CHAPTER 14

Introduction to Digital Filters

Digital filters are used for two general purposes: (1) separation of signals that have been combined, and (2) restoration of signals that have been distorted in some way. Analog (electronic) filters can be used for these same tasks; however, digital filters can achieve far superior results. The most popular digital filters are described and compared in the next seven chapters. This introductory chapter describes the parameters you want to look for when learning about each of these filters.

Цифровые фильтры используются для двух общих целей: (1) разделение сигналов, которые были объединены, и (2) восстановление сигналов которые были искажены некоторым способом. Аналоговые (электронные) фильтры могут использоваться для тех же самых задач; однако, цифровые фильтры могут достигать далеко превосходящих результатов. Наиболее популярные цифровые фильтры описаны и сравнены в следующих семи главах. Эта вводная глава описывает параметры, которые Вы хотите искать при изучении относительно каждого из этих фильтров.

Filter Basics
Основы Фильтра

Digital filters are a very important part of DSP. In fact, their extraordinary performance is one of the key reasons that DSP has become so popular. As mentioned in the introduction, filters have two uses: signal *separation* and signal *restoration*. Signal separation is needed when a signal has been contaminated with interference, noise, or other signals. For example, imagine a device for measuring the electrical activity of a baby's heart (EKG) while still in the womb. The raw signal will likely be corrupted by the breathing and heartbeat of the mother. A filter might be used to separate these signals so that they can be individually analyzed. Signal restoration is used when a signal has been distorted in some way. For example, an audio recording made with poor equipment may be filtered to better represent the sound as it actually occurred. Another example is the deblurring of an image acquired with an improperly focused lens, or a shaky camera.

Цифровые фильтры - очень важная часть ЦОС. Фактически, их экстраординарная эффективность - одно ключевых заключений доводов разума, что ЦОС стал настолько популярным. Как упомянуто во введении, фильтры имеют два использования: *разделение* сигнала и *восстановление* сигнала. Разделение Сигнала необходимо, когда сигнал был загрязнен интерференцией, шумом, или другими сигналами. Например, вообразите устройство для измерения электрического действия сердца младенца (ЭКГ) в то время как он еще в матке. Необработанный сигнал будет вероятно искажен дыханием и биением сердца матери. Фильтр мог бы использоваться, чтобы отделить эти сигналы так, чтобы они могли быть индивидуально проанализированы. Восстановление сигнала используется, когда сигнал был искажен некоторым способом. Например, звуковая регистрация, сделанная с плохим оборудованием может быть фильтрована, чтобы лучше представить звук, поскольку это фактически произошло. Другой пример – устранение размытости изображения, приобретенного с ненадлежащим образом сфокусированной линзой, или шаткой камерой.

These problems can be attacked with either analog or digital filters. Which is better? Analog filters are cheap, fast, and have a large dynamic range in both amplitude and frequency. Digital filters, in comparison, are vastly superior in the level of performance that can be achieved. For example, a low-pass digital filter presented in Chapter 16 has a gain of 1 +/- 0.0002 from DC to 1000 hertz, and a gain of less than 0.0002 for frequencies above 1001 hertz. The entire transition occurs within only 1 hertz. Don't expect this from an op amp circuit! Digital filters can achieve *thousands* of times better performance than analog filters. This makes a dramatic difference in how filtering problems are approached. With analog filters, the emphasis is on handling limitations of the electronics, such as the accuracy and stability of the resistors and capacitors. In comparison, digital filters are so good that the performance of the filter is frequently ignored. The emphasis shifts to the limitations of the *signals*, and the *theoretical* issues regarding their processing.

Эти проблемы могут быть атакованы или с аналоговыми или с цифровыми фильтрами. Который является лучше? Аналоговые фильтры дешевы, быстры, и имеют большой динамический диапазон и по амплитуде и по частоте. Цифровые фильтры, для сравнения, являются значительно выше по уровню эффективности, которая может быть достигнута. Для примера, цифровой низкочастотный фильтр, представленный в главе 16 имеет увеличение(коэффициент усиления) 1 + /- 0.0002(единицу) от постоянного тока до 1000 герц, и увеличение(усиление меньше чем 0.0002 для частот более чем 1001 герц. Полная передача происходит в пределах только 1 герца. Не ожидайте это от op amp(операционных усилителей) схем! Цифровые фильтры могут достигать эффективности в *тысячи* раз лучшей, чем аналоговые фильтры. Это делает драматическую разность в том, как проблемы фильтрации достигаются. С аналоговыми фильтрами, акцент находится при обработке ограничений электроники, типа точности и стабильности резисторов и конденсаторов. Для сравнения, цифровые фильтры настолько хороши, что эффективность фильтра часто игнорируется. Акцент сдвигается к ограничениям *сигналов*, и *теории* (*теоретических проблем*) относительно их обработки.

It is common in DSP to say that a filter's input and output signals are in the *time domain*. This is because signals are usually created by sampling at regular intervals of *time*. But this is not the only way sampling can take place. The second most common way of sampling is at equal intervals in *space*. For example, imagine taking simultaneous readings from an array of strain sensors mounted at one centimeter increments along the length of an aircraft wing. Many other domains are possible; however, time and space are by far the most common. When you see the term *time domain* in DSP, remember that it may actually refer to samples taken over time, or it may be a general reference to any domain that the samples are taken in.

Это обычно в ЦОС, чтобы говорить, что сигналы ввода и вывода фильтра находятся в домене времени. Это - то, потому что сигналы обычно создаются, производя выбор равномерно времени. Но это - не единственный путь производить выбор может иметь место. второй наиболее обычный путь осуществления выборки - в равных интервалах в пространстве. Например, вообразите брать одновременные чтения от массива датчиков напряжения, установленных в одних приращениях сантиметра по длине крыла самолета. Много других доменов возможны; однако, время и пространство(расстояние) намного наиболее обычны. Когда Вы видите термин домен времени в ЦОС, помните, что это может фактически относиться к выборкам, требующим течения времени, или это может быть общая ссылка на любой домен, что выборки приняты.

As shown in Fig. 14-1, every linear filter has an **impulse response**, a **step response** and a **frequency response.** Each of these responses contains complete information about the filter, but in

a different form. If one of the three is specified, the other two are fixed and can be directly calculated. All three of these representations are important, because they describe how the filter will react under different circumstances.

Как показано в рис. 14-1, каждый линейный фильтр имеет **импульсную передаточную функцию**, **реакцию на скачок(ступеньку)** и **частотную характеристику**. Каждый из этих ответов содержит полную информацию относительно фильтра, но в различной форме. Если один из этих трех определен, другой два установлены(фиксированы, определены, постоянны) и могут быть непосредственно рассчитаны. Все три из этих представлений важны, потому что они описывают, как фильтр реагирует при различных обстоятельствах.

The most straightforward way to implement a digital filter is by *convolving* the input signal with the digital filter's *impulse response*. All possible linear filters can be made in this manner. (This should be obvious. If it isn't, you probably don't have the background to understand this section on filter design. Try reviewing the previous section on DSP fundamentals). When the *impulse response* is used in this way, filter designers give it a special name: the **filter kernel**.

Наиболее прямой способ осуществлять цифровой фильтр - *скручивая* входной сигнал с *импульсной передаточной функцией* цифрового фильтра. Все возможные линейные фильтры могут быть сделаны этим способом. (Это должно быть очевидно. Это если Вы, вероятно, не имеете подготовки, чтобы понять этот раздел по проекту фильтра. Пробуйте рассмотреть предыдущий раздел по основным принципам ЦОС). Когда импульсная передаточная функция используется таким образом, проектировщики фильтра дают этому специальное название **ядро фильтра**.

There is also another way to make digital filters, called **recursion**. When a filter is implemented by convolution, each sample in the output is calculated by *weighting* the samples in the input, and adding them together. Recursive filters are an extension of this, using previously calculated values from the *output*, besides points from the *input*. Instead of using a filter kernel, recursive filters are defined by a set of **recursion coefficients**. This method will be discussed in detail in Chapter 19. For now, the important point is that all linear filters have an impulse response, even if you don't use it to implement the filter. To find the impulse response of a recursive filter, simply feed in an impulse, and see what comes out. The impulse responses of recursive filters are composed of sinusoids that exponentially decay in amplitude. In principle, this makes their impulse responses *infinitely long*. However, the amplitude eventually drops below the round-off noise of the system, and the remaining samples can be ignored. Because of this characteristic, recursive filters are also called **Infinite Impulse Response or IIR** filters. In comparison, filters carried out by convolution are called **Finite Impulse Response or FIR** filters.

Имеется также другой способ делать цифровые фильтры, называемые **рекурсией**. Когда фильтр осуществлен скручиванием, каждая выборка в выходе рассчитана *взвешиванием* выборки во вводе, и сложения их вместе. Рекурсивные фильтры - расширение(продление) этого, используя предварительно расчетные значения от *выхода*, помимо точек от *ввода*. Вместо использования ядра фильтра, рекурсивные фильтры определены набором **коэффициентов рекурсии**. Этот метод будет обсужден подробно в главе 19. Пока, важный пункт - то, что все линейные фильтры имеют импульсную передаточную функцию, даже если Вы не используете это, чтобы осуществить фильтр. Находить импульсную передаточную функцию рекурсивного фильтра, просто подают импульс, и смотрят то, что выйдет. Импульсные передаточные функции рекурсивных фильтров составлены из синусоид, которые по экспоненте распадаются в амплитуде. В принципе, это делает их импульсные передаточные функции *бесконечно длинными*. Однако, амплитуда в конечном счете понижается ниже шума округления системы, и остающиеся выборки могут игнорироваться.

Из-за этой характеристики, рекурсивные фильтры также называются **Бесконечной Импульсной Передаточной Функцией** или **БИХ-фильтры(фильтры с бесконечной импульсной характеристикой**). Для сравнения, фильтры, выполненные скручиванием называются **Фильтрами с Конечной Импульсной передаточной функцией** или **КИХ-фильтрами(фильтрами с конечной импульсной характеристикой)**.

As you know, the *impulse response* is the output of a system when the input is an *impulse*. In this same manner, the *step response* is the output when the input is a *step* (also called an *edge*, and an *edge response*). Since the step is the integral of the impulse, the step response is the integral of the impulse response. This provides two ways to find the step response: (1) feed a step waveform into the filter and see what comes out, or (2) integrate the impulse response. (To be mathematically correct: *integration* is used with continuous signals, while *discrete integration*, i.e., a running sum, is used with discrete signals). The frequency response can be found by taking the DFT (using the FFT algorithm) of the impulse response. This will be reviewed later in this chapter. The frequency response can be plotted on a linear vertical axis, such as in (c), or on a logarithmic scale (decibels), as shown in (d). The linear scale is best at showing the passband ripple and roll-off, while the decibel scale is needed to show the stopband attenuation.

Как Вы знаете, *отклик на импульс*(*импульсная передаточная функция*) - выход системы, когда ввод - *импульс*. В этой же самой манере, *реакция(отклик) на скачок(ступеньку)* - выход, когда ввод – *ступенька* (также называемый *краем*, и *ответом края.* Так как ступенька(шаг) - интеграл импульса, реакция(отклик) на скачок - интеграл импульсной передаточной функции. Это обеспечивает два способа найти реакцию на скачок: (1) подают форму волны ступеньки в фильтр, и смотрят на то, что будет на выходе, или (2) интегрируют импульсную передаточную функцию. (Чтобы быть математически точным: интегрирование используется с непрерывными сигналами, в то время как дискретное интегрирование, то есть, выполняющая сумма, используется с дискретными сигналами). Частотная характеристика может быть найдена, беря ДПФ (используя алгоритм БПФ) импульсной передаточной функции. Это будет рассмотрено позже в этой главе. График частотной характеристики можно составлять на линейной вертикальной оси, типа в (c), или в логарифмическом масштабе (децибелы), как показано в (d). Линейный масштаб лучший при показе пульсации(неравномерности) полосы пропускания и спада(завала), в то время как масштаб в децибелах необходим, чтобы показать полосу затухания((полосу задерживания) ослабления.

Don't remember decibels? Here is a quick review. A **bel** (in honor of Alexander Graham Bell) means that the power is changed by a *factor of ten*. For example, an electronic circuit that has 3 bels of amplification produces an output signal with 10x10x10 = 1000 times the power of the input. A **decibel (dB)** is one-tenth of a bel. Therefore, the decibel values of: -20dB, -10dB, 0dB, 10dB & 20dB, mean the power ratios: 0.01, 0.1, 1, 10, & 100, respectively. In other words, every *ten* decibels mean that the power has changed by a factor of ten.

Не помните что такое децибелы? Имеется быстрый обзор. **Бел** (в честь Александра Грэма, Бэлла) означает что мощность изменена *коэффициентом(фактором) десять*. Например, электронная схема, которая имеет усиление 3 бела, производит сигнал выхода в 10x10x10 = 1000 раз больше мощности ввода. **Децибел** (**dB**) - десятая часть бела. Поэтому, значения децибела: -20dB, -10dB, 0dB, 10dB и 20dB, означают отношения(коэффициенты) мощностей: 0.01, 0.1, 1, 10, и 100, соответственно. Другими словами, каждые десять децибелов подразумевают, что мощь изменилась коэффициентом десять(в десять раз).

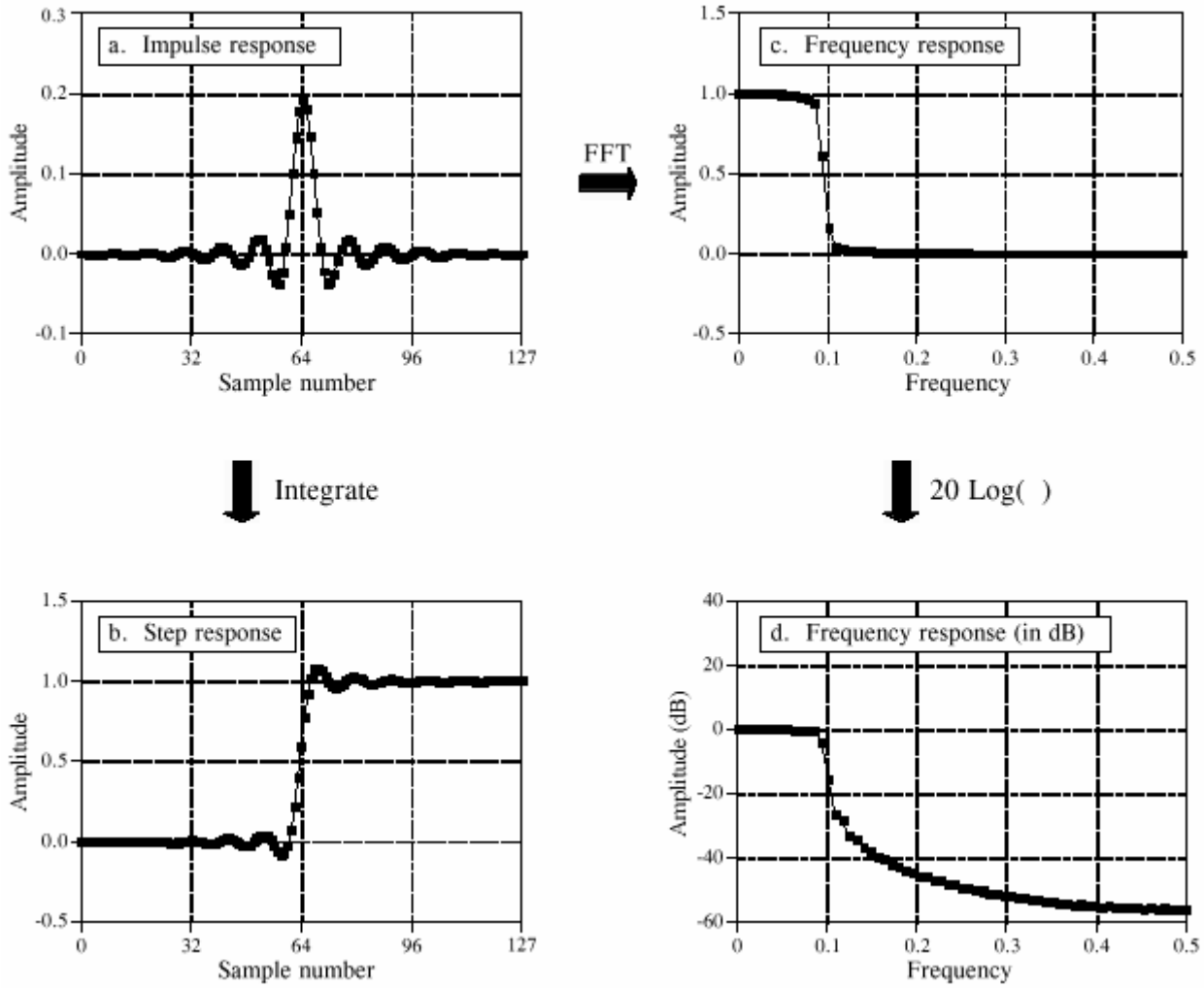

FIGURE 14-1
Filter parameters. Every linear filter has an impulse response, a step response, and a frequency response. The step response, (b), can be found by discrete integration of the impulse response, (a). The frequency response can be found from the impulse response by using the Fast Fourier Transform (FFT), and can be displayed either on a linear scale, (c), or in decibels, (d).

РИСУНОК 14-1. Параметры Фильтра.
Каждый линейный фильтр имеет импульсную передаточную функцию, реакцию на скачок, и частотную характеристику. Реакция на скачок, (b), может быть найдена дискретным интегрированием импульсной передаточной функции, (a). Частотная характеристика может быть найдена от импульсной передаточной функции, используя Быстрое преобразование Фурье (БПФ), и может быть отображена или в линейном масштабе, (c), или в децибелах, (d).

Here's the catch: you usually want to work with a signal's *amplitude*, not its *power*. For example, imagine an amplifier with 20dB of gain. By definition, this means that the power in the signal has increased by a factor of 100. Since amplitude is proportional to the square-root of power, the amplitude of the output is 10 times the amplitude of the input. While 20dB means a factor of 100 in power, it only means a factor of 10 in amplitude. Every *twenty* decibels mean that the amplitude has changed by a factor of ten. In equation form:

Имеется арретир(за что зацепиться): Вы обычно хотите работать с *амплитудой* сигнала, не с его *мощностью*. Например, вообразите усилитель с усилением 20dB. По определению, это означает, что мощность сигнала увеличилась на коэффициент 100. Так как амплитуда пропорциональна к квадратному корню из мощности, амплитуда выхода – в 10 раз больше

амплитуды ввода. В то время как 20dB в мощности означает коэффициент(фактор) 100, в амплитуде это означает коэффициент(фактор) только 10. Каждые двадцать децибелов подразумевают, что амплитуда изменилась коэффициентом(фактором) десять. В форме уравнения:

EQUATION 14-1. Definition of decibels.
Decibels are a way of expressing a *ratio* between two signals. Ratios of power (P1 & P2) use a different equation from ratios of amplitude (A1 & A2).

$$\mathrm{dB} = 10 \log_{10} \frac{P_2}{P_1}$$

14-1 УРАВНЕНИЕ. Определение децибелов.
Децибелы - путь выражения *отношения* между двумя сигналами. Отношения(коэффициенты) мощностей (P1 и P2) используют различные уравнения от отношений амплитуды (A1 и A2).

$$\mathrm{dB} = 20 \log_{10} \frac{A_2}{A_1}$$

The above equations use the base 10 logarithm; however, many computer languages only provide a function for the base *e* logarithm (the natural log, written $\log_e x$ or $\ln x$). The natural log can be use by modifying the above equations: dB = 4.342945 log *e* (P_2 /P_1) and dB = 8.685890 log *e* (A_2 /A_1).

Вышеупомянутые уравнения используют логарифм с основанием 10; однако, много машинных языков обеспечивают функцию только для логарифма с основанием *e* (натуральный логарифм, записываемый $\log_e x$ или $\ln x$). Натуральный логарифм может быть использован, изменяя вышеупомянутое уравнение: dB = 4.342945 log *e* (P_2 /P_1) и dB = 8.685890 log *e* (A_2 / A_1).

Since decibels are a way of expressing the ratio between two signals, they are ideal for describing the gain of a system, i.e., the ratio between the output and the input signal. However, engineers also use decibels to specify the amplitude (or power) of a *single* signal, by referencing it to some standard. For example, the term: **dBV** means that the signal is being referenced to a 1 volt rms signal. Likewise, **dBm** indicates a reference signal producing 1 mW into a 600 ohms load (about 0.78 volts rms).

Так как децибелы - способ выражения отношения между двумя сигналами, они идеальны для описания усиления системы, то есть, отношения между выходным и входным сигналом. Однако, инженеры также используют децибелы, чтобы определить амплитуду (или мощность) отдельного сигнала, ссылаясь при этом на некоторый стандарт. Например, термин: **dBV** означает, что говорится о сигнале со среднеквадратичным значением один вольт. Аналогично, **dBm** указывает что сигнал, выделяет на нагрузке 600 ом мощность 1 mW(миливатт) (приблизительно 0.78 среднеквадратичного значения в вольтах).

If you understand nothing else about decibels, remember two things: First, -3dB means that the amplitude is reduced to 0.707 (and the power is 60dB = 1000 therefore reduced to 0.5). Second, memorize the following conversions between decibels and *amplitude* ratios:

Если Вы не совсем понимаете что-нибудь относительно децибелов, помните две вещи: Во первых, -3dB означает, что амплитуда ослаблена в 0.707 раза (и мощность 60dB = 1000 поэтому приведена(слаблена к 0.5). Во вторых, запомните следующие преобразования между децибелами и амплитудными отношениями:

40dB =	100
20dB =	10
0dB =	1
-20dB =	0.1

(c) АВТЭКС, Санкт-Петербург, http://www.autex.spb.ru, e-mail: info@autex.spb.ru

-40dB = 0.01
-60dB = 0.001

How Information is Represented in Signals
Как Информация представлена в Сигналах

The most important part of any DSP task is understanding how *information* is contained in the signals you are working with. There are many ways that information can be contained in a signal. This is especially true if the signal is manmade. For instance, consider all of the modulation schemes that have been devised: AM, FM, single-sideband, pulse-code modulation, pulse-width modulation, etc. The list goes on and on. Fortunately, there are only two ways that are common for information to be represented in naturally occurring signals. We will call these: **information represented in the time domain**, and **information represented in the frequency domain**.

Наиболее важная часть любой задачи ЦОС понимать, как *информация* содержится в сигналах, с которыми Вы работаете. Имеются много способов, которыми информация может содержаться в сигнале. Это - особенно истина, если сигнал искусственный. Например, рассмотрите все модуляционные схемы, которые были изобретены: АМ, ЧМ, кодово-импульсная модуляция с единственной боковой полоса, модуляция ширины импульса, и т.д. Список продолжается и продолжается. К счастью, имеются только два пути, которые являются обычными для информации, которая будет представлена в естественно появляющихся сигналах. Мы назовем их: информация, представленная в домене времени, и информация, представленная в частотном домене.

Information represented in the time domain describes when something occurs and what the amplitude of the occurrence is. For example, imagine an experiment to study the light output from the sun. The light output is measured and recorded once each second. Each sample in the signal indicates what is happening at that instant, and the level of the event. If a solar flare occurs, the signal directly provides information on the time it occurred, the duration, the development over time, etc. Each sample contains information that is interpretable without reference to any other sample. Even if you have only one sample from this signal, you still know something about what you are measuring. This is the simplest way for information to be contained in a signal.

Информация, представленная в домене времени описывает, когда кое-что происходит и, какова амплитуда местонахождения. Например, вообразите, что эксперимент изучает световой выход(светоотдачу) от солнца. Световой выход измерен и зарегистрирован один раз в каждую секунду. Каждая выборка в сигнале указывает то, что случается в тот момент, и уровень случая. Если солнечная вспышка происходит, сигнал непосредственно обеспечивает информацию относительно времени, когда это произошло, продолжительность, развитие через какое-то время, и т.д. Каждая выборка содержит информацию, которая является поддающейся толкованию, независимо от любой другой выборки. Даже если Вы имеете только одну выборку от этого сигнала, Вы все еще знаете кое-что относительно того, что Вы измеряете. Это - самый простой путь для информации, которую нужно содержать в сигнале.

In contrast, information represented in the frequency domain is more indirect. Many things in our universe show periodic motion. For example, a wine glass struck with a fingernail will vibrate, producing a ringing sound; the pendulum of a grandfather clock swings back and forth; stars and planets rotate on their axis and revolve around each other, and so forth. By measuring the frequency, phase, and amplitude of this periodic motion, information can often be obtained about the system producing the motion. Suppose we sample the sound produced by the ringing

wine glass. The fundamental frequency and harmonics of the periodic vibration relate to the mass and elasticity of the material. A single sample, in itself, contains no information about the periodic motion, and therefore no information about the wine glass. The information is contained in the *relationship* between many points in the signal.

Напротив, информация, представленная в частотном домене более косвенная. Много вещей в нашей области показывают периодическое движение. Например, бокал, нажатый с ногтем будет вибрировать, производя звук окружения; маятник дедушки синхронизирует колебание назад и вперед; звезды и планеты вращаются на их оси и вращаются вокруг друг друга, и т.д. Измеряя частоту, фазу, и амплитуду этого периодического движения, информация может часто получаться относительно системы, производящей движение. Предположим, что мы производим выборку звука, произведенного бокалом окружения. Фундаментальная частота и гармоники периодической вибрации касаются массы и адаптационной способности материала. Отдельная выборка, сама по себе, не содержит никакой информации относительно периодического движения, и поэтому никакой информации относительно бокала. Информация содержится в *отношениях* между многими точками в сигнале.

This brings us to the importance of the step and frequency responses. The *stepresponse* describes how information represented in the *time domain* is being modified by the system. In contrast, the *frequency response* shows how information represented in the *frequency domain* is being changed. This distinction is absolutely critical in filter design because it is not possible to optimize a filter for both applications. Good performance in the time domain results in poor performance in the frequency domain, and vice versa. If you are designing a filter to remove noise from an EKG signal (information represented in the time domain), the step response is the important parameter, and the frequency response is of little concern. If your task is to design a digital filter for a hearing aid (with the information in the frequency domain), the frequency response is all important, while the step response doesn't matter. Now let's look at what makes a filter optimal for time domain or frequency domain applications.

Это приносит нам к важности шага и частотных характеристик. *Реакция(отклик) на скачок* описывает, как информация, представленная в *домене времени* изменяется системой. Напротив, *частотная характеристика* показывает, как изменяется информация, представленная в *частотном домене*. Это различие абсолютно критическое в проекте фильтра, потому что не возможно оптимизировать фильтр для обоих приложений. Хорошая эффективность в домене времени приводит к плохой эффективности в частотном домене, и наоборот. Если Вы разрабатываете фильтр, чтобы удалить шум из сигнала ЭКГ (информация, представленная в домене времени), реакция на скачок - важный параметр, и частотная характеристика имеет небольшое беспокойство. Если ваша задача состоит в том, чтобы проектировать цифровой фильтр для слухового аппарата (с информацией в частотном домене), частотная характеристика весь важна, в то время как реакция на скачок не имеет значение. Теперь давайте смотреть то, что делает фильтр оптимальным для приложений в домене времени или частотном домене.

Time Domain Parameters
Параметры Домена Времени

It may not be obvious why the step response is of such concern in time domain filters. You may be wondering why the impulse response isn't the important parameter. The answer lies in the way that the human mind understands and processes information. Remember that the step, im-

pulse and frequency responses all contain identical information, just in different arrangements. The step response is useful in time domain analysis because it matches the way humans view the information contained in the signals.

Не может быть очевидно, почему реакция на скачок имеет такое беспокойство в фильтрах домена времени. Вы можете задаваться вопросом, почему импульсная передаточная функция не важный параметр. Ложь ответа в пути, которым человеческое мнение понимает и обрабатывает информацию. Помните, что шаг(ступенька), импульс и частотные характеристики все содержат идентичную информацию, только в различных размещениях. Реакция на скачок полезна в анализе домена времени, потому что это соответствует пути, которым люди рассматривают информацию, содержащуюся в сигналах.

For example, suppose you are given a signal of some unknown origin and asked to analyze it. The first thing you will do is divide the signal into regions of similar characteristics. You can't stop from doing this; your mind will do it automatically. Some of the regions may be smooth; others may have large amplitude peaks; others may be noisy. This segmentation is accomplished by identifying the points that separate the regions. This is where the step function comes in. The step function is the purest way of representing a division between two dissimilar regions. It can mark when an event starts, or when an event ends. It tells you that whatever is on the *left* is somehow different from whatever is on the *right*. This is how the human mind views time domain information: a group of step functions dividing the information into regions of similar characteristics. The step response, in turn, is important because it describes how the dividing lines are being modified by the filter.

Например, предположите, Вам дают сигнал некоторого неизвестного происхождения и просят его анализировать. Первая вещь, которую Вы будете делать - делить сигнал на области с подобными характеристиками. Вы не можете не выполнять этого; ваше сознание будет делать это автоматически. Некоторые из областей могут быть гладки; другие могут иметь большие амплитудные пики; другие могут быть шумные. Эта сегментация выполняется, идентифицируя точки, которые отделяют области. Это - то, где ступенчатая функция входит. Ступенчатая функция - самый чистый путь представления раздела между двумя несходными областями. Это может отмечать, когда начинается случай(событие, или когда случай(событие) заканчивается. Это сообщает Вам, что безотносительно является, *слева* так или иначе отличается того, что - *справа*. Это - то, как человеческое мнение рассматривает информацию домена времени: группа ступенчатых функций, разделяющих информацию в области подобных характеристик. Реакция на скачок, в свою очередь, является важной, потому что это описывает, как разделительные линии изменяются фильтром.

The step response parameters that are important in filter design are shown in Fig. 14-2. To distinguish events in a signal, the duration of the step response must be shorter than the spacing of the events. This dictates that the step response should be as *fast* (the DSP jargon) as possible. This is shown in Figs. (a) & (b). The most common way to specify the **risetime** (more jargon) is to quote the number of samples between the 10% and 90% amplitude levels. Why isn't a very fast risetime always possible? There are many reasons, noise reduction, inherent limitations of the data acquisition system, avoiding aliasing, etc.

Параметры реакции на скачок, которые являются важными в проекте фильтра, показываются в рис. 14-2. Чтобы различать(отличать) события в сигнале, продолжительность реакции на скачок должна быть короче чем интервал событий. Это диктует, что реакция на скачок должна быть так *быстра* (жаргон ЦОС) насколько возможно. Это показано в сис. (a) и (b). Наиболее обычный способ определить **переходное быстродействие (время на-**

растания; время установления; длительность время нарастания импульса; длительность фронта импульса) (большее количество жаргона) состоит в том, чтобы цитировать число выборок между 10% и 90% уровнями амплитуд. Почему не очень быстрое время нарастания, всегда возможно? Имеются много причин, шумовое приведение, свойственные ограничения системы сбора данных, избежание наложения спектров, и т.д.

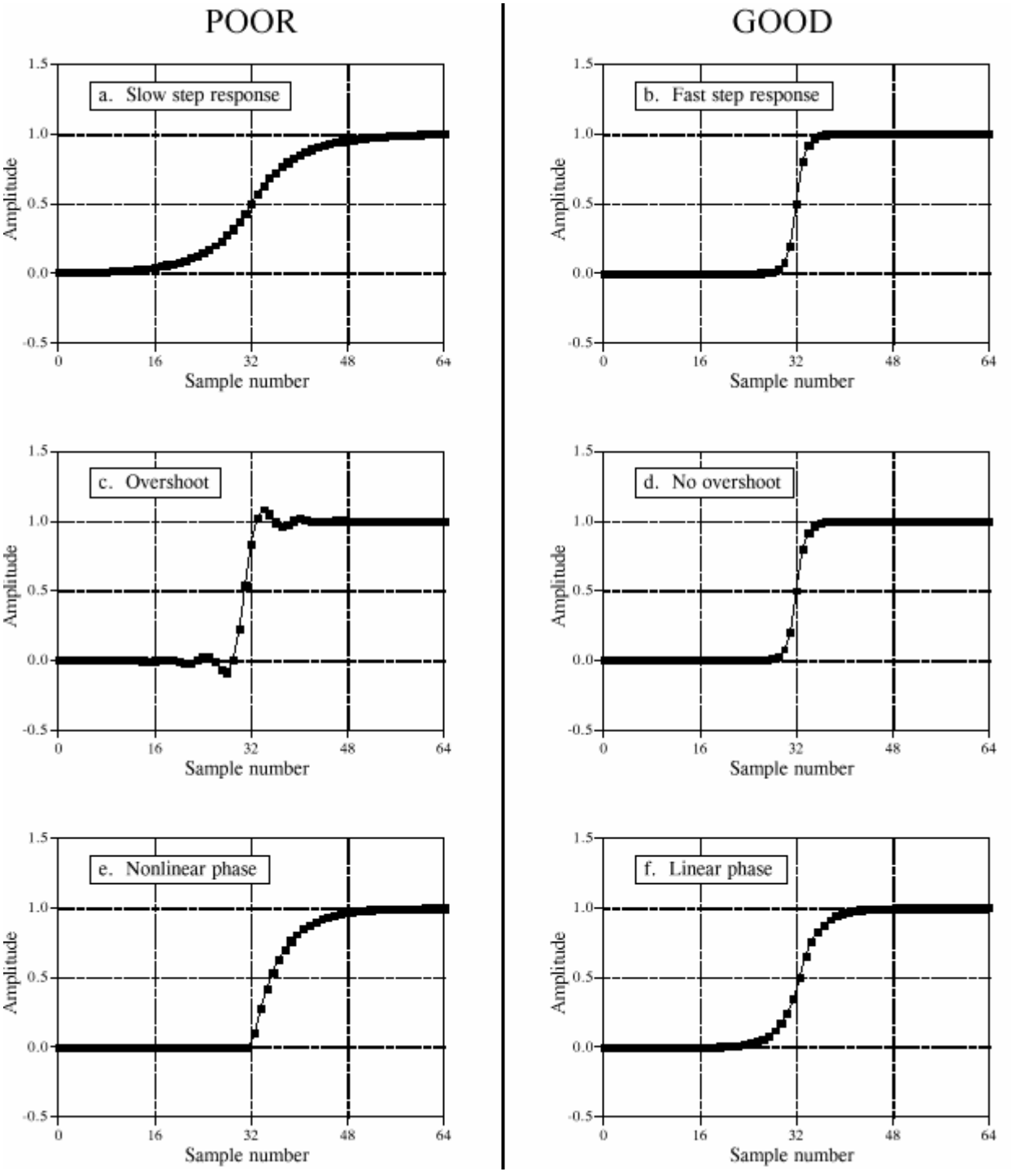

FIGURE 14-2. Parameters for evaluating *time domain* performance.
The step response is used to measure how well a filter performs in the time domain. Three parameters are important: (1) transition speed (risetime), shown in (a) and (b), (2) overshoot, shown in (c) and (d), and (3) phase linearity (symmetry between the top and bottom halves of the step), shown in (e) and (f).

(РИСУНОК 14-2. Параметры для оценки эффективности домена времени.

(c) АВТЭКС, Санкт-Петербург, http://www.autex.spb.ru, e-mail: info@autex.spb.ru

Реакция на скачок используется, чтобы иметь параметры, как хорошо фильтр исполняет в время домен. Три параметра важны: (1) переходное быстродействие (время нарастания), показанное в (a) и (b), (2) перерегулирование, показанное в (c) и (d), и (3) фазовая линейность (симметрия между верхними и нижними половинами ступеньки), показанной в (e) и (f).

Figures (c) and (d) shows the next parameter that is important: **overshoot** in the step response. Overshoot must generally be eliminated because it changes the amplitude of samples in the signal; this is a basic distortion of the information contained in the time domain. This can be summed up in one question: Is the overshoot you observe in a signal coming from the thing you are trying to measure, or from the filter you have used?

Рисунки (c) и (d) показывают следующий параметр, который является важным: **Перерегулирование** в реакции на скачок. Перерегулирование должно вообще устраняться, потому что это изменяет амплитуду выборок в сигнале; это - основное искажение информации, содержащейся в домене времени. Это может быть суммировано в одном вопросе: Является ли перерегулирование, которое Вы наблюдаете в сигнале, исходящим из вещи, которую Вы пробуете измерять, или от фильтра, который Вы использовали?

Finally, it is often desired that the upper half of the step response be symmetrical with the lower half, as illustrated in (e) and (f). This symmetry is needed to make the *rising edges* look the same as the *falling edges*. This symmetry is called **linear phase**, because the frequency response has a phase that is a straight line (discussed in Chapter 19). Make sure you understand these three parameters; they are the key to evaluating time domain filters.

Наконец, это часто желательно, чтобы верхняя половина реакции на скачок была симметрическая с более низкой половиной, как иллюстрировано в (e) и (f). Эта симметрия необходима, чтобы делать вид *нарастания граней*(нарастающий фронт) тем же самым как *падающие грани*. Эта симметрия называется **линейной фазой**, потому что частотная характеристика имеет фазу, которая является прямой линией (обсужденная в главе 19). Удостоверитесь, что Вы понимаете эти три параметра; они - ключ к оценке фильтров домена времени.

Frequency Domain Parameters
Параметры Частотного Домена

Figure 14-3 shows the four basic frequency responses. The purpose of these filters is to allow some frequencies to pass unaltered, while completely blocking other frequencies. The **passband** refers to those frequencies that are passed, while the **stopband** contains those frequencies that are blocked. The **transition band** is between. A **fast roll-off** means that the transition band is very narrow. The division between the passband and transition band is called the **cutoff frequency.** In analog filter design, the cutoff frequency is usually defined to be where the amplitude is reduced to 0.707 (i.e., -3dB). Digital filters are less standardized, and it is common to see 99%, 90%, 70.7%, and 50% amplitude levels defined to be the cutoff frequency.

Рисунок 14-3 показывает четыре основных частотных характеристики. Цель этих фильтров состоит в том, чтобы позволить некоторым частотам проходить неизменными, при полном блокировании других частот. **Полоса пропускания** относится к тем частотам, которые пропускают, в то время как **полоса задерживания**(полоса затухания, полоса ослабления) содержит те частоты, которые блокированы. Переходная полоса - между. Быстрый завал(спад) означает, что переходная полоса очень узкая. Раздел(деление) между полосой пропускания и переходной полосой называется частотой останова. В аналоговом проекте фильтра, **частота останова**(**граничная частота**; **частота отсечки**) обычно определяется,

чтобы быть, где амплитуда сокращена к 0.707 (то есть, -3dB). Цифровые фильтры меньше стандартизированы, и общее(обычно) видеть 99 %, 90 %, 70.7 %, и 50% уровней амплитуд, определенные, чтобы быть частотой останова(граничной частотой).

Figure 14-4 shows three parameters that measure how well a filter performs in the frequency domain. To separate closely spaced frequencies, the filter must have a **fast roll-off**, as illustrated in (a) and (b). For the passband frequencies to move through the filter unaltered, there must be no **passband ripple**, as shown in (c) and (d). Lastly, to adequately block the stopband frequencies, it is necessary to have good **stopband attenuation**, displayed in (e) and (f).

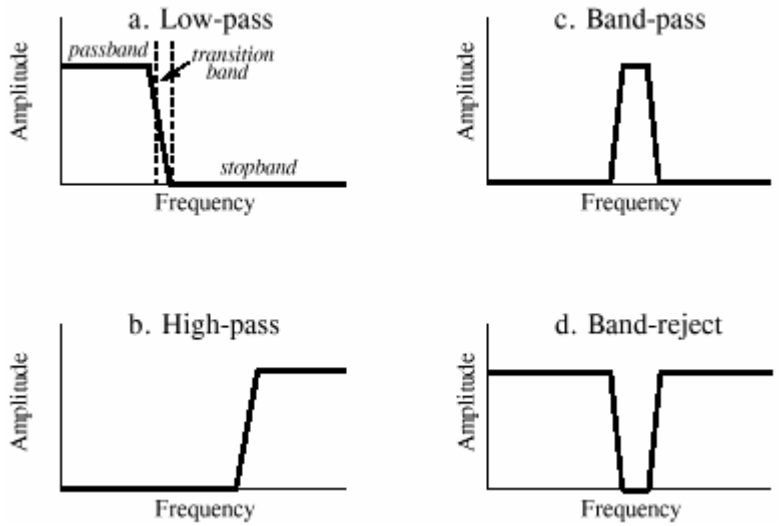

FIGURE 14-3
The four common frequency responses. Frequency domain filters are generally used to pass certain frequencies (the *passband*), while blocking others (the *stopband*). Four responses are the most common: low-pass, high-pass, band-pass, and band-reject.

ЧИСЛО(РИСУНОК) 14-3
Четыре общих(обычных) частотных характеристики. Фильтры частотного домена вообще используются, чтобы передать некоторые частоты (полоса пропускания), при блокировании других (Полоса задерживания(полоса затухания, полоса ослабления)). Четыре ответа наиболее общие(обычны): фильтр низких частот, фильтр верхних частот, полосовой, и полосовой – заграждающий(режекторный) фильтр.

Рисунок 14-4 показывает три параметра, которые имеют размеры как хорошо фильтр исполняет в частотном домене. Чтобы отделить близко расположенные частоты, фильтр должен иметь **быстрый завал(спад)**, как иллюстрировано в (a) и (b). Для частот полосы пропускания, чтобы двигаться через неизменный фильтр, не должно иметься никакой **неравномерности полосы пропускания**(неравномерности пульсаций; ряби), как показано в (c) и (d). Наконец, чтобы соответственно блокировать частоты полосы задерживания(полосы затухания; полосы ослабления), необходимо иметь хорошее ослабление полосы задерживания(полосы затухания; полосы ослабления), отображенное в (e) и (f).

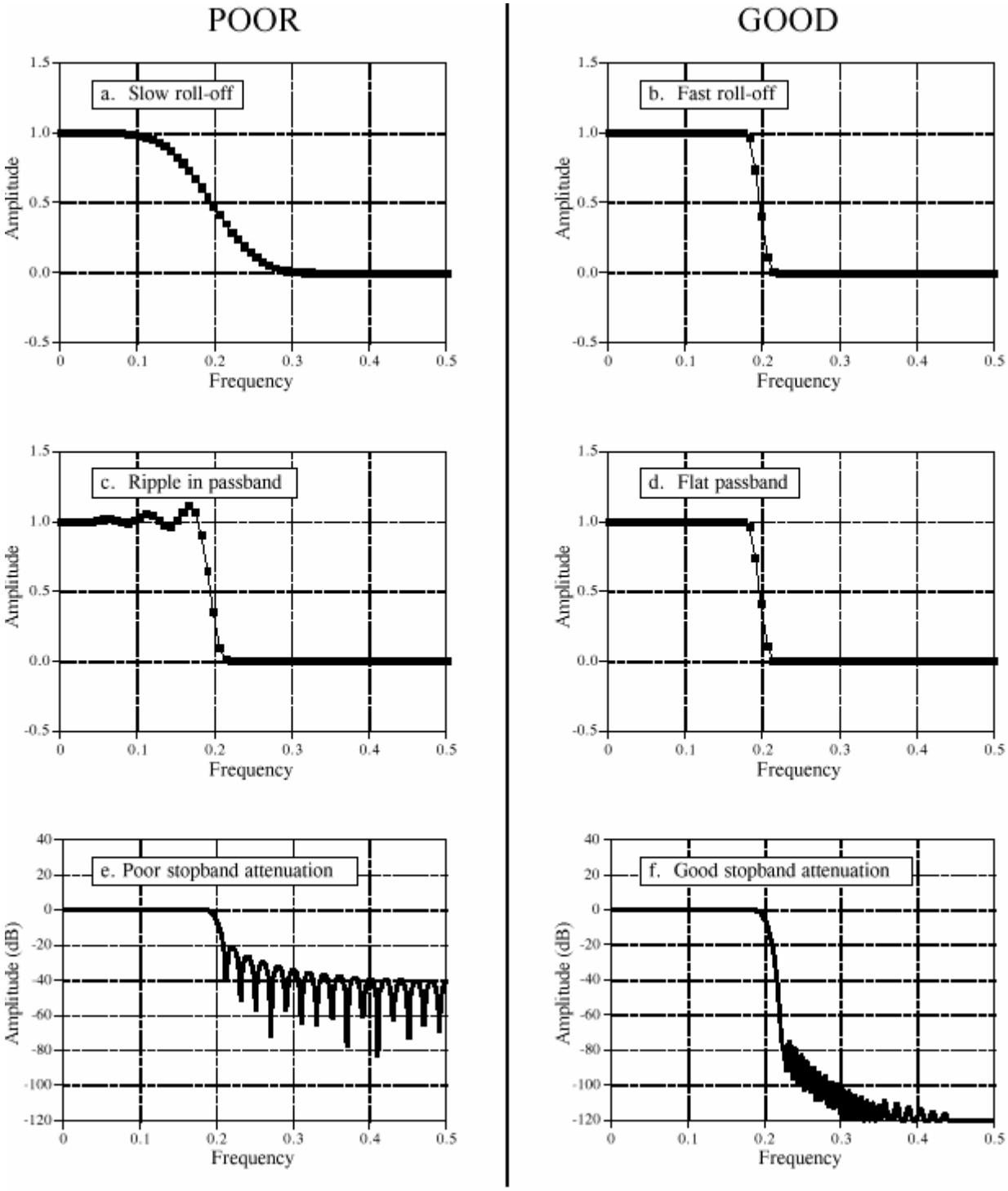

FIGURE 14-4 . Parameters for evaluating *frequency domain* performance.
The frequency responses shown are for low-pass filters. Three parameters are important: (1) roll-off sharpness, hown in (a) and (b), (2) passband ripple, shown in (c) and (d), and (3) stopband attenuation, shown in (e) and (f).

РИСУНОК 14-4. Параметры для оценки эффективности частотного домена.
Показанные частотные характеристики - для фильтров нижних частот. Три параметра важны: (1) резкость завала(спада), показанная в (a) и (b), (2) неравномерности в полосе пропускания, показанных в (c) и (d), и (3) ослаблении полосы задерживания(полосы затухания; полосы ослабления), показанных в (e) и (f).

(c) АВТЭКС, Санкт-Петербург, http://www.autex.spb.ru, e-mail: info@autex.spb.ru

Why is there nothing about the *phase* in these parameters? First, the phase isn't important in most frequency domain applications. For example, the phase of an audio signal is almost completely random, and contains little useful information. Second, if the phase is important, it is very easy to make digital filters with a *perfect* phase response, i.e., all frequencies pass through the filter with a zero phase shift (also discussed in Chapter 19). In comparison, analog filters are ghastly in this respect.

Почему там в этих параметрах нет ничего относительно *фазы*? Во первых, фаза не важна в большинстве приложений частотного домена. Например, фаза звукового сигнала почти полностью случайна, и содержит немного полезной информации. Во вторых, если фаза важна, очень просто делать цифровые фильтры с *совершенным* фазовым ответом, то есть, проход всех частот через фильтр с нулевым сдвигом фаз (также обсуждается в главе 19). Для сравнения, аналоговые фильтры ужасны в этом отношении.

Previous chapters have described how the DFT converts a system's impulse response into its frequency response. Here is a brief review. The quickest way to calculate the DFT is by means of the FFT algorithm presented in Chapter 12. Starting with a filter kernel N samples long, the FFT calculates the frequency spectrum consisting of an N point *real part* and an N point *imaginary part*. Only samples 0 to of the FFT's real and imaginary parts $N/2$ contain useful information; the remaining points are duplicates (negative frequencies) and can be ignored. Since the real and imaginary parts are difficult for humans to understand, they are usually converted into polar notation as described in Chapter 8. This provides the magnitude and phase signals, each running from sample 0 to sample (i.e. $N/2 + 1$, samples in each signal). For example, an impulse response of 256 points will result in a frequency response running from point 0 to 128. Sample 0 represents DC, i.e., zero frequency. Sample 128 represents one-half of the sampling rate. Remember, no frequencies higher than one-half of the sampling rate can appear in sampled data.

Предыдущие главы описали, как ДПФ преобразовывает импульсную передаточную функцию системы в его частотную характеристику. Имеется краткий обзор. Самый быстрый способ вычислить ДПФ - посредством алгоритма БПФ, представленного в главе 12. Начинают с ядра фильтра длительностью N выборок, БПФ вычисляют спектр частот, состоящий из N точек *вещественной части*, и N *точек мнимой* части. Только выборки 0 из БПФ в вещественных и мнимых частях $N/2$, содержат полезную информацию; остающиеся точки - дубликаты (отрицательные частоты) и могут игнорироваться. Так как вещественные и мнимые части трудны для людей, чтобы понять, они обычно преобразовываются в полярную систему обозначений как описано в главе 8. Это обеспечивает величину и фазы сигналов, каждая выполняющаяся от выборки 0, к выборке $N/2$ (то есть, $N/2 + 1$ выборки в каждый сигнал). Например, ответ импульса 256 точек приведет к частотной характеристике, выполняющейся от точки от 0 до 128. Выборка 0 представляет постоянный ток, то есть., нулевая частота. Выборка 128 представляет половину частоты выборки. Вспомните(не забудьте), никакие частоты выше чем половина частоты выборки не могут появляться в выбранных данных.

The number of samples used to represent the impulse response can be arbitrarily large. For instance, suppose you want to find the frequency response of a filter kernel that consists of 80 points. Since the FFT only works with signals that are a power of two, you need to add 48 zeros to the signal to bring it to a length of 128 samples. This *padding with zeros* does not change the impulse response. To understand why this is so, think about what happens to these added zeros

when the input signal is convolved with the system's impulse response. The added zeros simply *vanish* in the convolution, and do not affect the outcome.

Число выборок используемых чтобы представить импульсную передаточную функцию, может быть произвольно большим. Для образца, предположите, что Вы хотите найти частотную характеристику ядра фильтра, которое состоит из 80 точек. Так как БПФ работает только с сигналами, которые являются мощью два, Вы должны прибавить 48 нулей к сигналу, чтобы принести это к длине 128 выборок. Это *дополнение нулями* не изменяет импульсную передаточную функцию. Чтобы понимать, почему это - так, думайте относительно того, что случается с этими добавленными нулями, когда входной сигнал свернут(скручен) с импульсной передаточной функцией системы. Добавленные нули просто обращаются в нуль в скручивании(свертке), и не затрагивают результат.

Taking this a step further, you could add *many* zeros to the impulse response to make it, say, 256, 512, or 1024 points long. The important idea is that longer impulse responses result in a closer spacing of the data points in the frequency response. That is, there are more samples spread between DC and one-half of the sampling rate. Taking this to the extreme, if the impulse response is padded with an *infinite* number of zeros, the data points in the frequency response are infinitesimally close together, i.e., a continuous line. In other words, the frequency response of a filter is really a *continuous* signal between DC and one-half of the sampling rate. The output of the DFT is a *sampling* of this continuous line. What length of impulse response should you use when calculating a filter's frequency response? As a first thought, try $N = 1024$, but don't be afraid to change it if needed (such as insufficient resolution or excessive computation time).

Идя далее, Вы могли прибавлять *много* нулей к импульсной передаточной функции, чтобы делать это, скажем, 256, 512, или 1024 точками длиной. Важность идеи состоит в том, что более длинные импульсные передаточные функции приводят к меньшему интервалу между точками данных в частотной характеристике. То есть имеется большее количество выборок распространенных между постоянным током и половиной частоты выборки. Беря это до крайности, если импульсная передаточная функция дополняется *бесконечным* числом нулей, точки данных в частотной характеристике – находятся бесконечно мало близко друг к другу, то есть, непрерывная линия. Другими словами, частотная характеристика фильтра - действительно непрерывный сигнал между постоянным током и половиной частоты выборки. Выход ДПФ - осуществление выборки этой непрерывной линии. Какую длину импульсной передаточной функции Вы должны использовать при вычислении частотной характеристики фильтра? Как первая мысль, пробуйте $N = 1024$, но не бойтесь изменить это если необходимо (типа недостаточной разрешающей способности или чрезмерного времени вычисления).

Keep in mind that the "good" and "bad" parameters discussed in this chapter are only generalizations. Many signals don't fall neatly into categories. For example, consider an EKG signal contaminated with 60 hertz interference. The information is encoded in the *time domain*, but the interference is best dealt with in the *frequency domain*. The best design for this application is bound to have trade-offs, and might go against the conventional wisdom of this chapter. Remember the number one rule of education: *A paragraph in a book doesn't give you a license to stop thinking*.

Имейте в виду, что "хорошие" и "плохие" параметры, обсужденные в этой главе - только обобщения. Много сигналов не попадают точно в категории. Например, рассмотрите сигнал ЭКГ, загрязненный интерференцией 60 герц. Информация закодирована в *домене времени*, но с интерференций лучше иметь дело в *частотном домене*. Лучший проект для этого приложения связан с тем, чтобы пойти на компромисс, и пойти против обычной

мудрости этой главы. Помните(не забудьте) номер одно правило образования: *параграф в книге не дает Вам права, чтобы прекратить думать*.

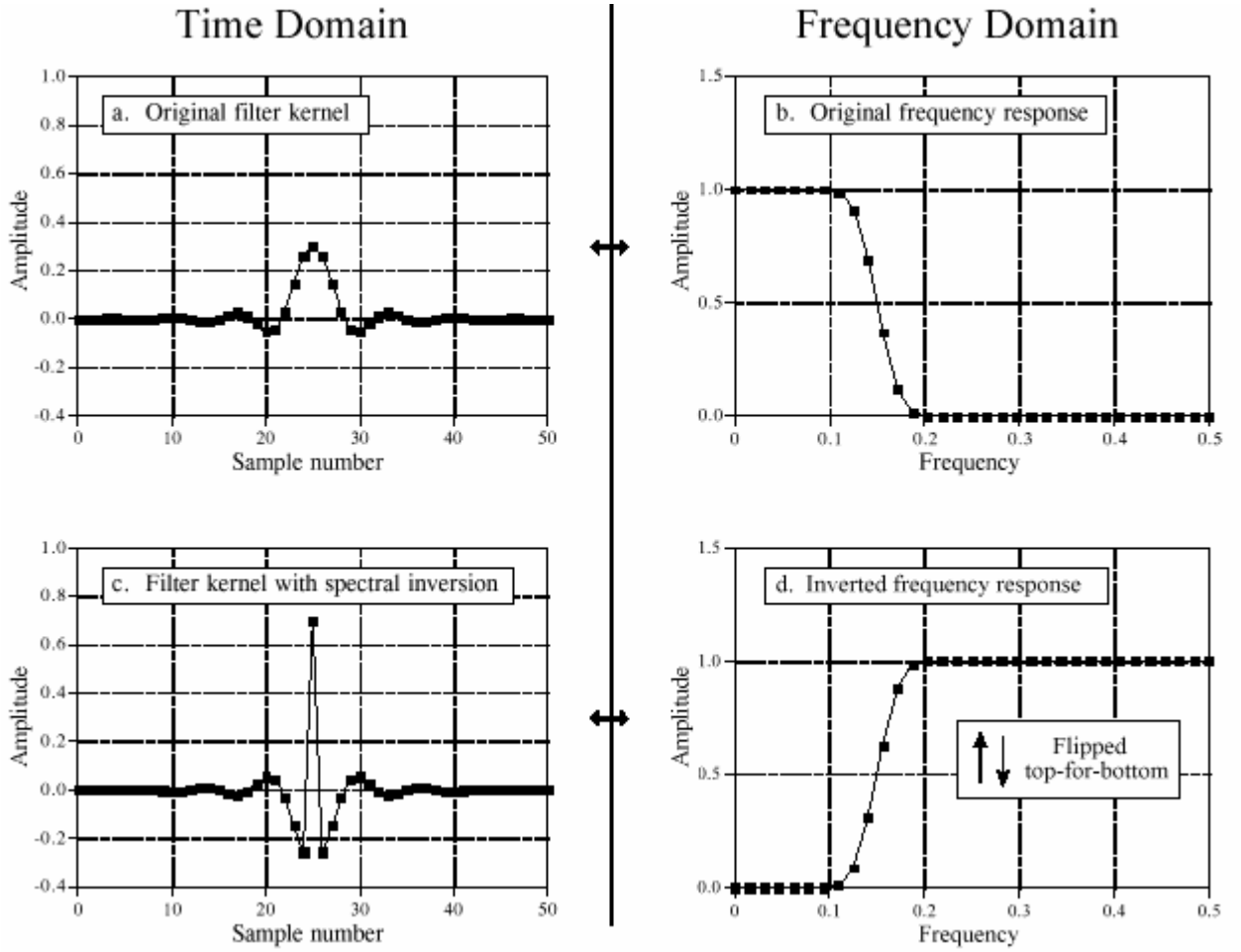

FIGURE 14-5. Example of spectral inversion.
The low-pass filter kernel in (a) has the frequency response shown in (b). A high-pass filter kernel, (c), is formed by changing the sign of each sample in (a), and adding one to the sample at the center of symmetry. This action in the time domain *inverts* the frequency spectrum (i.e., flips it top-for-bottom), as shown by the high-pass frequency response in (d).

РИСУНОК 14-5. Пример спектральной инверсии.
Ядро фильтра нижних частот в (a) показывает частотную характеристику в (b). Ядро фильтра верхних частот, (c), сформировано, изменяя знак каждой выборки в (a), и прибавляя единицу к выборке в центре симметрии. Это действие в домене времени инвертирует спектр частот (то есть, зеркально отражает это "вершина - основание "), как показано ответом частоты(частотной характеристикой) фильтра верхних частот в (d).

High-Pass, Band-Pass and Band-Reject Filters
Фильтр верхних частот, Полосовые и Полосовые - заграждающие(режекторные) Фильтры

High-pass, band-pass and band-reject filters are designed by starting with a low-pass filter, and then converting it into the desired response. For this reason, most discussions on filter design only give examples of low-pass filters. There are two methods for the low-pass to high-pass conversion: **spectral inversion** and **spectral reversal**. Both are equally useful.

Фильтр верхних частот, полосовые и полосовые(заграждающие) фильтры разрабатываются, начиная с фильтра нижних частот, и затем преобразовывая это в желательную характеристику. По этой причине, большинство обсуждений по проектированию фильтра, только дают примеры фильтров нижних частот. Имеются два метода для преобразования фильтра низких частот в фильтр верхних частот: **спектральная инверсия** и **спектральное реверсирование**. Оба одинаково полезны.

An example of *spectral inversion* is shown in 14-5. Figure (a) shows a low-pass filter kernel called a windowed-sinc (the topic of Chapter 16). This filter kernel is 51 points in length, although many of samples have a value so small that they appear to be zero in this graph. The corresponding frequency response is shown in (b), found by adding 13 zeros to the filter kernel and taking a 64 point FFT. Two things must be done to change the low-pass filter kernel into a high-pass filter kernel. First, change the sign of each sample in the filter kernel. Second, add *one* to the sample at the center of symmetry. This results in the high-pass filter kernel shown in (c), with the frequency response shown in (d). Spectral inversion *flips* the frequency response *top-for-bottom*, changing the passbands into stopbands, and the stopbands into passbands. In other words, it changes a filter from low-pass to high-pass, high-pass to low-pass, band-pass to band-reject, or band-reject to band-pass.

Пример *спектральной инверсии* показывается на рисунке 14-5. Рисунок (a) показывает ядро фильтра нижних частот называемое windowed-sinc (взвешенный синус?)тема главы 16). Это ядро фильтра - 51 точка длинной, хотя многие из выборок имеют настолько маленькие значения, что они, кажется, нулевыми в этой диаграмме(графике). Соответствующая частотная характеристика показанная в (b), найдена, прибавляя 13 нулей к ядру фильтра и беря(делая) 64 точка БПФ. Две вещи должны быть сделаны, чтобы изменить ядро фильтра нижних частот в ядро фильтра верхних частот. Во первых, изменить знак каждой выборки в ядре фильтра. Во вторых, прибавить тот к выборке в центре симметрии. Это приводит к ядру фильтра верхних частот, показанному в (c), с частотной характеристикой, показанной в (d). Спектральная инверсия зеркально отражает частотную характеристику " вершина - основание ", изменяя полосы пропускания в полосы задерживания(полосы затухания; полосы ослабления), и полосы задерживания(полосы затухания; полосы ослабления) в полосы пропускания. Другими словами, это изменяет фильтр низких частот в фильтр верхних частот; фильтр верхних частот в фильтр низких частот; полосовой в полосовой - заграждающий); или полосовой - заграждающий в полосовой.

Figure 14-6 shows why this two step modification to the time domain results in an inverted frequency spectrum. In (a), the input signal, $x[n]$, is applied to two systems in parallel. One of these systems is a low-pass filter, with an impulse response given by $h[n]$. The other system does *nothing* to the signal, and therefore has an impulse response that is a delta function, $\sigma[n]$. The overall output, $y[n]$, is equal to the output of the all-pass system *minus* the output of the low-pass system. Since the low frequency components are subtracted from the original signal, only the high frequency components appear in the output. Thus, a high-pass filter is formed.

Рисунок 14-6 показывает, почему эти два шага модификации домена времени приводят к перевернутому(инвертированному) спектру частот. В (a), входной сигнал, $x[n]$, применяется к двум системам параллельно. Одна из этих систем - фильтр нижних частот, с импульсной передаточной функцией, данной $h[n]$. Другая система не делает *ничего* к сигналу, и поэтому имеет импульсную передаточную функцию, которая является дельта функцией, $\sigma[n]$. Полный выход, $y[n]$, является равным выходу системы пропускающей все отрицательные частоты, системы с низким проходом. Так как низкочастотные компоненты вычитаются от первоначального сигнала, только компоненты высокой частоты появляются в выходе. Таким образом, фильтр верхних частот сформирован.

(c) АВТЭКС, Санкт-Петербург, http://www.autex.spb.ru, e-mail: info@autex.spb.ru

This could be performed as a two step operation in a computer program: run the signal through a low-pass filter, and then subtract the filtered signal from the original. However, the entire operation can be performed in a signal stage by combining the two filter kernels. As described in Chapter 7, parallel systems with added outputs can be combined into a single stage by adding their impulse responses. As shown in (b), the filter kernel for the high-pass filter is given by: σ[*n*] - *h*[*n*]. That is, change the sign of all the samples, and then add one to the sample at the center of symmetry.

Это могло быть выполнено операцией в два шага в компьютерной программе: выполните сигнал через фильтр нижних частот, и затем вычтите фильтрованный сигнал от оригинала. Однако, полная операция может быть выполнена в стадии сигнала, объединяя два ядра фильтра. Как описано в главе 7, параллельные системы с добавленными выходами могут быть объединены в отдельную стадию, прибавляя их импульсные передаточные функции. Как показано в (b), ядро фильтра для фильтра верхних частот дается: σ[*n*] - *h*[*n*]. То есть измените(замените) знак всех выборок, и затем прибавьте тот к выборке в центре симметрии.

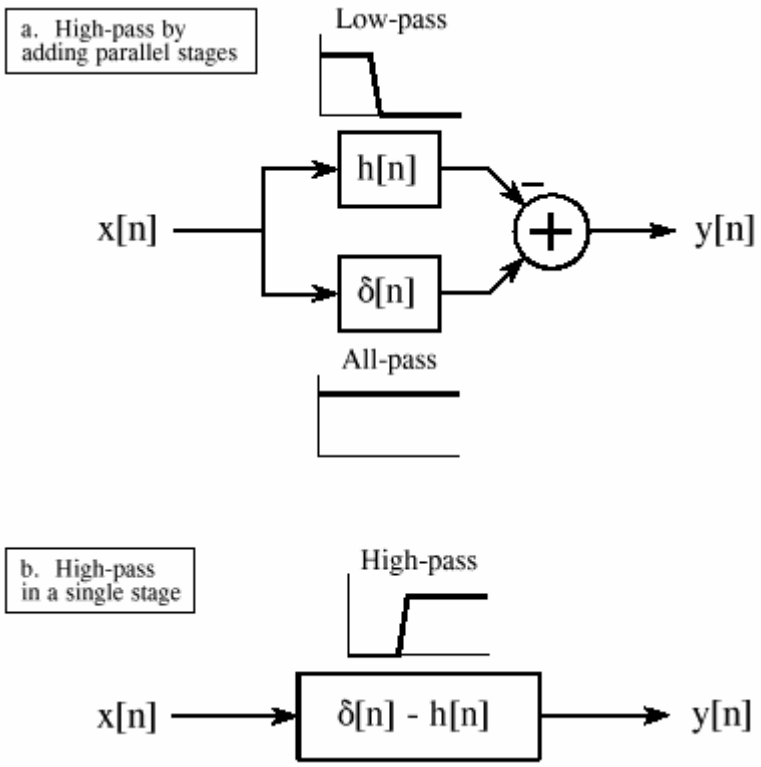

FIGURE 14-6. Block diagram of spectral inversion.
In (a), the input signal, *x*[*n*], is applied to two systems in parallel, having impulse responses of *h*[*n*] and σ[*n*]. As shown in (b), the combined system has an impulse response of σ[*n*] - *h*[*n*]. This means that the frequency response of the combined system is the *inversion* of the frequency response of *h*[*n*].

РИСУНОК 14-6. Блок-схема спектральной инверсии.
В (a), входной сигнал, *x*[*n*], применяется к двум параллельным системам, имея импульсные передаточные функции *h*[*n*] и σ[*n*]. Как показано (b), объединенная система имеет импульсную передаточную функцию σ[*n*] - *h*[*n*]. Это означает, что частотная характеристика объединенной системы - инверсия частотной характеристики *h*[*n*].

For this technique to work, the low-frequency components exiting the low-pass filter must have the same phase as the low-frequency components exiting the all-pass system. Otherwise a complete subtraction cannot take place. This places two restrictions on the method: (1) the original

filter kernel must have left-right symmetry (i.e., a zero or linear phase), and (2) the impulse must be added at the center of symmetry.

Для этой методики, чтобы работать, низкочастотные компоненты, выходящие из фильтра нижних частот должны иметь ту же самую фазу как низкочастотные компоненты, выходящие из пропускающей все частоты системы. Иначе полное вычитание не может иметь место. Это размещает два ограничения на метод: (1) первоначальное ядро фильтра должно иметь лево - правую симметрию (то есть, нулевая или линейная фаза), и (2), импульс должен быть добавлен в центре симметрии.

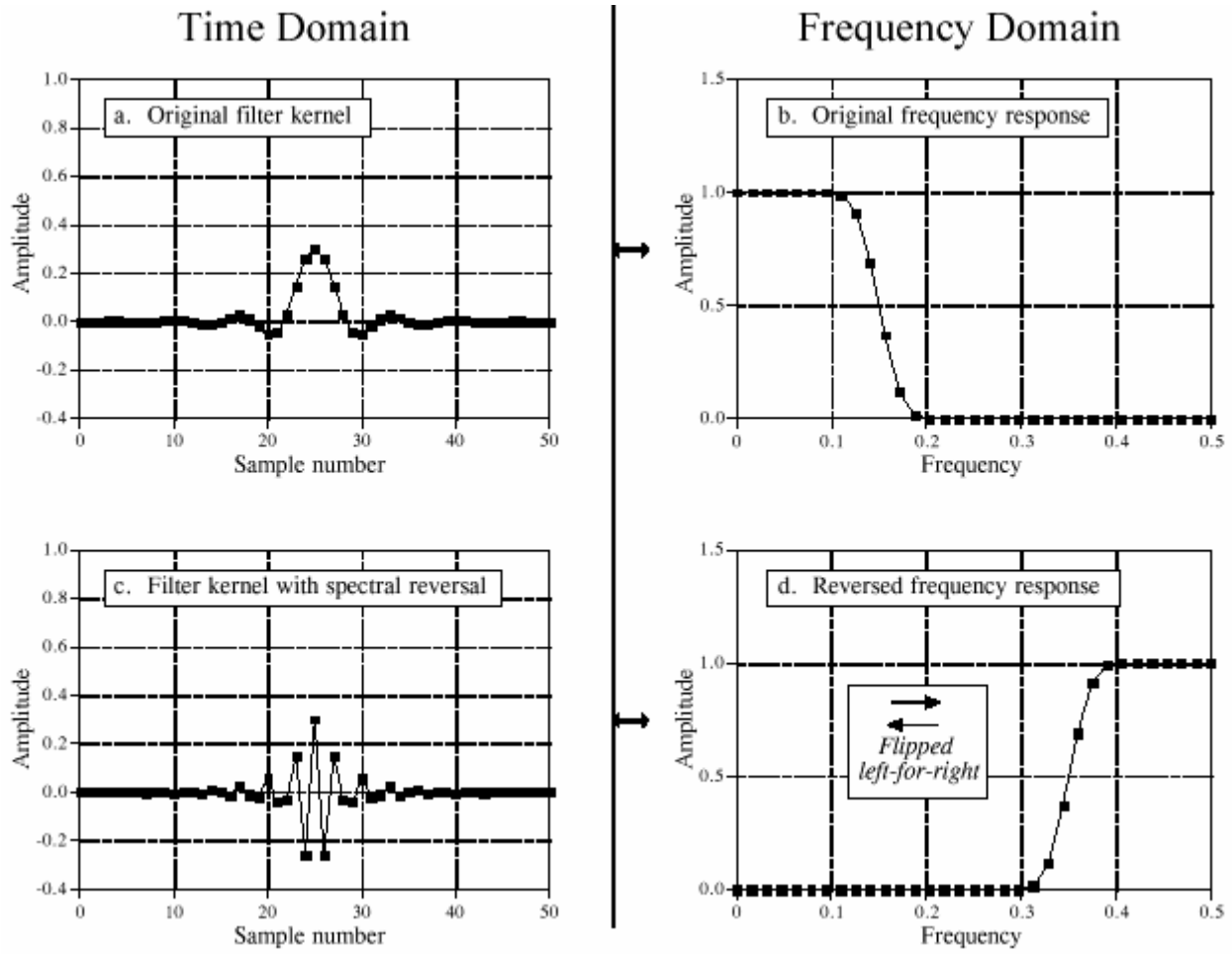

FIGURE 14-7. Example of spectral reversal.
The low-pass filter kernel in (a) has the frequency response shown in (b). A high-pass filter kernel, (c), is formed by changing the sign of every other sample in (a). This action in the time domain results in the frequency domain being flipped *left-for-right*, resulting in the high-pass frequency response shown in (d).

РИСУНОК 14-7. Пример спектрального реверсирования.
Ядро фильтра нижних частот в (a) показывает частотную характеристику в (b). Ядро фильтра верхних частот, (c), сформировано, изменяя знак каждой другой выборки в (a). Это действие в домене времени приводит к зеркально отражаемому частотного домена, слева - направо, приводя к частотной характеристике фильтра верхних частот, показанной в (d).

The second method for low-pass to high-pass conversion, *spectral reversal*, is illustrated in Fig. 14-7. Just as before, the low-pass filter kernel in (a) corresponds to the frequency response in (b). The high-pass filter kernel, (c), is formed by *changing the sign of every other sample* in (a). As shown in (d), this flips the frequency domain *left-for-right*: 0 becomes 0.5 and 0.5 becomes 0. The cutoff frequency of the example low-pass filter is 0.15, resulting in the cutoff frequency of the high-pass filter being 0.35.

(c) АВТЭКС, Санкт-Петербург, http://www.autex.spb.ru, e-mail: info@autex.spb.ru

Второй метод для преобразования фильтра низких частот в фильтр верхних частот, *спектральным реверсированием*, иллюстрированным в рис. 14-7. Так же, как прежде, ядро фильтра нижних частот в (a) соответствует частотной характеристике в (b). Ядро фильтра верхних частот, (c), сформировано, *изменяя знак каждой другой(дополнительной) выборки* в (a). Как показано в (d), это зеркально отражает частотный домен, *лево вправо*: 0 становится 0.5, и 0.5 становится 0. Граничная частота фильтра нижних частот примера - 0.15, приводя к граничной частоте фильтра верхних частот, являющегося 0.35.

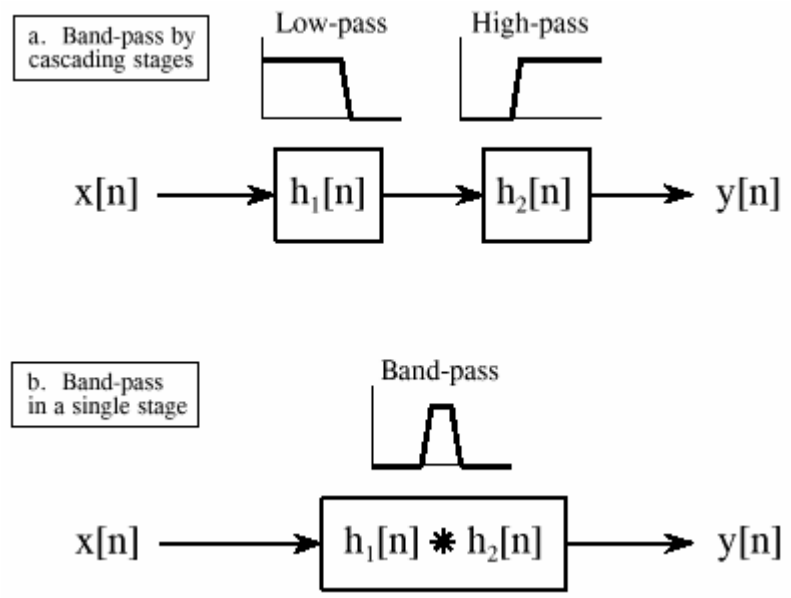

FIGURE 14-8. Designing a band-pass filter.
As shown in (a), a band-pass filter can be formed by cascading a low-pass filter and a high-pass filter. This can be reduced to a single stage, shown in (b). The filter kernel of the single stage is equal to the *convolution* of the low-pass and high-pass filter kernels.

РИСУНОК 14-8. Проектирование полосового фильтра.
Как показано в (a), полосовой фильтр может быть сформирован, располагая каскадом фильтр нижних частот и фильтр верхних частот. Это может быть сокращено к единственной этапе(каскаде), показан в (b). Ядро фильтра единственного этапа(каскада) равно скручиванию ядер фильтра верхних частот и низких частот.

Changing the sign of every other sample is equivalent to multiplying the filter kernel by a sinusoid with a frequency of 0.5. As discussed in Chapter 10, this has the effect of *shifting* the frequency domain by 0.5. Look at (b) and imagine the negative frequencies between -0.5 and 0 that are of mirror image of the frequencies between 0 and 0.5. The frequencies that appear in (d) are the negative frequencies from (b) shifted by 0.5.

Изменение(замена) знака каждой другой выборки эквивалентно умножению ядра фильтра синусоидой с частотой 0.5. Как обсуждено в главе 10, это имеет эффект смещения частотного домена(области) 0.5. Смотрите (на (b), и вообразите отрицательные частоты между -0.5 и 0, которые имеют зеркальное изображение(образ) частот между 0 и 0.5. Частоты, которые появляются в (d) - отрицательные частоты от (b), сдвинутой 0.5.

Lastly, Figs. 14-8 and 14-9 show how low-pass and high-pass filter kernels can be combined to form band-pass and band-reject filters. In short, *adding* the filter kernels produces a *band-reject* filter, while *convolving* the filter kernels produces a *band-pass* filter. These are based on the way cascaded and parallel systems are be combined, as discussed in Chapter 7. Multiple combination

of these techniques can also be used. For instance, a band-pass filter can be designed by adding the two filter kernels to form a stop-pass filter, and then use *spectral inversion* or *spectral reversal* as previously described. All these techniques work very well with few surprises.

Наконец, Рис. 14-8 и 14-9 показывают, как ядра фильтра низких частот и фильтра верхних частот могут быть объединены, чтобы формировать полосовые и полосовые заграждающие фильтры. Короче говоря, *добавляя* ядра фильтров, производят *полосовой заграждающий* фильтр, при скручивании ядер фильтров производится *полосовой фильтр*. Они основаны на пути, которым каскадные и параллельные системы являются быть объединенными, как обсуждено в главе 7. Множественная комбинация этих методов может также использоваться. Например, полосовой фильтр может быть разработан, прибавляя два ядра фильтра, чтобы формировать stop-pass фильтр (фильтр *прохода останова)*, и затем использовать спектральную инверсию или спектральное реверсирование как предварительно описано. Все эти методы работают очень хорошо с немногими неожиданностями.

Filter Classification
Классификация Фильтра

Table 14-1 summarizes how digital filters are classified by their *use* and by their *implementation*. The use of a digital filter can be broken into three categories: *time domain*, *frequency domain* and *custom*. As previously described, time domain filters are used when the information is encoded in the shape of the signal's waveform. Time domain filtering is used for such actions as: smoothing, DC removal, waveform shaping, etc. In contrast, frequency domain filters are used when the information is contained in the amplitude, frequency, and phase of the component sinusoids. The goal of these filters is to separate one band of frequencies from another. Custom filters are used when a special action is required by the filter, something more elaborate than the four basic responses (high-pass, low-pass, band-pass and band-reject). For instance, Chapter 17 describes how custom filters can be used for *deconvolution*, a way of counteracting an unwanted convolution.

Таблица 14-1 суммирует, как цифровые фильтры классифицированы их *использованием* и их *выполнением*. Использование цифрового фильтра может в трех категориях(классах): *домен времени*, *частотный домен* и *заказной*. Как предварительно описано, фильтры домена времени используются, когда информация закодирована в форме формы волны сигнала. Фильтрация домена Времени используется для таких действий как: сглаживание, удаление постоянного тока, формирование формы волны, и т.д. Напротив, фильтры частотного домена используются, когда информация содержится в амплитуде, частоте, и фазе составляющих синусоид. Цель этих фильтров состоит в том, чтобы отделить одну полосу частот от другой. Заказные фильтры используются, когда требуется какое либо специальное действие фильтра, более сложное чем четыре основные характеристики (фильтр верхних частот, фильтр нижних частот, полосовой и полосовой заграждающий). Например, глава 17 описывает, как заказные фильтры могут использоваться для *деконволюции*, пути противодействия нежелательному скручиванию(свертке).

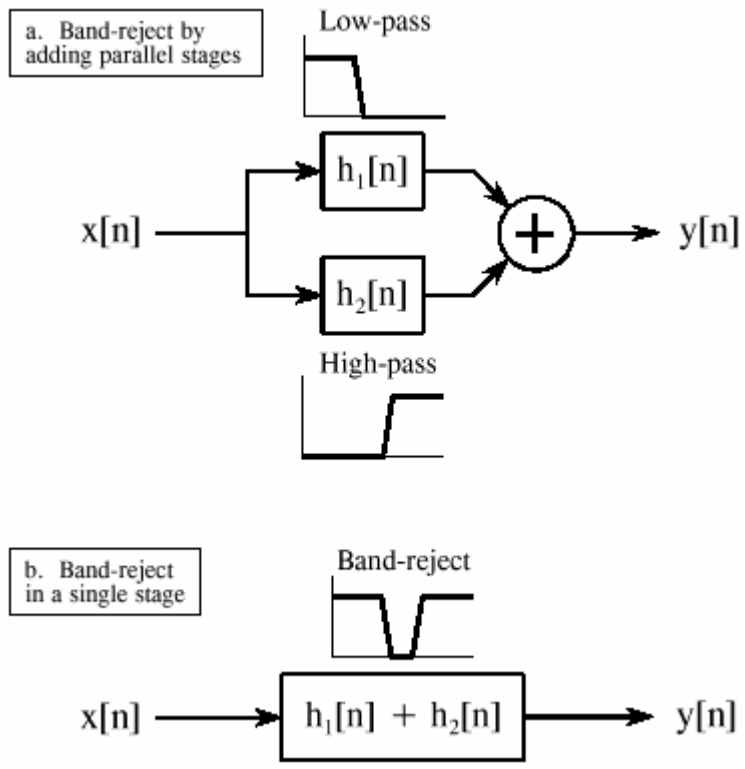

FIGURE 14-9. Designing a band-reject filter.
As shown in (a), a band-reject filter is formed by the parallel combination of a low-pass filter and a high-pass filter with their outputs added. Figure (b) shows this reduced to a single stage, with the filter kernel found by *adding* the low-pass and high-pass filter kernels.

РИСУНОК 14-9. Проектирование полосового заграждающего фильтра.
Как показано в (a), полосовой заграждающий фильтр сформирован параллельной комбинацией фильтра нижних частот и фильтра верхних частот с их добавленными выходами. Рисунок (b) показывает, что это приводило к единственной стадии, с ядром фильтра, найденным, прибавляя ядра фильтра верхних частот и низких частот.

		FILTER IMPLEMENTED BY:	
		Convolution *Finite Impulse Response (FIR)*	Recursion *Infinite Impulse Response (IIR)*
FILTER USED FOR:	Time Domain *(smoothing, DC removal)*	Moving average (Ch. 15)	Single pole (Ch. 19)
	Frequency Domain *(separating frequencies)*	Windowed-sinc (Ch. 16)	Chebyshev (Ch. 20)
	Custom *(Deconvolution)*	FIR custom (Ch. 17)	Iterative design (Ch. 26)

TABLE 14-1
Filter classification. Filters can be divided by their *use*, and how they are *implemented*.

ТАБЛИЦА 14-1.
Классификация Фильтра. Фильтры могут быть разделены их использованием, и как они *осуществлены*.

Digital filters can be implemented in two ways, by *convolution* (also called *finite impulse response* or *FIR*) and by *recursion* (also called *infinite impulse response* or *IIR*). Filters carried out by convolution can have far better performance than filters using recursion, but execute much more slowly.

Цифровые фильтры могут быть осуществлены двумя способами, *скручиванием*(сверткой) (*также называемый конечным ответом импульса* или *КИХ*) и *рекурсией* (так*же называемый бесконечной импульсной передаточной функцией* или *БИХ*). Фильтры, выполненные скручиванием(сверткой) могут иметь гораздо лучшую эффективность чем фильтры, использующие рекурсию, но выполняться намного более медленно.

The next six chapters describe digital filters according to the classifications in Table 14-1. First, we will look at filters carried out by convolution. The *moving average* (Chapter 15) is used in the time domain, the *windowed-sinc* (Chapter 16) is used in the frequency domain, and *FIR custom* (Chapter 17) is used when something special is needed. To finish the discussion of FIR filters, Chapter 18 presents a technique called FFT convolution. This is an algorithm for increasing the speed of convolution, allowing FIR filters to execute faster.

Следующие шесть глав описывают цифровые фильтры согласно классификациям в таблице 14-1. Во первых, мы будем смотреть на фильтры, выполненные скручиванием(сверткой). *Скользящее среднее значение* (глава 15) используется в домене времени, *windowed-sinc* (глава 16) используется в частотном домене, и *КИХ заказной* (глава 17) используется, когда кое-что специальное необходимо. Чтобы заканчивать обсуждение КИХ-фильтров, глава 18 представляет методику под названием скручивание(свертка) БПФ. Это - алгоритм для увеличения быстродействия скручивания(свертки), разрешение КИХ-фильтров, чтобы выполниться быстрее.

Next, we look at recursive filters. The *single pole* recursive filter (Chapter 19) is used in the time domain, while the *Chebyshev* (Chapter 20) is used in the frequency domain. Recursive filters having a custom response are designed by *iterative techniques*. For this reason, we will delay their discussion until Chapter 26, where they will be presented with another type of iterative procedure: the neural network.

(c) АВТЭКС, Санкт-Петербург, http://www.autex.spb.ru, e-mail: info@autex.spb.ru

Затем, мы смотрим на рекурсивные фильтры. Однополюсный рекурсивный фильтр (глава 19) используется в домене времени, в то время как *Чебышев* (глава 20) используется в частотном домене. *Рекурсивные фильтры*, имеющие заказной ответ разработаны итерационными методами. По этой причине, мы задержим их обсуждение до главы 26, где они будут представлены с другим типом итерационной процедуры: нервная сеть(система).

As shown in Table 14-1, *convolution* and *recursion* are rival techniques; you must use one or the other for a particular application. How do you choose? Chapter 21 presents a head-to-head comparison of the two, in both the time and frequency domains.

Как показано в таблице 14-1, *скручивание*(*конволюция*) и *рекурсия* - конкурирующие методы; Вы должны использовать один или другой для специфического приложения. Как Вы выбираете? Глава 21 представляет сравнение голова к голове(доменная граница между доменами с встречным направлением?) этих двух доменов, и времени и частоты.