



MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest





MAGYAR TUDOMÁNAOS AKADÉMIA  
STÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATO INTÉZETE

ГИБКИЕ АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ:  
НАДЕЖНОСТЬ И ПРОБЛЕМЫ УПРАВЛЕНИЯ

Д-р Кочиш Я., Фетисов В.А.

Tanulmányok 162/1984.

A kiadásért felelős:

*Dr. VAMOS TIBOR*

ISBN 963 311 181 1

ISSN 0324 - 2951

С О Д Е Р Ж А Н И Е

Введение .....	5
1. Концепция гибких автоматизированных систем .....	7
2. Анализ надежности функционирования ГАП .....	21
2.1. Надежность средств вычислительной техники .....	21
2.1.1. Самоустраняющиеся отказы и сбои .....	21
2.1.2. Искажения информации в системах передачи данных .....	24
2.1.3. Искажения данных при накоплении и хранении в памяти вычислительных систем .....	26
2.2. Надежность программного обеспечения .....	28
2.2.1. Основные понятия теории надежности комплексов программ .....	28
2.2.2. Критерии оценки надежности функционирования комплексов программ .....	32
2.2.3. Возмущения, влияющие на надежность программно- го обеспечения .....	39
3. Модели планирования и управления функционирова- нием ГАП .....	42
3.1. Информационная модель технологического участка ..	42
3.2. Модели календарного планирования .....	47
3.2.1. Система ограничений .....	48
3.2.2. Критерии качества .....	49
3.2.3. Методы решения задачи календарного планиро- вания .....	52
3.3. Модель управления технологическим участком .....	55
3.4. Внутрисменное оперативное управление .....	60
4. Пути повышения надежности ГАП .....	63
4.1. Временное резервирование технических систем .....	63
4.1.1. Источники временной избыточности .....	63
4.1.2. Отказы систем с временной избыточностью .....	66
4.1.3. Основные критерии и характеристики .....	67
4.2. Обеспечение надежности комплексов программ .....	72
4.2.1. Методы повышения надежности функционирования программ .....	72

4.2.2. Резервирование программ и данных .....	76
4.2.3. Временная избыточность программного обеспечения .....	79
5. Методика расчета временной избыточности .....	82
5.1. Количественный анализ ГАП с временной избыточностью .....	82
5.1.1. Отказ обесценивает работу, сделанную на всех предыдущих этапах .....	83
5.1.2. Отказ обесценивает результаты работы, сделанной на одном этапе .....	86
5.1.3. Поэтапная проверка абсолютно надежна .....	88
5.2. Способы использования временной избыточности ....	89
Заключение .....	92
Указатель литературы .....	93

ГИБКИЕ АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ:  
НАДЕЖНОСТЬ И ПРОБЛЕМЫ УПРАВЛЕНИЯ

КОЧИШ Я., ФЕТИСОВ В.А.<sup>\*</sup>

Центр по применению вычислительной техники  
Будапештского Технического Университета

Введение

Возможность создания гибких автоматизированных производств (ГАП) на основе широкого использования вычислительной техники, автоматизированного оборудования и роботоманипуляторов открывает новый этап в автоматизации промышленности. ГАП являются высокопроизводительными и дорогостоящими производственными комплексами, использование которых оказывается экономически целесообразным при условии их полной загрузки, работе в три смены, без выходных. Отсюда вытекают повышенные требования к надежности функционирования всех компонент гибких систем: оборудования, средств вычислительной техники и программного обеспечения. Наличие в составе ГАП сложных комплексов программного обеспечения систем управления и обработки информации значительно усложняют проблему обеспечения надежности. Теоретический и практический уровень современной теории надежности технических устройств достаточно высок, и ни одна сложная техническая система не проектируется без одновременного анализа ее будущей надежности. Для обеспечения заданной надежности аппаратуры применяется широкий спектр методов и средств, позволяющих из относительно ненадежных компонент создавать высоконадежные сложные системы. Искажения программ и данных

---

<sup>\*</sup>  
В настоящее время проходит стажировку в Будапештском Техническом Университете

не только возникают в связи с аномалиями работы аппаратуры (надежность таких все еще уникальных систем, какими как правило являются ГАП, хотя и высока, но все же конечна), но и могут проявляться при безотказной работе ЭВМ и систем передачи данных. Высокая ответственность функционирования ГАП обуславливает необходимость повышенного внимания к надежности их функционирования. Тот факт, что на первой международной конференции по гибким производственным системам (Брайтон, Великобритания, октябрь 1982 г.) практически отсутствовали доклады по проблемам надежности, не должен настраивать нас слишком оптимистически. На наш взгляд это объясняется вопросами пристрастия фирм, производящих автоматизированное производственное оборудование и ЭВМ, и нежеланием опубликовывать данные о реальной надежности своих изделий.

Экономические требования, предъявляемые к гибким производственным системам, предполагают не столько безотказную работу, но, главным образом, гарантированное исполнение заказов в поставленные сроки. В качестве комплексного метода, обеспечивающего повышение надежности функционирования и гарантирующего соблюдение временных (календарных) ограничений, выступает использование временного резервирования (временной избыточности).

Временная избыточность не только обеспечивает реализацию структурного резервирования и информационной избыточности, но и выступает в качестве резерва системы оперативного управления.

В этом смысле, временная избыточность обеспечивает повышение устойчивости всей системы. В данной работе авторы рассматривают те разделы проблемы обеспечения надежности функционирования ГАП, которые близки их научным интересам, и, конечно, не претендуют на полноту охвата всей проблемы.

## I. Концепция гибких автоматизированных систем

Прежде чем перейти к изложению базовых аспектов концепции ГАП формулируем основные цели, которые преследуются при создании автоматизированного производства. К этим целям относят:

- резкое повышение производительности труда в процессе изготовления единичной и мелкосерийной продукции, благодаря более высокой загрузке оборудования;
- быстрота реагирования на изменяющиеся требования заказчиков;
- быстрые темпы роста производства при нарастающем дефиците рабочей силы;
- повышение рентабельности и уменьшение объемов незавершенного производства;
- повышение качества изготавливаемой продукции и устранение ошибок и нарушений технологических режимов;
- решение социальных вопросов (освобождение человека от малоквалифицированного труда, улучшение условий труда и устранение ручных трудоемких, вредных и тяжелых операций).

Экономически с созданием и внедрением ГАП связывают повышение фондоотдачи оборудования, увеличение коэффициента сменности его работы, сокращение длительности технологического цикла.

Уже простое перечисление этих целей показывает всю серьезность современного подхода к автоматизации производства.

Гибкое автоматизированное производство - производственная единица (линия, участок, цех, небольшой завод), состоящая из производственных компонентов, гибких модулей и макромодулей, работа которых координируется с помощью ЭВМ (или сети ЭВМ) как единое целое многоуровневой системой управления.

Подобное определение с незначительными вариациями характерно как для советских авторов, так и для зарубежных [1,9].

В состав дискретной автоматизированной технологической системы входят: организационная, исполнительная и информационно-управляющая системы /рис. 1.1/, обеспечивающие выполнение производственной программы /по количеству и номенклатуре/ с заданными критериями эффективности.

Организационная система ГАП есть совокупность средств, моделей и методов, определяющая цели и критерии функционирования ГАП и обеспечивающая работу всей системы в экстремальных ситуациях, при выходе из строя информационно-управляющей системы или возникновения резких отклонений в ходе технологического процесса. Эта система реализуется на основе взаимодействия ГАП с автоматизированной системой управления производством более высокого уровня /АСУП/, автоматизированной системой научных исследований /АСНИ/, системой автоматизированного проектирования /САПР/, автоматизированной системой технологической подготовки производства /АСТПП/, отделом материально-технического снабжения /МТС/ центральным инструментальным складом /ЦИС/ и т.д.

На рис. 1.2. представлено системное окружение ГАП. В каждом конкретном случае возможны изменения в составе взаимодействующих систем или агрегирование нескольких из перечисленных систем в одну.

Исполнительная система обеспечивает выполнение всех требуемых операций согласно маршрутным технологическим процессам и состоит из совокупности производственных компонентов, гибких модулей и макромодулей. Производственными компонентами ГАП являются автоматический склад, транспортная система, участки технического контроля, комплектования инструмента, оснастки, заготовок и др.



Рис. 1.1. Структура дискретной-технологической системы

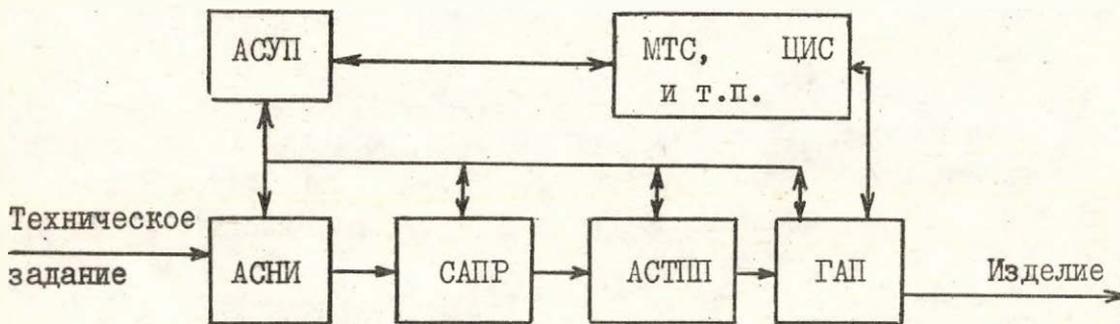


Рис. 1.2. Системное окружение ГАП

Гибкие модули ГАП - это технологическое оборудование с числовым программным управлением: программируемые автоматы, станки, автооператоры контрольно-измерительные устройства и др.

Макромодули состоят из нескольких единиц оборудования, управляемого из промежуточного устройства, например: станок с работой - манипулятором и устройством контроля.

Информационно-управляющая система предназначена для обеспечения взаимодействия ГАП с АСУП, АСНИ, САПР, АСТПП и элементами ГАП между собой в соответствии с заданными маршрутными технологическими процессами и производственной ситуацией при выполнении плана. Как правило, информационно-управляющая система реализуется в виде многоуровневой вычислительной системы или сети мини- и микро- ЭВМ. Сопряженные автоматизированные системы обеспечивают информационно-управляющую технологической информацией и управляющими программами для всех элементов исполнительной системы, а службы типа МТС и ЦИС реализуют материальные потоки: предметы труда, инструмент и др.

В настоящее время на концептуальном уровне ГАП не определены количественные критерии, позволяющие рекомендовать тот уровень "гибкости" системы, который в полной мере отвечал бы потребностям производства. Качественно, тип системы зависит от степени изменяемости целей, технологических операций и производственной среды. Малая их изменяемость обеспечивается "жесткими" системами типа автоматических линий. В табл. I. приведена качественная классификация различных производств по степени изменяемости цели, среды и технологических операций. Основное отличие ГАП от уже существующих производств - наличие гибких организационной, информационно-управляющей и исполнительной систем, характеризующихся большой изменяемостью.

Таблица I.

Классификация производств по степени изменяемости целей, среды и технологических операций

Изменяемость технологических операций	Изменяемость цели			
	малая		большая	
	изменяемость производственной среды			
	малая	большая	малая	большая
Малая	ОС-жесткая ИС-жесткая ИУС-жесткая (автоматическая линия с жестким управлением)	ОС-жесткая ИС-жесткая ИУС-гибкая (автоматическая линия с гибким управлением)	ОС-гибкая ИС-жесткая ИУС-жесткая (автоматическая переналаживаемая линия)	ОС-гибкая ИС-гибкая ИУС-гибкая (автоматическая переналаживаемая линия с адаптивным управлением)
Большая	ОС-жесткая ИС-гибкая ИУС-жесткая (автоматизированные центры)	ОС-жесткая ИС-гибкая ИУС-гибкая (сборочные участки поточного производства)	ОС-гибкая ИС-гибкая ИУС-жесткая (гибкое автоматическое производство)	ОС-гибкая ИС-гибкая ИУС-гибкая (гибкое автоматизированное производство с адаптивной структурой)

В / 9 / для оценки вариативности технологической среды и технологических операций предложена следующая методика.

В процессе выполнения технологических операций и люди и машины совершают движения, которые можно разделить на решительные, однозначно определенные для данного технологического процесса, и приноровительные, которые требуют для выполнения дополнительной информации о среде. Все движения оцениваются приведенным временем исполнения.

Тогда, вариативность технологической среды  $W_c$  есть отношение суммы времени приноровительных движений  $\tau_{i\text{пр}}$  к суммарному времени  $T$  выполнения всех движений в технологическом процессе:

$$W_c = \sum_i \tau_{i\text{пр}} / T \quad (\text{I.I.})$$

Для получения характеристики вариативности движений необходимо учесть частоту смены различных их типов в отдельных операциях, т.к. при замене человека на таких операциях требуется либо несколько устройств, либо одно устройство с многозвенной кинематикой и гибкой системой управления.

Показатель вариативности движений в операциях ( $W_{дв.}$ ) определяется следующим образом:

$$W_{дв.} = K_{т} \cdot \omega_j / K_0, \quad (I.2)$$

где  $K_{т}$  - число различных типов движений ;

$K_0$  - общее число движений ;

$\omega_j$  - частота использования  $j$ -го технологического процесса при функционировании ГАП. Введение  $\omega_j$  обеспечивает учет вариативности технологии.

Используя данные критерии можно провести классификацию (табл. 2.) исполнительных устройств, обеспечивающих реализацию технологического процесса.

Таким образом, если удастся оценить вариативность технологического процесса, то можно определить наиболее эффективный тип оборудования. Однако этих показателей для окончательного решения может быть недостаточно, так как на практике выбор оборудования зависит от экономических факторов, имеющих решающее значение.

Источники вариативности очень различны.

Нестабильность параметров сырья /химический состав, размеры, прочность и т.д./ приводит к вариативности времени исполнения технологических операций. Нестабильность оборудования, его конечная надежность порождает вариативность процесса планирования, приводя к коррекции сменно-суточных заданий и расписаний.

В дальнейшем рассматривается вариативность технологических процессов, вызываемая проблемами надежности функционирования оборудования и программного обеспечения.

Концептуальное рассмотрение ГАП завершим кратким анализом функциональной структуры (рис. I.3) типовой системы, где информационно-управляющая система представлена двухуров-

Таблица 2.

Классификация оборудования и управления по коэффициентам  
вариативности среды и движений

Вариативность Ва- риатив- ность среды движений	Жесткая кинематика	Гибкая кинематика
$W_{\text{ср.}} = 0$ (программное управление)	Автоматы с жест- ким программным управлением	Работы-манипуля- торы с програм- ным управлением
$W_{\text{ср.}} > 0$ (адаптивное управление)	Автоматы с дат- чиками и система- ми обработки первичной информа- ции.	Адаптивные работы

новой интегрированной вычислительной системой. Нижний уровень системы (ЛУО) обеспечивает локальное управление отдельными видами оборудования и макромодулями, а верхний (УВО) - обеспечивает взаимодействие оборудования, т.е. осуществляет планирование и оперативное управление хода технологического процесса. Организационная система ГАП реализуется программными средствами мини - ЭВМ, либо ЭВМ более высокого уровня управления.

На нижнем уровне управления, где установлены микро - ЭВМ решаются следующие задачи:

- синтаксический и семантический анализ и перевод командных операторов технологического языка управления элементом ГАП (модулем, макромодулем) в последовательность макрокоманд управления оборудованием;
- передача информации о нормальном или аварийном завершении выполнения микрокоманд и операций;
- обработка прерываний по нормальному или аварийному завершению выполнения микрокоманд и операций.

На этом же уровне встроенными системами контроля и диагностики решаются задачи обеспечения надежности функционирования элементов ГАП:

- контроль выполнения оборудованием микрокоманд с помощью системы активного контроля;
- компенсация случайных сбоев оборудования;
- анализ работоспособности оборудования;
- контроль и компенсация изменений размеров инструмента и систематических погрешностей оборудования /автоматическая подналадка оборудования/;
- тестирование оборудования;
- аварийный останов оборудования при нарушении границ рабочей зоны.

На уровне управления взаимодействием оборудования с помощью мини - ЭВМ (рис. 1.3.) решаются следующие задачи.

I. При реализации функции "Календарное планирование":

- анализ планового задания на принципиальную составимость календарного плана;
- формирование библиотеки производственных, директивных, экономических и технологических ограничений, оптимизирующих функций;
- формирование библиотеки функций упорядочения /предпочтений/ для партий, технологических процессов и всех видов исполнительных средств;
- определение "узких" мест по видам исполнительных средств, т.е. элементов ГАП, которые должны быть максимально загружены;
- формирование и оценка качества варианта календарного плана;
- подсчет плановой загрузки оборудования;
- расчет таблицы соответствия сроков запуска-выпуска партий в плановом периоде.

Функция "Оперативный учет" может быть реализована следующей совокупностью задач:

- учет загрузки всех видов исполнительных средств;
- учет всех видов простоев исполнительных средств по различным причинам;
- учет сбоев, отказов, поломок исполнительных средств;
- учет задержек поставок заготовок, инструмента, оснастки;
- учет заготовок, инструмента, оснастки;

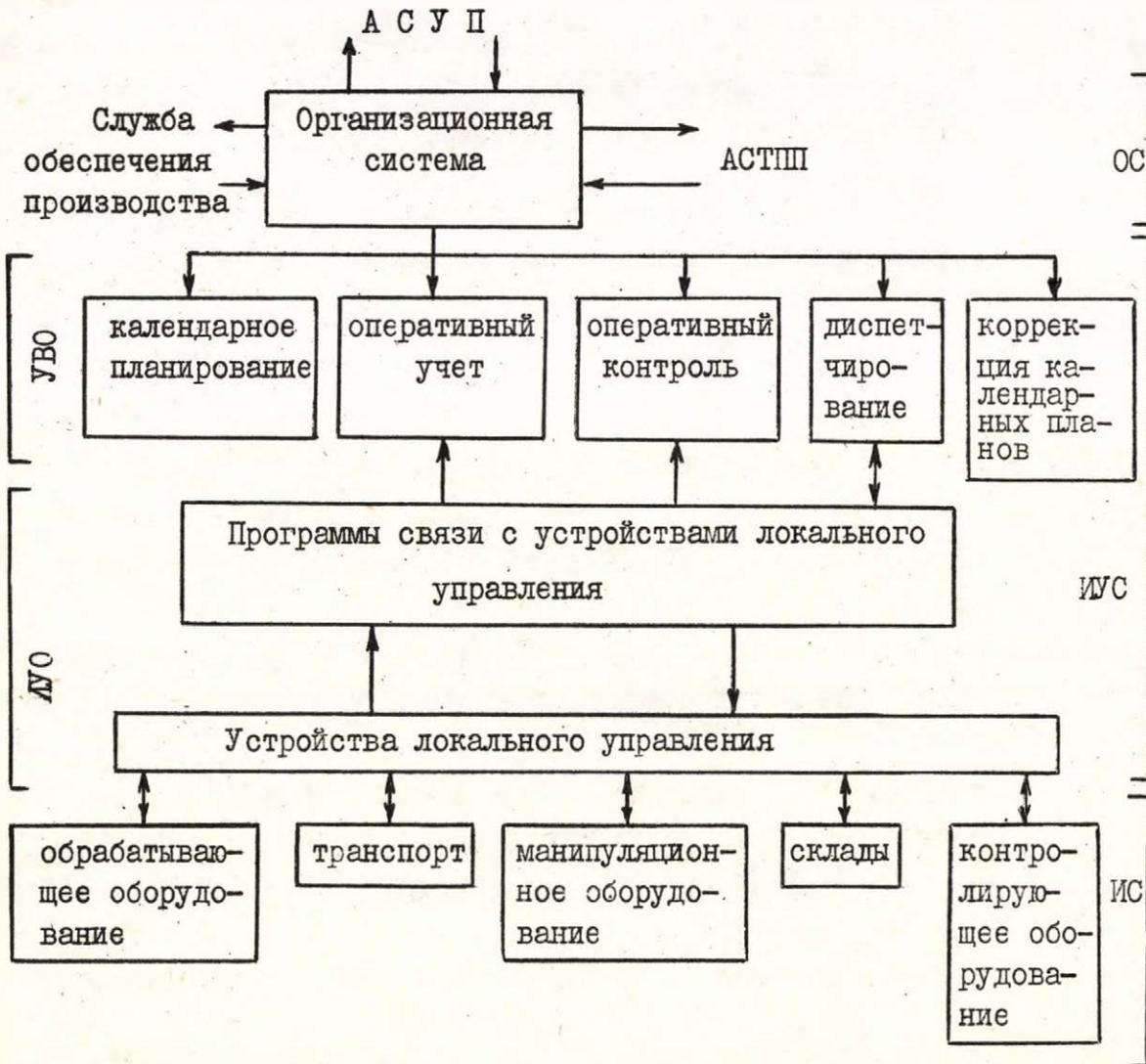


Рис. 1.3. Функциональная структура ГАП

- учет изделий на выходе ГАП;
- учет брака по всем партиям;
- расчет экономических показателей.

Функция "Оперативный контроль" зачастую реализуется совместно с функцией "Оперативный учет" и включает в себя решение задач:

- контроль всех видов исполнительных средств по состоянию;
- контроль обрабатывающего оборудования по браку деталей;
- контроль и предсказание потребности в заготовках, инструменте, оснастке;
- контроль и предсказание потребности в управляющих программах для исполнительных средств с программным управлением;
- контроль за прохождением всех партий, согласно технологическому маршруту и календарному плану;
- контроль за экономическими показателями.

Реализация функции "Диспетчирование" (оперативное управление взаимодействием оборудования) в значительной степени зависит от принятого в конкретном исполнении ГАП метода управления. Например, в автоматизированном технологическом комплексе (АТК), реализованном на Днепропетровском электровагоностроительном заводе данная функция реализуется следующими задачами [9]:

- анализ сигналов исполнительных средств;
- анализ величины рассогласования фактического и планового времени завершения операции;
- сдвиг вышедших из графика операций согласно временных резервов календарного плана;

- анализ технологического процесса для определения следующей операции;
- анализ состояния всей системы по "портретам" партий деталей и исполнительных средств;
- анализ календарного плана и выбор исполнительных средств, инструмента, оснастки и программ управления для выполнения следующей операции;
- выдача оператору таблиц загрузки оборудования, схем прохождения партий деталей по технологическим маршрутам;
- корректировка временных резервов календарного плана при внесении изменений в ход технологического процесса.

Функция "Коррекция календарного плана" в ряде систем не подвергается автоматизации и по-прежнему возложена на человека-мастера участка или оператора ГАП. Базовыми задачами как в ручном, так и в автоматизированном варианте, следует считать следующие:

- контроль выполнения календарного плана предыдущего периода планирования;
- анализ отклонений от календарного плана, определение партий деталей, идущих с нарушением сроков как опережающих, так и отстающих;
- анализ возможности завершения обработки отстающих партий в срок за счет опережающих партий;
- формирование запросов к оператору или системе управления более высокого уровня в случае невозможности выполнения планового задания в срок;
- контроль обеспеченности календарного плана очередного этапа планирования;
- формирование плановых заданий для пересчета календарных планов последующих этапов;

- коррекция календарного плана текущего периода планирования с использованием временных резервов

В дальнейшем изложении более подробно будут рассмотрены те функции и задачи информационно-управляющей системы ГАП, которые обеспечивают бесперебойную работу комплекса при наличии сбоев и отказов и реализуют оперативное управление ходом технологического процесса.

## 2. Анализ надежности функционирования ГАП

### 2.1. Надежность средств вычислительной техники

Вычислительные системы, являющиеся основой гибких автоматизированных систем, функционируют в более сложных условиях чем ЭВМ, используемые в других типах автоматизированных систем, от них часто требуется круглосуточная безотказная работа при очень высокой достоверности результатов решения задач. В условиях ГАП аварийный переход на ручное управление затруднен или даже невозможен, и полный выход ЭВМ из строя на сколько-нибудь длительное время приводит к катастрофическим последствиям. Поэтому ГАП оснащаются высоконадежными вычислительными системами, практически исключающими полный останов ЭВМ.

В значительной степени надежность функционирования вычислительной системы определяется такими факторами, как самоустраняющиеся сбои и отказы, а так же искажения информации в системах передачи данных и памяти ЭВМ.

#### 2.1.1. Самоустраняющиеся отказы и сбои

Самоустраняющиеся отказы и сбои в аппаратуре вычислительных систем являются фактором, существенно влияющим на конечную надежность функционирования ГАП. За последние годы достигнуты значительные успехи в повышении надежности вычислительных систем. Особенно велики результаты по снижению вероятности полного отказа аппаратуры.

Существуют системы, характеризующиеся средним временем наработки на отказ, исчисляемым десятками тысяч часов, однако для однопроцессорных ЭВМ наработка на устойчивый отказ, как правило, измеряется сотнями часов.

Значительно чаще происходят сбои или труднообнаруживаемые кратковременные отказы. Большинство из них выявляется и устраняется средствами аппаратурного контроля, не влияя на исполнение программ. Однако некоторая часть аппаратурных сбоев может приводить к искажениям исполнения программ или к искажениям переменных. Причинами таких сбоев и отказов являются преимущественно внешние воздействия на аппаратуру, влияющие на нарушение контактов и пропадание сигналов или индустриальные электрические помехи, что наиболее характерно для ГАП. Это приводит к тому, что обнаруживаемые тестами сбои и самоустраняющиеся отказы происходят на один-два порядка чаще, чем устойчивые отказы. Еще чаще происходят сбои, которые не удается обнаружить и зафиксировать при функционировании комплекса программ в процессе нормальной обработки информации и управления. Такие сбои проявляются в случайные моменты времени, и практически невозможно добиться их повторяемости.

Трудность их регистрации и изучения, а также незаинтересованность фирм, производящих ЭВМ и оборудование ГАП, в выявлении характеристик сбоев приводят к тому, что достоверные данные о них практически отсутствуют. Тем не менее искажения переменных и процесса исполнения программ из-за сбоев аппаратуры иногда приводят к закливанию, остановку

или искажению массивов данных.

Если среднее время наработки на устойчивый отказ в однопроцессорной ЭВМ составляет 100 часов, то интервал времени между обнаруживаемыми сбоями и самоустраняющимися отказами составляет около 1 часа. Еще чаще происходят сбои, которые невозможно зарегистрировать, например, искажения младших разрядов переменных, являющихся результатом измерения гладких физических величин. При среднем быстродействии ЭВМ 100 тыс. операций в секунду это соответствует выполнению произвольной операции с вероятностью искажения около  $10^{-7} - 10^{-8}$ .

## 2.1.2. Искажения информации в системах передачи данных

ЭВМ связываются с внешними абонентами, источниками и потребителями информации с помощью систем передачи данных. Внешними абонентами могут быть измерительные комплексы, исполнительные системы ГАП или ЭВМ, входящие в вычислительную систему. Удаление ЭВМ, входящих в систему, колеблется от десятков метров до многих километров. Компонентами информационной сети системы передачи данных (СПД) являются средства каналообразования, передачи и приема данных, повышения достоверности передачи, центры коммутации и узлы связи, а также каналы передачи данных.

Транспортировка сообщений по каналам связи характеризуется рядом параметров:

- скоростью передачи или обратной величиной, равной времени доставки сообщения зафиксированного объема в заданный адрес;
- надежностью или вероятностью правильной доставки сообщения адресату;
- достоверностью принятого сообщения, которую определяют по вероятности возникновения в сообщении необнаруженной ошибки.

Эти параметры являются взаимозависимыми, так как при увеличении избыточности повышается надежность и достоверность и снижается скорость передачи. Вероятность ошибок меняется от  $10^{-3}$  до  $10^{-6}$  на 1 бит сообщений в зависимости от скорости передачи информации.

Лучшими характеристиками среди телефонных каналов обладают кабельные ( $\sim 3 \cdot 10^{-4}$ ). Телеграфный проводной канал имеет в среднем вероятность ошибки на знак (байт) порядка  $1,9 \div 10^{-3}$ .

По экспериментальным данным [ 4 ] на различных линиях связи большинство прерываний (более 50%) имело длительность меньше 0,1 сек и более 80% - менее 1 сек. Однако такие перерывы более редки, чем шумовые искажения небольшой группы разрядов. Специальные системы передачи данных характеризуются достоверностью передачи, достигающей  $10^{-8}$  на символ. В большинстве случаев (применительно к ГАП, где протяженность линий связи не велика) на входе ЭВМ достоверность данных составляет  $10^{-6}$  -  $10^{-7}$  на символ.

Далеко не каждое искажение способно вызвать сбой или отказ при исполнении программ. Часть искажений приходится на результаты измерения квазинепрерывных переменных или на переменные, которые подвергаются глубокому дополнительному контролю и не могут привести к сбою или отказу.

В среднем, доля искажений, приводящих к сбою или отказу, находится в диапазоне  $10^{-2}$  -  $10^{-5}$ . Если предположить, что  $10^{-2}$  искажений приведет к отказам, то для данных, передаваемых по телефонным каналам со скоростью 300 символов в секунду и достоверностью  $10^{-6}$ , вероятность передачи символа, вызывающего отказ, составляет примерно  $10^{-8}$ .

Что соответствует темпу передачи символа, вызывающего отказ, один раз в 100 часов. В результате вычислительная система в целом будет иметь наработку на отказ около 100 часов. Таким образом, передача данных по каналам связи имеет существенное влияние на надежность функционирования комплексов программ.

### 2.1.3. Искажения данных при накоплении и хранении в памяти вычислительных систем

Достоверные данные, поступившие в ЭВМ, могут быть искажены или потеряны до начала их обработки основными функциональными алгоритмами в процессе накопления в памяти, предварительного упорядочения, селектирования и перекодирования. Эти искажения обусловлены ограниченностью ресурсов реальных вычислительных систем, флюктуациями потоков сообщений от внешних абонентов и длительностями их обработки. Наибольшее значение имеют задержка сообщений в памяти ЭВМ свыше допустимого времени из-за ограниченной производительности вычислительной системы и потеря (стирание) сообщений в буферных накопителях.

При неограниченной памяти наличие предельной производительности ЭВМ не приводит к потере сообщений или их искажению, однако при перегрузке ряд сообщений может попасть в состояние столь долгого ожидания обработки, что это эквивалентно их отсутствию.

Динамические характеристики внешних абонентов позволяют установить порог допустимой длительности ожидания сообщений. Например, если загрузка вычислительной системы на 10% превышает допустимую, то не менее 10% сообщений своевременно не попадут на обработку. Такие задержанные сообщения следует рассматривать как искаженные, так как содержащаяся в большинстве случаев в них информация о внешних абонентах не соответствует времени начала их обработки и в данном примере вероятность искажения достигает 0,1.

Характеристики искажений, обусловленные задержкой, могут меняться при использовании различных дисциплин распределения производительности вычислительных систем, учитываю-

ших характеристики потоков сообщений и длительностей их обработки.

Запаздывания в обработке мало влияют на надежность функционирования комплексов программ, однако при длительных перегрузках возможно прекращение обработки сообщений и нарушение целостности решения функциональных задач.

Разрушение логической связности исполнения программ может быть эквивалентно отказу функционирования.

На задержку и временное искажение сообщений влияют структура и типы памяти. В современных вычислительных системах объем внешней памяти может увеличиваться почти неограниченно, однако затраты производительности на обмен данными между внешней и оперативной памятью является существенным ограничением на объем памяти всей системы.

Ограниченность оперативной памяти отражается на надежности функционирования комплексов программ прежде всего из-за конечного объема буферных накопителей вычислительной системы для приема и выдачи сообщений. В реальных системах [ 4 ] вероятность потери не менее  $10^{-4}$ , а буферные накопители рассчитываются на вероятность потери сообщений около  $10^{-2} + 10^{-3}$  для сообщений низших приоритетов.

При этом реальная производительность системы используется в среднем на 90-95%. При перегрузках эта вероятность возрастает до 0,1 и выше. Таким образом, неопределенность динамических характеристик процесса обработки информации всегда сохраняет некоторую вероятность отказа в системе по этим причинам.

## 2.2. Надежность программного обеспечения

В последнее время в области интересов и практики исследования надежности вышел новый вид изделий - сложные комплексы программного обеспечения систем управления и обработки информации. При эксплуатации таких комплексов возникают сбои и отказы, обусловленные искажениями программ и данных. Эти искажения не только возникают в связи с аномалиями работы аппаратуры, но могут проявиться при безотказной работе ЭВМ, реализующей данный комплекс программ. Отсутствие старения и физического разрушения привело к появлению мнения о полной неприменимости методов существующей теории надежности для исследования надежностных характеристик программ. Однако анализ сбоев и отказов при длительном функционировании сложных комплексов программ позволяет выявить аналогии со сбоями и отказами аппаратуры. Более того, отсутствие в большинстве случаев физического разрушения и необходимости ремонта программ резко повысило возможность автоматического восстановления программ после их отказов без участия человека. Возникла задача количественного исследования и создания методов оперативного восстановления программ и данных, обеспечивающих сокращение длительности восстановления и снижения последствий отказа до уровня результатов кратковременного сбоя.

### 2.2.1. Основные понятия теории надежности комплексов программ

Программы для ЭВМ можно разделить на три основных типа. К первому относятся программы, разрабатываемые для решения инженерных и научно-исследовательских задач. Они характеризуются неполным использованием ресурсов вычислительных систем, их эксплуатация носит кратковременный характер, отсутствуют жесткие ограничения на допустимую длительность ожидания результатов, практически всегда имеется возможность достаточно строго проконтролировать выходные данные и при необходимости поставить

контрольные эксперименты. К этому типу программ практически не применимы основные понятия теории надежности.

Второй тип представлен сложными комплексами программ для информационно-справочных систем и систем автоматизированной обработки информации, которые функционируют вне реального времени. Для таких комплексов программ техническими документами могут быть определены функции и характеристики, а также промежуток времени, на который должны сохраняться заданные показатели. Однако, изменение комплекса программ в процессе развития и модернизации системы приводит к тому, что содержание и значения показателей надежности оказываются нестационарными.

К третьему типу относятся комплексы программ автоматического или автоматизированного управления, непосредственно входящие в контур управления и функционирующие в реальном масштабе времени. Такие комплексы программ практически полностью используют ресурсы ЭВМ по памяти и производительности, снабжаются подробной документацией и эксплуатируются многие годы. Эти комплексы в значительной мере определяют степень автоматизации производства. Комплексы программ этого типа обладают всеми характерными чертами промышленных изделий и к ним в наибольшей степени применимы основные подходы и понятия теории надежности. Реальная надежность программного обеспечения нередко оказывается ниже, чем надежность аппаратурных средств и определяет надежность функционирования системы в целом.

К задачам анализа надежности программного обеспечения можно отнести следующие [ 4 ]:

- формулирование основных понятий, используемых при исследовании параметров и показателей надежности программ;
- выбор и обоснование критериев надежности комплексов программ;
- выявление и исследование основных факторов, определяющих характеристики надежности сложных программных

- исследование характеристик искажений исходных данных от различных типов источников и их влияние на надежность функционирования комплексных программ;
- исследование ошибок в программах, динамики изменения при отладке и модернизации и влияния на надежность;
- разработка и исследование методов структурного синтеза сложных комплексов программ, повышающих их надежность;
- исследование методов и средств контроля и защита от искажений вычислительного процесса и данных в памяти путем ввода различных видов избыточности;
- разработка методов прогнозирования характеристик надежности комплексов программ с учетом их сложности, структурного построения и технологии проектирования.

Естественно, в данной работе будут лишь частично рассмотрены некоторые из перечисленных задач, применительно к программному обеспечению ГАП.

Уточним фундаментальные понятия теории надежности (сбой, отказ, восстановление, надежность и т.д.) при их использовании для анализа характеристик функционирования комплексов программ.

Отказ при исполнении программ. Рассмотрим специфику нарушения работоспособности программ в предложении абсолютной безотказности аппаратуры вычислительной системы. Отказ при исполнении комплекса программ может появиться вследствие: нарушения кодов записи программ в памяти команд; стирания или искажения данных в оперативной или долговременной памяти; нарушения нормального хода вычислительного процесса.

Перечисленные искажения могут действовать совместно. Отказ может проявляться в виде программного останова или заикливания, систематического пропуска исполнения некоторой группы команд, однократного или систематического искажения данных и т.д. Программные отказы приводят к прекращению выдачи абонентам информации и управляющих воздействий или к зна-

чительному искажению их содержания и темпа выдачи, соответствующих нарушению работоспособности комплекса программы.

Основной причиной отказа является конфликт между исходными данными, подлежащими обработке и характеристиками программы, осуществляющей ее обработку. При этом под исходными данными понимается как вновь поступившая информация, так и вся информация накопленная за время предыдущего функционирования программы. Исходные данные могут находиться в области, определенной техническим заданием, однако вне области, проверенной при тестировании и испытаниях программы на надежность. Кроме того, реальные исходные данные могут иметь значения, отличные от определяемых техническим заданием.

Сбой при исполнении программ. Понятие "сбой" в теории надежности трактуется [ 10 ] как самоустраняющийся отказ, не требующий внешнего вмешательства для замены отказавших компонент. При конфликтах исходных данных не требуется замены или ремонта материальных компонент. Восстановление после программного отказа в принципе всегда может быть осуществлено программными средствами без вмешательства человека. Основным признаком классификации сбоев и отказов становится длительность восстановления, т.е. необходимо установление порогового значения, что возможно сделать при анализе динамических характеристик абонентов - потребителей. В этом случае существенны следующие динамические параметры системы: инерционность объекта управления; среднее время и частота (темп) решения задачи; необходимая длительность отклика; средний интервал времени между однотипными сообщениями, поступающими на обработку. Для систем типа ГАП эти параметры в настоящее время имеют следующие значения [ 7 ]<sub>1</sub>: среднее время решения -  $1 \div 10$  сек; время отклика -  $10 \div 1$  сек; средний интервал между однотипными сообщениями -  $10 \div 1$  сек; пороговое время восстановления между областью сбоев и отказов -  $1 \div 10$  сек.

### Правильный и надежный комплекс программ

В ряде работ [7,8] отмечается отсутствие тождественности понятий "правильная" и "надежная" программа. Понятие "правильная" рассматривается статистически, вне временного функционирования. Правильная программа должна обеспечивать выходные данные, соответствующие эталонным, в области изменения исходных данных, заданных требованиями технического задания. Надежная программа должна обеспечивать низкую вероятность отказа в процессе функционирования. Быстрая реакция на искажения программ, данных или вычислительного процесса и восстановление работоспособности за время, меньшее порогового, позволяют обеспечить высокую надежность программ. Следовательно, отказ при функционировании программы является понятием динамическим и произойдет при совмещении следующих событий:

- появлению на входе программы данных, попадающих в непроверенные при тестировании и испытаниях области;
- обработке этих данных компонентами программы, содержащими ошибку, достаточную для появления отказовой ситуации;
- длительности восстановления после возникновения отказовой ситуации, превышающей пороговое значение.

Восстановление. Отсутствие физического разрушения компонент функционирующего комплекса программ выдвигает в качестве главной задачу восстановления за время, не превышающее порогового значения.

### 2.2.2. Критерии оценки надежности функционирования комплексов программ

Отказы и сбои по степени их влияния на функционирование комплекса программ и на всю систему управления в целом делятся на три группы [10]:

- искажения вычислительного процесса и данных, вызывающие полное прекращение выполнения функций системой управления на длительное или неопределенное время - отказ, в значительной степени обеспечивающий результаты предыдущего функционирования;
- искажения, кратковременно прерывающие функционирование системы и мало искажающие накопленные данные и выдаваемые результаты - частичный отказ или длительный сбой, в некоторой степени обеспечивающий предыдущие результаты;
- искажения, кратковременно и мало отражающиеся на вычислительном процессе и обрабатываемых данных - сбой, практически не обеспечивающие результаты функционирования комплекса программ.

В зависимости от глубины контроля и длительности запаздывания в обнаружении отказа, а также в зависимости от качества средств, осуществляющих восстановление, одни и те же ситуации искажений вычислительного процесса или данных могут быть отнесены к различным типам отказов и сбоев.

Наиболее типичными полными отказами являются:

- заикливание, т.е. последовательное повторяющееся исполнение группы команд, которое не прекращается без внешнего вмешательства, блокируя функционирование всех остальных программ данного комплекса;
- останов ЭВМ и полное прекращение решения функциональных задач, при этом может сохраняться возможность приема и выдачи информации и выполнения некоторых функций, стимулируемых прерываниями;
- значительное искажение или полная потеря накопленных данных о состоянии внешних абонентов и процесса их функционирования;

- прекращение или недопустимое снижение темпа решения некоторых задач, пропуск или потеря необработанных групп сообщений вследствие перегрузки ЭВМ по пропускной способности.

В несколько меньшей степени на вычислительный процесс и обрабатываемые данные влияют искажения, приводящие к следующим типам частичных отказов или длительных сбоев:

- искажение заданной последовательности вызова программ, приводящие к пропуску исполнения отдельных подпрограмм или их частей, что в свою очередь может привести к неправильному или неполному решению некоторых задач и к искажению выходных результатов;
- использование и обработка искаженных исходных данных, отражающиеся на логике решения задач и приводящие к искажению отдельных накопленных и выводимых данных.

В зависимости от повторяемости и глубины распространения искажения квалифицируются как частичные отказы либо как сбои с длительными последствиями.

Для оценки надежности программ, как правило, приходится ограничиваться интегральными показателями наработки на отказ и средним временем восстановления. Качество отладки определяется частотой отказов и значениями ошибок в программах и искажений исходных данных.

Существующие математические модели позволяют оценивать характеристики ошибок в программах и прогнозировать их надежность при проектировании и эксплуатации. Модели имеют вероятностный характер и достоверность прогнозов зависит от точности исходных данных и глубины прогнозирования во времени. Они предназначены для оценки [6]:

- показателей надежности комплексов программ в процессе отладки;
- количества ошибок, оставшихся невыявленными;
- времени, необходимого для обнаружения следующей ошибки в функционирующей программе;
- времени, необходимого для выявления всех ошибок с заданной вероятностью.

Основными среди существующих являются следующие четыре модели:

- экспоненциальная модель изменения количества ошибок в зависимости от времени отладки;
- модель, учитывающая дискретно-понижающуюся частоту появления ошибок как линейную функцию времени тестирования и испытаний;
- модель, базирующаяся на распределении Вейбула;
- модель, основанная на дискретном гипергеометрическом распределении.

Наиболее обоснованной представляется первая модель, основные предположения которой заключаются в следующем [11]:

1. Любые ошибки в программе являются независимыми и проявляются в случайные моменты времени с постоянной средней интенсивностью при отсутствии корректировок на всем интервале времени функционирования программы.
2. Типы исполняемых в программе команд перемешаны и время работы между ошибками определяется средним временем выполнения команды на данной ЭВМ и средним числом команд, исполняемых между ошибками.
3. Потенциальное множество тестов при отладке должно покрывать все множество реальных исходных данных при нормальном функционировании комплекса программ.

4. Ошибка, являющаяся причиной искажения результатов фиксируется и исправляется после завершения тестирования либо вообще не обнаруживается. Из этих свойств следует, что при нормальных условиях эксплуатации количество ошибок, проявляющихся в некотором интервале времени, распределено по закону Пуассона. В результате длительность непрерывной работы между ошибками распределена экспоненциально. Параметр распределения изменяется после обнаружения и исправления очередной ошибки. При стационарных условиях тестирования или эксплуатации интенсивность обнаружения и устранения ошибок убывает в зависимости от времени. Количество обнаруживаемых и исправляемых ошибок в единицу времени пропорционально их общему количеству в программе на данном интервале времени.

Предположим, что в начале отладки ( $\tau = 0$ ) в комплексе программ содержалось  $N_0$  ошибок. После отладки в течении времени  $\tau$  осталось  $n_0$  ошибок и устранено  $n$  ошибок.

Время  $\tau$  учитывает только длительность исполнения программы на ЭВМ без простоев. При постоянных условиях отладки интенсивность обнаружения ошибок пропорциональна числу оставшихся ошибок.

$$dn/d\tau = k \cdot N_0 - k \cdot n,$$

где  $K$  - коэффициент, учитывающий масштаб изменения времени, используемого для описания процесса обнаружения ошибок, быстродействие ЭВМ и т.п.

Учитывая, что в начале отладки ( $\tau = 0$ ) отсутствуют обнаруженные ошибки, то решение имеет вид:

$$n = N_0 [1 - \exp(-k \cdot \tau)].$$

Количество оставшихся ошибок  $n_0 = N_0 \cdot \exp(-k \cdot \tau)$ .

Наработка на отказ равна величине, обратной интенсивности обнаружения ошибок:

$$T = \frac{1}{dn/d\tau} = \exp(k \cdot \tau) / k \cdot N_0$$

Если считать, что начальному количеству ошибок  $N_0$  соответствовала наработка на отказ -  $T_0$ , то функция наработки на отказ от длительности проверок будет иметь вид:

$$T = T_0 \cdot \exp(\tau / N_0 \cdot T_0)$$

Если известны моменты обнаружения ошибок  $t_i$  и каждый раз в эти моменты устраняется одна ошибка, то используя метод максимального правдоподобия, можно получить уравнение для определения  $N_0$ :

$$\sum_{i=1}^n 1/(N_0 - i + 1) = n \cdot \sum_{i=1}^n t_i / [N_0 \cdot \sum_{i=1}^n t_i - \sum_{i=1}^n (i-1) \cdot t_i],$$

а также выражение для расчета коэффициента пропорциональности

$$K = n / [N_0 \cdot \sum_{i=1}^n t_i - \sum_{i=1}^n (i-1) \cdot t_i].$$

В процессе отладки наработка на отказ увеличивается с  $T_1$  до  $T_2$  путем устранения  $\Delta n$  ошибок за время отладки (чистое время решения)  $\Delta \tau$ :

$$\Delta n = N_0 \cdot T_0 \cdot [1/T_1 - 1/T_2],$$

$$\Delta \tau = [N_0 \cdot T_0 \cdot \ln(T_2/T_1)] / K.$$

Вторая модель [8] построена на основе гипотезы о том, что частота появления ошибок линейно зависит от времени испытаний  $t_i$  между моментами обнаружения последовательных ошибок:

$$\lambda(t_i) = K \cdot [N_0 - i + 1] \cdot t_i;$$

где  $K$  - коэффициент пропорциональности, обеспечивающий равенство единице площади под кривой вероятности обнаружения ошибок.

Наработка на отказ соответствует распределению Релея:

$$T(t_i) = \exp \left\{ -\kappa \cdot [N_0 - i + 1] \cdot t_i^2 / 2 \right\}.$$

Используя функцию максимального правдоподобия, получаем оценку для  $N_0$  и  $\kappa$ :

$$N_0 = \left[ 2 \cdot n / \kappa + \sum_{i=1}^n (i-1) t_i^2 \right] / \sum_{i=1}^n t_i^2.$$

$$\kappa = \left[ \sum_{i=1}^n 2 / (N_0 - i + 1) \right] / \sum_{i=1}^n t_i^2.$$

Особенностью третьей модели является учет ступенчатого характера изменения надежности при устранении очередной ошибки. Если ошибки не устраняются, то интенсивность отказов остается постоянной:

$$T(t) = \exp(-\lambda \cdot t).$$

Отсюда плотность распределения наработки на отказ

$$f(t) = \lambda \cdot \exp(-\lambda \cdot t), \quad t > 0, \lambda > 0.$$

Для аппроксимации изменения частоты отказов от времени используется функция следующего вида:

$$\lambda(t) = \lambda \cdot \beta \cdot t^{\beta-1}$$

Если  $0 < \beta < 1$ , то интенсивность отказов снижается по мере отладки (или эксплуатации) и плотность функции распределения наработки на отказ описывается двухпараметрическим распределением Вейбула.

$$f(t) = \lambda \cdot \beta \cdot t^{\beta-1} \cdot \exp(-\lambda \cdot t^\beta).$$

Четвертая модель основана на дискретном гипергеометрическом распределении [11]. Метод базируется на процедуре

искусственного ввода в реальную программу ошибок. Методом максимального правдоподобия получены выражения для оценки количества ошибок в программе на основе данных о количестве выявленных реальных и введенных ошибок.

### 2.2.3. Возмущения, влияющие на надежность программного обеспечения

#### 2.2.3.1. Искажения исходных данных

От абонента, являющегося первичным источником исходных данных, до комплекса программ, сообщения проходят ряд этапов передачи, хранения, кодирования и преобразования. На каждом этапе данные подвергаются искажениям, характеристики которых необходимо учитывать при анализе надежности.

Рассмотрим обобщенные характеристики количества ошибок, возможных при организации информационного обмена.

Достоверность ввода со знаковых индикаторов (дисплеев) выше чем при работе с перфораторами и составляет  $10^{-4}$ – $10^{-5}$  на знак. Перенос данных из первичных документов на бланки характеризуется частотой ошибок, равной  $(1 \div 3) \cdot 10^{-4}$  на знак. При этом основная доля ошибок связана с нечетким заполнением исходных документов (38%), описками (20%), пропусками цифр в числе (17%), ошибками в символических шифрах (14%)

Вывод данных из ЭВМ и печать на бланках характеризуются вероятностью искажения  $\sim (1 \div 3) \cdot 10^{-4}$ . Эти искажения обусловлены неправильным печатаньем знаков ( $\sim 25 \div 30\%$ ), нечетким изображением символов ( $\sim 20\%$ ).

Приведенные оценки показывают, что достоверность ручной подготовки данных может заметно влиять на надежность функционирования комплексов программ.

### 2.2.3.2. Ошибки в комплексах программ

С точки зрения технологии разработки программ ошибки можно разделить на следующие типы:

- программные, вызванные неправильной записью на языке программирования и ошибками трансляции;
- алгоритмические, связанные с неполным формированием необходимых условий решения и некорректной постановкой задач;
- системные, обусловленные отклонением функционирования программ в системе и характеристик взаимодействующих объектов от предполагавшихся при проектировании.

Программные ошибки по количеству и типам определяются степенью автоматизации программирования и глубиной автоматического контроля записи программ. Количество программных ошибок зависит от квалификации специалистов-разработчиков, от общего объема программ, глубины логического и информационного взаимодействия отдельных частей алгоритма. На начальных этапах разработки и автономной отладки подпрограмм программные ошибки составляют около одной трети всех ошибок. Ошибки использования типов операций на начальных этапах разработки достигает 14%, а затем быстро убывают при повышении квалификации программистов. Ошибки переменных составляют около 13%, а ошибки управления и организации циклов - около 10%.

Каждая программная ошибка влечет за собой необходимость изменения около шести команд, что существенно ниже, чем при алгоритмических и системных ошибках. На этапе эксплуатации программные ошибки составляют 3% от общего количества ошибок, выделяемых в единицу времени.

Алгоритмические ошибки труднее поддаются обнаружению методами формального автоматического контроля. Ошибки, обусловленные неполным учетом условий решений задач, являются

наиболее частыми в этой группе и составляют до 70% всех алгоритмических ошибок или около 30% общего количества ошибок.

Ошибки сопряжения функциональных подпрограмм составляют 6-8% общего количества и квалифицируются как ошибки некорректной постановки задачи. Для исправления алгоритмической ошибки приходится изменять до 14 команд.

Системные ошибки определяются неполной информацией о реальных процессах, происходящих в источниках и потребителях информации. При автономной и вначале комплексной отладки доля системных ошибок невелика (10%), но она существенно возрастает (до 35-40%) на завершающих этапах отладки. В процессе эксплуатации системные ошибки являются преобладающими (~ 80% всех ошибок). При этом в среднем для исправления системной ошибки корректируется до 25 команд.

Развитие технологии создания сложных комплексов программ и средств автоматизации программирования приводит к изменению начального уровня и интенсивности устранения ошибок в программах, однако имеющийся опыт разработок могут служить первым приближением при последующих оценках.

### 3. Модели планирования и управления функционированием ГАП

#### 3.1. Информационная модель технологического участка

Все многообразие задач, решаемых на различных уровнях автоматизированной системы управления производством, связывается в единое целое информационными потоками, позволяющими при обособленном рассмотрении каждой задачи учесть ее специфику и глобальные цели всей системы. В связи с этим рассмотрим общую информационную модель технологического участка, опирающуюся на результаты, полученные в [ 9 ].

Конфигурация участка задается наличием парком оборудования (складов, манипуляторов, транспорта, обрабатывающего и контрольного оборудования), составом операторов - рабочих и наладчиков, пространственными и функциональными связями между ними. Программа функционирования участка задается объемными планами на плановый период, набором технологических процессов (маршрутных технологий), нормативной информацией о связях между технологическими операциями и оснасткой, приспособлениями, инструментом, программами обработки для станков с ЧПУ и роботов-манипуляторов.

За плановый период  $T$  пл. на участке необходимо обработать множество партий деталей  $D = \{d_e \mid e = \overline{1, L}\}$ .

Над каждой партией должна быть выполнена упорядоченная совокупность операций  $\Theta_i = \{O_{ik} \mid k = \overline{1, K_i}\}$ ,

где  $K_i$  - количество операций в  $i$ -м технологическом процессе;

Каждая партия характеризуется вектором параметров:

$$\bar{P}_e = \{d_e, \Theta_i, \varepsilon_e, \tau_e, \bar{\tau}_e, K_{ie}^o, K_{ie}^k, \pi_e\},$$

где  $\varepsilon_e$  - количество деталей в  $e$ -ой партии;

$\tau_e$  - плановый срок запуска партии в обработку;

$\bar{\tau}_e$  - плановый срок выпуска партии;

$K_{ie}^o$  - номер операции в  $\Theta_{ie}$  которого начинается изготовление партии ;

$K_{ie}^k$  - номер операции, которой заканчивается изготовление партии в плановом периоде ;

$\pi_e$  - приоритет партии.

Множество векторов  $\bar{P}_e$  составляют производственную программу участка на период  $T_{пл.}$ :

$$P = \{ \bar{P}_e | e = \overline{1, L} \}.$$

На участке имеется множество всего оборудования

$$A = \{ v_n | n = \overline{1, N} \},$$

где  $n$  - порядковый номер оборудования, независимо от его вида.

Множество  $A$  может состоять из групп однородного оборудования:

$$A = \{ B_m | m = \overline{1, M} \}; \cap B_m = \{ v_n^m | n = \overline{1, N}, m = \overline{1, M} \};$$

$$B_\gamma \cap B_\mu = \emptyset, \forall \gamma \neq \mu.$$

Оборудование может быть сгруппировано по различным признакам, например, по взаимозаменяемости относительно какой-либо технологической операции:

$$G_k = \{ v_{km} | m \in M_k \},$$

где  $M_k$  - множество номеров групп однородного оборудования, входящих в  $k$ -ю технологическую группу.

Пространственное расположение оборудования задается для линейных конфигураций вектором расстояний между группами однородного оборудования  $\bar{D}$  и единицами его в группах  $\bar{d}$ .

Для сложных конфигураций задается список  $S = \{ v_n, x_n \}$ ,

где  $x_n$  - координаты.

Оборудование участка разделяется на рабочие места:

$$F_{\gamma}^{\circ} = \{v_k | k \in K_{\gamma}\}, \gamma \in \Gamma,$$

где  $\Gamma$  - множество номеров рабочих мест,  
 $K_{\gamma}$  - множество номеров оборудования, объединенных в рабочее место (станок, робот, накопитель и т.п.).

Для выполнения технологической операции используется инструмент, оснастка и приспособления. Обозначим:

$$\{H_{\omega} | \omega = \overline{1, \Omega}\}$$

- множество типов инструмента и оснастки,  $\{Q_{\nu} | \nu = \overline{1, \Omega'}\}$   
 - множество типов приспособлений. На участке имеется  $\Psi_{\omega}$   
 инструмента или оснастки  $\omega$  типа,  $H_{\omega} = \{h_{\psi}^{\omega} | \psi = \overline{1, \Psi_{\omega}}\}$   
 и  $X_{\nu}$  приспособлений  $\nu$ -го типа  $Q_{\nu} = \{q_x^{\nu} | x = \overline{1, X_{\nu}}\}$ .

Для множеств  $H_{\omega}$  и  $Q_{\nu}$  выполняются следующие соотношения:

$$H_{\omega_1} \cap H_{\omega_2} = \emptyset, \forall \omega_1 \neq \omega_2;$$

$$\bigcup_{\omega} H_{\omega} = H; \sum_{\omega} \Psi_{\omega} = |H|;$$

$$Q_{\nu_1} \cap Q_{\nu_2} = \emptyset, \forall \nu_1 \neq \nu_2;$$

$$\bigcup_{\nu} Q_{\nu} = Q; \sum_{\nu} X_{\nu} = |Q|.$$

Для выполнения манипуляционных операций загрузки-разгрузки оборудования на участке имеется некоторое множество рабочих (роботов)  $R = \{R_n | n = \overline{1, N}\}$ .

Рабочий может иметь несколько профессий из множества  $V = \{v_{\lambda} | \lambda = \overline{1, \Lambda}\}$  и соответственно несколько разрядов (квалификаций) из множества  $U = \{u_{\alpha} | \alpha = \overline{1, A}\}$ , что задается множеством векторов  $R_n = \{(v_{\lambda n}, u_{\alpha n}) | n = \overline{1, N}\}$ .

Рабочие, взаимозаменяемые по профессиям и квалификации, объединяются в группы (работы - по выполняемым программам):

$$\Delta v = \{ R_{zv} \mid z \in 1, \bar{z}, v = 1, \bar{v} \},$$

где  $v$  - количество групп,  $|\bar{z}|$  - количество рабочих в группе, причем  $\bigcup_v \Delta v = R$ ,  $\Delta v_m \cap \Delta v_\lambda \neq \emptyset, \forall v_m \neq v_\lambda$  - для рабочих,  $\Delta v_m \cap \Delta v_\lambda = \emptyset, \forall v_m \neq v_\lambda$  - для роботов.

За каждым рабочим местом может быть закреплено некоторое количество рабочих  $\Pi_\beta = \{ R_{z\beta} \mid z \in H, \beta = 1, \bar{\beta} \},$

так что  $\Pi_{\beta_m} \cap \Pi_{\beta_\lambda} \neq \emptyset, \forall \beta_m \neq \beta_\lambda$ . В свою очередь за каждым рабочим может быть закреплено несколько рабочих мест

$$K_z = \{ \Pi_{\beta z} \mid \beta z \in B, z = 1, \bar{z} \},$$

причем  $K_{z_1} \cap K_{z_2} \neq \emptyset, \forall z_1 \neq z_2$ .

Множество накопителей участка  $Y = \{ Y_\epsilon \mid \epsilon = 1, \bar{\epsilon} \}$  разделяется на группы: долговременные (склады) и промежуточные (магазины у оборудования). Каждый накопитель характеризуется следующим вектором параметров:

$$\bar{Y}_\epsilon = \{ A_\epsilon^0, e_\epsilon, \delta_\epsilon, \Pi_\epsilon^A \},$$

где  $A_\epsilon^0$  - адрес накопителя (координаты начальной ячейки),

$e_\epsilon$  - емкость накопителя (количество ячеек),

$\delta_\epsilon$  - тип накопителя ( $\delta_\epsilon = I$  - склад,

$\delta_\epsilon = 0$  - магазин);

$\Pi_\epsilon^A$  - номер рабочего места, у которого установлен магазин или номер склада.

В маршрутных технологических процессах можно выделить 6 функциональных видов операций: операции обработки, контрольные, манипуляционные, транспортные, складские, внешние.

Операции обработки характеризуются следующим образом:

$$\bar{X}_{ik}^1 = \{ (B_m, \hat{O}_p, \Delta v, \Phi_j^{k_i}, t_j^{k_i}) \mid j = 1, \bar{j} \},$$

где  $k, i$  - определяют тип  $k^0$ -ой операции  $i$ -го технологического процесса;  $j$  - число способов, которыми можно выполнить операцию;  $\Phi_j^{k_i}$  - программа для оборудования с ЧПУ;  $B_m, \Delta$  - группа оборудования и рабочих, обеспечивающих  $k$  - ю операцию;  $t_j^{k_i}$  - время выполнения операции при вы-

боре  $j$  - го способа реализации;  $\hat{O}_p$  - комплект инструмента, приспособлений и оснастки.

Длительность операции включает время установки - снятия детали  $t_j^{k_i''}$ , время штучной обработки  $t_j^{k_i'}$ , время переналадки  $t_j^{k_i n}$ , т.е.:

$$t_j^{k_i} = t_j^{k_i'} + t_j^{k_i''} + t_j^{k_i n}$$

При большом времени переналадок или дефиците наладчиков переналадка планируется как самостоятельная операция.

Контрольные операции характеризуются теми же параметрами, что и обрабатывающие, но выделены в отдельный вид, так как алгоритмы обработки результатов этих операций различны - вызывают различное функционирование системы управления участком.

$$\bar{X}_{ik}^2 = \{V_m, \hat{O}_p, \Phi^{ik}, t_{ik}\}, t_{ik} = t_j^{k_i} \cdot \epsilon_e \cdot \hat{\rho}_{ik},$$

где  $\hat{\rho}_{ik}$  - процент контролируемых элементов из партии. При организации контроля в процессе обработки детали вектор  $\bar{X}_{ik}^2$  может не вводиться, а расширяются функции анализа момента окончания обрабатывающей операции.

Манипуляционные операции характеризуются одновременной занятостью двух исполнительных элементов (робот и транспорт, робот и станок и т.п.). Манипуляционные операции могут быть партионными и поддетальными

$$\bar{X}_{ik}^3 = \{V_m, \Delta y, \Phi^{ik}, t^{ik} \cdot [1 + (\epsilon_e - 1) \cdot \delta]\},$$

где  $\delta = 0$ , если операция поддетальная,  $\delta = 1$  для партионной операции.

Транспортные операции обеспечивают перемещение партий или тар из одного адреса в другой:

$$\bar{X}_{ik}^4 = \{ (V_m, \gamma_j^{ek}, \underline{A}^{ek}, \bar{A}^{ek}) \mid j = 1, \dots, J_m \},$$

где  $\gamma_j^{ek}$  - количество тар, требуемых для транспортировки  $e$ -й партии;  $\underline{A}^{ek}, \bar{A}^{ek}$  - начальный и конечный адреса транспортировки.

Складские операции являются статическими и их информационное описание носит справочный и учетный характер:

$$\bar{X}_{ik}^5 = \{ V_m, A_{ek}^m, E_{ek} \},$$

где  $A_{ek}^m$  - номер ячейки в складе  $m$ , где хранится партия  $e$  на  $k$ -ой операции;  $E_{ek}$  - количество ячеек, необходимых для хранения партии.

Операции, выполненные на других участках называют внешними и их достаточно характеризовать только временем исполнения

$$\bar{X}_{ik}^6 = \{ t_{ik} \}.$$

### 3.2. Модели календарного планирования

В соответствии с практической важностью задач календарного планирования в общей иерархии задач управления производством им уделяется значительное внимание. Классической формулировкой задачи считается следующая. Необходимо изготовить  $L$  различных деталей, которые должны пройти обработку на  $K$  станках при заданных временах обработки на каждом станке.

Требуется определить порядок запуска деталей на обработку так, чтобы общее время, необходимое для выпуска всех деталей, было минимальное.

Не имея возможности подробно проанализировать все многообразие известных моделей скажем только, что в общем виде для  $k > 2$  задача не решена и сформулируем ее в терминах рассмотренной информационной модели участка.

Дополнительно введем следующие обозначения:  $\underline{\tau}_{ek}, \overline{\tau}_{ek}$  - время фактического начала и конца  $k$ -й операции на  $e$ -й партии;  $\underline{\tau}_{nj}, \overline{\tau}_{nj}$  - время фактического начала и конца  $j$ -й операции на  $n$ -м оборудовании;  $\overline{\tau}_{ek_e}$  - время фактического завершения всего технологического процесса на  $e$ -й партии;  $t_{ek}^p$  - время пролеживания  $e$ -й партии перед  $k$ -й операцией;  $t_{nj}^p$  - время простаивания  $n$ -го оборудования перед  $j$ -й операцией.

Длительность прохождения партии через систему

$$t_e = \overline{\tau}_{ek_e} - \underline{\tau}_e = \sum_{k=1}^{K_e} t_{ek} + \sum_{k=1}^{K_e} t_{ek}^p.$$

При планировании считаем, что операция, начатая над  $e$ -й партией на  $n$ -м оборудовании, не прерывается:

$$\overline{\tau}_{ek} = \underline{\tau}_{ek} + t_{ek}^p.$$

### 3.2.1. Система ограничений

Система ограничений на выбор комбинаций, возможных в качестве элементов расписания, определяется в следующем виде [ 9 ].

1. Все переменные задачи - неотрицательные, действительные.
2. Соблюдение технологической последовательности для каждой партии:

$$\underline{\tau}_{e, k+1} \geq \overline{\tau}_{ek}, \forall e, k.$$

Неналожение во времени по партиям, оборудованию, рабочим или работам:

$$\bar{t}_{ek} = \underline{t}_{ek} + t_{ek} \leq \underline{t}_{e,k+1}, \forall e, k;$$

$$\bar{t}_{nj} = \underline{t}_{nj} + t_{nj} \leq \underline{t}_{n,j+1}, \forall n, j;$$

$$\bar{t}_{\eta j} = \underline{t}_{\eta j} + t_{\eta j} \leq \underline{t}_{\eta,j+1}, \forall \eta, j.$$

3. Соблюдение сроков запуска - выпуска партий:

$$\underline{t}_{e,j} \geq \underline{t}_e; \bar{t}_{ek_e} \leq \bar{t}_e, \forall e, j=1.$$

4. Ограниченность ресурсов по обрабатываемому оборудованию:

$$\sum_{e=1}^L \sum_{k=1}^{K_e} t_{ek} \cdot \delta_{ek}^m + t_m^n \leq T_m \cdot N_m,$$

где  $\delta_{ek}^m = \begin{cases} 1, & \text{если } k \text{ операция } e\text{-й партии} \\ & \text{требует оборудование } m\text{-й группы,} \\ 0, & \text{в противном случае;} \end{cases}$   
 $t_m^n$  - плановые простои  $n$ -го оборудования в плановом периоде;  
 $T_m$  - временной ресурс единицы оборудования  $m$ -й группы;  
 $N_m$  - количество оборудования в группе.

Аналогичные соотношения должны соблюдаться по ресурсу рабочего времени роботов, операторов, инструмента, оснастки и т.д.

Все временные параметры задаются в дискретном виде.

Дискрет задается как максимальная величина времени, в течении которого не совершается существенных событий, т.е. как наибольший общий делитель для множества  $\{t_{ik}^e\}, \forall i, k, e.$

### 3.2.2. Критерии качества

При экспериментальных постановках задачи календарного планирования формулируются целевые функции, определяющие оптимальное расписание. Таких целевых функций было предложено

множество. При появлении эвристических моделей и методов решения число критериев расписания начало множится еще быстрее. Выбор критерия всегда связан с конкретной ситуацией на производстве.

Наиболее реальной представляется точка зрения [ 9 ], что не существует общего критерия оптимальности календарного плана работы технологического участка и вопрос построения или выбора критерия должен решаться конкретно для каждого участка с учетом производственных ситуаций, условий его работы и целей календарного планирования.

Общая постановка задачи построения критерия состоит в том, чтобы задать некоторую числовую функцию  $F$ , определенную на всех планах-графиках и ставящая каждому графику  $G_k$  в соответствие некоторое число  $F(G_k)$ , причем для наилучшего плана-графика функция принимает экстремальное значение.

Рассмотрим некоторые, наиболее употребляемые на практике, критерии.

Обозначим:  $\bar{\tau}_e = \bar{\tau}_{eke}$  - время окончания выполнения заказа  $d_e$ ;  $\tau_e, \bar{\tau}_e$  - плановое время запуска - выпуска;  $\Delta t_e = \bar{\tau}_e - \tau_e$  - плановая продолжительность пребывания  $d_e$  на участке;  $\tau_{te} = \bar{\tau}_e^p - \tau_e$  - продолжительность технологического процесса по выполнению  $d_e$ ;  $D_e = \bar{\tau}_e^p - \bar{\tau}_e$  - величина отступления от планового срока выпуска;  $D_e^d = \max(0, D_e)$  задержка выполнения  $d_e$ ;  $D_e^o = \max(0, -D_e)$  - опережение выполнения  $d_e$ ;  $\tau_{ej}^1$  - время выпуска первой детали из  $d_e$  со станков  $j$ -й группы;  $\delta_{ej} = \tau_{ej} - \tau_{e, j-1}$  - продолжительность ожидания  $d_e$  перед операцией  $j$ ;  $\bar{\delta}_e = \sum_j \delta_{ej}$  - суммарное ожидание (пролеживание)  $d_e$  на участке;  $\Delta t_e$  - фактическая продолжительность пребывания  $d_e$  на участке,  $\Delta t_e^p = \bar{\tau}_e^p - \tau_e^p$ .

В [ 3 ] регулярным критерием назван критерий вида  $F(G_k) = F(\bar{\tau}_1, \bar{\tau}_2, \dots, \bar{\tau}_r)$ , неубывающий по  $\bar{\tau}_e$ , для которых  $F(G_{k1}) \leq F(G_{k2})$  всегда, когда по крайней мере для одного  $e$  -  $f_e(G_{k1}) \leq f_e(G_{k2})$ .

Примерами регулярных критериев служат следующие:

1. Максимальный момент окончания -

$$\bar{\tau}_{\max}(G_r) = \max_e \bar{\tau}_e(G_r);$$

2. Среднее значение моментов окончания -

$$\bar{\tau}(G_r) = \frac{1}{L} \sum_e \bar{\tau}_e(G_r);$$

3. Максимальная длительность пребывания -

$$\Delta t_{\max}^p(G_r) = \max_e \Delta t_e^p(G_r);$$

4. Средняя длительность пребывания -

$$\Delta t^p(G_r) = \frac{1}{L} \sum_e \Delta t_e^p(G_r);$$

5. Максимальное отступление от плана -

$$D_{\max}(G_r) = \max_e D_e(G_r);$$

6. Среднее отступление от плана -

$$D(G_r) = \frac{1}{L} \sum_e D_e(G_r);$$

7. Максимальная задержка выполнения -

$$D_{\max}^d(G_r) = \max_e D_e^{de}(G_r);$$

8. Средняя задержка выполнения -

$$D^d(G_r) = \frac{1}{L} \sum_e D_e^{de}(G_r);$$

9. Максимальное ожидание -

$$S_{\max}(G_r) = \max_e \delta_e(G_r);$$

10. Среднее ожидание -

$$S(G_r) = \frac{1}{L} \sum_e \delta_e(G_r).$$

Среди множества нерегулярных критериев следует отметить:

1. Среднее опережение сроков выпуска -

$$D^o(G_r) = \frac{1}{L} \sum_e D_e^o(G_r);$$

2. Максимальное опережение сроков выпуска -

$$D_{\max}^0(G_e) = \max_e D_e^0(G_e).$$

Критерии, связанные с экономическими соображениями, строятся на основе функций, ставящих в соответствие уже рассмотренным критерием стоимостные величины.

Широко распространены критерии, учитывающие издержки, связанные с межоперационными запасами, наладками и простоями. Общий подход в этом вопросе в том, что критерий - показатель решения задачи календарного планирования - должен, во-первых, быть наиболее важным в данной постановке задачи, и, во-вторых, выражаться количественно через переменные величины, варьируемые в процессе решения.

### 3.2.3. Методы решения задачи календарного планирования

Исторически, первыми были попытки решить задачу календарного планирования с помощью моделей математического программирования. Эти попытки не дали сколько-нибудь обнадеживающих результатов в силу огромной размерности моделей. Методы полного перебора еще более бесперспективны. Из методов направленного перебора наибольшую известность получил метод "ветвей и границ". Основная проблема метода - в определении зависимости оценок перспективности вершин и общего критерия качества.

В тех случаях, когда эта зависимость соответствует реальной ситуации алгоритм дает приемлимое решение. Довольно много попыток решить задачу методом Монте-Карло. В этой области полученные результаты в основном дали возможность промоделировать различные способы задания правил выбора и их влияния на качество полученных расписаний для различных производственных ситуаций. Всё многообразие подходов и методов и безуспешность их использования в реальных производственных ситуациях приводят к необходимости использования

эвристических методов решения. При этом под эвристикой понимается правило выбора претендента на включение в комбинацию на поле расписания в конфликтной ситуации.

Рассмотрим некоторые правила и функции приоритета, приведенные в литературе [3,9].

1. Правило случайного отбора:

$$Y_1 = \text{rand}(\omega_1, \omega_2, \dots, \omega_n),$$

где  $\omega_i$  - участники конфликта.

2. Обработка в порядке поступления:

$$Y_2 = \min_i \bar{t}_{ik},$$

где  $\bar{t}_{ik}$  - время поступления заказа  $d_e$  на операцию  $k$ .

3. Правило кратчайшей операции:

$$Y_3 = \min_j t_{ik}^j.$$

4. Чередование правил - в порядке поступления и кратчайшей операции:

$$Y_4 = Y(Y_2, Y_3).$$

5. Усеченное правило кратчайшей операции:

$$Y_5 = Y(Y_3, t_{ekm}^0, c),$$

где  $t_{ekm}^0$  - время ожидания заказа  $d_e$  перед станком  $m$ ;  
 $c$  - верхняя граница ожидания.

6. Правило наиболее раннего планового срока выпуска:

$$Y_6 = \min_e \bar{t}_e.$$

7. Правило жесткости сроков:

$$Y_7 = \min_e \frac{\bar{t}_e - t_e^0}{T_e^0},$$

где  $T_e^0$  - минимальная длительность технологического процесса при отсутствии пролеживания.

8. Правило наименьшего прошедшего времени:

$$Y_8 = \min_e (t - \tau_e),$$

где  $t$  - текущее время.

9. Правило соблюдения сроков выпуска:

$$Y_9 = \min_e (\bar{\tau}_e - t)$$

10. Правило жесткого соблюдения сроков выпуска:

$$Y_{10} = \min_e \left( \frac{\bar{\tau}_e - \tau_e}{\pi_e^0} + \bar{\tau}_e - t \right).$$

11. Правило соблюдения сроков выпуска с учетом начала обработки заказа:

$$Y_{11} = \min_e \left( \frac{\bar{\tau}_e - t}{\pi_e^0 - t_e} \right),$$

где  $t_e$  - фактическое время, прошедшее с момента начала обработки заказа к моменту  $t$ .

12. Правило минимума оставшегося времени обработки:

$$Y_{12} = \min_e \tau_e.$$

13. Правило минимума оставшихся операций:

$$Y_{13} = \min_e U(e, \kappa).$$

14. Правило максимальной оставшейся обработки:

$$Y_{14} = \max_e \tau_e.$$

15. Правило максимума оставшихся операций:

$$Y_{15} = \max_e U(e, \kappa).$$

16. Комбинация правил  $Y_3$  и  $Y_{14}$ . Применение вначале правила  $Y_3$  позволяет быстро включить станки, последующий переход к  $Y_{14}$  обеспечивает выполнение в срок наиболее длительных заказов.

17. Рандомизированные правила предпочтения. Предлагается сочетание детерминированных функций приоритета со случайным отбором.

### 18. Эвристические приемы.

Среди всего многообразия применяемых эвристических приемов отметим использования ранжировки /упорядочения/ как статистической, т.е. до начала процесса решения, так и динамической /с пересчетом "весов" объектов расписания на каждом шаге синтеза решения/. Здесь представляется перспективным метод двойного ранжирования [9], превосходящий метод "ветвей и границ" по качеству решения, так как использует больше семантической информации на каждом шаге решения.

Заканчивая краткое рассмотрение моделей и методов решения задач календарного планирования, заметим, что основные достижения в этой области впереди, в частности, при имитационном моделировании реальных производственных ситуаций.

### 3.3. Модель управления технологическим участком

Основной целью функционирования технологического участка является выпуск продукции объема  $V$  в течении планового периода  $T_{пл}$ . при заданных ресурсах  $\hat{R}$ . Кроме основной, задается некоторое множество целей  $\hat{z} = \{z_i\}$ , которые отражают реальную экономическую и производственную ситуации. Такими целями могут быть скорейший выпуск деталей, сокращение времени переналадки, минимальное отклонение от плановых сроков и т.д. При планировании работы участка эти цели заложены в алгоритм и полученное расписание обеспечивает их достижение при функционировании участка в стационарном режиме. При нарушении технологического процесса, реально возникающем в частности из-за конечной надежности его элементов, требуется вводить определенное управление, обеспечивающее достижение поставленных целей.

Объектом управления является совокупность исполнительных средств и технологических процессов. Входными переменными  $\{\bar{x}_i\}$  являются материальные потоки (заготовки, инструмент,

приспособления) и информационные - плановое задание (P), информация о незавершенном производстве, о заготовках, поступивших с внешних операций.

Управляемыми параметрами  $\{\hat{y}_i\}$  системы являются порядок и сроки запуска всех деталиеопераций из P.

Возмущения  $\{\hat{f}_i\}$  возникают от внутренних и внешних причин.

К внешним относятся: несвоевременная доставка на участок заготовок, полуфабрикатов, инструмента, оснастки; внеплановые заказы; нарушение сроков планового ремонта оборудования; отсутствие энергии; отсутствие вспомогательных материалов.

К внутренним причинам относятся: поломка инструмента, оборудования; невыход на работу или преждевременный уход рабочих; брак.

Процесс функционирования участка характеризуется параметрами партий в технологических процессах и состоянием оборудования. Динамический "портрет" партии задается набором параметров  $\pi_0 = \{O_e, O_e^{ik}, V_e^{ik}, q_e^{ik}, R_e^{ik}, a_e^t, t_{ek}^H, \hat{G}_1, \hat{G}_2, \hat{G}_3\}$ .

Параметры партий характеризуют различные ее состояния: количественные (число деталей в партии после каждой операции с учетом брака -  $a_e^t$ ); пространственные (местонахождение - склад, станок, транспорт -  $V_e^{ik}$ ); временные (моменты начала  $t_{ek}^H$  и окончания обработки); технологические (какой операцией  $O_e^{ik}$  и  $q_e^{ik}$  какого технологического процесса, с какой оснасткой роботом или рабочим  $R_e^{ik}$  - занята партия); логические (начат техпроцесс над партией или нет -  $\hat{G}_1$ , обрабатывается или ожидает -  $\hat{G}_2$ , годная или брак -  $\hat{G}_3$ ).

Динамические "портреты" состояния оборудования характеризуются следующими параметрами.

I. Обработывающее, манипуляционное и контрольное оборудование

$$\pi_1 = \{d_e, \theta^i, O_k^i, \hat{O}_k^i, \hat{P}_1, \hat{P}_2, \hat{P}_3, \hat{P}_4\},$$

где  $d_e, \theta^i, O_k^i, \hat{O}_k^i$  - параметры, определяющие партию, техпроцесс, операцию и оснастку, которыми занято оборудование  $v_n$ ;  $\hat{P}_1 \div \hat{P}_4$  логические переменные, указывающие на исправность оборудования ( $\hat{P}_1$ ), занятость ( $\hat{P}_2$ ), налаженность ( $\hat{P}_3$ ), работает или простаивает ( $\hat{P}_4$ )

2. "Портрет" транспортного оборудования

$$\pi_2 = \{d_e, \hat{P}_1, \hat{P}_2, \hat{P}_4, \epsilon_e\},$$

Характеризуется партией  $d_e$ , числом тар  $\epsilon_e$ , исправностью  $\hat{P}_1$  и занятостью  $\hat{P}_2$ , состоянием: работает или ожидает -  $\hat{P}_4$ .

3. "Портрет" рабочего  $\pi_3 = \{R_{22}, v_n, \Delta v, O_k^i, \hat{P}_5, \hat{P}_3, \hat{P}_2\}$  определяет, что он должен быть на рабочем месте у оборудования  $v_n$ , выполнять операцию  $O_k^i$ . Логические переменные определяют наличие рабочего на рабочем месте -  $\hat{P}_5, \hat{P}_3$  - ожидает ли работает,  $\hat{P}_2$  - занят или нет.

4. "Портрет" склада должен быть представлен двумя типами характеристик:

- для деталей  $\pi_4 = \{A_\epsilon, d_e, \epsilon_e, \hat{P}_2\}$  определяет занятость  $\hat{P}_2$  ячейки с адресом  $A_\epsilon$  партией  $d_e$  с числом деталей (или тар)  $\epsilon_e$ ;
- для оснастки  $\pi_5 = \{A_\epsilon, \hat{O}_e^i, \epsilon_e, \hat{P}_2\}$ , где  $\hat{O}_e^i$  - комплекты оснастки, хранящиеся в ячейке  $A_\epsilon$ .

"Портреты"  $\pi_1 \div \pi_5$  определяют динамическое состояние  $\hat{S}$  участка в любой дискретный момент времени  $\tau_g$ . Управление ведется не по временным интервалам  $\Delta t$ , а по существенным моментам  $\tau_g(\hat{S})$ , когда происходят события, скачкообразно изменяющие состояние системы,

Система оперативного управления производственным

участком может быть представлена в виде классической кибернетической системы управления с обратной связью.

Основными функциональными подсистемами являются:

- оперативный учет, который включает опрос, сбор и передачу первичной информации;
- оперативный контроль, т.е. анкету поступившей информации, выявление фактических и по возможности предполагаемых отклонений от плана; определение причин отклонений;
- оперативное регулирование, которое заключается в принятии оперативных решений, направленных на устранение несоответствия между запланированным и реальным ходом производства, и реализации принятых решений путем передачи управляющих воздействий.

На рис. 3.1. представлена функциональная структура такой системы управления.

На вход системы поступает целевое указание  $Z^*$ . Блок  $P_1$  на основании целеуказания, включающего плановое задание  $P = \{P_e\}$  и оптимизирующие требования, и обобщенной информации о состоянии объекта управления  $\Phi^* = \{\varphi_e^*\}$  составляет пооперационные календарные планы-графики работы участка на плановый период, формирует плановые задания  $P^* = \{P_k^*\}$  для всех компонент участка. Блок  $P_2$  осуществляет обработку и учет первичной информации, которая поступает в виде сигналов изменения состояния компонент участка  $K^* = \{K_s^*\}$ , и информации о неуправляемых, но наблюдаемых воздействиях на систему со стороны среды  $X^* = \{X_j^*\}$ . Элемент сравнения (блок  $P_3$ ) осуществляет оперативный контроль, сравнивая реальное состояние  $\hat{S} = \{S_k^*\}$  с планом  $P^*$ . Например, отклонения от запланированного времени начала или окончания текущей операции  $\Delta_e^{*ik}$ . Если степень рассогласования меньше некоторой

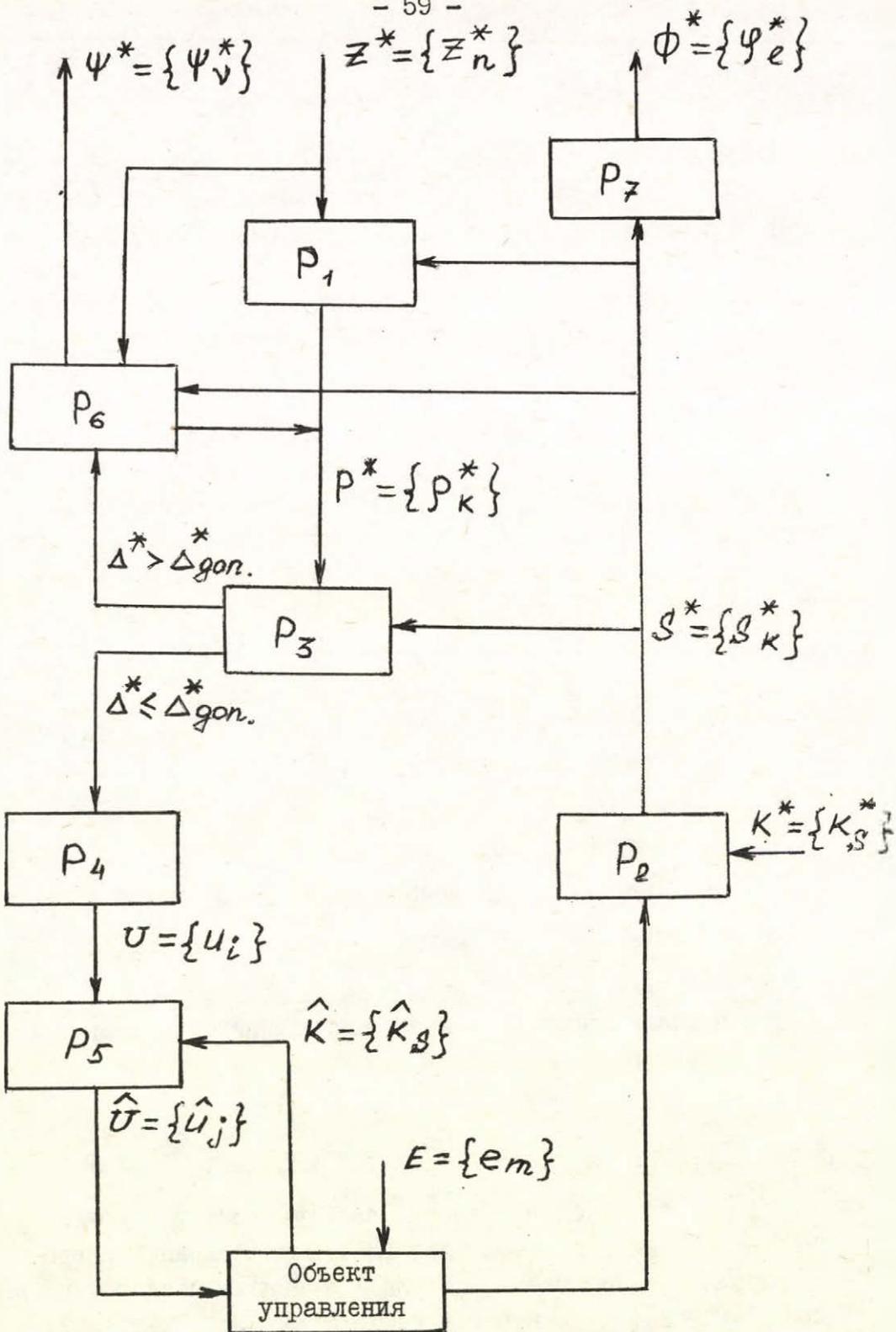


Рис. 3.1. Функциональная структура системы оперативного управления технологическим участком

допустимой величины ( $\Delta_e^{*ik} \leq \Delta_{гор}^*$ ), то информация о нем поступает в блок  $P_4$ , который осуществляет оперативное регулирование процесса, вырабатывая управление  $U = \{u_i\}$ . Ресурсами блока  $P_4$  являются временные резервы, заложенные в систему на этапе планирования. Блок 5 - это устройство локального управления оборудованием, обрабатывающее рабочие сигналы  $\bar{U}$ .

Если рассогласование  $\Delta_e^{*ik} > \Delta_{гор}^*$ , то информация поступает в блок  $P_6$ , который корректирует плановое задание  $P^*$  и выдает сообщение  $\Psi^*$  на верхний уровень управления. Превышение  $\Delta_{гор}^*$  означает нехватку временных резервов и потребность участка в дополнительных ресурсах. После выделения дополнительных ресурсов,  $P_6$  в интерактивном режиме с диспетчером участка решает следующие задачи:

- коррекцию и формирование нового планового задания на оставшийся период с учетом, возникших отклонений и директив диспетчера;
- коррекцию модели участка с учетом отклонений и дополнительных ресурсов;
- составление календарного плана на оставшийся плановый период.

Блок  $P_7$  осуществляет статистический учет. В системе таким образом реализуется принцип управления с обратной связью по состоянию  $\bar{S}$ .

#### 3.4. Внутрисменное оперативное управление

Рассмотрим более подробно работу блока  $P_4$  и использование временных резервов операций. Внутрисменное оперативное управление предназначено для ликвидации последствий нарушений, приводящих к изменению длительности операций. Метод управления [9] основан на коррекции сменно-суточных заданий за счет временных резервов расписания. Пусть задано

некоторое оборудование  $\ell$ . Определим локальный резерв операции на  $\ell$ -м объекте следующим образом:

$$\Delta \tau_{rk} = \tau_{r,k+1} - \bar{\tau}_{rk},$$

где  $\bar{\tau}_{rk}$  - момент начала операции, непосредственно следующей на данном оборудовании за данной  $k$ -й операцией. Если операция последняя в расписании для данного оборудования, то

$$\Delta \tau_{rk} = \max(\bar{\tau}_r, \bar{\tau}_{rke}) - \bar{\tau}_{rk}.$$

Локальный резерв - это интервал времени, на которой можно увеличить длительность операции, не сдвигая момента начала следующей за ней операции.

Минимальный локальный резерв партии - операции равен

$$\delta(\tau_{rk}) = \min_r \Delta \tau_{rk}.$$

Общий резерв  $k_i$ -й операции на  $\ell$ - партии определится по следующему рекуррентному соотношению:

$$\tau_{ek} = \begin{cases} \Delta \tau_{eke} & \text{для последней операции,} \\ \min(\Delta \tau_{ek} + \tau_{e,k-1}) & \text{для остальных.} \end{cases}$$

Можно показать, что увеличение длительности операции на величину  $\tau_{ek}$  не приведет к превышению  $\bar{\tau}_e$  ни для одной из партий  $\ell$  расписания, хотя и может привести к сдвигу вправо некоторых операций.

Для повышения устойчивости расписания введем порог чувствительности системы оперативного управления на изменение длительности операции -  $\delta_{ek}$ .

Тогда стратегия внутрисменного оперативного управления строится следующим образом.

Пусть длительность выполнения операции увеличилась на величину  $\Delta t_{ek}$  т.е.

$$\bar{t}'_{ek} = \bar{t}_{ek} + \Delta t_{ek}.$$

Тогда, если  $\Delta \bar{t}_{ek} \leq \delta_{ek}$ , то операция считается закончившейся в срок и продолжается управление по прежнему расписанию. При  $\delta_{ek} < \Delta t_{ek} < \Delta \bar{t}_{ek}$ , то отклонение не затрагивает других операций расписания.

Если же  $\Delta \bar{t}_{ek} < \Delta t_{ek} \leq T_{ek}$ , то необходима коррекция расписания путем сдвига вправо соответствующих операций за счет временных резервов календарного плана. В случае  $\Delta t_{ek} > T_{ek}$  необходимы дополнительные ресурсы.

#### 4. Пути повышения надежности ГАП

##### 4.1. Временное резервирование технических систем

##### 4.1.1. Источники временной избыточности

Для повышения надежности технических систем (в том числе и ГАП) широко применяются пять видов избыточности (резервирования): нагрузочная, структурная (аппаратурная), функциональная, информационная и временная. Если первые четыре вида избыточности уже давно находятся в центре внимания специалистов по надежности и каждому из них посвящена обширная техническая литература, то теоретические исследования временной избыточности начались сравнительно недавно (70-е годы). На наш взгляд именно временная избыточность позволяет повысить надежность функционирования ГАП, причем при отсутствии новых капитальных вложений в производство, лишь за счет некоторого снижения производительности системы.

О временной избыточности говорят в тех случаях, когда системе в процессе функционирования представляется возможность израсходовать некоторое время для восстановления ее технических характеристик.

Можно указать несколько основных источников резерва времени [12]. Прежде всего он может создаваться за счет увеличения времени, выделяемого системе для выполнения задания. Вторым основным источником является запас производительности, который позволяет уменьшить минимальное время выполнения задания и создать резерв без увеличения оперативного (рабочего) времени системы. Запас производительности можно образовать, увеличивая быстродействие элементов системы или объединяя несколько устройств низкой производительности в единый комплекс. В системах, результат работы которых оценивается объемом производимого продукта (что

имеет место в нашем случае), резерв времени можно создать за счет внутренних запасов выходной продукции. Пока запас не исчерпан, продукция поступает из специальных накопителей на выход системы и смежные с ней системы "не замечают" частичного или даже полного прекращения ее функционирования.

Четвертым источником резерва времени является функциональная инерционность. В работе многих технических систем допускаются незначительные перерывы без потери качества функционирования. Они могут использоваться, в частности, для устранения отказа подключением аппаратного резерва. Таким свойством обладает, например, подсистема диспетчерского управления ГАП.

Резерв времени можно расходовать не только на ремонт и переключение аппаратного резерва, но и на обнаружение отказов, повторение работ, обесцененных отказом, ожидание загрузки в работоспособном состоянии.

По характеру последствий все отказы можно разделить на три группы: необесценивающие, частично обесценивающие и полностью обесценивающие. Отказ считается необесценивающим, если система после восстановления работоспособности может возобновить работу с того же места, на котором она была прервана. В системе с необесценивающими отказами отсутствует необходимость в повторении работ и поэтому вся наработка между соседними отказами является полезной.

В системе с полностью обесценивающими отказами последствия настолько тяжелы, что приходится всю работу, сделанную к моменту отказа, выполнять заново. Вся наработка до возникновения отказа оказывается бесполезной, если она меньше заданной величины, и должна быть включена в потери рабочего времени. Полезной же признается только та часть наработки, которая не прерывалась отказами.

Частично обесценивающие отказы характерны для систем с периодическим контролем работоспособности, а также для некоторых систем с непрерывным контролем, у которых периодически фиксируются и сохраняются промежуточные результаты работы. Существует определенная аналогия между аппаратурной и временной избыточностью. Она проявляется, в частности, в том, что на временное резервирование можно распространить существующую классификацию и выделить такие способы резервирования, как: общее, групповое, отдельное, полное, частичное, целой и дробной кратности и т.д.

При общем резервировании выделяемый резерв времени можно израсходовать на восстановление работоспособности любого элемента системы. Отдельное резервирование характерно для многофазных систем, состоящих из нескольких последовательно соединенных устройств с промежуточными накопителями. Создавая запас продукции в своем выходном накопителе, каждое устройство обеспечивает себя собственным резервом времени, который не может использовать ни одно последующее устройство. В том случае, когда устанавливается один накопитель на группу устройств, имеет место групповое резервирование.

В некоторых системах значение выделяемого резерва времени устанавливается заранее, до начала работы, и он предназначен для компенсации любых потерь рабочего времени (этот случай будет подробно рассмотрен в дальнейшем изложении). При очередном нарушении работоспособности для устранения отказа и его последствий используется лишь та часть первоначального резерва, которая не была израсходована при предыдущих отказах. Такой резерв времени называют не-пополняемым, а систему, им обладающую, кумулятивной [12].

Резерв времени следует рассматривать как один из многих видов избыточности, которые можно вводить совместно или порознь для повышения надежности. Как и другие виды

избыточности, временная избыточность вносит новые элементы в трактовку основных понятий теории надежности и прежде всего понятия отказа.

#### 4.1.2. Отказы систем с временной избыточностью

В системах без временной избыточности отказом считается событие, состоящее в нарушении работоспособности. При этом отказавшая система не может выполнить задание. В системе с резервом времени нарушение работоспособности не означает срыва задания, так как в течение резервного времени она допускает восстановление. Поэтому отказом такой системы является событие, после возникновения которого система уже не способна выполнить задание при данных условиях эксплуатации. Отказы систем с резервом времени могут быть внезапными и постепенными, устойчивыми и самоустраняющимися (сбоями).

Внезапный отказ возникает в том случае, когда при нарушении работоспособности системы происходит обесценивание проделанной работы такого объема, что для ее повторения уже недостаточно оставшегося резерва времени.

Постепенные отказы являются наиболее распространенными, так как обычно резерв времени расходуется не мгновенно, а с момента нарушения работоспособности до момента, когда резерв оказывается исчерпанным, может пройти значительное время. Постепенный отказ может произойти вследствие как постепенного ухудшения технических характеристик, например, производительности, так и внезапного нарушения работоспособности.

Отказ называют устойчивым [ 5 ], если после окончания резерва времени на восстановление работоспособности затрачивается некоторое время. При наличии самоустраняющихся

отказов потери рабочего времени равны выделенному резерву.

Для фиксирования момента отказа необходимо непрерывно вести статистику потерь рабочего времени и иметь четкие признаки, по которым можно было бы своевременно зафиксировать срыв задания.

В невосстанавливаемой кумулятивной системе, в которой отказы элементов приводят к снижению производительности, срыв задания следует фиксировать в тот момент, когда производительность падает ниже допустимого уровня, зависящего от времени. Чтобы установить факт срыва задания в многоканальной кумулятивной системе, необходимо вести учет потерь рабочего времени во всех ее каналах. Если все каналы взаимозаменяемы, то срыв задания фиксируется в тот момент, когда суммарные потери времени во всех каналах достигают предельно допустимого уровня.

В многофазной системе при раздельном временном резервировании отказ есть событие, заключающееся в нарушении ее работоспособности, поскольку время выполнения задания в этом случае не увеличивается по сравнению с минимальным временем, затрачиваемым системой без резерва времени. И только при общем резервировании допускаются перерывы в работе. В этом случае признаки срыва задания формируются так же, как и в однофазных системах, опираясь на статистику простоев за время функционирования.

#### 4.1.3. Основные критерии и характеристики

Основным критерием надежности системы с временной избыточностью является вероятность  $P_{0\bar{t}}$  безотказного функционирования, т.е. вероятность того, что в течение оперативного времени не возникает отказ. Обозначая через  $T_0$  время до первого отказа системы, а через  $t$  оперативное время,

можно записать:

$$P_{\text{оф}} = P\{T_0 > t\}.$$

Это определение является основным и его можно использовать при любом способе временного резервирования и для любых технических систем. Для кумулятивных систем удобнее пользоваться следующим определением. Вероятностью безотказной работы называется вероятность того, что время выполнения задания  $t_{вз}$  не превысит допустимое, т.е.

$$P_{\text{оф}} = P\{t_{вз} \leq t\}.$$

Или, вероятность безотказного функционирования есть вероятность того, что полезное время  $t_n$  системы в оперативном интервале  $(0, t)$  достигнет или превзойдет некоторый заданный уровень  $t_3$ , равный минимальному времени выполнения задания при отсутствии отказов, т.е.

$$P_{\text{оф}} = P\{t_n(t) \geq t_3\}.$$

Вероятность  $P_{\text{оф}}$  можно представить как функцию трех аргументов: минимального времени  $t_3$  выполнения задания, оперативного времени  $t$  и совокупности  $\omega$  технических характеристик системы, в том числе и временных, которые определяют условия использования и пополнения резерва времени. Если вместо  $t$  задавать резервное время  $t_n$ , то  $P_{\text{оф}}$  выражается уже другой функцией -  $P(t_3, t_n, \omega)$ .

Вероятность отказа системы с временной избыточностью равна

$$Q(t_3, t_n, \omega) = 1 - P(t_3, t_n, \omega).$$

Частота и интенсивность отказов системы находятся из следующих выражений:

$$Q(t_3, t_n, \omega) = \frac{\partial}{\partial t_3} Q(t_3, t_n, \omega),$$

$$\lambda(t_3, t_n, \omega) = \frac{Q(t_3, t_n, \omega)}{P(t_3, t_n, \omega)} = -\frac{\partial}{\partial t_3} \ln P(t_3, t_n, \omega).$$

Средняя наработка до первого отказа определяется из интегрального соотношения

$$T_{cp}(t_n, \omega) = \int_0^{\infty} x d_x Q(x, t_n, \omega),$$

где  $d_x Q(x, t_n, \omega) = Q(x+dx, t_n, \omega) - Q(x, t_n, \omega)$

дифференциал функции  $Q(x, t_n, \omega)$  по аргументу  $x$ .

Если заявка на выполнение задания поступает в некоторый заранее известный момент (например, при управлении ГАП по расписанию) или произвольно выбранный момент (управление по текущему состоянию), то основным показателем надежности системы является вероятность безотказного функционирования при выполнении ожидаемой заявки или коэффициент готовности. По формуле полной вероятности имеем

$$P(t_3, t_n, \omega, \tau) = \sum_{i \in M} P_i(\tau) P^{(i)}(t_3, t_n, \omega, \tau),$$

где  $M$  - множество благоприятных состояний системы  
 $P_i(\tau)$  - вероятность того, что в начале оперативного интервала времени система окажется в состоянии  $i$  из множества  $M$ ;

$P^{(i)}(t_3, t_n, \omega, \tau)$  - вероятность безотказного функционирования при условии, что в начале оперативного интервала времени система находится в состоянии  $i$ .

В восстанавливаемых системах множество  $M$  совпадет с множеством всех возможных состояний.

При непрерывном множестве  $M$  сумма в последней формуле заменяется интегралом. Если момент  $\tau$  выбирается произвольно на стационарном участке функционирования, то при  $\tau \rightarrow \infty$  получаем

$$P(t_3, t_n, \omega) = \sum_{i \in M} P_i \cdot P^{(i)}(t_3, t_n, \omega),$$

где  $P_i$  - стационарные вероятности состояний.

Функция готовности за заданное время по определению будет равна:

$$K_r(t_n, \omega, \tau) = K_r(\tau) + (1 - K_r(\tau)) \cdot P\{t_B' \leq t_B^*(t_n, \omega)\},$$

где  $K_r(\tau)$  - вероятность застать систему в момент  $\tau$  в работоспособном состоянии;  $t_B'$  - интервал времени от момента  $\tau$  до момента восстановления работоспособности;  $t_B^*(t_n, \omega)$  - предельно допустимое значение  $t_B'$ .

Коэффициент готовности за заданное время как предельное значение функции готовности будет

$$K_r(t_n, \omega) = K_r + (1 - K_r) \cdot P\{t_B' \leq t_B^*(t_n, \omega)\}.$$

При статистическом определении коэффициент готовности за заданное время вычисляется следующим образом:

$$K_r^* = \left( \sum_{i=1}^n t_{H_i} + \sum_{i=1}^n t_{B_i}' \right) / \left( \sum_{i=1}^n t_{H_i} + \sum_{i=1}^n t_{B_i} \right),$$

где  $t_{H_i}$  - наработка между  $i-1$  и  $i$ -м отказами;  $t_{B_i}$  - время восстановления после  $i$ -го отказа;  $t_{B_i}'$  - часть интервала времени восстановления, зависящая от величины резерва времени и условий функционирования системы;  $n$  - количество отказов за время наблюдения за системой.

Составление данных уравнений ещё не является гарантией того, что удастся получить приемлимые для практического использования точные и даже приближенные формулы. Поэтому на практике широко распространены модели с ограничениями на систему, что позволяет получить инженерные методики. В разделе 5. приведена одна из них.

#### 4.2. Обеспечение надежности комплексов программ

##### 4.2.1. Методы повышения надежности функционирования программ

В процессе проектирования недостаточно создать правильные программы, выдающие необходимые результаты при идеальных исходных данных и абсолютном отсутствии возмущений. Требуется разработать надежные программы, устойчивые к различным возмущениям и способные сохранять требуемое качество обработки в реальных условиях функционирования.

Любые аномалии при исполнении программ необходимо блокировать и ограничивать по последствиям на уровне сбоя с максимально быстрым восстановлением.

Для этого в комплексы программ вводятся средства, решающие следующие задачи [ 7 ]:

- оперативное обнаружение искажений программ, вычислительного процесса, промежуточных и результирующих данных;
- селекцию и распознавание искажений, оценку возможных последствий для функционирования комплекса программ;
- принятие решения и выбор операций для восстановления нормального функционирования;
- реализацию принятого решения;
- регистрацию и накопление данных о выявленных искажениях и оперативно принятых мерах для их ликвидации или ограничения области влияния.

В табл. 4. представлены основные методы, позволяющие создавать надежное программное обеспечение. Основное внимание в дальнейшем изложении будет уделено использованию избыточности, обеспечивающей защиту вычислительного процесса и информации программно-алгоритмическими методами.

## Проблемная классификация надежности программного обеспечения

Факторы, определяющие надежность программ	Методы проектирования надежных комплексов программ	Методы повышения надежности программ за счет избыточности
1	2	3
<p>Особенности объектов управления:</p> <ul style="list-style-type: none"> <li>- требования к показателям надежности,</li> <li>- инерционность объектов,</li> <li>- необходимое время реакции,</li> <li>- средний темп обмена сообщениями.</li> </ul> <p>Характеристики искажений исходных данных:</p> <ul style="list-style-type: none"> <li>- искажения, обусловленные взаимодействием человека с ЭВМ,</li> <li>- искажения в системах передачи данных,</li> <li>- искажения данных при накоплении и хранении.</li> </ul>	<p>Структурное проектирование программ и данных:</p> <ul style="list-style-type: none"> <li>- программных модулей,</li> <li>- комплексов программ и взаимодействия модулей,</li> <li>- массивов данных,</li> <li>- контроль выполнения правил структурирования.</li> </ul> <p>Тестирования программ:</p> <ul style="list-style-type: none"> <li>- детерминированное,</li> <li>- статистическое,</li> <li>- контроль пропускной способности в реальном времени.</li> </ul>	<p>Контроль состояния и функционирования программ:</p> <ul style="list-style-type: none"> <li>- использование избыточности: <ul style="list-style-type: none"> <li>- программной,</li> <li>- информационной,</li> <li>- временной,</li> </ul> </li> <li>= контроль программ, данных и решения задач при профилактических работах,</li> <li>= предстартовый контроль,</li> <li>= оперативный контроль.</li> </ul> <p>Программное восстановление:</p> <ul style="list-style-type: none"> <li>- текстов программ,</li> <li>- исправление искаженных данных,</li> <li>- корректировка вычислительного процесса.</li> </ul>

Таблица 4.  
Продолжение

I	2	3
<p>Характеристики ошибок в программах:</p> <p>= статистические характеристики:</p> <ul style="list-style-type: none"> <li>- программ,</li> <li>- массивов данных,</li> <li>- вычислительного процесса,</li> </ul> <p>-- характеристики классов ошибок:</p> <ul style="list-style-type: none"> <li>- программных,</li> <li>- алгоритмических,</li> <li>- системных.</li> </ul>	<p>Испытания и определение надежности программ:</p> <ul style="list-style-type: none"> <li>- экспериментальное определение надежности в нормальных условиях функционирования,</li> <li>- форсированные испытания,</li> <li>- расчетно-экспериментальные методы определения надежности программ.</li> </ul>	

Под временной избыточностью понимается использование части производительности ЭВМ для контроля исполнения программ и восстановления вычислительного процесса.

При проектировании комплекса программ должен быть предусмотрен запас производительности, который будет использоваться для контроля и повышения надежности функционирования. Значение временной избыточности зависит от требований к надежности функционирования системы и составляет от 5-10% производительности однопроцессорной ЭВМ до трех- и четырехкратного дублирования производительности ЭВМ в мажоритарных вычислительных комплексах.

Информационная избыточность состоит в дублировании исходных и промежуточных данных, обрабатываемых комплексом программ. Избыточность используется для обеспечения достоверности данных, которые в наибольшей степени влияют на нормальное функционирование комплекса программ и требуют значительного времени для восстановления. Данные защищаются двух- и трехкратным дублированием с соответствующей дисциплиной контроля сохранности и периодическим обновлением. Для менее важных данных информационная избыточность используется в виде помехозащитных кодов, позволяющих только обнаружить искажение.

Искаженные данные исключаются из обработки и происходит их естественное обновление в процессе последующего функционирования.

Программная избыточность используется для контроля и обеспечения достоверности важных решений по управлению и обработке информации. Она заключается в применении в комплексах программ нескольких вариантов программ, различающихся методами решения задачи или программной реализацией одного того же метода. Тем самым создается возможность сопоставить результаты обработки одинаковых исходных данных разными программами и исключить

искажение результатов, обусловленное программными ошибками или сбоями ЭВМ. Программная избыточность необходима для реализации программы контроля и оперативного восстановления данных с использованием информационной избыточности и для функционирования средств защиты, использующих временную избыточность.

#### 4.2.2. Резервирование программ и данных

Искажения программ и данных являются одной из причин, существенно снижающих надежность функционирования программ. Методы защиты от искажений базируются на использовании информационной избыточности и многократного резервирования защищаемых данных или программ. В зависимости от этапов жизненного цикла программ к их защите предъявляются разные требования и применяются разные методы.

В процессе разработки программ корректируются и при завершении отладки изменениям подвергается очередная версия. Для гарантии сохранения программ изменяемая версия имеет несколько копий, а кроме того, хранятся несколько предыдущих версий. Длительность комплексной отладки и испытаний может достигать 2-3 лет, однако предистории обычно бывают не старше полугода и их число не более двух. Разрушение программ чаще всего обусловлено необнаруженными ошибками. Допустимая вероятность разрушения всех магнитных копий во время отладки должна быть обеспечена на уровне  $10^{-4}$ . [ 6 ].

В процессе эксплуатации и сопровождения комплекса программ корректировки проводятся эпизодически для устранения обнаруженных ошибок или для расширения функций программ и улучшения их качества.

Периодичность изменений базовой версии для различной комплектации комплекса программы может составлять 0,5 - 1 год и иметь 2-3 предыстории. Сложные комплексы программ управления имеют время "жизни" 5-10 лет и должны сохраняться на этом интервале с вероятностью около 0,9999.

В [6] проведен анализ трех стратегий резервирования информационных массивов (программ или данных), которые наиболее часто встречаются на практике.

В первой стратегии используется  $\tilde{K}$  тождественных копий массивов. При разрушении основного массива последовательно применяются очередные копии для восстановления всех копий.

Вторая стратегия характеризуется применением только предысторий для обновления и восстановления текущих данных. Каждая предыстория содержит данные на некоторый предшествующий момент времени и все изменения, внесенные до момента формирования следующей предыстории. Если разрушился массив текущих данных, то его можно восстановить по ближайшей предыстории.

Третья стратегия наиболее часто используется на практике и базируется на хранении копий и предысторий. При разрушении массива первые попытки восстановления производятся с использованием  $X$  копий, а при разрушении всех копий последовательно применяются  $Y$  предысторий.

Причем  $\tilde{K} = X + Y$ .

Для всех стратегий считается, что известна вероятность разрушения  $q = 1 - p_m$  за единичный интервал использования. При хранении без функционирования массивы не раз-

рушаются. Время создания одной копии равно  $t_k$ , периодичность решения задачи  $\Theta$ .

Для оценки стратегий применяется два критерия: вероятность успешного решения -  $P$  и средняя продолжительность использования ЭВМ для решения основных функциональных задач  $E$ .

$$P_1 = 1 - q^{\tilde{k}+1}; \quad E_1 = \tilde{k} \cdot t_k + \Theta \cdot P_M^{-1} (1 - q^{\tilde{k}+1});$$

$$P_2 = P_M (P_M^{\tilde{k}+2} - q^{\tilde{k}+2})^{-1} \cdot (P_M^{\tilde{k}+1} + q^{\tilde{k}+1});$$

$$E_2 = \frac{\Theta}{q - P_M} \left\{ \tilde{k} + 1 - \frac{(\tilde{k} + 2) \cdot [1 - (q \cdot P_M^{-1})^{\tilde{k}+1}]}{1 - (q \cdot P_M^{-1})^{\tilde{k}+2}} \right\};$$

$$P_3 = 1 - (q^{y+2} - P_M^{y+2})^{-1} \cdot q^{\tilde{k}+1} \cdot (q - P_M);$$

$$E_3 = x \cdot t_k + \Theta \cdot P_M^{-1} (1 - q^x) + \frac{\Theta}{q - P_M} \cdot x \cdot \left\{ y + 1 - \frac{(y + 2) \cdot [1 - (q \cdot P_M^{-1})^{y+1}]}{1 - (q \cdot P_M^{-1})^{y+2}} \right\} \cdot q^x$$

Очевидно, что повышение вероятности неразрушения массивов за счет резервирования ведет к ухудшению временных показателей и снижению полезной производительности ЭВМ. Из приведенных формул видно, что вероятность неразрушения всех массивов монотонно возрастает при увеличении вероятности сохранения каждого массива. Стратегия I всегда лучше стратегии 3, которая лучше стратегии 2, что обусловлено увеличе-

нием затрат на содержание предысторий и увеличение вероятности их разрушения по сравнению с использованием копий.

При  $P_M = 0,9$  и сохранении четырех копий вероятность разрушения всех массивов снижается до  $10^{-5}$

#### 4.2.3. Временная избыточность программного обеспечения

Объекты с временным резервированием для анализа их надежности можно представить в виде условной схемы "объект-время" [  $\neq$  ], состоящей из двух параллельно включенных компонент: технического объекта и резерва времени. Такая схема аналогична схеме ненагруженного однократного дублирования замещением, в котором при отказе объекта включается резерв времени.

Для комплексов программ управления ГАП в реальном времени наиболее адекватна схема с мгновенно пополняемым резервом времени и наличием допустимого времени  $t_g$ , в пределах которого отсутствие работоспособности не учитывается как отказ. Затраты времени на контроль и восстановления, не превышающее  $t_g$ , рассматриваются как соответствующие неработоспособному состоянию, не входящему в состояние отказа. Резерв времени пополняется до исходного значения независимо от числа предыдущих отказов, времени на их устранение и наработки на отказ.

Суммарное время восстановления работоспособности не ограничено.

Коэффициент готовности  $K_r^*(T, t_g)$  такой системы включает в себя два слагаемых: вероятность того, что в момент поступления задачи на обработку комплекс программ окажется в работоспособном состоянии  $K_r$  - и вероятность

того, что задача застанет программы в состоянии контроля и восстановления, однако эти операции закончатся до использования допустимого резерва времени  $t_g$ :

$$K_r^*(t_g, T) = K_r + (1 - K_r) \cdot P\{t_B \leq t_g\} = K_r + (1 - K_r)(1 - q),$$

где  $P\{t_B \leq t_g\}$  вероятность проведения всех работ по контролю и восстановлению за время, меньшее  $t_g$ , а  $q$  - вероятность того, что отказовая ситуация приведет к полному отказу при функционировании комплекса программ.

Предположим, что реализован дискретный и достоверный контроль работоспособности и отказы возникают только в рабочем режиме. Нарботка между соседними отказовыми ситуациями  $T$  имеет такое же распределение, как и наработка до первого отказа. Нарботка  $T$  распределена экспоненциально, а среднее время восстановления работоспособности равно  $t_3$ . Отказ в комплексе программ фиксируется и учитывается, если сумма случайных величин времени ожидания до обнаружения отказа  $t_2$  и времени восстановления  $t_3$  превышает  $t_g$ .

В [ 5 ] приведены расчетные соотношения:

$$K_r = \frac{1 - \exp(-t'/T)}{t'/T + (t_3/T)[1 - \exp(-t'/T)]},$$

$$q = \begin{cases} 1 - \frac{T \cdot (\exp(t_g/T) - 1) - t_3(1 - \exp(-t_g/t_3))}{[\exp(t'/T) - 1] \cdot (T + t_3)}, & t_g \leq t' \\ \frac{\exp(-t_g/t_3) \cdot t_3}{[\exp(t'/T) - 1] \cdot (T + t_3)} \cdot \left[ \exp\left(\frac{t' \cdot (T + t_3)}{T \cdot t_3}\right) - 1 \right], & t_g \geq t' \end{cases}$$

где  $t'$  - период контроля.

При увеличении  $t_g$  уменьшается количество отказовых ситуаций. В простейшем случае можно предположить, что  $t_s \leq T, t' \leq T$  и контроль производится мгновенно. При этом временное резервирование изменяет частотность отказовых ситуаций на величину  $q$ , равную отношению коэффициентов простоя при  $t_g = 0$  и выбранном значении  $t_g$ :

$$q = (1 - K_r^*) / (1 - K_r).$$

В результате ввод временного резервирования  $t_g$  повышает наработку на отказ до величины  $T(t_g) = T/q$ . В общем случае для определения наработки на отказ необходимо вычислить вероятность  $q$  отнести искажение к отказу и учесть все затраты времени на интервале функционирования комплекса программ между регистрируемыми отказами.

## 5. Методика расчета временной избыточности

### 5.1. Количественный анализ ГАП с временной избыточностью

Итак, с позиций теории надежности, ГАП представляет собой контролируемую восстанавливаемую систему. Произведем оценку характеристик надежности системы и свяжем эти характеристики с производительностью и способами организации контроля правильности функционирования.

Сформулируем задачу следующим образом [2].

Требуется выполнить одним устройством производственное задание объема  $V$  за время  $t$ . Величины  $V$  и  $t$  измеряются в одних и тех же единицах. (Это могут быть машино-часы, машино-минуты или любые другие единицы, в которых измеряется и планируется работа данной ГАП). Под устройством понимается единая, неделимая и не различимая в конкретной постановке часть системы, например, обрабатывающий центр, станок с ЧПУ, включающие как собственно обрабатывающие устройства с манипуляторами и транспортной системой, так и систему автоматизированного управления ими.

Задание  $V$  разбивается на  $n$  равных этапов, выполняемых последовательно. Для механообработки в этап включается собственно время обработки детали и подготовительно-заключительное время. После каждого этапа тем или иным способом проводится проверка результатов работы на этапе и в случае необходимости либо восстановительной (если имел место отказ системы), либо профилактический ремонт.

Поток отказов устройства принимается пуассоновским, интенсивность отказов -  $\lambda$ . Время проверки  $\tau_n$  и профилактического ремонта и время восстановительного ремонта  $\tau_B$  принимается постоянными и равными  $d$  и  $h$  соответственно.

В общем случае можно считать, что эти времена являются случайными с математическими ожиданиями  $M[\tau_n] = d$ ,

$$M[\tau_B] = h.$$

При этих условиях определим:

1. Вероятность  $P_z(v, t)$  выполнения производственного задания объема  $V$  за время  $t \geq V$  при  $z$  этапах.
2. Математическое ожидание  $M[T]$  рабочего времени системы, затраченного на выполнение задания  $V$  при  $z$  этапах.

Рассмотрим несколько возможных вариантов организации работ ГАП, учитывающих типы отказов, возникающих в системе.

5.1.1. Отказ обесценивает работу, сделанную на всех предыдущих этапах.

Схема организации выполнения производственного задания в данном случае приведена на рис. 5.1.А.

Выполнение задания начинается в условной момент времени  $t=0$ . В конце каждого этапа, в том числе и последнего, производится проверка. Если проверка дала положительный результат, то система переходит к выполнению следующего этапа, в противном случае, система переходит в режим восстановления. Результаты работы всех предшествующих этапов обесцениваются и на следующем этапе система начинает выполнение всего задания сначала. В механообработке подобная ситуация возникает в ГАП с конвейерным способом организации работы, когда ритм движения конвейера постоянный, операции над деталью или партией деталей производятся за одно и тоже время, а наличие отказа /брака/ на любой из операций заставляет удалять деталь с конвейера и начинать обработку снова.

Таким образом, задание будет выполнено, если за время  $t$  система будет успешно работать подряд  $z$  этапов /т.е. весь технологический цикл/. Определим для данного случая вероятность

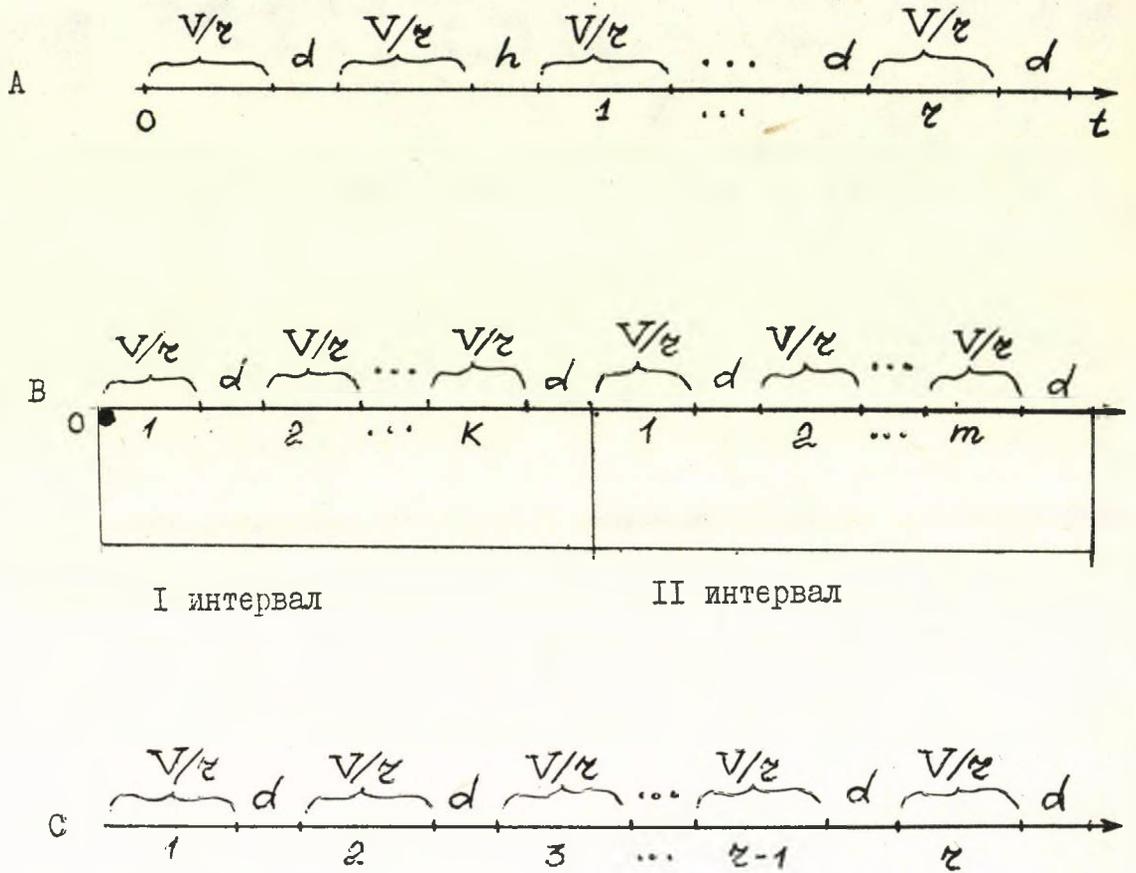


Рис. 5.1. Организация процесса функционирования ГАП при выполнении задания

Примем  $\tau_n = \tau_z = d = h$ . Длительность каждого этапа  $t_0 = V/z + d$ . Вероятность успешной работы системы на каждом этапе будет равна

$$p = p(V/z) = \exp(-\lambda \cdot V/z) \quad (5.1)$$

При этом принимается, что во время проведения диагностики и ремонтов новых отказов не возникает.

За время  $t$  в этом случае будет возможным проведение  $n$  этапов

$$n = [t / (V/z + d)], \quad (5.2)$$

где  $[ \cdot ]$  - целая часть подскобочного выражения. Т.е. если будет определена вероятность  $P_z(n)$  того, что среди  $n$  возможных за время  $t$  этапов будет хотя бы одна последовательность из  $z$  успешных этапов, то будет определена искомая вероятность  $P_z(V, t)$ . Для подобной задачи В. Феллером [13] было определено приближенное решение:

$$P_z(n) \approx 1 - \frac{1 - p \cdot x_0}{q(1 + z - z \cdot x_0)} \cdot \frac{1}{x_0^{n+1}}, \quad (5.3)$$

где  $p$  - вероятность успешности одного этапа, а  $x_0$  - корень уравнения:

$$q \cdot x \cdot (1 + p \cdot x + \dots + p^{z-1} \cdot x^{z-1}) = 1, \quad q = 1 - p.$$

Формула (5.3) дает хорошие приближения при больших  $n$  и для  $p$  не слишком близких к единице. В практических приложениях наибольший интерес представляют случаи, когда  $p$  близко к единице, а  $n$  не очень велико. В [ ] предложен метод точного вычисления  $P_z(n) = P_z(V, t)$ .

$$P_z(n) = P_z(n-1) + q \cdot \rho^z [1 - P_z(n-z-1)], \quad n > z \quad (5.4)$$

Практическое вычисление  $P_z(n)$  для небольших  $n$  можно вести по следующим рекуррентным соотношениям:

$$P_z(0) = P_z(1) = \dots = P_z(z-1) = 0;$$

$$P_z(z+k) = \rho^z (1 + kq), \quad k = 0, 1, \dots, z;$$

$$P_z(2z+i) = \rho^z \left\{ 1 + q \left[ i + z - i\rho^z - \frac{i(i-1)}{2} q\rho^z \right] \right\},$$

$$i = 1, \dots, z.$$

Для случая, когда  $d = \text{const}$  и  $h = \text{const}$  в [2] приведена формула определения математического ожидания  $M[T]$  случайного времени  $T$  выполнения задания объема  $V$ , разбитого на  $z$  этапов

$$M[T] = \frac{(1 - \exp(-\lambda V))}{(1 - \exp(-\lambda V/z)) \cdot \exp(-\lambda V)} \cdot$$

$$\left[ \exp(-\lambda V/z) \cdot d + (1 - \exp(-\lambda V/z)) \cdot h + V/z \right] \quad (5.5)$$

### 5.1.2. Отказ обесценивает результаты работы, проделанной на одном этапе

Схема организации выполнения системой производственного задания в этом случае представлена на рис. 5.1.В

Данный случай в реальных ГАП наиболее характерен при оценки работоспособности систем управления. Проверка правильности работы системы управления обычно производится путем повторения вычислений до появления двух совпадающих результатов. Совпадающие результаты считаются правильными. Организация каждой проверки - передача данных, сравнение, выдача команд на повторный счет - занимает время, равное  $d$ .

За это же время  $d$  в случае необходимости должна быть восстановлена работоспособность системы управления. Из рис. 5.1.В. видно, что весь цикл управления состоит из  $z$  интервалов. На каждом интервале случайное число раз (до появления двух совпадающих результатов) повторяются вычисления на одном из этапов. Длительность работы на этапе  $t_2 = V/z + d$ . По-прежнему, считаем, что на промежутках  $d$  сбои и отказы не возникают, а в начальный момент система исправна. За время  $t$  можно провести  $n$  первоначальных вычислений и проверок:

$$n = [t / (V/z + d)],$$

где  $[ \cdot ]$  - целая часть.

Чтобы успеть выполнить все задание за время  $t$ , на каждом из  $z$  интервалов должно быть не менее 2 успешных проверок, т.е. за весь период работы допускается не более чем  $n - 2z$  отказов.

В этом случае, используя биномиальный закон, искомая вероятностей определяется по следующей формуле:

$$P_z(V, t) = \sum_{k=2z}^n C_n^k \exp(-k\lambda V/z) [1 - \exp(-\lambda V/z)]^{n-k} \quad (5.6)$$

А учитывая тот факт, что продолжительность каждого минимального этапа функционирования равна  $V/z + d$ , математическое ожидание времени  $M[\tau]$ , необходимого для выполнения всего задания будет равно:

$$M[\tau] = 2(V + z \cdot d) \cdot \exp(\lambda V/z) \quad (5.7)$$

### 5.1.3. Поэтапная проверка абсолютно надежна

На рис. 5.1.С. представлена схема организации выполнения задания в данном случае.

После завершения работы на каждом этапе с помощью абсолютно надежных устройств диагностики проводится проверка правильности работы системы. В случае обнаружения неправильного результата работа на данном этапе должна быть повторена. Для ГАП типа механообработки это соответствует широко распространенному случаю обработки деталей или партий деталей на специализированном оборудовании. Заготовки подаются из магазина, имеющего определенный страховой запас, а передача обработанной детали производится только после ее проверки. Если обнаружен брак, то деталь на дальнейшую обработку не поступает из магазина берется новая заготовка и выполнение операции повторяется.

Для выполнения производственного задания в течении времени  $t$  необходимо, чтобы из возможного числа этапов  $n$  не более чем  $n - z$  были неуспешными. В соответствии с этим искомая вероятность определится следующим образом:

$$P_z(V, t) = \sum_{k=z}^n C_n^k \cdot \exp(-k\lambda V/z) [1 - \exp(-\lambda V/z)]^{n-k} \quad (5.8)$$

Математическое ожидание времени, необходимого для выполнения всего задания, будет равно:

$$M[\tau] = (V + z \cdot d) \cdot \exp(\lambda V/z). \quad (5.9)$$

Сравнивая три варианта организации диагностики системы управления и обрабатывающего оборудования при наличии временной избыточности можно сделать вывод, что наиболее предпочтителен третий вариант - с проведением проверки после выполнения каждого этапа.

## 5.2. Способы использования временной избыточности

Итак, можно считать общепризнанным [5, 12], что временная избыточность представляет собой наиболее эффективный, экономически целесообразный способ повышения надежности автоматизированных систем. Ряд авторов [9] утверждает, что временная избыточность обязательно присутствует в любой реальной системе. В качестве источников появления временной избыточности указываются межоперационные пролеживания и внутренние простои. Но, если учесть современные тенденции в организации функционирования ГАП, то станет ясным, что эти источники временной избыточности не являются регулярными, более того система календарного планирования в качестве одного из основных оптимизационных критериев оперирует именно минимизацией временных потерь подобного типа. По нашему мнению, временная избыточность должна вводиться в расписание работы системы в плановом порядке. Рассмотрим один из возможных способов определения величины временной избыточности.

По-прежнему требуется выполнить задание объемом  $V$  за время  $t$ . Задание  $V$  разбито на  $n$  равных этапов (элементарных работ, операций).

После каждого этапа с помощью абсолютно надежных устройств контроля проводится проверка правильности функционирования системы на данном этапе. В случае обнаружения неправильного результата данный этап повторяется (т.е. рассматривается схема организации функционирования системы рис. 5.1.С).

В отношении системы сделаем следующие предположения:

1. В начальный момент система исправна.
2. В перерывах между проведением этапов выполнения задания новых сбоев и отказов не происходит.
3. Поток отказов, приводящий к неправильному функционированию, применяется пуассоновским, интенсивность отказов -  $\lambda$ .

4. Восстановление работоспособности системы происходит в перерывах между выполнениями этапов. К моменту начала следующего этапа, либо повторения данного система исправна.
5. Задана требуемая работоспособность системы, т.е. вероятность  $P_n(V, t)$  того, что задание объема  $V$  будет выполнена к моменту времени  $t$ .

Т.е. мы пришли к решению обратной задачи, известна левая часть уравнения (5.8), требуется определить  $n$ . Решая уравнение (5.8) относительно  $n$ , получим общее количество этапов, которое с заданной вероятностью система успеет выполнить за время  $t$ . Тогда количество избыточных, резервных этапов будет равно  $n - z$ . Решение уравнения (5.8) относительно  $n$  возможно путем последовательной подстановки целочисленных значений  $n$ , начиная с  $n = z, z+1, \dots$

В работе [ 4 ] предложено определять данную величину, исходя из моделирования производственной системы простой цепью Маркова. Очевидно, возможны отличные от предложенных способы расчета временных резервов, учитывающие более сложные потоки отказов и не накладывающие столь строгие ограничения на модель системы.

С инженерной точки зрения предлагаемая методика сводится к следующему. На основе статистического анализа функционирования ГАП или предпроектного анализа системы определяются операции или функции, наиболее сильно влияющие на конечную работоспособность всей системы.

Задается требуемая надежность функционирования и, зная вид потока отказов и его интенсивность, рассчитывается количество резервных этапов-операций.

Для ГАП типа механообработки это означает увеличение времени обработки партии деталей на резервируемой операции. При календарном планировании функционирования ГАП в качестве времени обработки рассматривается уже это - суммарное время. Естественно, увеличение производственного цикла снижает производительность системы, но в каждом конкретном случае необходимо решить (с экономической точки зрения) что важнее: производительность или гарантированный результат, вынужденный простой системы при отсутствии брака или дополнительные капитальные затраты на структурное резервирование оборудования и систем управления.

### Заключение

Успех автоматизации управления производством любого уровня зависит от достоверности информации, надежности функционирования всех компонент системы, организации и экономической эффективности.

Создание гибких автоматизированных систем предполагает пристальное внимание к этим проблемам. Исходя из рассмотренной концепции гибких автоматизированных систем и анализа надежности функционирования оборудования, средств вычислительной техники и программного обеспечения, можно сделать вывод, что в качестве комплексного метода повышения эффективности ГАП может рассматриваться временная избыточность.

Предлагаемая методика расчета и использования временной избыточности обеспечивает с заданной вероятностью выполнение системой производственного задания в отведенное время. Используемые в работе математические модели построены при определенных предположениях, в частности, о стационарности процессов восстановления работоспособного состояния. Это, безусловно, идеализирует модели, но позволяет получить соотношения, пригодные к использованию в инженерной практике. Широкое внедрение гибких автоматизированных систем в реальное производство позволит получить статистический материал о параметрах потоков сбоев и отказов и усложнить используемые модели.

Авторы считают своим приятным долгом выразить благодарность академику Вामошу Т., явившемуся инициатором данной работы, доктору технических наук Шомло Л., взявшему на себя труд по рецензированию, а также профессору Игнатьеву М.Б. и доценту Барта Д. за плодотворные дискуссии по проблемам надежности и управления гибкими автоматизированными системами.

Указатель литературы

1. Buzacott J.A. The Fundamental Principles of Flexibility in Manufacturing Systems. Proceedings of the 1th International Conference of Flexible Manufacturing Systems. Brighton, U.K, 1982, p.13-22.
2. Гаркави А.Л., Гоголевский В.Б., Грабовецкий В.П. Надежность контролируемых восстанавливаемых устройств с временной избыточностью. В кн: Теория надежности и массовое обслуживание. М., Наука, 1969, с. 108-118
3. Конвей Р.В., Масквелл В.А., Миллер Л.В. Теория расписаний. М., Наука, 1975.
4. Кочиш Я., Фетисов В.А. Об одном методе повышения надежности функционирования автоматизированных систем. *Periodica Polytechnica*. 1984. (в печати)
5. Креденцер Б.П. Прогнозирование надежности систем с временной избыточностью. К., Наукова думка, 1978.
6. Кульба В.В., Пелихов В.П., Шелков А.Б. Стратегии резервирования информационных массивов. - В кн: Сборник трудов Института проблем управления. М., 1978, вып. 16 с. 26-42.
7. Липаев В.В. Надежность программного обеспечения АСУ. М, Энергоиздат, 1981.
8. Musa I.A. The theory of software reliability and its applicotions. *IEEE Transactions of software Engineering*. 1975, vol. SE - 1, 3, p. 312 -327.
9. Мясников В.А., Игнатъев М.Б., Перовская Е.И., Модели планирования и управления производством. М., Экономика, 1982 г.
10. Parnas D.L. The influence of software structure on reliability. *Proceedings of the International Conference on Reliable Software Los Angeles*, 1975, p. 358-362.

11. Schick G. I, Wolverson R.W. An analysis of Computing software reliability models. I IEEE Transactions of Software Engineering, 1978, vol. SE-4, 2, p. 104 - 119.
12. Черкасов Г.И. Надежность технических систем с временной избыточностью. М, Советское радио, 1974.
13. Феллер В. Введение в теорию вероятностей и ее приложение. М, Мир, 1964.

Kocsis J. - a műszaki tudományok  
kandidátusa, BME Számítástechnikai  
Alkalmazási Központ

Budapest, 1111  
Műgyetem rkp.  
9. R. ép. III. em. 11

Фетисов В.А. - кандидат технических  
наук  
Старший преподаватель кафедры ЭВМ  
Ленинградского института авиацион-  
ного приборостроения

С С С Р, 190000  
Ленинград  
Ул. Герцена, 67

A TANULMÁNSOROZATBAN 1983-BAN MEGJELENTEK

- 140/1983 Operation Research Software Descriptions (Vol.1.)  
Szerkesztette: Prékopa András és Kéri Gerzson
- 141/1983 Ngo The Khanh: Prefix-mentes nyelvek és egyszerű  
determinisztikus gépek
- 142/1983 Pikler Gyula: Dialógussal vezérelt interaktív  
gépészeti CAD rendszerek elméleti és gyakorlati  
megfogalmazása
- 143/1983 Márkus Zsuzsanna: Modellelméleti és univerzális  
algebrai eszközök a természetes és formális nyelvek  
szemantikaelméletében
- 144/1983 Publikációk '81 /Szerkesztette: Petróczy Judit/
- 145/1983 Telcs András: Belső állapotú bolyongások
- 146/1983: Varga Gyula: Numerical Methods for Computation of  
the Generalized Inverse of Rectangular Matrices
- 147/1983 Proceedings of the joint Bulgarian-Hungarian  
workshop on "Mathematical, Cybernetics and data  
Processing" /Szerkesztette: Uhrin Béla/
- 148/1983 Sebestyén Béla: Fejezetek a részecskefizikai  
elektronikus kísérleteinek adatgyűjtő, -feldolgozó  
rendszerei köréből
- 149/1983 L. Keviczky, J. Hethéssy: A general approach for  
deterministic adaptive regulators based on explicit  
identification
- 150/1983 IFIP TC.2 WORKING CONFERENCE "System Description  
Methodologies" May 22-27. 1983. Kecskemét.  
/Szerkesztette: Knuth Előd/

- 151/1983 Márkus Zsuzsanna: On First Order Many-Sorted LOGIC
- 152/1983 Operations Research Software Descriptions /Vol.2./ Edited by A. Prékopa and G. Kéri
- 153/1983 T.M.R. Ellis: The automatic generation of user-adaptable application-oriented language processors based on quasi-parallel modules

#### 1984-BEN MEGJELENTEK

- 155/1984 Deák, Hoffer, Mayer, Németh, Potecz, Prékopa, Straziczky: Termikus erőműveken alapuló villamos-energiarendszerek rövidtávu, optimális, erőművi menetrendjének meghatározása hálózati feltételek figyelembevételével.
- 156/1984 Radó Péter: Relációs adatbáziskezelő rendszerek összehasonlító vizsgálata
- 157/1984 Ho Ngoc Luat: A geometriai programozás fejlődései és megoldási módszerei
- 158/1984 PROCEEDINGS of the 3rd International Meeting of Young Computer Scientists.  
Edited by J. Demetrovics, and J. Kelemen
- 159/1984 Bertók Péter: A system for monitoring the machining operation in automatic manufacturing systems
- 160/1984 Ratkó István: Válogatott számítástechnikai és matematikai módszerek orvosi alkalmazása  
/kandidátusi értekezés/
- 161/1984 Hannák László: Többértékű logikák szerkezetéről



