

CHAPTER  
17

## Custom Filters Заказные Фильтры

Most filters have one of the four standard frequency responses: low-pass, high-pass, band-pass or band-reject. This chapter presents a general method of designing digital filters with an *arbitrary* frequency response, tailored to the needs of your particular application. DSP excels in this area, solving problems that are far above the capabilities of analog electronics. Two important uses of custom filters are discussed in this chapter: *deconvolution*, a way of restoring signals that have undergone an unwanted convolution, and *optimal filtering*, the problem of separating signals with overlapping frequency spectra. This is DSP at its best.

Большинство фильтров имеет одну из четырех стандартных частотных характеристик: фильтр нижних частот, фильтр верхних частот, полосовой или полосовой(заграждающий). Эта глава представляет общий метод проектирования цифровых фильтров с произвольной частотной характеристикой, приспособленной к потребностям вашего специфического приложения. ЦОС превосходит других в этой области, решая проблемы, которые являются далеко выше возможностей аналоговой электроники. Два важных использования заказных фильтров обсуждены в этой главе: *деконволюция*, путь восстановления сигналов, которые подверглись нежелательному скручиванию(свертке), и *оптимальной фильтрации*, проблема отделения сигналов с перекрыванием на частотные спектры. Это - ЦОС в лучшем случае.

### Arbitrary Frequency Response Произвольная Частотная характеристика

The approach used to derive the windowed-sinc filter in the last chapter can also be used to design filters with virtually *any* frequency response. The only difference is how the desired response is moved from the frequency domain into the time domain. In the windowed-sinc filter, the frequency response and the filter kernel are both represented by *equations*, and the conversion between them is made by evaluating the *mathematics* of the Fourier transform. In the method presented here, both signals are represented by *arrays of numbers*, with a *computer program* (the FFT) being used to find one from the other.

Подход используемый чтобы получать windowed-sinc фильтр в последней главе может также использоваться, чтобы проектировать фильтры с фактически любой частотной характеристикой. Единственная разность - то, как желательный ответ перемещен от частотного домена в домен времени. В фильтре windowed-sinc, частотная характеристика и ядро фильтра оба представлены *уравнениями*, и преобразование между ними сделано, оценивая *математику* преобразования Фурье. В методе, представленном здесь, оба сигнала представлены *массивами чисел*, с *компьютерной программой* (БПФ) используемой, чтобы найти одно от другого.

Figure 17-1 shows an example of how this works. The frequency response we want the filter to produce is shown in (a). To say the least, it is very irregular and would be virtually impossible to obtain with analog electronics. This ideal frequency response is *defined* by an array of numbers

that have been selected, not some mathematical equation. In this example, there are 513 samples spread between 0 and 0.5 of the sampling rate. More points could be used to better represent the desired frequency response, while a smaller number may be needed to reduce the computation time during the filter design. However, these concerns are usually small, and 513 is a good length for most applications.

Рисунок 17-1 показывает пример того, как это работает. Частотная характеристика, которую мы хотим, чтобы фильтр произвел, показывается в (а). Чтобы меньше говорить, это очень неправильно и было бы фактически невозможным *получить(определить)* с аналоговой электроникой. Эта идеальная частотная характеристика определена массивом чисел(номеров), которые были отобраны, не некоторое математическое уравнение. В этом примере, имеется 513 выборки распространенных между 0 и 0.5 частоты выборки. Большее количество точек могло использоваться, чтобы лучше представить желательную частотную характеристику, в то время как меньшее число может быть необходимо, чтобы привести время вычисления в течение проекта фильтра. Однако, эти беспокойства обычно маленькие, и 513 - хорошая длина для большинства приложений.

Besides the desired *magnitude* array shown in (a), there must be a corresponding *phase* array of the same length. In this example, the phase of the desired frequency response is entirely *zero* (this array is not shown in Fig. 17-1). Just as with the magnitude array, the phase array can be loaded with any arbitrary curve you would like the filter to produce. However, remember that the first and last samples (i.e., 0 and 512) of the phase array must have a value of *zero* (or a multiple of  $2\pi$ , which is the same thing). The frequency response can also be specified in rectangular form by defining the array entries for the *real* and *imaginary parts*, instead of using the magnitude and phase.

Помимо желательного массива *величин*, показанного в (а), должен иметься соответствующий *фазовый* массив той же самой длины. В этом примере, фаза желательной частотной характеристики полностью нулевая (этот массив не показывается в рис. 17-1). Так же, как с массивом величин, фазовый массив может быть загружен любой произвольной кривой, которую Вы хотели бы, чтобы фильтр произвел. Однако, помните, что первые и последние выборки (то есть, 0 и 512) фазового массива должны иметь значение *нуля* (или множитель  $2\pi$ , который является той же самой вещью). Частотная характеристика может также быть определена в прямоугольной форме, определяя входы массива для *вещественных* и *мнимых частей*, вместо использования величины и фазы.

The next step is to take the Inverse DFT to move the filter into the time domain. The quickest way to do this is to convert the frequency domain to rectangular form, and then use the Inverse FFT. This results in a 1024 sample signal running from 0 to 1023, as shown in (b). This is the impulse response that corresponds to the frequency response we want; however, it is not suitable for use as a filter kernel (more about this shortly). Just as in the last chapter, it needs to be *shifted, truncated, and windowed*. In this example, we will design the filter kernel with  $M = 40$ , i.e., 41 points running from sample 0 to sample 40. Table 17-1 shows a computer program that converts the signal in (b) into the filter kernel shown in (c). As with the windowed-sinc filter, the points near the ends of the filter kernel are so small that they appear to be zero when plotted. Don't make the mistake of thinking they can be deleted!

Следующий шаг - брать Обратный ДПФ, чтобы переместить фильтр в домен времени. Самый быстрый способ сделать это состоит в том, чтобы преобразовать частотный домен к прямоугольной форме, и затем использовать Обратное БПФ. Это приводит к 1024 выборкам сигнала, выполняющимся от 0 до 1023, как показано в (b). Это - импульсная передаточная функция, которая соответствует частотной характеристике, которую мы хотим; однако, это не подходящее для использования как ядро фильтра (больше относительно этого вскоре). Также, как в последней главе, требуется быть *сдвинуть, усечь, и обработать(реализовать) методом окна*. В этом примере, мы будем проектировать ядро фильтра с  $M = 40$ , то есть, 41 точкой выполняемых от выборки 0, до выборки 40. Таблица 17-1 показывает компьютерную программу, которая преобразовывает сигнал в (b) в ядро фильтра, показанное в (c). Как с фильтром windowed-sinc, точки около концов ядра фильтра настолько маленькие, что они, кажутся, нулевыми когда составляется график. Не делайте ошибку из размышления их, возможного удаления!

```

100          'CUSTOM FILTER DESIGN
110          'This program converts an aliased 1024 point impulse response into an M+1 point
120          'filter kernel (such as Fig. 17-1b being converted into Fig. 17-1c)
130          '
140 DIM REX[1023]          'REX[ ] holds the signal being converted
150 DIM T[1023]           'T[ ] is a temporary storage buffer
160 '
170 PI = 3.14159265
180 M% = 40                'Set filter kernel length (41 total points)
190 '
200 GOSUB XXXX            'Mythical subroutine to load REX[ ] with impulse response
210 '
220 FOR I% = 0 TO 1023    'Shift (rotate) the signal M/2 points to the right
230 INDEX% = I% + M%/2
240 IF INDEX% > 1023 THEN INDEX% = INDEX% - 1024
250 T[INDEX%] = REX[I%]
260 NEXT I%
270 '
280 FOR I% = 0 TO 1023
290 REX[I%] = T[I%]
300 NEXT I%
310 '                    'Truncate and window the signal
320 FOR I% = 0 TO 1023
330 IF I% <= M% THEN REX[I%] = REX[I%] * (0.54 - 0.46 * COS(2*PI*I%/M%))
340 IF I% > M% THEN REX[I%] = 0
350 NEXT I%
360 '                    'The filter kernel now resides in REX[0] to REX[40]
370 END

```

TABLE 17-1

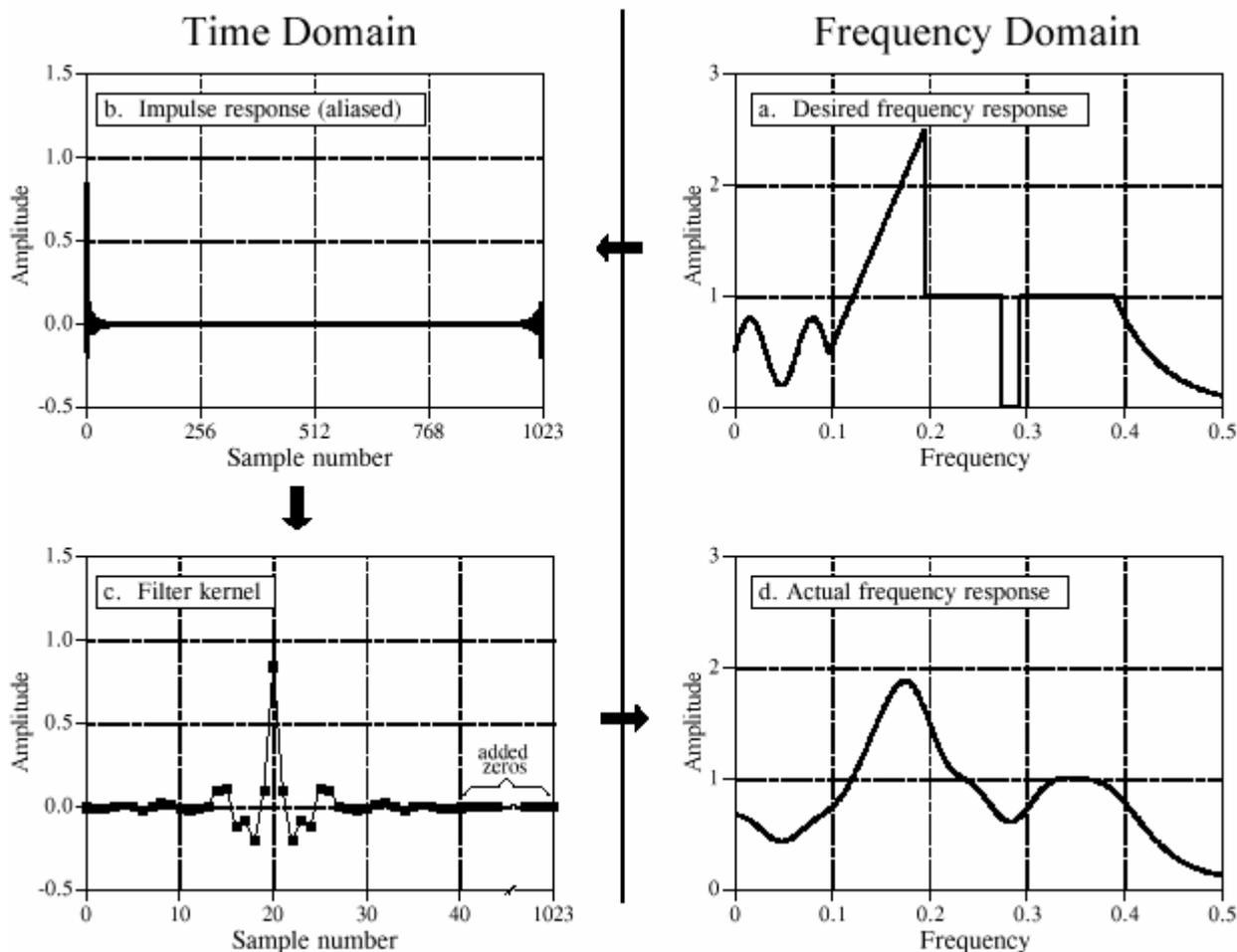


FIGURE 17-1. Example of FIR filter design.

Figure (a) shows the desired frequency response, with 513 samples running between 0 to 0.5 of the sampling rate. Taking the Inverse DFT results in (b), an *aliased* impulse response composed of 1024 samples. To form the filter kernel, (c), the aliased impulse response is truncated to  $M + 1$  samples, shifted to the right by samples, and multiplied by a Hamming or Blackman window. In this  $M/2$  example,  $M$  is 40. The program in Table 17-1 shows how this is done. The filter kernel is tested by padding it with zeros and taking the DFT, providing the actual frequency response of the filter, (d).

РИСУНОК 17-1. Пример проекта КИХ-ФИЛЬТРА.

Рисунок (а) показывает желательную частотную характеристику, с 513 выборками, выполняющимися между 0 до 0.5 частоты выборки. Взятие Обратного ДПФ приводит (к (b), *смешанная*(эффект наложения?) импульсная передаточная функция, составленная из 1024 выборок. Чтобы формировать ядро фильтра, (c), смешанная импульсная передаточная функция усечена к  $M + 1$  выборкам, сдвинута направо  $M/2$  выборками, и умножена окном Хэмминга или Блэкума. В этом примере,  $M$  есть 40. Программа в таблице 17-1 показывает, как это сделано. Ядро фильтра проверено, дополняя это нулями и беря ДПФ, обеспечивая фактическую частотную характеристику фильтра, (d).

The last step is to *test* the filter kernel. This is done by taking the DFT (using the FFT) to find the actual frequency response, as shown in (d). To obtain better resolution in the frequency domain, pad the filter kernel with zeros before the FFT. For instance, using 1024 total samples (41 in the filter kernel, plus 983 zeros), results in 513 samples between 0 and 0.5.

Последний шаг должен проверить ядро фильтра. Это сделано, беря ДПФ (используя БПФ) чтобы найти фактическую частотную характеристику, как показано в (d). Чтобы получить лучшую разрешающую способность в частотном домене, дополните ядро фильтра нулями перед БПФ. Например, использование 1024 полных выборок (41 в ядре фильтра, плюс 983 нуля), приводят к 513 выборкам между 0 и 0.5.

As shown in Fig. 17-2, the length of the filter kernel determines how well the *actual* frequency response matches the *desired* frequency response. The exceptional performance of FIR digital filters is apparent; virtually any frequency response can be obtained if a long enough filter kernel is used.

Как показано в рис. 17-2, длина ядра фильтра определяет, как хорошо *фактическая* частотная характеристика соответствует *желательной* частотной характеристике. Исключительная эффективность цифровых фильтров с КИХ очевидна; фактически любая частотная характеристика может быть получена, если используется достаточно длинное ядро фильтра.

This is the entire design method; however, there is a subtle *theoretical* issue that needs to be clarified. Why isn't it possible to directly use the impulse response shown in 17-1b as the filter kernel? After all, if (a) is the Fourier transform of (b), wouldn't convolving an input signal with (b) produce the *exact* frequency response we want? The answer is *no*, and here's why.

Это - полный метод проекта; однако, имеется тонкая *теоретическая* проблема, которая должна быть разъяснена. Почему не возможно непосредственно использовать импульсную передаточную функцию, показанную в 17-1b как ядро фильтра? В конце концов, если бы (a) - преобразование Фурье из (b), не было бы скручиванием входного сигнала с (b), произведет *точную* частотную характеристику, которую мы хотим? Ответ - *нет*, и есть почему.

When designing a custom filter, the desired frequency response is defined by the values in an array. Now consider this: what does the frequency response do *between* the specified points? For simplicity, two cases can be imagined, one "good" and one "bad." In the "good" case, the frequency response is a smooth curve between the defined samples. In the "bad" case, there are wild fluctuations between. As luck would have it, the impulse response in (b) corresponds to the "bad" frequency response. This can be shown by padding it with a large number of zeros, and then taking the DFT. The frequency response obtained by this method will show the erratic behavior between the originally defined samples, and look just awful.

При проектировании заказного фильтра, желательная частотная характеристика определена значениями в массиве. Теперь рассмотрите это: что частотная характеристика делает между указанными точками? Для простоты, два случая могут предполагаться, один "хороший" и один "плохой". В "хорошем" случае, частотная характеристика - гладкая кривая *между* определенными выборками. В "плохом" случае, между точками имеется дикое флуктуации. В лучшем случае эта импульсная передаточная функция в (b) будет соответствовать "плохой" частотной характеристике. Это можно показать, дополняя это большим количеством нулей, и затем беря ДПФ. Частотная характеристика, полученная этим методом, покажет ошибочное поведение между первоначально определенными выборками, и выглядеть только ужасной.

To understand this, imagine that we force the frequency response to be what we want by defining it at an infinite number of points between 0 and 0.5. That is, we create a continuous curve. The inverse DTFT is then used to find the impulse response, which will be *infinite* in length. In other words, the "good" frequency response corresponds to something that cannot be represented in a computer, an infinitely long impulse response. When we represent the frequency spectrum with  $N/2+1$  samples, only  $N$  points are provided in the time domain, making it unable to correctly contain the signal. The result is that the infinitely long impulse response wraps up (aliases) into

the  $N$  points. When this aliasing occurs, the frequency response changes from "good" to "bad." Fortunately, windowing the  $N$  point impulse response greatly reduces this aliasing, providing a smooth curve between the frequency domain samples.

Чтобы понимать это, вообразите, что мы вынуждаем частотную характеристику быть тем, что мы хотим, определяя это в бесконечном числе точек между 0 и 0.5. То есть мы создаем непрерывную кривую. Инверсия DTFT тогда используется, чтобы найти импульсную передаточную функцию, которая будет *бесконечна* в длине. Другими словами, "хорошая" частотная характеристика соответствует кое-чему, что не может быть представлено в компьютере, бесконечно длинной импульсной передаточной функцией. Когда мы представляем спектр частот с  $N/2+1$  выборками, только  $N$  точки - обеспечены в домене времени, делая это неспособным правильно содержать сигнал. Результат - то, что бесконечно длинная импульсная передаточная функция обертывает(оборачивает) (псевдонимы; ложные частоты) в точки  $N$ . Когда это наложение спектров происходит, частотная характеристика изменяется от "хорошей" до "плохой". К счастью, работа с окнами импульсной передаточной функции точки  $N$  очень приводит это наложение спектров, обеспечивая гладкую кривую между выборками частотного домена.

Designing a digital filter to produce a given frequency response is quite simple. The hard part is finding what frequency response to use. Let's look at some strategies used in DSP to design custom filters.

Проектирование цифрового фильтра, чтобы произвести данную частотную характеристику весьма просто. Жесткая часть – обнаружить(заключить; сделать заключение) какую частотную характеристику использовать. Давайте рассмотрим некоторые стратегии, используемые в ЦОС, чтобы проектировать заказные фильтры.

## **Deconvolution**

### **Деконволюция**

Unwanted convolution is an inherent problem in transferring analog information. For instance, all of the following can be modeled as a convolution: image blurring in a shaky camera, echoes in long distance telephone calls, the finite bandwidth of analog sensors and electronics, etc. Deconvolution is the process of filtering a signal to compensate for an undesired convolution. The goal of deconvolution is to recreate the signal as it existed *before* the convolution took place. This usually requires the characteristics of the convolution (i.e., the impulse or frequency response) to be known. This can be distinguished from **blind deconvolution**, where the characteristics of the parasitic convolution are *not* known. Blind deconvolution is a much more difficult problem that has no general solution, and the approach must be tailored to the particular application.

Нежелательная свертка - свойственная проблема в передаче аналоговой информации. Например, все следующее может быть смоделировано как свертка: размывание изображения(образа) в шаткой камере, эхо(отражение; микрофонный эффект?) в телефонных звонках на длинные расстояния, конечная ширина полосы частот аналоговых датчиков и электроники, и т.д. Деконволюция(обращение свертки) - процесс фильтрации сигнала, чтобы компенсировать нежелательную свертку. Цель деконволюции состоит в том, чтобы освежить сигнал, поскольку это существовало *прежде*, чем свертка имело место. Это обычно требует чтобы были известны характеристики свертки (то есть, импульсная или частотная характеристики). Это может отличить (деконволюцию?) от **слепой деконволюции**, где

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

характеристики паразитной свертки *не известны*. Слепая деконволюция - намного более трудная проблема, которая не имеет никакого общего решения, и подход должен быть приспособлен к специфическому приложению.

Deconvolution is nearly impossible to understand in the *time domain*, but quite straightforward in the *frequency domain*. Each sinusoid that composes the original signal can be changed in amplitude and/or phase as it passes through the undesired convolution. To extract the original signal, the deconvolution filter must *undo* these amplitude and phase changes. For example, if the convolution changes a sinusoid's amplitude by 0.5 with a 30 degree phase shift, the deconvolution filter must amplify the sinusoid by 2.0 with a -30 degree phase change.

Деконволюцию почти невозможно понять в домене времени, но весьма прямо в частотном домене. Каждая синусоида, которая составляет первоначальный сигнал, может быть изменена в амплитуде и-или фазе, поскольку это проходит через нежелательную свертку. Чтобы извлечь первоначальный сигнал, фильтр деконволюции должен отменить эту амплитуду и изменение фазы. Для примера, Если свертка изменяет амплитуду синусоиды 0.5 с 30 градусами сдвига фазы, фильтр деконволюции должен усилить синусоиду 2.0 с -30 градусами изменения фазы.

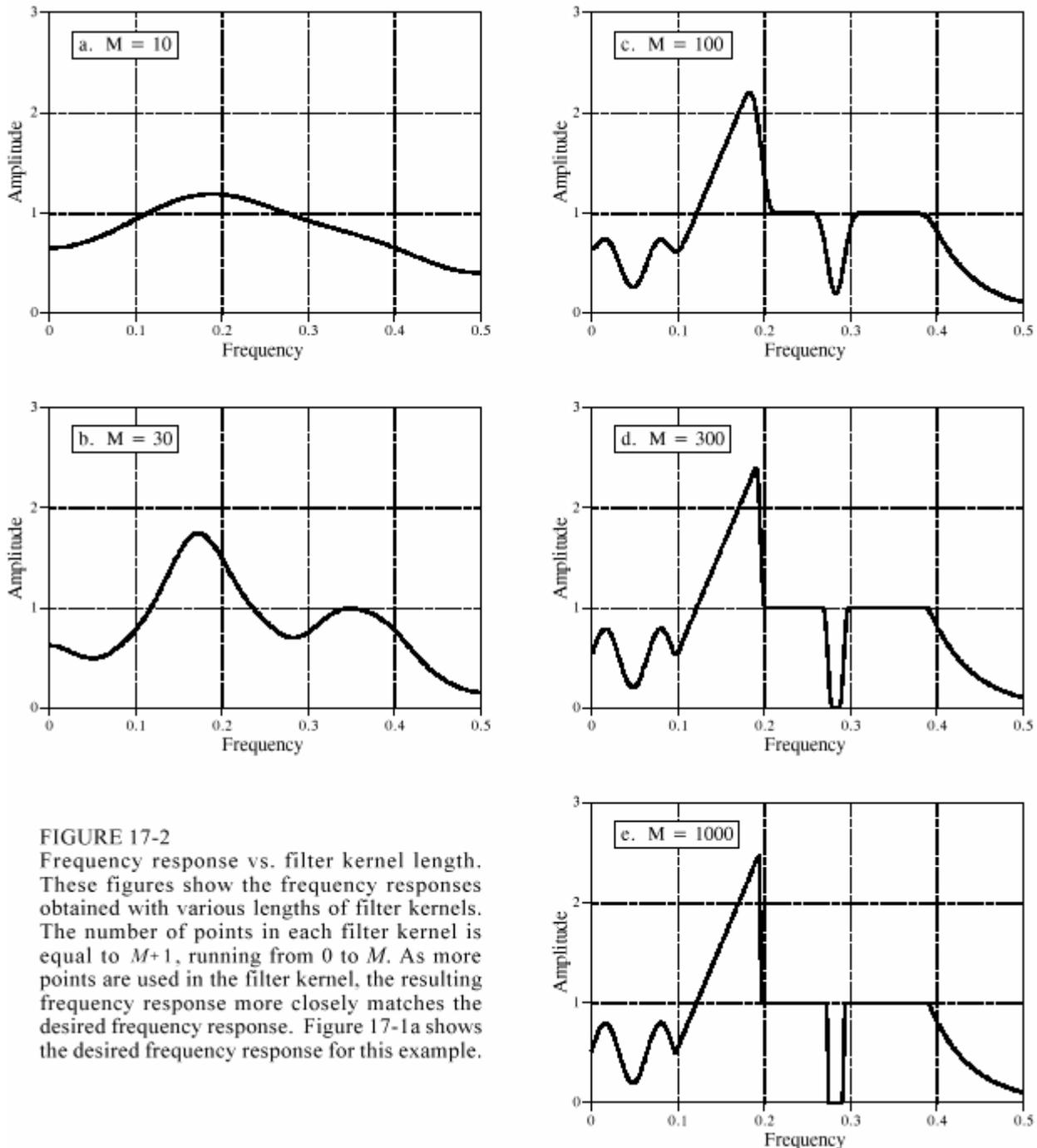


FIGURE 17-2  
 Frequency response vs. filter kernel length. These figures show the frequency responses obtained with various lengths of filter kernels. The number of points in each filter kernel is equal to  $M+1$ , running from 0 to  $M$ . As more points are used in the filter kernel, the resulting frequency response more closely matches the desired frequency response. Figure 17-1a shows the desired frequency response for this example.

FIGURE 17-2  
 Frequency response vs. filter kernel length. These figures show the frequency responses obtained with various lengths of filter kernels. The number of points in each filter kernel is equal to  $M + 1$  running from 0 to  $M$ . As more, points are used in the filter kernel, the resulting frequency response more closely matches the desired frequency response. Figure 17-1a shows the desired frequency response for this example.

РИСУНОК 17-2.  
 Частотная характеристика против длины ядра фильтра. Эти рисунки показывают частотные характеристики, полученные с различными длинами ядер фильтра. Число точек в каждом ядре фильтра равно от  $M + 1$  выполняемому от 0 до  $M$ . Как большее количество точек используются в ядре фильтра, заканчивающаяся частотная характеристика более близко соответствует желательному частотному ответу(частотной характеристике). Рисунок 17-1а показывает желательную частотную характеристику для этого примера.

The example we will use to illustrate deconvolution is a *gamma ray detector*. As illustrated in Fig. 17-3, this device is composed of two parts, a *scintillator* and a *light detector*. A scintillator is a special type of transparent material, such as sodium iodide or bismuth germanate. These com-

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

pounds change the energy in each gamma ray into a brief burst of visible light. This light is then converted into an electronic signal by a light detector, such as a photodiode or photomultiplier tube. Each pulse produced by the detector resembles a *one-sided exponential*, with some rounding of the corners. This shape is determined by the characteristics of the scintillator used. When a gamma ray deposits its energy into the scintillator, nearby atoms are excited to a higher energy level. These atoms randomly *deexcite*, each producing a single photon of visible light. The net result is a light pulse whose amplitude decays over a few hundred nanoseconds (for sodium iodide). Since the arrival of each gamma ray is an *impulse*, the output pulse from the detector (i.e., the one-sided exponential) is the *impulse response* of the system.

Пример, который мы будем использовать, чтобы иллюстрировать деконволюцию - *датчик гамма лучей*. Как иллюстрировано в рис. 17-3, это устройство составлено из двух частей, *сцинтиллятора* и *светоприемника*. Сцинтиллятор - специальный тип прозрачного материала, типа *иодида натрия* или *германата висмута*. Эти составы изменяют энергию каждого гамма луча в краткий световой импульс(вспышку). Этот световой импульс тогда преобразован в электронный сигнал светоприемником, типа лампы фотоумножителя или фотодиода. Каждый импульс, произведенный датчиком походит на *одностороннюю показательную функцию*, с некоторым округлением углов. Эта форма определена в соответствии с характеристиками используемого сцинтиллятора. Когда гамма луч вносит его энергию в сцинтиллятор, близлежащие атомы возбуждены к более высокому энергетическому уровню. Эти атомы беспорядочно дезактивируются, при это каждый создает единственный(отдельный) фотон видимого светового излучения(вспышку). Схемный результат - световой импульс, чья амплитуда распадается более чем несколько сотен наносекунд (для иодида натрия). Так как прибытие каждого гамма луча - импульс, импульс выхода от датчика (то есть, односторонняя показательная функция) - импульсная передаточная функция системы.

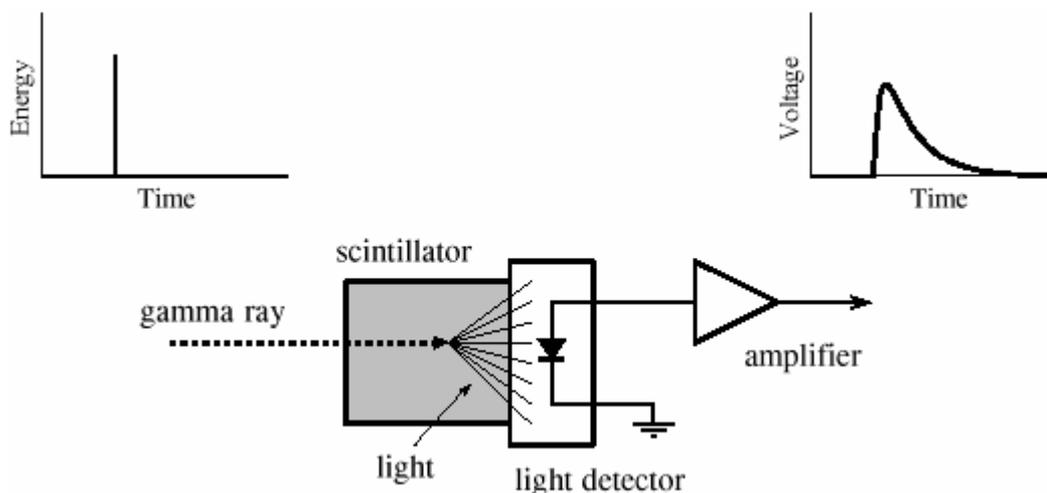


FIGURE 17-3

Example of an unavoidable convolution. A gamma ray detector can be formed by mounting a *scintillator* on a *light detector*. When a gamma ray strikes the scintillator, its energy is converted into a pulse of light. This pulse of light is then converted into an electronic signal by the light detector. The gamma ray is an *impulse*, while the output of the detector (i.e., the *impulse response*) resembles a one-sided exponential.

РИСУНОК 17-3. Пример неизбежной свертки.

Датчик гамма луча может быть сформирован, устанавливая сцинтиллятор на светоприемнике. Когда луч гаммы попадает на сцинтиллятор, его энергия преобразована в импульс света. Этот импульс света тогда преобразован в электронный сигнал светоприемником. Гамма луч - импульс, в то время как выход датчика (то есть, импульсная передаточная функция) походит на одностороннюю показательную функцию.

Figure 17-4a shows pulses generated by the detector in response to randomly arriving gamma rays. The information we would like to extract from this output signal is the *amplitude* of each pulse, which is proportional to the *energy* of the gamma ray that generated it. This is useful information because the energy can tell interesting things about where the gamma ray has been. For example, it may provide medical information on a patient, tell the age of a distant galaxy, detect a bomb in airline luggage, etc.

Рисунок 17-4а показывает импульсы, сгенерированные датчиком в ответ на беспорядочно прибывающие гамма-лучи. Информация, которую мы хотели бы извлечь из этого сигнала выхода - амплитуда каждого импульса, который является пропорциональным к энергии гаммы луча, который генерировал это. Это - полезная информация, потому что энергия может сообщить интересные вещи относительно того, где гаммы луч был. Например, это может обеспечивать медицинскую информацию относительно пациента, сообщать возраст отдаленной галактики, обнаруживать бомбу в багаже на авиалиниях, и т.д.

Everything would be fine if only an occasional gamma ray were detected, but this is usually not the case. As shown in (a), two or more pulses may overlap, shifting the measured amplitude. One answer to this problem is to *deconvolve* the detector's output signal, making the pulses narrower so that less pile-up occurs. Ideally, we would like each pulse to resemble the original impulse. As you may suspect, this isn't possible and we must settle for a pulse that is finite in length, but significantly shorter than the detected pulse. This goal is illustrated in Fig. 17-4b.

Все было бы прекрасно, если только случайный гамма луч был обнаружен, но это - обычно не случай. Как показано в (а), два или больше импульса могут накладываться, сдвигая измеренную(ритмичную) амплитуду. Один ответ на эту проблему *обращение свертки* сигнала выхода датчика, делая импульсы более узкими так, чтобы меньше накопления(группирования импульсов; наложения импульсов?), происходило. Идеально, мы хотели бы, чтобы каждый импульс походил на первоначальный импульс. Как Вы можете

подозревать, это не возможно, и мы должны согласиться на импульс, который является конечным в длине, но значительно короче чем обнаруженный импульс. Эта цель иллюстрирована в рис. 17-4b.

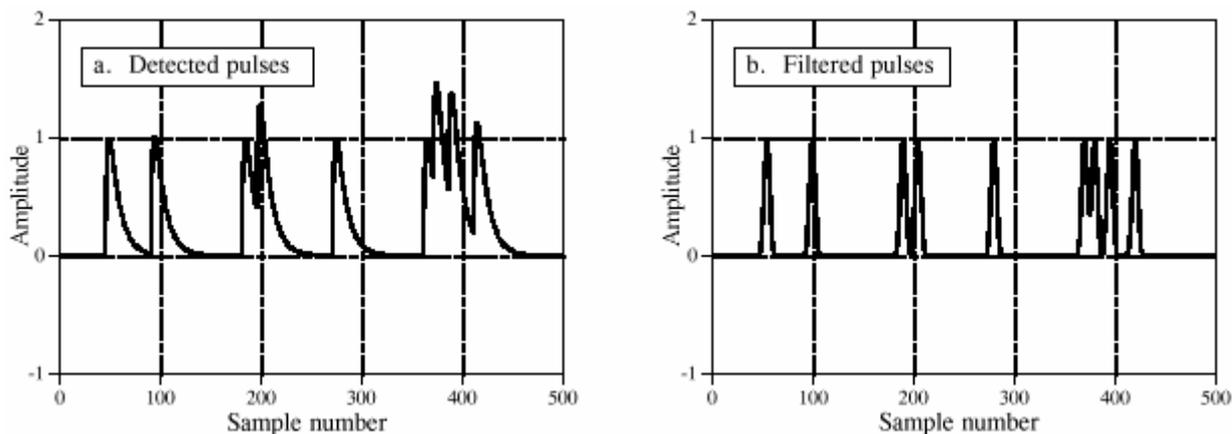


FIGURE 17-4

Example of deconvolution. Figure (a) shows the output signal from a gamma ray detector in response to a series of randomly arriving gamma rays. The deconvolution filter is designed to convert (a) into (b), by reducing the width of the pulses. This minimizes the amplitude shift when pulses land on top of each other.

РИСУНОК 17-4. Пример деконволюции.

Рисунок (а) показывает сигнал выхода от датчика гамма луча в ответ на ряд беспорядочно прибывающих гамма-лучей. Фильтр деконволюции предназначен, чтобы преобразовать (а) в (б), приводя ширину импульсов. Это минимизирует амплитудный сдвиг, когда импульсы попадают на вершине друг друга.

Even though the detector signal has its information encoded in the *time domain*, much of our analysis must be done in the *frequency domain*, where the problem is easier to understand. Figure 17-5a is the signal produced by the detector (something we know). Figure (c) is the signal we wish to have (also something we know). This desired pulse was arbitrarily selected to be the same shape as a Blackman window, with a length about one-third that of the original pulse. Our goal is to find a filter kernel, (e), that when convolved with the signal in (a), produces the signal in (c). In equation form: if  $a * e = c$  and given  $a$  and  $c$ , find  $e$ .

Даже при том, что сигнал датчика кодирует его информацию в *домене времени*, многое из нашего анализа должно быть сделано в *частотном домене*, где проблема проще, чтобы понять. Рисунок 17-5а - сигнал, произведенный датчиком (кое-что, что мы знаем). Рисунок (с) - сигнал, мы желаем иметь (также кое-что, что мы знаем). Этот желательный импульс был произвольно отобран, чтобы быть той же самой формой как окно Блэкмана, с длиной приблизительно третья часть таковой первоначального импульса. Наша цель состоит в том, чтобы найти ядро фильтра, (е), что когда свернуто с сигналом в (а), производит сигнал в (с). В форме уравнения: если  $a * e = c$ , и даны  $a$  и  $c$ , находят  $e$ .

If these signals were combined by addition or multiplication instead of convolution, the solution would be easy: *subtraction* is used to "de-add" and *division* is used to "de-multiply." Convolution is different; there is not a simple inverse operation that can be called "deconvolution." Convolution is too messy to be undone by directly manipulating the time domain signals.

Если бы эти сигналы были объединены добавлением или умножением вместо свертки, решение было бы просто: *вычитание* используется, чтобы "де-прибавлять" ("предотвращать добавление"), и *деление* используется, чтобы "де-умножать" ("де-").

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

мультиплицировать"). Свертка различна; не имеется простой обратной операции, которая может называться "деконволюцией". Свертка слишком грязна, чтобы быть уничтоженной, непосредственно управляя сигналами домена времени.

Fortunately, this problem is simpler in the frequency domain. Remember, *convolution* in one domain corresponds with *multiplication* in the other domain. Again referring to the signals in Fig. 17-5: if  $b \times f = d$ , and given  $b$  and  $d$ , find  $f$ . This is an easy problem to solve: the frequency response of the filter, ( $f$ ), is the frequency spectrum of the desired pulse, ( $d$ ), *divided* by the frequency spectrum of the detected pulse, ( $b$ ). Since the detected pulse is asymmetrical, it will have a *nonzero* phase. This means that a *complex* division must be used (that is, a magnitude & phase divided by another magnitude & phase). In case you have forgotten, Chapter 9 defines how to perform a complex division of one spectrum by another. The required filter kernel, ( $e$ ), is then found from the frequency response by the custom filter method (IDFT, shift, truncate, & multiply by a window).

К счастью, эта проблема более простая в частотном домене. Помните, свертка в одном домене соответствует *умножению* (*мультиплицированию*) в другом домене. Снова что касается сигналов в рис. 17-5: если  $b \times f = d$ , даны и  $b$  и  $d$ , находят  $f$ . Это - простая проблема решить: частотная характеристика фильтра, ( $f$ ), является спектром частот желательного импульса, ( $d$ ), разделенная спектром частот обнаруженного импульса, ( $b$ ). Так как обнаруженный импульс асимметричен, это будет иметь фазу отличную от нуля. Это означает, что комплексное деление должно использоваться (то есть величина и фаза, разделенная другой величиной и фазой). В случае, если Вы забыли, глава 9 определяет, как исполнить комплексное деление одного спектра другим. Требуемое ядро фильтра, ( $e$ ), тогда найдено от частотной характеристики методом заказного фильтра (IDFT, сдвигом, усечением, и умножением окном).

There are limits to the improvement that deconvolution can provide. In other words, if you get greedy, things will fall apart. Getting greedy in this example means trying to make the desired pulse excessively narrow. Let's look at what happens. If the desired pulse is made narrower, its frequency spectrum must contain more high frequency components. Since these high frequency components are at a very low amplitude in the detected pulse, the filter must have a very high gain at these frequencies. For instance, ( $f$ ) shows that some frequencies must be multiplied by a factor of *three* to achieve the desired pulse in ( $c$ ). If the desired pulse is made narrower, the gain of the deconvolution filter will be even greater at high frequencies.

Имеются пределы уточнению, которое деконволюция может обеспечивать. Другими словами, если Вы становитесь жадными, вещи развалятся. Жадность в этом примере - желание сделать желательный импульс чрезмерно узким. Давайте смотреть то, что случается. Если желательный импульс сделан более узким, его спектр частот должен содержать большее количество компонентов высокой частоты. Так как эти компоненты высокой частоты - в очень низкой амплитуде в обнаруженном импульсе, фильтр должен иметь очень высокое усиление в этих частотах. Например, ( $f$ ) показывает, что некоторые частоты должны быть умножены на коэффициент три, чтобы достичь желательного импульса в ( $c$ ). Если желательный импульс сделан более узким, усиление фильтра деконволюции будет даже больше в высоких частотах.

The problem is, small errors are very unforgiving in this situation. For instance, if some frequency is amplified by 30, when only 28 is required, the deconvolved signal will probably be a mess. When the deconvolution is pushed to greater levels of performance, the characteristics of the unwanted convolution must be understood with greater *accuracy* and *precision*. There are

always unknowns in real world applications, caused by such villains as: electronic noise, temperature drift, variation between devices, etc. These unknowns set a limit on how well deconvolution will work.

Проблема, маленькие ошибки очень неумолимы в этом положении(ситуации). Например, если некоторая частота усилена 30, когда требуется только 28, сигнал восстановленный методом обращения свертки будет вероятно беспорядочен. Когда деконволюция помещена в большие уровни эффективности, характеристики нежелательной свертки должны быть поняты с большей *точностью* и *прецизионностью*. Имеются всегда неизвестные в реальных(вещественных) мировых приложениях, вызванных такими злодеями как: электронный шум, дрейф температуры, вариация(влияния) между устройствами, и т.д. Эти неизвестные устанавливают предел на том, как хорошо деконволюция будет работать.

Even if the unwanted convolution is *perfectly* understood, there is still a factor that limits the performance of deconvolution: *noise*. For instance, most unwanted convolutions take the form of a low-pass filter, reducing the amplitude of the high frequency components in the signal. Deconvolution corrects this by amplifying these frequencies. However, if the amplitude of these components falls below the inherent noise of the system, the information contained in these frequencies is lost. No amount of signal processing can retrieve it. It's gone forever. Adios! Goodbye! Trying to reclaim this data will only amplify the noise. As an extreme case, the amplitude of some frequencies may be completely reduced to *zero*. This not only obliterates the information, it will try to make the deconvolution filter have *infinite* gain at these frequencies. The solution: design a less aggressive deconvolution filter and/or place limits on how much gain is allowed at any of the frequencies.

Даже если нежелательная свертка *совершенно* понята, имеется все еще фактор, который ограничивает эффективность деконволюции: *шум*. Например, наиболее нежелательные свертки принимают форму фильтра нижних частот, приводя амплитуду компонентов высокой частоты в сигнале. Деконволюция исправляет это, усиливая эти частоты. Однако, если падение амплитуды этих компонентов ниже свойственного шума системы, информация, содержащаяся в этих частотах потеряна. Никакое количество обработки сигналов не может восстановить(отыскивать) это. Это ушло навсегда. Пока! До свидания! Ну и ну! Попытка исправлять эти данные только усилит шум. Как критический случай, амплитуда некоторых частот может быть полностью сокращена, к *нулю*. Этот не только стирает информацию, это будет пробовать заставить фильтр деконволюции иметь *бесконечное* усиление в этих частотах. Решение: Проектируйте менее агрессивный фильтр деконволюции, и-или разместите пределы в то, сколько усиление позволяет в любой из частот.

How far can you go? How greedy is too greedy? This depends totally on the problem you are attacking. If the signal is well behaved and has low noise, a significant improvement can probably be made (think a factor of 5-10). If the signal changes over time, isn't especially well understood, or is noisy, you won't do nearly as well (think a factor of 1-2). Successful deconvolution involves a great deal of testing. If it works at some level, try going farther; you will know when it falls apart. No amount of theoretical work will allow you to bypass this iterative process.

Как далеко Вы можете идти? Насколько жадный слишком жадно? Это зависит полностью от проблемы, на которую Вы нападаете. Если сигнал хорошо ведет себя и имеет низкий шум, существенное уточнение может вероятно быть сделано (думайте о коэффициенте 5-10). Если сигнал с течением времени, не особенно хорошо понят, или шумный, Вы не будете делать почти также (думайте о коэффициенте 1-2). Успешная деконволюция включает в себя(подразумевает) много испытания. Если это работает на некотором уровне, по-

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

пытка, идущая дальше; Вы будете знать, когда это разваливается. Никакое количество теоретической работы не позволит Вам обходить этот итеративный процесс.

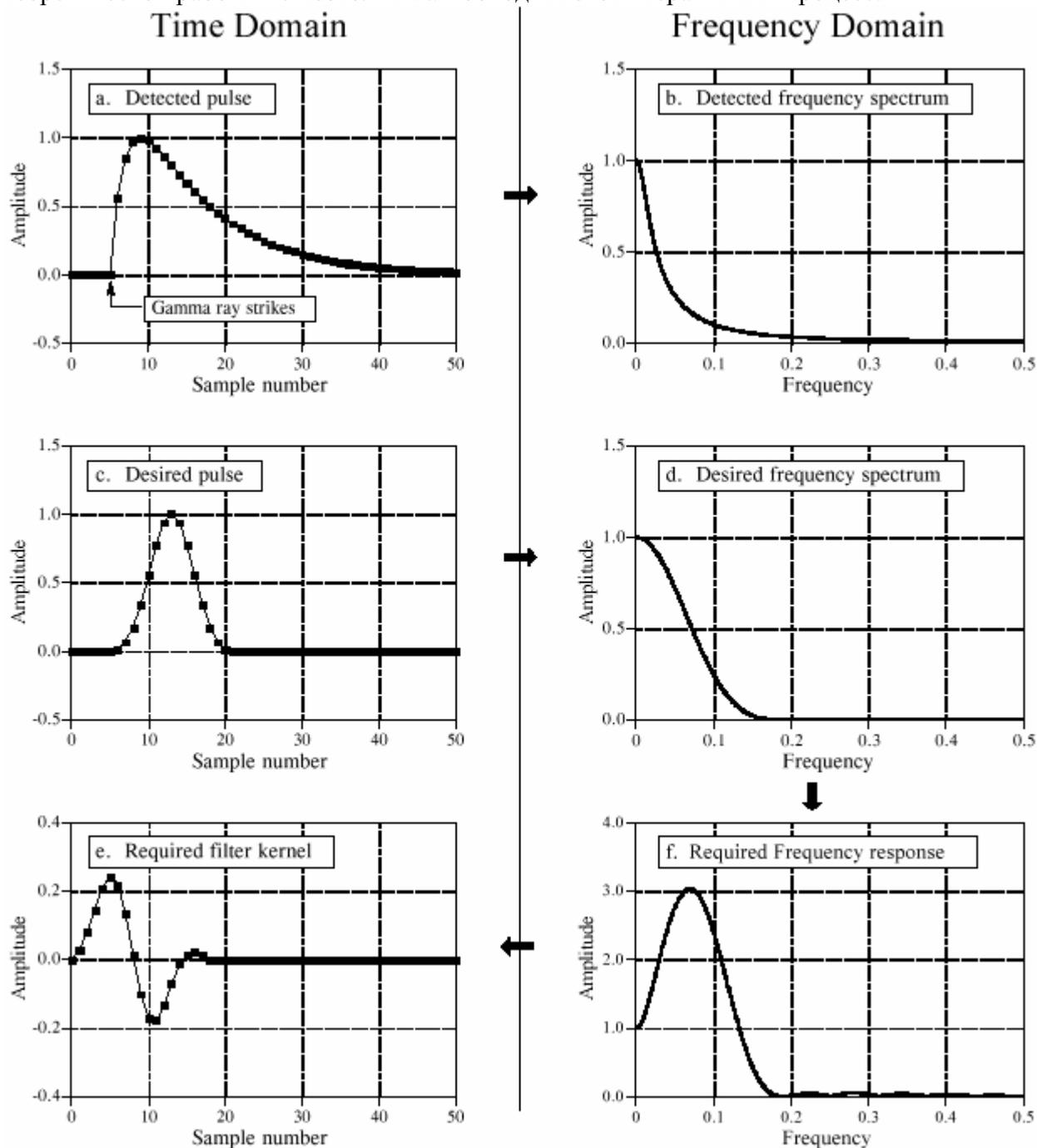


FIGURE 17-5. Example of deconvolution in the time and frequency domains.

The impulse response of the example gamma ray detector is shown in (a), while the desired impulse response is shown in (c). The frequency spectra of these two signals are shown in (b) and (d), respectively. The filter that changes (a) into (c) has a frequency response, (f), equal to (b) divided by (d). The filter kernel of this filter, (e), is then found from the frequency response using the custom filter design method (inverse DFT, truncation, windowing). Only the magnitudes of the frequency domain signals are shown in this illustration; however, the phases are nonzero and must also be used.

**ЧИСЛО(РИСУНОК) 17-5**

Пример деконволюции во доменах времени и частоты. Пример импульсная передаточной функции датчика гамма луча показывается в (a), в то время как желательная импульсная передаточная функция показывается в (c). Частотные спектры из этих двух сигналов показываются в (b) и (d), соответственно. Фильтр, который изменяет(заменяет) (a) в (c), имеет частотную характеристику, (f), равный (b), разделенному (d). Ядро (c) АВТЭКС, Санкт-Петербург, <http://www.autex.spb.ru>, e-mail: [info@autex.spb.ru](mailto:info@autex.spb.ru)

фильтра этого фильтра, (е), тогда найдено от частотной характеристики, используя метод проекта заказного фильтра (обратный ДПФ, усечение, работа с окнами). Только величины сигналов домена частоты показаны в этой иллюстрации; однако, фазы отличные от нуля и должны также использоваться.

Deconvolution can also be applied to *frequency domain* encoded signals. A classic example is the restoration of old recordings of the famous opera singer, Enrico Caruso (1873-1921). These recordings were made with very primitive equipment by modern standards. The most significant problem is the *resonances* of the long tubular recording horn used to gather the sound. Whenever the singer happens to hit one of these resonance frequencies, the loudness of the recording abruptly increases. Digital deconvolution has improved the subjective quality of these recordings by the loud spots in the music. We will only describe the general method; for a detailed description, see the original paper: T. Stockham, T. Cannon, and R. Ingebretsen, "Blind Deconvolution Through Digital Signal Processing", *Proc. IEEE*, vol. 63, Apr. 1975, pp. 678-692.

Деконволюция может также применяться к частотному домену(области) закодированные сигналы. Классический пример - восстановление старых регистрации известного певца оперы, Enrico Caruso (1873-1921). Эти регистрация были сделаны с очень примитивным оборудованием по современным стандартам. Наиболее значительная проблема - резонансы длинного трубчатого рожка регистрации, имел обыкновение собирать звук. Всякий раз, когда певец, случается, нажимает одну из этих резонансных частот, громкость регистрации резко увеличивается. Цифровая деконволюция улучшила субъективное качество этих регистрации громкими пятнами(ячейками) в музыке. Мы только опишем общий метод; для детального описания, см. первоначальную бумагу: Т. Stockham, Т. Орудие, и R. Ingebretsen, " Слепая Деконволюция Через Цифровую Обработку сигналов ", *Proc. ИИЭР*, vol. 63, апрель 1975, pp. 678-692.

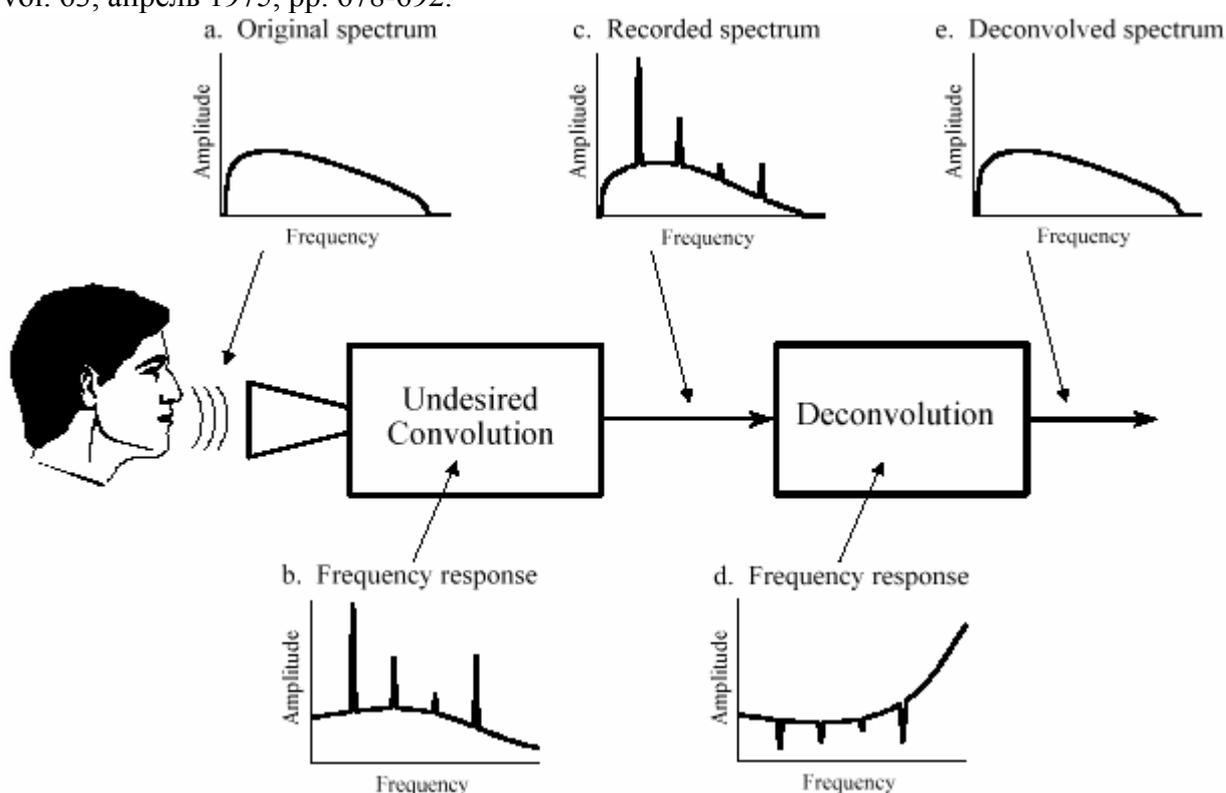


FIGURE 17-6

Deconvolution of old phonograph recordings. The frequency spectrum produced by the original singer is illustrated in (a). Resonance peaks in the primitive equipment, (b), produce distortion in the recorded frequency spectrum, (c). The frequency response of the deconvolution filter, (d), is designed to counteracts the undesired convolution, restoring the original spectrum, (e). These graphs are for illustrative purposes only; they are not actual signals.

РИСУНОК 17-6. Деконволюция старых записей фонографа.

Спектр частот, произведенный первоначальным певцом иллюстрирован в (а). Резонансные пики в примитивном оборудовании, (b), производят искажение в зарегистрированном спектре частот, (c). Частотная характеристика фильтра деконволюции, (d), разработана к противодействию нежелательной свертки, восстанавливая первоначальный спектр, (e). Эти диаграммы(графики) - для целей иллюстрации; они - не фактические сигналы.

Figure 17-6 shows the general approach. The frequency spectrum of the original audio signal is illustrated in (a). Figure (b) shows the frequency response of the recording equipment, a relatively smooth curve except for several sharp resonance peaks. The spectrum of the recorded signal, shown in (c), is equal to the true spectrum, (a), multiplied by the uneven frequency response, (b). The goal of the deconvolution is to *counteract* the undesired convolution. In other words, the frequency response of the deconvolution filter, (d), must be the *inverse* of (b). That is, each peak in (b) is cancelled by a corresponding dip in (d). If this filter were perfectly designed, the resulting signal would have a spectrum, (e), identical to that of the original. Here's the catch: the original recording equipment has long been discarded, and its frequency response, (b), is a mystery. In other words, this is a *blind deconvolution* problem; given only (c), how can we determine (d)?

Рисунок 17-6 показывает общий подход. Спектр частот первоначального звукового сигнала иллюстрирован в (а). Рисунок (b) показывает частотную характеристику оборудования регистрации, относительно гладкая кривая, если бы не несколько пиков острого резонанса. Спектр зарегистрированного сигнала, показанного в (c), является равным истинному спектру, (a), мультиплицированным(умноженным) нечетной частотной характеристикой, (b). Цель деконволюции состоит в том, чтобы *противодействовать* нежелательной свертке. Другими словами, частотная характеристика фильтра деконволюции, (d), должна быть *инверсия* (b). То есть каждый пик в (b) отменен соответствующим падением в (d). Если бы этот фильтр был разработан совершенно, заканчивающийся сигнал имел бы спектр, (e), идентичный таковому оригинала. Имеется арретир: оригинал оборудования на котором сделана запись давно был отвергнут(списан), и его частотная характеристика, (b), - тайна(мистерия; детективный роман). Другими словами, это - *проблема слепой деконволюции*; дано только (c), как мы можем определить (d)?

Blind deconvolution problems are usually attacked by making an estimate or assumption about the unknown parameters. To deal with this example, the *average spectrum* of the original music is assumed to match the *average spectrum* of the same music performed by a present day singer using modern equipment. The *average spectrum* is found by the techniques of Chapter 9: break the signal into a large number of segments, take the DFT of each segment, convert into polar form, and then average the magnitudes together. In the simplest case, the unknown frequency response is taken as the average spectrum of the old recording, divided by the average spectrum of the modern recording. (The method used by Stockham et al. is based on a more sophisticated technique called *homomorphic* processing, providing a better estimate of the characteristics of the recording system).

Слепые проблемы деконволюции обычно атакуются, делая оценку или предположение относительно неизвестных параметров. Чтобы иметь дело с этим примером, *средний спектр* первоначальной музыки принят, чтобы соответствовать *среднему спектру* той же самой музыки, выполненной существующим певцом сегодняшнего дня, использующим современное оборудование. *Средний спектр* найден методами главы 9: разбейте сигнал в большое количество сегментов, берите ДПФ каждого сегмента, преобразуйте в полярную форму, и затем усредните величины вместе. В самом простом случае, неизвестная частотная характеристика принята как средний спектр старой записи, разделенный средним спектром современной записи. (Метод, используемый Stockham\_ом и другими. Основан на бо-

лее сложной(искусственной) методике, называемой *гомоморфной* обработкой, обеспечивающей лучшую оценку характеристик системы записи).

## **Optimal Filters**

### **Оптимальные Фильтры**

Figure 17-7a illustrates a common filtering problem: trying to extract a waveform (in this example, an exponential pulse) buried in random noise. As shown in (b), this problem is no easier in the frequency domain. The signal has a spectrum composed mainly of low frequency components. In comparison, the spectrum of the noise is *white* (the same amplitude at all frequencies). Since the spectra of the signal and noise *overlap*, it is not clear how the two can best be separated. In fact, the real question is how to define what "best" means. We will look at three filters, each of which is "best" (optimal) in a different way. Figure 17-8 shows the filter kernel and frequency response for each of these filters. Figure 17-9 shows the result of using these filters on the example waveform of Fig. 17-7a.

Рисунок 17-7а иллюстрирует обычную проблему фильтрации: пробуя извлекать форму волны (в этом примере, показательный импульс) скрытый в случайном шуме. Как показано в (b), эта проблема - не проще в частотном домене. Сигнал составляет спектр главным образом низкочастотных компонентов. Для сравнения, спектр шума - *белый* (та же самая амплитуда во всех частотах). Начиная со спектров сигнала и шумового *перекрытия*, это не чисто как эти два могут лучше всего быть раз(от)деленными. Фактически, реальный вопрос - как определить то, какой "лучше всего" означает. Мы будем смотреть на три фильтра, каждый из которых "лучший" (оптимален) различным способом. Рисунок 17-8 показывает ядро фильтра и частотную характеристику для каждого из этих фильтров. Рисунок 17-9 показывает результат использования этих фильтров на примере формы волны рис. 17-7а.

The **moving average filter** is the topic of Chapter 15. As you recall, each output point produced by the moving average filter is the average of a certain number of points from the input signal. This makes the filter kernel a rectangular pulse with an amplitude equal to the reciprocal of the number of points in the average. The moving average filter is optimal in the sense that it provides the fastest step response for a given noise reduction.

**Фильтр скользящего среднего** - тема главы 15. Как Вы помните, каждая точка выхода, произведенная фильтром скользящего среднего - среднее число некоторого числа точек от входного сигнала. Это делает ядро фильтра, прямоугольным импульсом с амплитудой равной обратной величине числа точек в среднем. Фильтр скользящего среднего оптимален в смысле, что это обеспечивает самую быструю реакцию на скачок для данного шумового приведения.

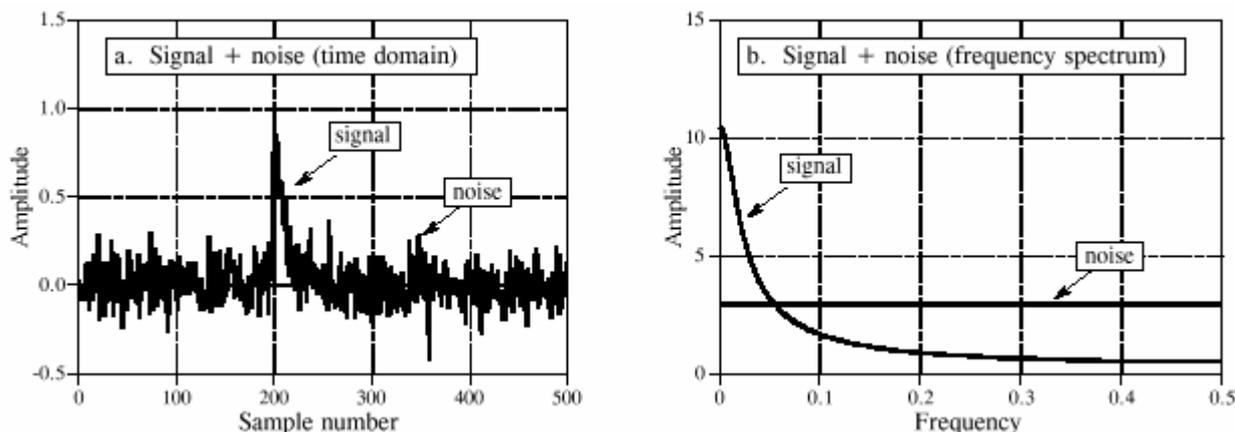


FIGURE 17-7

Example of optimal filtering. In (a), an exponential pulse buried in random noise. The frequency spectra of the pulse and noise are shown in (b). Since the signal and noise overlap in both the time and frequency domains, the best way to separate them isn't obvious.

РИСУНОК 17-7. Пример оптимальной фильтрации.

В (а), показательный импульс, скрытый в случайном шуме. Частотные спектры импульса и шума показыва-ются в (b). Начиная с сигнала и шумового перекрытия и в домене времени и частотном домене(в обоих), лучший способ отделить их не очевиден.

The **matched filter** was previously discussed in Chapter 7. As shown in Fig. 17-8a, the filter kernel of the matched filter is the same as the target signal being detected, except it has been flipped left-for-right. The idea behind the matched filter is *correlation*, and this flip is required to perform *correlation* using *convolution*. The amplitude of each point in the output signal is a measure of how well the filter kernel *matches* the corresponding section of the input signal. Recall that the output of a matched filter does not necessarily look like the signal being detected. This doesn't really matter; if a matched filter is used, the shape of the target signal must already be known. The matched filter is optimal in the sense that the top of the peak is farther above the noise than can be achieved with any other linear filter (see Fig. 17-9b).

**Согласованный фильтр** был предварительно обсужден в главе 7. Как показано в рис. 17-8а, ядро фильтра согласованного фильтра - то же самое, как обнаруживаемый целевой сигнал, кроме этого было зеркально отражено, лево-право. Идея позади(за спиной) согла-сованного фильтра - *корреляция*, и это зеркальное отражение, требуется, чтобы исполнить *корреляцию*, используя *свертку*. Амплитуда каждой точки в сигнале выхода - мера того, как хорошо ядро фильтра соответствует соответствующему разделу входного сигнала. Вспомните, что выход согласованного фильтра не обязательно напоминает обнаруживаемый сигнал. Это действительно не имеет значения; если согласованный фильтр использу-ется, форма целевого сигнала должна уже быть известна. Согласованный фильтр оптима-лен в смысле, что вершина пика дальше выше шума чем может быть достигнуто с любым другим линейным фильтром (см. рис. 17-9b).

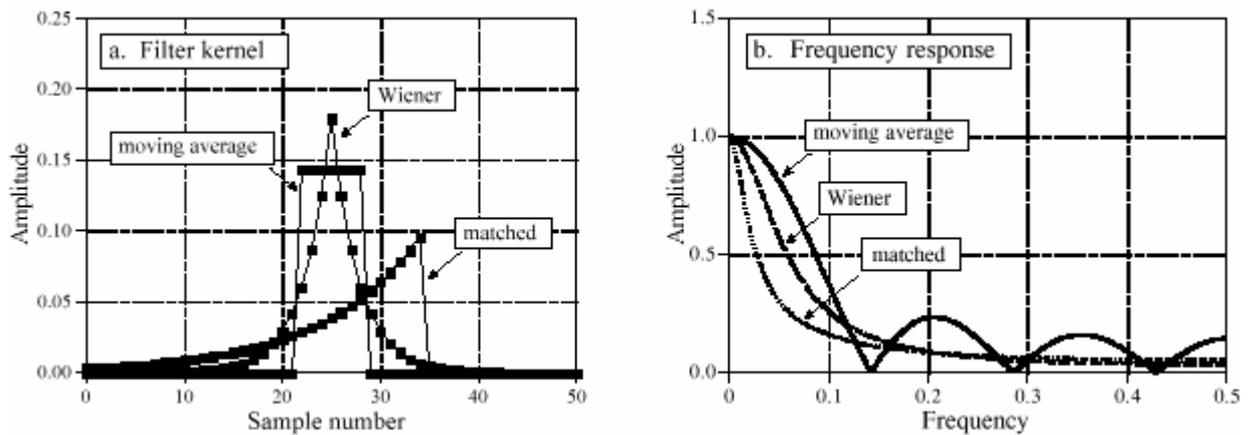


FIGURE 17-8

Example of optimal filters. In (a), three filter kernels are shown, each of which is optimal in some sense. The corresponding frequency responses are shown in (b). The moving average filter is designed to have a rectangular pulse for a filter kernel. In comparison, the filter kernel of the matched filter looks like the signal being detected. The Wiener filter is designed in the frequency domain, based on the relative amounts of signal and noise present at each frequency.

РИСУНОК 17-8. Пример оптимальных фильтров.

В (а), показаны ядра трех фильтров, каждый из которых оптимален в некотором смысле. Соответствующие частотные характеристики показываються в (b). Фильтр скользящего среднего предназначен(разработан), чтобы иметь прямоугольный импульс для ядра фильтра. Для сравнения, ядро фильтра согласованного фильтра напоминает обнаруживаемый сигнал. Винеровский фильтр предназначен в частотном домене, основан на относительных количествах сигнала и шумового представления в каждой частоте.

The **Wiener filter** (named after the optimal estimation theory of Norbert Wiener) separates signals based on their frequency spectra. As shown in Fig. 17-7b, at some frequencies there is mostly signal, while at others there is mostly noise. It seems logical that the "mostly signal" frequencies should be passed through the filter, while the "mostly noise" frequencies should be blocked. The Wiener filter takes this idea a step further; the gain of the filter *at each frequency* is determined by the relative amount of signal and noise *at that frequency*:

**Винеровский фильтр** (названный по имени теории оптимальной оценки Norbert\_a Винера) отделяет сигналы, основанные на их частотных спектрах. Как показано в рис. 17-7b, в некоторых частотах имеется главным образом сигнал, в то время как в других имеется главным образом шум. Кажется логически, что "главным образом сигнал" частоты нужно пропустить через фильтр, в то время как "главным образом шум" частоты должен быть блокирован. Винеровский фильтр берет эту идею шаг далее; усиление фильтра *в каждой частоте* определена соотношением сигнала и шума *в той частоте*:

EQUATION 17-1. The Wiener filter.

The frequency response, represented by  $H[f]$ , is determined by the frequency spectra of the noise,  $N[f]$ , and the signal,  $S[f]$ . Only the magnitudes are important; all of the phases are zero.

$$H[f] = \frac{S[f]^2}{S[f]^2 + N[f]^2}$$

УРАВНЕНИЕ 17-1. Винеровский фильтр.

Частотная характеристика, представленная  $H[f]$ , определена частотными спектрами шума,  $N[f]$ , и сигналом,  $S[f]$ . Только величины - важны; все фазы нулевые.

This relation is used to convert the spectra in Fig. 17-7b into the Wiener filter's frequency response in Fig. 17-8b. The Wiener filter is optimal in the sense that it maximizes the ratio of the signal power to the noise power (over the length of the signal, not at each individual point). An appropriate filter kernel is designed from the Wiener frequency response using the custom method.

Это отношение используется, чтобы конвертировать спектры в рис. 17-7b в частотный ответ Винеровского фильтра в рис. 17-8b. Винеровский фильтр оптимален в смысле, что это максимизирует отношение мощности сигнала к мощности шума (по длине сигнала, не в каждой индивидуальной точке). Соответствующее ядро фильтра разработано от Винеровской частотной характеристики, используя заказной метод.

While the ideas behind these optimal filters are mathematically elegant, they often fail in practicality. This isn't to say they should never be used. The point is, don't hear the word "optimal" and stop thinking. Let's look at several reasons why you might *not* want to use them.

В то время как идеи позади(за спиной) этих оптимальных фильтров математически изящна, они часто терпят неудачу в практичности. Это не должно говорить, что они никогда не должны использоваться. Пункт, не слышите слово "оптимальное" и прекращаете о них думать. Давайте смотреть на несколько причин, почему Вы не могли бы хотеть использовать их.

First, the difference between the signals in Fig. 17-9 is very unimpressive. In fact, if you weren't told what parameters were being optimized, you probably couldn't tell by looking at the signals. This is usually the case for problems involving overlapping frequency spectra. The small amount of extra performance obtained from an optimal filter may not be worth the the increased program complexity, the extra design effort, or the longer execution time.

Во первых, разность между сигналами в рис. 17-9 очень невпечатляющая. Фактически, если Вам не говорили, что параметры оптимизировались, Вы вероятно не могли сообщить разглядывая сигналы. Это обычно имеет место для проблем при включении перекрытия на частотные спектры. Маленькое количество дополнительной эффективности, полученной от оптимального фильтра не может стоить увеличенной сложности программы, дополнительные конструкторские работы, или более длинное время выполнения.

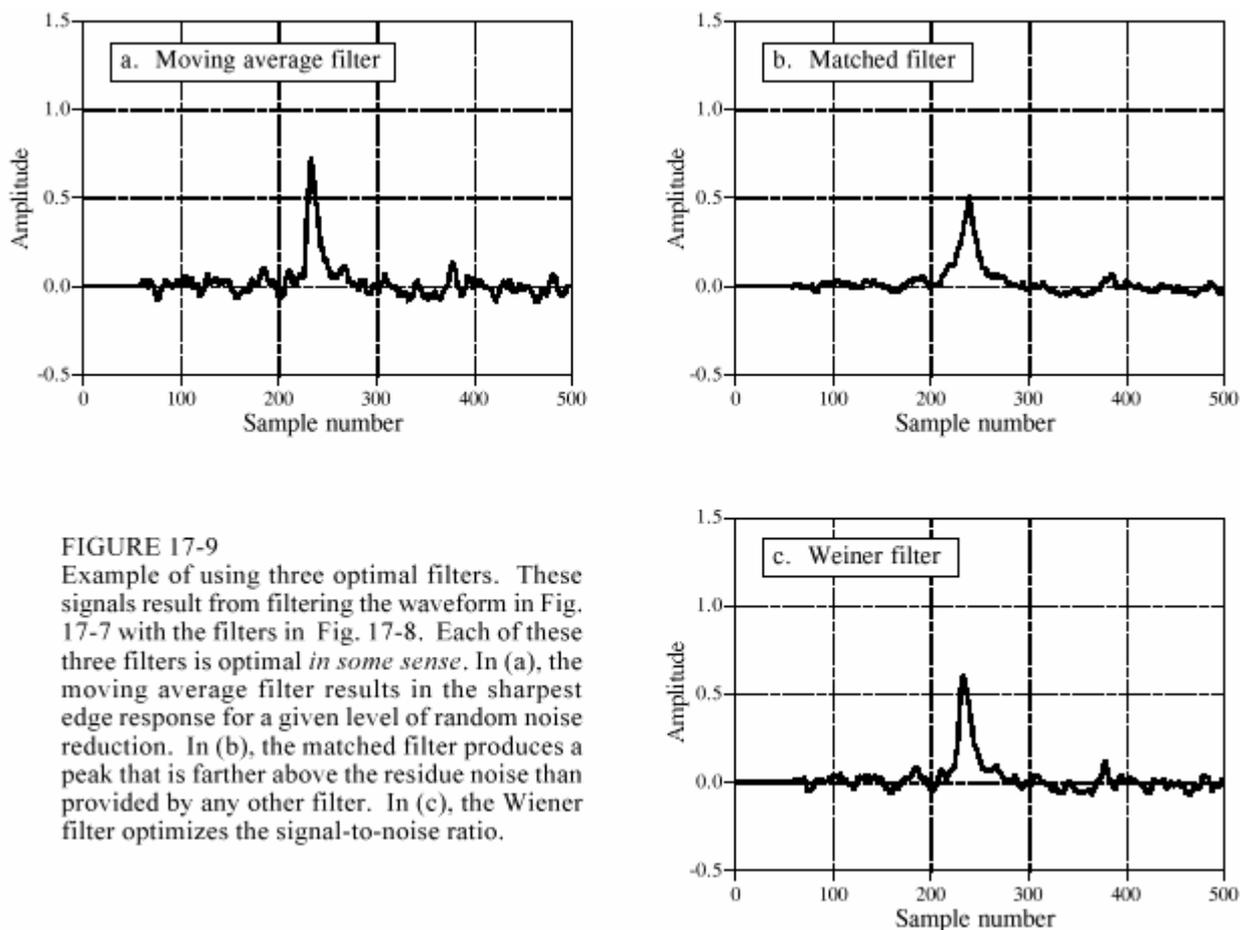


FIGURE 17-9 Example of using three optimal filters. These signals result from filtering the waveform in Fig. 17-7 with the filters in Fig. 17-8. Each of these three filters is optimal *in some sense*. In (a), the moving average filter results in the sharpest edge response for a given level of random noise reduction. In (b), the matched filter produces a peak that is farther above the residue noise than provided by any other filter. In (c), the Wiener filter optimizes the signal-to-noise ratio.

FIGURE 17-9 Example of using three optimal filters. These signals result from filtering the waveform in Fig. 17-7 with the filters in Fig. 17-8. Each of these three filters is optimal *in some sense*. In (a), the moving average filter results in the sharpest edge response for a given level of random noise reduction. In (b), the matched filter produces a peak that is farther above the residue noise than provided by any other filter. In (c), the Wiener filter optimizes the signal-to-noise ratio.

РИСУНОК 17-9. Пример использования трех оптимальных фильтров. Эти сигналы следствие фильтрации формы волны в рис. 17-7 с фильтрами в рис. 17-8. Каждый из этих трех фильтров оптимален в некотором смысле. В (а), фильтр скользящего среднего приводит к самому острому ответу края(фронта) для данного уровня случайного шумового приведения. В (b), согласованный фильтр производит пик, который является дальше выше остаточного шума чем обеспеченный любым другим фильтром. В (с), Винеровский фильтр оптимизирует отношение(коэффициент) " сигнал к шуму ".

Second: The Wiener and matched filters are completely determined by the characteristics of the problem. Other filters, such as the windowed-sinc and moving average, can be tailored to your liking. Optimal filter advocates would claim that this diddling can only reduce the effectiveness of the filter. This is very arguable. Remember, each of these filters is optimal in one specific way (i.e., "in some sense"). This is seldom sufficient to claim that the entire problem has been optimized, especially if the resulting signals are interpreted by a human observer. For instance, a biomedical engineer might use a Wiener filter to maximize the signal-to-noise ratio in an electrocardiogram. However, it is not obvious that this also optimizes a physician's ability to detect irregular heart activity by looking at the signal.

Во вторых: Винер и согласованные фильтры полностью определены в соответствии с характеристиками проблемы. Другие фильтры, типа windowed-sinc и скользящего среднего, могут быть приспособлены к вашей симпатии. Оптимальные защитники фильтра утверждали бы, что этот diddling(сдвиг элементов изображения) может только приводить эф-

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

эффективность фильтра. Это очень спорно. Помните, каждый из этих фильтров оптимален одним специфическим способом (То есть, " в некотором смысле "). Это достаточно редко требовать, чтобы полная проблема была оптимизирована, особенно, если заканчивающиеся сигналы интерпретируются человеческим глазом. Например, биомедицинский инженер мог бы использовать Винеровский фильтр, чтобы максимизировать отношение " сигнал к шуму " в электрокардиограмме. Однако, не очевидно, что это также оптимизирует способность врача обнаружить неправильное действие сердца глядя на сигнал.

Third: The Wiener and matched filter must be carried out by *convolution*, making them extremely slow to execute. Even with the speed improvements discussed in the next chapter (FFT convolution), the computation time can be excessively long. In comparison, *recursive* filters (such as the moving average or others presented in Chapter 19) are much faster, and may provide an acceptable level of performance.

Третье: Винеровский и согласованный фильтр должен быть выполнен сверткой, делая их чрезвычайно медленными, чтобы выполняться. Даже с уточнениями быстродействия, обсужденными в следующей главе (свертка БПФ), время вычисления может быть чрезмерно долго. Для сравнения, рекурсивные фильтры (типа скользящего среднего или другие представленные в главе 19) - намного быстрее, и могут обеспечивать приемлемый уровень эффективности.