ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»

К.Е. КЛИМЕНТЬЕВ

СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

САМАРА 2008

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»

К.Е. КЛИМЕНТЬЕВ

СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

Утверждено Редакционно-издательским советом университета
в качестве обзорного курса лекций

С А М А Р А
Издательство СГАУ
2008

УДК СГАУ/ 004 (075)
ББК 32.97
К 492

Рецензент канд. техн. наук. В.Г. Иоффе

Климентьев К.Е.
Системы реального времени: обзорный курс лекций / *К.Е. Климентьев*. – Самара: Самар. гос. аэрокосм. ун-т., 2008. – 52 с.

ISBN 978-5-7883-0717-6

В курсе лекций рассматриваются специализированное оборудование, системное и прикладное программное обеспечение автоматизированных систем сбора данных и управления, работающих в режиме реального времени. Также обсуждаются основы проектирования и реализации таких систем.

Пособие предназначено для студентов заочной формы обучения специальности 230102 – Автоматизированные системы обработки информации и управления. Может быть полезно студентам очной и очно-заочных форм обучения, а также студентам, обучающимся по смежным специальностям. Рассчитано на 8-12 часов лекционных занятий.

Разработано на кафедре информационных систем и технологий СГАУ.

УДК СГАУ/ 004 (075)
ББК 32.97

ISBN 978-5-7883-0717-6

СОДЕРЖАНИЕ

ТЕМА 1. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

1.1. Системы реального времени как АСУ

Рассмотрим две наиболее часто встречающиеся схемы отношений между человеком, компьютером и окружающим их внешним миром (см. рис. 1.1).

а) человек - посредник между ЭВМ и внешним миром

б) ЭВМ - посредник между человеком и внешним миром

Рис. 1.1. Варианты применения ЭВМ

Случай (а) встречается чаще, он характерен для систем обработки информации. Согласно этой схеме, человек непосредственно взаимодействует с объектами внешнего мира (воспринимает от них информацию и оказывает на них воздействие), а ЭВМ играет вспомогательную роль: выполняет вычисления и преобразования информации, которые нужны человеку. Большинство систем, построенных по схеме (а), представляют собой АИС (*Автоматизированные информационные системы*), АРМ (*автоматизированные рабочие места*), САПР (*системы автоматизированного проектирования*) и т.п.

Схема (б) соответствует тому случаю, когда человек (в силу собственного несовершенства) не способен самостоятельно взаимодействовать с объектами внешнего мира. ЭВМ и системы на их основе играют роль посредника между человеком и объектами внешнего мира. Большинство систем, построенных по схеме (б), представляют собой АСУ – *Автоматизированные системы управления.*

Нужно различать понятия «автоматический» и «автоматизированный». *Автоматические системы* (или просто *автоматы*) – это устройства, выполняющее жестко заданный набор действий без участия человека. *Автоматизированные системы* обычно работают при прямом или косвенном участии человека-оператора в качестве звена общей цепи управления объектом.

Также нужно различать понятия «управление» и «регулирование». Под *управлением* обычно понимают сложную функцию автоматизированной системы, направленную на сохранение определенной структуры управляемого объекта внешнего мира, поддержание определенного режима его деятельности, реализацию определенной программы поведения объекта, обеспечение достижения им определенных целей. *Регулирование* – частный случай управ-

ления, представляющий собой принудительную стабилизацию небольшого количества (иногда – всего одного) физических параметров, от которых зависит поведение объекта.

Системами реального времени (СРВ) называются автоматизированные системы с жесткими ограничениями на временны*е* (динамические) характеристики работы. Формально это условие может быть записано в виде:

$$T_{min} < T < T_{max}, \quad (1.1)$$

где T – некоторая временн*ая* характеристика (например, время реакции на внешнее событие, длительность выполнения какой-либо операции, момент наступления какого-либо события и т.п.);

T_{min} и $T_{max,}$ - предельно допустимые значения (границы) этой характеристики.

Выход за эти границы считается отказом в работе СРВ.

Основное предназначение СРВ - взаимодействие с объектами реального мира в темпе процессов, протекающих в этих объектах. Большинство АСУ является СРВ, а среди АИС такие системы встречаются редко.

Иногда используются также термины: *системы жесткого реального времени* (СЖРВ) и *системы мягкого реального времени* (СМРВ). В СЖРВ нарушение (1.1) недопустимо, в СМРВ это изредка может происходить.

1.2. Классификация АСУ РВ

Рассмотрим следующие классы АСУ, являющихся СРВ:

- *АСУ ТП* - АСУ технологическими процессами (например, система управления ядерным реактором АЭС или система управления конвейером автозавода);
- *АСНИ* - автоматизированные системы научных исследований и комплексных испытаний (например, система вибрационных испытаний компонентов ракетной техники);
- *встроенные системы управления* (предназначенные для управления работой простых технических объектов - мобильных телефонов, стиральных машин, станков и т.п.) и *бортовые системы управления* (предназначенные для управления автомобилями, танками, самолетами, ракетами и т.п.).

Также приведем примеры АСУ, не относящихся к классу СРВ: *АСУП* - АСУ промышленным производством в целом (например, система управления материальными и финансовыми потоками предприятия) и ОАСУ (например, АСУ отрасли - Единой энергетической системы).

1.3. Обобщенная структура АСУ РВ

АСУ представляют собой сложные программно-аппаратные комплексы, функционирующие на основе специализированных математических методов. Основными задачами, решаемыми типичной АСУ, являются:

- сбор данных о состоянии и поведении управляемого объекта;
- преобразование, сохранение, передача и отображение этих данных;
- анализ данных и принятие решений;
- выдача управляющих воздействий на объект.

Обобщенная структура автоматизированной системы управления изображена на рис. 1.2.

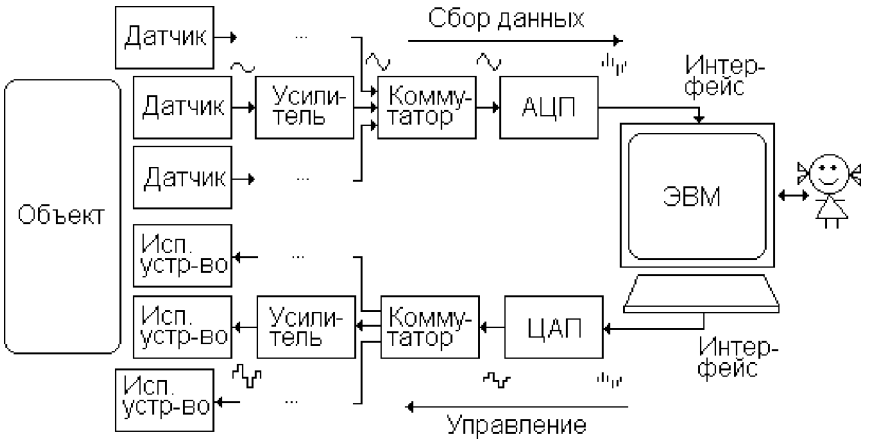


Рис. 1.2. Обобщенная структурная схема АСУ

На практике также встречаются частные случаи этой структуры:

- если в системе присутствуют только каналы сбора данных и измерения, то это *ИИС – информационно-измерительная система*;
- если принятие решений и управление осуществляется непосредственно человеком, то это *АСДУ – автоматизированная система диспетчерского управления*.

Вопросы и задания к Теме 1

1. Являются ли системами реального времени АРМ (автоматизированное рабочее место) бухгалтера? САПР – система автоматизированного проектирования? Интерактивная компьютерная игра? Сетевой сервер?

2. Как вы думаете, является ли системой реального времени АСУ конвейера по сколачиванию табуреток? А если для сколачивания одной табуретки установлены временные ограничения порядка 3 часов? Порядка 3 секунд?

3. Придумайте и приведите примеры СЖРВ и СМРВ.

4. Придумайте и приведите собственные примеры АСУ ТП, АСНИ, встроенных и бортовых систем управления, отличные от приведенных в п. 1.2.

5. Объясните разницу между терминами «автоматическое регулирование» и «автоматизированное управление». Придумайте и приведите примеры.

6. Как вы думаете, почему АСУП не являются системами реального времени?

7. Придумайте и приведите пример АИС, которая является СРВ.

ТЕМА 2. АППАРАТНЫЕ СРЕДСТВА СРВ И ИНТЕРФЕЙСЫ

Аппаратные средства СРВ условно можно разделить на две большие группы:

- средства вычислительной техники (ЭВМ с ее стандартными устройствами и интерфейсами);
- специализированные устройства для связи ЭВМ с объектом.

2.1. Устройства связи с объектом

УСО (устройства связи с объектом) – это комплекс устройств, обеспечивающих взаимодействие объектов внешнего мира и ЭВМ. Рассмотрим примеры таких устройств.

1. *Датчики* (или *первичные измерительные преобразователи*) - это устройства, выполняющие преобразование значения физической величины (температуры, давления, перемещения и т.п.) в электрический сигнал. При этом информация может быть заключена в величине напряжения, тока или частоты изменения сигнала.

Пример: *резистивный датчик температуры* (синонимы: *терморезистор*, *термометр сопротивления*, *резистивный преобразователь температуры*). Принцип его действия основан на зависимости электрического сопротивления металлов (например, Cu - меди или Pt - платины) от температуры. Конструктивно датчик может выглядеть как металлическая проволока, намотанная на диэлектрический сердечник и помещенная в теплопроводящий корпус (см. рис. 2.1).

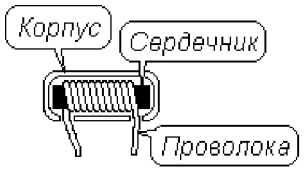


Рис. 2.1. Вариант устройства терморезистора

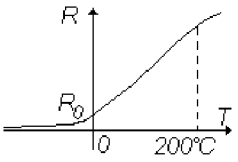


Рис. 2.2. Функция преобразования терморезистора

Важнейшей технической характеристикой любого аналогового датчика является зависимость величины выходного сигнала от значения физической величины на его входе – *функция преобразования* (синонимы: *характеристика преобразования*, или *градуировочная характеристика*, или просто *градуировка*). Для терморезисторов она нелинейна (см. рис. 2.2.), но на некоторых интервалах значений входной величины (например, для меди – на интервале 0..200°C) может быть приблизительно описана линейной функцией

(например, функцией вида $R = R_0(1+\alpha T)$, где α - термоэлектрический коэффициент меди, R_0 – сопротивление при 0°С).

Кроме функции преобразования, основными характеристиками датчиков являются:

- рабочий *диапазон* измеряемых значений;
- *погрешность* – характеристика, отражающая свойство датчика выполнять преобразование с определенной ошибкой, например, с величиной не более 0,1% от ширины диапазона входной величины.

(Примечание: обычно указывают характеристику *основной погрешности*, т.е. погрешности датчика в нормальных условиях работы; и функцию *дополнительной погрешности*, возникающей при отклонении условий работы от нормальных).

Другими характеристиками датчиков, часто указываемыми в документации на устройство, являются *чувствительность* (первая производная от функции преобразования), *постоянная времени* (временной интервал, в течение которого завершаются переходные процессы в датчике), АЧХ – *амплитудно-частотная характеристика* (функция, характеризующая влияние частоты изменения физической величины на ошибку преобразования) и т.п.

Существуют также цифровые датчики, преобразующие некую характеристику состояния объекта (например, состояние клапана – «закрыт» или «открыт») в значения цифрового сигнала 0 («ложь») или 1(«истина»).

Используемый иногда термин «интеллектуальный датчик» соответствует компактному, но сложному устройству, содержащему, кроме собственно датчика схемы, преобразования сигнала управляющий микроконтроллер и, возможно, цифровой интерфейс (см. ниже).

2. *Промежуточные измерительные преобразователи* – устройства, сохраняющие вид представления сигнала (например, напряжение остается напряжением) и его форму, но изменяющие его величину. Они необходимы в тех случаях, когда электрический сигнал, поступающий с датчика, слишком слаб по величине, либо слишком силен, либо засорен помехами и т.п. К устройствам этого типа относятся разнообразные усилители, нормализаторы, фильтры и пр. Нередко они конструктивно входят в состав датчика (например, мостовая схема подключения термосопротивлений). Как правило, промежуточные преобразователи тоже имеют функцию преобразования простого вида $Y=A\cdot X$, которую можно описать единственным числом - коэффициентом усиления A.

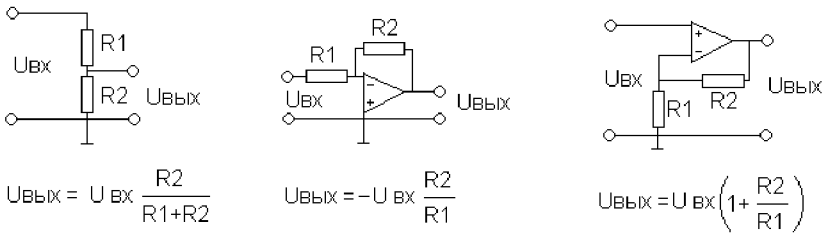


Рис 2.3. Простейшие схемы для изменения уровня напряжения

Существуют как простые усилители, представляющие собой электронные схемы на основе резисторов и операционных усилителей (см. рис. 2.3), так и сложные многофункциональные устройства с программируемым извне или автоматически настраиваемым коэффициентом усиления.

3. *Аналогово-цифровые преобразователи* (АЦП) – устройства, предназначенные для преобразования значения электрического сигнала в число.

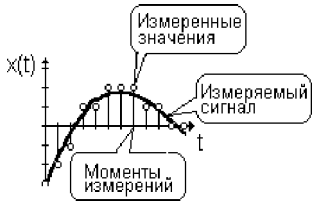


Рис. 2.4. Квантование сигнала по уровню и дискретизация по времени

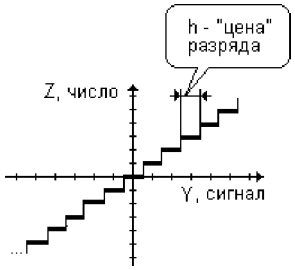


Рис. 2.5. Функция преобразования АЦП

Такое преобразование происходит не мгновенно (этот временной интервал называется *временем преобразования АЦП*), поэтому динамически изменяющийся непрерывный электрический сигнал, поступающий на вход АЦП, на выходе преобразуется в дискретную последовательность чисел. Типичное время преобразования современных АЦП: от 10-7 до 10-3 с.

Каждое число на выходе у АЦП представляется конечным количеством двоичных разрядов (эта характеристика называется *разрядностью АЦП*), следовательно, множество всевозможных числовых значений на выходе у АЦП ограничено: например, если разрядность АЦП N=2, то всего возможно 4 различных результата преобразования: $00_2=0_{10}$, $01_2=1_{10}$, $10_2=2_{10}$ и $11_2=3_{10}$. Типичные значения разрядности современных АЦП: от 8 до 24.

Таким образом, работа АЦП заключается в выполнении двух операций (см. рис. 2.4):

- квантование сигнала по уровню;
- дискретизация сигнала по времени.

Функция преобразования АЦП имеет вид «ступенчатой кривой» (см. рис. 2.5) и может быть описана формулой

$$Z = \left[\frac{Y + 0.5}{h} \right], \qquad (2.1)$$

где Y – преобразуемый электрический сигнал; Z – числовой результат преобразования; *h* – величина входного сигнала, соответствующая одному биту (разряду) выходного числа; [.] – округление числа до целого значения. Таким образом, в любом АЦП происходит преобразование близких значений электрического сигнала в одно общее числовое значение, что приводит к *погрешности квантования*, не превышающей по величине h/2 (см. рис. 2.5).

Итак, наиболее важные технические характеристики АЦП:

- *входной диапазон* (диапазон измеряемых значений тока или напряжения, например ±5В);
- *разрядность* (количество двоичных разрядов, которыми представляется выходное значение, например, 16);
- *время преобразования* (например, 0,00001 сек).

4. *Цифро-аналоговые преобразователи* (ЦАП) – устройства, предназначенные для преобразования числовой величины в электрический сигнал (напряжение или ток). Свойства и технические характеристики ЦАП аналогичны АЦП.

5. *Коммутаторы* аналоговых сигналов – устройства, осуществляющие физическое переключение (коммутацию) связей между различными устройствами. Различают *мультиплексоры* (устройства, способные подключать несколько входов на один выход), *демультиплексоры* (устройства, способные подключать один вход к нескольким выходам) и комбинированные коммутаторы. Существуют также коммутаторы цифровых сигналов – «*свитчи*».

6. *Исполнительные устройства* – устройства, предназначенные для организации непосредственного воздействия на объект. Примером ИУ могут служить шаговый двигатель, электрический нагреватель и т.п.

2.2. Интерфейсы

Для передачи электрических сигналов между различными устройствами (например, между датчиком и усилителем или между усилителем и АЦП) используются аналоговые *линии связи*, т.е. обычные электрические провода и кабели. В случае же, когда необходимо передавать цифровую информацию, используются специальные *интерфейсы* – комплексы программных и аппаратных средств и протоколов (алгоритмов взаимодействия), предназначен-

ных для обеспечения конструктивной, электрической и логической совместимости различных устройств и их компонентов.

Обычно интерфейс (см. рис. 2.6) состоит из:

- *шины* (или *магистрали*) – набора электрических линий (проводов), по которым передаются информационные и служебные сигналы;
- *адаптеров* приемника и передатчика – специальных устройств, которые реализуют протоколы обмена информационными и служебными сигналами.

Рис. 2.6. Интерфейс

По способу передачи данных по шине интерфейсы можно разделить на *параллельные* и *последовательные* (см. рис. 2.7). Параллельные интерфейсы позволяют передавать элементы информации (например, отдельные биты) независимо друг от друга по разным линиям шины; теоретически, они обеспечивают высокую скорость передачи данных, но сложней в реализации и использовании. Последовательные интерфейсы передают элементы информации один за другим по небольшому числу линий (например, всего по одной линии), – этот вид передачи данных прост и экономичен, но медленен.

Однако практической реализации быстрых параллельных интерфейсов мешает эффект взаимного электрического влияния сигналов, распространяющихся по соседним линиям шины.

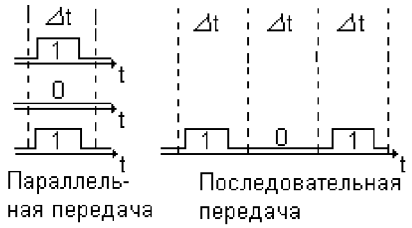


Рис. 2.7. Параллельная и последовательная передача данных

По назначению интерфейсы можно разделить на *внутренние* (или *системные*) - предназначенные для объединения составных частей одного устройства, и *внешние* – предназначенные для организации информационного обмена между независимыми устройствами.

Примеры интерфейсов.

1. *Последовательный интерфейс RS-232* можно отнести к классу внешних интерфейсов. Он позволяет передавать данные между двумя устройства-

ми со скоростями до 115 килобит/с на расстояние до 15 метров (при уменьшении скорости возможно увеличение дальности передачи до 1 км). Международный стандарт на этот интерфейс описывает передачу данных по 25 линиям, но в простейшем случае достаточно кабеля с двумя линиями: одна используется для передачи данных, другая - как общая «земля».

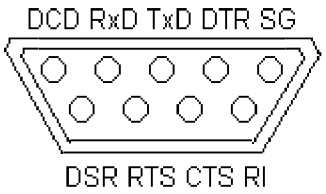


Рис. 2.8. Разъем RS-232

Также стандарт описывает протокол передачи, предусматривающий организацию данных в виде набора битов, который включает начальный («стартовый») бит, конечный («стоповый») бит, бит контроля целостности посылки, блок информационных битов и т.п.

Интерфейс RS-232 используется для подключения к различным типам ЭВМ манипуляторов «мышь», модемов, контрольно-кассовых аппаратов, медицинских анализаторов, АТС, цифровых осциллографов, программаторов и т.п. В ПЭВМ имеется устройство, организующее обмен данными по стандарту RS-232, это COM-порт.

2. *Последовательные интерфейсы RS-422* и *RS-485* являются дальнейшим развитием RS-232. В отличие от RS-232 они позволяют соединять вместе не два, а бо́льшее количество устройств (до 32); передавать данные с более высокой скоростью на бо́льшие расстояния (от 62,5 кбит/с на 1200 м до 10 мбит/с на 10 м). Но международные стандарты на эти интерфейсы не описывают никаких протоколов передачи данных, поэтому RS-422 и RS-485 служат как физическая база для организации других интерфейсов (например, класса «полевая шина» - *ModBus*, *ProfiBus* и т.п.).

3. *Параллельный интерфейс КАМАК* относится к классу системных интерфейсов (см. рис. 2.9). Он является исторически первым и наиболее типичным представителем семейства интерфейсов, служащих базисом для построения *магистрально-модульных систем* (ММС). В основе таких систем лежит магистраль, заключенная в открытый с одной стороны металлический корпус – *крейт*. Различные функциональные компоненты системы (так называемые *функциональные модули*), вставляемые в разъемы магистрали, располагаются внутри крейта, словно книги в полке книжного шкафа – этим обеспечивается высокая надежность системы и устойчивость ко внешним воздействиям. В КАМАК предусмотрено всего 25 разъемов, они называются «*станциями*». Номенклатура модулей огромна: существуют модули усилителей и нормализаторов, модули АЦП, модули ЦАП, модули мультиплексоров и демультиплексоров, модули внешних запоминающих устройств (винчестеров, дисководов, флэш-накопителей и пр.), модули таймеров и счетчиков, модули управления роботами, модули декодирования сигналов с видеокамер и т.п. В крейте КАМАК обязательно должен присутствовать один «главный»

модуль (так называемый «*крейт-контроллер*»), который управляет передачей данных по шине между отдельными модулями. Он может быть оформлен как интерфейс к внешней ПЭВМ либо содержать внутри себя микроЭВМ.

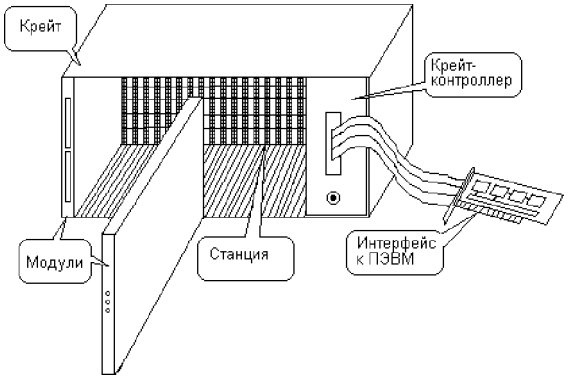


Рис. 2.9. Крейт и модули в стандарте КАМАК

Стандарт КАМАК был разработан в конце 1960-х годов, к середине 1980-х годов он морально устарел и был вытеснен более современными интерфейсами. Однако простота и высокая надежность архитектуры КАМАК позволяют эксплуатировать системы на ее основе до сих пор (прежде всего, в институтах ядерных исследований и на атомных электростанциях).

4. *Параллельные интерфейсы VME/VXI, CompactPCI, PXI* и пр. относятся к классу системных интерфейсов и также предназначены для построения ММС. Они являются развитием архитектурной идеологии, использовавшейся ранее в интерфейсе КАМАК: предусматривают наличие крейта и функциональных модулей. (Примечание: в дешевых магистрально-модульных системах, например, в NI FieldPoint, крейт может отсутствовать). VME/VXI и CompactPCI могут использоваться в качестве «полноценного» системного интерфейса, позволяющего на основе общей шины объединять процессорный модуль, модуль оперативной памяти, модули внешних устройств и прочие компоненты ЭВМ. В системах, основанных на этих интерфейсах, получили распространение *мезонинные технологии*, предусматривающие организацию функциональных модулей также на базе некоторых (более простых и миниатюрных) магистрально-модульных архитектур.

5. Прочие интерфейсы. Среди прочих интерфейсов, используемых в АСУ и СРВ, можно отметить:

последовательный внешний интерфейс *CANbus*, часто используемый для организации распределенных бортовых (автомобильных и авиационных) систем управления;

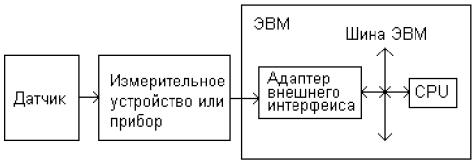
последовательный внешний интерфейс *Industrial Ethernet*, являющийся оптимизированной под задачи реального времени модификацией сетевого интерфейса Ethernet;

комбинированный интерфейс *HART*, позволяющий одновременно передавать цифровые данные и аналоговые сигналы.

Последовательный внешний интерфейс USB, параллельные системные интерфейсы PCI, ISA, AGP, параллельный сетевой интерфейс Ethernet редко используются в системах реального времени.

В зависимости от используемых интерфейсов возможны различные варианты объединения компонентов УСО и подключения к ЭВМ.

1. Автономное исполнение. Компоненты УСО представляют собой комбинированные измерительные и управляющие устройства, сочетающие в себе усилители, АЦП, коммутаторы, микроконтроллеры и пр. Информационная связь с ЭВМ осуществляется посредством одного из стандартных интерфейсов связи (RS-232, RS-485, USB, приборного интерфейса GPIB и т.п.). Также возможно использование этих интерфейсов для сетевого объединения автономных компонентов друг с другом.

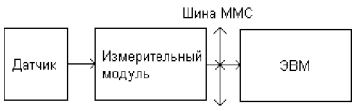


а) Функциональная схема

б) NI-6008 с интерфейсом USB

Рис. 2.10. УСО в виде автономного устройства

2. В составе магистрально-модульных систем. Компоненты УСО, ЭВМ и ее внешние устройства представляют собой функциональные модули («кубики») магистрально-модульных систем на базе КАМАК, VME/VXI, PXI, CompactPCI и пр. Обмен данными производится по системной шине магистрально-модульной системы.

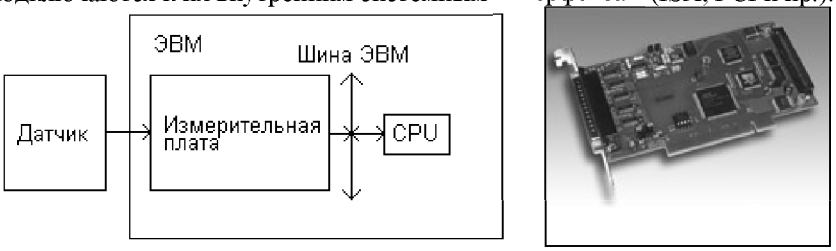


а) Функциональная схема

б) Система на базе NI FieldPoint

Рис. 2.11. УСО на базе магистрально-модульных интерфейсов

3. Подключение к внутренним интерфейсам универсальных ЭВМ. Компоненты УСО изготавливаются в виде плат расширения универсальных ЭВМ и подключаются к их внутренним системным интерфейсам (ISA, PCI и пр.).



а) функциональная схема

б) Плата L-Card Д-780

Рис. 2.12. УСО в виде компонентов ЭВМ

2.3. Средства вычислительной техники

Основные требования, предъявляемые к средствам вычислительной техники, используемым для организации СРВ:

- надежность и работа в расширенном диапазоне внешних условий;
- наличие развитой системной службы времени;
- легкая конфигурируемость;
- простота системы команд и способов адресации;
- возможность подключения большого количества внешних устройств;
- наличие развитой системы обработки прерываний.

ЭВМ, используемые в АСУ и СРВ, построены на базе различных процессоров: Intel 80x86 и Pentium, Motorola MC68x00, PowerPC и пр. Рассмотрим основные классы средств вычислительной техники, применяемые в АСУ и СРВ.

1. «*Промышленные*» *ПЭВМ.* Частично решить проблему надежности и устойчивости к неблагоприятным внешним условиям помогает специализированное «инженерное» исполнение, предусматривающее металлический корпус, ударо- и вибростойкое шасси, мощную систему охлаждения, нестандартную компоновку материнских плат и т.п. Типичная промышленная персональная ЭВМ снабжается большим количеством различных сетевых и системных интерфейсов для подключения внешних устройств.

Назначение – сбор данных и управление многими объектами с большим числом точек доступа в масштабах цеха или предприятия; сбор, обработка, визуализация и хранение потоков данных, поступающих из локальных узлов, в качестве которых выступают промышленные контроллеры и микроконтроллеры.

Пример: промышленная ЭВМ ROBO-2000. Процессор: Intel Pentium IV, тактовая частота 3 ГГц. ОЗУ – 512 Мб. Наличие встроенного сторожевого таймера. Количество слотов PCI – 8, PCI Express x1 – 3, PCI Express x16 – 1. Адаптеров Ethernet – 1. Винчестер – 120 Гб. Корпус стальной, шириной 19" и высотой 4U. Масса – 19 кг. Стоимость – порядка 1600$.

Рис. 2.13. Модель ROBO-2000

2. *Промышленные контроллеры и программируемые логические контроллеры* (ПЛК) – полноценные микроЭВМ, которые невелики по размерам, недороги, обладают сравнительно небольшой вычислительной мощностью (тактовая частота процессора – несколько десятков МГц, объем оперативной памяти – до нескольких МГб), но развитыми средствами коммуникаций. Обычно к ним не подключаются ни клавиатура, ни монитор, а весь обмен с оператором идет через имеющиеся интерфейсы (например, через RS-232). Часто промышленные контроллеры выполняются в виде функциональных модулей магистрально-модульных систем. Основное назначение – решение несложных задач сбора и обработки данных и локального управления в масштабе станка, промышленной установки, автономного агрегата и т.п. Типичное применение – работа в качестве одного из узлов сети и/или управляющего модуля магистрально-модульной системы. Примеры:

- ADVANTECH I-7188 (процессор AMD 80386, тактовая частота 40 МГц, ОЗУ 256 Кб, электронный Flash-диск 512 Кб, разъемы RS-232 и RS-485 – 4 шт);
- PEP/Kontron SMART I/O (процессор – MC 68030, тактовая частота 20 МГц, ОЗУ – 512 Кб, встроенная флэш-память объемом 256 Кб с операционной системой OS-9, разъемы RS-232 и RS-485 – 2 шт., разъемов для сменных модулей - 3).

а) PEP/Kontron Smart IO

б) Mitsubishi FX3U

в) Advantech I-7188

Рис. 2.14. Примеры промышленных и программируемых логических контроллеров

3. *Цифровые сигнальные процессоры* (DSP) и *цифровые микроконтроллеры* (МК) – компактные и дешевые устройства, предназначенные для решения

несложных типовых задач автоматизации управления во встроенных и бортовых системах. DSP и МК часто выполняются в виде одной микросхемы (или в виде «*чипсета*» – комплекта микросхем), интегрирующей в себе ряд устройств: АЛУ, ОЗУ, ПЗУ для хранения программ, АЦП, ЦАП, простой интерфейс передачи данных и т.п. Основное отличие МК от DSP заключается в том, что DSP проблемно-ориентированы, в их системе команд присутствуют специализированные машинные команды, реализующие алгоритмы цифровой обработки сигналов; МК более универсальны. Примеры: микросхемы фирм ESS и YAMAHA в звуковых картах ПЭВМ; чипсеты Rockwell в модемах; микроконтроллер AT43USB351M с ПЗУ 24 Кб, ОЗУ 1 Кб, 10-разрядным АЦП, 8 и 16-битовыми таймерами-счетчиками и программируемым USB-интерфейсом.

Альтернативой микроконтроллерам могут служить ПЛИС (программируемые логические интегральные схемы).

Вопросы и задания к Теме 2

1. Измерительный канал состоит из двух последовательно соединенных компонентов: 1) с функцией преобразования $Y=A\cdot X+B$; 2) с функцией преобразования $Y=C\cdot X+D$? Какова полная функция преобразования всего канала? Какую формулу надо использовать, чтобы по полученному на выходе измерительного канала значению рассчитать значение, поданное на его вход? Как поступить в случае нелинейных функций преобразования?

2. Сигнал на выходе датчика изменяется в пределах ±1В, а АЦП рассчитан на диапазон ±7,5В. Предложите схему усилителя и номиналы резисторов.

3. Пределы измерения АЦП составляют ±5В, разрядность – 8 бит. Какова погрешность дискретизации в вольтах? В процентах?

4. Время преобразования АЦП составляет 22,7 мкс. Какова максимальная частота дискретизации сигнала, которую можно достичь с этим АЦП?

5. Данные по интерфейсу RS-232 передаются пакетами, состоящими из одного стартового, 8 информационных и одного стопового бита со скоростью 115 кбит/с. Какова продолжительность передачи одного бита? Одного пакета?

ТЕМА 3. ПРОГРАММНЫЕ СРЕДСТВА СРВ

3.1. Операционные системы

Основные требования, предъявляемые к операционным системам (ОС), используемым в АСУ и СРВ:

- предсказуемость поведения во временной области;
- масштабируемость (т.е. возможность получать сверхкомпактные и сверхбыстрые варианты ОС за счет отключения ряда компонентов и функций).

ОС, удовлетворяющие требованию предсказуемости поведения во временной области, называются *операционными системами реального времени* (ОС РВ). ОС, удовлетворяющие требованию масштабируемости, называются *встраиваемыми операционными системами*.

Современные ОС, предназначенные для использования в АСУ и СРВ, обычно удовлетворяют обоим требованиям.

3.1.1. Архитектурные особенности

В общем случае операционная система состоит из нескольких «слоев», каждый из которых выполняет свой набор функций (см. рис. 3.1).

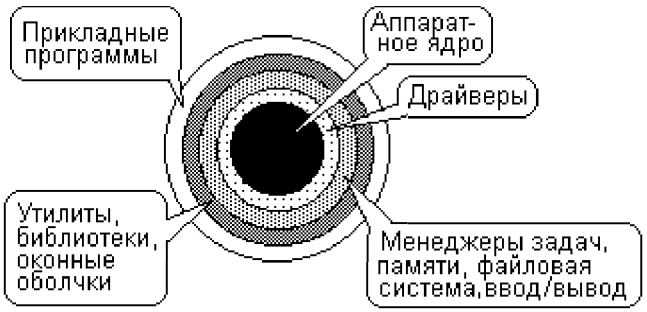


Рис. 3.1. Многослойная структура программного обеспечения

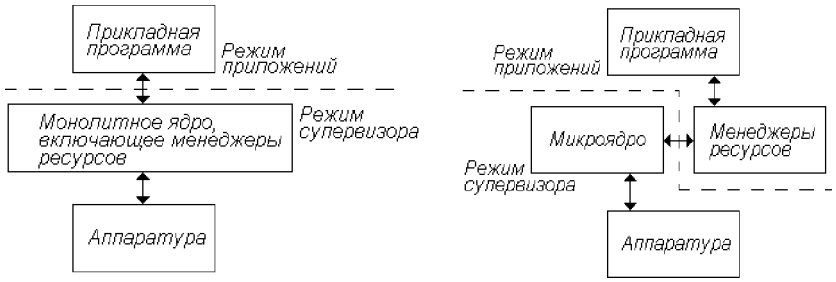
Практически все современные процессоры (Intel 80x86/Pentium, Motorola MC 68K, PowerPC и т.п.) поддерживают, как минимум, два режима выполнения программного кода:

- «привилегированный режим» (или «режим супервизора», или «режим ядра», или «kernel mode», или «защищенный режим»);
- «непривилегированный режим» (или «режим приложений»).

Основное отличие между ними заключается в том, что программы, работающие в режиме супервизора, имеют непосредственный доступ ко всем ресурсам ЭВМ - ко внешним устройствам, к оперативной памяти по физиче-

ским адресам и пр.; программы же режима приложений работают в виртуальной среде, сформированной программами режима супервизора.

Различают два больших класса архитектур операционных систем: «*монолитная*» и «*микроядерная*» (см. рис. 3.2).



а) Монолитная архитектура б) Микроядерная архитектура

Рис. 3.2. Варианты архитектур операционных систем

Все компоненты «монолитной» ОС работают в режиме супервизора в едином адресном пространстве. Главное достоинство таких ОС – высокая производительность. Главный недостаток – невозможность внесения каких-либо изменений в структуру операционной системы в процессе ее эксплуатации, т.е. плохая масштабируемость. Другой недостаток – невысокая реактивность системы, т.к. если внешнее событие, требующее немедленной реакции, происходит во время выполнения задач уровня ядра, то обработка этого события задерживается до возвращения на уровень приложений.

Особенность «микроядерных» ОС – наличие компактного и быстродействующего «микроядра», работающего в режиме супервизора, а все остальные компоненты операционной системы, включая менеджеры ресурсов, при этом работают в непривилегированном режиме. «Микроядерный» подход обеспечивает хорошую гибкость и масштабируемость операционной системы, малое время реакции на внешние события. С другой стороны, «микроядерные» операционные системы отличаются относительно невысокой производительностью, т.к. при работе происходят частые переключения из режима в режим.

Встраиваемые ОСРВ преимущественно строятся в соответствии с «микроядерной» архитектурой. В фирменной документации на ОСРВ обычно указывают числовые значения временных характеристик, таких как:

- предельное время переключения с задачи на задачу;
- предельная задержка между возникновением прерывания и началом его обработки;

- предельное время выполнения запроса прикладной программы к ядру ОС;
- предельное время переключения из режима «супервизора» в «непривилегированный» режим и обратно, и т.п.

3.1.2. Организация многозадачности

Поскольку программное обеспечение СРВ должно взаимодействовать со многочисленными процессами внешнего мира и реагировать на многочисленные события (в том числе и происходящие одновременно), то встраиваемые ОСРВ должны поддерживать работу в *режиме многозадачности*. В этом режиме несколько программ (называемых *задачами* или *вычислительными процессами*) могут выполняться одновременно.

На самом деле, если процессор один, то задачи выполняются не одновременно, а поочередно – короткими временны*ы*ми отрезками. Пока выполняется одна задача, другие - ожидают. Для организации ожидания обычно используются две очереди (см. рис. 3.3):

- очередь готовых для исполнения задач;
- очередь задач, заблокированных отсутствием необходимых ресурсов.

В очереди второго типа могут, например, располагаться задачи, требующие обращения к диску, занятому другой - выполняющейся в настоящий момент задачей.

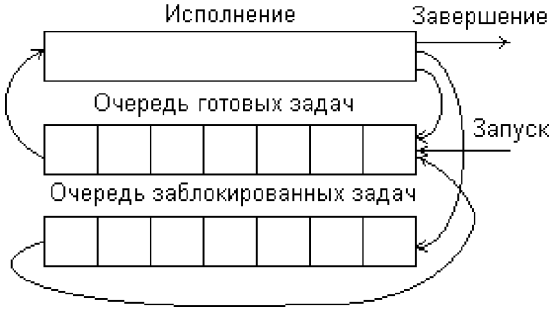


Рис. 3.3. Граф многозадачности

Существуют два основных способа организации многозадачности:

- кооперативный;
- вытесняющий.

Кооперативная многозадачность – режим работы операционной системы, при котором различные задачи сами принимают решения о передаче управления друг другу в те моменты времени, когда им это «удобно». Точнее, текущая задача отдает управление операционной системе, а уж та извлекает другую задачу из очереди и отдает ей управление. (Примечание: переда-

ча управления другой задаче может также происходить по инициативе пользователя или внешнего устройства, возбудившего аппаратное прерывание.) С одной стороны, при правильном проектировании программного обеспечения с учетом всех влияющих факторов такой подход позволяет добиваться высокой реактивности СРВ на внешние события, предсказуемости ее поведения и эффективного использования ресурсов ЭВМ. С другой стороны, это усложняет процесс разработки, налагает повышенные требования на квалификацию разработчика, служит потенциальным источником ошибок, уменьшает устойчивость системы. В частности, «зависание» одной задачи приводит к невозможности передачи управления следующей задаче и, как следствие, к отказу всей системы.

Вытесняющая многозадачность – режим работы ОС, при котором переключением задач занимается только она сама. Каждой задаче выделяется короткий временной отрезок (типичное значение 0,02 сек), так называемый *квант времени* (*timeslice*), по истечении которого задача принудительно прерывается («вытесняется»), и управление передается задаче, находящейся в голове очереди готовых задач. По исчерпании очереди управление вновь передается первой задаче и т.д. Такой режим работы в общем случае не приводит к высокой реактивности системы, но обеспечивает гарантированный отклик на внешнее событие в пределах временной задержки, которую можно рассчитать заранее. Программное обеспечение, работающее в условиях вытесняющей многозадачности, устойчиво к ошибкам – «зависание» одной из задач приведет всего лишь к потере квантов времени, выделенных этой задаче, но работоспособность системы в целом сохранится.

ОСРВ, как правило, позволяют использовать различные режимы многозадачности: кооперативную - для задач жесткого реального времени и вытесняющую - для пользовательской работы и задач мягкого реального времени.

Для управления последовательностью выполнения задач (т.е. порядком размещения в очереди) им присваиваются числовые *приоритеты*. Чем выше приоритет, тем ближе к голове очереди располагается задача. Основное правило: *если в очереди появляется задача с приоритетом выше, чем приоритет исполняющейся в настоящий момент задачи, то текущая задача немедленно «вытесняется», а выполняться начинает новая, более приоритетная задача*. Алгоритм, в соответствии с которым операционная система назначает приоритеты и передает управление от одной задачи к другой, называется «*дисциплиной диспетчеризации*».

Наиболее простой дисциплиной диспетчеризации является «*карусельная дисциплина*» (или «*круговорот*», или «*Round Robin*»), в соответствии с которой задачи имеют равные приоритеты (т.е. порядок их расположения в очереди постоянен) и получают равные кванты времени (см. рис. 3.4).

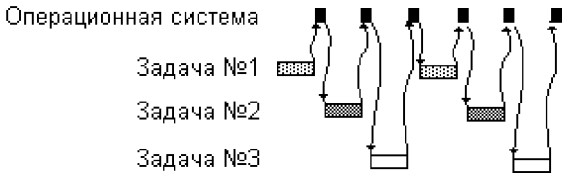


Рис. 3.4. Дисциплина «Round Robin»

«Круговорот» редко используется в СРВ, т.к. может приводить к недопустимо длинным задержкам выполнения критических задач.

Существуют специальные дисциплины диспетчеризации, оптимизированные для задач РВ, например, «Earliest Deadline First», предусматривающая динамическое присвоение более высоких приоритетов тем задачам, которым до завершения осталось меньше времени.

Реальное использование различных режимов многозадачности в операционных системах сталкивается с рядом проблем. Пример: проблема «*инверсии приоритетов*». Пусть за ресурс Р конкурируют задача А с высоким приоритетом, задача Б со средним приоритетом и задача В с низким приоритетом. В некий момент времени ресурсом Р владеет задача В, а выполняется задача Б. В этой ситуации задача Б будет выполняться неограниченно долго (по крайней мере, до своего завершения), а задачи А и В не получат возможности для работы.

Для решения подобных проблем в операционных системах офисного назначения применяется «*разгон приоритетов*»: операционная система периодически выбирает из «хвоста» очереди давно не выполняющиеся задачи и присваивает им на один квант времени максимальный приоритет. В системах, ориентированных на задачи РВ, применяется «*наследование приоритетов*»: всем задачам, конкурирующим за общий ресурс, кратковременно присваивается один и тот же приоритет, равный максимальному среди всех таких задач.

3.1.3. Синхронизация задач

Порядок выполнения задач может определяться не только операционной системой, но и самими задачами.

Если одновременный доступ нескольких параллельно выполняющихся задач к одному ресурсу невозможен или нежелателен (например, к дисковому файлу, открытому для записи), то такой ресурс называется *критическим*. Фрагменты задач, в которых может осуществляться попытка доступа к критическим ресурсам, являются и называются *критическими секциями*. Для того, чтобы параллельно выполняющиеся задачи могли корректно обращаться к общим ресурсам, им необходимо *синхронизировать* (т.е. согласовать) свое выполнение. Встречается также *асинхронное* параллельное выполнение за-

дач, при котором они непосредственно не взаимодействуют, но ориентируются на ход каких-либо внешних процессов, например, на течение времени; на прерывания от устройств; на состояние объектов внешнего мира.

Модель «поставщик-потребитель» - самая простая и наиболее часто встречающаяся модель задач, конкурирующих за общий ресурс. Задача-«поставщик» способна выполнять операции: 1) «произвести»; 2) «передать». Задача-«потребитель» способна выполнять операции: 1) «принять»; 2) «употребить». Передача данных осуществляется через буфер, который является критическим ресурсом, причем:

- запрещен одновременный доступ к буферу со стороны разных задач;
- запрещена попытка чтения из пустого буфера;
- запрещена попытка записи в заполненный буфер.

Соответственно, критическими секциями являются фрагменты программ, выполняющие операции «передать» и «принять». Модель «поставщик-потребитель» может быть обобщена на произвольное количество задач и буферов.

Простейшее решение проблемы синхронизации, основанное на блокирующем «флаге», может выглядеть примерно так, как изображено на рис. 3.5.

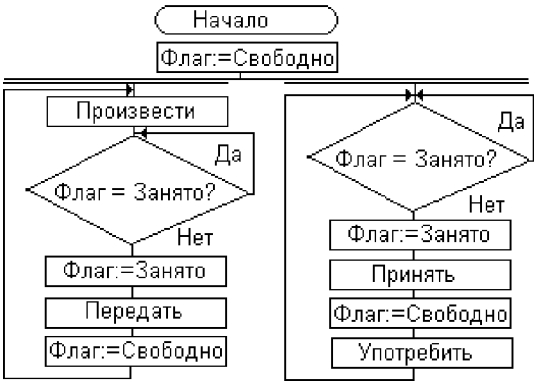


Рис. 3.5. Алгоритм с блокирующим флагом

Доказано, что это решение некорректно для случая, когда параллельно работающие задачи выполняются с разными скоростями. Удовлетворительное решение проблемы синхронизации конкурирующих задач с использованием флагов занятости выглядит следующим образом («*алгоритм Деккера-Холта*», см. рис. 3.6).

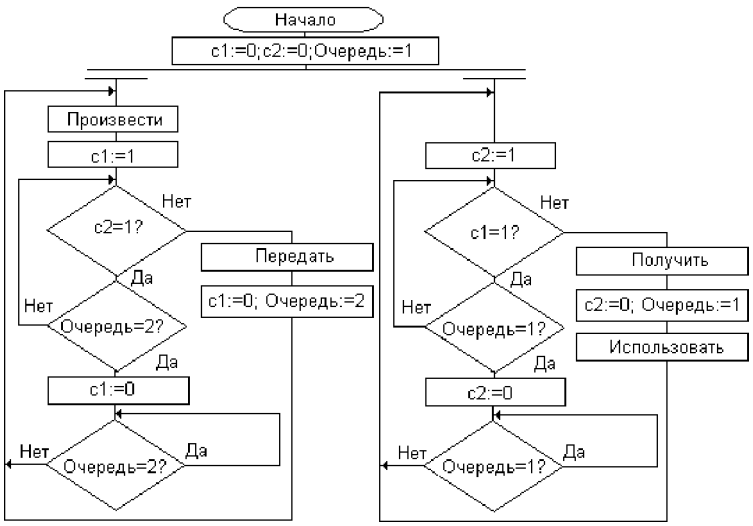


Рис. 3.6. Алгоритм Деккера-Холта

Недостаток этого метода - громоздкость реализации, особенно для случая большого количества конкурирующих задач.

Наилучшим методом решения проблемы является использование *семафора Дейкстры* - целой переменной S, с которой связана очередь ожидающих задач и над которой разрешено производить две неделимые (т.е. не прерываемые во время выполнения) операции.

1. Операция «оградить» («lock»):

- S:=S-1
- если S> 0, то текущая задача продолжает работу, иначе встает в конец очереди ожидания.

2. Операция «освободить» («unlock»):

- если S<=0 , то запускается первая в очереди задача
- S:=S+1.

Семафоры применяются следующим образом.

Для обеспечения невозможности одновременного доступа достаточно всего одного семафора S1: операция «оградить» используется как «открывающая скобка» перед критической секцией, а операция «освободить» используется как «закрывающая скобка» после критической секции. Для невозможности чтения из пустого буфера и записи в заполненный буфер используются еще два семафора – S2 и S3 (см. рис. 3.7).

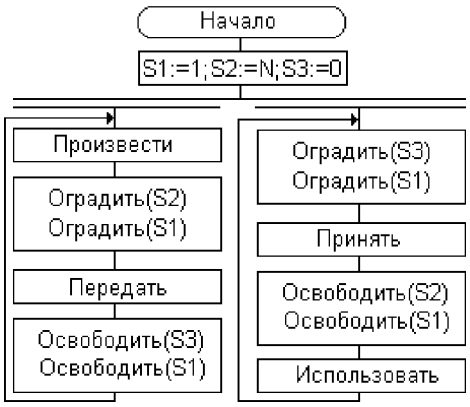


Рис. 3.7. Решение задачи «поставщик-потребитель» с использованием семафоров

Для синхронизации конкурирующих задач в различных операционных системах также применяется множество других механизмов и их модификаций: *мьютексы*, *сигналы*, *сообщения* и пр.

3.1.4. Примеры операционных систем реального времени

Итак, сформулируем расширенные требования к операционным системам с точки зрения пригодности для использования в АСУ и СРВ:

- масштабируемость;
- конфигурируемость, т.е. возможность включения/отключения функциональных особенностей, способных повлиять на выполнение программ в режиме РВ (например, виртуальной памяти);
- наличие разнообразных, мощных и гибких средств управления приоритетной многозадачностью, включая средства для борьбы с «инверсией приоритетов»;
- наличие разнообразных, мощных и гибких средств синхронизации задач;
- предсказуемость поведения (особенно, во временной области);
- соответствие стандартам (прежде всего, стандарту POSIX, определяющему интерфейсы доступа прикладных программ к ядру ОС);
- открытость и документированность.

Примерами операционных систем, удовлетворяющих этим требованиям, являются OS-9/9000, QNX (см. рис. 3.8), WxWorks, OS Lynx. Благодаря хорошей масштабируемости, эти ОС работают и на «больших» ЭВМ, и на промышленных контроллерах.

Рис. 3.8. Внешний вид рабочего стола QNX Neutrino

Операционные системы семейства MS Windows не удовлетворяют почти ни одному из этих требований, но (благодаря своим развитым графическим средствам) достаточно широко используются для построения систем мягкого реального времени на базе промышленных ЭВМ. Существуют решения, позволяющие модифицировать MS Windows таким образом, чтобы часть перечисленных требований были удовлетворены (например, пакет RTX фирмы VenturCom, «исправляющий» менеджер задач Windows). Также имеются разработки самой фирмы Microsoft: Windows CE и Windows XPE, обладающие свойствами масштабируемости и конфигурируемости.

Существует отечественная встраиваемая операционная система реального времени ОС2000, используемая вооруженными силами для построения бортовых систем управления и являющаяся модифицированной и русифицированной разновидностью VxWorks.

3.2. Языковые средства для создания приложений

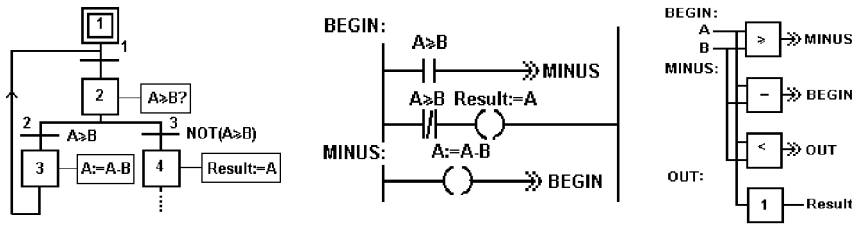
Задачи РВ решаются при помощи разбиения на сравнительно небольшое множество типовых подзадач, таких как: опрос источников данных; обработка данных средствами типовых алгоритмов; накопление собранных данных и результатов обработки; визуализация собранных данных и результатов обработки; передача собранных данных и результатов обработки; диалоговое взаимодействие с оператором и т.п. Все эти задачи целесообразно решать на основе стандартизации и унификации языков программирования, алгоритмов и т.п. Рассмотрим ряд типовых решений.

Международный стандарт IEC 1131-3 описывает синтаксис и семантику пяти специализированных языков программирования ПЛК: SFC, LD, FBD, ST и IL. Применением этих языков обеспечивается легкая переносимость реализационных решений с одной платформы на другую.

1. *SFC* (Sequential Function Chart) – высокоуровневый графический язык, используемый для описания алгоритма в виде набора связанных пар шагов и переходов (см. рис. 3.9, а).

2. *LD* (Ladder Diagram) - графический язык программирования для описания алгоритмов в виде множества релейных схем, в которых логические и вычислительные операции представлены в виде множества «шин», «контактов» и «катушек» (см. рис. 3.9, б).

3. *FBD* (Functional Block Diagram) - в этом графическом языке используются функциональные блоки - элементы типа "логическое И", "логическое ИЛИ", блоки сравнения, сложения и вычитания и прочие элементы, соединенные линиями (см. рис. 3.9, в).



а) Язык SFC б) Язык LD в) Язык FBD

Рис. 3.9. Графические языки программирования ПЛК

4. *ST* (Structured Text) - текстовый высокоуровневый язык общего назначения, по синтаксису сходный с языком Паскаль (см. рис. 3.10, а).

5. *IL* (Instruction List) - текстовый язык низкого уровня, похожий на язык Ассемблера (см. рис. 3.10, б).

```
WHILE A>=B DO
A:=A-B
END_WHILE;
Result:=A;
```

а) Язык ST

```
LD A
BEGIN: GE
       JMPNC OUT
       SUB B
 JMP BEGIN
FINISH:ST Result
```

б) Язык IL

Рис. 3.10. Текстовые языки программирования ПЛК

Написание, отладка и сборка программ производится на ПЭВМ в операционной системе, имеющей развитые графические средства (например, Windows или QNX), а затем результат (подготовленный к работе программный модуль) передается на ПЛК, не имеющий ни клавиатуры, ни монитора, через один из стандартных интерфейсов, например, через RS-232. Подобная технология называется *кросс-платформенной технологией* (или просто «*кросс-технологией*»).

3.2.2. Инструментальные среды

Разработано большое количество специализированных инструментальных сред, предназначенных для упрощения решения задач автоматизации и основанных на идеях визуального и графического программирования. Типовые алгоритмы сбора, обработки, передачи, накопления и визуализации данных складываются из отдельных, заранее заготовленных и отлаженных «кубиков».

Такие пакеты получили название SCADA-систем (Supervisory Control And Data Asquisition – диспетчерское управление и сбор данных). Примеры SCADA-систем, ориентированных на создание программного обеспечения для АСУ ТП:

- WinCC фирмы Siemens;
- InTouch фирмы Wanderware;
- отечественная система TraceMode фирмы АдАстра.

Рис. 3.11. Пример мнемосхемы

Создание управляющего программного обеспечения при помощи SCADA-систем заключается в наполнении разработчиком базы знаний, описывающей автоматизируемую промышленную установку, цех и т.п. Для этого необходимо определить:

- множество компонентов, составляющих автоматизируемый объект (станков, двигателей, трубопроводов, исполнительных устройств, датчиков и т.п.), включающее и их характеристики (типы, диапазоны измерений, функции преобразования и т.п.);
- множество отношений между ними, задающее алгоритм функционирования системы (обычно, в виде продукций – т.е. условий и сопоставленных им действий);
- графическую модель автоматизируемого цеха, станка, установки, конвейера и т.п. – так называемую *мнемосхему* (см. рис. 3.11).

Таким образом, разработка программного обеспечения средствами SCADA-систем выполняется одновременно с проектированием АСУ.

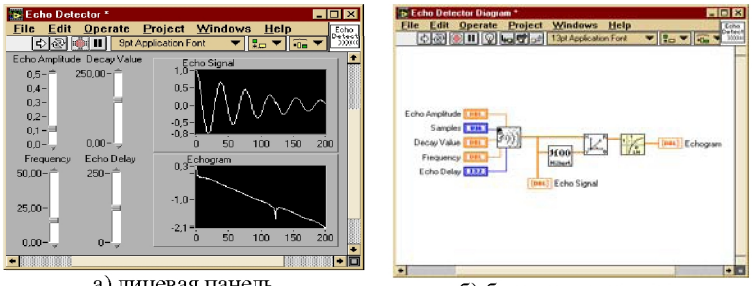
Идеологически близки к SCADA-системам инструментальные среды лабораторной автоматизации, ориентированные на создание программного обеспечения для АСНИ, например, LabVIEW фирмы National Instruments. Они основаны на концепции *виртуального прибора* – программной модели реального или воображаемого прибора или устройства. В процессе разработки программного обеспечения программно реализуются не только средства управления (рукоятки, кнопки, лампочки и т. п.), но и логика работы устройства. Связь программы с физическими объектами осуществляется через интерфейсные узлы, представляющие собой драйвера внешних устройств — АЦП, ЦАП, контроллеров промышленных интерфейсов и т.п.

Программа LabVIEW и состоит из двух частей:

- *лицевой панели*, описывающей внешний интерфейс виртуального прибора (см. рис. 3.12, а);
- *блочной диаграммы*, описывающей логику работы виртуального прибора (см. рис. 3.12, б).

Лицевая панель виртуального прибора содержит средства ввода-вывода, такие как: кнопки, переключатели, светодиоды, верньеры, шкалы, информационные табло и т. п. Они используются человеком для управления виртуальным прибором.

Блочная диаграмма содержит функциональные узлы, являющиеся источниками, приемниками и средствами обработки данных. Также компонентами блочной диаграммы являются терминалы («задние контакты» объектов лицевой панели) и управляющие структуры (являющиеся аналогами таких элементов текстовых языков программирования, как условный оператор «IF», операторы цикла «FOR» и «WHILE» и т. п.). Функциональные узлы и терминалы объединены в единую схему линиями связей.



а) лицевая панель б) блочная диаграмма

Рис. 3.12. Пример виртуального прибора LabView

Использование SCADA-систем и инструментальных сред типа LabVIEW позволяет создавать надежное и эффективное программное обеспечение АСУ, обладающее качественным *человеко-машинным интерфейсом*, в очень короткие сроки.

3.2.3. Языки реального времени

Программное обеспечение для АСУ и СРВ можно писать и на обычных языках программирования. При этом наиболее целесообразно использование специализированных *языков реального времени*, в синтаксис (или в стандартные библиотеки) которых заложены возможности работы с параллельно выполняющимися задачами:

- Modula-2;
- Ada;
- Java.

Эти возможности не зависят от используемой ЭВМ и операционной системы. Например, TopSpeed Modula-2 фирмы JPI обеспечивает кооперативную и вытесняющую многозадачность для программ, выполняющихся даже в принципиально однозадачной операционной системе MS-DOS.

3.3. Программирование в реальном времени

Рассмотрим типовые проблемы программирования в РВ.

1. Выполнение определенных действий в требуемые моменты времени может быть реализовано с использованием разных подходов.

Во-первых, при помощи синхронной задержки текущей задачи на определенный интервал или до определенного момента времени. Например, системный запрос Sleep() в операционной системе MS Windows перемещает поток в очередь заблокированных задач на указанный в параметрах вызова интервал времени (или до конца кванта, если указан 0).

Во-вторых, при помощи асинхронной передачи управления задаче по внешнему запросу, связанному с определенным моментом или интервалом

времени. Роль внешнего запроса может играть событие, произошедшее в другой задаче, или прерывание от внешнего устройства, например, от таймера. В MS Windows системный запрос timeSetEvent() регистрирует указанную в параметрах процедуру в качестве обработчика события, которое будет автоматически возбуждено ядром операционной системы через интервал времени, также указанный в параметрах.

Однако, при использовании программных механизмов, основанных на "отмеривании" относительных интервалов времени, может возникнуть "эффект накопления ошибок" (см рис. 3.13), поэтому предпочтение следует отдавать механизмам, работающим с абсолютными моментами времени.

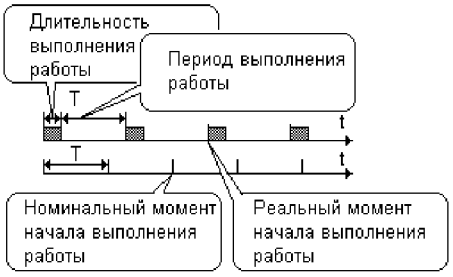
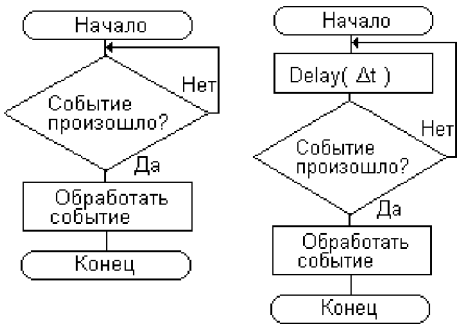


Рис. 3.13. Накопление ошибки

2. Реагирование на внешние события также может быть реализовано поразному:

- синхронно - при помощи "поллинга" ("polling"), т.е. циклической проверки наступления события (см. рис. 3.14);
- асинхронно - с принудительной передачей управления обработчику события в момент появления запроса (например, прерывания).

Достоинство первого подхода - потенциально высокая реактивность. Основной недостаток - в условиях однозадачности или кооперативной многозадачности невозможно реагировать на несколько одновременных событий. А при вытесняющей многозадачности текущая задача, выполняющая циклический опрос, может быть прервана в любой момент на недопустимо большое время. Частичное решение проблемы заключается в комбинировании методов вытесняющей и кооперативной многозадачности: повысить приоритет задачи и внутри цикла самостоятельно передавать управление остальным задачам на некоторый интервал времени Δt, имеющий допустимую величину.



а) Неправильно б) Правильно

Рис. 3.14. Пример организации "поллинга" (циклического опроса)

При использовании второго подхода для обработки многочисленных внешних событий возникает необходимость введения очередей запросов и приоритетности для источников запросов. Также используется "двухступенчатая" обработка: очень короткие "первичные" обработчики только ставят запросы в очередь; длительные "вторичные" обработчики реализуются как отдельные параллельно работающие задачи, извлекающие из очереди задания в соответствии с приоритетами и выполняющие требуемые действия по обработке.

Вопросы и задания по Теме 3

1. В операционной системе реализована вытесняющая многозадачность с квантом времени длиной 1 мс и дисциплиной «EDF». Одновременно запускаются три задачи с длительностями 5, 3 и 2 мс, которым первоначально устанавливаются равные приоритеты. Нарисуйте «диаграмму Ганнта» (т.е. диаграмму, аналогичную рис. 3.4) в предположении, что время переключения контекста (время, затрачиваемое операционной системой для переключения с задачи на задачу) близко к 0.

2. Поясните ситуацию с «инверсией приоритетов». Почему задача Б будет блокировать выполнение задачи А?

3. Как вы думаете, почему применение «разгона приоритетов» нежелательно в операционных системах реального времени?

4. Придумайте пример, иллюстрирующий некорректное взаимодействие задач при использовании алгоритма с блокирующими флагами.

5. Каким должно быть первоначальное значение семафорной переменной S для предотвращения одновременного доступа к единственному ресурсу двух задач? Трех задач? К двум ресурсам?

6. Составьте программу на языке FBD для расчета дискриминанта квадратного уравнения.

ТЕМА 4. ПРИМЕР ПРОЕКТИРОВАНИЯ И РЕАЛИЗАЦИИ

Особенности проектирования и практической реализации систем реального времени рассмотрим на следующем конкретном примере: «Автоматизированная система для измерения биений вала».

4.1. Постановка задачи

На электростанциях для передачи крутящего момента между рабочим колесом и подвижной обмоткой гидроагрегатов и парогенераторов используется горизонтально расположенный стальной цилиндрический вал диаметром D=0.8 м., частота вращения которого F=68.2 об/мин (см. рис. 4.1). Необходимо измерять «биения» вала, т.е. отклонения ΔZ внешней поверхности вращения от номинального положения Z_0, вызванные неидеальностью формы вала в поперечном сечении; прогибом вала; прецессией оси вращения вала. В процессе измерения требуется контролировать выполнение условия

$|Z_0-\Delta Z| \le \Delta Z_{max}=3$ мм.

Измерения необходимо производить в течение одного оборота вала на участке длиной L=5 м с шагом $L_ш$=0,5 м. Предельная абсолютная погрешность измерения биений – 0,2 мм, позиционирования вдоль вала – 10 мм.

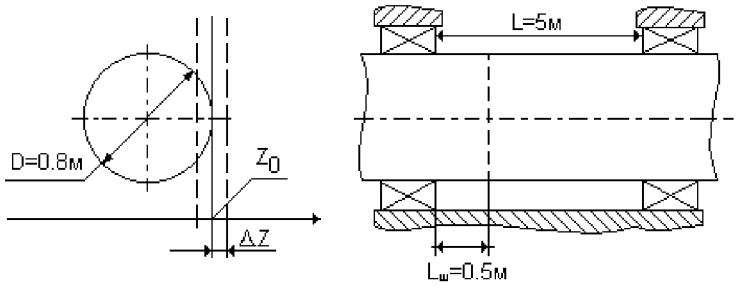


Рис. 4.1. Вал рабочего колеса

4.2. Общее описание системы

Датчик Д1, используемый для измерения «биений», располагается на подвижной каретке, способной двигаться по направляющим рейкам вдоль вала. Для перемещения каретки используются ролики, вращаемые электродвигателем постоянного тока ЭД. Контроль положения каретки выполняется при помощи датчика Д2. Выходные сигналы датчиков Д1 и Д2 имеют унифицированные пределы 0..10В, следовательно, промежуточные преобразователи для подключения к АЦП не требуются. Для управления электродвигателем ЭД необходим электрический сигнал определенного уровня и опреде-

ленной мощности, который генерируется специализированным управляющим контроллером УК, совмещенным с усилителем мощности.

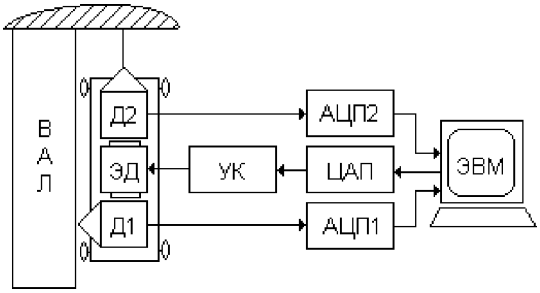


Рис. 4.2. Функциональная схема АС

4.3. Описание компонентов УСО

1. Измерения «биений» вала выполняются при помощи бесконтактного вихретокового датчика линейных перемещений.

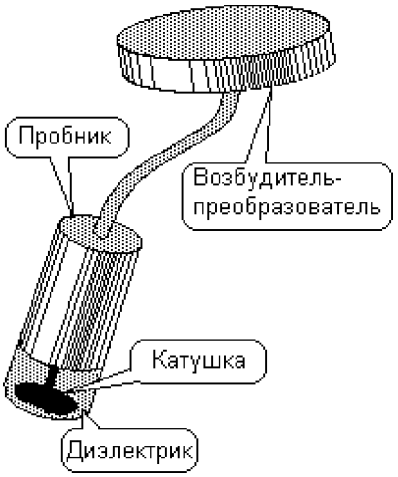


Рис. 4.3. Вихретоковый датчик

Вихретоковый датчик (см. рис. 4.3) состоит из бесконтактного пробника, на конце которого в диэлектрической втулке размещается плоская катушка индуктивности. Электроника датчика, соединенная с катушкой коаксиальным кабелем, генерирует сигнал возбуждения электромагнитных колебаний в катушке. В результате, на поверхности контролируемого металлического объекта возникают вихревые токи, в свою очередь, изменяющие активное и индуктивное сопротивление катушки, улавливаемое преобразователем.

Для измерений «биений» вала используется датчик со следующими техническими характеристиками: пределы измерения 0..5 мм; основная погрешность – 3%; полоса частот – до 2 КГц; унифицированный выходной сигнал 0..10В; питание – 24 В.

2. Измерение позиционирования каретки вдоль вала выполняется при помощи тросикового датчика Д2 линейных перемещений.

Принцип действия (см. рис. 4.4): на барабан внутри датчика намотан в один слой измерительный тросик, при вытягивании которого барабан вращается, а потенциометрический измеритель преобразует угол поворота в электрический сигнал. Возврат тросика и намотка на барабан осуществляются с помощью спиральной возвратной пружины.

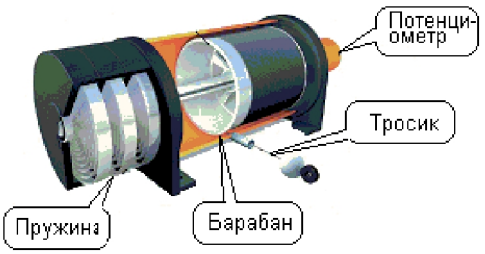


Рис. 4.4. Тросиковый датчик

Используется модель со следующими характеристиками: диапазон измерений – 0..6 м; унифицированный выходной сигнал 0..10В; разрешающая способность 2 мм; максимальная скорость вытягивания троса – 3 м/с; питание – 24 В.

3. Для перемещения каретки вдоль вала используется мотор-редуктор постоянного тока со следующими характеристиками: мощность - 70 Вт; питание – 24В; частота вращения ротора – до 2400 об/мин; передаточное число редуктора – 15; частота вращения вала – до 80 об/мин; возможен реверс.

Широко распространены два метода внешнего управления оборотами электродвигателей (см. рис. 4.5):

- для мощных асинхронных двигателей переменного тока чаще применяется метод, использующий ШИМ (широтно-импульсную модуляцию) – при этом частота вращения ротора пропорциональна ширине прямоугольных импульсов, подаваемых на управляющий вход со специального *частотного преобразователя*;
- для маломощных двигателей постоянного тока чаще применяется управление уровнем напряжения на управляющем входе.

Для управления рассмотренным выше мотор-редуктором используется специализированный контроллер второго типа (см. рис. 4.6).

4.5. Два метода управления оборотами

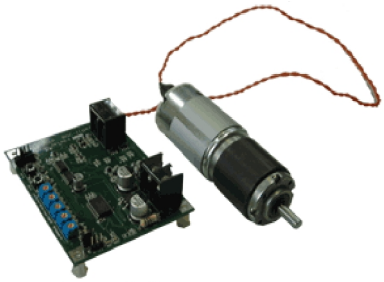


Рис. 4.6. Электродвигатель с контроллером

4. Для ввода аналоговых сигналов с датчиков и для вывода управляющих аналоговых сигналов на контроллер двигателя применяется автономный модуль, подключаемый к ЭВМ по интерфейсу USB (см. рис. 2.10). Технические характеристики модуля:

- 8-канальный 12-битовый АЦП, максимальная частота опроса 10 КГц, унифицированные диапазоны входных сигналов 4..20 мА, 0..5В, 0..10В, ±2,5В, ±5В;
- 2-канальный 12-битовый ЦАП, максимальная частота вывода 150 Гц, унифицированный диапазон выходных сигналов – 0..5В;
- 24 канала цифрового ввода-вывода;
- 32-битовый таймер-счетчик с частотой пересчета 5 МГц.

4.4. Выбор частоты опроса датчика биений

При измерении случайно изменяющегося во времени сигнала (*случайного процесса*) существует общее правило, позволяющее выбирать частоту дискретизации этого сигнала. Оно базируется на *теореме Котельникова* (известной так же, как *критерий Найквиста*). Вот это правило: *частота дискретизации сигнала выбирается из условия* Fизм ≥ 2×Fmax, где Fmax – *максимальная частота изменения сигнала*.

Имея априорную информацию о поведении процесса (например, выполнив кратковременные измерения с максимально возможной частотой), можно оценить частоту его изменения. (Примечание: при этом предполагается выполнение условий *стационарности* случайного процесса – т.е. неизменности его параметров во времени; и *эргодичности* случайного процесса – т.е. неизменности его параметров от реализации к реализации.) Для этого, воспользовавшись преобразованием Фурье, можно получить и проанализировать спектр сигнала, т.е. представление его в виде суммы синусоид со всевозможными частотами.

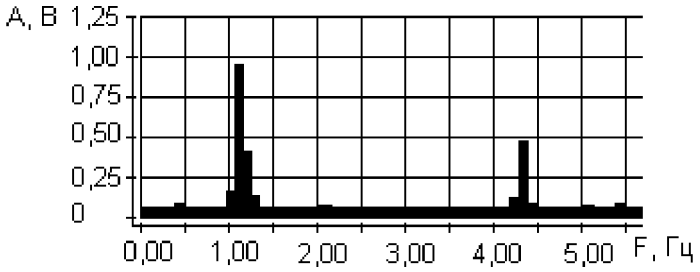


Рис. 4.7. Спектр сигнала «биений» вала

Анализ спектра «биений» вала (см. рис. 4.7) позволяет сделать вывод о значительном вкладе в формирование сигнала всего двух частот: $F_1 \approx 1{,}14$ Гц и $F_2 \approx 4{,}2$ Гц. Следовательно, достаточно измерять сигнал с частотой $2 \times F_2 \approx 8.4$ Гц. Выбираем $F_{изм}=100$ Гц (период $\Delta T_{изм}=0{,}01$ с, 88 измерений на оборот).

4.5. Расчет погрешности измерительных каналов

Любые измерения выполняются с ошибкой, т.е. с *погрешностью* – отклонением измеренного значения величины от ее истинного значения: $\Delta X = X_{изм} - X$. Свой вклад в эту ошибку вносят:

- неидеальность метода измерений (эта составляющая полной погрешности называется *методической погрешностью*);
- неидеальность использованных технических средств (эта составляющая называется *инструментальной* или *приборной погрешностью*).

Важной технической характеристикой любых средств измерений, будь то линейка, градусник, датчик, АЦП или целый измерительный канал автоматизированной системы, является приписываемая им величина инструментальной погрешности. Она характеризует предельное значение погрешности, которая может возникать при использовании данного средства измерения, и указывается в одной из трех форм:

1. абсолютной $\Delta = Xизм - X;$

2. относительной $\gamma = \Delta / X;$

3. приведенной $\delta = \dfrac{\Delta}{X_{max} - X_{min}} \times 100\% \cdot$

В качестве математической модели погрешности обычно принимается нормально распределенная случайная величина, которая может быть охарактеризована одним из двух способов:

1. как совокупность математического ожидания m (характеризующего *систематическую составляющую* погрешности) и дисперсии σ^2 (характеризующей *случайную составляющую*);

2. в виде минимальной Δ_{min} и максимальной Δ_{max} границ возможных значений и доверительной вероятности $P_{дов}$ попадания в этот интервал.

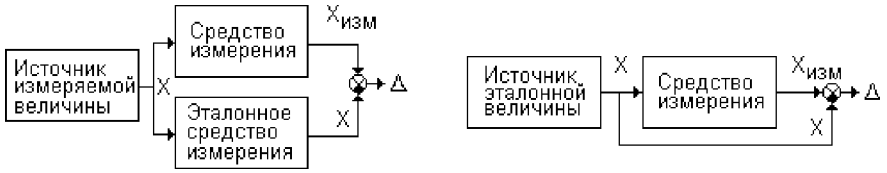
Между двумя этими формами представления случайной величины существует простая связь:

$$m = (\Delta_{min}+\Delta_{max})/2; \quad (4.1)$$

$$\Delta = \Delta_{max}-m = m-\Delta_{min}= K\cdot\sigma, \quad (4.2)$$

где K – некий коэффициент, зависящий от $P_{дов}$ и выбираемый из таблицы квантилей нормального распределения (например, при $P_{дов}$=0,95 принимают K=1,96).

Если средство измерений существует, то свойственная ему предельная погрешность измерений Δ определяется экспериментально в результате применения одного из двух методов (см. рис. 4.8). Именно эта характеристика указывается в документации на средство измерения.



а) Метод эталонного компонента б) Метод источника эталонной величины

Рис. 4.8. Два метода определения погрешности средств измерений

На этапе проектирования, когда средство измерений еще не существует, можно приближенно рассчитать свойственную ему погрешность измерений по известным характеристикам погрешности отдельных компонентов.

Первый способ расчета полной погрешности средства измерений заключается в том, что погрешности отдельных компонентов, пересчитанные на вход всей измерительной цепи, представляются в форме (m, σ^2), а расчет характеристик суммарной погрешности выполняется по правилам:

$$m_\Sigma = m_1+m_2+\ldots m_M = \sum_{i=1}^{M} m_i ; \quad \text{.....} \quad .(4.3)$$

$$\sigma_\Sigma = \sqrt{\sigma_1^2+\sigma_2^2+\ldots+\sigma_M^2} = \sqrt{\sum_{i=1}^{M}\sigma_i^2} . \quad .(4.4)$$

Второй способ расчета:

$$\Delta_{min\Sigma}=\Delta_{min1}+\Delta_{min2}+\ldots+\Delta_{minM}; \quad (4.5)$$

$$\Delta_{max\Sigma}=\Delta_{max1}+\Delta_{max2}+\ldots+\Delta_{maxM}. \quad (4.6)$$

Первым способом обычно складываются основные, а вторым – дополнительные погрешности компонентов (см. п. 2.1).

В нашем случае средствами измерения

являются измерительные каналы автоматизированной системы. Рассчитаем их погрешности. Аппаратная часть обоих каналов имеет одинаковую структуру (см. рис. 4.9).

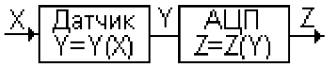


Рис. 4.9. Структура измерительных каналов

Для простоты расчетов предположим, что у всех компонентов систематическая составляющая погрешности пренебрежимо мала, т.е. равна нулю, а функции преобразования линейны.

Датчик измерения перемещений каретки имеет погрешность (разрешающую способность) Δ_1=2 мм. В 12-разрядном АЦП с диапазоном измерения 0..10В погрешность дискретизации составляет h/2=(10В/4096)·0,5=0,0012В. Пересчитаем ее на вход датчика, т.е. определим, какое перемещение на входе датчика вызвало бы такое изменение напряжения у него на выходе. Для этого определим линейную функцию преобразования датчика Y=(10В/6000мм)·X=0,00167·X В/мм, следовательно, искомая погрешность на входе датчика составила бы Δ_2= 0,0012/0,00167=0,73 мм. Считая, что K=1,96, в соответствие с (4.2), получаем $\theta_1=\Delta_1/K$=1,02 мм и $\theta_2=\Delta_2/K$=0,37 мм. Таким образом, с.к.о. полной погрешности, согласно (4.4), составит $\theta_{ИК1}=\sqrt{\theta_1^2+\theta_2^2}$=1,086 мм, что, согласно (4.2), соответствует $\Delta_{\Sigma ИК1}=K\cdot\theta_{ИК1}$=2,12 мм.

Датчик измерения биений вала имеет основную погрешность 3% от ширины диапазона измерения (0..5 мм), что составляет Δ_1=0,15 мм. Погрешность дискретизации АЦП имеет порядок 10^{-4} мм и пренебрежимо мала, поэтому $\Delta_{\Sigma ИК2}$=0,15 мм.

Итак, можно сделать вывод, что рассчитанные характеристики погрешности измерительных каналов автоматизированной системы удовлетворяют требованиям, приведенным в п. 4.1.

4.6. Адаптивное управление перемещением каретки

Одной из типичных задач автоматизированного управления является регулирование параметров поведения инертного технического процесса. Примеры: нагревание или охлаждение до нужной температуры больших объемов вещества; перемещение и точное позиционирование массивных объектов и т.п.

Решение этой задачи осуществляется введением в контур управления обратной связи со включенными в нее регуляторами (см. рис. 4.10).

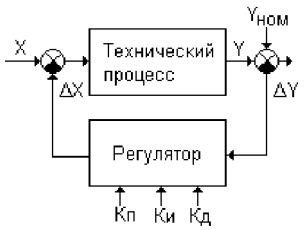


Рис. 4.10. Управление с обратной связью

На этом рисунке X – параметр, управляющий техническим процессом; Y – выход, характеризующий поведение технического процесса; $Y_{ном}$ – номинальное значение выходного параметра, соответствующее «правильному» поведению технического процесса; ΔY – ошибка поведения технического процесса; ΔX - корректирующее воздействие на управляющий параметр; $K_п$, $K_и$ и $K_д$ – параметры регулятора.

В общем случае, работа регулятора может быть описана следующей формулой:

$$\Delta X = \Delta X_п + \Delta X_и + \Delta X_д = K_п \cdot \Delta Y + K_и \cdot \int \Delta Y dt + K_д \cdot \frac{d\Delta Y}{dt} \cdot \qquad (4.7)$$

В соответствии с этой формулой, корректирующее воздействие ΔX есть результат сложения трех составляющих:

- $\Delta X_п = K_п \cdot \Delta Y$ – пропорциональной, зависящей от текущего значения ошибки;
- $\Delta X_и = K_и \cdot \int \Delta Y dt$ - интегральной, зависящей от ошибки, накопленной к текущему моменту времени;
- $\Delta X_д = K_д \cdot \frac{d\Delta Y}{dt}$ - дифференциальной, зависящей от текущей скорости изменения ошибки.

«Вкладом» той или иной составляющей в ΔX можно управлять при помощи коэффициентов $K_п$, $K_и$ и $K_д$. Регуляторы, в которых задействованы все три составляющие, называются *ПИД-регуляторами*. Возможны частные случаи: П-регуляторы, ИД-регуляторы, ПИ-регуляторы и т.п.

ПИД-регулятор может быть оформлен в виде несложной программы. Кроме того, для систем полностью автоматического управления ПИД-регуляторы выпускаются в виде автономных устройств с унифицированными аналоговыми входами и выходами (см. рис. 4.11).

Рис. 4.11. Аппаратный ПИД-регулятор

Для задачи перемещения и точного позиционирования каретки параметры общей схемы регулирования могут быть проинтерпретированы следующим образом: X – скорость вращения электродвигателя; Y – текущее поло-

жение каретки; $Y_{ном}$ – «цель», т.е. положение, которое каретка должна в итоге занять; ΔY – расстояние до «цели»; ΔX – корректирующее приращение или уменьшение скорости вращения двигателя.

4.7. Алгоритмы и методы обработки данных

Данные, поступающие на ЭВМ с управляемого объекта, не могут быть использованы без обработки. Различают два этапа обработки:

- *первичный* (или *предварительный*), в ходе которого (обычно, в режиме реального времени) приводят данные к удобному для дальнейшего использования виду – очищают от помех, выявляют и исправляют искажения, удаляют избыточность, рассчитывают оценки измеряемых величин и их погрешности и т.п.;
- *вторичный* (или *основной*), в процессе которого строят и рассчитывают параметры математических моделей, выполняют их анализ и делают выводы, используемые для формирования управляющих воздействий на объект.

Иногда часть задач, характерных для вторичной обработки, также решают в реальном времени – это режим *экспресс-анализа данных*.

Обычно наличие обоих этапов характерно для АСНИ и для АСУ ТП, управляющих сложными процессами, на ход которых влияют многочисленные разнородные факторы. В остальных случаях и для остальных классов автоматизированных систем обычно хватает этапа первичной обработки.

Далее будут рассмотрены алгоритмы и методы, характерные именно для этого этапа.

4.7.1. Классификация сигналов

Методы, используемые для обработки данных, сильно зависят от вида сигнала, в котором эти данные содержатся (см. рис. 4.12).

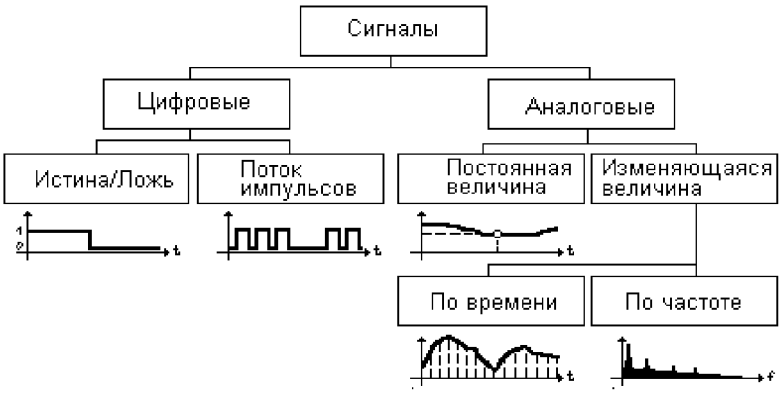


Рис. 4.12. Классификация сигналов

Цифровые сигналы – это сигналы, значения которых представимы в виде двух логических уровней: «0» и «1». Данные подобного вида поступают с цифровых датчиков, реле, сигнализаторов состояния и т.п. Информация может быть заключена как в самом значении сигнала, так и в частоте его изменения, в интервалах времени между изменениями, в количестве этих изменений и т.п.

Аналоговые сигналы - это сигналы, значения которых непрерывны по уровню. Информация в сигналах подобного вида может быть заключена как в одиночном значении (рассматриваемом безотносительно значений, измеренных в другие моменты времени), в характере изменения значения по времени, в частоте его изменения и т.п.

Рассмотренная классификация не является взаимоисключающей, т.е. один и тот же сигнал, в зависимости от решаемой задачи, может быть измерен и обработан как принадлежащий к разным классам.

4.7.2. Первичная обработка данных

Поскольку автоматизированная система расположена на территории электростанции, то на результат любых измерений оказывают влияние сильные электромагнитные помехи, приводящие к значительным случайным погрешностям. Поэтому целесообразно производить многократные измерения физических величин и рассчитывать их оценки в результате статистической обработки.

1. Данные, поступающие с датчика, измеряющего положение подвижной каретки, представляют собой постоянную величину. Результат измерения постоянной величины обычно представляется в форме $\widetilde{X} \pm \Delta X_{изм}$, где $\widetilde{X}$ - приближенная оценка измеряемой величины; $\Delta X_{изм}$ – приближенная оценка погрешности измерений, представленная в виде верхней и нижней границ диапазона значений. Расчеты ведутся по формулам:

- $$\widetilde{X} = \frac{1}{N}\sum_{i=1}^{N} X_i \,, \qquad (4.8)$$
- $$\Delta X_{изм} = K \cdot \widetilde{\sigma}_{изм}, \qquad (4.9)$$
- $$\widetilde{\sigma}_{изм} = \sqrt{S^2 + \theta^2} \,, \qquad (4.10)$$
- $$S^2 = \frac{1}{N-1}\sum_{i=1}^{N} \left(X_i - \widetilde{X}\right)^2 , \qquad (4.11)$$

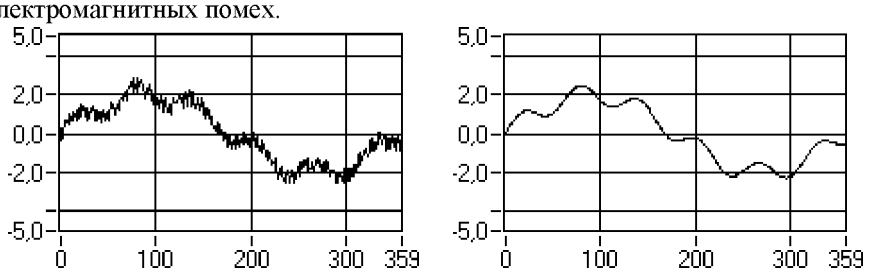
где N - количество наблюдений («замеров»); X_i – результат i-го наблюдения («замера»); K – некий коэффициент, зависящий от N и от $P_{дов}$; $\widetilde{\sigma}_{изм}$ - оценка с.к.о. полной ошибки измерений; S^2 - дисперсия случайной составляющей погрешности измерений; θ^2 – дисперсия инструментальной погрешности,

приписываемой использованному средству измерения (см. раздел «Расчет погрешности измерительных каналов»).

2. Данные, поступающие с датчика, который измеряет «биения» вала, представляют собой изменяющийся во времени случайный процесс вида

$$X(t) = g(t)+\xi(t), \qquad (4.12)$$

где g(t) – «*тренд*», т.е. детерминированная функция, описывающая поведение процесса во времени; ξ(t) – случайный «шум», вызванный воздействием электромагнитных помех.



а) Исходный «зашумленный» процесс б) Сглаженный процесс, «тренд»

Рис. 4.13. Сглаживание эмпирических данных

Для разделения процесса на «тренд» и «шум» используются разные подходы. Проиллюстрируем один из них, основанный на методах цифровой фильтрации.

Метод *скользящего среднего по n точкам* представляет собой применение для каждого i-го отсчета выборки следующей системы правил (на примере n=3):

$$\begin{aligned} \tilde{Z}_1 &= (5Z_1 + 2Z_2 - Z_3)/2n, \quad i = 1 \\ \tilde{Z}_i &= (Z_{i-1} + Z_i + Z_{i+1})/n, \quad 1 < i < n \\ \tilde{Z}_n &= (5Z_n + 2Z_{n-1} - Z_{n-2})/2n, \quad i = n \end{aligned} \qquad (4.13)$$

где $\tilde{Z}_i$ - новое, «сглаженное» значение отсчета.

Обратите внимание, что этот метод работает «с задержкой», т.е., например, при n=3 для выполнения расчетов для i-го элемента выборки необходимо дождаться появления (i+1)-го элемента. Можно избавиться от задержки, применив правило $\tilde{Z}_i = (Z_{i-n+1} + Z_{i-n+2} + \ldots + Z_i)/n$, но в этом случае погрешность сглаживания будет выше.

Также можно использовать *метод медианной фильтрации по n точкам*, основанный на правиле:

$$\tilde{Z}_i = \underset{Z}{\mathrm{med}}(Z_{i-n+1}, Z_{i-n+2}, \ldots, Z_i), \qquad (4.14)$$

где med(·) означает среднее по величине из группы значений.

После сглаживания необходимо рассчитать Z_{min} и Z_{max} – предельные значения в выборке и проверить выполнение условия Z_{min} >-3 мм и Z_{max}<+3 мм.

4.7.3. Оптимизация циклических расчетных алгоритмов

Введем обозначения: ω - числовая последовательность; F(ω) - некоторая функция, вычисляемая по числовой последовательности; * - операция присоединения нового члена к числовой последовательности; x - новый член последовательности. Функция F(ω) называется *индуктивной,* если существует такая функция G, что: F(ω*x) = G(F(ω),x), т.е. по значению функции и по значению нового члена последовательности можно вычислить новое значение функции.

Пример 1. Пусть F(ω) - сумма членов последовательности. F(ω*x)=F(ω)+x= G(F(ω),x). Значит, F(ω) - индуктивная.

Пример 2. Пусть F(ω)=max(ω)-min(ω) - разность между максимальным и минимальным членом последовательности. Зная значение этой функции и новый член x, новое значение функции вычислить невозможно. Значит, F(ω) - не индуктивная.

Если F(ω) - не индуктивная функция, но существует другая такая индуктивная функция H(ω), что F(ω*x)=G(F(ω), H(ω), x), то H(ω) - *индуктивное расширение* для F(ω).

Пример 3. Пусть F(ω) - количество элементов ω, равных max(ω). Функция max(ω) - индуктивное расширение для F(ω), т.к. F(ω*x) = F(ω)+1, если x=max(ω), и F(ω*x)=F(ω) - иначе.

Метод индуктивных функций удобен для построения алгоритмов расчета по выборке «скользящих» числовых оценок, т.е. оценок, получаемых в процессе сбора данных, а не по результатам накопления полного их набора.

Пример 4. Пусть F(ω) - среднее арифметическое членов последовательности. Индуктивное расширение для нее N(ω) - количество членов последовательности, поскольку F(ω*x) = ((F(ω)·(N(ω)))+x)/N(ω*x). Таким образом, можно получать новые значения среднего арифметического по части выборки и «пересчитывать» их по мере получения новых членов:

```
(* Традиционный метод *)
mx := 0; n := 0;
```

```
(* «Индуктивный» метод *)
mx := 0; n := 0;
```

```
while not конец do begin
 mx := mx + Новое_значение;
 n:=n+1;
end;
mx := mx / n;
```

```
while not конец do begin
 x := Новое_значение;
 mx := (mx * n + x)/(n+1);
 n:=n+1;
end;
```

Подобная модификация возможна также для алгоритмов расчета моментов высших порядков, например, дисперсии.

4.8. Алгоритмы сбора данных и управления

Алгоритм работы автоматизированной системы имеет циклический характер и может быть представлен схемой, изображенной на рис. 4.14. Особенности реализации отдельных его частей (блоков) пояснены ниже.

1. При сборе данных с датчиков (блоки 2 и 3 алгоритма) возникает задача расчета значений измеряемых величин. Измерительные каналы имеют структуру, изображенную на рис. 4.9. Функции преобразования датчиков линейны $Y=A\cdot X$, а функции преобразования АЦП в соответствие с (2.1) имеют вид $Z=[(Y+0.5)/h]=[(Y+0.5)/(D/2^N)]=[2^N\times(Y+0.5)/D]=[Y\times(2^N/D)+2^{N-1}/D]$, где N-разрядность АЦП; D – входной диапазон АЦП; $[\cdot]$ – операция выделения целой части. Следовательно, полная функция преобразования измерительного канала будет иметь вид $Z=[X\times A\times(2^N/D)+2^{N-1}/D]$, а приближенное значение X можно вычислять по формуле $X\approx(Z\times D)/(A\times 2^N)-0.5/A$.

Для датчика, измеряющего «биения» вала, чувствительность A=2 В/мм, а для датчика, измеряющего положение каретки, A=0,00167 В/мм. АЦП, используемые в обоих измерительных каналах, идентичны и имеют N=12 и D=10 В. Следовательно, формулы для расчетов значений измеряемых величин:

$X=0.0012\cdot Z-0.25$ мм – для канала измерения «биений» вала;

$X=1.4619\cdot Z-299.4$ мм – для канала измерения положения каретки.

2. Измерение положения каретки (блок 2 алгоритма) выполняется многократно (n=30) с максимально возможной частотой, расчет значения измеряемой величины выполняется в соответствие с формулой (4.8) по «скользящему» алгоритму, рассмотренному в разделе «Оптимизация циклических расчетных алгоритмов». При управлении перемещением подвижной каретки (блок 2 алгоритма) используется алгоритм П-регулятора. Этот алгоритм работает в дискретном времени, поскольку электродвигатель способен воспринимать управляющие воздействия, поступающие на его вход с периодом не меньше $\Delta t_2=0,2$ сек. Завершение перемещения каретки (блок 2.10) выполняется при одновременном достижении двух условий: 1) абсолютное значение ошибки позиционирования не превышает $L_ш=10$ мм; 2) абсолютное значение скорости перемещения не превышает $L_ш/\Delta t_2=10$ мм/0,2с=50 мм/с. Для дости-

жения необходимой «точности» управления значение коэффициента пропорционального усиления П-регулятора должно выбираться из условия $K_п \le 1/\Delta t_2$.

3. Измерение биений вала (блок 3 алгоритма) в каждом сечении выполняется n=88 раз с периодом $\Delta T_{изм}$=0,01 с. После измерений (блок 4) выполняется сглаживание выборки методом медианной фильтрации по 3 точкам и отображение графика биений на экране ЭВМ, а также контроль выборочных значений на выполнение условия $|Z_i\text{-}2.5|<3$.

4. Перемещение каретки осуществляется на роликах радиусом R=25 мм, следовательно максимальная скорость перемещения составит $|V_{max}|=80\times 2\pi R=209$ мм/с, а обратная функция преобразования канала управления кареткой U=V/41,8+5 В·с/мм.

Рис. 4.14. Общий алгоритм

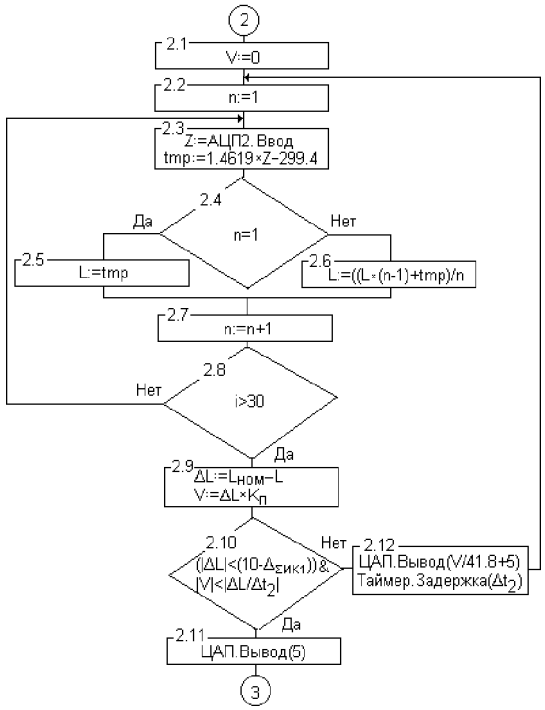


Рис. 4.16. Перемещение каретки

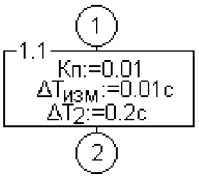


Рис. 4.15. Инициализация работы

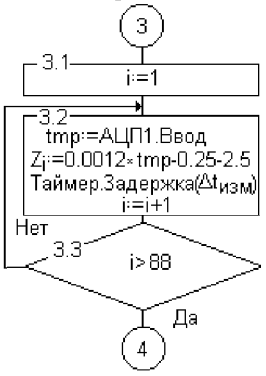


Рис. 4.17. Измерение биений вала

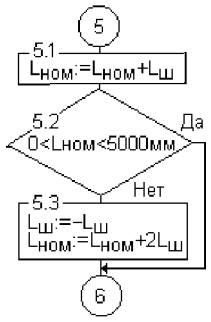


Рис. 4.19. Расчет нового положения каретки

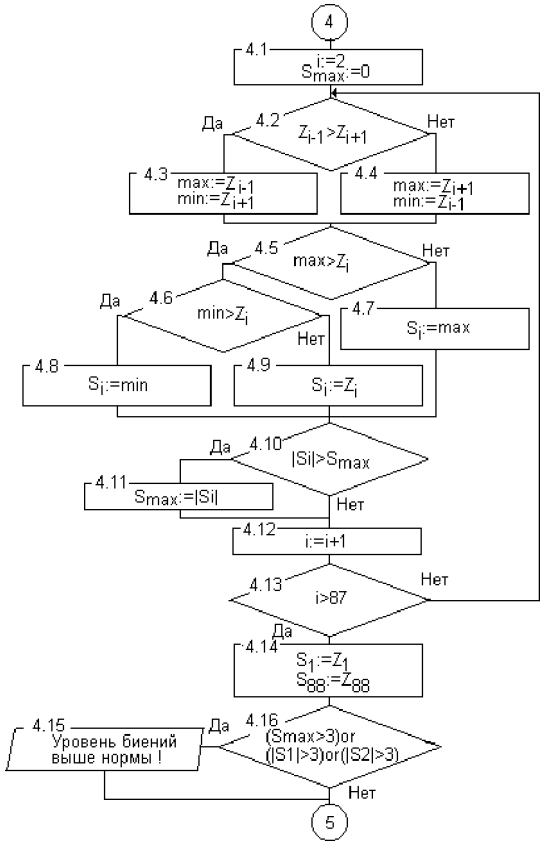


Рис. 4.18. Обработка результатов измерения биений вала

Вопросы и задания по Теме 4

1. С какой частотой целесообразно оцифровывать слышимый звук?

2. Функция преобразования датчика $Y=A \cdot X+B$, погрешность на выходе равна Δ. Пересчитайте ее на вход.

3. Почему для управления кареткой достаточно П-регулятора? Придумайте условия, при которых потребовался бы ПД-регулятор.

4. Придумайте пример сигнала, который может быть измерен и обработан и как цифровой, и как аналоговый.

5. Сгладьте выборку (1, 5, 2, 4) методом «скользящего среднего по 3 точкам» и методом «медианной фильтрации по 3 точкам».

6. Является ли индуктивной функция «Сумма четных членов ряда»?

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

К темам 1, 2, 3 и 4:

Олссон, Г.Цифровые системы автоматизации и управления / Г. Олссон, Дж. Пиани. – СПб.: Невский Диалект, 2001. – 557 с.

К теме 3:

Ослэндер, Д.М. Управляющие программы для механических систем: объектно-ориентированное проектирование систем реального времени / Д.М. Ослэндер, Дж. Р. Риджли, Дж. Д Ринггенберг. – М.: БИНОМ, 2004. – 413 с.

Андреев, Е.Б. SCADA-системы: взгляд изнутри / Е.Б. Андреев, Н.А. Куцевич, О.В. Синенко. – М.: Издательство «РТСофт», 2004. – 176 с.

Тревис, Дж. LabVIEW для всех / Дж. Тревис. – М.: ДМК Пресс, 2004. – 544 с.

Зыль, С.Н. QNX Momentics: основы применения / С.Н. Зыль. – СПб.: БХВ-Петербург, 2005 – 256 с.

К темам 2 и 4:

Болтон, У. Карманный справочник инженера-метролога / У. Болтон. – М.: Издательский дом «Додэка-XXI», 2002. – 384 с.

Парк, Дж. Сбор данных в системах контроля и управления / Дж. Парк, С. Маккей. – М.: Группа ИДТ, 2006. – 504 с.

Учебное издание

СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

Обзорный курс лекций

Редактор Ю.Н.Литвинова
Доверстка Ю.Н.Литвинова

Подписано в печать 10.12.08. Формат 60x84$^{1}/_{16}$.
Бумага офсетная. Печать офсетная.
Усл. печ. л. 3,25
Тираж 100 экз. Заказ .Арт.

Самарский государственный
аэрокосмический университет
443086 Самара, Московское шоссе, 34.

Изд-во Самарского государственного
аэрокосмического университета
443086 Самара, Московское шоссе, 34.