



Applications of the DFT Приложения ДПФ

The Discrete Fourier Transform (DFT) is one of the most important tools in Digital Signal Processing. This chapter discusses three common ways it is used. First, the DFT can calculate a signal's *frequency spectrum*. This is a direct examination of information encoded in the frequency, phase, and amplitude of the component sinusoids. For example, human speech and hearing use signals with this type of encoding. Second, the DFT can find a system's frequency response from the system's impulse response, and vice versa. This allows systems to be analyzed in the *frequency domain*, just as convolution allows systems to be analyzed in the *time domain*. Third, the DFT can be used as an intermediate step in more elaborate signal processing techniques. The classic example of this is *FFT convolution*, an algorithm for convolving signals that is hundreds of times faster than conventional methods.

Дискретное преобразование Фурье (ДПФ) - одно из наиболее важных инструментальных средств в Цифровой Обработке Сигналов. Эта глава обсуждает три обычных способа, которыми ДПФ используется. Во первых, ДПФ может вычислять *спектр частот* сигнала. Это - прямая экспертиза информации, закодированной в частоте, фазе, и амплитуде составляющих синусоид. Например, человеческая речь и слышимость(громкость) используют сигналы с этим типом кодирования. Во вторых, ДПФ может находить частотную характеристику системы от импульсной передаточной функции системы, и наоборот. Это позволяет системам быть проанализированным в *частотном домене*, так же, как скручивание позволяет системам быть проанализированным в *домене времени*. Третье, ДПФ может использоваться как промежуточный шаг в более сложном методе обработки сигналов. Классический пример этого - *скручивание сигналов БПФ*, алгоритм для скручивания сигналов, который в сотни раз быстрее, чем обычные методы.

Spectral Analysis of Signals Спектральный Анализ Сигналов

It is very common for information to be encoded in the sinusoids that form a signal. This is true of naturally occurring signals, as well as those that have been created by humans. Many things oscillate in our universe. For example, speech is a result of vibration of the human vocal cords; stars and planets change their brightness as they rotate on their axes and revolve around each other; ship's propellers generate periodic displacement of the water, and so on. The *shape* of the time domain waveform is not important in these signals; the key information is in the *frequency*, *phase* and *amplitude* of the component sinusoids. The DFT is used to extract this information.

Спектральный анализ сигналов очень обычен для информации, которая будет закодирована в синусоидах формирующих сигнал. Это истинно из естественно появления сигналов, также как сигналов созданных людьми. Многие вещи колеблются в нашей вселенной. Например, речь - результат вибрации человеческих голосовых связок; звезды и планеты изменяют их яркость, потому что они вращаются вокруг своих осей и вращаются вокруг друг друга; гребной винт судна генерируют периодическое смещение воды, и так далее. *Форма* формы колебаний домена времени не важна в этих сигналах; ключевая информация находится в *частоте*, *фазе* и *амплитуде* составляющих синусоид. ДПФ используется, чтобы извлечь эту информацию.

An example will show how this works. Suppose we want to investigate the sounds that travel through the ocean. To begin, a microphone is placed in the water and the resulting electronic signal amplified to a reasonable level, say a few volts. An analog low-pass filter is then used to remove all frequencies above 80 hertz, so that the signal can be digitized at 160 samples per second. After acquiring and storing several thousand samples, what next?

Покажем на примере, как это работает. Предположим, что мы хотим исследовать звуки, которые распространяются в океане. Для начала поместим в воду микрофон и полученный электронный сигнал, усилим до разумного уровня, скажем несколько ватт. Затем используем аналоговый фильтр нижних частот, чтобы удалить все частоты более чем 80 герц, так, чтобы сигнал мог быть оцифрован с частотой 160 выборок в секунду. Допустим, мы приобрели(получили) и сохранили несколько тысяч выборок, что дальше?

The first thing is to simply *look* at the data. Figure 9-1a shows 256 samples from our imaginary experiment. All that can be seen is a noisy waveform that conveys little information to the human eye. For reasons explained shortly, the next step is to multiply this signal by a smooth curve called a **Hamming window**, shown in (b). (Chapter 16 provides the equations for the Hamming and other windows; see Eqs. 16-1 and 16-2, and Fig. 16-2a). This results in a 256 point signal where the samples near the ends have been reduced in amplitude, as shown in (c).

Первое, мы можем просто *смотреть* на данные. На рисунке 9-1а показано 256 выборок полученных в результате нашего мысленного эксперимента. Все, что мы можем заметить - шумная форма волны(сигнала), которая передает небольшую информацию человеческому глазу. По причинам, которые будут вскоре объяснены, следующим шагом должно быть умножение этого сигнала на гладкую кривую называемую **окном Хемминга**, показанную в (b). (Глава 16 обеспечивает уравнения для окна Хемминга и других окон; см. уравнения, 16-1, 16-2, и рис. 16-2а). Это приводит к 256 точкам сигнала, где выборки около концов были сокращены в амплитуде, как показано в (c).

Taking the DFT, and converting to polar notation, results in the 129 point frequency spectrum in (d). Unfortunately, this also looks like a noisy mess. This is because there is not enough information in the original 256 points to obtain a well behaved curve. Using a longer DFT does nothing to help this problem. For example, if a 2048 point DFT is used, the frequency spectrum becomes 1025 samples long. Even though the original 2048 points contain more information, the greater number of samples in the spectrum dilutes the information by the same factor. Longer DFTs provide better frequency resolution, but the same noise level.

Получение ДПФ, и преобразование к полярной системе обозначений, приводит к 129 точкам частотного спектра в (d). К сожалению, это также напоминает шумный беспорядок. Это происходит потому, что в 256 точках оригинала имеется недостаточно информации, чтобы получить хорошо ведущую себя кривую(кривую с хорошими свойствами). Использование более длинного ДПФ не делает ничего, чтобы помочь в решении этой проблемы. Например, если используется 2048 точек ДПФ, спектр частот становится длиной 1025 выборок. Даже при том что оригинал, 2048 точек, содержит больше информации(подробную информацию), большое число выборок в спектре, разбавляет информацию тем же самым фактором. Более длинные ДПФ обеспечивают лучшую частотную разрешающую способность, но тот же самый уровень шума.

The answer is to use more of the original signal in a way that doesn't increase the number of points in the frequency spectrum. This can be done by breaking the input signal into many 256 point *segments*. Each of these segments is multiplied by the Hamming window, run through a 256 point DFT, and converted to polar notation. The resulting frequency spectra are then *aver-*

aged to form a single 129 point frequency spectrum. Figure (e) shows an example of averaging 100 of the frequency spectra typified by (d). The improvement is obvious; the noise has been reduced to a level that allows interesting features of the signal to be observed. Only the *magnitude* of the frequency domain is averaged in this manner; the *phase* is usually discarded because it doesn't contain useful information. The random noise reduces in proportion to the *square-root* of the number of segments. While 100 segments is typical, some applications might average *millions* of segments to bring out weak features.

Ответ - необходимо использовать больше первоначального сигнала способом, который не увеличивает число точек в частотном спектре. Это может быть сделано, разбивая входной сигнал на 256 точек во многих *сегментах*. Каждый из этих сегментов умноженный на Окно Хемминга, пробегает 256 точек ДПФ, и преобразован в полярную систему обозначений. Полученные спектры частоты тогда *усреднены*, чтобы формировать отдельные 129 точек сигнала спектра частот. Рисунок (e) показывает пример среднего из 100 частотных спектров, которые символизированы(олицетворены) (d). Улучшение(уточнение) очевидно; шум был сокращен(уменьшен) до уровня, который позволяет интересным(значимым) особенностям сигнала наблюдаться. Только *величина* частотного домена усреднена этим способом; *фаза* обычно отвергается, потому что не содержит полезной информации. Случайный шум приводится в соотношении(пропорционально?) к *квадратному корню* числа сегментов. В то время как типично 100 сегментов, некоторые приложения, чтобы обнаружить слабые особенности(сигнала), могли бы усреднять *миллионы* сегментов.

There is also a second method for reducing spectral noise. Start by taking a very long DFT, say 16,384 points. The resulting frequency spectrum is high resolution (8193 samples), but very noisy. A low-pass digital filter is then used to *smooth* the spectrum, reducing the noise at the expense of the resolution. For example, the simplest digital filter might average 64 adjacent samples in the original spectrum to produce each sample in the filtered spectrum. Going through the calculations, this provides about the same noise and resolution as the first method, where the 16,384 points would be broken into 64 segments of 256 points each.

Имеется также второй метод для уменьшения спектрального шума. Начнем, беря очень длинное ДПФ, скажем 16384 точки. Полученный спектр частот имеет высокое разрешение (8193 выборки), но очень шумный. Тогда используется цифровой фильтр с низким проходом, чтобы *пригладить* спектр, приводя(уменьшая) шум за счет разрешающей способности. Например, самый простой цифровой фильтр мог бы усреднить 64 смежные выборки в первоначальном спектре, чтобы произвести каждую выборку в фильтрованном спектре. Проходя вычисления, это обеспечивает относительно тот же самый шум и разрешающую способность, что и первый метод, где 16384 точек были бы разбиты на 64 сегмента по 256 точек каждый.

Which method should you use? The first method is easier, because the digital filter isn't needed. The second method has the *potential* of better performance, because the digital filter can be tailored to optimize the trade-off between noise and resolution. However, this improved performance is seldom worth the trouble. This is because both noise and resolution can be improved by using *more data* from the input signal. For example, imagine breaking the acquired data into 10,000 segments of 16,384 samples each. This resulting frequency spectrum is high resolution (8193 points) *and* low noise (10,000 averages). Problem solved! For this reason, we will only look at the averaged segment method in this discussion.

Который метод Вы должны использовать? Первый метод проще, потому что нет необходимости использовать цифровой фильтр. Второй метод имеет *потенциал* лучшей эффективности, потому что, чтобы оптимизировать отношение между шумом и разрешающей (с) АВТЭКС, Санкт-Петербург, <http://www.autex.spb.ru>, e-mail: info@autex.spb.ru

способностью может быть приспособлен(специально сделанный) цифровой фильтр. Однако, эта улучшенная эффективность редко стоит тревоги(волнений). Это потому, что шум и разрешающая способность могут быть улучшены, использованием *большого* количество данных из входного сигнала. Например, вообразите разбиение приобретенных данных на 10000 сегментов по 16384 выборок каждый. Этот заканчивающийся спектр частот с высоким разрешением (8193 точки) и низким шумом (среднее 10000). Решенная Проблема! По этой причине в этом обсуждении, мы будем рассматривать только метод усредненного сегмента.

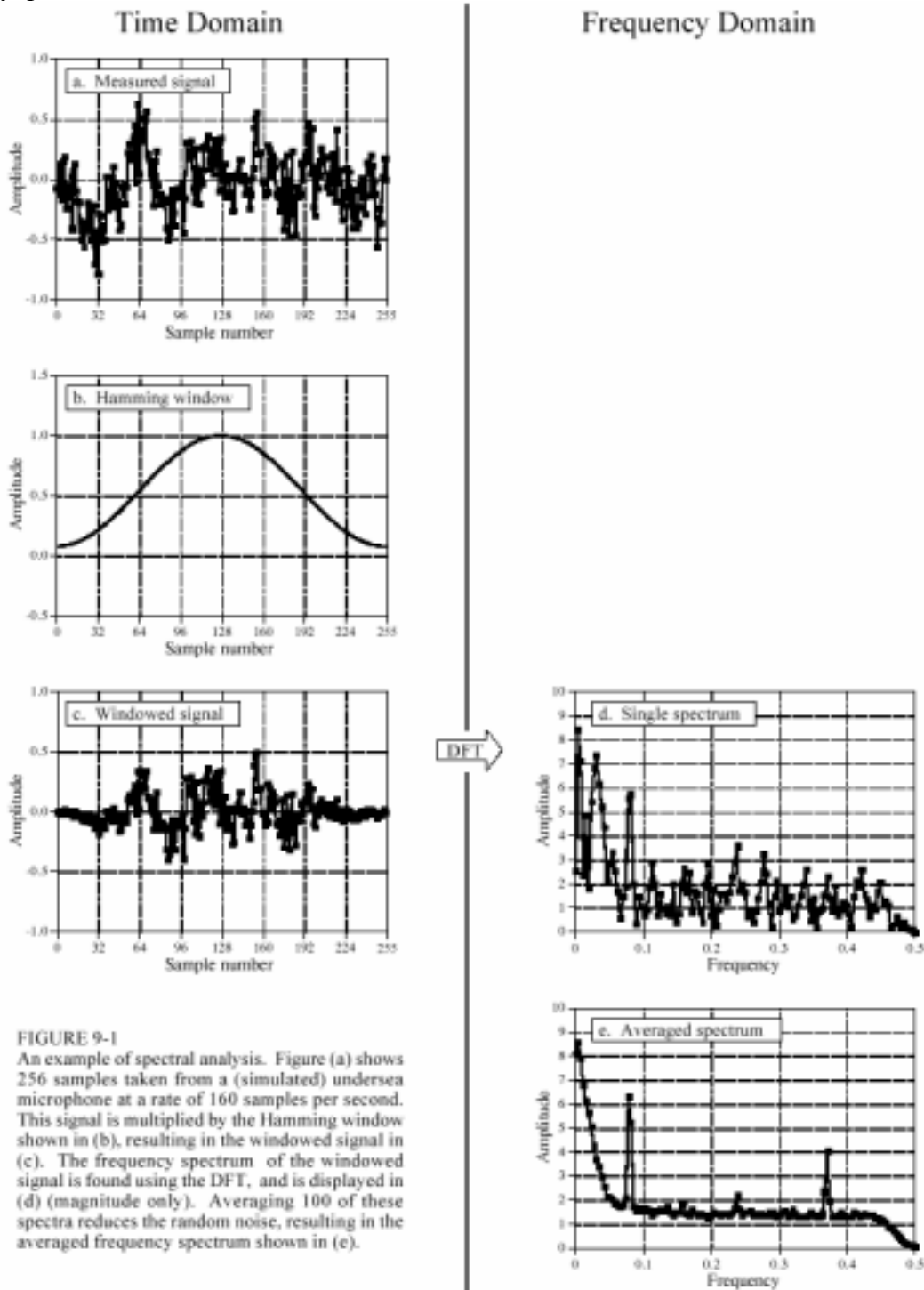


FIGURE 9-1
An example of spectral analysis. Figure (a) shows 256 samples taken from a (simulated) undersea microphone at a rate of 160 samples per second. This signal is multiplied by the Hamming window shown in (b), resulting in the windowed signal in (c). The frequency spectrum of the windowed signal is found using the DFT, and is displayed in (d) (magnitude only). Averaging 100 of these spectra reduces the random noise, resulting in the averaged frequency spectrum shown in (e).

РИСУНОК 9-1. Пример спектрального анализа.

Рисунок (а) показывает 256 выборок, принятые от (имитированного) подводного микрофона с частотой 160 выборок в секунду. Этот сигнал умножен на Окно Хемминга, показанное в (b), приводя к сигналу обрабатываемого методом окна в (c). Спектр частот сигнала реализуемого посредством организации окна найден,

используя ДПФ, и отображен в d) (только величина). Среднее этих 100 спектров приводит случайный шум, приводя к усредненному спектру частот, показанному в (e).

Figure 9-2 shows an example spectrum from our undersea microphone, illustrating the features that commonly appear in the frequency spectra of acquired signals. Ignore the sharp peaks for a moment. Between 10 and 70 hertz, the signal consists of a relatively flat region. This is called **white noise** because it contains an equal amount of all frequencies, the same as white light. It results from the noise on the time domain waveform being *uncorrelated* from sample-to-sample. That is, knowing the noise value present on any one sample provides no information on the noise value present on any other sample. For example, the random motion of electrons in electronic circuits produces white noise. As a more familiar example, the sound of the water spray hitting the shower floor is white noise. The white noise shown in Fig. 9-2 could be originating from any of several sources, including the analog electronics, or the ocean itself.

На рисунке 9-2 показан пример спектра от нашего подводного микрофона, иллюстрируя особенности, какие обычно появляются в частотных спектрах приобретенных сигналов. Игнорируя крутые мгновенные пики, между 10 и 70 герц сигнал состоит из относительно плоской области. Это называется **белым шумом**, потому что содержит равное количество всех частот, так же самое как белый свет(световое излучение). Это следует из шума в форме волны домена времени, являющимся *некоррелированным* от "выборки к выборке". То есть, знание значения представленного на любой выборке шума не обеспечивает никакой информации относительно значения шума представленного на любой другой выборке. Например, беспорядочное движение электронов в электронных цепях производит белый шум. Как более знакомый пример, звук водного распыления падающей на пол воды из душа - белый шум. Белый шум, показанный на рис. 9-2 мог бы происходить из любого из нескольких источников, включая аналоговую электронику, или непосредственно океан.

Above 70 hertz, the white noise rapidly decreases in amplitude. This is a result of the roll-off of the antialias filter. An ideal filter would pass all frequencies below 80 hertz, and block all frequencies above. In practice, a perfectly sharp cutoff isn't possible, and you should expect to see this gradual drop. If you don't, suspect that an aliasing problem is present.

Выше 70 герц, белый шум быстро уменьшается в амплитуде. Это - результат завала фильтра для устранения эффектов наложения спектров. Идеальный фильтр передал бы все частоты ниже, и блок всех частот выше 80 герц. Практически, совершенно крутой порог не возможен, и Вы должны ожидать видеть это постепенное снижение. Если Вы не делаете, подозревайте(полагайте), что проблема наложения спектров присутствует.

Below about 10 hertz, the noise rapidly increases due to a curiosity called **1/f noise** (one-over-f noise). 1/f noise is a mystery. It has been measured in very diverse systems, such as traffic density on freeways and electronic noise in transistors. It probably could be measured in all systems, if you look low enough in frequency. In spite of its wide occurrence, a general theory and understanding of 1/f noise has eluded researchers. The cause of this noise can be identified in some specific systems; however, this doesn't answer the question of why 1/f noise is everywhere. For common analog electronics and most physical systems, the transition between white noise and 1/f noise occurs between about 1 and 100 hertz.

Ниже 10 герц шум быстро увеличивается из-за редкой(любопытной) вещи, называемой **1/f шум** (one-over-f шум). 1/f шум – загадка. Это было измерено в очень иных системах, типа плотность(интенсивность) трафика(движения) на автострадах и электронного шума в транзисторах. Это вероятно могло бы быть измерено во всех системах, если Вы смотрите(ищите) достаточно низкую частоту. Несмотря на его широкое местонахождение

ние(распространение), общая теория и понимание $1/f$ шума уклонились от исследователей. Причина этого шума может быть идентифицирована(отождествлена) в некоторых специфических системах; однако это не отвечает на вопрос того, почему $1/f$ шум появляется всюду. Для общей(обычной) аналоговой электроники и наиболее физических систем, перемещение(переход, превращение) между белым шумом и $1/f$ шумом находится приблизительно между 1 и 100 герцами.

Now we come to the sharp peaks in Fig. 9-2. The easiest to explain is at 60 hertz, a result of electromagnetic interference from commercial electrical power. Also expect to see smaller peaks at multiples of this frequency (120, 180, 240 hertz, etc.) since the power line waveform is not a *perfect* sinusoid. It is also common to find interfering peaks between 25-40 kHz, a favorite for designers of switching power supplies. Nearby radio and television stations produce interfering peaks in the megahertz range. Low frequency peaks can be caused by components in the system vibrating when shaken. This is called *microphonics*, and typically creates peaks at 10 to 100 hertz.

Теперь мы прибываем в крутые пики на рис. 9-2. Самое простое объяснение - 60 герц, результат электромагнитной интерференции(помехи) от промышленной электрической энергии(сети). Также ожидайте видеть меньшие пики кратные этой частоте (120, 180, 240 герц, и т.д.) так как форма волны линии электропередачи - *не совершенная* синусоида. Также обычно встретить вмешивающиеся пики между 25-40 кГц, фаворит(самый ближайший кандидат) для конструкторов в переключении источников питания. Близлежащее радио и телевизионные станции производят вмешивающиеся пики в диапазоне мегагерц. Низкочастотные пики могут быть вызваны компонентами в вибрирующей системе, когда она подвержена колебаниям(вибрации, качаниям). Это называется *микрофонным эффектом*, и типично создает пики от 10 до 100 герц.

Now we come to the actual signals. There is a strong peak at 13 hertz, with weaker peaks at 26 and 39 hertz. As discussed in the next chapter, this is the frequency spectrum of a nonsinusoidal periodic waveform. The peak at 13 hertz is called the fundamental frequency, while the peaks at 26 and 39 hertz are referred to as the second and third harmonic respectively. You would also expect to find peaks at other multiples of 13 hertz, such as 52, 65, 78 hertz, etc. You don't see these in Fig. 9-2 because they are buried in the white noise. This 13 hertz signal might be generated, for example, by a submarine's three bladed propeller turning at 4.33 revolutions per second. This is the basis of *passive* sonar, identifying undersea sounds by their frequency and harmonic content.

Теперь мы прибываем в актуальные(фактические) сигналы. Имеется сильный пик в 13 герц, с более слабыми пиками в 26 и 39 герц. Как обсуждено в следующей главе, это - спектр частот несинусоидальной периодической формы волны. Пик в 13 герц называется фундаментальной(собственной) частотой, в то время как пики в 26 и 39 герц упомянуты как вторая и третья гармоника соответственно. Вы также ожидали бы найти другие пики кратные 13 герц, типа 52, 65, 78 герц, и т.д. Вы не видите их на рис. 9-2, потому что они захоронены(замаскированы) в белом шуме. Этот сигнал 13 герц мог бы быть сгенерирован, например, субмаринами с тремя лопастями гребного винта со скоростью вращения 4.33 оборота в секунду. Это - основание *пассивного* гидролокатора, идентифицируя подводные звуки их частотой и содержанием гармоник.

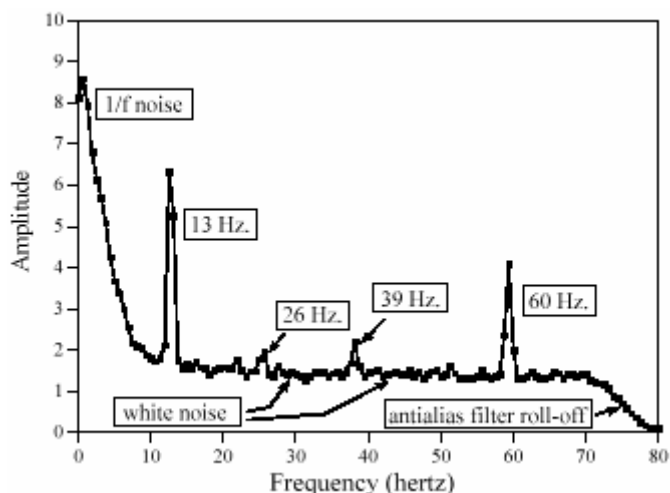


FIGURE 9-2

Example frequency spectrum. Three types of features appear in the spectra of acquired signals: (1) random noise, such as white noise and 1/f noise, (2) interfering signals from power lines, switching power supplies, radio and TV stations, microphonics, etc., and (3) real signals, usually appearing as a fundamental plus harmonics. This example spectrum (magnitude only) shows several of these features.

РИСУНОК 9-2. Пример спектра частот.

Три типа особенностей появляются в спектрах приобретенных сигналов: (1) случайный шум, типа белого шума и 1/f шума, (2) вмешивающиеся сигналы(помехи) от линий электропередачи, переключения источников питания, радио и телевизионных станций, микрофонный эффект, и т.д. и (3) реальные(настоящие) сигналы, обычно появляясь как плюс фундаментальной(основной) гармоники(частоты). Этот пример спектра (только величины) показывает несколько из этих особенностей.

Suppose there are peaks very close together, such as shown in Fig. 9-3. There are two factors that limit the frequency resolution that can be obtained, that is, how close the peaks can be without merging into a single entity. The first factor is the length of the DFT. The frequency spectrum produced by an N point DFT consists of $N/2+1$ samples equally spaced between zero and one-half of the sampling frequency. To separate two closely spaced frequencies, the sample spacing must be *smaller* than the distance between the two peaks. For example, a 512 point DFT is sufficient to separate the peaks in Fig. 9-3, while a 128 point DFT is not.

Предположим, что имеются пики, расположенные очень близко друг к другу, типа пиков, показанных на рис. 9-3. Имеются два фактора, ограничивающих частотную разрешающую способность, которая может быть получена, то есть, как близко пики могут быть без того, чтобы объединиться в единственный объект. Первый фактор - длина ДПФ. Спектр частот, произведенный ДПФ N точек, состоит из $N/2+1$ выборок, одинаково раздельных между нулем и единицей - половина выборочной частоты. Чтобы отделить две близко расположенные частоты, интервал выборки должен быть *меньше*, чем расстояние между двумя пиками. Например, ДПФ 512 точек, является достаточным, чтобы отделить пики в рис. 9-3, в то время как ДПФ 128 точек является недостаточным.

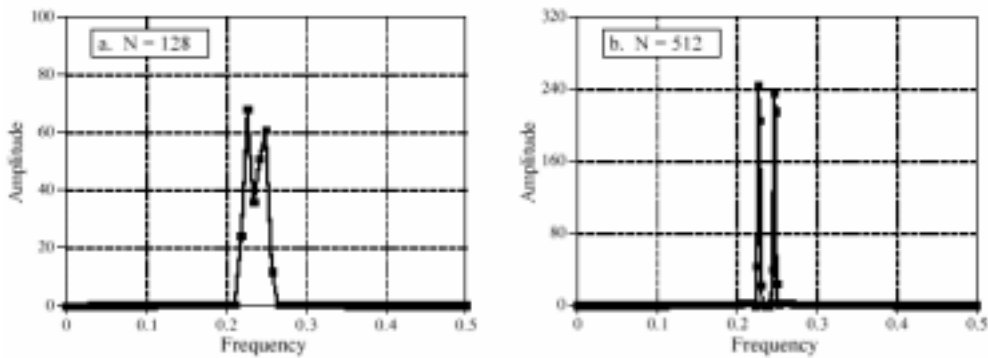


FIGURE 9-3

Frequency spectrum resolution. The longer the DFT, the better the ability to separate closely spaced features. In these example magnitudes, a 128 point DFT cannot resolve the two peaks, while a 512 point DFT can.

РИСУНОК 9-3. Разрешающая способность спектра частот.

Чем длиннее ДПФ, тем лучше способность отделить близко расположенные особенности. В этом примере величины, ДПФ 128 точек не может отделить два пика, в то время как ДПФ 512 точек может.

The second factor limiting resolution is more subtle. Imagine a signal created by adding two sine waves with only a slight difference in their frequencies. Over a short segment of this signal, say a few periods, the waveform will look like a *single* sine wave. The closer the frequencies, the longer the segment must be to conclude that more than one frequency is present. In other words, the *length* of the signal limits the frequency resolution. This is distinct from the first factor, because the *length of the input signal* does not have to be the same as the *length of the DFT*. For example, a 256 point signal could be padded with zeros to make it 2048 points long. Taking a 2048 point DFT produces a frequency spectrum with 1025 samples. The added zeros don't change the shape of the spectrum, they only provide more samples in the frequency domain. In spite of this very close sampling, the ability to separate closely spaced peaks would be only slightly better than using a 256 point DFT. When the DFT is the same length as the input signal, the resolution is limited about equally by these two factors. We will come back to this issue shortly.

Второй фактор, ограничивающий разрешающую способность более тонкий(искусный). Вообразите сигнал, созданный, сложением двух волн синуса с очень небольшой разностью в их частотах. По короткому сегменту этого сигнала, скажем нескольких периодов, форма волны напоминает(подобна) *единственную(одиночную)* волну синуса. Чем ближе частоты, тем длиннее должен быть сегмент, чтобы заключить(сделать вывод), что больше чем одна частота присутствует. Другими словами, длина сигнала ограничивает частотную разрешающую способность. Это отлично от первого фактора, потому что *длина(длительность) входного сигнала* не должна быть той же самой как *длина(длительность) ДПФ*. Например, 256 точек сигнала могут дополняться нулями, чтобы делать это длиной 2048 точек. При взятии 2048 точек ДПФ производит спектр частот с 1025 выборками. Добавленные нули не изменяют форму спектра, они только обеспечивают большее количество выборок в частотном домене. Несмотря на это, при очень близком осуществление выборок, способность отделить близко расположенные пики была бы только слегка лучше, чем при использовании 256 точек ДПФ. Когда ДПФ - той же самой длины как входной сигнал, разрешающая способность ограничена этими двумя факторами относительно одинаково. Мы возвратимся к этой проблеме вскоре.

Next question: What happens if the input signal contains a sinusoid with a frequency *between* two of the basis functions? Figure 9-4a shows the answer. This is the frequency spectrum of a signal composed of two sine waves, one having a frequency *matching* a basis function, and the other with a frequency *between* two of the basis functions. As you should expect, the first sine wave is represented as a single point. The other peak is more difficult to understand. Since it cannot be represented by a single sample, it becomes a peak with **tails** that extend a significant distance away.

Следующий вопрос: Что случается, если входной сигнал содержит синусоиду с частотой *между* двумя из базисных функций? Рисунок 9-4а показывает ответ. Это - спектр частот сигнала, составленного из двух волн синуса, одна имеет частоту, *соответствующую* базисной функции, и другая с частотой *между* двумя из базисных функций. Как Вы должны ожидать, первая волна синуса представлена как единственная точка. Другой пик более трудно понять. Так как он не может быть представлен единственной(отдельной) выборкой, это становится пиком с **хвостами (срезами импульса, крыльями, конечными фазами)**, которые расширяют значащее расстояние(интервал) далеко.

The solution? Multiply the signal by a Hamming window before taking the DFT, as was previously discussed. Figure (b) shows that the spectrum is changed in three ways by using the window. First, the two peaks are made to look more alike. This is good. Second, the tails are greatly reduced. This is also good. Third, the window reduces the resolution in the spectrum by making the peaks wider. This is bad. In DSP jargon, windows provide a trade-off between *resolution* (the width of the peak) and *spectral leakage* (the amplitude of the tails).

Решение? Умножьте сигнал на окно Хемминга перед взятием ДПФ, как было предварительно обсуждено. Рисунок (b) показывает, что используя окно, спектр изменен тремя способами. Во первых, два пика сделаны, чтобы выглядеть более подобными. Это хорошо. Во вторых, хвосты (конечные фазы) очень сокращены. Это также хорошо. Третье, окно приводит разрешающую способность в спектре, делая пики более широкими. Это плохо. На жаргоне ЦОС, окна обеспечивают обмен между *разрешающей способностью (шириной пика)* и *спектральной утечкой (амплитудой хвостов(просачиванием спектральных составляющих))*.

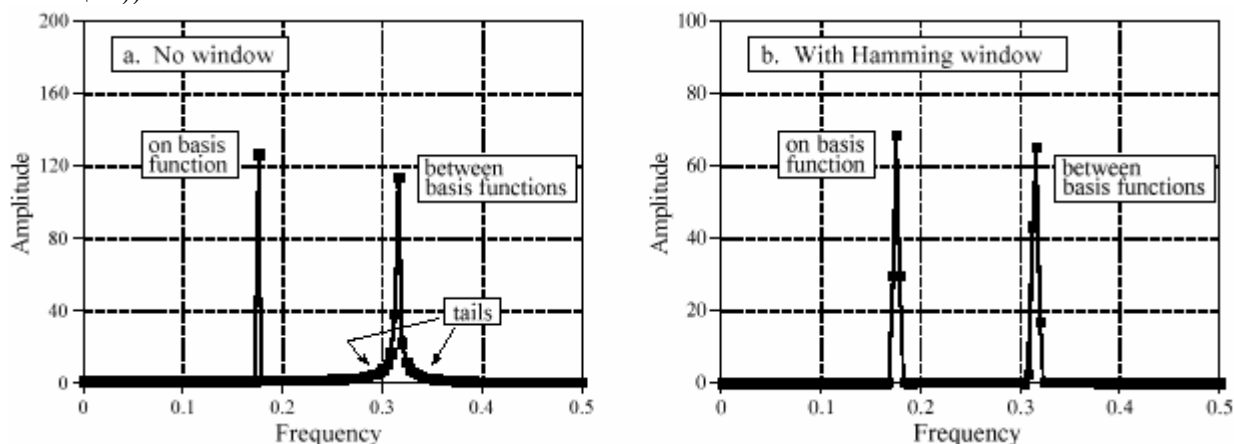


FIGURE 9-4. Example of using a window in spectral analysis. Figure (a) shows the frequency spectrum (magnitude only) of a signal consisting of two sine waves. One sine wave has a frequency exactly equal to a basis function, allowing it to be represented by a single sample. The other sine wave has a frequency *between* two of the basis functions, resulting in *tails* on the peak. Figure (b) shows the frequency spectrum of the same signal, but with a Blackman window applied before taking the DFT. The window makes the peaks look the same and reduces the tails, but broadens the peaks.

РИСУНОК 9-4. Пример использования окна в спектральном анализе.

НАУЧНО-ТЕХНИЧЕСКОЕ РУКОВОДСТВО ПО ЦИФРОВОЙ ОБРАБОТКЕ СИГНАЛОВ

Рисунок (а) показывает спектр частот (только величину) сигнала, состоящего из двух волн синуса. Одна волна синуса имеет частоту, точно равную базисной функции, позволяя этому быть представленной единственной выборкой. Другая волна синуса имеет частоту *между* двумя из базисных функций, приводя к *хвостам* (крыльям) на пике. Рисунок (б) показывает спектр частот того же самого сигнала, но с окном Blackman'a, прикладным перед взятием ДПФ. Окно делает вид пиков тем же самым и приводит (уменьшает) хвосты (крылья импульса), но расширяет пики.

To explore the theoretical aspects of this in more detail, imagine an infinitely long discrete sine wave at a frequency of 0.1 the sampling rate. The frequency spectrum of this signal is an infinitesimally narrow peak, with all other frequencies being zero. Of course, neither this signal nor its frequency spectrum can be brought into a digital computer, because of their infinite and infinitesimal nature. To get around this, we change the signal in two ways, both of which distort the true frequency spectrum.

Чтобы исследовать теоретические аспекты этого более подробно, вообразите бесконечно длинную дискретную волну синуса частотой 0.1 частоты выборки. Спектр частот этого сигнала - бесконечно маленький узкий пик, со всеми другими частотами, являющимися нулевыми. Конечно, ни этот сигнал ни его спектр частот не могут быть принесены в цифровой компьютер, из-за их бесконечности и бесконечно малого характера (природы). Чтобы обойти это, мы изменяем сигнал двумя способами, оба из которых искажают истинный спектр частот.

First, we *truncate* the information in the signal, by multiplying it by a window. For example, a 256 point *rectangular window* would allow 256 points to retain their correct value, while all the other samples in the infinitely long signal would be set to a value of zero. Likewise, the Hamming window would *shape* the retained samples, besides setting all points outside the window to zero. The signal is still infinitely long, but only a finite number of the samples have a nonzero value.

Во первых, мы *усекаем* информацию в сигнале, умножая это окном. Например, *прямоугольное окно* 256 точек позволило бы 256 точкам сохранять их правильное значение, в то время как все другие выборки в бесконечно длинном сигнале будут установлены в значение нуля. Аналогично, Окно Хемминга *формировало* бы сохраненные выборки, помимо установки всех точек вне окна к нулю. Сигнал - все еще бесконечно длинный, но только конечное число выборок имеет значение отличное от нуля.

How does this windowing affect the frequency domain? As discussed in Chapter 10, when two time domain signals are *multiplied*, the corresponding frequency domains are *convolved*. Since the original spectrum is an infinitesimally narrow peak (i.e., a delta function), the spectrum of the windowed signal is the spectrum of the window shifted to the location of the peak. Figure 9-5 shows how the spectral peak would appear using four different window options (If you need a refresher on dB, look ahead to Chapter 14). Figure 9-5a results from a rectangular window. Figures (b) and (c) result from using two popular windows, the Hamming and the Blackman (as previously mentioned, see Eqs. 16-1 and 16-2, and Fig. 16-2a for information on these windows).

Как эта работа с окнами затрагивает частотный домен? Как обсуждено в главе 10, когда два сигнала домена времени умножены, соответствующие частотные домены *свернуты* (скручены). Так как первоначальный спектр - бесконечно мало узкий пик (то есть, треугольная (дельта) функция, спектр взвешенного сигнала - спектр окна, сдвинутого к расположению пика. На рисунке 9-5 показано, как спектральный пик появился бы, используя четыре различных параметра окна (Если Вы нуждаетесь освежить в памяти что такое dB (децибел), загляните вперед в главу 14). Рисунок 9-5а следует из прямоугольного (П- (c) АВТЭКС, Санкт-Петербург, <http://www.autex.spb.ru>, e-mail: info@autex.spb.ru

образного) окна. Рисунки (b) и (c) следствие использования двух популярных окон, Хемминга и Blackman'a (как предварительно упомянуто, смотри уравнения, 16-1 и 16-2, и рис. 16-2а для информации относительно этих окон).

As shown in Fig. 9-5, all these windows have degraded the original spectrum by broadening the peak and adding tails composed of numerous side lobes. This is an unavoidable result of using only a portion of the original time domain signal. Here we can see the tradeoff between the three windows. The Blackman has the widest main lobe (bad), but the lowest amplitude tails (good). The rectangular window has the narrowest main lobe (good) but the largest tails (bad). The Hamming window sits between these two.

Как показано в рис. 9-5, все эти окна ухудшили первоначальный спектр, расширяя пик и прибавляя остатки(срезы импульса), составленные из многочисленных побочных лепестков. Это - неизбежный результат использования только части первоначального сигнала домена времени. Здесь мы можем видеть компромисс между этими тремя окнами. Blackman имеет самый широкий основной лепесток (плохо), но самые низкие амплитудные хвосты (хорошо). Прямоугольное(П-образное) окно имеет самый узкий основной лепесток (хорошо), но самые большие хвосты (плохо). Окно Хемминга находится между этими двумя.

Notice in Fig. 9-5 that the frequency spectra are continuous curves, not discrete samples. After windowing, the time domain signal is still infinitely long, even though most of the samples are zero. This means that the frequency spectrum consists of $\infty/2+1$ samples between 0 and 0.5, the same as a continuous line.

Обратите внимание, что частотные спектры на рис. 9-5 – непрерывные кривые, не дискретные выборки. После работы с окнами, сигнал домена времени - все еще бесконечно длинный, даже при том, что большинство выборок нулевое. Это означает, что спектр частот состоит из $\infty/2+1$ выборок между 0 и 0.5, то же самое как непрерывная линия.

This brings in the second way we need to modify the time domain signal to allow it to be represented in a computer: *select N points from the signal*. These N points must contain all the non-zero points identified by the window, but may also include any number of the zeros. This has the effect of *sampling* the frequency spectrum's continuous curve. For example, if N is chosen to be 1024, the spectrum's continuous curve will be sampled 513 times between 0 and 0.5. If N is chosen to be much larger than the window length, the samples in the frequency domain will be close enough that the peaks and valleys of the continuous curve will be preserved in the new spectrum. If N is made the same as the window length, the fewer number of samples in the spectrum results in the regular pattern of peaks and valleys turning into irregular tails, depending on where the samples happen to fall. This explains why the two peaks in Fig. 9- 4a do not look alike. Each peak in Fig 9-4a is a *sampling* of the underlying curve in Fig. 9-5a. The presence or absence of the tails depends on where the samples are taken in relation to the peaks and valleys. If the sine wave exactly matches a basis function, the samples occur exactly at the valleys, eliminating the tails. If the sine wave is between two basis functions, the samples occur somewhere along the peaks and valleys, resulting in various patterns of tails.

Это вводит второй способ, которым мы должны изменить сигнал домена времени, чтобы позволить ему быть представленными в компьютере: *выберите N точек из сигнала*. Эти точки N должны содержать все точки отличные от нуля, идентифицированные окном, но могут также включать любое число нулей. Это имеет эффект *осуществления выборки* непрерывной кривой спектра частот. Например, если N выбран, чтобы быть 1024, непрерывная кривая спектра будет производить выбор 513 раз между 0 и 0.5. Если N выбран, на- (с) АВТЭКС, Санкт-Петербург, <http://www.autex.spb.ru>, e-mail: info@autex.spb.ru

много больше, чем длина окна, выборки в частотном домене будут достаточно близко, так что пики и точки минимума непрерывной кривой будут сохраняться в новом спектре. Если N сделан тем же самым как длина окна, меньшее количество числа выборок в спектре приводит к регулярному образцу пиков и точек минимума, превращающихся в неправильные хвосты(крылья, срезы импульса), в зависимости от того, куда выборки попадают. Это объясняет, почему два пика на рис. 9-4а не выглядят аналогично. Каждый пик на рис. 9-4а – осуществление выборки основной кривой на рис. 9-5а. Присутствие или отсутствие хвостов зависят от того, где выборки являются принятым относительно пиков и точек минимума. Если волна синуса точно соответствует базисной функции, выборки происходят точно в точках минимума, устраняя хвосты. Если волна синуса - между двумя базисными функциями, выборки происходят где-нибудь по пикам и точкам минимума, приводя к различным образцам хвостов(крыльев).

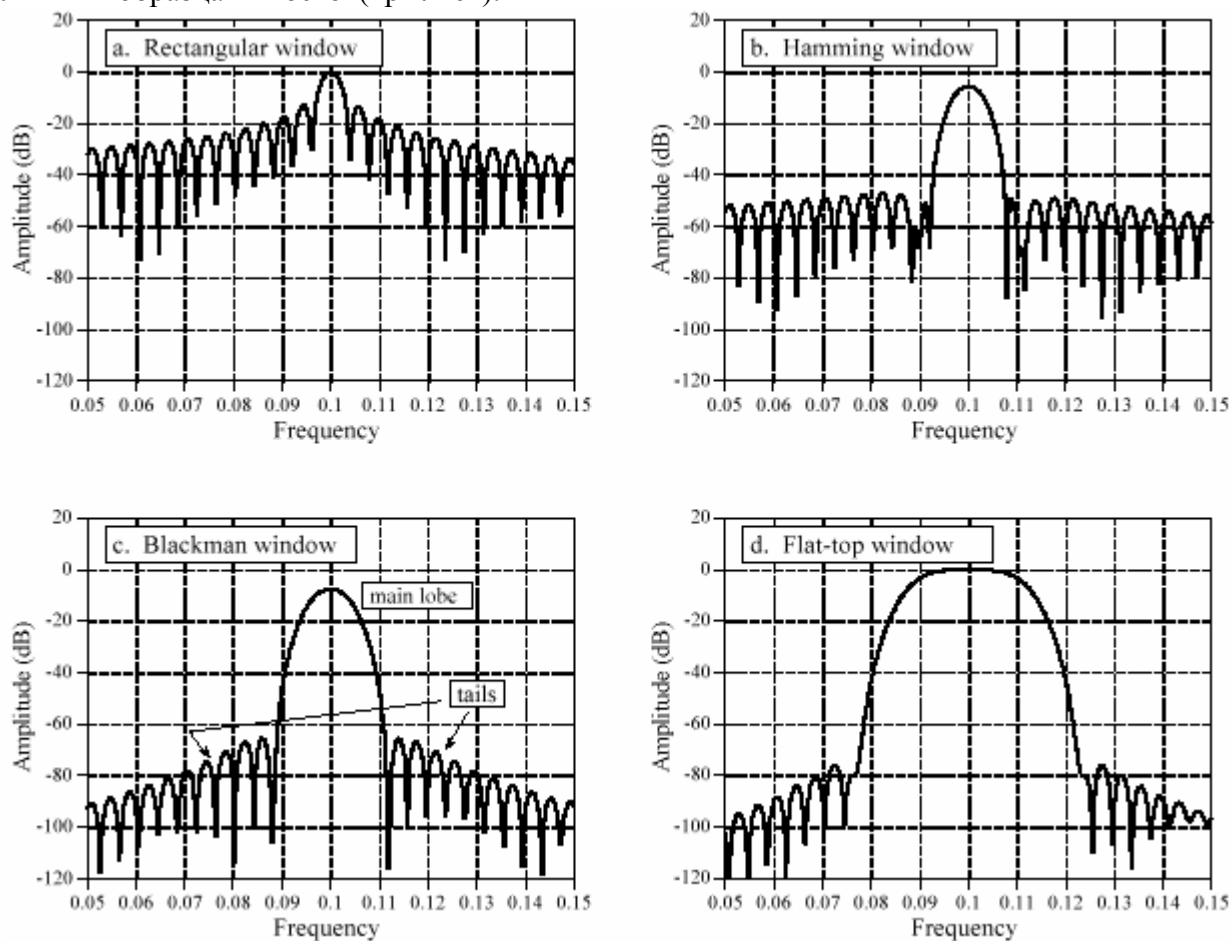


FIGURE 9-5. Detailed view of a spectral peak using various windows. Each peak in the frequency spectrum is a central lobe surrounded by tails formed from side lobes. By changing the window shape, the amplitude of the side lobes can be reduced at the expense of making the main lobe wider. The rectangular window, (a), has the narrowest main lobe but the largest amplitude side lobes. The Hamming window, (b), and the Blackman window, (c), have lower amplitude side lobes at the expense of a wider main lobe. The flat-top window, (d), is used when the amplitude of a peak must be accurately measured. These curves are for 255 point windows; longer windows produce proportionately narrower peaks.

РИСУНОК 9-5. Детальный вид спектрального пика, используя различные окна. Каждый пик в спектре частот - центральный лепесток, окруженный хвостами(крыльями), сформированными из побочных лепестков. Изменяя форму окна, амплитуда побочных лепестков может быть сокращена за счет создания более широкого основного лепестка. Прямоугольное окно, (а), имеет самый узкий основной лепесток, но самые большие амплитуды побочных лепестков. Окно Хемминга, (b), и окно Blackman`а, (c), имеет амплитуды побочных лепестков ниже, за счет более широкого основного лепестка. Плоское(П-образное) окно (d), используется когда амплитуда пика должна быть точно измерена. Эти кривые - для окон 255 точек; более длинные окна производят пропорционально более узкие пики.

As it turns out, the shape we want for a flat-top window is exactly the same shape as the filter kernel of a low-pass filter. We will discuss the theoretical reasons for this in later chapters; for now, here is a cookbook description of how the technique is used. Chapter 16 discusses a low-pass filter called the *windowed-sinc*. Equation 16-4 describes how to generate the filter kernel (which we want to use as a window), and Fig. 16-4a illustrates the typical shape of the curve. To use this equation, you will need to know the value of two parameters: M and fc . These are found from the relations: $M = N-2$, and $fc = s/N$, where N is the length of the DFT being used, and s is the number of samples you want on the flat portion of the peak (usually between 3 and 5). Table 16-1 shows a program for calculating the filter kernel (our window), including two subtle features: the normalization constant, K , and how to avoid a divide-by-zero error on the center sample. When using this method, remember that a DC value of *one* in the time domain will produce a peak of amplitude *one* in the frequency domain. However, a sinusoid of amplitude *one* in the time domain will only produce a spectral peak of amplitude *one-half*. (This is discussed in the last chapter: *Synthesis, Calculating the Inverse DFT*).

Как это собирается(выключается?), форма которую мы хотим для плоского(п-образного) окна – точно та же самая форма, как ядро фильтра нижних частот. Мы обсудим теоретические причины для этого в более поздних главах; пока, имеется описание “поваренной книги” того, как используется методика. Глава 16 обсуждает фильтр нижних частот называемый *windowed-sinc*(*взвешенный синус?*). Уравнение 16-4 описывает, как генерировать ядро фильтра (который мы хотим использовать как окно), а рис. 16-4а иллюстрирует типичную форму кривой. Чтобы использовать это уравнение, Вы будете должны знать значение двух параметров: M и fc . Они найдены от отношений: $M = N-2$, и $fc = s/N$, где N - длина используемого ДПФ, и s - число выборок, которые Вы хотите (сделать?) на плоской части пика (обычно от 3 до 5). В таблице 16-1 показана программа для вычисления ядра фильтра (наше окно), включая две тонких особенности: константа нормализации, K , и как избегать ошибки деления на ноль на средней выборке(при усреднении выборки?). При использовании этого метода, помните, что значение постоянного тока *единица* в домене времени произведет пик амплитуды *единица* в частотном домене. Однако, синусоида амплитуды *единица* в домене времени произведет спектральный пик только *половины амплитуды*. (Это обсуждено в прошлой главе: *Синтез, Вычисление Обратного ДПФ*).

Frequency Response of Systems

Частотная характеристика Систем

Systems are analyzed in the *time domain* by using convolution. A similar analysis can be done in the *frequency domain*. Using the Fourier transform, every input signal can be represented as a group of cosine waves, each with a specified amplitude and phase shift. Likewise, the DFT can be used to represent every output signal in a similar form. This means that any linear system can be *completely* described by how it changes the amplitude and phase of cosine waves passing through it. This information is called the system's **frequency response**. Since both the impulse response and the frequency response contain complete information about the system, there must be a one-to-one correspondence between the two. Given one, you can calculate the other. The relationship between the impulse response and the frequency response is one of the foundations

of signal processing: *A system's frequency response is the Fourier Transform of its impulse response.* Figure 9-6 illustrates these relationships.

Системы анализируются в *домене времени*, используя скручивание(свертку). Подобный анализ может быть сделан в *частотном домене*. Используя анализ Фурье, каждый входной сигнал может быть представлен как группа волн косинуса, каждый с указанной амплитудой и сдвигом фаз. Аналогично, ДПФ может использоваться, чтобы представить каждый сигнал выхода в подобной форме. Это означает, что любая линейная система может быть *полностью* описана тем, как она изменяет амплитуду и фазу волн косинуса, проходящих через нее. Эта информация называется **частотной характеристикой** системы. С тех пор и импульсная передаточная функция и частотная характеристика содержат полную информацию относительно системы, должно иметься взаимно-однозначное соответствие между ними. Учитывая одну, Вы можете вычислить другую. Отношения между импульсной передаточной функцией и частотной характеристикой – одна из основ обработки сигналов: *частотная характеристика системы – анализ Фурье его импульсной передаточной функции*. Рисунок 9-6 иллюстрирует эти отношения.

Keeping with standard DSP notation, impulse responses use lower case variables, while the corresponding frequency responses are upper case. Since $h[n]$ is the common symbol for the impulse response, $H[f]$ is used for the frequency response. Systems are described in the time domain by convolution, that is: $x[n] * h[n] = y[n]$. In the frequency domain, the input spectrum is *multiplied* by the frequency response, resulting in the output spectrum. As an equation: $X[f] \times H[f] = Y[f]$. That is, *convolution* in the time domain corresponds to *multiplication* in the frequency domain.

Сохраняя стандарт системы обозначений ЦОС, импульсные передаточные функции использует переменные строчных букв, в то время как соответствующий частотные характеристики - верхний регистр. В то время как $h[n]$ - общий(обычный) символ для импульсной передаточной функции, $H[f]$ используется для частотной характеристики. Системы описаны в домене времени скручиванием(сверткой) – $x[n] * h[n] = y[n]$. В частотном домене, входной спектр *умножен* на частотную характеристику, приводя к спектру выхода. Как в уравнении: $X[f] \times H[f] = Y[f]$. То есть *скручивание(свертка)* в домене времени соответствует *умножению* в частотном домене.

Figure 9-7 shows an example of using the DFT to convert a system's impulse response into its frequency response. Figure (a) is the impulse response of the system. Looking at this curve isn't going to give you the slightest idea what the system does. Taking a 64 point DFT of this impulse response produces the frequency response of the system, shown in (b). Now the function of this system becomes obvious, it passes frequencies between 0.2 and 0.3, and rejects all others. It is a band-pass filter. The *phase* of the frequency response could also be examined; however, it is *more* difficult to interpret and *less* interesting. It will be discussed in upcoming chapters.

На рисунке 9-7 показан пример использования ДПФ, чтобы преобразовать импульсную передаточную функцию системы в его частотную характеристику. Рисунок (а) - импульсная передаточная функция системы. Взгляд на эту кривую не собирается давать Вам самую небольшую идею, что система делает. При взятии 64 точек ДПФ этой импульсной передаточной функции производит частотную характеристику системы, показанную в (b). Теперь функция этой системы становится очевидной, это передает частоты между 0.2 и 0.3, и отклоняет все другие. Это - полосовой фильтр. *Фаза* частотной характеристики могла также быть исследована; однако, это *более* трудно интерпретировать и *менее* интересно. Это будет обсуждено в последующих главах.

Figure (b) is very jagged due to the low number of samples defining the curve. This situation can be improved by **padding** the impulse response with zeros before taking the DFT. For example, adding zeros to make the impulse response 512 samples long, as shown in (c), results in the higher resolution frequency response shown in (d).

Рисунок (b) очень неровно оборван из-за низкого числа выборок, определяющих кривую. Это положение(ситуация) может быть улучшено, **дополняя** импульсную передаточную функцию нулями перед взятием ДПФ. Например, добавление нулей, чтобы сделать импульсную передаточную функцию длиной 512 выборок, как показано в (c), приводит к более высокой разрешающей способности частотной характеристики, показанной в (d).

How much resolution can you obtain in the frequency response? The answer is: *infinitely* high, if you are willing to pad the impulse response with an *infinite* number of zeros. In other words, there is nothing limiting the frequency resolution except the length of the DFT. This leads to a very important concept. Even though the impulse response is a *discrete* signal, the corresponding frequency response is *continuous*. An N point DFT of the impulse response provides $N/2+1$ *samples* of this continuous curve. If you make the DFT longer, the resolution improves, and you obtain a better idea of what the continuous curve looks like. Remember what the frequency response represents: amplitude and phase changes experienced by cosine waves as they pass through the system. Since the input signal can contain *any* frequency between 0 and 0.5, the system's frequency response *must* be a continuous curve over this range.

Какую разрешающую способность Вы можете получить в частотной характеристике? Ответ: *бесконечно* высокую, если Вы желаете(готовы) дополнить импульсную передаточную функцию *бесконечным* числом нулей. Другими словами, кроме длины ДПФ ничто больше не может ограничить частотную разрешающую способность. Это ведет к очень важной концепции. Даже притом, что импульсная передаточная функция - *дискретный* сигнал, соответствующая частотная характеристика *непрерывна*. N точек, ДПФ импульсной передаточной функции обеспечивают $N/2+1$ *выборки* этой непрерывной кривой. Делая ДПФ длиннее, Вы улучшаете разрешающую способность, и получаете, лучшую идею, что непрерывная кривая напоминает. Помните то, что представляет частотная характеристика: изменения амплитуды и фазы, испытываемые волнами косинуса, при их прохождении через систему. Так как входной сигнал может содержать *любую* частоту между 0 и 0.5, частотная характеристика системы *должна быть* по этому диапазону непрерывной кривой.

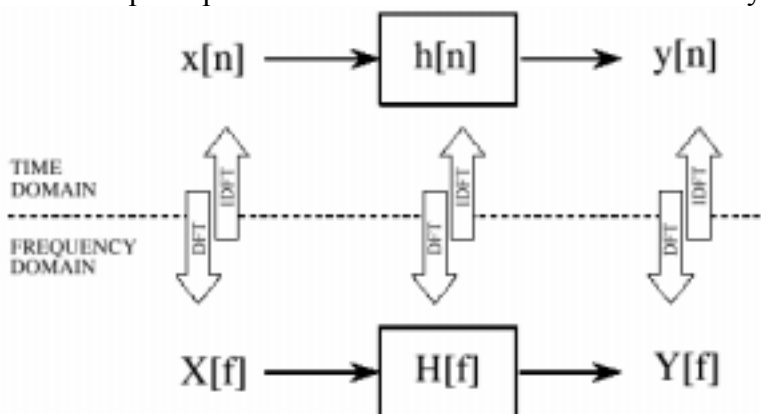


FIGURE 9-6

Comparing system operation in the time and frequency domains. In the time domain, an input signal is *convolved* with an impulse response, resulting in the output signal, that is, $x[n] * h[n] = y[n]$. In the frequency domain, an input spectrum is *multiplied* by a frequency response, resulting in the output spectrum, that is, $X[f] \times H[f] = Y[f]$. The DFT and the Inverse DFT relate the signals in the two domain.

РИСУНОК 9-6

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

Сравнение операции системны в доменах времени и частоты. В домене времени, входной сигнал *свернут*(скручен) с импульсной передаточной функцией, приводя к сигналу выхода, то есть $x[n] * h[n] = y[n]$. В частотном домене, входной спектр *умножен* на частотную характеристику, приводя к спектру выхода, то есть $X[f] \times H[f] = Y[f]$. ДПФ и Обратный ДПФ связывают сигналы в двух доменах.

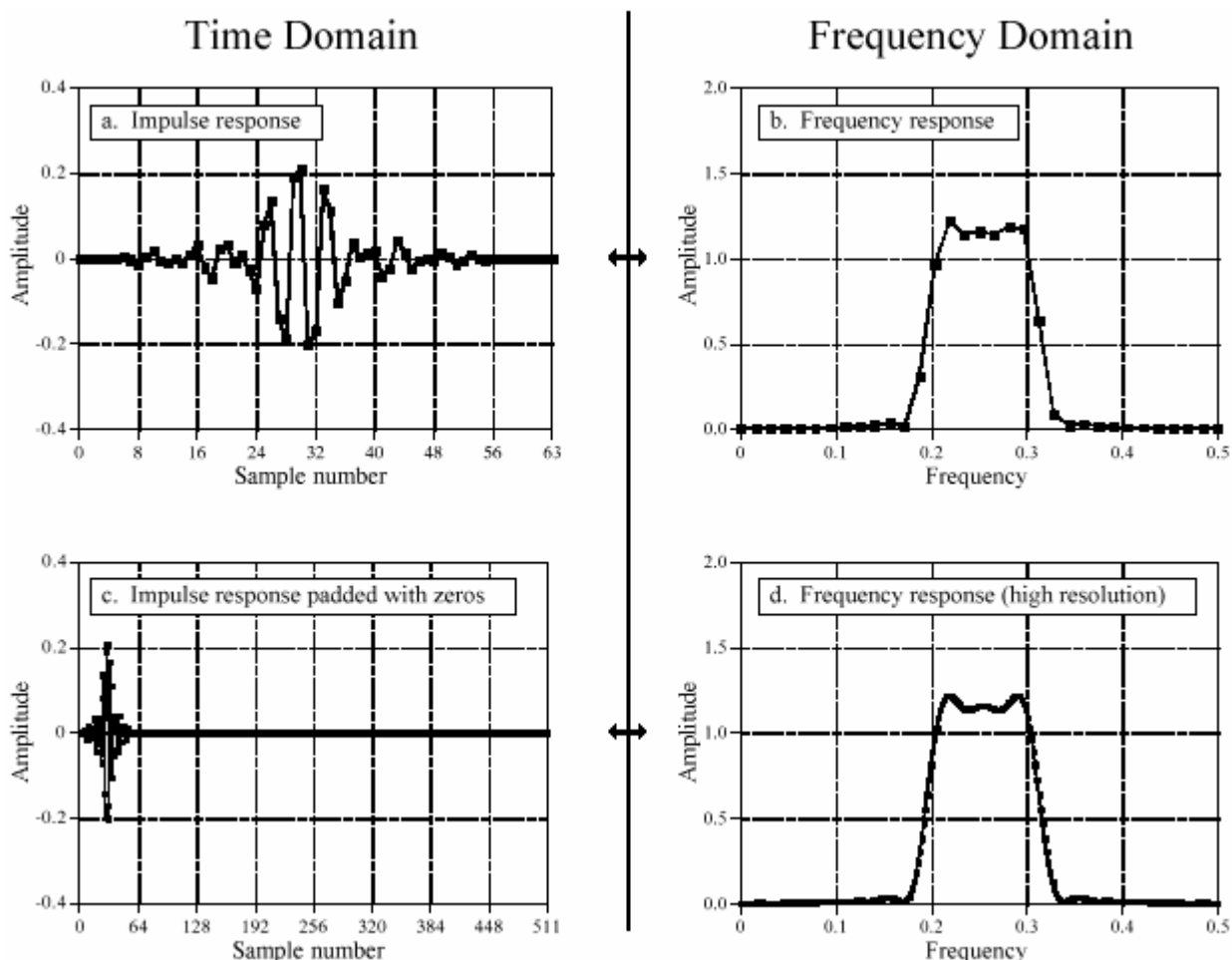


FIGURE 9-7 Finding the frequency response from the impulse response. By using the DFT, a system's impulse response, (a), can be transformed into the system's frequency response, (b). By padding the impulse response with zeros (c), higher resolution can be obtained in the frequency response, (d). Only the magnitude of the frequency response is shown in this example; discussion of the phase is postponed until the next chapter.

РИСУНОК 9-7

Нахождение частотной характеристики от импульсной передаточной функции. Используя ДПФ, импульсная передаточная функция системы, (a), может быть преобразована в частотную характеристику системы, (b). Дополняя импульсную передаточную функцию нулями (c), более высокая разрешающая способность может быть получена в частотной характеристике, (d). Только величина частотной характеристики показывается в этом примере; обсуждение фазы отложено до следующей главы.

This can be better understood by bringing in another member of the Fourier transform family, the **Discrete Time Fourier Transform (DTFT)**. Consider an N sample signal being run through an N point DFT, producing an $N/2+1$ sample frequency domain. Remember from the last chapter that the DFT considers the time domain signal to be *infinitely long* and *periodic*. That is, the N points are repeated over and over from negative to positive infinity. Now consider what happens when we start to pad the time domain signal with an ever increasing number of zeros, to obtain a finer and finer sampling in the frequency domain. Adding zeros makes the period of the time domain *longer*, while simultaneously making the frequency domain samples *closer together*.

Это может быть лучше понято, вводя другой элемент семейства анализа Фурье, **Дискретное Преобразование Фурье Времени (DTFT)**. Предположите сигнал выборки N , который пробегает N точек ДПФ, производя $N/2+1$ выборку частотного домена. Вспомните из прошлой главы, что ДПФ полагает, что сигнал домена времени будет *бесконечно длинным* и *периодическим*. То есть точки N повторены много раз от отрицательной до положительной бесконечности. Теперь рассмотрите то, что случается, когда мы начинаем дополнять сигнал домена времени при увеличении числа нулей, чтобы получить более тонкое и более тонкое(подробное?) осуществление выборки в частотном домене. Добавление нулей делают период из домена времени *дольше*, при одновременном создании выборки частотного домена *ближе вместе(чаще, ближе друг к другу)*.

Now we will take this to the extreme, by adding an *infinite* number of zeros to the time domain signal. This produces a different situation in two respects. First, the time domain signal now has an infinitely long period. In other words, it has turned into an *aperiodic* signal. Second, the frequency domain has achieved an infinitesimally small spacing between samples. That is, it has become a *continuous signal*. This is the DTFT, the procedure that changes a discrete aperiodic signal in the time domain into a frequency domain that is a continuous curve. In mathematical terms, a system's frequency response is found by taking the DTFT of its impulse response. Since this cannot be done in a computer, the DFT is used to calculate a *sampling* of the true frequency response. This is the difference between what you do in a computer (the DFT) and what you do with mathematical equations (the DTFT).

Теперь мы будем прибавлять *бесконечное* число нулей к сигналу домена времени до крайности. Это производит различную ситуацию в двух отношениях (касательствах). Во первых, сигнал домена времени теперь имеет бесконечно длительный период. Другими словами, превращается в аperiodический сигнал. Во вторых, частотный домен достиг бесконечно маленького интервала между выборками. То есть стал непрерывным сигналом. Это – DTFT, процедура, которая изменяет дискретный аperiodический сигнал в домене времени в частотный домен, который является непрерывной кривой. Выражаясь математически, частотная характеристика системы найдена, беря DTFT его импульсной передаточной функции. Так как это не может быть сделано в компьютере, для вычисления выборки из истинной частотной характеристики используется DTFT. Это - разница между тем, что Вы делаете в компьютере (ДПФ) и что Вы делаете с математическими уравнениями (DTFT).

Convolution via the Frequency Domain

Конволюция (скручивание, свертка) через Частотный Домен

Suppose that you despise convolution. What are you going to do if given an input signal and impulse response, and need to find the resulting output signal? Figure 9-8 provides an answer: transform the two signals into the frequency domain, multiply them, and then transform the result back into the time domain. This replaces one convolution with two DFTs, a multiplication, and an Inverse DFT. Even though the intermediate steps are very different, the output is *identical* to the standard convolution algorithm.

Предположим, что Вы презираете скручивание(свертку). Что Вы собираетесь делать, если даны входной сигнал и импульсная передаточная функция, и необходимо найти заканчивающийся сигнал выхода? Рисунок 9-8 обеспечивает ответ: преобразуйте два сигнала в частотный домен, (пере)умножьте их, и затем преобразуйте результат назад в домен времени. Это заменяет одно скручивание двумя ДПФ, умножение и Обратный ДПФ. Даже

НАУЧНО-ТЕХНИЧЕСКОЕ РУКОВОДСТВО ПО ЦИФРОВОЙ ОБРАБОТКЕ СИГНАЛОВ

при том, что промежуточные шаги очень различны, выход *идентичен* стандартному алгоритму скручивания(свертки).

Does anyone hate convolution enough to go to this trouble? The answer is yes. Convolution is avoided for two reasons. First, convolution is *mathematically* difficult to deal with. For instance, suppose you are given a system's impulse response, and its output signal. How do you calculate what the input signal is? This is called **deconvolution**, and is virtually impossible to understand in the time domain. However, deconvolution can be carried out in the frequency domain as a simple *division*, the inverse operation of multiplication. The frequency domain becomes attractive whenever the complexity of the Fourier Transform is less than the complexity of the convolution. This isn't a matter of which you like better; it is a matter of which you hate less.

Достаточно ли все (каждый, любой) ненавидят скручивание(свертку), чтобы идти к этой неприятности(затруднению)? Ответ - да. Скручивания избегают по двум причинам. Во первых, со скручиванием трудно иметь дело *математически*. Например, предположим, Вам дают импульсную передаточную функцию системы, и его сигнал выхода. Как Вы вычислите, каков входной сигнал? Это называется **деконволюцией**, и фактически невозможно понять в домене времени. Однако, деконволюция может быть выполнена в частотном домене как простое *деление*, операция обратная умножению. Частотный домен становится притягивающим(предпочтительным) всякий раз, когда сложность анализа Фурье – меньше чем сложность конволюции. Это не вопрос, что Вы любите больше; это – вопрос, что Вы ненавидите меньше.

The second reason for avoiding convolution is *computation speed*. For example, suppose you design a digital filter with a kernel (impulse response) containing 512 samples. Using a 200 MHz personal computer with floating point numbers, each sample in the output signal requires about one millisecond to calculate, using the standard convolution algorithm. In other words, the throughput of the system is only about 1,000 samples per second. This is 40 times too slow for high-fidelity audio, and 10,000 times too slow for television quality video!

Вторая причина для ухода от скручивания – *быстродействие вычисления*. Например, предположите, что Вы проектируете цифровой фильтр с ядром (импульсной передаточной функцией), содержащей 512 выборок. Используя персональный компьютер с тактовой частотой 200 МГц с числами плавающей запятой, используя стандартный алгоритм скручивания, каждая выборка в сигнале выхода требует для вычисления приблизительно одну миллисекунду. Другими словами, производительность системы - приблизительно только 1000 выборок в секунду. Это - 40 раз медленнее для высококачественного звуковоспроизведения, и в 10000 раз медленнее для высококачественного телевидения!

The standard convolution algorithm is slow because of the large number of multiplications and additions that must be calculated. Unfortunately, simply bringing the problem into the frequency domain via the DFT doesn't help at all. Just as many calculations are required to calculate the DFTs, as are required to directly calculate the convolution. A breakthrough was made in the problem in the early 1960s when the *Fast Fourier Transform* (FFT) was developed.

Стандартный алгоритм скручивания медленен из-за большого количества умножений и сложений, которые должны быть рассчитаны. К сожалению, просто принося проблему в частотный домен через ДПФ, не помогает вообще. Так много вычислений требуются, чтобы вычислить ДПФ, как требуются, чтобы непосредственно вычислить скручивание. Крупное достижение было сделано в проблеме в начале 1960-ых, когда было разработано *Быстрое Преобразование Фурье* (БПФ).

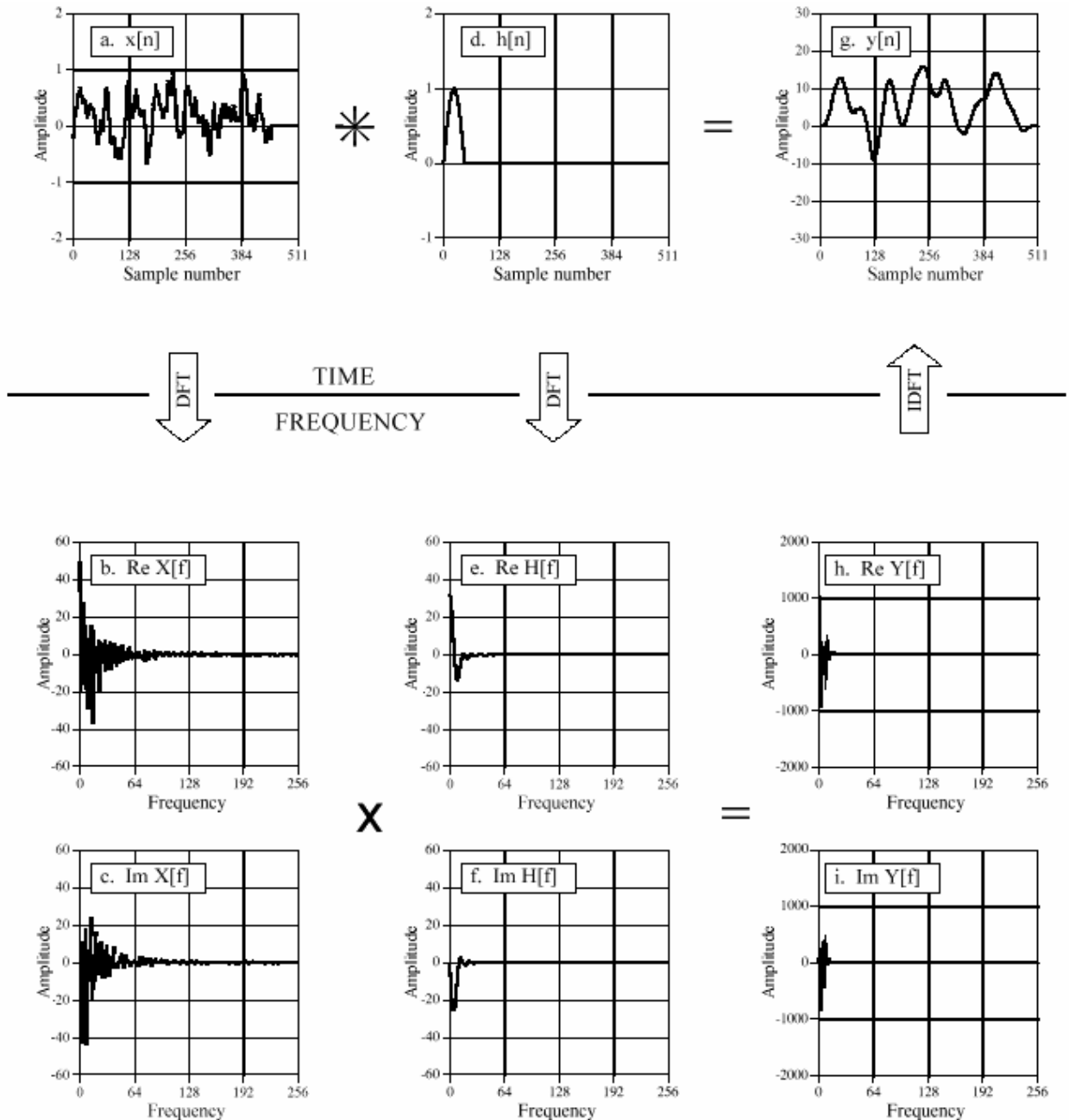


FIGURE 9-8

Frequency domain convolution. In the *time domain* $x[n]$, is convolved with $h[n]$ resulting in $y[n]$, as is shown in Figs. (a), (d), and (g). This same procedure to be accomplished in the *frequency domain*. The DFT is used to find the frequency spectrum of the input signal, (b) & (c), and the system's frequency response, (e) & (f). Multiplying these two frequency domain signals results in the frequency spectrum of the output signal, (h) & (i). The Inverse DFT is then used to find the output signal, (g).

РИСУНОК 9-8. Скручивание частотного домена.

В *домене времени*, $x[n]$, свернут(скручен) с $h[n]$ приводя в $y[n]$, как показано в рис. (a), (d), и (g). Это та же самая процедура, которая будет выполнена в частотном домене. ДПФ используется, чтобы найти спектр частот входного сигнала, (b) и (c), и частотной характеристики системы, (e) и (f). Умножение этих двух сигналов частотного домена приводит к спектру частот сигнала выхода, (h) и (i). Обратный ДПФ тогда используется, чтобы найти сигнал выхода, (g).

The FFT is a clever algorithm for rapidly calculating the DFT. Using the FFT, convolution by multiplication in the frequency domain can be hundreds of times faster than conventional convolution. Problems that take hours of calculation time are reduced to only minutes. This is why people get excited about the FFT, and processing signals in the frequency domain. The FFT will be presented in Chapter 12, and the method of FFT convolution in Chapter 18. For now, focus on how signals are convolved by frequency domain multiplication.

БПФ – искусный(умный, талантливый) алгоритм для быстро вычисления ДПФ. Используя скручивание БПФ умножением в частотном домене, может быть в сотни раз быстрее, чем обычное скручивание. Проблемы, которые занимают часы времени вычисления, сокращены только к минутам. Это - то, почему люди возбуждаются относительно БПФ, и обработки сигналов в частотном домене. БПФ будет представлено в главе 12, и методе скручивания БПФ в главе 18. Пока, фокус(сконцентрируем внимание) на том, как сигналы скручены умножением частотным доменом.

To start, we need to define how to multiply one frequency domain signal by another, i.e., what it means to write: $X[f] * H[f] = Y[f]$. In polar form, the magnitudes are multiplied: $MagY[f] = MagX[f] * MagH[f]$ and the phases, are added: $PhaseY[f] = PhaseX[f] + PhaseH[f]$. To understand this, imagine a cosine wave entering a system with some amplitude and phase. Likewise, the output signal is also a cosine wave with some amplitude and phase. The polar form of the frequency response directly describes how the two amplitudes are related and how the two phases are related.

Чтобы начать, мы должны определить, как умножить один сигнал частотного домена на другой, то есть то, что означает запись: $X[f] * H[f] = Y[f]$. В полярной форме, величины умножены: $MagY[f] = MagX[f] * MagH[f]$ и фазы, сложены $PhaseY[f] = PhaseX[f] + PhaseH[f]$: Чтобы понимать это, вообразите волну косинуса ввода системы с некоторой амплитудой и фазой. Аналогично, сигнал выхода – также волна косинуса с некоторой амплитудой и фазой. Полярная форма частотной характеристики непосредственно описывает, как эти две амплитуды связаны и как эти две фазы связаны.

When frequency domain multiplication is carried out in *rectangular form* there are cross terms between the real and imaginary parts. For example, a sine wave entering the system can produce both cosine and sine waves in the output. To multiply frequency domain signals in rectangular notation:

Когда умножение частотных доменов выполнено в *прямоугольной форме* имеются перекрестные термины между реальными(вещественными) и мнимыми(несобственными) частями. Например, волна синуса, вводимая в систему, может производить, и волну косинуса и волну синуса в выходе. Умножение сигналов частотного домена сообщают в прямоугольной системе обозначений:

EQUATION 9-1

Multiplication of frequency domain signals rectangular form: $Y[f] = X[f] \times H[f]$.

$$Re Y[f] = Re X[f] Re H[f] - Im X[f] Im H[f] \quad \text{in}$$

УРАВНЕНИЕ 9-1

Умножение сигналов частотного домена прямоугольной форме: $Y[f] = X[f] \times H[f]$.

$$Im Y[f] = Im X[f] Re H[f] + Re X[f] Im H[f] \quad \text{в}$$

Focus on understanding multiplication using *polar notation*, and the idea of cosine waves passing through the system. Then simply accept that these more elaborate equations result when the same operations are carried out in rectangular form. For instance, let's look at the *division* of one frequency domain signal by another. In polar form, the division of frequency domain signals is achieved by the inverse operations we used for multiplication. To calculate: $H[f] = Y[f] / X[f]$, divide the magnitudes and subtract the phases, i.e., $MagH[f] = MagY[f] / MagX[f]$, $PhaseH[f] = PhaseY[f] - PhaseX[f]$. In rectangular form this becomes:

Фокус(Сконцентрируем внимание) на понимании умножения, используя *полярную систему обозначений*, и идеи относительно волн косинуса, проходящих через систему. Тогда просто допустим, если те же самые операции выполнены в прямоугольной форме, они кончатся более сложными уравнениями. Например, давайте смотреть на деление одного сигнала частотного домена на другой. В полярной форме, деление сигналов частотного домена достигнуто операциями обратными тем, которые мы использовали для умножения. Вычислить: $H[f] = Y[f] / X[f]$ делите величины, и вычтите фазы, то есть, $MagH[f] = MagY[f] / MagX[f]$, $PhaseH[f] = PhaseY[f] - PhaseX[f]$. В прямоугольной форме это становится:

EQUATION 9-2

Division of frequency domain signals in rectangular form, where:

$$ReH[f] = \frac{ReY[f] ReX[f] + ImY[f] ImX[f]}{ReX[f]^2 + ImX[f]^2}$$

УРАВНЕНИЕ 9-2

Деление сигналов частотного домена в прямоугольной форме, где $H[f] = Y[f] / X[f]$.

$$ImH[f] = \frac{ImY[f] ReX[f] - ReY[f] ImX[f]}{ReX[f]^2 + ImX[f]^2}$$

Now back to frequency domain convolution. You may have noticed that we cheated slightly in Fig. 9-8. Remember, the convolution of an N point signal with an M point impulse response results in $N + M - 1$ an point output signal. We cheated by making the last part of the input signal all *zeros* to allow this expansion to occur. Specifically, (a) contains 453 nonzero samples, and (b) contains 60 nonzero samples. This means the convolution of the two, shown in (c), can fit comfortably in the 512 points provided.

Теперь назад к скручиванию частотного домена. Вы, возможно, заметили, что мы обманывали слегка в рис. 9-8. Помните, скручивание сигнала точки N с точкой M результата импульсной передаточной функции в точке $N+M-1$ сигнала выхода. Мы обмануты, делая последнюю часть входного сигнала всеми нулями, чтобы позволить это расширение, чтобы произойти. Определенно, (a) содержит 453 выборки отличные от нуля, и (b) содержит 60 выборок отличных от нуля. Это означает скручиванию этих двух, показанных в (c), может соответствовать удобно в 512 обеспеченных точках.

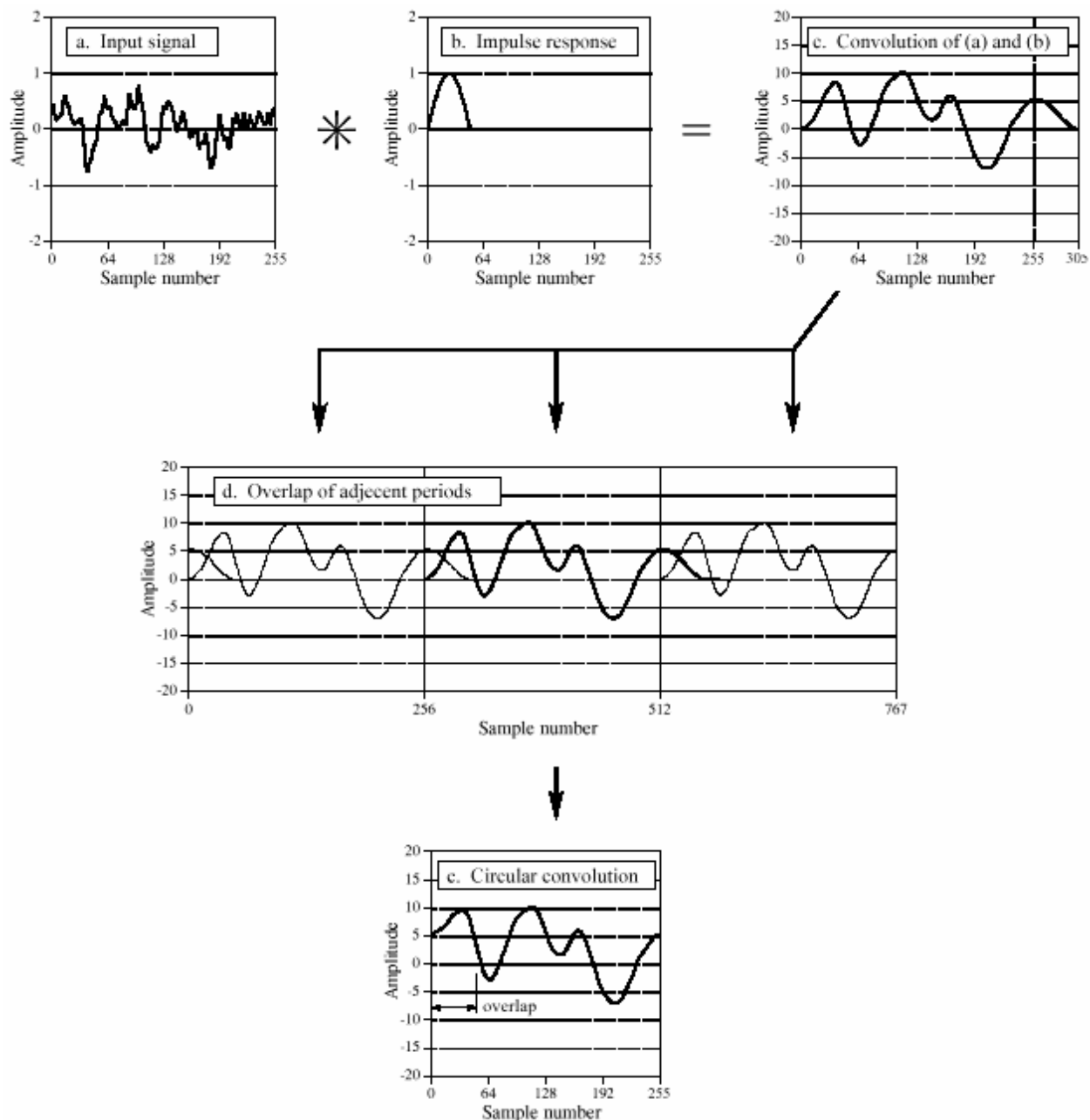


FIGURE 9-9

Circular convolution. A 256 sample signal, (a), convolved with a 51 sample impulse response, (b), results in a 306 sample signal, (c). If this convolution is performed in the frequency domain using 256 point DFTs, the 306 points in the correct convolution cannot fit into the 256 samples provided. As shown in (d), samples 256 through 305 of the output signal are pushed into the next period to the right, where they *add* to the beginning of the next period's signal. Figure (e) is a single period of the resulting signal.

РИСУНОК 9-9. Круговое скручивание(свертка).

256 выборок сигнала, (а), скрученные с 51 выборкой импульсной передаточной функции, (b), приводят к 306 выборкам сигнала, (c). Если это скручивание выполнено в частотном домене, используя 256 точек ДПФ, 306 точек в правильном скручивании не могут вписываться в 256 обеспеченных выборок. Как показано в (d), 256 выборок через 305 сигнала выхода помещены в следующий период направо, где они прибавляются к началу следующего периода сигнала. Рисунок (e) - единственный(отдельный) период заканчивающегося сигнала.

Now consider the more general case in Fig. 9-9. The input signal, (a), is 256 points long, while the impulse response, (b), contains 51 nonzero points. This makes the convolution of the two signals 306 samples long, as shown in (c). The problem is, if we use frequency domain multiplication to perform the convolution, there are only 256 samples *allowed* in the output signal. In other words, 256 point DFTs are used to move (a) and (b) into the frequency domain. After the multiplication, a 256 point Inverse DFT is used to find the output signal. How do you squeeze 306 values of the correct signal into the 256 points provided by the frequency domain algorithm? The answer is, you can't! The 256 points end up being a distorted version of the correct signal. This process is called **circular convolution**. It is important because you want to *avoid* it.

Теперь рассмотрите большее количество общего случая(более общий случай?) в рис. 9-9. Входной сигнал, (a), имеет длительность 256 точек, в то время как импульсная передаточная функция, (b), содержит 51 точку отличную от нуля. Это делает скручивание из двух сигналов длительностью 306 выборок, как показано в (c). Проблема, если мы используем умножение частотных доменов, чтобы выполнить скручивание, имеются только 256, *позволенных* в сигнале выхода, выборок. Другими словами, Чтобы переместить (a) и (b) в частотный домен, используется 256 точек ДПФ. После умножения, используется 256 точек Обратного(Инверсного) ДПФ, чтобы найти сигнал выхода. Как Вы сожмете 306 значений правильного сигнала в 256 точек, обеспеченных алгоритмом частотного домена? Ответ, Вы не можете! 256 точек заканчиваются являющийся искаженной версией правильного сигнала. Этот процесс называется круговым скручиванием. Это важно, потому что Вы хотите *избегать* этого.

To understand circular convolution, remember that an N point DFT views the time domain as being an infinitely long periodic signal, with N samples per period. Figure (d) shows three periods of how the DFT views the output signal in this example. Since $N = 256$, each period consists of 256 points: 0-255, 256-511, and 512-767. Frequency domain convolution tries to place the 306 point *correct output signal*, shown in (c), into each of these 256 point periods. This results in 49 of the samples being pushed into the neighboring period to the right, where they overlap with the samples that are legitimately there. These overlapping sections add, resulting in each of the periods appearing as shown in (e), the *circular convolution*.

Для понимания кругового скручивания, помните, что точку N , ДПФ рассматривает доменом времени, как являющимся бесконечно длинным периодическим сигналом, с N выборками на период. Рисунок (d) показывает три периода того, как ДПФ рассматривает сигнал выхода в этом примере. С тех пор как(если) $N = 256$, каждый период состоит из 256 точек: 0-255, 256-511, и 512-767. Скручивание частотного домена пробует разместить 306 точек *правильного сигнала выхода*, показанного в (c), в каждом периоде из 256 точек. Это приводит к 49 из выборкам, помещаемым в соседний период направо, где они накладываются с выборками, которые находятся там законно. Эти разделы перекрывания складываются, приводя к каждому из периодов, появляющихся как показано в (e), *кругового скручивания*.

Once the nature of circular convolution is understood, it is quite easy to avoid. Simply pad each of the signals being convolved with enough zeros to allow the output signal room to handle the $N+M-1$ points in the correct convolution. For example, the signals in (a) and (b) could be padded with zeros to make them 512 points long, allowing the use of 512 point DFTs. After the frequency domain convolution, the output signal would consist of 306 nonzero samples, plus 206 samples with a value of zero. Chapter 18 explains this procedure in detail.

Как только характер(природа) кругового скручивания понят, весьма просто избежать. Просто дополните каждый из скручиваемых сигналов, достаточным количеством нулей, позволяя возможности сигналу выхода обрабатывать $N+M-1$ точек в правильном скручивании. Например, сигналы в (а) и (b) могут дополняться нулями, чтобы делать их 512 точками длиной, позволяя использование 512 точек ДПФ. После скручивания частотного домена, сигнал выхода состоял бы из 306 выборок отличных от нуля, плюс 206 выборок со значением нуля. Глава 18 объясняет эту процедуру подробно.

Why is it called *circular* convolution? Look back at Fig. 9-9d and examine the center period, samples 256 to 511. Since all of the periods are the same, the portion of the signal that flows out of this period to the *right*, is the same that flows into this period from the *left*. If you only consider a single period, such as in (e), it *appears* that the right side of the signal is somehow *connected* to the left side. Imagine a snake biting its own tail; sample 255 is located next to sample 0, just as sample 100 is located next to sample 101. When a portion of the signal exits to the right, it magically reappears on the left. In other words, the N point time domain behaves as if it were *circular*.

Почему это называется *круговым* скручиванием? Оглянемся назад на рис. 9-9d и исследуем средний период, выборки от 256 до 511. Так как все периоды - те же самые(одинаковые), часть сигнала, которая вытекает из этого периода *вправо*, - та же самая, которая втекает в этот период *слева*. Если Вы рассматриваете только единственный период, типа в (e), *видно*, что правая сторона сигнала так или иначе *связана* с левой стороной. Вообразите змею, кусающую свой собственный хвост; выборка 255 расположена, рядом с выборкой 0, так же, как выборка 100, расположена, рядом с выборкой 101. Когда часть сигнала выходит справа, это волшебным образом вновь появляется слева. Другими словами, точка N , домена времени ведет себя, как будто это было круговое.

In the last chapter we posed the question: does it really matter if the DFT's time domain is viewed as being N points, rather than an infinitely long periodic signal of period N ? Circular convolution is an example where it *does* matter. If the time domain signal is understood to be *periodic*, the distortion encountered in circular convolution can be simply explained as the signal expanding from one period to the next. In comparison, a rather bizarre conclusion is reached if only N points of the time domain are considered.

That is, frequency domain convolution acts as if the time domain is somehow wrapping into a circular ring with sample 0 being positioned next to sample $N-1$.

В прошлой главе мы изложили вопрос: действительно ли имеет значение, что домен времени ДПФ рассмотрен как являющийся N точками, скорее, чем бесконечно длинный периодическим сигналом периодов N ? Круговое скручивание - пример, где это *имеет* значение. Если сигнал домена времени, как понимают, является *периодическим*, искажение, с которым сталкиваются в круговом скручивании, можно просто объяснить как сигнал, расширяющийся с одного периода к следующему. На сравнении, довольно причудливое заключение достигнуто, если только рассматриваются N точек домена времени. То есть скручивание частотного домена действует, как будто домен времени как то обертывает в круговое кольцо с выборкой 0 позиционируемой рядом с выборкой $N-1$.