



The Discrete Fourier Transform Дискретное преобразование Фурье

Fourier analysis is a family of mathematical techniques, all based on decomposing signals into sinusoids. The discrete Fourier transform (DFT) is the family member used with *digitized* signals. This is the first of four chapters on the **real DFT**, a version of the discrete Fourier transform that uses real numbers to represent the input and output signals. The **complex DFT**, a more advanced technique that uses complex numbers, will be discussed in Chapter 31. In this chapter we look at the mathematics and algorithms of the Fourier decomposition, the heart of the DFT.

Анализ Фурье, (гармонический анализ) - семейство математических методов, основанное на разложении сигналов на синусоиды. Дискретное преобразование Фурье (ДПФ) - элемент семейства, используемый с *цифровыми* сигналами. Это первая из четырех глав по **реальному(вещественному) ДПФ**, версия дискретного анализа Фурье, который, для того чтобы представить сигналы ввода и вывода, используя вещественные числа. **Комплексное ДПФ**, более продвинутая методика, которая использует комплексные числа, будет обсуждено в главе 31. В этой главе мы рассмотрим математику и алгоритмы декомпозиции Фурье, основы(сердца) ДПФ.

The Family of Fourier Transform Семейство Преобразований Фурье

Fourier analysis is named after **Jean Baptiste Joseph Fourier** (1768-1830), a French mathematician and physicist. (Fourier is pronounced: *for-ē-a*, and is always capitalized).). While many contributed to the field, Fourier is honored for his mathematical discoveries and insight into the practical usefulness of the techniques. Fourier was interested in heat propagation, and presented a paper in 1807 to the Institut de France on the use of sinusoids to represent temperature distributions. The paper contained the controversial claim that any continuous periodic signal could be represented as the sum of properly chosen sinusoidal waves. Among the reviewers were two of history's most famous mathematicians, Joseph Louis Lagrange (1736-1813), and Pierre Simon de Laplace (1749-1827).

Анализ Фурье(спектральный анализ) назван по имени **Джеана Баптиста Джозефа Фоуриера** (1768-1830), французского математика и физика. (Фурье объявлен: *for-ē-a*, и всегда печатается прописными буквами). В то время как многие вносили вклад в исследования, Фурье интересен своими математическими открытиями и пониманием практической полезности методов. Фурье был заинтересован распространением теплоты, и представил в 1807 бумагу на использовании синусоид в Институт Франции, чтобы представить температурные распределения. Бумага содержала спорное утверждение, что любой непрерывный периодический сигнал может быть представлен как сумма должным образом выбранных синусоидальных волн. Среди рецензентов были два из наиболее известных в хронологии математиков, Лагранж Джозеф Луи (1736-1813), и Pierre Саймон Лаплас (1749-1827).

While Laplace and the other reviewers voted to publish the paper, Lagrange adamantly protested. For nearly 50 years, Lagrange had insisted that such an approach could not be used to represent signals with *corners*, i.e., discontinuous slopes, such as in square waves. The Institut de France bowed to the prestige of Lagrange, and rejected Fourier's work. It was only after Lagrange died that the paper was finally published, some 15 years later. Luckily, Fourier had other things to keep him busy, political activities, expeditions to Egypt with Napoleon, and trying to avoid the guillotine after the French Revolution (literally!).

В то время как Лаплас и другие рецензенты голосовали, чтобы издать бумагу, Лагранж непреклонно возражал. Почти 50 лет, Лагранж упорно утверждал, что такой подход не может использоваться, чтобы представить сигналы с *углами*, то есть, прерываемые наклонами, типа прямоугольных меандров (волн, колебаний). Институт Франции, преклонясь перед престижем Лагранжа, отклонил работу Фурье. Эта бумага была издана только после того, как Лагранж умер, это приблизительно 15 годами позже. К счастью, Фурье имел другие дела сохраняющие его занятость, политические действия, экспедиции в Египет с Наполеоном, и попытка(трудное уклонение) избежать гильотины после Французской Революции (буквально!).(с трудом избежать гильотины?)

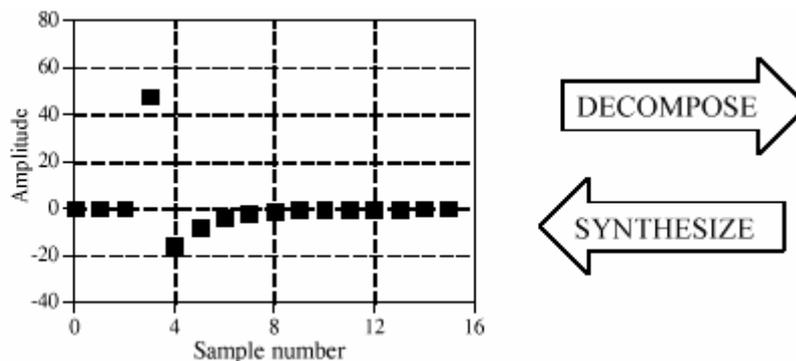
Who was right? It's a split decision. Lagrange was correct in his assertion that a summation of sinusoids cannot form a signal with a corner. However, you can get *very* close. So close that the difference between the two has *zero energy*. In this sense, Fourier was right, although 18th century science knew little about the concept of energy. This phenomenon now goes by the name: *Gibbs Effect*, and will be discussed in Chapter 11.

Кто был прав? Это – решение, при котором голоса разделились(Частично Лагранж и частично Фурье). Лагранж был прав, утверждая, что суммирование(совокупность) синусоид не может формировать сигнал с уголковыми вершинами. Однако, Вы можете подойти к этому *очень* близко. Так близко, что разность между этими двумя имеет *нулевую энергию*. Хотя наука 18-ого столетия знала немного относительно концепции энергии, в этом смысле, был прав Фурье. Это явление теперь известно под названием *Эффект Гиббса*, и будет обсуждено в главе 11.

Figure 8-1 illustrates how a signal can be decomposed into sine and cosine waves. Figure (a) shows an example signal, 16 points long, running from sample number 0 to 15. Figure (b) shows the Fourier decomposition of this signal, nine cosine waves and nine sine waves, each with a different frequency and amplitude. Although far from obvious, these 18 sinusoids add to produce the waveform in (a). It should be noted that the objection made by Lagrange only applies to *continuous* signals. For *discrete* signals, this decomposition is mathematically exact. There is no difference between the signal in (a) and the *sum* of the signals in (b), just as there is no difference between 7 and 3+4.

Рисунок 8-1 иллюстрирует, как сигнал может быть разложен в волны косинуса и синуса. Рисунок (а) показывает пример сигнала длительность 16 точек, выполняющийся(текущий) от выборки 0 до выборки 15. Рисунок (б) показывает декомпозицию Фурье этого сигнала, девять волн косинуса и девять волн синуса, с различной частотой и амплитудой каждая. Хотя далеко от очевидного(явного), эти 18 синусоид складываясь, производят форму волны в (а). Должно быть отмечено, что возражение, сделанное Лагранжем относится только к *непрерывным* сигналам. Для *дискретных* сигналов, эта декомпозиция математически точна. Не имеется никакой разности между сигналом в (а) и сумме сигналов в (б), также, как не имеется никакой разности между 7 и 3+4.

FIGURE 8-1a
(see facing page)



Why are sinusoids used instead of, for instance, square or triangular waves? Remember, there are an infinite number of ways that a signal can be decomposed. The goal of decomposition is to end up with something *easier* to deal with than the original signal. For example, impulse decomposition allows signals to be examined one point at a time, leading to the powerful technique of convolution. The component sine and cosine waves are simpler than the original signal because they have a property that the original signal does not have: *sinusoidal fidelity*. As discussed in Chapter 5, a sinusoidal input to a system is guaranteed to produce a sinusoidal output. Only the amplitude and phase of the signal can change; the frequency and wave shape must remain the same. Sinusoids are the only waveform that have this useful property. While square and triangular decompositions are *possible*, there is no general reason for them to be *useful*.

Почему используются синусоиды, например, прямоугольные или треугольные волны? Помните, имеется бесконечное число способов, которыми сигнал может быть разложен. Цель декомпозиции(разложения) состоит в том, чтобы закончить кое-чем с чем *проще* иметь дело чем с первоначальным сигналом. Например, декомпозиция импульса позволяет сигналам быть исследованным одной точкой одновременно(в время), ведя к мощной методике конволюции. Составляющие волны синуса и волны косинуса более просты чем первоначальный сигнал, потому что они имеют свойство, которого первоначальный сигнал не имеет: *синусоидальная точность(верность)*. Как обсуждено в главе 5, синусоидальный ввод к системе, как гарантируют произведение синусоидального выхода. Только амплитуда и фаза сигнала могут изменяться; частота и форма волны должны остаться теми же самими. Синусоиды - единственная форма волны, которые имеют это полезное свойство. В то время как квадратные и треугольные декомпозиции *возможны*, не имеется никакой общей причины(аргумента) для них, чтобы быть *полезным*.

The general term: *Fourier transform*, can be broken into four categories, resulting from the four basic types of signals that can be encountered. A signal can be either *continuous* or *discrete*, and it can be either *periodic* or *aperiodic*. The combination of these two features generates the four ategories, described below and illustrated in Fig. 8-2.

Общий термин: *преобразование Фурье*, может быть разбит на четыре категории, следуя из четырех основных типов сигналов, с которыми можно сталкиваться. Сигнал может быть *непрерывен* или *дискретен*, и может быть *периодическим* или *апериодическим*. Комбинация из этих двух особенностей формирует(порождает) четыре категории, описанные ниже и иллюстрированные рисунком 8-2.

Aperiodic-Continuous

This includes, for example, decaying exponentials and the Gaussian curve. These signals extend to both positive and negative infinity *without* repeating in a periodic pattern. The Fourier Transform for this type of signal is simply called the **Fourier Transform**.

Апериодически - непрерывные

Это включает, например, затухающие экспоненциальные(показательные функции) и кривую нормального распределения(кривую гаусса). Эти сигналы простираются и на положительную и на отрицательную бесконечность без того, чтобы повториться в периодическом образце. Преобразование Фурье для этого типа сигнала просто называется **Трансформантой Фурье**.

Periodic-Continuous

Here the examples include: sine waves, square waves, and any waveform that repeats itself in a regular pattern from negative to positive infinity. This version of the Fourier transform is called the **Fourier Series**.

Периодически - непрерывные

Здесь примеры включают: волны синуса, квадратные волны, и любую форму волны, которая повторяет себя в регулярном образце от отрицательной до положительной бесконечности. Эта версия преобразования Фурье называется **Рядом Фурье**.

Aperiodic-Discrete

These signals are only defined at discrete points between positive and negative infinity, and do not repeat themselves in a periodic fashion. This type of Fourier transform is called the **Discrete Time Fourier Transform**.

Апериодически - дискретный

Эти сигналы определены только в дискретных точках между положительной и отрицательной бесконечностью, и не повторяют себя периодическим способом. Этот тип преобразования Фурье называется **Преобразование Фурье с Дискретным Временем**.

Periodic-Discrete

These are discrete signals that repeat themselves in a periodic fashion from negative to positive infinity. This class of Fourier Transform is sometimes called the Discrete Fourier Series, but is most often called the **Discrete**.

Периодически - дискретный

Они - дискретные сигналы, которые повторяют сами себя периодическим способом от отрицательной до положительной бесконечности. Этот класс преобразования Фурье иногда называется Дискретным Рядом Фурье, но наиболее часто он называется **Дискретным**.

Fourier Transform.

You might be thinking that the names given to these four types of Fourier transforms are confusing and poorly organized. You're right; the names have evolved rather haphazardly over 200 years. There is nothing you can do but memorize them and move on. These four classes of signals all extend to positive and negative *infinity*. Hold on, you say! What if you only have a

finite number of samples stored in your computer, say a signal formed from 1024 points. Isn't there a version of the Fourier Transform that uses finite length signals? No, there isn't. Sine and cosine waves are *defined* as extending from negative infinity to positive infinity. You cannot use a group of infinitely long signals to synthesize something finite in length. The way around this dilemma is to make the finite data *look like* an infinite length signal. This is done by imagining that the signal has an infinite number of samples on the left and right of the actual points. If all these "imagined" samples have a value of zero, the signal looks *discrete* and *aperiodic*, and the Discrete Time Fourier Transform applies. As an alternative, the imagined samples can be a duplication of the actual 1024 points. In this case, the signal looks discrete and periodic, with a period of 1024 samples. This calls for the Discrete Fourier Transform to be used.

Трансформанта Фурье.

Вы могли бы думать, что названия, данные этим четырем типам преобразований Фурье запутывающие и плохо организованы. Вы правы; названия развились довольно случайно(без системно) более чем 200 лет. Не имеется ничего, что Вы можете сделать, кроме как запомнить их и манипулировать ими. Все эти четыре класса сигналов простираются(распространяются) от положительной до отрицательной *бесконечности*. Против, скажете Вы! Если Вы имеете только конечное число выборок, сохраненных в вашем компьютере, говорят сигнал, сформирован из 1024 точек. Не имеется версии трансформанты Фурье, которая использует сигналы конечной длины? Нет, не имеется. Волны синуса и волны косинуса *определены* как распространяющиеся от отрицательной до положительной бесконечности. Вы не можете использовать группу бесконечно длинных сигналов, чтобы синтезировать кое-что конечное в длине. Путь обойти эту дилемму состоит в том, чтобы заставить конечные данные *напомнить(быть похожими на)* сигнал бесконечной длины. Это делается, воображая, что сигнал имеет бесконечное число выборок слева и справа от фактических точек. Если все эти "предполагаемые" выборки имеют значение нуля, виды(типы, формы) сигнала, *дискретные* и *апериодические*, и применяется Преобразование Фурье с Дискретным Временем. Как альтернатива, предполагаемые выборки могут быть дубликатами(повторяющимися) фактических 1024 точек. В этом случае, сигнал выглядит дискретным и периодическим, с периодом 1024 выборки. Это требует использования Дискретного Преобразования Фурье.

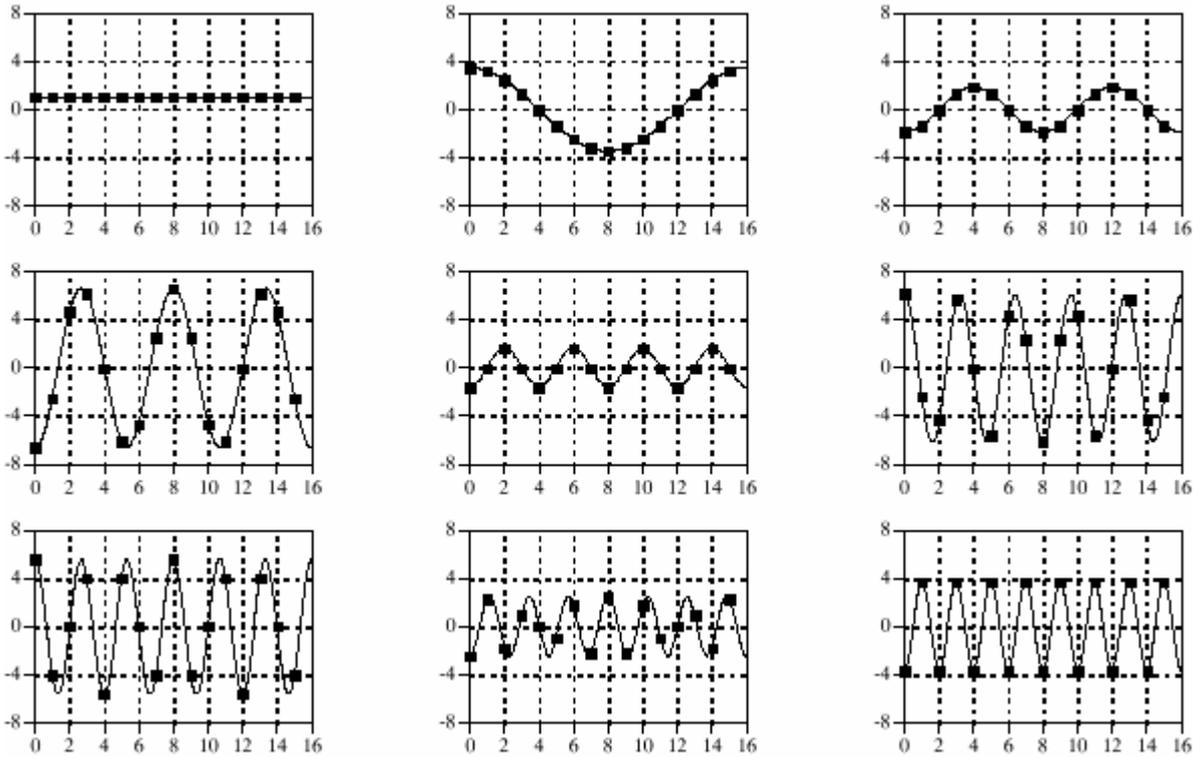
As it turns out, an *infinite* number of sinusoids are required to synthesize a signal that is *aperiodic*. This makes it impossible to calculate the Discrete Time Fourier Transform in a computer algorithm. By elimination, the only type of Fourier transform that can be used in DSP is the DFT. In other words, digital computers can only work with information that is *discrete* and *finite* in length. When you struggle with theoretical issues, grapple with homework problems, and ponder mathematical mysteries, you may find yourself using the first three members of the Fourier transform family. When you sit down to your computer, you will only use the DFT. We will briefly look at these other Fourier transforms in future chapters. For now, concentrate on understanding the Discrete Fourier Transform.

Как оказалось, чтобы синтезировать сигнал который является *апериодическим*, требуется *бесконечное* число синусоид, Это делает невозможным вычислить Преобразование Фурье с Дискретным Временем в компьютерном(машинном) алгоритме. Исключение, единственный тип преобразования Фурье, который может использоваться в ЦОС – ДПФ. Другими словами, цифровые компьютеры могут работать только с информацией, которая является *дискретной* и *конечной* в длине. Когда Вы боретесь с теоретическими проблемами, пытайтесь решить проблемы домашней работой, и обдумыванием математической тайны, Вы можете использовать первые три члена семейства преобразований Фурье. Когда Вы садитесь за ваш компьютер, Вы будете использовать

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

только ДПФ. в будущих главах мы кратко рассмотрим 3 другие преобразования Фурье. Пока же, упор на понимание Дискретного Преобразования Фурье.

Cosine Waves



Sine Waves

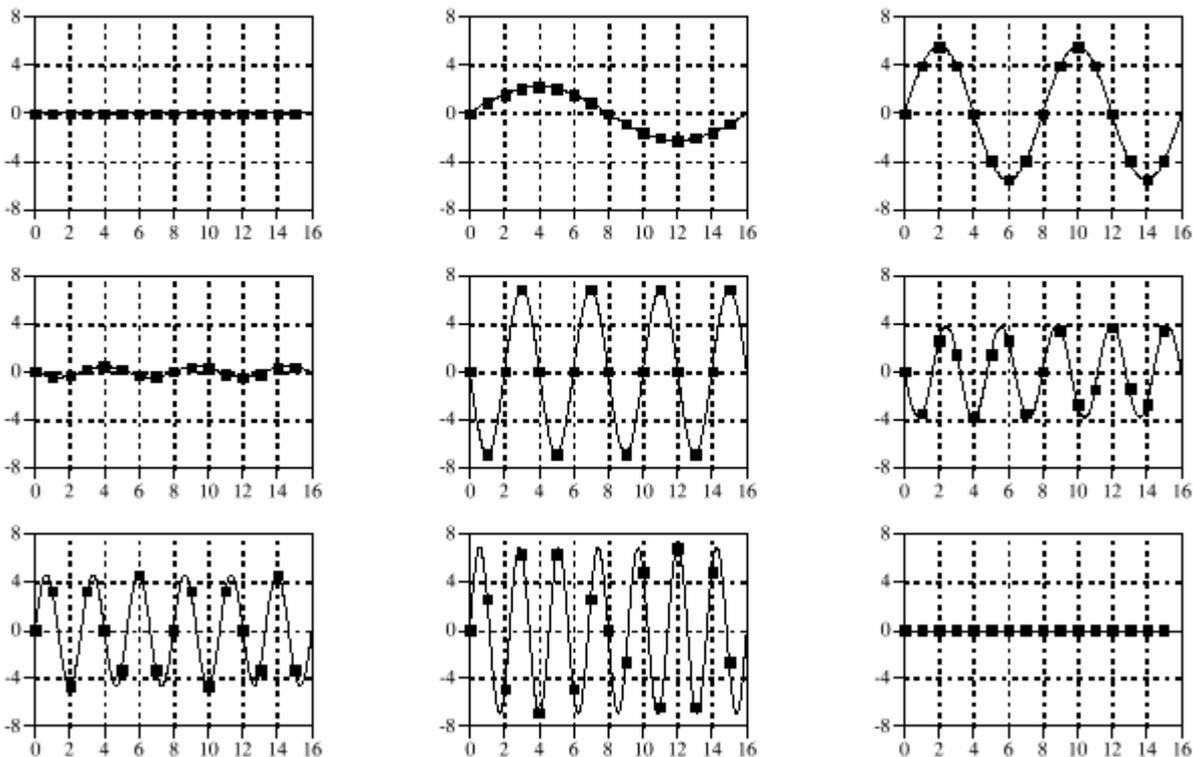


FIGURE 8-1b. Example of Fourier decomposition.

A 16 point signal (opposite page) is decomposed into 9 cosine waves and 9 sine waves. The frequency of each sinusoid is fixed; only the amplitude is changed depending on the shape of the waveform being decomposed

РИСУНОК 8-1b. Пример декомпозиции Фурье.

16 точек сигнала (противоположная страница – на рисунке 8b?) разложен на 9 волн косинуса и 9 волн синуса. Частота каждой синусоиды постоянна; только амплитуда изменена в зависимости от формы разлагаемой формы волны.

Type of Transform	Example Signal
Fourier Transform <i>signals that are continuous and aperiodic</i>	
Fourier Series <i>signals that are continuous and periodic</i>	
Discrete Time Fourier Transform <i>signals that are discrete and aperiodic</i>	
Discrete Fourier Transform <i>signals that are discrete and periodic</i>	

FIGURE 8-2

Illustration of the four Fourier transforms. A signal may be continuous or discrete, and it may be periodic or aperiodic. Together these define four possible combinations, each having its own version of the Fourier transform. The names are not well organized; simply memorize them.

РИСУНОК 8-2 . Иллюстрация этих четырех преобразований Фурье.

Сигнал может быть непрерывен или дискретен, и может быть периодическим или аperiodическим. Вместе они определяют четыре возможных комбинации, каждая имеет его собственную версию преобразования Фурье. Названия плохо организованы; просто запомните их.

Look back at the example DFT decomposition in Fig. 8-1. On the face of it, it appears to be a 16 point signal being decomposed into 18 sinusoids, each consisting of 16 points. In more formal terms, the 16 point signal, shown in (a), must be viewed as a single period of an infinitely long periodic signal. Likewise, each of the 18 sinusoids, shown in (b), represents a 16 point segment from an infinitely long sinusoid. Does it really matter if we view this as a 16 point signal being synthesized from 16 point sinusoids, or as an infinitely long periodic signal being synthesized from infinitely long sinusoids? The answer is: *usually no, but sometimes, yes*. In upcoming chapters we will encounter properties of the DFT that seem baffling if the signals are viewed as finite, but become obvious when the periodic nature is considered. The key point to understand is that this periodicity is invoked in order to use a *mathematical tool*, i.e., the DFT. It is usually meaningless in terms of where the signal originated or how it was acquired.

Посмотрим назад на пример декомпозиции ДПФ на рис 8-1. На лице этого, кажется, есть 16 точек сигнала, разложенные на 18 синусоид, каждая из которых состоит из 16 точек. В более формальном выражении, 16 точек сигнала, показанные в (а), должны рассматриваться как единственная(отдельная) точка бесконечно длинного периодического сигнала. Аналогично, каждая из 18 синусоид, показанных в (b), представляет 16 точек сегмента бесконечно длинной синусоиды. Действительно ли имеет значение, рассматриваем ли мы это как 16 точек сигнала синтезируемые от 16 точек синусоиды, или как бесконечно длинный периодический сигнал, синтезируемый от бесконечно длинной синусоиды? Ответ: *не всегда, но иногда, да*. В последующих главах мы столкнемся со свойствами ДПФ, которые кажутся затруднительными(неблагоприятными), если сигналы рассматриваются как конечные, но станут очевидны, когда рассматривается периодический характер. Ключевой пункт, чтобы понять - то, что эта периодичность вызвана, чтобы использовать *математический инструмент*, то есть, ДПФ. Не имеет значения условие того, где сигнал произошел или как он был приобретен.

Each of the four Fourier Transforms can be subdivided into **real** and **complex** versions. The real version is the simplest, using ordinary numbers and algebra for the synthesis and decomposition. For instance, Fig. 8-1 is an example of the **real DFT**. The complex versions of the four Fourier transforms are immensely more complicated, requiring the use of *complex numbers*. These are numbers such as: $3+4j$, where j is equal to $\sqrt{-1}$ (electrical engineers use the variable j , while mathematicians use the variable, i). Complex mathematics can quickly become overwhelming, even to those that specialize in DSP. In fact, a primary goal of this book is to present the fundamentals of DSP *without* the use of complex math, allowing the material to be understood by a wider range of scientists and engineers. The complex Fourier transforms are the realm of those that specialize in DSP, and are willing to sink to their necks in the swamp of mathematics. If you are so inclined, Chapters 30-33 will take you there.

Каждое из этих четырех преобразований Фурье может быть подразделено на **вещественные** и **комплексные** версии. Вещественная версия самая простая, используя обычные числа и алгебру для синтеза и декомпозиции. Например, рис. 8-1 - пример **вещественного ДПФ**. Комплексные версии этих четырех преобразований Фурье очень сильно усложнены, требуя использование *комплексных чисел*. Они - числа типа: $3+4j$, где j равно $\sqrt{-1}$ (инженеры-электрики, используют переменную j , в то время как математики используют переменную, i). Комплексная математика может быстро стать подавляющей, даже для тех, кто специализируется в ЦОС. Фактически, первичная цель этой книги состоит в том, чтобы представить основные принципы ЦОС *без* использования комплексной математики, позволяя материалу быть понятным для более широкого диапазона ученых и инженеров. Комплексные преобразования Фурье - область тех, кто специализируется в ЦОС, и желает по шею погрузиться в болото математики. Если Вы склонны к этому, Вас ожидают главы 30-33.

The mathematical term: **transform**, is extensively used in Digital Signal Processing, such as: Fourier transform, Laplace transform, Z transform, Hilbert transform, Discrete Cosine transform, etc. Just what is a transform? To answer this question, remember what a *function* is. A function is an algorithm or procedure that changes one value into another value. For example, $y = 2x + 1$ is a function. You pick some value for x , plug it into the equation, and out pops a value for y . Functions can also change *several* values into a single value, such as: $y = 2a + 3b + 4c$, where a , b , and c are changed into y .

Математический термин: **трансформанта**, экстенсивно(широко) используется в Цифровой Обработке Сигналов, типа: трансформанта Фурье, трансформанта Лапласа, Z
(с) АВТЭКС, Санкт-Петербург, <http://www.autex.spb.ru>, e-mail: info@autex.spb.ru

трансформанта, Гильбертова трансформанта, Дискретная трансформанта Косинуса, и т.д. Только, что является трансформантой? Чтобы ответить на этот вопрос, вспомните, что такое функция. Функция - алгоритм или процедура, которая изменяет одно значение в другое значение. Например, $y = 2x+1$ - функция. Вы выбираете некоторое значение для x , подставляете это в уравнение, и получаете значение для y . Функции могут также изменять несколько значений в одиночное(единое) значение, типа: $y = 2a+3b+4c$, где a , b , и c изменены(заменены) на y .

Transforms are a direct extension of this, allowing both the input and output to have *multiple* values. Suppose you have a signal composed of 100 samples. If you devise some equation, algorithm, or procedure for changing these 100 samples into another 100 samples, you have yourself a transform. If you think it is useful enough, you have the perfect right to attach your last name to it and expound its merits to your colleagues. (This works best if you are an eminent 18th century French mathematician). Transforms are not limited to any specific type or number of data. For example, you might have 100 samples of discrete data for the input and 200 samples of discrete data for the output. Likewise, you might have a continuous signal for the input and a continuous signal for the output. Mixed signals are also allowed, discrete in and continuous out, and vice versa. In short, a transform is any fixed procedure that changes one chunk of data into another chunk of data. Let's see how this applies to the topic at hand: the Discrete Fourier transform.

Трансформанты - прямое продолжение этого, позволяя обоим и вводу и выводу иметь *многочисленные(кратные)* значения. Предположим, что Вы составляете сигнал из 100 выборок. Если Вы придумаете(изобретете) некоторое уравнение, алгоритм, или процедуру для изменения(замены) этих 100 выборок на другие 100 выборок, Вы самостоятельно(сами) получили трансформанту. Если Вы думаете, что это достаточно полезно, Вы имеете полное право прикрепить вашу фамилию к этому(назвать это вашей фамилией) и разьяснять ее достоинства вашим коллегам. (Это работает лучше всего, если Вы выдающийся Французский математик 18-ого столетия). Трансформанты не ограничены любым специфическим типом или числом данных. Например, Вы могли бы иметь 100 выборок дискретных данных для ввода и 200 выборок дискретных данных для выхода. Аналогично, Вы могли бы иметь непрерывный сигнал для ввода и непрерывный сигнал для вывода. Смешанные сигналы также позволяют, дискретные в непрерывные, и наоборот. Короче говоря, трансформанта - любая установленная процедура, которая изменяет(заменяет) одну порцию данных на другую порцию данных. Давайте посмотрим, как это применяется(относится) к текущей теме(разделу): Дискретное преобразование Фурье.

Notation and Format of the Real DFT

Система обозначений(Запись) и Формат Реального ДПФ

As shown in Fig. 8-3, the discrete Fourier transform changes an N point input signal into two $N/2+1$ point output signals. The input signal contains the signal being decomposed, while the two output signals contain the *amplitudes* of the component sine and cosine waves (scaled in a way we will discuss shortly). The input signal is said to be in the **time domain**. This is because the most common type of signal entering the DFT is composed of samples taken at regular intervals of *time*. Of course, any kind of sampled data can be fed into the DFT, regardless of how it was acquired. When you see the term "time domain" in Fourier analysis, it may actually refer to samples taken over time, or it might be a general reference to any discrete signal that is being

decomposed. The term **frequency domain** is used to describe the amplitudes of the sine and cosine waves (including the special scaling we promised to explain).

Как показано на рис. 8-3, дискретное преобразование Фурье заменяет сигнал ввода точки N двумя сигналами точки выхода $N/2+1$. Входной сигнал содержит анализируемый (распадающийся, разлагаемый) сигнал, в то время как два сигнала выхода содержат *амплитуды* составляющих волн синуса и косинуса (масштабируемые способом, который мы коротко (вскоре) обсудим). Входной сигнал, как считают, находится в **домене времени**. Это - то, потому что наиболее общий (обычный) тип сигнала, вводящего (входящего) ДПФ составлен из выборок, принятых равномерно *времени*. Конечно, любой вид (тип) произведенных выборок данных может быть подан в ДПФ, независимо от того, как это было приобретено. Когда Вы видите термин "домен времени" в анализе Фурье, это может фактически относиться к выборкам, взятым через какое-то время, или это могло бы быть общей ссылкой на любой дискретный сигнал, который разлагается. Термин **Домен частоты** используется, чтобы описать амплитуды волн синуса и косинуса (включая специальное масштабирование, которое мы обещали объяснить).

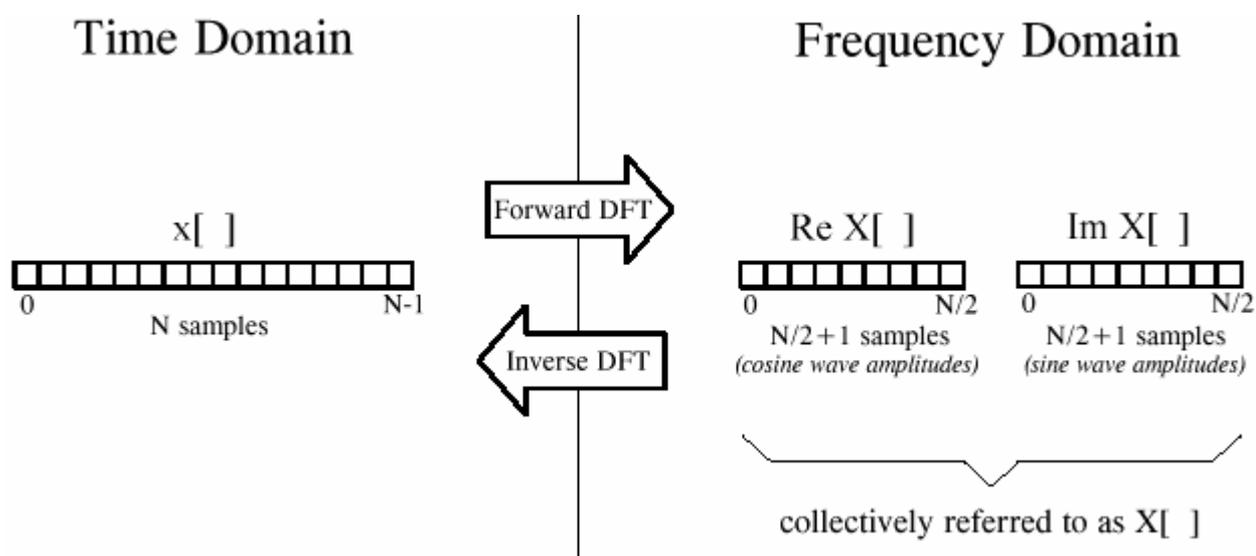


FIGURE 8-3

DFT terminology. In the time domain, $x[n]$ consists of N points running from 0 to $N-1$. In the frequency domain, the DFT produces two signals, the real part, written $Re X[n]$, and the imaginary part, written $Im X[n]$. Each of these frequency domain signals are $N/2 + 1$ points long, and run from 0 to $N/2$. The Forward DFT transforms from the time domain to the frequency domain, while the Inverse DFT transforms from the frequency domain to the time domain. (Take note: this figure describes the **real DFT**. The **complex DFT**, discussed in Chapter 31, changes N complex points into another set of N complex points).

РИСУНОК 8-3

Терминология ДПФ. В домене времени, $x[n]$ состоит из точек N , выполняющихся от 0 до $N-1$. В частотном домене, ДПФ производит два сигнала, вещественную часть, записываемую: $Re X[n]$, и мнимую часть, записываемую $Im X[n]$. Каждый из этих частотных доменов сигнала - длится $N/2+1$ точек, и работает (протекает) от 0 до $N/2$. Прямой ДПФ преобразовывает домен времени в частотный домен, в то время как Обратный ДПФ преобразовывает частотный домен в домен времени. (Заметьте: рисунок описывает **реальный (вещественный) ДПФ**. **Комплексный ДПФ**, обсужденный в главе 31, изменяет (заменяет) комплексные точки N в другой набор комплексных N точек).

The frequency domain contains exactly the same information as the time domain, just in a different form. If you know one domain, you can calculate the other. Given the time domain signal, the process of calculating the frequency domain is called **decomposition, analysis, the forward DFT**, or simply, **the DFT**. If you know the frequency domain, calculation of the time

domain is called **synthesis**, or the **inverse DFT**. Both synthesis and analysis can be represented in equation form and computer algorithms.

Частотный домен содержит точно ту же самую информацию как домен времени, только в другой форме. Если Вы знаете один домен, Вы можете вычислить другой. Учитывая сигнал домен(а) времени(я), процесс вычисления частотного домена называется **декомпозицией, анализом, прямым ДПФ**, или просто, **ДПФ**. Если Вы знаете частотный домен, вычисление домена времени называется **синтезом**, или **обратным ДПФ**. И синтез и анализ могут быть представлены в форме уравнения и компьютерных(машинных) алгоритмах.

The number of samples in the time domain is usually represented by the **variable N** . While N can be any positive integer, a power of two is usually chosen, i.e., 128, 256, 512, 1024, etc. There are two reasons for this. First, digital data storage uses binary addressing, making powers of two a natural signal length. Second, the most efficient algorithm for calculating the DFT, the Fast Fourier Transform (FFT), usually operates with N that is a power of two. Typically, N is selected between 32 and 4096. In most cases, the samples run from 0 to $N-1$, rather than 1 to N .

Число выборок в домене времени обычно представляется **переменной N** . В то время как N может быть любым положительным целым числом, обычно выбирается показатель два, то есть, 128, 256, 512, 1024, и т.д. Для этого имеется две причины. Во первых, хранение данных в цифровой форме использует адресацию в двоичном коде, создание степеней два естественная длина сигнала. Во вторых, наиболее эффективный алгоритм для вычисления ДПФ, Быстрое преобразование Фурье (БПФ), обычно работает с N , который является показателем(степенью) два. Как правило, N отобран между 32 и 4096. В большинстве случаев, выборки, выполняются(протекают) от 0 до $N-1$, скорее чем 1 к N .

Standard DSP notation uses **lower case letters** to represent time domain signals, such as $x[n]$, $y[n]$ and $z[n]$. The corresponding **upper case letters** are used to represent their frequency domains, that is, $X[n]$, $Y[n]$, and $Z[n]$. For illustration, assume an N point time domain signal is contained in $x[n]$. The frequency domain of this signal is called $X[n]$, and consists of two parts, each an array of $N/2+1$ samples. These are called the **Real part of $X[n]$** , written as: **$ReX[n]$** , and the **Imaginary part of $X[n]$** : written as: **$Im X[n]$** . The values $Re X[n]$ in are the amplitudes of the cosine waves, while the values in $Im X[n]$ are the amplitudes of the sine waves (not worrying about the scaling factors for the moment). Just as the time domain runs from $x[0]$ to $x[N-1]$, the frequency domain signals run from $ReX[0]$ to $ReX[N/2]$, and from $ImX[0]$ to $Im X[N/2]$. Study these notations carefully; they are critical to understanding the equations in DSP. Unfortunately, some computer languages don't distinguish between lower and upper case, making the variable names up to the individual programmer. The programs in this book use the array $XX[n]$ to hold the time domain signal, and the arrays $REX[n]$ and $IMX[n]$ to hold the frequency domain signals.

Стандарт системы обозначений ЦОС для представления сигналов домена времени использует **символы строчных букв**, типа, $x[n]$, $y[n]$ и $z[n]$. Соответствующие **символы верхнего регистра** используются, чтобы представить их частотные домены, то есть $X[n]$, $Y[n]$, и $Z[n]$. Для иллюстрации(пояснения), примите(предположите) точка N сигнала домена времени содержится в $x[n]$. Частотный домен этого сигнала называется $X[n]$, и состоит из двух частей, каждый массив из выборок $N/2+1$. Они называются **Вещественной частью из $X[n]$** , записываемой как: **$ReX[n]$** , и **Мнимой частью из $X[n]$** : записываемой как: **$Im X[n]$** . Значения $Re X[n]$ - амплитуды волн косинуса, в то время как значения $Im X[n]$ - амплитуды волн синуса (не вызывающий беспокойства относительно коэффициентов(факторов) масштабирования в течение(для) момента). Также, как домен (с) АВТЭКС, Санкт-Петербурга, <http://www.autex.spb.ru>, e-mail: info@autex.spb.ru

времени выполняется от $x[0]$ к $x[N-1]$, сигналы частотного домена, выполненные от $ReX[0]$ к $ReX[N/2]$, и от $ImX[0]$ к $Im X[N/2]$. Изучите эти системы обозначений тщательно; Они критические(важны) для понимания уравнений в ЦОС. К сожалению, некоторые машинные языки не различают нижний и верхний регистр, делая имена переменной по усмотрению индивидуального программиста. Программы в этой книге используют массив $XX []$, чтобы считать сигнал доменом времени, и массивы $REX[]$ и $IMX[]$ чтобы проводить(считать) сигналы частотным доменом.

The names *real part* and *imaginary part* originate from the complex DFT, where they are used to distinguish between *real* and *imaginary* numbers. Nothing so complicated is required for the real DFT. Until you get to Chapter 31, simply think that "real part" means the *cosine wave amplitudes*, while "imaginary part" means the *sine wave amplitudes*. Don't let these suggestive names mislead you; everything here uses ordinary numbers. Likewise, don't be misled by the *lengths* of the frequency domain signals. It is common in the DSP literature to see statements such as: "The DFT changes an N point time domain signal into an N point frequency domain signal." This is referring to the *complex DFT*, where each "point" is a complex number (consisting of real and imaginary parts). For now, focus on learning the real DFT, the difficult math will come soon enough.

Вещественная часть и *мнимая часть* названий(имен) происходят из комплексного ДПФ, где они используются, чтобы различать *реальные* и *комплексные* числа. Ничто столь сложного не требуется для реального(вещественного) ДПФ. Пока Вы не приметесь за главу 31, просто думайте, что "вещественная часть" означает *амплитуды волны косинуса*, в то время как "мнимая часть" означает *амплитуды волны синуса*. Не позволяйте этим наводящим на размышления названиям ввести Вас в заблуждение; все здесь использует обычные числа. Аналогично, не будьте введены в заблуждение *длинами* сигналов частотных доменов. Это обычно в литературе ЦОС, чтобы видеть инструкции(формулировки) типа: "ДПФ изменяет(заменяет) точку N сигнала домена времени в точку N сигнала частотного домена." Это - что касается *комплексного(вещественного) ДПФ*, где каждая "точка" - комплексное число (состоящее из реальной и мнимой частей). Пока, фокус(центр внимания) направлен на изучение реального(вещественного) ДПФ, трудная математика начнется достаточно скоро.

The Frequency Domain's Independent Variable **Независимая Переменная Частотного Домена**

Figure 8-4 shows an example DFT with $N=128$. The time domain signal is contained in the array: $x[0]$ to $x[127]$. The frequency domain signals are contained in the two arrays: $ReX[0]$ to $ReX[64]$, and $Im X[0]$ to $Im X[64]$. Notice that 128 points in the time domain corresponds to 65 points in each of the frequency domain signals, with the frequency indexes running from 0 to 64. That is, N points in the time domain corresponds to $N/2+1$ points in the frequency domain (not $N/2$ points). Forgetting about this extra point is a common bug in DFT programs.

На рисунке 8-4 показан пример ДПФ с $N=128$. Сигнал домена времени содержится в массиве: $x[0]$ к $x[127]$. Сигналы частотного домена содержатся в двух массивах: $ReX[0]$ к $ReX[64]$, и $ImX[0]$ к $ImX[64]$. Обратите внимание, что 128 точек в домене времени соответствуют 65 точкам в каждом из сигналов частотного домена, с частотными индексами, выполняющимися от 0 до 64. То есть N точек в домене времени соответствуют $N/2+1$ точкам в частотном домене (не $N/2$ точкам)). Упущение относительно этого дополнительного пункта - обычная ошибка в программах ДПФ.

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

The horizontal axis of the frequency domain can be referred to in **four different ways**, all of which are common in DSP. In the first method, the horizontal axis is labeled from 0 to 64, corresponding to the 0 to $N/2$ samples in the arrays. When this labeling is used, the index for the frequency domain is an integer, for example, $Re X[k]$ and $Im X[k]$, where k runs from 0 to $N/2$ in steps of one. Programmers like this method because it is how they write code, using an index to access array locations. This notation is used in Fig. 8-4b.

Горизонтальная ось частотного домена может быть упомянута четырьмя различными способами, все из которых обычны в ЦОС. В первом методе, горизонтальная ось помечена от 0 до 64, соответствуя выборкам 0 к $N/2$ в массивах. Когда это маркирование используется, индекс для частотного домена - целое число, например, $ReX[k]$ и $ImX[k]$, где k выполняется от 0 до $N/2$ в шагах один(единица). Программисты подобно этому методу, потому что то, как они записывают код, используя индекс(указатель, номер), чтобы обратиться к расположениям массива. Эта система обозначений используется в рис. 8-4b.

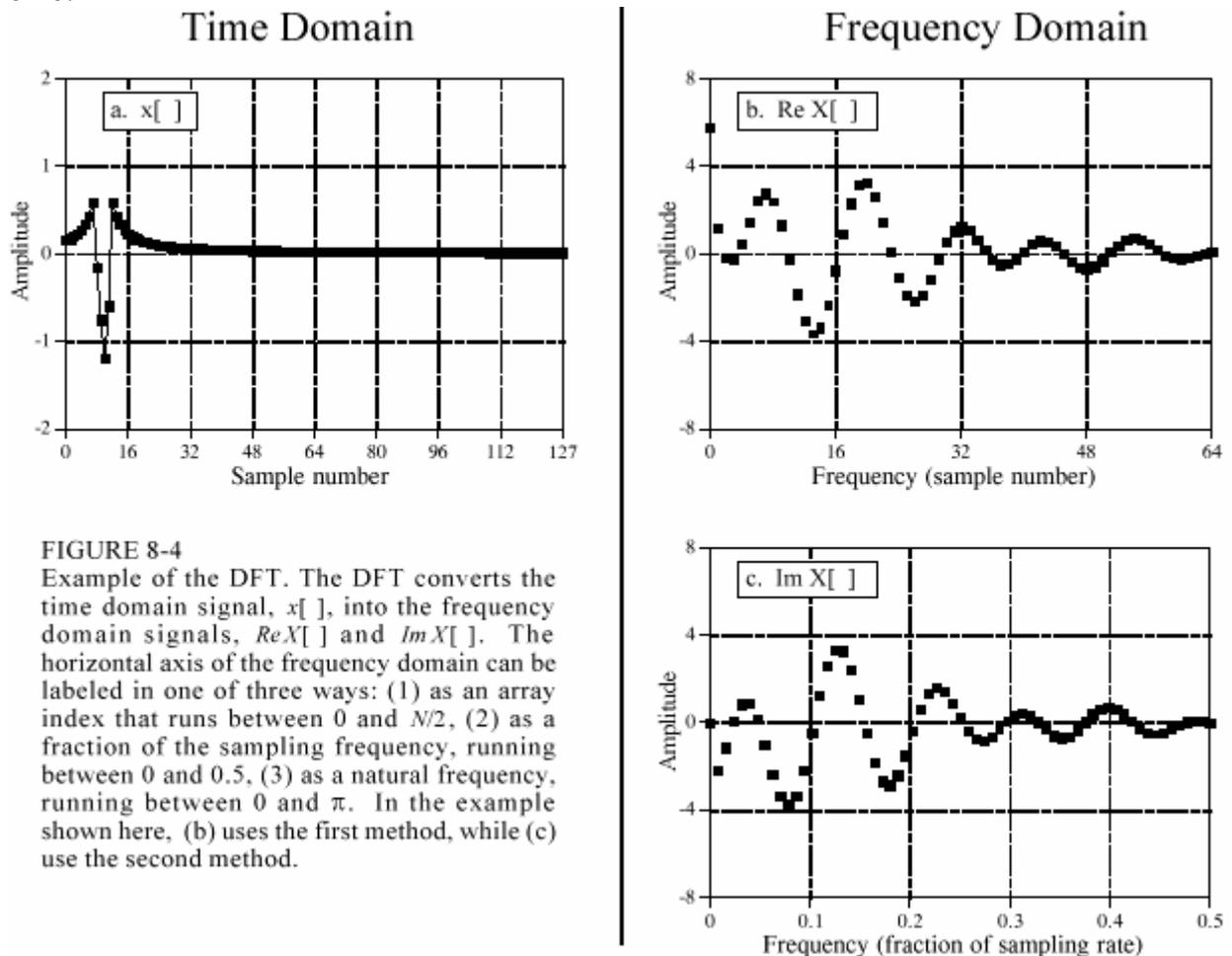


FIGURE 8-4
 Example of the DFT. The DFT converts the time domain signal, $x[n]$, into the frequency domain signals, $ReX[k]$ and $ImX[k]$. The horizontal axis of the frequency domain can be labeled in one of three ways: (1) as an array index that runs between 0 and $N/2$, (2) as a fraction of the sampling frequency, running between 0 and 0.5, (3) as a natural frequency, running between 0 and π . In the example shown here, (b) uses the first method, while (c) use the second method.

FIGURE 8-4
 Example of the DFT. The DFT converts the time domain signal, $x[n]$, into the frequency domain signals, $ReX[k]$ and $ImX[k]$. The horizontal axis of the frequency domain can be labeled in one of three ways: (1) as an array index that runs between 0 and $N/2$, (2) as a fraction of the sampling frequency, running between 0 and 0.5, (3) as a natural frequency, running between 0 and π . In the example shown here, (b) uses the first method, while (c) use the second method.

РИСУНОК 8-4

Пример ДПФ. ДПФ преобразовывает сигнал домена времени, $x[n]$, в сигналы частотного домена, $ReX[k]$ и $ImX[k]$. Горизонтальная ось частотного домена может быть помечена одним из трех способов: (1) как

индекс(номер?) массива, который выполняется между 0 и $N/2$, (2) как дробь(доля,часть) частоты посылок(выборочной частоты), выполняющейся между 0 и 0.5, (3) как собственная(натуральная, собственная) частота, выполняющаяся между 0 и π . В примере показанном здесь, (b) использует первый метод, в то время как (c) используют второй метод.

In the second method, used in (c), the horizontal axis is labeled as a *fraction of the sampling rate*. This means that the values along the horizontal axis always run between 0 and 0.5, since discrete data can only contain frequencies between DC and one-half the sampling rate. The index used with this notation is f , for frequency. The real and imaginary parts are written: $ReX[f]$ and $ImX[f]$, where f takes on $N/2+1$ equally spaced values between 0 and 0.5. To convert from the first notation, k , to the second notation, f , divide the horizontal axis by N . That is, $f = k/N$. Most of the graphs in this book use this second method, reinforcing that discrete signals only contain frequencies between 0 and 0.5 of the sampling rate.

Во втором методе, используемом в (c), горизонтальная ось помечена как дробь(доля, часть) частоты выборки(частоты дискретизации). Это означает, что значения по горизонтальной оси, всегда выполняемой между 0 и 0.5, начиная с дискретных данных могут содержать только частоты между постоянным током(DC) и половиной частоты выборки(дискретизации). Индекс, используемый с этой системой обозначений - f , для частоты. Реальные и мнимые части написаны: $ReX[f]$ и $ImX[f]$, где f принимает $N/2+1$ равно раздельные значения между 0 и 0.5. Преобразовывая из первой системы обозначений, k , к второй системе обозначений, f , делят горизонтальную ось N . То есть $f = k/N$. Большинство графиков в этой книге используют этот второй метод, укрепляя(усиливая, упрочняя), что дискретные сигналы содержат только частоты между 0 и 0.5 частоты выборки(дискретизации).

The third style is similar to the second, except the horizontal axis is multiplied by 2π . The index used with this labeling is ω , a lower case Greek *omega*. In this notation, the real and imaginary parts are written: $ReX[\omega]$ and $ImX[\omega]$, where ω takes on $N/2+1$ equally spaced values between 0 and π . The parameter, ω , is called the **natural frequency**, and has the units of **radians**. This is based on the idea that there are 2π radians in a circle. Mathematicians like this method because it makes the equations shorter. For instance, consider how a cosine wave is written in each of these first three notations: using k : $c[n] = \cos(2\pi kn/N)$, using f : $c[n] = \cos(2\pi fn)$ and using ω : $c[n] = \cos(\omega n)$.

Третий стиль подобен второму, кроме горизонтальной оси умноженной на 2π . Индекс, используемый с этим маркированием ω , строчная греческая буква *omega*. В этой системе обозначений, реальные и мнимые части написаны: $ReX[\omega]$ и $ImX[\omega]$, где ω принимает одинаково раздельные значения $N/2+1$ между 0 и π . Параметр, ω , называется собственной частотой, и измеряется в радианах. Это основано на идее, что в круге содержится 2π радиан. Математики подобны(сходны) этому методу, потому что это делает уравнения короче. Например, рассмотрите, как волна косинуса написана в каждой из этих первых трех систем обозначений: используя k : $c[n] = \cos(2\pi kn/N)$, используя f : $c[n] = \cos(2\pi fn)$ и используя ω : $c[n] = \cos(\omega n)$.

The fourth method is to label the horizontal axis in terms of the analog frequencies used in a *particular* application. For instance, if the system being examined has a sampling rate of 10 kHz (i.e., 10,000 samples per second), graphs of the frequency domain would run from 0 to 5 kHz. This method has the advantage of presenting the frequency data in terms of a *real world* meaning. The disadvantage is that it is tied to a particular sampling rate, and is therefore not applicable to general DSP algorithm development, such as designing digital filters.

Четвертый метод состоит в том, чтобы маркировать горизонтальную ось в единицах аналоговых частот, используемых в специфическом приложении. Например, если исследуемая система имеет частоту выборки(частоту дискретизации) 10 кГц (то есть, 10,000 выборок в секунду), диаграммы(графики) частотного домена работали бы от 0 до 5 кГц. Этот метод имеет преимущество представления частотных данных в терминах реального мирового значения(международных единицах?). Недостаток - то, что это связано со специфической частотой выборки(дискретизации), и поэтому не применимо к развитию генерального(универсального, общего вида) алгоритма ЦОС, типа проектирования цифровых фильтров

All of these four notations are used in DSP, and you need to become comfortable with converting between them. This includes both graphs and mathematical equations. To find which notation is being used, look at the independent variable and its range of values. You should find one of four notations: k (or some other integer index), running from 0 to $N/2$; f , running from 0 to 0.5; ω , running from 0 to π ; or a frequency expressed in hertz, running from DC to one-half of an actual sampling rate.

Все эти четыре системы обозначений используются в ЦОС, и Вам должно стать удобными преобразование между ними. Это включает и диаграммы(графики) и математические уравнения. Чтобы найти, которая система обозначений используется, смотрите на независимую переменную и ее диапазон значений. Вы должны найти одну из четырех систем обозначений: k (или некоторый другой целочисленный индекс), выполняющийся от 0 до $N/2$; f , выполняющееся от 0 до 0.5; ω , выполняющийся от 0 до π ; или частота, выраженная в герцах, выполняющемся от постоянного тока до половины фактической частоты выборки(дискретизации).

DFT Basis Functions

Базисные Функции ДПФ

The sine and cosine waves used in the DFT are commonly called the DFT **basis functions**. In other words, the output of the DFT is a set of numbers that represent amplitudes. The basis functions are a set of sine and cosine waves with *unity* amplitude. If you assign each amplitude (the frequency domain) to the proper sine or cosine wave (the basis functions), the result is a set of *scaled* sine and cosine waves that can be added to form the time domain signal.

Волны синуса и косинуса, используемые в ДПФ обычно называются **базисными функциями** ДПФ. Другими словами, продукция(выход) ДПФ - набор чисел, которые представляют амплитуды. Базисные функции - набор волн синуса и косинуса с амплитудой единицы. Если Вы назначаете каждую амплитуду (частотный домен) к надлежащей волне синуса или косинуса (базисной функции), результат - набор масштабируемых волн синуса и косинуса, которые могут быть сложены, чтобы сформировать сигнал домена времени.

The DFT basis functions are generated from the equations:
Базисные функции ДПФ сгенерированы уравнениями:

EQUATION 8-1

Equations for the DFT basis functions. In these equations, $c_k[i]$ and $s_k[i]$ are the cosine and sine waves, each N points in length, running from $i = 0$ to $N-1$. The parameter, k , determines the frequency of the wave. In an N point DFT, k takes on values between 0 and $N/2$.

$$c_k[i] = \cos(2\pi ki/N)$$

$$s_k[i] = \sin(2\pi ki/N)$$

EQUATION 8-1

Equations for the DFT basis functions. In these equations, $ck[i]$ and $sk[i]$ are the cosine and sine waves, each N points in length, running from $i=0$ to $N-1$. The parameter, k , determines the frequency of the wave. In an N point DFT, k takes on values between 0 and $N/2$

УРАВНЕНИЕ 8-1.

Уравнения для базисных функций ДПФ. В этих уравнениях, $ck[i]$ и $sk[i]$ - волны косинуса и синуса, каждой точки N в длине, выполняющейся от $i=0$ до $N-1$. Параметр, k , определяет частоту волны. В N точках ДПФ, k принимает значения между 0 и $N/2$

where: $ck[i]$ is the cosine wave for the amplitude held in $ReX[k]$, and $sk[]$ is the sine wave for the amplitude held in $Im X[k]$. For example, Fig. 8-5 shows some of the 17 sine and 17 cosine waves used in an $N=32$ point DFT. Since these sinusoids add to form the input signal, they must be the same *length* as the input signal. In this case, each has 32 points running from $i=0$ to 31. The parameter, k , sets the frequency of each sinusoid. In particular $c_1[]$, is the cosine wave that makes *one* complete cycle in N points, $c_5[]$ is the cosine wave that makes *five* complete cycles in N points, etc. This is an important concept in understanding the basis functions; the frequency parameter, k , is equal to the number of complete cycles that occur over the N points of the signal.

Где: $ck[]$ - волна косинуса для амплитуды, поддержанной(удовлетворяющей) $ReX[k]$ и – $sk[]$ волна синуса для амплитуды, проведенной(поддержанной) в $ImX[k]$. Например, рис. 8-5 показывает часть(некоторые) из 17 волн синуса и 17 волн косинуса используя $N=32$ точек в ДПФ. Так как эти синусоиды слагают формируемый входной сигнал, они должны быть той же самой *длины* как входной сигнал. В этом случае, каждый имеет 32 точки, выполняющиеся от $i=0$ до 31. Параметр, k , устанавливает частоту каждой синусоиды. В частности $c_1[]$, является волной косинуса, которая делает *один* полный цикл(период) в N точках, $c_5[]$ - волна косинуса, которая делает *пять* полных циклов(периодов) в N точках, и т.д. Это - важная концепция в понимании базисных функция; частотный параметр, k , является равным числу полных циклов(периодов), которые происходят по N точкам сигнала.

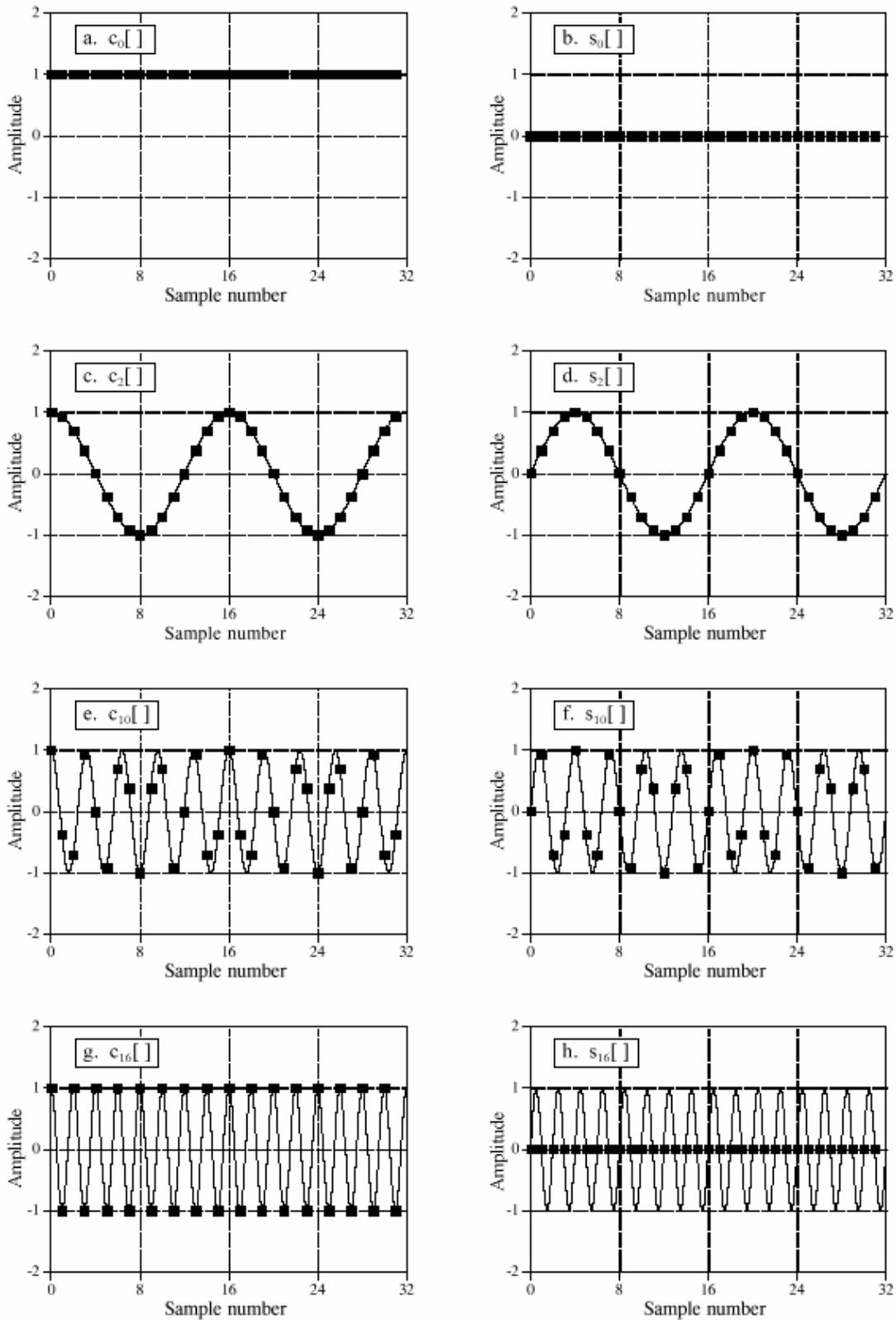


FIGURE 8-5
 DFT basis functions. A 32 point DFT has 17 discrete cosine waves and 17 discrete sine waves for its basis functions. Eight of these are shown in this figure. These are discrete signals; the continuous lines are shown in these graphs only to help the reader's eye follow the waveforms.

РИСУНОК 8-5

Базисные функции ДПФ. 32 точки ДПФ имеют 17 дискретных волн косинуса и 17 дискретных волн синуса для его базисных функций. Восемь из них показываються в этом рисунке. Дискретные сигналы; показаны на

этих графиках непрерывными линиями только для того, чтобы помочь глазу читателя следовать за формами волны.

Let's look at several of these basis functions in detail. Figure (a) shows the cosine wave $c_0[n]$ of zero frequency, which is a constant value of one. This means that $Re X[0]$ holds the average value of all the points in the time domain signal. In electronics, it would be said that $Re X[0]$ holds the **DC offset**. The sine wave of zero frequency, $s_0[n]$ is shown in (b), a signal composed of all *zeros*. Since this can not affect the time domain signal being synthesized, the value of $Im X[0]$ is *irrelevant*, and always set to zero. More about this shortly.

Давайте рассмотрим несколько из этих базисных функций подробно. Рисунок (а) показывает косинус волны, это является в косинусом волны $c_0[n]$ нулевой частоты, которая является константой значением единица. Это означает, что $Re X[0]$ содержит среднее значений всех точек в сигнале домена времени. В электронике, сказало бы что $Re X[0]$ содержит **смещение постоянного тока (смещение DC)**. Синус волна нулевой частоты, $s_0[n]$ показанный в (b), сигнал, составленный из всех нулей. Так как это не может затрагивать синтезируемый сигнал домена времени, значение $Im X[0]$ несоответствующее, и всегда устанавливается на нуль. Больше относительно этого вскоре.

Figures (c) & (d) show $c_2[n]$ & $s_2[n]$, the sinusoids that complete *two* cycles in the N points. These correspond to $Re X[2]$ & $Im X[2]$, respectively. Likewise, (e) & (f) show $c_{10}[n]$ & $s_{10}[n]$ the sinusoids that complete *ten* cycles in the N , points. These sinusoids correspond to the amplitudes held in the arrays $Re X[10]$ & $Im X[10]$. The problem is, the samples in (e) and (f) no longer *look* like sine and cosine waves. If the continuous curves were not present in these graphs, you would have a difficult time even detecting the pattern of the waveforms. This may make you a little uneasy, but don't worry about it. From a mathematical point of view, these samples do form discrete sinusoids, even if your eye cannot follow the pattern.

Рисунки (с) и (d) показывают $c_2[n]$ и $s_2[n]$, эти синусоиды *два* полных цикла(периода) в N точках. Они соответствуют(записываются) $Re X[2]$ и $Im X[2]$, соответственно. Аналогично, (e) и (f) показывают $c_{10}[n]$ и $s_{10}[n]$ синусоиды что полные десять циклов(периодов) в N , точек. Эти синусоиды соответствуют амплитудам, проведенным(поддержанным) в массивах $Re X[10]$ и $Im X[10]$. Проблема, выборки в (e) и (f) больше не *напоминают* косинус и синус волны. Если бы непрерывные кривые не были представлены в этих графиках, Вам было бы трудно точно обнаружить(увидеть) образец формы волны. Это может делать Вам немного непросто, но не волнуйтесь относительно этого. С математической точки зрения, эти выборки формируют дискретные синусоиды, даже если ваш глаз не может следовать за образцом.

The highest frequencies in the basis functions are shown in (g) and (h). These are $c_{N/2}[n]$ & $s_{N/2}[n]$ or in this example, $c_{16}[n]$ & $s_{16}[n]$. The discrete cosine wave alternates in value between 1 and -1, which can be interpreted as sampling a continuous sinusoid at the *peaks*. In contrast, the discrete sine wave contains all zeros, resulting from sampling at the *zero crossings*. This makes the value of $Im X[N/2]$ the same as $Im X[0]$, always equal to zero, and not affecting the synthesis of the time domain signal.

Самые высокие частоты в базисных функциях показываются в (g) и (h). Они - $c_{N/2}[n]$ и $s_{N/2}[n]$ или в этом примере, $c_{16}[n]$ & $s_{16}[n]$. Дискретные волны косинуса изменяют значения между 1 и -1, которое может интерпретироваться как осуществление выборки непрерывной синусоиды в *пиках*. Напротив, дискретная волна синуса содержит все нули, следуя, производя выборки при нулевых пересечениях. Это делает значение $Im X[N/2]$ тем

же самым как $Im X[0]$, всегда равным нулю, и не воздействующим(не имеющим влияния) на синтез сигнала домена времени.

Here's a puzzle: If there are N samples entering the DFT, and $N+2$ samples exiting, where did the extra information come from? The answer: two of the output samples contain *no* information, allowing the other N samples to be fully independent. As you might have guessed, the points that carry no information are $Im X[0]$ and $Im X[N/2]$, the samples that always have a value of zero.

Имеется проблема: Если имеются N выборок ввода ДПФ, и $N+2$ выборок выхода, где(откуда) дополнительная информация исходила? Ответ: две из выборок выхода *не* содержат никакой информации, позволяя другим N выборкам быть полностью независимыми. Поскольку Вы могли бы предположить, точки, которые не несут никакой информации - $Im X[0]$ и $Im X[N/2]$, выборки, которые всегда имеют значение нуля.

Synthesis, Calculating the Inverse DFT **Синтез, Вычисление Обратного ДПФ**

Pulling together everything said so far, we can write the **synthesis equation**:

Перемещение, вместе все сказало пока, мы можем записать уравнение синтеза:

(Учитывая сказанное, мы можем записать уравнение синтеза):

$$x[i] = \sum_{k=0}^{N/2} Re \bar{X}[k] \cos(2\pi ki/N) + \sum_{k=0}^{N/2} Im \bar{X}[k] \sin(2\pi ki/N)$$

EQUATION 8-2

The synthesis equation. In this relation, $x[i]$ is the signal being synthesized, with the index, i , running from 0 to $N-1$. $Im \bar{X}[k]$ and $Re \bar{X}[k]$ hold the amplitudes of the cosine and sine waves, respectively, with k running from 0 to $N/2$. Equation 8-3 provides the normalization to change this equation into the inverse DFT.

УРАВНЕНИЕ 8-2

Уравнение синтеза. В этом отношении, $x[i]$ - синтезируемый сигнал, с индексом i , выполняющийся от 0 до $N-1$. $Im \bar{X}[k]$ and $Re \bar{X}[k]$ проводят(держат) амплитуды волн косинуса и синуса, соответственно, с k , выполняющимся от 0 до $N/2$. Уравнение 8-3 обеспечивает нормализацию чтобы изменить(заменить) это уравнение в обратное ДПФ.

In words, any N point signal, $x[i]$, can be created by adding $N/2+1$ cosine waves and $N/2+1$ sine waves. The amplitudes of the cosine and sine waves are held in the arrays $Im \bar{X}[k]$ and $Re \bar{X}[k]$ respectively. The synthesis equation multiplies these amplitudes by the basis functions to create a set of scaled sine and cosine waves. Adding the scaled sine and cosine waves produces the time domain signal, $x[i]$.

В словах, любая точка N сигнала $x[i]$, может быть создана, прибавляя $N/2+1$ волны косинуса и $N/2+1$ волны синуса. Амплитуды волн косинуса и синуса проведены(поддержаны) в массивах $Im \bar{X}[k]$ and $Re \bar{X}[k]$ соответственно. Уравнение синтеза умножает эти амплитуды на базисные функции, чтобы создать набор масштабируемых волн синуса и косинуса. Добавление масштабируемых волн синуса и косинуса производит сигнал домена времени, $x[i]$.

In Eq. 8-2, the arrays are called $Im \bar{X}[k]$ and $Re \bar{X}[k]$, rather than $Im X[k]$ and $Re X[k]$. This is because the *amplitudes needed for synthesis* (called in this $Re X[k]$ discussion: $Im \bar{X}[k]$ and $Re \bar{X}[k]$), are slightly different from the *frequency domain of a signal* (denoted by: $Im X[k]$ and $Re X[k]$). This is the scaling factor issue we referred to earlier. Although the conversion is only a simple

normalization, it is a common bug in computer programs. Look out for it! In equation form, the conversion between the two is given by:

В уравнении 8-2, массивы называются $Im \bar{X}[k]$ and $Re \bar{X}[k]$, скорее чем $Im X[k]$ и $Re X[k]$. Это - то, потому что амплитуды, необходимые для синтеза (вызвавшие это обсуждение: $Im \bar{X}[k]$ and $Re \bar{X}[k]$), слегка отличаются от сигнала частотного домена (обозначенного: $Im X[k]$ и $Re X[k]$). Это - проблема коэффициента(фактора) масштабирования, которую мы упомянули ранее. Хотя преобразование - только простая нормализация, это - общая(обычная) ошибка в компьютерных программах. Имейте это ввиду! Преобразование между этими двумя(между ними) дается в форме уравнения:

EQUATIONS 8-3

Conversion between the sinusoidal amplitudes and the frequency domain values. In these equations, $Re \bar{X}[k]$ and $Im \bar{X}[k]$ hold the amplitudes of the cosine and sine waves needed for synthesis, while $Re X[k]$ and $Im X[k]$ hold the real and imaginary parts of the frequency domain. As usual, N is the number of points in the time domain signal, and k is an index that runs from 0 to $N/2$.

$$Re \bar{X}[k] = \frac{Re X[k]}{N/2}$$

$$Im \bar{X}[k] = - \frac{Im X[k]}{N/2}$$

УРАВНЕНИЯ 8-3

Преобразование между синусоидальными амплитудами и значениями частотного домена. В этих уравнениях, $Re \bar{X}[k]$ и $Im \bar{X}[k]$ проводят(держат) амплитуды волн косинуса и синуса, необходимых для синтеза, в то время как $Re X[k]$ и $Im X[k]$ проводят(держат) реальные(вещественные) и мнимые части частотного домена. Как обычно, N - число точек в сигнале домена времени, и k - индекс, который выполняется от 0 до $N/2$.

except for two special cases:

$$Re \bar{X}[0] = \frac{Re X[0]}{N}$$

$$Re \bar{X}[N/2] = \frac{Re X[N/2]}{N}$$

Suppose you are given a frequency domain representation, and asked to synthesize the corresponding time domain signal. To start, you must find the amplitudes of the sine and cosine waves. In other words, given $Im X[k]$ and $Re X[k]$, you must find $Im \bar{X}[k]$ and $Re \bar{X}[k]$. Equation 8-3 shows this in a mathematical form. To do this in a computer program, three actions must be taken. First, divide all the values in the frequency domain by $N/2$. Second, change the sign of all the imaginary values. Third, divide the first and last samples in the real part, $Re X[0]$ and $Re X[N/2]$, by two. This provides the amplitudes needed for the synthesis described by Eq. 8-2. Taken together, Eqs. 8-2 and 8-3 *define* the inverse DFT.

Предположим, Вам дают представление частотного домена, и просят синтезировать соответствующий сигнал домена времени. Чтобы начать, Вы должны найти амплитуды волн косинуса и синуса. Другими словами, дается $Im X[k]$ и $Re X[k]$, Вы должны найти $Im \bar{X}[k]$ and $Re \bar{X}[k]$. Уравнение 8-3 представляет это в математической форме. Чтобы делать это в компьютерной программе, должны быть приняты три действия. Во первых, делите все значения в частотном домене $N/2$. Во вторых, измените знак всех мнимых значений. Третье, делите первые и последние выборки в вещественной части, $Re X[0]$ и $Re X[N/2]$, на два. Это обеспечивает амплитуды необходимые для синтеза, описанного уравнением 8-2. Взятые вместе, уравнения 8-2 и 8-3 определяют обратное ДПФ.

The entire Inverse DFT is shown in the computer program listed in Table 8-1. There are two ways that the synthesis (Eq. 8-2) can be programmed, and both are shown. In the first method, each of the scaled sinusoids are generated one at a time and added to an accumulation array, which ends up becoming the time domain signal. In the second method, each sample in the time domain signal is calculated one at a time, as the sum of all the corresponding samples in the

cosine and sine waves. Both methods produce the same result. The difference between these two programs is very minor; the inner and outer loops are swapped during the synthesis.

Полное Обратный ДПФ показано в компьютерной программе, показанной в таблице 8-1. Имеются два пути(способа), которыми синтез (уравнение 8-2) может быть запрограммирован, и оба показаны. В первом методе, каждая из масштабируемых синусоид сгенерирована по одной и добавлена к массиву накоплений, который, заканчиваясь, становится сигналом домена времени. Во втором методе, каждая выборка в сигнале домена времени рассчитана одновременно, как сумма всех соответствующих выборок в волнах косинуса и синуса.

НАУЧНО-ТЕХНИЧЕСКОЕ РУКОВОДСТВО ПО ЦИФРОВОЙ ОБРАБОТКЕ СИГНАЛОВ

```
100 " ОБРАТНОЕ ДИСКРЕТНОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ
110 'The time domain signal, held in XX[ ], is calculated from the frequency domain signals,
120 'held in REX[ ] and IMX[ ].
130 '
140 DIM XX[511] 'XX[ ] holds the time domain signal
150 DIM REX[256] 'REX[ ] holds the real part of the frequency domain
160 DIM IMX[256] 'IMX[ ] holds the imaginary part of the frequency domain
170 '
180 PI = 3.14159265 'Set the constant, PI
190 N% = 512 'N% is the number of points in XX[ ]
200 '
210 GOSUB XXXX 'Mythical subroutine to load data into REX[ ] and IMX[ ]
220 '
230
240 ' 'Find the cosine and sine wave amplitudes using Eq. 8-3
250 FOR K% = 0 TO 256
260 REX[K%] = REX[K%] / (N%/2)
270 IMX[K%] = -IMX[K%] / (N%/2)
280 NEXT K%
290 '
300 REX[0] = REX[0] / 2
310 REX[256] = REX[256] / 2
320 '
330 '
340 FOR I% = 0 TO 511 'Zero XX[ ] so it can be used as an accumulator
350 XX[I%] = 0
360 NEXT I%
370 '
380 ' Eq. 8-2 SYNTHESIS METHOD #1. Loop through each
390 ' frequency generating the entire length of the sine and cosine
400 ' waves, and add them to the accumulator signal, XX[ ]
410 '
420 FOR K% = 0 TO 256 'K% loops through each sample in REX[ ] and IMX[ ]
430 FOR I% = 0 TO 511 'I% loops through each sample in XX[ ]
440 '
450 XX[I%] = XX[I%] + REX[K%] * COS(2*PI*K%*I%/N%)
460 XX[I%] = XX[I%] + IMX[K%] * SIN(2*PI*K%*I%/N%)
470 '
480 NEXT I%
490 NEXT K%
500 '
510 END
```

Альтернативный код для строк от 380 до 510

```
380 ' Уравнение 8-2 МЕТОД СИНТЕЗА #2. Loop through each
390 ' sample in the time domain, and sum the corresponding
400 ' samples from each cosine and sine wave
410 '
420 FOR I% = 0 TO 511 'I% loops through each sample in XX[ ]
430 FOR K% = 0 TO 256 'K% loops through each sample in REX[ ] and IMX[ ]
440 '
450 XX[I%] = XX[I%] + REX[K%] * COS(2*PI*K%*I%/N%)
460 XX[I%] = XX[I%] + IMX[K%] * SIN(2*PI*K%*I%/N%)
470 '
480 NEXT K%
490 NEXT I%
500 '
510 END
```

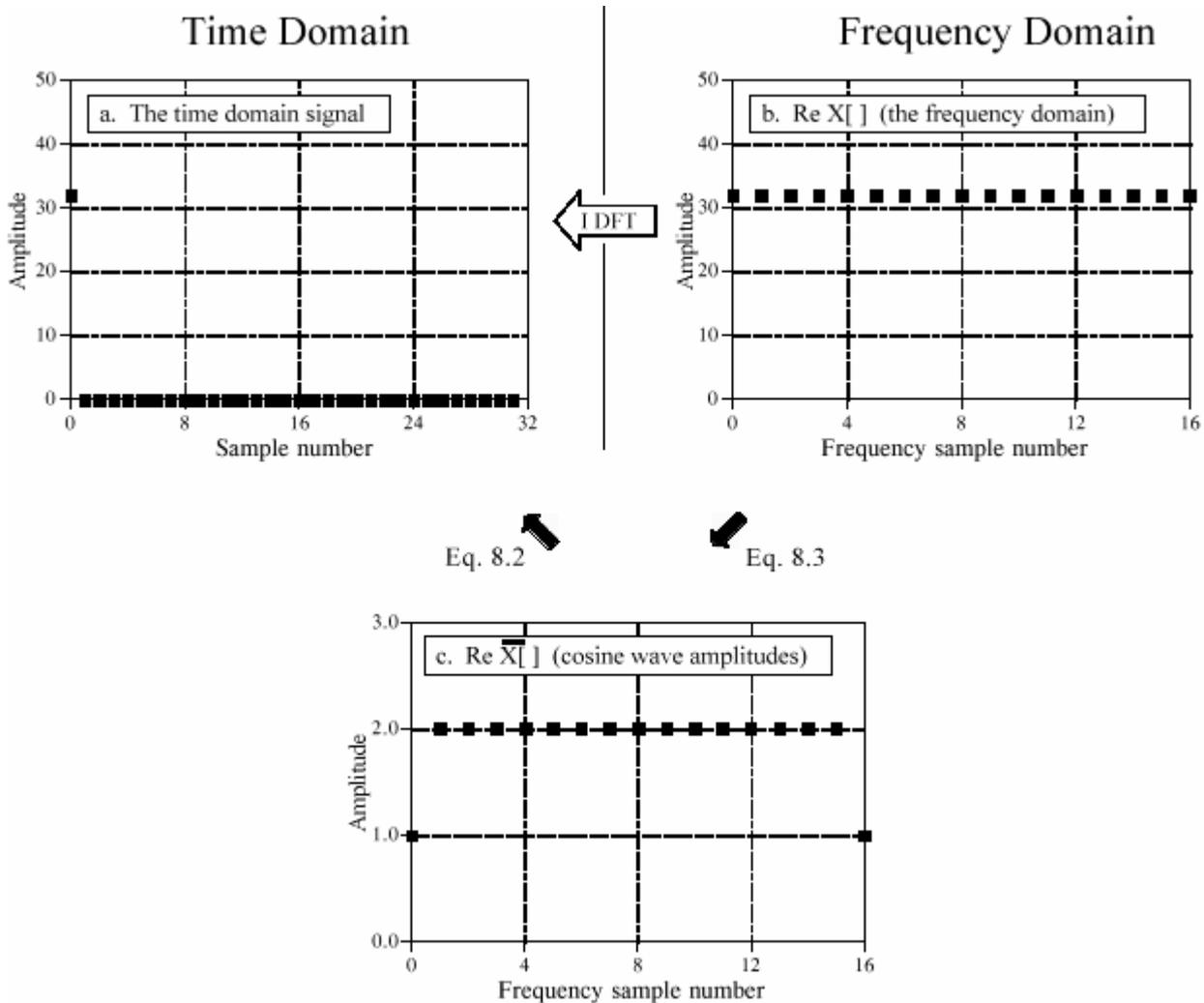


FIGURE 8-6 Example of the Inverse DFT. Figure (a) shows an example time domain signal, an impulse at sample zero with an amplitude of 32. Figure (b) shows the real part of the frequency domain of this signal, a constant value of 32. The imaginary part of the frequency domain (not shown) is composed of all zeros. Figure(c) shows the amplitudes of the cosine waves needed to reconstruct (a) using Eq. 8-2. The values in (c) are found from (b) by using Eq. 8-3.

РИСУНОК 8-6

Пример Обратного(инвертированного) ДПФ. Рисунок (а) показывает пример сигнала домена времени, импульс в выборке нуль с амплитудой 32. Рисунок (б) показывает вещественную часть частотного домена этого сигнала, постоянное значение 32. Мнимая часть частотного домена (не показана) составлена из всех нулей. Рисунок (с) показывает амплитуды волн косинуса, необходимых, чтобы восстановить (а), используя уравнение 8-2. Значения в (с) найдены от (б), используя уравнение 8-3.

Figure 8-6 illustrates the operation of the Inverse DFT, and the slight differences between the frequency domain and the amplitudes needed for synthesis. Figure 8-6a is an example signal we wish to synthesize, an impulse at sample zero with an amplitude of 32. Figure 8-6b shows the frequency domain representation of this signal. The real part of the frequency domain is a constant value of 32. The imaginary part (not shown) is composed of all zeros. As discussed in the next chapter, this is an important DFT pair: an impulse in the time domain corresponds to a

constant value in the frequency domain. For now, the important point is that (b) is the *DFT* of (a), and (a) is the *Inverse DFT* of (b).

Рисунок 8-6 иллюстрирует операцию Обратного(инвертированного) ДПФ, и небольшие различия между частотным доменом и амплитудами, необходимыми для синтеза. Рисунок 8-6а - пример сигнала, который мы желаем синтезировать, импульс в выборке нуль с амплитудой 32. Рисунок 8-6б показывает частотное представление домена этого сигнала. Вещественная часть частотного домена - постоянное значение 32. Мнимая часть (не показана) составлена из всех нулей. Как обсуждено в следующей главе, это - важная пара ДПФ: импульс в домене времени соответствует постоянному значению в частотном домене. Пока, важный пункт - тот (b) - ДПФ (a), и (a) – Обратный(инвертированный) ДПФ (b).

Equation 8-3 is used to convert the frequency domain signal, (b), into the amplitudes of the cosine waves, (c). As shown, all of the cosine waves have an amplitude of *two*, except for samples 0 and 16, which have a value of *one*. The amplitudes of the sine waves are not shown in this example because they have a value of zero, and therefore provide no contribution. The synthesis equation, Eq. 8-2, is then used to convert the amplitudes of the cosine waves, (b), into the time domain signal, (a).

Уравнение 8-3 используется, чтобы преобразовать сигнал домена частоты, (b), в амплитуды волн косинуса, (c). Как показано, все волны косинуса имеют амплитуду *два*, исключая выборки 0 и 16, которые имеют значение *единица*. Амплитуды волн синуса не показываются в этом примере, потому что они имеют значение нуля, и поэтому не вносят никакого вклада. Уравнение синтеза, уравнение 8-2, тогда используется, чтобы преобразовать амплитуды волн косинуса, (b), в сигнал домена времени, (a).

This describes *how* the frequency domain is different from the sinusoidal amplitudes, but it doesn't explain *why* it is different. The difference occurs because the frequency domain is defined as a **spectral density**. Figure 8-7 shows how this works. The example in this figure is the real part of the frequency domain of a 32 point signal. As you should expect, the samples run from 0 to 16, representing 17 frequencies equally spaced between 0 and $\frac{1}{2}$ of the sampling rate. *Spectral density* describes how much signal (amplitude) is present *per unit of bandwidth*. To convert the sinusoidal amplitudes into a spectral density, divide each amplitude by the bandwidth represented by each amplitude. This brings up the next issue: how do we determine the bandwidth of each of the discrete frequencies in the frequency domain?

Это описывает, *как* частотный домен отличается от синусоидальных амплитуд, но это не объясняет, *почему* это различно. Разность происходит, потому что частотный домен определен как **спектральная плотность**. На рисунке 8-7 показано, как это работает. Пример в этом рисунке - вещественная часть частотного домена из 32 точек сигнала. Поскольку Вы должны ожидать, выборки, выполненные от 0 до 16, представляют 17 частот, одинаково раздельных между 0 и $\frac{1}{2}$ из частоты выборки(частоты дискретизации). *Спектральная плотность* описывает, сколько сигнала (амплитуд) присутствует *на модуль ширины полосы частот*. Чтобы преобразовывать синусоидальные амплитуды в спектральную плотность, делите каждую амплитуду на ширину полосы частот, представленную каждой амплитудой. Это поднимает следующую проблему: как мы определяем ширину полосы частот каждой из дискретных частот в частотном домене?

As shown in the figure, the bandwidth can be defined by drawing dividing lines between the samples. For instance, sample number 5 occurs in the band between 4.5 and 5.5; sample number (c) АВТЭКС, Санкт-Петербург, <http://www.autex.spb.ru>, e-mail: info@autex.spb.ru

6 occurs in the band between 5.5 and 6.5, etc. Expressed as a fraction of the total bandwidth (i.e., the $N/2$), bandwidth of each sample is $2/N$. An exception to this is the samples on each end, which have one-half of this bandwidth, $1/N$. This accounts for the $2/N$ scaling factor between the sinusoidal amplitudes and frequency domain, as well as the additional factor of two needed for the first and last samples.

Как показано на рисунке, ширина полосы частот может быть определена, вычерчивая разделительные линии между выборками. Например, пятая выборка происходит в полосе между 4.5 и 5.5; шестая выборка происходит в полосе между 5.5 и 6.5, и т.д. Выраженным как дробь(доля, фракция) полной ширины полосы частот (то есть, $N/2$), ширина полосы частот каждой выборки - $2/N$. Выраженным как дробь(доля) полной ширины полосы частот (то есть, $N/2$), ширина полосы частот каждой выборки - $2/N$. Исключение к этому - выборки на каждом конце, которые имеют половину этой ширины полосы частот, $1/N$. Это объясняет масштабирующий коэффициент(фактор) $2/N$ между синусоидальными амплитудами и частотным доменом, также как дополнительным коэффициентом(фактором) два, необходимым для первой и последней выборок.

Why the negation of the imaginary part? This is done solely to make the *real DFT* consistent with its big brother, the *complex DFT*. In Chapter 29 we will show that it is necessary to make the mathematics of the complex DFT work. When dealing only with the real DFT, many authors do not include this negation. For that matter, many authors do not even include the $2/N$ scaling factor. Be prepared to find both of these missing in some discussions. They are included here for a tremendously important reason: The most efficient way to calculate the DFT is through the Fast Fourier Transform (FFT) algorithm, presented in Chapter 12. The FFT generates a frequency domain defined according to Eq. 8-2 and 8-3. If you start messing with these normalization factors, your programs containing the FFT are not going to work as expected.

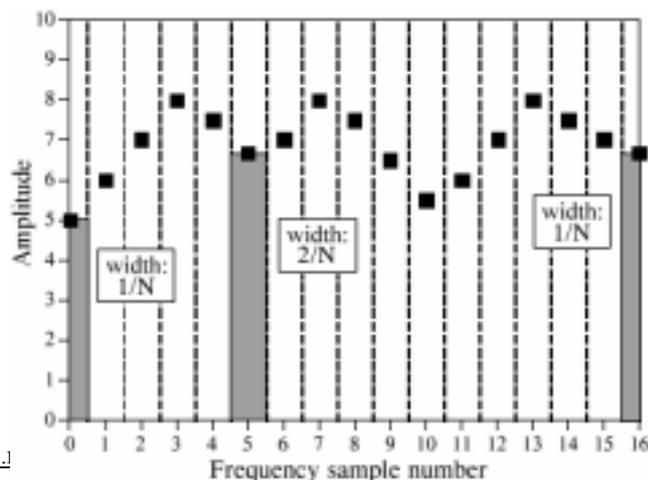
Почему отрицание мнимой части? Это сделано исключительно, чтобы делать *реальный ДПФ* совместимым с его большим(старшим) братом, *комплексным ДПФ*. В главе 29 мы покажем, что необходимо делать математику из комплексной работы ДПФ. При контакте только с реальным ДПФ, много авторов не включают это отрицание. В этом отношении, много авторов делаю, даже не включают $2/N$ масштабирующий коэффициент(фактор). Будьте готовы, чтобы найти оба из этих отсутствующих в некоторых обсуждениях. Они включены здесь по чрезвычайно важной причине: наиболее эффективный способ вычислять ДПФ - через Быстрое преобразование Фурье алгоритм (БПФ), представленный в главе 12. БПФ генерирует частотный домен, определенный согласно уравнению 8-2 и 8-3. Если Вы смешиваете сначала с этими коэффициентами(факторами) нормализации, ваши программы, содержащие БПФ не собираются работать как ожидается.

FIGURE 8-7

The bandwidth of frequency domain samples. Each sample in the frequency domain can be thought of as being contained in a frequency band of width $2/N$, expressed as a fraction of the total bandwidth. An exception to this is the first and last samples, which have a bandwidth only one-half this wide, $1/N$.

РИСУНОК 8-7

Ширина полосы частот в выборках частотного домена. О каждой выборке в частотном домене можно думать как содержащейся в полосе частот шириной $2/N$, выраженной как дробь(доля) полной ширины полосы частот. Исключение к



этому - первая и последняя выборки, которые имеют полосу частот, только половину этой ширины, $1/N$.

Analysis, Calculating the DFT

The DFT can be calculated in three completely different ways. First, the problem can be approached as a set of *simultaneous equations*. This method is useful for understanding the DFT, but it is too inefficient to be of practical use. The second method brings in an idea from the last chapter: *correlation*. This is based on detecting a known waveform in another signal. The third method, called the Fast Fourier Transform (FFT), is an ingenious algorithm that decomposes a DFT with N points, into N DFTs each with a single point. The FFT is typically hundreds of times faster than the other methods. The first two methods are discussed here, while the FFT is the topic of Chapter 12. It is important to remember that all three of these methods produce an identical output. Which should you use? In actual practice, *correlation* is the preferred technique if the DFT has less than about 32 points, otherwise the *FFT* is used.

Анализ, Вычисление ДПФ

ДПФ может быть рассчитан тремя полностью различными способами. Во первых, к проблеме можно приближаться как набор *системы уравнений*. Этот метод полезен для понимания ДПФ, но слишком неэффективен для практического использования. Второй метод вводит идею от прошлой главы: *корреляция*. Это основано на обнаружении известной формы волны в другом сигнале. Третий метод, называемый Быстрым преобразованием Фурье (БПФ), является изобретательным алгоритмом, который анализирует(расчленяет) ДПФ с пунктами(точками) N , в N ДПФ каждый с единственной(отдельной) точкой. БПФ - типично в сотни раз быстрее чем другие методы. Первые два метода обсуждены здесь, в то время как БПФ - тема(раздел) главы 12. Важно помнить, что все три из этих методов производят идентичную продукцию(выход). Который Вы должны использовать? В фактической практике, корреляция - привилегированная методика, если ДПФ имеет меньше чем приблизительно 32 точки, в иначе(в противном случае) используется БПФ.

DFT by Simultaneous Equations

Think about the DFT calculation in the following way. You are given N values from the time domain, and asked to calculate the N values of the frequency domain (ignoring the two frequency domain values that you know must be zero). Basic algebra provides the answer: to solve for N unknowns, you must be able to write N linearly independent equations. To do this, take the first sample from each sinusoid and add them together. The sum must be equal to the first sample in the time domain signal, thus providing the first equation. Likewise, an equation can be written for each of the remaining points in the time domain signal, resulting in the required N equations. The solution can then be found by using established methods for solving simultaneous equations, such as Gauss Elimination. Unfortunately, this method requires a tremendous number of calculations, and is virtually never used in DSP. However, it is important for another reason, it shows *why* it is possible to decompose a signal into sinusoids, how *many* sinusoids are needed, and that the basis functions must be linearly independent (more about this shortly).

ДПФ Системой уравнений

Думайте относительно вычисления ДПФ следующим способом. Вам дают значения N от домена времени, и просят вычислить значения N частотного домена (игнорируя два частотных значения домена, которые Вы знаете, должны быть нулевыми). Основная

алгебра обеспечивает ответ: чтобы решить для N неизвестных, Вы должны быть способны записать N линейно независимые уравнения. Чтобы делать это, берите первую выборку от каждой синусоиды, и сложите их вместе. Сумма должна быть равна первой выборке в сигнале домена времени, таким образом, обеспечивая первое уравнение. Аналогично, уравнение может быть написано в каждой из остающихся точек в сигнале домена времени, приводя к требуемым уравнением N . Решение может тогда быть найдено, используя установленные методы для решения системы уравнений, типа Устранения Гаусса (элиминации, устранения, исключения Гаусса). К сожалению, этот метод требует огромного числа вычислений, и фактически никогда не используется в ЦОС. Однако, это важно по другой причине, это показывает, *почему* возможно анализировать(расчленив) сигнал в синусоиды, *как много(сколько)* синусоид необходимо, и что базисные функция должны быть линейно независимы (больше относительно этого вскоре).

DFT by Correlation

Let's move on to a better way, the *standard way* of calculating the DFT. An example will show how this method works. Suppose we are trying to calculate the DFT of a 64 point signal. This means we need to calculate the 33 points in the real part, and the 33 points in the imaginary part of the frequency domain. In this example we will only show how to calculate a single sample, $Im X[3]$, i.e., the amplitude of the sine wave that makes three complete cycles between point 0 and point 63. All of the other frequency domain values are calculated in a similar manner.

ДПФ Корреляцией

Давайте перейдем к лучшему пути, *стандартному пути* вычисления ДПФ. Пример покажет, как этот метод работает. Предположим, что мы пробуем вычислять ДПФ из 64 точек сигнала. Это означает, что мы должны вычислить 33 точки в вещественной части, и 33 точки в мнимой(несобственной) части частотного домена. В этом примере мы только покажем, как вычислить единственную(отдельную) выборку, $Im X[3]$, то есть, амплитуда волны синуса, которая делает три полных цикла(периода) между точкой 0 и точкой 63. Все другие частотные значения домена рассчитаны подобным способом.

Figure 8-8 illustrates using correlation to calculate $Im X[3]$. Figures (a) and (b) show two example time domain signals, called: respectively. The first signal $x1[]$, is composed of nothing but a sine wave that makes three cycles between points 0 and 63. In contrast, $x2[]$ is composed of several sine and cosine waves, *none* of which make three cycles between points 0 and 63. These two signals illustrate what the algorithm for calculating $Im X[3]$ must do. When fed $x1[]$, the algorithm must produce a value of 32, the amplitude of the sine wave present in the signal (modified by the scaling factors of Eq. 8-3). In comparison, when the algorithm is fed the other signal $x2[]$, a value of zero must be produced, indicating that this particular sine wave is not present in this signal.

Рисунок 8-8 иллюстрирует использование корреляции, чтобы вычислить $Im X[3]$. Рисунки (a) и (b) показывают два примера сигнала домена времени, называемые: $x1[]$ и $x2[]$, соответственно. Первый сигнал $x1[]$, составлен только из волна синуса, которая делает три цикла(периода) между точками 0 и 63. Напротив, $x2[]$ является составленный из нескольких волн синуса и косинуса, *ни одна* из которых не делает три цикла(периода) между точками 0 и 63. Эти два сигнала иллюстрируют то, что должен делать алгоритм для вычисления $Im X[3]$. Когда подано $x1[]$, алгоритм должен произвести значение 32, амплитуда волны синуса представлена в сигнале (изменяемая коэффициентами масштабирования уравнения 8-3). Для сравнения, когда алгоритм подан другого сигнала, $x2[]$, значение должно быть произведено нуля, указывая, что эта специфическая волна синуса - не представлена в этом сигнале.

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

The concept of correlation was introduced in Chapter 7. As you recall, to detect a known waveform contained in another signal, multiply the two signals and add all the points in the resulting signal. The single number that results from this procedure is a measure of how similar the two signals are. Figure 8-8 illustrates this approach. Figures (c) and (d) both display the signal we are looking for, a sine wave that makes 3 cycles between samples 0 and 63. Figure (e) shows the result of multiplying (a) and (c). Likewise, (f) shows the result of multiplying (b) and (d). The sum of all the points in (e) is 32, while the sum of all the points in (f) is zero, showing we have found the desired algorithm.

Концепция корреляции была представлена в главе 7. Как Вы повторно вызываете, обнаруживать известную форму волны, содержащуюся в другом сигнале, умножаете(мультиплицируете) два сигнала и прибавляете все точки в заканчивающемся сигнале. Единственное число (номер, критерий), который следует из этой процедуры - мера того, насколько подобны два сигнала. Рисунок 8-8 иллюстрирует этот подход. Рисунки (c) и (d) оба отображают сигнал, мы смотрим для, волны синуса, которая делает 3 цикла между выборками 0 и 63. Рисунок (e) показывает результат умножения (a) и (c). Аналогично, (f) показывает результат умножения (b) и (d). Сумма всех точек в (e) - 32, в то время как сумма всех точек в (f) нулевая, показывая мы нашли желательный алгоритм.

The other samples in the frequency domain are calculated in the same way. This procedure is formalized in the *analysis equation*, the mathematical way to calculate the frequency domain from the time domain:

Другие выборки в частотном домене рассчитаны таким же образом. Эта процедура формализована в *уравнении анализа*, математический способ вычислить частотный домен от домена времени:

EQUATION 8-4

The analysis equations for calculating the DFT. In these equations $x[i]$, is the time domain signal being analyzed, and are the frequency $Re X[k]$ & $Im X[k]$ domain signals being calculated. The index i runs from 0 to $N-1$, while the index k runs from 0 to $N/2$.

$$Re X[k] = \sum_{i=0}^{N-1} x[i] \cos(2\pi k i / N)$$

$$Im X[k] = - \sum_{i=0}^{N-1} x[i] \sin(2\pi k i / N)$$

Уравнения анализа для вычисления ДПФ. В этих уравнениях $x[i]$, является анализируемым сигналом домена времени, и $Re X[k]$ и $Im X[k]$ - рассчитываемые сигналы частотного домена. Индекс i , выполняется от 0 до $N-1$, в то время как индекс k выполняется от 0 до $N/2$.

In words, each sample in the frequency domain is found by multiplying the time domain signal by the sine or cosine wave being looked for, and adding the resulting points. If someone asks you what you are doing, say with confidence: "I am correlating the input signal with each basis function." Table 8-2 shows a computer program for calculating the DFT in this way.

В словах, каждая выборка в частотном домене найдена, умножая(мультиплицируя) сигнал домена времени волной синуса или косинуса, разыскивая, и прибавляя заканчивающиеся точки. Если кто - то спрашивает Вас, что Вы делаете, говорите с самоуверенно: " я коррелирую входной сигнал с каждой базисной функцией. В таблице 8-2 показана компьютерная программа для вычисления ДПФ таким образом.

The analysis equation does *not* require special handling of the first and last points, as did the synthesis equation. There is, however, a negative sign in the imaginary part in Eq. 8-4. Just as

before, this negative sign makes the *real DFT* consistent with the *complex DFT*, and is not always included.

Уравнение анализа *не* требует специальной обработки первых и последних точек, также, как и уравнение синтеза. Имеется, однако, отрицательный сигнал в мнимой(несобственной) части уравнения 8-4. Также, как прежде, этот отрицательный знак делает *реальный(вещественный) ДПФ* совместимым с *комплексным ДПФ*, и не всегда включается.

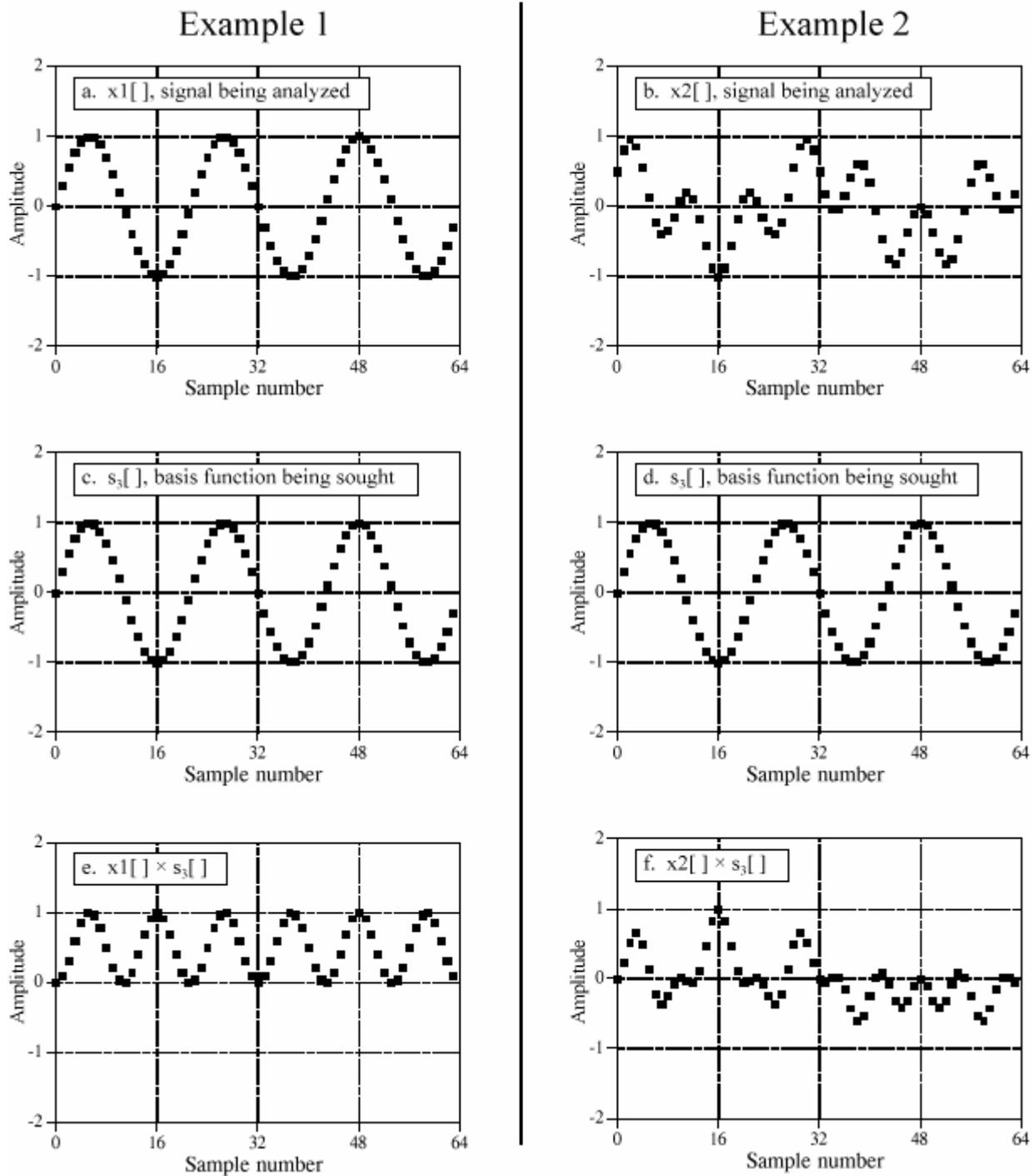


FIGURE 8-8
 Two example signals, (a) and (b), are analyzed for containing the specific basis function shown in (c) and (d). Figures (e) and (f) show the result of multiplying each example signal by the basis function. Figure (e) has an
 (c) АВТЭКС, Санкт-Петербург, <http://www.autex.spb.ru>, e-mail: info@autex.spb.ru

average of 0.5, indicating that $x_1[n]$ contains the basis function with an amplitude of 1.0. Conversely, (f) has a zero average, indicating that $x_2[n]$ does not contain the basis function.

РИСУНОК 8-8

Два примера сигнала, (a) и (b), проанализированы для содержания специфической базисной функции, показанной в (c) и (d). Рисунки (e) и (f) показывают результат умножения каждой выборкой сигнала базисной функции. Рисунок (e) имеет среднее число 0.5, указывая, что $x_1[n]$ содержит базисную функцию с амплитудой 1.0. Наоборот, (f) имеет нулевое среднее число, указывая, что $x_2[n]$ не содержит базисную функцию.

In order for this correlation algorithm to work, the basis functions must have an interesting property: each of them must be completely *uncorrelated* with all of the others. This means that if you multiply any two of the basis functions, the sum of the resulting points will be equal to zero. Basis functions that have this property are called **orthogonal**. Many other

Для этого алгоритма корреляции, чтобы работать, базисные функции должны иметь интересное свойство: каждая из них должна быть полностью *некоррелированной* со всеми другими. Это означает, что, если Вы умножаете(мультиплицируете) любые две из базисных функций, сумма заканчивающихся точек будет равна нулю. Базисные функции, которые имеют это свойство, называются **ортогональными**. Много других ортогональных базисных функций существуют, включая: квадратные волны, треугольные волны, импульсы, и т.д. Сигналы могут быть расчленены в них другие ортогональные базисные функции, используя корреляцию, также, как сделано здесь с синусоидами. Это не должно предполагать, что это является *полезным*, только то, что это является *возможным*.

As previously shown in Table 8-1, the *Inverse DFT* has two ways to be implemented in a computer program. This difference involves *swapping* the inner and outer loops during the synthesis. While this does not change the output of the program, it makes a difference in how you *view* what is being done. The *DFT* program in Table 8-2 can also be changed in this fashion, by swapping the inner and outer loops in lines 310 to 380. Just as before, the output of the program is the same, but the way you *think* about the calculation is different. (These two different ways of viewing the DFT and inverse DFT could be described as "input side" and "output side" algorithms, just as for convolution).

Как предварительно показано в Таблице 8-1, *Обратный ДПФ* имеет два способа, которые будут осуществлены в компьютерной программе. Эта разность включает в себя(подразумевает) *подкачку* внутренних и внешних циклов в течение синтеза. В то время как это не изменяет продукцию(выход) программы, это делает разность в том, как Вы рассматриваете то, что сделается. Программа ДПФ в таблице 8-2 может также быть изменена этим способом, подкачкой внутренних и внешних электрических циклов в строках от 310 до 380. Также, как прежде, выход программы - тот же самый, но путь, которым Вы *думаете* относительно вычисления, различен. (Эти два различных пути просмотра ДПФ и обратного ДПФ могли быть описаны как алгоритмы "стороны ввода" и "стороны вывода", также, как для скручивания(конволюции)).

As the program in Table 8-2 is written, it describes how an individual sample in the frequency domain is affected by all of the samples in the time domain. That is, the program calculates each of the values in the frequency domain in succession, not as a group. When the inner and outer loops are exchanged, the program loops through each sample in the time domain, calculating the

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

contribution of that point to the frequency domain. The overall frequency domain is found by adding the contributions from the individual time domain points. This brings up our next question: what kind of contribution does an individual sample in the time domain provide to the frequency domain? The answer is contained in an interesting aspect of Fourier analysis called *duality*.

Как написана программа в таблице 8-2, это описывает, как на индивидуальную выборку в частотном домене воздействуют всеми выборками в домене времени. То есть программа вычисляет каждое из значений в частотном домене последовательно, не как группу. Когда внутренние и внешние циклы обменены, циклы программы через каждую выборку в домене времени, вычисляя вклад той точки к частотному домену. Полный частотный домен найден, прибавляя вклады от индивидуальных точек домена времени. Это поднимает наш следующий вопрос: какой вклад производит индивидуальная выборка в домене времени, обеспечивая частотный домен? Ответ содержится в интересном аспекте анализа Фурье называемого *дуальностью (принципом двойственности)*.

Duality

Дуальность

The synthesis and analysis equations (Eqs. 8-2 and 8-4) are strikingly similar. To move from one domain to the other, the known values are multiplied by the basis functions, and the resulting products added. The fact that the *DFT* and the *Inverse DFT* use this same mathematical approach is really quite remarkable, considering the totally different way we arrived at the two procedures. In fact, the only significant difference between the two equations is a result of the time domain being *one* signal of N points, while the frequency domain is *two* signals of $N/2+1$ points. As discussed in later chapters, the *complex DFT* expresses both the time and the frequency domains as complex signals of N points each. This makes the two domains completely symmetrical, and the equations for moving between them virtually *identical*.

Синтез и уравнения анализа (уравнения 8-2 и 8-4) поразительно подобны. Чтобы двигаться от одного домена до другой, известные значения умножены на базисные функции, и заканчивающиеся произведения добавленные. Факт, что *ДПФ* и *Обратный ДПФ* использует это тот же самый математический подход, действительно весьма замечательный, рассматривая полностью различный путь, которым мы достигли этих двух процедур. Фактически, единственная существенная разность между этими двумя уравнениями - результат домена времени, являющегося *одним* сигналом точки N , в то время как частотный домен - *два* сигнала точек $N/2+1$. Как обсуждено в более поздних главах, *комплексный ДПФ* выражает оба домена, и время и частоты, поскольку комплексные сигналы N точек каждого. Это делает эти два домена, полностью симметрическими, и уравнения для перемещения между ними фактически *идентичными*.

This symmetry between the time and frequency domains is called **duality**, and gives rise to many interesting properties. For example, a single point in the frequency domain corresponds to a sinusoid in the time domain. By duality, the inverse is also true, a single point in the time domain corresponds to a sinusoid in the frequency domain. As another example, convolution in the time domain corresponds to multiplication in the frequency domain. By duality, the reverse is also true: convolution in the frequency domain corresponds to multiplication in the time domain. These and other duality relationships are discussed in more detail in Chapters 10 and 11.

Эта симметрия между доменом времени и частотным доменом называется **дуальностью**, и вызывает много интересных свойств. Например, единственная(отдельная) точка в (с) АВТЭКС, Санкт-Петербург, <http://www.autex.spb.ru>, e-mail: info@autex.spb.ru

частотном домене передает синусоиду в домене времени. Дуальностью, инверсия - также истина, единственная(отдельная) точка в домене времени соответствует синусоиде в частотном домене. Как другой пример, скручивание(свертка) в домене времени соответствует умножению в частотном домене. Дуальностью, перемена(обратное движение,противоположное) - также истина: скручивание(свертка) в частотном домене передает умножению в время домен. Эти и другие отношения дуальности обсуждены более подробно в главах 10 и 11.

Polar Notation

Полярная Система обозначений

As it has been described so far, the frequency domain is a group of amplitudes of cosine and sine waves (with slight scaling modifications). This is called **rectangular** notation. Alternatively, the frequency domain can be expressed in **polar** form. In this notation, $ReX[]$ $Im X[]$ are replaced with two other arrays, called the **Magnitude of X[]**, written in equations as: **Mag X[]** and the **Phase of X[]**, written as: **Phase X[]**. The magnitude and phase are a pair-for-pair replacement for the real and imaginary parts. For example, $Mag X[0]$ and $Phase X[0]$ are calculated using only $ReX[0]$ and $Im X[0]$. Likewise, and $Mag X[14]$ and $Phase X[14]$ calculated using only $ReX[14]$ and $Im X[14]$, and so forth. To understand the conversion, consider what happens when you add a cosine wave and a sine wave of the same frequency. The result is a cosine wave of the same frequency, but with a new amplitude and a new phase shift. In equation form, the two representations are related:

EQUATION 8-5

The addition of a cosine and sine wave results in a cosine wave with a different amplitude and phase shift. The information contained in A & B is transferred to two other variables, M and θ .

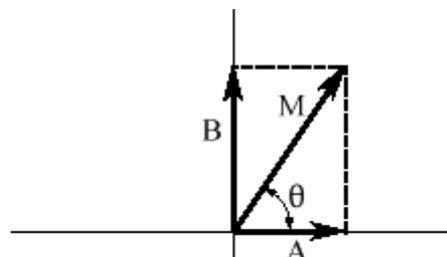
$$A \cos(x) + B \sin(x) = M \cos(x + \theta)$$

УРАВНЕНИЕ 8-5

Сложение волн косинуса и синуса приводит к волне косинуса с различной амплитудой и сдвигом фаз. Информация, содержащаяся в A и B передана(перемещена) двум другим переменным, M и θ .

The important point is that no information is lost in this process; given one representation you can calculate the other. In other words, the information contained in the amplitudes A and B , is also contained in the variables M and θ . Although this equation involves sine and cosine waves, it follows the same conversion equations as do simple vectors. Figure 8-9 shows the analogous vector representation of how the two variables, A and B , can be viewed in a rectangular coordinate system, while M and θ are parameters in polar coordinates.

Важный пункт - то, что никакая информация не потеряна в этом процессе; учитывая одно представление(отображение,обозначение) Вы можете вычислить другое. Другими словами, информация, содержащаяся в амплитудах A и B , также содержится в переменных M и θ . Хотя это уравнение включает в себя(содержит) синус и косинус волны, это следует за теми же самыми конверсионными уравнениями также, как и простые векторы. Рисунок 8-9 показывает аналогичное векторное представление как эти две переменные, A и B , могут быть просмотрены в прямоугольной системе координат, в то время как M и θ - параметры в полярных координатах.



$$M = (A^2 + B^2)^{1/2}$$
$$\theta = \arctan(B/A)$$

FIGURE 8-9

Rectangular-to-polar conversion. The addition of a cosine wave and a sine wave (of the same frequency) follows the same mathematics as the addition of simple vectors.

РИСУНОК 8-9

Прямоугольно-полярное преобразование. Сложение волны косинуса и волны синуса (той же самой частоты) следует за той же самой математикой как добавление простых векторов.

In polar notation, $Mag X[k]$ holds the amplitude of the cosine wave (M in Eq. 8-4 and Fig. 8-9), while $Phase X[k]$ holds the phase angle of the cosine wave (θ in Eq. 8-4 and Fig. 8-9). The following equations convert the frequency domain from rectangular to polar notation, and vice versa:

В полярной системе обозначений, $Mag X[k]$ проводит(держит) амплитуду волны косинуса (M в уравнении 8-4 и рис. 8-9), в то время как фаза $X[k]$ проводит(держит) фазовый угол волны косинуса (θ в уравнении 8-4 и рис. 8-9). Следующие уравнения преобразовывают частотный домен из прямоугольной системы обозначений в полярную, и наоборот:

EQUATION 8-6

Rectangular-to-polar conversion. The rectangular representation of the frequency domain, and $ReX[k]$ $Im X[k]$ changed into the polar form, $Mag X[k]$ and $Phase X[k]$

$$MagX[k] = (ReX[k]^2 + ImX[k]^2)^{1/2}$$

$$PhaseX[k] = \arctan\left(\frac{ImX[k]}{ReX[k]}\right)$$

EQUATION 8-7

Polar-to-rectangular conversion. The two arrays, and $Mag X[k]$ $Phase X[k]$ converted into and $ReX[k]$ $Im X[k]$

$$ReX[k] = MagX[k] \cos(PhaseX[k])$$

$$ImX[k] = MagX[k] \sin(PhaseX[k])$$

Rectangular and polar notation allow you to think of the DFT in two different ways. With rectangular notation, the DFT decomposes an N point signal into $N/2 + 1$ cosine waves and $N/2 + 1$ sine waves, each with a specified *amplitude*. In polar notation, the DFT decomposes an N point signal into $N/2 + 1$ cosine waves, each with a specified *amplitude* (called the *magnitude*) and *phase shift*. Why does polar notation use cosine waves instead of sine waves? Sine waves cannot represent the DC component of a signal, since a sine wave of zero frequency is composed of all zeros (see Figs. 8-5 a & b).

Прямоугольная и полярная система обозначений позволят Вам думать о ДПФ двумя различными способами. С прямоугольной системой обозначений, ДПФ расчленяет сигнал точки N в $N/2+1$ волны косинусов и $N/2+1$ волны синусов, каждый с заданной *амплитудой*. В полярной системе обозначений, ДПФ расчленяет сигнал точки N в $N/2+1$ волны косинусов, каждый с заданной *амплитудой* (называемой *магнитудой*? (*величиной абсолютной*)) и *сдвигом фаз*. Почему полярная система обозначений использует волны

косинуса вместо волн синуса? Волны синуса не могут представлять компонент постоянного тока сигнала, так как волна синуса нулевой частоты составлена из всех нулей (см. рис. 8-5 а и б).

Even though the polar and rectangular representations contain exactly the same information, there are many instances where one is easier to use than the other. For example, Fig. 8-10 shows a frequency domain signal in both rectangular and polar form. Warning: Don't try to understand the shape of the real and imaginary parts; your head will explode! In comparison, the polar curves are straightforward: only frequencies below about 0.25 are present, and the phase shift is approximately proportional to the frequency. This is the frequency response of a low-pass filter.

Даже при том, что полярные и прямоугольные представления содержат точно ту же самую информацию, имеются много примеров, где одна из них более легкая для использования, чем другая. Например, рис. 8-10 показывает частотный домен сигнала и в прямоугольной и в полярной форме. Предупреждение: Не пробуйте понять форму реальных(вещественных) и мнимых(несобственных) частей; ваша голова взорвется! Для сравнения, полярные(полярные кривые) прямые: только частоты ниже приблизительно 0.25 присутствуют, и сдвиг фаз приблизительно пропорционален к частоте. Это - частотная характеристика фильтра нижних частот.

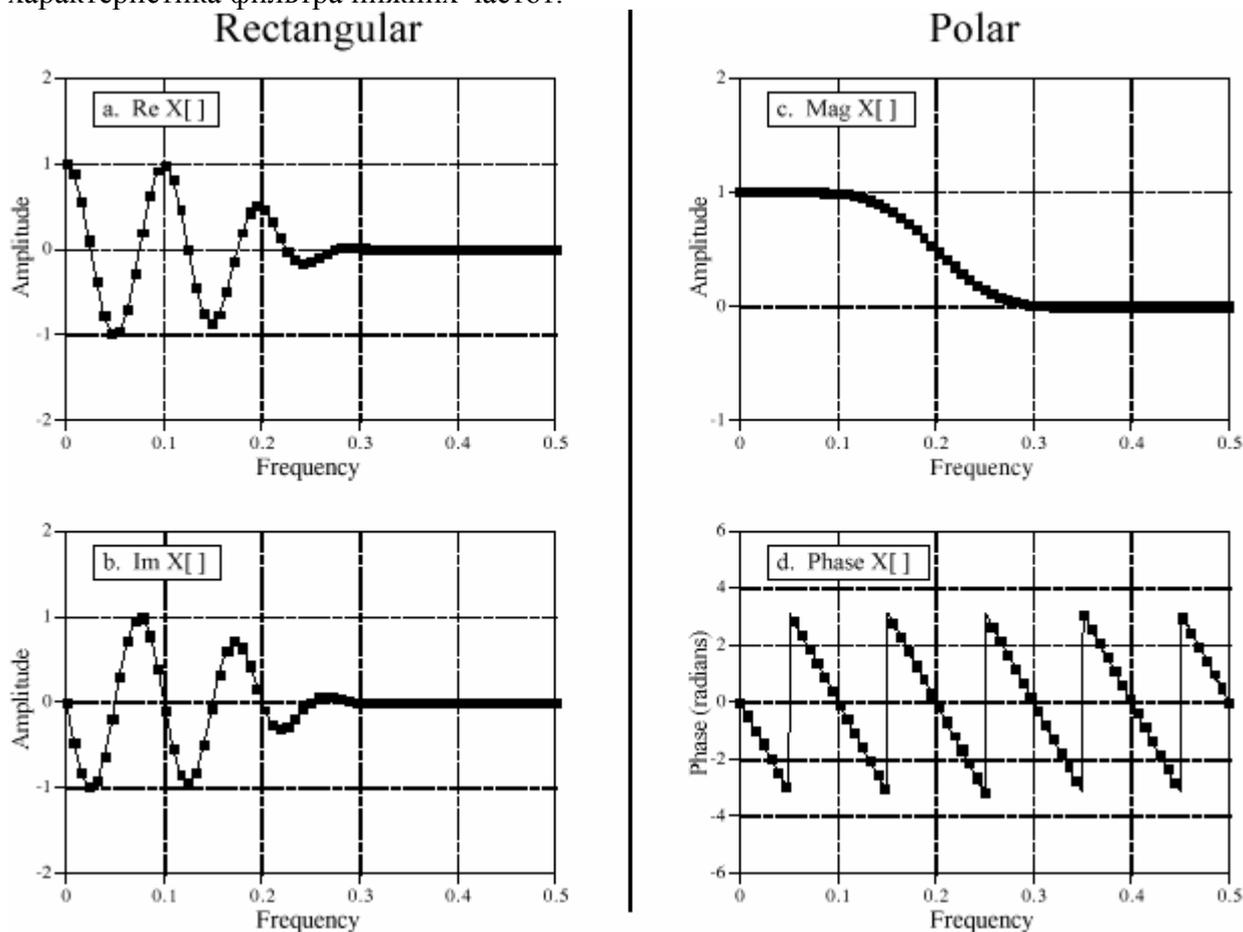


FIGURE 8-10

Example of rectangular and polar frequency domains. This example shows a frequency domain expressed in both rectangular and polar notation. As in this case, polar notation usually provides human observers with a better understanding of the characteristics of the signal. In comparison, the rectangular form is almost always used when math computations are required. Pay special notice to the fact that the first and last samples in the phase must be zero, just as they are in the imaginary part.

РИСУНОК 8-10

НАУЧНО-ТЕХНИЧЕСКОЕ РУКОВОДСТВО ПО ЦИФРОВОЙ ОБРАБОТКЕ СИГНАЛОВ

Пример прямоугольных и полярных частотных доменов. Этот пример показывает частотный домен, выраженный и в прямоугольной и в полярной системе обозначений. Как в этом случае, полярная система обозначений обычно обеспечивает человеческих наблюдателей лучшим пониманием характеристик сигнала. Для сравнения, прямоугольная форма почти всегда используется, когда требуются математические вычисления. Обратите внимание на специальное примечание - на факт, что первые и последние выборки в фазе должны быть нулевые, также, как они находятся в мнимой(несобственной) части.

When should you use rectangular notation and when should you use polar? Rectangular notation is usually the best choice for calculations, such as in equations and computer programs. In comparison, graphs are almost always in polar form. As shown by the previous example, it is nearly impossible for *humans* to understand the characteristics of a frequency domain signal by looking at the real and imaginary parts. In a typical program, the frequency domain signals are kept in rectangular notation until an observer needs to look at them, at which time a rectangular-to-polar conversion is done.

Когда Вы должны использовать прямоугольную систему обозначений и когда Вы должны использовать полярную? Прямоугольная система обозначений - обычно лучший выбор для вычислений, типа в уравнениях и компьютерных программах. Для сравнения, диаграммы(графики) - почти всегда в полярной форме. Как показано предыдущим примером, это почти невозможно для *людей(с человеческой точки зрения)*, чтобы понять характеристики сигнала частотного домена смотря(глядя) на реальные(вещественные) и мнимые(несобственные) части. В типичной программе, сигналы частотного домена сохраняются в прямоугольной системе обозначений, пока наблюдатель не должен посмотреть на них, в которое время сделано преобразование прямоугольного к полярному.

Why is it easier to understand the frequency domain in polar notation? This question goes to the heart of why decomposing a signal into sinusoids is *useful*. Recall the property of *sinusoidal fidelity* from Chapter 5: if a sinusoid enters a linear system, the output will also be a sinusoid, and at exactly the same frequency as the input. Only the amplitude and phase can change. Polar notation directly represents signals in terms of the amplitude and phase of the component cosine waves. In turn, systems can be represented by how they modify the amplitude and phase of each of these cosine waves.

Почему это проще должно понять частотный домен в полярной системе обозначений? Этот вопрос идет к основе(сердцу) того, почему разложение сигнала в синусоиды является полезным. Повторно вызовите свойство синусоидальной точности от главы 5: если синусоида вводит линейную систему, выход будет также синусоида, и в точно та же самая частота как ввод. Только амплитуда и фаза могут изменяться. Полярная система обозначений непосредственно представляет сигналы в терминах амплитуды и фазы составляющих волн косинуса. В свою очередь, системы могут быть представлены тем, как они изменяют амплитуду и фазу каждой из этих волн косинуса.

Now consider what happens if rectangular notation is used with this scenario. A mixture of cosine and sine waves enter the linear system, resulting in a mixture of cosine and sine waves leaving the system. The problem is, a cosine wave on the input may result in both cosine and sine waves on the output. Likewise, a sine wave on the input can result in both cosine and sine waves on the output. While these cross-terms can be straightened out, the overall method doesn't match with why we wanted to use sinusoids in the first place.

Теперь рассмотрите то, что случается, если прямоугольная система обозначений используется с этим сценарием. Смесь волн косинуса и синуса вводит линейную систему, приводя к смеси волн косинуса и синуса, оставляя систему. Проблема, волна косинуса на

вводе может приводить, и к косинусу и волнам синуса на выходе. Аналогично, волна синуса на вводе может приводить, и к волнам косинусу и синуса на выходе. В то время как это пересечение - сроки может быть разглажено, полный метод не соответствует тому, почему мы хотели использовать синусоиды во-первых.

Polar Nuisances

Полярные Помехи

There are many nuisances associated with using polar notation. None of these are overwhelming, just really annoying! Table 8-3 shows a computer program for converting between rectangular and polar notation, and provides solutions for some of these pests.

Имеются много помех, связанных с использованием полярной системы обозначений. Ни одна из них не непреодолима, только действительно раздражает! Таблица 8-3 показывает компьютерную программу для преобразования между прямоугольной и полярной системой обозначений, и обеспечивает решения к некоторым из этих вредителей(помех).

Nuisance 1: Radians vs. Degrees

Помеха 1: Радианы против Градусов

It is possible to express the phase in either *degrees* or *radians*. When expressed in degrees, the values in the phase signal are between -180 and 180. Using radians, each of the values will be between $-\pi$ and π , that is, between -3.141592 to 3.141592. Most computer languages require the use radians for their trigonometric functions, such as cosine, sine, arctangent, etc. It can be irritating to work with these long decimal numbers, and difficult to interpret the data you receive. For example, if you want to introduce a 90 degree phase shift into a signal, you need to add 1.570796 to the phase. While it isn't going to kill you to type this into your program, it does become tiresome. The best way to handle this problem is to define the constant, $PI = 3.141592$, at the beginning of your program. A 90 degree phase shift can then be written as $PI/2$. Degrees and radians are both widely used in DSP and you need to become comfortable with both.

Возможно выразить фазу или в *градусах* или *радианах*. Когда выражено в градусах, значения в фазовом сигнале - между -180° и 180° . Используя радианы, каждое из значений будет между $-\pi$ и π , то есть от -3.141592 до 3.141592. Большинство машинных языков требует использования радианов для их тригонометрических функций, типа косинуса, синуса, арктангенса, и т.д. Может быть раздражающе работать с этими длинными десятичными числами, и трудно интерпретировать данные, которые Вы получаете. Например, если Вы хотите представить сдвига фазы сигнала в 90° , Вы должны добавить 1.570796 к фазе. В то время как это не собирается уничтожить Вас, чтобы напечатать это в вашу программу, это станет утомительным. Лучший способ обрабатывать эту проблему состоит в том, чтобы определить константу, $PI = 3.141592$, в начале вашей программы. 90° сдвига фаз может тогда быть написано как $PI/2$. И градусы, и радианы широко используются в ЦОС, и Вы должны стать удобными с обоими.

```
100                                'RECTANGULAR-TO-POLAR & POLAR-TO-RECTANGULAR CONVERSION
110 '
120 DIM REX[256]                   'REX[ ] holds the real part
130 DIM IMX[256]                   'IMX[ ] holds the imaginary part
```

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

```

140 DIM MAG[256]      'MAG[ ] holds the magnitude
150 DIM PHASE[256]    'PHASE[ ] holds the phase
160 '
170 PI = 3.14159265
180 '
190 GOSUB XXXX        'Mythical subroutine to load data into REX[ ] and IMX[ ]
200 '
210 '
220 '                'Rectangular-to-polar conversion, Eq. 8-6
230 FOR K% = 0 TO 256
240 MAG[K%] = SQR( REX[K%]^2 + IMX[K%]^2 )                'from Eq. 8-6
250 IF REX[K%] = 0 THEN REX[K%] = 1E-20                  'prevent divide by 0 (nuisance 2)
260 PHASE[K%] = ATN( IMX[K%] / REX[K%] )                'from Eq. 8-6
270 '                                                    'correct the arctan (nuisance 3)
280 IF REX[K%] < 0 AND IMX[K%] < 0 THEN PHASE[K%] = PHASE[K%] - PI
290 IF REX[K%] < 0 AND IMX[K%] >= 0 THEN PHASE[K%] = PHASE[K%] + PI
300 NEXT K%
310 '
320 '
330 '                                                    'Polar-to-rectangular conversion, Eq. 8-7
340 FOR K% = 0 TO 256
350 REX[K%] = MAG[K%] * COS( PHASE[K%] )
360 IMX[K%] = MAG[K%] * SIN( PHASE[K%] )
370 NEXT K%
380 '
390 END

```

Таблица 8-3

Nuisance 2: Divide by zero error

Помеха 2: ошибка Деления на ноль

When converting from rectangular to polar notation, it is very common to find frequencies where the real part is zero and the imaginary part is some nonzero value. This simply means that the phase is exactly 90 or -90 degrees. Try to tell your computer this! When your program tries to calculate the phase from: $Phase X[k] = \arctan(Im X[k] / Re X[k])$, a *divide by zero error* occurs. Even if the program execution doesn't halt, the phase you obtain for this frequency won't be correct. To avoid this problem, the real part must be tested for being zero before the division. If it is zero, the imaginary part must be tested for being positive or negative, to determine whether to set the phase to $\pi/2$ or $-\pi/2$, respectively. Lastly, the division needs to be bypassed. Nothing difficult in all these steps, just the potential for aggravation. An alternative way to handle this problem is shown in line 250 of Table 8-3. If the real part is zero, change it to a negligibly small number to keep the math processor happy during the division.

При преобразовании прямоугольной системы обозначений в полярную, очень общее(обычно) найти частоты, где вещественная часть нулевая, и мнимая(несобственная) часть - некоторое значение отличное от нуля. Это просто означает, что фаза - точно 90° или -90°. Попытайтесь сообщить это вашему компьютеру! Когда ваша программа пробует вычислять фазу от: $Phase X[k] = \arctan(Im X[k] / Re X[k])$, происходит ошибка деления на ноль. Даже если выполнение программы не остановится, фаза которую Вы получите для этой частоты, будет неправильна. Чтобы избежать этой проблемы, перед делением вещественная часть должна быть проверена не является ли она нулем. Если это ноль, мнимая(несобственная) должна быть проверена часть, будучи положительной или отрицательной, определить, установить ли фазу в $\pi/2$ или $-\pi/2$, соответственно. Наконец, деление должно быть обойдено. Нет ничего трудного во всех этих шагах, только потенциал для ухудшения. Альтернативный способ обрабатывать эту проблему

показывается в строке 250 таблицы 8-3. Если вещественная часть нулевая, изменение ее на маленькое отрицательное число, чтобы сохранить математический процессор счастливым(довольным) в течение деления.

Nuisance 3: Incorrect arctan

Помеха 3: Неправильный арктангенс

Consider a frequency domain sample where $ReX[k] = 1$ and $Im X[k] = 1$. Equation 8-6 provides the corresponding polar values of $Mag X[k] = 1.414$ and $Phase X[k] = 45^\circ$. Now consider another sample where $ReX[k] = -1$ and $Im X[k] = -1$. Again, Eq. 8-6 provides the values of $Mag X[k] = 1.414$ and $Phase X[k] = 45^\circ$. The problem is, the phase is wrong! It should be -135° . This error occurs whenever the real part is negative. This problem can be corrected by testing the real and imaginary parts after the phase has been calculated. If both the real and imaginary parts are negative, subtract 180° (or π radians) from the calculated phase. If the real part is negative and the imaginary part is positive, add 180° (or π radians). Lines 340 and 350 of the program in Table 8-3 show how this is done. If you fail to catch this problem, the calculated value of the phase will only run between $-\pi/2$ and $\pi/2$, rather than between $-\pi$ and π . Drill this into your mind. If you see the phase only extending to ± 1.5708 , you have forgotten to correct the ambiguity in the arctangent calculation.

Рассмотрите выборку частотного домена где $ReX[k] = 1$ и $Im X[k] = 1$. Уравнение 8-6 обеспечивает соответствующие полярные значения $Mag X[k] = 1.414$ и $Phase X[k] = 45^\circ$. Теперь рассмотрите другую выборку где $ReX[k] = -1$ и $Im X[k] = -1$. Снова, уравнение 8-6 обеспечивает значения $Mag X[k] = 1.414$ и $Phase X[k] = 45^\circ$. Проблема, фаза неправильна! Это должно быть -135° . Эта ошибка происходит всякий раз, когда вещественная часть отрицательна. Эта проблема может быть исправлена, проверяя реальные(вещественные) и мнимые части после того, как фаза была рассчитана. Если и реальные(вещественные) и мнимые части отрицательны, вычитается 180° (или π радиан) от рассчитанной фазы. Если вещественная часть отрицательна, а мнимая(несобственная) часть положительна, прибавляется 180° (или π радиан). Строки 340 и 350 из программы в таблице 8-3 показывают, как это сделано. Если Вы будете не в состоянии охватить(поймать,понять) эту проблему, расчет значения фазы будет работать только между $-\pi/2$ и $\pi/2$, скорее чем между $-\pi$ и π . Это отверстие(пробел) в вашем мнении(знании, интеллекте). Если Вы видите фазу, продлевающуюся только на ± 1.5708 , Вы забыли исправить неоднозначность в вычислении арктангенса.

Nuisance 4: Phase of very small magnitudes

Помеха 4: Фаза очень маленьких величин

Imagine the following scenario. You are grinding away at some DSP task, and suddenly notice that part of the phase doesn't look right. It might be noisy, jumping all over, or just plain *wrong*. After spending the next hour looking through hundreds of lines of computer code, you find the answer. The corresponding values in the magnitude are so small that they are buried in round-off noise. If the magnitude is negligibly small, the phase doesn't have any meaning, and can assume unusual values. An example of this is shown in Fig. 8-11. It is usually obvious when an *amplitude* signal is lost in noise; the values are so small that you are forced to suspect that the values are meaningless. The phase is different. When a polar signal is contaminated with noise, the values in the phase are random numbers between $-\mathbf{B}$ and \mathbf{B} . Unfortunately, this often *looks* like a real signal, rather than the nonsense it really is.

Вообразите следующий сценарий. Вы вкальваете(работаете) над некоторой задачей ЦОС, и внезапно(вдруг) обращаете внимание, что часть фазы не выглядит правой(правильной). Это могло бы быть шумом, переходящий на всем протяжении, или очевидно неправильным(ошибочным, ложным). После расхода следующего часа, просматривая сотни строк компьютерного кода, Вы находите ответ. Соответствующие значения в величине настолько маленькие, что они захоронены в окружающем шуме. Если величина отрицательная маленькая, фаза не имеет никакого значения(смысла), и может принимать необычные значения. Пример этого показывается в рис. 8-11. Обычно очевидно, когда амплитудный сигнал потерян в шуме; значения настолько маленькие, что Вы вынуждены подозревать, что значения являются бессмысленными. Фаза различна. Когда полярный сигнал загрязнен шумом, значения в фазе - случайные числа между -2π и 2π . К сожалению, это часто напоминает реальный(вещественный) сигнал, скорее чем бессмыслицу(абсурд), это действительно(реально).

Nuisance 5: 2π ambiguity of the phase
Помеха 5: неоднозначность фазы 2π

Look again at Fig. 8-10d, and notice the several discontinuities in the data. Every time a point looks as if it is going to dip below -3.141592 , it snaps back to 3.141592 . This is a result of the periodic nature of sinusoids. For example, a phase shift of θ is exactly the same as a phase shift of $\theta+2\pi$, $\theta+4\pi$, $\theta+6\pi$, etc. Any sinusoid is unchanged when you add an integer multiple of 2π to the phase. The apparent discontinuities in the signal are a result of the computer algorithm picking its favorite choice from an infinite number of equivalent possibilities. The smallest possible value is always chosen, keeping the phase between $-\pi$ and π .

Посмотрите снова на рис. 8-10d, и обратите внимание на несколько разрывов в данных. Каждый раз точка смотрится(выглядит), как будто собирается опускаться ниже -3.141592 , это поддерживается обратно к 3.141592 (не дает опуститься ниже 3.141592). Это – результат(следует из) периодического характера синусоид. Например, сдвиг фаз θ - точно тот же самый как - сдвиг фаз $\theta+2\pi$, $\theta+4\pi$, $\theta+6\pi$, и т.д. Любая синусоида неизменяема, когда Вы прибавляете кратное 2π к фазе. Очевидные разрывы в сигнале - результат компьютерного алгоритма, выбирающего его благоприятный выбор от бесконечного числа эквивалентных возможностей. Всегда выбирается самое маленькое возможное значение, сохраняя фазу между $-\pi$ и π .

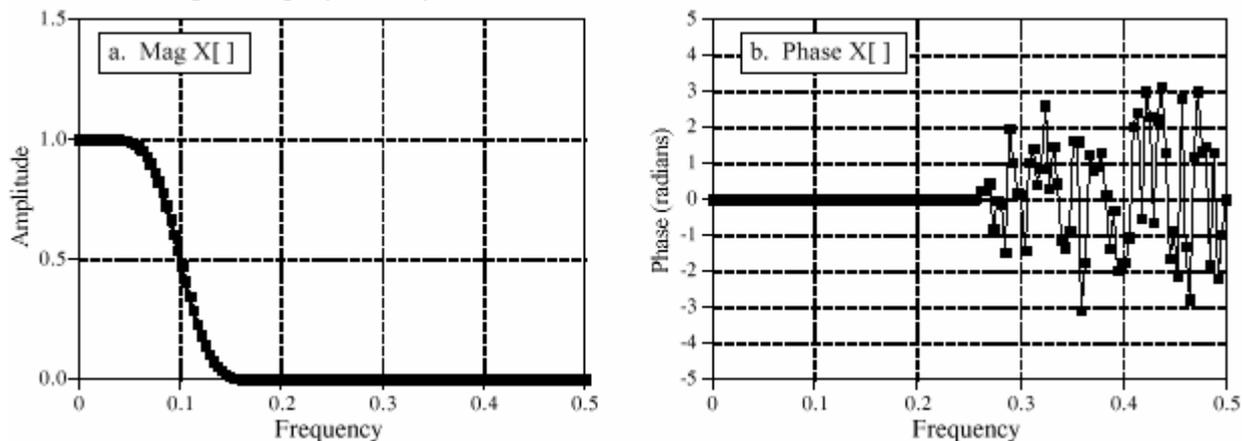


FIGURE 8-11

The phase of small magnitude signals. At frequencies where the magnitude drops to a very low value, round-off noise can cause wild excursions of the phase. Don't make the mistake of thinking this is a meaningful signal.

РИСУНОК 8-11

Фаза сигналов маленькой величины. В частотах, где величина понижается к очень низкому значению, окружающий шум может вызывать дикие отклонения фазы. Не делайте ошибку из размышления, что это является значимым сигналом.

FIGURE 8-12

Example of phase unwrapping. The top curve shows a typical phase signal obtained from a rectangular-to-polar conversion routine. Each value in the signal must be between $-\pi$ and π (i.e., -3.14159 and 3.14159). As shown in the lower curve, the phase can be *unwrapped* by adding or subtracting integer multiplies of 2π from each sample, where the integer is chosen to minimize the discontinuities between points.

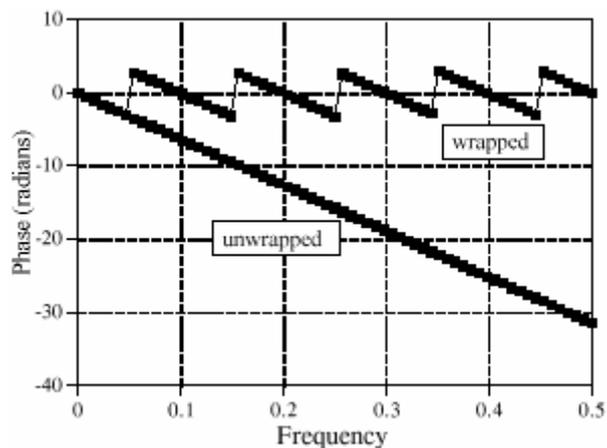


РИСУНОК 8-12

Пример фазового разворачивания. Верхняя кривая показывает типичный фазовый сигнал, полученный из прямоугольного к полярной конверсионной подпрограммой(подпрограммой преобразования). Каждое значение в сигнале должно быть между $-\pi$ и π (то есть, -3.14159 и 3.14159). Как показано в более низкой кривой, фаза может быть *развернута*, прибавляя, или вычитая целое число умножая на 2π каждую выборку, где целое число выбрано, чтобы минимизировать разрывы между точками.

It is often easier to understand the phase if it does not have these discontinuities, even if it means that the phase extends above π , or below $-\pi$. This is called **unwrapping the phase**, and an example is shown in Fig. 8-12. As shown by the program in Table 8-4, a multiple of 2π is added or subtracted from each value of the phase. The exact value is determined by an algorithm that minimizes the difference between adjacent samples.

Часто проще понять(интерпретировать) фазу, если это не имеет этих разрывов, даже если это означает, что расширяет(продлевает?) фазу выше π , или ниже $-\pi$. Это называется, разворачиванием фазы, и пример показывается в рис. 8-12. Как показано программой в таблице 8-4, множитель 2π добавлен или вычитается от каждого значения фазы. Точное значение определено алгоритмом, который минимизирует разность между смежными выборками.

Nuisance 6: The magnitude is always positive (π ambiguity of the phase)
Помеха 6: величина всегда положительна (неоднозначность фазы π)

Figure 8-13 shows a frequency domain signal in rectangular and polar form. The real part is smooth and quite easy to understand, while the imaginary part is entirely zero. In comparison, the polar signals contain abrupt discontinuities and sharp corners. This is because the magnitude must always be positive, *by definition*. Whenever the real part dips below zero, the magnitude remains positive by changing the phase by π (or $-\pi$, which is the same thing). While this is not a problem for the mathematics, the irregular curves can be difficult to interpret.

Рисунок 8-13 показывает частотный домен сигнала в прямоугольной и полярной форме. Вещественная часть гладка и весьма облегчает, чтобы понять, в то время как мнимая часть полностью нулевая. Для сравнения, полярные сигналы содержат резкие разрывы и крутые углы. Это - то, потому что величина должна всегда быть положительна, *по определению*. Всякий раз, когда падение вещественной части ниже нуля, величина(магнитуда) остается уверенной, изменяя фазу π (или $-\pi$, который является той же самой вещью). В то время как это - не, проблема для математики, неправильные кривые может быть трудно интерпретировать.

```

100 ' PHASE UNWRAPPING
110 '
120 DIM PHASE[256] 'PHASE[ ] holds the original phase
130 DIM UWPHASE[256] 'UWPHASE[ ] holds the unwrapped phase
140 '
150 PI = 3.14159265
160 '
170 GOSUB XXXX 'Mythical subroutine to load data into PHASE[ ]
180 '
190 UWPHASE[0] = 0 'The first point of all phase signals is zero
200 '
210 ' 'Go through the unwrapping algorithm
220 FOR K% = 1 TO 256
230 C% = CINT( (UWPHASE[K%-1] - PHASE[K%]) / (2 * PI) )
240 UWPHASE[K%] = PHASE[K%] + C%*2*PI
250 NEXT K%
260 '
270 END
    
```

Таблица 8-4

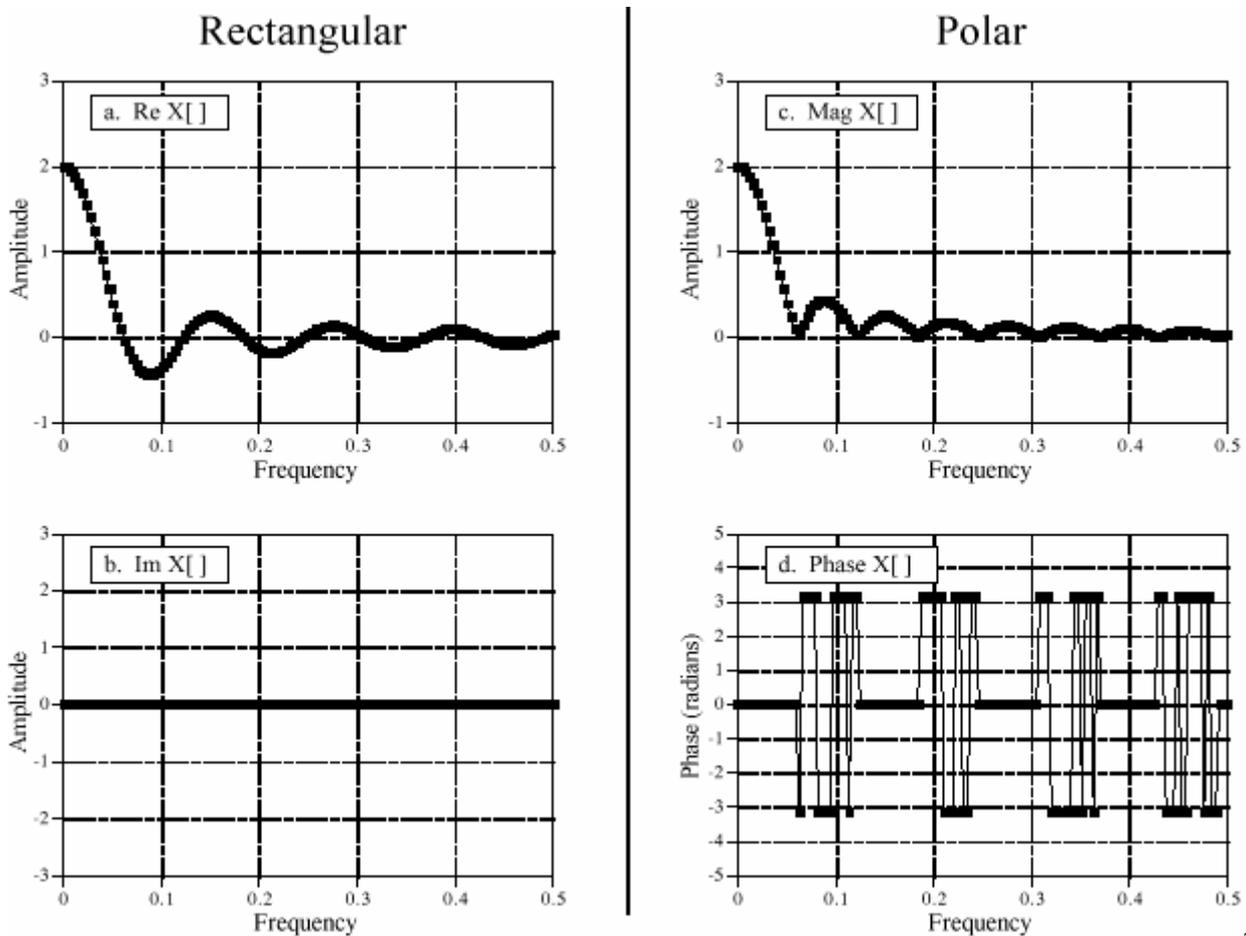


FIGURE 8-13

Example signals in rectangular and polar form. Since the magnitude must always be positive (by definition), the magnitude and phase may contain abrupt discontinuities and sharp corners. Figure (d) also shows another nuisance: random noise can cause the phase to rapidly oscillate between π or $-\pi$.

Пример сигнала в прямоугольной и полярной форме. Так как величина(магнитуда) должна всегда быть положительна (по определению), величина, и фаза может содержать резкие разрывы и крутые углы. Рисунок (d) также показывает другую помеху: случайный шум может заставлять фазу быстро колебаться между π или $-\pi$.

One solution is to allow the magnitude to have *negative* values. In the example of Fig. 8-13, this would make the magnitude appear the same as the real part, while the phase would be entirely zero. There is nothing wrong with this if it helps your understanding. Just be careful not to call a signal with negative values the "magnitude" since this violates its formal definition. In this book we use the weasel words: *unwrapped magnitude* to indicate a "magnitude" that is allowed to have negative values.

Одно решение состоит в том, чтобы позволить величине иметь *отрицательные* значения. В примере рис. 8-13, это заставило бы величину появиться тот же самый как вещественная часть, в то время как фаза была бы полностью нулевая. Не имеется ничего неправильно с этим, если это помогает вашему пониманию. Только будьте внимательным, чтобы не вызвать сигнал с отрицательным значением "величины", так как это нарушает ее формальное определение. В этой книге мы используем переплетающиеся слова: *развернутая величина*, чтобы указать "величину", которой позволяют иметь отрицательные значения.

Nuisance 7: Spikes between π and $-\pi$

Помеха 7: Выбросы между π и $-\pi$

Since π and $-\pi$ represent the same phase shift, round-off noise can cause adjacent points in the phase to rapidly switch between the two values. As shown in Fig. 8-13d, this can produce sharp breaks and spikes in an otherwise smooth curve. Don't be fooled, the phase isn't really this discontinuous.

С тех пор $-\pi$ и π представьте тот же самый сдвиг фаз, окружающий шум может заставлять смежные точки в фазе быстро переключаться (принимать) между двумя значениями. Как показано в рис. 8-13 d, это может производить острые перерывы и выбросы в иначе гладкой кривой. Не будьте введены в заблуждение, эта фаза не действительно не прерывиста.