

МЕТОДИ, УСТРОЙСТВА И СИСТЕМИ
ЗА СЪБИРАНЕ И ПРЕОБРАЗУВАНЕ НА
ИНФОРМАЦИЯ

1. ОБЩА СТРУКТУРА И ПРЕДНАЗНАЧЕНИЕ НА СИСТЕМИТЕ ЗА СЪБИРАНЕ И ПРЕОБРАЗУВАНЕ НА ИНФОРМАЦИЯ	1
<i>Какви знания и умения ще придобиете?</i>	1
<i>Къде се прилагат?</i>	1
ВЪВЕДЕНИЕ	1
1.1. ПРЕДНАЗНАЧЕНИЕ НА СИСТЕМИТЕ ЗА СЪБИРАНЕ И ОБРАБОТКА НА ИНФОРМАЦИЯ	2
1.1.1. <i>Предназначение</i>	2
1.1.2. <i>Взаимодействие</i>	2
1.1.3. <i>Процеси</i>	3
1.2. ОБЩА БЛОКОВА СХЕМА НА СИСТЕМИТЕ ЗА СЪБИРАНЕ И ОБРАБОТКА НА ИНФОРМАЦИЯ	4
1.3. ИНФОРМАЦИОННИ ПОТОЦИ В СИСТЕМИТЕ ЗА СЪБИРАНЕ И ОБРАБОТКА НА ИНФОРМАЦИЯ И ТЯХНАТА ИНТЕНЗИВНОСТ	6
1.4. ЗАДАЧИ, РЕШАВАНИ ОТ СИСТЕМИТЕ ЗА СЪБИРАНЕ И ОБРАБОТКА НА ИНФОРМАЦИЯ	7
1.5. ОСНОВНИ ПРОБЛЕМИ ПРИ ПРОЕКТИРАНЕ НА СИСТЕМИТЕ ЗА СЪБИРАНЕ И ОБРАБОТКА НА ИНФОРМАЦИЯ	8
1.5.1. <i>Проблеми при избор на датчици и изпълнителни механизми</i>	9
1.5.2. <i>Проблеми при избор на управляващата компютърна система</i>	10
1.5.3. <i>Проблеми с комуникацията с човека оператор</i>	11
1.5.4. <i>Проблеми с надеждността на системата</i>	11
1.5.5. <i>Проблеми при изграждане на хранващия блок</i>	12
ОБЩЕНИЕ	13
Контролни въпроси	14

2. ВЪВЕЖДАНЕ И ИЗВЕЖДАНЕ НА СИГНАЛИ ОТ КОМПЮТЪРНАТА СИСТЕМА.....	1
ВЪВЕДЕНИЕ	1
2.1. ВЪВЕЖДАНЕ НА ПОТЕНЦИАЛНИ И ИМПУЛСНИ СИГНАЛИ	2
2.1.1. Дефиниране	2
2.1.2. Типична схематехника на входните вериги, галванично развързване.....	3
2.1.3. Защита от импулсни смущения	4
2.2. ВЪВЕЖДАНЕ НА АНАЛОГОВИ СИГНАЛИ.....	7
2.2.1. Дефиниране	7
2.2.2. Класическа блокова схема на многоканален АЦП.....	7
2.2.3. Повишаване на точността и динамиката.....	11
2.2.4. Роля и разчет на входния филтър.....	15
2.2.5. Проблем с галваничното развързване при въвеждане на аналогови сигнали.....	17
2.3. ИЗВЕЖДАНЕ НА ПОТЕНЦИАЛНИ И ИМПУЛСНИ СИГНАЛИ	19
2.3.1. Видове и реализация.....	19
2.3.2. Типова схематехника с галванично развързване.....	20
2.4. ИЗВЕЖДАНЕ НА АНАЛОГОВИ СИГНАЛИ.....	23
2.4.1. Типова схематехника на изходните блокове	23
2.4.2. Реализация на многоканален ЦАП.....	23
2.4.3. Проблеми с галваничното развързване при извеждане на аналогови сигнали.....	24
ОБОБЩЕНИЕ.....	25

3. ПЪРВИЧНА ОБРАБОТКА НА СИГНАЛИТЕ В КОМПЮТЪРНАТА СИСТЕМА	1
ВЪВЕДЕНИЕ	1
3.1. ЦЕЛИ И ЗАДАЧИ НА ПЪРВИЧНАТА ОБРАБОТКА	1
3.2. БАЛАНС НА РЕШАВАНИТЕ ЗАДАЧИ ОТ ПЪРВИЧНАТА ОБРАБОТКА ЗА АПАРАТНА И ПРОГРАМНА РЕАЛИЗАЦИЯ.....	2
3.3. НОРМАЛИЗАЦИЯ	4
3.3.1. <i>Нормализация на амплитудните параметри</i>	4
3.3.2. <i>Нормализация на времевите параметри</i>	4
3.4. ФУНКЦИИ НА ПРЕОБРАЗУВАНЕ	5
3.4.1. <i>Линейно-отсечкова интерполация</i>	6
3.4.2. <i>Интерполация с полином</i>	7
3.4.3. <i>Апроксимация по метода на най-малките квадрати</i>	8
3.4.4. <i>Сплайн апроксимация</i>	9
3.4.5. <i>Апроксимация, производна и интеграл</i>	10
3.5. СТАТИСТИЧЕСКА ОБРАБОТКА	11
3.5.1. <i>Статистическа обработка в амплитудна област</i>	11
3.5.2. <i>Статистическа обработка във времева област</i>	12
3.6. ФИЛТРАЦИЯ.....	13
3.6.1. <i>Филтрация преди въвеждането на сигнала в компютърната система</i>	13
3.6.2. <i>Филтрация след въвеждането на сигнала в компютърната система</i>	13
ОБОБЩЕНИЕ	14

4. ТЕОРЕТИЧНИ ОСНОВИ НА ДИСКРЕТИЗАЦИЯТА И КВАНТУВАНЕТО НА СИГНАЛИТЕ.....	1
ВЪВЕДЕНИЕ	1
4.1. КЛАСИФИКАЦИЯ НА СИГНАЛИТЕ	2
4.1.1. Аналогов сигнал	2
4.1.2. Дискретен сигнал.....	4
4.1.3. Квантуван сигнал	6
4.2. ПРОБЛЕМ ЗА ИЗБОР НА ЧЕСТОТАТА НА ДИСКРЕТИЗАЦИЯ	8
4.2.1. Теорема за дискретизацията.....	8
4.2.2. Апроксимационна формула на Шенон.....	9
4.2.3. Други методи за апроксимация.....	11
4.3. ИЗБОР НА БРОЯ НА НИВАТА ЗА КВАНТУВАНЕ	12
4.3.1. Плътност на вероятност на амплитудата.....	12
4.3.2. Квантуване и плътност на вероятността на амплитудата	13
4.3.3. Теорема за квантуването	14
4.3.4. Оценка на строгостта на изискването за квантуване.....	14
4.4. ИЗТОЧНИЦИ НА ГРЕШКА ПРИ РЕАЛНА ДИСКРЕТИЗАЦИЯ	15
4.4.1. Дискретизация с периодични импулси с произволна форма.....	15
4.4.2. Дискретизация с усредняване	16
4.4.3. Дискретизация на сигнали с крайна продължителност.....	17
ОБОБЩЕНИЕ	18

5. ОСНОВНИ МЕТОДИ ЗА ОБРАБОТКА НА СИГНАЛИТЕ	1
ВЪВЕДЕНИЕ.....	1
5.1 ПРЕОБРАЗУВАНЕ НА ФУРИЕ.....	2
5.1.1. Дефиниция на преобразуването на Фурие.....	2
5.1.2. Физически смисъл на преобразуването на Фурие.....	2
5.1.3. Условия за съществуването на преобразуването на Фурие.....	3
5.1.4. Основни свойства на преобразуването на Фурие.....	3
5.1.5. Фурие-образи на някои функции	4
5.1.6. Дискретно преобразуване на Фурие	6
5.1.7. Алгоритмично реализиране на дискретното преобразуване на Фурие	7
5.1.8. Използване на теглови функции при преобразуването на Фурие.....	8
5.2. ИЗЧИСЛОВАНЕ НА МОЩНОСТ И ЕНЕРГИЯ НА СИГНАЛА	11
5.2.1. Мощност на сигнала.....	11
5.2.2. Енергия на сигнала.....	11
5.2.3. Мощност на сигнала в честотна област	11
5.2.4. Енергия на сигнала в честотна област	12
5.3. КОРЕЛАЦИОННА И АВТОКОРЕЛАЦИОННА ФУНКЦИИ	12
5.3.1. Автокорелационна функция на сигнал	12
5.3.2. Корелационна (взаимнокорелационна) функция на два сигнала.....	13
5.3.3. Физически смисъл и основни приложения.....	14
ОБОБЩЕНИЕ	16

Глава 6. Цифрова филтрация

6. ЦИФРОВА ФИЛТРАЦИЯ	1
ВЪВЕДЕНИЕ	1
6.1. ЦИФРОВ ФИЛТЪР	2
6.1.1. Дефиниция за цифров филтър	2
6.1.2. Уравнение на дискретните системи	2
6.1.3. Рекурсивни и нерекурсивни дискретни системи	3
6.2. ОСНОВНИ ХАРАКТЕРИСТИКИ НА ЦИФРОВИТЕ ФИЛТРИ	3
6.2.1. Времени характеристики	3
6.2.2. Устойчивост и реализуемост	4
6.2.3. Честотни характеристики	5
6.3. ФОРМИ НА ПРЕДСТАВЛЕНИЕ НА ЦИФРОВИТЕ ФИЛТРИ	7
6.4. АНАЛИЗ НА ЦИФРОВИТЕ ФИЛТРИ	8
6.4.1. Теоретичен анализ	8
6.4.2. Симулация и моделиране на цифровия филтър	8
6.5. СИНТЕЗ НА ЦИФРОВИТЕ ФИЛТРИ	9
6.5.1. Входни параметри за синтез на цифров филтър	9
6.5.2. Избор типа на филтъра	10
6.5.3. Апроксимация на желаната амплитудночестотна характеристика	10
6.5.4. Оценка на необходимата точност	11
6.6. РЕАЛИЗАЦИЯ НА ЦИФРОВИТЕ ФИЛТРИ	12
6.6.1. Програмна реализация на цифров филтър	12
6.6.2. Програмна реализация в реално време	13
6.6.3. Апаратна реализация	13
6.6.4. Реализация посредством специализирани интегрални схеми	13
6.6.5. Реализация с процесор за цифрова обработка на сигнали	13
6.7. ПРЕДИМСТВА И НЕДОСТАТЪЦИ НА ЦИФРОВИТЕ ФИЛТРИ	14
6.7.1. Предимства на цифровите филтри	14
6.7.2. Недостатъци на цифровите филтри	14
ОБОБЩЕНИЕ	15

Глава 7. Некласически методи за обработка на сигналите

7. НЕКЛАСИЧЕСКИ МЕТОДИ ЗА ОБРАБОТКА НА СИГНАЛИТЕ.....	1
ВЪВЕДЕНИЕ	1
7.1. СПЕКТРАЛНА ЛУПА	2
7.1.1. Същност на проблема	2
7.1.2. Спектрална лупа с честотно изместване и цифрова филтрация	3
7.1.3. Спектрална лупа при записан сигнал	6
7.2. КЕПСТРАЛЕН АНАЛИЗ.....	8
7.2.1. Дефиниция и терминология	8
7.2.2. Предимства пред автокорелационния анализ.....	9
7.2.3. Предимства пред спектралния анализ.....	9
7.2.4. Приложение на кепстралния анализ	10
7.3. СКАНИРАЩ АНАЛИЗ	11
7.3.1. Същност на сканиращия анализ	11
7.3.2. Представяне на резултатите от сканиращия анализ	11
7.3.3. Практическо приложение на сканиращия анализ	12
ОБОБЩЕНИЕ	13

8. СИНХРОННО ДЕТЕКТИРАНЕ И СИНХРОНИЗИРАН АЦП	1
ВЪВЕДЕНИЕ	1
8.1. МЕТОД НА СИНХРОННОТО ДЕТЕКТИРАНЕ.	1
<i>8.1.1. Област на приложение на метода</i>	<i>1</i>
<i>8.1.2. Блокова схема на метода на синхронното детектиране</i>	<i>2</i>
<i>8.1.3. Теория на метода на синхронното детектиране</i>	<i>3</i>
8.2. СИНХРОНИЗИРАН АЦП.....	5
<i>8.2.1. Теоретични основи</i>	<i>5</i>
<i>8.2.2. Модул синхронизиран АЦП за микрокомпютър</i>	<i>5</i>
8.3. УПРАВЛЕНИЕ И ИЗПОЛЗВАНЕ НА СИНХРОНИЗИРАНИ АЦП	8
<i>8.3.1. Снемане на амплитудночестотни и фазочестотни характеристики</i>	<i>8</i>
<i>8.3.2. Снемане на импулсна характеристика на обект</i>	<i>8</i>
<i>8.3.3. Снемане на преходна характеристика на обект</i>	<i>9</i>
<i>8.3.4. Допълнителни приложения</i>	<i>9</i>
ОБОБЩЕНИЕ	9

10. ОПТИМИЗАЦИОННИ МЕТОДИ В ИНЖЕНЕРНАТА ПРАКТИКА.....	1
ВЪВЕДЕНИЕ	1
10.1. ОБЩИ СВЕДЕНИЯ ЗА ОПТИМИЗАЦИЯТА.....	2
10.1.1. Етапи на оптимизацията.....	2
10.1.2. Избор на критерий за качество.....	3
10.1.3. Видове ограничения	4
10.1.4. Локални и глобален екстремум	4
10.1.5. Постановка на задачата за оптимизация	5
10.1.6. Протичане на процеса на оптимизация.....	5
10.1.7. Условия за край на оптимизационния процес.....	6
10.1.8. Въвеждане на ограниченията посредством наказателни функции	8
10.2. МЕТОДИ ЗА ЕДНОПАРАМЕТРИЧНА ОПТИМИЗАЦИЯ	9
10.2.1. Метод на сканирането	10
10.2.2. Модифициран метод на сканирането	10
10.2.3. Метод на дихотомията	12
10.2.4. Метод на златното сечение	13
10.2.5. Метод на Брент	14
10.3. МЕТОДИ ЗА МНОГОПАРАМЕТРИЧНА ОПТИМИЗАЦИЯ.....	14
10.3.1. Метод на Гаус-Зайдел	15
10.3.2. Мрежово търсене на Мишке	16
10.3.3. Градиентни методи.....	16
10.3.4. Конфигурационен метод на Хук и Джийвс	19
10.3.5. Симплекс метод	21
10.4. ПРИМЕРИ ЗА ИЗПОЛЗВАНЕТО НА ОПТИМИЗАЦИОННИТЕ МЕТОДИ .. Error! Bookmark not defined.	
10.4.1. Подобряване на точността на измерителна система	19
10.4.2. Изчисляване на малък антенен филтър.....	23
ОБОБЩЕНИЕ.....	28

Глава 11. Управление и комуникация в системите за събиране и обработка на информация

11. УПРАВЛЕНИЕ И КОМУНИКАЦИИ В СИСТЕМИТЕ ЗА СЪБИРАНЕ И ОБРАБОТКА НА ИНФОРМАЦИЯ.....	1
ВЪВЕДЕНИЕ.....	1
11.1. ОПЕРАЦИОННИ СИСТЕМИ ЗА РЕАЛНО ВРЕМЕ.....	2
11.1.1. Дефиниране.....	2
11.1.2. Основни понятия в операционните системи за реално време.....	3
11.1.3. Атрибути на заданията.....	4
11.1.4. Граф на състоянията на заданията.....	5
11.1.5. Функции и реализация.....	9
11.2. СТРУКТУРИ ОТ ДАННИ ЗА КОМУНИКАЦИЯ МЕЖДУ ЗАДАНИЯТА.....	10
11.2.1. Флаг.....	11
11.2.2. Опашка.....	11
11.3. ЛОКАЛНИ КОМПЮТЪРНИ МРЕЖИ.....	14
11.3.1. Общи сведения за локалните мрежи.....	14
11.3.2. Локална мрежа за индустриално приложение.....	21
ОБОБЩЕНИЕ.....	31

Глава 12. Системи за управление, събиране и обработка на данни в промишлеността

12. СИСТЕМИ ЗА УПРАВЛЕНИЕ, СЪБИРАНЕ И ОБРАБОТКА НА ДАННИ В ПРОМИШЛЕНОСТТА.....	12-1
ВЪВЕДЕНИЕ.....	12-1
12.1. ПРЕДНАЗНАЧЕНИЕ И СПЕЦИФИЧНИ ОСОБЕНОСТИ.....	12-2
12.1.1. <i>Предназначение</i>	12-2
12.1.2. <i>Специфични особености на индустриалните системи</i>	12-2
12.2. БЛОКОВА СХЕМА НА СИСТЕМА ЗА СЪБИРАНЕ И ОБРАБОТКА НА ИНФОРМАЦИЯ ЗА ЦЕЛИТЕ НА УПРАВЛЕНИЕТО НА ПРОМИШЛЕНИ ОБЕКТИ.....	12-5
12.2.1. <i>Промислени контролери</i>	12-6
12.2.2. <i>Концентратори</i>	12-8
12.2.3. <i>Централен компютър</i>	12-9
12.3. ОБОБЩЕН МОДЕЛ НА ПРОМИШЛЕН КОНТРОЛЕР.....	12-12
12.3.1. <i>Захранване на промишления контролер</i>	12-12
12.3.2. <i>Аналогови входове</i>	12-12
12.3.3. <i>Цифрови входове</i>	12-13
12.3.4. <i>Аналогови изходи</i>	12-14
12.3.5. <i>Цифрови изходи</i>	12-14
12.3.6. <i>Пулт за управление</i>	12-15
12.3.7. <i>Връзка с по-горното ниво</i>	12-15
12.4. РЕШАВАНИ ЗАДАЧИ ОТ ПРОМИШЛЕНИТЕ КОНТРОЛЕРИ.....	12-15
12.4.1. <i>Промислени контролери в непрекъснатите производства</i>	12-15
12.4.2. <i>Промислени контролери в дискретните производства</i>	12-20
ОБОБЩЕНИЕ.....	12-22

1. Обща структура и предназначение на системите за събиране и преобразуване на информация



Ключови

думи

- системи за събиране и обработка на информация
- информационни потоци
- решавани задачи
- проблеми при проектирането

Какви знания и умения ще придобиете?

Къде се прилагат?

В тази тема ще се запознаете с предназначението на системите за събиране и обработка на информация, с тяхната обобщена блокова схема, с информационните потоци в тях, с решаваните от тези системи задачи и с основните проблеми при проектирането на такива системи.

Получените знания ще намерят приложение в първият етап от проектирането на всяка система за събиране и обработка на информация — изграждането на нейната обща структура.

Въведение

Настоящата тема е посветена на системотехниката, архитектурата на изграждане на системите за събиране и обработка на информация. Разгледани са предназначението и основните принципи при изграждане на тези системи. Обърнато е специално внимание на проблемите при проектирането на системите за събиране и обработка на информация.

1.1. Предназначение на системите за събиране и обработка на информация

1.1.1. Предназначение

Системите за събиране и обработка на информация са предназначени за изследване, изучаване на явления и процеси, протичащи във външна за тях среда и въздействие върху тях. Като правило “сърцето” на една такава система е компютър. Тези системи са изключително многообразни и могат да варират от една малка микропроцесорна система за управление на битов домакински уред до сложен изчислителен комплекс за управление на производството в цял цех.

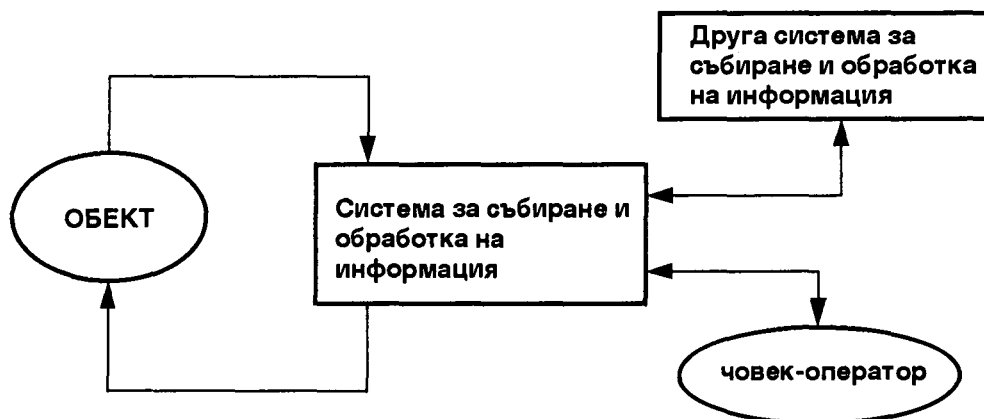


Независимо от тяхното многообразие взаимодействат с околната среда по сходен начин и в тях протичат едни и същи процеси. Разликата е в сложността на алгоритмите за обработка и многообразието на входните и изходните величини на системата.

1.1.2. Взаимодействие

В изпълнение на поставените задачи системата за събиране и обработка на информация взаимодейства с външни за нея страни, както е показано на фиг. 1.1:

а) *обект* — от когото се събира информация и върху когото се въздейства. Това може да бъде пещ за изпичане на хляб, управляема ракета или дори човек — когато системата за събиране и обработка на данни е с медицинско предназначение;



Фиг. 1.1. Взаимодействие на система за събиране и обработка на информация.

б) *човек-оператор* — за него са предназначени резултатите от работата на системата, той ги получава и може да подава команди с които да влияе на процеса на събирането и обработката;

в) *друга система за събиране и обработка на информация*, когато няколко такива системи са свързани заедно за постигане на по-сложни цели и задачи.

1.1.3. Процеси

Най-общо казано системата за събиране и обработка на информация възприема сигнали, данни от обекта, извършва тяхната обработка, въздейства обратно върху обекта и извежда резултатите. В процеса на работата на системата едновременно протичат три процеса:

а) *Възприемане* — на сигнали, данни, измерване на електрически и неелектрически величини посредством система от датчици, получаване на команди от оператора посредством клавиатура или друго устройство. Като част от този процес е и довеждането на измерените величини до техните физически стойности. Например, при измерване на температура посредством термодвойка да се получи не напрежението на двата края на термодвойката, а температурата в градуси.

б) *Обработка* — в зависимост от поставените пред системата за събиране и обработка на данни цели и задачи, получените в първата фаза резултати от възприемането и запомнени предварително данни се извършва обработка, най-често на няколко етапа. Получените резултати се съхраняват с цел по-нататъшно използване и/или извеждане.

в) *Извеждане* — на получените резултати от обработката, като част от този процес е и преобразуването на физическата стойност, например от необходимите обороти в минута на електродвигател до непосредствената команда, която трябва да се подаде към изпълнителното устройство. Извеждането на резултатите може да става в няколко посоки:

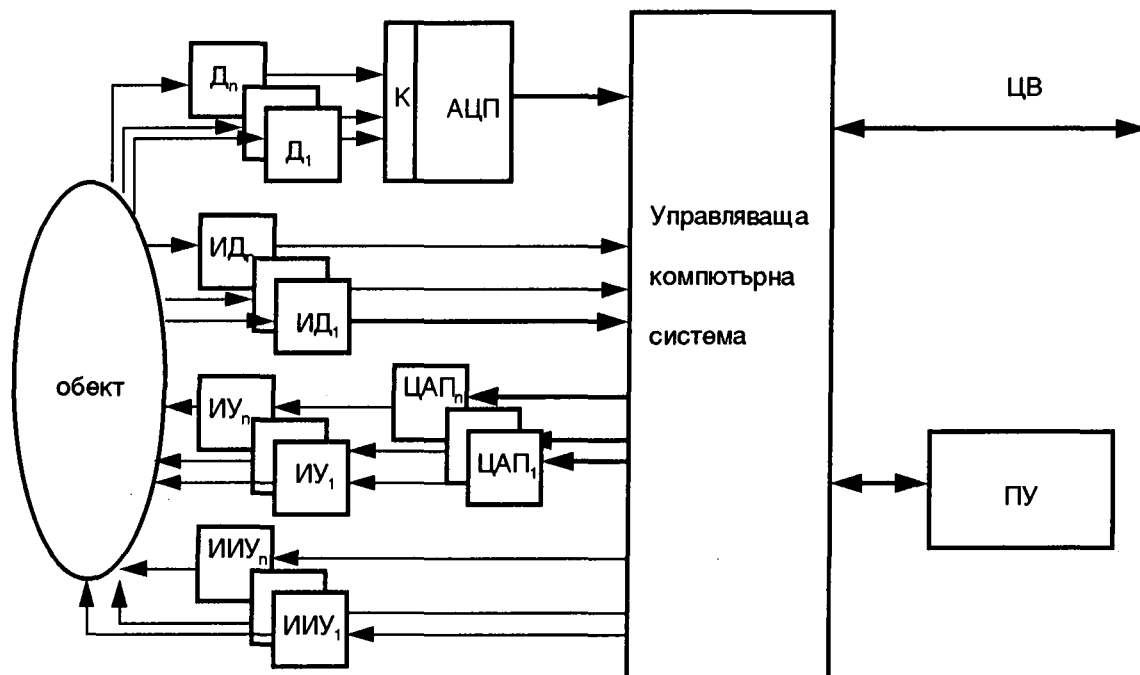
- към обекта, от който са въведени данни, сигнали и др. при което върху него си извършва въздействие посредством изпълнителни механизми;
- към човека, за сведение, документиране, обработка, получаване на следващи команди;
- към по-високото стъпало в йерархията на голяма система за събиране и обработка на данни, ако нашата система е част от такава.

1.2. Обща блокова схема на системите за събиране и обработка на информация

Обобщена блокова схема на система за събиране и обработка на информация е показана на фиг. 1.2. Основните елементи са:

- D_1, D_2, \dots, D_n — датчици с аналогов изход;
- ID_1, ID_2, \dots, ID_n — датчици с импулсен или потенциален изход;
- K — аналогов мултиплексор;
- АЦП — аналогово-цифров преобразувател;
- управляващ компютър — от проста микропроцесорна система до сложен многопроцесорен управляващ комплекс;
- ЦАП₁, ЦАП₂, . . . , ЦАП_n — цифрово-аналогови преобразуватели;
- ИУ₁, ИУ₂, . . . , ИУ_n — изпълнителни устройства с аналогов изход;

- ИИУ₁, ИИУ₂, . . . , ИИУ_n — изпълнителни устройства с импулсен или потенциален изход;
- ПУ — операторски пулт;
- ЦВ — цифрова връзка с друга система за събиране и обработка на информация.



Фиг. 1.2. Обща блокова схема на система за събиране и обработка на информация.

Така изградената блокова схема на система за събиране и обработка на информация позволява взаимодействието с обекта и с човека-оператор и протичането на трите процеса.



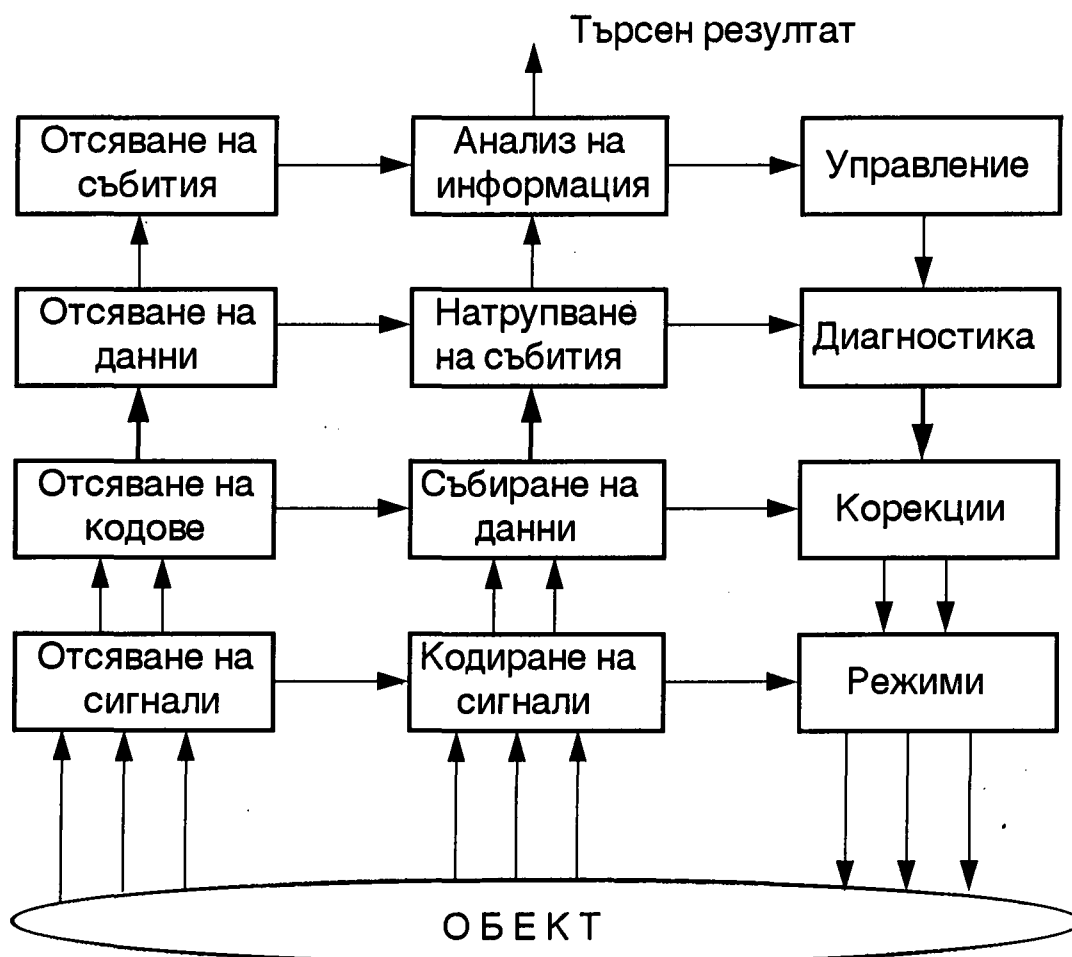
Не е задължително всички елементи от блоковата схема да бъдат представени в конкретната система за събиране и обработка на информация. Например, в един компютърен електрокардиограф, предназначен за записване на биоелектричната активност на човешкото

сърце, липсват всички елементи, предназначени за въздействие върху обекта, в случая пациента.

1.3. Информационни потоци в системите за събиране и обработка на информация и тяхната интензивност

Блоковата схема на информационните потоци в системите за събиране и обработка на информация е показана на фиг. 1.3. Информацията се движи в три колони и преминава през четири нива.

Първата колона е поток от информация спомагащ вземането на решения за обработка и отсяване, като този поток от информация е с посока отдолу нагоре;



Фиг. 1.3. Информационни потоци в системите за събиране и обработка на информация.

Втората колона е потока от текуща информация, снета от обекта, непрекъснато обработвана, и е с посока също от долу нагоре;

Третата колона е потока от управляваща информация, посоката му е отгоре надолу.

Първото ниво е апаратурата за възприемане и преобразуване на информацията в електрически сигнали и цифрови кодове — датчици, нормализация нивата на сигналите, АЦП, компаратори, тригери, аналогови филтри.

Второто ниво са апаратно-програмни средства за отсяване на кодове — функционални преобразуватели, времеви врати, цифрови компаратори, цифрови филтри.

Третото ниво на обработка включва отсяването на данните, натрупване на събития. Като правило се реализира по програмен начин от управляващата компютърна система.

Четвъртото ниво включва обобщаването и изобразяването на натрупаната информация, също като правило реализирано по програмен път в управляващия компютър.



Основното правило при изграждането на системите за събиране и обработка на информация е така да се разпределят задачите между апаратната и програмната част и между фазите на обработка, че интензивността на трите информационни потока да е с един порядък по-малка спрямо по-голямото ниво.

1.4. Задачи, решавани от системите за събиране и обработка на информация

На всяко от четирите нива се решават специфични задачи. Потокът от информация при преминаване в следващото по-горно ниво не само намалява количествено, но и претърпява качествени изменения, изразява се с други понятия.

- ⇒ **Първо ниво** — преобразуване на входната информация в електрически сигнали, нормализиране на амплитудата им, премахване на смущаващи въздействия посредством аналогова филтрация и преобразуването им в цифров код, т.е. извършване на процеса на дискретизация.
- ⇒ **Второ ниво** — допълнително отсяване на постъпилите кодове посредством времева, амплитудна и цифрова филтрация, функционално преобразуване с цел премахване на нелинейността на датчиците посредством апроксимация на техните характеристики. На това ниво процесът на въвеждане и първична обработка на информацията приключва с получаване на физическите стойности на въвежданите величини.
- ⇒ **Трето ниво** — алгоритми за управление и вземане на решение за въздействие върху обекта. Това е нивото на програмните регулатори и по-сложните алгоритми за управление. На това ниво се извършва и статистическа обработка на измерваните величини, изчисляване на средноквадратично отклонение, отхвърляне на измерванията вън от 2.5 или 3 пъти средноквадратичното отклонение и т.н. Тук се извършват и по-сложните обработки на входните сигнали — спектри, корелации, геометрични преобразувания, намиране на характерни точки и др.
- ⇒ **Четвърто ниво** — статистически обработки за по-дълъг период от време, оптимизационни процедури, идентификация на обекта за управление, адаптивност на системата към променящи се външни условия и параметри на обекта, прогнозиране хода на процеса, представяне на данните за оператора, възприемане на командите му и др.

1.5. Основни проблеми при проектиране на системите за събиране и обработка на информация

Проектирането и разработването на системи за събиране и обработка на информация е сложен, многокритериален и итерационен процес, обикновено решаван от колективи.

- ⇒ *Сложен*, защото обхваща широк спектър от инженерните науки — от датчиковите системи през първичната обработка до алгоритмите за управление и идентификация на обектите и за обработка на сигналите.
- ⇒ *Многокритериален*, поради разнопосочността и противоречивостта на изискванията към разработваната система — цена, бърздействие, габарити, консумация, функции и др.
- ⇒ *Итерационен*, защото като правило добрата система за събиране и обработка на информация се получава след разработването на няколко системи със същото или със сходно предназначение с отчитане на показаните недостатъци от предишните разработки. Последното важи като се започне от ниво печатна платка и се завърши с ниво цяла система.

1.5.1. Проблеми при избор на датчици и изпълнителни механизми

В началния стадий на проектирането е известно какви величини трябва да се възприемат от обекта. Тук проблемът е в избора на необходимите за това датчици и техния принцип на работа. Трябва да се направят разчети и да се определи необходимата точност на отчитане на съответната величина, а оттам и точността на датчиците. Много често една и съща фирма предлага цяла гама от датчици за една и съща величина (например температура, налягане и др.), които съществено се различават в цената си в зависимост от постиганата точност. От разстоянието между компютърната система и датчика се избира и начина за въвеждане на тази величина — от просто прекарване на свързващ кабел между датчика и входа на аналогово-цифровия преобразувател до разработване на малка микропроцесорна система специално за възприемане и първична обработка, като връзката със системата за събиране и обработка на информация се осъществява по цифров път с използването на някой от стандартните интерфейси (например RS232) или с помощта на специализиран интерфейс. Задъл-

жително се отчитат и условията при които ще работят датчиците и изпълнителните механизми — температурен интервал, смущения, вибрации, влага, разяждащи пари и разтвори и др. Температурата в стаята можем да измерим и с обикновен термистор и преобразувател съпротивление-напрежение, но за да измерим температурата на концентриран разтвор на киселина е необходимо да се намери датчик за температура, конструктивно изпълнен да издържи на тези условия.

1.5.2. Проблеми при избор на управляващата компютърна система

Преди да се започне избора или разработката на управляващата компютърна система е необходимо да се дефинират конкретните задачи, стоящи пред нея, на всяко от изброените по-горе нива. Трябва да се направи баланс на точността и от тук вътрешното представяне на входните и изходните величини. Когато искаме да обработваме масив от 1024 числа и е достатъчна точността на цяло дгубайтово число, необходимият обем памет е 2 К, но ако е необходимо да работим с осембайтови числа в плаваща запетая необходимият обем памет вече е 8 К. Балансът на точността пряко влияе и върху баланса на времето в компютърната система. Времето за умножение на две дгубайтови цели числа е много пъти по-малко от времето за умножение на две осембайтови числа в плаваща запетая. След уточняване на необходимите памет и бързодействие на управляващата компютърна система се извършва избора на нейната архитектура — дали ще изберем еднопроцесорна система с достатъчно мощен микропроцесор или многопроцесорна разпределена структура зависи от конкретната задача. Многопроцесорните системи биват *слабосвързани* и *силносвързани*. Слабосвързаните архитектури се състоят от компютърни системи, всяка работеща самостоятелно и решаваща нейната част от задачите на цялата система. Връзката между системите се осъществява посредством локална мрежа, стандартни (RS232, IEEE488) или специално разработени интерфейси или локална мрежа. При силносвързаните архитекту-

ри средата за комуникация е локалната шина на процесорите с поделение на общите ресурси — памет и периферия.

1.5.3. Проблеми с комуникацията с човека оператор.

След анализ на данните, които трябва да получава и въвежда оператора се извършва и избор на необходимата периферия за връзка с оператора. Това може да са няколко светодиода и бутон, но може да е дисплей с пълна клавиатура и мишка. Трябва сериозно внимание да се обърне и на средата в която ще се работи тази периферия — не е допустимо да се използва мишка в силно запрашена среда, защото много скоро тя просто ще откаже. Тук се отчитат и други фактори — например необходимостта от честа дезинфекция за електромедицинската апаратура, което изключва употребата на стандартните клавиатури за компютър. При всеки случай стремежът е пренасяне на комуникацията с човека оператор в място с нормални температури и замърсеност. Загължително се отчитат както психо-физиологичните особености на оператора (способността му да възприема информация, да задържа продължително време вниманието си, скоростта на преценка на обстановката и времето му за реакция), така и степента на неговата подготовка (работник, техник, инженер).

1.5.4. Проблеми с надеждността на системата

Системата за събиране и обработка на информация трябва да работи нормално и да реализира зададените изисквания (точност, бързодействие и др.) в предвидените в техническото задание условия — температурен интервал, вредни въздействия, вибрации, удари, електрически смущения и др. В зависимост от предназначението на системата, обаче, трябва да се гарантира и отказоустойчивостта ѝ. Това е способността да работи определено време без повреда и се постига с подходящи схемотехнични и конструктивни мерки, например защити на входните вериги от пренапрежение и импулсни смущения, кутии и касети за електронната апаратура които я предпазват от вредните въздействия. Внимателно трябва да се анализират и ситуацията на

отказ на един или няколко датчика или изпълнителни механизми с цел цялостно или поне частично запазване на работоспособността на системата в дадената аварийна ситуация — надеждност на системата.

1.5.5. Проблеми при изграждане на захранващия блок

Осигуряването на всички необходими захранващи напрежения в една система за събиране и преобразуване на информация е проблем често подценяван от разработчиците на такива системи. Получаването на няколкото захранващи напрежения, необходими за работата на системата изглежда на пръв поглед не особено сложен проблем. Трудностите тук са в няколко направления:

а) *По отношение на енергийния източник.* Много често в променливотоковата захранваща мрежа 220 V има импулсни смущения и кратковременно (до няколко секунди) отпадане — особено характерно за условията в индустрията. Това води до предвиждане на акумулаторно резервиране на захранването със съответните вериги за дозареждане на акумулатора.

б) *По отношение на запазването на данни при изключване на системата.* За запазване на натрупаните данни и резултати от обработката се налага използването на EEPROM памет, а когато обемът на данните е голям — батерийно или акумулаторно резервиране на цялата или на част от оперативната памет. Апаратно това изисква предвиждането на батерия или малък акумулатор (отново вериги за дозареждане!) и необходимите вериги за предотвратяване на записа в тази памет при преходните процеси по време на включване или изключване на устройството. Програмно трябва да се контролира целостта на данните посредством контролни суми и други подобни мерки.

в) *По отношение на осигуряване захранването на галванично отделените датчици.* Особено тежък е този проблем при електромедицинската апаратура. Входните усилватели на биологичните сигнали, които се снемат посредством свързани с пациента електроди, трябва да бъдат захранени съгласно изключително тежките изисквания на стандартите за електромедицинска апаратура. Тези изисквания включват нормиране

на изолационното съпротивление и пробивното напрежение между захранващата променливотокова мрежа и електроудите, прикрепени към пациента. Галванично отделени захранващи напрежения се подават и към някои от датчиците в индустриалните системи за събиране и обработка на информация.

г) *По отношение на разсейваната от захранващия блок топлина.* Захранващият блок отделя топлина в затворената кутия или касета на система за събиране и обработка на информация. Това създава проблеми с отделянето на тази топлина. Поставянето на вентилатор е възможно само при системи, работещи в относително чисто помещение. В противен случай преминаващия през касетата въздух оставя съдържащия се в него прах по радиаторите, интегралните схеми и останалите градивни елементи. Това довежда до затруднено охлаждане и дефектиране. Решението е в използването на високоефективни ключови захранващи блокове.

Обобщение

Системите за събиране и обработка на информация са предназначени за изследване, изучаване на явления и процеси, протичащи във външна за тях среда и въздействие върху тях. В процеса на своята работа те си взаимодействат с обекта за изучаване или управление, с човека-оператор и с други системи за събиране и обработка на информация.

Системите за събиране и обработка на информация се състоят от датчици, изпълнителни устройства, управляваща компютърна система, средства за комуникация с човека-оператор и интерфейс към други системи за събиране и обработка на информация.

Основното правило при изграждането на системите за събиране и обработка на информация е така да се разпределят задачите между апаратната и програмната част и между фазите на обработка, че интензивността на трите информационни потока да е с един порядък по-малка във всяко следващо ниво.

Задачите решавани от системите за събиране и обработка на информация са разпределени на четири нива:

- преобразуване на входната информация в цифров код — дискретизация;
- филтрация и функционално преобразуване с цел получаване на физическите стойности на въвежданите величини;
- алгоритми за управление и вземане на решение за въздействие върху обекта;
- статистически обработки за по-дълъг период от време, адаптивност на системата, представяне на данните за оператора и възприемане на командите от него.

Проектирането и разработването на системи за събиране и обработка на информация сложен, многокритериален и итерационен процес, обикновено решаван от колективи. Основните проблеми при разработката на системите за събиране и преобразуване на информация са свързани с:

- избора на датчици и изпълнителни механизми;
- определянето на архитектурата на управляващата компютърна система;
- комуникацията с човека-оператор;
- надеждността на системата;
- изграждане на захранващия блок.



Контролни въпроси

1. Какво е предназначението на системите за събиране и обработка на информация и къде се използват? Дайте примери!
2. С какво взаимодействат системите за събиране и обработка на информация?
3. Кои са съставните елементи на системите за събиране и обработка на информация?
4. Кои са основните задачи, решавани от системите за събиране и обработка на информация?

Глава 1. Обща структура и предназначение на системите за събиране и преобразуване на информация

5. Кои са основните групи проблеми при проектирането на системите за събиране и обработка на информация?

2. Въвеждане и извеждане на сигнали от компютърната система



Ключови думи

- аналогов сигнал
- импулсен сигнал
- АЦП
- ЦАП
- галванично раз-
вързване

Какви знания и умения ще придобиете?

Къде се прилагат?

В тази тема ще се запознаете с типовите решения и основните проблеми при въвеждането и извеждането на информация от компютърната система. Тази дейност е изключително важна за успешното решаване на проблемите при създаването на системи за събиране и преобразуване на информация.

Получените знания ще приложите при проектирането на входните и изходните вериги на системите за събиране и преобразуване на информация.

Въведение

Определящо значение за работата на системите за събиране и обработка на информация имат началният и крайният етап от взаимодействието с обекта: възприемане на информация от обекта и извеждане на информация, т.е. въздействие върху него. В настоящата тема са разгледани двете основни групи сигнали — импулсно-потенциални и аналогови. Дадени са типовите схемни решения на входните и изходните блокове на системите за събиране и преобразуване на информация. Входните сигнали се разглеждат след преобразуването им в електрическа величина от входните датчици. Изходните сигнали също се разглеждат докато са още електрически величини, т.е. до входа на изпълнителните устройства.

Предназначението на входните и изходните вериги на системата за събиране и преобразуване на информация е:

- да превърнат входните сигнали в разбираем за компютъра вид — на логически нули и единици, или обратно, от логически нули и единици в сигнали разбираеми за изпълнителните устройства.
- да осигурят защита от шумове и смущаващи сигнали;

- да осигурят защитата на системата от външни пренапрежения, токови удари, неправилно включване.

2.1. Въвеждане на потенциални и импулсни сигнали

2.1.1. Дефиниране

Потенциалните и импулсните сигнали се характеризират с две състояния — включено и изключено, затворено и отворено, има и няма. Разликата между тях е условна и се определя от честотата на промяна на състоянието им. Потенциалните сигнали се изменят сравнително рядко и управляващият компютър ги следи обикновено по програмен път. Импулсните сигнали изменят състоянието си по-често и за възприемането им компютърната система се нуждае от допълнителна апаратна част — тригери и броячи.

Освен с честотата на промяна на състоянието си потенциалните и импулсните сигнали се характеризират още:

- с начина по който ще се задават двете състояния — с напрежение, ток, съпротивление;
- с нивата на двете състояния.



Примери за потенциални сигнали:

- при алармена система за автомобил — изхода на датчика дали е отворен капак на багажника на автомобила;
- при компютърна система за управление на автоматична пералня — изходите на датчиците за достигнати температури 30°C, 60°C, 90°C на водата в казана.

Примери за импулсни сигнали:

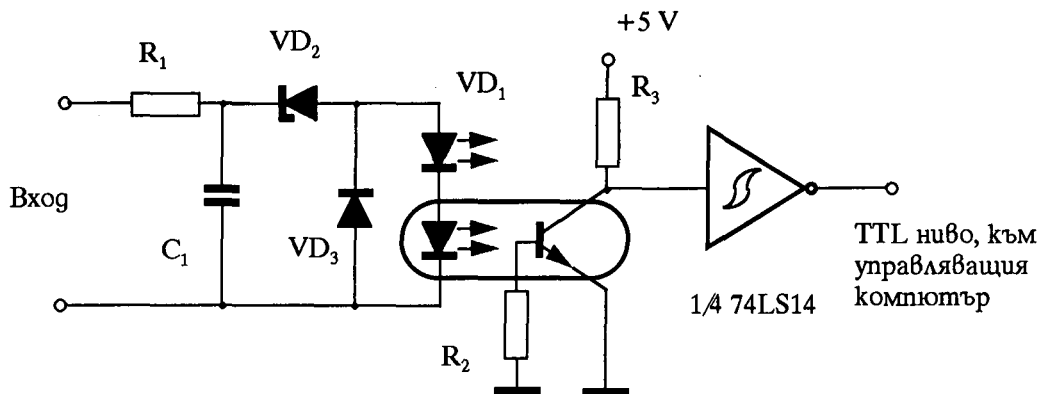
- при компютърна система за управление на видеокасетофон — броя на възпроизведените или записаните полукадри за управление на индикацията (брояча);
- при система за управление оборотите на постояннотоков електродвигател — импулсите на изхода на тахогенератора.

Примери за различни нива на двете състояния:

- при инфрачервен датчик за присъствие за алармена система в сграда — изходно напрежение 0 или 12 V;
- при контактен датчик за крайно положение — съпротивление, затворен или отворен контакт.

2.1.2. Типична схемотехника на входните вериги, галванично развързване

Един от най-сигурните методи за предпазване на входните вериги на системата за събиране и обработка на информация е галваничното им разделяне от датчиците.



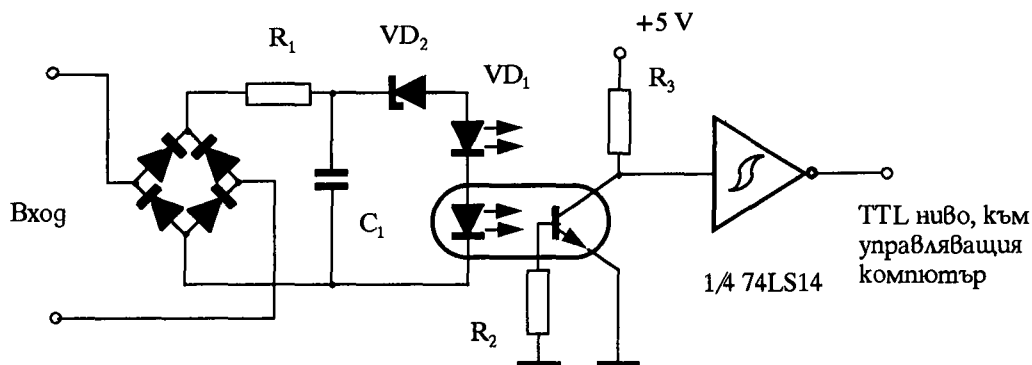
Фиг. 2.1. Типова схема на входна верига за постоянно напрежение

Типична схема на входна верига за постоянно напрежение или ток е показана на фиг. 2.1. Резисторът R_1 ограничава тока през светодиода на оптрона и същевременно, заедно с кондензатора C_1 , образува един Г-образен RC филтър против импулсни смущения. Светодиодът VD_1 служи за индициране състоянието на входа и светва заедно със светодиода на оптрона. Това се получава когато входното напрежение превиши пробивното напрежение на центровия диод VD_2 . Диодът VD_3 предпазва схемата при обратно подаване на входното напрежение. Резисторът R_2 стабилизира работата на фототранзистора в широк температурен интервал. След товарния резистор в колектора на фототранзистора на оптрона е включен тригер на Шмит, който служи да формира стръмни фронтове на изходния сигнал. Хистерезисът на този тригер предпазва и от многократно превключване в рамките на полегатия фронт на входния сигнал след RC филтъра. Резисторът R_1 трябва да гарантира такъв ток на светодиода, че превключването на изхода на тригера на Шмит да става по средата между двете състояния.

Ако входното напрежение е променливо се използва схемата, показана на фиг. 2.2. Тя работи по същия начин и се различава от предишната по наличието на двуполупериоден изправител (схема Греци) на входа и по липсата на диода VD_3 .

Когато не е необходимо галванично развързване схемотехниката на входните вериги на системите за събиране и преобразуване на информация е практически същата, като липсват оптроните.

При необходимост от възприемане на затварянето на контакти (релета, крайни изключватели, отдалечени бутони и др.), при които и двата проводника са изолирани от корпуса, нулата на мрежата и земята, входните вериги се опростяват и са съвсем тривиални.



Фиг. 2.2. Типова схема на входна верига за променливо напрежение

2.1.3. Защита от импулсни смущения

Системата за събиране и обработка на информация трябва правилно да възприема импулсните и потенциалните сигнали и в условията на смущаващи въздействия — индуктирани в свързващите датчиците и компютърната система кабели напрежителни импулси, породени от включването или изключването на мощни консуматори в хранващата мрежа. Трябва да се има пред вид, че в промишлени условия тези смущаващи въздействия може да са много силни и дори съизмерими с нивата на кодиране на двете състояния. Даже и в битови условия амплитудите на смущаващите въздействия не са за подценяване. Защитата от смущения във входните вериги за импулсни сигнали се извършва по апаратен път — с поставянето на подходящи RC или LC филтри. При потенциалните сигнали освен апаратните мерки се прилагат и програмни (алгоритмични) мерки за защита.

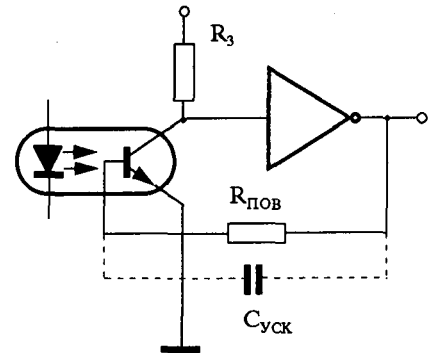
2.1.3.1. Апаратни мерки за защита от импулсни смущения

В разгледаните вече типови входни вериги има поставен RC филтър против импулсни смущения, който в комбинация с тригера на Шмит в повечето случаи предотвратява превръщането на смущаващите сигнала

ли в лъжливи преходи на сигнала с TTL ниво. Капацитетът на филтриращия кондензатор C_1 се ^{определя} ~~разчита~~ така, че RC филтърът да пропуска импулсните сигнали с най-високата работна честота и да спира импулсните смущения.

Съществена роля при наличието на импулсни смущения играе *ценера-вият диод* VD_2 . Той се отпущва и дава възможност да протече ток през светодиода на оптрона едва когато входното напрежение превиши пробивното му напрежение. То се избира около половината от активното ниво на входното напрежение.

При голяма амплитуда на смущаващите сигнали понякога е необходимо *да се увеличи хистерезиса на тригера на Шмит*. За целта се въвежда допълнителна положителна обратна връзка, както е показано на фиг. 2.3. При полегати фронтове на входните сигнали може да се постави ускоряващ кондензатор, както е показано с прекъснатата линия.

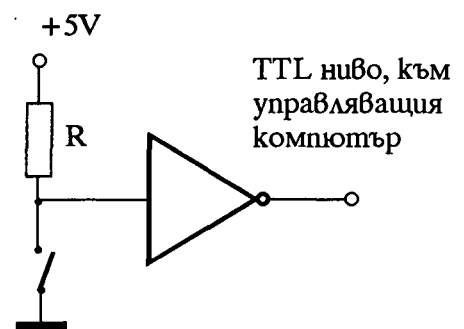


Фиг. 2. 3.

2.1.3.2. Програмни мерки за защита от импулсни смущения

След превръщането на входните потенциални сигнали в TTL ниво те се подават на входа на някой от портите на управляващата микропроцесорна система. Като правило по програмен път се вземат допълнителни мерки за защита от фалшиво възприемане.

На фиг. 2.4. е показана схема за въвеждане в компютърната система на състоянието на контакт — бутон, датчик и др. По-неже това е потенциален сигнал не са взети никакви апаратни мерки за подтискане на импулсни смущения. Особено при въвеждане състоянието на контакти е наличието на преходен процес при затваряне и отваряне на контакта поради трептенето на контактните пластини.



Фиг. 2.4.

Борбата с преходния процес и импулсните смущения се извършва по програмен път с въвеждането на така нареченото мъртво време. След констатиране на смяната на състоянието на контакта (например на-

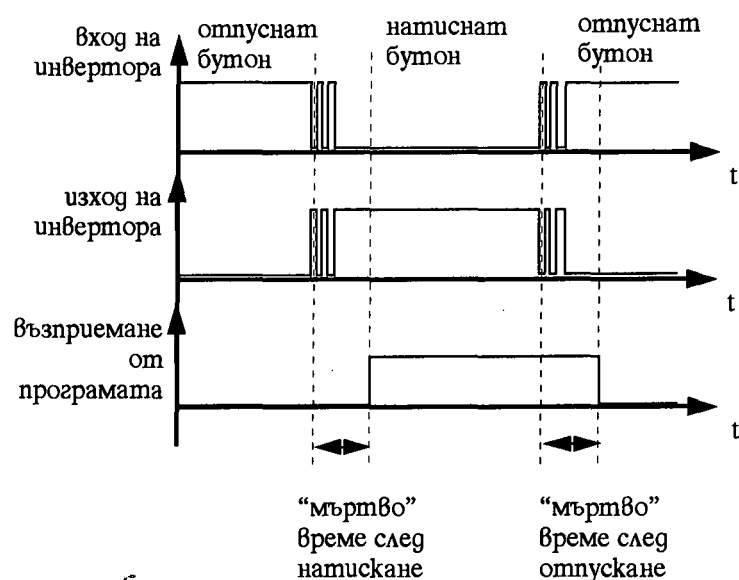
тискане на бутона) известно време не се следи неговото положение (мъртво време), както е показано на фиг. 2.5.

Ако след изтичане на мъртвото време състоянието продължава да е сменено (бутонът натиснат) значи това не е импулсно смущение, а истинска смяна на състоянието. В противен случай се касае за смущение. Продължителността на мъртвите времена зависи от типа на контакта и е от порядъка на няколко десетки милисекунди.

2.1.3.3. Допълнителни мерки за защита против импулсни смущения

Допълнителни мерки за защита против случайни импулсни смущения се вземат както на апаратно, така и на алгоритмично ниво.

Една от мерките е така наречената *времева врата*. Това означава промяната на състоянието на всеки вход да се следи само в точно определен отрязък от време — когато се очаква да се промени в зависимост от конкретната задача. През останалото време промяната на състоянието на този вход просто не се възприема — блокирана е апаратно или не се следи програмно.



Фиг. 2.6. Мъртво време при възприемане на потенциални сигнали

Друга възможна мярка е *повишаване на броя на проверките*, най-често програмните, свързани с времето между две промени на дадения вход или с комбинацията на няколко входни сигнала. Ясно е, че не е възможно датчиците за двете крайни положения на работен орган да бъдат затворени едновременно, или за части от секундата работния орган да се придвижи от едното положение в другото. Не е възможно, също така, да е сработил датчика в казана на една автоматична пералня за температурата на водата над 60°C , а да не е сработил датчика за температура над 30°C . В трите случая се касае или за импулсно смущение или за отказ на съответния датчик. Въвеждането на тези допълнителни проверки повишава възможността на системата за събира-

не и обработка на информация за откриване на такива повреди и за адекватна реакция — продължаване на работа доколкото е възможно без да се допускат неправилни действия, т.е. повишава се надеждността на системата.

2.2. Въвеждане на аналогови сигнали

2.2.1. Дефиниране

Аналоговите входни сигнали изменят плавно, непрекъснато, своята стойност между две величини. Физическите входни величини са преобразувани от датчиците в електрически, най-често в напрежение. За да бъде въведен и обработван в компютърната система аналоговият сигнал трябва да бъде преобразуван в цифров код — число, съответстващо на големината му. Тази процедура се извършва от аналогово-цифровия преобразувател — АЦП.



Примери за аналогови сигнали:

- при автоматичен регулатор на температура в пещ — изхода на датчика за температура;
- при електрокардиограф — електрическото напрежение, породено от биоелектричната активност на сърцето.

2.2.2. Класическа блокова схема на многоканален АЦП

2.2.2.1. Типова блокова схема

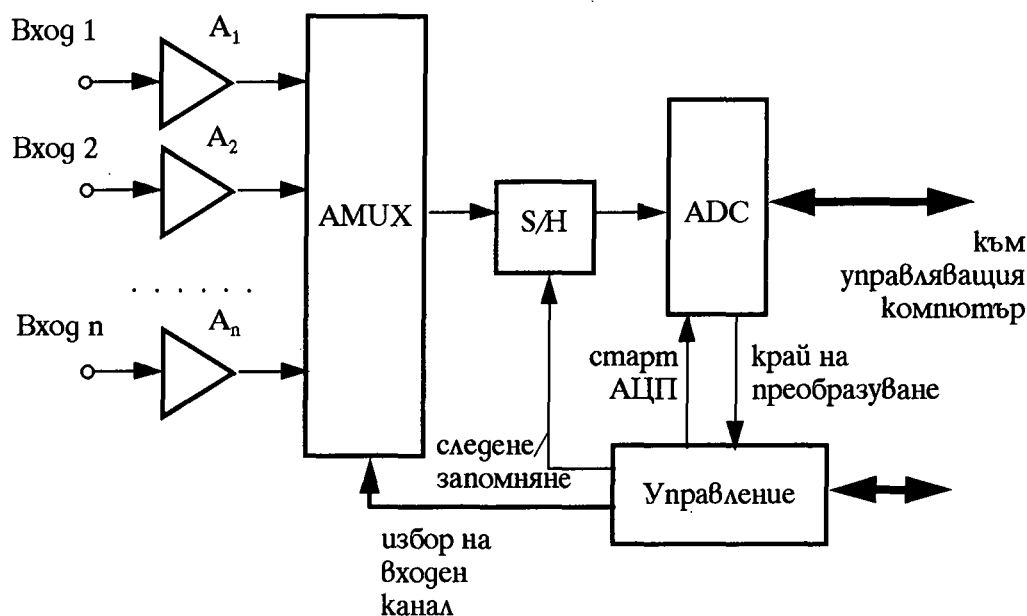
Системите за събиране и преобразуване на информация обикновено трябва ^{да} възприемат няколко аналогови сигнала. Класическата блокова схема на многоканален АЦП е показана на фиг. 2.7. Аналоговите сигнали се буферират и усилват при нужда от операционните усилватели A_1 до A_n и се подават на входовете на аналоговия мултиплексор АМУХ. Най-често използваната интегрална схема за аналогов мултиплексор е осем каналният аналогов мултиплексор МUX08. Избраният от блока “управление” входен сигнал излиза на изхода на мултиплексора и се подава на входа на схемата следене/запомняне — S/H. Типично тук се използва схемата следене/запомняне LF398. След запомняне на моментната стойност на сигнала по команда от управляващия блок се стартира аналогово-цифровия преобразувател ADC. Когато изтече времето за преобразуване на изхода на АЦП се появява числото, съответстващо

на амплитудата на входния сигнал. За АЦП най-често се използва монолитен чип, работещ по метода на поразрядното кодиране — AD574, българският CM757 и др.

2.2.2.2. Протичане на едно измерване

Измерването на една моментна стойност на входния сигнал протича в следния ред:

- ◆ Избира се необходимият входен канал и се изчаква времето за превключване на аналоговия мултиплексор и установяването на изхода му. Това време е от порядъка на няколко микросекунди и се определя от изходното съпротивление на усилвателите, съпротивлението на затворения ключ в аналоговия мултиплексор и входният капацитет на схемата следене/запомня-



Фиг. 2.7. Блокова схема на многоканален АЦП

не.

- ◆ Привежда се схемата следене/запомняне в състояние “следене” и се изчаква времето за установяване на изходното ѝ напрежение. Това време също е от порядъка на няколко микросекунди и се определя от изходното съпротивление на схемата следене/запомняне, големината на запомнящия кондензатор и входното съпротивление и капацитет на АЦП.

- ◆ Привежда се схемата следене/запомняне в състояние “запомняне”. Изчаква се преминаването на преходните процеси. Това време е от порядъка на 5 - 6 микросекунди.
- ◆ Стартира се аналогово-цифровото преобразуване в блока АЦП. Изчаква се времето за протичането му. В зависимост от използвания тип АЦП то е няколко десетки микросекунди — 25 до 100.
- ◆ Прочита се числото на изхода на АЦП и се запомня в паметта на системата за събиране и обработка на информация с което процедурата за едно измерване приключва.

Минималното време между две измервания е сумата от четирите времена за изчакване плюс времето за прочитане и запомняне на резултата от микропроцесорната система. За типичните осембитови микропроцесори това време е ограничено от бързодействието на микропроцесора и трудно може да бъде под двадесет микросекунди при програмно четене и запис на данните в паметта.

2.2.2.3. Реализиране на управляващия блок

В зависимост от бързодействието на АЦП процесът на управление на многоканалният АЦП може да бъде както изцяло апаратно реализиран (с тригери, броячи, чакащи мултивибратори и др.) за АЦП с бързодействие под 20 микросекунди, така и изцяло да се осъществява по програмен път, при което блокът “управление” се свежда до едни регистър в който управляващият микропроцесор може да записва. Три от изходните битове на регистъра се използват за избор на канал от аналоговия мултиплексор, един — за директно управление на състоянието на схемата следене/запомняне. Съществуват варианти и при начина на прочитане на резултата от измерването. За бързи АЦП целият масив от много измервания се записва директно в паметта на микропроцесорната система посредством DMA канал или дори в собствена бърза памет. При по-ниски бързодействия микропроцесорът чете направо от АЦП отчет след отчет.

2.2.2.4. Необходимост от схемата следене/запомняне и изисквания към нея

Предназначението на схемата следене/запомняне е:

- ◆ да запомни входното напрежение с точност по-добра от точността на АЦП;

◆ да загържи това напрежение за времето на преобразуване на АЦП.

За времето на запомнянето изходното напрежение намалява ^{поряди} ~~за време~~ ^U ~~платката~~ на разреждането на запомнящия кондензатор през входното съпротивление на повторителя в самата схема следене/запомняне и през утечните съпротивления на запомнящия кондензатор и в самата печатна платка. За времето на преобразуването намаляването на това напрежение трябва да бъде по-малко от половин дискрет на АЦП.

При много бавно изменящи се входни сигнали не е задължително използването на схема следене/запомняне. В този случай максималната честота на входния сигнал се определя от формулата:

$$v_{max} \leq \frac{1}{2\pi t_{np} 2^n},$$

където v_{max} е максималната честота на входния сигнал, t_{np} е времето за преобразуване на АЦП, а n е броят на разрядите му. За десетбитов АЦП с време на преобразуване 100 μ S тази честота е 1.554 Hz.

Схема следене/запомняне обикновено не е необходима при използване на АЦП от интегриращ тип, които са бавни и дават средната стойност на входното напрежение за времето на интегриране. Не се използва схема следене/запомняне и в случаите, когато АЦП работи на принципа на капацитивно преразпределение на заряда. Такива са АЦП в еднокристалните микроконтролери от серията MC68HC05 и MC68HC11.

2.2.2.5. Получаване стойността на входното напрежение

Важен момент е получаването на стойността на входното напрежение от числото N , получено от АЦП. Определящи са минималното U_{min} и максималното U_{max} входни напрежения и разрядността n на АЦП:

$$U_{вх} = U_{min} + N \cdot \frac{U_{max} - U_{min}}{2^n - 1}.$$

Например, за десет разряден АЦП с обхват от -5 V до +5 V при получено число 160 входното напрежение е -3.436 V. За дванадесет разряден АЦП с обхват от 0 V до +10 V и получено число 3100 входното напрежение е 7.57 V.

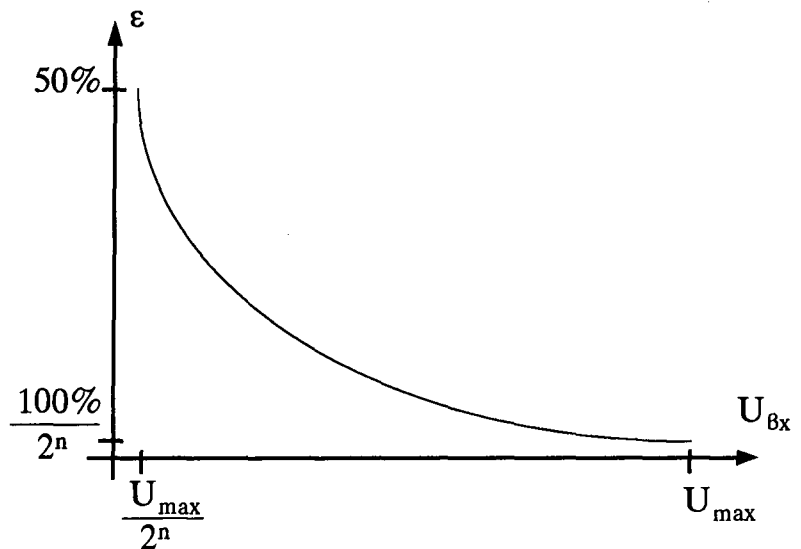
2.2.2.6. Грешки при измерване на входните величини посредством АЦП

При идеален АЦП абсолютната грешка в измерването на входното напрежение е половината от големината на един дискрет — множителя след N в предишната формула. Относителната грешка се дава от израза:

$$\varepsilon = \frac{\frac{1}{2} \text{LSB}}{U_{\text{вх}}}$$

Това означава, че при входно напрежение близко до максималното относителната грешка е много пъти по-малка отколкото при входни напрежения близки до нулата, където грешката може да бъде и 50%. За един десетбитов АЦП с обхват от 0 до 10 V при входно напрежение 9 V относителната грешка е 0.109%, докато при входно напрежение 100 mV тя е 9.77%. Абсолютната грешка и в двата случая е 9.77 mV. Зависимостта на относителната грешка от входното напрежение е показана на фиг. 2.8. При реалният АЦП към тази грешка се прибавят още:

- грешките на реалния аналогово-цифров преобразувател;
- грешката от пада на напрежение в ключовете на аналоговия мултиплексор;
- грешките в схемата следене/запомняне;
- грешките във входните операционни усилватели — коефициент на усилване, температурен дрейф;
- индуктирани външни смущения и шумове.



Фиг. 2.8. Относителна грешка на идеален АЦП в зависимост от входното напрежение

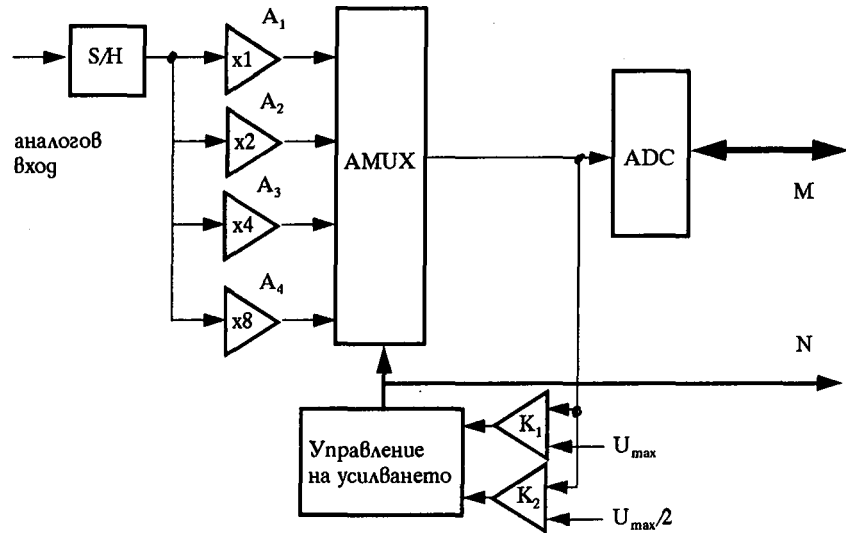
2.2.3. Повишаване на точността и динамиката

В редица случаи в практиката се налага въвеждането на сигнали със зададена относителна грешка, които имат голям динамичен диапазон — отношение между най-малката и най-голямата работна амплитуда

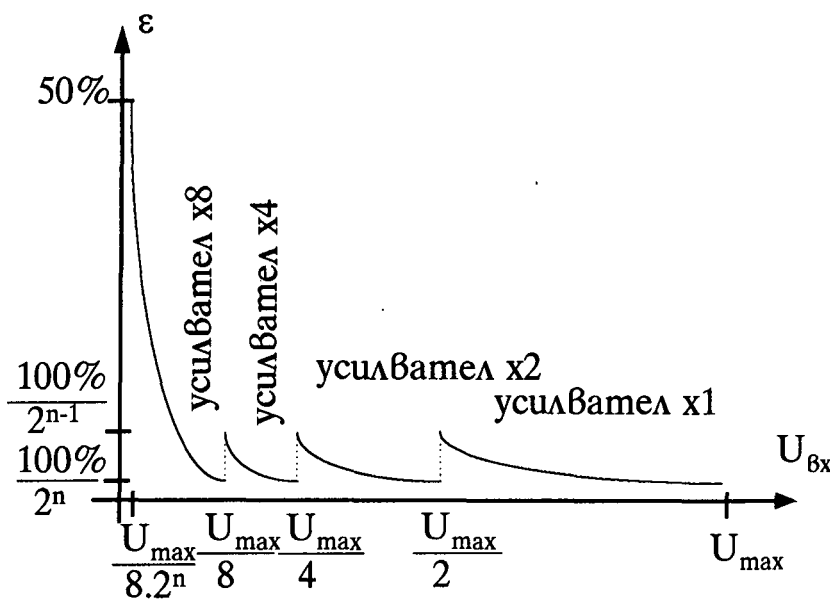
на входния сигнал. За да се получи необходимата относителна грешка при сигнала с най-малка амплитуда се налага използването на високо разрядни и скъпи АЦП. Съществуват и други решения при възникване на такъв проблем.

2.2.3.1. АЦП с плаваща запетая

Блоковата схема на АЦП с плаваща запетая е дадена на фиг. 2.9. Състои се от схема следене/запомняне S/H, операционни усилватели с коефициенти на усилване 1, 2, 4 и 8 пъти, аналогов мултиплексор AMUX, блок за управление на усилването и стартиране на АЦП, два компаратора и АЦП. Двамата компаратора сравняват входното напрежение с максималното и с половината му на максималното за АЦП.



Фиг. 2.9. Блокова схема на АЦП с плаваща запетая



Фиг. 2.10. Относителна грешка на АЦП с плаваща запетая

След запомняне на амплитудата на входния сигнал в схемата следене/запомняне се извършва избор на необходимото усилване от блока "управление на усилването". Той се стреми да поддържа входното напрежение на АЦП между максималното и половината му, използвайки изходите на двата компаратора. Ако напрежението е под половината

на максималното — превключва аналоговия мултиплексор АМУХ към изхода на усилвател с по-голям коефициент на усилване. Ако напрежението е по-голямо от максималното — превключва аналоговия мултиплексор към изхода на усилвател с по-малък коефициент на усилване. Процесът завършва след довеждане на входното напрежение между максималното входно напрежение на АЦП и половината му, или след превключване на аналоговия мултиплексор към изхода на усилвателя с най-голям (респективно най-малък) коефициент на усилване. След това се стартира АЦП. В резултат се получават две числа — M от АЦП и N — номера на канала на аналоговия мултиплексор. Входното напрежение се изчислява по формулата:

$$U_{вх} = \left(M \cdot \frac{U_{max}}{2^n - 1} \right) \cdot 2^{-N}$$

От представянето на входното напрежение с две числа — експонента и мантиса, както се представят числата в плаваща запетая в паметта на компютъра, идва и името на този вид АЦП.

Зависимостта на относителната грешка от входното напрежение е показана на фиг. 2.10. При сравнение с грешката на идеален АЦП (фиг. 2.8) се вижда, че относителната грешка остава под определена стойност за значително по-широк обхват на входните напрежения.

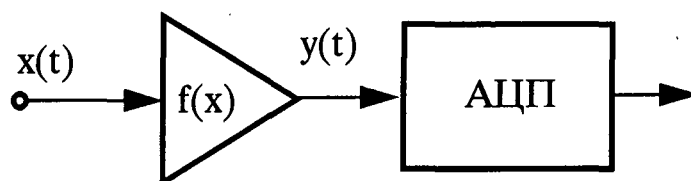
2.2.3.2. АЦП с нелинейна характеристика

В този случай пред АЦП се поставя аналогов блок с нелинейна преобразователна характеристика, както е показано на фиг. 2.11. Аналоговият блок пред АЦП обикновено реализира логаритмична характеристика с което се запазва постоянна относителна грешка в целия обхват на входните напрежения. Изхождайки от постановката, че Δy е постоянна величина (стъпката на квантуване на АЦП) в общия случай на функционален преобразувател, реализиращ функцията $y = f(x)$ пред входа на АЦП, може да се напише следната формула за относителната грешка:

$$\frac{dx}{x} = \frac{\Delta y}{f'(x) \cdot x}$$

При линейна функция $y = k \cdot x$ за относителната грешка се получава:

$$\frac{dx}{x} = \frac{\Delta y}{k \cdot x}$$



Фиг. 2.11. АЦП с нелинейна характеристика

което е и функцията, показана на фиг. 2.8. За логаритмична характеристика на входния функционален преобразувател $y = \ln(x)$ се получава твърде интересният резултат:

$$\frac{dx}{x} = \Delta y,$$

което означава постоянна относителна грешка в целия обхват. За да се избегне безкрайното усилване при входен сигнал, клонящ към нула, се извършва изместване на предавателната характеристика. При двуполлярни сигнали предавателната характеристика се прави и симетрична спрямо нулата. Една често използвана предавателна функция е:

$$f(x) = x_{max} \cdot \frac{\ln\left(1 + \frac{\mu \cdot x}{x_{max}}\right)}{\ln(1 + \mu)}$$

където μ е амплитудата, при която от линейна се преминава към логаритмична характеристика. Такива нелинейни характеристики се използват в цифровите телефонни централи. По принцип 256 нива на квантуване (8 битов АЦП) са напълно достатъчни за нормална разбираемост на говор. За подобряване на качеството и при слаби сигнали се използва нелинейна характеристика от този тип (така наречения μ - закон), която позволява довеждане на грешката при слаби сигнали както при 13 битов АЦП. При това потокът от данни си остава с интензивност 64 Kbit/s (8 бита с честота на дискретизация 8 000 Hz). При възпроизвеждане е необходимо след ЦАП да се използва нелинеен функционален преобразувател, реализиращ обратната функция, за възстановяване на нормалните амплитуди на сигнала.

Основният недостатък на това решение е в аналоговия функционален преобразувател. Трудно е да се направи аналогов нелинеен функционален преобразувател с висока точност (0.01%) и слаба зависимост на характеристиката от външни фактори (главно температурата). При серийно производство е необходима и добра повторяемост на тази характеристика при минимални настройки.

2.2.3.3. АЦП с цифрово реализиране на нелинейната характеристика

Цифровото реализиране на нелинейната характеристика запазва всички предимства, коментирани по-горе, като избягва основния недостатък — аналоговия нелинеен преобразувател. Едно решение за цифрово

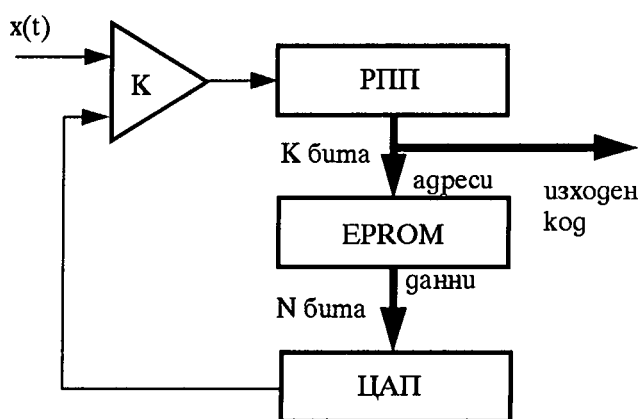
реализиране на нелинейната характеристика е показано на блоковата схема на фиг. 2.12. На практика това е един аналогово-цифров преобразувател с поразрядно кодиране, управляван от K -битовия регистър за последователни приближения РПП. Изходният му код се прекодира от една постоянна EPROM памет, където е записана нелинейната предавателна функция и се подава на N -битовия ЦАП. Обикновено $N > K$ за прецизно реализиране на предавателната функция. Стойността на дискретизирания отчет се получава след K такта на честотата на РПП. Това решение е подходящо за сравнително високи честоти на дискретизация и не изисква никаква допълнителна обработка от страна на управляващия микропроцесор. Като недостатък може да се отбележи необходимостта от използване на високо разряден ЦАП.

Друг възможен вариант на цифровото реализиране на нелинейната предавателна характеристика е поставянето на прекодиращата EPROM памет след АЦП. Недостатъкът тук е, че стойността на дискретизирания отчет ще се получи след N такта на честотата на РПП, което в общия случай е повече време.

Ако целта е просто да се намали обема памет за съхраняване на дискретизирания сигнал реализирането на нелинейната характеристика може да се извърши и по програмен път — след прочитане на 12-битовия отчет от АЦП той се прекодира до осембитова стойност с необходимата нелинейна функция. Това е възможно при ниски честоти на дискретизация — микропроцесорът трябва да има време да направи прекодирането по програмен път.

2.2.4. Роля и изисквания към входния филтър

За правилното дискретизиране на входните сигнали е необходимо на входа на АЦП да не се подават сигнали с честота по-висока от половината на честотата на дискретизация. В различните източници това изискване се нарича теорема на Шенон, теорема на Найкуист или теорема на Котелников. За тази цел пред блока аналогово-цифров преоб-

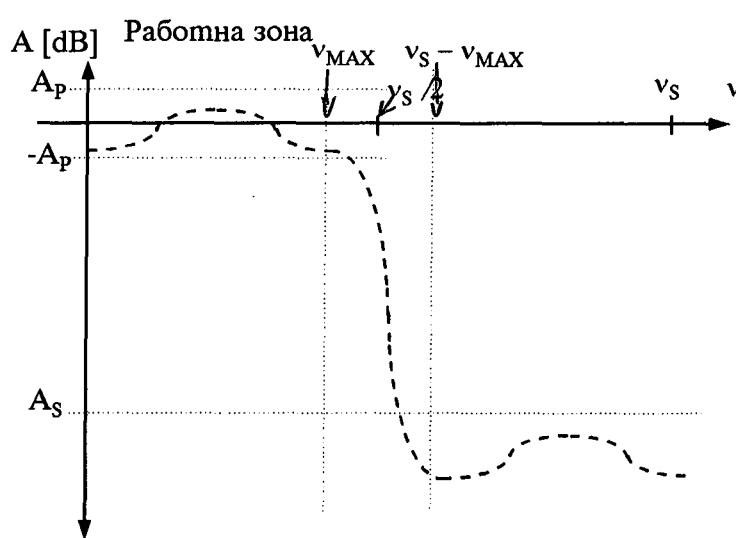


Фиг. 2.12. Цифрово реализиране на нелинейна характеристика на АЦП

зувател се поставя нискочестотен аналогов филтър. Примерната амплитудно-честотна характеристика на един такъв филтър е показана на фиг. 2.13. Той трябва да изпълни следните задачи:

- да пропусне всички сигнали с честота в рамките на работната честотна лента от 0 до $\nu_{m ax}$ като неравномерността в лентата на пропускане A_p трябва да бъде по-малка от половин дискрет на АЦП;
- да спре всички сигнали с честота по-голяма от $\nu_s - \nu_{m ax}$, като подтискането A_s е такова, че дори и сигнали с максимална входна амплитуда да се пропуснат с амплитуда по-малка от половин дискрет.

За един десет битов АЦП това означава $A_p = 0,0042 \text{ dB}$ (!) и $A_s = 66 \text{ dB}$. Много силно е и изискването за стръмността на АЧХ. За $\nu_s = 10\,000 \text{ Hz}$ и $\nu_{m ax} = 4\,000 \text{ Hz}$ се получава стръмност в областта 4 000 - 6 000 Hz около 120 dB/окт, което означава 20-звенеен филтър (!).



Фиг. 2.13. Амплитудно-честотна характеристика на филтър пред АЦП

Получените числени стойности показват, че формулираните в началото на тази точка изисквания са изключително силни, особено изискването за стръмност на АЧХ при прехода от зоната на пропускане към зоната на подтискане. В практиката се приема, че в честотната лента на входния сигнал няма сигнали с честота над $\nu_{m ax}$, като

при необходимост се увеличава честотата на дискретизация. Отделянето на полезния, необходимия, сигнал от смущаващите сигнали и шумове се извършва по методите на цифровата обработка на сигналите след като сигналът е дискретизиран и въведен в паметта на компютърната система.

Входен филтър все пак се поставя, но това е един 4 - 8 звенеен нискочестотен аналогов филтър. Изискването за равномерност на АЧХ в лентата на пропускане остава. Типичен пример за построяването на такъв филтър е с използване на интегралната схема MAX274 на фирмата MAXIM. Тя представлява един осемзвенеен аналогов филтър с

външни резистори, задаващи честотата на среза. Филтърът може да се проектира с апроксимация на честотната характеристика по Бесел, Чебишев, Бътнеруорд или с елиптична апроксимация.

Друг вариант е използването на специализирани интегрални схеми — филтър с превключваеми кондензатори. Те обикновено изискват опорна честота 100 или 255 пъти по-голяма от честотата на дискретизация и реализират АЧХ на един 6, 8, 10 или 16 звенен аналогов филтър. Не изискват външни елементи. Като недостатък може да се посочи повишения шум в изходното напрежение.

В редица случаи амплитудно-честотната характеристика на входните вериги е стандартизирана и не подлежи на промяна. Такива са АЧХ на усилвателите на електрокардиографски сигнал и на повечето биологични сигнали. Това е направено за да може електрокардиограма, направена в София да бъде гледана от специалист кардиолог в Париж или Москва. Да се измени АЧХ на входните вериги означава да се измени и формата на въведения сигнал. Такива са и АЧХ на входните вериги на някои стъпала в звукотехниката: за електродинамична гоза (RIAA корекция); входните и изходните вериги при запис на магнитна лента (DOLBY корекция при запис и възпроизвеждане). В тези случаи задачата на филтъра е да има АЧХ максимално близка до стандартизираната.

При всички случаи проблемът с аналоговия филтър пред АЦП не бива да се подценява.

2.2.5. Проблем с галваничното ^{разделяне} ~~развързване~~ при въвеждане на аналогови сигнали

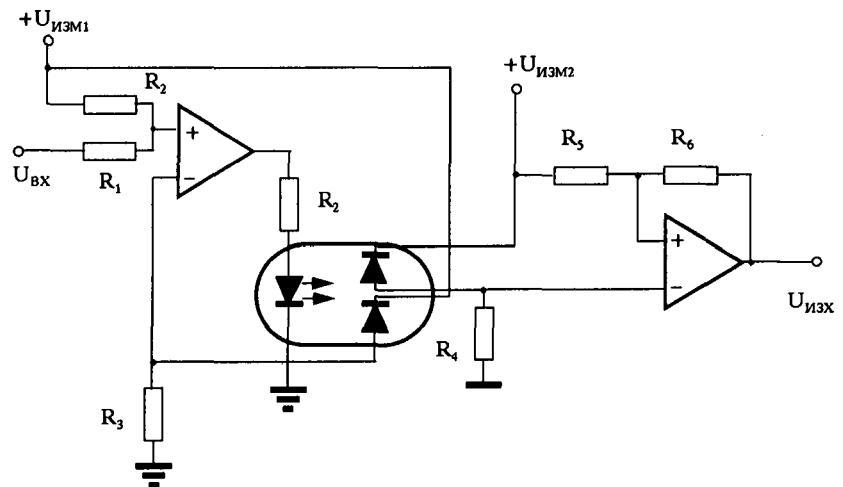
Всичко казано в т. 2.1.2 за необходимостта от галванично развързване при въвеждане на потенциални и импулсни сигнали в компютърната система важи и за аналоговите сигнали. Особено силно стои този проблем при въвеждането на биологични сигнали, които се снемат посредством електроди, прикрепени към тялото на пациента. Сериозна е и необходимостта от галванично развързване и при въвеждането на аналогови сигнали в системи предназначени да управляват промишлени обекти.

2.2.5.1. Галванично разделяне на ниво аналогов сигнал

Методите за галванично разделяне на ниво аналогов сигнал са обект на разглеждане от страна на аналоговата схемотехника. Типично използвани решения са:

- Аналоговият сигнал модулира (най-често амплитудно) високочестотно колебание, което преминава през трансформатор. След вторичната намотка на трансформатора следва линеен детектор и усилвател. Недостатъците на това решение са в сравнително големия брой елементи и използването на трансформатор. При използване на широчинно импулсна модулация е възможно и използването на оптрон с известна загуба на линейност.

- За галванично отделяне се използва диференциален оптрон. Вторият ^{светоодиод} светодиоди е в обратната връзка на управляващата светодиода част за да се компенсира нелинейността на оптрона. Типовото схемно решение е показано на фиг. 2.14. Недостатък на това решение е в по-голямата нелинейност на предавателната характеристика.



Фиг. 2.14. Галванично разделяне на аналогов сигнал посредством диференциален оптрон

2.2.5.2. Галванично разделяне с междинно преобразуване в честота

Аналоговият сигнал се подава на входа на генератор, чиято честота зависи от управляващото напрежение. Получената импулсна поредица, с честота пропорционална на амплитудата на аналоговия сигнал, преминава през оптрон по начините, описани за импулсните сигнали. Честотата се измерва посредством таймерната подсистема на управляващия микрокомпютър, която в този случай играе ролята на АЦП. Методът е много точен, позволява пренасянето на аналоговия сигнал във вид на импулсна поредица на голямо разстояние без загуба на точност, прецизното измерване на честотата не изисква скъпи елементи.

Недостатък е по-голямото време, необходимо за измерване на честотата с висока точност. Използва се за индустриални приложения, например за измерване на температурата в пещи. При добър разчет на честотите на генератора може да се използва и за сигнали с по-широка честотна лента, дори за биосигнали.

2.2.5.3. Галванично разделяне на ниво TTL сигнали

Между управляващата микропроцесорна система и блока АЦП се поставят оптрони — на шината за данни от АЦП към микропроцесорната система и на управляващите сигнали — от микропроцесорната система към блока АЦП. Това е едно сравнително скъпо решение с много елементи. Подходящо е за АЦП със сравнително малък брой разряди (до 8) и бързодействие между 30 и 100 μ S.

2.2.5.4. Галванично разделяне на ниво цифров интерфейс

В повечето съвременни еднокристални микроконтролери е вграден синхронният сериен интерфейс SPI (serial peripheral interface). Скоростта на обмен е от няколко десетки килобод до един мегабод и е програмируема. Съществуват и еднокристални АЦП със същия интерфейс. Галваничното развързване се осъществява с бързи оптрони.

Друг възможен вариант е изграждането на интелигентна система за управление на АЦП с микропроцесор и комуникация с главната система посредством някой от стандартните интерфейси — RS232, RS422 или IEEE488. По своята същност една такава система е също система за събиране и обработка на информация.

2.3. Извеждане на потенциални и импулсни сигнали

2.3.1. Видове и реализация

Разнообразието на изходните за системата за събиране и преобразуване на информация потенциални и импулсни сигнали е значително по-голямо от входните:

- контакт — реализира се посредством реле, чиито контакти са оразмерени за съответния ток и напрежение;
- напрежение — реализира се с помощта на транслатор на ниво от TTL до изискваните напрежения за двете логически нива;
- ток — реализира се посредством преобразувател на TTL нива в две стойности на изходящия ток;

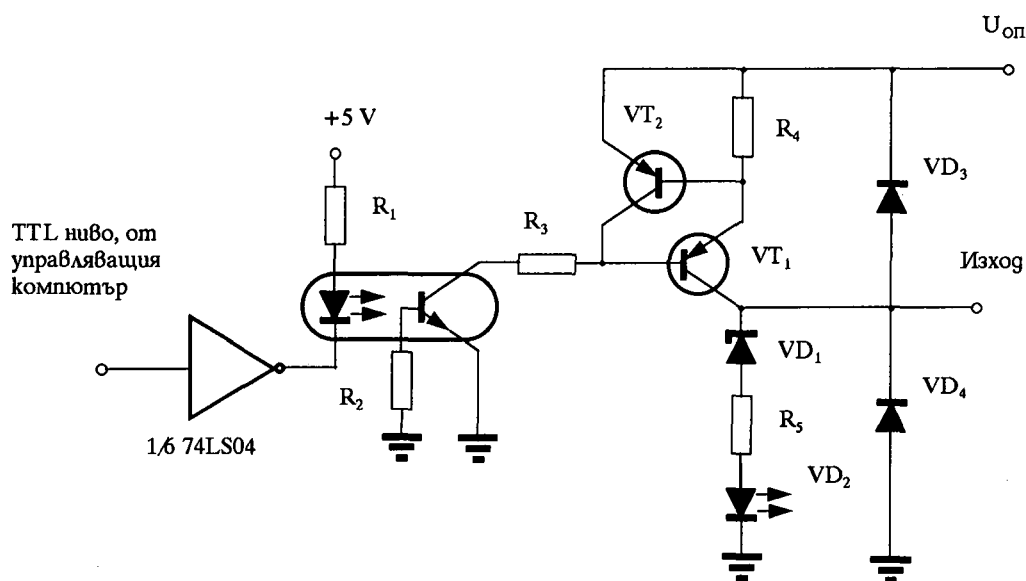
- включване на консуматор към постоянно напрежение — това може да стане както с реле, така и с помощта на достатъчно мощен транзисторен ключ;

- включване на консуматор към променливо напрежение — реализира се с реле или триак.

Независимо от типа на изходния сигнал, с цел да защитим компютърната част, е желателно да се реализира галванично развързване на компютърната система от изходните вериги. Когато изходът е релеен контакт това е изпълнено. В останалите случаи е наложително използването на оптрони. Желателно е вземането на мерки за защита на изходните вериги от неправилно включване, от външни пренапрежения и къси съединения.

2.3.2. Типова схемотехника с галванично развързване

Типична схема на изходна верига за индустриални цели е показана на фиг. 2.15. За разлика от входните вериги, тук е необходимо наличието на допълнително захранващо напрежение $U_{оп}$. Това напрежение служи за захранване на външните вериги и е галванично отделено от захранващото $+5\text{ V}$ напрежение за компютърната система. Резисторът R_1 ограничава тока през светодиода на оптрона. Фототранзисторът на оптрона, резисторът R_3 и транзисторът VT_1 работят така, че когато фототранзисторът е осветен, VT_1 е отпушен. Групата VT_2 и R_4 защитават изходния транзистор VT_1 от късо съединение в изхода. Диодите VD_3 и VD_4 предпазват изходните вериги при проникване на отрицателни напрежения или на напрежения по-високи от $U_{оп}$. Групата на светодиода VD_2 , ценеровият диод VD_1 и резисторът R_5 индицира състоянието на изхода. Резисторът е токоограничаващ.

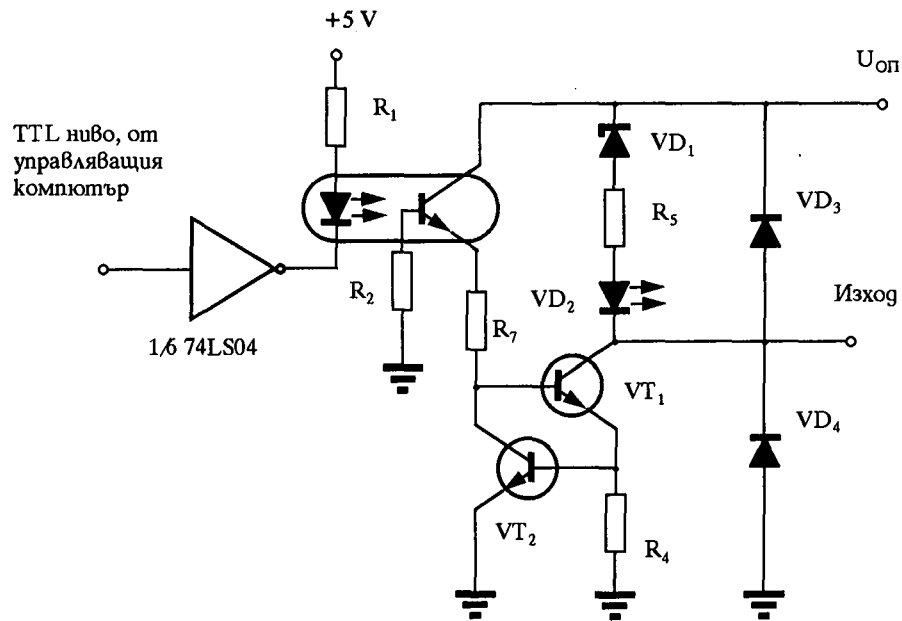


Фиг. 2.15. Изходна верига с галванично развързване

Пробивното напрежение на ценовия диод се избира половината от $U_{оп}$. При тази реализация индикаторният светодиод ще светне само когато изходното напрежение *наистина* превиши половината от операционното. Ако индикаторният светодиод е свързан последователно на светодиода на оптрона, той ще светне дори и когато има късо съединение в изхода.

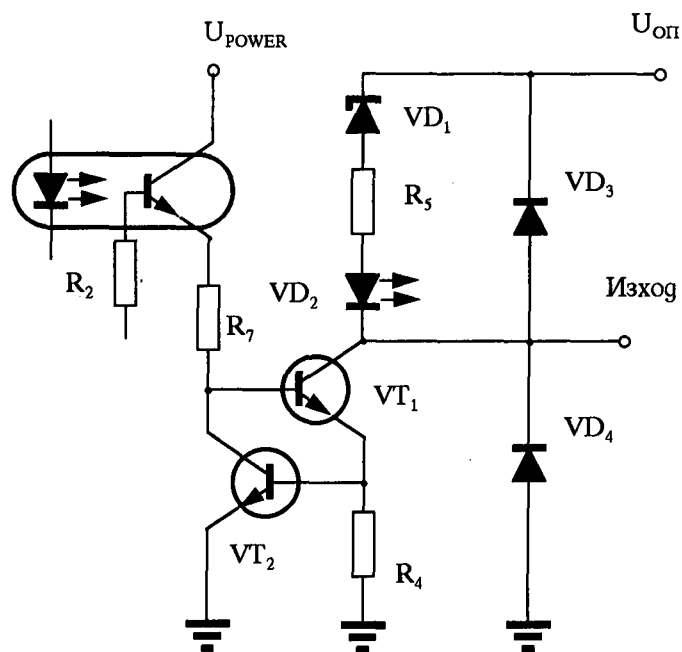
Разгледаното дотук схемно решение е за товар, свързан към галванично отделената маса. Ако искаме товарът да се включва към операционното напрежение се използва схемата, показана на фиг. 2.16. Предназначението на отделните вериги е същото, само е сменен типа на използваните транзистори.

За предпазване от подаване на фалшиви изходящи сигнали при включване на микропроцесорната система често маломощните управляващи вериги се хранят отделно, от напрежението U_{POWER} , както е показано на фиг. 2.17. Това напрежение се включва от микропроцесорната система по програмен път, след преминаване на началните процедури и тестове.



Фиг. 2.16. Изходна верига с галванично развързване и товар към $+U_{оп}$

Независимо от това, неактивното състояние на изхода трябва да съответства на неактивно състояние (каквото е след апаратен Reset) на изходите на микропроцесорната система.



Фиг. 2.17. Изходна верига с разделно захранване на управляващите вериги

2.4. Извеждане на аналогови сигнали

2.4.1. Типова схемотехника на изходните блокове

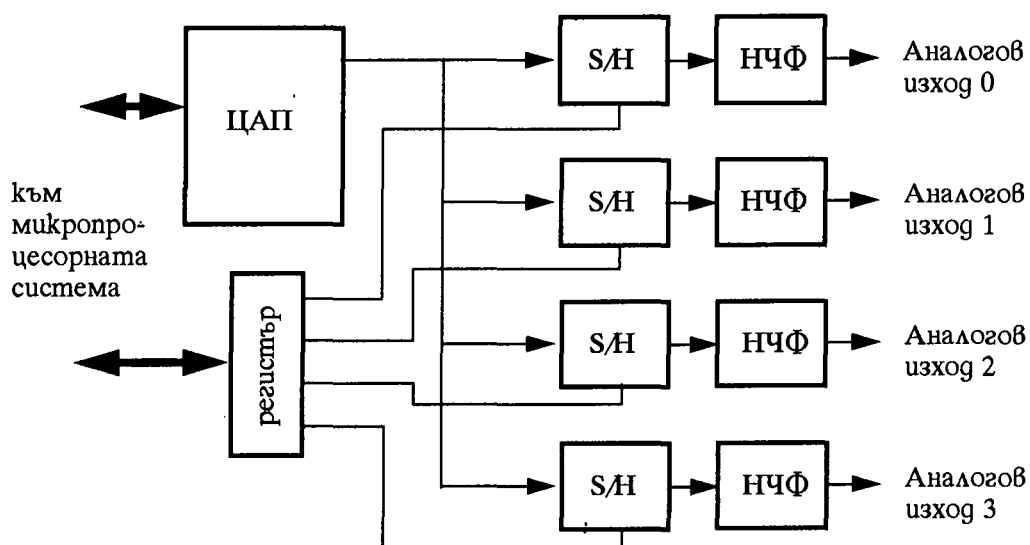
В общия случай аналоговите изходи се реализират посредством цифрово-аналогов преобразувател, като на всеки аналогов изход съответства един ЦАП. Аналоговият блок след ЦАП трябва да извърши следните дейности:

- да превърне изходното напрежение или ток на изхода на интегралната схема ЦАП в нормирана изходна величина — напрежение или ток в съответния обхват;
- да осигури защита от късо съединение в изходните вериги;
- да предпази от проникването на импулсни смущения в системата за събиране и преобразуване на информация.

Схемотехниката на изходните блокове няма никакви съществени особености, единственото изискване е да осигури определената от разрядността на ЦАП точност. При извеждане на сигнали с широка честотна лента след ЦАП задължително се поставя нискочестотен филтър. Изискванията към него са същите, както към филтъра пред АЦП.

2.4.2. Реализация на многоканален ЦАП

Необходимостта от извеждането на много аналогови сигнали има недостатъка от вграждането в системата на много интегрални схеми ЦАП. В редица случаи е приложимо решението на многоканалния ЦАП, чиято блокова схема е показана на фиг. 2.18.



Фиг. 2.18. Реализация на многоканален ЦАП

Микропроцесорната система подава на ЦАП число, съответстващо на напрежението, което трябва да се изведе от аналогов изход 0. Посредством запис в регистъра поставя съответната схема следене/запомняне за десетина микросекунди в режим “следене” и отново я връща в режим “запомняне”. След това извежда напрежението за аналогов изход 1 и го запомня в съответната схема следене/запомняне. По този начин опреснява напрежението на всички аналогови изходи. Предимството на това решение е в използването само на един ЦАП. Като недостатък може да се отбележи необходимостта от програмно опресняване на изходните напрежения. Интервалът на опресняване се определя от схемата следене/запомняне и от необходимата точност на изходното напрежение.

2.4.3. Проблеми с галваничното развързване при извеждане на аналогови сигнали

Всичко казано за необходимостта от галванично развързване на входните аналогови сигнали важи и за изходните аналогови сигнали.

2.4.3.1. Галванично разделяне на ниво аналогов сигнал

Методите за галванично разделяне на ниво аналогов сигнал вече са разгледани в точка 2.2.5. Типовите схемни решения, предимствата и недостатъците са същите.

2.4.3.2. Галванично разделяне с преобразуване в честота

Като ЦАП се използва част от таймерната подсистема на микропроцесорната система. Получената честота преминава през оптрон по начините, описани за импулсните сигнали. Честотата се превръща в аналогов сигнал по известните схемотехнични методи или с използване на интегрална схема — преобразувател честота-напрежение. Методът е много точен, позволява пренасянето на изходния аналогов сигнал във вид на импулсна поредица на голямо разстояние без загуба на точност. Недостатък е по-голямото време, необходимо за установяване на изходното напрежение поради по-голямата времеконстанта, необходима на преобразувателя честота-напрежение.

2.4.3.3. Галванично разделяне с широчинно-импулсна модулация

Методът е сходен с предишния. Разликата е в това, че честотата е постоянна, а се изменя коефициентът на запълване. Като ЦАП от-

ново се използва таймерната подсистема. След галваничното разделяне, посредством оптрон, се поставя нискочестотен филтър, който дава на изхода си средната стойност на напрежението. Предимство е по-малкото време за установяване на изходното напрежение.

2.4.3.4. Галванично разделяне на ниво TTL сигнали

Между управляващата микропроцесорна система и блока ЦАП се поставят оптрони — на шината за данни от микропроцесорната система към ЦАП. Това е едно сравнително скъпо решение с много елементи. Подходящо е за ЦАП със сравнително малък брой разряди.

2.4.3.5. Галванично разделяне на ниво цифров интерфейс

В повечето съвременни еднокристални микроконтролери е вграден синхронният сериен интерфейс SPI (serial peripheral interface). Скоростта на обмен е от няколко десетки килободга до един мегабод и е програмируема. Съществуват и еднокристални ЦАП със същия интерфейс. Галваничното развързване се осъществява с бързи оптрони.

Друг възможен вариант е изграждането на интелигентна система за управление на ЦАП с микропроцесор и комуникация с главната система посредством някой от стандартните интерфейси RS232, RS422 или IEEE488. По своята същност една такава система е също система за събиране и обработка на информация.

Обобщение

В настоящата глава бяха разгледани въпросите и проблемите по отношение на въвеждането и извеждането на импулсни, потенциални и аналогови сигнали в и от системата за събиране и обработка на информация. Разгледана беше типичната схематехника на входните и изходните блокове. Сериозно внимание беше обърнато на необходимостта от галванично разделяне на източника (респективно приемника) на сигнала от системата за събиране и обработка на информация, като един сериозен фактор за повишаване на сигурността на работа на системата. Разгледан беше, също така, и въпросът за нискочестотния аналогов филтър преди АЦП и след ЦАП, като бяха дефинирани и основните изисквания към него.



Задачи

1. Направете разчет на стойностите на елементите на входния блок от фиг. 2.1. за входно напрежение 24 V и максимална честота на входния сигнал 30 Hz.

2. Изведете формулата от точка 2.2.2 за максималната входна честота на сигнала без използване на схема следене/запомняне пред АЦП.

3. Начертайте графично примерната нелинейна преобразователна характеристика, дадена в точка 2.2.3.Б. Използвайте няколко стойности на коефициента μ .

4. Изведете формулата и начертайте графично относителната грешка в зависимост от входното напрежение на примерната нелинейна преобразователна характеристика, дадена в точка 2.2.3.Б за използваните в предишната задача стойности на коефициента μ .

5. Направете пълен разчет на стойностите на елементите от фиг. 2.16 за $U_{оп} = 24$ V и изходен ток до 50 mA.



Контролни въпроси

1. Какво е общото и каква е разликата между потенциалните и импулсните сигнали?

2. Защо е желателно галванично разделяне между източника и приемника на потенциално-импулсния или аналоговия сигнал?

3. Какви са изискванията към аналоговия филтър пред АЦП и след ЦАП?

4. Как може да се повиши динамиката на АЦП със запазване на относителната грешка по-малка от зададена стойност за по-широк интервал от стойности на входното напрежение?

3. Първична обработка на сигналите в компютърната система



Ключови думи

- първична обработка
- линеаризация
- апроксимация
- нормализация
- филтрация
- функции на преобразуване

Какви знания и умения ще придобиете? Къде се прилагат?

В тази тема ще се запознаете с типовите решения и основните проблеми при първичната обработка на въведената в компютърната система информация, допълнителното отсяване на кодове, линеаризация, апроксимация на нелинейните характеристики на входните датчици до получаване на физическите стойности на въвежданите величини.

Получените знания ще приложите при проектирането на първичната обработка на информацията в системите за събиране и преобразуване на информация.

Въведение

✓ Първичната обработка на въвежданата в компютърната система информация - сигнали, данни и др. включва обработката по апаратен и програмен път от въвеждането на информацията в компютърната система до получаването на стойностите на въвежданите величини в техните измерителни единици - олове, паскали, метри, секунди. Тя включва и допълнителното отсяване на груби грешки при въвеждането и премахване на фалшивите стойности. Трябва да ^{се}гарантира получаването на физическите стойности на въвежданите величини с необходимата за по-нататъшна обработка точност.

3.1. Цели и задачи на първичната обработка

Основните *цели* на първичната обработка на въведените сигнали и данни са:

- ⇒ получаване на стойностите на въвежданите величини в техните физически единици - олове, паскали, градуси, метри, секунди, херци със зададената точност;
- ⇒ премахване на евентуалните грешки при въвеждането, породени от шумове и случайни смущения;

- ⇒ отстраняване на всички излишни данни, които не носят информация за изследвания процес и биха затруднили по нататъшната обработка;
- ⇒ получаване на годни за по-нататъшна обработка данни и сигнали.

За постигането на целите на първичната обработка на информацията трябва да бъдат решени следните задачи:

- ⇒ промяна на амплитудните и времевите параметри на входните данни и подготовка за въвеждане в компютърната система - нормализация;
- ⇒ създаване на математически модели на входните датчици и получаване на функционална зависимост "въведено в компютъра число \leftrightarrow реална стойност на физическата величина" и запазване на необходимата точност - функции на преобразуване;
- ⇒ статистическа обработка на въведените данни и отстраняване на грешки, породени от случайни смущения - външни и в работата на веригите за въвеждане в компютърната система;
- ⇒ филтрация на данните в честотна, времева и амплитудна област и отстраняване на максимална част от излишната информация.

3.2. Баланс на решаваните задачи от първичната обработка за апаратна и програмна реализация

Изброените в предишната точка задачи в повечето случаи могат да бъдат решени както с помощта на апаратни средства, така и програмно. Един от основните въпроси при проектирането на системите за събиране и обработка на информация е баланса между задачите, които да бъдат реализирани от апаратната част и задачите, които да се решават от програмното осигуряване.

Основен критерий за добър баланс е минималната апаратна част на изгражданата система като цяло и добре натоварена с програмни задачи компютърна система. Парадоксът тук е, че прехвърлянето изцяло на всички задачи върху програмното осигуряване, изисква мощна компютърна система, респективно по-голяма апаратна част.



Пример за баланс на задачите

Необходимо е да се разработи електронен пулсомер, част от електромедицинска апаратура. Като входен сигнал се подава усилен и филтриран електрокардиографски сигнал. Пулсът трябва да се покаже на малък дисплей, да могат да се задават минимална и максимална стойности. При излизане на пулса от зададените граници е необходимо да се подаде алармен сигнал.

Първи вариант - чисто програмно решение. Изгражда се система на базата на микроконтролера MC68HC11 с външни оперативна и постоянна памети. Използва се вграденият в микроконтролера осембитов АЦП. Компютърната система въвежда електрокардиографския сигнал, като измерва моментната му стойност на всеки 4 mS. Полученият масив от моментни стойности се обработва, разпознават се ударите на сърцето (така наречения QRS комплекс). Измерва се времето между два последователни удара и се превръща в брой удара за минута. Индицира се получената стойност и ако тя е вън от зададените граници се активира алармения сигнал.

Втори вариант - комбинирано решение. Изгражда се система на базата на възможно най-малкия еднокристален микрокомпютър (MC68HC05, Z8, PIC16 или други подобни) в минимална конфигурация само за управление на индикацията и алармения сигнал. Ударите на сърцето се разпознават от апаратен блок (обикновено две диференциращи вериги и компаратор) и се подават като сигнал с TTL ниво на някой от цифровите входове на еднокристалния микрокомпютър. Той измерва времето между два последователни удара и го превръща в брой удара за минута. Индицира получената стойност и ако тя е вън от зададените граници активира алармения сигнал.

Трети вариант - чисто апаратно решение. Апаратно се разпознават ударите на сърцето, прави се цифров решаващ блок и цифров блок за управление на индикацията. С помощта на два цифрови компаратора се проверява дали честотата на ударите на сърцето е в зададените граници и евентуално се активира алармения сигнал.

При сравняване на трите варианта се вижда, че оптимален (с най-малка апаратна част) е вторият вариант. При него най-добре е балансирано разпределението на задачите между програмното осигуряване и апаратната част.



При проектиране на нова система за събиране и обработка на информация задължително се оставя запас от производителност на

компютърната система. Добре натоварената още при разработката система за събиране и обработка на информация няма възможност за развитие. Максимално използване на ресурсите се търси при системи, които ще се произвеждат в големи количества.

3.3. Нормализация

Процесът на довеждане на амплитудните и времевите параметри на входните сигнали до годни за въвеждане в компютърната система стойности се нарича нормализация.

3.3.1. Нормализация на амплитудните параметри

Нормализирането на амплитудните параметри на входните сигнали се извършва апаратно.

За аналоговите сигнали това са аналоговите усилватели със зададен коефициент на усилване в необходимата честотна лента. Тяхното предназначение е да подават на входа на АЦП сигнал с амплитуда близка до максималната за АЦП с цел да се гарантира необходимата точност на въвеждане на амплитудата.

При импулсните и импулсно потенциалните сигнали това са импулсните усилватели, компараторите и трансляторите на ниво.

3.3.2. Нормализация на времевите параметри

За времеви параметри се говори обикновено при импулсните сигнали. Това са продължителност на импулса, честота и коефициент на запълване. Апаратните блокове, които се използват, са разширители на продължителността на импулса, делители на честота, броячи.

Когато е необходимо просто да се отбележи пристигането на кратки импулси, например с цел преброяване, обикновено се използват разширители на продължителността им. Ако продължителността на пристигащите импулси е 1 mS, а компютърната система чете цифровия вход на всеки 10 mS е необходимо продължителността на входния импулс да се разшири, така че да се гарантира възприемането му, т.е. над 10 mS. При това максималната честота на входните импулси трябва да бъде под 50 Hz. Това прави период над 20 mS - над 10 mS импулс и над 10 mS пауза. Типичен разширител на продължителността на импулсите е чакащият мултивибратор.

Когато е необходимо да се измери продължителността на кратък импулс се използват разширители на продължителността на импулса. Типично това се осъществява чрез зареждане на кондензатор с калибриран ток през време на импулса и разреждането му след това с много

пъти по-малък ток. Измерва се времето за разреждане на кондензатора посредством таймерната подсистема на управляващия компютър.

При необходимост от измерване на честота или преброяване на бързо пристигащи импулси често се използва разделяне на многоразрядния брояч на две части - апаратна и програмна. Предварително се извършва делене на честотата им посредством апаратен брояч, чието съдържание може да се чете от компютърната система, а старшите разряди се реализират по програмен път, като се преброяват смените на състоянието на изхода на апаратния брояч.

3.4. Функции на преобразуване

За да получим входната величина в нейните физически измерителни единици е необходимо да се намери функция, която от числото, прочетено от АЦП, да даде стойността на входната величина. В най-простия случай тази функция е полином от първа степен:

$$y = k \cdot x + b,$$

където x е прочетеното от АЦП число, y е стойността на входната величина, а k и b са коефициентите на полинома. Друг често срещан случай е получаването на честотата на входния сигнал от броя входни импулси N за време T :

$$f = \frac{1}{N \cdot T} \quad f = \frac{N}{T}$$

И в двата случая входната величина е аргумент на дадена функция, чието стойност е измерваната величина. В реалната практика функциите на преобразуване са доста по-сложни - от една страна като вид, от друга страна - като брой коефициенти. Дори и в първия случай за да получим вярна стойност на входната величина е необходимо да се знаят точните стойности на двата коефициента. Редица датчици имат нелинейна характеристика на преобразуване на физическата величина в електрически сигнал. Понякога нелинейността е предопределена от самия метод на измерване, както е във втория случай.

И така, за да въведем с дадена точност входната величина е необходимо:

- ⇒ да се намери подходящ тип функция на преобразуване;
- ⇒ да се намерят точните стойности на коефициентите на тази функция.

Типът на функцията зависи от метода на преобразуване на физическата величина, от типа на датчика, от нормализиращите вериги и т.н. Неправилно е да се апроксимира експоненциална или логаритмична зависимост с полином и обратното.

Значително по-сложно стои въпросът с намиране на точните стойности на коефициентите на определената функция. В повечето случаи входните вериги се тарират чрез подаване на еталонни въздействия, измерени с уред поне с един клас по-висока точност от необходимата. След подаване на няколко различни по амплитуда въздействия се оформя таблица, съставена от твърдения от типа:

при подаване на входно въздействие u_i

в компютъра се въвеждаше число x_i .

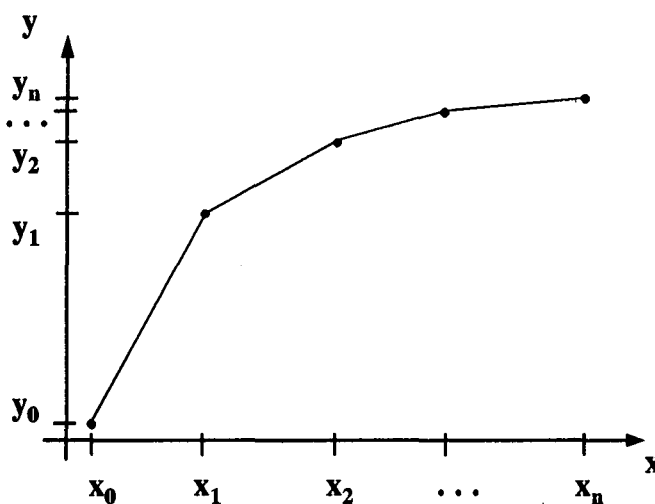
От получената в табличен вид функция на преобразуване е необходимо да се намерят точните стойности на коефициентите. Естествен е стремежът да се намерят колкото се може повече точки от функцията на преобразуване, за да бъде постигнута по-висока точност.

3.4.1. Линейно-отсечкова интерполация

Линейно-отсечковата интерполация е най-простата и често се използва. Идеята е да се свържат известните точки от предавателната функция с отсечки, както е показано на фиг. 3.1. При наличието на повече точки точността на апроксимацията в много случаи е достатъчна. Уравнението на права, свързваща точките с координати (x_{n-1}, y_{n-1}) и (x_n, y_n) се дава от израза:

$$y = \frac{y_n - y_{n-1}}{x_n - x_{n-1}} \cdot (x - x_{n-1}) + y_{n-1}$$

Границите и коефициентите на правата за всеки интервал се изчисляват предварително и се запомнят в паметта на компютърната система. След всяко въвеждане на входната величина x програмата проверява в кой интервал се намира входната величина и с помощта на коефициентите за този интервал намира физическата стойност на величината y . Недостатък на метода е в това, че програмата ще губи различно време при изчисляване на изходната величина, докато търси нейният интервал. Друг недостатък е, че преходната функция може да има различна от права форма между точките. Този недостатък може да се компенсира с повече точки от преходната характеристика, което, обаче, ще забави



Фиг. 3.1. Линейно отсечкова интерполация

още повече изчисляването на величината y . Основните предимства на метода са неговата простота и в избягване на необходимостта да търсим вида на функцията на преобразуване.

3.4.2. Интерполация с полином

Интерполирането по този метод намира полином, преминаващ през всички точки от таблицата. Дадени са $n+1$ таблични стойности $(x_i, y_i), i = 0, 1, \dots, n$. Да се прекара полином от n -та степен през всички точки. Съставя се системата уравнения:

$$a_n \cdot x_0^n + a_{n-1} \cdot x_0^{n-1} + \dots + a_1 \cdot x_0 + a_0 = y_0$$

$$a_n \cdot x_1^n + a_{n-1} \cdot x_1^{n-1} + \dots + a_1 \cdot x_1 + a_0 = y_1$$

...

$$a_n \cdot x_n^n + a_{n-1} \cdot x_n^{n-1} + \dots + a_1 \cdot x_n + a_0 = y_n$$

Тази система от $n+1$ уравнения от първа степен с $n+1$ неизвестни (коефициентите на полинома) е решима и в резултат ще получим интерполация с функцията:

$$y = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \dots + a_1 \cdot x + a_0.$$

Недостатък на метода е високата степен на полинома при наличието на по-голям брой точки - девета при известни десет точки от функцията на преобразуване. Високата степен означава и по-голямо време за получаване стойността на y . Това време частично може да бъде намалено с използване на формулата на Хорнер:

$$y = ((\dots (a_n \cdot x + a_{n-1}) \cdot x + \dots) \cdot x + a_1) \cdot x + a_0.$$

3.4.3. Апроксимация по метода на най-малките квадрати

За да се избегне високата степен на полинома задачата от предишната точка се променя по следния начин: да се прекара полином от степен k ($k < n$) максимално близко да дадените $n+1$ точки, както е показано на фиг. 3.2. Условието е сумата от квадратите на разликите между полинома и известните му стойности да е минимална.

Задачата може да се реши в общия случай на линейна комбинация от функции. Избираме апроксимиращата функция $g(x)$ да бъде линейна комбинация от функции:

$$g(x) = c_k \cdot g_k(x) + c_{k-1} \cdot g_{k-1}(x) + \dots + c_0 \cdot g_0(x).$$

В случай на апроксимация с полином $g_i(x) = x^i$. Грешката от апроксимация за всяка таблична стойност се определя от разликата $\varepsilon_i = g(x_i) - y_i$. Сумата от квадратите на грешките е:

$$E = \sum_{i=0}^n [g(x_i) - y_i]^2.$$

За да е минимална стойността на E трябва първите частни производни спрямо търсените коефициенти c_i да са нули:

$$\frac{\partial E}{\partial c_k} = \frac{\partial E}{\partial c_{k-1}} = \dots = \frac{\partial E}{\partial c_0} = 0.$$

Тъй като:

$$E = \sum_{i=0}^n [c_k \cdot g_k(x_i) + c_{k-1} \cdot g_{k-1}(x_i) + \dots + c_0 \cdot g_0(x_i) - y_i]^2,$$

то условието за минимум се свежда до намиране на решението на системата уравнения:

$$\frac{\partial E}{\partial c_0} = 2 \sum [c_k \cdot g_k(x_i) + c_{k-1} \cdot g_{k-1}(x_i) + \dots + c_0 \cdot g_0(x_i) - y_i] \cdot g_0(x_i) = 0$$

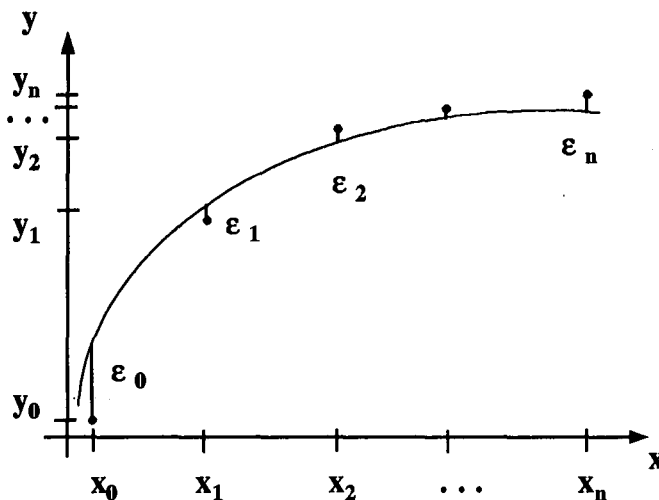
⋮

$$\frac{\partial E}{\partial c_k} = 2 \sum [c_k \cdot g_k(x_i) + c_{k-1} \cdot g_{k-1}(x_i) + \dots + c_0 \cdot g_0(x_i) - y_i] \cdot g_k(x_i) = 0$$

След преобразуване на тези k уравнения се получава лесно решимата линейна система уравнения от първа степен:

$$\begin{vmatrix} \sum g_0^2(x_i) & \sum g_0(x_i) \cdot g_1(x_i) & \dots & \sum g_0(x_i) \cdot g_k(x_i) \\ \vdots & \vdots & \ddots & \vdots \\ \sum g_k(x_i) \cdot g_0(x_i) & \sum g_k(x_i) \cdot g_1(x_i) & \dots & \sum g_k^2(x_i) \end{vmatrix} \begin{vmatrix} c_0 \\ c_1 \\ \vdots \\ c_k \end{vmatrix} = \begin{vmatrix} \sum g_0(x_i) \cdot y_i \\ \vdots \\ \sum g_k(x_i) \cdot y_i \end{vmatrix}$$

След решаване на системата уравнения получаваме коефициентите c_i пред функциите $g_i(x)$ с което сме намерили апроксимиращата функция. В зависимост от характера на таблично зададената функция



Фиг. 3.2. Апроксимация по метода на най-малките квадрати

като функции $g_i(x)$ могат да се използват и експоненциални, логаритмични, синусоидални и косинусоидални функции.

3.4.4. Сплайн апроксимация

Апроксимацията посредством сплайн-функции свързва всяка двойка точки от таблицата посредством кубична функция, като първите и вторите производни в точките на свързване трябва да са равни за да се получи гладка крива. Такива функции се наричат кубични сплайн-функции. Целта е да се определят ^{коэффициентите} коефициентите на всичките n кубични функции. В общ вид кубичните функции се записват така:

$$q_i(x) = k_{3i} \cdot x^3 + k_{2i} \cdot x^2 + k_{1i} \cdot x + k_{0i}, \quad i = 1, 2, \dots, n.$$

Първите $2n$ условия идват от това, че сплайните трябва да се съединяват един с друг в зададените точки:

$$q_i(x_i) = y_i, \quad i = 1, 2, \dots, n;$$

$$q_i(x_{i+1}) = y_{i+1}, \quad i = 1, 2, \dots, n-1.$$

Следващите $2n-2$ условия са, че първите и вторите производни трябва съответно да са равни във всички вътрешни точки:

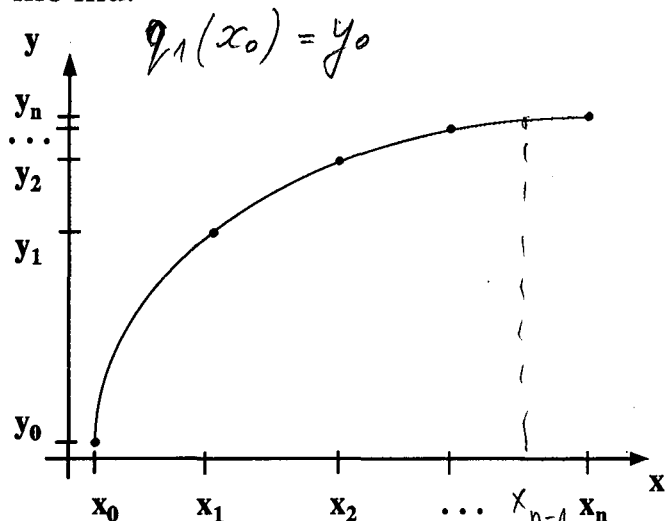
$$q'_{i+1}(x_i) = q'_i(x_i), \quad i = 1, 2, \dots, n-1;$$

$$q''_{i+1}(x_i) = q''_i(x_i), \quad i = 1, 2, \dots, n-1.$$

До тук имаме $4n$ неизвестни и $4n-2$ уравнения. Необходими са ни още две условия. Обикновено за такива се избират:

$$q''_1(x_0) = 0 \text{ и } q''_n(x_n) = 0.$$

След решаване на системата уравнения и получаване на коефициентите на всяка кубична сплайн-функция, намирането на y за произволна стойност на x се свежда до намиране на отсечката, в която лежи аргумента и изчисляване на един полином от трета степен с коефициентите за тази отсечка. Задачата за решаване на системата уравнения и намиране на коефициентите на полиномите може да бъде значително опростена и сведена до намирането само на n коефициента, което я прави със същата сложност като линейно-отсечковата интерполация. Не е изненадващо да се види, че кубичната сплайн-апроксимация



Фиг. 3.3. Сплайн апроксимация

дава значително по-добри резултати от апроксимация с полином от n -ти ред.

3.4.5. Апроксимация, производна и интеграл

Нужно е да се коментира какво ще стане, ако след апроксимиране на функцията на преобразуване на даден датчик, се наложи да се извърши числено диференциране или числено интегриране.

В общия случай производната на апроксимиращата функция няма нищо общо с производната на апроксимираната. Като правило числено диференциране след преминаването на измерваната величина през функция на преобразуване от ред по-голям от първи дава много лоши резултати. Може да се очаква получаването на някаква близка до реалността производна само ако са взети изключителни мерки при апроксимацията за запазване характера на производната.

✓ Върши
(трети)

Малко по-различно стои въпросът при числено интегриране. Ако се извършва числено интегриране по аргумента x в повечето случаи точността на получения интеграл е по-добра дори и от точността на апроксимация, понеже процесът на интегриране усреднява грешките. Ако, обаче, сме интерполирали нелинейната характеристика на един датчик за измерване на скорост и се опитваме да интегрираме показанията му във времето за да получим изминатия път, то грешката ϵ при интерполиране започва да се увеличава линейно с времето на интегриране $\epsilon_f = \epsilon \cdot t$.

3.5. Статистическа обработка

Предварителната статистическа обработка е приложима навсякъде където имаме набор от еднотипни данни: многократно измерена една и съща величина, периодично повтарящи се реализации на един и същи процес и т.н. Целта на предварителната статистическа обработка е да се получи усредняване на въведените данни, като се премахнат грубите грешки. В системите за събиране и обработка на информация по правило статистическата обработка се извършва по програмен път.

3.5.1. Статистическа обработка в амплитудна област

За статистическа предварителна обработка в амплитудна област говорим, когато имаме многократни измервания на една и съща величина. Типично имаме масив $\vec{Y} = [y_1, y_2, \dots, y_m]$ от измерените стойности, който се обработва по следния ред:

- изчислява се средноаритметичната стойност:

$$\bar{y} = \frac{1}{m} \sum_{i=1}^m y_i;$$

• изчислява се средноквадратичното отклонение на серията измервания:

$$\sigma = \sqrt{\frac{\sum_{i=1}^m (\bar{y} - y_i)^2}{m - 1}};$$

• изхвърлят се всички измервания вън от $\pm 3 \cdot \sigma$ за Гаусово разпределение на грешките;

• повтаря се процеса над останалите данни докато не се получи изхвърляне на измерване.

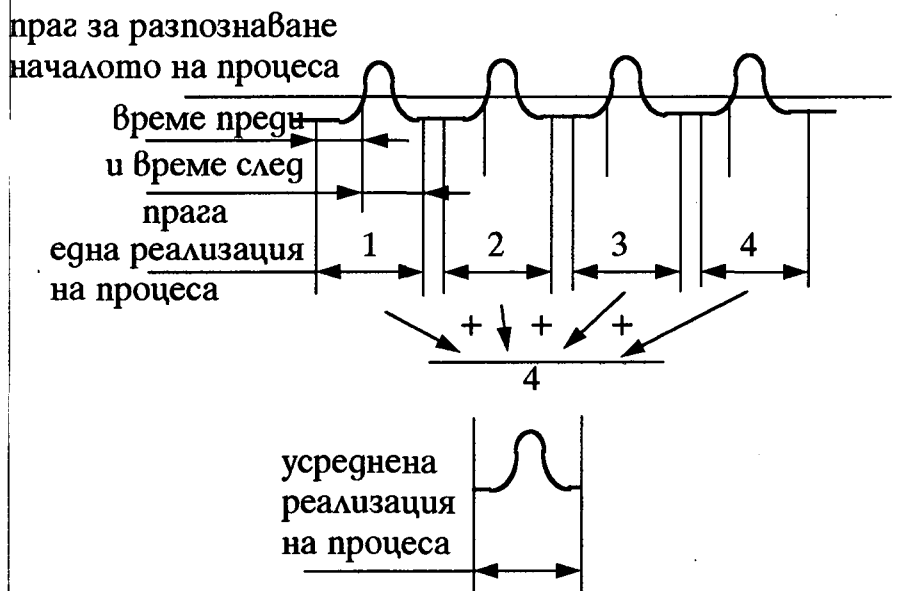
Точността на получената усреднена величина \bar{y} се дава от израза:

$$\sigma' = \sqrt{\frac{\sum_{i=1}^m (\bar{y} - y_i)^2}{m' \cdot (m' - 1)}},$$

където m' е броят на останалите след редуцирането измервания.

3.5.2. Статистическа обработка във времева област

За статистическа предварителна обработка във времева област говорим, когато имаме многократни реализации на един и същи процес във времето, като към полезния сигнал са прибавени шумове, смущения и грешки при въвеждането в компютърната система. За целта в масива от периодични последователни измервания на амплитудата на сигнала, съдържащ многократното повторение на изучавания процес се намират характерни точки, еднообразно



Фиг. 3.4. Статистическа обработка във времева област

разположени във всяка реализация - например начало, специфични върхове и т.н. Вземат се определено число измервания преди и определено число измервания след характерната точка на всяка реализация и се осредняват. Процесът е показан нагледно на фиг. 3.4. В получената средна стойност се намалява присъствието на смущаващи сигнали и шумове. Възможно е прилагането на отхвърляне на отделни реализации на процеса на принципа, описан в предишната точка.

3.6. Филтрация

Филтрацията в честотна област е подтискане на определени честоти от спектъра на въвеждания в компютърната система сигнал. При всяко положение стриктно трябва да се спазва принципа да се въвежда само полезния сигнал и всички честоти вън от неговия спектър да се подтискат.



Примери за сигнали с ограничена честотна лента

При въвеждане в компютърна система за събиране и преобразуване на информация на звуков сигнал от телефонна линия честотната лента трябва да бъде ограничена стриктно в рамките на спектъра на въвеждания сигнал: от 300 Hz до 3400 Hz. На същия принцип при въвеждане на електрокардиографски сигнали честотната лента трябва да бъде ограничена в рамките от 0.5 Hz до 120 Hz.

3.6.1. Филтрация преди въвеждането на сигнала в компютърната система

Филтрацията преди въвеждането на сигналите в компютърната система се реализира апаратно - посредством аналогови филтри със зададена честотна лента. Те се реализират или с операционни усилватели с честотнозависими обратни връзки или със специализирани интегрални схеми за построяване на такива филтри. Проектирането на аналогов филтър със зададена амплитудночестотна характеристика не е предмет на настоящата книга.



При проектирането на аналоговите филтри трябва да се обърне особено внимание на фазочестотната характеристика на проектирания филтър. Дори и при една и съща амплитудночестотна характеристика различните филтри имат различна фазочестотна характеристика. Лошата фазочестотна характеристика може да доведе до съществени промени във формата на въвеждания сигнал.

3.6.2. Филтрация след въвеждането на сигнала в компютърната система

Филтрацията след въвеждане на сигналите в компютърната система се реализира по програмен път с използване на алгоритмите на цифровата филтрация. Те се дискутират в отделна глава на тази книга.

Обобщение

В настоящата глава бяха разгледани методите за предварителна обработка на въведените в компютърната система сигнали.

Целта на първичната обработка на въведените сигнали и данни е получаване на стойностите на въвежданите величини в техните физически единици, премахване на евентуалните грешки при въвеждането, породени от шумове и случайни смущения, отстраняване на всички излишни данни и получаване на годни за по-нататъшна обработка данни и сигнали.

За постигането на целта на първичната обработка на информацията трябва да бъдат решени следните задачи: нормализация, функции на преобразуване, статистическа обработка и филтрация.

Задачите на първичната обработка могат да бъдат решени както с помощта на апаратни средства, така и програмно. Един от основните въпроси при проектирането на системите за събиране и обработка на информация е баланса между задачите, които да бъдат реализирани от апаратната част и задачите, които да се решават от програмното осигуряване.

Процесът на довеждане на амплитудните и времевите параметри на входните сигнали до годни за въвеждане в компютърната система стойности се нарича нормализация. Нормализирането на амплитудните параметри на входните сигнали се извършва апаратно посредством усилватели, компаратори и транслатори на ниво. Нормализирането на времевите параметри на импулсните сигнали се извършва с разширители на продължителността на импулса, делители на честота, броячи.

За да получим входната величина в нейните физически измерителни единици е необходимо да се намери функция, която от числото, прочетено от АЦП, да даде стойността на входната величина. Входната величина е аргумент на дадена функция, чиято стойност е измерваната величина. За да въведем с дадена точност входната величина е необходимо: да се намери подходящ тип функция на преобразуване и да се намерят точните стойности на коефициентите на тази функция. Ка-

то правило функцията на преобразуване се задава в табличен вид. За намиране на функцията на преобразуване се използват линейно-отсечковата интерполация, апроксимация с полином и сплайн-апроксимацията. Сериозно внимание трябва да се обърне на апроксимиращата функция ако в последствие ще се извършва числено диференциране или числено интегриране.

Предварителната статистическа обработка е приложима когато имаме набор от еднотипни данни: многократно измерена една и съща величина, периодично повтарящи се реализации на един и същи процес и т.н. Целта на предварителната статистическа обработка е да се получи усредняване на въведените данни, като се премахнат грубите грешки. В системите за събиране и обработка на информация по правило статистическата обработка се извършва по програмен път. Статистическата обработка може да се извърши в амплитудна и във времева област.

Филтрацията в честотна област е подтискане на определени честоти от спектъра на въвеждания в компютърната система сигнал. Стриктно трябва да се спазва принципа да се въвежда само полезния сигнал и всички честоти вън от неговия спектър да се подтискат.



Задачи

1. За тариране на електронна везна, реализирана с тензометричен мост, разработващите инженери Лили, Иван и Николай разполагат само с еталонна тежест от 4.5 килограма и затова решили да използват собственото си тегло. След претегляне с еталонен кантар се установило, че Лили тежи 48.5 килограма, Иван - 72, а Николай - 84 килограма.

G, kg	0	4.5	48.5	53	72	76.5	84	88.5
N	1	2	3	4	5	6	7	8
U, V	-1 mV	0.229	2.49	2.72	3.69	3.93	4.31	4.54

След това измерили напрежението на изхода на електронната везна, като всеки от тях заставал на везната сам и с еталонната тежест в ръце. Получените резултати са дадени в таблицата. Напишете програма извършваща апроксимиране на функцията на предаване на електронната везна по метода на най-малките квадрати с права:

$$G = k \cdot U + b,$$

където G е търсеното тегло, а U е напрежението на изхода на тензометричния мост. Намерете грешката на апроксимацията във всяка точка.

2. Извършете апроксимация на функцията от предишната задача с полином от по-висока степен. Какво става с грешката при апроксимация?

3. Дадена е функцията $y = 11892 \cdot e^{-0,00866 \cdot x}$. В интервала от -20 до +100 изчислете десет стойности на y . Апроксимирайте получената в табличен вид функция с линейно-отсечкова интерполация. Изчислете грешката от интерполация в няколко точки за всяка отсечка. Данните подредете в таблица. Изчислете средноквадратичната грешка при апроксимацията. Направете същото при апроксимация с полином от втора, трета и четвърта степен по метода на най-малките квадрати. За целта напишете програма на език за програмиране, който знаете. Сравнете получените резултати!

4. Напишете компютърна програма за сплайн апроксимация на функцията от предишната задача. Сравнете грешките с предишните решения. Сравнете формата на функциите на преобразуване при апроксимация с полиноми от трета и четвърта степен и сплайн апроксимация.

5. Напишете компютърна програма изчисляваща производните на апроксимираната функция, апроксимиращите полиноми от трета и четвърта степен и сплайн-апроксимацията от предишната задача. Начертайте графиките на производните и сравнете формата им! Изчислете грешките в точките от предишните задачи и средноквадратичната грешка. Сравнете грешката на производната с грешката при апроксимация!



Контролни въпроси

1. Какви са целите на предварителната обработка на информацията и какви задачи трябва да се решат за постигането им?

2. Защо е необходим баланс между задачите на предварителната обработка, които да се решават апаратно и тези, които да се решават програмно?

3. Какви са съображенията при избор на функция на преобразуване?

4. Какво става с грешката при апроксимация, когато последващо се изчисляват производна или интеграл?

5. Защо е необходима филтрация на входните сигнали?

4. Теоретични основи на дискретизацията и квантуването на сигналите



Ключови думи

- аналогов сигнал
- дискретизиран сигнал
- квантуван сигнал
- дискретизация
- квантуване
- теорема на Шенон
- грешки при реална дискретизация

Какви знания и умения ще придобиете?

Къде се прилагат?

В тази тема ще се запознаете с математическото дефиниране на трите основни вида сигнали при въвеждането и записването им в паметта на компютърната система - аналогов, дискретизиран и квантуван. Ще бъдат дефинирани условията за дискретизация без загуба на информация и необходимият брой нива за квантуване. Ще бъдат разгледани основните грешки при реална дискретизация.

Получените знания ще приложите при проектирането на системите за събиране и преобразуване на информация в частта касаеща вида и разрядността на АЦП, както и необходимата памет за въвеждане на сигнала с цел последваща обработка при грешки по зададените.

Въведение

При въвеждане на аналогови сигнали в компютърната система пред проектиращия инженер обикновено стоят три въпроса:

- ◇ каква честота на дискретизация да се използва;
- ◇ колко разряден да е аналогово-цифровия преобразувател;
- ◇ колко отчета от сигнала да се въведат.

Отговорите на тези въпроси, определящи:

- ⇒ типа (разрядност и бързодействие) на използвания аналогово-цифров преобразувател;
- ⇒ вида (качество и цена) на схемата следене/запомняне;
- ⇒ количеството отчети (т.е. необходимата памет), което трябва да се въведат,

са изключително важни за да се получи успешно реализиране на конкретния инженерен проблем - обработка на сигнала и получаване на резултат със зададена точност

4.1. Класификация на сигналите

4.1.1. Аналогов сигнал

4.1.1.1. Дефиниране на аналогов сигнал

Аналоговият сигнал $x(t)$ се описва от непрекъснатата или непрекъснатата по части функция, обикновено на времето t , при което сигналът е дефиниран в краен отрязък на времето и има ограничена амплитуда:

$$x(t) \Big|_{\substack{t' \leq t \leq t'' \\ x' \leq x \leq x''}}$$

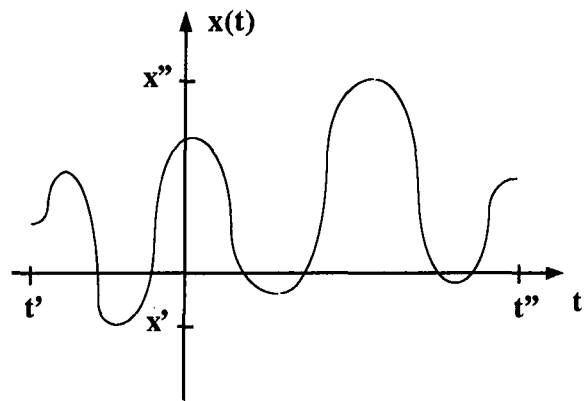
Примерен аналогов сигнал е показан на фиг. 4.1. На тези условия отговарят всички сигнали в природата. Аналоговите сигнали отразяват протичането на някакъв процес във времето, т.е. във времева област. Протичането на същия процес в честотна област се описва еднозначно от спектъра на сигнала $X(\nu)$, който се получава след преобразуване на Фурие. Примерен спектър на аналогов сигнал е показан на фиг. 4.2. Преминването от времева в честотна област и обратно е без загуба на информация:

$$x(t) \Leftrightarrow X(\nu).$$

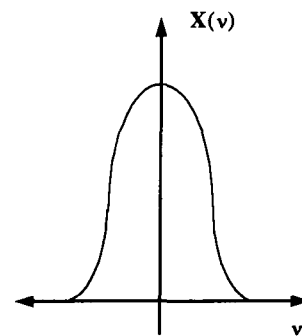
От тук нататък ще означаваме сигналите с малки латински букви, а техните спектри - със съответните главни латински букви. Символът \Leftrightarrow ще използваме за означаване на преобразуването на Фурие.

4.1.1.2. Свиване и теорема на Планшерел

Свиването е операция, която се извършва от измерителните прибори и датчици, в резултат на която се получава размито (неясно) изображение на (представа за) изучавания обект или процес. Един от основните параметри на всяка линейна система, през която преминават сигнали, е импулсната характеристика $h(t)$ - реакцията на системата при подаване на входа ѝ на дираков импулс:



Фиг. 4.1. Аналогов сигнал



Фиг. 4.2. Спектър на аналогов сигнал

$$\delta(t) = \begin{cases} 1, & t = 0 \\ 0, & t \neq 0 \end{cases}$$

При подаване на входа на системата на сигнал $e(t)$ изходният сигнал $s(t)$ се описва от уравнението:

$$s(t) = \int_{-\infty}^{+\infty} e(\tau) \cdot h(t - \tau) \cdot d\tau.$$

За краткост действието “свиване” ще означаваме от тук нататък със звездичка (*):

$$s(t) = e(t) * h(t)$$

Свиването има следните свойства:

а) дистрибутивност:

$$a(t) * [b(t) + c(t)] = a(t) * b(t) + a(t) * c(t);$$

б) комутативност:

$$a(t) * b(t) * c(t) = a(t) * c(t) * b(t) = c(t) * b(t) * a(t);$$

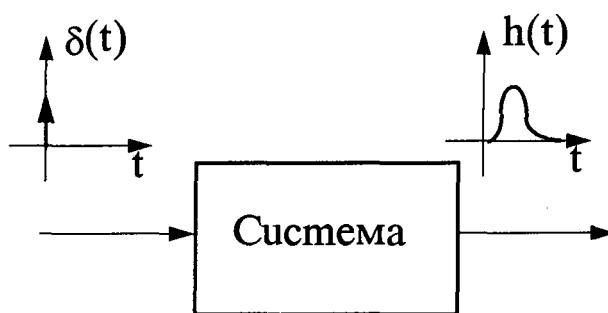
в) асоциативност:

$$[a(t) * b(t)] * c(t) = a(t) * [b(t) * c(t)] = a(t) * b(t) * c(t).$$

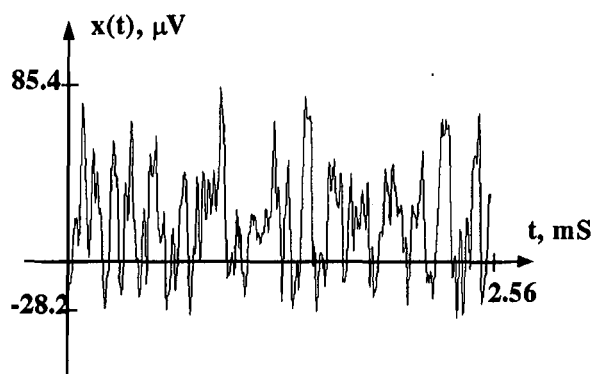
Съгласно теоремата на Планшерел преобразуването на Фурие на свиването на две функции е равно на произведението на спектрите на тези функции:

$$e(t) * h(t) \Leftrightarrow E(\nu) \cdot H(\nu) \text{ и обратно } E(\nu) * H(\nu) \Leftrightarrow e(t) \cdot h(t).$$

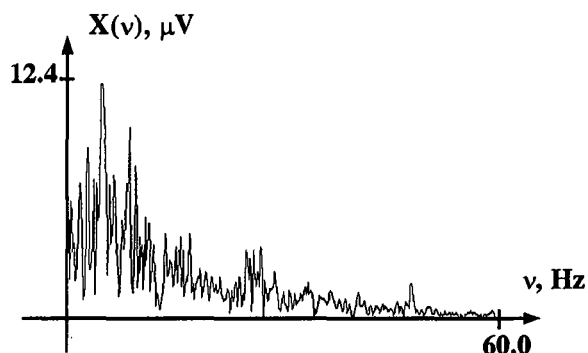
В литературата освен “свиване” се използват и термините “свертка” (от руски език) и “конволюция” (от английски език).



Фиг. 4.4. Импулсна характеристика



Фиг. 4.5. Електроенцефалограма на пълх



Фиг. 4.6. Спектър на сигнала от фиг. 4.5



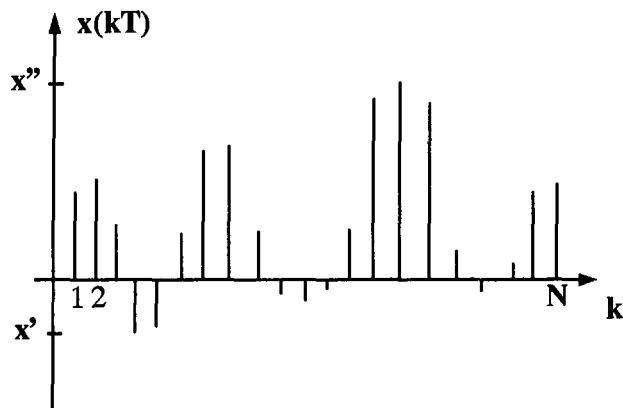
Пример за аналогов сигнал

На фиг. 4.5 е дадена електроенцефалограма на плъх, снета посредством имплантирани вътре в черепа електроди, а на фиг. 4.6 - амплитудния спектър на същия сигнал.

4.1.2. Дискретен сигнал

4.1.2.1. Дефиниране на дискретен сигнал

Дискретният сигнал се описва от гребеновидна функция $x(kT)$, която е ограничена по амплитуда и приема непрекъснати стойности в интервала $x' \leq x \leq x''$. Независимата променлива k приема само дискретни целочислени значения $k = 0, 1, 2, \dots, T$ се нарича период на дискретизация, а $F_s = \frac{1}{T}$ - честота на дискретизация. Дискретният сигнал изглежда примерно така, както е показано на фиг. 4.7.



Фиг. 4.7. Дискретен сигнал

4.1.2.2. Дискретизираща функция

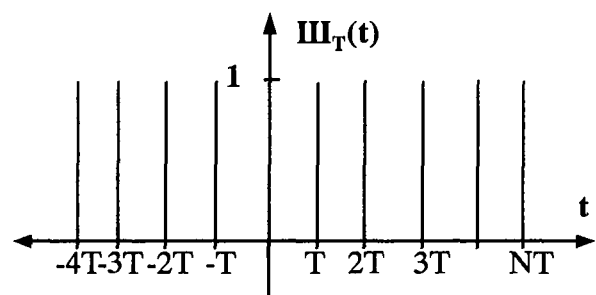
Функцията, която ще използваме за математическо моделиране на процеса на дискретизация, е гребеновидната функция $\text{Ш}_T(t)$, дефинирана както следва:

$$\text{Ш}_T(t) = \sum_{k=-\infty}^{+\infty} \delta(t - kT),$$

където $\delta(t)$ е дираков импулс. Гребеновидната функция $\text{Ш}_T(t)$ изглежда както е показано на фиг. 4.8. Спектърът на функцията $\text{Ш}_T(t)$ е също гребеновидна функция:

$$\text{Ш}_T(t) \Leftrightarrow F_s \cdot \text{Ш}_{F_s}(\nu),$$

или по-подробно:



Фиг. 4.8. Дискретизираща функция

$$\sum_{k=-\infty}^{+\infty} \delta(t - kT) \Leftrightarrow F_s \cdot \sum_{n=-\infty}^{+\infty} \delta(\nu - nF_s)$$

и изглежда както е показано на фиг. 4.9. (*n* е цяло число)

4.1.2.3. Процес на дискретизация

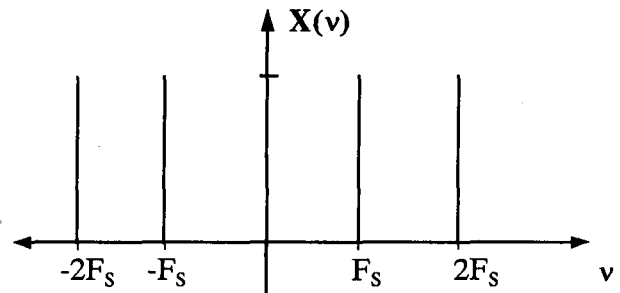
Дискретизация е процесът на получаване на дискретен от аналогов сигнал. Математически се описва като умножение на сигнала $x(t)$ с гребеновидната функция $\text{Ш}_T(t)$:

$$x(kT) = x(t) \cdot \text{Ш}_T(t).$$

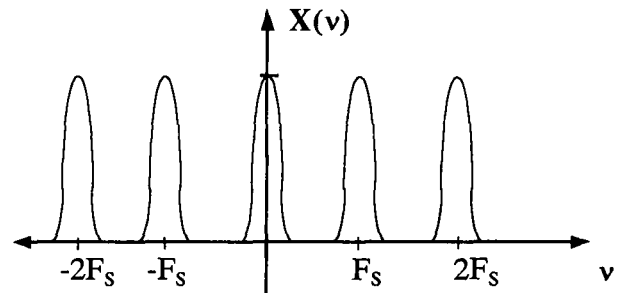
След преобразуване на Фурие и използване на теоремата на Планшерел се получава за спектъра на дискретизирания сигнал:

$$x(kT) \Leftrightarrow X(\nu) * [F_s \cdot \text{Ш}_{F_s}(\nu)].$$

От този израз следва, че спектърът на сигнала $x(kT)$ е периодичен, с период $F_s = \frac{1}{T}$, както е показано на фиг. 4.10.



Фиг. 4.9. Спектър на дискретизиращата функция



Фиг. 4.10. Спектър на дискретизиран сигнал



Основният извод от математическото описание на процеса на дискретизация е, че той довежда до “размножаване” на спектъра $X(\nu)$ на аналоговия сигнал $x(t)$ по оста на честотите с период F_s .

4.1.2.4. Къде се извършва дискретизацията

В реалните системи дискретизацията се извършва от схемата следене/запомняне. Именно тя запомня (фиксира) стойностите на входния аналогов сигнал $x(t)$ в моментите $0, T, 2T, \dots$, които се задава от управляващата блока АЦП микропроцесорна система.



Примери за дискретен и дискретизиран сигнали

Дискретен по своята същност е сигнал, който приема стойност през определени интервали и остава постоянен в тях. Пример за един

такъв сигнал е фиксингът на БНБ за курса на лева спрямо чуждестранните валути.

Дискретизиран сигнал е аналогов (непрекъснат) сигнал, чиито стойности знаем само в определени моменти от време, т.е. над него е извършен процес на дискретизация. Пример за един такъв сигнал е сведението за нивото на река Дунав на пристанище Русе. Нивото на реката на пристанището се изменя непрекъснато, плавно, но ежедневните сведения по радиото ни дават само стойностите на нивото измерено по обяд.

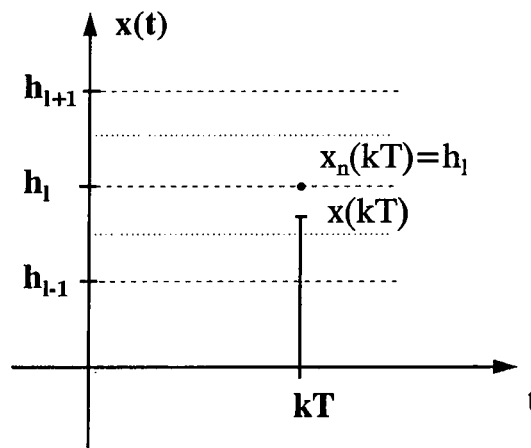
4.1.3. Квантуван сигнал

4.1.3.1. Дефиниране на квантуван сигнал

Квантуваният сигнал $x_n(t)$ е ограничен във времето сигнал ($t' \leq t \leq t''$), който приема само определени дискретни стойности (нива) - h_1, h_2, \dots, h_n . За нас особен интерес представляват квантуваните дискретизирани сигнали, които се описват от гребеновидната функция $x_n(kT)$, където независимата променлива k приема само дискретни целочислени значения $k = 0, 1, 2, \dots$, а $x_n \in \{h_1, h_2, \dots, h_n\}$. Сигналът $x_n(kT)$ ще приеме стойност h_l когато:

$h_l - \frac{h_l - h_{l-1}}{2} \leq x_n(kT) < h_l + \frac{h_{l+1} - h_l}{2}$,

т.е. най-близкото до неговата моментна стойност ниво, както е показано на фиг. 4.11.



Фиг. 4.11. Квантуване

4.1.3.2. Къде се извършва квантуването

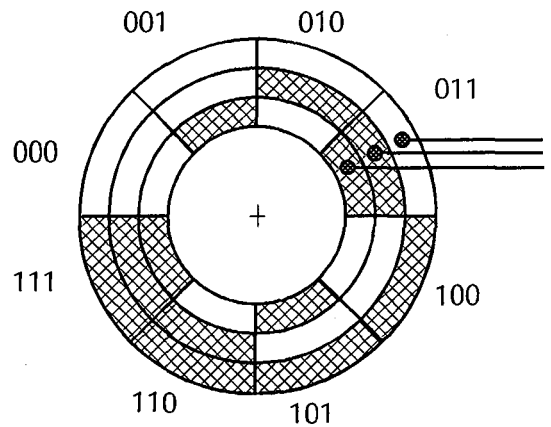
В реалните системи квантуването по амплитуда се извършва от аналогово-цифровия преобразувател. Той измерва фиксираната (дискретизираната) от схемата следене/запомняне моментна стойност на сигнала и подава към компютърната система число - номера на нивото, съответстващо на амплитудата на сигнала.

4.1.3.3. Номериране на нивата за квантуване

В една от предишните глави беше показано как от числото, получено от АЦП, да се изчисли входната величина. Тези формули са валидни

при еднакво разстояние между нивата на квантуване и присвояване на последователни номера на нивата от най-малкото към най-голямото. Това е и най-често срещания случай.

Понякога се налага номерата на нивата да са “разбъркани”, т.е. кодирани по различен начин. Нека вземем за пример един датчик за ъглово положение на вал, става на манипулатора на робот и т.н. Той представлява група от оптодвойки (в нашия случай три), наредени в една линия. Между двойките светодиод - фототранзистор е поставена пластина, свързана с вала, чието ъглово положение ще измерваме. На пластината са нанесени тъмни и светли участъци, както е показано на фиг. 4.12. Изходите на оптодвойките



Фиг. 4.12. Датчик за ъглово положение на вал

дават ъгловото положение на вала, квантувано на 2^N нива, където N е броят на оптодвойките. Тази конструкция на ъгловия датчик може да започне да дава грешни показания, когато валът е завъртян точно на границата на смяна на нивата - например между 111 и 000. В следствие на малко потрепенване или вибрация, температурна нестабилност и др., всяка от оптодвойките може да дава на изхода си ту логическа единица, ту логическа нула. При този най-тежък случай показанието на датчика за ъглово положение може да бъде произволно и различно във всеки момент от времето. За да се избегне недостатъкът на това квантуващо ъглово положение устройство е удачно да се използва друго нареждане (номерирание) на нивата (ъгъла на завъртане). Изключително удобен тук е кодът на Грей, при който на всяко следващо ниво (ъгъл) се присвоява число, различаващо се само с един

номер на нивото	0	1	2	3	4	5	6	7
код на нивото	000	001	011	010	110	111	101	100

двоичен разряд, както е показано в таблицата. При този начин на изработка на

пластината датчикът винаги ще дава показания с точност ± 1 разряд, което е нормално при процесите на квантуване и вярно определя положението на вала.

Когато за кодиране на номера на нивото са необходими например дванадесет двоични разряда, а за съхраняване на един отчет се използват шестнадесет двоични разряда (два байта) в някои случаи е оправдано използването на т.нар. кодиране с излишък. При него номерата на нивата се кодират с част от възможните стойности на едно двубайтово число, определени по такъв начин, че при предаване на данните по комуникационна линия (с потенциално въвеждане на грешки) да може да се отделят с най-голяма вероятност кодовете при които има допусната грешка. При част от начините за кодиране с излишък е възможно и възстановяване на най-вероятната истинска стойност на кода.



Примери за квантувани сигнали

Всички измерени с определена точност сигнали така или иначе са квантувани. Нивото на река Дунав в ежедневните сводки е квантувано с точност до сантиметър. Всички въведени в компютърна система измерени величини и сигнали също са квантувани, защото компютрите винаги работят с числа с определена точност.

4.2. Проблем за избор на честотата на дискретизация

4.2.1. Теорема за дискретизацията

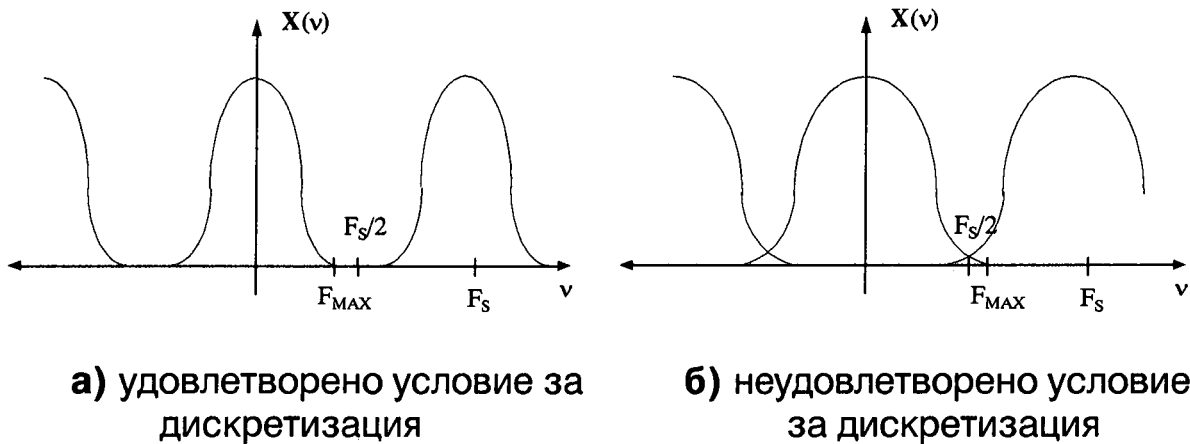
Ако е даден сигналът $x(t)$ за чиито спектър $X(\nu)$ е известно, че $X(\nu) \equiv 0$ за всяко $|\nu| \geq F_{max}$, то необходимо и достатъчно условие периодичното повторение на спектъра на сигнала, предизвикано от дискретизирането с честота F_s , да не измени повтарящия се спектър е $F_s \geq 2 \cdot F_{max}$.

Графично теоремата за дискретизацията е показана на фиг. 4.13, като на фиг. 4.13.а е показана сигнал удовлетворяващ условието за дискретизация, а на фиг. 4.13.б - сигнал, който не удовлетворява условието за дискретизация.

Максималната честота в спектъра на електроенцефалограмата, дадена като пример в точка 1.1, е около 60 Hz. Това означава, че този сигнал трябва да бъде въведен в паметта на компютърната система с честота на дискретизация по-голяма от 120 Hz.

В различни области на инженерната дейност до условието за дискретизация без загуба на информация са стигали в своето време различни специалисти. Затова теоремата за дискретизацията е известна в различните източници под различни имена - теорема на Котелников (работил в областта на предаването на информация чрез модулация),

теорема на Найкуист (работил в областта на автоматичните регулатори) и теорема на Шенон - американски учен от френски произход, от когото е доказана теоретично най-строго през четридесетте години на XX век.



Фиг. 4.13. Теорема за дискретизация на сигнали



Основният извод от теоремата за дискретизацията е, че при дискретизиране на сигнал и удовлетворено изискване на тази теорема процесът на дискретизация не предизвиква загуба на информация за аналоговия сигнал!

4.2.2. Апроксимационна формула на Шенон

След като при спазено условие на теоремата за дискретизацията няма загуба на информация трябва да видим начина по който може да възстановим моментните стойности на сигнала $x(t)$ във всеки момент на времето t , като използваме само дискретизираната поредица $x(kT)$.

За целта дефинираме правоъгълната функция $\Pi_{\frac{F_s}{2}}(v)$ както следва:

$$\Pi_{\frac{F_s}{2}}(v) = \begin{cases} 1, & |v| \leq \frac{F_s}{2} \\ 0, & |v| > \frac{F_s}{2} \end{cases}$$

След обратно преобразуване на Фурие тази функция има вида:

$$\mathcal{V} \quad \Pi_{\frac{F_s}{2}}(\nu) \Leftrightarrow F_s \cdot \frac{\sin(\pi \cdot F_s \cdot t)}{\pi \cdot F_s \cdot t}$$

Графично функцията на честотата $\Pi_{\frac{F_s}{2}}(\nu)$ и нейното представяне във времева област са показани на фиг. 4.14.

След като умножим спектъра на дискретизираната функция $x(kT)$, показан на фиг. 4.15, с правоъгълната функция $\Pi_{\frac{F_s}{2}}(\nu)$ от него ще остане

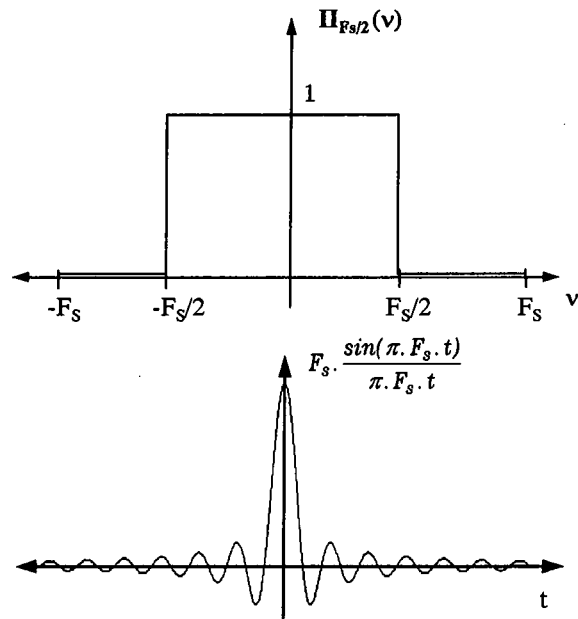
само централната му част - спектъра на сигнала $x(t)$:

$$\left[X(\nu) * F_s \sum_{n=-\infty}^{+\infty} \delta(\nu - n \cdot F_s) \right] \Pi_{\frac{F_s}{2}}(\nu) = X(\nu) \cdot F_s$$

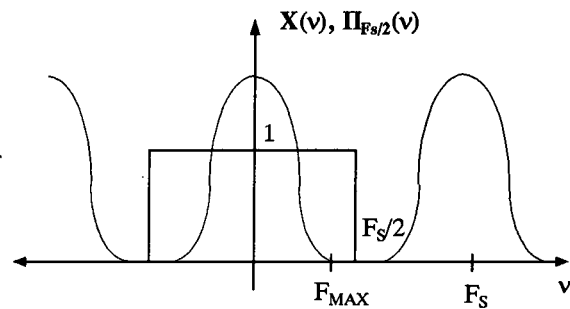
Графично това е илюстрирано на фиг. 4.15. На полученото равенство правим обратно преобразуване на Фурие и получаваме формула за изчисляване на моментните стойности на сигнала на времето $x(t)$ като използваме само дискретизираната поредица $x(kT)$:

$$x(t) = \sum_{k=-\infty}^{+\infty} x(kT) \cdot \frac{\sin(\pi \cdot F_s \cdot (t - kT))}{\pi \cdot F_s \cdot (t - kT)}$$

Тази формула се нарича апроксимационна формула на Шенон. Тя има фундаментално значение, тъй като теоремата за дискретизацията само ни дава условията, при които няма да имаме загуба на информация, докато апроксимационната формула на Шенон ни дава пътя по който да я получим.

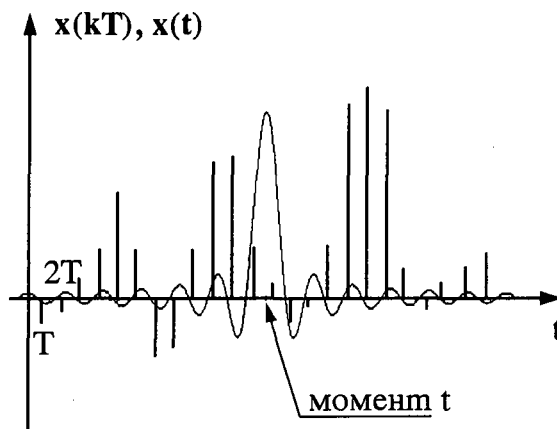


Фиг. 4.14. Правоъгълна функцията на честотата и представянето ѝ във времева област



Фиг. 4.15. Умножение на спектъра на дискретизирания сигнал с правоъгълна функция на честотата

Схематично процесът на получаване на стойността на аналоговия сигнал в момента t е показан на фиг. 4.16. Върху дискретизирания сигнал $x(kT)$ се наслажда функция-та $\frac{\sin(\pi \cdot F_s \cdot (t - kT))}{\pi \cdot F_s \cdot (t - kT)}$ и всеки отчет на дискретизираната поредица $x(kT)$ се умножава със стойността ѝ в момента kT . Сумата от получените произведения ни дава моментната стойност на сигнала $x(t)$.



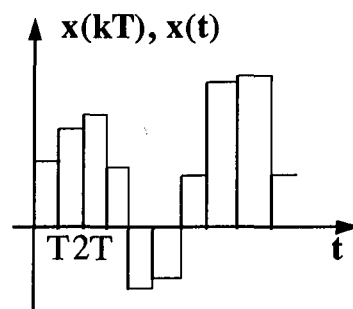
Фиг. 4.16. Апроксимация с формулата на Шенон

4.2.3. Други методи за апроксимация

Използването на апроксимационната формула на Шенон за практически цели е свързано с огромен брой изчисления - за получаването на стойността на сигнала $x(t)$ в един-единствен момент от времето се обработва целият масив от дискретизирани стойности $x(kT)$. В редица случаи е достатъчно използването на по-прости (и съответно по-неточни) методи за апроксимация.

4.2.3.1. Апроксимация от нулев ред

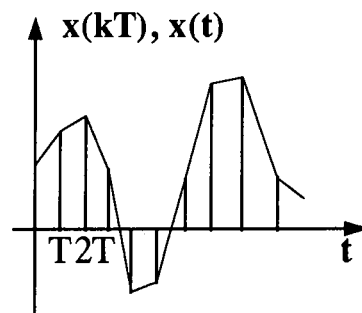
При апроксимацията от нулев ред се приема, че сигналът $x(t)$ е константен и равен на $x(kT)$ за времето от kT до $(k + 1)T$ както е показано на фиг. 4.17. Тази апроксимация с хоризонтални отсечки не изисква практически никакви изчисления. По този начин извършва апроксимацията цифрово-аналоговия преобразувател на изхода на системите за събиране и обработка на информация. През време T микропроцесорната система зарежда в ЦАП число, съответстващо на амплитудата $x(kT)$ на извеждания сигнал. ЦАП поддържа изходното си ниво постоянно до записването на нова стойност.



Фиг. 4.17. Апроксимация от нулев ред

4.2.3.2. Апроксимация от първи ред

Апроксимацията от първи ред свързва стойностите $x(kT)$ и $x((k+1)T)$ с права линия и е в същност линейно-отсечкова апроксимация. Показана е на фиг. 4.18. Изисква минимално време за изчисления и е напълно достатъчна, когато честотата на дискретизация превишава десетина пъти максималната честота в спектъра на сигнала $x(t)$. Апаратно може да бъде изпълнена с ЦАП и интегратор след него.



Фиг. 4.18. Апроксимация от първи ред

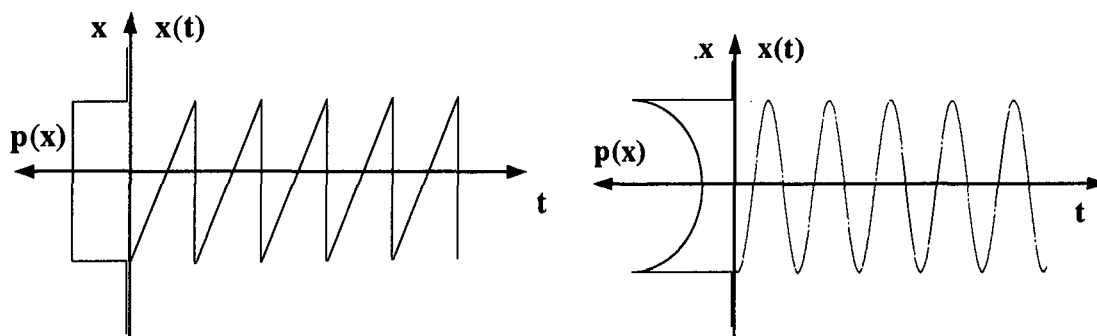
4.3. Избор на броя на нивата за квантуване

След като от теоремата за дискретизацията може да се определи каква да бъде минималната честота на дискретизация за всеки конкретен сигнал без загуба на информация, следва да отговорим на въпроса с каква точност трябва да се извърши квантуването на този сигнал, т.е. колко нива на квантуване трябва да се използват. Обикновено броя на нивата на квантуване се избира точна степен на 2 и определя разрядността на използвания АЦП.

4.3.1. Плътност на вероятност на амплитудата

За аналогия между квантуването и дискретизацията ще използваме понятието плътност на вероятност $p(x)$ на амплитудата на сигнала $x(t)$. При това $p(x) \cdot dx$ е вероятността амплитудата на сигнала $x(t)$ да е в интервала $[x, x + dx]$. Плътността на вероятност на амплитудата на два прости сигнала - тригонообразен и синусоидален, е показана на фиг. 4.19. Обърнете внимание, че колкото е по-голяма скоростта на изменение на амплитудата на сигнала, толкова е по-малка плътността на вероятност на амплитудата. За тригонообразния сигнал, чиято амплитуда се изменя с еднаква скорост, плътността на вероятност на амплитудата е константа. За синусоидалния сигнал плътността на вероятност на амплитудата е най-малка около нулата, където скоростта на изменение на амплитудата е най-голяма.

За плътността на вероятност на амплитудата е валидно равенството $\int_{-\infty}^{+\infty} p(x) \cdot dx = 1$, което преведено на прост език означава, че вероятността амплитудата на сигнала да бъде между $-\infty$ и $+\infty$ е единица.



а) на трионообразен сигнал

б) на синусоидален сигнал

Фиг. 4.19. Плътност на вероятност на амплитудата

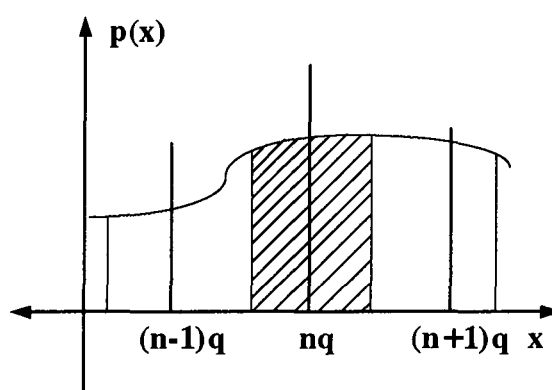
4.3.2. Квантуване и плътност на вероятността на амплитудата

След квантуване посредством равноотгалечени със стъпка q нива вероятността квантуваният сигнал да има амплитуда $n \cdot q$ се дава от израза:

$$p(x' = nq) = \int_{nq - \frac{q}{2}}^{nq + \frac{q}{2}} p(x) \cdot dx = \int_{-\infty}^{+\infty} p(x) \cdot \Pi_q(x - nq) \cdot dx,$$

където $\Pi_q(x) = \begin{cases} 1, & -\frac{q}{2} \leq x < +\frac{q}{2} \\ 0, & x \notin \left[-\frac{q}{2}, \frac{q}{2}\right) \end{cases}$.

Процесът на квантуване довежда до дискретизиране на $p(x)$ със стъпка q и се получава дискретна функция $p(nq)$, всеки от импулсите на която има за амплитуда част от повърхността под кривата $p(x)$, както е показано на фиг. 4.20.



Фиг. 4.20. Квантуване и плътност на вероятност на амплитудата

4.3.3. Теорема за квантуването

Нека $P(\alpha)$ е Фурие образът на $p(x)$. Ако съществува α_{max} такава, че $P(\alpha) \equiv 0$ за всяко $\alpha > \alpha_{max}$ то, съгласно теоремата на Шенон, ако стъпката на квантуване е $q < \frac{1}{2\alpha_{max}}$, може да се възстанови еднозначно $p(x)$ и оттам всичките *статистически* параметри и свойства на сигнала $x(t)$.

От тази теорема можем да определим максималната стъпка на квантуване за всеки конкретен сигнал.

4.3.4. Оценка на строгостта на изискването за квантуване

Теоремата за дискретизацията на аналогови сигнали предявява доста строги изисквания за честотата на дискретизация. Това довежда до получаването на съществен поток от данни при дискретизация на аналогов сигнал. Да видим дали това е така и при теоремата за квантуването. Нека имаме сигнал с плътност на вероятност на амплитудата, описвана от камбанообразната функция на Гаус:

$$p(x) = \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot e^{-\frac{x^2}{2\sigma^2}}.$$

Образът на Фурие на тази функция е:

$$P(\alpha) = e^{-\frac{4 \cdot \pi^2 \cdot \alpha^2 \cdot \sigma^2}{2}}.$$

Тази функция много бързо намалява с нарастването на α и при $\alpha = \frac{4}{2 \cdot \pi \cdot \sigma}$ нейната стойност е само 0,0034. Следователно, с точност

до няколко хилядни, може да се смята че $P(\alpha) \equiv 0$ за $\alpha > \frac{2}{\pi \cdot \sigma}$, което съгласно теоремата за квантуването води до максимална стъпка на квантуване $q = \frac{\pi \cdot \sigma}{4} \approx 0,8 \cdot \sigma$. Като се има пред вид, че за функцията на

Гаус може да се докаже, че в интервала $\pm 3 \cdot \sigma$ се вместиат 99% от стойностите на амплитудите на аналоговия сигнал, се получава за общия брой на нивата за квантуване $n = \frac{6 \cdot \sigma}{0,8 \cdot \sigma} \approx 8$ или триразряден

АЦП! Това означава, че един сигнал с гаусова плътност на вероятност на амплитудата може да бъде твърде грубо квантуван без статистическите му свойства да бъдат сериозно променени.



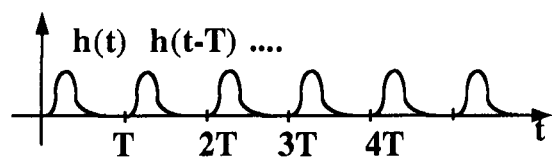
За съжаление възстановяването на точните моментни стойности на грубо квантувания сигнал, което по принцип е възможно, изисква сложни и продължителни математически изчисления. По тази причина въвеждането на аналогови сигнали в системите за събиране и обработване на информация се извършва поне с осемразряден АЦП. Затова от тук нататък ще приемем, че се извършва чиста дискретизация на сигнала и ще пренебрегваме влиянието на квантуването по амплитуда. При превишаване с такъв запас на изискванията на теоремата за квантуването неговото влияние се свежда до появата на т.нар. шум от квантуване с амплитуда равна на половината на стъпката за квантуване.

4.4. Източници на грешка при реална дискретизация

Всички изведени дотук зависимости са теоретични и работят в един идеален свят, в който съществуват безкрайно тесните диракови импулси и сигналите са известни в безкраен период от време. Съществено се различава протичането на процесите на дискретизация в реалната електронна апаратура, при работа в реалния свят в който няма устройства с безкрайно широка честотна характеристика и оттам липсват безкрайно тесни импулси, в света в който инженерът е притиснат от ограниченото количество памет и оттам сигналът за обработка му е известен в ограничен интервал от време.

4.4.1. Дискретизация с периодични импулси с произволна форма

Нека вместо с поредица от диракови импулси да осъществим дискретизацията на сигнала $x(t)$ посредством поредица от импулси с форма $h(t)$, както е показано на фиг. 4.21. За дискретизацията в момента $t = k' \cdot T$ се получава:



Фиг. 4.21. Дискретизираща поредица с произволна форма

$$x'(k'T) = \int_{-\infty}^{+\infty} x(t) \cdot h(t - k'T) \cdot dt = x(t) * h(t)|_{t=k'T}$$

Тъй като в крайна сметка се получават стойности само в моментите kT , то за дискретизираната посредством периодични импулси с произволна форма функция $x'(kT)$ се получава:

$$x'(kT) = [x(t) * h(t)] \cdot \sum_{k=-\infty}^{+\infty} \delta(t - kT),$$

което е еквивалентно на идеална дискретизация на сигнала $x(t) * h(t)$, който има спектър $X(\nu) \cdot H(\nu)$. Това означава, че дискретизираме сигнала $x(t)$, но преминал през филтър с импулсна характеристика $h(t)$.



Уравненията на дискретизацията посредством импулси с произволна форма моделират крайната амплитудночестотна характеристика на схемата следене/запомняне - най-често срещаното дискретизиращо устройство в електронната апаратура. В редица случаи качеството на схемата следене/запомняне определя върното въвеждане на входния сигнал. Затова много фирми, произвеждащи високоточни или високоскоростни АЦП, предпочитат да вградят комплекса входни усилватели, АЦП и схемата следене/запомняне в обща интегрална схема.

4.4.2. Дискретизация с усредняване

Редица случаи на практическо дискретизиране на сигнали се моделират добре посредством дискретизацията с усредняване. При нея един дискрет се получава като средната стойност на входния сигнал $x(t)$ за интервал θ , както е показано на фиг. 4.22. За дискретизацията в момента $t = k' \cdot T$ се получава:

$$x'(k'T) = \frac{1}{\theta} \cdot \int_{k'T}^{k'T+\theta} x(t) \cdot dt = \frac{1}{\theta} \cdot x(t) * \Pi_{\frac{\theta}{2}}(t)|_{t=k'T}.$$

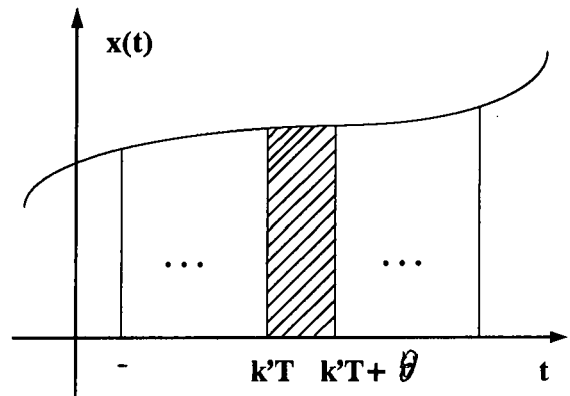
Тъй като в крайна сметка се получават стойности само в моментите kT , то за дискретизираната с усредняване функция $x'(kT)$ се получава:

$$x'(kT) = \frac{1}{\theta} \cdot \left[x(t) * \Pi_{\frac{\theta}{2}}(t) \right] \cdot \sum_{k=-\infty}^{+\infty} \delta(T - kT).$$

След преобразуване на Фурие на сигнала в средните скоби получаваме за спектъра на входния сигнал:

$$X'(\nu) = X(\nu) \cdot \frac{\sin(\pi \cdot \nu \cdot \theta)}{\pi \cdot \nu \cdot \theta} \cdot e^{-2 \cdot \pi \cdot j \cdot \nu \cdot \frac{\theta}{2}}.$$

В крайна сметка след дискретизация с усредняване в компютърната система се въвежда сигналът $x(t)$, но преминал през филтър с честотна характеристика определена от вторите два множителя. Първият множител изменя амплитудите на честотните съставки на $x(t)$, а вторият изменя само фазата им, понеже модулът му е винаги единица.



Фиг. 4.21. Дискретизация с усредняване



Дискретизацията с усредняване описва процесите, протичащи в АЦП от интегриращ тип, понеже при тях се получава резултат равен на средната стойност на входното напрежение за периода на интегриране. Тя също описва и процесите на дискретизация с междинно преобразуване в честота, тъй като честотомерът броя входните импулси за определено време, което е и времето за усредняване. Полезната страна на честотните изкривявания е, че филтърът на дискретизацията с усредняване не пропуска сигнали с честота $\frac{1}{\theta}$ и кратните ѝ. Това свойство се използва в джобните мултиметри, където АЦП е от интегриращ тип. Периодът на интегриране се подбира кратен на мрежовата честота - основен източник на смущения. Това стабилизира показанията на уреда.

4.4.3. Дискретизация на сигнали с крайна продължителност

Да разгледаме сигнала $x_{\Gamma}(t)$, получен от $x(t)$ посредством умножение с правоъгълната функция $\Pi_{\frac{\Gamma}{2}}(t)$:

$$x_{\Gamma}(t) = x(t) \cdot \Pi_{\frac{\Gamma}{2}}(t) = \begin{cases} x(t), & -\frac{\Gamma}{2} \leq t \leq +\frac{\Gamma}{2} \\ 0, & t \notin \left[-\frac{\Gamma}{2}, +\frac{\Gamma}{2}\right] \end{cases}$$

Нека Фурие образът на $x(t) \rightarrow X(\nu) \equiv 0$ за всяко $|\nu| > B$. Тогава:

$$X_{\Gamma}(\nu) = X(\nu) * \Gamma \cdot \frac{\sin(\pi \cdot \nu \Gamma)}{\pi \cdot \nu \Gamma}$$

Тъй като функцията $\frac{\sin(\pi \cdot \nu \cdot \Gamma)}{\pi \cdot \nu \cdot \Gamma}$ не е ограничена в крайна честотна област, то и функцията $X_{\Gamma}(\nu)$ също не е ограничена в крайна честотна област, което съгласно теоремата на Шенон не позволява дискретизирането на сигнала $x_{\Gamma}(t)$ без загуба на информация. На практика този извод отрича възможността да се дискретизира и въведе в паметта на компютърна система сигнал без загуба на информация.

Доказано е, обаче, че въвн от интервала $[-B, +B]$ за сигнал с продължителност Γ е справедливо:

$$|X_{\Gamma}(\nu)| < \frac{1}{B \cdot \Gamma}, |\nu| > B.$$

Оттук може да се докаже, че средноквадратичната грешка при възстановяване на моментните стойности на сигнала $x_{\Gamma}(t)$ от дискретизираните стойности посредством орязаната апроксимационна формула на Шенон:

$$x_{\Gamma}(t) = \left[\sum_{k=-\frac{N}{2}}^{+\frac{N}{2}} x(kT) \cdot \frac{\sin[\pi \cdot (t - kT) \cdot F_s]}{\pi \cdot (t - kT) \cdot F_s} \right] \cdot \Pi_{\frac{\Gamma}{2}}(t)$$

е от порядъка на $\varepsilon \approx \frac{1}{\sqrt{B \cdot \Gamma}}$. В апроксимационната формула $N = \frac{\Gamma}{T}$ е броят на отчетите, въведени в паметта на компютърната система за време Γ .



Дискретизацията на сигнали с крайна дължина моделира процесите на реалното въвеждане на сигнали в паметта на компютърните системи. Беше показано, че не е възможно въвеждане на сигнал с крайна дължина без загуба на информация. Грешката, обаче, може да бъде сведена под всяка зададена граница с увеличаване на дължината на сигнала, което е достатъчно за инженерни цели.

Обобщение

В тази глава бяха дефинирани понятията аналогов, дискретен и квантуван сигнал и бяха моделирани математически основните процеси при въвеждането на сигналите в компютърните системи.

Аналоговият сигнал е ограничена, непрекъсната или непрекъсната по части функция на времето с ограничена продължителност. Всеки аналогов сигнал има честотен спектър - представянето на същия про-

цес в честотна област. Разглеждането на процесите във времевата и в честотната област е еквивалентно. Свиването е процес на замазване на въвежданата информация в следствие ограничената честотна лента на входните устройства.

Дискретният сигнал се описва от гребеновидна функция, приемаща стойности само в дискретни моменти на времето. След дискретизиране (умножаване с гребеновидната дискретизираща функция) от аналоговия сигнал се получава дискретен сигнал.

Квантуваният сигнал приема само определен набор от стойности - нива на квантуване.

За не се получи загуба на информация е необходимо дискретизацията да се извърши с честота поне два пъти по-голяма от най-голямата честота в спектъра на входния сигнал - теорема за дискретизацията. Възстановяването на моментните стойности на сигнала се извършва с апроксимационната формула на Шенон. Възможно е и използването и на апроксимация от нулев и първи ред с известна загуба на точност.

Теоремата за квантуването поставя изискванията към броя на нивата за квантуване за да се запазят статистическите свойства на сигнала. Тези изисквания не са особено силни и затова в повечето разглеждания се приема, че дискретизацията се извършва точно, т.е. не се отчита влиянието на процеса на квантуване на амплитудата.

За моделиране на процесите на реална дискретизация се използват математическите модели за дискретизация с импулси с произволна форма (ограничената честотна лента на схемата следене/запомняне), за дискретизация с усредняване (интегриращи АЦП и АЦП с междинно преобразуване в честота) и за дискретизация на сигнали с крайна дължина (ограничената памет в компютърните системи).



Задачи

1. Начертайте начина на засенчване на пластината от фиг. 4.12 при използване на код на Грей за номерата на нивата.

2. Напишете компютърна програма, моделираща апроксимацията на дискретизирани сигнали. Дискретизирайте аналитично десетина периода на един синусоиден сигнал с честота на дискретизация десет пъти по-голяма от честотата му. Изчислете моментните стойности в сто точки на всеки период като използвате:

- апроксимационната формула на Шенон;
- апроксимация от нулев ред;
- апроксимация от първи ред.

Изчислете средноквадратичната грешка при всеки тип апроксимация и засечете времето, за което програмата я извършва. Резултатите погледете в таблица.

3. Намерете разрядността на АЦП за квантуване на тригономичния сигнал от фиг. 4.18 а. Използвайте формулата за получаване на спектъра на правоъгълна функция. Точността да бъде по-добра от 0,05%.

4. След междинно преобразуване в честота в компютърна система се въвежда сигнал, посредством измерване с честотомер с време за броене $\theta = 1 \text{ mS}$. Коя е най-високата честота, която може да бъде въведена с грешка под 0,1%?

5. В компютърна система се въвежда сигнал с честотна лента $B = 4 \text{ KHz}$. Въвеждат се 1024 отчета с честота на дискретизация $F_s = 10 \text{ KHz}$. Какво ще бъде средноквадратичното отклонение при изчисляване на моментните стойности на сигнала?



Контролни въпроси

1. Какво представлява свиването и какви свойства има?
2. Кое устройство извършва дискретизирането в реалните системи?
3. Къде се извършва квантуването на амплитудата на дискретизирания сигнал?
4. Може ли да се въведе в компютърна система сигнал с честотна лента в обхват по-висок от честотата на дискретизация? Ако да, при какви условия?
5. Кое свойство на реалните устройства моделира дискретизацията с импулси с произволна форма?
6. При кои случаи недостатъкът на дискретизацията с усредняване - честотните изкривявания се превръща в предимство?

5. Основни методи за обработка на сигналите



Ключови думи

- преобразуване на Фурие
- теглови функции
- мощност на сигнала
- енергия на сигнала
- автокорелационна функция
- корелационна функция

Какви знания и умения ще придобиете? Къде се прилагат?

В тази тема ще се запознаете с класическите методи за обработка на сигналите: намиране на спектъра на сигнала посредством преобразуването на Фурие, изчисляване на енергия и мощност на сигнала във времева и честотна област, намиране на взаимовръзката между два сигнала посредством изчисляването на корелационната им функция.

Получените знания ще приложите както за получаване на спектър, енергия, мощност и корелационна зависимост на сигнали, така и при създаването на алгоритми за по-сложни обработки на въведените в компютърната система сигнали.

Въведение

След успешното дискретизиране на аналоговия сигнал и въвеждането му в паметта на компютърната система е необходимо да се запознаем с основните, класическите, методи за неговата обработка. Те позволяват:

- ⇒ да извлечем необходимата ни информация и да я представим в друг, по-разбираем вид;
- ⇒ да отстраним въведените шумове;
- ⇒ да изчислим мощност и енергия на сигналите във времева и честотна област.

Класическите методи за обработка на сигналите са мощно средство за съставяне на по-сложни алгоритми за обработка на дискретизираните сигнали.

5.1 Преобразуване на Фурие

5.1.1. Дефиниция на преобразуването на Фурие

Преобразуването на Фурие ни дава спектъра на сигнала и се извършва посредством формулата:

$$X(\nu) = \int_{-\infty}^{+\infty} x(t) \cdot e^{-2 \cdot \pi \cdot j \cdot \nu \cdot t} \cdot dt.$$

$X(\nu)$ се нарича Фурие образ или честотен спектър на $x(t)$. В общия случай е комплексна функция. Прието е обозначението на това действие да става по следните начини:

$$X(\nu) = TF[x(t)] \quad \text{или} \quad X(\nu) \Leftrightarrow x(t).$$

Често се използват понятията амплитуден спектър:

$$|X(\nu)| = \sqrt{\{Re[X(\nu)]\}^2 + \{Im[X(\nu)]\}^2}$$

и фазов спектър:

$$\varphi(\nu) = \text{arctg} \frac{Re[X(\nu)]}{Im[X(\nu)]}$$

където $Re[X(\nu)]$ и $Im[X(\nu)]$ са съответно реалната и имагинерната част на $X(\nu)$, за които след използването на формулата на Ойлер се получава:

$$Re[X(\nu)] = \int_{-\infty}^{+\infty} x(t) \cdot \cos(2 \cdot \pi \cdot \nu \cdot t) \cdot dt$$

$$Im[X(\nu)] = \int_{-\infty}^{+\infty} x(t) \cdot \sin(2 \cdot \pi \cdot \nu \cdot t) \cdot dt.$$

Обратното преобразуване на Фурие позволява да получим формата на сигнала, имащ даден спектър и се дава от израза:

$$x(t) = \int_{-\infty}^{+\infty} X(\nu) \cdot e^{2 \cdot \pi \cdot j \cdot \nu \cdot t} \cdot d\nu.$$

5.1.2. Физически смисъл на преобразуването на Фурие

Функциите $x(t)$ и $X(\nu)$ описват в различна форма един и същ процес. Ако разглеждаме $x(t)$, то процесът се изучава в равнината амплитуда-време. Ако разглеждаме $X(\nu)$, то процесът се изучава в равнината амплитуда-честота. Не е задължително дименсиите да са обезателно за измерване на време. Ако едната функция има за аргумент, на-

пример, величина с дименсия метър, то нейният Фурие образ ще има за аргумент величина с дименсия метър⁻¹.

Преобразуването на Фурие разлага сигнала на сума от синусоидални сигнали с различна честота и фаза (респективно на сума от синусоидални и косинусоидални сигнали в комплексна област). Възможно е да се направи разложение по два периодични сигнала с произволна форма (правоъгълна, тригонообразна и т.н.), дефазирани на 90°. Това е така нареченият обобщен спектър. Избрано е преобразуването на Фурие, т.е. разложение по синуси и косинуси, защото синусите и косинусите са собствени функции на линейния филтър — при преминаването през него не изменят формата си, а само амплитудата. От тази гледна точка са най-удобни за работа.

5.1.3. Условия за съществуването на преобразуването на Фурие

Достатъчно условие за съществуването на преобразуването на Фурие е изпълнението на следните три условия:

- а) $x(t)$ да е ограничена за $t \in [-\infty, +\infty]$;
- б) $x(t)$ да е абсолютно интегрируема в интервала $t \in [-\infty, +\infty]$;
- в) $x(t)$ да има краен брой точки на прекъсване, максимуми и минимума във всеки ограничен времеви интервал.

Тези три условия са достатъчни, но не необходими за съществуването на преобразуването на Фурие. Може да се докаже, че ако $x(t)$ е квадратично сумируема в интервала $[-\infty, +\infty]$, то съществуват правото и обратното преобразуване на Фурие. Квадратичната сумируемост на $x(t)$ означава, че на $x(t)$ и $X(\nu)$ съответствува крайна енергия. Понеже всеки реален сигнал се регистрира в ограничен интервал от време, това условие е винаги изпълнено.

5.1.4. Основни свойства на преобразуването на Фурие

Преобразуването на Фурие има някои по-важни свойства, които често се използват в практиката или илюстрират реални физически процеси:

- а) линейност:

$$TF[a \cdot f(t) + b \cdot g(t)] = a \cdot F(\nu) + b \cdot G(\nu);$$

- б) четност — по-важните случаи, важещи за реални сигнали, са дадени в таблицата;

- в) подобие -

$x(t)$	$X(\nu)$
реална, четна	реална, четна
реална, нечетна	имагинерна, нечетна
реална, произволна	реална част — четна, имагинерна част — нечетна

$$TF[x(a.t)] = \frac{1}{|a|} \cdot X\left(\frac{\nu}{a}\right);$$

з) изместване -

$$TF[x(t - a)] = e^{-2.\pi.j.a.\nu} \cdot X(\nu) = X(\nu) \cdot \cos(2.\pi.a.\nu) - j \cdot X(\nu) \cdot \sin(2.\pi.a.\nu);$$

г) производна — $TF\left[\frac{d^n x(t)}{dt^n}\right] = (2.\pi.j.\nu)^n \cdot X(\nu).$

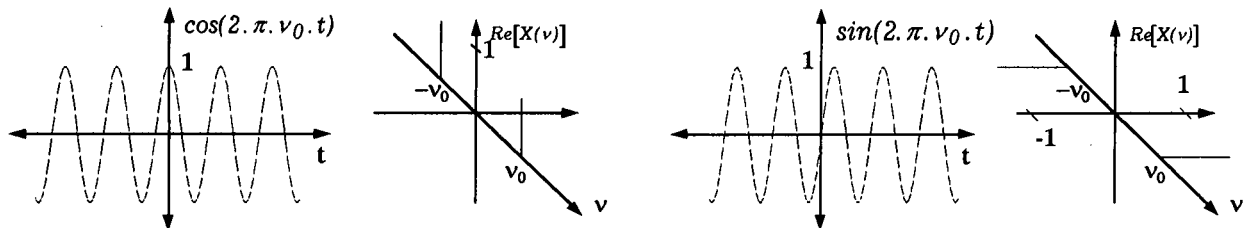


Първите три свойства са очевидни и се извеждат лесно от дефиницията на преобразуването на Фурие. При изместване във времето се предизвиква само изменение на фазовия спектър, понеже множителят $e^{-2.\pi.j.a.\nu}$ има модул единица и завърта само фазата пропорционално на честотата. От петото свойство се вижда, че спектърът на първата производната на сигнала може да се получи като умножим спектъра на сигнала с $2.\pi.j.\nu$. Това увеличава линейно с честотата амплитудата на всички честотни съставки на сигнала. От това следват някои важни изводи. При дискретизиране на реален сигнал обикновено се търси честотата, при която амплитудата на честотните съставки е под някаква минимална стойност. Приема се, че над тази честота спектърът е тъждествено равен на нула, прилага се теоремата на Шенон и се намира минималната честота на дискретизация. Ако искаме да запазим и първата (или и следващите) производна на сигнала, то спектърът на сигнала трябва да се умножи с $2.\pi.j.\nu$ (респективно с $(2.\pi.j.\nu)^n$) и тогава да се търси честотата, над която ще приемем, че спектърът е тъждествено равен на нула. Обикновено тази честота е значително по-висока.

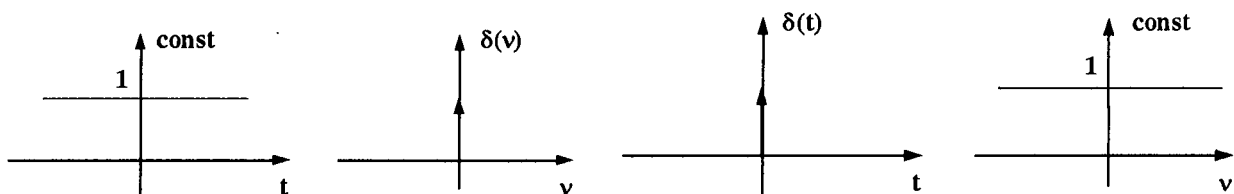
5.1.5. Фурие-образи на някои функции

В таблицата са дадени някои от най-често срещаните сигнали и техните Фурие образи. На следващите фигури те са показани графично. Обърнете внимание на дуализма на преобразуването на Фурие — съответствието между формата на сигнала и спектъра при константата и дираковия импулс, при прозоречната функция и сигнала, описван от функцията $\frac{\sin x}{x}$ и др.

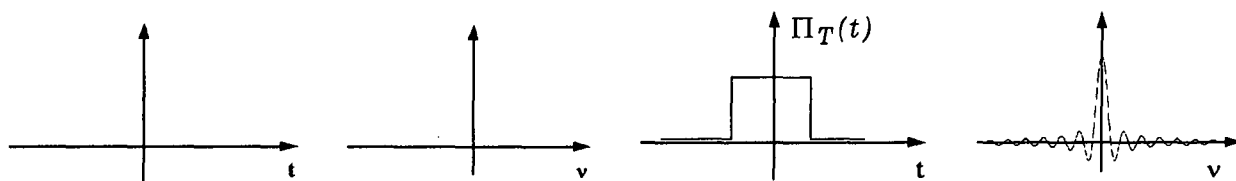
$x(t)$	$X(\nu)$
$\cos(2. \pi. \nu_0. t)$	$\frac{1}{2} \cdot [\delta(\nu - \nu_0) + \delta(\nu + \nu_0)]$
$\sin(2. \pi. \nu_0. t)$	$\frac{j}{2} \cdot [\delta(\nu - \nu_0) - \delta(\nu + \nu_0)]$
$const$	$\delta(\nu)$
$\delta(t)$	$const$
$\frac{\sin(2. \pi. \nu_0. t)}{2. \pi. \nu_0. t}$	$\frac{1}{2. \nu_0} \cdot \Pi_{\nu_0}(\nu)$
$\Pi_T(t)$	$2. T. \frac{\sin(2. \pi. \nu. T)}{2. \pi. \nu. T}$
$\text{III}_T(t) = \sum_{k=-\infty}^{+\infty} \delta(t - kT)$	$\text{III}_{F_s}(\nu) = \sum_{k=-\infty}^{+\infty} \delta(\nu - kF_s), F_s = \frac{1}{T}$
$\Pi_T(t) \cdot \cos(2. \pi. \nu_0. t)$	$2. T. \left[\frac{\sin[2. \pi. T. (\nu + \nu_0)]}{2. \pi. T. (\nu + \nu_0)} + \frac{\sin[2. \pi. T. (\nu - \nu_0)]}{2. \pi. T. (\nu - \nu_0)} \right]$



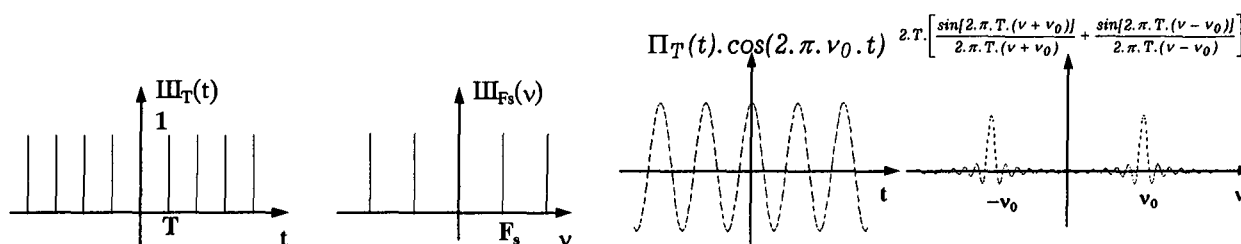
Фиг. 5.1. Форма на сигнала и спектъра на синусоидална функция



Фиг. 5.2. Форма на сигнала и спектъра на константа и дираков импулс



Фиг. 5.3. Форма на сигнала и спектъра на правоъгълна функция и функцията $\sin(x)/x$



Фиг. 5.4. Форма на сигнала и спектъра на дискретизираща функция и на усечена синусоида

5.1.6. Дискретно преобразуване на Фурие

Спектърът на дискретизиран сигнал може да бъде изчислен за всяка стойност на честотата ν_0 . Това се нарича дискретно преобразуване на Фурие и се дава от формулата:

$$X(\nu_0) = \sum_{k=-\frac{N}{2}}^{+\frac{N}{2}} x(kT) \cdot e^{-2\pi \cdot j \cdot \nu_0 \cdot k \cdot T},$$

където N е броят на дискретите. При реално изчисляване на спектъра на сигнал след дискретизиране на N точки от него се търсят обикновено N точки от неговия спектър в обхвата $-\frac{F_s}{2}, +\frac{F_s}{2}$. Това означава, че се дискретизира и самият спектър на сигнала в N точки със стъпка $\Delta\nu = \frac{F_s}{N}$. От дискретизирания по този начин спектър можем да намерим стойността на сигнала за всеки момент от време. Това се нарича дискретно обратно преобразуване на Фурие и за него е справедлива формулата:

$$x(t) = \sum_{n=-\frac{N}{2}}^{+\frac{N}{2}} X(n, \Delta\nu) \cdot e^{2 \cdot \pi \cdot j \cdot n \cdot \Delta\nu \cdot t}.$$

5.1.7. Алгоритмично реализиране на дискретното преобразуване на Фурие

Формулите за правото и обратното преобразуване на Фурие са свързани с голямо количество изчисления. Броят им, и респективно времето за изчисления, нараства с квадрата на броя на точките N . За ускорено получаване на правото и обратното преобразуване на Фурие са разработени множество бързи алгоритми. Най-популярен сред тях е бързото преобразуване на Фурие (Fast Fourier Transformation — FFT) — алгоритъм, разработен от Кули и Тюки през 1968 г. При него броят на изчисленията нараства пропорционално на $N \cdot \log_2 N$, т.е. приблизително линейно с броя на точките. Този алгоритъм е основен при написването на компютърни програми за спектрален анализ и е описан подробно в почти всички източници за обработка на сигнали. Затова няма да се спираме подробно на принципите на реализацията му. При зададени N точки от сигнала бързото преобразуване на Фурие дава N точки от спектъра му и обратно. Недостатък на алгоритъма е, че изисква броят на точките да е точна степен на двойката. Известен е също така и алгоритъмът на Виноградов, който е разработен за произволен брой точки и е малко по-бърз от алгоритъма на Кули и Тюки.

Често намира приложение преобразуването на Уолш, което е разложение на сигнала на правоъгълни функции с амплитуда единица. Спектърът на Уолш се получава само посредством операции събиране и изваждане (без умножения и тригонометрични функции!) и времето за изчисляване е много по-малко. След това от спектъра на Уолш се прави преход към спектъра — разложение по синусоидални функции. По аналогичен начин са разработени и алгоритмите на бързото преобразуване на Уолш.

5.1.8. Използване на теглови функции при преобразуването на Фурие

Умножаването на $x(t)$ с правоъгълната функция $\Pi_{\theta}(t)$ при работа

с ограничен временен интервал $-\frac{\theta}{2}, +\frac{\theta}{2}$ от сигнала $x(t)$ довежда до изменения в спектъра му и появата на амплитудни и фазови изкривявания. Сравнете идеалния спектър на синусоидален сигнал от фиг. 5.1 със спектъра на усечената във времето синусоида от фиг. 5.4. Дираковият импулс на честота ν_0 от фиг. 5.1 е заменен при спектъра на усечената синусоида със спектъра на правоъгълната функция $\Pi_{\theta}(t)$, който е по-

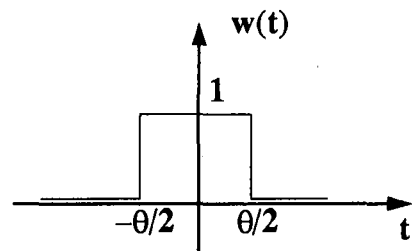
казан на фиг. 5.3. За частично компенсирание на изкривяванията на спектъра на сигнала се използва замяна на правоъгълната функция с друга, чиито спектър е по-подходящ за конкретната обработка. Тази функция се нарича теглова функция. Сигналят $x(t)$ се умножава с тегловата функция $w(t)$ и след това се извършва преобразуването на Фурие:

$$x(t) \cdot w(t) \Leftrightarrow X(\nu) * W(\nu).$$

Известни са много теглови функции, най-често използваните които са:

а) естествена теглова функция, показана на фиг. 5.5 и дефинирана от израза:

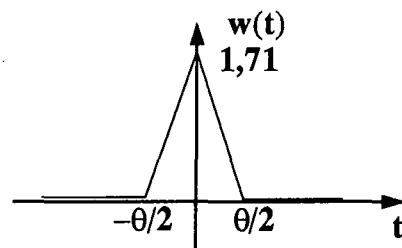
$$w(t) = \begin{cases} 1, & |t| \leq \frac{\theta}{2} \\ 0, & |t| > \frac{\theta}{2} \end{cases}$$



Фиг. 5.5. Естествена теглова функция

б) триъгълна теглова функция (функция на Бартлет), показана на фиг. 5.6 и дефинирана от израза:

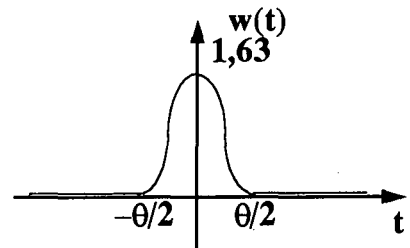
$$w(t) = \begin{cases} A \cdot \left(1 - \frac{2|t|}{\theta}\right), & |t| \leq \frac{\theta}{2}, A = \sqrt{3} \\ 0, & |t| > \frac{\theta}{2} \end{cases}$$



Фиг. 5.6. Триъгълна теглова функция

в) теглова функция на Тюки, показана на фиг. 5.7 и дефинирана от израза:

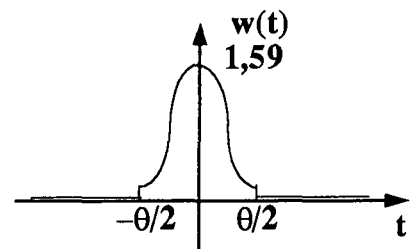
$$w(t) = \begin{cases} \frac{A}{2} \cdot \left(1 + \cos \frac{2 \cdot \pi \cdot t}{\theta}\right), & |t| \leq \frac{\theta}{2}, A = 1,63 \\ 0, & |t| > \frac{\theta}{2} \end{cases}$$



Фиг. 5.7. Теглова функция на Тюки

г) теглова функция на Хеминг, показана на фиг. 5.8 и дефинирана от израза:

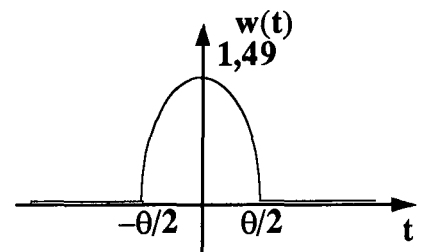
$$w(t) = \begin{cases} \frac{A}{2} \cdot \left(0,54 + 0,46 \cdot \cos \frac{2 \cdot \pi \cdot t}{\theta}\right), & |t| \leq \frac{\theta}{2}, A = 1,59 \\ 0, & |t| > \frac{\theta}{2} \end{cases}$$



Фиг. 5.8. Теглова функция на Хеминг

г) теглова функция на Макс, Фок и Бертие, показана на фиг. 5.9 и дефинирана от израза:

$$w(t) = \begin{cases} A \cdot \frac{\sin \frac{2 \cdot \pi \cdot t}{\theta}}{\frac{2 \cdot \pi \cdot t}{\theta}}, & |t| \leq \frac{\theta}{2}, A = 1,49 \\ 0, & |t| > \frac{\theta}{2} \end{cases}$$



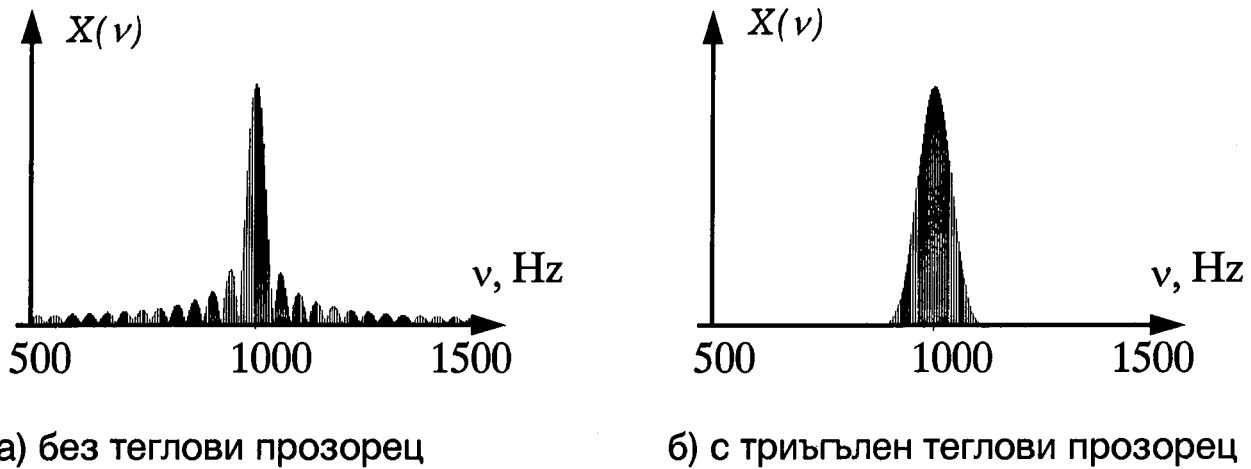
Фиг. 5.9. Теглова функция на Макс, Фок и Бертие

Коефициентите A са така разчетени, че да се запази общата енергия на сигнала. Всяка теглова функция внася свои специфични изкривявания в спектъра и се използва при различни случаи.



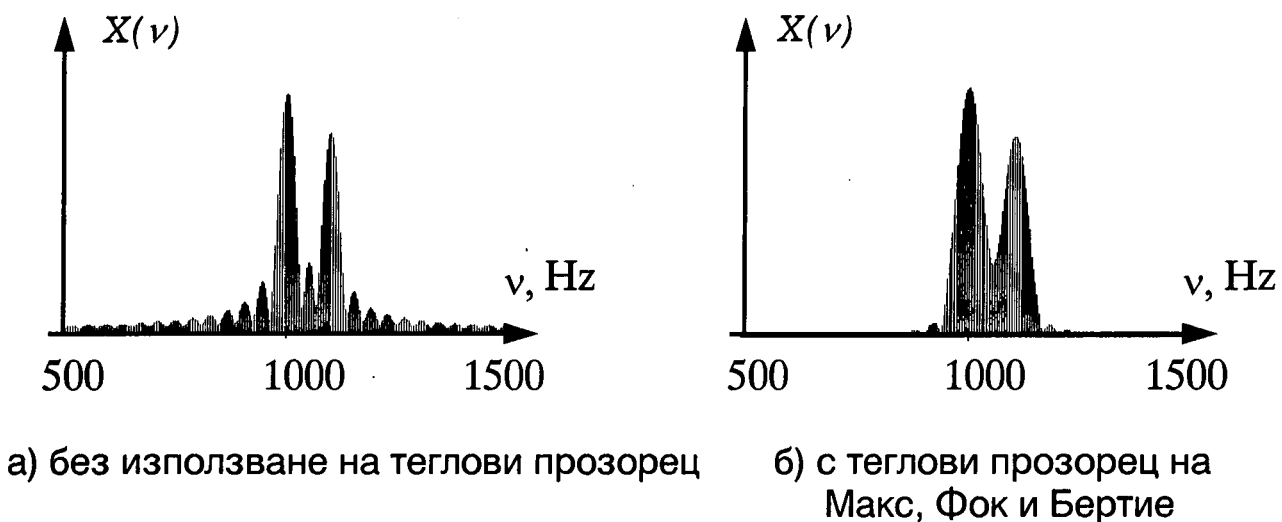
Примери за използване на теглови функции

Нека имаме синусодален сигнал с честота 1 000 Hz, дискретизиран в 256 точки с честота 10 000 Hz. На фиг. 5.10 са показани спектрите на сигнала без теглови прозорец и с използването на триъгълен теглови прозорец. Първият спектър е подобен на теоретично изведения на фиг. 5.4. При втория спектър единичната спектрална линия изпъква по-ясно. И двата спектъра са увеличени и са показани само в интервала от 500 до 1 500 Hz.



Фиг. 5.10. Спектри на синусоидален сигнал с честота 1 000 Hz

На фиг. 5.11 са показани спектрите на сигнал получен от сумата на две синусоиди — с честота 1 000 Hz и амплитуда 10 V, с честота 1 100 Hz и амплитуда 8 V. Сигналът е дискретизиран с честота 10 000 Hz, като от него са записани 256 отчета. Първият спектър е без използването на теглови прозорец, а вторият — с използването на теглови прозорец на Макс, Фок и Бертие. При втория спектър отчетливо се виждат само двете честотни съставки на входния сигнал. Двама спектъра са увеличени както при предишната фигура.



Фиг. 5.11. Спектри на сума от синусоидални сигнали с честота 1 000 и 1 100 Hz

От посочените примери може да се направи извода, че за изпъкването на единични спектрални линии е удачно използването на теглова-

ма функция на Бартлет, а за разделяне на две близки спектрални линии — тегловата функция на Макс, Фок и Бертие.

5.2. Изчисляване на мощност и енергия на сигнала

5.2.1. Мощност на сигнала

Средната мощност на сигнала от момента t_0 за интервал τ се дава от формулата:

$$P(t_0, \tau) = \frac{1}{\tau} \cdot \int_{t_0}^{t_0+\tau} |x(t)|^2 \cdot dt,$$

като при дискретизиран сигнал средната мощност на сигнала от момента $k_0 \cdot T$ за интервал $n_0 \cdot T$ се дава от формулата:

$$P(k_0 T, n_0 T) = \frac{1}{n_0} \cdot \sum_{i=k_0}^{k_0+n_0} |x(iT)|^2.$$

Моментната мощност (мощността) на сигнала се дава от границата:

$$P(t_0) = \lim_{\tau \rightarrow 0} \left(\frac{1}{\tau} \cdot \int_{t_0}^{t_0+\tau} |x(t)|^2 \cdot dt \right) = |x(t)|^2,$$

или в дискретен вид: $P(kT) = |x(kT)|^2$. ✓

5.2.2. Енергия на сигнала

Енергията на сигнала от момента t_0 за интервал τ се дава от формулата:

$$E(t_0, \tau) = \int_{t_0}^{t_0+\tau} |x(t)|^2 \cdot dt,$$

като при дискретизиран сигнал енергията на сигнала от момента $k_0 \cdot T$ за интервал $n_0 \cdot T$ се дава от формулата:

$$E(k_0 T, n_0 T) = \sum_{i=k_0}^{k_0+n_0} |x(iT)|^2.$$

5.2.3. Мощност на сигнала в честотна област

Спектралната мощност на сигнала за честотен интервал $[\nu_0, \nu_0 + \Delta\nu]$ се дава от формулата:

$$P(\nu_0, \Delta\nu) = \frac{1}{\Delta\nu} \cdot \int_{\nu_0}^{\nu_0 + \Delta\nu} |X(\nu)|^2 \cdot d\nu.$$

Спектралната плътност (честотната мощност) на сигнала се дефинира от формулата:

$$S_{xx}(\nu) = \lim_{\Delta\nu \rightarrow 0} \left(\frac{1}{\Delta\nu} \cdot \int_{\nu_0}^{\nu_0 + \Delta\nu} |X(\nu)|^2 \cdot d\nu \right) = |X(\nu)|^2.$$

5.2.4. Енергия на сигнала в честотна област

Енергия на сигнала за честотен интервал $[\nu_0, \nu_0 + \Delta\nu]$ се дава от формулата:

$$E(\nu_0, \Delta\nu) = \int_{\nu_0}^{\nu_0 + \Delta\nu} S_{xx}(\nu) \cdot d\nu$$

Равенството на енергиите, независимо от представянето във времевата или честотна област, се описва от формулата:

$$\int_{-\infty}^{+\infty} |x(t)|^2 \cdot dt = \int_{-\infty}^{+\infty} |X(\nu)|^2 \cdot d\nu,$$

което е част от теоремата на Парсевал.

5.3. Корелационна и автокорелационна функции

5.3.1. Автокорелационна функция на сигнал

5.3.1.1. Дефиниране на автокорелационна функция

Автокорелационната функция на сигнала $x(t)$ се дава от израза:

$$C_{xx}(\tau) = \lim_{T \rightarrow \infty} \left(\frac{1}{T} \cdot \int_0^T x(t) \cdot x(t - \tau) \cdot dt \right).$$

Установява се влиянието на пред-историята на процеса и неговото моментално развитие
 Ако $x(t)$ е периодична ф-я, а T е кратен на периода то се получава макс сходство и автокорел. ф-я има макс стойност

5.3.1.2. Теорема на Винер-Хинчин

Теоремата на Винер-Хинчин гласи, че Фурие образът на автокорелационната функция е спектралната плътност на сигнала:

$$S_{xx}(\nu) = \mathcal{TF}[C_{xx}(\tau)].$$

5.3.1.3. Свойства на автокорелационната функция

Основните свойства на автокорелационната функция на сигнала $x(t)$ са следните:

а) автокорелационната функция е четна функция:

$C_{xx}(\tau) = C_{xx}(-\tau)$, откъдето и от теоремата на Винер-Хинчин,

б) има максимум при $\tau = 0$, т.е. $|C_{xx}(\tau)| \leq C_{xx}(0)$, при което $C_{xx}(0)$ е средното значение на квадрата на сигнала, или дисперсията, или квадрата на ефективното значение, или средната мощност на $x(t)$;

в) автокорелационната функция на периодичен сигнал има същия основен период, съдържа всички честоти на основния сигнал и само тези честоти;

г) фазовият спектър на автокорелационната функция е тъждествено равен на нула.



От горните две свойства следва, че автокорелационната функция съхранява честотната информация и губи фазовата.

5.3.2. Корелационна (взаимнокорелационна) функция на два сигнала

5.3.2.1. Дефиниране на корелационната функция на два сигнала

Корелационната (взаимнокорелационната) функция на сигналите $x(t)$ и $y(t)$ се дава от израза:

$$C_{xy}(\tau) = \lim_{T \rightarrow \infty} \left(\frac{1}{T} \cdot \int_0^T x(t) \cdot y(t - \tau) \cdot dt \right).$$

5.3.2.2. Свойства на корелационната функция

Основните свойства на корелационната функция на сигналите $x(t)$ и $y(t)$ са следните:

а) $C_{xy}(\tau) = C_{yx}(-\tau)$;

б) $|C_{xy}(\tau)| \leq \frac{1}{2} \cdot [C_{xx}(0) + C_{yy}(0)]$;

в) ако двете функции са независими, то тяхната корелационна функция е тъждествено равна на нула;

г) корелационната функция на две функции с един и същи период е периодична функция със същия период;

г) корелационната функция на две функции с различни периоди е периодична функция с основен период равен на периода на биенето ($T = m \cdot T_1 = n \cdot T_2$, където m и n са минималните цели числа, удовлетворяващи равенството), т.е. с основна честота равна на разликата от двете честоти;

е) корелационната функция на произволна периодична функция $x(t)$ с гребеновидна функция със същия период е същата периодична функция $x(t)$;

ж) дистрибутивност:

ако $x(t) = a(t) + b(t)$, то $C_{xx}(\tau) = C_{aa}(\tau) + C_{bb}(\tau) + C_{ab}(\tau) + C_{ba}(\tau)$.

5.3.3. Физически смисъл и основни приложения

5.3.3.1. Физически смисъл на корелационната функция

Терминът “релация” означава връзка, подобие. В такъв случай “корелация” означава взаимна връзка, взаимно подобие. Стойността на корелационната функция $C_{xy}(\tau)$ е толкова по-голяма, колкото повече си приличат (са подобни) сигналите $x(t)$ и $y(t)$, разместени по времева ос на време τ . Това означава, че функцията $C_{xy}(\tau)$ ще има максимуми там, където двата сигнала си приличат най-много и минимуми там, където двата сигнала са максимално различни — в противоположна фаза, например. Оттук и свойството **B**), че корелационната функция на два независими сигнала е тъждествено равна на нула. Основните приложения на корелационната и автокорелационната функции са свързани със свойството им да отстраняват шумовете. Това се дължи на факта, че автокорелационната функция на случайните сигнали (шумовете) клони към нула при достатъчно голямо време за интегриране и те не са подобни с никой друг сигнал.

5.3.3.2. Основни приложения

а) Откриване на периодичен сигнал на фона на шум

Ако $x(t) = a(t) + b(t)$, където $a(t)$ е търсеният периодичен сигнал, а $b(t)$ — шум, то $C_{xx}(\tau) = C_{aa}(\tau) + C_{bb}(\tau) + C_{ab}(\tau) + C_{ba}(\tau)$ и:

◆ понеже сигналът и шумът са независими $C_{ab}(\tau) \rightarrow 0$ и $C_{ba}(\tau) \rightarrow 0$;

◆ $C_{bb}(\tau) \rightarrow 0$ за достатъчно голямо време за интегриране T ,

в следствие на което $C_{xx}(\tau) \approx C_{aa}(\tau)$.

б) Подобряване съотношението сигнал/шум

При изчисляване на автокорелационната функция на сигнал с честотна лента B , време за интегриране T и съотношение сигнал/шум R подобряването на съотношението сигнал/шум се дава от формулата:

$$G = \frac{2 \cdot B \cdot T}{2 + \frac{1}{R}},$$

където G е подобряването на съотношението сигнал/шум.

в) Намиране на сигнал с известен период на фона на шум

Нека търсим сигнал с известен период $a(t)$ на фона на шум $b(t)$. За целта се търси корелационната функция на сигнала $x(t) = a(t) + b(t)$ с опорен (спомогателен) сигнал $c(t)$ с основен период като на $a(t)$, при което $C_{xc}(\tau) = C_{ac}(\tau) + C_{bc}(\tau)$ и $C_{bc}(\tau) \rightarrow 0$ за достатъчно голямо време за интегриране T , понеже $b(t)$ и $c(t)$ са независими.

г) Измерване на изместването във времето на два сигнала

Нека са дадени сигналите $x(t)$ и $y(t) = k \cdot x(t - \Delta t)$. Вторият сигнал е получен като първият е изместен във времето на интервал Δt . Това може да е двойката сигнали въздействие на входа на обект и неговата реакция след известно закъснение. Тяхната корелационна функция $C_{xy}(\tau)$ ще има максимум при $\tau = \Delta t$. При това съществено се редуцира влиянието на въведените в $y(t)$ шумове и смущаващи сигнали.



При използването на корелационния анализ за търсене на взаимовръзка (причина — следствие) трябва особено внимание да се обърне на това дали анализираният сигнал не са следствие от друг, много по-силен фактор. Ако изчислим корелационната функция на сигналите:

а) ежемесечния брой продадени пасти за зъби в гр. Лом и

б) средномесечното ниво на река Дунав край същия град,

за период от няколко години, може да се окаже, че корелационната им функция има сериозен максимум. Този факт на силна връзка между тези два сигнала може да ни доведе до няколко извода:

- ◇ жителите на гр. Лом мият зъбите си в река Дунав;
- ◇ брегът на реката край града е висок;
- ◇ когато спадне нивото на реката те не могат да си мият зъбите и затова не купуват паста за зъби.

Тези шеговити и явно неверни изводи се дължат на факта, че нивото на реката зависи от валежите, от тях зависи и селскостопанската реколта, оттам материалното благосъстояние на населението на града и способността му да си купува паста за зъби.

Обобщение

В настоящата глава се запознахте с основните, класическите, методи за обработка на дискретизирани сигнали.

Беше дефинирано преобразуването на Фурие като основен метод за получаване на спектъра на сигнала и изброени условията за съществуването му. Физическият смисъл на преобразуването на Фурие е представянето на сигнала като сума от синусоиди с различна честота, амплитуда и фаза. Показани са и спектрите на най-често използваните сигнали. Дефинирани са основните свойства на преобразуването на Фурие.

Алгоритмичното реализиране на преобразуването на Фурие е свързано с голямо количество изчисления. Затова в компютърните програми най-често се използва алгоритъма за бързото преобразуване на Фурие (FFT), разработен от Кули и Тюки, както и негови модификации.

Тегловете прозорци внасят специфични изкривявания в спектъра на дискретизирания сигнал и се използват за по-ясното представяне на спектъра на дискретизирани сигнали. За целта преди преобразуването на Фурие сигналът се умножава по съответната теглова функция.

В тази глава са дефинирани и основните формули за получаване на енергията и мощността на сигнала във времева и честотна област.

Корелационната функция дава взаимовръзката, подобие, на изместени един спрямо друг сигнали. Тя запазва честотната информация и губи фазовата. При изчисляването на корелационната функция се отстраняват чувствително съставките, породени от случайни сигнали — шумовете. Това е и основното приложение на корелационния анализ.



Задачи

1. Използвайки свойството за Фурие образа на производната на сигнал изчислете в какъв честотен обхват диференцира диференцираща верига с времеконстанта $\tau = 10 \text{ mS}$, като се ограничите с точност 5%.

2. Като използвате начина за начертаване на спектри, показан на фиг. 5.1 начертайте Фурие образа на функцията $\sin(2\pi \nu_0 t + \varphi_0)$.

3. Даден е сигнал с честотна лента до 4 КHz и съотношение сигнал/шум 1:1. С честота на дискретизация 10 КHz от него са въведени в компютърна система 1024 отчета. Колко ще бъде съотношението сигнал/шум след изчисляване на автокорелационната му функция?



Контролни въпроси

1. При какви условия съществува преобразуването на Фурие?
2. Защо е избрано преобразуването на Фурие, което е разложение на сума от синусоидални сигнали?
3. Защо и как се използват тегловите функции при изчисляване на спектъра на сигнал?
4. Какво се запазва и какво се губи при изчисляване на автокорелационна функция на сигнал?
5. Какво ни дава корелационната функция на два сигнала?

6. Цифрова филтрация



Ключови думи

- цифрова филтрация
- рекурсивен цифров филтър
- нерекурсивен цифров филтър
- честотна характеристика
- импулсна характеристика
- проектиране на цифров филтър

Какви знания и умения ще придобиете?

Къде се прилагат?

В тази тема ще се запознаете с една от най-често използваните процедури за цифрова обработка на сигналите — цифровата филтрация. Ще можете да направите анализ на цифров филтър, т.е. да получите основните му характеристики — честотна (амплитудна и фазова), импулсна и преходна. Ще научите също така как по зададена честотна характеристика да проектирате цифров филтър.

Получените знания ще приложите при проектирането на цифрови филтри — неразделна част от обработката на дискретизиран сигнал и при създаване на програмното осигуряване на системите за събиране и обработка на информация.

Въведение

В редица случаи в практиката се налага да се отделят или да се подтиснат определени участъци от честотната лента на сигнала. В аналоговата схемотехника това се извършва от филтрите — устройства, реализиращи зададена амплитудночестотна характеристика. С нарастването на стръмността на прехода от зона на пропускане към зона на подтискане се увеличава и редът на аналоговия филтър — броя на честотнозависимите звена (RC и LC групи). Това довежда до повишаване сложността на аналоговия филтър, трудното му проектиране и настройка. Поради стареене на елементите с течение на времето честотните характеристики на аналоговия филтър се изменят.

Недостатъците на аналоговите филтри довеждат до разработката на исторически най-старата процедура за цифрова обработка на сигналите — цифровата филтрация. Добрите параметри на цифровите филтри и бурното развитие на технологията за производство на интегрални схеми отварят широко пътя на цифровите филтри в електронната комуникационна апаратура. По това време цифровия филтър може да бъде разгледан като една “черна кутия” с един аналогов вход и

един аналогов изход. Вътре в нея се извършва дискретизация и квантуване на входния сигнал от АЦП, цифровата му обработка (обикновено апаратно реализирана) и отново превръщане в аналогов сигнал посредством ЦАП.

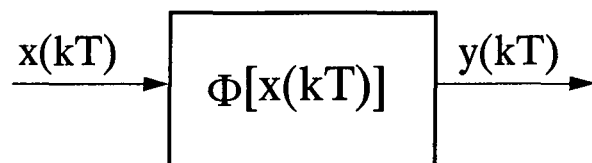
С бурното развитие на компютърните системи и технологията за производство на интегрални схеми цифровата обработка на дискретизирани сигнали става все по-често използван инструмент при анализа на биологични сигнали, говор, вибрации, акустични изследвания и в много други области. Независимо от това и до днес цифровата филтрация е сред най-често използваните в инструментариума на цифровата обработка на сигналите, част от които разгледахме в предишните глави.

6.1. Цифров филтър

6.1.1. Дефиниция за цифров филтър

Цифровият филтър е изчислителна процедура над дискретизиран сигнал, реализираща определена честотна характеристика. По своята същност той представлява една линейна дискретна система (фиг. 6.1), преобразуваща входната редица на дискретизирания сигнал $x(kT)$ в изходна редица $y(kT)$ по зададена функция:

$$y(kT) = \Phi[x(kT)].$$



Фиг. 6.1. Линейна дискретна система

6.1.2. Уравнение на дискретните системи

Линейните дискретни системи с постоянни във времето параметри се описват най-добре с разликкови уравнения:

$$\sum_{j=0}^{M-1} a_j \cdot y[(n-j)T] = \sum_{i=0}^{N-1} b_i \cdot x[(n-i)T],$$

където N и M са постоянни цели числа, a_j и b_i — константи независещи от n , т.е. от времето. След полагане $a_0 = 1$ и трансформиране на горното равенство се получава:

$$y(nT) = \sum_{i=0}^{N-1} b_i \cdot x[(n-i)T] \oplus \sum_{j=1}^{M-1} a_j \cdot y[(n-j)T],$$

т.е. текущата стойност на изходната редица в момента nT се определя от стойностите на:

- входната редица в настоящия момент време;

- входната редица в предходните $N - 1$ момента;
- изходната редица в предходните $M - 1$ момента.

6.1.3. Рекурсивни и нерекурсивни дискретни системи

Ако са известни началните условия на системата $y(-T), y(-2T), \dots, y[-(M - 1)T]$ горното уравнение дава възможност да се изчислят всички стойности на $y(nT)$, където $n = 0, 1, \dots$ за произволна входна поредица $x(nT)$. Системите (филтрите), които се описват с това уравнение се наричат *рекурсивни*. Друг често използван в литературата термин е БИХ филтри — филтри с безкрайна импулсна характеристика. В англоезичната литература терминът е IIR (infinite impulse response).

Когато $a_j = 0$ за всяко $j > 0$ филтърът се описва от уравнението:

$$y(nT) = \sum_{i=0}^{N-1} b_i \cdot x[(n - i)T].$$

Тези филтри се наричат *нерекурсивни* или КИХ филтри (филтри с крайна импулсна характеристика). В англоезичната литература терминът е FIR (finite impulse response).

6.2. Основни характеристики на цифровите филтри

Характеристиките на цифровите филтри са същите както на аналоговите. Разликата е в това, че както входните въздействия, така и изходните реакции са дискретни сигнали.

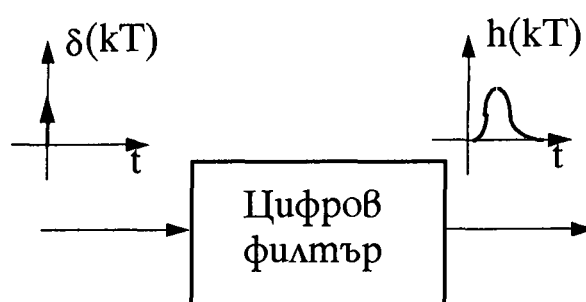
6.2.1. Времени характеристики

6.2.1.1. Импулсна характеристика

Реакцията на филтъра (изходния сигнал) $h(kT)$ при въздействието на единичен дираков импулс $\delta(kT)$ и нулеви начални условия се нарича импулсна характеристика, както е показано на фиг. 6.2.



Обърнете внимание, че при нерекурсивните цифрови филтри:



Фиг. 6.2. Импулсна характеристика на цифров филтър

- ♦ редицата $h(kT)$ е с крайна дължина (филтри с крайна импулсна характеристика!);
- ♦ притежава N елемента;
- ♦ стойностите на редицата са коефициентите на нерекурсивния филтър $h(kT) = b_k, k = 0, 1, \dots, N - 1$.

6.2.1.2. Свиване

Входната поредица $x(kT)$, изходната поредица $y(kT)$ и импулсната характеристика са свързани с формулата на свиването:

$$y(kT) = \sum_{m=0}^{+\infty} h(mT) \cdot x[(k-m)T], \quad x(kT) = 0 \text{ за } k < 0.$$

Тази формула дава възможност да се изчисли изходната поредица при произволна входна поредица и нулеви начални условия.

6.2.1.3. Преходна характеристика

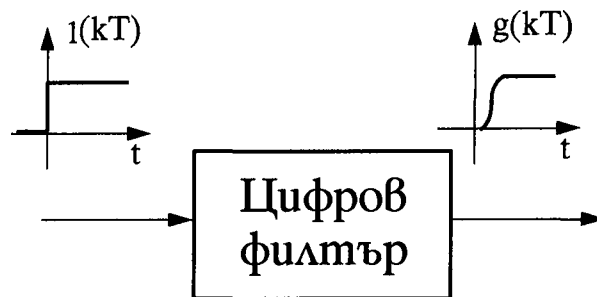
Преходната характеристика $g(kT)$ на цифровия филтър (фиг. 6.3) е реакцията на филтъра при нулеви начални условия и въздействие на единично стъпало, дефинирано както следва:

$$1(kT) = \begin{cases} 1, & k \geq 0 \\ 0, & k < 0 \end{cases}$$

Преходната характеристика може да се изчисли съгласно формулата на свиването:

$$g(kT) = \sum_{m=0}^{+\infty} h(mT) \cdot 1[(k-m)T] = \sum_{m=0}^k h(mT), \quad k \geq 0.$$

Очевидно е, че $h(kT) = g(kT) - g[(k-1)T]$.



Фиг. 6.3. Преходна характеристика на цифров филтър

6.2.2. Устойчивост и реализуемост

Всяка линейна дискретна система е устойчива, ако нейната реакция, т.е. изходната редица $y(kT)$ е ограничена за всяка ограничена входна редица $x(kT)$. Необходимо и достатъчно условие за устойчивост на линейната дискретна система е абсолютната сумируемост на импулсната характеристика:

$$\sum_{m=0}^{+\infty} |h(mT)| < \infty,$$

т.е. редът на отчетите на импулсната характеристика да е абсолютно сходим.

Линейната дискретна система е физически реализуема, ако нейната реакция, изходният дискретизиран сигнал, не изпреварва входния. Това означава, че в момента kT стойността на изходния сигнал $y(kT)$ да зависи само стойностите на входната последователност $x(k'T)$, $k' \leq k$ и изходната последователност $y(k'T)$, $k' < k$. Линейната дискретна система, респективно цифровия филтър, е физически реализуема ако е изпълнено:

$$h(kT) = 0 \text{ за } k < 0.$$

6.2.3. Честотни характеристики

Честотната характеристика филтър ни дава информация сигнали с каква честота пропуска и с каква честота подпуска цифровия филтър. Тя е комплексна функция на честотата. Честотната характеристика на цифровия филтър, а и на всяка линейна дискретна система, се дава от израза:

$$H(\nu) = \frac{\sum_{l=0}^{N-1} b_l \cdot e^{-j.l.2.\pi.\nu.T}}{1 + \sum_{k=1}^{M-1} a_k \cdot e^{-j.k.2.\pi.\nu.T}}.$$

Модулът на тази комплексна функция на честотата се нарича амплитудночестотна характеристика и се дава от израза:

$$A(\nu) = |H(\nu)|,$$

а аргументът — фазочестотна характеристика:

$$\varphi(\nu) = \arg[H(\nu)].$$

Развити тези формули изглеждат така:

$$A(\nu) = \sqrt{\frac{\left[\sum_{l=0}^{N-1} b_l \cdot \cos(l2\pi\nu T) \right]^2 + \left[\sum_{l=0}^{N-1} b_l \cdot \sin(l2\pi\nu T) \right]^2}{\left[1 + \sum_{l=1}^{M-1} a_l \cdot \cos(l2\pi\nu T) \right]^2 + \left[\sum_{l=1}^{M-1} a_l \cdot \sin(l2\pi\nu T) \right]^2}},$$

✓ ?

$$\varphi(\nu) = -\operatorname{arctg} \frac{\sum_{l=0}^{N-1} b_l \cdot \sin(l2\pi\nu T)}{\sum_{l=0}^{N-1} b_l \cdot \cos(l2\pi\nu T)} + \operatorname{arctg} \frac{\sum_{l=1}^{M-1} a_l \cdot \sin(l2\pi\nu T)}{1 + \sum_{l=1}^{M-1} a_l \cdot \cos(l2\pi\nu T)}.$$



За нерекурсивни цифрови филтри отпадат знаменателят на първия израз и втората част на втория израз. Честотната характеристика е Фурие образ на импулсната характеристика на цифровия филтър:

$$H(\nu) = TF[h(kT)].$$

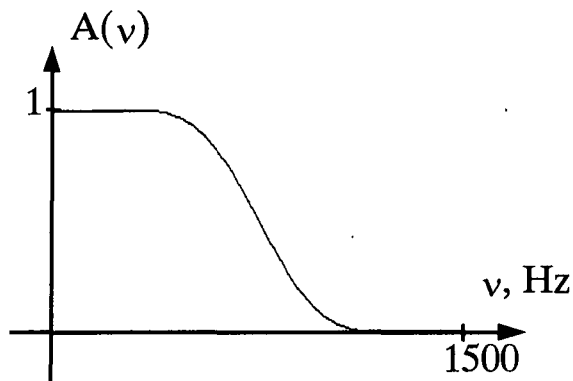


Пример за параметри на цифров филтър

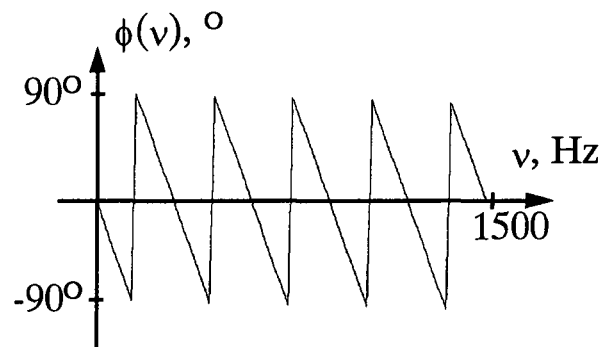
Нека е даден единадесет звенен нерекурсивен цифров филтър с честота на дискретизация 3000 Hz и със следните коефициенти:

$$b_0 = 0.012 \quad b_1 = 0.000 \quad b_2 = -0.062 \quad b_3 = 0.000 \quad b_4 = 0.300 \quad b_5 = 0.500 \\ b_6 = 0.300 \quad b_7 = 0.000 \quad b_8 = -0.062 \quad b_9 = 0.000 \quad b_{10} = 0.012.$$

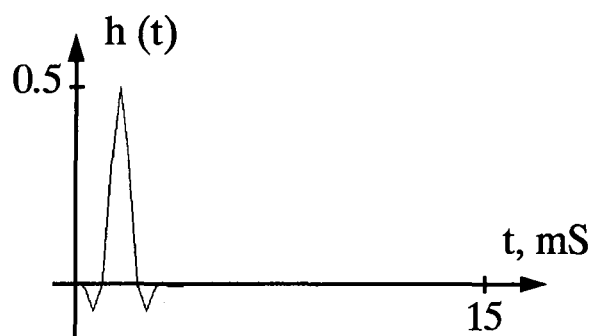
На фиг. 6.4 е показана теоретично изчислената амплитудночестотната му характеристика, а на фиг. 6.5 — фазочестотната му характеристика. На фиг. 6.6 е показан резултатът от симулацията на модела на филтъра с входно въздействие единичен дираков импулс, а на фиг. 6.7 — резултатът от въздействието с единично стъпало. Всички резултати са получени от реално действаща програма за анализ на цифрови филтри.



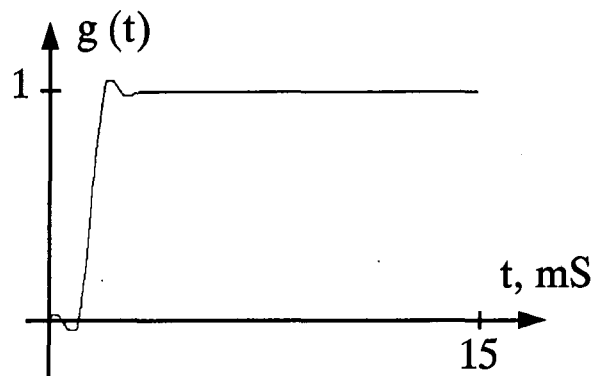
Фиг. 6.4. Теоретично изчислена амплитудночестотна характеристика на цифров филтър



Фиг. 6.5. Теоретично изчислена фазочестотна характеристика на цифров филтър



Фиг. 6.6. Импулсна характеристика на цифров филтър — резултат от симулация



Фиг. 6.7. Преходна характеристика на цифров филтър — резултат от симулация

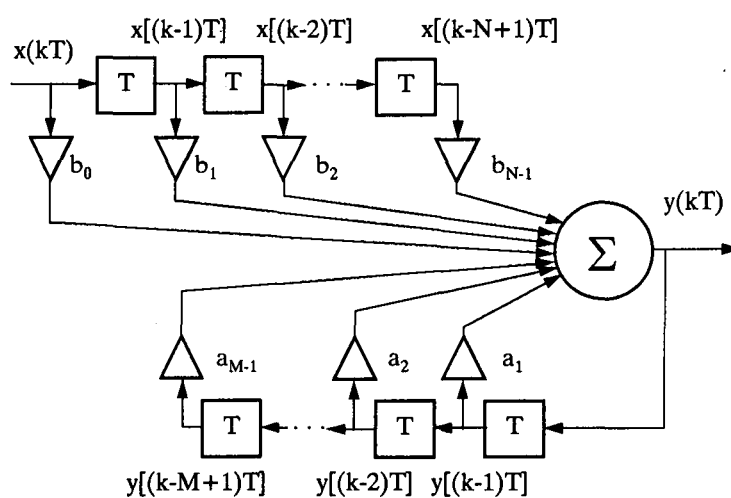
Обърнете внимание и на някои особености. Фазочестотната характеристика на този нерекурсивен цифров филтър е линейна. Наличието на положителни ъгли на фиг. 6.5 е само в резултат от теоретичния анализ — функцията $arctg$, използвана за анализа, връща резултат само в обхвата от -90° до $+90^\circ$. Линејната фазочестотна характеристика означава постоянно време на закъснение на всички честотни съставки на сигнала. Затова и при подаване на входа на филтъра на сигнали с широк спектър — дираковия импулс и единичното стъпало, резултатите са напълно удовлетворителни и този филтър притежава добри времеви параметри.

6.3. Форми на представяне на цифровите филтри

Цифровата филтрация е съвкупност от три прости действия:

- ◇ умножение с константа;
- ◇ закъснение на време T — периода на дискретизация;
- ◇ сумиране на числа.

Освен с формулите, дадени в т. 6.1.2, в литературата много често се прибегва до графично представяне на структурата на цифровите филтри. Пряката или каноничната форма на цифров филтър се изпол-



Фиг. 6.8. Канонична форма на цифров филтър

зва най-често и е показана на фиг. 6.8. Освен това се използва последователната форма за представяне на структурната схема на филтъра — като поредица от последователно свързани елементарни цифрови филтриращи звена. Друга използвана форма на представяне е паралелната, при която входният сигнал се подава едновременно на входовете на няколко филтриращи звена. Изходният сигнал се получава като сума от изходните сигнали на тези филтриращи звена.

6.4. Анализ на цифровите филтри

Анализът на един цифров филтър включва получаването на времевите и честотните характеристики на филтъра от неговите коефициенти.

6.4.1. Теоретичен анализ

От формулите, цитирани в точка 6.2, може лесно да бъде реализирана компютърна програма за теоретичен анализ на цифров филтър от определен ред (числата M и N) със зададени коефициенти. Програмата може да изчисли амплитудночестотната и фазочестотната характеристика на филтъра в обхвата от нула до половината от честотата на дискретизация в произволен брой точки. Резултатите могат да бъдат представени както като графична зависимост честота-коефициент на предаване, така и като диаграма на Найкуист.

6.4.2. Симулация и моделиране на цифровия филтър

Симулацията на цифровия филтър се извършва от компютърна програма, която реализира алгоритъма на работа на цифровия филтър. На входа на програмата се подават числови редици, моделиращи реални сигнали — дираков импулс, единично стъпало, синусоидални сигнали с различна честота. След завършване на симулацията се получава изходната поредица. Тя може да бъде представена графично или във вид на поредица от числа. По този начин могат да бъдат получени времевите характеристики на един цифров филтър — импулсна и преходна.

6.4.2.1. Моделиране без отчитане крайната точност

Когато програмата за моделиране на работата на цифровия филтър работи с числа с плаваща запетая се извършва моделиране на филтъра без да се отчита крайната точност на представяне на числата в реалния филтър. Получените резултати са практически еднакви с резултатите от теоретичния анализ на филтъра.

6.4.2.2. Моделиране с отчитане на крайната точност

За да се отчете крайната точност на работа на цифровия филтър при моделирането му е необходимо в моделиращата програма да се извършат някои допълнителни действия:

- ◆ Входната поредица за симулация да бъде закръглена до точността на работа на входния АЦП;
- ◆ Всички вътрешни действия на филтъра (умножения и събирания) да бъдат с ограничената точност на работа на филтъра;
- ◆ изходната поредица също трябва да бъде извеждана с точността на работа на ЦАП на изхода на филтъра.

Коригираната по този начин моделираща програма след симулация на реакцията на дадено входно въздействие ще дава вече резултати както реалния филтър. Те могат да се различават повече или по-малко от резултатите получени след теоретичния анализ.

6.5. Синтез на цифровите филтри

Въпросите за синтеза на цифровите филтри са подробно описани в литературата за цифрова филтрация и затова няма да бъдат засегнати с подробности. Те включват както добре познати части от проектирането на аналогови филтри (апроксимацията на амплитудно-фазовата характеристика), така и специфични за цифровите филтри особености (разрядност на АЦП и ЦАП, точност на вътрешното извършване на изчисленията). Тук ще бъдат разгледани само етапите на проектирането на цифровите филтри и отделни особености. Загължително след синтеза на цифровия филтър се извършва анализ на работата му и при незадоволителни резултати процесът на синтез се повтаря.

6.5.1. Входни параметри за синтез на цифров филтър

Преди започване на синтеза на цифровия филтър е необходимо да се уточнят ясно необходимите параметри и изисквания към проектирания цифров филтър.

6.5.1.1. Вид на филтъра

Видът на филтъра определя необходимата честотна характеристика. Биват нискочестотни, високочестотни, лентови и заграждащи. Специален вид филтри са тези на които честотната характеристика трябва да има точно определена форма, която се задава аналитично или таблично.

6.5.1.2. Изисквания към амплитудночестотната характеристика

Най-често срещаните изисквания към честотната характеристика са неравномерност на коефициента на предаване в лентата на пропускане и минимален коефициент на подтискане в лентата на непропускане. В случай на зададена честотна характеристика се предявява изискването за точност на покриването ѝ. Тези изисквания в крайна сметка задават разрешените зони, в които трябва да бъде честотната характеристика на готовия филтър.

6.5.1.3. Изисквания към фазочестотната характеристика

Изисквания от този тип се предявяват сравнително рядко. Най-често се предявява изискването за линейна фазочестотна характеристика. Това означава еднакво време на закъснение на сигнала от входа до изхода на филтъра, независимо от неговата честота. Филтрите с линейна фазочестотна характеристика имат много добри времеви характеристики — импулсна и преходна, което ги прави предпочитани в много случаи.

6.5.1.4. Максимална работна честота

Максималната работна честота еднозначно определя минималната честота на дискретизация. Съгласно теоремата на Шенон честотата на дискретизация трябва да бъде поне два пъти по-висока от максималната работна честота на цифровия филтър.

6.5.2. Избор типа на филтъра

Това е първият етап от проектирането на цифровите филтри. Основният въпрос тук е да се избере между рекурсивен и нерекурсивен филтър. Предимствата на рекурсивните филтри са, че с по-малко звена (респективно изчисления) се постига зададена стръмност на амплитудночестотната характеристика. Недостатъците са, че фазочестотната характеристика е нелинейна и това понякога довежда до не особено добри времеви параметри, също така при проектирането е необходимо да се прави проверка за устойчивост на филтъра. Точно линейната фазочестотна характеристика еднозначно определя нерекурсивен цифров филтър. В останалите случаи също се предпочитат нерекурсивните филтри, когато това е възможно и допустимо.

6.5.3. Апроксимация на желаната амплитудночестотна характеристика

Това е вторият и основен етап от проектирането на цифровите филтри. Тук трябва да се определят редът и коефициентите на цифро-

вия филтър, т.е. да се реши задачата за апроксимация на желаната амплитудночестотна характеристика.

6.5.3.1. Традиционни методи за апроксимация

Най-често тук се използват традиционните при проектирането на аналогови филтри апроксимации по Чебишев, Бътърворт или Кауер. Прилага се също така и апроксимация по метода на най-малките квадрати. При рекурсивни филтри тук се извършва и проверката за устойчивост. Всички тези методи за проектиране са много добре засегнати в литературата за цифрова филтрация и затова няма да се спираме подробно на тях.

6.5.3.2. Използване на преобразуването на Фурие

Един от елегантните методи за проектиране на нерекурсивни цифрови филтри е с използване на преобразуването на Фурие. Нека имаме зададена амплитудночестотна характеристика $A(\nu)$ и фазочестотна характеристика $\varphi(\nu)$. От тях получаваме комплексната честотна характеристика на цифровия филтър:

$$H(\nu) = A(\nu) \cdot e^{j\varphi(\nu)} = A(\nu) \cdot \cos[\varphi(\nu)] - j \cdot A(\nu) \cdot \sin[\varphi(\nu)].$$

След извършване на обратно преобразуване на Фурие се получава импулсната характеристика на цифровия филтър:

$$h(t) = TF[H(\nu)]^{-1}.$$

При практическо проектиране честотната характеристика се изчислява в M точки и след обратното преобразуване на Фурие се получават M точки от импулсната характеристика. Както беше отбелязано в т. 6.2.3 при нерекурсивните цифрови филтри точките от импулсната характеристика съвпадат с коефициентите на нерекурсивния цифров филтър.

Единственият проблем тук е да се определи броя на звената на нерекурсивния филтър. Едно възможно решение на проблема е чрез последователни итерации от синтез на филтъра и анализ на получения резултат. Числото M се увеличава до покриване на изискванията към филтъра. Този метод се подава добре на формализиране и лесно се реализира във вид на компютърна програма.

6.5.4. Оценка на необходимата точност

Третият етап от проектирането включва разглеждане на въпросите по разрядността на аналогово-цифровия преобразувател и регистрирете за съхранение на междинните резултати с цел постигане на зададената точност на амплитудночестотната характеристика.

Реалният цифров филтър се различава от идеалният цифров филтър, проектиран до този етап, по ограничената точност на представяне на числата. Това важи както за квантуването на входния аналогов сигнал и при превръщането на изходния сигнал от цифров код в аналогова величина, така и за точността (ограничения брой разряди) на извършване на изчисленията при реализиране на филтъра. Особено внимание трябва да се обърне на случаите, когато за ускоряване на бързодействието се използват действия с числа с фиксирана запетая. Ограничената точност на изчисленията се моделира с въвеждане на адитивен шум с дисперсия половината на ширината на стъпката на квантуване. Наличието на ограничена точност на изчисленията има следните последствия:

- ⇒ намаляване на работния динамичен диапазон (отношението на най-силния и най-слабия сигнал) на работа на филтъра в следствие на шума от квантуване;
- ⇒ изменения в проектираните амплитудночестотна и фазочестотна характеристики;
- ⇒ възможна неустойчивост на проектирания като устойчив рекурсивен цифров филтър.

Освен теоретичните разглеждания, добре описани в литературата, добре е да се извърши моделиране на проектирания реален цифров филтър и симулиране реакцията му на различни входни сигнали с отчитане на реалната точност на работата му.

6.6. Реализация на цифровите филтри

Практическото реализиране и използване на проектирания цифров филтър може да бъде осъществено по няколко начина: като част от програма за цифрова обработка на сигнал, като работеща в реално време програма, като апаратно реализиран цифров филтър, с използване на специализирани интегрални схеми, с използване на процесори за цифрова обработка на сигналите.

6.6.1. Програмна реализация на цифров филтър

Програмната реализация на цифровия филтър във време може да бъде един от етапите на обработка на дискретизирания и записан в паметта на компютърната система сигнал. Програмата в този случай не се различава от програмата за моделиране на работата на цифровия филтър. Повечето от проблемите, свързани с ограничената точност на изчисленията отпадат. За реализацията на програмата може да се използва както асемблер, така и език от високо ниво.

6.6.2. Програмна реализация в реално време

Програмната реализация на цифров филтър в реално време се нуждае от микропроцесорна система, към която са свързани АЦП и ЦАП. В този случай цифровият филтър е отделно устройство. Поради ограничената производителност на масово разпространените микропроцесорни фамилии този начин на реализация е с приложение за честоти на дискретизация до около 5 000 Hz.

6.6.3. Апаратна реализация

Целта тук е отново получаване на затворено устройство — филтър с един аналогов вход и с един аналогов изход, който има зададена честотна характеристика. Цялостната реализация е с помощта на стандартния набор от цифрови логически елементи — регистри, суматори и т.н, при което силно страда гъвкавостта и възможността за пренастройване на филтъра за сметка на значително по-голямата възможна честота на дискретизация — до десетки и стотици килохерци. Това е най-старият метод за реализиране на цифрови филтри.

6.6.4. Реализация посредством специализирани интегрални схеми

В този случай цифровият филтър се реализира посредством специализирана интегрална схема — цифров филтър. Към нея се свързват входният АЦП и изходният ЦАП. Интегралната схема притежава интерфейс към микропроцесорна система, която зарежда коефициентите на филтъра. Типична реализация са четири нерекурсивни филтъра от по 27 звена (!) в един корпус. Те могат и да се свързват последователно.

6.6.5. Реализация с процесор за цифрова обработка на сигнали

Специализираните процесори за цифрова обработка на сигналите имат редуцирана система от инструкции, които се използват най-често при цифровата обработка (например умножение с константа и прибавяне на резултата). Опростената система инструкции и специално разработената вътрешна архитектура ги прави високопроизводителни програмируеми устройства за цифрова обработка на сигнали, при което се съчетават гъвкавостта и лесното пренастройване на програмната реализация с бързодействието на чисто апаратната реализация. Типични представители са серията 32000 на фирмата Texas Instruments, серията 56000 на фирмата Motorola и др.



Едно типично приложение, реализирано със сигналния процесор 56000, е цифров изравнител на амплитудно честотната характеристика на висококачествен аудио усилвател (екулайзер). Сигналният процесор извършва цифрова филтрация на два звукови канала с честота на дискретизация 44 КHz и точност на входния АЦП и изходния ЦАП — 16 бита. В останалото време четете осемканален, сравнително по-бавен АЦП. С положението на потенциометрите, свързани към осемте канала на този АЦП, потребителят задава желаната амплитудночестотна характеристика. Сигналният процесор извършва проектиране в реално време на цифров филтър с такава характеристика и задава коефициентите му на по-бързата част от програмата, извършваща филтрацията в реално време.

6.7. Предимства и недостатъци на цифровите филтри

С предимствата на цифровите филтри е свързано все по-широкото им разпространение в съобщенията, системите за предаване на данни, обработката на биологически сигнали, резултати от изследването на машини и др. Като съставна част на сложна програмна система за обработка на сигнали цифровите филтри нямат алтернатива.

6.7.1. Предимства на цифровите филтри

Основните предимства на цифровите филтри са:

- ◆ точно покриване на предварително зададената амплитудночестотна характеристика;
- ◆ АЧХ не се изменя с времето както при аналоговите филтри в следствие на стареенето на елементите, т.е. имат висока стабилност на характеристиките;
- ◆ лесна настройка и пренастройка на цифровите филтри;
- ◆ в редица случаи реализацията им е с по-малки размери;
- ◆ имат по-висока надеждност.

6.7.2. Недостатъци на цифровите филтри

Посочените предимства се заплащат с недостатъците на цифровите филтри:

- ◆ по-сложната и скъпа апаратна част;
- ◆ появата на специфичен шум, свързан с процесите на квантуването;
- ◆ невъзможност за работа в реално време при по-високи честоти.

Обобщение

Цифровият филтър е изчислителна процедура над дискретизиран сигнал, реализираща определена честотна характеристика. По своята същност той представлява една линейна дискретна система, преобразуваща входната редица на дискретизирания сигнал в изходна редица по зададена функция. Цифровите филтри биват рекурсивни (с обратна връзка) и нерекурсивни.

Основните параметри на цифровите филтри са времеви (импулсна характеристика и преходна характеристика) и честотни (честотна, амплитудночестотна и фазочестотна характеристики).

Анализът на цифровите филтри включва намирането на основните параметри на зададен филтър. Това може да стане по теоретичните формули и чрез моделиране и симулация на работата на цифровия филтър с и без да се отчита ограничената точност на работата му.

Синтезът на цифровия филтър включва намирането на типа му, реда на филтъра и стойностите на коефициентите му. Извършва се на три етапа: избор на типа на филтъра, апроксимация на желаната честотна характеристика и оценка на необходимата точност за извършване на изчисленията.

В следствие на своите предимства цифровите филтри намират все по-широко приложение в съобщенията, системите за предаване на данни, обработката на биологически сигнали, резултати от изследването на машини и др. Като съставна част на сложна програмна система за обработка на сигнали цифровите филтри нямат алтернатива.



Задачи

1. Поради ограничения в паметта и бързодействието реда на един нискочестотен нерекурсивен цифров филтър е ограничен до пет. Намерете коефициентите на филтъра с най-тясна честотна лента!

Упътване: Изберете метода с използване на преобразуването на Фурие и погледнете фиг. 5.2.

2. Направете теоретичен анализ на филтъра — резултат от предишната задача. Това включва амплитудночестотната и фазочестотната характеристики, импулсната и преходната характеристики.

3. При разработката на цифров доплеров измерител на скоростта е необходимо да се реализира цифров филтър, отговарящ на следните условия:

- да подтиска честотите под 200 Hz;
- да подтиска честотите над 1500 Hz;

- коефициент на предаване при 200 Hz — единица;
- коефициентът на предаване в зоната на пропускане да се увеличи с квадрата на честотата и при 1500 Hz да достигне около 50 пъти.

Разделете този филтър на три последователно свързани звена и проектирайте всяко едно от тях.



Контролни въпроси

1. Какво представлява цифровият филтър?
2. Какво значи рекурсивен и нерекурсивен филтър?
3. Кои са основните параметри на цифровите филтри и как се дефинират?
4. Кои са етапите на синтезирането на цифровите филтри?
5. Кои са предимствата и недостатъците на цифровите филтри?

7. Некласически методи за обработка на сигналите



Ключови думи

- спектрален анализ
- спектрална луна
- ZOOM FFT
- кепстрален анализ
- сканиращ анализ

Какви знания и умения ще придобиете?

Къде се прилагат?

В тази тема ще се запознаете с неklasическите методи за обработка на сигналите. Те са мощно средство за получаване на допълнителна информация от вече дискретизирания сигнал.

Получените знания ще приложите при проектирането на алгоритмите за обработка на дискретизиран сигнал при конкретни приложения и при създаване на програмното осигуряване на системите за събиране и обработка на информация.

Въведение

В редица случаи в практиката се налага от дискретизирания сигнал да се извлече допълнителна информация за процесите, които той описва. Понякога изчислителната мощност на системата за събиране и обработка на информация не е достатъчна за прякото използване на класическите методи за обработка на сигналите, в други случаи не достига памет за тези обработки. Дискретизираните сигнали понякога отразяват процесите, които ни интересуват, в твърде завоалиран вид. В тези случаи се използват неklasическите методи за обработката им, които ни позволяват да получим необходимата ни информация.

Спектралната луна позволява да се получи увеличена част от спектъра на дискретизирания сигнал като се използва алгоритъма на бързото преобразуване на Фурие. По този начин можем да разгледаме и обработим само част от спектъра на сигнала.

Кепстралният анализ дава възможност да получим периодичността на изменение на спектъра на дискретизирания сигнал във времето.

Сканиращият спектрален анализ позволява да разгледаме как се изменя във времето спектъра на дискретизирания сигнал.

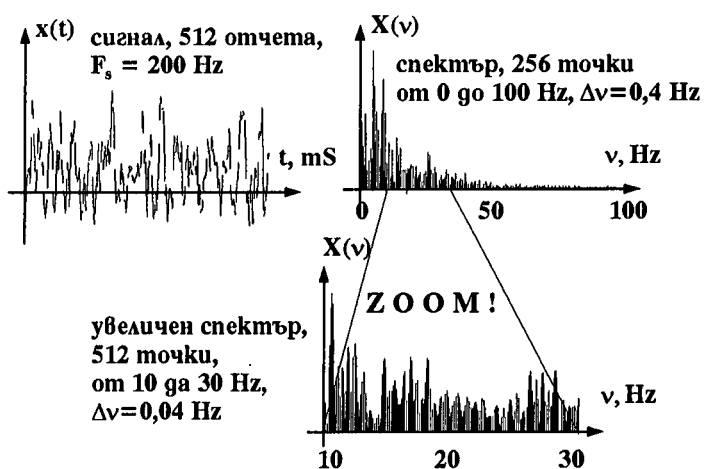
7.1. Спектрална лупа

7.1.1. Същност на проблема

Нека имаме дискретизиран сигнал, въведен в паметта на компютърната система. Искаме да разгледаме подробно част от неговия спектър. За яснота да си послужим с конкретен пример. С честота на дискретизация 200 Hz сме дискретизирали 512 отчета от аналогов сигнал. След бързо преобразуване на Фурие получаваме спектъра на сигнала отново в 512 точки в честотната област от 0 до 200 Hz (или в честотната област от -100 до +100 Hz, което е едно и също). Спектърът на реален сигнал е симетричен спрямо нулата, т.е. имаме 256 точки в обхвата от 0 до 100 Hz. В честотна област тези 256 точки са през интервал:

$$\Delta\nu = \frac{F_s}{N} = \frac{200}{512} \approx 0,4\text{Hz}.$$

Целта на обработката включва детайлно изследване на спектъра на сигнала в честотната област от 10 до 30 Hz. В този честотен обхват от 20 Hz имаме $N_1 = \frac{\Delta F}{\Delta\nu} \approx \frac{20}{0,4} = 50$ точки от спектъра. Да приемем, че за нуждите на нашата обработка са необходими десет пъти повече точки в този честотен обхват, което означава интервал $\Delta\nu_1 \leq 0,04\text{Hz}$, както е показано на фиг. 7.1. За решаването на този проблем има няколко възможности:



Фиг. 7.1. Същност на спектралната лупа

⇒ отказ от алгоритъма на бързото преобразуване на Фурие и изчисляване на спектъра на сигнала по формулата, дадена в точка 5.1.6, в разположени по наше желание точки;

⇒ въвеждане в паметта на компютърната система на десет пъти повече точки от сигнала и използване отново на бързото преобразуване на Фурие;

⇒ прилагане на процедура на спектралната лупа с използване на алгоритъма на бързото преобразуване на Фурие.

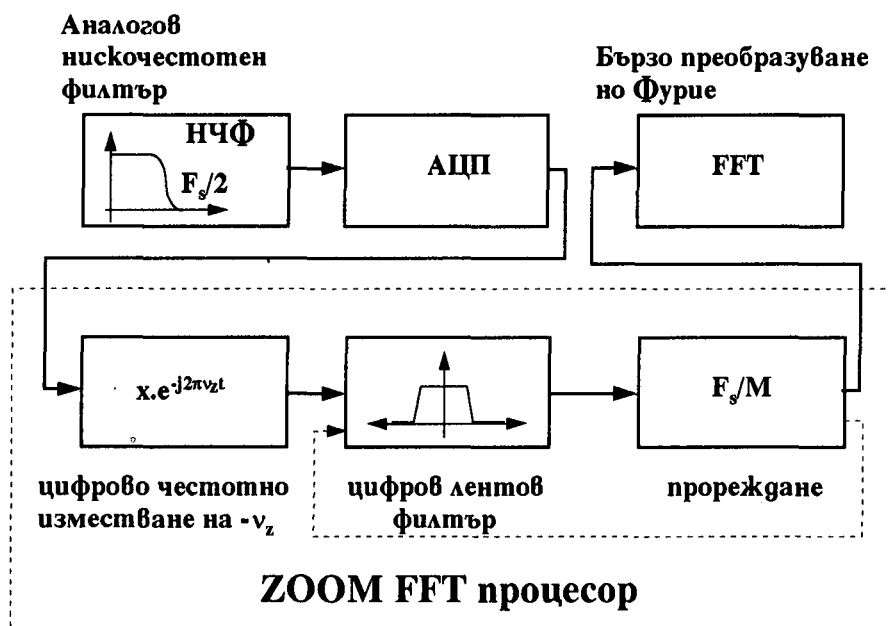
Първата възможност е свързана с голямо количество изчисления, както е показано в точка 5.1.7. За нашия случай ($N = 512$) времето за

изчисляване на 512 точки от спектъра в интересувания ни честотен интервал ще е около 57 пъти (!) по-голямо. Във втория случай, при условие че имаме достатъчно памет в компютърната система, за получаване на целия спектър на въведените 8192 точки (най-близката точна степен на двойката над 5120) ще е необходимо 23 пъти (!) повече време. За обработки от този тип алгоритмите на спектралната лупа нямат алтернатива.

Спектралната лупа в редица източници се нарича прозоречно преобразуване на Фурие. Друг често използван термин е ZOOM FFT. Терминът *zoom* в киното се използва за обектите с променливо фокусно разстояние, позволяващи да се увеличава плавно изображението. Самото плавно увеличаване на изображението (от кадър в който артистът е в цял ръст към кадър в който се вижда само лицето му) също се нарича *zoom*. В нашия случай ZOOM FFT може да се преведе като увеличаване на част от спектъра на сигнала с използване на алгоритъма на бързото преобразуване на Фурие.

7.1.2. Спектрална лупа с честотно изместване и цифрова филтрация

Този алгоритъм се използва в повечето спектрални анализатори. Повишаването на разрешаващата способност в честотна област се постига след изместване на честотата и нискочестотно филтриране.



Фиг. 7.2. Спектрална лупа с честотно изместване и цифрова филтрация

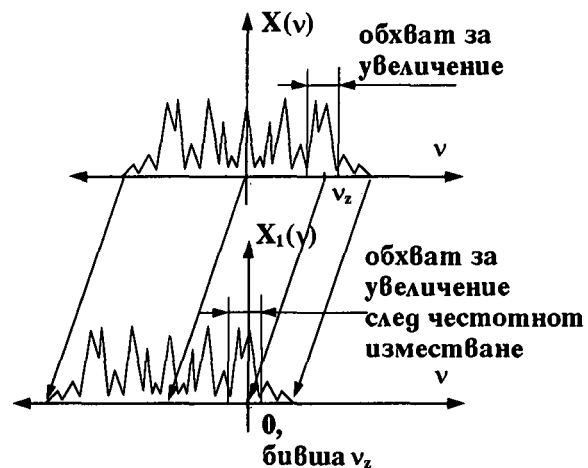
Блоковата схема за получаване на увеличени участъци от спектъра е показана на фиг. 7.2. Входните параметри са средна честота на увеличени интервал ν_z и коефициент на увеличение M . Получаването на увеличени спектъра протича на няколко етапа.

7.1.2.1. Нискочестотна аналогова филтрация и дискретизация

След преминаване през аналоговия нискочестотен филтър с честота на среза половината от честотата на дискретизация F_s аналоговият сигнал се подава към аналогово-цифровия преобразувател. Със зададен период на дискретизация T аналогово-цифровия преобразувател подава отчетите на входния сигнал $x(kT)$ за обработка.

7.1.2.2. Честотно изместване

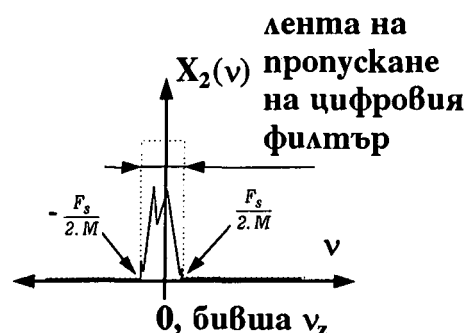
Всеки отчет на входния сигнал се умножава с единичен, въртящ се с честота ν_z , вектор — $e^{-j.2.\pi.\nu_z.k.T}$. Това умножение предизвиква изместване на целия честотен спектъра на сигнала на честота $-\nu_z$, както е показано на фиг. 7.3. От този момент нататък входният сигнал е вече поредица от комплексни числа.



Фиг. 7.3. Честотно изместване

7.1.2.3. Цифрова филтрация

Върху получения сигнал се прави филтрация с цифров филтър с честота на среза $\frac{F_s}{2.M}$. След цифровата филтрация остава само интересувашата ни част от спектъра на сигнала, изместена симетрично спрямо нулата, както е показано на фиг. 7.4.



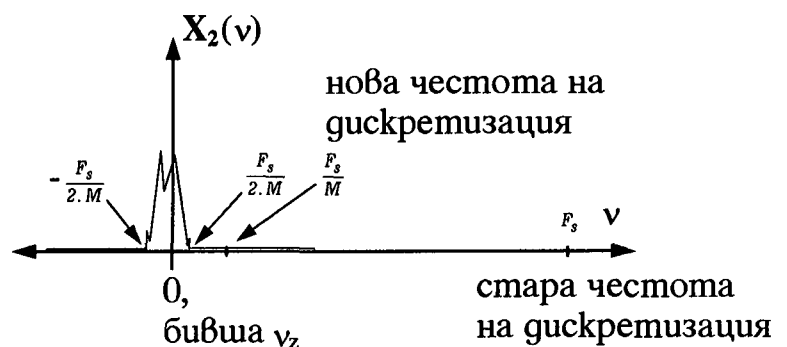
Фиг. 7.4. Цифрова филтрация

7.1.2.4. Прореждане

За да получим спектъра с търсената разрешаваща способност, след цифровата филтрация е необходимо да намалим M пъти честотата на дискретизация, както е показано на фиг. 7.5. Това може да се постигне по различни начини:

- ♦ пропуска се всяка M -та точка от потока дискрети, а останалите се отхвърлят, което е подходящо при спектрална луна в реално време;
- ♦ извършва се разместване на точките в масива, като мислено се подрежда масивът с филтрираните точки M пъти един след друг във времето и се взема всяка M -та точка (за да се използват всички точки от масива е необходимо броят им N и коефициентът на увеличение M да са взаимно прости числа, т.е. да нямат общи множители).

Първият начин на прореждане се използва в работещите в реално време спектрални анализатори, при които не се изисква запазване на входния сигнал. При това за получаването на N броя точки ще е необходимо време $M \cdot N \cdot T$. Вторият начин на прореждане е приложим когато имаме вече дискретизиран и запомнен сигнал.



Фиг. 7.5. Промяна на честотата на дискретизация чрез прореждане

7.1.2.5. Бързо преобразуване на Фурие

Над получените след прореждането N точки се извършва бързо преобразуване на Фурие. След честотното изместване дискретизираният сигнал вече е поредица от комплексни числа. Затова и спектърът му не е симетричен спрямо нулата — това е увеличеният M пъти спектър на сигнала с централна честота ν_z . По този начин се получава ZOOM FFT спектъра.



Обърнете внимание, че на фиг. 7.2 с пунктирна линия е показано, че след прореждането сигналът може още веднъж да премине през нискочестотна филтрация и прореждане, което означава да се постигне увеличение в спектъра от още M пъти.

Основните предимства на този алгоритъм за спектрална луна са:

- възможност за реализиране на големи коефициенти на увеличение;

- възможност за работа в реално време (ако изчислителната мощност на системата го позволява);
- може да не се изразходва памет за запазване на дискретизирания във времето входен сигнал.

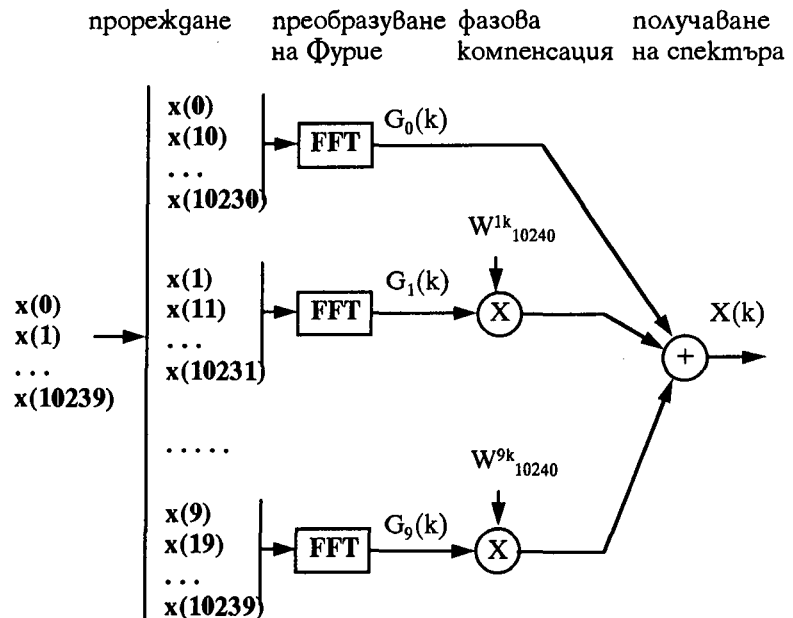
7.1.3. Спектрална лупа при записан сигнал

В редица случаи имаме продължителен запис на дискретизиран сигнал. Спектърът на този сигнал е с достатъчна разрешаваща способност, но паметта на системата за събиране и преобразуване на информация е недостатъчна за извършване на бързото преобразуване на Фурие над целия сигнал или не достига изчислителна мощност за получаването на целия спектър за необходимото време. С използването на този алгоритъм е възможно да се получи както целия, така и част от спектъра.

Нека за конкретност приемем, че са запомнени 10 240 дискрета (10 К). Общата блокова схема на алгоритъма в този случай е показана на фиг. 7.6. Получаването на спектъра протича през няколко етапа.

За целта се формират десет масиви по 1024 отчета (1 К), съставени по следния начин:

- ◇ първият масив $g_0(k)$ включва отчетите с номера 0,10,20,...,10230;
- ◇ вторият масив $g_1(k)$ включва отчетите с номера 1,11,21,...,10231;
- ◇ ... и т.н.;
- ◇ десетият масив $g_9(k)$ включва отчетите с номера 9,19,29,...,10239.



Фиг. 7.6. Спектрална лупа при записан сигнал

7.1.3.1. Прореждане

За целта се формират десет масиви по 1024 отчета (1 К), съставени по следния начин:

7.1.3.2. Бързо преобразуване на Фурие

Извършва се преобразуване на Фурие на всеки от десетте формиранни масива и се получават спектрите $G_0(k)$, $G_1(k)$, ..., $G_9(k)$. От всеки 1024 точки сигнал се получават 1024 точки спектър, така че $k = 0, \dots, 1023$. Понеже преобразуването на Фурие е линейна трансформация, комбинирането на десетте спектъра трябва да даде спектъра на целия 10 К запис.

7.1.3.3. Фазова компенсация

От получените десет масива само първият е с нулево изместване във времето спрямо целия запис. Вторият е изместен във времето на интервал T , третият — на интервал $2.T$ и т.н. Затова се налага преди комбинирането на спектрите да се извърши компенсация на това изместване във времето на отделните масиви. В честотна област изместването във времето се отразява като завъртане на фазите в спектъра на ъгъл $\Delta\varphi = -2.\pi.l.k / 10240$. Затова всеки елемент на спектъра $G_l(k)$ се умножава на компенсиращия фазата множител:

$$W_{10240}^{lk} = e^{-j \cdot \frac{2.\pi.l.k}{10240}} \quad l = 0, 1, \dots, 9.$$

7.1.3.4. Получаване на целия или на част от спектъра

Спектърът на дискретизирания сигнал се получава като се сумират фазово компенсираниите десет спектъра:

$$X(k) = \sum_{\substack{l=0 \\ k=0, \dots, 5119}}^9 W_{10240}^{lk}(k) \cdot G_l(k)$$

В случаите, когато е необходим само част от спектъра, се извършва изчисление на точките $X(k)$ за $k_{min} \leq k \leq k_{max}$, където k_{min} и k_{max} определят лявата и дясната честота на необходимият ни спектрален прозорец. При изчисляване на всички точки от спектъра резултатът ще е същият както при преобразуването на Фурие на целия сигнал. Удобството тук е, че спектърът може да се изчисли само частично и да не се губи време за изчисляване на останалата част.

Обърнете внимание, че тук $k = 0, \dots, 5119$, за да се получат точките от спектъра на 10240 дискрета. За целта се използва свойството за циклично повторение на спектъра на сигнала след дискретизация, така че $G_l(0) = G_l(1024) = G_l(2048) = \dots$



Основните предимства на този алгоритъм за спектрална лупа са:

- възможност за реализиране на фиксирани и не особено високи коефициенти на увеличение, тъй като те зависят от броя на дискретизираните точки (обеа памет на системата за събиране и обработка на информация);
- възможност за работа извън реално време над вече записан сигнал;
- по тази причина не се изисква изключително висока изчислителна мощност;
- запазва се дискретизирания във времето входен сигнал.

7.2. Кепстрален анализ

7.2.1. Дефиниция и терминология

Кепстърът $K(\theta)$ е обратна трансформация на Фурие на логаритъма на спектралната плътност (честотната мощност) $S_{xx}(\nu)$, както следва:

$$K(\theta) = TF\{\log[S_{xx}(\nu)]\}^{-1}.$$

Обърнете внимание на подобие то на дефиницията на кепстъра и теоремата на Винер-Хинчин за спектъра на корелационната функция:

$$C_{xx}(\tau) = TF\{S_{xx}(\nu)\}^{-1}.$$



Независимата променлива на кепстъра θ е с дименсия време. Тя дава периода на промяна на спектъра на сигнала. Максимум (върха) в кепстъра за стойност на независимата променлива θ_0 означава периодична промяна на спектъра на сигнала с период θ_0 .

Кепстралния анализ е обработка на сигнала, която няма реален физически смисъл. Въведените термини също са измислени с дефинирането на кепстралната функция. Независимата променлива се нарича *quersfrequency* (от размяната на първите две срички на *frequency* — честота). Кратните на основната кепстрална линия се наричат *rahnionics* (рахмоници — идва от обратното прочитане на първата сричка на *harmonics* — хармоници). Самото име кепстър (*cepstrum*) идва от обратното прочитане на първата сричка на спектър (*spectrum*).

7.2.2. Предимства пред автокорелационния анализ

При въвеждането на сигнала посредством датчици в компютърната система, той претърпява честотни изкривявания — амплитудно-честотните характеристики на датчиците, кабелите, усилвателите, схемата следене/запомняне и т.н. Нека целия път на сигнала от изследвания процес до въвеждането му в компютърната система има амплитудно-честотна характеристика $H_{xy}(\nu)$. При автокорелационен анализ получаваме за мощностния спектър (спектралната плътност) на сигнала:

$$S_{yy}(\nu) = S_{xx}(\nu) \cdot |H_{xy}(\nu)|^2,$$

където $S_{xx}(\nu)$ е спектралната плътност на изследвания процес, а $S_{yy}(\nu)$ е изкривената спектрална плътност, която всъщност получаваме. При кепстралния анализ след логаритмуването се получава:

$$\log S_{yy}(\nu) = \log S_{xx}(\nu) + 2 \cdot \log |H_{xy}(\nu)|,$$

и след обратно преобразуване на Фурие:

$$TF\{\log S_{yy}(\nu)\}^{-1} = TF\{\log S_{xx}(\nu)\}^{-1} + TF\{2 \cdot \log |H_{xy}(\nu)|\}^{-1}.$$



Предимството на кепстралния анализ пред автокорелационния анализ е, че мултипликативната грешка от честотните изкривявания на входните блокове се превръща в адитивна и оттам има по-малко влияние.

7.2.3. Предимства пред спектралния анализ

При сигнали със сложен спектър (много хармонични, взаимни влияния, модуляции) е много трудно да се извърши намирането на търсените характерни линии, които ни дават информация за процеса, който изследваме. При такъв тип сигнали кепстрът дава много ясна и чиста картина, удобна и лесна за тълкуване.

На фиг. 7.7 е показан сложен аналогов сигнал, получен с едновременна честотна и амплитудна модулация на синусоидален сигнал с амплитуда 1 V и честота 1000 Hz. Честотата на модулиращия сигнал е 100 Hz, а амплитудата му — 0,3 V.

Спектрът на този сигнал (използван е теглови прозорец на Хеминг) е показан на фиг. 7.8 и дава само относителна представа за честотата на основния сигнал и заеманата честотна лента. От този спектър не е възможно да се намери честотата на модулиращия сигнал.

Кепстърът на същия сигнал е показан на фиг. 7.9, като при получаването му отново е използван теглови прозорец на Хенинг. Отчетливо се вижда периода на модулиращата честота, но не е възможно да се намери периода на основния сигнал.



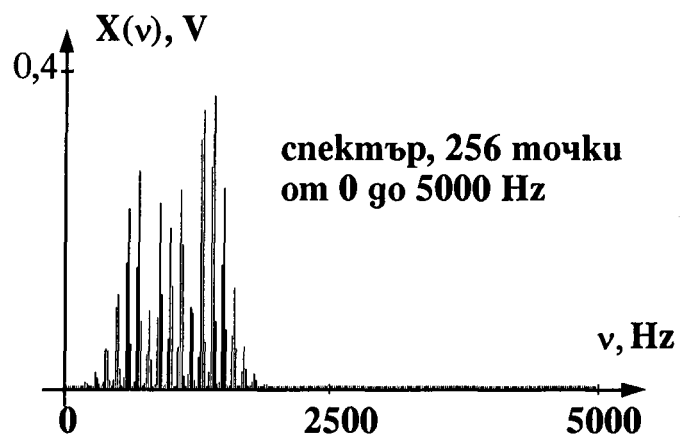
Кепстралният анализ не измества спектралния, той просто дава информация, която не се вижда ясно при разглеждане на спектъра. Двама метода на обработка на сигналите се допълват при изследване на сигнали със сложна форма.

7.2.4. Приложение на кепстралния анализ

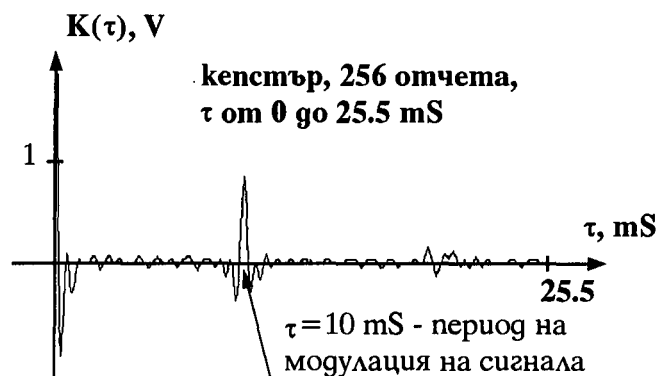
Основното приложение на кепстралния анализ е при изследване на вибрациите на машини и механизми — двигатели, генератори, турбини, предавателни кутии. В процеса на вибриране на механизмите всеки елемент “звъни” (създава вибрации) на собствената си резонансна честота. Тази честота зависи от механичните усилия, приложени върху този елемент. От тук следва, че вибрациите на механизма като цяло извършват честотна модулация на “звъненето” на всеки отделен елемент. Към сигнала, смет от тензодатчика, се прибавят и допълнителни шумове, резонансни явления и т.н. Това прави спектъра твърде богат на



Фиг. 7.7. Амплитудно и честотно модулиран сигнал



Фиг. 7.8. Спектр на сигнала



Фиг. 7.9. Кепстър на сигнала

спектрални линии и техните хармонични честоти. За сметка на това в кепстъра отчетливо се виждат основните периоди на вибрация на целия механизъм. Това прави възможно, посредством периодично запущени кепстри на вибрацията на машината, да се извърши количествен анализ на вибрациите и да се планира своевременно замяната на износени лагери, зъбни колела и др.

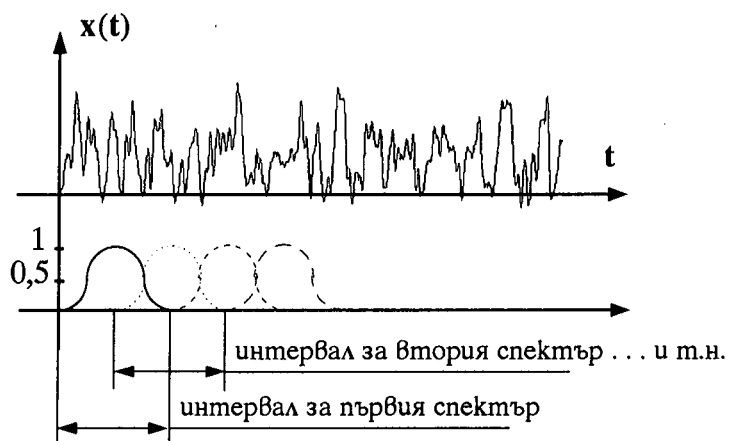
На същите принципи като прозоречното преобразуване на Фурие са разработени и алгоритмите за прозоречно изчисляване на кепстъра.

7.3. Сканиращ анализ

7.3.1. Същност на сканиращия анализ

В редица случаи предмет на изследване е сигнал, чието спектър не е стационарен, т.е. не е постоянен във времето. Тогава е интересно да се види в кои моменти от времето се появяват дадени честотни съставки, как се изменя амплитудата им във времето и кога изчезват.

За извършване на сканиращ спектрален анализ над дълъг запис от сигнала се използва последователност от теглови прозорци (най-често на Хенинг), припокриващи се на половината от нивото си, както е показано на фиг. 7.10. След умножаването на сигнала с тегловия прозорец се извършва преобразуване на Фурие и се получава усреднен спектър за първия интервал от време. След това тегловият прозорец се измества по оста на



Фиг. 7.10. Преместване на тегловия прозорец при сканиращ спектрален анализ

времето, прави се ново преобразуване на Фурие и се получава усреднения спектър на сигнала през втория интервал от време. Процесът се повтаря до достигане края на записа. Резултатът от сканиращия спектрален анализ е един двумерен масив от числа, в който по редове са наредени спектрите на сигнала за определените интервали от време.

7.3.2. Представяне на резултатите от сканиращия анализ

По своята същност резултатите от сканиращия анализ изискват графика с три измерения — време, честота и амплитуда.

Един от начините за визуално представяне е чрез правоъгълен масив от точки на екрана на компютъра. По едната ос е скалата на времето, а по другата — скалата на честотата. Яркостта на всяка точка зависи от амплитудата на съответната спектрална съставка в съответния момент от време. Исторически това е по-старият начин.

Друг възможен начин, използван най-широко днес, е представяне на резултатите от сканиращия спектрален анализ като тримерна графика. Това е по-прегледният начин при който обикновено потребителят има възможност да изменя ъгъла на завъртане на графиката и височината от която я гледа. По този начин може да се разгледат всички детайли на триизмерното изображение на резултатите от сканиращия спектрален анализ.

7.3.3. Практическо приложение на сканиращия анализ

В практиката сканиращият анализ се използва в редица области на изследванията на сигнали. Една такава област е изследването и анализа на говор. След получаване на изменението на спектъра на записания говорен сигнал във времето може да се направи анализ на появата на всяка гласна, нейните честотни съставки (т.нар. формантни честоти) и т.н. Получените резултати намират приложение при разработването на програми за анализ и синтез на говор.

Друго възможно приложение е сканиращ спектрален анализ на вибрациите на един двигател с вътрешно горене за един оборот. На получените спектри в зависимост от ъгъла на завъртане на колянвия вал отчетливо се виждат моментите на отваряне и затваряне на клапаните, момента на запалване на горивната смес и т.н. Получените резултати са ценно средство за надникване в процесите, протичащи в двигателя с вътрешно горене. В случаите на изследване на такива периодични процеси (един оборот на колянвия вал на двигателя) за отстраняване на случайни шумове и грешки е удачно записването на повече (поне десетина) оборота на двигателя и извършването на статистическото усредняване във времева област, описано в т. 3.5.2.

На същия принцип може да се извърши и сканиращ кепстрален анализ, като се получава графика на изменение на кепстъра на сигнала във времето.



Пример за сканиращ анализ

На фиг. 7.11 е показан въведен в компютърна система електрокардиографски сигнал — резултат от биоелектричната активност на чо-

вешкото сърце. Записът е 2048 отчета, честотата на дискретизация — 400 Hz. Отчетливо се виждат ударите на сърцето на пациента.

Извършен е сканиращ анализ, като продължителността на интервала е 128 отчета — 320 mS. Получени са 31 спектъра (защо?). Резултатите от сканиращия спектрален анализ на електрокардиографския сигнал са показани на фиг. 7.12. За прегледност спектрите са показани в интервала от 0 до 100 Hz. Вижда се нарастването на амплитудата на високочестотните съставки по време на ударите на сърцето.

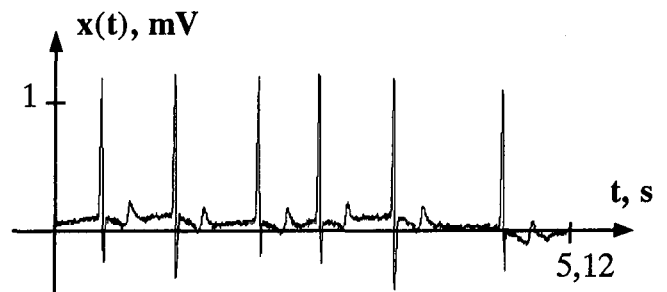
Обобщение

В тази глава се запознахте с някои неklasически методи за обработка на сигналите, чрез които от вече дискретизирания и въведен в паметта на компютърната система може:

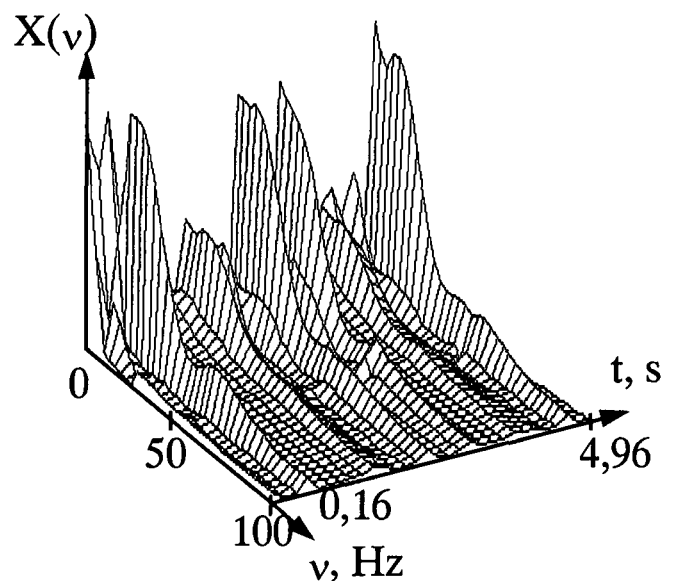
- ⇒ да се извлече допълнителна информация за процесите, които описва сигнала;
- ⇒ наличната информация да се представи в по-прегледен и удобен за оператора вид.

Спектралната луна позволява бързо да се получат увеличени и с по-добра разрешаваща способност части от спектъра на сигнала. Разгледани бяха два алгоритъма. Първият е удобен за работа в реално време и при високи коефициенти на увеличение и включва следните етапи:

- ◆ честотно изместване;
- ◆ нискочестотна цифрова филтрация;
- ◆ прореждане;



Фиг. 7.11. Електрокардиографски сигнал



Фиг. 7.12. Сканиращ спектрален анализ на електрокардиографски сигнал

◆ бързо преобразуване на Фурие.

Вторият алгоритъм на спектралната луна е удобен при вече записани сигнали и при фиксирани и не особено високи коефициенти на увеличение. Включва следните етапи:

- ◆ прореждане;
- ◆ бързо преобразуване на Фурие;
- ◆ фазова компенсация;
- ◆ сумиране и получаване на спектъра.

Кепстрът е обратното преобразуване на Фурие на логаритъма на мощностния спектър. Има независима променлива с дименсия време и дава периода на изменение на спектъра на сигнала. Предимството пред автокорелационния анализ е, че мултипликативната грешка от честотните изкривявания на входните блокове при кепстралния анализ се превръща в адитивна. Допълва спектралния анализ, като дава информация не за честотните съставки на сигнала, а за периодичността на изменение на спектъра му. Кепстралния анализ намира основно приложение при анализа на вибрациите на машини и механизми.

Сканиращият спектрален анализ ни дава как се изменя спектъра на сигнала във времето. За получаването на това изменение се използват преместващи се във времето теглови прозорци и бързо преобразуване на Фурие. Сканиращият спектрален анализ намира приложение при анализа на говор, вибрациите на двигатели с вътрешно горене и др. На същия принцип може да се направи и сканиращ кепстрален анализ.



Контролни въпроси

1. Кои са етапите и какви са основните предимства на спектралната луна с честотно изместване?
2. Къде намира приложение спектралната луна с прореждане?
3. Как се дефинира кепстрът и кои са предимствата му пред автокорелационния анализ?
4. В кои случаи е приложим сканиращия спектрален анализ?

8. Синхронно детектиране и синхронизиран АЦП



Ключови думи

- синхронно детектиране
- корелационна функция
- синхронизиран АЦП
- честотна характеристика
- импулсна характеристика
- преходна характеристика

Какви знания и умения ще придобиете? Къде се прилагат?

В тази тема ще се запознаете с едно практическо приложение на корелационния анализ за детектиране на сигнали на фона на силни шумове и намаляване на съотношението сигнал/шум. Освен с приложението на класическият метод на синхронното детектиране ще се запознаете и с неговата модификация — синхронизиран АЦП.

Получените знания за метода ще можете да приложите при конструиране на апаратура, възприемаща сигнали към които е прибавен шум.

Въведение

При създаване на системи за събиране и обработка на информация в много случаи сигналът, който трябва да се въведе в компютърната система, е примесен с голямо количество смущаващи сигнали и шумове, чиято амплитуда е сравнима с амплитудата на полезния сигнал.

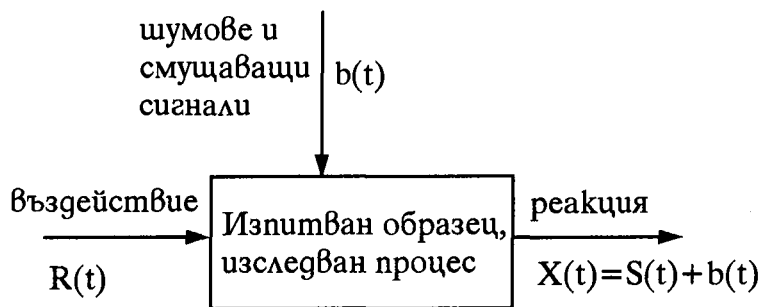
Въз основа на класическия метод на синхронното детектиране, който се реализира на базата на аналогова схемотехника, може да се разработи и използва негова модификация, която е приложима при компютъризираните устройства. Модифицираният метод на синхронното детектиране съчетава в себе си предимствата на класическия метод с мощта на програмната обработка на сигналите в съвременните системи за дискретизиране и цифрова обработка на сигналите.

8.1. Метод на синхронното детектиране

8.1.1. Област на приложение на метода

Синхронното детектиране се използва в случаите, когато полезният сигнал е сравним по амплитуда с шумовете. В следствие на това се създават затруднения за неговото точно измерване. На фиг. 8.1 е показана опитна постановка без използване на синхронно детектиране.

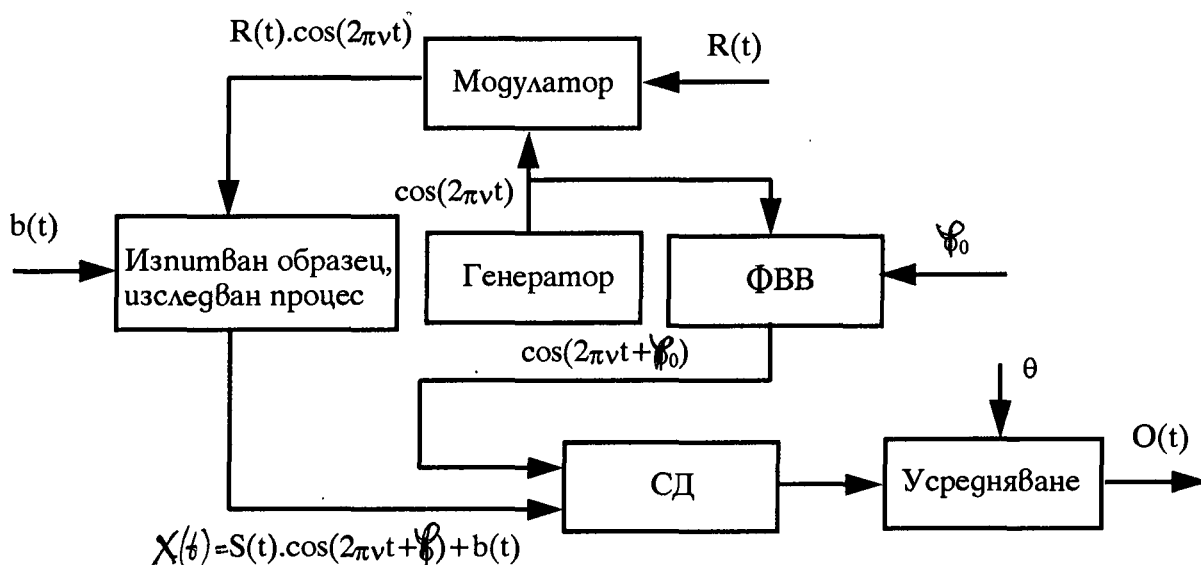
Нека $R(t)$ е въздействието върху изпитвания образец (изследваното явление), което заема честотната лента $(-\Delta B, +\Delta B)$ около нулевата честота. Шумът $b(t)$ заема по-широка честотна лента и спектърът на $R(t)$ се покрива от спектъра на $b(t)$. Това прави невъзможно използването на обикновено филтриране (по аналогов или цифров път) за отделянето на реакцията $S(t)$ на изпитвания образец (полезния сигнал) от регистрирания изходен сигнал $X(t) = S(t) + b(t)$.



Фиг. 8.1. Опитна постановка без използване на синхронно детектиране

8.1.2. Блокова схема на метода на синхронното детектиране

За успешното отделяне на полезния сигнал се използва опитната постановка от фиг. 8.2. Сигналят $R(t)$ модулира амплитудата на си-



Фиг. 8.2. Блокова схема на метода на синхронното детектиране

нусоцидален сигнал с честота ν и се получава въздействието $R(t). \cos 2\pi\nu t$. Честотата ν съществено превишава ΔB (около 100 пъти) и затова можем да считаме, че модулиращия сигнал $R(t)$ е в първо

приближение константа. Реакцията на изпитвания образец е $X(t) = S(t) \cdot \cos(2\pi\nu t + \varphi) + b(t)$. Тук φ е закъснението на реакцията спрямо въздействието изпитвания образец (изследваното явление). На синхронния детектор СД се подава сигналът $X(t)$ и дефазираният от фазовъртящата верига ФВВ на фаза φ_0 опорен сигнал с честота ν . Синхронният детектор СД умножава тези два сигнала. Изходният сигнал $O(t)$ се получава след преминаване през усредняващо звено - нискокочестотния филтър НЧФ с времеконстанта θ .

8.1.3. Теория на метода на синхронното детектиране

Изходният сигнал $O(t)$ на изхода е равен на:

$$\begin{aligned} O(t) &= \frac{1}{\theta} \int_0^{\theta} X(t) \cdot \cos(2\pi\nu t + \varphi_0) \cdot dt = \\ &= \frac{1}{\theta} \int_0^{\theta} [S(t) \cdot \cos(2\pi\nu t + \varphi) + b(t)] \cdot \cos(2\pi\nu t + \varphi_0) \cdot dt = \\ &= \frac{1}{\theta} \int_0^{\theta} S(t) \cdot \cos(2\pi\nu t + \varphi) \cdot \cos(2\pi\nu t + \varphi_0) \cdot dt + \frac{1}{\theta} \int_0^{\theta} b(t) \cdot \cos(2\pi\nu t + \varphi_0) \cdot dt. \end{aligned}$$

Понеже:

$$\begin{aligned} \cos(2\pi\nu t + \varphi) \cdot \cos(2\pi\nu t + \varphi_0) &= \frac{1}{2} [\cos(\varphi - \varphi_0) + \cos(4\pi\nu t + \varphi + \varphi_0)], \text{ то} \\ O(t) &= \frac{1}{2\theta} \int_0^{\theta} S(t) \cdot \cos(\varphi - \varphi_0) \cdot dt + \frac{1}{2\theta} \int_0^{\theta} S(t) \cdot \cos(4\pi\nu t + \varphi + \varphi_0) \cdot dt + \\ &+ \frac{1}{\theta} \int_0^{\theta} b(t) \cdot \cos(2\pi\nu t + \varphi_0) \cdot dt \Rightarrow \frac{1}{2\theta} \int_0^{\theta} S(t) \cdot \cos(\varphi - \varphi_0) \cdot dt = \\ &= \frac{1}{2} S^*(t) \cdot \cos(\varphi - \varphi_0), \end{aligned}$$

тъй като при нарастване на θ вторият и третият интеграл клонят към нула (средната стойност на синусоидален сигнал и на шумов сигнал при достатъчно голяма времеконстанта) и след извеждане на константата $\cos(\varphi - \varphi_0)$ пред интеграла остава $S^*(t)$ — средна стойност на реакцията $S(t)$.

Следователно, ако зададем такава фаза φ_0 на фазовъртящата верига ФВВ, че $\varphi = \varphi_0$, то изходният сигнал $O(t)$ ще бъде максимален по амплитуда и равен на половината на $S^*(t)$. По този начин като изменяме фазата φ_0 до получаването на максимален изходен сигнал $O(t)$

можем да получим както средната стойност на реакцията на изпитвания образец $S^*(t) = 2 \cdot O(t)$, така и закъснението на реакцията на изпитвания образец $\Delta t = \frac{\varphi_0}{2 \cdot \pi \cdot \nu}$.

В същност методът изчислява взаимнокорелационната функция на реакцията на обекта и на опорния сигнал — вж. т. 5.3.3.2.в. Времето за интегриране е времеконстантата на нискочестотния филтър. Силните филтриращи свойства се дължат на това, че методът е чувствителен не само към честотата на входния сигнал, както теснолентовия филтър, но и към фазата му.

В много литературни източници изложената дотук теория и метод се нарича синхронна демодулация. Терминът синхронно детектиране се използва за технически по-опростената версия, при която опорният сигнал е с правоъгълна форма. Прецизният аналогов умножител се заменя с управляем ключ.



Пример за използване на синхронното детектиране

Нека приемем, че се налага да измерим прозрачността на въздуха за целите на екологичен контрол на замърсеността му с прах, сажди и други механични включвания. За целта на определено разстояние (до няколко километра) ще разположим:

- ⇒ нашата апаратура, състояща се от излъчвател на светлина (обикновено лазер) и фотоприемник;
- ⇒ отражател — трипел-призма, която връща светлината в посоката от която е дошла, подобно на светлоотразителите на автомобилите и велосипедите.

Лазерът излъчва светлина с калибрирана интензивност. При абсолютна тъмнина попадналата във фотоприемника светлинна енергия ще зависи от поглъщането на въздуха за удвоеното разстояние между апаратурата и отражателя. Към върната светлинна енергия, обаче, се прибавя и светлината от звезди, изкуствени източници и др., които в този случай са смущаващи сигнали. Те ще намалят силно точността на измерването, независимо от взетите мерки: фокусираща тесноъгълна оптика на предавателя и приемника и поставянето на оптичен филтър пред приемника, пропускащ само светлината на лазера.

В този случай изходът е да модулираме светлината на лазера и да приложим метода на синхронното детектиране. Реализирането на уреда може да се извърши като се използва точно блоковата схема на описвания метод.

8.2. Синхронизиран АЦП

8.2.1. Теоретични основи

Използвайки математическия апарат на класическото синхронно детектиране може лесно да се докаже, че формата на модулация сигнал и сигнала за демодулиране не е от съществено значение. Това позволява построяването на синхронен детектор при който демодулиращ сигнал е дискретизиращата функция:

$$Ш_T(t) = \sum_{k=-\infty}^{+\infty} \delta(t - kT),$$

където $\delta(t)$ е импулса на Дирак:

$$\delta(t) = \begin{cases} 1 & \text{за } t = 0 \\ 0 & \text{за } t \neq 0 \end{cases}.$$

По своята същност $Ш_T(t)$ е гребеновидна функция, безкрайна сума от "иглите" на дираковите импулси. Това е функцията, при умножаването с която се описва теоретично процеса на аналогово-цифровото преобразуване. При дискретизация на входния сигнал посредством функцията $Ш_T(t)$ се извършва и синхронно детектиране на входния сигнал. В този случай ролята на синхронен детектор играе аналогово-цифровия преобразувател.

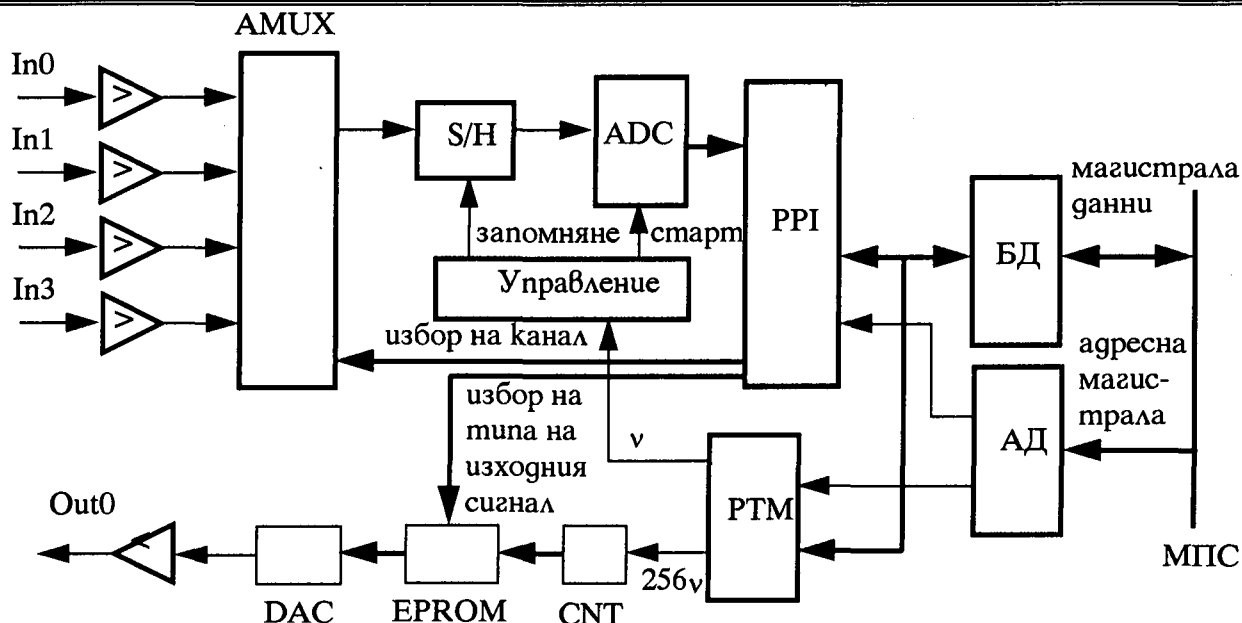
8.2.2. Модул синхронизиран АЦП за микрокомпютър

Примерната блокова схема на модула синхронизиран АЦП е показана на фиг. 8.3. Буферите за данни БД и адресният дешифратор АД осигуряват на микропроцесорната система МПС достъп до паралелния периферен интерфейсен адаптер РРІ и програмируемият таймер РТМ. Модулът се състои от няколко функционални блока:

- ◆ функционален генератор;
- ◆ аналогово-цифров преобразувател;
- ◆ синхронизираща част;
- ◆ нискочестотен филтър.

8.2.2.1. Функционален генератор

Функционалният генератор е реализиран посредством брояча CNT, тактуван от таймера с честота 256 ν , постоянната памет EPROM, цифрово-аналоговия преобразувател DAC и изходния му усилвател. Изходният сигнал на функционалния генератор е с честота ν . Всеки период на изходния сигнал е изчислен в 256 точки, чиито стойности са записани в постоянната памет. Формата на сигнала може да се задава посредством превключване на старшите адресни шини на постоянна-



Фиг. 8.3. Блокова схема на модул синхронизиран АЦП

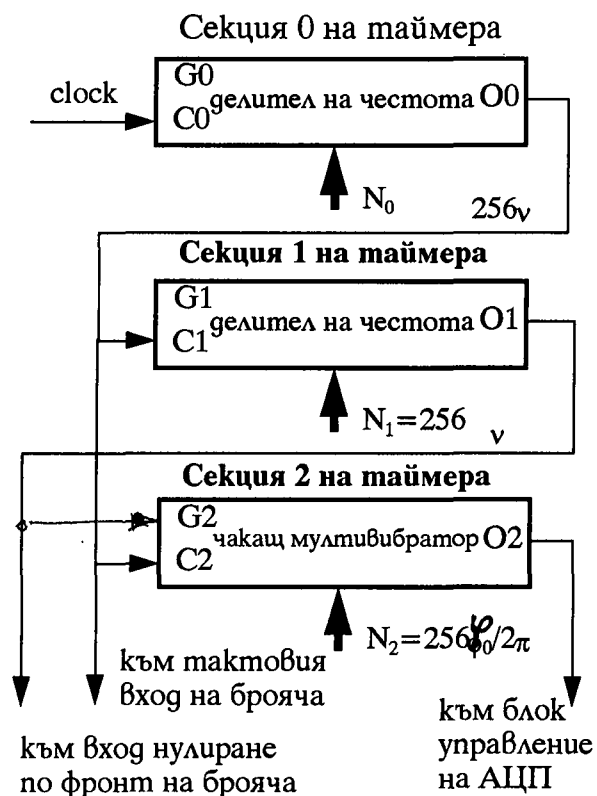
та памет. Това е генераторът на въздействието върху изследвания обект. Амплитудата $R(t)$ е постоянна и равна на опорното напрежение на ЦАП. При необходимост от промяната ѝ е необходимо да се постави втори ЦАП, чието изходно напрежение да се задава от микропроцесорната система. Изходното напрежение на втория ЦАП ще бъде опорно напрежение на ЦАП на функционалния генератор. Този блок изпълнява ролята на генератора и модулятора от блоковата схема от фиг. 8.2.

8.2.2.2. Аналогово-цифров преобразувател

Приемната част на модула синхронизиран АЦП е един четириканален аналогово-цифров преобразувател изграден по класическа схема с четири операционни усилвателя за входните сигнали, аналогов мултиплексор AMUX, схема следене/запомняне (S/H) и монолитен аналогово-цифров преобразувател, свързан към паралелния интерфейс адаптер PPI. Особеността тук е в управляващата логика, която стартира аналогово-цифровото преобразуване. След постъпване на активен фронт на входа на блока Управление се извършва привеждане на схемата следене/запомняне в състояние "запомняне", изчаква се времето за преходния процес (няколко микросекунди) и се подава стартиращ сигнал на АЦП. След края на преобразуването АЦП подава сигнал към микропроцесорната система посредством PPI за прочитане на резултата от преобразуването. Както вече беше отбелязано този блок играе ролята на синхронния детектор от блоковата схема на фиг. 8.2.

8.2.2.3. Синхронизираща част

Аналогово-цифровият преобразувател се стартира по сигнала, постъпващ от таймера РТМ с честота ν . Фазата на момента за стартиране може да бъде изменяна по програмен път спрямо генерирания от функционалния генератор въздействащ сигнал. За целта трите секции на програмуемия таймер РТМ са свързани по показаният на фиг. 8.4 начин. Първата секция е програмно управляем делител на еталонната честота clock. С помощта на коефициента на деление N_0 се задава честота 256ν пъти по-голяма от необходимата ни честота ν . Тази честота се подава за тактуване на брояча на функционалния генератор и на броячите на другите две секции на таймера. Втората секция дели на фиксираното число $N_1 = 256$ и се използва за получаване на сигнал с честота ν и нулева фаза. Той нулира броячът CNT на функционалния генератор с което се гарантира синфазната работа на функционалния генератор. Освен това се използва за запускане на третата секция на таймера, която работи като чакащ мултивибратор с програмуемо време на закъснение. Изходът на втория таймер се използва за запускане на аналогово-цифровото преобразуване. С помощта на числото N_2 се задава фазата φ_0 . По този начин третата секция се използва като фазовъртяща верига.



Фиг. 8.4. Свързване на секциите на програмуемия таймер

8.2.2.4. Нискочестотен филтър

Ролята на нискочестотен филтър се поема от програмното осигуряване за управление на модула синхронизиран АЦП. След натрупване на определено количество измервания (от десетина до хиляда) в зависимост от необходимото време за интегриране, се извършва усредняване на получените измервания. Полученото число е в същност изхода на синхронния детектор за зададената чрез таймера фаза.

8.3. Управление и използване на синхронизирания АЦП

Модулът синхронизиран АЦП и програмното осигуряване за управлението му могат да се използват в много различни режими на работа. Функционалните възможности се повишават значително от разнообразните изходни сигнали на функционалния генератор, записани в постоянната му памет и избирани програмно. Изходното напрежение на функционалния генератор се подава на входа на изследвания обект и се измерва реакцията му. Възможни са различни приложения на такава опитна постановка.

8.3.1. Снемане на амплитудночестотни и фазочестотни характеристики

Задава изходният сигнал да бъде синусоидален с дадена честота. Снема се реакцията на обекта, като се измерва амплитудата на изходния сигнал и фазовото закъснение. Процесът се повтаря с дадена стъпка за целият обхват от честоти.

Получаването на резултат от едно измерване (изходно напрежение при зададена фаза) се забавя от необходимото време за натрупване на определения брой дискрети за усредняване. За да се получи максимум на изходния сигнал е необходимо да се изменя фазата (числото във втората секция на таймера). За да се изчерпят всички възможни фази е необходимо да се направят 256 единични измервания. Максималната стойност на изходното напрежение и фазата, при която е получена, дават коефициента на предаване на изследвания обект (като се раздели на изходното напрежение на функционалния генератор) и как се изменя фазата за тази честота. Съществуват алгоритми за получаване на фазата и максималната стойност без да се правят всичките 256 измервания. Те ще бъдат коментирани в една от следващите глави. Възможно е да се намери фазата, при която изходното напрежение е максимално, с по-малко от шестнадесет измервания.

8.3.2. Снемане на импулсна характеристика на обект

Задава се изходният сигнал да бъде дираков импулс. Периодът на повторението му трябва да е по-голям от времето за затихване на реакцията на обекта — импулсната характеристика. Снема се реакцията на обекта, като последователно се увеличава закъснението на запускане на АЦП спрямо началото на импулса (фазата). Този метод се нарича изследване на процеси с развивка. Във времето разстоянието между получените точки (дискрети) от импулсната характеристика е един период на честотата 256*v*. Това позволява да се получи еквивалентна честота на дискретизация много пъти по-голяма от макси-

малната честота на работа на използваният АЦП. Подобен принцип се използва в т.нар. стробиращи осцилоскопи.

След получаване на точките от импулсната характеристика с изчисляване на спектъра ѝ можем да получим и честотните характеристики на обекта.

8.3.3. Снемане на преходна характеристика на обект

Задава се изходният сигнал да бъде правоъгълен и се снима реакцията на обекта по начина, описан в предишната точка. Периодът на правоъгълния сигнал трябва да бъде повече от два пъти по-голям от времето за приключване на преходния процес. От получената преходна характеристика може да бъде направена идентификация на обекта като линейна дискретна система (т.е. реда и коефициентите на уравнението на модела на обекта) с цел проектиране на регулатор или друга система за управление.

8.3.4. Допълнителни приложения

По подобен начин могат да бъдат измервани и спектър на сигнала, коефициент на нелинейни изкривявания на усилвател, коефициент на интермодуляционни изкривявания и т.н. Многообразието на приложенията на модула синхронизиран АЦП се допълва от наличието на няколко входни канала, което позволява снемането на характеристиките в няколко точки на изследвания процес или обект.

Обобщение

Синхронното детектиране се използва в случаите, когато полезният сигнал, резултат от въздействие върху някакъв процес или обект, е сравним по амплитуда с шумовете. В следствие на това се създават затруднения за неговото точно измерване. Методът се състои в модулиране на синусоидален сигнал с въздействието върху обекта. В синхронния детектор реакцията се умножава с опорен сигнал, получен след фазово изместване от генератора на синусоидалния сигнал. След умножаването се извършва усредняване. Получава се сигнал пропорционален на средната стойност на реакцията на обекта и на фазовата разлика му с опорния сигнал. Посредством изменение на дефазиранието на опорния сигнал се извършва намиране на максималната стойност на изходния сигнал. Резултатът от едно измерване е амплитуда на реакцията и фазовото ѝ изместване. Смуцаващите сигнали и шумове са практически изчезнали. Силните филтриращи свойства се дължат на това, че методът е чувствителен не само към честотата на входния сигнал, както теснолентовия филтър, но и към фазата му.

9. Многомерни сигнали и методи за тяхната обработка



Ключови думи

- многомерен сигнал
- изображение
- обработка на изображения
- многомерно преобразуване на Фурие
- многомерна цифрова филтрация
- теорема за дискретизация на многомерни сигнали

Какви знания и умения ще придобиете?

Къде се прилагат?

Тази глава е посветена на многомерните сигнали и методите за тяхната обработка. В реалната практика един от най-често срещаните многомерни сигнали е двумерното изображение. Част от въпросите са посветени изрично на обработката на изображения. Ще се запознаете с математическото дефиниране на трите основни вида многомерни сигнали - непрекъснат, дискретизиран и квантуван. Ще бъдат дефинирани условията за дискретизация без загуба на информация и необходимият брой нива за квантуване.

Получените знания ще приложите при проектирането на алгоритмите и програмите за обработка на многомерни сигнали. Приложими са както при обработката на медицински диагностични образи (рентгенограми, ехограми, томографски изображения), така и навсякъде, където се въвеждат и обработват изображения.

Въведение

Реалният живот е сложен и много често процесите в него не могат да бъдат описани от функция на един параметър. Затова се налага да бъдат дефинирани и разгледани многомерните сигнали - функции на повече от един параметър. Един от най-често срещаните многомерни сигнали е неподвижното изображение - яркостта е функция на два параметъра (широчина и височина). В случай на движещо се изображение яркостта е функция на три параметъра - широчина, височина и време. Разликата между едномерен сигнал и двумерен е качествена и изисква различен подход и методи за обработка, а разликата между сигналите с две, три и повече измерения е вече само количествена. Независимо от това, при разглеждането на многомерните сигнали ще се придър-

жаме към последователността с която разглеждахме едномерните сигнали в предишните глави.

Обработката на многомерните сигнали има няколко съществени различия спрямо обработката на едномерни сигнали:

- ⇒ обикновено за решаване на задачите по обработка на многомерни сигнали е необходимо да се използва значително по-голям обем от данни;
- ⇒ математическите методи за обработка на многомерни сигнали нямат тази завършеност, характерна за методите за обработка на едномерни сигнали;
- ⇒ многомерните системи имат значително повече степени на свобода, в резултат на което проектирането има гъвкавост, която не е характерна за едномерните системи.

9.1. Дефиниране на многомерен сигнал

9.1.1. Непрекъснат многомерен сигнал

Многомерният сигнал $a(x_1, x_2, \dots, x_n)$ се описва от непрекъснатата или непрекъснатата по части функция на независимите променливи x_1, x_2, \dots, x_n , приемащи стойности в даден обхват. Амплитудата на многомерния сигнал е също ограничена:

$$a(x_1, x_2, \dots, x_n) \left\{ \begin{array}{l} x'_1 \leq x_1 \leq x''_1 \\ x'_2 \leq x_2 \leq x''_2 \\ \vdots \\ x'_n \leq x_n \leq x''_n \\ a' \leq a \leq a'' \end{array} \right.$$

На тези условия отговарят всички сигнали в природата. Един типичен пример за многомерен сигнал е двумерното черно-бяло изображение. Стойността на функцията $a(x_1, x_2)$ е пропорционална на яркостта на изображението и зависи от x_1 и x_2 - координатите по широчина и височина в изображението. Както яркостта приема стойности в ограничен интервал - от черното до бялото, така и независимите променливи също приемат стойности в ограничен интервал - размерите на изображението.

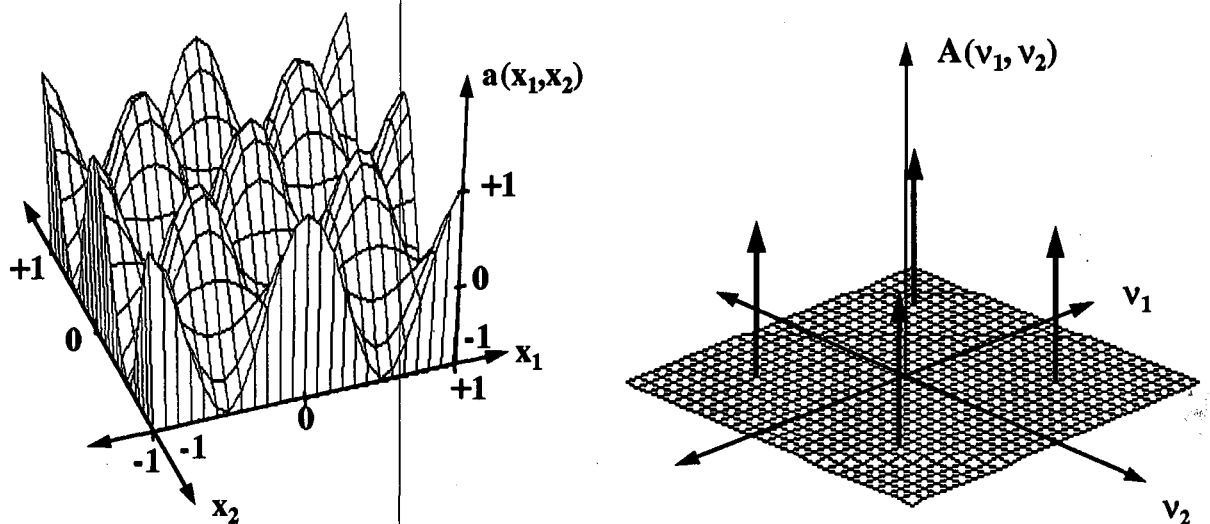
По аналогичен на едномерните сигнали начин се дефинира и спектъра на многомерен сигнал $A(\nu_1, \nu_2, \dots, \nu_n)$, който еднозначно представя същия сигнал, но в честотна област. Ако размерността на независимата променлива x_i е метър, то размерността на ν_i ще бъде

метър⁻¹, което има физически смисъл на брой линии за метър. Преминването в честотна област и обратно е без загуба на информация:

$$a(x_1, x_2, \dots, x_n) \Leftrightarrow A(v_1, v_2, \dots, v_n).$$

По аналогия с едномерните сигнали ще означаваме многомерните сигнали с малки латински букви, а техните спектри - със съответната главна латинска буква.

За краткост от тук нататък ще работим само с двумерни сигнали. Това важи за всички формули, примери и т.н. Формулите за сигнали с повече измерения са аналогични. На фиг. 9.1 са дадени двумерен косинусоидален сигнал и неговият двумерен амплитуден спектър. Направете аналогия с фиг. 5.1!



а) двумерен косинусоидален сигнал

б) двумерен амплитуден спектър на този сигнал

Фиг. 9.1. Многомерен сигнал и неговият спектър

9.1.2. Дискретен многомерен сигнал

9.1.2.1. Дефиниране

Дискретният многомерен сигнал се описва от гребеновидната функция $a(\vec{X}_k)$, която е ограничена по амплитуда. Независимата променлива \vec{X}_k е вектор, съдържащ списък от стойности на координати в които дискретният многомерен сигнал е дефиниран, т.е. притежава стойност. Частен случай на дискретния многомерен сигнал е двумерният дискретен сигнал с равноотдалечени координати на точките на

дискретизация $a(n_1, n_2)$, където n_1 и n_2 са цели числа, $x_1 = \Delta_1 \cdot n_1$, $x_2 = \Delta_2 \cdot n_2$. Тук Δ_1 и Δ_2 са периодите на дискретизация по двете координати.

9.1.2.2. Дискретизиращи функции

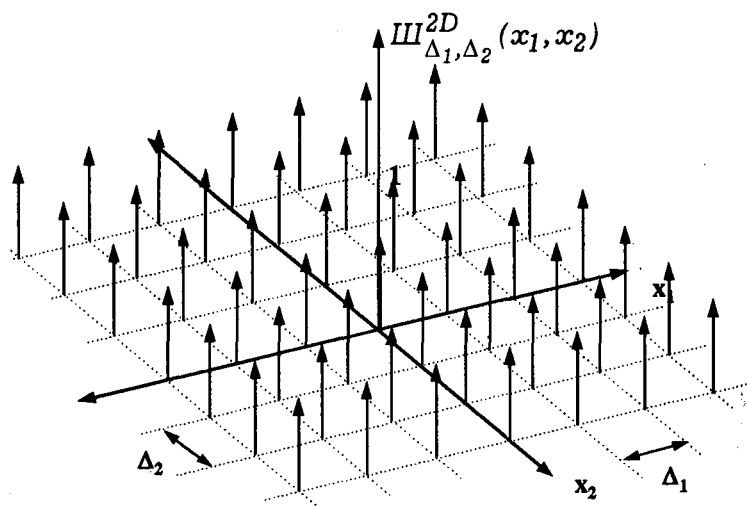
При едномерните сигнали използвахме дискретизираща функция, състояща се от равноотдалечени във времето гиракови импулси. Двумерният аналог на тази функция е

$$\text{Ш}_{\Delta_1, \Delta_2}^{2D}(x_1, x_2) = \sum_{n_1=-\infty}^{+\infty} \sum_{n_2=-\infty}^{+\infty} \delta(x_1 - n_1 \cdot \Delta_1, x_2 - n_2 \cdot \Delta_2),$$

където $\delta(x_1, x_2)$ е двумерен гираков импулс, дефиниран както следва:

$$\delta(x_1, x_2) = \begin{cases} 1, & x_1 = x_2 = 0 \\ 0, & x_1 \neq x_2 \neq 0 \end{cases}$$

Това е най-често използваната дискретизираща функция, показана е на фиг. 9.2. При нея двумерният сигнал е правоъгълна матрица от числа - стойностите на сигнала за съответната координата. Обикновено стъпката на дискретизация по двете оси е еднаква - $\Delta_1 = \Delta_2 = \Delta$. Дискретизи-

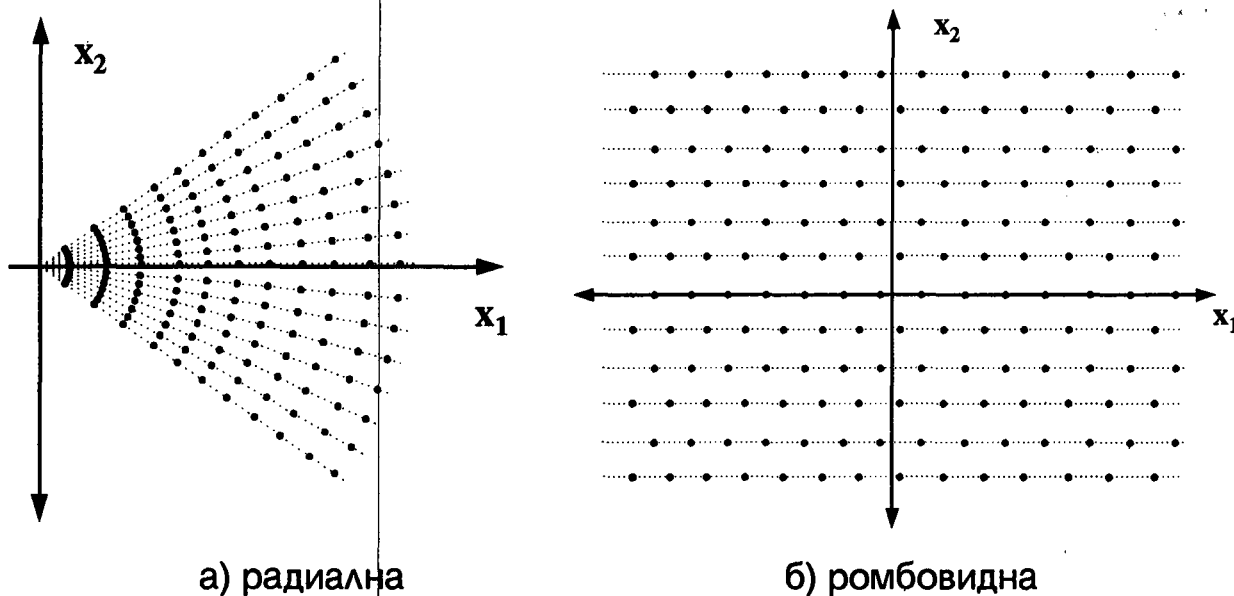


Фиг. 9.2. Правоъгълна двумерна дискретизираща функция с равномерна стъпка

раните по този начин двумерни сигнали се обработват най-лесно от компютърните системи и се изобразяват най-бързо на екрана на компютъра ако става въпрос за изображения.

По редица причини се използват и други дискретизиращи функции. Когато двумерният сигнал се формира като резултат от изпращане на импулс и приемане на отразен сигнал с движеща се приемо-предавателна част (радиолокация, ехография) точките на дискретизиращата функция се задават като ъгъл и разстояние, т.е. използват се радиални координати, както е показано на фиг. 9.3.а). В други случаи се предпочита ромбовидната дискретизираща функция, показана на фиг. 9.3.б), вместо правоъгълната - при един и същи обем на предавано-

то по теснолентов съобщителен канал изображение у зрителя се постига усещане за по-добро качество.



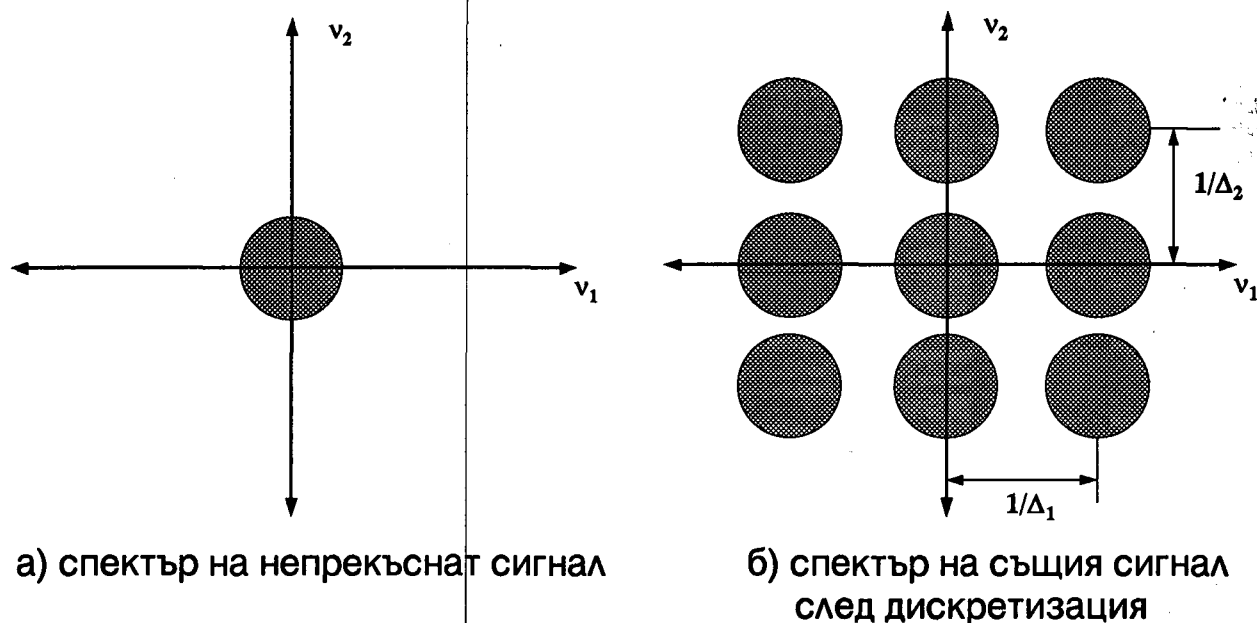
Фиг. 9.3. Други видове двумерни дискретизиращи функции



Почти всички методи за обработка на многомерни сигнали предполагат дискретизиране с правоъгълна дискретизираща функция. Затова, независимо от начина по който е дискретизиран и въведен в компютъра конкретния сигнал, след даден етап на обработка се извършва редискретизация с правоъгълна дискретизираща функция.

9.1.2.2.3. Процес на дискретизация

По аналогичен за едномерните сигнали начин дискретизацията е процес на получаване на дискретен многомерен сигнал от непрекъснат многомерен сигнал. Математически се описва като умножение на непрекъснатия сигнал с дискретизиращата функция. Спектърът на многомерния сигнал по аналогичен на едномерния случай начин се размножава по всяка от осите на честотата с период $\frac{1}{\Delta_i}$, където Δ_i е стъпката на дискретизация по съответната ос. Това е показано нагледно на фиг. 9.4. Този извод важи за правоъгълна дискретизираща функция. При използване на друг вид дискретизираща функция резултатът е по-добен, но математически се описва по-сложно.



Фиг. 9.4. Размножаване на спектъра на двумерен сигнал след дискретизация

9.1.3. Квантуван многомерен сигнал

Квантуваният многомерен сигнал приема само определени стойности (нива). Математическото описание на процеса на квантуването е абсолютно аналогично на квантуването на едномерен сигнал, което е дадено в т. 4.1.3. Всички въведени в паметта на компютърните системи многомерни сигнали са както дискретизирани, така и квантувани.

9.2. Многомерно преобразуване на Фурие и свойства

9.2.1. Многомерно преобразуване на Фурие

Многомерното преобразуване на Фурие ни дава спектъра на многомерния сигнал и се извършва посредством формулата:

$$A(v_1, v_2) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} a(x_1, x_2) \cdot e^{-2 \cdot \pi \cdot j \cdot (x_1 \cdot v_1 + x_2 \cdot v_2)} \cdot dx_1 \cdot dx_2.$$

$A(v_1, v_2)$ се нарича Фурие образ или честотен спектр на $a(x_1, x_2)$ и е комплексна функция на два аргумента. Аналогично на едномерния случай се дефинират понятията амплитуден и фазов многомерен спектр.

9.2.2. Многомерно обратно преобразуване на Фурие

Многомерното обратно преобразуване на Фурие се дава от израза:

$$a(x_1, x_2) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} A(v_1, v_2) \cdot e^{2 \cdot \pi \cdot j \cdot (v_1 \cdot x_1 + v_2 \cdot x_2)} \cdot dv_1 \cdot dv_2.$$

9.2.3. Условия за съществуване на многомерното преобразуване на Фурие

Може да се докаже, че достатъчно условие за съществуването на многомерното преобразуване на Фурие е многомерният сигнал да е абсолютно сумируем:

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} |a(x_1, x_2)| \cdot dx_1 \cdot dx_2 < \infty.$$

Многомерни сигнали, отговарящи на по-слабото условие:

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} |a(x_1, x_2)|^2 \cdot dx_1 \cdot dx_2 < \infty,$$

може да нямат непрекъснат Фурие образ, но той е напълно определен с изключение на точките на прекъсване.

9.2.4. Основни свойства на многомерното преобразуване на Фурие

Основните свойства на многомерното преобразуване на Фурие са подобни на свойствата на преобразуването на Фурие на едномерни сигнали: линейност, пространствено изместване, производна. Формулите за връзка са аналогични на дадените в т. 5.1.4 и затова няма да ги цитираме тук. Освен това има и някои нови свойства:

- а) транспониране - $a(x_2, x_1) \Leftrightarrow A(v_2, v_1)$;
- б) огледално отражение - $a(-x_1, x_2) \Leftrightarrow A(-v_1, v_2)$,
 $a(x_1, -x_2) \Leftrightarrow A(v_1, -v_2)$ и
 $a(-x_1, -x_2) \Leftrightarrow A(-v_1, -v_2)$.

9.2.5. Дискретно многомерно преобразуване на Фурие

Спектърът на дискретизиран многомерен сигнал може да бъде изчислен за всяка стойност на честотите v_1 и v_2 :

$$A(v_1, v_2) = \sum_{n_1 = -\frac{N}{2}}^{+\frac{N}{2}} \sum_{n_2 = -\frac{N}{2}}^{+\frac{N}{2}} a(n_1, n_2) \cdot e^{-j \cdot \frac{2\pi}{N} \cdot (n_1 \cdot v_1 + n_2 \cdot v_2)}$$

Формулата е дадена за дискретизирано изображение с еднакъв брой точки на дискретизация по двете оси - N и правоъгълна дискретизираща функция. По аналогичен начин се получава и формулата за обратното многомерно преобразуване на Фурие. Тя ни дава възможност от дискретизирания многомерен спектър да получим стойността на коя да е точка на многомерния сигнал. Всичко казано в т. 5.1.7 за обема на изчисленията се повдига на степен - размерността на сигнала. Това означава повдигане на квадрат за двумерни сигнали.

9.3. Дискретизиране и квантуване на многомерни сигнали

9.3.1. Теорема за дискретизацията на многомерни сигнали

Ако за спектъра $A(v_1, v_2)$ на двумерния сигнал $a(x_1, x_2)$ е известно, че $A(v_1, v_2) \equiv 0$ за всяко $v_1 > v_{1max}$ и $v_2 > v_{2max}$ то необходимо и достатъчно условие периодичното повторение на спектъра, предизвикано от дискретизацията със стъпка Δ_1 и Δ_2 по всяка от осите да не измени повтарящия се спектър е $\Delta_1 < 2 \cdot \frac{1}{v_{1max}}$ и $\Delta_2 < 2 \cdot \frac{1}{v_{2max}}$. Това е двумерният аналог на теоремата на Шенон за дискретизирането на едномерни сигнали.

9.3.2. Апроксимационна формула за многомерни сигнали

След извършване на математически операции, подобни на дадените в т. 4.2.2 се получава апроксимационната формула за двумерни сигнали:

$$\tilde{a}(x_1, x_2) = \sum_{n_1} \sum_{n_2} a(n_1, n_2) \cdot \frac{\sin(\pi \cdot \frac{x_1 - n_1 \cdot \Delta_1}{\Delta_1})}{\pi \cdot \frac{x_1 - n_1 \cdot \Delta_1}{\Delta_1}} \cdot \frac{\sin(\pi \cdot \frac{x_2 - n_2 \cdot \Delta_2}{\Delta_2})}{\pi \cdot \frac{x_2 - n_2 \cdot \Delta_2}{\Delta_2}}$$

Тази формула е двумерният аналог на апроксимационната формула на Шенон. Абсолютно по същия начин може да се използва и апроксимация от нулев и от първи ред, както и сплайн апроксимация. Разликата е, че тук координати са повече с което нараства и сложността на използваните формули.



Апроксимационната формула в този си вид е валидна само за най-простата дискретизираща функция - правоъгълната. При използване на друга дискретизираща функция математическото извеждане на формулата за апроксимация използва значително по-сложен мате-

матически апарат. Затова няма да се спираме на случаите с различна от правоъгълната дискретизираща функция.

9.3.3. Избор на брой нива за квантуване

Въпросът за избора на броя на нивата за квантуване тук е значително по-сложен, макар че могат да се направят математически изводи по подобие на направените за едномерния случай. Много често броят на нивата се определя от предназначението на дискретизирания и квантуван многомерен сигнал. В случай на въвеждане на черно-бели изображения, предназначени само за разглеждане от човек, са достатъчни 30-40 нива на сивото, защото толкова се възприемат от човешкото око. По тази причина повечето от АЦП, предназначени за въвеждане на телевизионни изображения са шест или най-много осем битови, което означава от 64 до 256 нива на сивото. Ако ще се извършват някакви обработки над въведеното изображение 30-40 нива на сивото са крайно недостатъчни.

9.4. Многомерна цифрова филтрация

Многомерните цифрови филтри са многомерни дискретни системи. Подобно на едномерните те имат своите честотни, амплитудночестотни и фазочестотни характеристики. Реакцията на многомерен гураков импулс се нарича многомерна импулсна характеристика на филтъра. Синтезът на двумерни и многомерни цифрови филтри се отличава съществено от синтеза на едномерните. Както при едномерните, обаче, различаваме рекурсивни и нерекурсивни филтри.

9.4.1. Нерекурсивни многомерни цифрови филтри

Нерекурсивните многомерни цифрови филтри са винаги устойчиви. Едно от най-важните им преимущества пред рекурсивните е във възможността да се синтезира нерекурсивен филтър с нулева фазочестотна характеристика, подобно на едномерните нерекурсивни цифрови филтри с линейна фазочестотна характеристика. Многомерните цифрови филтри с нулева фазочестотна характеристика са основен инструмент при обработката на изображения. Ненулевата фазочестотна характеристика на многомерния филтър довежда до разсейване на контрастните точки, линии и граници на филтрираното изображение.

9.4.1.1. Дефиниране на нерекурсивен многомерен цифров филтър

Всяка точка на изходния двумерен сигнал $b(n_1, n_2)$ се получава от входния двумерен сигнал $a(n_1, n_2)$ по формулата:

$$b(n_1, n_2) = \sum_{k_1} \sum_{k_2} h(k_1, k_2) \cdot a(n_1 - k_1, n_2 - k_2).$$

Тук матрицата $h(k_1, k_2)$ с коефициентите на нерекурсивния филтър е в същност импулсната характеристика на двумерния цифров филтър, подобно на едномерния цифров нерекурсивен филтър.

9.4.1.2. Реализиране на многомерен нерекурсивен цифров филтър

Един от често използваните методи за синтез на многомерен нерекурсивен филтър е многомерен вариант на метода, описан в т. 6.5.3.2. Други методи, които се използват, са синтез с помощта на теглови прозорци и синтез по метода на най-малките квадрати. Те са описани в специализираната литература за обработка на изображения и многомерни сигнали.



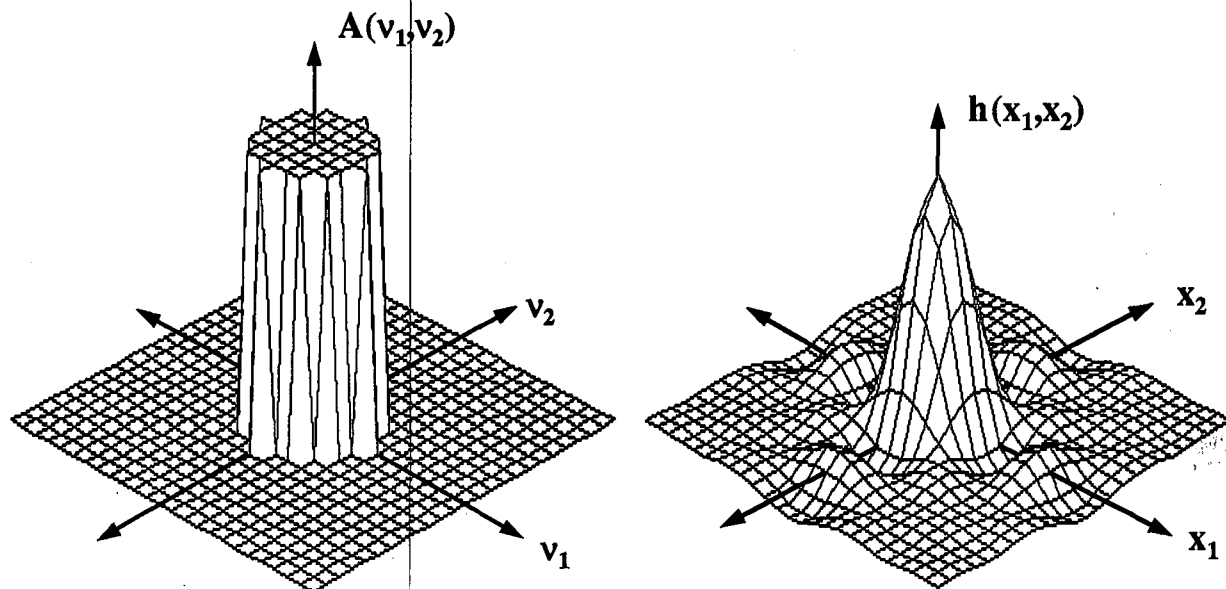
Пример за многомерен нерекурсивен цифров филтър

Нека да е необходимо да се проектира двумерен цифров филтър, аналог на нискочестотния едномерен филтър, който да пропуска сигнали с честоти ν_1 и ν_2 , отговарящи на условието $\sqrt{\nu_1^2 + \nu_2^2} \leq \rho$ и да спира сигналите с честоти неотговарящи на това условия. Тази амплитудночестотна характеристика е двумерен аналог на правоъгълната едномерна функция, разглеждана в глава 4. Графично е показана на фиг. 9.5.а). За целта се подготвя един масив с размерност $N \times N$, където N е желаният размер на филтъра. След извършване на обратно двумерно преобразуване на Фурие се получава импулсната характеристика на проектирания филтър, която е показана на фиг. 9.5.б). Тази импulsна характеристика е двумерен аналог на функцията $\frac{\sin(x)}{x}$. В случай на проектиране на нерекурсивен двумерен цифров филтър с нулева фаза това е направо матрицата с коефициентите на филтъра.

9.4.2. Рекурсивни многомерни цифрови филтри

Всяка точка на изходния двумерен сигнал $b(n_1, n_2)$ се получава от входния двумерен сигнал $a(n_1, n_2)$ по формулата:

$$b(n_1, n_2) = \sum_{k_1} \sum_{k_2} m(k_1, k_2) \cdot a(n_1 - k_1, n_2 - k_2) - \sum_{l_1} \sum_{l_2} n(l_1, l_2) \cdot b(n_1 - l_1, n_2 - l_2)$$



а) желана амплитудночестотна характеристика б) получена импулсна характеристика, респективно коефициентите на нерекурсивния филтър

Фиг. 9.5. Проектиране на двумерен нерекурсивен цифров филтър с нулева фаза

Във втората сума $(l_1, l_2) \neq (0, 0)$. Тук матрицата $m(k_1, k_2)$ е с коефициентите на нерекурсивната част, матрицата $n(l_1, l_2)$ е с коефициентите на рекурсивната част.

Самият филтър като изчислителна процедура е доста сложен - трябва да се предвиди последователност на изчисления при която предишните стойности, използвани в рекурсивната част, да са вече изчислени. Това още повече касае синтеза на многомерните рекурсивни цифрови филтри. Цялостна и стройна теория не е създадена, повечето методи за синтез са итеративни.

9.4.3. Реализация на многомерните цифрови филтри

Поради сложността си и големия брой изчисления многомерните цифрови филтри като правило се реализират във вид на програма за процесор с общо предназначение, когато многомерната филтрация е

част от сериозна обработка на многомерен сигнал. При специализирани системи или когато се търси висока производителност се използва програмна реализация на многомерен филтър за процесор за цифрова обработка на сигнали.

9.5. Цифрова обработка на изображения

Неподвижното изображение е двумерен сигнал, чиято амплитуда определя яркостта на съответната точка. Английският термин за точка от изображението, често използван и в литературата на български език е пиксел (от pictures element - елемент на картината). В зависимост от нуждите яркостта на всяка точка може да бъде квантувана на различен брой нива: две (само черно и бяло), 16, 256 или повече степени на сивото.

9.6.1. Въвеждане на изображения в компютърните системи

Основните начини за въвеждане на изображения в компютърните системи са:

- ⇒ скенер - за въвеждане на черно-бели и цветни изображения от лист хартия, фотоснимка или диапозитив с разрешаваща способност от 150 до над 1200 точки на инч (от 6 до над 48 точки на милиметър);
- ⇒ телевизионна камера и специализиран модул АЦП - за въвеждане на неподвижни или движещи се изображения с размерност от 160 на 100 точки до 600 на 800 точки и честота от един до тридесет кадъра за секунда;
- ⇒ специализирани модули АЦП за въвеждане на сигнали от радиолокационни станции, ехографски апарати и др.;
- ⇒ специализирана апаратура за въвеждане на данни, от които след математически операции се възстановява изображение, например томография, сеизмографска информация, океанографска информация, радиоастрономия и др.

В повечето случаи изображението се получава след развивка във времето. При телевизията изображението се предава ред след ред докато се формира един кадър, след което се започва предаването на следващия кадър и т.н. В радиолокаторите и ехографите развивката е радиална - изстрелва се сондиращ импулс и се приемат отраженията от обектите, след това приемо-предавателната част се завърта на даден ъгъл, отново се изстрелва сондиращ импулс и т.н.

Ако не се използват никакви методи за цифрова компресия на данните изображенията заемат значително място в паметта на компю-

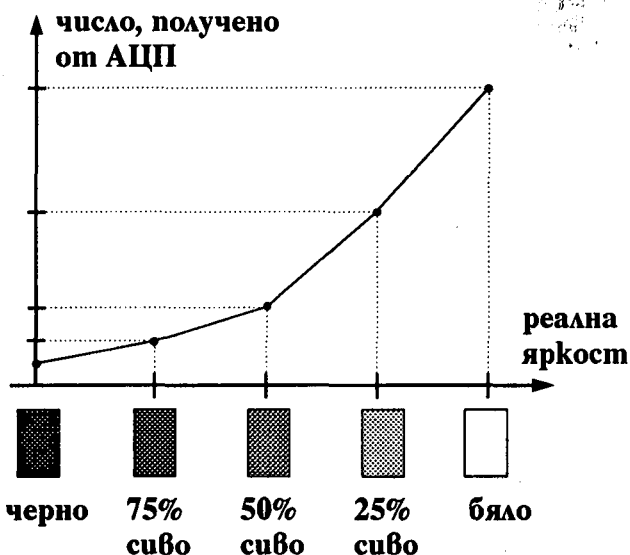
търната система. Например едно черно-бяло изображение 600 на 800 точки с 256 степени на сивото (един байт на точка) заема в паметта 480 000 байта.

9.6.2. Предварителна обработка на изображението

Целта на предварителната обработка на изображението е да се компенсират изкривяванията, внесени в него от системата за въвеждането му в компютърната система. Всяка от изброените предварителни обработки се прилага само при необходимост. След максимално компенсиране на тези изкривявания полученото изображение става годно не само за визуализация. По него могат да се извършват и количествени измервания на яркост, геометрични размери, разстояния и др. Като правило калибрирането на входния тракт на изображението се извършва чрез въвеждане на тестови изображения, или чрез обработка на изображения с известни характеристики.

9.6.2.1. Коририране на преходните изкривявания

Целта е да се коригира евентуалната нелинейност на преобразуването на амплитудата по целия канал - от яркостта на обекта до числото, получено в компютърната система. Използват се методите, описани в т. 3.4. Подава се тестово изображение, в което има участъци с различна степен на сивото, например: бяло, 10% сиво, 25% сиво, 50% сиво, 75% сиво, 90% сиво и черно. От въведеното изображение се локализират координатите на всеки участък и неговата яркост се усреднява. В резултат се получава предавателната характеристика: число в компютъра - яркост на обекта. Една примерна предавателна характеристика е показана на фиг. 9.6. Извършва се нейната апроксимация. След въвеждане на реално изображение се прави преизчисляване на стойността на всеки елемент от него с помощта на апроксимиращата функция.



Фиг. 9.6. Коририране на преходните изкривявания

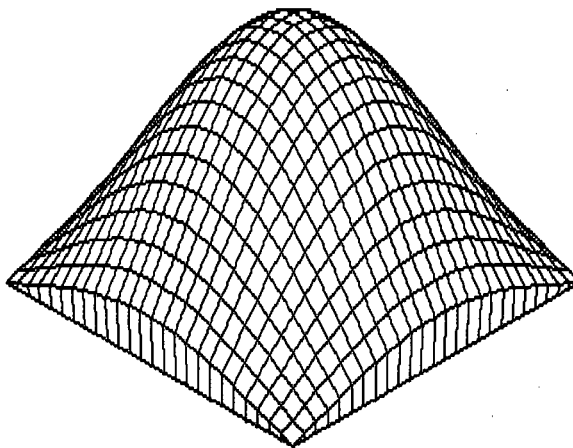
В редица случаи се прави умишлено изкривяване на преходната функция. В телевизията, например, преди предаване на яркостния сиг-

нал се прави така наречената γ - корекция, която прави преходната характеристика по-стръмна при по-малките яркости на сигнала. Това компенсира нелинейната зависимост на яркостта на екрана на телевизора в зависимост от подаденото на катода на кинескопа напрежение. С това се подобрява предаването на яркостта на по-тъмните обекти при възпроизвеждане на екрана на телевизора. При въвеждане в компютърна система на изображения от телевизионен сигнал е необходимо да се направи обратна γ корекция, с което да се компенсират предварителните изкривявания въведени в телевизионния център.

За коригиране на яркостния канал в рентгенографията се използват тестови изображения на метални плочи с различна дебелина, които спират различно рентгеновото излъчване. Където не е възможно подаване на тестово изображение се търсят косвени методи. Например при обработка на изображения на звезди в астрономията за настройка може да се използват наблюдения на няколко звезди с известна яркост (звездна величина).

9.6.2.2. Коририране на амплитудните изкривявания

Амплитудните изкривявания се проявяват като зависимост на въведената яркост от геометричните координати. Тестовото изображение е равномерно сива повърхност. След въвеждане се прави оценка на неравномерността на яркостта на отделните точки на изображението. Типично изкривяване тук е зависимост на яркостта от ъгъла под който пада сканиращия изображението лъч (обикновено $\cos \alpha_1 \cdot \cos \alpha_2$) по всяка от осите, както е показано на фиг. 9.7. Тези изкривявания са характерни за видеокамерите и някои типове оптически скенери. Яркостта на въведеното равномерно сиво изображение е най-голяма в центъра и намалява към краищата на изображението. При необходимост се извършва преизчисляване на яркостта на изображението в зависимост от координатите на всяка точка по намерена от тестовото изображение апроксимираща двумерна функция. Обикновено се използва метода на най-

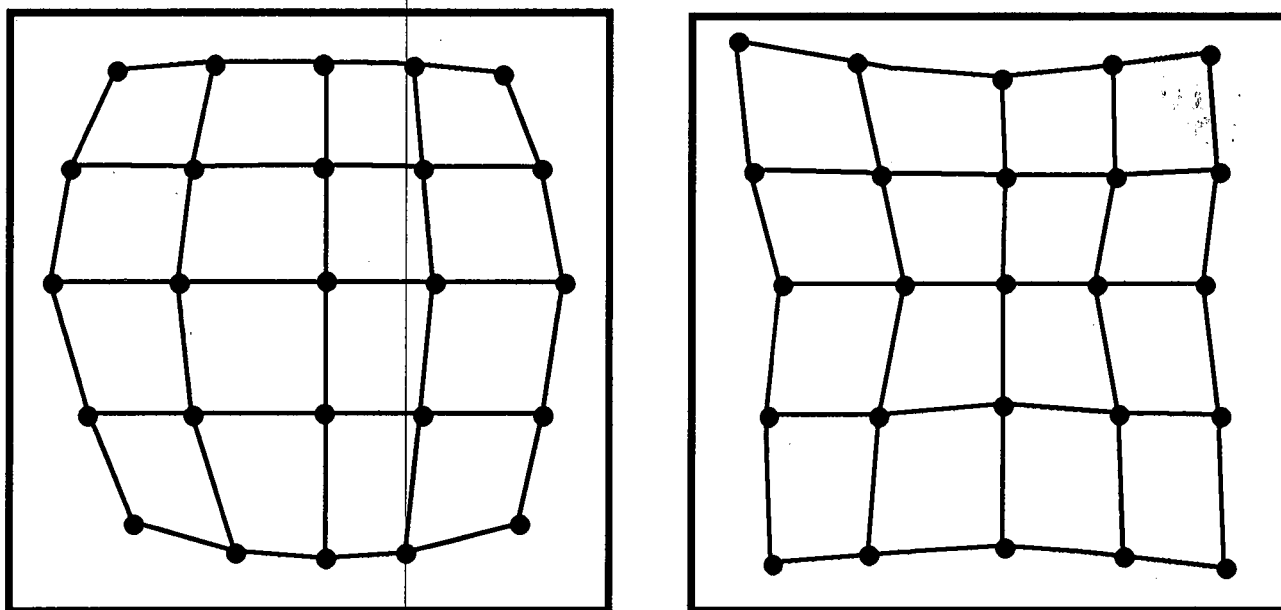


Фиг. 9.7. Амплитудни изкривявания в зависимост от геометричните координати - яркост на равномерно сива повърхност

малките квадрати. В отделни случаи тази и предишната корекция може да се съчетаят.

9.6.2.3. Коригиране на геометричните изкривявания

Геометричните изкривявания са изместване на точките на дискретизиращата функция от желаните координати. Например при въвеждане на изображения от телевизионна камера може да се окаже, че растрът (дискретизиращата функция) не е точно правоъгълен в следствие на нелинейното изменение на отклоняващото напрежение, което движи сканиращия лъч в телевизионната камера. Типични изкривявания са тип "възглавница" или тип "бъчва". На фиг. 9.8 са показани примерни геометрични изкривявания при въвеждане на мрежа от равноотдалечени точки.



Фиг. 9.8. Геометрични изкривявания на дискретизиращата функция

Геометричните изкривявания може да бъдат предизвикани и от фокусиращата оптика. Тестовото изображение в този случай е правоъгълна мрежа. След въвеждане се снемат координатите на пресечните точки на мрежата. Извършва се апроксимация на нелинейната дискретизираща функция. Така се получават истинските геометрични координати на точките на дискретизиращата функция. След това, с използване на апроксимиращата формула от т. 9.3.2, се извършва редискретизация на въведеното изображение с цел получаване на равномерна дискретизираща функция. Обработката заема изключително много време и се прави само в случай на значителни изкривя-

вания и необходимост от извършване на количествени измервания на разстояния, площи и др. от въведеното изображение.

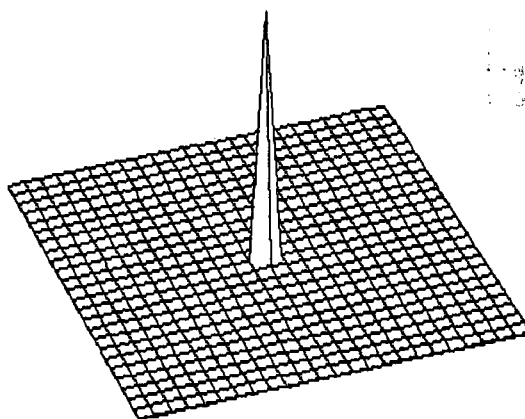
Тази обработка е почти задължителна при въвеждане на изображения с полярна дискретизираща функция, както е при радарните изображения и при изображенията от ехографска апаратура и се прави винаги, когато ще се извършват геометрични измервания или някаква по-сложна обработка на въведеното изображение.

9.6.2.3. Коригиране на честотните изкривявания

Честотните изкривявания са предизвикани от крайната лента на пропускане на апаратурата за въвеждане на изображението в компютърната система. В следствие на това при наличие във входното изображение на рязък преход от черно към бяло във въведеното изображение той се замазва и става повече или по-малко плавен. Черните точки на бял фон или белите точки на черен фон (двумерни гуракови импулси) също се замазват. Тяхната форма точно съответства на импулсната характеристика на цялата система за въвеждане на изображението в компютъра. На практика се извършва процес на двумерно свиване, подобно на едномерното свиване, описано в т. 4.1.1.2. Процесът се описва от модел от вида:

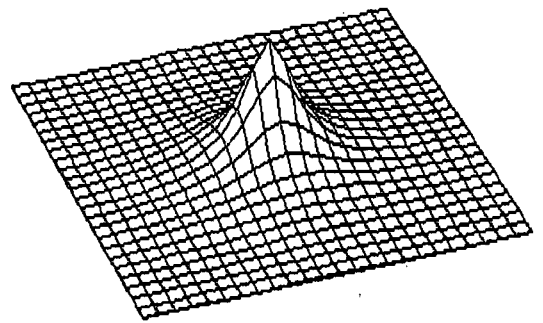
$$b(x_1, x_2) = D[a(x_1, x_2)],$$

където $a(x_1, x_2)$ е входното изображение, $b(x_1, x_2)$ - изходното, а $D[.]$ е изкривяващия оператор. Ако е известно входното изображение и изкривяващия оператор и се търси изходното изображение задачата се нарича *реализация на системата*. Типичен пример е двумерната цифрова филтрация. Ако са известни входното и изходното изображение и се търси изкривяващия оператор задачата се нарича *идентификация на системата*. Ако задачата е да се намери входното изображение по известно изкривено изходно изображение, както в този случай, тя се нарича *обратна задача*.



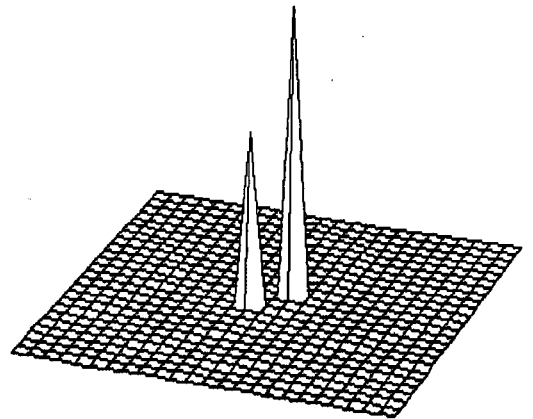
Фиг. 9.9. Тестово изображение - бяла точка на черен фон

Едно от елементарните решения на обратната задача се свежда до синтеза на двумерен цифров филтър с честотна характеристика обратна на честотната характеристика на апаратурата за въвеждане на изображения в компютърната система. Тестовото изображение е ^{ка} двумерен гираков импулс, като ^{на} позания на фиг. 9.9, или контрастен квадрант (например бял на черен фон). Като резултат в първия случай ще получим импулсната характеристика (фиг. 9.10), а във втория - преходната характеристика на изкривяващия оператор.



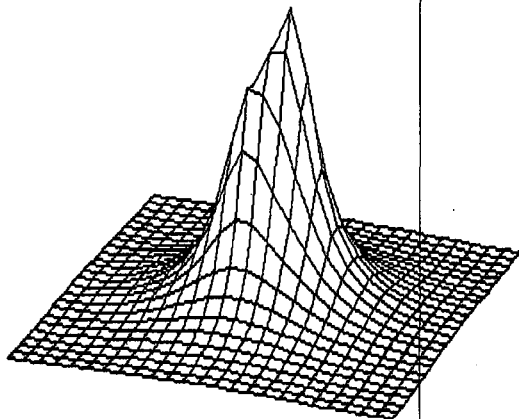
Фиг. 9.10. Тестовото изображение въведено с честотни изкривявания - импулсна характеристика на входния тракт

Задачата се свежда до синтеза на двумерен филтър, чрез който от изходното изображение да получим входното. След синтеза му всяко въведено изображение преминава през този филтър и получаваме коригирано изображение, максимално близко до реалното. Методът е годен за работа при сериозни изкривявания в честотна област предизвикани от входните устройства. Естествено не може да получим информация в честотна област изцяло потисната от тях. Трябва да се прилага внимателно, тъй като обикновено нараства шума в коригираното изображение.

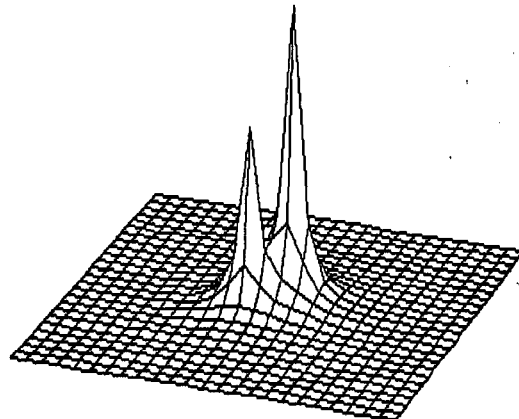


Фиг. 9.11. Реално изображение - бяла и светлосива точки на черен фон

На фиг. 9.11 е показано реално изображение, състоящо се от бяла и светлосива точки на черен фон. На фиг. 9.12 е показано същото изображение, въведено в компютърната система посредством входния тракт, чиято импулсна характеристика е показана на фиг. 9.10. След преминаване през подходящо проектиран коригиращ филтър се получава изображението показано на фиг. 9.13. Корекцията не е пълна, но полученият резултат е със значително по-добро качество от въведеното изображение.



Фиг. 9.12. Реалното изображение въведено в компютърната система



Фиг. 9.13. Реалното изображение след коригиране на честотните изкривявания на входния тракт

Съществуват и значително по-мощни итерационни методи за възстановяване на изкривени изображения, които се прилагат в конкретни случаи за конкретна апаратура.

9.6.3. Някои типични процедури за обработка на изображения

Освен общите за всички многомерни сигнали обработки (двумерен спектър, двумерна корелация и др.) при изображенията съществуват и някои специфични само за тях видове обработка. Тяхната цел е да се получи изображение, което да се вижда по-добре, цялото или отделни части от него. Нека приемем, че изображението е във вид на правоъгълна матрица с размерност M на N .



Фиг. 9.14. Тестово изображение

Стойността на всеки елемент на матрицата на изображението $a(i, j)$ определя яркостта му. Яркостта (респективно стойността на елементите) се изменя от $AMIN$ до $AMAX$. За изображение със 128 степени на сивото тези стойности са съответно 0 и 127. Повечето обработки ще бъдат дадени във вид на кратки програми на псевдоезик,

подобен на езика С. Тестовото изображение, върху което ще се демонстрират типовите обработки, е показано на фиг. 9.14.

9.6.3.1. Регулиране на яркостта и контраста

За да се повиши (понижи) яркостта на изображението е необходимо да се прибави (респективно извади) константа от всеки елемент на изображението. Нека константата да е ADB , която със своя знак определя повишаване или понижаване на яркостта. Обработката на изображението изглежда така:

```
for (i=0; i<N; i++) {
  for (j=0; j<M; j++) {
    a[i][j] = a[i][j] + ADB;
    if (a[i][j] > AMAX) a[i][j] = AMAX;
    if (a[i][j] < AMIN) a[i][j] = AMIN;
  }
}
```

За изменение на контраста (разликата между най-тъмния и най-светлия елемент) е необходимо да се умножи всеки елемент на изображението с константа. Когато константата е по-голяма от единица контрастът на изображението се повишава, когато е по-малка - контрастът се понижава. Нека константата е ADC , при което обработката на изображението изглежда така:

```
for (i=0; i<N; i++) {
  for (j=0; j<M; j++) {
    a[i][j] = ADC * a[i][j];
    if (a[i][j] > AMAX) a[i][j] = AMAX;
    if (a[i][j] < AMIN) a[i][j] = AMIN;
  }
}
```

Изменението на яркостта и контраста имат съществено значение при визуализацията на изображението. Както вече беше казано човешкото око различава до тридесетина степени на сивото. В много случаи в компютърната система имаме изображение с значително по-голям брой степени на сивото.



Фиг. 9.15. Повишаване на яркостта

Един такъв пример е възстановеното изображение на среза на човешкото тяло след рентгенова томография. Типично степените на сивото са 1024. Това позволява чрез изменение на яркостта и контраста да се разглеждат детайли както от слабо пропускащите рентгеновото излъчване органи (костите), така и детайли от пропускащите добре рентгеновото излъчване органи (меките тъкани и вътрешните органи - стомах, бял дроб, сърце и др.).



Фиг. 9.16. Повишаване на контраста

9.6.3.2. Автоматично регулиране на яркостта и контраста

Много често въведеното черно-бяло изображение изменя своята яркост в по-тесен от максималния интервал. Например при 256 възможни степени на сивото, яркостта на въведеното изображение може да е от 34 до 180. Чрез процедурата на за автоматично регулиране на яркостта и контраста на това изображение се цели точката с минимална яркост да има стойност 0, а точката с максимална яркост да има стойност 256. Процедурата на изравняването протича на два етапа - намиране на минималната и максималната яркост в изображението и извършване на самата обработка:



Фиг. 9.16. Автоматично регулиране на яркостта и контраста на изображението

```
amin = AMAX;  
amax = AMIN;  
for (i=0; i<N; i++) {  
    for (j=0; j<M; j++) {  
        if (a[i][j] > amax) amax = a[i][j];  
        if (a[i][j] < amin) amin = a[i][j];  
    }  
}  
  
adc = (AMAX - AMIN) / (amax - amin);
```

```

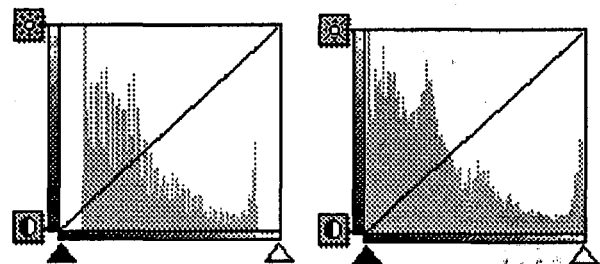
for (i=0; i<N; i++) {
  for (j=0; j<M; j++) {
    a[i][j] = adc * a[i][j] - amin;
    if (a[i][j] > AMAX) a[i][j] = AMAX;
    if (a[i][j] < AMIN) a[i][j] = AMIN;
  }
}

```

На фиг. 9.16. е показано обработеното по този начин тестово изображение.

9.6.3.3. Хистограма на яркостта на изображението

Хистограмата на яркостта на изображението се получава като се преброят точките с дадена степен на яркост. Резултатът е един масив $H[AMAX]$ с брой на елементите $AMAX$, всеки елемент $H[i]$ на който ни дава броя на точките с яркост i . На фиг. 9.17 са показани хистограмите на яркостта на тестовото изображение и на изображението от фиг. 9.16. Хистограмата е ценен помощник при регулиране на яркостта и контраста на изображението и позволява още да го подобрим чрез стесняване на интервала между $amin$ и $amax$. Нека в примера за изравняването минимална стойност 34 да има само една-единствена точка от изображението, а действителната минимална яркост да е 56. Това ще се види много добре на хистограмата и операторът ще може да укаже нова минимална и максимална стойност на яркостта за изравняване.



а) тестово

б) от фиг. 9.16

Фиг. 9.17. Хистограма на яркостта на изображения

9.6.3.3. Увеличаване

Целта на увеличаването е да се разгледа подробно участък с размери K на L точки от изображение с размери M на N , $K \leq M$ и $L \leq N$. Като правило размерът на участъка, респективно броят на точките в него, не е голям. За да се подобри изображението се извършва увеличаване на броя на точките, най-често отново до M на N . Това означава редискретизация на част от изображението с нова дискретизираща функция. Въпрос на изисквания към качеството е използваната апроксимационна функция - от нулев или първи ред, сплайн апроксимация или дори апроксимиращата формула от т. 9.3.2, което е

свързано с изключително голям брой изчисления. На фиг. 9.17 са показани изрязан участък от тестовото изображение и същият участък след редискретизация. Повишаването на качеството е очевидно.



а) изрязан участък от изображението



б) същия участък след редискретизация

Фиг. 9.17. Увеличаване на участък от изображението

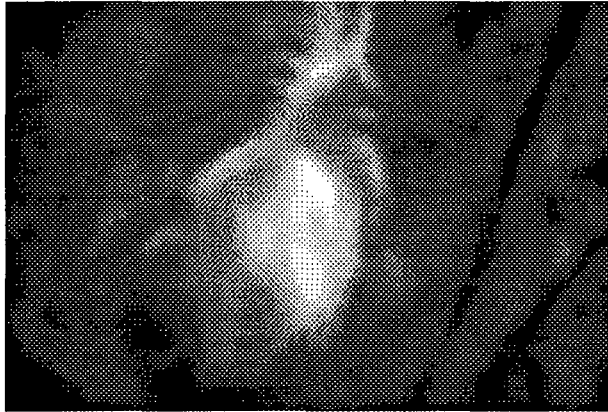


На подобен принцип са изградени цифровите системи за увеличаване на изображението, монтирани в съвременните видеокамери. Оптически, посредством вариообектив, обектът може да бъде приближаван около десет - петнадесет пъти. След това специализиран процесор за цифрова обработка на сигналите може да даде увеличение още до десет - петнадесет пъти. Това означава обработка в реално време на 50 полукадъра в секунда. Като правило се използва сплайн апроксимация като едно добро съчетание между качество на полученото увеличено изображение и скоростта за изчисляването му.

9.6.3.4. Настройка на рязкостта

При обработка на изображения за научни цели, както и за целите на изкуството, възниква необходимостта или да се замъглят, замажат, контурите на изображението, или обратно, да се изострят тези контури, да се направят по рязки. Това означава изображението да премине през един двумерен цифров филтър. Когато този филтър пропуска по-слабо високите честоти се получава замъгляване, замазване на контурите. Когато филтърът пропуска по-добре високите честоти се получава изостряне на контурите в изображението, повишава се

неговата рязкост. На фиг. 9.18.а) е показано тестовото изображение



а) намаляване



б) увеличаване

Фиг. 9.18. Настройка на рязкостта на изображението

преминало през нискочестотен двумерен цифров филтър. Филтърът е *нелин* некурсивен, с размерност на матрицата 10×10 и стойност на всеки елемент $0,01$. На практика яркостта ^{на} всяка точка от изходното изображение усреднява стойността на съответната точка от входното със стойностите на околните ѝ точки. На фиг. 9.18.б) е показано тестовото изображение преминало през високочестотен двумерен цифров филтър. Вижда се изпъкването на контурите в изображението.

Високочестотен филтър в комбинация с прагов елемент (ако точката от изображението е с по-голяма яркост от праговата и се присвоява максимална стойност, ако не - минимална) позволява да се отделият само контурите в изображението. На фиг. 9.19 е показано тестовото изображение обработено с високочестотен филтър с линейно нарастваща амплитудночестотна характеристика и прагов елемент настроен на половината яркост. След обработката изображението е инвертирано за по-добър изглед. Фил-



Фиг. 9.19. Отделяне на контурите посредством високочестотен филтър и прагов елемент

търът с линейно нарастваща амплитудночестотна характеристика е всъщност намиране на първата производна - вж. свойство 5.1.4.g) на преобразуването на Фурие.

9.6.4. Особености при цветни изображения

Всеки цвят може да се получи като комбинация от три основни цвята. При цветните изображения имаме три канала на яркостта - по един за всеки цвят. Това означава три канала при въвеждане на цветни изображения и автоматично утрояване на обема памет, необходима за съхраняване на изображението. Различават се две основни системи за разлагане на изображението по основни цветове - адитивна и субтрактивна.

9.6.4.1. Адитивна система за разлагане на цветовете

Основните цветове тук са червеното, зеленото и синьото. Известна е като RGB система (от red, green, blue - червено, зелено и синьо). По трите канала се записва яркостта на съответния цвят за всяка точка. При нулева яркост по всички канали точката е черна, при максимална яркост по всички канали - бяла. Затова и системата се нарича адитивна - сумата от опорните цветове дава бяло. Използва се най-често, тъй като е най-удобна при представяне на цветовете на екрана на телевизор или на монитора на компютър. По този начин се разлагат изображенията в цветната телевизия и във видеопаметта на компютърните системи.

9.6.4.2. Субтрактивна система за разлагане на цветовете

Основните цветове са светлосиньото, пурпурното и жълтото, затова тази система се нарича CMYK (от cyan, magenta, yellow). При нулева яркост на трите канала изображението е бяло, при максимална яркост - черно. Удобна е при подготовка на цветни изображения за печат на хартия, където и намира най-голямо приложение. Това са и цветовете на мастилата на цветните печатащи устройства. Когато не е печатано с нито един от цветовете остава белия цвят на хартията, когато са положени и трите основни цвята се получава черен цвят.

9.6.4.3. Обработка на цветни изображения

Всички изброени дотук обработки могат да бъдат извършени и над цветни изображения, като се обработват и трите цветови канала едновременно. Резултатите са същите, като при черно-белите изобра-

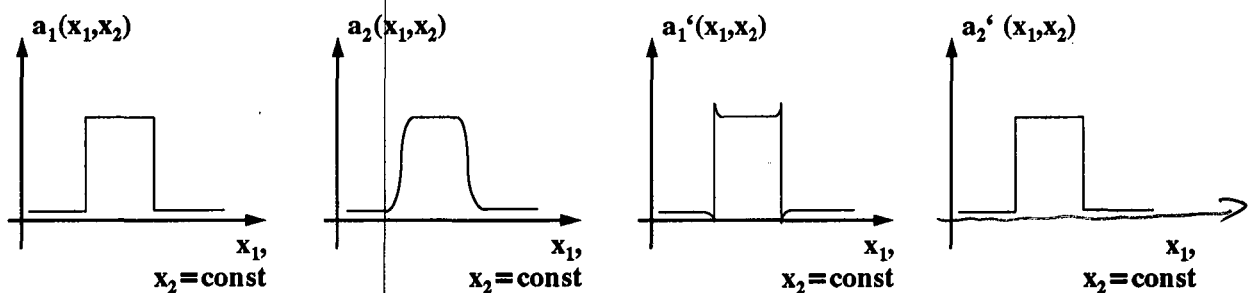
жения. Цветът на всяка точка не се променя, а се изменя само яркостта ѝ.

9.6.4.4. Цветови корекции

Когато обработваме по различен начин всеки от трите цветови канала цветът на всяка точка от изображението се променя. Това позволява да се нанесат цветови корекции както в цялото изображение, така и в отделни участъци от него. Намирам приложение за целите на науката и изкуството и при подготовка за печат на цветни снимки, диапозитиви и др.

9.6.5. Подготовка на изображението за визуализация

Когато обработеното изображение се покаже на екрана на компютъра, то се възприема от човека чрез неговия зрителен орган - окото. След това изображението се обработва и осмисля от мозъка. След направени психофизиологични изследвания на възприемането на образи от човека се е установило, че комплексът око-мозък замазва резките смени в яркостта, т.е. действа като двумерен филтър с намалено предаване на високите честоти. За подобряване на възприемането на изображенията в някои случаи се прибегва до предварително изкривяване. Това става с двумерен цифров филтър с повдигане на високите честоти в изображението. Той подчертава ръбовете така, че след възприемане от окото и обработка от мозъка да се получи усещане еквивалентно на истинското изображение. Всичко това, разбира се, е твърде условно, но подобреното възприемане на детайлите от така подготвени изображения от страна на човека е доказано. На фиг. 9.20 този процес е показан нагледно. На фиг. 9.20.а) е показан срез по една ос на примерно изображение - една бяла ивица на черен фон. На фиг.



а) оригинално б) възприемане в) предизкривено г) възприемане

Фиг. 9.20. Възприемане на изображение без и с предварителна корекция

9.20.б) е показано как най-вероятно се възприема това изображение от системата око-мозък. На фиг. 9.20.в) е показано тестовото изображение с въведена корекция - един двумерен цифров филтър с повишаване на коефициента на предаване към високите честоти, а на фиг. 9.20.г) - как предварително изкривеното изображение се възприема най-вероятно от системата око-мозък.

Обобщение

Предмет на разглеждане в тази глава са многомерните сигнали. Подобно на едномерните те се разделят на непрекъснати, дискретизирани и квантувани. Представянето в честотна област на многомерния сигнал се нарича многомерен спектър. За разлика от едномерната дискретизираща функция при многомерните дискретизиращи функции многообразието е голямо.

Дефинирана е теоремата за дискретизацията на многомерни сигнали по аналогия с теоремата на Шенон в едномерния случай. За възстановяване на стойността на сигнала между точките на дискретизация се използва многомерния аналог на апроксимационната формула на Шенон. Приложими са също така и апроксимациите от нулев и първи ред, както и сплайн апроксимацията.

Многомерното преобразуване на Фурие ни дава многомерния спектър на един многомерен сигнал. Преобразуването на Фурие на многомерен сигнал съществува при крайна енергия на сигнала както при едномерните сигнали. Освен изведените за едномерния случай свойства са дадени и две нови - транспониране и огледално отражение.

Многомерният цифров филтър е една линейна дискретна многомерна система. Параметрите му са честотни, амплитудночестотни и фазочестотни характеристики в честотна област и импулсна характеристика като реакция на въздействието на многомерен дираков импулс. Подобно на едномерните и многомерните цифрови филтри биват рекурсивни и нерекурсивни.

Изображенията са двумерен частен случай на многомерните сигнали. Всеки елемент на дискретния двумерен сигнал на изображението носи информация за яркостта на съответната точка. След въвеждането си в компютърната система всяко изображение преминава през предварителна обработка, същинска обработка и подготовка за визуализация. Предварителната обработка компенсират изкривяванията внесени от приемната апаратура: преходни, амплитудни, геометрични и честотни. Всички обработки на многомерните сигнали могат да бъдат извършени и над изображения, но има и някои специфични за изоб-

раженията действия с тях. Такива са регулиране на яркостта и контраста, изравняване и настройка на рязкостта. При цветните изображения цветът на всяка точка се разлага на три основни цвята. Това увеличава тройно обема на данните и съответно времето за обработка. При цветните изображения освен вече изброените начини за обработка се прибавят и цветовете корекции. Подготовката на изображението за визуализация включва предварителна компенсация на изкривяванията на системата око-мозък на оператора и подобрява възприемането на изображението от човека.



Задачи

1. Пред екрана на осцилоскоп е поставен диапозитив 6 на 4.5 см. С помощта на два ЦАП компютърът движи лъча на осцилоскопа и възприема яркостта на преминалата през диапозитива светлина посредством оптически приемник разположен на 10 см. от центъра на изображението. Изходното напрежение от фотоприемника след усилване се измерва посредством АЦП. По този начин изображението се сканира и въвежда в компютърната система. ^{на}Имерете теоретичната формула и напишете програма извършваща корекция на амплитудните изкривявания!

2. За случая от предишната задача направете разчет от колко нива на сивото ~~нагоре~~ във въведеното изображение има смисъл да се прави тази корекция.

3. Начертайте амплитудно честотната характеристика на филтъра от фиг. 9.18.а). За целта разгледайте фиг. 5.3 и фиг. 9.5.



Контролни въпроси

1. Дайте дефиниция на многомерен непрекъснат и многомерен дискретен сигнал!

2. Кое налага използването на различни многомерни дискретизиращи функции?

3. Кои нови свойства има многомерното преобразуване на Фурие?

4. Какво означава многомерен нерекурсивен филтър с нулева фаза?

5. Кои са основните коригиращи процедури при предварителната обработка на въведените в компютърната система изображения?

6. Кои са основните обработки на изображения и къде намират приложение?

10. Оптимизационни методи в инженерната практика



Ключови

думи

- оптимизация
- еднопараметрична оптимизация
- многопараметрична оптимизация
- наказателна функция

Какви знания и умения ще придобиете?

Къде се прилагат?

В тази тема ще се запознаете с оптимизационните методи и тяхното приложение в системите за събиране и обработка на информация. Те се използват както при проектирането на самите системи, така и при решаването на конкретните проблеми в процеса на работата на една такава система.

Получените знания ще приложите при проектирането на системите за събиране и преобразуване на информация и при разработката на конкретните алгоритми за работата им.

Въведение

Често срещани проблеми в инженерната практика са удовлетворяването на противоречиви изисквания, проектирането на системи в които трябва да се получат максимално възможните параметри, минимизиране на грешката на измерителна система и много други задачи, които се свеждат до намирането на минимум или максимум на някаква сложна зависимост. Тъй като аналитичният вид на тази зависимост в повечето случаи не е известен или е изключително сложен, известните от математиката методи за намиране на екстремум чрез нулите на първата производна са неприложими. Този клас проблеми се решават чрез методите на математическата оптимизация.

10.1. Общи сведения за оптимизацията

За успешното решаване на даден проблем с използване на методите на математическата оптимизация е необходимо:

- ◆ ясно и точна да се дефинират *параметрите за оптимизация*;
- ◆ да се намери подходящ *критерий за качество*;
- ◆ да се дефинират *ограниченията* на оптимизационния процес;
- ◆ да се избере подходящ *метод (алгоритъм) за оптимизация*;
- ◆ да се намери добра *стартова точка* (начални стойности на параметрите за оптимизация).

Оптимизацията обикновено се извършва от компютърна програма с **три основни части**:

- ⇒ главна програма за задаване на начални стойности на параметрите и печат на резултатите;
- ⇒ подпрограма за изчисляване на критерия за оптимизация;
- ⇒ подпрограма за реализиране на избрания метод за оптимизация, която в процеса на работата си извиква подпрограмата за изчисляване на критерия за оптимизация.

Първите две се пишат и настройват за всеки конкретен проблем, докато третата се пише и настройва еднократно и може да се използва и при решаването на други оптимизационни задачи.

10.1.1. Етапи на оптимизацията

За да се използват методите на математическата оптимизация трябва да се премине през следните етапи:

- ◆ избор на параметрите за оптимизация;
- ◆ избор на критерий за качество;
- ◆ определяне на ограниченията в които ще се търси екстремума;
- ◆ избор на подходящ метод за оптимизация;
- ◆ определяне на началната точка за оптимизация;
- ◆ написване на програмата и настройка с тестов пример;
- ◆ получаване на решението с реални данни.

10.1.2. Избор на критерий за качество

Критерият за качество е израз, даващ количествена оценка на качеството, чийто екстремум (минимум или максимум) се търси. Математически това е една хиперповърхнина в $n+1$ мерното пространство, където n е броят на параметрите за оптимизация. Като правило критерият се определя от няколко изисквания (частни критерии) и се формира по различни начини.

Критерий за максимална (минимална) сума:

$$Q = \sum_{i=1}^n a_i \cdot q_i$$

където q_i са частните критерии, а a_i - коефициента на тежест на съответния частен критерий. Недостатъкът на този начин на формиране на критерия е в ~~трудността~~ и субективността при определяне на ~~коефициентите на тежест~~. При едни и същи условия с различни коефициенти могат да се получават различни решения, оптимални за едни или други изисквания. Много близките до нула частни критерии имат и много малка тежест.

Критерий за максимално (минимално) произведение:

$$Q = \prod_{i=1}^n q_i$$

където q_i са частните критерии. При този начин на формиране на критерия близките до нула частни критерии имат значително по-голяма тежест.

Минимаксен критерий:

$$Q = \min\{\max[a_i \cdot q_i]\}$$

където q_i са частните критерии, а a_i - коефициента на тежест на съответния частен критерий. Недостатъкът на този начин на формиране на критерия е в ~~трудността~~ при определяне на коефициентите на тежест. При едни и същи условия с различни коефициенти могат да се получават различни решения, оптимални за едни или други изисквания. В този случай се търси минимума на критерия Q , като при всяко изчисляване той приема стойността на

най-лошата, максималната, стойност на частния критерий умножена по съответния коефициент на тежест. Този начин на формиране на критерия е удобен, например, при оптимизиране на точността на измерителен уред, при което се гарантира точност по-добра от получената при оптимизацията стойност на критерия.

10.1.3. Видове ограничения

Ограниченията са онези физически граници на стойностите, в които да се потърси екстремума на критерия за оптимизация. Стесняват задачата и я превръщат от математическа задача за намиране на ~~екстремум в реален инженерен проблем~~. Обикновено се задават като равенства или неравенства, на които трябва да отговарят параметрите за оптимизация, частните или общия критерии или междинни резултати:

$$C_i \neq (x_1, x_2, \dots, x_n) = 0$$

.....

$$a_i \leq Z_i(x_1, x_2, \dots, x_n) \leq b_i$$

Ограниченията при оптимизация биват:

- ◆ Ограничения на параметрите за оптимизация. Обикновено се задават границите в които може да се изменя стойността на всеки от параметрите за оптимизация, но е възможно и да се задават като неравенства - i -тият параметър трябва да има стойност по-голяма (по-малка) от зададената.
- ◆ Ограничения на изходните параметри. Задават се като интервал от стойности и/или като гранични стойности на частните или на общия критерий за оптимизация.
- ◆ Ограничения по междинни резултати. Задават се като равенства или неравенства на междинни резултати при изчисляване на частните или общия критерии за оптимизация.

10.1.4. Локални и глобален екстремум

В зададеното пространство на параметрите критерият за качество може да има няколко екстремума (минимуми или максимуми) - локални екстремуми. Най-добрият от тях се нарича глобален екстремум. Всички алгоритми за оптимизация предполагат, че хиперповърх-

нината в $n+1$ мерното пространство е унимодална, т.е. има единствен екстремум в пространството на параметрите, зададено чрез ограниченията. При подозрение за няколко екстремума работното пространство на параметрите се разбива на части и във всяка част се прави отделно оптимизация и намиране на екстремума. Вземат се най-добрите от тях - глобалният екстремум.

10.1.5. Постановка на задачата за оптимизация

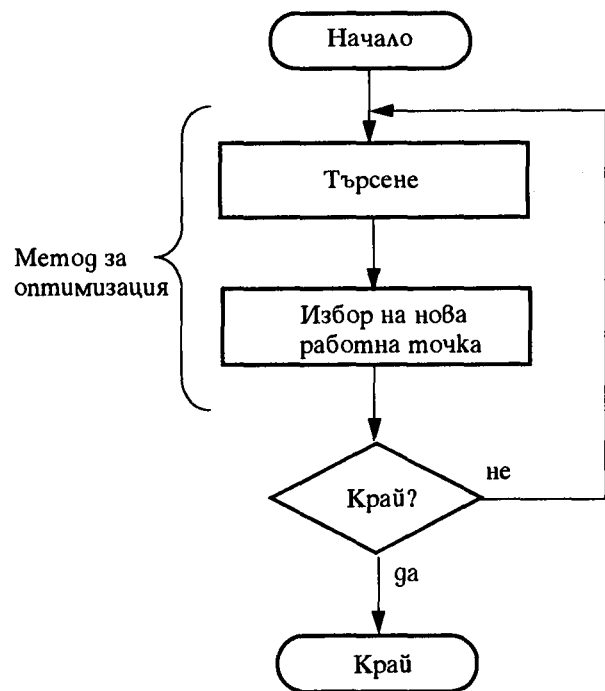
Преди решаване на задачата за оптимизация трябва да бъде уточнено следното:

- ⇒ вектор на параметрите за оптимизация $\vec{X} = (x_1, x_2, \dots, x_n)$;
- ⇒ критерий за качество $Q(\vec{X})$;
- ⇒ ограничения в процеса на оптимизация: $A(\vec{X}) = 0$, $B(\vec{X}) < 0$, $C(\vec{X}) > 0$;
- ⇒ начална точка за започване на оптимизацията \vec{X}_0 .

10.1.6. Протичане на процеса на оптимизация

Оптимизационните методи се движат към целта стъпка по стъпка, итеративно, като преминават през различни работни точки - вектора на променливите за оптимизация \vec{X} . Преди първата итерация на работната точка се присвояват стойностите на началната точка за започване на оптимизацията. Всяка итерация протича на три етапа, както е показано на фиг. 10.1:

- ◇ Първият етап се нарича търсене и представлява проучване на околността на работната точка. По начина на реализиране на търсенето се различават и различните методи за оптимизация.
- ◇ Вторият етап е избор на нова, по-добра работна точка - резултат от търсенето.
- ◇ Третият етап на всяка итерация е проверката за край на оптимизационния процес чрез сравняване на старата и новата работна точка. Ако условията за край не са изпълнени се извършва следваща итерация с нейните три етапа.



Фиг. 10.1. Оптимизация.

Приема се, че изчисляването на критерия за оптимизация изисква най-много време. Затова при сравняване на различните методи за оптимизация се предпочита този, който достига до екстремума с минимален брой изчисления на критерия за оптимизация. Съществен момент при решаване на конкретна оптимизационна задача е изборът на началната точка. Колкото тя е по-близка до екстремума, толкова по-бързо той ще бъде достигнат.

10.1.7. Условия за край на оптимизационния процес

Кога да приключи процесът на оптимизация зависи и от спецификата на оптимизационната задача. Обикновено се използват няколко условия за край на оптимизационния процес.

Придвижване по всяка от координатите по-малко от зададено малко число

След всяка итерация се изчисляват разликите

$$\Delta_i = |x_i^{(k)} - x_i^{(k-1)}|$$

където $x_i^{(k)}$ е стойността на параметъра за оптимизация x_i в k -тата итерация. Когато по всички координати преместването на работната точка Δ_i е по-малко от зададено малко число ε_i , се приема, че е достигнат екстремума. Неудобството тук е, че трябва да се задават ~~голям брой малки числа ε_i~~ . Не е изключено оптимизационният процес да приключи преждевременно.

Преместване на работната точка по-малко от зададено малко число

След всяка итерация се изчислява разстоянието на което се е придвижила работната точка:

$$\Delta = \sqrt{\sum_{i=1}^n (x_i^{(k)} - x_i^{(k-1)})^2}$$

и когато придвижването на работната точка е по-малко от зададено малко число ε е се приема, че е достигнат екстремума. Удобството тук е, че се задава само едно малко число, но е необходимо числените стойности на ~~параметрите за оптимизация да са от един порядък!~~ Това условие за край е неприложимо, например, когато оптимизираме стойностите на резистори (десетки и стотици килооми - 10^3 ома) и кондензатори (десетки и стотици пикофаради - 10^{-12} фарада). За целта е необходимо числените стойности на резисторите и кондензаторите във вектора на параметрите да са в килооми и пикофаради, при което порядъците им се изравняват. Това трябва да се отчете и в подпрограмата за изчисляване на критерия за оптимизация.

Извършване на определен брой итерации

Това допълнително условие се поставя за ~~да се избегне зациклянето на програмата за оптимизация~~ при неправилно зададено основно условие за край - например твърде малко число ε . Програмата приключва принудително след извършването на определен брой стъпки.



Трябва да се има пред вид, че изчисляването на критерия за оптимизация става с определена точност. Малкото число е трябва да се значително по-голямо от точността на изчисляване на критерия за оптимизация, понеже самата стойност на критерия има флукуации, определени от точността на изчисляване.

10.1.8. Въвеждане на ограниченията посредством наказателни функции

Ограниченията са в зависимост от решавания проблем. За да може да се използва написаната подпрограма за оптимизация и при решаването на други оптимизационни задачи трябва проверката за нарушаване на ограниченията да се изведе от тази подпрограма и да се прехвърли в проблемно зависимата част - подпрограмата за изчисляване на критерия за оптимизация. За целта се използват така наречените наказателни функции. При нарушаване на някое от ограниченията критерият за оптимизация се влошава изкуствено, с което се "наказва" оптимизационната процедура за навлизането в неразрешена зона от стойности и тя връща работната точка обратно в разрешената зона. Критерият за оптимизация придобива следния вид:

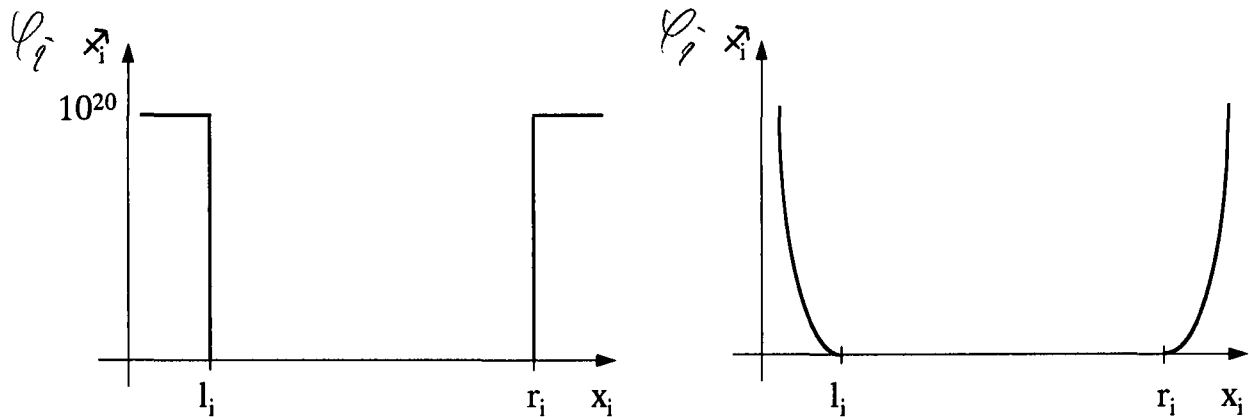
$$Q(\bar{X}) = F(\bar{X}) + \sum_{i=1}^l \varphi_i(\bar{X})$$

където:

- * $F(\bar{X})$ е началният критерий за оптимизация, при който трябва да се спазват зададените ограничения;
- * $\varphi_i(\bar{X})$ са наказателните функции за нарушаването на всяко от ограниченията;
- * $Q(\bar{X})$ е модифицираният критерий за оптимизация, който от гледна точка на оптимизационната процедура няма ограничения, тъй като изкуствено е влошен в забранените зони.

Най-често използвани наказателни функции

А. При нарушаване на ограниченията към критерия за оптимизация се прибавя голяма константа, например 10^{20} . Неудобството тук е, че в точката на нарушаване на ограниченията



критерият за оптимизация има точка на прекъсване, което смущава работата на някои алгоритми за оптимизация, използващи производната на критерия за оптимизация.

Б. При нарушаване на ограничението наказателната функция има вида:

$$\varphi_i = (l_i - x_i)^2,$$

където l_i е граничната стойност. По този начин критерият за качество се влошава плавно при излизане от разрешената зона и няма точки на прекъсване.

10.2. Методи за еднопараметрична оптимизация

Методите и алгоритмите за еднопараметрична оптимизация се отличават със своята простота. Постановката на задачата е намирането на екстремума на критерия за качество при изменението на параметъра за оптимизация в зададен интервал $[a, b]$. Всички алгоритми работят при наличието на един екстремум в този интервал.

Интервал на неопределеност е отсечката в която със сигурност се намира екстремума. В началния момент това е зададеният интервал на стойностите на параметъра за оптимизация. Алгоритмите за еднопараметрична оптимизация извършват

стесняване на интервала на неопределеност до предварително зададено малко число ε .

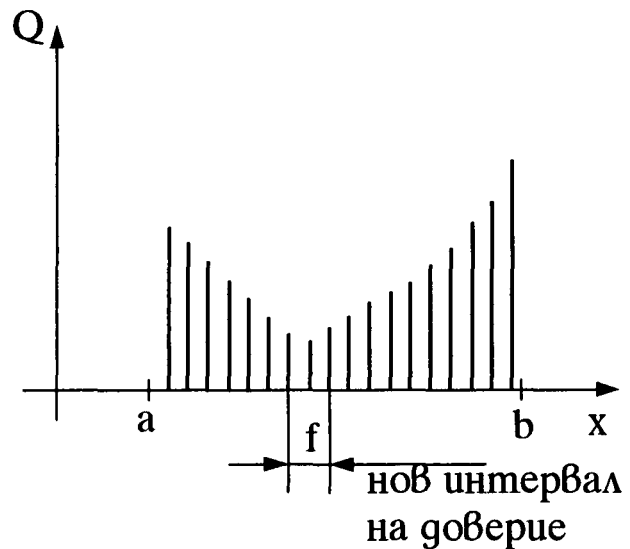
Особеност на методите за еднопараметрична оптимизация е тяхната *детерминираност*, т.е. за всеки метод може предварително да се каже след колко изчисления на критерия за оптимизация може да се стесни интервала на доверие да зададената стойност.

10.2.1. Метод на сканирането

В зададеният интервал на стойностите $[a, b]$ се изчислява стойността на критерия за оптимизация в N равноотдалечени точки и се взема стойността параметъра за оптимизация при която имаме най-добра (минимална или максимална) стойност на критерия, както е показано на фиг. 10.2. Стесняването на интервала на доверие се дава от израза:

$$f = \frac{2}{N+1}.$$

Това е най-простият и най-неефективният метод за еднопараметрична оптимизация. За да стесним интервала на доверие хиляда пъти ($f=0.001$) е необходимо да изчислим 1999 пъти критерия за оптимизация.



Фиг. 10.2. Метод на сканирането.

10.2.2. Модифициран метод на сканирането

Идеята тук е да приложим няколко пъти метода на сканирането. След първата итерация в новия интервал на доверие отново се прилага метода на сканирането и т.н. до стесняване на интервала на доверие до зададената стойност, както е показано на фиг. 10.3. Стесняването на интервала на доверие се дава от израза:

$$f = \left(\frac{2}{N+1}\right)^I,$$

където N броят на точките за изчисляване на критерия при всяка итерация, а I - броят на итерациите. Общият брой оценки на критерия за оптимизация е:

$$J = N \cdot I.$$

Възниква въпросът за стойността на N при която интервалът на доверие достига зададено стесняване f с минимален брой оценки. След полагане $I = \frac{J}{N}$ и логаритмуване се достига до равенството:

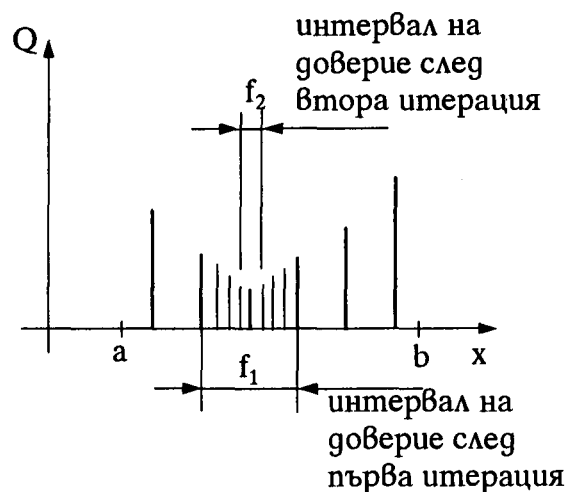
$$J = \frac{\ln\left(\frac{1}{f}\right) \cdot N}{\ln\left(\frac{N+1}{2}\right)}.$$

Тъй като $\ln\left(\frac{1}{f}\right) = const$ (зададеното стесняване на интервала на доверие), то разглеждаме функцията $\frac{N}{\ln\left(\frac{N+1}{2}\right)}$ на целочисления параметър N ($N > 1$). Първите седем стойности на функцията са дадени в таблицата.

N	2	3	4	5	6	7	8
$\frac{N}{\ln\left(\frac{N+1}{2}\right)}$	4.93	4.33	4.36	4.55	4.79	5.05	5.98

Тя има минимум при $N = 3$, което означава, че зададено стесняване на

интервала на доверие f ще се получи с най-малък брой изчисления на критерия за оптимизация при деление на интервала на доверие на три. За да стесним интервала на доверие хиляда пъти ($f = 0.001$) по този метод за еднопараметрична оптимизация е необходимо да изчислим 30 пъти критерия за оптимизация. Броят на итерациите е 10, а стесняването на интервала на доверие е 1024 пъти.

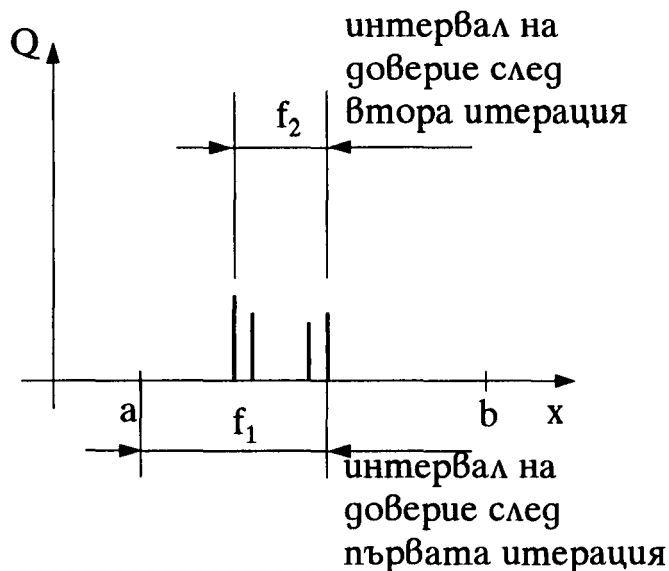


Фиг. 10.3. Модифициран метод на сканирането.

10.2.3. Метод на дихотомията

Името на метода идва от гръцките думи "ди" (две) и "томос" (режа) и преведено буквално означава "разрязване на две". С

модифицираният метод на сканирането се изчерпаха възможностите за повишаване ефективността на методите за еднопараметрична оптимизация при запазване на равноотдалеченото разполагане на стойностите на параметъра за оптимизация при извършване на оценките. Нека при извършване на две оценки в точките x_1 и x_2 , лежащи в отсеч-



Фиг. 10.4. Метод на дихотомията.

ката $[a, b]$, $z_1 = x_1 - a$, $z_2 = x_2 - x_1$ и $z_3 = b - x_2$.

Тогава ако $Q(x_1) > Q(x_2)$ новият интервал на доверие ще бъде $z_1 + z_2$, а ако $Q(x_1) < Q(x_2)$ новият интервал на доверие ще бъде $z_2 + z_3$. Вижда се, че за да е минимален новият интервал на доверие и при двата случая е необходимо $z_1 = z_3$ и $z_2 = \varepsilon \rightarrow 0$. Това означава, че при всяка итерация се извършват по две оценки в средата на интервала на доверие, отдалечени на малко число ε , както е показано на фиг. 10.4. Новият интервал на доверие ще е левият, ако $Q(x_1) > Q(x_2)$ и десният ако $Q(x_1) < Q(x_2)$. След всяка итерация новият интервал на доверие е:

$$f = \frac{1}{2} + \frac{\varepsilon}{2},$$

което означава приблизително два пъти стесняване. За да стесним интервала на доверие хиляда пъти ($f=0.001$) по този метод за еднопараметрична оптимизация е необходимо да изчислим 20 пъти критерия за оптимизация. Броят на итерациите е 10, а стесняването на интервала на доверие е малко по-малко от 1024 пъти.

10.2.4. Метод на златното сечение

Ако отсечката z е разделена на две части z_1 и z_2 , казваме че е разделена в златно сечение ако:

$$\frac{z_1}{z} = \frac{z_2}{z_1} = 0.618.$$

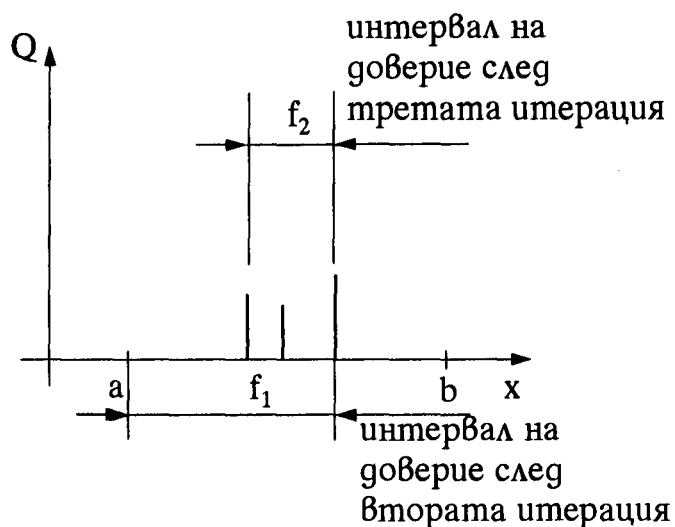
С метода на дихотомията се изчерпаха възможностите за повишаване ефективността на методите за еднопараметрична оптимизация без използване на оценките от предишните итерации. Методът на златното сечение включва оценка в точката, разделяща интервала на доверие в златно сечение. С тази първа итерация не се постига стесняването му.

Отново се извършва оценка на критерия за оптимизация в

точката, разделяща по-голямата отсечка в златно сечение. На базата на двете оценки избираме отсечката с по-голяма стойност на критерия и интервала на доверие се стеснява 0.618 пъти. При следващата итерация имаме вече една оценка, тъй като се оказва, че точката, деляща новият интервал на доверие в златно сечение е едната от двете предишни - фиг. 10.5. По този начин с всяка следваща итерация се стеснява интервала на доверие 0.618 пъти. Стесняването на интервала на доверие след I итерации се дава от израза:

$$f = 0.618^{I-1}.$$

За да стесним интервала на доверие хиляда пъти ($f=0.001$) по този метод за еднопараметрична оптимизация е необходимо да изчислим 16 пъти критерия за оптимизация. Стесняването на интервала на доверие е 1365 пъти.



Фиг. 10.5. Метод на златното сечение

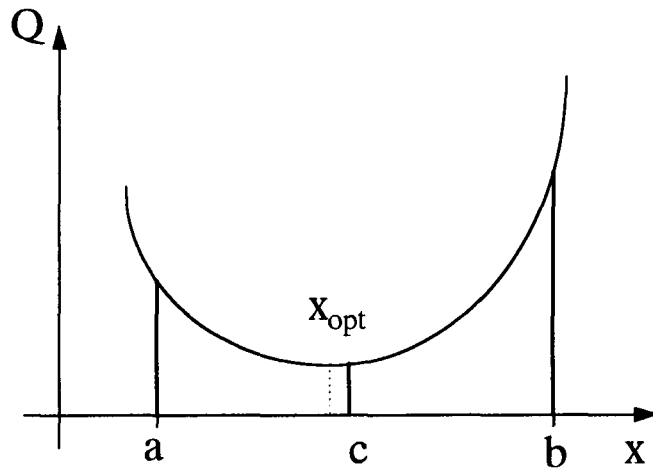
10.2.5. Метод на Brent

Това е приблизителен метод за намиране на екстремума на функция на едни параметър. Изчислява се стойността на критерия за оптимизация Q в двата края на интервала на доверие a и b и в средата c . Нека получените стойности са Q_a , Q_b и Q_c .

През получените три точки се прекарва крива от втора степен (парабола) $Q = u \cdot x^2 + v \cdot x + w$, както е показано на фиг. 10.6. Коэффициентите на параболата u , v и w се изчисляват от линейната система от три уравнения с три неизвестни:

$$\begin{cases} Q_a = a^2 \cdot u + a \cdot v + w \\ Q_c = c^2 \cdot u + c \cdot v + w \\ Q_b = b^2 \cdot u + b \cdot v + w \end{cases}$$

Екстремумът на параболата се намира по чисто математически път - нулата на първата производна $x = -\frac{v}{2u}$. Този екстремум се приема за приблизително решение на еднораметричната оптимизация.



Фиг. 10.6. Метод на

Брент

10.3. Методи за многопараметрична оптимизация

Методите за многопараметричната оптимизация се различават съществено от методите за еднораметрична оптимизация.

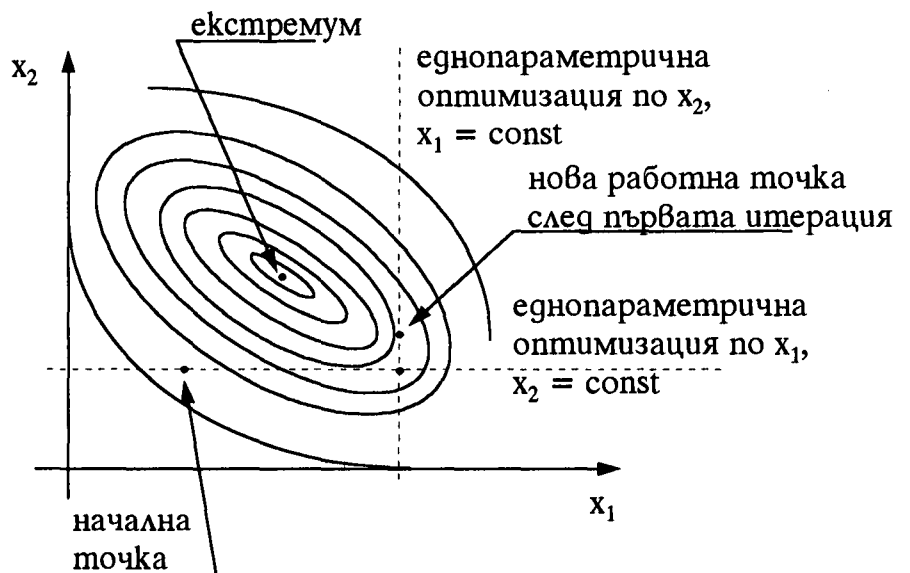
- ⇒ Характерно тук е, че силно намалява вероятността за унимодална хиперповърхнина в работната зона. Това налага редица проверки за да гарантираме намирането на глобалния екстремум. Като минимум програмата се запуска няколко пъти от различни начални точки.
- ⇒ Друга характерна особеност е, че с увеличаване на броя на параметрите усилията за намиране на екстремума на критерия за оптимизация нарастват експоненциално.

⇒ При многопараметричната оптимизация рядко може да се прогнозира броят на изчисленията на критерия за оптимизация за достигане на екстремума със задана точност.

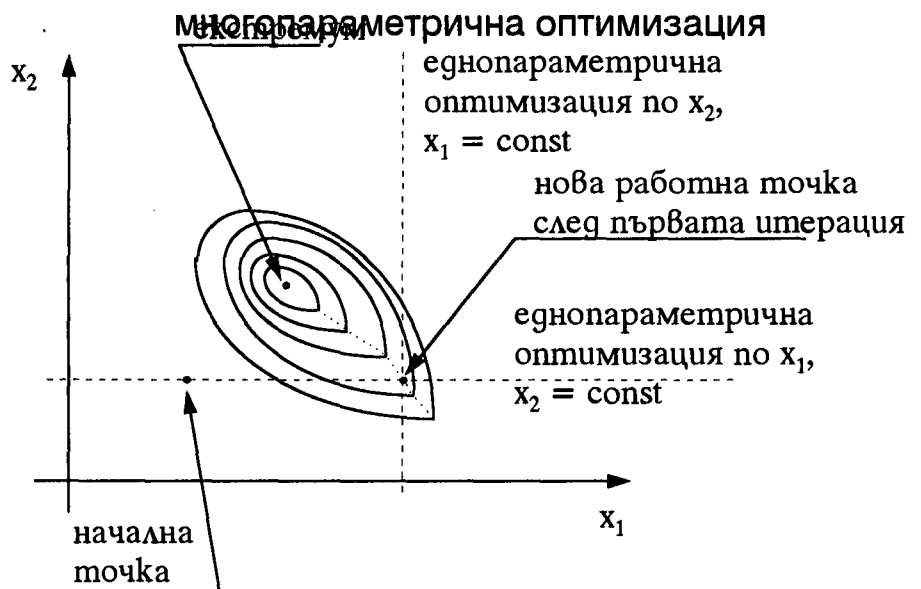
10.3.1. Метод на Гаус-Зайдел

Методът на Гаус-Зайдел използва ефективните методи за еднопаметрична оптимизация. Всяка итерация представлява последователни еднопаметрични оптимизации по всеки един от параметрите, като другите се приемат за константи - фиг. 10.7. Използва се някой от вече изброените критерии за край.

Методът не дава правилно решение при наличие на гребен в многомерната хиперповърхнина на критерия за оптимизация, както е показано на фиг. 10.8. Намерената след първата итерация нова работна точка лежи на гребена и ще остане същата и след втората итерация. Критерият за край ще я обяви за екстремум, което не отговаря на истината. Удобен е като стартова процедура за намиране на по-добра начална точка за



Фиг. 10.7. Метод на Гаус-Зайдел за



Фиг. 10.8. Метод на Гаус-Зайдел при наличие на

другите методи за многопараметрична оптимизация.

10.3.2. Мрежово търсене на Мишке

Този метод е многомерен еквивалент на модифицирания метод на сканирането. Първоначалната област на търсене е квадрат, куб или хиперкуб. Зоната на търсене се разделя на равни части, прекарва се мрежа и се прави оценка във всеки от възлите. Най-добрата точка става център на нов квадрат, куб или хиперкуб и се прави следващата итерация. Един от малките методи за многопараметрична оптимизация, който позволява предварително изчисляване на броя на итерациите за постигане на зададена точност за локализиране на екстремума. Тъй като този брой е отчайващо голям, това е методът използван като последно средство за намиране на екстремум.

10.3.3. Градиентни методи

Това са цяла група методи, използващи изчисляването на градиента на критерия за оптимизация. Както е известно от математиката градиентът е вектор, указващ посоката на най-голямо нарастване на функцията с дължина пропорционална на стръмността на нарастване. Еквивалентен е на първата производна на функция на един параметър и се дава от формулата:

$$\text{grad} \bar{Q} = \frac{dQ}{dx_1} \cdot \bar{e}_1 + \frac{dQ}{dx_2} \cdot \bar{e}_2 + \dots + \frac{dQ}{dx_n} \cdot \bar{e}_n,$$

където $\frac{dQ}{dx_i}$ е частната производна на критерия за оптимизация спрямо параметъра x_i , а \bar{e}_i е единичен вектор с посока по оста на x_i . Посоката на градиента указва най-бързото нарастване на функцията, а дължината на вектора е пропорционална на стръмността на нарастването.

В точката $\bar{X} = (x_1, x_2, \dots, x_n)$ градиентът числено се изчислява по формулата:

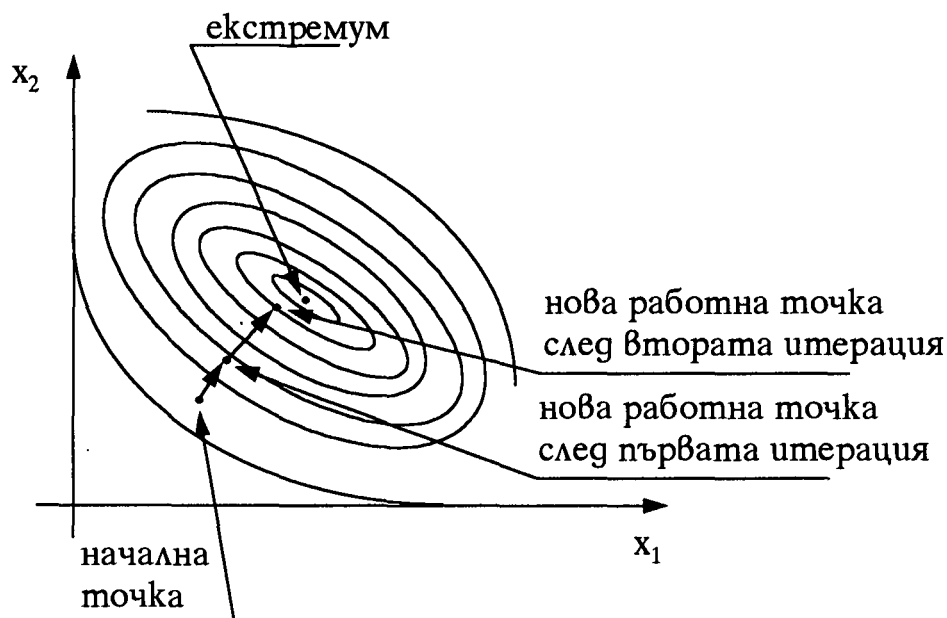
$$\text{grad} \bar{Q} = \frac{\bar{Q}_1(x_1 + \Delta_1) - Q(\bar{X})}{\Delta_1} \cdot \bar{e}_1 + \frac{\bar{Q}_2(x_2 + \Delta_2) - Q(\bar{X})}{\Delta_2} \cdot \bar{e}_2 + \dots + \frac{\bar{Q}_n(x_n + \Delta_n) - Q(\bar{X})}{\Delta_n} \cdot \bar{e}_n$$

Функцията $\bar{Q}_i(\bar{x})$ изчислява стойността на критерия за оптимизация, като заменя i -тият параметър за оптимизация с \bar{x} и взема стойностите на останалите от текущия \bar{X} .

Тук константите Δ_n трябва да бъдат достатъчно малки, за да изчислим частната производна в точката, и достатъчно големи, за да избегнем флуктуациите при изчисляване на критерия за качество, определени от точността на изчисляването му. За изчисляване на градиента на критерия за качество са необходими $n+1$ оценки.

10.3.3.1. Чисто градиентен метод

Всяка итерация представлява изчисляване на градиента и извършване на стъпка по посока на градиента - фиг. 10.9. Ако новата точка е по-добра, стъпката се увеличава при следващата итерация, а ако е по-лоша - стъпката се намалява. Процедурата продължава до намаляване на стъпката под зададена минимална стойност. Методът е универсален, но не е особено ефективен - изисква много пъти изчисляване на критерия за оптимизация.



Фиг. 10.9. Чисто градиентен метод за многопараметрична оптимизация

10.3.3.2. Метод на най-бързото спускане

Съчетава предимствата на градиентните методи с ефективността на методите за еднопараметрична оптимизация. Всяка итерация представлява изчисляване на градиента и извършване на еднопараметрична оптимизация по посока на градиента - фиг. 10.10. Намерената най-добра точка става работна точка за следващата итерация. Използва се някой от вече изброените критерии за край.

Еднопараметричната оптимизация протича на два етапа:

- ◆ *Първият етап е локализиране на отсечката за еднопараметрична оптимизация.* Той се извършва като последователно се изчислява критерия за оптимизация с нарастваща стъпка докато се намира по-добра точка. Ако на n -тата оценка намерим по-лоша точка, отсечката в която ще се извърши еднопараметричната оптимизация е между $(n-2)$ -та и n -тата оценки, както е показано на фиг. 10.11.



Фиг. 10.10. Метод на най-бързото спускане

- ◆ *Вторият етап е еднопараметрична*



10 18 Фиг. 10.11. Еднопараметрична

оптимизация в отсечката локализираща екстремума по посока на градиента. Приложим е кой да е от методите за едноточков параметрична оптимизация. Най-често тук се използват методите на дихотомията и на златното сечение. Тук е възможно използването и на приблизителният метод на Брент, като параболата се прекарва през вече извършените оценки на първия етап.

Методът на най-бързото спускане е един от най-често използваните методи за оптимизация поради неговата универсалност и ефективност.

10.3.4. Конфигурационен метод на Хук и Джийвс

Този метод се основава на принципа, че всяка успешна система от премествания (конфигурации) си струва да се повтори отново. Една итерация протича на следните етапи:

- ◆ около работната точка със стъпка Δ се извършва търсене - оценки на критерия за оптимизация;
- ◆ ако не се намери по-добра точка, стъпката се намалява и се преминава към следващата итерация;
- ◆ ако се намери по-добра точка се извършва преместване на работната точка в тази посока с увеличена стъпка:
$$x_{io}^{(k+1)} = x_i^{(k+1)} + a \cdot (x_i^{(k+1)} - x_i^{(k)}),$$
 където i е номерът на параметъра, k - итерацията, а a е константа по-голяма от единица.

При този метод стъпката може да достигне стойности стотици пъти по-големи от началната. Процедурата се прекратява като се



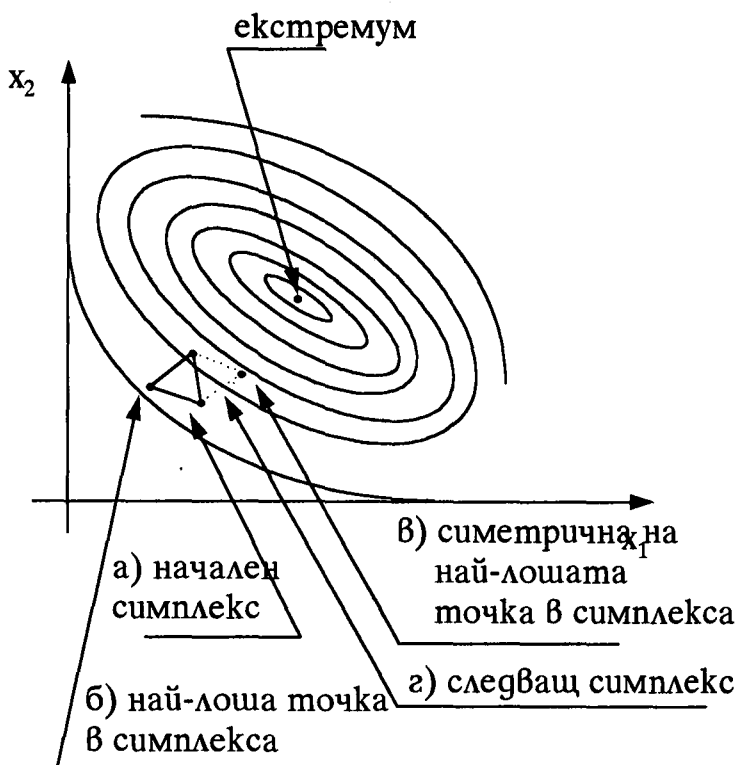
Фиг. 10.12. Конфигурационен метод на Хук и Джийвс

използва някой от вече изброените критерии за край.

Методът на Хук и Джийвс е универсален и независимо от по-малката си ефективност намира широко приложение поради простотата си. Подобно на метода на Гаус-Зайдел не работи добре при наличието на гребен в хиперповърхнината на критерия за оптимизация.

10.3.5. Симплекс метод

Симплексът е затворена n -мерна фигура с праволинейни страни и $n+1$ върха. В едномерното пространство това е отсечката, в двумерното - триъгълникът, в примерното - тетраедърът и т.н. Вместо начална работна точка при този метод се задава начален симплекс. Прави се оценка на критерия за оптимизация във всеки от върховете. Намира се симетричната спрямо останалите на най-лошата



Фиг. 10.13. Симплекс-метод за многопараметрична оптимизация

точка и това е новият работен симплекс - фиг. 10.13. При следващата итерация се прави оценка на критерия за оптимизация само в новата точка и процедурата се повтаря. По този начин центърът на тежестта на симплекса се придвижва към екстремума. Процедурата се повтаря докато новата точка е по-добра от старата.

Задаването на голям по размери начален симплекс ускорява процедурата по намиране на екстремума, но намалява точността на локализирането му. Намаляването на размерите на началния симплекс подобрява точността, но силно забавя процедурата на намиране на екстремума. Затова съществува

модификация на метода, известна като алгоритъм на Нелдер и Мид, при която се започва с голям по размери симплекс. След като не може да се намери по-добра точка размерите на симплекса се намаляват към най-добрата точка и процедурата се повтаря. Това продължава до намаляването на размерите на симплекса под зададено малко число.

Обобщение

В настоящата глава бяха разгледани общите принципи на математическата оптимизация. Всяка оптимизационна програма е един цикъл, състоящ се от търсене, извършване на преход към нова работна точка и проверка за край. Обърнато беше внимание на използването на наказателни функции в критерия за оптимизация.

Последователно бяха разгледани методите за еднопараметрична оптимизация: на сканирането, модифициран метод на сканирането, метод на дихотомията, метод на златното сечение и приблизителният метод на Брент.

Обърнато беше сериозно внимание на особеностите на многопараметричната оптимизация: силно намалява вероятността за унимодална хиперповърхнина в работната зона, което налага редица проверки за да гарантираме намирането на глобалния екстремум; с увеличаване на броя на параметрите усилията за намиране на екстремума на критерия за оптимизация нарастват експоненциално; рядко може да се прогнозира броят на изчисленията на критерия за оптимизация за достигане на екстремума със зададена точност.

Разгледани бяха методите за многомерна оптимизация:

- метод на Гаус-Зайдел, подходящ за намиране на стартова точка за другите методи;
- мрежовото търсене на Мишке, като последно средство за намиране на екстремум;
- групата на градиентните методи, в която влиза особено популярния метод на най-бързото спускане;
- конфигурационния метод на Хук и Джиуиус, най-простия за програмиране;
- симплекс-методът с модификацията му като алгоритъм на Нелдер и Мид.

Разгледаните методи за еднопараметрична и многопараметрична оптимизация са само част от най-често използваните методи за оптимизация. При написването на сложни оптимизационни процедури могат да бъдат използвани редица по-ефективни, но и същевременно по-сложни методи за оптимизация, описани в литературата.

Обхватът на приложение на методите на математическата оптимизация в реалната инженерна практика е изключително широк. Малко знания по програмиране, малко досетливост - и можем да постигнем съществено подобряване на работата на електронните устройства, без да се налага никакви допълнителни изменения в апаратната част.



Обърнете внимание, че оптимизационните методи са приложими само при напълно ясна постановка на задачата за оптимизация:

⇒ ясно и точно дефинирани кои да са параметрите за оптимизация;

⇒ вече създаден математически апарат за изчисляване на критерия за оптимизация.

Ясно е, че чрез оптимизация не може да се подобри работата на електронната апаратура ако е пропуснат важен параметър за оптимизация. Няма да се получи и добър резултат ако е избран неподходящ критерий за оптимизация.



Задачи

1. Един интересен метод за намиране на корен на полином от общ вид е чрез минимизиране на критерий за качество, който приема стойности, клонящи към нула, когато се достига коренът. Опитайте този метод, като използвате някой от методите за еднопараметрична оптимизация, за да намерите корените на уравнението:

$$x^5 + 21x^4 + 158x^3 + 502x^2 + 609x + 245 = 0.$$

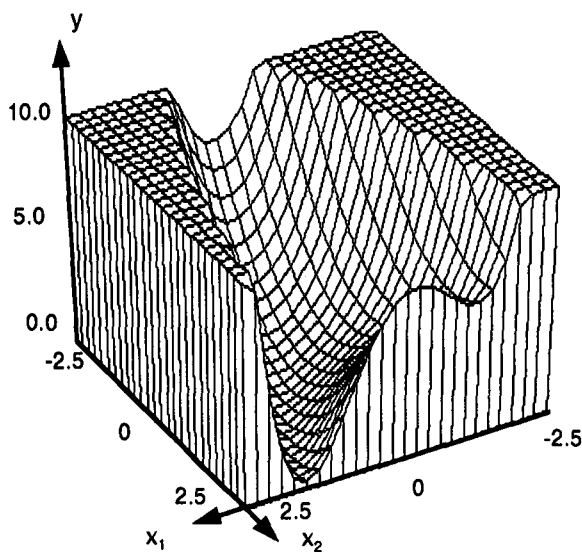
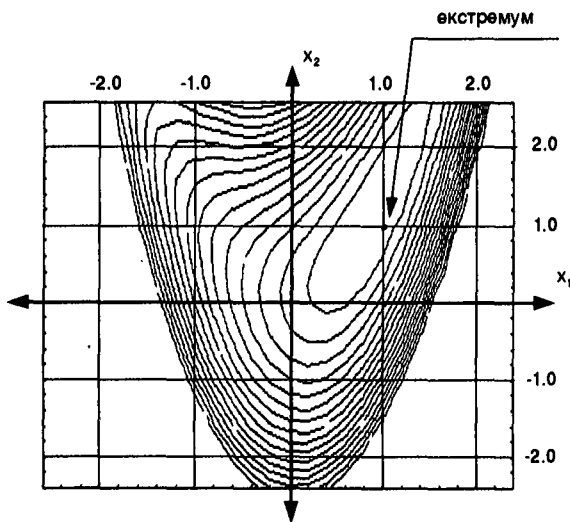
2. Въпреки че е доста проста в математическо отношение, функцията на Розенброк:

$$y = k \cdot (x_2 - x_1^2)^2 + (1 - x_1)^2,$$

има извит обрaг, както се вижда и от следващата фигура. Функцията е изчислена за стойност на параметъра $k = 1$. В топографското изображение линиите са начертани за стойности на y през 0.5 - от 0 до 8. В обемното изображение стойността на функцията е ограничена

до 10 за да се види по-ясно обрaзът. С увеличаването на стойността на параметъра k обрaзът става по-стръмен. Екстремумът е в точката $(1.0,1.0)$, но ако се избере стартова точка във втори квадрант, често се получава лоша сходимост. Да се реши оптимизационната задача със стартова точка $(-1.0,1.0)$ и стойност на параметъра $k = 100$ като се използват:

- а) методът на Гаус-Зайдел;
- б) някой градиентен метод;
- в) някой конфигурационен метод.





Контролни въпроси

1. Кои са етапите на оптимизационния процес?
2. Как може да се формира от частните критерии общият критерий за оптимизация?
3. Какви могат да бъдат критериите за край на оптимизационния процес?
4. Защо използването на наказателна функция превръща оптимизационната задача от такава с ограничения в такава без ограничения?
5. Направете оценка на еднопараметричните методи за оптимизация от гледна точка на тяхната простота и от гледна точка на тяхната ефективност!
6. Кога е неприложим методът на Гаус-Зайдел за многомерна оптимизация?
7. Защо много често е неприемливо използването на мрежовото търсене на Мишке като метод за многомерна оптимизация?
8. Кои са двете фази на едномерната оптимизация по посока на градиента при методът на най-бързото спускане?

11. Управление и комуникации в системите за събиране и обработка на информация



Ключови думи

- операционна система за реално време
- задание
- събитие
- опашка от данни
- флаг
- локална мрежа
- слоеве на локалните мрежи
- протокол на локална мрежа

Какви знания и умения ще придобиете?
Къде се прилагат?

В тази глава се разглеждат въпросите за управлението и комуникациите в системите за събиране и преобразуване на информация — системното програмно осигуряване. Ще научите основните понятия в операционните системи за реално време и ще получите представа за общата структура на програмното осигуряване на една система за събиране и обработка на информация.

Получените знания ще приложите при създаването на проекта за системното програмно осигуряване на система за събиране и преобразуване на информация.

Въведение

В началните глави на тази книга глави бяха дадени апаратните особености на системите за събиране и обработка на информация. В следващите бяха обсъдени основните методи въз основа на които се създават алгоритмите за обработка на данните и информацията в тези системи. Навсякъде се предполагаше, че реализацията на тези алгоритми е програма за компютърна система. Това е така наречения приложен слой на програмното осигуряване на системите за събиране и обработка на информация. Всяка такава система трябва да изпълнява множество задачи, които реализират алгоритмите за събиране и обработка. Тези задачи са свързани, решават общи проблеми, и трябва да могат да обменят данни помежду си. Как да се организира управлението на ресурсите на компютърната система и на най-важният от тях — времето на процесора (процесорите), как да се осигу-

ри комуникацията и предаването на данни между отделните програми е предмет на системното програмно осигуряване. На тези въпроси е посветена настоящата глава. Разглеждането включва само начално запознаване с основните понятия и принципи на управлението и комуникацията в работещите в реално време системи.

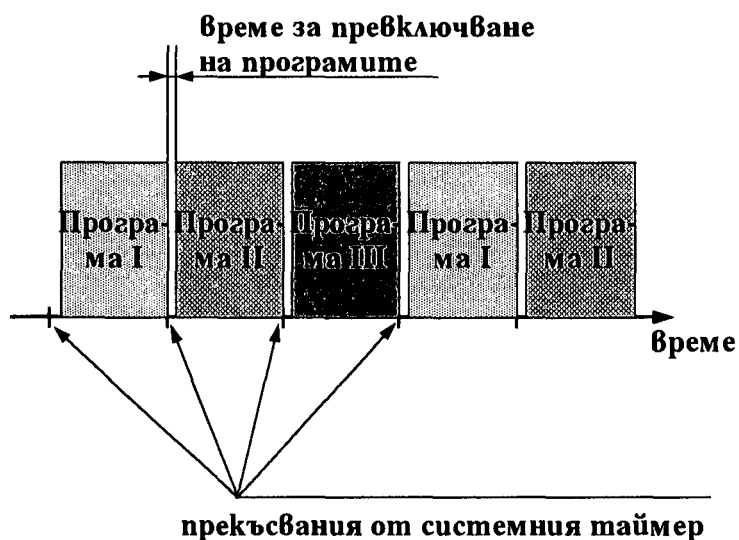
11.1. Операционни системи за реално време

11.1.1. Дефиниране

Операционна система за реално време е програма позволяващи едновременното изпълнение на много програми, всяка от които може да приема информация от външната среда, да извежда навън информация и да обменя данни с другите програми.

Операционната система за реално време е по своята същност многозадачна операционна система. Терминът “реално време” означава, че системата като цяло ще реагира на дадено външно въздействие по-рано от зададен краен интервал от време.

Една от най-простите реализации на операционна система за реално време е програма която обслужва периодичното прекъсване от системния таймер на компютърната система. Показана е на фиг. 11.1. След всяко прекъсване от таймера се запомня докъде е стигнал процесорът в изпълнението на прекъснатата програма и след това времето му се предоставя на следващата. Това продължава до изтичане на интервала време на таймера, след което процесът се повтаря. Ако



Фиг. 11.1. Реализиране едновременното изпълнение на три програми

имаме три програми, които се редуват по този начин, и превключваме от програма в програма достатъчно често, за външен наблюдател ще изглежда че трите програми се изпълняват едновременно, всяка една от три пъти по-бавен процесор.

11.1.2. Основни понятия в операционните системи за реално време

11.1.2.1. Задание

Заданието е самостоятелна програма, работеща под управлението на операционната система за реално време. Заданията са от едно до няколко десетки.

11.1.2.2. Събитие

Събитието е основна комуникационна единица в операционната система за реално време. Заданията могат да изчакват настъпването на дадено събитие и да генерират настъпването на събитие.

11.1.2.3. Ресурс

Това е памет, входно/изходен канал, машинното време на процесора и др. Ресурсите са под управлението на операционната система за реално време. Всеки ресурс може да бъде свободен или зает. Заданието може да поиска от операционната система за реално време ресурс, да изчаква освобождаването му, да го използва и да го освобождава.

11.1.2.4. Пощенска кутия

Пощенската кутия е структура от данни в която едно съобщение се изпраща от едно задание и се получава от друго. Заданията могат да поставят съобщение в пощенската кутия, да вземат съобщение от пощенската кутия и да проверяват състоянието ѝ — празна или не.

11.1.2.5. Семафор

Семафорът е флаг, който се използва за синхронизиране на заданията във времето. Има две състояния — вдигнато и спуснато. Заданията могат да установяват семафора във всяко от двете състояния и да проверяват текущото му състояние.

11.1.2.6. Системна единица за време

Системната единица за време е минималното време в рамките на операционната система за реално време и е равна на интервала между прекъсванията ^{от} на таймера.

11.1.3. Атрибути на заданията

Всяко задание има своите характеристики — атрибути. Атрибутите се поддържат и проверяват от операционната система за реално време. В различни реализации на операционни системи атрибутите могат да варират, но най-често срещаните са приоритет и състояние.

11.1.3.1. Приоритет на заданието

Приоритетът на заданието е число определящо важността на заданието по отношение на другите задания. Обикновено колкото е по-голямо числото толкова е по-важно заданието. На заданията с по-голям приоритет операционната система дава по-голяма част от машинното време. В различните операционни системи за реално време този механизъм може да бъде реализиран по различен начин.

11.1.3.2. Състояние на заданието

Състоянието на заданието е атрибут задаван и променян непрекъснато от операционната система за реално време. В различните операционни системи състоянията варират като имена, но като цяло се различават четири основни състояния на заданието.

Блокирано. Заданието в блокирано състояние съществува в паметта на компютърната система, но не участва в конкуренцията за получаване на машинно време на процесора. Всички обръщания към него се игнорират.

Чакащо. Заданието в това състояние изчаква или настъпването на събитие или освобождаването на ресурс.

Готовност. Заданието в състояние “готовност” има всички условия за изпълнението си и участва в конкуренцията за получаване на част от машинното време на процесора.

Активно. Заданието е в състояние “активно” когато се изпълнява от процесора. Броят на активните задания в рамките на операционната система за реално време е равен на броя на процесорите, т.е. ^{при един процесор} ~~най-често~~ само едно задание може да бъде в това състояние.

11.1.4. Граф на състоянията на заданията

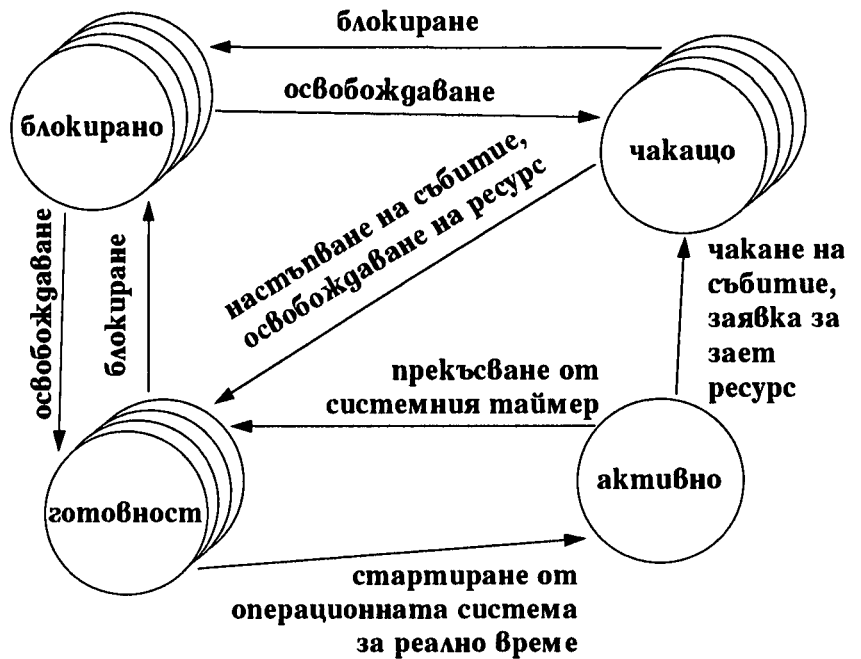
Графът на състоянията на заданията определя възможните преходи на заданието от едно състояние в друго. Преминаването на едно задание от едно състояние в друго се определя от операционната система за реално време. Схематично графът за преминаване на заданията от едно състояние в друго е показан на фиг. 11.2.

11.1.4.1. Преход “активно” — “готовност”

Преходът на едно задание от активно състояние в готовност се извършва при прекъсване от системния таймер или ако заданието се откаже от повече машинно време чрез обръщане към операционната система за реално време. И в двата случая управлението се предава на операционната система за реално време. Тя променя състоянието на заданието от активно в готовност и запомня съдържанието на всички необходими регистри за да може това задание да бъде стартирано отново.

11.1.4.2. Преход “готовност” — “активно”

След като някое от заданията е излязло от активно състояние и преминало в готовност операционната система за реално време избира едно от всичките задания, намиращи се в състояние готовност. Възстановява се съдържанието на регистрите на процесора каквото е било при предишното прекъсване на работата на това задание и то се стартира отново от точката където е било прекъснато. Различните операционни системи за реално време се различават по критериите, които използват за да изберат кое от заданията в състояние готовност да стартират. Един от основните критерии е приоритетът на заданието. При всяко положение, обаче, трябва да се гарантира, че всяко задание ще получи част от машинното време на процесора след краен интервал от време независимо от приоритета си и приоритета на останалите задания. Тъй като операционната система е също програма времето за търсене на най-подходящо задание е загубено от гледна точка на заданията. Затова се избягва да се правят прекалено сложни алгоритми за избор на заданието за стартиране. Срещат се и операционни системи за реално време при които всички задания са с равен приоритет и се стартират последователно.



Фиг. 11.2. Граф за преминаване на заданията в различните състояния

11.1.4.3. Преход “активно” —“чакащо”

Активното задание преминава в чакащо състояние когато се обърне към операционната система за реално време и заяви, че иска да изчака настъпването на определено събитие. Същият преход се извършва и ако активното задание поиска да ползва ресурс, който в момента е зает. Операционната система променя състоянието на заданието от активно в чакащо и запомня съдържанието на всички необходими регистри за да може това задание да бъде стартирано отново. Запомня се също така и кое събитие или кой ресурс чака заданието. След това операционната система за реално време избира едно от всичките задания, намиращи се в състояние готовност и го стартира по описания вече начин.

11.1.4.4. Преход “чакащо” —“готовност”

Когато активното задание се обърне към операционната система за реално време и генерира настъпването на дадено събитие от своя страна операционната система за реално време преглежда списъка на чакащите задания. **Всички** задания, които чакат настъпването на точно това събитие се привеждат в състояние готовност.

По същия начин се процедурира и ако активното задание се обърне към операционната система за реално време и освободи даден ресурс. Проверява се дали има задание, което чака освобождаването на този ресурс. Ако не се намери такова задание ресурсът се освобождава. Ако има задание (задания), чакащи този ресурс, се избира **едно** от тях и му се присвоява състояние готовност с което то вече участва в конкуренцията за машинно време на процесора и може да ползва ресурса. Алгоритъмът за избор на кое задание да се даде освободения ресурс трябва да се съобразява с приоритета на чакащите задания, но трябва и да гарантира получаването на ресурса след краен брой освобождавания независимо от приоритета на чакащото задание и приоритета на останалите задания.

И при двата случая управлението се връща на активното задание.

11.1.4.5. Преход към “блокирано” и обратно

По заявка на активното задание операционната система за реално време може да приведе някое от заданията в състояние блокирано, независимо от състоянието на това задание в момента — готовност или чакащо. Също така по заявка на активното в момента задание операционната система за реално време може да извади блокираното задание и да го върне в състоянието в което е било преди блокирането — готовност или чакащо. Въпрос на реализация на самата операционна система за реално време е да се проверяват и някои допълнителни условия, например само задание с по-висок приоритет да може да блокира друго задание, само заданието, което е блокирало друго задание да може да го освобождава и т.н. След изпълнение на заявката за блокиране управлението се връща на активното задание.



Преминаване на заданията от едно състояние в друго

Нека имаме система за управление на индустриален обект. Веднага след стартиране на заданията под управление на операционната система за реално време една част от тях се обръщат към нея и заявяват, че ще изчакват събитие номер 14, например. В състояние готовност остават само заданията, които извършват непосредственото управление на индустриалния обект. Освен тях има още едно кратко, но с висок приоритет задание, което извършва проверки на най-важните параметри от работата на управлявания обект. При нормални стойности на тези параметри предоставя остатъка от процесорното си време на останалите задания. В случай на отклонение на важен параметър от нормалните граници ситуацията се променя. Това е еквивалентно на възникване на аварийна ситуация. Заданието наблюдател генерира въпросното събитие номер 14 и блокира част или всичките задания, управляващи индустриалния обект в нормален режим. Чакащите събитие задания преминават в състояние готовност, участват в конкуренцията за процесорно време и поемат управлението на промишления обект. Те са написани за работа при възникване на аварийна ситуация и вземат необходимите мерки за овладяването ѝ. След стабилизиране на обстановката заданията за уп-

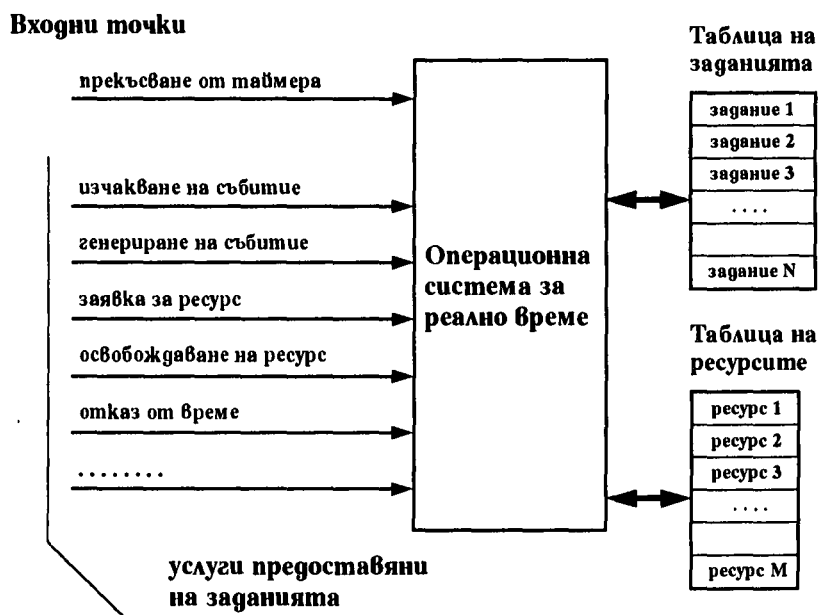
равление на обекта в аварийна ситуация се обръщат към операционната система за реално време със заявка за изчакване на въпросното събитие номер 14. Заданието наблюдател извежда от състояние блокирано заданията за управление на индустриалния обект в нормални условия и системата за управление придобива отново нормалното си състояние.

11.1.5. Функции и реализация

Една обобщена блокова схема на операционна система за реално време е дадена на фиг. 11.3.

11.1.5.1. Входни точки

Операционната система за реално време е програма, която управлява разпределението на ресурсите и обслужва заявките от страна на заданията. За тази цел тя трябва да има определени входни точки. Основна входна точка е обслужването на прекъсването от системния таймер. След всяко прекъсване управлението се предава на тази точка и се извършват дейностите, описани в т. 11.1.4.1. Останалите входни точки са заявки за услуги, които предоставя операционната система за реално време — искане на ресурс, генериране на събитие и т.н. Достъпът до входните точки за заявки



Фиг. 11.3. Обобщена блокова схема на операционна система за реално време

може да бъде програмно организиран различно за различните процесори, но най-често става чрез механизма на програмните прекъсвания.

11.1.5.2. Таблици

За своята работа операционната система използва няколко основни таблици в които съхранява необходимата информация. Една от най-важните таблици е таблицата на заданията. В нея за всяко задание е отделено място в което се съхраняват атрибутите му (приоритет и състояние), стойностите на регистрите на процесора при прекъсване на работата на това задание и др. Друга важна таблица е таблицата на ресурсите. За всеки ресурс е отделено място, в което са записани идентификатора на ресурса, състоянието му (свободен или зает) и друга информация необходима за управлението му. Някои операционни системи за реално време поддържат и таблица на събитията, както и други масиви с данни необходими за работата им.

11.1.5.3. Реализация

Обикновено операционните системи за реално време се реализират на асемблер за съответния процесор. Все по-често напоследък се срещат и операционни системи за реално време, реализирани на езика С. Той съчетава предимствата на езиките от високо ниво (бързо написание и настройка, кратка и прегледна програма) с предимствата на програмите, написани на асемблер (бързина в изпълнението и малък обем). Операционните системи за реално време не са големи програми като брой на операторите поради изискването да са бързи и поради ограничения брой услуги, които обикновено предоставят на заданията.

11.2. Структури от данни за комуникация между заданията

Заданията, работещи под управлението на операционната система за реално време, имат нужда да обменят данни помежду си тъй като често са част от един конвейер за обработка на потока данни. За целта системното програмно осигуряване трябва да осигури средствата за обмен на информация между заданията. В някои реализации това са услуги, предоставяни от самата операционна система за ре-

ално време. В групи решения това са отделно разработени средства — програми или макроси. Най-често използваните структури от данни за обмен на информация са флагът и опашката.

11.2.1. Флаг

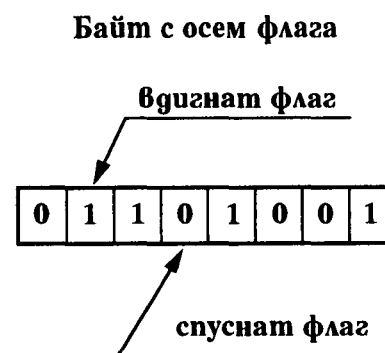
Флагът има две състояния — вдигнат и спуснат. Използва се като семафор. Програмно се реализира като един бит от даден байт в оперативната памет. Битът също има две състояния — логическа нула и логическа единица, на които се присвоява значение на спуснат и вдигнат флаг. Друга възможна програмна реализация на флага е той да заема цял байт. Ако стойността на байта е нула флагът е спуснат, ако е различна от нула — вдигнат. С флага се извършват следните основни действия (примитиви):

- ◆ вдигане на флага;
- ◆ спускане на флага;
- ◆ проверка на флага.

Извършването на тези действия се реализира програмно. Обикновено даден флаг се вдига и спуска от едно и също задание, а останалите само го проверяват. Много често флагът е средство за комуникация и синхронизиране само между две задания.

11.2.2. Опашка

Опашката служи за обмен на съобщения (масиви от данни) между заданията. Чрез нея се реализират пощенските кутии за предаване на съобщения между заданията. Едно съобщение е неделим елемент от данни. Опашката обикновено реализира връзка между две задания само в една посока — едното задание изпраща съобщения, другото ги чете и обработва. За реализиране на двупосочна връзка са необходими две опашки. Опашката е важно средство за осигуряване на максимално



Фиг. 11.4. Структура от данни флаг за комуникация между заданията

използване на времето на процесора от двете задания и служи като буфер между заданията при случаите на различен темп на постъпване и обработка на съобщенията.

11.2.2.1. Реализиране на опашката

Опашката е структура от данни съдържаща масив памет и два показалеца — входен и изходен. По своята същност това е един стек FIFO (first in first out — първи влязъл, първи излязъл). Когато опашката е празна входният и изходният показалец имат еднакви стойности. Основните примитиви за работа с опашката са:

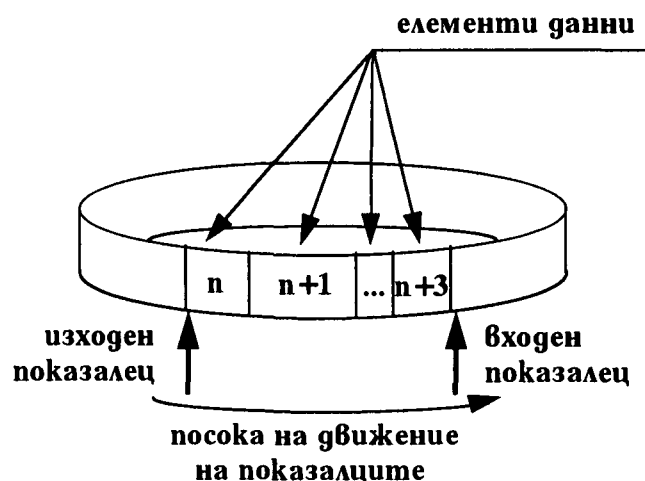
- ◆ въвеждане на елемент данни в опашката;
- ◆ извеждане на елемент данни от опашката.

Освен основните примитиви съществуват и допълнителни:

- ◇ проверка за празна или непразна опашка;
- ◇ копиране на първия елемент данни от опашката, като той остава в нея;
- ◇ изхвърляне на първия елемент от данни.

Движението на двата показалеца се извършва само напред. При достигане до края на масива памет показалецът се връща в началото му. По този начин отделената за опашката памет се превръща в един затворен кръгов буфер, както е показано на фиг. 11.5.

Въвеждането на елемент данни в опашката се извършва като елементът се копира от мястото в паметта, където е записан, в опашката. Използва се входният показалец. След въвеждането на еле-



Фиг. 11.5. Структура от данни опашка за комуникация между заданията

мента показалецът се премества напред и показва отново началото на свободното пространство. Критична ситуация тук е когато в опашката има по-малко свободно място от дължината на съобщението. В този случай заданието, което изпраща съобщението, може да постъпи по два начина. Първият е да се откаже от процесорно време, с което го предоставя на другите задания, включително и на заданието, което изтегля и обработва съобщенията от тази опашка. С това ускорява приближаването на момента когато ще бъде изтеглено съобщение от тази опашка и освободено място за новото съобщение. Вторият начин е, ако това е допустимо, да се изхвърли най-старото (първото!) съобщение и да се въведе новото.

Извеждането на елемент от опашката се извършва като се използва изходния показалец. Първият елемент в опашката се копира в посочено място в паметта на компютърната система. По време на ^{извеждането} извеждането изходния показалец се мести напред, освобождавайки място в масива памет.

Проверката за празна опашка се извършва като се сравнят стойностите на двата показалеца. Ако те са равни — опашката е празна.

Копирането на първия елемент от данни е същото като извеждането на елемент от данни, но не се извършва промяна на стойността на изходния показалец. С това елемента от данни остава в опашката.

Изтриването на елемент от данни е преместването на изходния показалец напред на разстояние равно на дължината на елемента. С това той изчезва от опашката.

Последните два примитива се използват когато елементът от данни (съобщението) се предава към друга компютърна система по някакъв съобщителен канал. Съобщението се копира и се предава по съобщителния канал, но остава в опашката. Изтрива се чак когато отсрещният компютър изпрати потвърждение за успешното му приемане. По този начин се избягва загубата на информация при възникване на грешки в съобщителната линия.

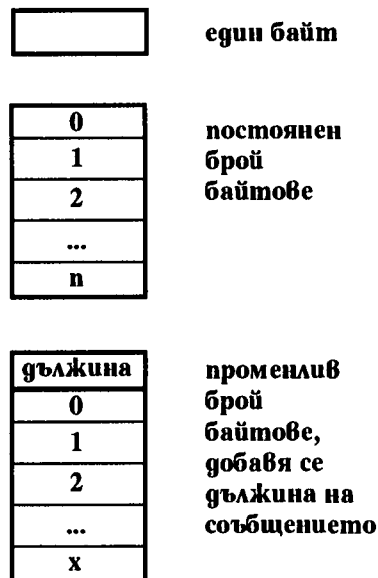
Примитивите за работа с опашката се реализират програмно и се осигурява интерфейс за работа с тях от приложния слой на програмното осигуряване на системата за събиране и обработка на информация.

11.2.2.2. Елемент данни

Елементът данни (съобщението) в опашката зависи от нуждите. Както е показано на фиг. 11.6 той може да бъде:

- един байт;
- фиксирано число байтове;
- променливо число байтове.

В третия случай пред съобщението се прибавят един или два служебни байта, носещи дължината на съобщението. Опашките обикновено се реализират за работа с третия тип елементи от данни. Прибягването до работа с опашки с фиксирана дължина на елемента се прави само при необходимост от повишено бързодействие или за икономия на памет.



Фиг. 11.6. Елемент данни

11.3. Локални компютърни мрежи

11.3.1. Общи сведения за локалните мрежи

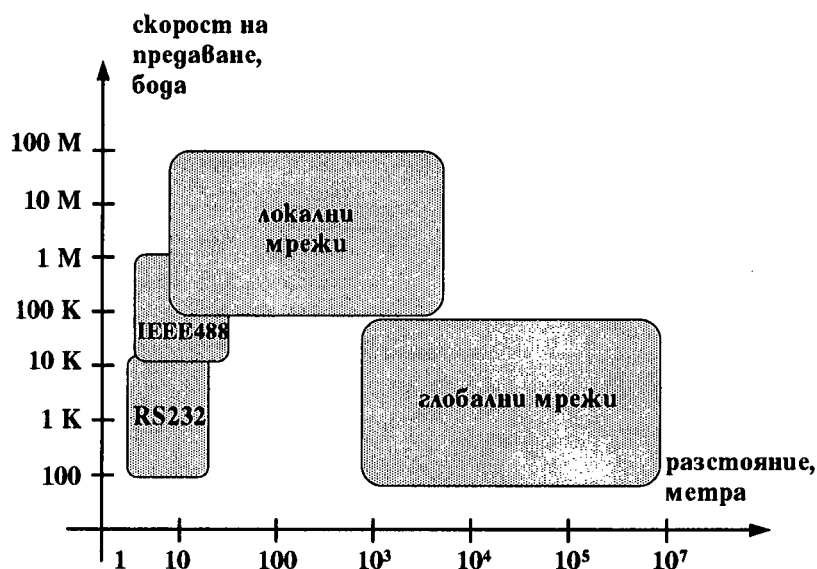
11.3.1.1. Дефиниране

Компютърната мрежа е комплекс от апаратни и програмни средства за обмен на данни между компютърни системи. В компютърната мрежа могат да се включат различен брой компютри. В зависимост от разстоянието на което се намират компютрите различаваме няколко вида компютърни мрежи:

- ⇒ глобални — разстояния от 10^3 до над 10^7 метра, скорост на предаване на данните от 100 бита/с до 100 Кбита/с;
- ⇒ локални (разстояния от 1 до 10^4 метра, скорост на предаване на данните от 100 Кбита/с до 100 Мбита/с).

Локалните компютърни мрежи свързват компютрите в рамките на една сграда или в няколко съседни сгради. Глобалните мрежи обхващат големи области — град, страна, целия свят. Те няма да бъдат обект на разглеждане в настоящата глава.

На фиг. 11.7 са показани местата на локалните и глобалните мрежи в зависимост от разстоянието и скоростта на предаване. За нагледност са дадени и местата на два от най-разпространените компютърни интерфейси — RS232 и IEEE488.



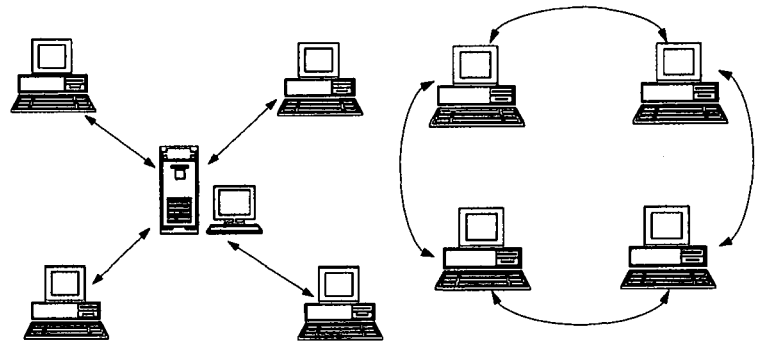
11.3.1.2. Топологии

Компютрите в рамките на локалната мрежа могат да бъдат свързани по различен начин — топология на локалната мрежа. Съществуват няколко основни типа топологии, показани на фиг. 11.8:

- звезда, при която всички компютри са свързани чрез отделен кабел към общ комуникационен възел или компютър;
- кръгова, при която всеки компютър е свързан с двата съседни;
- шинна, при която всички компютри са свързани към обща комуникационна среда, най-често коаксиален кабел;
- комбинирана, която представлява смес от две или повече от гореизброените.

11.3.1.3. Еталонен модел и слоеве

Един от основните принципи на мрежовите архитектури е разделянето на функциите на управлението на връзките на йерархично подредени слоеве. Всеки слой ползва услугите на по-долните слоеве. Еталонният модел на компютърната мрежа, препоръчан от ISO, включва седем слоя.

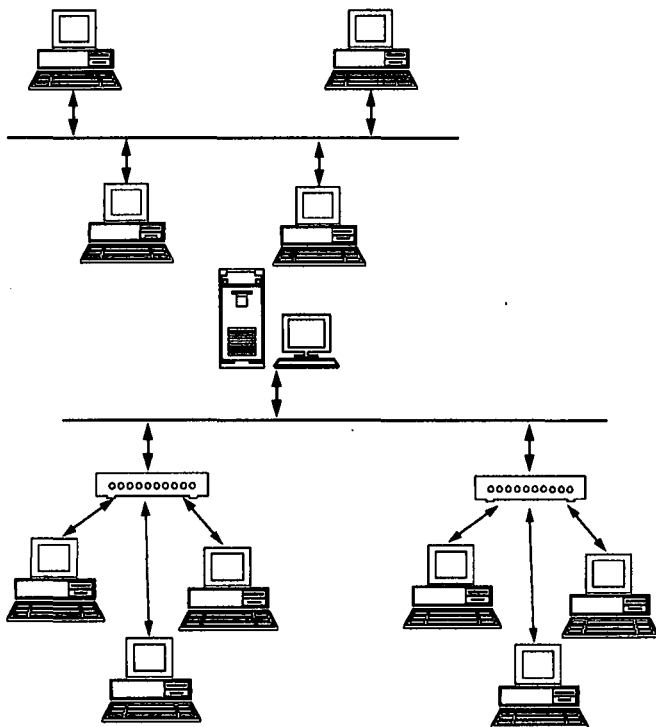


а) звезда

б) кръгова

Фиг. 11.8. Топологии на локалните мрежи

1. **Физическия слой (Physical Layer)** осигурява механични и функционални възможности за предаване на един бит от данни. Това е физическата среда за предаване на данните: кабелни линии, радиоканали, спътникови комуникационни линии и др., както и куплунги, връзки и др. Включва и начините за предаване на единица информация — електрически нива, начин на кодиране



в) шинна

г) комбинирана

Фиг. 11.8. (продължение) Топологии на локалните мрежи

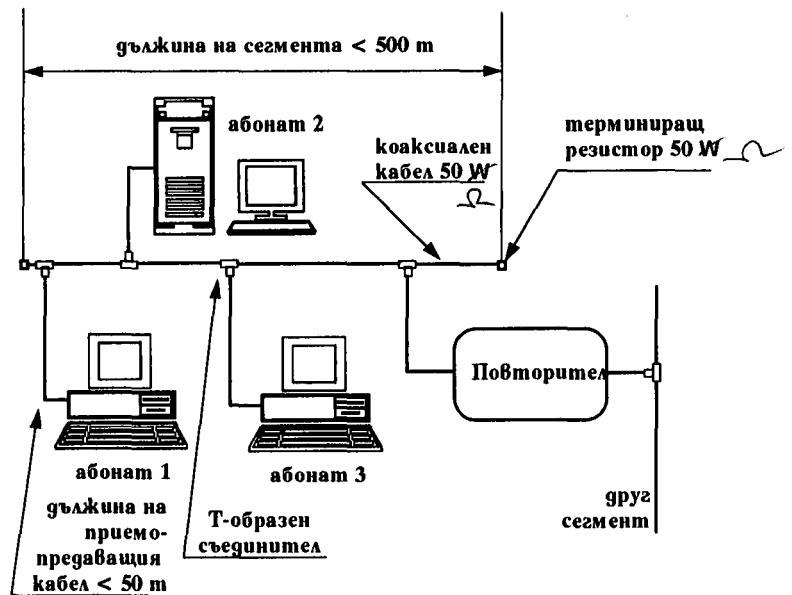
или модулация.

2. **Каналният слой (Data Link Layer)** обслужва предаването на един блок данни между два възела (две системи). Прибавят се различни методи за откриване и защита от грешки при предаването на данните във физическия слой.
3. **Мрежовият слой (Network Layer)** предоставя средства за установяване и подържане на мрежовите съединения и е само посредник между две компютърни мрежи.
4. **Транспортният слой (Transport Layer)** осигурява пренасянето на блок данни от компютъра-предавател до компютъра-приемник.
5. **Сесийният слой (Session Layer)** обслужва организирането на установяването на връзка, обмена на блокове от данни и прекратяването на връзката.
6. **Представителният слой (Presentation Layer)** извършва подготовката на данните за предаване — блокуване и др.
7. **Приложният слой (Application Layer)** са в същност приложните програми, които обменят данни помежду си.

При локалните мрежи първите четири слоя обикновено се описват заедно и се наричат протокол на локалната мрежа.

11.3.1.4. Локална мрежа Ethernet

Локалната мрежа Ethernet (фиг. 11.9) при създаването си е типичен представител на мрежите с шинна топология. Съобщителната среда е коаксиален кабел с импеданс 50Ω . Дължината на един сегмент е до 500 метра. В двата края на сегмента задължително се слагат терминиращи резистори със стойност 50Ω . Връзката с други кабелни сегменти се осъществява с помощта на повторител. Скоростта на предаване е 10 Мбита/сек. Информацията се кодира в манчестерски код — единицата се означава с преген фронт, а нулата — със заген — фиг. 11.10.

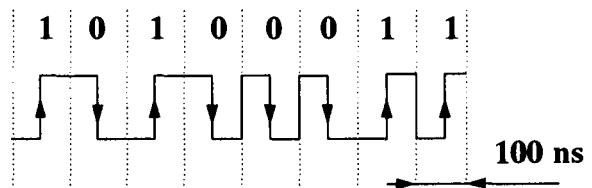


Фиг. 11.9. Локална мрежа Ethernet

Съвременните модификации на мрежата позволяват използването и на топология тип звезда, използване на усукана двойка като съобщителна среда и повишаване на скоростта на обмен на данните до 100 Мбита/сек.

Пакетът данни на локалната мрежа Ethernet (фиг. 11.11) има следната структура:

- ◆ Встъпителна част, съдържаща седем байта AA_{16} и един байт AB_{16} . Шестнадесетичното число AA_{16} има вид в двоичната бройна система 10101010_2 . След предаване в манчестерски код се получава честота 5 MHz. Тя служи за синхронизиране на генератора на приемника с този на предавателя.



Фиг. 11.10. Кодирание на информацията в Ethernet

Глава 11. Управление и комуникация в системите за събиране и обработка на информация

- ◆ Адрес на получателя, дължина 6 байта. Определя уникалния номер на компютърната система получател. Ако най-старшият бит е нула — следват 47 бита адрес на получателя на блока. Ако е единица — следват 47 бита групов адрес. Ако всички битове са единици получателите са всички станции в мрежата.
- ◆ Адрес на подателя, дължина 6 байта. Определя уникалния номер на компютърната система подател.
- ◆ Тип на пакета, два байта.
- ◆ Следват 46 до 1500 байта данни.
- ◆ Блокът завършва с четирибайтов контролен цикличен код, който се получава от предадените до този момент байтове и който приемащата станция също изчислява и сверява с получения. Той е системата за защита от грешки при предаване на данните.

Встъпителна част	адрес получател	адрес подател	тип на пакета	данни	цикличен код
8 байта	6 байта	6 байта	2 байта	46 до 1500 байта	4 байта

Фиг. 11.10. Формат на пакета в локалната мрежа Ethernet

Особеност на мрежите със шинна топология е възможността за възникване на конфликти. Всички абонати следят дали линията е заета или свободна и започват да предават само ако линията е свободна. Независимо от това е възможно да се получи така, че две или повече станции да започнат да предават едновременно, тъй като по време на встъпителната част една станция е започнала да предава, а останалите още не са разбрали това — извършва се апаратно синхронизиране на генераторите им. Конфликтната ситуация се разпознава апаратно и всеки абонат, разпознал конфликтната ситуация, е длъжен да предаде 4 - 6 байта с произволно съдържание. По този начин се гарантира, че всички абонати в мрежата ще разпознаят конфликт-

ната ситуация. След това всички абонати продължават да следят дали линията е заета или свободна, но не започват да предават преди изтичането на генериран отгелно във всяка станция случаен интервал от време.

11.3.1.5. Изграждане на малка локална мрежа

Изграждането на малка локална мрежа в рамките на една или няколко стаи с десетина компютъра включва монтиране на мрежовите платки във всеки компютър, поставяне на свързващите кабели и терминатори, инсталиране на мрежовото програмно осигуряване и настройка на мрежата. Локалната мрежа осигурява само предаването и приемането на блокове от данни. Останалото се извършва от допълнително инсталираното програмно осигуряване. Услугите, които предоставя една такава мрежова среда позволяват:

- да се обяви за общо ползване целия диск на компютъра или отделни директории него;
- да се използват обявените за общо ползване дискове или директории на другите компютри;
- да се обяви за общо ползване принтера, свързан към компютъра, и да се ползват обявените за общо ползване принтери на другите компютри;
- да се обяви за общо ползване модема, свързан към компютъра и да се ползват обявените за общо ползване модеми на другите компютри за връзка с отдалечени компютри, изпращане на факсове и др;
- да се изпращат съобщения (електронна поща) между компютрите в мрежовата среда.

В изградената по този начин мрежова среда може да се работи съвместно по големи проекти, изготвяне на обемисти документи, използване съвместно на големи масиви от данни и др. Освен данни и голяма част от програмите могат да се съхраняват на диска на един компютър, откъдето всички могат да ги извикват и стартират. Локалната мрежа е и неразделна част от горния етаж на големите системи за събиране и обработване на информация. Тенденцията е във

вграждане на способността да се работи в мрежа в самите операционни системи на компютрите, като работата в мрежа става присъща и неотделима част от всеки компютър.

11.3.2. Локална мрежа за индустриално приложение

Изискванията към локалната мрежа за индустриално приложение, както и за свързване на компютърните системи от ниските етажи на големите системи за събиране и преобразуване на информация, се различават съществено.

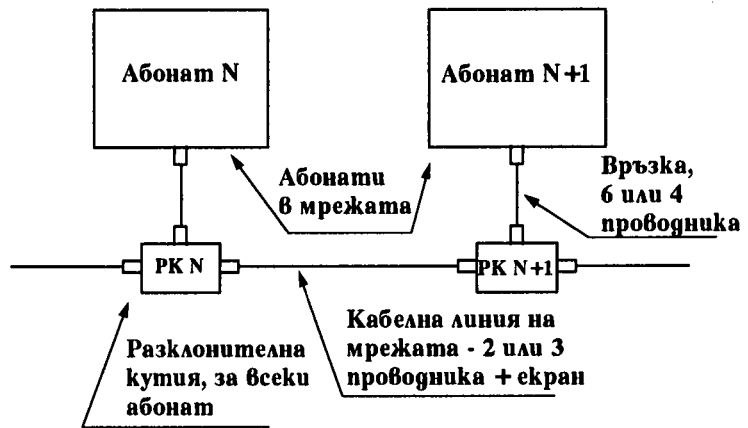
- ◇ Тук не е необходима висока скорост на предаване на данните, тъй като интензивността на информационните потоци с данни не е голяма, защото програми и файлове не се предават по такива мрежи. Освен това значително по-маломощните процесори на тези компютърни системи трудно ще поддържат протоколите на скоростните мрежи.
- ◇ Галваничното разделяне на всяка станция е абсолютно задължително по причини вече коментирани в трета глава.
- ◇ Нивото на електрическите и електромагнитните смущения в индустриални условия е много по-високо и се налага да се вземат изключителни мерки за устойчивост — както апаратни, така и програмни.
- ◇ Желателно е да се минимизира цената на апаратното осигуряване на локалната мрежа, понеже броят на компютърните системи от ниските етажи е значително по-голям.

Тук е дадено като пример реализирана в Техническия Университет в София локална мрежа за индустриално приложение. Тя е типична разработка за тези цели и дава достатъчно добра представа за проблемите и начините за решаването им при организиране на комуникацията в една разпределена система за събиране и обработка на информация.

11.3.2.1. Апаратно осигуряване

Мрежата за индустриални цели е изградена с шинна топология, както е показано на фиг.

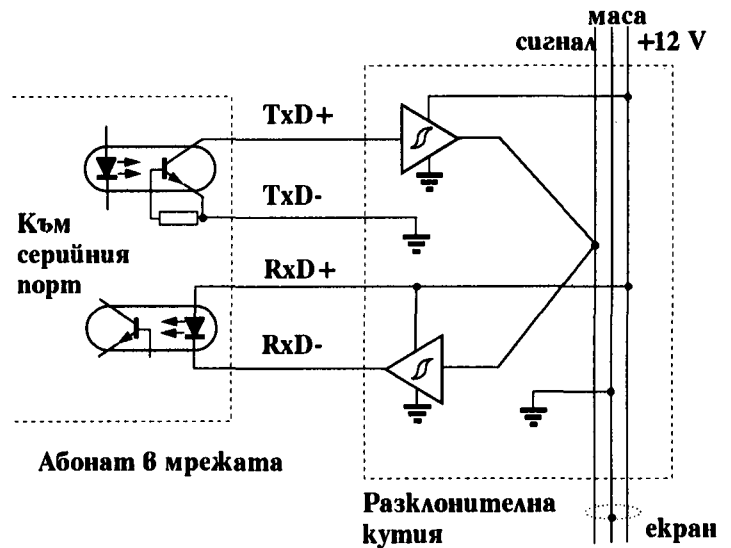
11.12. Носител на информацията е обикновен телефонен чифт, като всеки абонат се свързва към линията посредством разклонителна кутия. За предаване на данните се използват серийните канали на компютърните системи. Данните се предават асинхронно, във формат 8 бита, контрол по четност или нечетност и един стоп бит. Нивата на сигналите са 0 V и 12 V. Скоростта на предаване е от 300 до 9600 бода и зависи от дължината на кабела — колкото е по-дълъг кабелът толкова е по-голям капацитетът му и оттам по-ниска максималната скорост на предаване. Реализирано е оптронно развързване на информационните линии. Блоковата схема на свързване на абоната с разклонителната кутия и линията е показана на фиг. 11.13. Захранването на галванично отделените приемници и предаватели в разклонителните кутии може да се реализира общо — чрез трети кабел по линията, или всяка компютърна система да захранва своите приемник и предавател с галванично изолирано напрежение.



Фиг. 11.12. Локална мрежа за индустриални цели

Фиг. 11.12. Локална мрежа за индустриални цели

Мерките за шумоустойчивост са взети с екранирането на линията, прилагането в апаратния приемник на тригер на Шмид с широк хистерезис — около 4 V, и с натоварване на линията в двата края с резистори. Характеристиката на приемника е показана на фиг. 11.14.



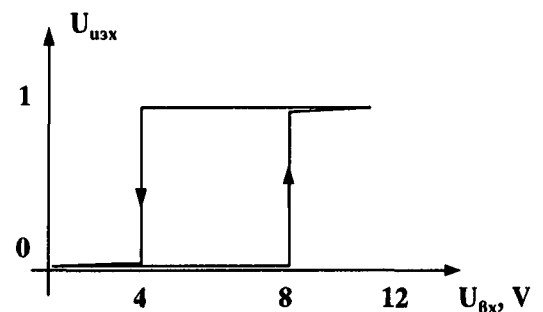
Фиг. 11.13. Апаратна част на локална мрежа за индустриално приложение

11.3.2.2. Протокол

Поради силната зашуменост на средата се налага данните да се блокуват на сравнително къси блокове. Пакетът данни на индустриалната локална мрежа има следната структура:

1. **Водещ символ**, 1 байт. Указва началото на предаването на блок данни. Може да бъде два типа — начало на предаването на съобщение (BC1) и начало на предаването на блок данни (BC2). Първият символ стои в началото на съобщението.

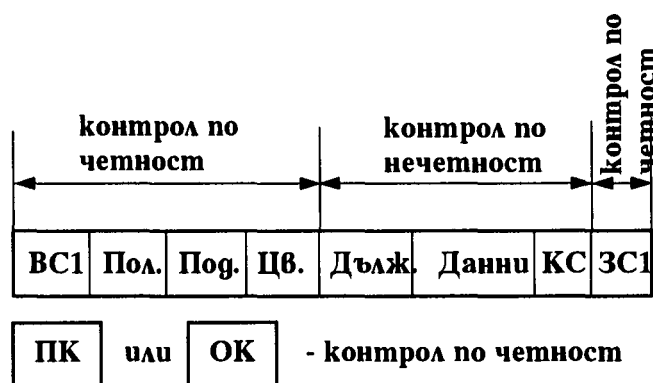
2. **Получател**, 1 байт. Указва адреса на станцията получател на съобщението. Размерът му ограничава броя на станциите в мрежата до 256.



Фиг. 11.14. Характеристика на приемника

3. *Подател*, 1 байт. Число указващо адреса на подателя на съобщението.
4. *Цвят на блока*, 1 байт. Има стойности нула или FF_{16} , които се редуват алтернативно при предаване на многоблоково съобщение. В литературата това се нарича оцветяване на блока в “синьо” или “оранжево”. Смисълът му ще бъде коментирани по-надолу.
5. *Дължина на блока*, 1 байт. Указва броя на байтовете в информационната част. Обикновено е до 128.
6. *Информационна част*, от 1 до 255 байта. Това са данните.
7. *Контролна сума*, 1 байт. Представява младшия байт на сумата на всички предадени досега байтове и служи за проверка на верността на приемането на блока.
8. *Завършващ символ*, 1 байт. Указва края на предаването на блок данни. Може да бъде два типа — край на предаването на съобщение (ЗС1) и край на предаването на блок данни (ЗС2). Първият символ стои в края на последния блок от съобщението. Вторият е в края на първия и следващите блокове на многоблоковото съобщение без последния блок.

За да се отличава служебната информация от информационната част тя се предава с контрол по четност, а байтовете на информационната част — с контрол по нечетност. Схематичното предаване на съобщение, състоящо се от един блок, е показано на фиг. 11.15. След като предаващата станция предаде блока по мрежата приемната станция връща квитанция за приемането на блока. Положителната квитанция (ПК) означава, че приемането е преминало нормално и изчислената контролна сума е съвпа-

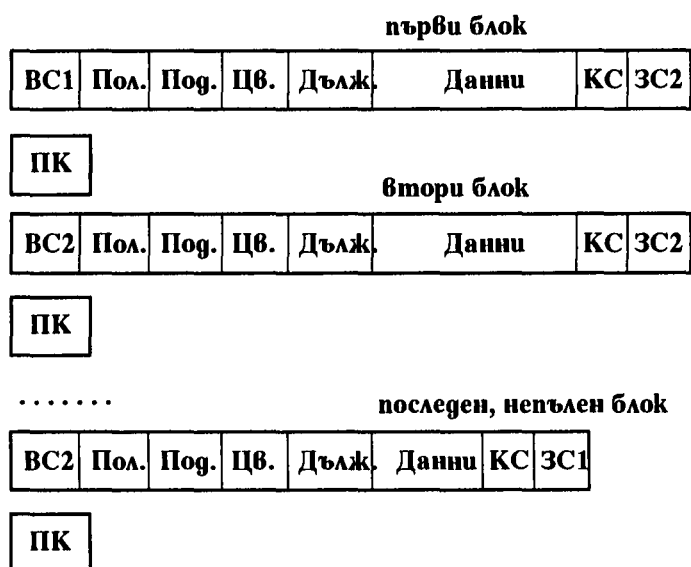


Фиг. 11.15. Предаване на съобщение от един блок

гнала с приетата. Отрицателната квитанция (ОК) изисква предавателя да повтори този блок поради възникнала грешка при приемането на блока.

Предаването на съобщение се извършва многоблоково, блок след блок, ако то е голямо и не се събира в един блок, както е показано на фиг. 11.16. Разделянето на кратки блокове се налага за да не се повтаря цялото дълго съобщение в случай на възникване на грешка, а само сгрешения блок. Освен това при дълго съобщение вероятността да възникне грешка е по-голяма.

Всички действия по мрежата имат своите защитни времена. Ако предаващата станция не получи квитанция до определено време повтаря предаването на текущия блок. Това налага и алтернативното оцветяване на блоковете. Ако квитанцията от страна на приемащата станция е положителна, но в следствие на смущение по линията не се приеме, то предаващата станция ще повтори нормално приетото съобщение. По цвета на блока приемната компютърна система ще разбере, че е нарушена последователността на предаване на блоковете. Прекъсването на предаването се извършва посредством предаване на сигнал BREAK (активиране на линията за време по-дълго от времето за предаване на един цял байт заедно със стартовите и стоповите му битове). Това се разпознава от всички станции, включително и от предаващата, като грешка във формата на предаваните данни.

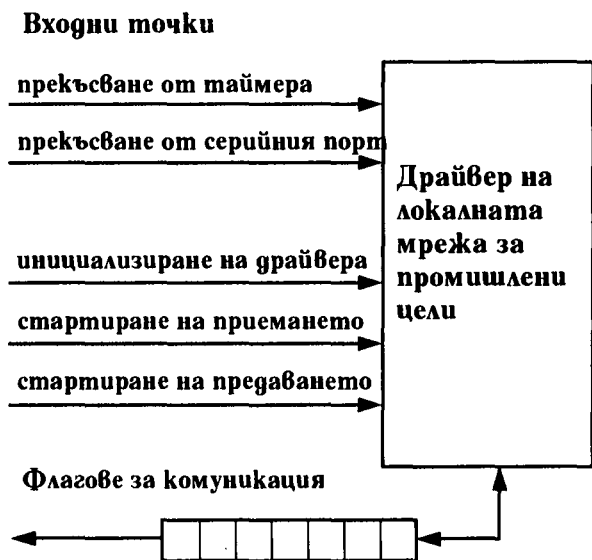


Фиг. 11.16. Предаване на съобщение от няколко блока

Всички станции слушат непрекъснато линията и започват да предават съобщение само когато линията е свободна. Времето за предаване на един символ при скорост на предаване 1200 бода е 9.16 ms. Станциите в мрежата ще разберат, че линията е заета чак 9.16 ms след започването на предаването на водещия символ на първия блок на съобщение от една станция. През това време и някоя друга станция може да започне да предава. Решаването на тази конфликтна ситуация става като всяка станция слуша, приема и сравнява собственото си предаване. В случай на разлика (още някой предава по линията) прекъсва предаването и извежда сигнала BREAK за да уведоми останалите. След това всяка станция продължава да приема, но не започва да предава преди да изчаква време пропорционално на адреса си в мрежата. По този начин се решават конфликтните ситуации в индустриалната мрежа.

11.3.2.3. Структура на драйвера

Драйверът на локалната мрежа за индустриални цели е програма, реализираща всички функции по изпращане и приемане на съобщения по мрежата, защитните времена, повторенията при неуспешно предаване и др. Като алгоритъм програмата не е сложна. Сложното тук е, че драйверът трябва да работи само в прекъсване и да не отнема много време от процесора. Схематично структурата на драйвера е показана на фиг. 11.17. Най-общо казано програмата се състои от инициализираща част, приемник и предавател. Освен собствената си работна зона в оперативната памет драйверът поддържа и дума на състоянието, посредством която чрез установяване и изчистване



Фиг. 11.17. Драйвер за локалната мрежа

на флагове комуникира с приложното програмно осигуряване. Драйверът има следните входни точки:

1. **Прекъсване от системния таймер.** Управлението се предава тук при всяко прекъсване от системния таймер. Това е “чувството за време” на драйвера. Тази входна точка само модифицира броячите на защитните времена и не задържа за дълго процесора.
2. **Прекъсване от серийния канал.** Това е прекъсването от серийния порт, който подава заявка при постъпване на символ от мрежата и при завършване на изпращането на символ. Драйверът обработва всеки постъпил символ в зависимост от режима в който работи. Следи дали линията е заета, дали не пристига съобщение за този абонат, а в режим на предаване — дали полученият символ е същият който е предал.
3. **Инициализация на мрежата.** Тази входна точка се извиква от програмата за начално установяване на компютърната система. Тук се извършва инициализация на серийния порт, задаване на векторите за прекъсване и присвояване на адреса на станцията. Приемникът и предавателят се установяват в подтиснато състояние. След това управлението се връща на викащата програма.
4. **Стартиране на приемането.** Тази входна точка се активира от приложния слой на програмното осигуряване. Той указва адреса на входния буфер и максималната му дължина. Драйверът прави необходимите промени в думата на състоянието (установява необходимите флагове) с което привежда приемника в активно състояние и връща управлението на приложната програма без да задържа продължително процесора. От тук нататък приемникът започва да следи линията и да приема съобщения за тази станция. След приемане на съобщение приемникът маркира това с установяване на съответните флагове в думата на състоянието и преминава в подтиснато състояние. Приложната програма е длъжна да

проверява периодично флагите в думата на състоянието и след приемане на съобщение да го изтегли за обработка и да активира отново приемника от тази входна точка.

5. **Стартиране на предаването.** Тази входна точка също се активира от приложния слой на програмното осигуряване само при условие, че предавателят е в подтиснато състояние, т.е. е завършил предаването на предишното състояние. Приложната програма указва адреса на изходния буфер, дължината на съобщението и позивната на получателя. Драйверът прави необходимите промени в думата на състоянието (установява необходимите флагове) с което привежда предавателя в състояние на готовност и връща управлението на приложната програма без да загърбва продължително процесора. От тук нататък предавателят проверява дали линията е заета и след освобождаването ѝ започва да предава съобщението. След завършване на предаването предавателят маркира това с установяване на съответните флагове в думата на състоянието и преминава в подтиснато състояние.

11.3.2.4. Организиране на приемането на потока от съобщения

В работата на една система за събиране и обработка на информация участват много (десетина и повече) задания под управлението на операционната система за реално време. Тук ще се спрем само на едно типично решение за организирането на потока от съобщения по локална мрежа — приемането на съобщението, обработката му и предаването на евентуалния отговор. По принцип всяка отделна част (компютърна система) от една разпределена слабосвързана система за събиране и преобразуване на информация (вж. глава първа) трябва да може да работи самостоятелно. Обикновено съобщенията по локалната мрежа са кратки команди за смяна на режима или за изискване на натрупана информация. Съобщението-отговор, което се изпраща също не е с голяма дължина. Потокът от съобщения трябва

да се приема и обработва заедно с подържане на нормалната работа на цялата система. Възможно е в даден момент да се получат няколко съобщения едно след друго с темп по-голям от способностите на системата да ги обработи. Това налага и буферизирането на съобщенията.

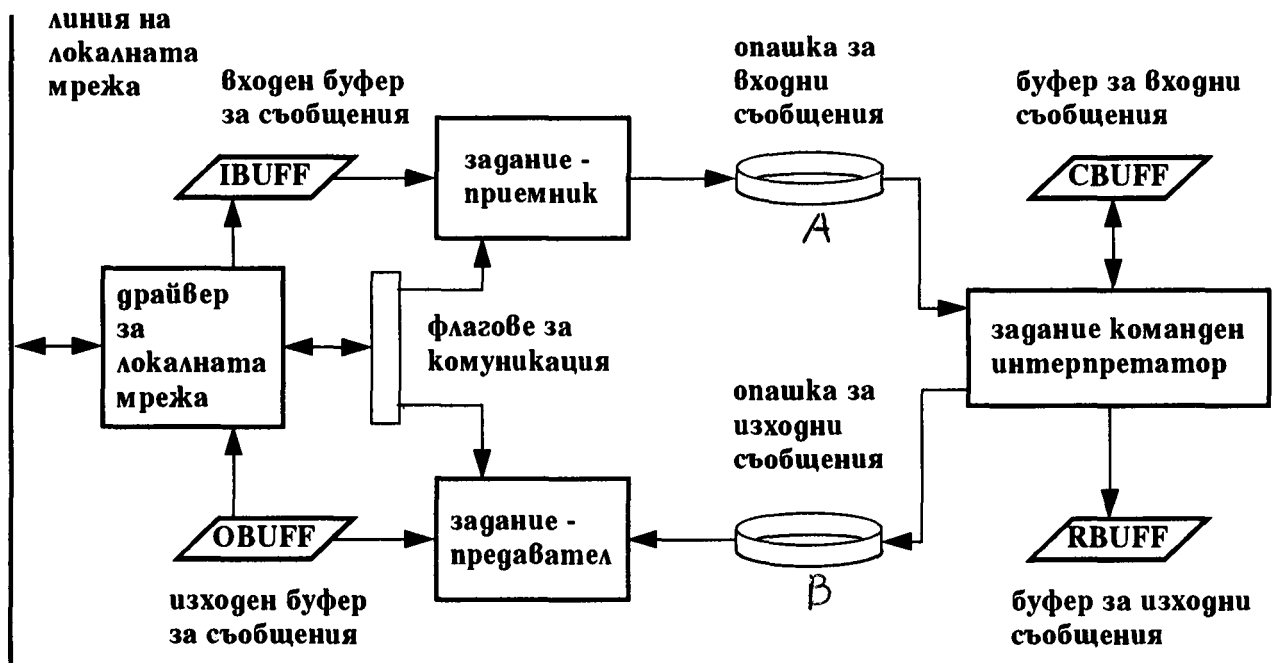
За приемане, буферизиране, обработка и предаване на съобщения по локалната мрежа се грижат три от заданията под управлението на операционната система за реално време: задание приемник, задание интерпретатор на съобщенията и задание предавател (фиг. 11.18). За комуникация между тях се използват опашките А и В, с обем достатъчен да поеме съобщенията докато се обработва едно от тях. Неразделна част от комуникацията чрез локалната мрежа е драйверът за локалната мрежа, който комуникира със заданията посредством думата на състоянието с шестнадесет флага. Програмата за начална инициализация на компютърната система е длъжна да стартира точката за инициализация на драйвера за мрежата още преди стартиране на операционната система за реално време.

Заданието приемник след стартирането си активира приемника на драйвера и след това само проверява дошло ли е съобщение. Ако не е пристигнало съобщение се обръща към операционната система за реално време и се отказва от повече процесорно време с което предоставя своето време на другите задания. В случай на пристигане на съобщение във входния буфер IBUFF (индицира се с установяване на флаг в думата на състоянието) това задание го прехвърля заедно с позивната на подателя в опашката за входни съобщения А и отново активира приемника.

Заданието интерпретатор на съобщенията проверява празна ли е опашката за входни съобщения А. Ако да — обръща се към операционната система за реално време и се отказва от повече процесорно време с което предоставя своето време на другите задания. В случай на непразна опашка изтегля съобщението в буфера за входящи съобщения SBUFF и го обработва. Съдържанието на съобщението зависи от конкретната програма и може да съдържа команда за

изпълнение или за команда за даване на информация. В зависимост от съдържанието му може да се формира съобщение-отговор в буфера RBUFF. След това отговорът, заедно с позивната на подателя, се изпраща в опашката за изходни съобщения В.

Заданието предавател проверява празна ли е опашката за изходни съобщения В. Ако да — обръща се към операционната система за реално време и се отказва от повече процесорно време с което предоставя своето време на другите задания. В случай на непразна опашка се прави проверка дали предавателят на грайвера на локалната мрежа не е активен. Ако предавателят на грайвера за локалната мрежа е активен се изчаква да премине в подтиснато състояние, т.е. да довърши предаването на предишното съобщение. След това се изтегля съобщението в изходния буфер OBUFF и се активира предавателя на грайвера на локалната мрежа, който веднага щом се освободи линията започва да предава съобщението. През това време



Фиг. 11.18. Организиране на приемането, обработката и предаването на съобщенията

заданието предавател чака постъпването на следващото съобщение в опашката за изходящи съобщения В.

Тази примерна организация на приемането, обработката и изпращането на съобщения по локалната мрежа под управлението на операционна система за реално време е една добра илюстрация на целия материал от тази глава — засягат се всички структури от данни за комуникация между заданията и се организира комуникация между компютърни системи.

Обобщение

Операционна система за реално време е програма позволяваща едновременното изпълнение на много програми всяка от които може да приема информация от външната среда, да извежда навън информация и да обменя данни с другите програми. Терминът “реално време” означава, че системата като цяло ще реагира на дадено външно въздействие по-рано от зададен краен интервал от време.

Основните понятия в операционната система за реално време са: задание с неговите атрибути (приоритет и състояние), събитие, ресурс, пощенска кутия, семафор и системна единица за време. Различават се следните основни състояния на едно задание: готовност, активно, чакащо, блокирано. Заданието преминава от едно състояние в друго под управлението на операционната система за реално време.

Заданията под управлението на операционната система за реално време комуникират помежду си посредством флагове и опашки (стек FIFO).

За връзка между отделните компютърни системи се използва компютърна мрежа — комплекс от апаратни и програмни средства за обмен на данни. Различават се локални и глобални компютърни мрежи. Всяка компютърна мрежа може да бъде изградена с различна топология — звезда, кръг, шинна и комбинирана.



Контролни въпроси

1. Що е операционна система за реално време?

Методи, устройства и системи за събиране и преобразуване на информация

2. Кои са основните състояния на заданието и кого то преминава от едно състояние в друго?
3. Кои са основните примитиви за работа с опашка?
4. Що е компютърна мрежа и кои са нейните характеристики?
5. Кои са особеностите на локалните мрежи за работа в промишлени условия?

12. Системи за управление, събиране и обработка на данни в промишлеността



Ключови думи

- системи за управление в промишлеността
- промишлен контролер
- дискретно производство
- непрекъснато производство
- промишлен регулатор

Какви знания и умения ще придобие?

Къде се прилагат?

Тази глава е посветена на особеностите при изграждането на системите за събиране и обработка на информация, предназначени за работа в промишлени условия. Разглежда се общата структура на разпределена система за управление на промишлени обекти и особеностите на нейното най-ниско ниво — промишления контролер.

Получените знания ще приложите при проектирането на разпределени системи за управление на индустриални обекти.

Въведение

От произвежданите в света микропроцесори и микроконтролери повече от половината са осембитови еднокристални микропроцесори и микроконтролери. Те са предназначени за въграждане в системи за управление, главно на обекти в промишлеността и в бита. Около 40% са четирибитови еднокристални микроконтролери, предназначени за въграждане в битова електроапаратура — микровълнови фурни, автоматични перални, съдомиялни машини, дистанционни управления, управление на аудио и видео апаратура и др. Шестнадесет и тридесет и два разрядните микропроцесори (от които толкова много се възхищаваме), специализираните процесори за цифрова обработка на сигнали (с тяхната огромна изчислителна мощност) заемат по-малко от десет процента от произведените бройки процесори на световния

пазар. В ценово отношение нещата се променят, разбира се, защото последната категория процесори са значително по-скъпи.

Определено може да се твърди, че създаването на микропроцесорите и развитието на микропроцесорната техника има съществено положително влияние върху развитието на човешката цивилизация като цяло. То се отразява като повишаване на качеството на живот в битата и повишаване на производителността и качеството на изделията в промишлеността. И това се дължи не на това, че секретарката на директора на завода работи на мощен тридесет и два разряден персонален компютър и печата кореспонденцията на лазерен принтер, а защото в същия завод работят десетки и стотици микропроцесори, някъде в цеховете, затворени в металните корпуси на машините, които повишават производителността и качеството на производството.

12.1. Предназначение и специфични особености

12.1.1. Предназначение

Системите за управление, събиране и обработка на данни в промишлеността са частен случай на обобщения модел на системите за събиране и обработка на информация детайлно разгледан в първа глава. Те са предназначени да поемат непосредственото управление на машините, извършващи производството, да осигурят предаването на команди към тези машини и да събират информация за протичането на производствения процес. В случая обектът, разглеждан в първа глава, е машината осъществяваща производствения процес.

12.1.2. Специфични особености на индустриалните системи

12.1.2.1. Среда за работа

Средата, в която ще работят системите за управление, събиране и обработка на данни има своите специфични особености в промишлени условия:

- ⇒ значително ниво на електромагнитни смущения, които индуцират силни смущаващи сигнали в свързващите кабели;

- ⇒ силна замърсеност на средата с прах, вредни изпарения, водни пари, кондензирана влага и гр.;
- ⇒ силни смущения по захранващата мрежа, свързани с включването и изключването на мощни консуматори, често с индуктивен характер, водещи до токови удари;
- ⇒ нестабилно захранващо напрежение в мрежата и кратковременни прекъсвания;
- ⇒ значително по-широк интервал на изменение на температурата.

12.1.2.2. Допълнителни изисквания

Към системите за управление на промишлени процеси се предявяват и някои допълнителни изисквания:

- ◆ значително по-високи изисквания за отказоустойчивост и надеждност, защото спирането на една промишлена система за управление означава спиране на цялата производствена линия или цех, което е свързано с големи загуби;
- ◆ с тях да може да работи персонал с ниска квалификация — работник или техник;
- ◆ да са устойчиви на неправилни команди и на опити за умишлено подаване на неверни команди с цел предизвикване на спиране на производствения процес.

12.1.2.3. Мерки за приспособяване към условията за работа

За да мога системите за управление, събиране и обработка на информация да работят при тези условия и да отговарят на специфичните изисквания се взема цял комплекс от мерки: схемотехнични, конструктивни, програмни и системотехнични.

12.1.2.3.1. Схемотехнични мерки

Схемотехничните мерки означават така да се изгради схемата на системата за управление, събиране и обработка на информация, че да работи съгласно изискванията и при най-тежкото струпване на неблагоприятните условия.

Това касае особено схемотехниката на входните и изходните блокове. Повечето от особеностите бяха разгледани във втора глава. Загължително се поставят защиты на всички входове и изходи от пренапрежения, неправилно включване и повреда на датчиците и изпълнителните механизми. Работата при тежки условия означава и презапасяване на елементите по мощност, гранични напрежения, максимални токове и др.

Загължителна особеност на микропроцесорните системи с повишени изисквания за надеждност е системата за наблюдение на работата на процесора (watch dog). В програмното осигуряване са поставени инструкции, които да рестартират тази система. Те се изпълняват периодично в процеса на нормалната работа на микропроцесорната система. Ако в следствие на токов удар или силно електромагнитно смущение управляващият микрокомпютър излезе от програмата си (увисне), то след изтичането на определено време системата за наблюдение на процесора ще подаде апаратен сигнал за начално установяване (Reset).

Нестабилното захранващо напрежение и честото му кратковременно отпадане предявява сериозни изисквания и към захранващия блок. Те са описани подробно в първа глава. Като правило промишлените системи имат резервирано захранване, локално или централно, или поне батерийно захранване на оперативната памет.

12.1.2.3.2. Конструктивни мерки

Конструктивните мерки включват подбор на кутия, оформяне на пулта за управление и индикация, отвеждане на топлината от кутията и др. Работещите непосредствено при машините компютърни системи за управление като правило се поставят в херметизирани кутии за предпазване от влага, пари, прах. Това означава минимизиране на консумацията от захранващия блок, за може да се отведе топлината от кутията. Херметизацията изисква и вземане на допълнителни мерки за конструктивното оформяне на пулта за управление — бутони и индикация. Куплунзите трябва да са механически устойчиви

и да не позволяват неправилно включване или разменяне поради ниската квалификация на обслужващия персонал.

12.1.2.3.3. Програмни мерки

Програмните мерки включват повишена устойчивост на алгоритмите за управление и програмната им реализация при въвеждане на информацията от обекта и нейната обработка. Особено внимание се обръща на начините за възстановяване на нормалната работа на системата след аварийно рестартиране и при възникване на аварийни ситуации в управлявания процес.

12.1.2.3.4. Системотехнични мерки

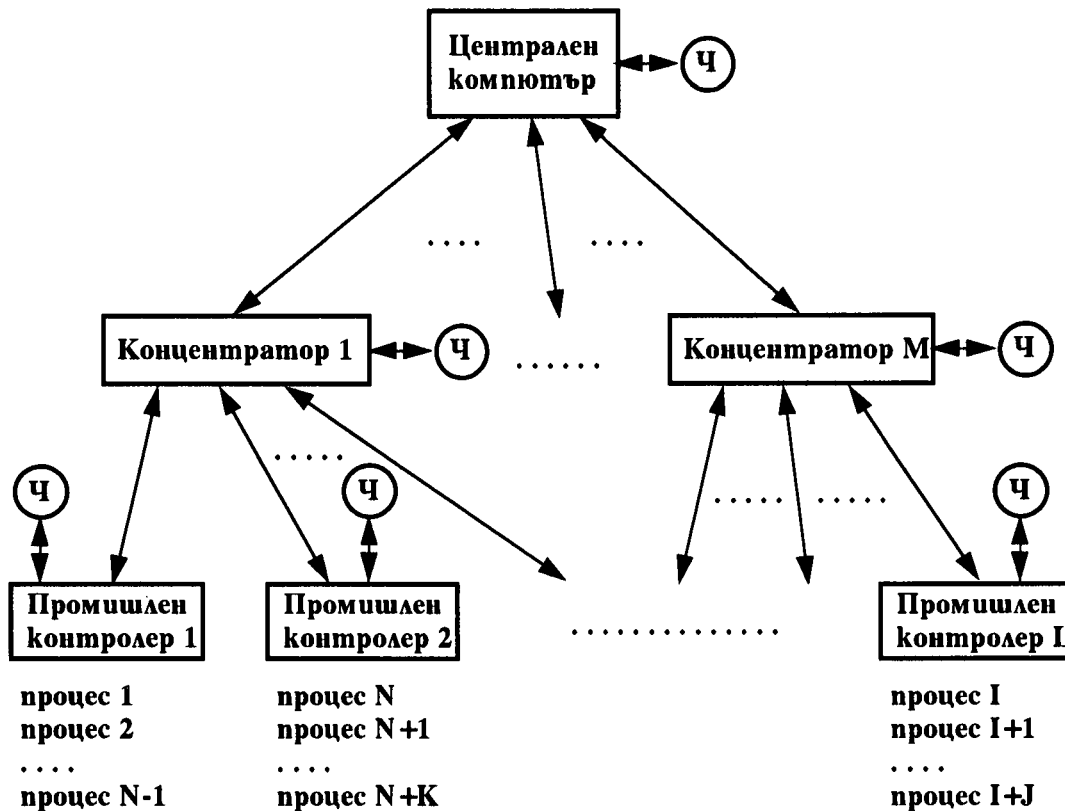
За повишаване на надеждността на цялата система за управление, събиране и обработка на данни се търси такава архитектура, която да позволява продължаването на работата и при отказ на част от компонентите. Най-често се изгражда многокомпютърна, разпределена, слабосвързана система. Всеки компонент на системата е желателно да може да работи самостоятелно, без връзка с другите. Управлението от оператора да може да се извършва и от локален пулт.

Удовлетворяването на всички тези изисквания съществено оскъпява системите за управление, събиране и обработка на информация. Затова в самите помещения на цеховете се разполага само минимално необходимите компоненти от разпределената система. Те работят при най-тежките условия и за тях се вземат всички мерки. Останалата част от компонентите на разпределената система се разполага в отделни помещения, където условията са значително облекчени.

12.2. Блокова схема на система за събиране и обработка на информация за целите на управлението на промишлени обекти

Системите за управление, събиране и обработка на информация от промишлени обекти се изграждат като правило с пирамидална архитектура и с въвеждането на строга йерархия на решаваните задачи и изпълняваните функции. Обикновено системата има три основни етажа, както е показано на фиг. 12.1 — ниво промишлени контролери,

ниво концентратори и нивото на централния компютър. На всеки от етажите човекът-оператор (означен с Ч) може да окаже въздействие върху управлението на производствения процес.



Фиг. 12.1. Обща блокова схема на система за събиране и обработка на информация за целите на управлението на промишлени обекти

12.2.1. Промислени контролери

12.2.1.1. Роля и функции

Първият етаж е нивото на компютърните системи за непосредствена обработка на производствения процес. Тези компютърни системи се наричат промишлени контролери. Те възприемат информацията директно от обекта за управление, извършват необходимите програмни действия и подават информация към изпълнителните механизми за въздействие върху производствения процес. Един промишлен контролер обикновено управлява няколко контура или процеса —

цифрови регулатори, изпълнителни механизми и др. Промислените контролери могат самостоятелно да подържат необходимите производствени режими, да извършват непосредственото управление на производствения процес и да натрупват статистическа информация за хода му. Всеки промишлен контролер управлява обикновено една машина или част от целия производствен процес. Притежава малък операторски пулт от който при нужда може да се задават параметри на производствения процес.

12.2.1.2. Апаратна реализация

Промисленият контролер е микропроцесорна система, реализирана най-често на базата на някоя от разпространените осембитови микропроцесорни фамилии. В отделни случаи при извършването на по-сложни функции (машинно зрение и разпознаване на обекти) или за управлението на по-сложни процеси се използват и процесори с по-голяма изчислителна мощност — шестнадесет и тридесет и два разрядни.

12.2.1.3. Програмно осигуряване

За реализирането на програмното осигуряване на промисления контролер се използват главно асемблерни езици, като част от програмното осигуряване се пише и на език от високо ниво, обикновено С. Най-често работят под управлението на операционна система за реално време. Връзката с концентраторите се осъществява посредством някакъв вид интерфейс или с помощта на индустриална локална мрежа. За комуникация с оператора понякога се използват и специализирани езици за описване на действията на промисления контролер. За задаване на необходимите действия на металорежеща машина с цифрово програмно управление, например, се използва специализиран език за описване на детайла, който трябва да се изработи. Командите на този език се интерпретират от промисления контролер. Намерили са разпространение и така наречените релейно-контактни езици за програмиране на промислени контролери, при които алгоритъмът на работа се описва като електрическа схема, изградена с ре-

лета, контакти, закъснителни елементи и броячи. Получената “електрическа схема” се компилира до програма за процесора на промишления контролер.

12.2.2. Концентратори

12.2.2.1. Роля и функции

Следващото ниво на промишлените разпределени системи за управление е нивото на концентраторите. Това са компютърни системи, притежаващи информационна връзка с промишлените контролери. Всеки концентратор е свързан с определен брой от промишлените контролери, които управляват свързана част от производството — линия, цех и др. Концентраторът “отговаря” за тази част от производствения процес. Периодично изтегля от промишлените контролери натрупаната информация за хода му и също извършва управление, но на по-високо ниво. Задава общите параметри на производствения процес на промишлените контролери. Има значително по-развити средства за комуникация с оператора, които му позволяват да представя повече информация и да възприема по-широк спектър от команди.

12.2.2.2. Апаратна реализация

Апаратната реализация на концентраторите силно варира в зависимост от разположението им (в цеха или в отделно помещение) и функциите им. Разположените в производствените помещения концентратори трябва да отговарят конструктивно на изискванията за работа при тези условия. Концентраторът може да бъде изработен на базата на някоя от по-мощните осем или шестнадесет битови микропроцесорни фамилии. При необходимост от запомняне и обработка на по-големи обеми от информация (т.е. наличие на дискова памет и дискова операционна система) може да се постави и съвместима с персоналните компютри система. Това са така наречените индустриални персонални компютри. Поради високите изисквания за надеждност и специалните изисквания към конструкцията — от компютъра до клавиатурата и монитора, те са значително по-скъпи.

12.2.2.3. Програмно осигуряване

Делът на написаната на асемблер част от програмното осигуряване намалява значително — това са обикновено грайверите за връзка с промишлените контролери (за управление на връзката по дадения интерфейс или за индустриалната локална мрежа). Основната част на програмното осигуряване на концентраторите е написано на език от високо с общо предназначение, обикновено С. В редица случаи, при обработка на големи масиви от данни, се използват и специализирани езици за управление на бази данни.

12.2.3. Централен компютър

12.2.3.1. Роля и функции

Централният компютър на системата за управление, събиране и обработка на информация за индустриални цели е свързан информационно с всички концентратори. Неговата роля в разпределената система за управление е да изтегли натрупаната в концентраторите информация, да я обработи и обобщи по подходящ начин и да спомогне оператора във вземането на решение за управлението на производството като цяло. Притежава силно развита периферия за връзка с човека — клавиатура, дисплей, печатащо устройство и др. Много често е свързан и в локалната мрежа на администрацията на цялото предприятие.

12.2.3.2. Апаратна реализация

Като правило това е мощен персонален компютър снабден с необходимата периферия — цветен дисплей, клавиатура, мишка, качествен принтер. Разположен е във от производствените помещения, в следствие на което отпадат голяма част от изискванията към конструкцията му. Остават строгите изисквания за отказоустойчивост и надеждност. В редица случаи могат да се използват и компютри от друг тип, например мини машини, обикновено по съображения за съвместимост с останалата изчислителна техника в предприятието.

12.2.3.3. Програмно осигуряване

Основната част от програмното осигуряване на централния компютър се пише на езици от високо ниво, обикновено специализирани езици за обработка на бази от данни. Те позволяват бързо написване на програмното осигуряване и подготовка за печат на необходимите справки, документи, отчети и др.



Изграждане на разпределена промишлена система

Нека имаме един завод за производство на безалкохолни напитки или бира. Ще разгледаме част от разпределената система за управление на този завод. Линията за бутилиране на произведените безалкохолни напитки обхваща следните машини:

- ◇ депалетизатор — машина, която сема касите ред след ред от палетата и ги поставя на транспортната линия;
- ◇ декрейтер — машина, която изважда бутилките от касите;
- ◇ миячна машина за измиване на бутилките и обработката им с дезинфекционни разтвори;
- ◇ пълначна машина, която налива определено количество питие във всяка бутилка и я затваря;
- ◇ контролър на нивото в напълнените бутилки, като пълните под нормата се отстраняват;
- ◇ машина за залепване на етикети;
- ◇ крейтер — машина за групиране и нареждане на бутилките в празните каси, придвижени по отделна транспортна линия от декрейтера;
- ◇ палетизатор — машина за нареждане на касите на палетата и опаковането му.

Всяка от тези машини се управлява от собствен промишлен контролер. Той изпълнява необходимите операции самостоятелно, със зададен темп на работа и натрупва статистическа информация за преминалите бройки палети, каси и бутилки.

Всички промишлени контролери на машините от линията за бутилиране са свързани с концентратор. Това е микропроцесорна система, отговаряща за управлението на линията като цяло. Концентраторът

се свързва периодично с промишлените контролери от линията, разпитва ги за текущото им състояние и изтегля натрупаната статистическа информация. След обработка на получените данни взема необходимите решения за управлението на линията. При спиране на някоя от машините от линията, например поради задръстване на транспортната лента, счупване на бутилка или по друга причина, концентраторът подава команди към предходните машини да намалят темпа на работа или въобще да спрат за да се избегне принудителното им спиране при препълване на транспортните ленти, което е свързано с удрянето на бутилки, каси и повишава вероятността от лавинообразно нарастване на принудително спрелите машини. При силно нарастване на принудителните престои подава команди към промишлените контролери да намалят темпа на работа с което се намалява вероятността за принудително спиране и оттук се получава нарастване на производителността на цялата линия. Събраната от промишлените контролери статистическа информация се обобщава и съхранява.

Компютърът от най-горното ниво има информационна връзка с концентраторите на всички производствени звена — производство на напитките, бутилиране, складове за суровина и за готова продукция и т.н. Той периодично изтегля събраната статистическа информация, обобщава я и изготвя необходимите отчетни документи за изтеклата смяна, денонощие, месец и т.н. Тук също се извършва процес на управление, обикновено с активното участие на човека.

12.3. Обобщен модел на промишлен контролер

Промисленият контролер, както вече беше отбелязано, е микропроцесорна система предназначена за работа в промишлени условия. Схемотехнично и конструктивно тя трябва да е изпълнена така, че да работи нормално в тези условия. Ако разгледаме промисления контролер като затворена кутия, той трябва да има следните връзки с външния свят: захранване, аналогови входове, цифрови входове, аналогови изходи, цифрови изходи, операторски пулт, връзка с по-горното ниво на системата за управление на промишлени обекти — фиг. 12.2.

12.3.1. Захранване на промисления контролер

Промислените контролери се захранват най-често от променливотоковата мрежа 220 V, 50 Hz. В отделни случаи захранването се извършва от специализирана захранваща мрежа с напрежения 48 V, 110 V и гр., с постоянно напрежение или с променливо напрежение с честота 50, 60 или 400 Hz.



Фиг. 12.2. Обобщен модел на промишлен контролер

12.3.2. Аналогови входове

Аналоговите входове са предназначени за свързване към датчиците с аналогов изход, които възприемат различни аналогови параметри от производствения процес — температура, налягане, концентрация на разтвори, рН на разтвори, дебит на флуиди и гр. Много често тези датчици имат своя собствена електронна част — усилватели, функционални преобразуватели. За да могат да се свързват датчици на различни производители към промислените контролери обикновено параметрите на аналоговите изходи на датчиците са стандартизира-

ни. Аналоговите входове на промишлените контролери най-често са оразмерени да възприемат:

- аналогово напрежение, от 0 до 10 V, от -5 до +5 V;
- ток, от 0 до 100 mA, от 4 до 20 mA.

Предаването по кабел на аналогов сигнал чрез ток в индустриални условия е по-устойчиво на смущения, защото входното съпротивление на токовия вход на промишления контролер е близко до нулата. Това натоварване на кабела намалява амплитудата на индуктираните смущаващи сигнали. Вторият стандарт за предаване на аналогов сигнал чрез ток (от 4 до 20 mA) позволява и да се провери свързан ли е датчикът — ако входният ток е под 4 mA това означава, че датчикът не е свързан, повреден е или има прекъсване на свързващите кабели.

12.3.3. Цифрови входове

Цифровите входове служат за възприемане на сигнал с две състояния — включено/изключено, наличие/отсъствие. Обикновено са контакти, които при затворено състояние подават напрежение на цифровия вход на промишления контролер. При поставяне на по-сложни датчици — преминаване на обект, ниво на течност и др., които имат собствена електронна част, изходът им е обикновено транзистор в ключов режим. Той подава постоянно напрежение към входа на промишления контролер. Съществуват датчици, които подават към промишления контролер импулсни поредици със зададени параметри — честота или коефициент на запълване. Най-често срещаните параметри на входния сигнал на цифровите входове на промишлените контролери са:

- променливо напрежение, 12 V, 48 V, 110 V, 220 V;
- постоянно напрежение, 5 V, 10 V, 12 V, 24 V;
- постоянен ток, 20 mA, 100 mA, 4 или 20 mA.

Последното се използва при системи с особено високи изисквания към надеждността — транспорт, авиация, енергопроизводство, електроснабдяване и др. Този стандарт позволява на промишления конт-

ролер да различи състоянието на липса на сигнал от прекъсната връзка с датчика.

12.3.4. Аналогови изходи

Аналоговите изходи служат за свързване на промишления контролер към изпълнителни устройства, чиито параметри могат да се регулират плавно в някакъв диапазон — серво система за управление на оборотите на високомоментен постоянен ток електродвигател, управление на ъгъла на отваряне на шибър за подаване на флуиди, задаване положението на работен органи, изходна мощност на електрически нагревател и др. За да могат промишлените контролери да се свързват към електронното управление на изпълнителни механизми, произведено от различни фирми се прибегва до уеднаквяване на параметрите на изходните сигнали на промишлените контролери. Най-често срещаните са както при аналоговите входове:

- аналогово напрежение, от 0 до 10 V, от -5 до +5 V;
- ток, от 0 до 100 mA, от 4 до 20 mA.

12.3.5. Цифрови изходи

Цифровите изходи служат за управление на изпълнителни механизми с две състояния — включено и изключено. Това са електродвигатели, помпи, клапани. В отделни случаи посредством цифровите изходи се формират импулсни поредици със зададени параметри — честота, коефициент на запълване. В зависимост от предназначението цифровите изходи могат да бъдат:

- аналогово напрежение, 0 или 5 V;
- контакт на реле за напрежения до 300 V и токове до 10 A;
- тиристор или триак, които могат да включват безконтактно товари в електрическата мрежа с напрежение 220 V с мощност до 2000 W;
- постоянен ток 4 или 20 mA.

Управлението на по-мощни консуматори се извършва като промишленият контролер управлява контактор включващ и изключващ мощния консуматор.

12.3.6. Пулт за управление

Пултът за управление на промишления контролер обикновено е течнокристална буквено цифрова индикация от един или два реда и клавиатура с десетина бутона. В редица случаи може да бъде и няколко светодиода и бутона. Понякога фирмите производителки прибягват до използването на преносими пултове за управление, които се носят от оператора и се включват към промишления контролер само при нужда. Посредством пулта за управление обикновено се извършват сервизни дейности — проверка състоянието на входовете, задаване на състояние на изходите. При отказ на системата за връзка с по-горното ниво от пулта за управление могат да се разглежда натрупаната информация и да се задават основни параметри.

12.3.7. Връзка с по-горното ниво

Обикновено това е специализирана промишлена локална мрежа. В отделни случаи се прибягва до използването на някой от стандартните интерфейси за обмен на данни като RS232 или RS422.

12.4. Решавани задачи от промишлените контролери

В зависимост от приложението се различават две основни групи промишлени контролери — за непрекъснато и за дискретно производство. Непрекъснатите производства са химията, металургията, биореакторите, част от производства в хранително-вкусовата промишленост. При тях преобладават промишлените контролери с аналогови входове и изходи. Цифровите входове и изходи имат спомагателни функции. Дискретните производства са машиностроенето, електрониката и др. При тях преобладават промишлените контролери с цифрови входове и изходи. Сравнително по-рядко се налага използването на аналогови входове и изходи.

12.4.1. Промислени контролери в непрекъснатите производства

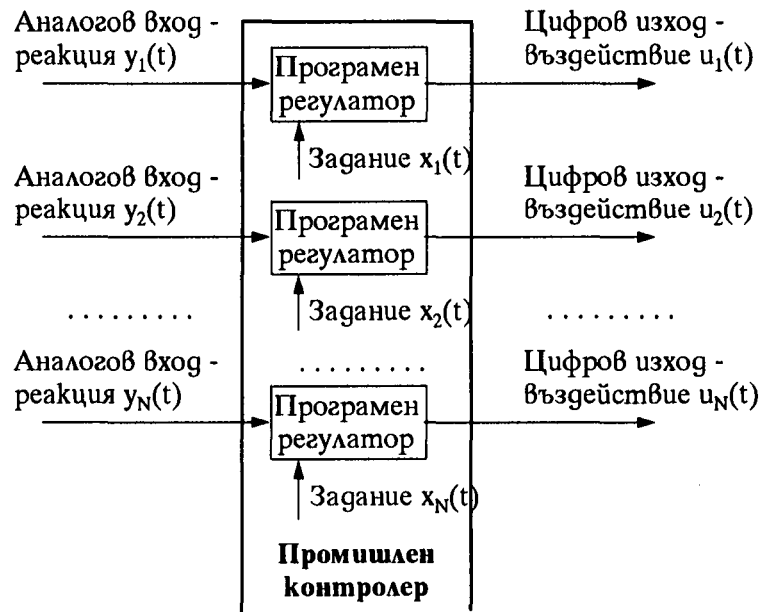
Основен елемент реализиран от промишлените контролери в непрекъснатите производства е промишленият регулатор. Блоквата схема на промишлен контролер за непрекъснатите производства е

показана на фиг. 12.3. В един промишлен контролер се реализират обикновено до осем промишлени регулатора.

12.4.1.1. Промислен регулатор

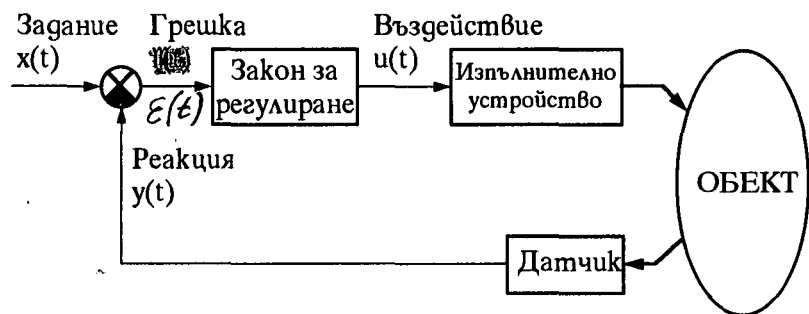
Всеки промишлен регулатор е част от един кръг за подържане на даден параметър (фиг. 12.4), който включва:

- ◆ обект за управление,
- ◆ датчик за снемане на аналогова величина,
- ◆ аналогов вход,
- ◆ закон за регулиране,
- ◆ аналогов изход и
- ◆ изпълнително устройство за въздействие върху обекта за управление.



Типичен пример е пещ (обект за управление), термодатчик (платина 100 или термодвойка със съответните им усилватели), аналогов вход на промишления контролер (един канал от АЦП), закон за регулиране (обикновено ПИД), аналогов изход (ЦАП или импулсен изход с широчинно импулсна модулация), изпълнително устройство (отваряне на шибер за подаване на горивото или регулиране на изходната мощност на електрически нагревател).

Фиг. 12.3. Промислен контролер за непрекъснатите производства



Фиг. 12.4. Промислен регулатор за подържане на зададен параметър

Най-често стойността на подгържаната величина се задава от по-горното ниво на системата за управление — концентратора, но съществуват реализации в които на промишленият контролер може да се задава закона за изменението във времето на желаната стойност на изходната величина (температурата в предишния пример) за извършване на даден технологичен процес. Този закон също се изпраща от по-високото ниво в системата за управление.

12.4.1.2. Закон за регулиране и параметри

В промишлените регулатори обикновено се реализира ПИД закон за регулиране на изходната величина:

$$u(t) = k_p \cdot \varepsilon(t) + \frac{1}{\tau_i} \cdot \int_0^t \varepsilon(\theta) \cdot d\theta + \tau_d \cdot \frac{d\varepsilon(t)}{dt},$$

където $u(t)$ е въздействието върху обекта, $\varepsilon(t) = x(t) - y(t)$ е грешката между желаната стойност $x(t)$ и отчетената изходна стойност $y(t)$, k_p , τ_i и τ_d са параметрите на регулатора, съответно коефициент на пропорционалност, интегрираща и диференцираща времеконстанти.

Тъй като компютърните промишлени регулатори работят в дискретно време, то дискретният ПИД закон за регулиране има следният вид:

$$u(kT) = k_p \cdot \varepsilon(kT) + \frac{T}{\tau_i} \cdot \sum_{i=0}^k \varepsilon(iT) + \tau_d \cdot \frac{\varepsilon(kT) - \varepsilon((k-1)T)}{T},$$

където T е периодът на дискретизация. Горната формула е свързана с голямо количество изчисления и изисква голям обем от памет за запазване на цялата история на процеса. Затова в практиката се работи по модифицирани формули. Нека извадим от $u(kT)$ предишната стойност на въздействието $u((k-1)T)$:

$$\begin{aligned}\Delta u(kT) &= u(kT) - u((k-1)T) = \\ &= k_p \cdot (\varepsilon(kT) - \varepsilon((k-1)T)) + \frac{T}{\tau_i} \cdot \varepsilon(kT) + \\ &+ \frac{\tau_d}{T} \cdot (\varepsilon(kT) - 2 \cdot \varepsilon((k-1)T) + \varepsilon((k-2)T))\end{aligned}$$

Прегрупираме горното равенство и означаваме за по-кратко момента време като индекс:

$$\begin{aligned}\Delta u_k &= u_k - u_{k-1} = \\ &= \left(k_p + \frac{T}{\tau_i} + \frac{\tau_d}{T} \right) \cdot \varepsilon_k + \left(-k_p - 2 \cdot \frac{\tau_d}{T} \right) \cdot \varepsilon_{k-1} + \left(\frac{\tau_d}{T} \right) \cdot \varepsilon_{k-2}\end{aligned}$$

и в крайна сметка получаваме за въздействието върху обекта за управление:

$$u_k = u_{k-1} + q_0 \cdot \varepsilon_k + q_1 \cdot \varepsilon_{k-1} + q_2 \cdot \varepsilon_{k-2},$$

където от параметрите на регулатора предварително са изчислени константите:

$$q_0 = k_p + \frac{T}{\tau_i} + \frac{\tau_d}{T}$$

$$q_1 = -k_p - 2 \cdot \frac{\tau_d}{T}$$

$$q_2 = \frac{\tau_d}{T}$$

Изведената по този начин формула се изчислява много по-бързо и изисква памет само за съхранение на двете предишни стойности на грешката и предишната стойност на въздействието. Цифровите регулатори имат още два параметъра:

- зона на нечувствителност ε_{min} , под която грешката се приема за нулева;
- максимална стойност на изходното въздействие u_{max} , като в процеса на работа на регулатора не се допуска числото u да превишава тази стойност.

Параметрите на всеки от цифровите регулатори в промишления контролер трябва да могат да се задават от компютърната система от по-високото ниво. При използване на друг закон за управление (П или ПИ) съответните времеконстанти се задават нули. Промис-

лените контролери за непрекъснатите производства имат и някои допълнителни функции, подпомагащи процеса на проектирането на регулаторите и подържането на оптимални от дадена гледна точка параметри на цифровия регулатор. Такива са непрекъснат запис във времето на поредицата въздействия — реакция $(u(kT), y(kT))$, възможността за подаване на тестови въздействия и запис на реакцията и др. Тези функции са незаменими в началния етап на запускане на системата за управление, идентификацията на обекта и проектиране на регулатора.

12.4.1.3. Проектиране на промишлен регулатор

Проектирането на цифровите регулатори за конкретен обект за управление протича на няколко етапа:

- ⇒ Идентификация на обекта за управление. Извършва се обикновено чрез подаване на тестово въздействие — единично стъпало или дираков импулс. Записва се реакцията на системата обект+изпълнително устройство като функция на времето. Намират се стойностите на коефициентите на линейна дискретна система (вж. глава шеста) с максимално близка реакция на същото въздействие, т.е. изгражда се модел на обекта. Тук най-често се използва методът на най-малките квадрати. Обикновено се търси моделиране с линейна дискретна система от нулев, първи или най-много втори ред.
- ⇒ Намира се експериментално или се изчислява зависимостта на установената във времето изходна стойност на системата обект+изпълнително устройство от въздействието $y_{уст} = f(u)$, където това е възможно.
- ⇒ Намират се коефициентите на ПИД регулатора, като за целта се формулират изискванията към проектирания промишлен регулатор: максимално пререгулиране, вид на преходния процес и др. За намиране на числените стойности на коефициентите се използва различен математически

апарат — z-преобразуване, преобразуване на Лаплас, оптимизационни методи и др. Независимо от използвания метод в края на проектирането се прави симулация на системата регулатор+обект+изпълнително устройство, като се отчитат и дискретността на въвежданите входни величини и дискретността на въздействието както във времето (дискретизация), така и в амплитудата (квантуване).

Възможно е този процес да се повтаря и по време на нормалната експлоатация на регулатора, като за идентификация се използва записаната от промишления контролер поредица въздействие-реакция на системата обект+изпълнително устройство. По този начин се осъществява адаптиране на регулатора при промяна във времето на параметрите на обекта (пещта е навън и нейните параметри се изменят при смяната на сезоните) или на изпълнителното устройство (използват се горива с различна калоричност). В много редки случаи, при силно нелинейни системи или при бързо изменящи се във времето параметри се прави адаптивна в реално време система при която процесът на изменените на коефициентите на регулатора е част от алгоритъма за управление.

12.4.2. Промислени контролери в дискретните производства

Основните обекти за управление тук са електродвигатели, клапани, вентили. Входовете се свързват към крайни изключватели, датчици за преминаване, датчици за присъствие.

12.4.2.1. Функционална зависимост

Най-общо казано векторът на изходните променливи $\vec{O}(kT)$ в момента време k е функция на вектора на входните променливи $\vec{I}(kT)$, на вектора на вътрешните променливи $\vec{V}(kT)$ и на дискретното време k :

$$\vec{O}(kT) = f(\vec{I}(kT), \vec{O}((k-1)T), \vec{V}(kT), k).$$

✓ 12.4.2.² Основни понятия и действия

Основните понятия, посредством които се задава общия алгоритъм на управление са контурът, закъснителното устройство, вътрешна променлива (логическа или цяло число) и броячът. Логическите функции, които се изпълняват със състоянията на цифровите входове и вътрешните логически променливи са основните логически операции: И, ИЛИ, Изключващо ИЛИ, НЕ. Броячите са цели числа, отброяващи брой импулси или събития. Основните действия са увеличаване или намаляване с единица и сравнение ($>$, $<$, $=$ и т.н.).

При написването на програмата за управление на промишлените контролери освен разпространените езици за програмиране с общо предназначение (асемблер, С) намират приложение и специализираните релейно-контактни езици.

✓ 12.4.2.³ Контури

Основните елементи за изграждането на алгоритмите за управление са контурите — група от цифрови входове и изходи, свързани с отделен елемент в обекта за управление. Типичен контур за управление в дискретното производство е един пневматичен цилиндър. Той се управлява посредством два цифрови изхода. Всеки изход указва на превключващия вентил да пусне състен въздух от едната или от другата страна на буталото на пневматичния цилиндър. Положението на буталото се възприема от два датчика — за двете крайни положения. Като обект за управление изпълнява две команди — премини в едното или в другото крайно положение. Друг подобен контур е движението на работен орган посредством електродвигател. Тук също са необходими два изхода — за включване на електродвигателя да върти в едната или в другата посока. Два цифрови входа, обикновено контакти, дават информация за заемането от работния орган на едното или другото крайно положение, при което програмата на промишления контролер трябва да изключи електродвигателя.

Обобщение

Системите за управление, събиране и обработка на данни в промишлеността са предназначени да поемат непосредственото управление на машините извършващи производството, да осигурят предаването на команди към тези машини и да събират информация за протичането на производствения процес.

Специфичните особености на индустриалните системи идват от неблагоприятната среда за работа — значително ниво на електромагнитни смущения, силна замърсеност с прах, вредни изпарения, водни пари, кондензирана влага и др., силни смущения по захранващата мрежа, нестабилно захранващо напрежение в мрежата и кратковременни прекъсвания, значително по-широк интервал на изменение на температурата. Освен това към тях се поставят и редица допълнителни изисквания: значително по-висока степен на отказоустойчивост и надеждност, с тях да може да работи персонал с ниска квалификация, да са устойчиви на неправилни команди и на опити за умишлено подаване на неверни команди с цел предизвикване на спиране на производствения процес.

За да мога системите за управление, събиране и обработка на информация да работят при тези условия и да отговарят на специфичните изисквания се вземат цял комплекс от мерки: схемотехнични, конструктивни, програмни и системотехнични.

Удовлетворяването на всички тези изисквания съществено оскъпява системите за управление, събиране и обработка на информация. Затова в самите помещения на цеховете се разполага само минимално необходимите компоненти от разпределената система. Те работят при най-тежките условия и за тях се вземат всички мерки. Останалата част от компонентите на разпределената система се разполага в отделни помещения, където условията не са толкова неблагоприятни.

Системите за управление, събиране и обработка на информация от промишлени обекти се изграждат като правило с пирамидална архитектура и с въвеждането на строга йерархия на решаваните задачи и изпълняваните функции. Обикновено системата има три основни етажа.

Промислени контролери — нивото на компютърните системи за непосредствено управление на производствения процес. Те възприемат информацията директно от обекта за управление, извършват необходимите програмни действия и подават информация към изпълнителните механизми за въздействие върху производствения процес. Могат самостоятелно да подържат необходимите производствени режими, да извършват непосредственото управление на производствения процес и да натрупват статистическа информация за хода му. Всеки промишлен контролер управлява обикновено една машина или част от целия производствен процес. Притежава малък операторски пулт от който при нужда може да се задават параметри на производствения процес.

Концентратори — следващото ниво на промишлените разпределени системи за управление. Това са компютърни системи, притежаващи информационна връзка с промишлените контролери. Концентраторът периодично изтегля от промишлените контролери натрупаната информация и също извършва управление, но на по-високо ниво. Задава общите параметри на производствения процес на промишлените контролери. Има значително по-развити средства за комуникация с оператора, които му позволяват да представя повече информация и да възприема по-широк спектър от команди.

Централният компютър на системата за управление, събиране и обработка на информация за индустриални цели е свързан информационно с всички концентратори. Неговата роля в разпределената система за управление е да изтегли натрупаната в концентраторите информация, да я обработи и обобщи по подходящ начин и да спомогне оператора във вземането на решение за управлението на производството като цяло. Притежава силно развита периферия за връзка с човека — клавиатура, дисплей, печатащо устройство и др. Много често е свързан и в локалната мрежа на администрацията на цялото предприятие.

В зависимост от приложението се различават две основни групи промишлени контролери — за непрекъснато и за дискретно производ-

ство. Непрекъснатите производства са химията, металургията, био-реакторите, част от производствата в хранително-вкусовата промишленост. При тях преобладават промишлените контролери с аналогови входове и изходи. Цифровите входове и изходи имат спомагателни функции. Дискретните производства са машиностроенето, електрониката и др. При тях преобладават промишлените контролери с цифрови входове и изходи. Сравнително по-рядко се налага използването на аналогови входове и изходи.



Контролни въпроси

1. Какво е предназначението на системите за управление, събиране и обработка на данни в промишлеността?
2. Кои са специфичните условия на средата в промишлени условия?
3. Какви допълнителни изисквания се предявяват към индустриалните системи за управление?
4. Каква е общата архитектура на промишлените системи за управление?
5. Какво представлява промишления контролер?
6. По какво се различават промишлените контролери за работа в дискретното и непрекъснатите производства?