q_1^B)$. Далее аналогичный процесс закрепления повторяется для вершины q_2^B , затем для вершины q_3^B и т.д. вплоть до вершины $q_{M_1}^B$. В результате реализации такой последовательной процедуры размещения вершин множества Q_B по N работоспособным ПУ с учетом имеющегося (ранее полученного) размещения вершин множества $Q_A = Q_{\text{оп}}/Q_B$ получаем новое размещение вершин всего графа $G(Q, X)$ по работоспособности ПУ.

Как и в случае рассмотренного выше алгоритма поярусного размещения, в данном случае до завершения процедуры размещения величины $T_{\text{реш}}$ и ΔD не могут быть оценены. Поэтому процедуру поиска размещения, удовлетворяющего условиям $T_{\text{реш}} < T_{\text{ц}}^3$ и $\Delta D < \Delta D_{\text{дон}}$, также имеет смысл разбить на два этапа: этап поиска размещения, обеспечивающего выполнение первого условия, и этап поиска модификации полученного размещения для удовлетворения второго условия. При выполнении второго этапа можно использовать метод, аналогичный описанному в предыдущем параграфе.

При использовании данного алгоритма время $T_{\text{раз}}$ решения задачи размещения можно оценить следующим образом:

$$\begin{aligned} T_{\text{раз}} &= T_{\text{э1}} + T_{\text{э2}} = M_B(T_B^* + NT_{\text{п}}) + M_B(T_B + NT_{\text{п}}) = \\ &= M_B\left(\frac{1}{M_{\text{оп}}}T_B + T_B + 2NT_{\text{п}}\right) \approx M_B(T_B + 2NT_{\text{п}}), \end{aligned}$$

где $T_{\text{э1}}$ – время выполнения этапа 1, т.е. поиск распределения, удовлетворяющего требованию $T_{\text{реш}} < T_{\text{ц}}^3$;

$T_{\text{э2}}$ – время выполнения этапа модификации полученного на этапе 1 размещения с целью выравнивания нагрузки на ПУ. При оценке $T_{\text{э2}}$ можно исходить из того, что переносится каждая из M_B подзадач и, соответственно, выполняется M_B итераций, а в ходе одной итерации одним ЛД рассматривается один вариант размещения (считаем, что один ЛД «принимает» одну подзадачу), причем на каждый вариант размещения тратится T_B времени;

$$M_B = |Q_B|;$$

T_B^* – время вычисления по формуле (3.9);

T_v – время вычисления по формуле (3.10) и (3.11), приблизительно можно считать, что $T_v = M_{оп} \cdot T_B^*$;

T_n – время передачи сообщений между ЛД.

3.4.4. Резюме

В результате проведенных исследований дана формальная постановка задачи размещения подзадач задачи управления по процессорным узлам и сформулированы необходимые условия ее допустимого решения, а именно допустимое решение задачи размещения должно обеспечивать время решения задачи управления менее заданного времени цикла управления, а также отклонение нагрузки отдельных ПУ от равномерной менее заданного порогового значения.

Показано, что решение задачи размещения должно выполняться за время, не превышающее время восстановления вычислительного процесса. Для сокращения времени решения задачи размещения предложено распараллеливать процесс ее решения с помощью множества ЛД отдельных ПУ.

Разработан ряд алгоритмов размещения при различных способах организации вычислительного процесса в ВП СУ ВС, обеспечивающих достижение допустимого размещения и ориентированных на параллельную реализацию с помощью множества ЛД.

3.5. Реконфигурация вычислительных подсистем систем управления с интегрированной модульной архитектурой

3.5.1. Особенности функционирования вычислительных подсистем с интегрированной модульной архитектурой

Выше рассмотрены обобщенные подходы к проблеме повышения надежности ВП СУ за счет использования способа резервирования производительности отдельных ПУ, методов распределенного диспетчирования и реконфигурации ВП СУ при возникновении отказов. Применение данных методов в ВП СУ ВС с архитектурой ИМА требует учета ряда особенностей организации работы последних. Рассмотрим эти особенности более подробно. При этом под задачей будем понимать комплекс программ (приложений), обеспечивающих общую функциональность ВП СУ ИМА, под приложением будем понимать часть задачи, которая выполняет определенный набор функций, а подзадачей считать одну из копий такого приложения.

ВП СУ ВС с архитектурой ИМА являются системами жесткого реального времени, поэтому имеют ряд особенностей в организации работы как отдельных ПУ (на уровне ОС), так и КС, которые описаны в стандартах ARINC 653 (в части высокоуровневой программной модели ИМА и ARINC 664 (в части КС) (рисунок 3.32). Данные особенности направлены

как на обеспечение детерминированности работы системы управления ВС в целом, так и на повышение ее надежности.

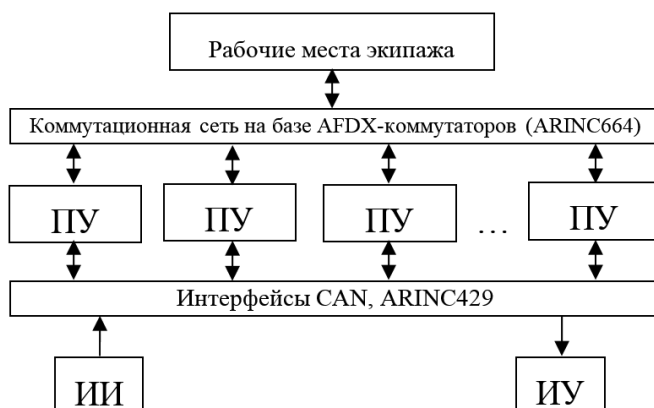


Рис. 3.32. Обобщенная структура ВП СУ ВС с архитектурой ИМА

В соответствии с данным стандартом все подзадачи, запущенные на ПУ, должны выполняться в изолированных областях, именуемых разделами, как это показано на рисунке 3.33.

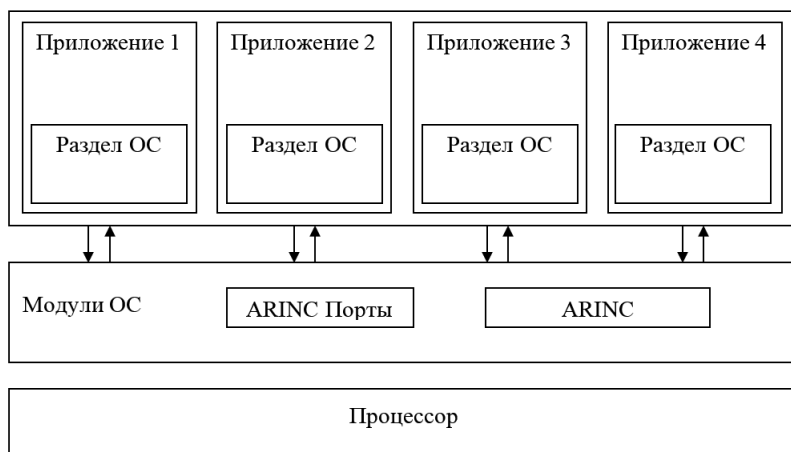


Рис. 3.33. Иллюстрация пространственного разделения

Раздел подобен виртуальной машине: он содержит данные, его собственный контекст, атрибуты конфигурации, и т.д. В разделе может выполняться один или более процессов, составляющих подзадачу. Согласно параметрам раздела, процессы могут выполняться одновременно для достижения требований реального времени, как это показано на рисунке 3.34. Процессы не видны за пределами раздела.

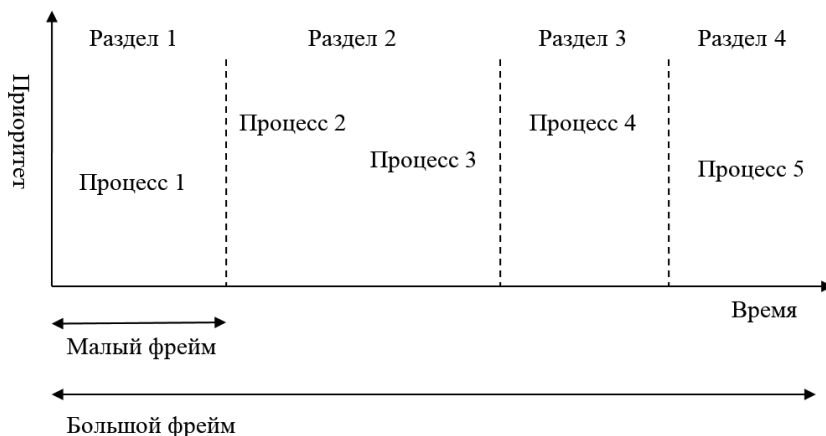


Рис. 3.34. Иллюстрация временного разделения

Ресурсы системы, требующиеся для выполнения процессов в разделе, выделяются статически во время построения системы и при инициализации системы. Создание и инициализация процессов происходит при инициализации раздела. Раздел может переинициализировать любой из своих процессов в любое время. Каждый процесс создается только однажды на протяжении всей жизни раздела. В случае сбоя процесса раздел может перезапустить или завершить данный процесс.

Каждый процесс имеет приоритет и режим работы: синхронный (периодический) или асинхронный. Оба типа процессов могут выполняться в одном разделе. Если выполнение процесса прервано концом временного окна раздела, данный процесс продолжит выполнение после возобновления работы раздела.

Связь между разделами осуществляется через сообщения. Сообщение определено как непрерывный блок данных конечной длины. Сообщение посылается от одиночного источника к одному или нескольким адресатам. Адресат сообщения представляет собой раздел, а не процесс в пределах раздела.

Основным механизмом осуществления связи между разделами с помощью сообщений является канал (порт ARINC). Канал определяет логическую связь между одним источником и одним или большим количеством адресатов, где источником и адресатами могут быть или разделы, или псевдоразделы. Он также определяет режим передачи сообщений от источника к адресатам вместе с характеристиками сообщений, которые должны быть посланы от этого источника к этим адресатам.

Разделы имеют доступ к каналам через порты. Канал состоит из одного или большего количества портов и соответствующих ресурсов. Порт предоставляет требуемые ресурсы, которые позволяют определенному разделу или посылать, или получать сообщения в определенном канале. Разде-

лам или псевдоразделам разрешается осуществлять обмен сообщениями по многочисленным каналам через порты соответствующих источников и адресатов. Канал описывает маршрут подключения одного порта отправки сообщения к одному или нескольким портам получения.

Если кратко выделить наиболее существенные с точки зрения возможности реконфигурации ВП СУ ВС моменты в части особенностей (ограничений) стандарта ARINC 653, то таковыми являются:

- статическое задание конфигурации ПУ в момент начала работы, т.е. количество разделов и их ресурсы не могут в дальнейшем изменяться без перезапуска ПУ;

- статическое задание в момент начала работы информационных каналов (портов ARINC), через которые приложения (подзадачи), выполняемые в разделе, могут взаимодействовать с приложениями других разделов и приложениями на других ПУ.

В целях обеспечения работы в жестком реальном времени при организации информационного взаимодействия между ПУ в ВП СУ на базе ИМА применяется концепция виртуальных связей (по стандарту ARINC 664). Данные связи фактически являются продолжением каналов, предусмотренных стандартом ARINC 653 (внутри ПУ), в случае необходимости информационного взаимодействия между приложениями на разных ПУ. Можно провести аналогию между электрическими соединениями внутри микросхемы (каналы ARINC 653) и соединениями между микросхемами (виртуальные связи ARINC 654).

Стандарт ARINC 664 основан на Ethernet-интерфейсе (можно работать с UDP и TCP (как опция)), но при этом является дублированным (рисунок 3.35).

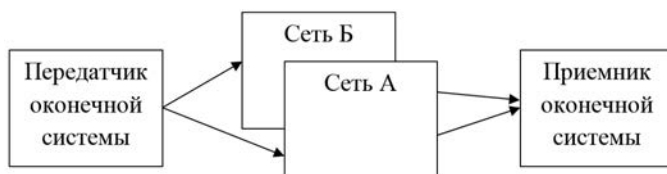


Рис. 3.35. Дублированная передача данных в стандарте ARINC 664

Однако стандарт ARINC 664 накладывает серьезные ограничения на возможности реконфигурации ВП СУ на базе ИМА вследствие статически задаваемых конфигурационных данных, загружаемых в коммуникационное оборудование в момент начала работы:

- все виртуальные связи должны быть определены сразу;
- виртуальные связи назначаются конкретным разделам-источникам и разделам-приемникам.

Если в процессе работы ВП СУ на базе ИМА происходит отказ ПУ и вместо него необходимо включить в работу резервный, то необходимо:

- либо произвести переинициализацию коммуникационного оборудования;
- либо предусмотреть в конфигурационных данных дополнительные связи.

Переинициализация коммуникационного оборудования занимает определенное время (секунды), в течение которого передача информации между всеми или частью ПУ ВП СУ на базе ИМА будет невозможна, что для авиационных систем недопустимо.

При реализации второго варианта необходимо следить за тем, чтобы возможностей коммуникационного оборудования «хватило» для «основных» и дополнительных связей, т.к. для тех и других ресурсы будут выделены сразу.

3.5.2. Способы «преодоления» ограничений стандартов ARINC 653 и ARINC 664 при выполнении процедуры реконфигурации

Выше приведены основные особенности ВП СУ на базе ИМА на базе стандартов ARINC 653 и ARINC 664, а также возникающие при этом ограничения, затрудняющие в некоторой степени проведение процедуры реконфигурации ВП СУ ВС при возникновении отказов. Рассмотрим способы «преодоления» этих ограничений.

Обратим внимание на тот факт, что ограничения, связанные со стандартом ARINC 653, относятся в первую очередь к отдельно взятому ПУ, поэтому могут быть сняты с помощью перезапуска конкретного(ых) ПУ. Это допустимо, например, для резервных ПУ, т.к. на них изначально нет запущенных подзадач. Однако подход к резервированию, основанный только на использовании резервных ПУ, как это было показано выше, не позволяет в полной мере использовать имеющиеся ресурсы в целях обеспечения заданного уровня отказоустойчивости, т.к. на свободные ресурсы «основных» ПУ в случае отказа можно разместить подзадачи с отказавших устройств. Учитывая, что нормальным считается использование ресурсов ПУ на 50%–90% [30], то потенциально из двух ПУ, нагруженных на 50%, можно парировать один отказ без использования резервных устройств. Поэтому необходимо найти возможность использования данных ресурсов.

Все подзадачи могут быть разделены на две категории – критичные и не критичные. Выполнение не критичных задач в случае необходимости может быть прекращено. В связи с этим возможны следующие способы использования ресурсов «основных» ПУ:

- размещение в первоначальной конфигурации ВП СУ на базе ИМА не критичных задач на отдельных ПУ, которые можно в случае необходимости перезапустить, не затрагивая критичные подзадачи;
- создание на ПУ, содержащих критичные подзадачи, дополнительных копий таких подзадач, находящихся в «теплом» резерве, т.е. не выполняющих реальных вычислений.

Первый способ, показанный на рисунке 3.36, позволяет в случае отказа размещать подзадачи отказавших узлов сначала на свободные ресурсы ПУ только с не критичными подзадачами, а затем и вместо последних (если размещаемые подзадачи критичные).

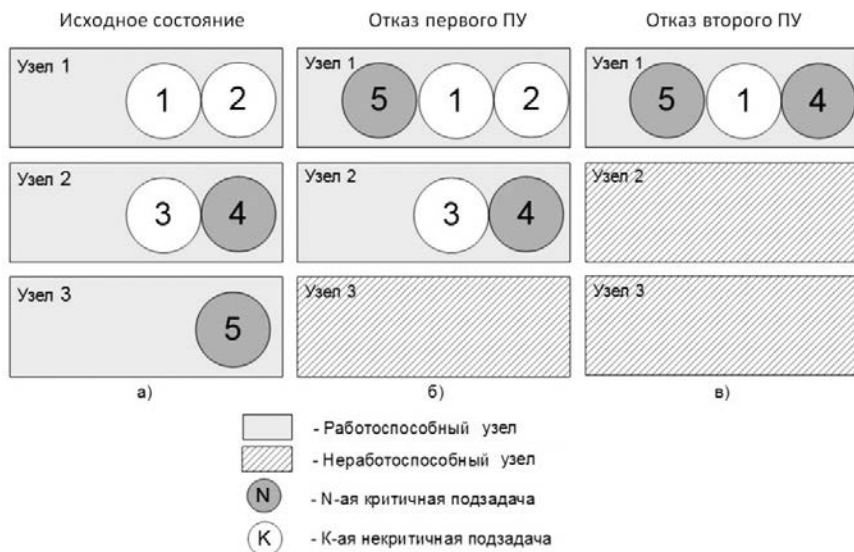


Рис. 3.36. Вариант реализации резервирования с одним ПУ, на котором размещены только некритичные подзадачи, имеющем запас ресурсов

Второй способ, показанный на рисунке 3.37, проигрывает первому в гибкости, однако при наличии неиспользованных ресурсов вполне может быть использован, т.к. не требует в случае переноса подзадачи перезапуска целевого ПУ.

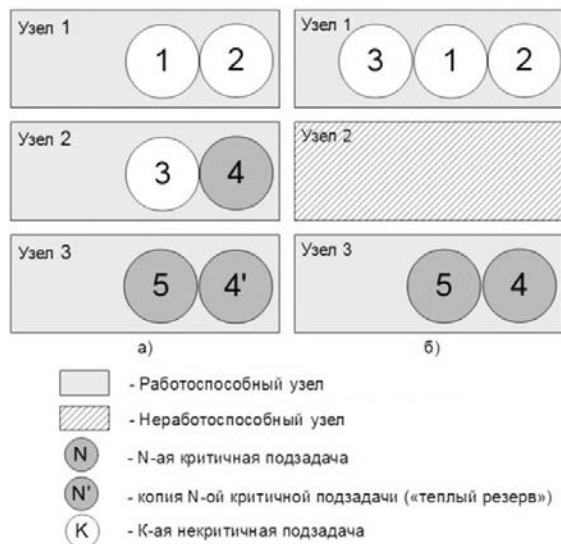


Рис. 3.37. Вариант реализации резервирования с одним ПУ, на котором размещены только некритичные подзадачи, и дополнительной копией критичной задачи, находящейся в «теплом» резерве

Более сложные проблемы вносят ограничения, накладываемые в части коммуникационной среды ВП СУ ИМА. Как уже отмечалось выше, перезапуск коммуникационного оборудования приводит к останову всей ВП СУ на базе ИМА до момента завершения данной процедуры. Поскольку перезапуск занимает несколько секунд, то во многих случаях прекращение работы ВП СУ на базе ИМА на такой промежуток времени не является допустимым.

Альтернативой является резервирование на этапе первоначального конфигурирования коммуникационного оборудования дополнительных связей, как это показано на рисунке 3.38, которые в случае отказов могут быть использованы для обеспечения взаимодействия подзадач после их перераспределения в результате реконфигурации (на рисунке дополнительные связи показаны серым цветом). В дальнейшем будем считать, что процедура перезагрузки коммуникационного оборудования является недопустимой.

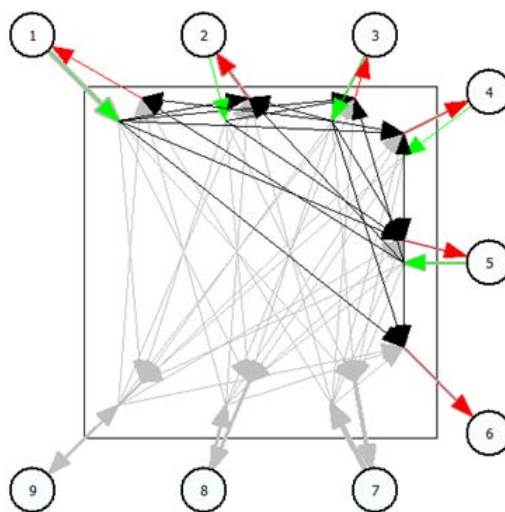


Рис. 3.38. Пример конфигурации коммуникационной среды с дополнительными связями

3.5.3. Алгоритм реконфигурации вычислительной подсистемы с учетом ограничений стандартов ARINC 653 и ARINC 664

С учетом приведенных выше соображений можно предложить следующий алгоритм реконфигурации ВП СУ на базе ИМА при возникновении отказов.

В случае выхода из строя одного из ПУ в ВП СУ на базе ИМА необходимо произвести перераспределение подзадач с вышедшего из строя узла на работоспособные ПУ. При этом алгоритм перераспределения подзадач должен учитывать следующие ограничения:

- 1) при перераспределении подзадач на работоспособный ПУ предполагается его перезапуск, т. е. те подзадачи, которые уже работали на этом узле на время его перезапуска, становятся временно неработоспособными;

2) каждое приложение, работающее в рамках ВП СУ на базе ИМА, может иметь более одной копии, т. е. в ВП СУ на базе ИМА может одновременно выполняться несколько программ, выполняющих одно и то же задание, что повышает надежность функционирования ВП СУ на базе ИМА в целом;

3) задачи, которые не требуют переконфигурации (т.н. «некритичные» подзадачи), могут выполняться в единственном экземпляре;

4) задачи, которые требуют переконфигурации (т. н. «критичные» подзадачи), должны выполняться во множестве копий (при этом число копий должно быть на одну больше, чем число предполагаемых парируемых отказов);

5) в процессе реконфигурации число копий «некритичных» подзадач по возможности должно быть больше одной, однако в некоторых случаях, когда другие варианты недоступны, допускается полная остановка всех копий подобных подзадач;

6) в процессе реконфигурации число копий «критичных» подзадач изменяться не должно, допускается остановка только тех копий подзадач, которые располагались на вышедшем из строя узле;

7) если в процессе реконфигурации (восстановления работоспособности) продолжают происходить выходы из строя новых узлов, допускается временное отключение «критичных» подзадач (на время перераспределения подзадач по работоспособным узлам); в этом случае парирование отказов осуществляется за счет наличия избыточного количества копий «критичных» подзадач, рассмотренного в пункте 4.

Ниже приведен алгоритм перераспределения «некритичной» подзадачи.

Алгоритм 3.4.

1) Если данная подзадача является последней копией, начать ее перераспределение, перейдя к п. 2, иначе – переразмещение подзадачи не производить, перейти к п. 5.

2) Найти полностью незагруженный (резервный) работоспособный узел и переразместить подзадачу на него, после чего закончить процесс переразмещения данной подзадачи, перейдя к п. 5.

3) Если полностью незагруженный узел не был найден – найти работоспособный узел с минимальной нагрузкой, который загружен только «некритичными» подзадачами и перераспределить подзадачу на него, после чего закончить процесс переразмещения данной подзадачи, перейдя к п. 5.

4) Если ни резервного узла, ни полностью занятого «некритичными» подзадачами узла не найдено – полностью отказаться от перераспределения данной «некритичной» подзадачи, сообщив об этом пользователю.

5) Конец процесса переразмещения «некритичной» подзадачи.

Алгоритм перераспределения «критичной» подзадачи выглядит следующим образом.

Алгоритм 3.5.

1) Найти полностью незагруженный (резервный) работоспособный узел и переразместить подзадачу на него, после чего закончить процесс переразмещения данной подзадачи, перейдя к п. 5.

2) Если полностью незагруженный узел не был найден – найти работоспособный узел с минимальной нагрузкой, который загружен только «некритичными» подзадачами, и перераспределить подзадачу на него, после чего закончить процесс перераспределения данной подзадачи, перейдя к п. 5.

3) Если на работоспособном узле с «некритичными» подзадачами недостаточно вычислительных ресурсов для размещения «критичной» подзадачи – «удалять» «некритичные» подзадачи до тех пор, пока ресурсов не станет достаточно для размещения «критичной» подзадачи, после чего перерасместить рассматриваемую подзадачу на данный узел и закончить процесс перераспределения, перейдя к п. 5.

4) Если перерасместить подзадачу не получается – выдать пользователю сообщение об ошибке.

5) Конец процесса перераспределения «некритичной» подзадачи.

Общая схема алгоритма реконфигурации ВП СУ на базе ИМА представлена на рисунке 3.39.

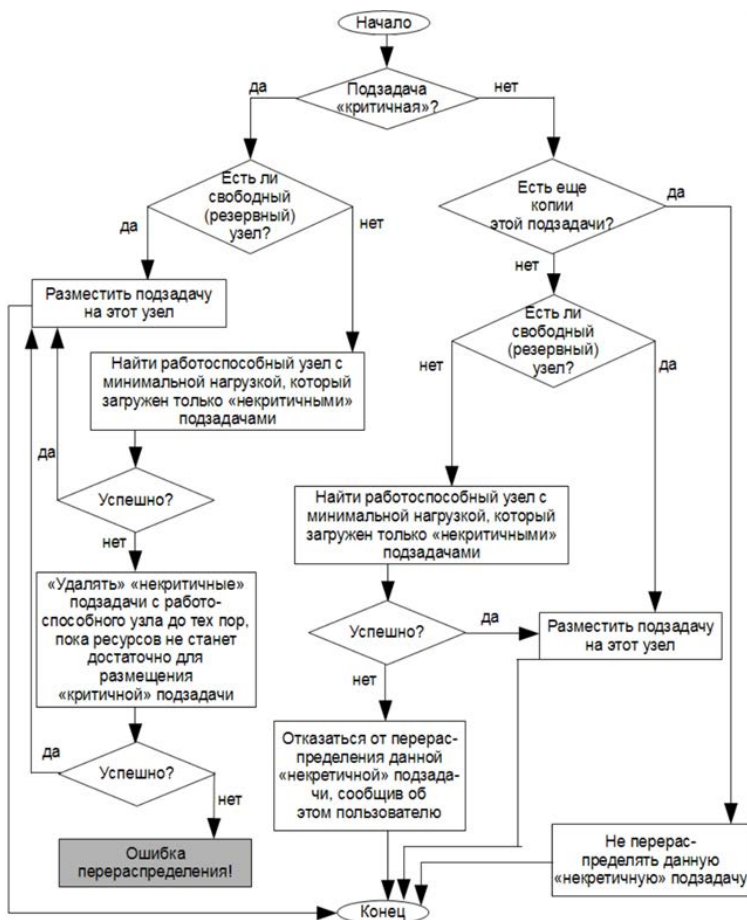


Рис. 3.39. Схема алгоритма перераспределения подзадач в ВП СУ на базе ИМА при выходе из строя узла (узлов)

Рассмотрим работу данного алгоритма на примере. Предположим, что в наличии имеется ВП СУ на базе ИМА со следующей начальной конфигурацией, изображенной на рисунке 3.40:

а) в системе имеется 4 ПУ, каждый вычислительной мощностью 100 условных единиц;

б) состав подзадач: 1-я, 2-я и 3-я подзадачи – «критичные», 4-я и 5-я – «некритичные»;

в) трудоемкость подзадач одинаковая – 40 условных единиц (т. е. на каждый узел можно распределить не более 2-х подзадач, при этом останется резерв по вычислительной мощности в 20 условных единиц);

г) начальное распределение подзадач: подзадачи 1 и 2 – на 1-м узле, подзадачи 3 и 4 – на 2-м, 5-я подзадача – на 3-м узле; узел номер 4 подзадачами не загружен и находится в резерве.

Предположим, произошел отказ 1-го и 2-го ПУ. В этом случае необходимо переконфигурировать систему таким образом, чтобы в обязательном порядке восстановить работоспособность подзадач 1–3 и по возможности – подзадач 4 и 5. Шаги по реконфигурированию системы описаны ниже и проиллюстрированы на рисунке 3.41.

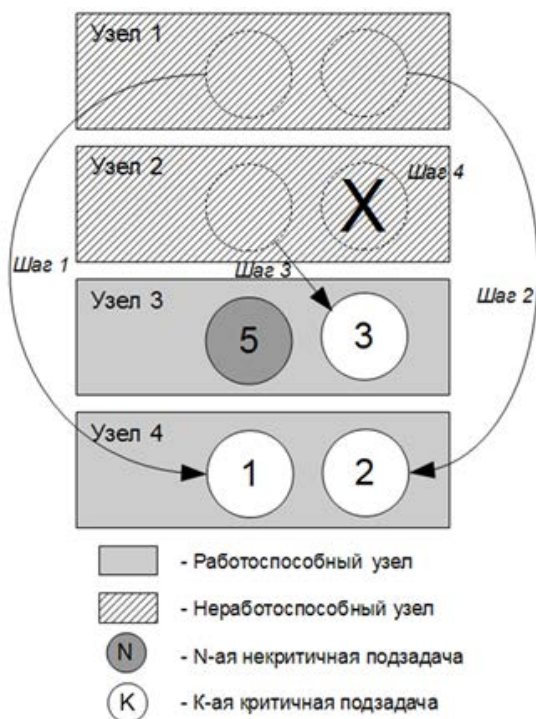
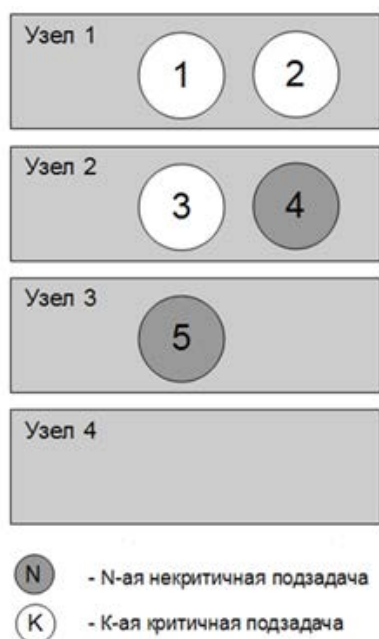


Рис.3.40. Начальная конфигурация ВП СУ на базе ИМА (пример)

Рис. 3.41. Реконфигурация ВП СУ на базе ИМА после отказа части узлов

Шаги 1 и 2. В соответствии с изложенным выше алгоритмом, подзадачи 1 и 2 (как «критичные») перераспределяются на полностью незанятый (резервный) узел номер 4.

Шаг 3. Т.к. свободных узлов нет, подзадача 3, являясь тоже «критичной», распределяется на узел номер 3. При этом на время перезапуска узла мы временно приостанавливаем работу подзадачи 5, что допускается, т. к. подзадача номер 5 «критичной» не является.

Шаг 4. Подзадача номер 4 не может быть перераспределена, поскольку уже нет ни полностью ненагруженных узлов, ни узлов с «некритичными» подзадачами и свободными вычислительными ресурсами. В соответствии с алгоритмом подзадача оказывается «утерянной», что допустимо ввиду ее статуса «некритичной».

3.5.4. Реализация реконфигурации процессорных узлов с помощью планировщика расписаний

В качестве основы для реализации процедуры реконфигурации ВП СУ на базе ИМА может быть использован механизм переключения расписаний. В некоторых реализациях операционных систем (ОС) реального времени (РВ) на базе ARINC 653 предусмотрена возможность использования нескольких расписаний работы разделов в ходе работы ПУ. Например, планировщик расписаний предоставляется ОС VxWorks 653 в виде библиотеки *arincSchedLib*. Расписания формируются на этапе подготовки ПУ к работе.

В расписании определяется, каким разделам и в какие временные интервалы предоставляется возможность выполнения. Это позволяет в расписании предоставлять процессорное время только тем разделам, работа которых необходима в рамках текущей конфигурации ВП СУ на базе ИМА. Таким образом, каждое из расписаний может соответствовать одной из возможных конфигураций ПУ в рамках работы ВП СУ на базе ИМА.

Применение механизма переключения расписаний позволяет реализовать реконфигурацию ПУ без перезагрузки, что делает реконфигурацию ПУ практически мгновенной. Однако существует ограничение на количество расписаний, накладываемое операционной системой. Например, количество расписаний для ОС РВ VxWorks 653 равно 16. Поэтому при формировании набора конфигураций для каждого ПУ следует укладываться в существующие ограничения. Кроме того, в начале работы ПУ все разделы, работа которых предусмотрена во всех расписаниях, должны быть загружены в оперативную память, что также накладывает определенные ограничения.

Рассмотрим, как влияют указанные ограничения на возможности применения скользящего резервирования и резервирования производительности. Предположим, что для обеспечения заданного уровня надежности необходимо обеспечить возможность парирования трех отказов ПУ ВП СУ.

В случае скользящего резервирования для парирования отказа трех ПУ необходимо наличие трех резервных узлов. При использовании механизма

ARINC-планировщика в ОС PB VxWorks возникает проблема в виде ограничений на число расписаний, доступных для работы данного планировщика: более 16 вариантов (комбинаций) работы подзадач реализовать не удастся.

Исходя из этого, получается, что три резервных узла смогут парировать только отказы 15 ПУ (еще одно расписание планировщика резервируется для обеспечения работы резервного ПУ во время отсутствия сбоя в ВП СУ). Соответственно, в случае, если число основных ПУ более 15 (но не более 30), для обеспечения работы остальных ПУ потребуется еще три резервных ПУ (рисунок 3.42). Таким образом, в любой из возможных конфигураций ВП не менее 3 ПУ будет находиться в резерве, что существенно ухудшает характеристики в плане производительности, массы и габаритов. Кроме того, резервные узлы должны иметь объем оперативной памяти, достаточный для размещения разделов (функциональных программ) со всех основных ПУ (до 15 ПУ). Таким образом, полезная вычислительная мощность ВП СУ, включающей 30 ПУ при использовании резервирования ПУ, составит 80% от максимальной и будет равна производительности 24 ПУ.

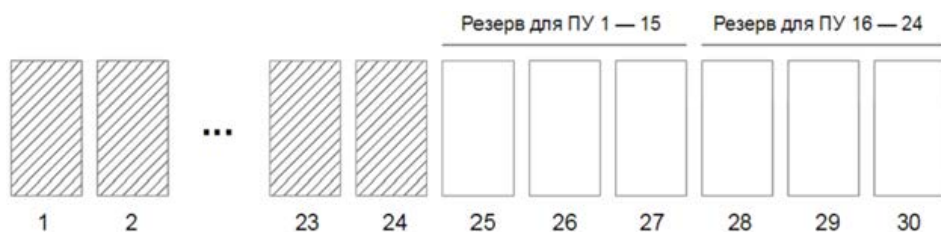


Рис. 3.42. Пример схемы, описывающей использование скользящего резервирования ВП СУ

В случае резервирования производительности при количестве ПУ от 4 до 30 и необходимости парирования трех отказов можно обеспечить эффективное использование всех ПУ при количестве расписаний не более 16 (рисунок 3.43). Предположим, что все ПУ ВП СУ на базе ИМА загружены равномерно: 80% производительности используется для работы подзадач (заштрихованная область), 20% – резерв производительности (незаштрихованная область). При таком соотношении полезная вычислительная мощность также будет равна производительности 24 ПУ.

В случае если из строя выходит ПУ номер 1, нам требуется перераспределить его подзадачи на 4 других ПУ. Пусть для этого выделяются ПУ 2 – 5).

Обозначим расписания для обеспечения реконфигурации парой чисел, разделенных точкой (над изображением ПУ). Например, «третья четверть вычислительной нагрузки с первого ПУ» будет обозначена как «1.3», «третья четверть вычислительной нагрузки с двенадцатого ПУ» как «12.3» и т. д.

В том случае, если в условной «группе резерва» происходит или уже произошел отказ ПУ (например, в первой группе из ПУ номер 2, 3, 4, 5 вышел из строя ПУ номер 2), требуется произвести перераспределение вычис-

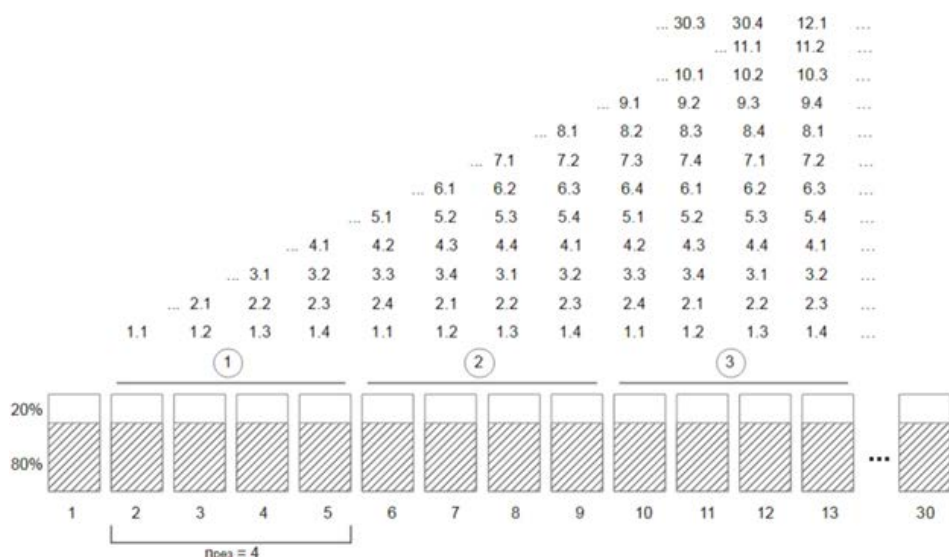


Рис. 3.43. Пример схемы, описывающей использование расписаний ARINC-планировщиков

лительных ресурсов на другой работоспособный ПУ. В рассматриваемом примере для этого существует вторая «группа резерва», состоящая из ПУ с номерами 6, 7, 8, 9. Если же и во второй группе «резервных» ПУ произойдет отказ, то необходимо задействовать третью группу. При формировании «резервных» групп необходимо применять кольцевой принцип нумерации узлов, т.е. при достижении ПУ 30 «резервирование» продолжается с ПУ 1. Всего понадобится 12 расписаний для обеспечения парирования отказов 3 любых ПУ из 30 и одно для первоначальной конфигурации, когда отказов нет, т.е. всего 13.

Если же рассматривать общий случай, т.е. B отказов, то, соответственно, необходимо создать B «резервных групп», каждая из которых состоит из $n_{рез}$ ПУ (в рассмотренном случае $n_{рез} = 4$). Следующее неравенство позволяет оценить возможность реализации резервирования производительности в общем случае:

$$1 + B \cdot n_{рез} \leq N_{расп},$$

где B – число парируемых отказов,

$n_{рез}$ – число ПУ в «резервной группе»,

$N_{расп}$ – максимальное число доступных расписаний.

Обратим внимание на тот факт, что количество конфигураций не зависит от числа ПУ в ВП СУ. Это означает, что в случае применения резервирования производительности ограничение на количество расписаний ОС РВ не влияет на количество ПУ в ВП СУ.

Если $N_{расп} = 16$, то при $B = 3$ максимальное значение $n_{рез}$ будет равно 5. Соответственно, в каждом ПУ минимально возможный резерв производительности составит 1/6 производительности ПУ. Следовательно, макси-

мальная полезная вычислительная мощность ВП СУ, включающей 30 ПУ, в случае применения резервирования производительности будет равна производительности 25 ПУ. Это на производительность одного ПУ больше, чем в случае резервирования ПУ.

Сравним показатели скользящего резервирования и резервирования производительности для различных конфигураций ВП СУ при условии обеспечения ее равной реальной вычислительной мощности. Сравнение проведем по следующим показателям: необходимое число ПУ, необходимый объем оперативной памяти. Будем полагать, что во всех случаях обеспечивается парирование отказа трех ПУ.

В таблице 3.2 приведены данные по необходимому числу ПУ, из которых видно, что оба подхода к резервированию дают возможности для реализации процесса реконфигурирования ВП СУ в случае отказа ее частей. Однако при использовании скользящего резервирования при числе ПУ более 18 требуется вводить в состав системы больше ресурсов для обеспечения равной эффективной вычислительной мощности, так как при этом уже не удастся вписаться в ограничение целевой ОС по количеству слотов расписания планировщика.

Табл. 3.2. Число ПУ, необходимое для реализации резервирования при ограничении количества конфигураций

Скользящее резервирование (всего = основные + резервные)	Резервирование производительности	Реальная вычислительная мощность ВП СУ
$30 = 24 + 6$	29	24 ПУ
$18 = 15 + 3$	18	18 ПУ
$9 = 6 + 3$	9	6 ПУ
$4 = 1 + 3$	4	1 ПУ

При оценке объема необходимой оперативной памяти будем считать, что он пропорционален загрузке ПУ. То есть в случае скользящего резервирования будем считать, что для работы основных ПУ необходим объем оперативной памяти в 1 у.е. (100% загрузка). Тогда для резервных ПУ потребуется объем памяти $H - \left\lfloor H / (N_{расч} - 1) \right\rfloor \cdot H$ у.е, где H – число основных ПУ (суммируем загрузку для всех конфигураций с учетом ограничения на количество расписаний). В случае резервирования производительности для обеспечения размещения подзадач для первоначальной конфигурации (без отказов) понадобится $H / (H + B)$ у.е., с учетом резервных конфигураций – еще $B^2 n_{рез} / (H + B)$ у.е. Данные о необходимом объеме памяти для различных конфигураций при 3-х парируемых отказах приведены в таблице 3.3.

Табл. 3.3. Объемы памяти, необходимые для работы различных конфигураций ВП СУ

Число ПУ	Объем оперативной памяти при резервировании ПУ		Объем оперативной памяти при резервировании производительности
	Основные ПУ(у.е.)	Резервные ПУ(у.е.)	
30	1	12	2,4
18	1	15	3,33
9	1	6	3,67
4	1	1	1

В случае резервирования ПУ необходимо применять узлы, значительно отличающиеся по объему оперативной памяти, что делает невозможной унификацию применяемых ПУ.

3.5.5. Организация работы с контекстными данными при реализации реконфигурации за счет смены расписаний

Некоторым подзадачам для возобновления работы после перемещения на другой ПУ в ходе реконфигурации могут потребоваться данные, отражающие работу в первоначальном месте размещения (контекстные данные). В связи с этим возникает необходимость:

- сохранения текущих данных активных экземпляров подзадач;
 - предоставления этих данных подзадаче в случае переноса на другой ПУ.
- Возможны следующие способы реализации такой функциональности.

Первый способ хранения/предоставления контекстных данных представляет собой «крайний» вариант делегирования функционала: все функции, связанные с организацией хранения и предоставления контекстных данных, возлагаются на ЛД. При этом ЛД может обмениваться информационными сообщениями с активными подзадачами, а подзадача, становясь активной после реконфигурации, может запросить у ЛД актуальные контекстные данные.

Достоинством данного способа является необходимость внесения минимальных изменений в подзадачи, поскольку реализация функционала возложена на ЛД, а функции подзадачи ограничиваются только отправкой обновленных контекстных данных и запросом на актуальные данные в случае реконфигурации.

Недостатком является потенциально переменная величина раздела ЛД на разных ПУ вследствие различного объема контекстных данных у разных подзадач и разного количества подзадач, данные которых требуется на разных ПУ.

Второй способ хранения/предоставления контекстных данных предполагает организацию хранения контекстных данных в разделе подзадачи. При этом на ЛД возлагаются задачи управления процессами обновления данных как в процессе синхронизации данных, так и в процессе реконфигурации.

Достоинством такого способа является сохранение размера раздела ЛД постоянным и не зависящим от объемов контекстных данных.

Недостатком данного способа является необходимость выделения неактивным экземплярам подзадач квантов процессорного времени с целью обновления локальных копий контекстных данных. Данный способ является промежуточным в плане нагрузки на ЛД/подзадачу.

Третий способ хранения/предоставления контекстных данных предполагает полное делегирование функций обеспечения работы распределенного хранения контекстных данных подзадаче.

В этом случае контекстные данные хранятся в разделе подзадачи, все функции по обеспечению их сохранения/предоставления делегированы подзадаче, что определяет основной недостаток такого метода: программный код подзадачи усложняется и, помимо этого, как и в предыдущем способе, неактивным экземплярам подзадач требуется выделить кванты процессорного времени. Также всем экземплярам одной подзадачи необходимо предоставить данные об их расположении.

С учетом перечисленных достоинств и недостатков целесообразно реализовать первый способ хранения/предоставления контекстных данных с учетом особенностей организации вычислений в соответствии со стандартом ARINC653. При этом у каждой подзадачи, требующей при реконфигурации предоставления контекстных данных, необходимо предусмотреть два дополнительных порта для отправки и получения таких данных. Такие порты используются для информационного обмена между разделами. Каждый входной порт может иметь буфер для временного хранения полученных данных, что дает возможность организации хранения контекстных данных непосредственно в порте и отсутствии необходимости организации специальных буферов в разделе ЛД.

Таким образом, в процессе своей работы активный экземпляр подзадачи через порт отправки контекстных данных передает их ЛД, который в соответствии с конфигурационными данными определяет расположение неактивных экземпляров данной подзадачи, передает полученные данные ЛД соответствующих ПУ, которые, в свою очередь, помещают их в порт получения контекстных данных неактивного экземпляра подзадачи. В случае реконфигурации экземпляра подзадачи, ставший активным, считывает контекстные данные из порта получения и выполняет необходимые действия для начала нормальной работы.

3.5.6. Организация программных средств локального диспетчера процессорного узла вычислительной подсистемы

На основе описанных выше подходов был разработан прототип программных средств ЛД для ВП СУ на базе ИМА. В его состав входят:

- программная модель для формирования наборов конфигураций с учетом возможных отказов. В ВП СУ на базе ИМА необходимо заранее подго-

товить набор всех возможных конфигураций, т.к. в КС при инициализации системы должны быть сформированы все необходимые связи с учетом всех вариантов перераспределения подзадач;

- локальный диспетчер, выполняющий мониторинг и реконфигурацию отдельного ПУ и ВП СУ на базе ИМА в целом путем взаимодействия с ЛД других ПУ.

Программная модель формирования допустимых конфигураций ВП СУ на базе ИМА позволяет осуществлять:

- моделирование отказов ПУ ВП СУ;
- сбор статистической информации о процессе моделирования отказов ПУ ВП СУ;
- моделирование отказов коммутаторов коммуникационной среды ВП СУ;
- сбор статистической информации о процессе моделирования отказов коммутаторов коммуникационной среды ВП СУ.

В процессе моделирования отказов ПУ ВП СУ осуществляется перебор всех допустимых типов отказов и выдача в качестве результата допустимых конфигураций перераспределения (реконфигурации) подзадач ВП СУ на оставшихся работоспособных ПУ. В случае невозможности создания подобной конфигурации программная модель выдает соответствующее предупреждение пользователю.

Кроме указанных выше основных функций программная модель обладает дополнительным функционалом, облегчающим проведение экспериментальных исследований в заявленной предметной области. К их числу относятся:

- визуализация структуры бортовой ВП СУ в различные моменты времени (при различных вариантах отказов как ПУ ВП СУ, так и коммутаторов коммуникационной среды ВП СУ);
- визуализация информационных потоков (коммуникационных связей) между узлами и коммутаторами РИУС;
- выдача статистической информации о процессе моделирования (на экран и в файл) в удобной форме.

Основными задачами программной модели являются:

- генерация данных для реконфигурации распределенной ВП СУ на базе ИМА в случае выхода из строя части ее узлов;
- создание графа избыточных связей для повышения надежности системы в случае отказов ее частей;
- создание маршрутной карты для передачи информации в ВП СУ на базе ИМА посредством узлов-посредников в случае выхода из строя частей коммуникационной среды, а также для анализа эффективности начальной конфигурации ВП СУ на базе ИМА.

Локальный диспетчер представляет собой приложение, запускаемое на отдельном компьютере под управлением ОС Windows, он обеспечивает мо-

делирование в приближенном к реальному масштабу времени процессов, происходящих при работе локального диспетчера в ходе распределенного диспетчирования работы ВП СУ на базе ИМА. Его задачами являются:

- запуск на выполнение подзадач ЗУ на «своем» ПУ в соответствии с конфигурационными данными, полученными от модели формирования наборов конфигураций, а также исходными данными каждой подзадачи (готовятся в процессе разработки подзадачи и интеграции подзадач в рамках ЗУ ИМА);

- взаимодействие с ЛД других ПУ в соответствии с конфигурационными данными в целях мониторинга «работы» других ПУ и фиксации их «отказов»;

- выполнение реконфигурации «своего» в случае обнаружения отказов других ПУ в соответствии с конфигурационными данными, полученными от модели формирования наборов конфигураций;

- отображение диагностической информации, отражающей ход моделирования работы ВП СУ на базе ИМА.

Работа с прототипом программных средств ЛД осуществляется следующим образом:

- 1) определение начальной конфигурации ВП СУ на базе ИМА и графа задачи, представление их описания в формате программной модели;

- 2) генерация посредством программной модели допустимых конфигураций ВП СУ на базе ИМА для заданных количества и видов отказов;

- 3) настройка при помощи полученных конфигураций локальных диспетчеров;

- 4) загрузка исходных данных подзадач;

- 5) запуск прототипа программных средств ЛД на выполнение задачи управления.

Общая схема совместной работы программной модели создания допустимых конфигураций ВП СУ на базе ИМА и прототипа программных средств мультиагентного диспетчера приведена на рисунке 3.44.

Определение начальной конфигурации ВП СУ на базе ИМА и графа задачи выполняется с помощью программной модели. Главное окно приложения показано на рисунке 3.45.

Работа с моделью начинается с загрузки описания начальной конфигурации ВП СУ на базе ИМА, содержащей данные:

- о количестве и производительности узлов в полностью исправной системе;

- о количестве коммутаторов, каналах связи между узлами и коммутаторами и пропускной способности этих каналов;

- о количестве и трудоемкости подзадач задачи управления, связях между ними и объемах передаваемых данных.

Описание конфигурации может быть подготовлено в виде текстового файла, кроме того, предусмотрена возможность импорта/экспорта конфи-

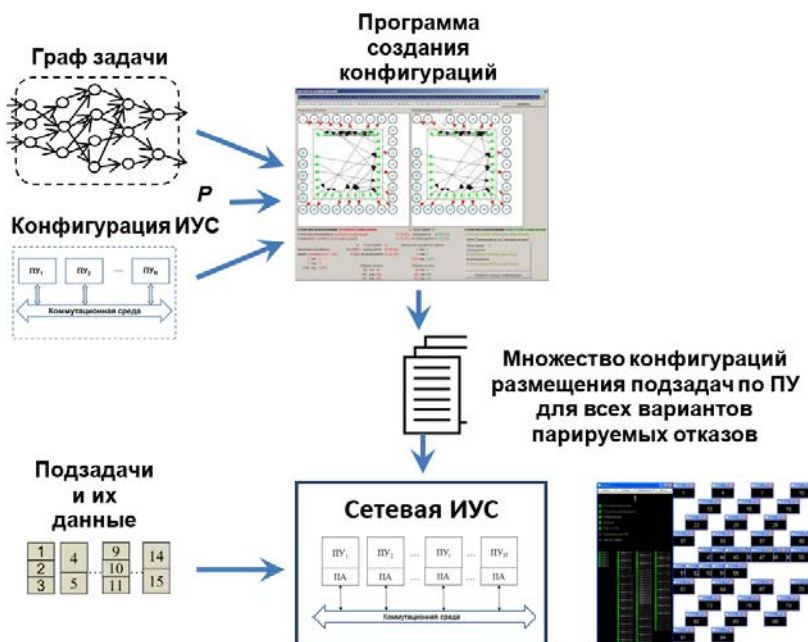


Рис. 3.44. Схема подготовки конфигураций ВП СУ на базе ИМА с использованием прототипа программных средств ЛД

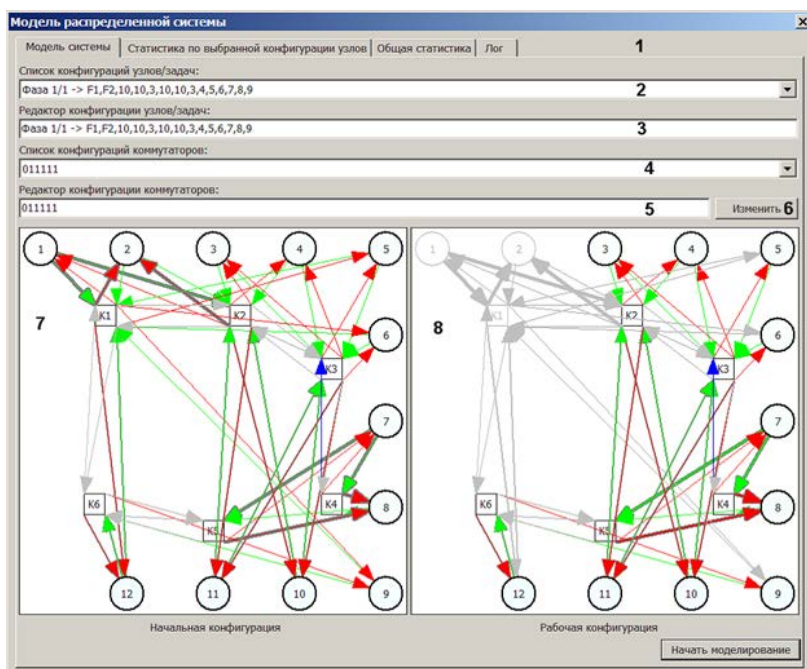


Рис. 3.45. Окно вывода информации о конфигурациях ВП СУ в случае отказа части системы

После этого выполняется генерация конфигураций, каждая из которых соответствует отказу определенного числа узлов, которое задается в конфигурационном файле. Каждая из конфигураций определяет порядок перераспределения задач с вышедших из строя узлов.

После окончания генерации вариантов реконфигурации списки конфигураций для случаев отказов узлов и коммутаторов выводятся пользователю.

Для настройки локальных диспетчеров полученные конфигурационные данные записываются в заданные каталоги на узлы, на которых размещены

ЛД. Для обеспечения корректной работы подзадач ЗУ в каталоги их размещения записываются исходные данные, зависящие от реализуемых подзадачей функций. Моделирование работы ВП СУ на базе ИМА выполняется в среде ОС Windows с возможностью взаимодействия с прототипами диспетчера для ОС на базе ARINC 653. ЛД обеспечивает реализацию алгоритмов, приведенных в пункте 3.3.3 с учетом особенностей ИМА (пункт 3.5.3). На рисунке 3.47 приведена его экранная форма, в которой отображается информация о текущем состоянии системы в целом (запущена, выполняется размещение и т.д.), отдельно каждого локального диспетчера (агента), а также размещение и состояние подзадач задачи управления.



Рис. 3.47. Экранная форма прототипа программных средств мультиагентного диспетчера

Исследования, проведенные с применением прототипа программных средств локального диспетчера, подтвердили работоспособность предложенных алгоритмов как в части формирования конфигураций, так и в части организации децентрализованного диспетчирования. В ходе исследований моделировалась работа ВП СУ на базе ИМА с числом ПУ до 30 и числом подзадач до 60.

Было показано, что:

- возможно формирование наборов конфигураций как при использовании скользящего резервирования, так и резервирования производительности;
- при меньшем числе ПУ резервирование производительности до 2–х раз более эффективно по необходимому для обеспечения реконфигурации ВП СУ на базе ИМА без перезагрузки коммуникационного оборудования количеству виртуальных связей в КС, при большем числе ПУ оба способа резервирования сопоставимы по данному показателю;
- разработанные алгоритмы обеспечивают согласованное принятие решений коллективом локальных диспетчеров, надежно функционируют в условиях меняющихся внешних условий, обеспечивают перераспределение подзадач в случае возникновения отказов.

3.5.7. Резюме

В результате проведенных исследований разработаны методы повышения надежности ВП СУ ВС с архитектурой ИМА, основанные на предложенном выше способе резервирования производительности и децентрализованном диспетчировании ресурсов и учитывающие особенности ОС РВ по стандарту ARINC 653 и КС по стандарту ARINC 664.

Разработаны программные средства ЛД, реализующие разработанные методы и алгоритмы децентрализованного диспетчирования работы ВП СУ ВС и ее реконфигурацию при возникновении отказов. Также программные средства могут применяться в технологической цепочке проектирования ЛД ВП СУ ВС на основе языка описания архитектуры AADL, обеспечивая при этом повышение уровня автоматизации за счет автоматического дополнения отказонеустойчивой структуры ВП до отказоустойчивой необходимыми компонентами и связями.

ЗАКЛЮЧЕНИЕ

Подводя итог, отметим, что применение разработанных алгоритмов реконфигурации позволяет обеспечить отказоустойчивость СУ ВС в случае возникновения отказов в вычислительной и исполнительной подсистемах, а как следствие – повысить безопасность полетов и сократить количество авиационных происшествий.

Особенность полученных результатов заключается в том, что реализация разработанных алгоритмов в СУ современных и перспективных ВС позволит обеспечить максимально достижимый уровень пилотажных характеристик для состояния, обусловленного отказами, в интересах безопасного продолжения и завершения полета.

Приведем основные итоги и рекомендации по использованию предлагаемых подходов в решении практических задач.

Реконфигурация исполнительной подсистемы СУ. Разработан аналитический метод реконфигурации СУ ВС при отказах исполнительной подсистемы, основанный на едином алгебраическом подходе к синтезу точных, оптимальных и робастных законов управления. В основу метода положены разработанные авторами методы аналитического решения линейных матричных уравнений как в детерминированной постановке, так и с учетом структурных и параметрических неопределенностей.

Канонические методы, используемые при анализе систем и синтезе управления, могут быть применены как для решения задач реконфигурации СУ, так и для решения других задач теории систем. Принципы, положенные в основу канонического метода решения матричных уравнений, составляют основу получения оптимальных и робастных решений в аналитическом виде за счет использования только алгебраических (арифметических) операций.

При получении приближенных решений матричных уравнений, в которых не удастся обеспечить выполнение условия разрешимости, удалось сохранить принцип эквивалентности преобразований, обеспечивающий минимальную параметризацию и аналитичность получаемых решений. При этом элементы канонического разложения определяются с помощью сингулярного разложения, а эквивалентные преобразования реализуются с помощью комбинации ортогональных преобразований и операций взвешивания. В результате все достоинства алгоритмов точного решения сохраняются и при описании всего множества приближенных решений, оптимальных в смысле минимума нормы ошибок.

Использование единого подхода к алгебраическим (основанным на каноническом разложении) и численным (основанным на сингулярном разложении) преобразованиям, позволяет осуществлять синтез точных (символьных), оптимальных и робастных решений с учетом требований к точности решения.

Использование единого аналитического подхода к решению задач реконфигурации ИП позволяет сформировать целостную методологию обеспечения отказоустойчивости СУ при различных постановках задачи, неопределенностях моделей ВС и возмущений. Эффективность реконфигурации ИП СУ достигается за счет предлагаемой логики переключения активных и пассивных режимов работы системы реконфигурации в зависимости от требуемой степени точности и робастности решений, определяемой условиями разрешимости задачи реконфигурации и достоверности информации о параметрах ВС с работоспособной и не работоспособной ИП СУ.

Предлагаемый подход является наиболее общим и позволяет получить множество реализаций законов реконфигурации, среди которых можно выбрать варианты с минимальным числом элементов, т.е. более простые в реализации схемы. Кроме того, он не требует учета логики функционирования штатных СУ и может работать совместно с СУ любой сложности.

Изложенный аналитический подход не предполагает выполнения в бортовых алгоритмах матричных операций при решении задачи реконфигурации и может быть реализован в реальном времени в виде классической структурной схемы.

Полученные аналитические зависимости коэффициентов законов реконфигурации от параметров полета допускают их настройку по режимам полета без повторного вычисления матриц реконфигурации на каждом шаге, т.е. практически решается проблема реконфигурации при нестационарной модели самолета.

Алгоритмы априорной оценки условий разрешимости и точности решений задач реконфигурации с учетом ограничений на отклонение поверхностей управления позволяют снижать энергетические затраты на их отклонение при допустимом и контролируемом снижении качества реконфигурации управления.

Алгоритм выбора единой числовой матрицы робастной реконфигурации, обеспечивающей удовлетворительное качество управления для всех режимов полета самолета без коррекции передаточных чисел, обеспечивает достаточные свойства робастности управления во всех режимах полета.

Для всех исследуемых сценариев построены оптимальные законы реконфигурации управления. Анализ результатов позволяет сформировать перечень критических с точки зрения безопасности полета самолета отказов по степени возрастания совокупных ошибок изменений аэродинамических параметров:

- отказ всех рулей высоты;
- отказ одного из элеронов;
- отказ стабилизатора;
- отказ руля направления;
- отказ одной секции интерцепторов.

Исследования показали, что существуют широкие возможности по ре-

конфигурации ИП СУ самолетов при отказах рулей высоты и элеронов. Отказы стабилизатора, руля направления и интерцепторов являются самыми сложными для компенсации вследствие недостаточной эффективности работоспособных поверхностей управления и ограничений на их максимальные отклонения. Для повышения безопасности полета при возникновении данных отказов целесообразно рассмотреть возможность использования для реконфигурации управления дополнительные поверхности управления: флапероны, предкрылки, тормозные щитки и т.д. При увеличении высоты и скорости полета эффективность реконфигурации СУ повышается.

Предлагаемая в работе методология аналитической реконфигурации СУ при отказах ИП в дальнейшем существенно облегчит применение аналитического подхода для построения активно-адаптивных и интеллектуальных СУ, обеспечивающих компенсацию динамических (постепенных и перемежающихся) компонентных отказов. В последствие предполагается расширить область применения разработанных подходов для моделей ВС с неполностью доступным измерению вектором состояния, в том числе и безынерционных. Развитие аналитических методов решения задач, а также получения решения с заданной точностью позволит решать многокритериальные задачи оптимизации с учетом возмущений, требований на устойчивость, а также ограничений на амплитуду и скорость отклонения поверхностей управления.

Реконфигурация вычислительной подсистемы СУ. Показано, что современные ВП СУ ВС строятся с использованием ИМА. Предложен новый способ обеспечения надежности таких ВП – способ резервирования производительности, позволяющий обеспечить требуемые характеристики надежности без существенного увеличения аппаратных затрат (без введения в состав ВП СУ дополнительных ПУ), а только за счет создания необходимого резерва производительности в имеющихся ПУ.

Показано, что способ резервирования производительности позволяет использовать все ресурсы работоспособных ПУ для улучшения характеристик ВП СУ, поскольку не требуется выделять специальные резервные ПУ, которые в процессе работы СУ ВС не участвуют в решении задачи управления.

Получены выражения, определяющие резерв производительности отдельных ПУ, необходимый для достижения заданного уровня отказоустойчивости ВП СУ.

Получены выражения, определяющие зависимость вероятности безотказной работы отдельного ПУ, входящего в состав ВП СУ, от температуры его нагрева, которая, в свою очередь, зависит от вычислительной нагрузки данного ПУ. Показано, что чем более равномерно вычислительная нагрузка будет распределена между всеми ПУ, входящими в состав ВП, тем выше будет вероятность ее безотказной работы.

Проведено исследование метода повышения надежности ВП СУ за счет выравнивания вычислительной нагрузки отдельных ПУ, показавшее, что

использование данного метода позволяет повысить надежность характеристики ВП.

Показано, что для организации отказоустойчивого функционирования ВП СУ с использованием способа резервирования производительности ПУ и реконфигурации при возникновении отказов необходимо обеспечить выполнение процедуры перераспределения подзадач задачи управления на исправных ПУ. Данная процедура не тривиальна и требует применения специального ПУ, выполняющего роль диспетчера, что ухудшает не только массогабаритные, но и надежность характеристики ВП СУ.

Предложен способ распределенной организации диспетчера с помощью множества ЛД, физически реализуемых на отдельных ПУ ВП СУ и совместно участвующих в процессе диспетчирования ресурсов ВП на принципах мультиагентного взаимодействия. Данный подход лишен перечисленных выше недостатков, поскольку не требует ввода в состав ВП СУ дополнительных устройств.

Проведен анализ особенностей функционирования ВП с распределенным диспетчером при различных способах организации вычислительного процесса в ВП СУ ВС и получены оценочные выражения, определяющие временные затраты на реализацию основных режимов диспетчирования (тестирование, цикл управления, восстановление вычислительного процесса). Для каждого из режимов разработан алгоритм работы ЛД, реализуемый на отдельном ПУ.

Дана формальная постановка задачи размещения подзадач задачи управления по процессорным узлам и сформулированы необходимые условия ее допустимого решения, а именно, допустимое решение задачи размещения должно обеспечивать время решения задачи управления менее заданного времени цикла управления, а также отклонение нагрузки отдельных ПУ от равномерной менее заданного порогового значения.

Показано, что решение задачи размещения должно выполняться за время, не превышающее время восстановления вычислительного процесса. Для сокращения времени решения задачи размещения предложено распараллеливать процесс ее решения с помощью множества ЛД отдельных ПУ.

Разработан ряд алгоритмов размещения при различных способах организации вычислительного процесса в ВП СУ ВС, обеспечивающих достижение допустимого размещения и ориентированных на параллельную реализацию с помощью множества ЛД.

Разработаны методы повышения надежности ВП СУ ВС с архитектурой ИМА, основанные на предложенном выше способе резервирования производительности и децентрализованном диспетчировании ресурсов и учитывающие особенности ОС РВ по стандарту ARINC 653 и КС по стандарту ARINC 664.

Разработаны программные средства ЛД, реализующие разработанные методы и алгоритмы децентрализованного диспетчирования работы ВП СУ

ВС и ее реконфигурацию при возникновении отказов. Также программные средства могут применяться в технологической цепочке проектирования ЛД ВП СУ ВС на основе языка описания архитектуры AADL, обеспечивая при этом повышение уровня автоматизации за счет автоматического дополнения отказонеустойчивой структуры ВП до отказоустойчивой необходимыми компонентами и связями.

СПИСОК ЛИТЕРАТУРЫ

1. *Акимов А.Н., Воробьев В.В., Коноплев Ю.К., Шабалин В.А.* Отказоустойчивость систем управления летательных аппаратов. М.: ВВИА им. Н. Е. Жуковского, 2005. 142 с.
2. Анализ состояния безопасности полетов в гражданской авиации Российской Федерации в 2018 году. М.: Росавиация, 2019. 89 с.
3. Антимиров В. М., Трапезников М. Б. Поколения бортовых цифровых вычислительных систем // Вопросы атомной науки и техники. Серия: Физика радиационного воздействия на радиоэлектронную аппаратуру. 2012. № 2. С. 38–46.
4. *Барский А.Б.* Планирование параллельных вычислительных процессов. М.: Машиностроение, 1980. 192 с.
5. *Буков В.Н., Рябченко В.Н., Косьянчук В.В., Зыбин Е.Ю.* Решение линейных матричных уравнений методом канонизации // Сборник тезисов докладов Международной математической конференции, посвященной столетию начала работы Д. А. Граве (1863–1939) в Киевском университете. Киев: Изд. Киевского нац. ун-та, 2002. С.73–74.
6. *Буков В.Н., Рябченко В.Н., Косьянчук В.В., Зыбин Е.Ю.* Решение линейных матричных уравнений методом канонизации // Вестник Киевского университета. Серия: Физико-математические науки. 2002 № 1. С. 19–28.
7. *Вавилов Ю.А.* Системы автоматического управления полетом. М.: ВВИА им. проф. Н. Е. Жуковского, 2009. 412 с.
8. *Валях Е. Николаев И.А., Степанов А.М.* Последовательно-параллельные вычисления. М.: Мир, 1985. 456 с.
9. *Вежбицкий В.М.* Численные методы (линейная алгебра и нелинейные уравнения): Учеб. пособие для вузов. М.: Высш. шк., 2000. 266 с.
10. *Воеводин В.В., Воеводин Вл.В.* Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 599 с.
11. *Головкин Б.А.* Классификация методов диспетчеризации работы многопроцессорных и многомашинных вычислительных систем // Управляющие машины и системы. 1982. № 3. С. 3–11.
12. *Головкин Б.А.* Расчет характеристик и планирование параллельных вычислительных процессов. М.: Радио и связь, 1983. 272 с.
13. *Горлов М., Строгонов А.* Геронтология кремниевых интегральных схем [Электронный ресурс], URL: http://www.chipnews.ru/html.cgi/arhiv_i/00_02/stat_57.htm (дата обращения 16.09.2020).
14. *Захаров В.А., Козмидиади В.А., Кузьмин А.В., Попов А.С., Шулятников Д.С.* Планирование выполнения заданий сервисных приложений в распределенной среде // Системы и средства информатики. 2008. № 18. С. 36–48.
15. *Земляков С.Д., Рутковский В.Ю., Силаев А.В.* Реконфигурация систем управления летательными аппаратами при отказах // Автоматика и телемеханика. 1996. № 1. С. 3–20.

16. *Зыбин Е.Ю., Косьянчук В.В.* Аналитический синтез многосвязных отказоустойчивых систем управления с упрощённой схемной реализацией // Известия РАН. Теория и системы управления. 2010. № 1. С. 108–117.
17. *Зыбин Е.Ю., Косьянчук В.В., Кульчак А.М.* Аналитическое решение задачи оптимальной реконфигурации системы управления летательного аппарата при отказе нескольких органов управления // Мехатроника, автоматизация, управление. 2014. № 7. С. 59–66.
18. *Зыбин Е.Ю., Косьянчук В.В., Сельвесюк Н.И., Сапожников А.В., Кульчак А.М.* Робастная к изменению режима полета реконфигурация управления воздушным судном при отказах приводов // Материалы Всероссийской научно-технической конференции «XI Научные чтения, посвященные памяти Н.Е. Жуковского»: сборник докладов. М.: Издательский дом Академии имени Н. Е. Жуковского. 2014. С. 331–336.
19. *Зыбин Е.Ю., Мисриханов М.Ш., Рябченко В.Н.* О минимальной параметризации решений линейных матричных уравнений // Вестник ИГЭУ. 2004. № 6. С. 127–131.
20. *Каляев И.А.* Децентрализованные системы компьютерного управления: монография / И.А. Каляев, Э.В. Мельник. Ростов н/Д: Изд-во ЮНЦ РАН, 2011. 196 с.
21. *Каляев И.А.* Итерационная процедура оптимизации планирования решения связанной задачи на кластере компьютеров с применением технологии мобильных агентов / И.А. Каляев, Э.В. Мельник, В.В. Волков, Ли Джилин // Интеллектуальные роботы ИР–2002: материалы Первой научн. молод. школы. Таганрог: Изд-во ТРТУ. 2002. С. 252–254.
22. *Каляев И.А.* Организация распределенных вычислений на основе коллективного принятия решений / И.А. Каляев, Э.В. Мельник, В.В. Волков, Ли Джилин // СуперЭВМ и многопроцессорные вычислительные системы: материалы Междунар. науч.-техн. конф. Таганрог: Изд-во ТРТУ. 2002. С. 192–195.
23. *Каляев И.А.* Реконфигурируемые мультиконвейерные вычислительные структуры / И.А. Каляев, И.И. Левин, Е.А. Семерников, В.И. Шмойлов; под общ. ред. И.А. Каляева. 2-е изд., перераб. и доп. Ростов н/Д: Изд-во ЮНЦ РАН, 2009. 344 с.
24. *Колпаков К.М.* История развития бортовых цифровых вычислительных машин в России // ITweek. 1999. № 32. [Электронный ресурс], URL: <https://www.itweek.ru/industrial/article/detail.php?ID=51809> (дата обращения: 12.09.2020).
25. *Константинов С.В., Редько П.Г., Ермаков С.А.* Электрогидравлические рулевые приводы систем управления полетом маневренных самолетов. М.: Янус-К, 2006. 315 с.
26. *Косьянчук В.В.* Обеспечение отказоустойчивости систем управления при отказе приводов управляющих поверхностей летательных аппаратов с механической проводкой управления // Проблемы безопасности полетов. 2007. № 9. С.37–42.

27. *Косьянчук В.В.* Синтез систем управления многосвязных объектов с неполным вектором управления // Труды IX Международной Четаевской конференции «Аналитическая механика, устойчивость и управление движением». Т. 3. Иркутск. 2007. С. 97–102.

28. *Крюков В.В., Майоров В.С., Шахгельдян К.И.* Алгоритм баланса нагрузки для обеспечения режима реального времени в распределенной системе сбора и обработки данных // Информационные технологии. 2004. №7. С. 11–17.

29. *Кульчак А.М., Косьянчук В.В., Зыбин Е.Ю.* Реконфигурация комплексной системы управления воздушного судна при отказах приводов с учетом ограничений на управление // Научный вестник МГТУ ГА. 2018. Т. 21. № 6. С. 65–78.

30. *Липаев В.В.* Методы обеспечения качества крупномасштабных программных средств. М.: СИНТЕГ, 2003. 520 с.

31. *Мелентьев В.А.* Моделирование систем, устойчивых к отказам заданной кратности // Труды VII Международной конференции «Идентификация систем и задачи управления» SICPRO. 2008. Т. 8. С. 1210–1223.

32. *Мельник Э.В.* Децентрализованная реализация механизмов обеспечения отказоустойчивости и взаимодействия компонентов функционального программного обеспечения в распределенных управляющих системах / Э.В. Мельник, С.С. Поливанов, С.С. Семенистый, К.В. Погорелов // Материалы 4-й Всерос. мультikonф. по проблемам управления (МКПУ-2011). Таганрог: Изд-во ТТИ ЮФУ. 2011. Т.1. С. 324–328.

33. *Мельник Э.В.* Децентрализованный алгоритм организации распределенных вычислений / Э.В. Мельник, И.В. Блушвили, С.А. Ховансков // Интеллектуальные и многопроцессорные системы – 2003: материалы Международ. науч.-техн. конф. Таганрог: Изд-во ТРТУ. 2003. Т.2. С. 27–30.

34. *Мельник Э.В.* Модифицированный децентрализованный алгоритм организации распределенных вычислений / Э.В. Мельник, И.В. Блушвили, С.А. Ховансков // Интеллектуальные и многопроцессорные системы – 2005: материалы Международ. науч.-техн. конф. Таганрог: Изд-во ТРТУ. 2005. Т.2. С. 104–106.

35. *Мельник Э.В.* Отказоустойчивая мультиагентная среда выполнения сложных распределенных программ / Э.В. Мельник, И.С. Пуха // Искусственный интеллект. Интеллектуальные и многопроцессорные системы – 2006: материалы Седьмой Международ. науч.-техн. конф. Таганрог: Изд-во ТРТУ. 2006. Т.2. С. 113–116.

36. *Мисриханов М.Ш., Зыбин Е.Ю.* Синтез нечувствительного к возмущениям закона управления боевого одновинтового вертолета // Современные методы управления многосвязными динамическими системами: сборник. Вып. 3 / Под ред. А.А. Красовского. М.: Энергоатомиздат, 2004. С. 122–131.

37. *Никифоров В.О., Слита О.В., Ушаков А.В.* Интеллектуальное управление в условиях неопределенности. СПб.: СПбГУ ИТМО, 2011. 232 с.

38. *Петрова В.Т.* Лекции по алгебре и геометрии: Учебник для вузов. В 2 ч. М.: Гуманит. изд. центр ВЛАДОС, 1999. Ч. 1. 312 с.; Ч. 2. 344 с.
39. Системы дистанционного управления магистральных самолетов / Б.С. Алешин, С.Г. Баженов, Ю.И. Диденко, Ю.Ф. Шелюхин. М.: Наука, 2013. 292 с.
40. Системы электрогидравлических рулевых приводов комплексов управления полетом самолетов / под ред. д-ра техн. наук, проф. С.В. Константинова. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2019. 518 с.
41. *Смелянский Р.Л., Костенко В.А.* Проблемы разработки итерационных алгоритмов для построения расписаний с одновременным нахождением необходимого количества ресурсов и их характеристик // Искусственный интеллект (Донецк). 2002. № 2. С. 141–150.
42. Справочник по теории автоматического управления // Под общ. ред. А.А. Красовского. М.: Наука, 1987. 712 с.
43. *Строгонов А.* Долговечность интегральных схем и производственные методы ее прогнозирования // ChipNews. 2002. № 6. С. 44–49.
44. *Тель Ж.* Введение в распределенные алгоритмы. М.: Изд-во МЦНМО, 2009. 616 с.
45. *Труханов В.М.* Новый подход к обеспечению надежности сложных систем. М.: Издательский дом «Спектр». 2010. 247 с.
46. *Уилкинсон Д., Райни К.* Справочник алгоритмов на языке АЛГОЛ. Линейная алгебра. М.: Машиностроение, 1976. 389 с.
47. *Федосов Е.А.* Проект создания нового поколения интегрированной модульной авионики с открытой архитектурой // Полет. 2008. № 8. С. 15–22.
48. *Федосов Е.А., Косьянчук В.В., Сельвесюк Н.И.* Интегрированная модульная авионика // Радиоэлектронные технологии. 2015. № 1. С. 66–71.
49. *Хорошевский В.Г.* Архитектура вычислительных систем: учеб. пособие для вузов. М.: Изд-во МГТУ им. Н. Э. Баумана, 2008. 520 с.
50. *Ченцов А.Г.* Экстремальные задачи маршрутизации и распределения заданий: вопросы теории. М., Ижевск: Регулярная и хаотическая динамика. 2008. 240 с.
51. *Ackermann J.* Multi-model approaches to robust control system design / J. Ackermann // Uncertainty and Control. Springer, Berlin, Heidelberg. 1985. P. 108–130.
52. *Adams R.J.* Robust multivariable flight control. Springer-Verlag, 1994. 164 p.
53. *Ahn C.I., Kim Y., Kim H.J.* Adaptive sliding mode controller design for fault tolerant flight control system // Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit. 2006. P. 6089.
54. *Alswailem S.I.* Application of robust control in unmanned vehicle flight control system design. PhD thesis, Cranfield University, 2004.
55. *Alwi H., Edwards C., Tan C.P.* Fault Detection and Fault Tolerant Control Using Sliding Modes. Springer Verlag, 2011. 340 p.

56. *Bajpai G., Chang B.C., Lau A.* Reconfiguration of flight control systems for actuator failures // IEEE Aerospace and Electronics Systems Magazine. 2001. Vol. 16. No. 9. P. 29–33.
57. *Bieber P., Noulard E., Pagetti C., Planche T., Vialard F.* Preliminary design of future reconfigurable IMA platforms // ACM Sigbed Review. 2009. Vol. 6. No. 3. P. 1–5.
58. *Bodson M.* Evaluation of optimization methods for control allocation // AIAA Journal of Guidance, Control, and Dynamics. 2002. Vol. 25. No. 4. P. 703–711.
59. *Bodson M.* Reconfigurable nonlinear autopilot // AIAA Journal of Guidance, Control, and Dynamics. 2003. Vol. 26. No. 5. P. 719–727.
60. *Bodson M., Pohlchuck W.A.* Command limiting in reconfigurable flight control // Journal of Guidance, Control, and Dynamics. 1998. Vol. 21. No. 4. P. 639–646.
61. *Boskovic J.D., Mehra R.K.* Multiple-model adaptive flight control scheme for accommodation of actuator failures // Journal of Guidance, Control, and Dynamics. 2002. Vol. 25. No. 4. P. 712–724.
62. *Boskovic J.D., Li S.-M., Mehra R.K.* Robust adaptive variable structure control of spacecraft under control input saturation // Journal of Guidance, Control, and Dynamics. 2001. Vol. 24. No. 1. P. 14–22.
63. *Diao Y., Passino K.M.* Stable fault-tolerant adaptive fuzzy/neural control for turbine engine // IEEE Transactions on Control Systems Technology. 2001. Vol. 9. No. 3. P. 494–509.
64. *Diao Y., Passino K.M.* Intelligent fault-tolerant control using adaptive and learning methods // Control Engineering Practice. 2002. Vol. 10. No. 8. P. 801–817.
65. *Eberhardt R.L., Ward D.* Indirect adaptive flight control system interactions // International Journal of Robust and Nonlinear Control. 1999. Vol. 9. No. 14. P. 1013–1031.
66. *Ganguli S., Marcos A., Balas G.* Reconfigurable LPV control design for Boeing 747-100/200 longitudinal axis // Proceedings of the 2002 American control conference. 2002. P. 3612–3617.
67. *Gao Z., Antsaklis P.J.* Stability of the pseudo-inverse method for reconfigurable control systems // International Journal of Control. 1991. Vol. 53. No. 3. P. 717–729.
68. *Genry V.* Thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm // Journal of Optimization problems and Applications. 1985. No. 45. P. 51–51.
69. *Gustavsson I., Ljung L., Soderstrom, T.* Identification of processes in closed loop – identifiability and accuracy aspects // Automatica. 1977. Vol. 13. No. 1. P. 59 – 75.
70. *Hajiyev C., Caliskan F.* Integrated sensor/actuator FDI and reconfigurable control for fault-tolerant flight control system design // The Aeronautical Journal. 2001. Vol. 105. No. 1051. P. 525–533.

71. *Hearkegard O.* Back stepping and control allocation with applications to flight control. PhD thesis, Linköping University, Sweden, 2003.
72. *Ho L.-W., Yen G.G.* Reconfigurable control system design for fault diagnosis and accommodation // *International Journal of Neural Systems*. 2002. Vol. 12. No. 6. P. 497–520.
73. *Jiang J., Zhang Y.M.* Accepting performance degradation in fault tolerant control system design // *IEEE Transactions on Control Systems Technology*. 2006. Vol. 14. No. 2. P. 284–292.
74. *Kale M.M., Chipperfield A.J.* Stabilized MPC formulations for robust reconfigurable flight control // *Control Engineering Practice*. 2005. Vol. 3. No. 6. P. 771–788.
75. *Kapila V., Grigoriadis K.* Actuator saturation control. CRC Press, 2002. 320 p.
76. *Kim K.-S., Lee, K.-J., Kim Y.* Reconfigurable flight control system design using direct adaptive method // *Journal of Guidance, Control, and Dynamics*. 2003. Vol. 26. No. 4. P. 543–550.
77. *Konstantopoulos I. K., Antsaklis P. J.* An optimization approach to control reconfiguration // *Dynamics and Control*. 1999. Vol. 9. No. 3. P. 255–270.
78. *Kuhn H.W.* Variants of the Hungarian method for assignment problems // *Naval Research Logistics Quarterly*. 1956. Vol. 3. No. 4 P. 253–258.
79. *Liao F., Wang J.L., Yang G.-H.* Reliable robust flight tracking control: An LMI approach // *IEEE Transactions on Control Systems Technology*. 2002. Vol. 10. No. 1. P. 76–89.
80. *Looze, D.P., Weiss, J.L., Eterno, J.S., Barrett, N.M.* An automatic design approach for restructurable control systems // *IEEE Control Systems Magazine* 1985. Vol. 5. No. 2. P. 16–22.
81. *Lopez-Toribio C.J., Patton R.J., Daley S.* Takagi-Sugeno fuzzy fault-tolerant control of an induction motor // *Neural Computing and Applications*. 2000. Vol. 19. No. 1. P. 19–28.
82. *Maciejowski J.M.* Modeling and predictive control: Enabling technology for reconfiguration // *Annual Reviews in Control*. 1999. Vol. 23. P. 13–23.
83. *Magni J.-F., Bennani S., and Terlouw J.* Robust Flight Control: A design challenge. Springer-Verlag, 1997. 649 p.
84. *Maybeck P.S.* Multiple model adaptive algorithms for detecting and compensating sensor and actuator/surface failures in aircraft flight control systems // *International Journal of Robust and Nonlinear Control*. 1999. Vol. 9. No. 14. P. 1051–1070.
85. *Mercadante R., Wheeler P.N.* Control reconfigurable combat aircraft piloted simulation development // *Proc. IEEE National Aerospace and Electronics Conf.* 1988. Vol. 2. P. 512–519.
86. *Mhaskar P., Gani A., Christofides P.D.* Fault-tolerant control of nonlinear processes: Performance-based reconfiguration and robustness // *International Journal of Robust and Nonlinear Control*. 2006. Vol. 16. No. 3. P. 91–111.

87. *Musgrave J. L., Guo T.-H., Wong E., Duyar, A.* Real-time accommodation of actuator faults on a reusable rocket engine // *IEEE Transactions on Control Systems Technology*. 1997. Vol. 5. No. 1. P. 100–109.
88. *Niksefat N., Sepeshri N.* A QFT fault-tolerant control for electro hydraulic positioning systems // *IEEE Transactions on Control Systems Technology*. 2002. Vol. 10. No. 4. P. 626–632.
89. *Pachter M., Chandler P.R., Mears M.* Reconfigurable tracking control with saturation // *Journal of Guidance, Control, and Dynamics*. 1995. Vol. 18. No. 5. P. 1016–1022.
90. *Patton R.J.* Fault-tolerant control: the 1997 situation // *IFAC Proceedings Volumes*. 1997. Vol. 30. No. 18. P. 1029–1051.
91. *Puig V., Quevedo J.* Fault-tolerant PID controllers using a passive robust fault diagnosis approach // *Control Engineering Practice*. 2001. Vol. 9. No. 11. P. 1221–1234.
92. *Rattan K.S.* Evaluation of control mixer concept for reconfiguration of flight control system // *Proc. IEEE National Aerospace and Electronics Conf.* 1985. P. 560–569.
93. *Rotondo D., Ponsart J.-C., Theilliol D., Nejari F., Puig V.* A virtual actuator approach for the fault tolerant control of unstable linear systems subject to actuator saturation and fault isolation delay // *Annual Reviews in Control*. 2015. Vol. 39. P. 68–80.
94. *Sheynin Y.* Distributed Interrupts for Real-Time Control in Space Wire-Based On-Board Systems / Y. Sheynin, S. Gorbachev. 9th Military and Aerospace Programmable Logic Devices (MAPLD) International Conference. Space Wire Seminar. September 26–28, 2006, Washington.
95. *Shin J.Y., Wu N.E., Belcastro C.* Adaptive linear parameter varying control synthesis for actuator // *Journal of Guidance, Control, and Dynamics*. 2004. Vol. 27. No. 4. P. 787–794.
96. *Shtessel Y., Buffington J., Banda S.* Multiple timescale flight control using reconfigurable sliding modes // *Journal of Guidance, Control, and Dynamics*. 1999. Vol. 22. No. 6. P. 873–883.
97. *Stengel R.F.* Intelligent failure-tolerant control // *IEEE Control Systems*. 1991. Vol. 11. No. 4. P. 14–23.
98. *Stoustrup J., Zhou K.* Robustness Issues in Fault Diagnosis and Fault Tolerant Control // *Journal of Control Science and Engineering*. 2008. Vol. 2008. P. 1–3.
99. Sukhoi SuperJet. Реальность против домыслов [Электронный ресурс], URL: <http://superjet.wikidot.com/wiki:aircraft-control> (дата доступа 06.12.2020).
100. *Tao G., Chen S., Joshi S.M.* An adaptive actuator failure compensation controller using output feedback // *IEEE Transactions on Automatic Control*. 2002. Vol. 47. No. 3. P. 506–511.
101. *Tao G., Chen S., Tang X., Joshi, S.M.* Adaptive control of systems with actuator failures. Springer Science & Business Media, 2013.

102. UAV Navigation AP04 autopilot [Электронный ресурс], URL: <https://www.uavnavigation.com/support/kb/autopilots/ap04> (дата обращения: 16.09.2020).
103. *Vaglienti B., Hoag R., Niculescu M.* Piccolo System User's Guide // Cloud Cap Technologies [Электронный ресурс], URL: <http://cloudcaptech.com> (дата обращения: 16.09.2020).
104. *Van den Hof P. M. J., Schrama R. J. P.* Identification and control – closed-loop issues // *Automatica*. 1995. Vol. 31. No. 12. P. 1751–1770.
105. *Van der Veen G., van Wingerden J.-W., Bergamasco M., Lovera M., Verhaegen M.* Closed-loop subspace identification methods: an overview // *IET Control Theory & Applications*. 2013. Vol. 7. No. 10. P. 1339–1358.
106. *Wang Q., Li Q., Cheng N., Song J.* Robust control allocation method in the presence of control effector failure // 2014 IEEE International Conference on Information and Automation (ICIA). 2014. P. 660–664.
107. *Wills L., Kannan S., Sander S., Guler M., Heck B., Prasad, J.V.R., Schrage D., Vachtsevanos G.* An open platform for reconfigurable control // *IEEE Control Systems Magazine*. 2001. Vol. 21. No. 3. P. 49–64.
108. *Yang G.-H., Wang J.L., Soh Y.C.* Reliable H1 controller design for linear systems // *Automatica*. 2001. Vol. 37. No. 5. P. 717–725.
109. *Yang G.-H., Wang J.L., Soh Y.C.* Reliable LQG control with sensor failures // *IEE Proceedings – Control Theory and Applications*. 2000. Vol. 147. No. 4. P. 433–439.
110. *Yang Z., Blanke M.* The robust control mixer module method for control reconfiguration // *Proceedings of the 2000 American control conference*. 2000. P. 3407–3411.
111. *Yu X., Jiang J.* A survey of fault-tolerant controllers based on safety-related issues // *Annual Reviews in Control*. 2015. Vol. 39. P. 46–57.
112. *Zhang Y.M., Jiang J.* An active fault-tolerant control system against partial actuator failures // *IEE Proceedings – Control Theory and Applications*. 2002. Vol. 149. No. 1. P. 95–104.
113. *Zhang Y.M., Jiang J.* Design of restructurable active fault-tolerant control systems // *IFAC Proceedings Volumes*. 2002. Vol. 35. No. 1. P. 101–106.
114. *Zhang Y.M., Jiang J.* Fault tolerant control system design with explicit consideration of performance degradation // *IEEE Transactions on Aerospace and Electronic Systems*. 2003. Vol. 39. No. 3. P. 838–848.
115. *Zhang Y., Jiang J.* Bibliographical review on reconfigurable fault-tolerant control systems // *Annual Reviews in Control*. 2008. Vol. 32. No. 2. P. 229–252.
116. *Zhou K., Ren Z.* A new controller architecture for high performance, robust, and fault-tolerant control // *IEEE Transactions on Automatic Control*. 2001. Vol. 46. No. 10. P. 2688–2693.
117. *Zuo Z., Ho D., Wang Y.* Fault tolerant control for singular systems with actuator saturation and nonlinear perturbation // *Automatica*. 2010. Vol. 46. No. 3. P. 569–576.

Желтов С.Ю., Каляев И.А., Косьянчук В.В.,
Мельник Э.В., Зыбин Е.Ю.

РЕКОНФИГУРАЦИЯ СИСТЕМ УПРАВЛЕНИЯ ВОЗДУШНЫХ СУДОВ

Формат 70х100 1/16
Гарнитура Times
Усл.-п. л. 16,58. Уч.-изд. л. 9,4
Тираж 300 экз.

Издатель – Российская академия наук

Публикуется в авторской редакции

Корректура и верстка – ООО «ВИН»

Отпечатано в экспериментальной цифровой типографии РАН

Издается по решению Научно-издательского совета
Российской академии наук (НИСО РАН) от 12.02.2021 г.
и распространяется бесплатно