



TEKTRONIX®

The FFT: Fundamentals and Concepts

by Robert W. Ramirez

\$25.00

(single copy price)

TEKTRONIX®

**The FFT:
Fundamentals and
Concepts**

by Robert W. Ramirez

Tektronix, Inc.
P.O. Box 500
Beaverton, Oregon 97077

ACKNOWLEDGEMENTS

It has been said that writing is somewhat like giving birth, and the pains of labor begin with the first page and run strong through publication. I think this is a fair comparison and, accordingly, feel justified in enjoying the rewards of publication.

But writing is not so much a solitary task as some may think. Many minds influence the words before they are written, and many hands touch the pages before they are bound. So it is the author's duty to acknowledge those who have shared in the task.

First of all, I would like to thank Tektronix, Inc., for making this space available and for the instruments and technical expertise that made **The FFT: Fundamentals and Concepts** possible. I would also like to thank the typesetters and printers at Tektronix, Inc., for putting these pages together. And I would like to give special acknowledgement to Lyle Ochs for his patience in explaining to me the subtle aspects of the FFT, to Dale Aufrecht for a swift and sure editorial pen, and to Ted Niimi for juggling the more than ninety pieces of artwork that I felt were necessary for seeing the Fourier transform in action.

Robert W. Ramirez

Copyright © 1975 by Tektronix, Inc., Beaverton, Oregon. Printed in the United States of America. All rights reserved. Contents of this publication may not be reproduced in any form without permission of Tektronix, Inc.

TABLE OF CONTENTS

	Page
PREFACE	v
PART I INTRODUCTION TO FOURIER ANALYSIS	
SECTION 1	TIME AND FREQUENCY: TWO BASES OF DESCRIPTION
	1-1
	Time Histories Need Time Bases
	1-2
	Sinusoids Look Different from a Frequency Viewpoint
	1-6
	Nonsinusoidal Waveforms are Composed of Sinusoids
	1-10
SECTION 2	FOURIER TECHNIQUES DESCRIBE THE FREQUENCY DOMAIN
	2-1
	FOURIER SERIES GIVES SPECTRA FOR PERIODIC WAVEFORMS
	2-2
	The Question of Existence
	2-2
	Fourier Series Gives Discrete Spectra
	2-3
	Using the Fourier Series—Matters of Practicality
	2-8
	FOURIER INTEGRAL GIVES SPECTRA FOR NONPERIODIC WAVEFORMS
	2-10
	The Fourier Integral is Related to the Fourier Series
	2-10
	Understanding Frequency-Domain Diagrams
	2-15
	Functions are Even or Odd or the Sum of Even and Odd Functions
	2-23
	Periodic or Nonperiodic?—It's Your Point of View
	2-27
	A Summary of Some Important Fourier Transform Properties
	2-33

PART II

DIGITAL FOURIER ANALYSIS

SECTION 3	INTRODUCTION TO THE DISCRETE AND FAST FOURIER TRANSFORMS	3-1
	Widowing and Sampling are Old Ideas with New Names	3-2
	The Discrete Fourier Transform Works on Sampled Data	3-5
	How the FFT Came About	3-9
	Putting the Algorithm to Work—Hardware or Software FFT?	3-11
SECTION 4	UNDERSTANDING FFT RESULTS	4-1
	TEK BASIC FORMATS FOR THE FFT	4-2
	Providing Time-Domain Data for the FFT	4-2
	FFT Results Are Normally in Rectangular Form	4-4
	FFT Results Can Be in Polar Form	4-6
	RECOGNIZING THE REALITIES OF DIGITAL FOURIER ANALYSIS	4-13
	What Is Being Transformed?	4-14
	What Are the Effects of Analog-to-Digital Conversion?	4-19
	Other Sources of Error	4-41
SECTION 5	A GUIDE TO USING THE FFT	5-1
	Some Important Properties of the FFT	5-1
	Some Guidelines for Improving FFT Results	5-2
SECTION 6	A BRIEF LOOK AT SOME FFT APPLICATIONS	6-1
	Distortion Analysis	6-1
	Vibration and Mechanical Signature Analysis	6-3
	Frequency Response Estimation (Transfer Functions)	6-5
	Convolution	6-7
	Correlation	6-8
	Power Spectra	6-13

The FFT

PART III

MATHEMATICS OF THE FFT

SECTION 7

AN ALGORITHM FOR COMPUTING THE DFT 7-1

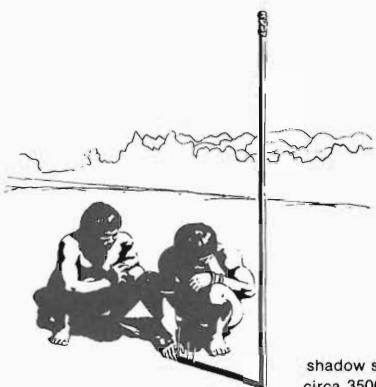
The Sande-Tukey Algorithm for Computing The DFT 7-2

BIBLIOGRAPHY

B-1

INDEX

I-1



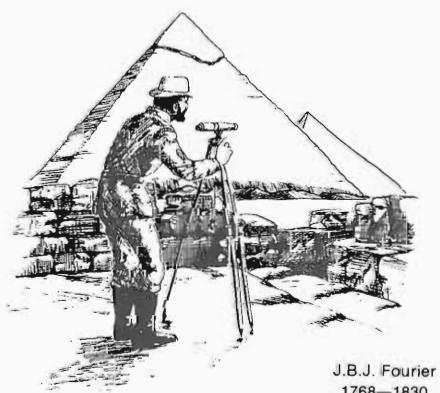
shadow stick
circa 3500 BC



sundial
circa 200 BC

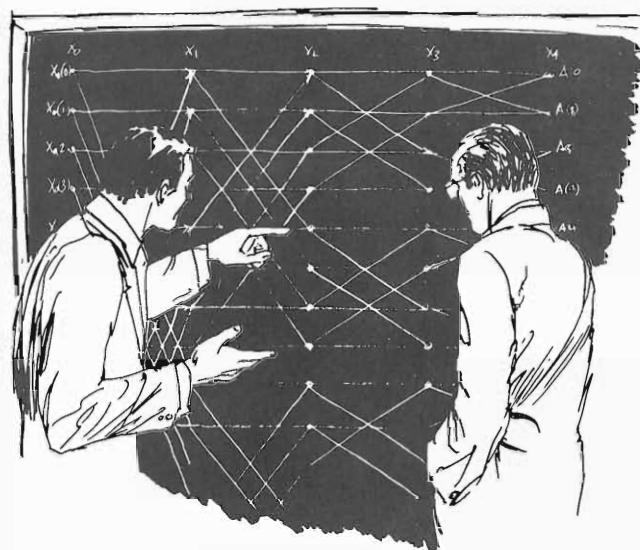


Galileo's pendulum
1582

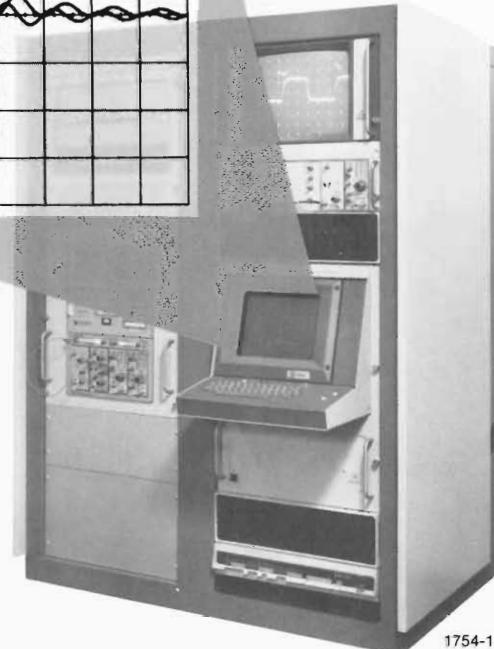
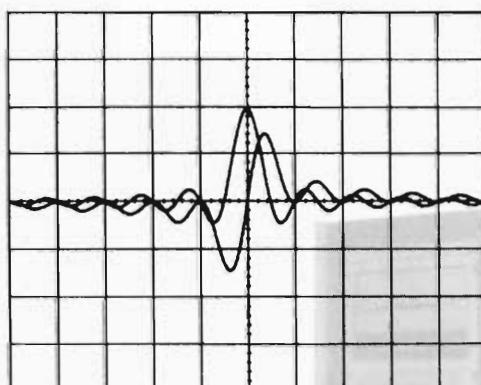


J.B.J. Fourier
1768—1830

The FFT . . .



FFT development and implementation
1960's & 1970's



1754-1

PREFACE

Fourier analysis is not a new subject. It has been around since the early 1800's when J.B.J. Fourier developed the initial concepts and theory. Since then, numerous papers and books dealing with Fourier theory have been published. And the Fourier series and integral have found their way into various college curricula.

So, why another book on Fourier theory?

In answer, Fourier analysis exists in a different context today. It used to be a pencil-and-paper issue, an interesting mathematical approach to getting frequency-domain information, but generally too difficult to apply in most practical cases. And even with the arrival of the digital computer, useful Fourier analyses were too time consuming and computer expensive for widespread use. Then in the 1960's J.W. Cooley and J.W. Tukey published "An Algorithm for the Machine Calculation of Complex Fourier Series." This algorithm became known as the Fast Fourier Transform, or FFT, and has become the new context for Fourier analysis. This is not just a digital context, but a new context that allows quick, economical application of Fourier techniques to a wide variety of analyses.

Thus, the FFT is becoming a general analysis tool. FFT routines are found in most comprehensive software libraries, and FFT analyzers are becoming a more frequently encountered item. But even more than this, the FFT has taken its place, along with more common operations, in general-purpose instrumentation. A good example of this is the Tektronix Digital Processing Oscilloscope, which provides signal processing operations such as waveform addition, multiplication, integration, convolution, correlation, and fast Fourier transformation.

That is the FFT today. What does tomorrow hold? Perhaps, with shrinking computer sizes, microprocessors, etc., a general-purpose FFT instrument will be sitting on your workbench next to your ohmmeter.

Looking to using the FFT now and in the future, you need to be familiar with Fourier theory, not only in the classical context, but in the digital context too. You won't have to know all the details of the equations and derivations, but you do need to know the concepts they embody. You do need to know what to expect in the frequency domain and how digital techniques affect the frequency domain.

For the most part, these things can be shown through simple diagrams and pictures and can be discussed in simple terms. That is the approach that will be taken in the following pages. PART I is an introduction to classical Fourier theory with a slant

The FFT

toward later discussions of digital theory. PART II covers the digital approach to Fourier analysis and uses the Tektronix Digital Processing Oscilloscope and DPO TEK BASIC software to provide specific examples and cathode-ray-tube photos of FFT results. Every attempt has been made throughout to fully illustrate each concept and to discuss each concept in easy-to-understand terms.

Why another book on Fourier theory?—to bridge the gap between classical theory and practical use, and to do it in language that people of different backgrounds and technical levels can understand.

PART I

INTRODUCTION TO FOURIER ANALYSIS

You will find two sections in PART I. The first is but a few pages—just enough to start you thinking about time and frequency as two related concepts. Though the relationship may not be intuitively obvious, an important relationship does exist. And this relationship becomes more obvious in the second section, where the Fourier transform is explored. A good grasp of the concepts in these two sections is necessary for understanding the digital analysis techniques covered in PART II.

SECTION 1

TIME AND FREQUENCY: TWO BASES OF DESCRIPTION

Time. That is one of the fundamental concerns of human beings. How many times during the day do we look at a clock or check our watches? Since birth, our lives have been geared to time. There is a time to wake up, a time to eat, a time to work, a time to play, and a time to go to sleep. We measure each day of our lives in time and use it to order the events that concern and affect us.

Time is universal. All people recognize its passage. All people live by it. Time in itself is central to many philosophical questions: Does time flow by us, or do we advance through time? And the measurement of time is an established science (horology) with a long history.

As far back as 3500 BC, people were known to have erected poles and towers to cast shadows. The length of the shadow being an indication of the time of day. By the eighth century BC, the Egyptians had refined this shadow concept to a fairly accurate sundial and had also developed water clocks as substitutes for night and on cloudy days. Later, the Romans and Greeks refined these devices further, but it wasn't until the 14th century that anything resembling a modern timepiece was developed. Then in 1582, Galileo observed the constancy of the pendulum, and Christiaan Huygens, in 1665, incorporated Galileo's observations in the first pendulum clock. Until the advent of electrically driven clocks, the pendulum clock was the most accurate timepiece

The FFT

available. Now, by the 1967 agreement of the International Conference on Weights and Measures, the atomic clock is the ultimate standard.

Time Histories Need Time Bases

Today, one second is equal to 9,192,631,770 transitions between two specified hyperfine levels of the Cesium-133 atom. But why so much precision in measuring the passage of time? The answer: Science demands it. A great deal of scientific theory is couched in terms of time histories. Furthermore, experimental proof of these theories requires time-domain measurements. And the precision of these measurements depends upon our ability to measure time.

As an example, Galileo reportedly used his own pulse as a timepiece in making his original pendulum observations. Each complete swing of a large pendulum took so many heart beats. With no greater precision than this, it is a natural experimental conclusion to say that a pendulum always shows the same simple harmonic motion. But theory tells us that this is not the case. In fact, for large displacements, the time for a complete swing of a pendulum is greater than for small displacements (Fig. 1-1). Proving this experimentally, however, requires a more precise timepiece than what Galileo had access to.

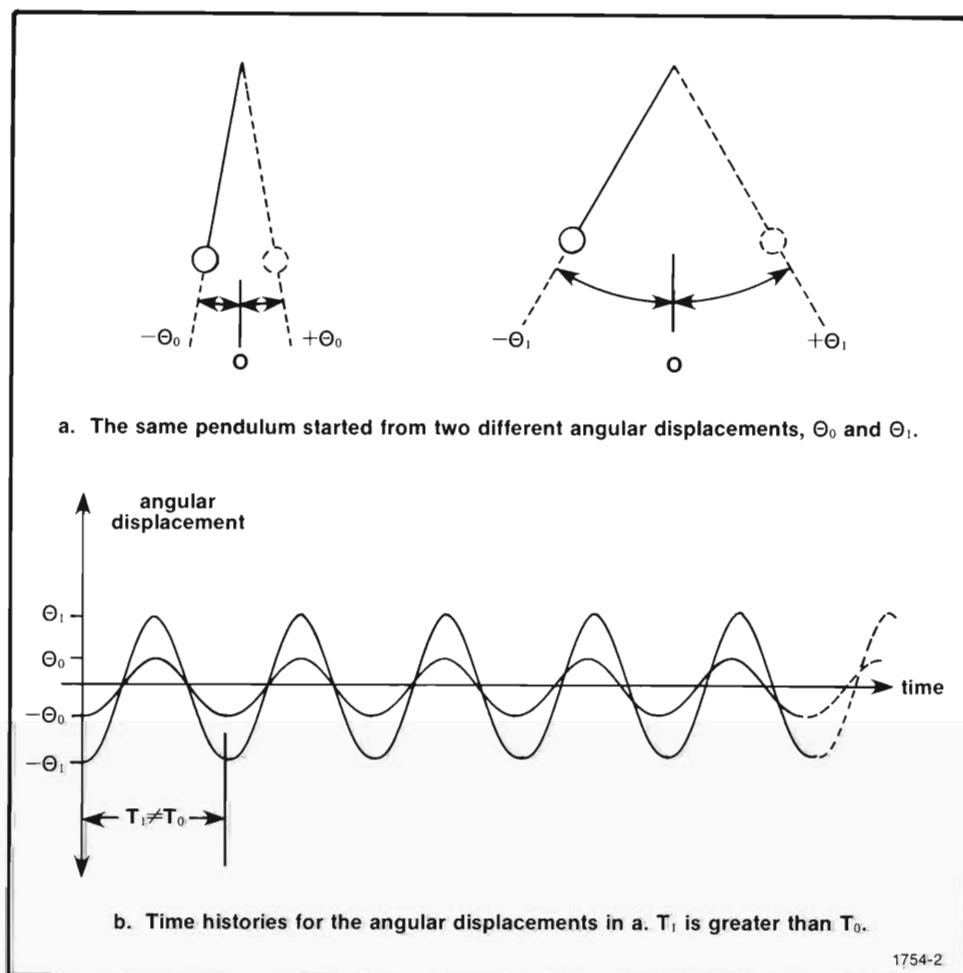


Fig. 1-1. The period of a pendulum, T , varies according to its angular displacement, Θ .

The FFT

An electronic oscillator is in many ways analogous to a mechanical pendulum. The output of a sine-wave oscillator has a time history that closely resembles the time history of angular displacement for a pendulum. Galileo's concept of counting pulse beats can also be applied to measure the time for a complete voltage swing from an oscillator. The modern version of this concept is used in frequency counters and time interval counters, where an electronic pulse is used instead of a human pulse. This is shown in Fig. 1-2.

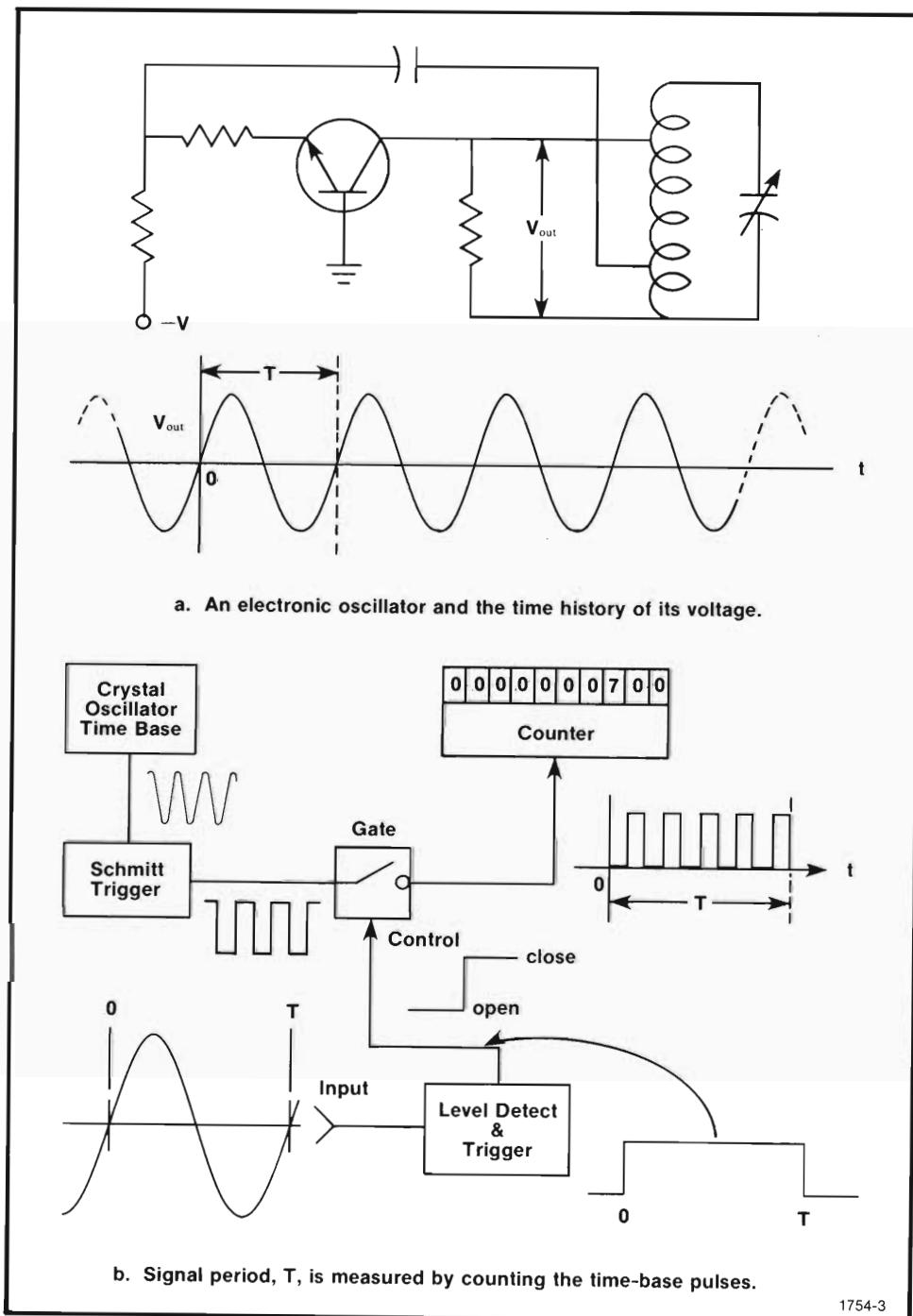


Fig. 1-2. Electronic version of Galileo's pendulum observations.

1754-3

The FFT

There is a time base involved in all time-domain measurements. In the case of Fig. 1-2b, the time base is used directly to measure the period of the signal. In other types of measurements, the time base is used to generate a time axis for an amplitude history. The precisely controlled speed of the paper drive for a chart recorder is an example of this later case. Another example is the oscilloscope, which uses a ramp voltage to drive the Cathode Ray Tube trace at a constant horizontal rate. In both examples, the measured quantity drives the pen or the CRT trace in a direction normal to the time base. The result is a time history of amplitude variation, as shown in the CRT photograph of Fig. 1-3.

The CRT photograph in Fig. 1-3 gives a complete time-domain description of a sinusoid. It is complete because at least one full repetition of the waveform is displayed, and from this we are assured that the waveform is sinusoidal. In general, we can only be sure of the description within the bounds of the display area. What happens outside of the display is not recorded and is unknown. In the case of Fig. 1-3, however, experience and common sense lead us to assume continuation of the waveform beyond the confines of the display.

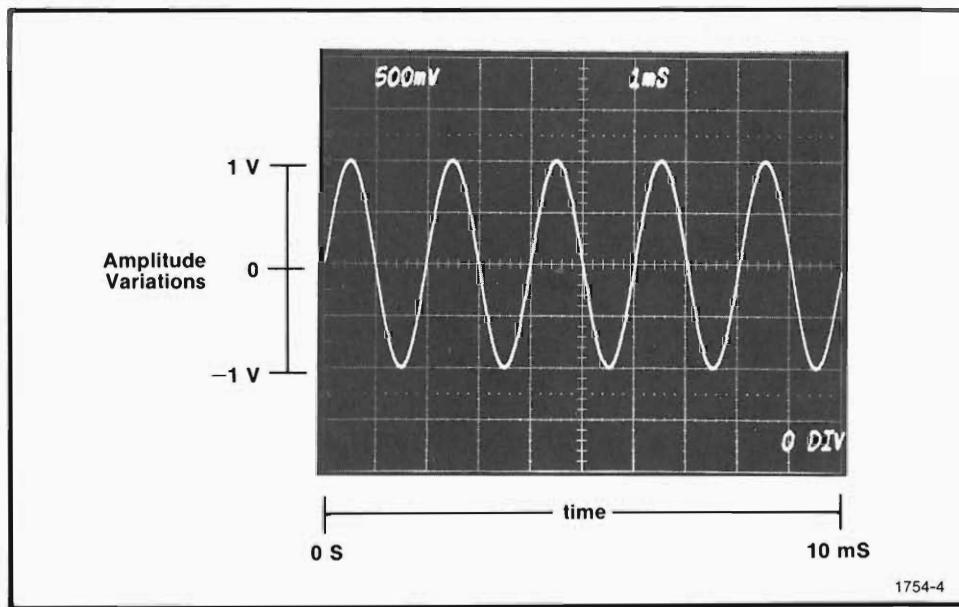


Fig. 1-3. Time history of sinusoidal amplitude variations obtained with an oscilloscope.

CRT Photos

It is conventional in oscilloscope measurements, such as Fig. 1-3, to assign time zero to the left side of the display. Then, according to the time-base setting, time proceeds to the right. In Fig. 1-3, the CRT readout in the upper, right-hand portion indicates the time increments for each major division of the graticule. Vertical amplitude scaling is indicated by the readout in the upper, left-hand corner. Also, the zero reference for the amplitude is given in the lower, right-hand corner. A 0 DIV indicates the center graticule line as the reference. A 3 DIV indicates the third graticule line above center for the reference, and a -3 DIV indicates the third line below center.

This system of CRT readout is not a feature of oscilloscopes in general. Many types of oscilloscopes force you to remember instrument settings and zero reference locations. The Tektronix, Inc., instruments and software used to provide examples for later discussion do, however, have the readout feature.

The FFT

Sinusoids Look Different from a Frequency Viewpoint

Once a sinusoid is completely described with respect to time, we can go on and construct a description with respect to frequency. This is shown in Fig. 1-4.

Fig. 1-4 depicts a three dimensional waveform space, with amplitude as one axis and time and frequency as the other two axes. The time and amplitude axes define something that can be called a "time plane". In the same manner, the frequency and amplitude axes define a "frequency plane" normal to the time plane.

The time history of a sinusoid, such as in Fig. 1-3, can be treated as a projection on the time plane. In concept, the sinusoid can be thought of as actually existing at some distance from the time plane. This distance is measured along the frequency axis and is equal to the reciprocal of the waveform period. Similarly, the sinusoid also has a projection on the frequency plane. This projection takes the form of an impulse with an amplitude equal to the sinusoid's amplitude. Because of symmetry, however, we need only project half of the total amplitude swing. This is shown in Fig. 1-4 by the positive amplitude impulse in the magnitude diagram. The position of this impulse on the frequency axis coincides with the frequency of the sinusoid.

The single impulse in the magnitude diagram defines both the amplitude and frequency of the sinusoid. With only this information, the waveform can be reconstructed in the time domain. Some additional information is needed, however, to fix the sinusoid's position with respect to the zero time reference. This additional information is provided by a phase diagram, which also consists of an impulse located on a frequency axis. The amplitude of this latter impulse indicates the phase.

Phase diagrams for sinusoids can be determined by looking at the positive peak closest to time zero. For the case of Fig. 1-4, the positive peak occurs after time zero by an amount equal to one fourth the period. There are 360 degrees in a cycle or period. The peak is shifted from zero by one fourth of this; thus, the phase is $360^\circ/4$, or 90° . And since the positive peak occurs after time zero, the sinusoid is said to be delayed. As a matter of convention, delay is denoted by negative phase. If the closest positive peak had been located before time zero, then the sinusoid would have been advanced. An advance is denoted by positive phase. These conventions are further illustrated in Fig. 1-5, and more examples are provided in Fig. 1-6.

In looking at Fig. 1-6, it should be pointed out that the total range of shift is -180° to $+180^\circ$, or 360° . With no reference fixed to the sinusoid, an actual shift out of the $360^\circ = 2\pi$ range corresponds to a shift within the 2π range. For example, a sinusoid advanced by $360^\circ + 90^\circ = 450^\circ$ is not generally distinguishable from the same sinusoid advanced by just 90° and would be represented as having 90° shift. This system of representing phase within a 2π range is referred to as modulo 2π phase. If a reference can be attached to the sinusoid, then shifts beyond the 2π range can be represented as such. This latter approach is referred to as a continuous phase representation and is detailed in PART II.

The FFT

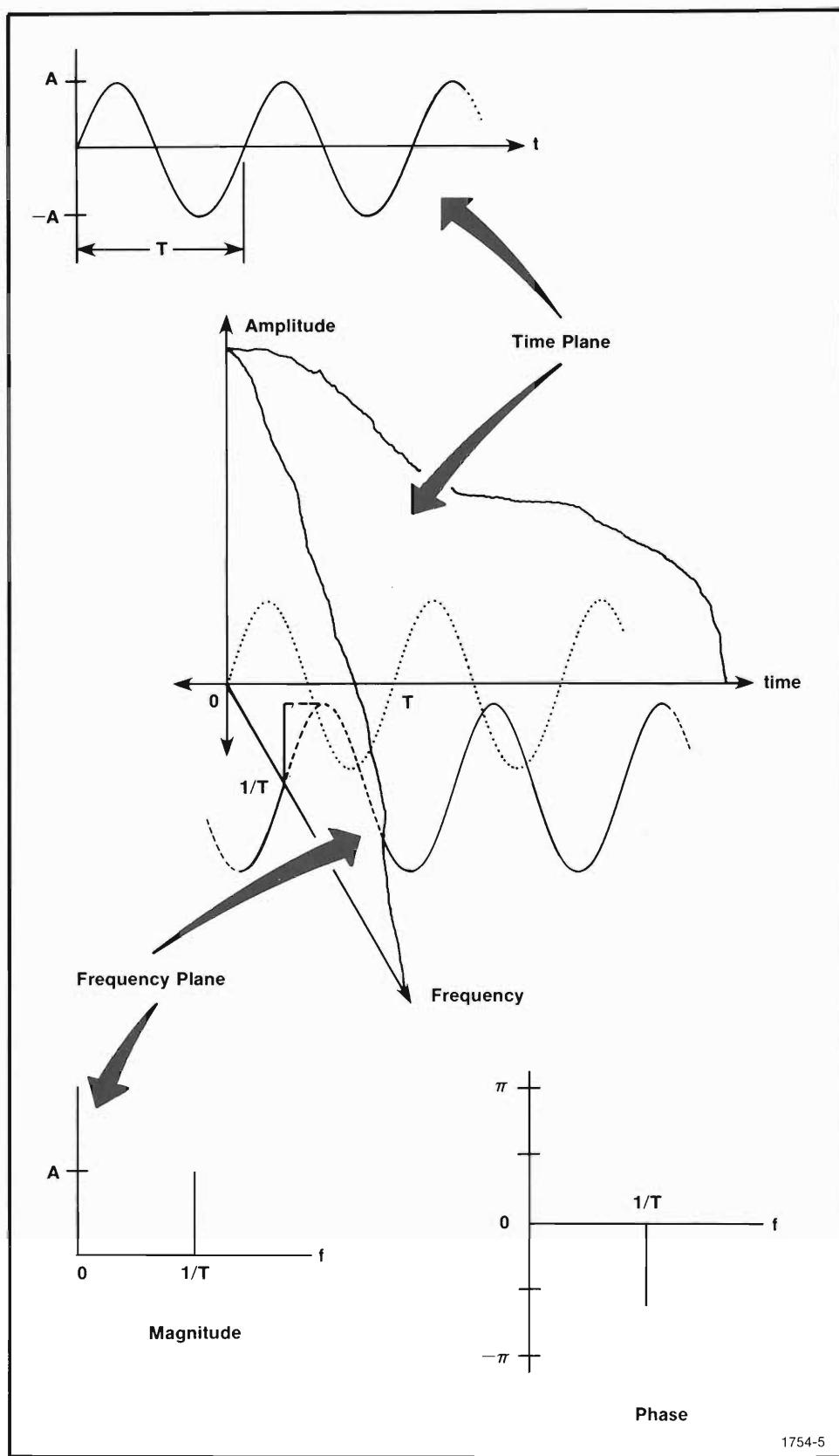


Fig. 1-4. Time and frequency description of a sine wave.

The FFT

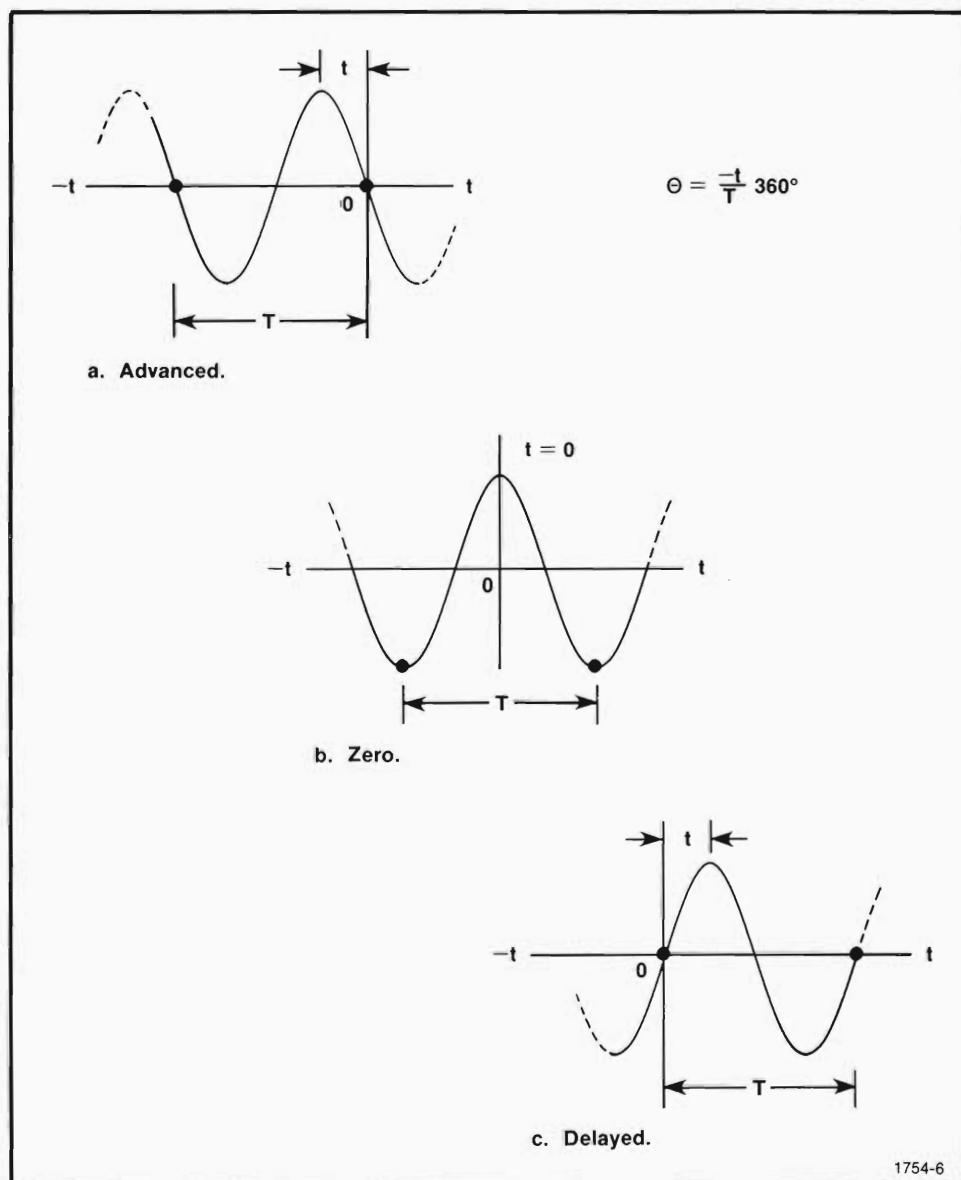


Fig. 1-5. Determining the phase of a sinusoid.

The FFT

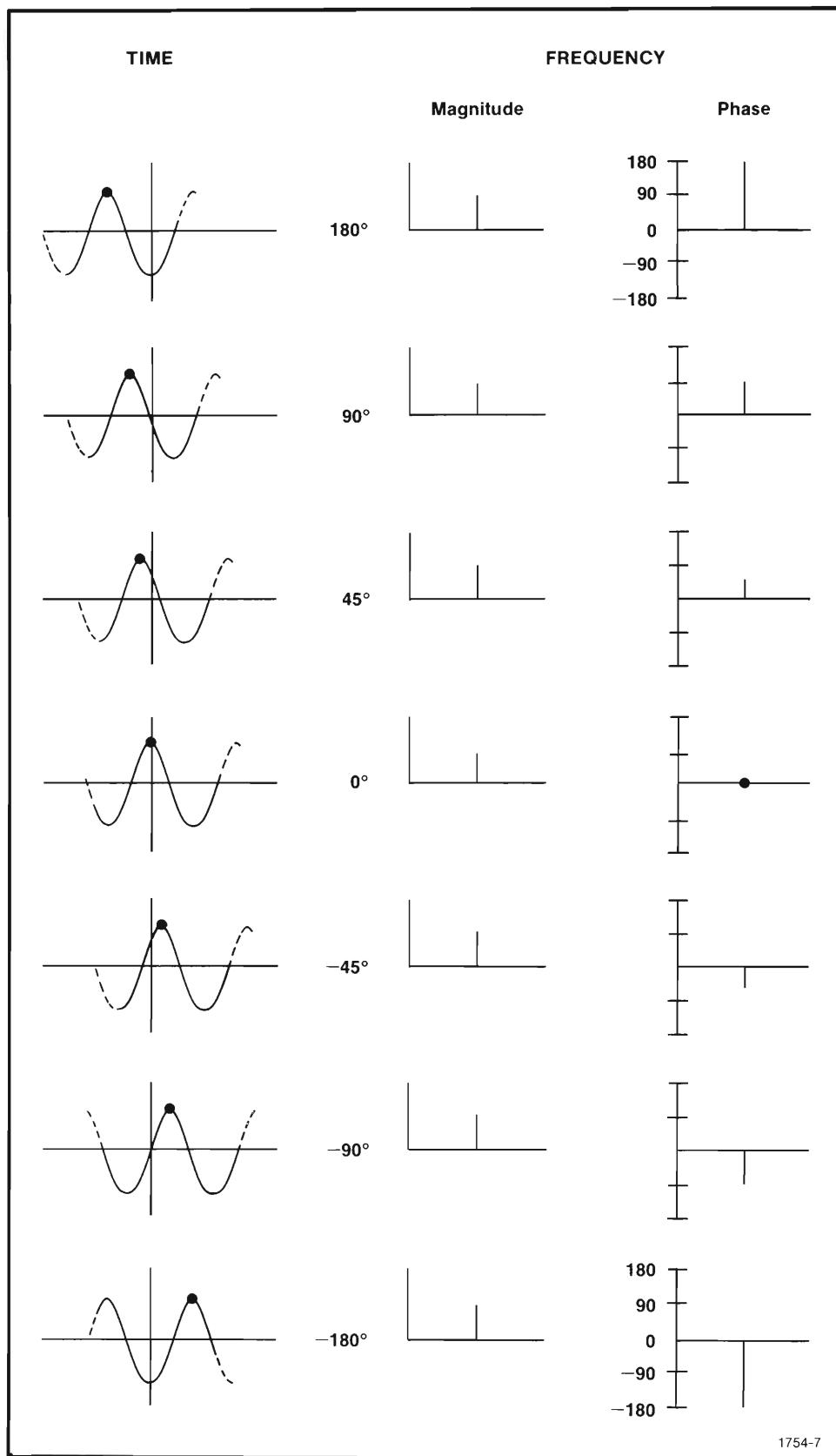


Fig. 1-6. Descriptions of the same sinusoid at various phases.

The FFT

Nonsinusoidal Waveforms are Composed of Sinusoids

By using the description conventions developed thus far, we can build nonsinusoidal waveforms. For example, let's start with the frequency description of a sinusoid having a frequency of F_o and a phase of -90° (Fig. 1-7a). Now let's take another sinusoid with a frequency of $2F_o$. Also, for the sake of illustration, let's say that the amplitude of this sinusoid is one half that for F_o . And to add some interest, let's also give the sinusoid at $2F_o$ a phase of -45° . This second sinusoid is completely described by Fig. 1-7b, and the frequency description for the sum of the two sinusoids is shown in Fig. 1-7c.

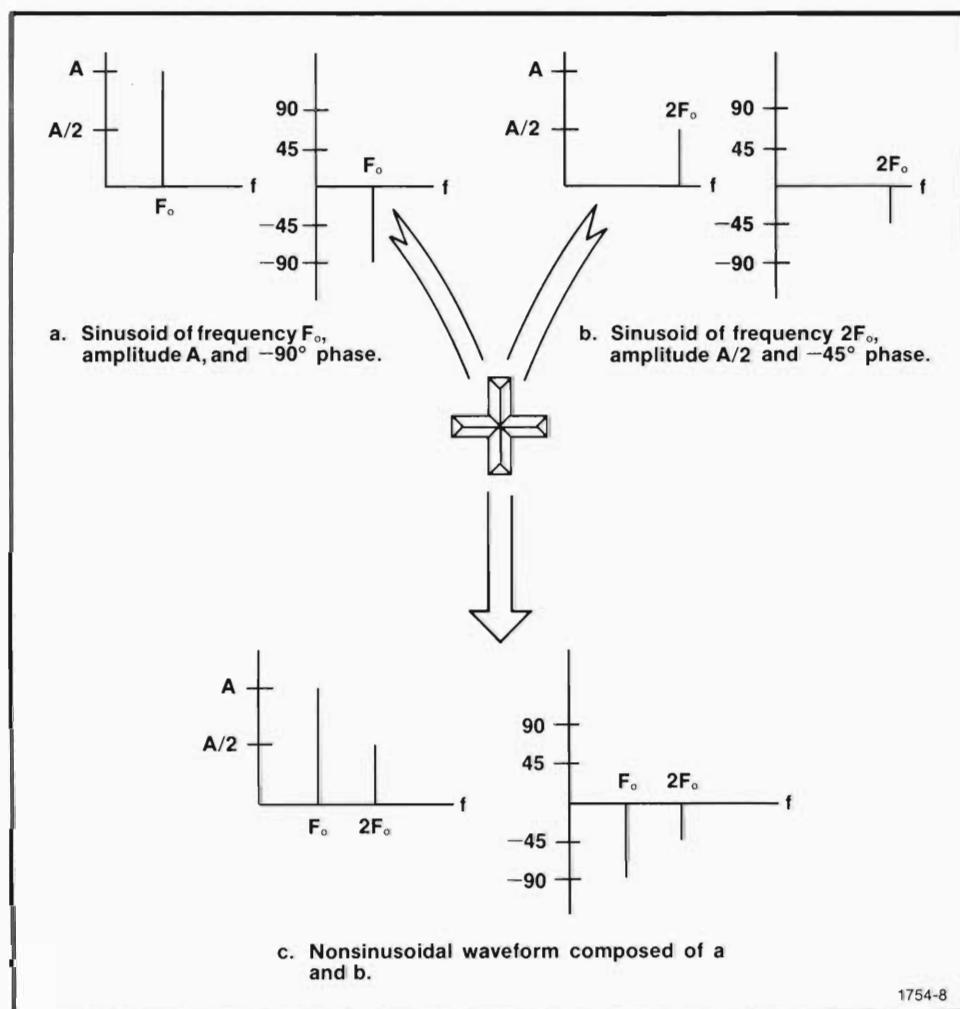


Fig. 1-7. Summing frequency descriptions of sinusoids.

The FFT

To complete the picture, let's recast Fig. 1-7 in terms of three dimensions. This is shown in Fig. 1-8, where the concepts of projecting onto a "time plane" and a "frequency plane" are used. Additionally, the idea of summing multiple time-plane projections is introduced. Projections onto the time plane (or into the time domain, if you prefer) are shown by dotted lines, and their sum is indicated by a heavy solid line.

We could continue on in the manner of Figs. 1-7 and 1-8, and by adding various sinusoids, variously shaped projections on the time plane are obtained. By judiciously selecting each frequency component and adjusting its phase and amplitude, a wide

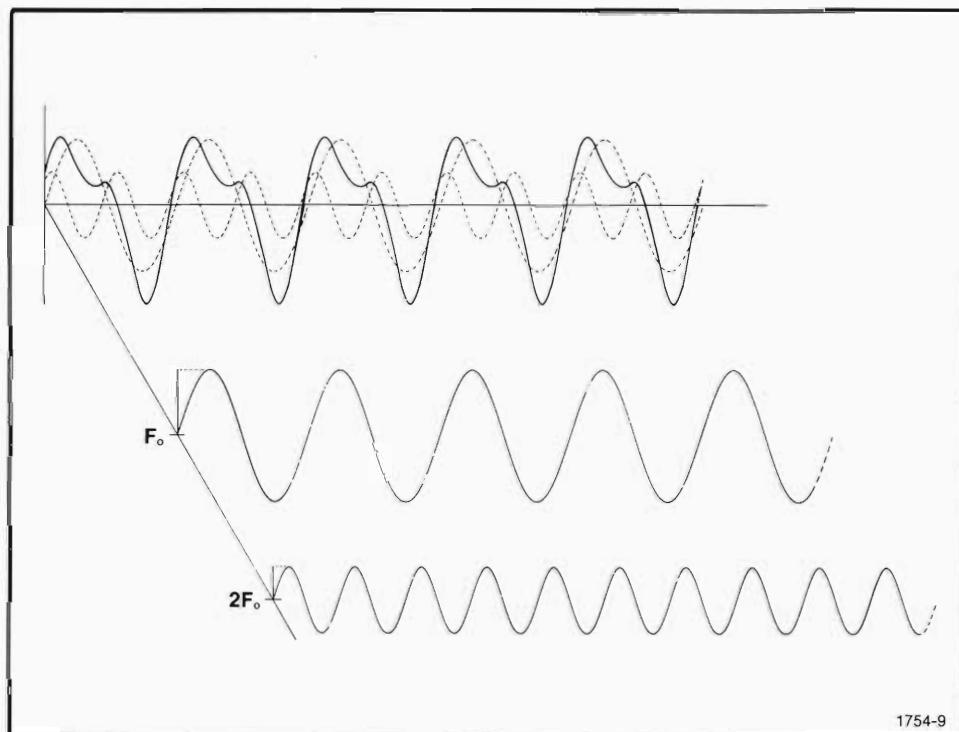


Fig. 1-8. Composing a nonsinusoidal waveform by summing sinusoids.

1754-9

The FFT

variety of repetitive waveforms can be constructed. Each is characterized by a unique combination of sinusoids. If any one of the sinusoids is changed in frequency, amplitude, or phase, then the waveform's time-plane projection changes. This latter point can be illustrated by recomposing the waveform in Fig. 1-8 with the phase at $2F_0$ changed from -45° to $+45^\circ$ (see Fig. 1-9).

For the most part, waveforms are measured in the time domain. Time-based measurements have historical precedence, and they are the most familiar data format. But time histories tell only one side of the story.

Without a direct look at the frequency domain, waveshape changes are the only indication that some frequency components have been modified. In many cases, rules of thumb are enough for interpreting these changes. Passing the square wave in Fig. 1-10a, for example, through a low-pass filter produces the predictable results in Fig. 1-10b. We know that fast rise times imply high frequencies. And, by its nature, a low-pass

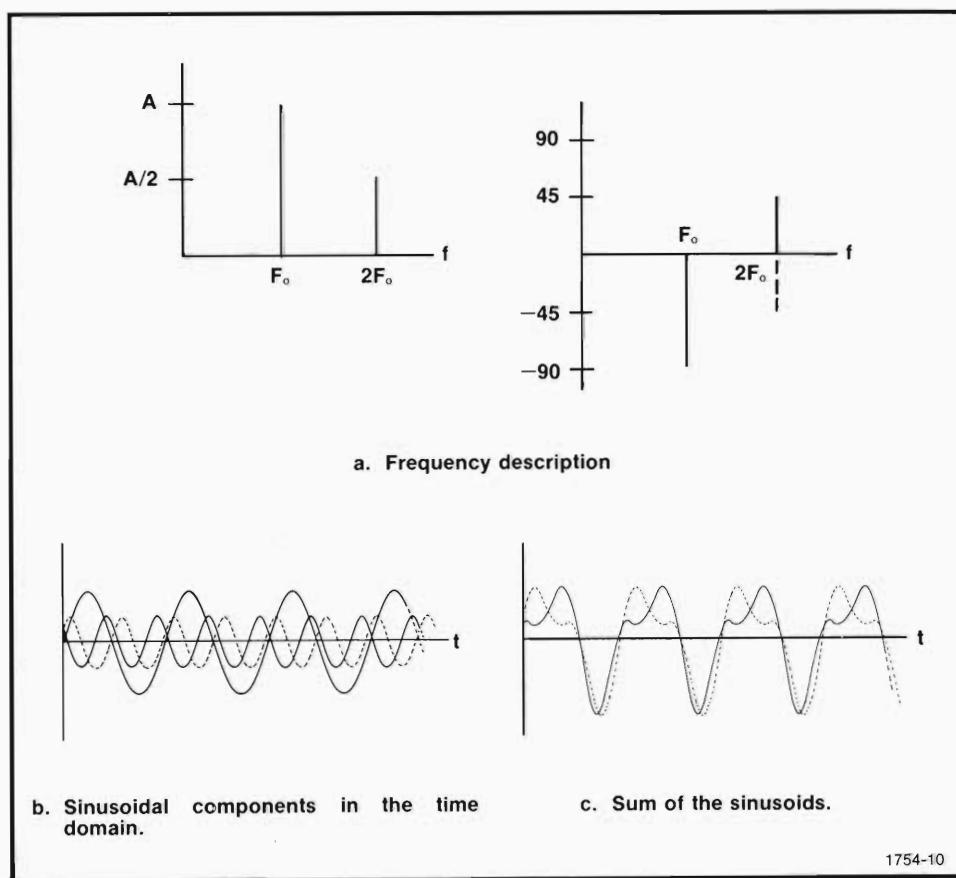


Fig. 1-9. Changing any frequency component, in any way, changes the time-domain description: The phase at $2F_0$ is changed from -45° to $+45^\circ$. The original condition is shown by dotted lines and the new condition by solid lines.

The FFT

filter attenuates high frequencies; thus, the slowed rise time in Fig. 1-10b is expected. But how specific can you be about Fig. 1-10b? Some frequency axes are given in Fig. 1-10c. Can you decompose the waveform in Fig. 1-10b into its frequency components and indicate their magnitudes and phases on these axes?

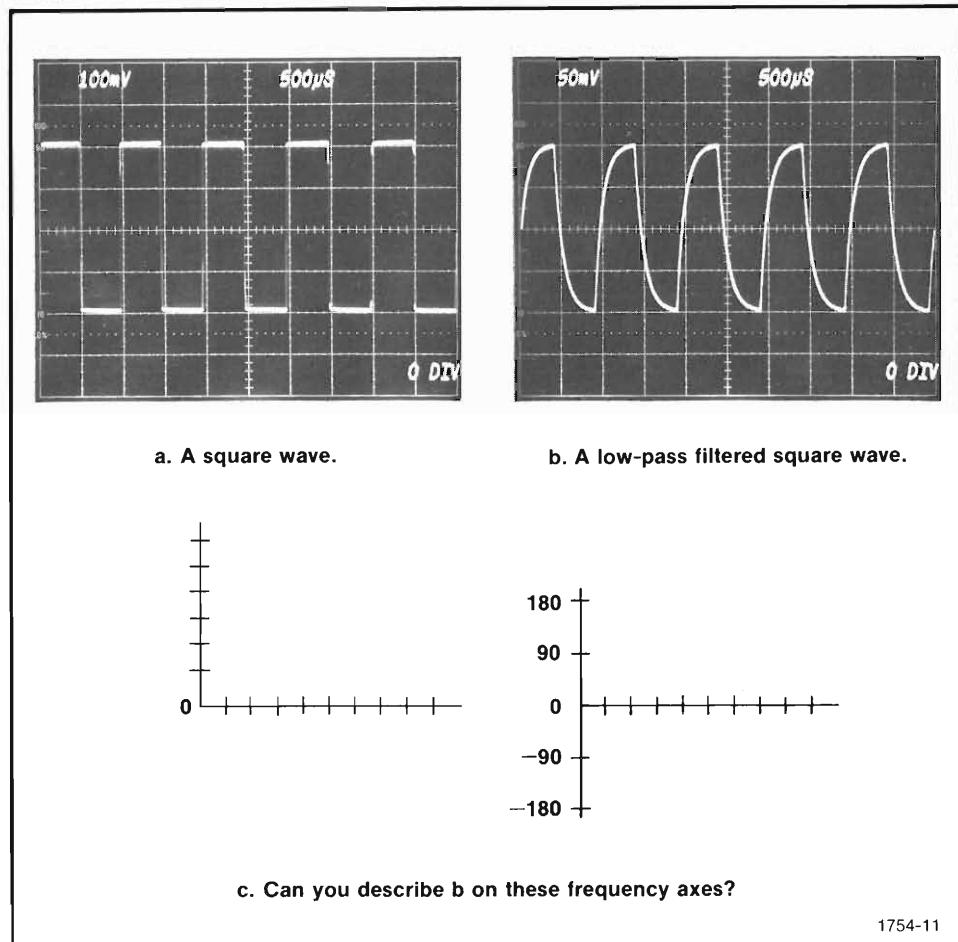


Fig. 1-10. The time domain does not tell the whole story.

1754-11

Something more than rules of thumb are needed to fill in Fig. 1-10c. If the waveform in Fig. 1-10b is available for measurement, it can be applied to a frequency-based oscilloscope. More often referred to as a spectrum analyzer, a frequency-based oscilloscope uses a filtering system to decompose a time-domain waveform into a magnitude diagram of its frequency components. A spectrum analyzer does not, however, provide phase information. For many analyses, just knowing the frequencies and their magnitudes is sufficient, but phase information can be just as important in many other cases, as was shown by Fig. 1-9.

Another and more complete approach is to apply the rigorous mathematical technique known as Fourier analysis. This allows you to describe a time-domain

The FFT

waveform in terms of frequency domain magnitude and phase. Or, if you prefer, Fourier analysis also gives results in the form of real and imaginary parts of the complex frequency domain. Unfortunately, the classic mathematical approach is frustrating for all but the simplest waveforms. If the waveform cannot be mathematically formulated, then classic Fourier techniques cannot be applied. A waveform can, however, be sampled and digitized. Then the discrete Fourier transform (DFT) can be used to Fourier transform the sampled and digitized waveform.

Since its general introduction in 1965, the fast Fourier transform (FFT) has been the commonly used algorithm for evaluating the DFT. This algorithm can be implemented through either computer software or hardware. Its major advantage is the speed with which it analyzes large numbers of samples. And, combined with standard measurement concepts, the FFT effectively converts a time-based instrument to a frequency-based instrument.

Parts II and III of this manual are devoted to digital techniques of Fourier analysis with the FFT algorithm. But before this subject can be approached, the continuous Fourier theory that it emulates must be understood. This Fourier theory for continuous, unsampled waveforms, embodied in the Fourier series and Fourier integral, is the subject of the next section.

SECTION 2

FOURIER TECHNIQUES DESCRIBE THE FREQUENCY DOMAIN

Born the son of an Auxerre tailor in 1768, Jean-Baptiste-Joseph Fourier grew to become one of France's major 19th-century administrators, historians, and mathematicians. His accomplishments began in 1798 when he went to Egypt with Napoleon. While there, he acted as an advisor on engineering and diplomatic matters and served as secretary of the *Institut d'Egypte* (Cairo). Also, he undertook an intensive study of Egyptian antiquities.

Fourier returned to France in 1801 and was appointed Prefect of the *Isère département* in 1802. He served at Grenoble in this capacity until 1814. During this time, he was recognized as an able administrator, and Napoleon granted him the title of Baron in 1809. This period also marked the beginning of Fourier's most important scientific contributions.

Fourier contributed heavily to compiling the *Description de l'Egypte*, which covered the cultural and scientific results of Napoleon's invasion of Egypt. This work, issued in 21 volumes over the period from 1808 to 1825, contained much of the information Fourier had gained from studying Egyptian antiquities. The attention these volumes drew to the ancient Egyptian civilization resulted in Egyptology being recognized as a new and separate discipline.

Between his administrative functions and his contributions to Egyptology, it is amazing that Fourier still found time to do pioneering work in mathematical physics. His interest in heat conduction led him to begin work in 1807 on *Théorie analytique de la chaleur* (English translation, 1878—The Analytical Theory of Heat). This initial work was completed in Paris and published in 1822; it shows how a mathematical series of sine and cosine terms can be used to analyze heat conduction in solid bodies. The series that Fourier proposed, and which bears his name, is of the form $y = \frac{1}{2} a_0 + (a_1 \cos x + b_1 \sin x) + (a_2 \cos 2x + b_2 \sin 2x) \dots$ This Fourier series was probably the first systematic application of a trigonometric series to a problem solution. Fourier spent the rest of his life working on his concept and expanded it to include the Fourier integral before his death in 1830. Both the Fourier series and the Fourier integral allow transformation of physically realizable time-domain waveforms to the frequency domain and vice versa. They are the mathematical tools for what is currently referred to as Fourier analysis.

Today, application areas of the Fourier series and integral transcend the original heat-conduction application. For example, a few of the many areas that benefit from Fourier analysis are linear systems, antennas, mechanical vibration, optics, biomedicine, and various random processes and boundary-value problems.

The FFT

It is interesting that such a far-reaching technique did not gain acceptability during Fourier's time. Various infinite series had been used prior to Fourier, most notably by Leonard Euler who found them an acceptable analysis tool. Many mathematicians, however, distrusted the use of series, and one influential mathematician wrote in 1828: "Divergent series are the invention of the devil, and it is shameful to base on them any demonstration whatsoever." Although Fourier's series is generally convergent, its validity did not escape the questioning of that era. Later work by P.G.L. Dirichlet (1805-1859), Bernhard Riemann (1826-1866), Henri Lebesgue (1875-1941), and others finally resolved any doubts about the validity of the Fourier series and the integral. For our part, let's accept the conclusions of these great mathematicians and simply go on to understand the fundamentals and concepts involved in Fourier analysis.

FOURIER SERIES GIVES SPECTRA FOR PERIODIC WAVEFORMS

If a Fourier series can be written for a waveform, then the components of the series completely describe the frequency content of the waveform.

The Question of Existence

The first condition that must be met for constructing a Fourier series is that the waveform be periodic. Precisely speaking, if the waveform is represented by $x(t)$ and a constant time, T , and it exists such that $x(t) = x(t + T)$ holds for all time, t , then $x(t)$ is periodic with a period of T . In short, the waveform must repeat itself in time. Familiar examples of this include sine waves, cosine waves, square waves, etc. The most important thing to remember here, and for later discussions, is that periodicity is for all time. That is, the waveform must begin at minus infinity and repeat itself out to plus infinity.

The condition of periodicity is rarely, if ever, met in the physical world! No oscillators or pendulums were in existence at time equal to minus infinity, and if they were, it is doubtful their operation would continue until plus infinity. However, for the sake of practicality, the rules can be bent a little. Periodicity, for the purpose of writing a Fourier series, can be defined over an observable interval. In other words, a square wave generator can be considered to have a periodic output from the time it is turned on to the time it is turned off. The Fourier series that is written, however, describes the square wave as though it started at minus infinity and continued to plus infinity.

The remaining conditions for existence of a Fourier series are referred to as the Dirichlet conditions. Briefly, these require that:

1. If the function has discontinuities, their number must be finite in any period.
2. The function must contain a finite number of maxima and minima during any period.

The FFT

3. The function must be absolutely integrable in any period; that is,

$$\int_0^T |x(t)| dt < \infty,$$

where $x(t)$ describes the function.

These conditions, along with that of periodicity, establish the existence or nonexistence of a Fourier series for any $x(t)$.

There are some functions for which a Fourier series does not exist. For all practical purposes, though, the Dirichlet conditions are met when a periodic function accurately specifies a physical occurrence. We can be confident that no oscillator exists that does not have a frequency spectrum associated with its output.

Fourier Series Gives Discrete Spectra

If we look at the square wave in Fig. 2-1a, we can quickly satisfy ourselves that it is periodic. We can also see in b, c, and d that it also meets the Dirichlet conditions. Thus, the Fourier series for this waveform exists.

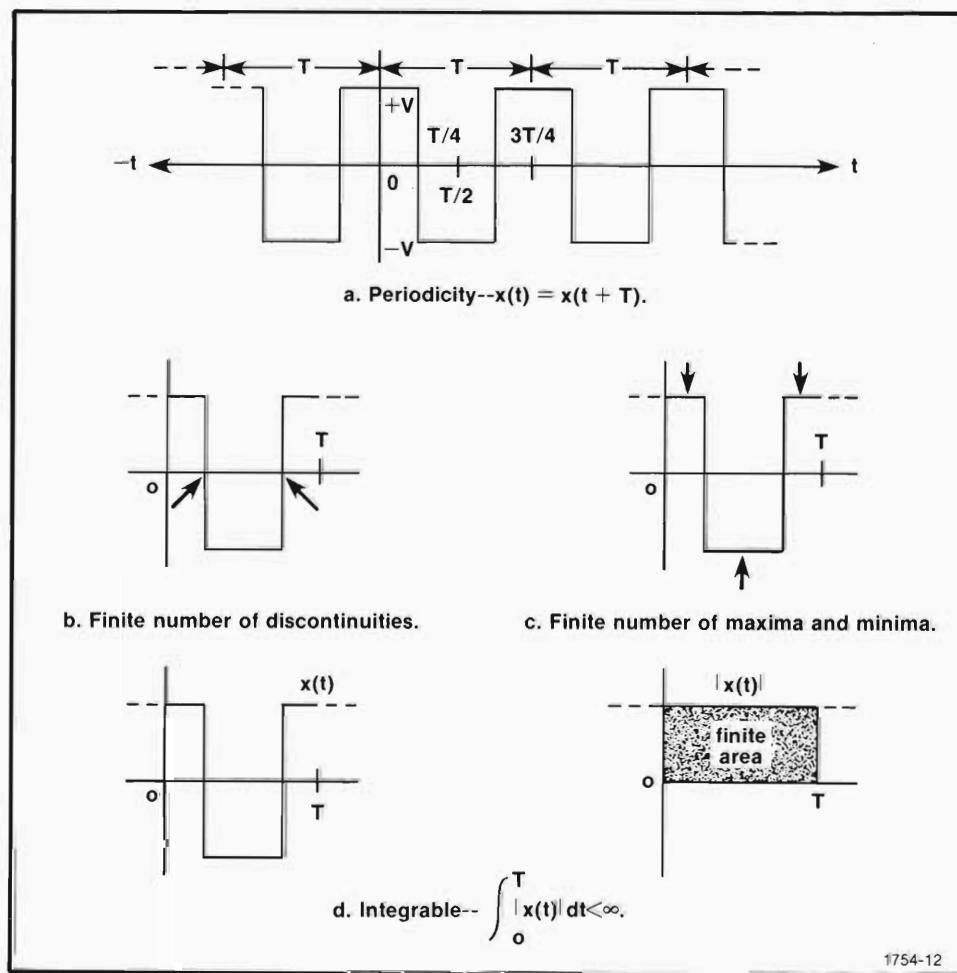


Fig. 2-1. A square wave meets the conditions for the existence of its Fourier series.

The FFT

With the question of existence out of the way, the Fourier series for a square wave can be written. It is not within the scope of this work to fully explore the writing of a Fourier series. That subject is amply covered in most complete textbooks on network analysis. Only a brief synopsis is given here for reference and as a matter of definition. The greater interest is in the laying of a foundation for building up to the concepts of the discrete Fourier transform (DFT) and fast Fourier transform algorithm (FFT).

The Fourier Series. The general form of the Fourier series is $x(t) = a_0 + a_1 \cos \omega_0 t + a_2 \cos 2\omega_0 t + \dots + a_n \cos n\omega_0 t + \dots + b_1 \sin \omega_0 t + b_2 \sin 2\omega_0 t + \dots + b_n \sin n\omega_0 t + \dots$, where $\omega_0 = 2\pi f_0$.

$$[Also frequently written as x(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega_0 t + b_n \sin n\omega_0 t)]$$

The Fourier series for a specific waveform is written by using salient features of the waveform to find specific values for the coefficients in the above series. First of all, ω_0 is taken from the period of $x(t)$ and is equal to $2\pi/T$ (also, $f_0 = 1/T$). The a_0 coefficient is the dc term and is equal to the average value of $x(t)$ over one period. This is determined by

$$a_0 = \frac{1}{T} \int_0^T x(t) dt$$

The remaining coefficients, a_n and b_n , are evaluated for $n = 1, 2, 3, \dots$ by

$$a_n = \frac{2}{T} \int_0^T x(t) \cos n\omega_0 t dt$$

and

$$b_n = \frac{2}{T} \int_0^T x(t) \sin n\omega_0 t dt.$$

For the particular case of the square wave in Fig. 2-1a, the Fourier coefficients evaluate to give the following series. In this series, $4/\pi$ is a constant resulting from the integration, and V is the peak voltage of the square wave. Also, notice that this series contains only cosine terms. This is because of the square wave's symmetric arrangement about time zero. (More on symmetry later.)

$$x(t) = \frac{4V}{\pi} (\cos \omega_0 t - \frac{1}{3} \cos 3\omega_0 t + \frac{1}{5} \cos 5\omega_0 t - \frac{1}{7} \cos 7\omega_0 t + \dots),$$

where $\omega_0 = 2\pi f_0$.

This series is a complete description of the frequency content of the square wave. From it, diagrams of both the magnitude spectrum and phase spectrum can be constructed according to the conventions discussed in Section 1. These diagrams are shown in Fig. 2-2.

The FFT

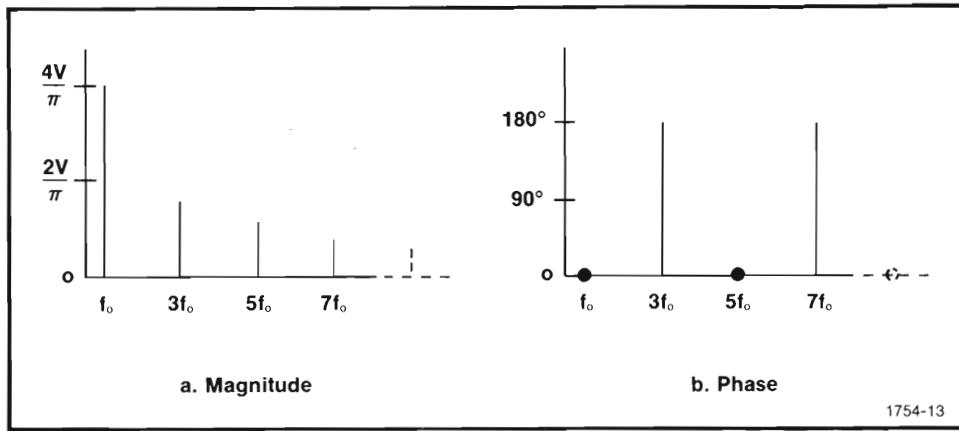


Fig. 2-2. Magnitude and phase spectra for the square wave in Fig. 2-1a.

Spectral Diagrams. For most people, Nature's own rainbow is the first experience with a spectral diagram. The rainbow, however, was never really thought of as a "spectrum" until Newton adapted that term in his 1672 paper to the Royal Society describing the continuous bands of color produced by passing light through a prism. Later, Joseph Fraunhofer (1787-1826) used diffraction gratings to study light spectra. What he saw were distinct lines instead of the continuous color spectra that Newton observed with his prism. And he discovered that the sun and stars have distinct line spectra associated with the light they emanate. Still later, in the mid-1800's, Gustav Kirchhoff and Robert W. Bunsen carried Fraunhofer's work further. (People in electronics will recognize Kirchhoff for his fundamental circuit laws. And Bunsen? Well, what chemistry or physics lab is complete without a Bunsen burner?) Kirchhoff and Bunsen found that each chemical element, when heated to incandescence, radiates its own distinct color of light. Hence, each chemical element is distinguishable by a unique line spectrum. Kirchhoff used these findings to analyze the chemical composition of various unknown substances. Thus, Bunsen and Kirchhoff launched the fundamentals of spectrum analysis.

Fourier analysis is spectrum analysis. Instead of light, though, Fourier analysis operates on wave shapes. And like light from heated compositions, different wave shapes have different spectra. For example, the square wave in Fig. 2-1a has the line spectra shown in Fig. 2-2. The square wave is made up of sinusoids having specific frequencies with specific amplitudes and phases. No other wave shape has the same component relationships!

The spectra in Fig. 2-2 are referred to as discrete or line spectra because each spectral component is discretely located on a frequency axis. Its location is indicated by a line. The length of the line indicates either magnitude or phase, depending upon which quantity is being considered.

Line spectra are characteristic of periodic waveforms. This is because periodic waveforms have discrete frequencies only. And when analyzed by the Fourier series,

The FFT

only those discrete frequencies that make up the periodic waveform appear in the Fourier series.

In the case of Fig. 2-2a, the line spectrum for magnitude is constructed from the Fourier series by first plotting the amplitude of the fundamental frequency. The fundamental frequency ($f_o = \omega_o/2\pi$) is the reciprocal of the waveform's period and is indicated in the Fourier series by ω_o . The magnitude of the fundamental is given by the first trigonometric term in the series ($n=1$). For the square wave example discussed thus far, the fundamental magnitude is $4V/\pi$. The diagram of Fig. 2-2a is constructed by placing the fundamental spectral line at f_o and giving it an amplitude of $4V/\pi$.

Subsequent Fourier terms are plotted in the same manner. Each term is some integer multiple of the fundamental frequency and is referred to as a harmonic. The fundamental is sometimes referred to as the first harmonic because f_o is multiplied by one, but integer multiples greater than one are always referred to as harmonics. In the case of Fig. 2-2a, the square wave is made up of odd harmonics. These are shown with spectral lines at $3f_o, 5f_o, 7f_o, \dots, nf_o$. The harmonic magnitudes are given by the Fourier coefficients in the series, and for the square wave, they are $1/3, 1/5, 1/7, \dots, 1/(2n-1)$ of the fundamental magnitude.

The line spectrum for phase is constructed in nearly the same manner as that for magnitude. There is a spectral line in the phase diagram for each component shown in the magnitude diagram, and these are placed on a frequency axis in the same manner. The difference is that the lengths of the lines in the phase spectrum indicate phase instead of magnitude. In Fig. 2-2b, the fundamental and the fifth harmonic are positive cosines and have zero phase. Their spectral line lengths are zero, so a heavy dot is used to indicate the presence of these zero-phase components. The third and seventh harmonics are negative cosines and therefore have phases of 180° . This is indicated by the lengths of the spectral lines at $3f_o$ and $7f_o$.

Gibb's Phenomenon. If you plot each cosine component from Fig. 2-2 against time and add up the waveforms, the sum approximates the original square wave. This is shown in Fig. 2-3a and b.

The exact square wave is not regained in Fig. 2-3b for two reasons. First, the total number of frequency components are not added in. And second, something called Gibb's phenomenon is happening.

Let's look at the number of components used first. Only those components indicated in Fig. 2-2 and 2-3a are used to get Fig. 2-3b. The Fourier series, however, specifies that an ideal square wave contains odd harmonics out to infinity. Since it is impossible to reconstruct an ideal waveform by physically adding an infinite number of components, a lesser number is used in Fig. 2-3. This should not be too disconcerting, though. The idea of cutting off or truncating a series is really not foreign at all. Few would dispute 0.3333 as a legitimate approximation of $1/3$. But we all know that the decimal equivalent of $1/3$ is really an "infinite series" of threes.

The FFT

If more accuracy is desired in representing $1/3$, 0.3333333333 can be used. For even more accuracy, even more threes can be added. This same idea is true for the Fourier series. For more accuracy, more Fourier terms are used. This is shown in Fig. 2-3c, where the first 20 odd harmonics are summed to get more accuracy in representing the ideal square wave. But even at this, the deviation from ideal is obvious.

In short, the original waveform can never be regained exactly unless you add in all of the terms from its Fourier series. And even at that, some waveforms still cannot be regained exactly. Those that cannot be regained exactly are the types that contain instantaneous transitions (discontinuities).

When discontinuities exist in the original waveform, adding up its Fourier terms does provide the exact original at every point except the discontinuity. At the discontinuity there will always be an overshoot. This overshoot is referred to as Gibb's phenomenon, and is always equal to 8.95% of the discontinuity amplitude.

Gibb's phenomenon and the effects of truncating the Fourier series are quite apparent in the square wave examples of Fig. 2-3. In both cases, the maximum

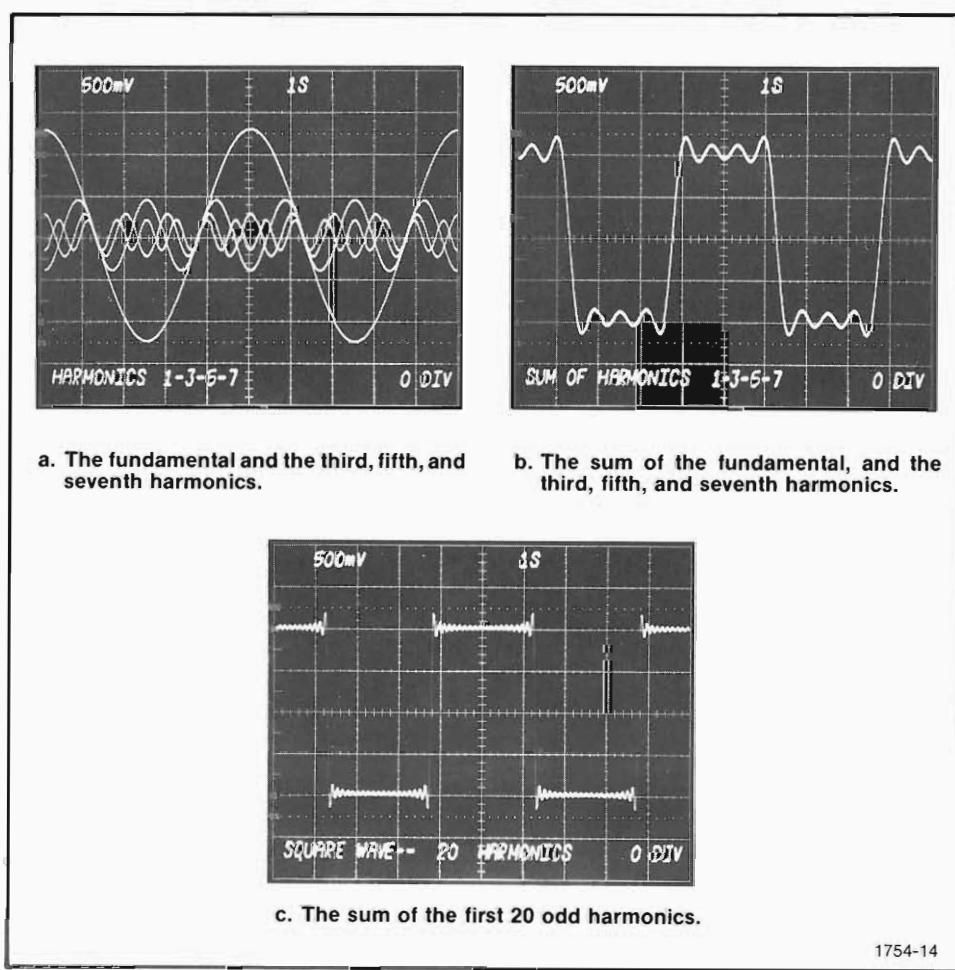


Fig. 2-3. Truncated Fourier series for a square wave: Truncating an infinite series results in some error in representation.

The FFT

overshoot is constant at 8.95% of the discontinuity. The ringing that follows, however, increases in frequency and decays quicker as more series terms are added. If all the terms are added in, the picture looks like Fig. 2-4, the ringing disappears, but Gibb's phenomenon still occurs.

Like using 0.333 to represent 1/3, the key thing here is recognizing that anything less than the full series is an approximation. If you know what to expect from truncating a Fourier series, or at discontinuities, then you can act accordingly.

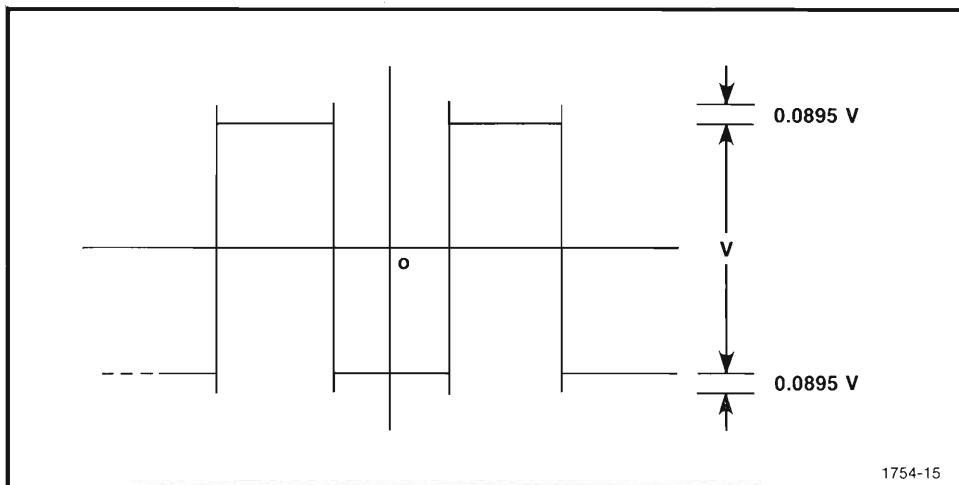


Fig. 2-4. Gibb's phenomenon in the limit. Overshoot is 8.95% of the discontinuity and never disappears.

Using the Fourier Series—Matters of Practicality

So far, a square wave is the only waveform that has been discussed. Its Fourier series has been given, and if the period of any square wave of interest is known, this can be used with the series to determine its exact spectral components. It's simply a matter of taking the reciprocal of the period to get the fundamental, f_0 . Then this is used in the series to find the frequency and magnitude of each component.

But the world is not made up exclusively of square waves. What about other waveforms?

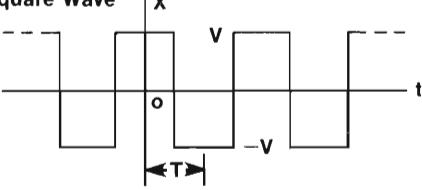
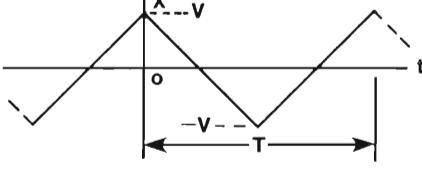
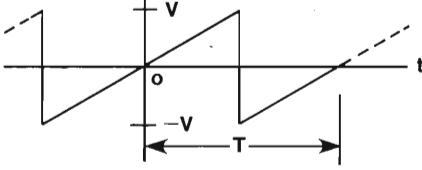
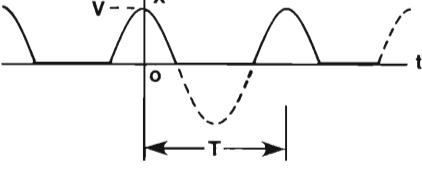
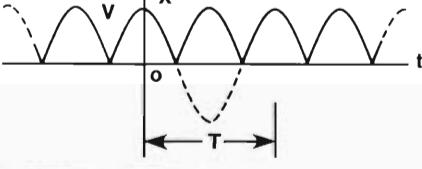
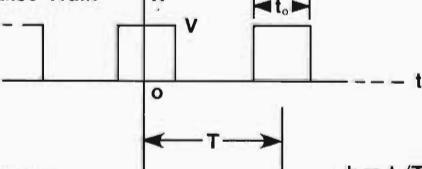
As long as you can mathematically describe a periodic waveform as a function of time, $x(t)$, that meets the Dirichlet conditions, its Fourier series can be written. However, this may not always be easy in practice. Fortunately, many textbooks contain Fourier series for most common wave shapes, and much work and agony can be saved by referring to them.

For your convenience, some common waveforms and their Fourier series are given in Table 2-1. Real-life waveforms rarely fit these tabulated wave shapes exactly.

The FFT

Square waves with zero rise time cannot realistically be generated, and some nonsymmetries and distortions from ideal invariably occur. But, if the actual waveform approximates the ideal fairly closely, the Fourier series for the ideal waveform can give some good estimates of the frequency spectrum.

TABLE 2-1
Some common Waveforms and Their Fourier Series

Wave Shape	Fourier Series -- $\omega_0 = 2\pi/T$
Square Wave 	$x(t) = \frac{4V}{\pi} (\cos \omega_0 t - \frac{1}{3} \cos 3\omega_0 t + \frac{1}{5} \cos 5\omega_0 t - \frac{1}{7} \cos 7\omega_0 t + \dots)$
Triangular Wave 	$x(t) = \frac{8V}{\pi^2} (\cos \omega_0 t + \frac{1}{9} \cos 3\omega_0 t + \frac{1}{25} \cos 5\omega_0 t + \dots)$
Sawtooth Wave 	$x(t) = \frac{2V}{\pi} (\sin \omega_0 t - \frac{1}{2} \sin 2\omega_0 t + \frac{1}{3} \sin 3\omega_0 t - \frac{1}{4} \sin 4\omega_0 t + \dots)$
Half-Wave Rectifier 	$x(t) = \frac{V}{\pi} (1 + \frac{\pi}{2} \cos \omega_0 t + \frac{2}{3} \cos 2\omega_0 t - \frac{2}{15} \cos 4\omega_0 t + \frac{2}{35} \cos 6\omega_0 t - \dots (-1)^{n/2+1} \frac{2}{n^2-1} \cos n\omega_0 t \dots)$ <p style="text-align: center;">n even</p>
Full-Wave Rectifier 	$x(t) = \frac{2V}{\pi} (1 + \frac{2}{3} \cos 2\omega_0 t - \frac{2}{15} \cos 4\omega_0 t + \frac{2}{35} \cos 6\omega_0 t - \dots (-1)^{n/2+1} \frac{2}{n^2-1} \cos n\omega_0 t \dots)$ <p style="text-align: center;">n even</p>
Pulse Train 	$x(t) = V[k + \frac{2}{\pi} (\sin k\pi \cos \omega_0 t + \frac{1}{2} \sin 2k\pi \cos 2\omega_0 t + \dots + \frac{1}{n} \sin nk\pi \cos n\omega_0 t + \dots)]$ <p style="text-align: right;">$k = t_o/T$</p>

FOURIER INTEGRAL GIVES SPECTRA FOR NONPERIODIC WAVEFORMS

The Fourier series is a useful tool for investigating the spectrum of a periodic waveform, but the world is not made up exclusively of periodic waveforms. What about nonperiodic waveforms—waveforms that don't repeat themselves in a regular fashion? Surely they too must have a frequency spectrum. After all, a bolt of lightning is nonperiodic, and certainly you have heard the familiar splatter of its spectrum on a common radio receiver.

Indeed, nonperiodic waveforms do have various frequency components. And the Fourier integral is the tool used to investigate the frequency spectra of nonperiodic waveforms.

The Fourier Integral is Related to the Fourier Series

The Fourier series and the Fourier integral, as analysis tools, are separate and distinct. One is intended for use with periodic waveforms and the other for use with nonperiodic waveforms. Thus, it is tempting to plunge directly into the Fourier integral. But this would be a disservice, for a subtle relationship exists between the two that is useful in interpreting later analysis concepts.

So, let's not leave the Fourier series just yet. Instead, let's draw the integral out of the series by considering a periodic waveform whose period is allowed to approach infinity. Though this development may not be generally considered rigorous, it is certainly enlightening. And enlightenment is the goal!

Getting the Integral out of the Series. This is going to take some math. But it's really not going to do any good to wander through a seemingly endless string of equations. Actually, there are only four major steps or equations that you need to be aware of. So those are clearly numbered below, and if you're pressed for time, you can just look at these numbered equations before going on. However, if you're a little curious about the transition between these steps, some of that information is given also. The full details are not given, but if you're interested in the complete exercise, it is included in many text books—*Network Analysis* by M.E. Van Valkenburg (2nd. ed., Prentice-Hall, Inc.) and *Basic Network Theory* by Paul M. Chirlian (McGraw-Hill Book Company), to name two good examples.

To begin, let's restate the general form of the Fourier series. This is

1

$$x(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos 2\pi n f_0 t + b_n \sin 2\pi n f_0 t).$$

The FFT

More appropriate to developing the integral, this should be reexpressed in exponential form. This is done by expressing $\cos 2\pi n f_0 t$ as

$$(e^{j2\pi n f_0 t} + e^{-j2\pi n f_0 t})/2 \text{ and } \sin 2\pi n f_0 t \text{ as } (e^{j2\pi n f_0 t} - e^{-j2\pi n f_0 t})/2j,$$

where e is the base of the natural logarithm and j is the imaginary unit of the complex number system ($j = \sqrt{-1}$). By some further manipulation and assignment of new variables, the more-compact, exponential form of the Fourier series is reached. This is

2

$$x(t) = \sum_{n=-\infty}^{\infty} c_n e^{j2\pi n f_0 t},$$

where c_n is evaluated for $n = -\infty, \dots, -2, -1, 0, 1, 2, \dots, \infty$ by

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} x(t) e^{-j2\pi n f_0 t} dt.$$

For each n , c_n is evaluated to give the magnitude and phase of the harmonic component of $x(t)$ having frequency $n f_0$.

With the Fourier series in exponential form, the next step is to recognize that each harmonic is separated by an amount $\Delta f = 1/T$. Now, with some further manipulations which again are omitted for the sake of getting on with the subject, the series can be placed into a form that lets us look at the limit as T goes to infinity. This form is

3

$$x(t) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{n=-\infty}^{\infty} X(n f_0) e^{j2\pi n f_0 t},$$

but since $\Delta f = 1/T$,

$$x(t) = \lim_{\Delta f \rightarrow 0} \sum_{n=-\infty}^{\infty} X(n f_0) e^{j2\pi n f_0 t} \Delta f.$$

Now, as Δf goes to zero (period, T , goes to infinity), the properties of the summation approach those of an integral.

All of this simply says that when the period (T) goes to infinity, the Fourier series reduces to

4

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi f t} df.$$

The "Fourier coefficients" here have become a function of a continuous frequency variable, f , and are given by

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt.$$

The FFT

Together, these two integrals are referred to as the Fourier transform pair. The former is generally referred to as the inverse Fourier transform and the latter as either the direct Fourier transform or simply the Fourier transform.

The Fourier Integral as a Transform. As implied by its development, the Fourier integral is only applicable to nonperiodic waveforms—waveforms with infinite periods. (An infinite period simply implies that the waveform does not repeat itself.)

A nonperiodic waveform, given by $x(t)$ and subject to the Dirichlet conditions, can be transformed to a function of frequency by using

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt.$$

When this is done, $X(f)$ is generally referred to as "the Fourier transform of $x(t)$." And in the same manner, using

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft} df,$$

the frequency-domain function, $X(f)$, can be "inverse transformed" back to the time-domain function, $x(t)$.

Dirichlet Conditions for Transform Existence

In order for $x(t)$ to be transformed by the Fourier integral, it must be nonperiodic. In addition to this, the following Dirichlet conditions must be met for existence of the transform.

1. For $-\infty \leq t \leq \infty$, $x(t)$ must contain a finite number of maxima and minima.
2. If $x(t)$ contains discontinuities, they must be finite in number over the range $-\infty \leq t \leq \infty$.
3. The function, $x(t)$, must be integrable in the sense that

$$\int_{-\infty}^{\infty} |x(t)| dt < \infty.$$

For all practical purposes, these conditions are met by any nonperiodic $x(t)$ that can be physically generated.

The FFT

The transform action of the Fourier integral, as compared to the Fourier series, is shown in Fig. 2-5. The major thing to notice there is that they both apply to different classes of waveforms. Also, notice that their frequency-domain or spectral descriptions are different. The Fourier series technique provides magnitudes and phases at specific, discrete frequencies. The Fourier integral, on the other hand, evaluates to a continuous function of frequency. To look into this a little further, let's go back to the Fourier series again and let the period go to infinity. But this time, let's do it with just pictures—no mathematics.

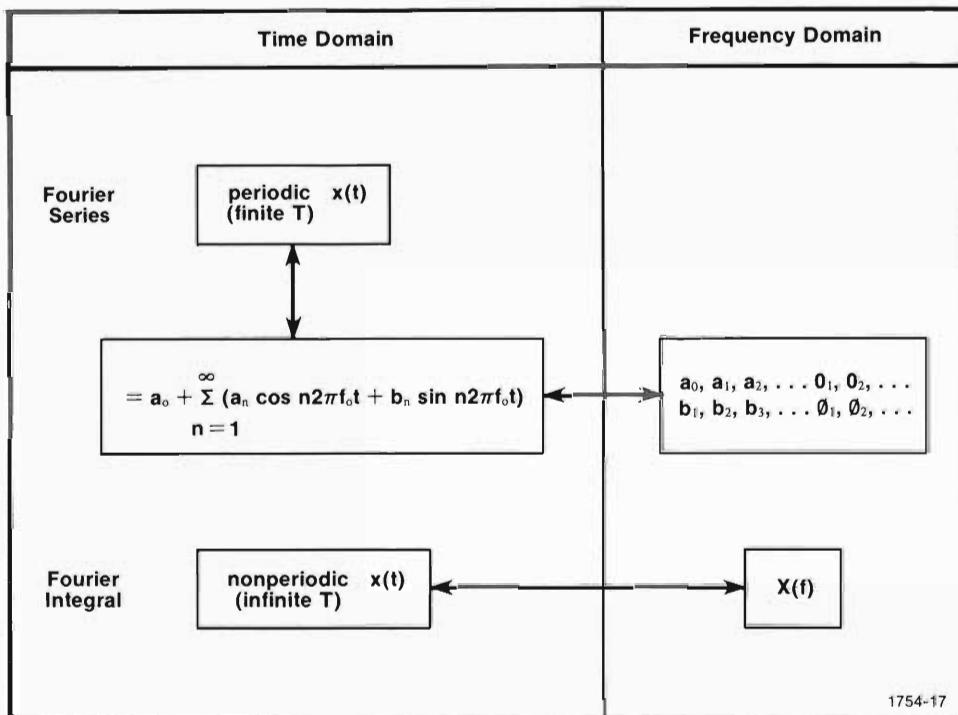


Fig. 2-5. The Fourier series and the Fourier integral—two different paths between time and frequency.

Infinite T Causes a Continuous Spectrum. To see how the Fourier integral arrives at a continuous frequency spectrum, let's start with the Fourier series and a periodic waveform. In particular, let's use a train of square pulses arranged so that pulse width is exactly one half the period. This is shown in Fig. 2-6a. The magnitude spectrum for this pulse train, taken from its Fourier series, is also shown in Fig. 2-6a. Except for a dc component, this magnitude spectrum is the same as those shown for previous square wave examples.

Next, in Fig. 2-6b, the period of the pulse train is doubled while pulse width is held constant. The effect on the magnitude spectrum is two additional frequencies around each of the original components from Fig. 2-6a (excepting the dc component). Also, notice that the amplitude of each component in Fig. 2-6b is decreased from Fig. 2-6a. This comes from a reduced duty factor (pulse width over period), which causes a reduction of average waveform energy over the period. Since the average waveform

The FFT

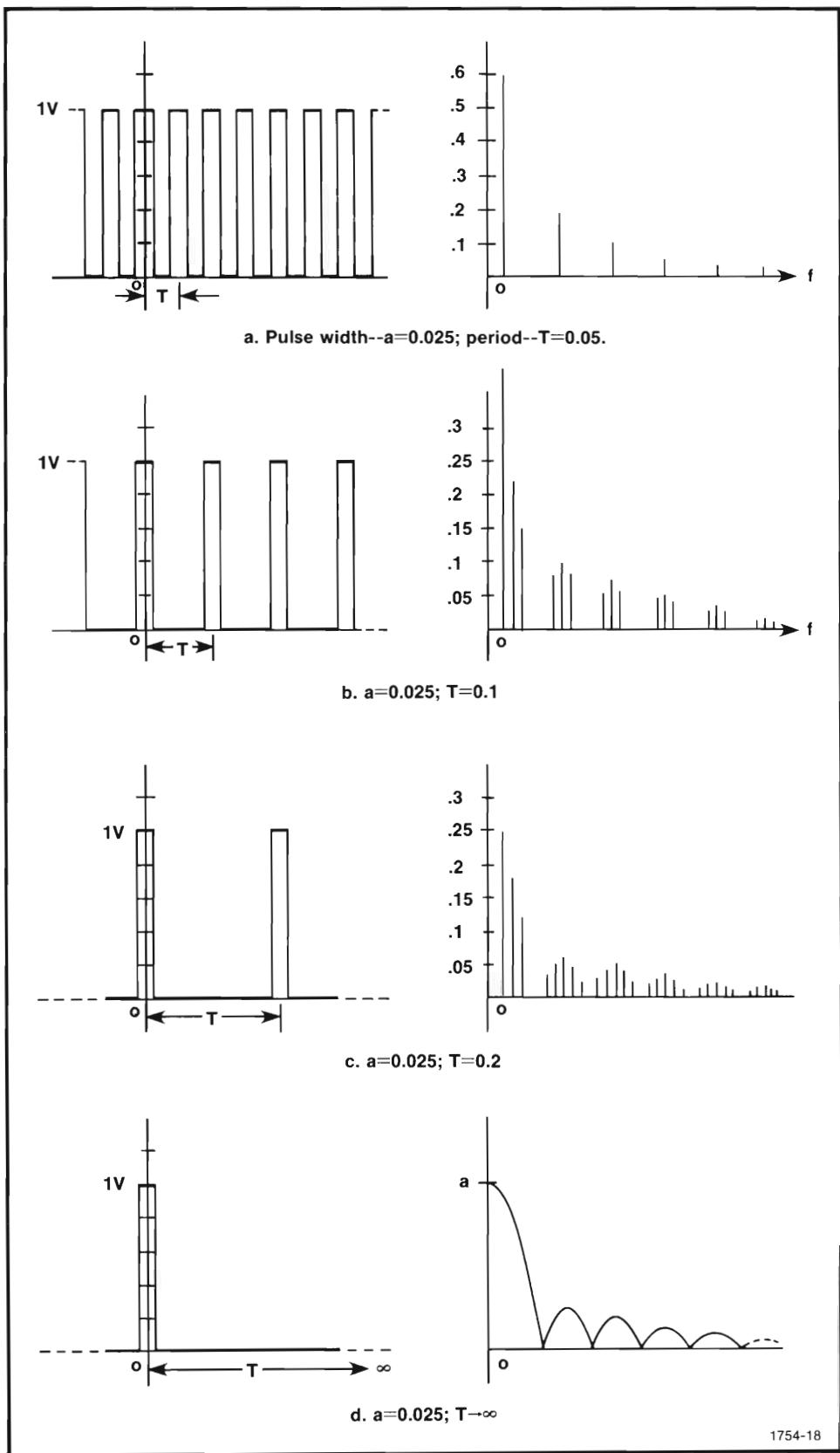


Fig. 2-6. As the period, T , goes to infinity, the discrete spectral lines become closer. When T becomes infinity, the spacing between lines is zero and a continuous spectrum exists.

The FFT

energy is reduced, it is natural that a reduction be seen in the magnitude spectrum: The frequency components needn't contribute as much to the total waveform.

In Fig. 2-6c, the period is increased again and the pulse width still held constant. Do you see what's happening to the magnitude spectrum?

More components are packed into the magnitude spectrum, and their amplitudes are decreased. Further increases in the period cause even more spectral components to occur with even closer spacing. As the period goes to infinity (Fig. 2-6d), the spacing between components goes to zero. In other words, the series converges to the Fourier integral. The resulting magnitude spectrum is called a continuous spectrum because it is defined at every frequency—there is zero spacing between frequency components. This implies that a single pulse is made up of an infinite number of sinusoidal components.

A similar demonstration for the phase spectrum is trivial in the case of Fig. 2-6 since those pulse arrangements all have zero phase. However, the same idea applies, and phase obtained by the Fourier transform is a continuous function of frequency, too.

Understanding Frequency-Domain Diagrams

Frequency-domain diagrams are the key to understanding Fourier analysis. They are the means for exposing the subtle nuances of the mathematics; without them, analysis is reduced to dreary comparisons of formulas and numbers.

Frequencies can be Negative or Positive. People are pretty comfortable in the time domain, so the idea of negative time isn't too unsettling. It's pretty easy to picture yesterday as negative time, right now as time zero, and tomorrow as positive time. Our language is very generous in supplying us with terms for supporting this concept, too. Then, now, later; past, present, future—these are all familiar to us and picturable in a variety of ways.

But negative frequency? That's not quite as comfortable. There really aren't any other words for it, and "minus 40 Hz" doesn't conjure up any picture different than "plus 40 Hz." Nevertheless, the mathematics of the Fourier integral require introduction of negative frequencies. Look again at the transform pair.

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt \quad x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft} df$$

Both $X(f)$ and $x(t)$ are defined over frequencies and times from minus infinity to plus infinity.

Actually, the concept of negative frequency isn't any more difficult than that of negative time, as long as a few basic ideas are kept in mind. These ideas can be explored with the aid of Fig. 2-7.

The FFT

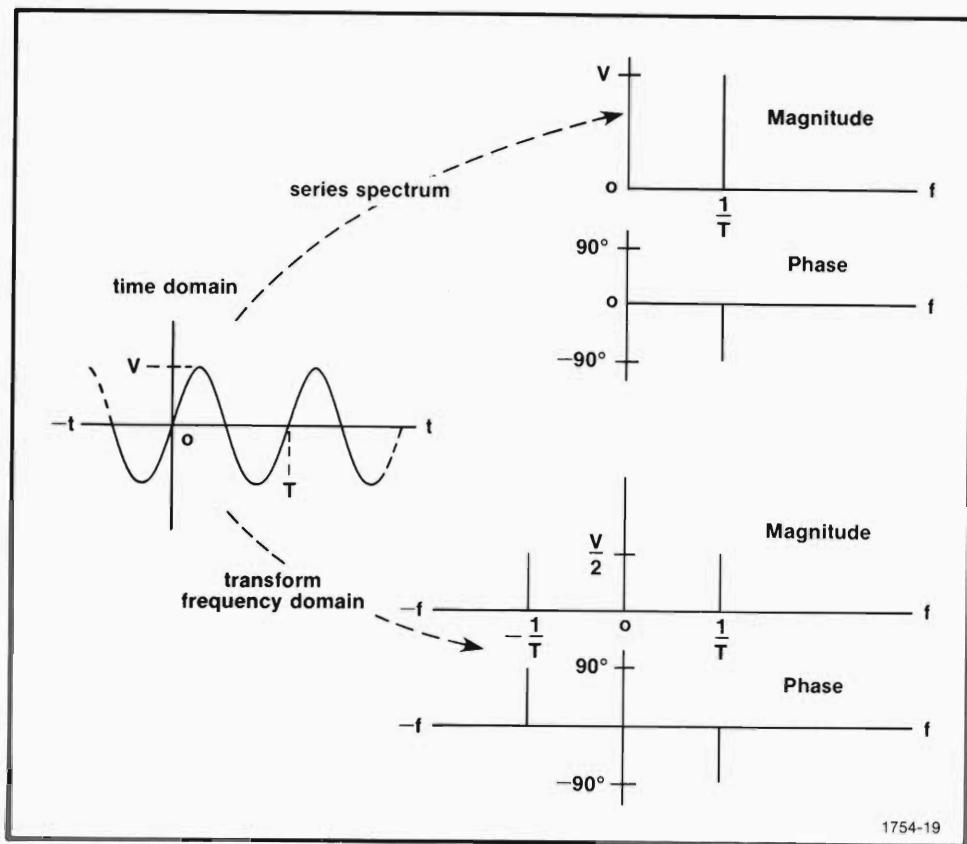


Fig. 2-7. Comparison of diagramming conventions in the series spectrum and the transform frequency domain.

Fig. 2-7 shows a sine wave that is periodic in the time domain from minus infinity to plus infinity. Also shown in Fig. 2-7 are several spectral diagrams for this sine wave.

Let's look at the uppermost pair of these diagrams. This pair contains the magnitude and phase information that would be obtained by writing the Fourier series for the sine wave. Notice that the amplitude, V , of the time-domain sine wave is reflected exactly in the magnitude spectrum. Also, the positioning of the time-domain sine wave, relative to time zero, indicates a delay of -90° . This same delay is also shown in the phase diagram. (Since it was covered thoroughly in Section 1, this format should be familiar to you.)

Now let's look at the lower pair of diagrams in Fig. 2-7. These, also, are spectral diagrams; however, they describe the sine wave in the frequency domain of the Fourier transform. In theory, a periodic waveform extending from minus infinity to plus infinity cannot be transformed to the frequency domain by the Fourier integral. For the sake of illustration, however, this restriction is ignored momentarily. The lower pair of diagrams show what the frequency domain would look like if an infinite extent sine wave could be Fourier transformed. If this breach of theory bothers you, think of the sine wave as being the only illustrated component of some arbitrary pulse. Then its transformation, as part of the pulse, is in concert with theory.

The FFT

Setting aside any remaining hesitancy about theory, go ahead and look closely at the lower pair of diagrams. Compare them with the diagrams of the series spectrum. The first difference you'll probably notice is the presence of negative frequencies in the frequency-domain diagrams. This should be expected since the transform integral defines both positive and negative frequencies.

Now focus your attention solely on the frequency-domain magnitude diagram. Notice that there are two spectral components there, one at the positive frequency of the sine wave and one at its negative frequency. Also, notice that each magnitude is one half that of the time-domain waveform (also, each one is half the magnitude of the one in the series spectrum). Since the spectrum is divided between positive and negative frequencies, doesn't it seem reasonable to divide the energy in the same manner?

Now let's look at the frequency-domain phase diagram. The phase for positive frequency duplicates the phase shown in the series spectrum; it is -90° at $1/T$. Phase in the negative frequency domain is simply an inverted image of the positive domain. For this example, it is $+90^\circ$ at $-1/T$. Are you wondering why phase isn't halved like magnitude? It's because phase is just a position indicator, not an energy indicator.

The frequency-domain diagrams in Fig. 2-7 embody most of the conventions of magnitude and phase description for the Fourier transform's frequency domain. In review, these conventions are:

1. **The magnitudes in the positive and negative frequency domain exactly duplicate each other.** Except for the dc component, the magnitudes are equally divided between the positive and negative frequency domain. For every frequency indicated in the positive frequency domain, one of equal magnitude is indicated at the same frequency in the negative domain. Their sum equals the amplitude of the corresponding sinusoidal component in the time domain. In the case of dc, its frequency is zero and there is no division of magnitude.

There is one qualification here that should be noted. This convention is true only for real-valued signals. In the case of complex signals, those of the form $x(t) = a(t) + jb(t)$, the negative frequency domain will not mirror the positive frequency domain. However, the large majority of signals encountered in a measurement situation are real. And then, the negative frequency domain does mirror the positive frequency domain.

2. **Phase in the positive frequency domain is duplicated in the negative frequency domain, except the images are inverted.** This simply entails a sign change when passing between positive and negative frequencies. The amount of phase in the time domain is reflected exactly in both the positive and negative frequencies; unlike magnitude, it is not halved.

Even with all of these conventions in mind, you may still be wondering: "Just exactly what does a negative-frequency sine wave look like?"

The FFT

Well, let's look at a picture of one. Fig. 2-8 shows a three-dimensional space that can be associated with the Fourier transform. This is an extension of the three-dimensional pictorial aid presented with Figs. 1-4 and 1-8 in Section 1. The difference in Fig. 2-8 is that negative frequencies are shown and frequency-domain conventions are used.

In keeping with frequency-domain conventions, two sinusoids are shown in Fig. 2-8. One is the negative frequency term of a time-domain component and the other is the positive frequency term.

Each of these sinusoids is of equal amplitude, and each passes through the frequency axis at points equal to plus and minus $1/T$. Their positive amplitude

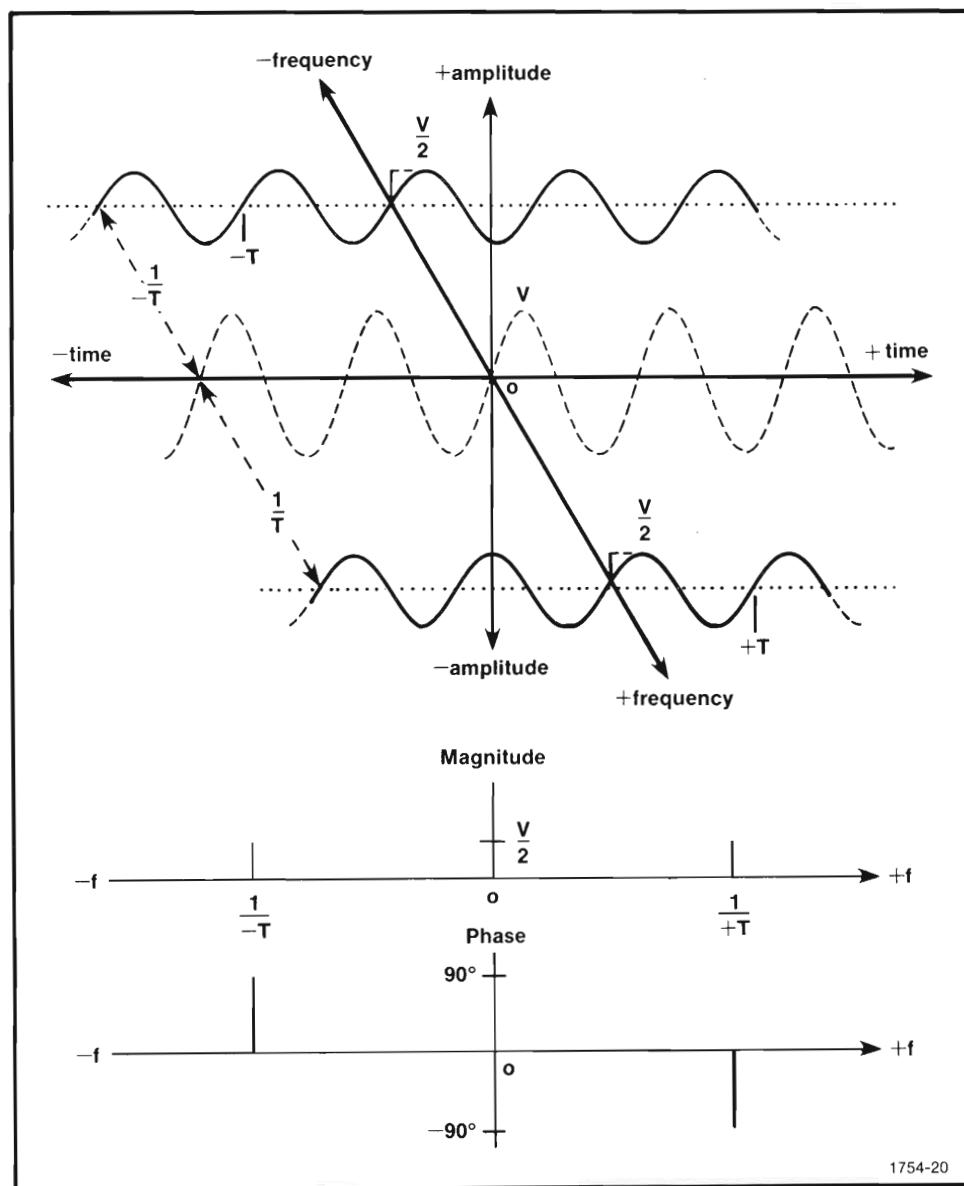


Fig. 2-8. A 3-D look at positive and negative frequencies.

The FFT

projection onto the frequency-amplitude plane at these two points forms the frequency-domain magnitude diagram. Also, their full projection onto the amplitude-time plane is summed to equal the time-domain component they represent (shown by a dashed line). In order for this to occur, both the positive and negative frequency components must have the same phase, and in Fig. 2-8, they do. The positive frequency term is arranged for -90° phase (a sine wave). The corresponding term in negative frequency has exactly the same arrangement with time; however, it is said to have a $+90^\circ$ phase. This sign change is in accordance with the conventions of frequency-domain description and is shown in the phase diagram at the bottom of Fig. 2-8.

So! A negative-frequency sine wave looks exactly like a positive-frequency sine wave! They're just located at different points on the frequency axis.

Now that some of the conventions for frequency-domain diagrams have been covered, let's go back to Fig. 2-6d and redraw it for the Fourier integral's frequency domain. This redrawn version is shown in Fig. 2-9. Notice that the time-domain pulse (Fig. 2-9a) is a more general version with an amplitude of "V" and a width of $2T_0$. Also, the pulse is centered about time zero; this is a zero phase positioning.

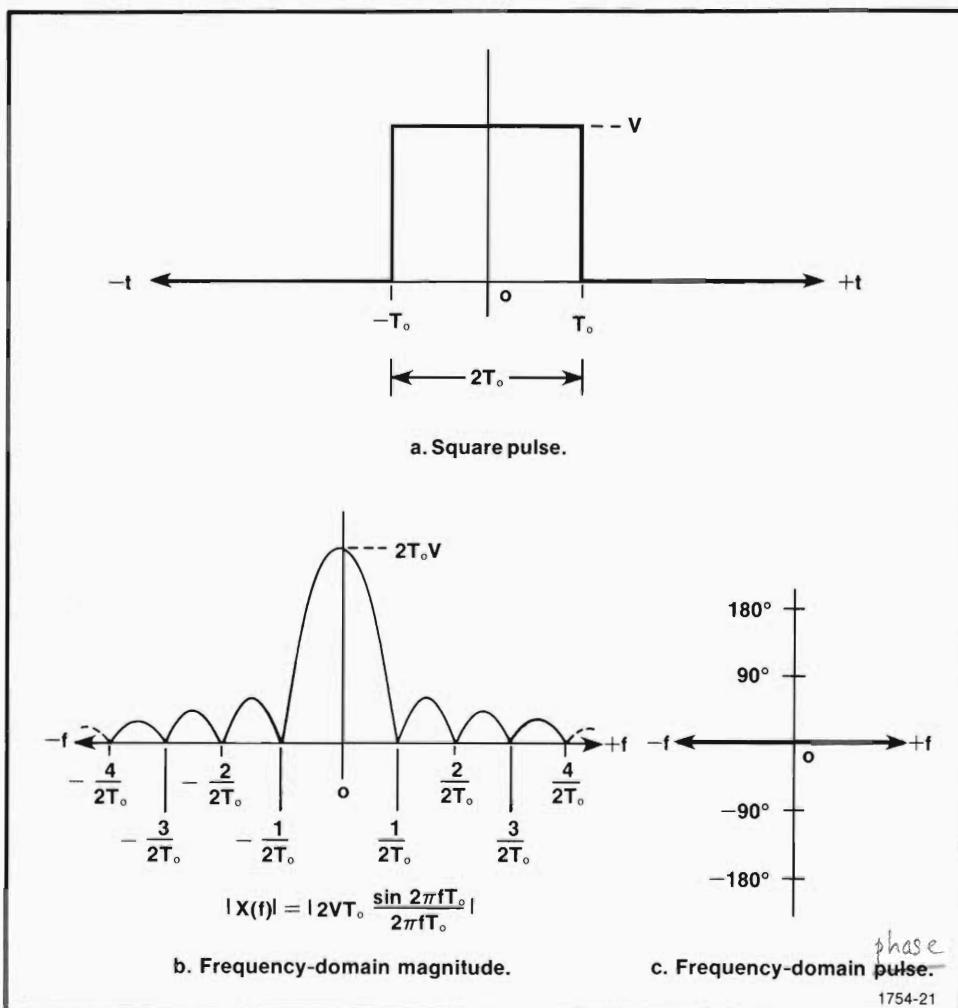


Fig. 2-9. A general square pulse and its frequency-domain magnitude and phase.

The FFT

The frequency-domain diagrams for the pulse are shown in Fig. 2-9b and c. In Fig. 2-9b, the magnitude is shown as a continuous function of frequency, and the positive-frequency magnitude is exactly mirrored by the negative frequencies. The phase is shown in Fig. 2-9c as continuously zero for the same range of frequencies. This zero phase indicates that the square pulse in Fig. 2-9a is made up entirely of cosine waves. Their magnitudes correspond to Fig. 2-9b. Every frequency of cosine wave is present in the pulse except where the magnitude is zero at $\pm 1/2T_0$, $\pm 2/2T_0$, $\pm 3/2T_0$, $\pm 4/2T_0$,...

As a final note, Fig. 2-9 fits any square pulse that is centered at time zero. Just substitute the amplitude of the pulse for V and the width for $2T_0$.

Frequency-Domain Descriptions Can Be in Rectangular Form. Up to this point, the frequency domain has been described only in terms of magnitude and phase. This is probably the best introductory approach since sinusoids and phase angles are generally familiar items. Also, introduction through magnitude and phase lends itself well to various graphical explanations and descriptions.

But, there is more than one way to look at the frequency domain. In fact, the actual computation involved in Fourier transformation doesn't necessarily lead directly to magnitude and phase results. It's often more convenient to obtain frequency-domain results in a "rectangular form" such that

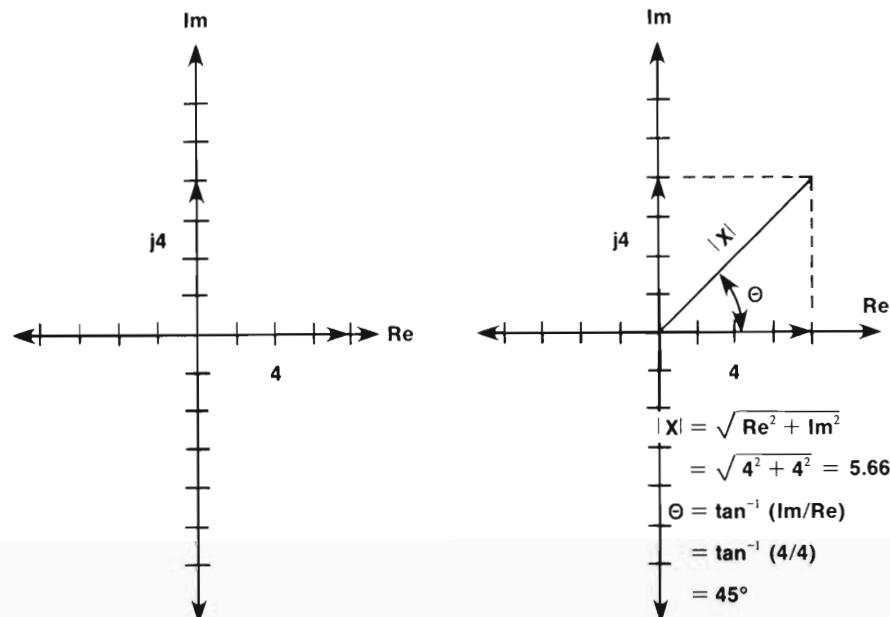
$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt = \text{Re}(f) + j\text{Im}(f).$$

In this relationship, the time-domain function being transformed is $x(t)$. The frequency-domain function for $x(t)$ is given by $\text{Re}(f) + j\text{Im}(f)$.

In the above results, $\text{Re}(f)$ is referred to as the "real part", and $\text{Im}(f)$ is referred to as the "imaginary part." Together, these two parts are referred to as a complex-valued function in rectangular form—complex because the function is of more than one part and rectangular because complex quantities are often portrayed as vectors in rectangular coordinates. This really is no different than magnitude and phase, which is a function with two parts—magnitude and phase—and is an equivalent system. Also, magnitude and phase can be portrayed as a rotating vector in polar coordinates. Thus, magnitude and phase results are often referred to as results in "polar form." The relationship between the rectangular and polar forms is shown by vectors in Fig. 2-10. If you are not at all excited by vectors (I'm not), you might prefer to look at Fig. 2-11. The same relationship is shown there through the time-domain and frequency-domain functions for a single square pulse.

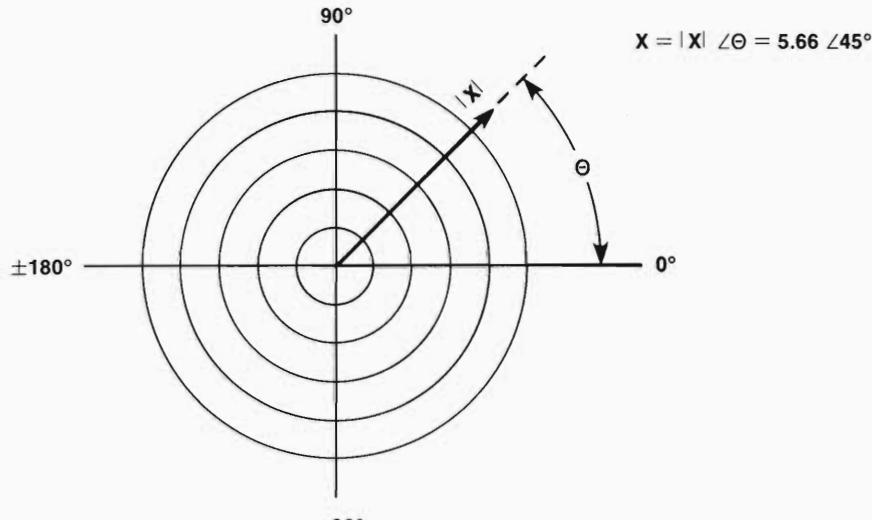
The FFT

a. Given a complex number: $x = Re + jIm = 4 + j4$.



b. X as vectors in rectangular coordinates.

c. Conversion to magnitude and phase.



d. Magnitude and phase in polar coordinates.

1754-22

Fig. 2-10. Complex number expressed in rectangular and polar form.

The FFT

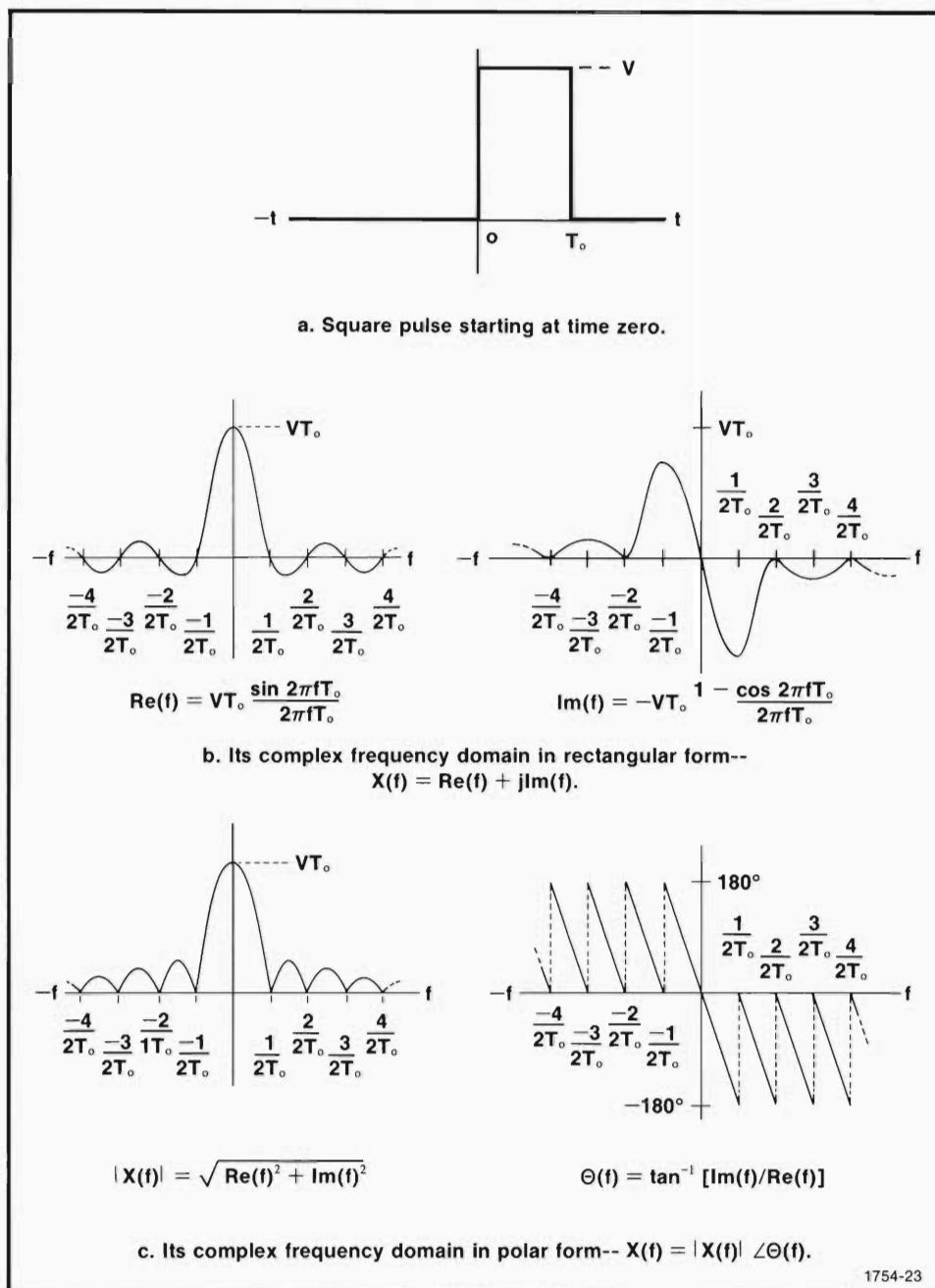


Fig. 2-11. A square pulse and its frequency domain functions.

The FFT

Functions are Even or Odd or the Sum of Even and Odd Functions

If you can identify even and odd time-domain functions, then you can predict some frequency-domain features just by looking at the waveform. For example—

Even Functions are the Sum of Cosines Only. Fig. 2-11 is useful for pointing out the relationship between the rectangular and polar form of the Fourier transform. The real interest, however, arises when this square pulse is compared to the one in Fig. 2-9. These two pulses really aren't too different in the time domain, but look at their frequency domains. One has zero phase and the other doesn't.

What's so different in the time domain that makes such a difference in the frequency domain?

Well, first of all, you might notice that the pulse in Fig. 2-9 has twice the width of that in Fig. 2-11.

Does this make the difference?

Not really.

Both of the square pulses are given in general enough form so that pulse width doesn't affect the shape of the frequency domain. In other words, we could shrink the pulse in Fig. 2-9 to a width of T_0 , and there would still be zero phase when it is transformed to the frequency domain.

So, if it isn't pulse width, then what makes the difference?

Compare the two pulses again. In particular, look at each pulse's location relative to time zero.

In Fig. 2-9, the square pulse starts at some negative time and ends at an equivalent positive time. The left half of the pulse is the same as the right half. And if you think about it, you can picture the two halves being folded about time zero to come together for an exact match. The pulse is symmetric about time zero. Mathematically, this is stated by the following equality.

$$x(t) = x(-t)$$

Any function, whether it is periodic or nonperiodic, that meets the $x(t) = x(-t)$ condition is said to be an *even function of time*. The same thing can be said for the frequency domain. A function that meets the condition of $X(f) = X(-f)$ is an *even function of frequency*.

Table 2-1, back in the discussion of the Fourier series, contains several examples of even functions of time. All of the waveforms there, except the sawtooth wave, are even functions of time. They meet the $x(t) = x(-t)$ requirements. If you look at the series for each of these even functions, you should notice something else they have in common. They are all composed of cosine terms. The same is true for the square pulse in Fig. 2-9.

The FFT

It is an even function and is composed entirely of cosine waves and therefore has zero phase.

All even functions of time are made up entirely of cosine waves and have zero phase. Or, if you prefer thinking in terms of the rectangular form, the Fourier transform of an even function of time gives a real and even function of frequency. The imaginary part of the transform is zero.

Odd Functions Have Sines Only. Now, look at the sawtooth wave in Table 2-1. It isn't an even function, but it does appear to have some symmetry. In fact, it meets the condition for an *odd function of time*, which is given by $x(t) = -x(-t)$. In terms of looking at the waveform, it is odd if the positive time portion can be sign reversed and folded about time zero for an exact match. This idea of folding and matching to determine evenness or oddness is shown in Fig. 2-12.

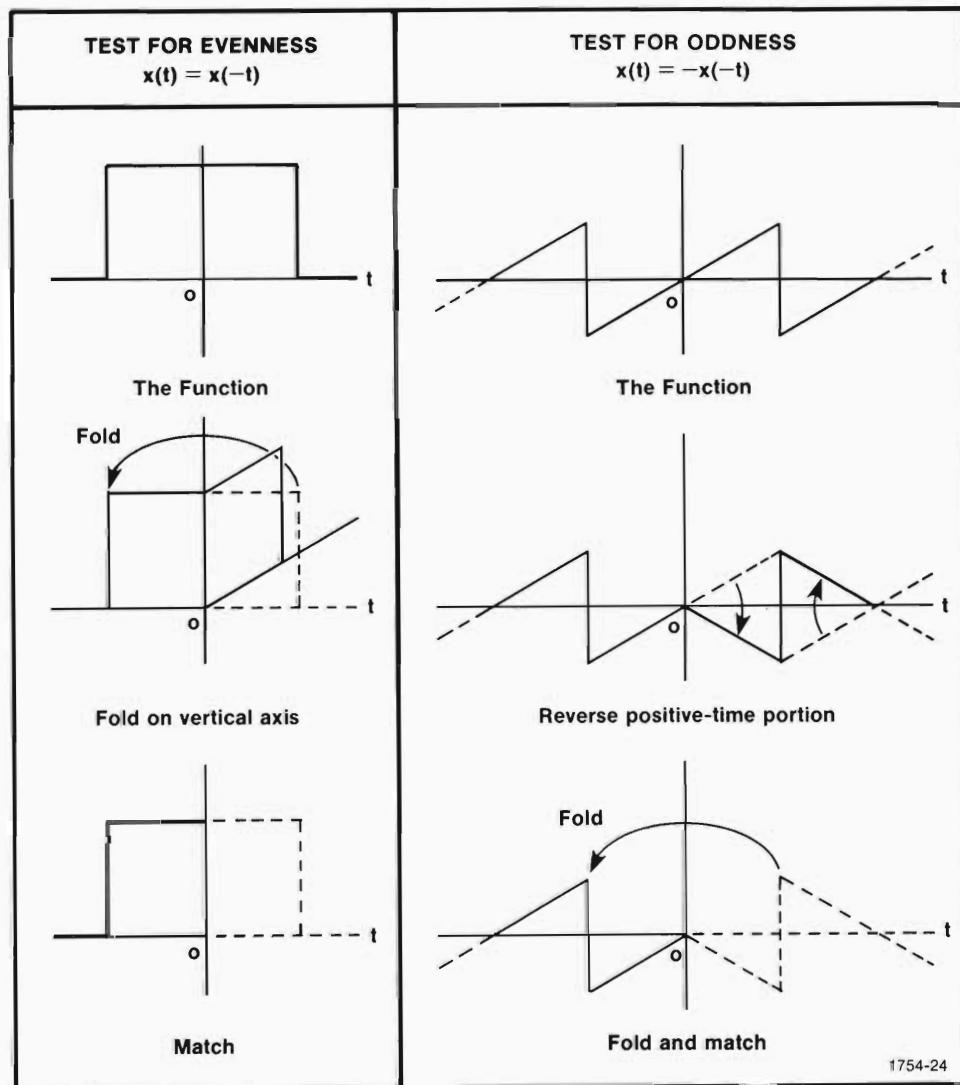


Fig. 2-12. Visual test for even or odd function.

The FFT

Returning to the sawtooth wave in Table 2-1, notice that its Fourier series is made up of sine terms only. This is characteristic of all odd functions of time. They are made up entirely of sine waves and have 90° phase. Or, in terms of the rectangular form, odd functions of time transform to odd and imaginary functions of frequency. The real part of the frequency domain for any odd function is always zero.

So far, we haven't looked at a pulse that is an odd function of time. So, let's look at one. Fig. 2-13a shows such a pulse, and you can prove its oddness for yourself by mentally going through the process shown in Fig. 2-12. The rectangular and polar forms

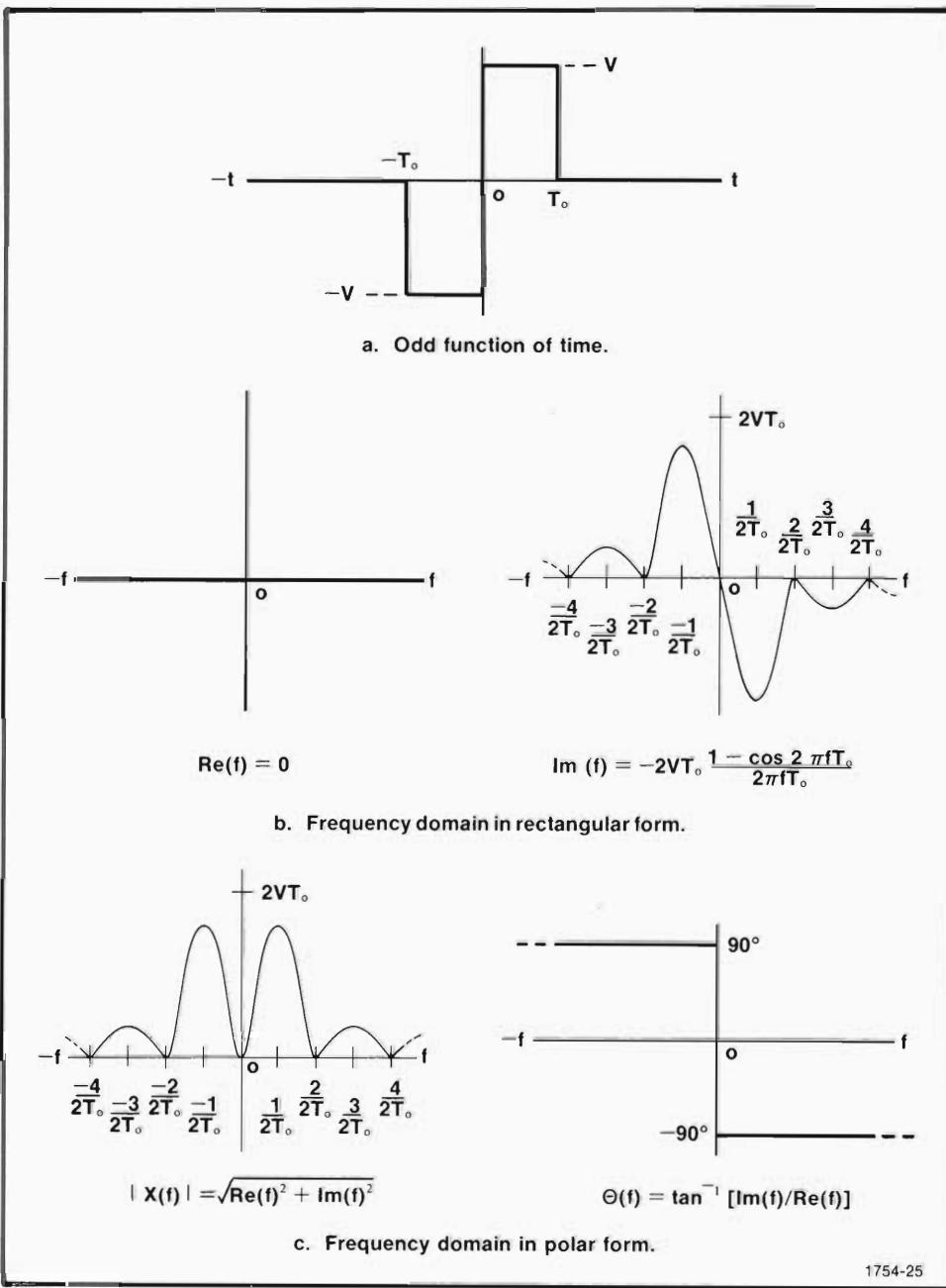


Fig. 2-13. Pulse that is an odd function and its Fourier transform to the frequency domain.

The FFT

of this pulse's frequency-domain functions are shown in Fig. 2-13b and c. Notice that the real part of the rectangular form is zero, as it should be. The imaginary part contains all of the frequency-amplitude information.

Look closely at the imaginary part in Fig. 2-13b. Is it even or odd?

Just like the time domain waveform it came from, the imaginary part is odd, also. This is another feature of oddness and evenness. Odd functions in one domain transform to odd, imaginary functions in the opposite domain, and even functions transform to even, real functions.

Some Functions are Part Even and Part Odd. Let's go back to the square pulse in Fig. 2-11. If you test this pulse for evenness and oddness, you'll find that it fits neither case. But, look at its frequency-domain in the rectangular form. Then look at the pulses in Fig. 2-9 and Fig. 2-13. Do you see how their frequency domains, except for a multiplying constant, might somehow be combined to equal the frequency domain of the pulse in Fig. 2-11?

In fact, the square pulse in Fig. 2-11 is neither even or odd, but is the sum of an even and an odd function. This summation is shown more clearly in Fig. 2-14. Notice

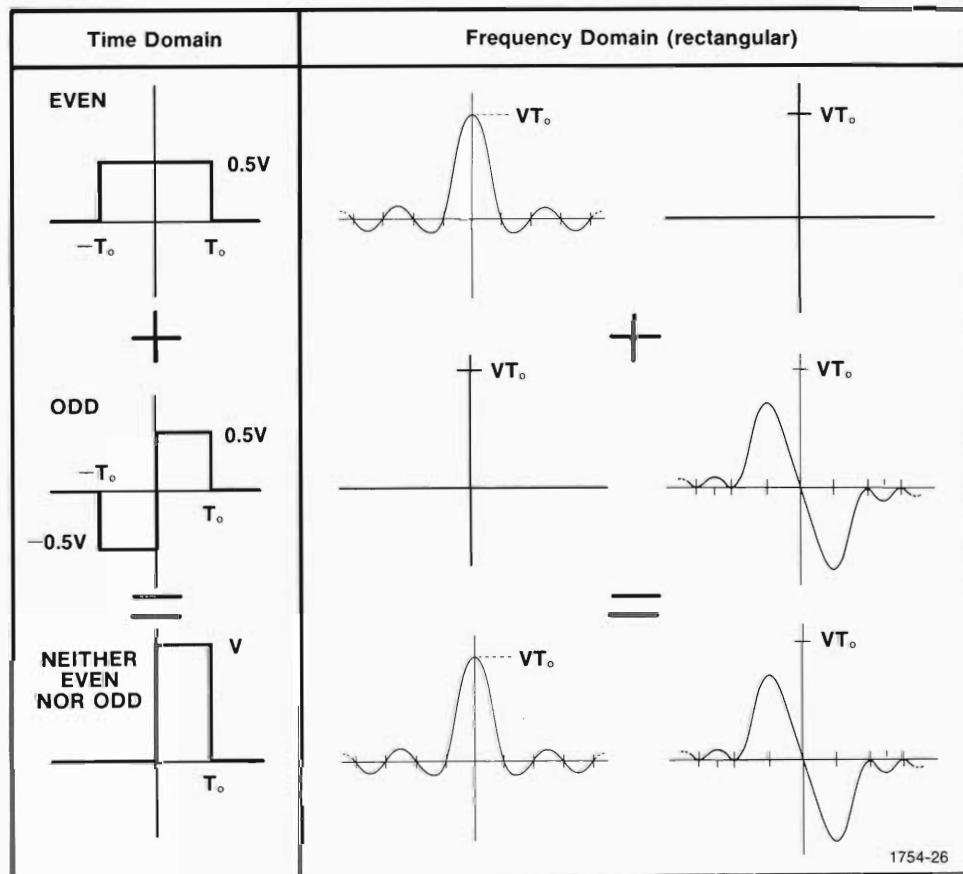


Fig. 2-14. Waveforms are either even or odd or the sum of even and odd parts.

The FFT

that the frequency-domain sum is done in rectangular form. This is because sums cannot be done directly in the polar form. So, one of the big advantages of rectangular form is directness of addition and subtraction.

After looking at Fig. 2-14, do you see how the even and odd parts sum to a function that is neither even nor odd? The conclusion to be drawn from Fig. 2-14 is that an arbitrary function can be classed as either even or odd or the sum of even and odd parts.

Actually, the square pulse in Fig. 2-14 can be made into an even function by removing some time delay. To do this, simply move the square pulse to the left until it is centered on the time origin. Then, the square pulse is an even function. Its imaginary part becomes zero, and its real part remains nonzero. In terms of the polar form, time shifting a waveform has no effect on the frequency-domain magnitude and is only reflected as a change in phase.

Shifting functions to take advantage of waveform symmetries is a standard mathematical operation. If a function can be arranged to be even or odd, both Fourier analysis and the results are simplified. There are still functions, however, that cannot be shifted for evenness or oddness. One such function is shown in Fig. 2-15. There are many others like this in real-life analyses.

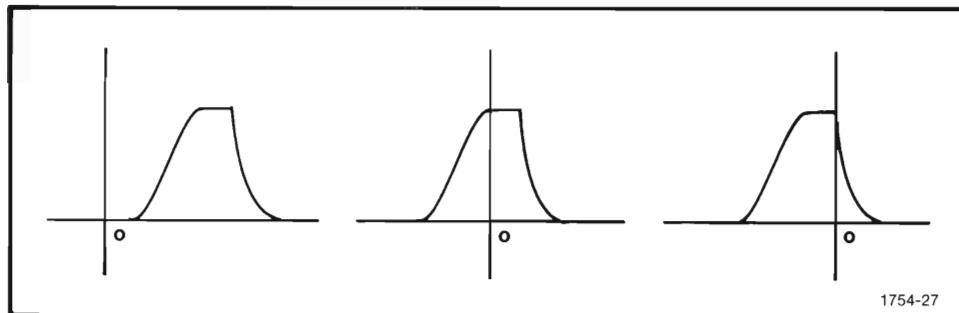


Fig. 2-15. Some functions cannot be shifted for evenness or oddness.

Periodic or Nonperiodic?—It's Your Point of View

Up to this point, the discussions of the Fourier series and the Fourier integral have taken two narrow points of view. Every waveform example has been defined to be either periodic (repeating itself from $-\infty$ to ∞) or nonperiodic (transient or pulse, occurring only once over infinite time). Everything has been done in accord with the strict theoretical definitions of periodic and nonperiodic. But, if we stick to these definitions in practice, some conflicts are bound to occur.

Consider, for example, a "sine-wave" oscillator—an actual physical circuit. When we turn on the oscillator and look at its output with an oscilloscope, we see something

The FFT

that certainly looks like a sine wave and keeps repeating in a periodic fashion. And when we turn off the oscillator, the output ceases. Everything is working like it should. We have a circuit that generates a periodic waveform, a sine wave. Right?

Wrong! Not if we are going to stay with the purely mathematical definition of periodicity. We turned the oscillator on, watched the output repeat itself for awhile, then turned it off. The oscillator's output didn't repeat over all time from minus infinity to plus infinity. In fact, the oscillator wasn't even built until quite some time after minus infinity. It generated a sinusoid for only a finite window in the infinite continuum of time. It's as if a theoretical sine wave had been multiplied by a single square pulse in the manner of Fig. 2-16.

"But", you might say, "theoretical definitions aside, it's periodic as far as I'm concerned—at least for the time I looked at it." And that's a good point of view, a practical point of view. For any physically generated waveform that repeats itself—square waves, sawtooths, etc.—you'll probably want to take the periodic point of view

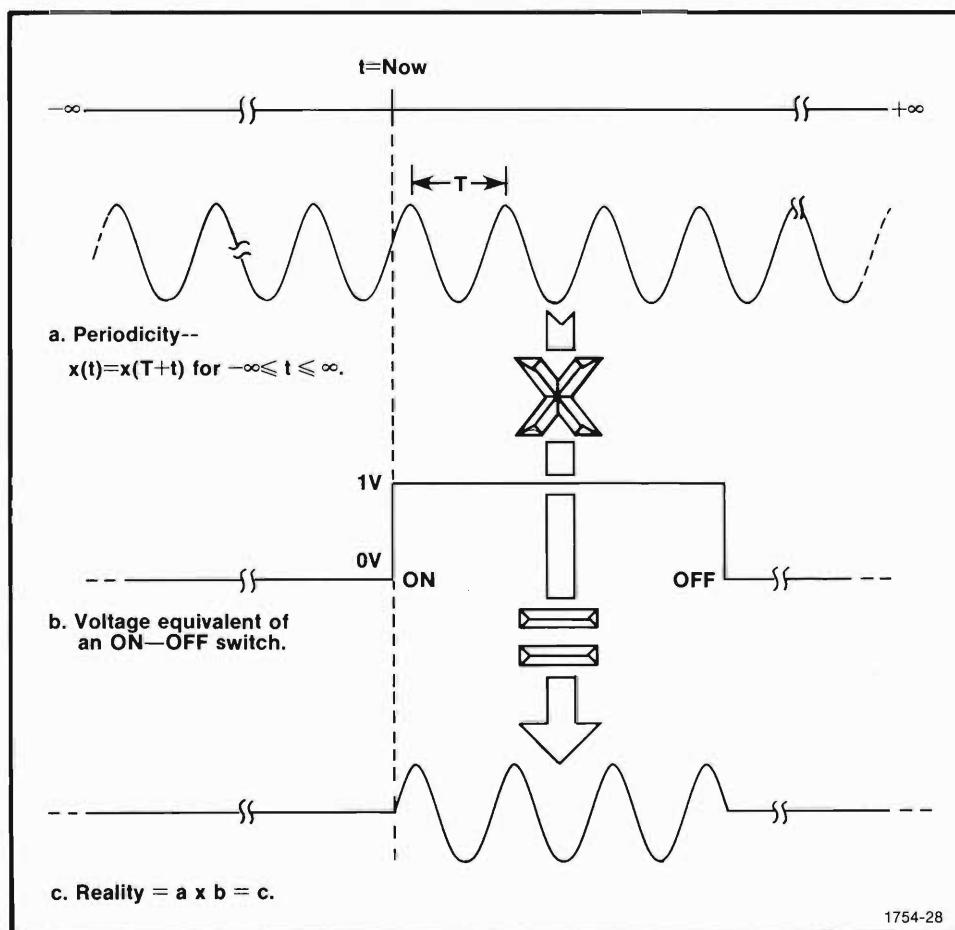


Fig. 2-16. Physically generated waveforms are either pulses (b) or windowed representations (c) of theoretical waveforms.

The FFT

and use the Fourier series for any analyses. All you need for the Fourier-series approach is one complete cycle of the waveform. But remember, using the Fourier series does imply theoretical periodicity. You'll be treating the waveform as if it did extend forever beyond the edges of the finite time window you are actually dealing with. The analysis results will contain the discrete harmonics that make up the periodic waveform.

On the other hand, you might want to take a different point of view. For example, let's say that you have a sine wave oscillator that you want to use in a remote-control application. Maybe you want to send a very short burst of the sine wave for tone control of an unmanned aircraft, or a robot, or to simply activate a switch remotely. Whatever the case, you'll be interested in the effects of gating the sine wave on and off (windowing). Then you'll want to take a nonperiodic point of view and do your analyses with the Fourier integral.

To explore the nonperiodic point of view further, let's go ahead and see how the Fourier integral is applied to a windowed waveform. And maybe more importantly, let's find out what happens when a periodic waveform is windowed.

Fourier Transform of a Rectangular Window. The rectangular window that we are going to use is shown in Fig. 2-17. You'll probably recognize it immediately as our old friend the square pulse with different proportions.

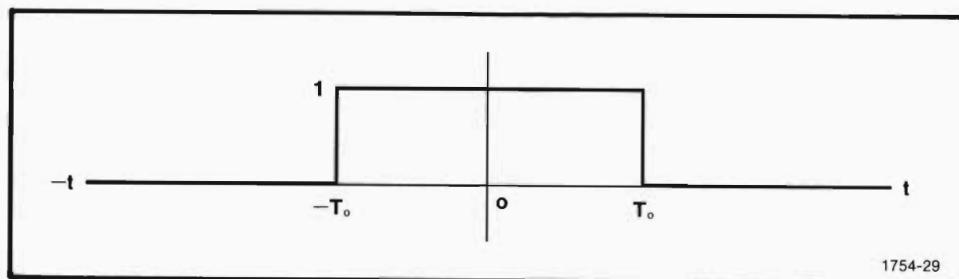


Fig. 2-17. A rectangular time window.

We already know what the frequency domain of the rectangular window looks like (Fig. 2-9); that will be of even more interest shortly. But first, let's look at how the Fourier integral is used to get the window's frequency domain.

As you may recall, the integral for transformation to the frequency domain is given by

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt.$$

For the case of the rectangular window, however, this can be simplified somewhat.

To do this, notice that the window has a value of zero everywhere except over the interval from $-T_o$ to T_o . In the nonzero interval, the window has a constant amplitude of one. Since the Fourier transform of zero is zero, it doesn't make much sense to apply the

The FFT

Fourier integral over anything but the nonzero interval. So, the integral can be changed to

$$X(f) = \int_{-T_0}^{T_0} x(t) e^{-j2\pi ft} dt.$$

And since the rectangular window, $x(t)$, has a constant value during the interval, the integral is further reduced to

$$X(f) = \int_{-T_0}^{T_0} e^{-j2\pi ft} dt.$$

This is a fairly innocuous expression for those who are familiar with calculus and can be evaluated with standard textbook methods. But let's not worry about mechanics of evaluation. This particular integral evaluates to

$$X(f) = 2T_0 \frac{\sin 2\pi f T_0}{2\pi f T_0} + j0,$$

which is the Fourier transform in rectangular coordinates (note that the imaginary part is zero). It has exactly the same magnitude and phase as shown earlier in Fig. 2-9. So, a rectangular window is the same in every respect as a square pulse.

Rectangularly Windowed Waveforms. If you recall the initial discussion of the Fourier transform, you may remember that it doesn't exist for periodic waveforms. But that needn't stop us!

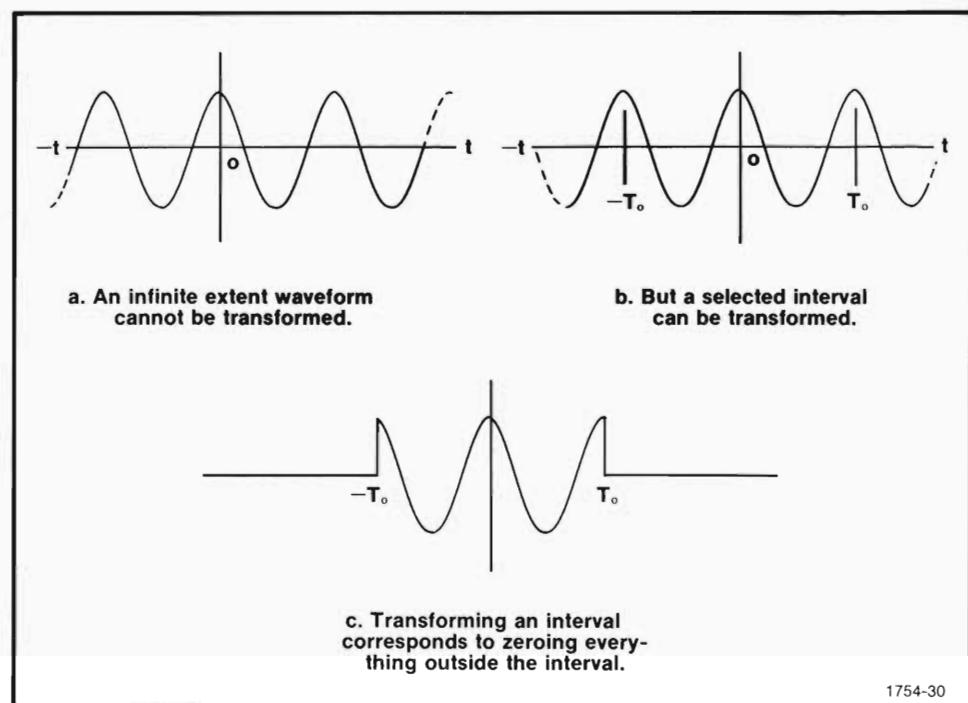
To transform sine waves, square waves, sawtooth waves, and other periodic waveforms, all you need to do is limit your view of the waveform. In short, just transform a selected interval of the waveform instead of the whole thing. This is done in the same manner as discussed for the rectangular window and is illustrated in Fig. 2-18. But keep in mind that the spectra associated with a strictly periodic signal and a windowed version of the same signal will necessarily be different. One will have a discrete spectrum (periodic), and the other (nonperiodic) will have a continuous spectrum.

In Fig. 2-18a, an infinite extent waveform is shown—in this case, a cosine wave. Since it cannot be transformed in its entirety, an interval for transformation has been marked off in Fig. 2-18b. Notice that this interval, from $-T_0$ to T_0 , is the same as indicated for the rectangular window in Fig. 2-17. The act of Fourier transforming over this interval with the Fourier integral, however, implies a zero-valued waveform outside of the interval. So what is really transformed is shown in Fig. 2-18c. It is as if the cosine wave were actually multiplied by the rectangular window in Fig. 2-17. And in essence, that is what happens whenever Fourier transformation is done over a finite interval.

But what does this mean in terms of the frequency domain? Obviously, Fig. 2-18c is not the same as Fig. 2-18a.

The actual process of windowing a waveform is shown in Fig. 2-19. The same

The FFT



1754-30

Fig. 2-18. Transforming a selected interval can change the waveform.

cosine wave and rectangular window are used. Additionally, each step is shown in the frequency domain so its effects there can be seen. In this particular case, both the window and the waveform are purposely selected to be even functions. This way, phase is zero and we can concentrate on demonstrating the concept solely with the magnitude spectrum. Using a more general case, with both magnitude and phase, would do nothing more than complicate the basic idea.

In Fig. 2-19a, an infinite extent cosine wave is shown in the time domain and as two spectral components at frequencies $\pm 1/T$. Following it, in Fig. 2-19b, is a rectangular window in the time domain and its $\sin x/x$ magnitude in the frequency domain. Multiplying the time-domain window and cosine wave results in the product shown in Fig. 2-19c. This product is referred to as a rectangularly windowed cosine wave, and the magnitude of its Fourier transform is shown in Fig. 2-19d.

Notice that the transform magnitude of the windowed cosine wave is not the product of the frequency-domain functions for the cosine wave and the window. Multiplication in the time domain does not correspond to multiplication in the frequency domain.

Yet, you may notice that Fig. 2-19d does exhibit some of the traits of both the cosine wave and the rectangular window. The $|\sin x/x|$ of the window appears twice in Fig. 2-19d, once in positive frequency and once in negative frequency. And the peaks of

The FFT

the double $|\sin x/x|$ occur at the frequency of the cosine wave ($\pm 1/T$). It's as if the two functions were somehow "rolled together" to give a new function having the major features of its constituents. And, in fact, the two functions are rolled together by a mathematical process called convolution. Fig. 2-19d is the result of convolving the cosine wave's frequency domain with the window's frequency domain.

So, multiplication in the time domain is equal to convolution in the frequency domain. And the transform of a windowed function is equal to the convolution of the function's frequency domain and the window's frequency domain.

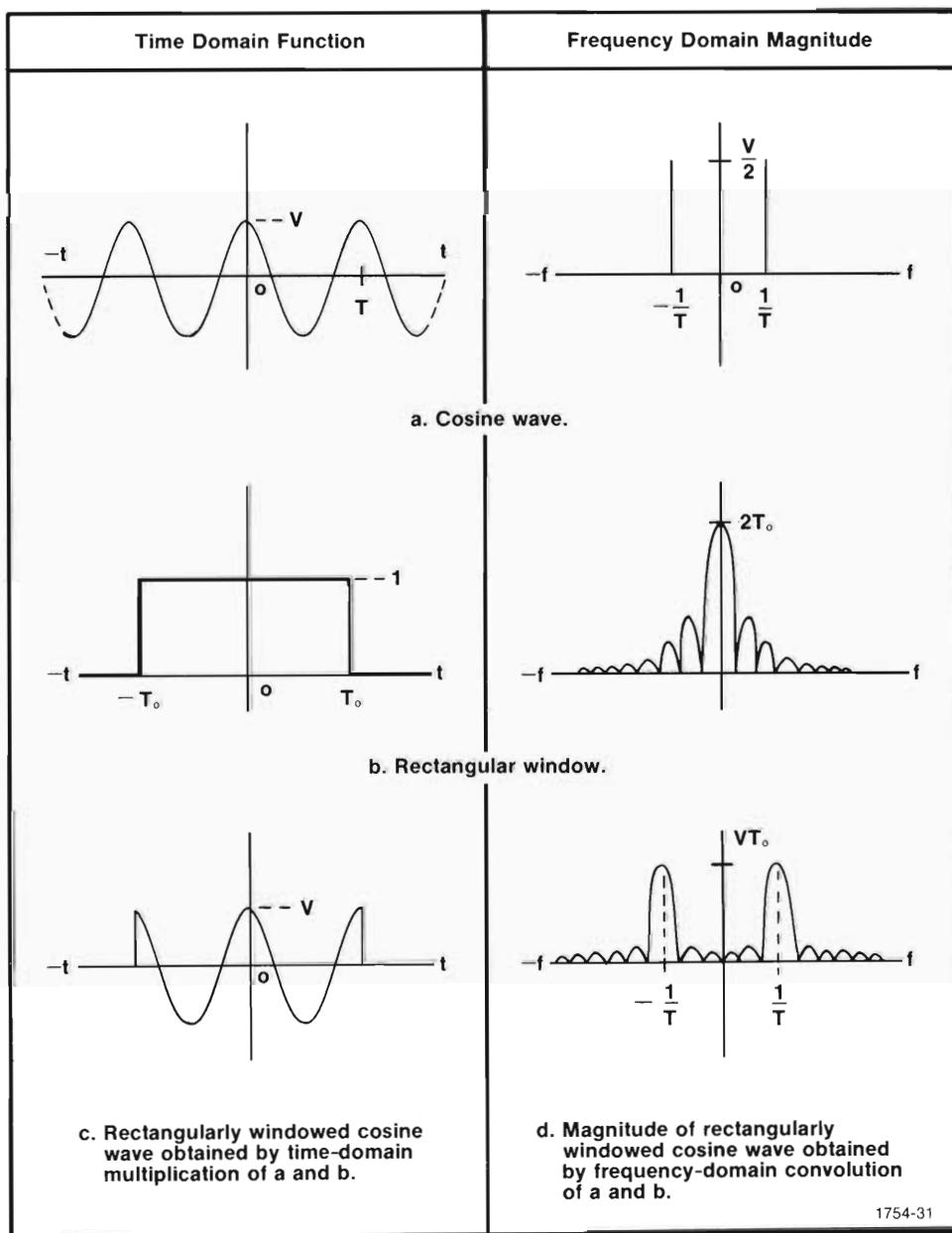


Fig. 2-19. The process of rectangular windowing.

The FFT

The topics of windowing and convolution are covered again and in more detail in Part II. It is important at this time, however, to realize that:

1. Infinite extent waveforms (sinusoids, square waves, etc.) can be transformed to the frequency domain by the Fourier integral.
2. To do this, however, you must multiply the waveform by a finite time window.
3. And this multiplication in the time domain is equivalent to convolution in the frequency domain, so the transform of a windowed signal is the convolution of the transform of the window and the signal's infinite extent counterpart.

As a final note, the converse of 3 is also true. Multiplication in the frequency domain is equivalent to convolution in the time domain.

A Summary of Some Important Fourier Transform Properties

Thus far, several important properties of the Fourier transform have been covered. For example, the idea that an arbitrary waveform is made up of odd and even parts has been discussed to some extent. However, some properties have only been presented subtly in discussing other aspects of the transform, and some important properties haven't been discussed at all. To remedy this, the more important properties of the Fourier transform are:

1. **The Fourier Transform Has an Inverse.** Although most of our discussion and examples have focused on transformation from the time domain to the frequency domain, the opposite may also be done. That is, a frequency-domain function can be transformed to obtain its corresponding time-domain function. This is the other half of the Fourier integral pair and is illustrated below in Fig. 2-20.

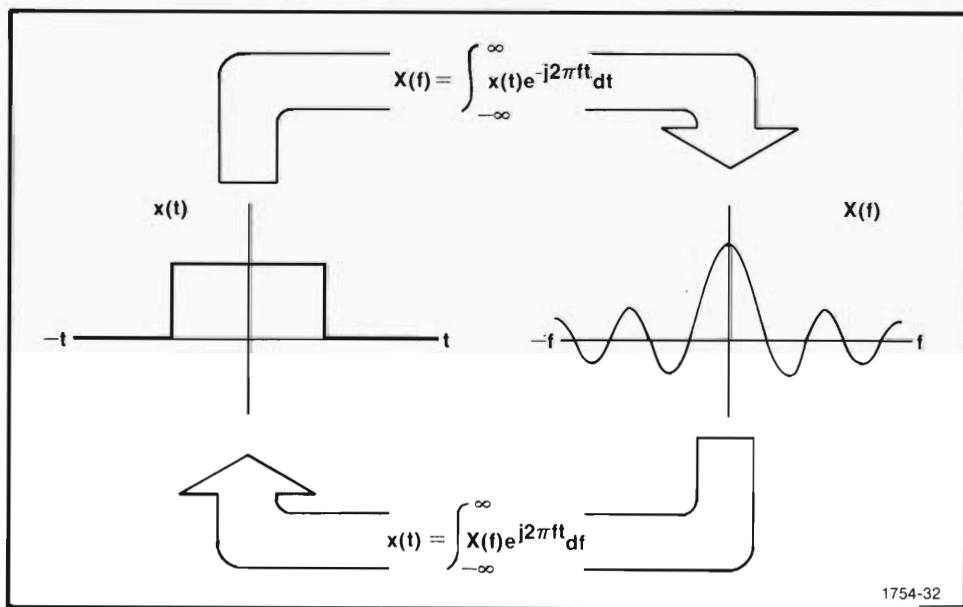


Fig. 2-20. The Fourier transform has an inverse.

The FFT

2. **Even Functions Have Real Parts Only.** If a real function given by $x(t)$ satisfies the relation $x(t) = x(-t)$, then it transforms to the real part of the frequency domain only and will be an even function of frequency. The imaginary part of the frequency domain will be zero. This property has already been discussed to some length and is illustrated in Figs. 2-12 and 2-14.

3. **Odd Functions Have Imaginary Parts Only.** If a real function given by $x(t)$ satisfies the relation $x(t) = -x(-t)$, then it transforms to the imaginary part of the frequency domain only. It will also be an odd function of frequency. This property has already been discussed and is illustrated by Figs. 2-12, 2-13, and 2-14.

4. **Arbitrary Functions Are the Sum of Even and Odd Parts.** Any function may be expressed as the sum of even and odd parts. In an even function, the odd part is zero. In an odd function, the even part is zero. Where a function is neither even nor odd, it is the sum of nonzero even and odd parts, and its frequency domain has a real and an imaginary part. This concept is summed up by Fig. 2-14.

5. **A Component Added in Time Is a Component Added in Frequency (linearity).** Suppose you have two functions given by $x(t)$ and $y(t)$, and they Fourier transform to $X(f)$ and $Y(f)$. Then, $x(t) + y(t)$ transforms to $X(f) + Y(f)$. This is most often referred to as linearity.

The groundwork for this property was laid back in Section 1 and is illustrated with Figs. 1-7 and 1-8. Also, this property is implied in the discussion of summing odd and even parts and is further illustrated in Fig. 2-14.

6. **Time Scaling Affects Frequency and Amplitude Scaling.** Suppose you have a function, given by $x(t)$, that is Fourier transformable to an $X(f)$. Now suppose you wish to rescale $x(t)$ in time by a factor "k", where k is a nonzero constant. Then $x(kt)$ Fourier transforms to $X(f/k)/|k|$.

In other words, a time-scale expansion corresponds to a frequency-scale compression and increased frequency-domain amplitude. A time-scale compression causes the opposite reaction in frequency—an expanded frequency scale and decreased amplitude. This time-scaling property is more clearly demonstrated in Fig. 2-21. There the scaling is constant, and the property is demonstrated through variation of the pulse-width parameter.

7. **Frequency Scaling Affects Time and Amplitude Scaling.** This property is comparable to time scaling. Suppose you have a function given by $x(t)$, and it is Fourier transformable to an $X(f)$. Now suppose you wish to rescale $X(f)$ in frequency by a factor of "k", where k is a nonzero constant. Then $X(kf)$ inverse transforms to $x(t/k)|k|$. This is more clearly demonstrated in Fig. 2-22.

The FFT

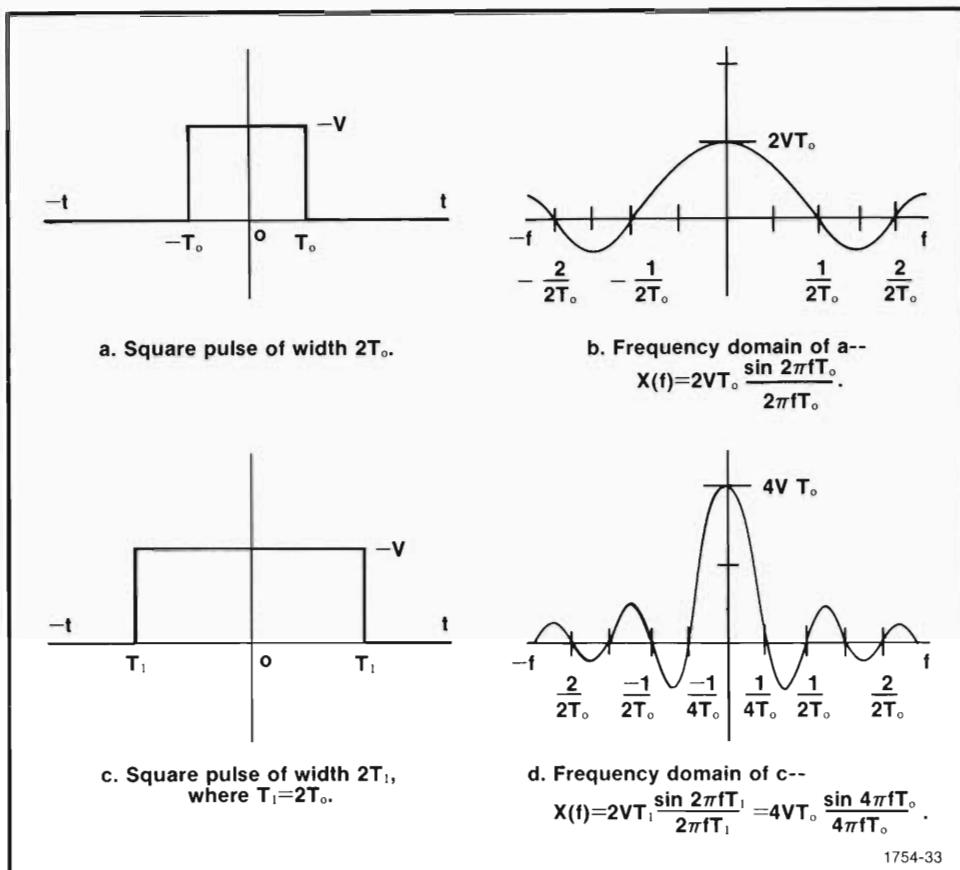


Fig. 2-21. Time expansion compresses frequency, increases amplitude.

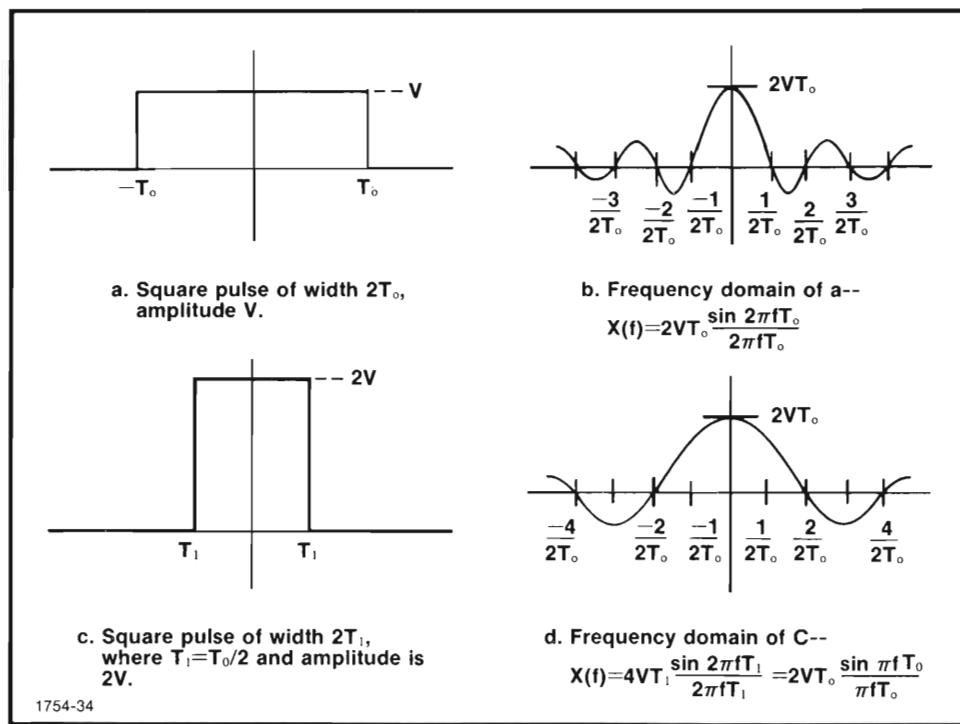
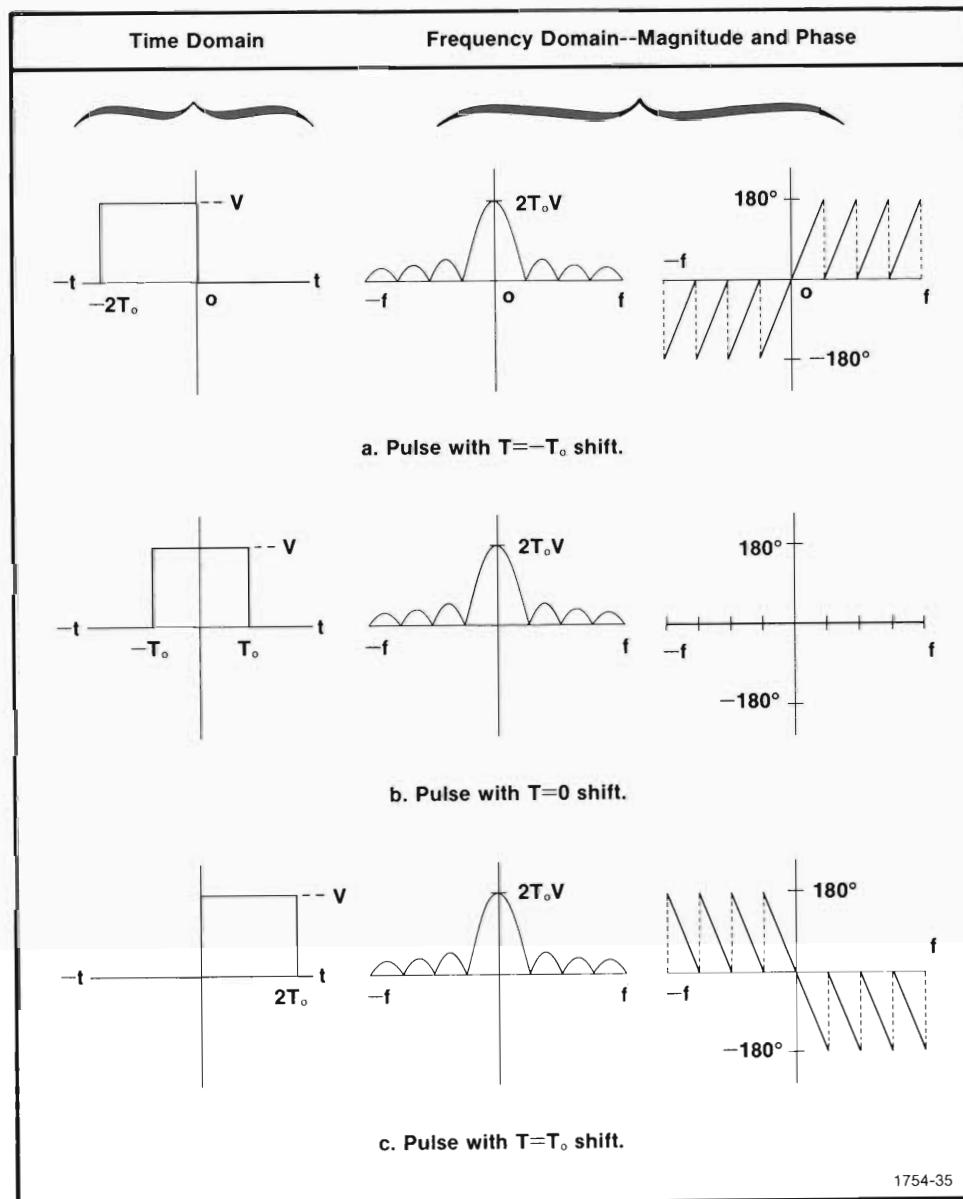


Fig. 2-22. Frequency expansion compresses time, increases amplitude.

The FFT

8. **Time Shifting Only Affects Phase Angle.** Suppose you have a function given by $x(t)$ that is Fourier transformable to an $X(f)$. Now, what happens if you shift $x(t)$ by a constant time, T ?

When you shift $x(t)$ by a constant time, T , $x(t)$ becomes $x(t-T)$, which Fourier transforms to $X(f)e^{-j2\pi fT}$. This is more clearly demonstrated in Fig. 2-23, where the frequency-domain magnitude and phase are shown. Notice that time shifting affects phase only; magnitude remains constant throughout.



1754-35

Fig. 2-23. Time shifting affects phase only.

The FFT

9. Frequency Shifting Causes Time-Domain Modulation. If $X(f)$ is inverse transformable to an $x(t)$, then shifting $X(f)$ by some constant frequency, F , results in an $X(f-F)$ that inverse transforms to $x(t)e^{j2\pi tF}$. This corresponds to a cosinusoid in the time domain being modulated by $x(t)$.

Fig. 2-24 demonstrates this shifting property. Only the real part of the frequency domain is dealt with in Fig. 2-24, so the resulting time-domain functions are even. Notice that the frequency of the cosine wave equals that of the frequency shift, F .

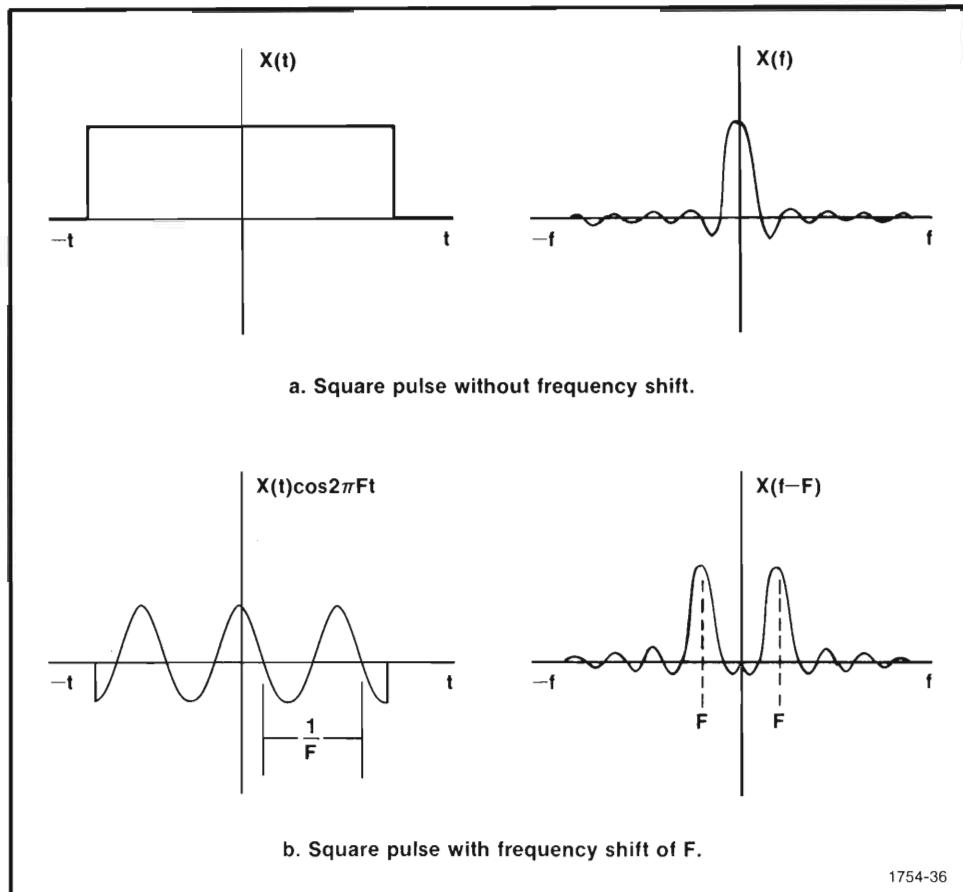


Fig. 2-24. Frequency shifting causes time-domain modulation. Notice the correspondence to windowing in Fig. 2-19.

Using the Fourier Integral—Matters of Practicality

Though it may be more versatile than the Fourier series, the Fourier integral still faces much the same matters of practicality. Chief among these is the requirement that the waveform to be transformed must be mathematically describable. If the function cannot be set to equation, then the Fourier integral cannot be directly applied. Secondary to this is the complexity of a function's description and the mental investment required for transforming it. In some cases, it is just not worth the effort to do the transform analytically.

The FFT

For many common waveforms, you can refer to standard tables and charts in engineering handbooks and texts to obtain Fourier transforms. But beyond the scope of standard waveform tables, something else has to be done—real waveforms rarely match idealized descriptions in tables. This is where digitized waveforms and digital computation come in. This is where the Discrete Fourier Transform (DFT) comes in. The digital approach to Fourier transformation is the subject of Parts II and III of this manual. There you will see how the DFT, evaluated by a Fast Fourier Transform algorithm (FFT), closely approximates the analog transform and will be introduced to the concepts necessary for practical Fourier analysis. More importantly, you will see how the FFT reduces your analysis efforts to little more than a standard oscilloscope measurement.

PART II

DIGITAL FOURIER ANALYSIS

The major concepts and properties of the Fourier transform were discussed in PART I. PART I is theory, but theory is frequently difficult to apply directly, without tools. The Discrete Fourier Transform (DFT) and the Fast Fourier Transform (FFT) are your tools for quick and easy application of Fourier theory.

In PART II, we explore digital Fourier analysis and discuss the application of Fourier theory through the DFT. The effects of digitizing a waveform are looked at, and Fourier analysis with the DFT is discussed. The bulk of this discussion, however, centers on the FFT, which is simply an efficient algorithm for computing the DFT. The major point to keep in mind throughout this discussion is that the DFT (and FFT) is viewed as a discrete approximation of the Fourier integral, the analog counterpart of the DFT. The quality of this approximation depends upon your understanding of Fourier theory and your ability to apply it to acquiring and digitizing analog signals, your ability to massage the data before transforming it, and your ability to interpret the discrete results. PART II gives you the background you'll need for addressing these requirements.

SECTION 3

INTRODUCTION TO THE DISCRETE AND FAST FOURIER TRANSFORMS

The DFT and FFT operate on finite sequences—sets of data with each point discretely and evenly spaced in time. However, the waveforms that we usually want to transform—real-life waveforms—are analog in nature. They are continuous in time, and they must be sampled at discrete points before the DFT or FFT can be applied. And, to be processed by a digital computer, these sampled points must be digitized. Understanding two basic concepts of the full analog-to-digital conversion, namely windowing and sampling, will put you a long way down the road to appreciating the power of the FFT and understanding its results.

The FFT

Windowing and Sampling are Old Ideas with New Names

If you have ever calculated the values of a waveform and plotted them on graph paper, then you have done windowing and sampling.

To see how this works, let's graph a cosine wave. For example, let's use a cosine wave having a peak amplitude of 1 volt and a frequency of 12.5 Hz. In terms of mathematics, this is represented by $x(t) = \cos 2\pi 12.5t$.

Now, think about graphing $x(t) = \cos 2\pi 12.5t$.

When the graph is done, we want everyone to recognize it as a cosine wave. Four repetitions of the waveform should be enough for that, so all we have to do is calculate values over a time interval of about $4/12.5 = 0.32$ seconds. And, so we can see the cosine wave without too much interpolating between points, let's use 32 equally spaced points. In other words, let's plot the cosine wave at every 0.01 seconds from 0.00 seconds through 0.31 seconds. The values at each of these points are listed in Table 3-1 and are plotted in Fig. 3-1. Notice that the beginning point of each cycle has a value of 1 and the end point a value of 0.707. The last-point value on a cycle is not 1 because that is the beginning value of the next cycle. Seeing this pattern is important for later discussions of the FFT and is explained further there.

TABLE 3-1

Values of $x(t) = \cos 2\pi 12.5t$ at every 0.01 second over the interval from 0.00 second through 0.31 second.

1754-37

t	x(t)	t	x(t)	t	x(t)	t	x(t)
0.00	1.000	0.08	1.000	0.16	1.000	0.24	1.000
0.01	0.707	0.09	0.707	0.17	0.707	0.25	0.707
0.02	0.000	0.10	0.000	0.18	0.000	0.26	0.000
0.03	-0.707	0.11	-0.707	0.19	-0.707	0.27	-0.707
0.04	-1.000	0.12	-1.000	0.20	-1.000	0.28	-1.000
0.05	-0.707	0.13	-0.707	0.21	-0.707	0.29	-0.707
0.06	0.000	0.14	0.000	0.22	0.000	0.30	0.000
0.07	0.707	0.15	0.707	0.23	0.707	0.31	0.707

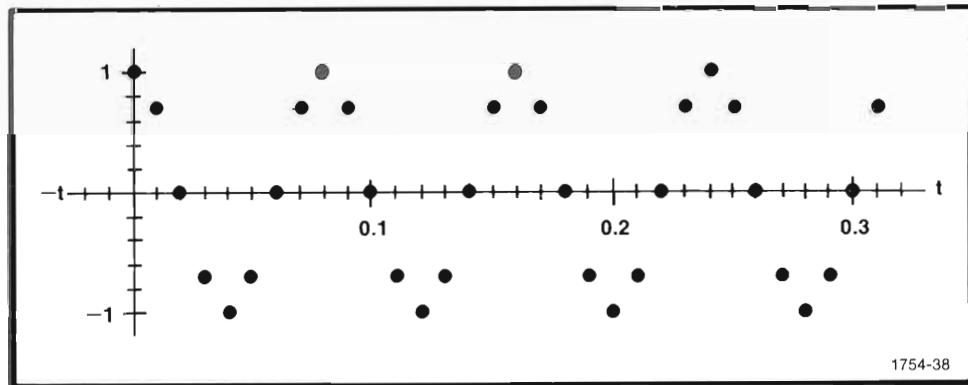


Fig. 3-1. A windowed and sampled cosine wave.

The FFT

Look at Fig. 3-1 again.

Fig. 3-1 is a plot of a windowed and sampled cosine wave. It is windowed by virtue of deciding to plot the waveform over a finite time interval, in this case 0.00 through 0.31 seconds. It is sampled by virtue of the decision to find and show its actual values at only 32 discrete points in that interval. So you see, the fundamental concepts of windowing and sampling date back at least to the concept of graph paper.

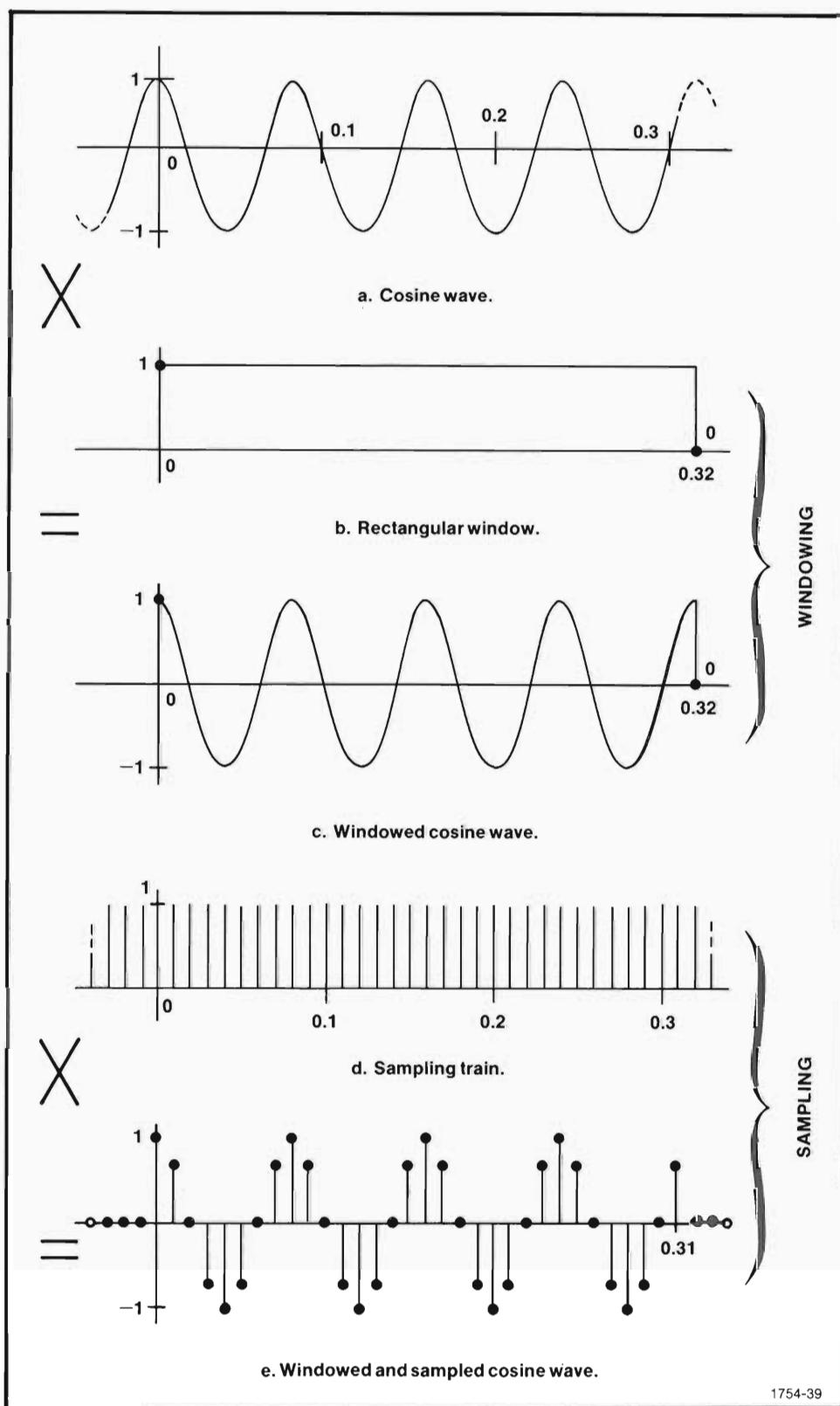
In terms of waveform processing, Fig. 3-1 can be thought of as being obtained through the multiplications shown in Fig. 3-2. Fig. 3-2a is the waveform, in this case a cosine wave, that is going to be windowed, and Fig. 3-2b is a rectangular window (a square pulse). Their product, shown in Fig. 3-2c, is a rectangularly windowed cosine wave. In Fig. 3-2b and c, notice the heavy dots at the window edges. These aren't actually part of the window, but are illustrative devices for signifying the values at the window edges. The window in Fig. 3-2b is defined to begin at time zero with an amplitude of one and continues up to 0.32 seconds with a constant amplitude of one. The window ends at 0.32 seconds with an amplitude of zero. Such precise definition of window edges becomes more important in later discussions of the periodicity assumed by the FFT.

After windowing, sampling is done with the train of unity amplitude impulses shown in Fig. 3-2d. Each impulse in this train rises from zero to infinity and returns to zero, all instantaneously. Thus, each is nonzero at only a single, discrete time. Also, each impulse encloses an area equal to one, and the train of impulses is graphically represented by a train of spikes having height equal to their integral or area. When this impulse train, or sampling train, is multiplied with the windowed cosine wave, the product (Fig. 3-2e) is a windowed and sampled cosine wave. The train of impulses then has an area equal to the corresponding cosine value. The time locations of the samples correspond to the time locations of the impulses. And, since each impulse exists at a single, discrete time, each sample exists at a single, discrete time. The discrete sample times and sample values are indicated in Fig. 3-2e by the heavy dots.

Notice that the windowed and sampled cosine wave in Fig. 3-2e is exactly the same as the graphed cosine wave in Fig. 3-1, except the impulses aren't shown in Fig. 3-1. Also, the values in Table 3-1 apply equally to Fig. 3-1 and Fig. 3-2e. So, Fig. 3-2 is just another way to get the results of Table 3-1 and Fig. 3-1.

In terms of physical processes, the results in Fig. 3-2e and Table 3-1 can be obtained by using an analog-to-digital converter. This process does not actually involve physical generation and multiplication of waveforms as shown in Fig. 3-2, but the conversion is analogous and produces the same end result. For example, the window in Fig. 3-2b corresponds to triggering the input of an analog-to-digital converter on and off. When the converter is triggered on, the waveform to be digitized is allowed to pass; when switched off, the waveform is blocked. The result is equivalent to multiplying the waveform to be digitized by a square pulse. The window length or pulse width corresponds to the length of time the analog-to-digital converter is on.

The FFT



1754-39

Fig. 3-2. Windowing and sampling equates to several waveform multiplications.

The FFT

In practice, sampling is done by the same means. The windowed waveform is looked at through much smaller, closely spaced sampling windows. When the sampling window is gated on, a capacitor or some other device is allowed to assume the value of the waveform during the sample time. A single value is thus obtained at each sample point on the waveform, as shown in Figs. 3-1 and 3-2e, and these values can be stored in a memory corresponding to Table 3-1. In most cases, the sample values are converted to digital data and stored in a memory compatible with a computer. Then a computer or some related digital device can be used to process the data.

There are a number of ways that windowing and sampling can be done. As was just discussed, some analog-to-digital converters window first and then sample. Others sample first, then window the desired block of sampled data. In still others, windowing and sampling occur at the same time. In any case, you will find that the end result is always the same—a windowed and sampled waveform.

There are, however, different windowing shapes and sampling schemes that may be used. Fig. 3-2 uses a square pulse for a window. A cosine squared pulse could have been used instead, and the results would have been different. These various window shapes are discussed later in Section 5.

As far as sampling is concerned, samples can be taken over equally spaced intervals, logarithmically spaced intervals, or by any other spacing scheme. But the use of equally spaced samples is probably more widespread than any other technique, and further discussions are confined to cases using equal-spaced sampling. When the term "sampling" is used in future discussions, it implies equal-spaced sampling.

The Discrete Fourier Transform Works on Sampled Data

Fig. 3-2e can be transformed to the frequency domain by applying the Fourier integral, over the window interval, to the product of the waveform and impulse train functions. In fact, if you sit down with pencil and paper and do the transform, paying attention to notation and performing the correct manipulations, you'll come up with an expression known as the discrete Fourier transform (DFT). It looks like this:

$$X_d(k\Delta f) = \Delta t \cdot \sum_{n=0}^{N-1} x(n\Delta t) e^{-j2\pi k \Delta f n \Delta t}$$

This expression allows you to transform a time series of samples, like Table 3-1, to a series of frequency-domain samples.

By some further manipulations, you can also develop the inverse DFT. It looks like this:

$$x(n\Delta t) = \Delta f \cdot \sum_{k=0}^{N-1} X_d(k\Delta f) e^{j2\pi k \Delta f n \Delta t}$$

This expression allows you to transform a series of frequency-domain samples,

The FFT

computed by the DFT, back to a series of time-domain samples.

In both of these discrete expressions, the variables have the following definitions:

N—N is the number of samples being considered.

$\Delta t - \Delta t$ is the time between samples, referred to as the "sampling interval".

From this, $N\Delta t$ gives the window length, often referred to as the "time record length".

$\Delta f - \Delta f$ is the sample interval in the frequency domain and is equal to $1/N\Delta t$.

n—n is the time sample index. Its values are $n=0, 1, 2, \dots, N-1$.

k—k is the index for the computed set of discrete frequency components. Its values are $k=0, 1, 2, \dots, N-1$.

$x(n\Delta t) - x(n\Delta t)$ is the discrete set of time samples that define the waveform to be transformed.

$X_d(k\Delta f) - X_d(k\Delta f)$ is the set of Fourier coefficients obtained by the DFT of $x(n\Delta t)$.

e—e is the base of the natural logarithm.

j—j is the symbol of complex notation, indicating the imaginary part of a complex quantity ($j = \sqrt{-1}$).

By continuing with some substitutions, letting $\Delta t=1$ so that $\Delta f=1/N$, you can arrive at the more common form of the DFT and the inverse DFT. These are:

$$\text{DFT} - X_d(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}$$

$$\text{and inverse DFT} - x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X_d(k)e^{j2\pi kn/N}$$

Since the $1/N$ before the summation in $x(n)$ is simply a scaling term, it can be included in either (not both) the direct or inverse expression. For the examples given here, $1/N$ is shifted to the direct expression, $X_d(k)$. This alternate formulation gives the dc term, $X_d(0)$, as the arithmetic mean of the time samples.

For computational convenience, Euler's identity ($e^{\pm j\Theta} = \cos \Theta \pm j \sin \Theta$) is used to change the complex exponential to give the DFT and inverse DFT as:

$$X_d(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \cos 2\pi kn/N - j x(n) \sin 2\pi kn/N$$

The FFT

and

$$x(n) = \sum_{k=0}^{N-1} X_d(k) \cos 2\pi kn/N + jX_d(k) \sin 2\pi kn/N$$

Now, with these expressions, it is a fairly straightforward set of operations to compute the discrete Fourier transform for any string of waveform samples. For example, you can compute the DFT of the cosine wave from the values listed in Table 3-1. There are 32 samples there, so k and n take on values of 0, 1, 2, ..., N-1. The values of $x(n)$ are taken directly from the table—where $x(0)$ corresponds to time zero and has a value of 1.000, $x(1)$ corresponds to 0.01 seconds and has a value of 0.707, and so on. Each Fourier coefficient, $X_d(k)$, is computed by summing $x(n)\cos 2\pi kn/N - j\sin 2\pi kn/N$ for all values of n at each k. For example, let k equal zero and sum $x(n) \cos 2\pi kn/N - j\sin 2\pi kn/N$ for $n=0$ to $N-1$. Then let k equal one and do the summation again for all n. This goes on until the summation for $k=N-1$ is reached and completed. When that is done, you have the set of 32 Fourier coefficients for the 32 time-domain samples of the cosine wave. These Fourier coefficients define the cosine wave's complex frequency domain at 32 discrete frequencies.

If you have taken up pencil and paper to try out the DFT on Table 3-1, it won't take you long to realize that there are 32 terms to be summed for each of the 32 values of k. That means there are $32 \times 32 = 1024$ major operations for doing a 32-point DFT. Although each operation is in itself fairly easy, doing it 1024 times is not. There has to be a better way!

A DFT Program Makes it Easier. Computing the DFT is essentially a repetitive task. The major operations are the same, over and over again. The only things that change are the index values, k and n, and the sample values, $x(n)$. So, the DFT is well suited to evaluation by computer program.

Put away your pencil and paper! Here is a program that generates the values in Table 3-1 and computes the Fourier coefficients by straightforward evaluation of the

DPO TEK BASIC or WDI TEK BASIC Program for 32-point DFT	
100 DIMENSION X(31),XR(31),XI(31) 105 LET PI=3.1416:LET X=0:LET EM=1 110 FOR I=0 TO 31 115 LET X(I)=COS(2*PI*12.5*I/100) 120 NEXT I 130 STOP	{ Define data arrays. Generate values of Table 3-1 in array X.
200 LET XR=0:LET XI=0 205 FOR K=0 TO 31 210 FOR N=0 TO 31 215 LET AG=2*PI*K*N/32 220 LET XR(K)=XR(K)+X(N)*COS(AG)/32 225 LET XI(K)=XI(K)-X(N)*SIN(AG)/32 230 NEXT N 235 NEXT K 240 STOP	{ Compute 32-point DFT of array X into the real array, XR, and the imaginary array, XI.
300 PRINT "A[AL" 305 PRINT "32-POINT DFT:";#20;"COEFFICIENT";#40;"FOURIER COEFFICIENTS" 310 PRINT "#20;" INDEX";#35;"REAL PART";#54;"IMAGINARY PART" 315 FOR I=0 TO 31:PRINT #23;I:#32:XR(I);#54:XI(I):NEXT I 320 STOP	Print results. ← {

1754-40

The FFT

DFT expression. It works with either Tektronix, Inc., DPO TEK BASIC software or WDI TEK BASIC software. Just enter the program on your Tektronix Computer Display Terminal, and then type GOTO 100 and press RETURN to generate the cosine data of Table 3-1. Then enter GOTO 200 to evaluate the DFT (evaluation will take several minutes). To print the DFT results, enter GOTO 300; the results for the data in Table 3-1 are shown in the left half of Table 3-2.

TABLE 3-2
32-point Fourier transformation of the data in Table 3-1.

Index	DFT		FFT	
	Real Part	Imaginary Part	Real Part	Imaginary Part
0	DC	-3.052E-5	0	0
1	0	-7.629E-6	3.052E-5	0
2	-3.052E-5	-7.629E-6	3.052E-5	0
3	0	2.289E-5	3.052E-5	0
4	0.5	-2.289E-5	0	3.052E-5
5	6.104E-5	-3.815E-5	-3.052E-5	6.104E-5
6	-3.052E-5	-7.629E-6	0	6.104E-5
7	-3.052E-5	-3.815E-5	0	6.104E-5
8	0	-3.052E-5	0	0
9	1.221E-4	-6.104E-5	0	0
10	-6.104E-5	1.526E-5	0	0
11	0	3.052E-5	3.052E-5	0
12	3.052E-5	0	0.5	-3.052E-5
13	6.104E-5	-3.815E-5	0	-3.052E-5
14	0	-6.867E-5	3.052E-5	-3.052E-5
15	-3.052E-5	6.867E-5	3.052E-5	-3.052E-5
16	0	1.526E-5	3.052E-5	0
17	0	4.578E-5	3.052E-5	3.052E-5
18	3.052E-5	-4.578E-5	3.052E-5	3.052E-5
19	3.052E-5	2.289E-5	0	3.052E-5
20	3.052E-5	-9.918E-5	0.5	3.052E-5
21	6.104E-5	5.341E-5	3.052E-5	0
22	-1.631E-4	-9.918E-5	0	0
23	-1.221E-4	8.392E-5	0	0
24	-1.221E-4	3.815E-5	0	0
25	-6.104E-5	9.155E-5	0	-6.104E-5
26	-1.526E-4	1.526E-5	0	-6.104E-5
27	6.104E-5	4.578E-5	-3.052E-5	-6.104E-5
28	0.5002	3.052E-5	0	-3.052E-5
29	1.221E-4	-1.144E-4	3.052E-5	0
30	4.273E-4	1.144E-4	3.052E-5	0
31	-2.136E-4	-1.068E-4	3.052E-5	0

1754-41

Frequency interval is given by $\Delta f = 1/N\Delta t$. Bold type indicates frequency components for cosine wave in Table 3-1; others are effectively zero.

*The Nyquist frequency is the highest frequency sinusoid that can be defined at a given sampling rate. For equally spaced samples (Δt), the Nyquist frequency is $1/2\Delta t$.

The FFT

Now enter FFT X,XR,XI from the terminal. The same Fourier coefficients are computed by the FFT algorithm. Notice how much faster this is. To print these FFT results, enter GOTO 300; these results are shown in the right half of Table 3-2. (Slight variations between the DFT and FFT results in Table 3-2 are due to different round offs in the two routines.)

With regard to this DFT—FFT program comparison, there are several things that should be pointed out. First, the DFT program in lines 200 through 240 is a rather inelegant, but straightforward, approach to evaluating the DFT. It is easy to follow, but the results aren't in an order corresponding to our frequency-domain diagramming convention. Furthermore, for real signals, this approach uses twice as many iterations as needed. Since the negative frequency domain duplicates the positive frequency domain for real signals, it is only necessary to compute the coefficients for index 0 through 16. Doing this cuts the execution time in half, but it still doesn't approach the speed of the FFT statement. The second thing that should be pointed out is that the example program is in a high-level language (TEK BASIC). It would execute faster if it were implemented in low-level assembly language, like the algorithm for the FFT statement. But even then, the direct evaluation of the DFT wouldn't be as fast as the FFT algorithm. In short, the FFT algorithm is to date the most elegant and time effective approach to evaluating the discrete Fourier transform.

How the FFT Came About

When you consider the 32^2 major operations required for the 32-point DFT just computed, it isn't difficult to understand why discrete Fourier analysis was generally avoided by scientists working before the development of the modern digital computer. And for many applications 32 samples is not enough. The required N samples for defining useful functions frequently runs into the hundreds. So the prospect of N^2 operations by hand calculation was enough to discourage discrete Fourier techniques as an analysis tool.

But some fields of study beg for Fourier analysis. In fact some information can only be gained by Fourier analysis, and there is nothing to do but plunge forward with the calculations.

Accordingly, it was standard in the days of hand calculation to be as concerned with minimization as well as application possibilities. Scientists welcomed papers describing means for reducing the number of calculations in an analysis technique almost as much as they welcomed the new analysis technique itself. It was in such a minimization paper (for 12- and 24-point transforms) in 1903 that C. Runge described the technique that later became known as the FFT. Later, in 1942, a more generalized approach was advanced by Danielson and Lanczos. By recognizing certain symmetries and periodicities, they reduced evaluation of the DFT for $N=2^k$ points to $N \log_2 N$ major operations. The significant savings of $N \log_2 N$ operations versus N^2 operations is apparent in Fig. 3-3. Evaluation by hand, however, is still a staggering task for reasonable values of N. For example, using an efficient algorithm, L.H. Thomas of the IBM Watson Laboratory reported spending three months in 1948 doing a transform with

The FFT

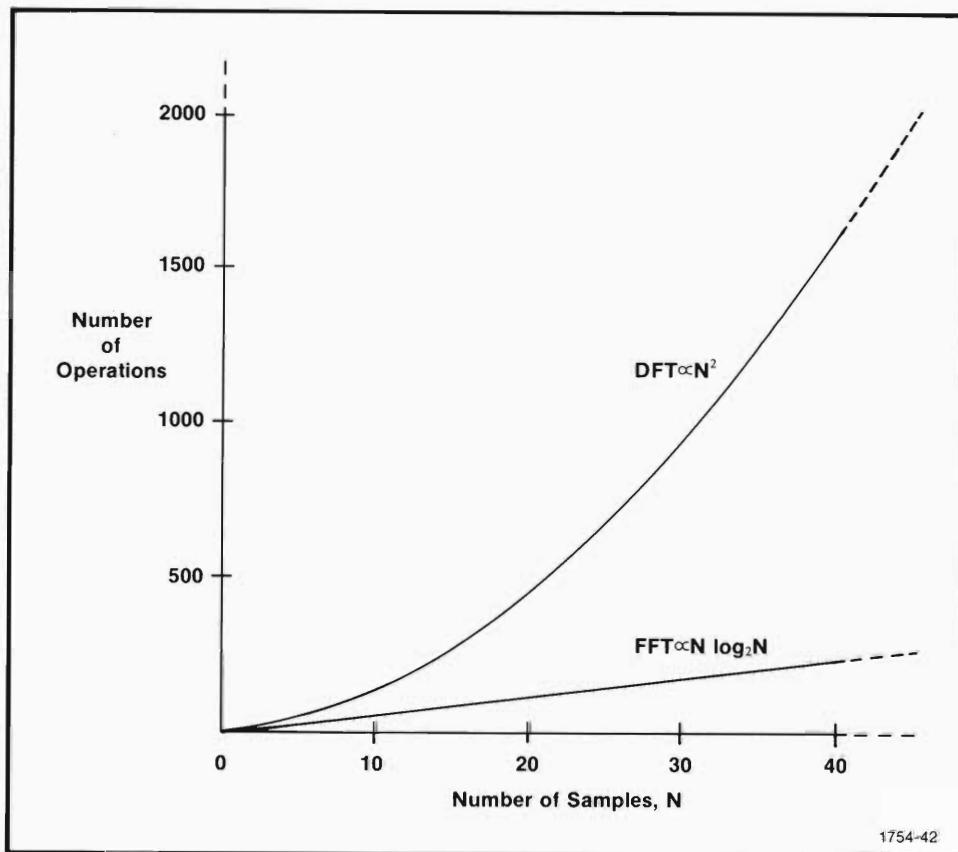


Fig. 3-3. How fast is fast? The more operations to be done, the longer it takes. The FFT reduces the number of operations, and for large N, the time advantage is tremendous. For example, the FFT for 512 samples has a better than 50 to one advantage over the DFT.

the aid of an office accounting machine. It's no wonder people steered clear of Fourier analysis whenever possible!

Then came the digital computer, but without widespread use, the techniques of Danielson and Lanczos were generally unknown, and Fourier analysis was still done by the direct approach, using N^2 operations. Even with a digital computer, N^2 operations takes significant time for large N. The cost of doing Fourier analysis by discrete methods was prohibitive in many cases.

Still, Fourier analysis is necessary to some research. In the early 1960's, R.L. Garwin was studying solid helium and had a great need for Fourier techniques. His need caused him to contact a colleague, J.W. Tukey, and ask for an efficient means of evaluating the discrete Fourier transform. Tukey supplied him with the essence of what was to become known as the fast Fourier transform. Then Garwin approached the director of Mathematical Sciences at IBM with the problem of programming the algorithm. As a result, J.W. Cooley became involved.

The FFT

In Cooley's words: "Garwin came to the computing center at IBM Research in Yorktown Heights to have the algorithm programmed. I was new at the computing center and was doing some of my own research. Since I was the only one with nothing important to do, they gave me this problem to work on. It looked interesting, but I thought that what I was doing was more important; however, with a little prodding from Garwin, I got the problem out in my spare time and gave it to him. It was his problem and I thought I would hear no more about it and went back to doing some real work."¹

But Cooley didn't hear the last of it. Garwin saw a wide range of applications for the program besides studying helium, and began contacting various scientists and making the possibilities known. "As a result of his publicity," Cooley relates, "I started to get letters requesting programs and write-ups. Requests for a paper then started arriving. I was asked to write a paper and did so, asking Tukey to co-author it. He did, and the paper in Mathematics of Computation, in 1965, was the result." This paper describes the Cooley-Tukey algorithm for evaluating the discrete Fourier transform.

For the most part, the Cooley-Tukey algorithm is simply referred to as the FFT. There are also a number of other algorithms, offshoots of the Cooley-Tukey algorithm, that are also lumped under the term "FFT," and any algorithm that provides the $N \log_2 N$ advantage is generally referred to as an FFT.

Putting the Algorithm to Work—Hardware or Software FFT?

Garwin's approach to the FFT was a software approach. General-purpose computers were available to him, and he had already obtained by computer simulations a very large amount of nuclear spin data for helium. All he needed was a program to Fourier transform the data. A software implementation of the FFT was the natural approach.

Software implementation of the FFT has some distinct advantages, too. The FFT algorithm can be programmed in a general enough manner so that the number of points to be transformed can be varied. If you have 512 samples, the program does a 512-point FFT; if you have 64 samples, a 64-point transform is done. With the flexibility offered by software, the FFT algorithm can do any length transform you desire, within the limits of the specific routine and computer memory size. (Quite often, though, the length is restricted to a power of two—2, 4, 8, 16, 32, 64, 128, ... This offers algorithm simplicity and additional execution speed.) Also, with general-purpose software, you can choose to do a variety of further processing or analysis if you like, and you can put the results in just about any form you want—tables, graphs, diagrams, etc. In short, you make the software FFT fit the data instead of making the data fit the FFT.

There are some tradeoffs in a general-purpose software implementation, though. The options that make it general purpose require decisions and interpretations by

¹James W. Cooley's keynote address at the 1968 Arden House Workshop on Fast Fourier Transform Processing, *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-17 No. 2, June 1969.

The FFT

software, and this takes additional time. Of course, execution can be speeded up by tailoring the FFT routine to a specific need, but then the program may not be flexible enough to handle other applications.

Another tradeoff is that a software FFT is, strictly speaking, an "off-line" operation; the signal data is acquired, stored, and then processed. The results are not "real-time" results, but occur some finite time after the input signal. However, from a practical viewpoint, real-time results often don't have to be instantaneous results. If the time between acquiring a signal and getting FFT results is milliseconds and signal variations are on the order of seconds or more, then the FFT results might effectively be considered real-time results. On the other hand, signal variations on the order of microseconds would disqualify a millisecond return of results from being considered as real-time results. But then, a large block of FFT applications do not require real-time results.

As a final point, a software implementation of the FFT algorithm simply does not execute as fast as the corresponding hardware implementation. Compared to a software approach, where a general-purpose machine is directed in every step of the task, a hardware FFT has the instructions built into it. It just goes ahead and does its specific task—no questions asked, and usually very little flexibility offered. In fact, a dedicated hardware FFT processor—designed for a specific, narrow application—can be so fast as to be considered an "on-line" analyzer, providing results that are "real-time" for all intents and purposes. But you can't tell it to do anything else but that one specific task. You have to make your data fit that hardware FFT!

Regardless of the FFT implementation, the final results are the same for the same algorithm. The algorithm itself is simply a method of evaluation. How you choose to implement the method—hardware, software, or pencil and paper—makes no difference in the results. The hardware FFT of a waveform is the same as the software FFT of the same waveform, and both of these are the same as the pencil-and-paper FFT for that same waveform. The major consideration is how quickly you get the results. And this depends upon the hardware design, or the software design and the computer's instruction speed, or how fast you are with pencil and paper.

SECTION 4

UNDERSTANDING FFT RESULTS

In Section 2, continuous, periodic signals and their transformation to the frequency domain by a Fourier series were discussed. Then continuous, nonperiodic signals and their transformation by the Fourier integral were discussed. After that, the idea of transforming windowed signals with the Fourier integral was introduced, and the concept of interpreting results from a periodic or nonperiodic point of view was introduced.

Then in Section 3, we went a step further. Windowed signals were sampled and transformed by the discrete Fourier transform, and the FFT was introduced as an efficient method of evaluating the DFT.

The DFT, or FFT, is a third type of Fourier transform. It is related to the Fourier series and Fourier integral, but it is defined only for discrete values over a finite interval. From a digital viewpoint, the FFT provides the exact transform for the discrete values provided. A continuous, periodic, or nonperiodic signal can be windowed and sampled to provide discrete values, and these discrete values can also be transformed by the FFT. The results from transforming a windowed and sampled function with the FFT will also be exactly as they should be as long as they are viewed from a digital viewpoint. That is, the FFT provides the correct frequency domain information for the windowed and sampled version of the waveform. If, however, you wish to interpret the FFT results from a continuous viewpoint, as is usually the case, then the interpretation must be done with care. An interpretation from the continuous viewpoint must always be tempered with the realities of the windowed and sampled data.

This section, Section 4, looks at FFT results in general and then goes on to explore the effects of windowing and sampling and how they influence FFT results from a continuous viewpoint. For convenience, the examples in this section are built around the Sande-Tukey FFT algorithm used in Tektronix, Inc., DPO TEK BASIC and WDI TEK BASIC software. (Where the Cooley-Tukey algorithm uses a process called decimation in time, the Sande-Tukey algorithm uses decimation in frequency. While the methods differ, the two algorithms produce identical results.) Also, because of their convenient size for publication, displayed results are taken from the cathode ray tube of a Tektronix, Inc., Digital Processing Oscilloscope. Although these examples are obtained via specific software and instrumentation, the concepts they illustrate are applicable to FFT's in general.

The FFT

TEK BASIC DATA FORMATS FOR THE FFT

For the most part, the examples in this section deal with 512-element data arrays. These are the standard waveform arrays of DPO TEK BASIC and WDI TEK BASIC software; however, arrays of other lengths can, in practice, be used with the FFT. For example, a 32-element array was used in Section 3 to introduce the DFT and the FFT.

Providing Time-Domain Data for the FFT

In the most fundamental sense, there are two types of time-domain information available for any type of analysis. There is information that is directly measurable or observable, and there is theoretical information generated by mathematical formulas or other means of speculation.

For the case of measurable information, like the waveform in Fig. 1-10 of Section 1 that was used to point out the need for Fourier analysis, the information can be acquired and sampled by standard measurement techniques. Then the samples from the acquisition window can be digitized and stored in a data array for further processing. In the case of DPO TEK BASIC and WDI TEK BASIC software, the length of a waveform data array is 512 elements, and each array element corresponds to a waveform sample from the acquisition window. Array element zero corresponds to time zero at the left edge of the window, as indicated in Fig. 4-1, and array element 511 corresponds to 1 sample less than the maximum time at the right edge of the window. The remaining elements, 1 through 510, correspond to equally spaced points between the window edges. The digital values stored in each array element correspond to the sampled values of the waveform at each element's corresponding time point. In short, the waveform array corresponds to the table of values you might construct from point-by-point measurement of the waveform. The sampled waveform, shown in Fig. 4-1, would then correspond to a graph you might draw from the table of point-by-point measurements.

For the case of theoretical data, the computed values are simply entered into the data array in much the same manner that Table 3-1 in Section 3 was constructed. However, in the case of Table 3-1, the array would be a 32-element array. To place the same theoretical cosine wave, over the same time window, into a standard 512-element array for TEK BASIC, you would need to compute samples at every 0.625 milliseconds (0.32 seconds/512 time increments) from 0 seconds to 0.32 seconds. In either case, an FFT of the theoretical data can be done. In the former, a 32-point FFT is done, and a 512-point FFT is done for the latter. The major difference, of course, is that 512 samples over the same interval provides better resolution.

The great majority of measured and theoretical time-domain data is real-valued data. That is, it is real not in the sense that it is physical, but real in the sense that it is not complex-valued data. It doesn't have an imaginary part, or at least the imaginary part isn't considered. Thus, you only need one waveform array to store real, time-domain data. However, two additional arrays are required for storage of the corresponding frequency-domain data. In DPO TEK BASIC and WDI TEK BASIC, for example, the real

The FFT

waveform could be stored in array A. Then the FFT statement can be entered as FFT A,B,C. After the statement is executed, waveform arrays B and C contain the complex frequency-domain results for the FFT of array A.

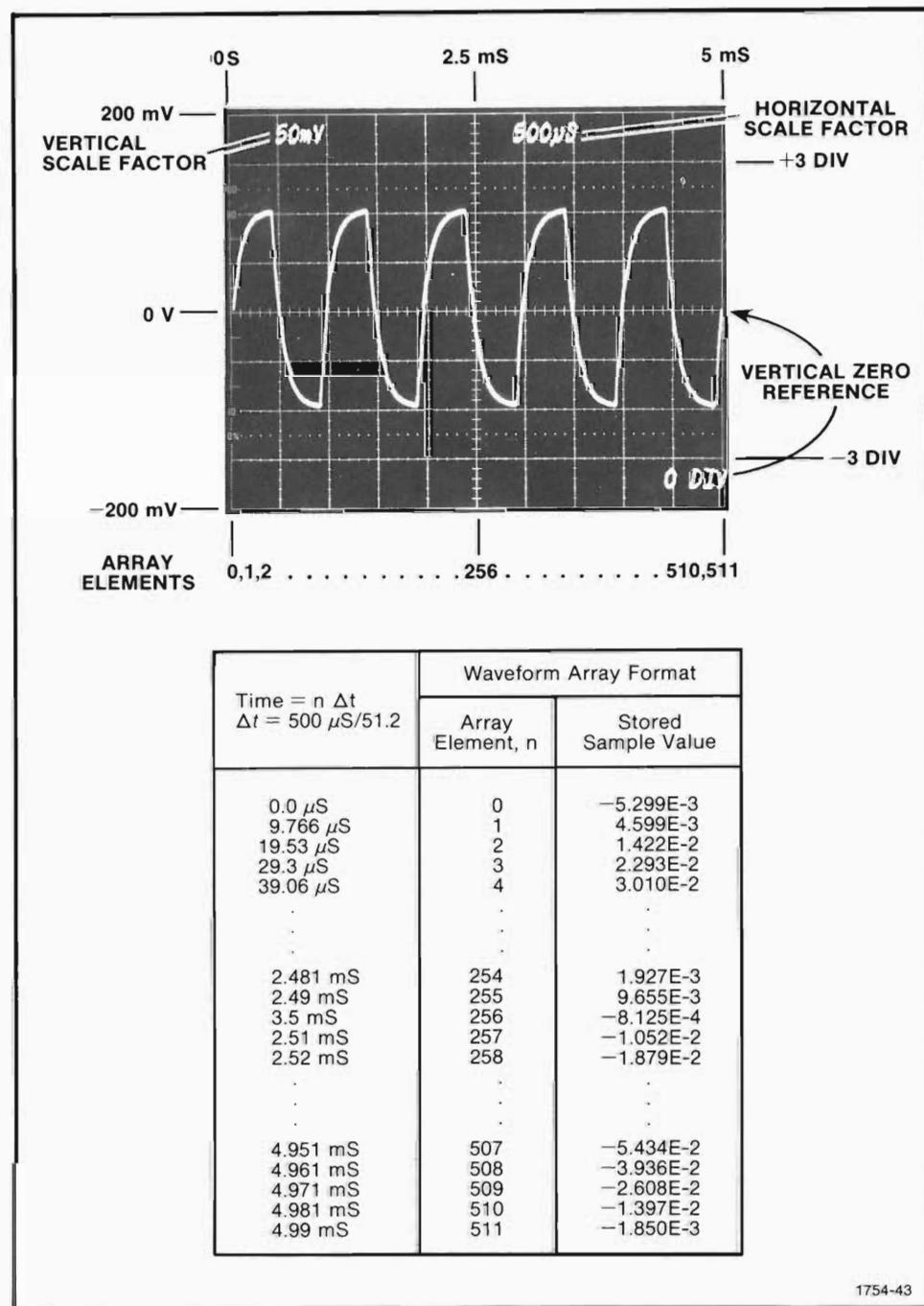


Fig. 4-1. Array format for time-domain waveforms.

The FFT

There are also instances where complex data may be acquired. Studying complex demodulation, for example, might require you to consider time-domain signals having both a real and an imaginary part. In order to transform complex signals by the FFT statement, you'll need to acquire the real part into one waveform array and the imaginary part into another waveform array. Then the complex form of the FFT, an option provided in DPO TEK BASIC and WDI TEK BASIC, can be applied. For example, the real part of the time-domain could be stored in array A and the imaginary part in array B. Then the FFT statement can be entered as

```
FFT A,B,C,D
```

The complex frequency-domain resulting from transforming arrays A and B are placed in arrays C and D.

FFT Results Are Normally in Rectangular Form

Let's take the time-domain waveform in Fig. 4-1 and transform it to the frequency domain with the FFT of DPO TEK BASIC. To begin, let's say the waveform is stored in array A. The cathode-ray-tube (CRT) display of this array is shown in Fig. 4-2a. This

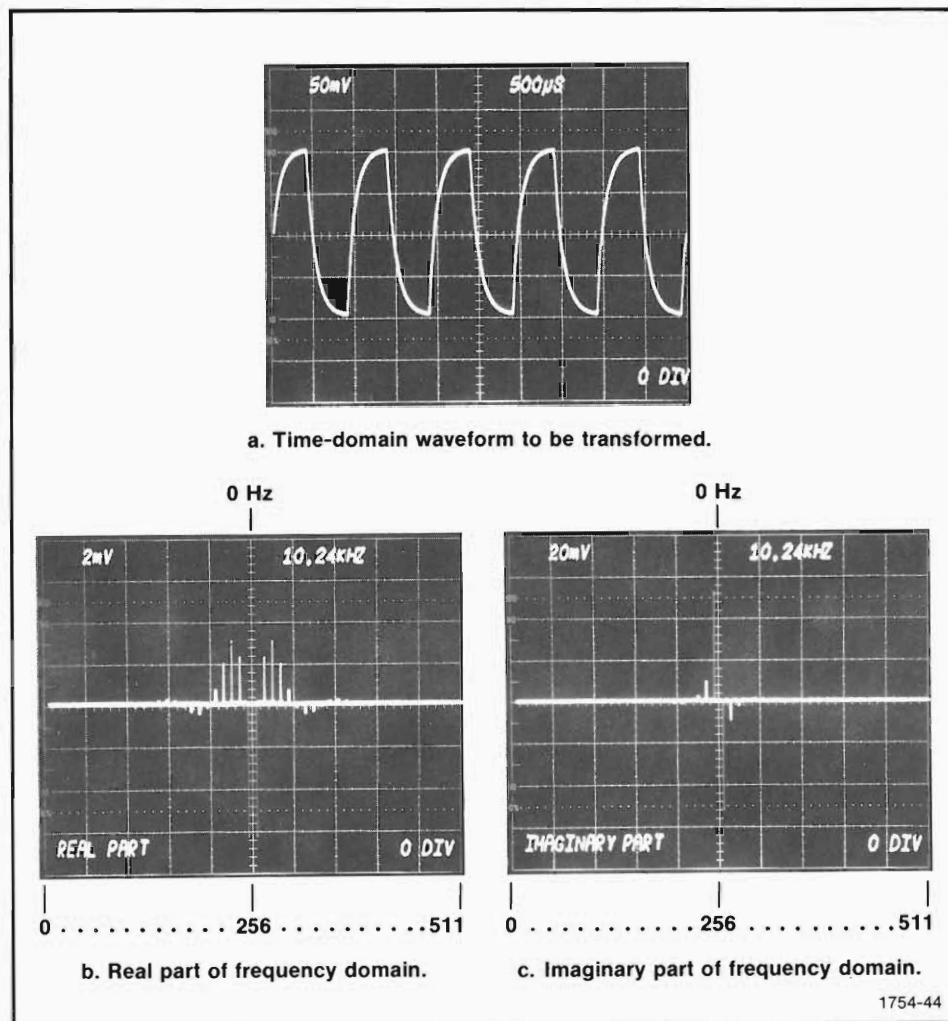


Fig. 4-2. The FFT of a time-domain waveform results in two arrays that represent the complex frequency domain in rectangular form.

The FFT

waveform array can be transformed to the frequency domain by entering

FFT A,B,C

from the Computer Display Terminal. The real part of the frequency domain is stored in array B and shown in Fig. 4-2b, and the imaginary part is in array C and is shown in Fig. 4-2c.

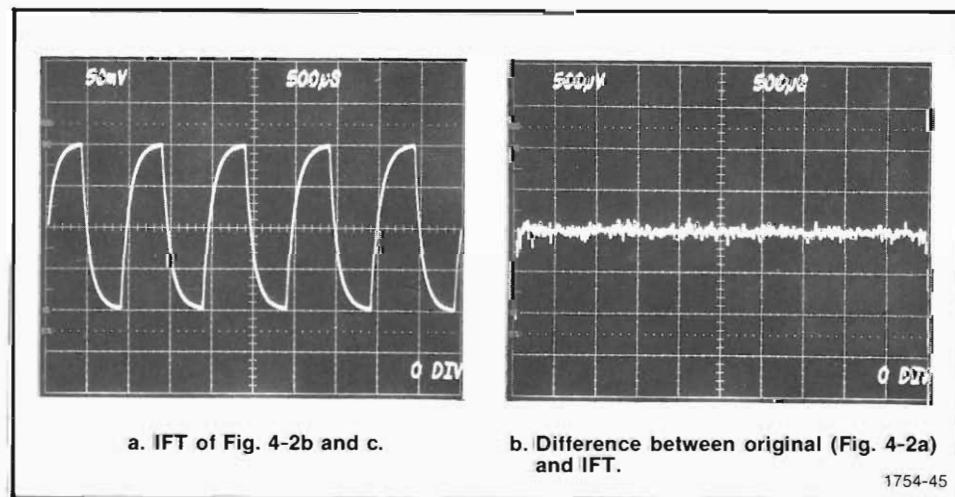
Notice how these results closely resemble the line spectrum of a periodic waveform. A periodic point of view follows naturally from Fig. 4-2a.

The arrangement of the Fourier coefficients in the two frequency-domain arrays is such that zero Hertz occurs at array element 256. This corresponds very closely to the center of the CRT display. The positive portion of the frequency domain extends over the elements from 257 through 511; this corresponds to the right half of the display. The negative frequency domain extends over the left half of the display and takes in array elements 0 to 256. Element zero corresponds to the highest frequency defined by the time-domain samples and is referred to as the Nyquist frequency. This arrangement is general to FFT results, no matter what point of view is taken. It also corresponds directly to the diagramming conventions in Section 2.

DPO TEK BASIC and WDI TEK BASIC also allow inverse transformation of frequency-domain arrays in the format of Fig. 4-2 back to the time domain. This is done by essentially the same FFT algorithm with only minor changes in the input and output routines. As an example, consider the two arrays of complex Fourier coefficients displayed in Fig. 4-2b and c. These arrays (B and C) can be transformed back to the time domain by entering

IFT B,C,D

The result of this inverse transform is shown in Fig. 4-3a. The difference between it and the original waveform in Fig. 4-2a is shown in Fig. 4-3b. Notice that this difference is slight—effectively zero compared to the waveform magnitudes—and can be attributed to arithmetic roundoff error.



1754-45

Fig. 4-3. The IFT of the frequency domain in rectangular form returns the original, time-domain waveform.

The FFT

FFT Results Can Be in Polar Form

The rectangular form is the most direct result for the FFT and is also more appropriate to further processing. But most people are not entirely comfortable viewing FFT results in rectangular form. Maybe it's because first introductions to Fourier theory and the frequency domain are through magnitude and phase, the polar form. Widespread use of spectrum analyzers may also account for this seeming preference since they present frequency-domain information in the form of magnitude (spectrum analyzers using analog techniques don't provide phase information). Whatever the reasons, people seem most comfortable viewing FFT results in polar form. And, if you are simply interested in the magnitude of a specific component, then the polar form is naturally desirable.

Converting FFT results from rectangular form to polar form is a simple operation. Magnitude is obtained from the real and imaginary parts by taking the square root of the sum of the squares (magnitude = $\sqrt{Re^2 + Im^2}$). Phase is the arctangent of the imaginary part divided by the real part (phase = $\arctan Im/Re$).

Modulo 2π Phase. For convenience, DPO TEK BASIC and WDI TEK BASIC have a polar form for the FFT statement. For example, if you have time-domain data stored in array A and wish to see frequency-domain magnitude in array B and phase in array C, then enter the following statement from the Computer Display Terminal:

```
FFT A,B,C,POLAR
```

The FFT algorithm itself isn't changed by the added POLAR argument. Intermediate results are still in rectangular form, but the POLAR switch causes software to go ahead and convert the results to polar form. For the example statement, magnitude is stored in array B and phase in array C. The format of these two arrays is the same as described for the rectangular form except vertical scaling is in terms of magnitude and phase. (It should be noted here that the arctangent routine used by the POLAR switch returns values in the range of 2π radians— $+\pi$ to $-\pi$. Thus, the phase resulting from the POLAR switch is referred to as modulo 2π phase. In WDI TEK BASIC, an additional argument can be used after the POLAR switch to construct "continuous" phase. When this delay argument is used, module 2π phase is converted to continuous phase, and the delay specified by the argument is removed from the phase function.)

An example of FFT polar results is shown in Fig. 4-4. The time-domain waveform in Fig. 4-4a is the same waveform used for the rectangular-form example. The polar-form results from FFTing this waveform are shown in Fig. 4-4b and c. The magnitude portion is fairly clear and is what would be expected for a low-pass filtered square wave. It closely resembles a line spectrum; thus, it is appealing to the continuous, periodic point of view. However, the phase portion in Fig. 4-4c does need some further explanation.

First of all, phase can't exist for a frequency component that doesn't exist. Yet, if we view Fig. 4-4a as a periodic waveform and interpret the FFT magnitude results in Fig. 4-4b as a line spectrum, Fig. 4-4c seems to say the opposite. There appears to be phase

The FFT

in Fig. 4-4c where magnitudes in Fig. 4-4b appear to be zero. In reality, though, the magnitude spectrum in Fig. 4-4b is not discrete in the sense of the periodic point of view. Also, the real and imaginary spectra (Fig. 4-2b and c), from which the magnitude and phase were computed, are not line spectra either. All of these spectra have low-level "noise" components residing between the higher-level harmonics. For interpreting the results from a continuous, periodic point of view, the low-level components are low enough to be considered zero. However, from a computing standpoint, these low-level components still exist, and they are significant in terms of computing phase. For example: A value of 10^{-6} might, for the purpose of interpretation, be considered zero when compared to a value of one. But, for the purpose of computation, $\arctan 1/1$ and $\arctan 10^{-6}/10^{-6}$ are the same—they both are equal to 45 degrees (0.7854 radians).

So even the smallest components can contribute significantly to phase computations. This is desirable when you are looking for information that is contained in low-level components, but for the present example, it may lead the uninformed to confusion.

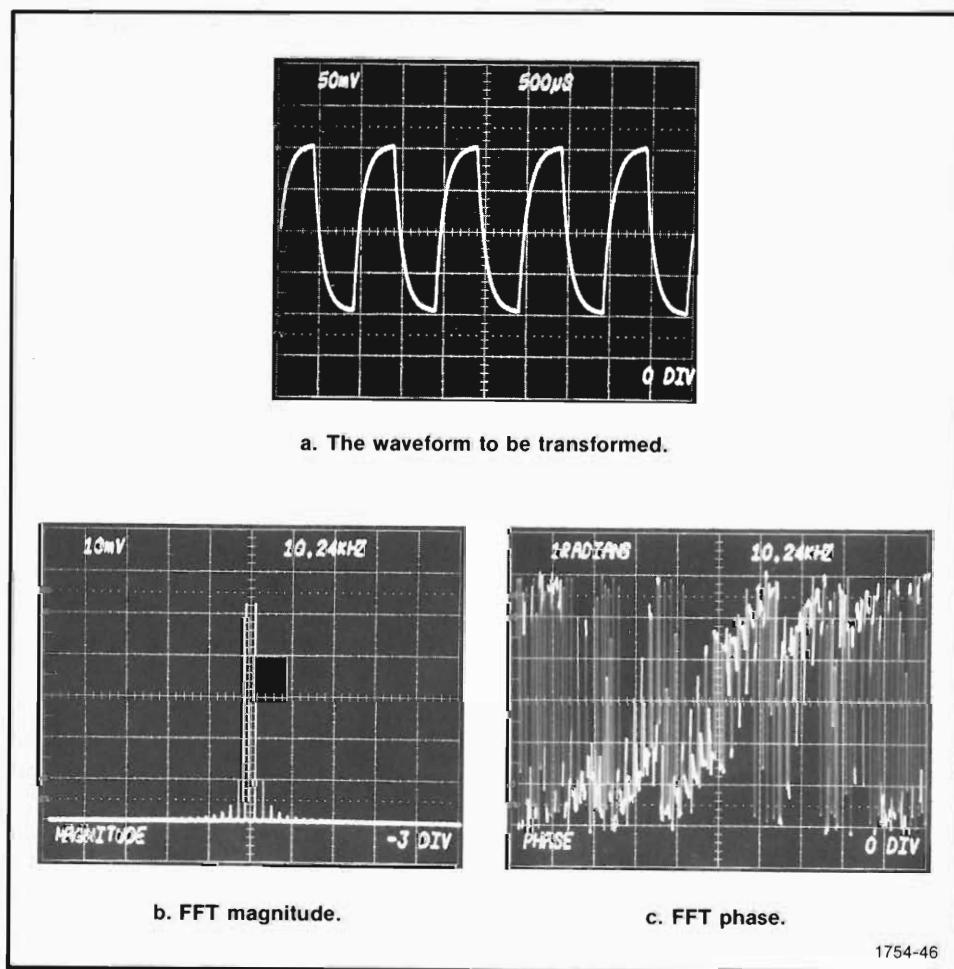


Fig. 4-4. FFT results can be expressed in polar form.

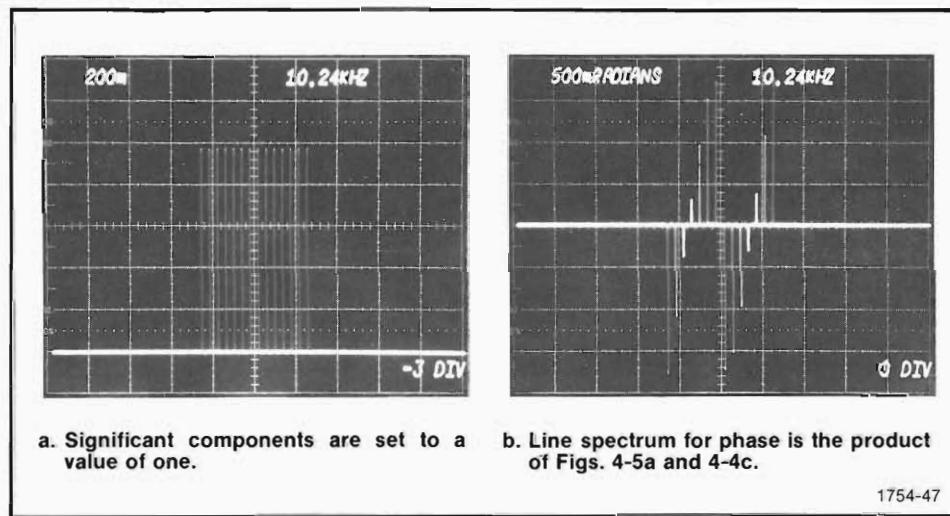
The FFT

How do low-level components come about? Well, in reality, Fig. 4-4a doesn't show an ideal, continuous, periodic waveform. The waveform used for this example was an actual, physical waveform and was subject to distortions, noise, and all of the other vagaries of reality. So some of the low-level components were actually present in the waveform. Adding to this, the waveform was windowed and sampled and the samples were digitized. Then the digitized samples were transformed by an FFT algorithm, and this FFT algorithm was executed by a physical computing system with physical limitations. In their own way, all of these things may contribute additional low-level components.

As you can see, a lot of things separate the realities of measurement and analysis from the niceties of theory. Most of these things will be discussed shortly in greater detail, but for now, let's just accept the fact that they all contribute to the FFT results.

Still, the filtered square wave in Fig. 4-4a is certainly appealing to a continuous, periodic point of view. However, to take a periodic point of view for interpreting phase results, you must consider phase to be valid only at the frequencies of interest. With that in mind, the significant frequency components can be picked from the magnitude spectrum. Then they can be placed in an array and given a value of one, while all other array elements are given a value of zero. The resulting array looks like Fig. 4-5a. Then, by multiplying Fig. 4-5a with the phase in Fig. 4-4c, a line spectrum for phase is obtained. This line spectrum for phase in Fig. 4-5b and the magnitude spectrum in Fig. 4-4b are what you want to look at if you wish to interpret the frequency domain of the filtered square wave from a periodic point of view, a point of view relating to the Fourier series.

But, as has been pointed out in previous discussions, a nonperiodic point of view may be appropriate for many analysis situations. Take Fig. 4-6a for example. The pulse shown there certainly appeals to a nonperiodic point of view. You would probably want to interpret its frequency domain in terms of what would be expected from the Fourier integral.



a. Significant components are set to a value of one.

b. Line spectrum for phase is the product of Figs. 4-5a and 4-4c.

1754-47

Fig. 4-5. Interpreting phase from a periodic point of view requires looking only at the significant frequency components.

The FFT

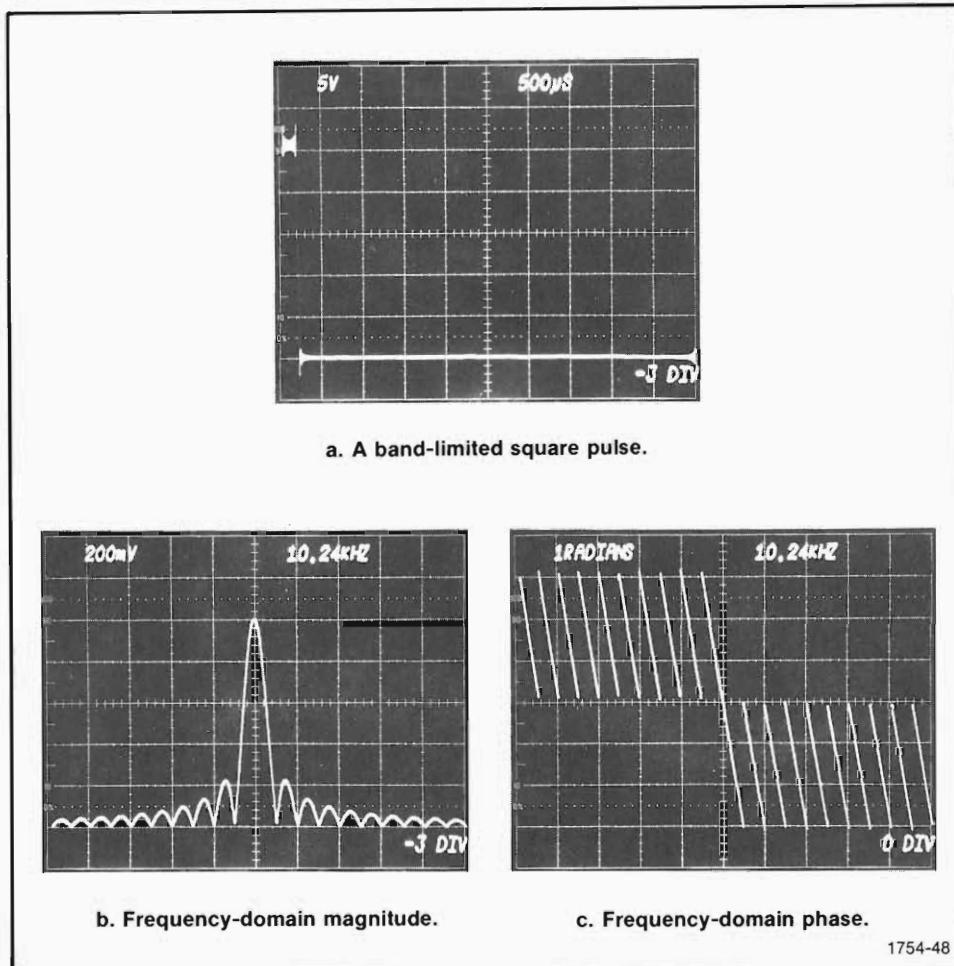


Fig. 4-6. A case where the nonperiodic point of view is a natural choice and agrees with the FFT results.

The frequency domain of the band-limited pulse in Fig. 4-6a is shown in Fig. 4-6b and c. In particular, notice the phase display in 4-6c. It is well ordered, not like Fig. 4-4c, and certainly appears to agree with what is expected from the nonperiodic point of view. In fact, compare the frequency domain of Fig. 4-6 with the theoretical frequency domain of the Fourier integral in Fig. 2-11c. Notice how closely they match.

This last example certainly supports the nonperiodic point of view for interpreting DFT and FFT results. And this should be expected since the DFT is derivable from the Fourier integral. But remember that the Fourier integral is also derivable from the Fourier series. So the periodic point of view of the Fourier series is also applicable to the DFT and FFT as long as you take into account the signal modifications occurring through the chain of derivation from the series, through the integral, to the DFT.

Continuous Phase. Up to this point, phase has only been discussed in terms of modulo 2π phase; that is, phase has been restricted to a range of $\pm\pi$. This is sufficient for most interpretations, but phase can in theory and practice exceed the $\pm\pi$ range. This is shown in Fig. 4-7.

The FFT

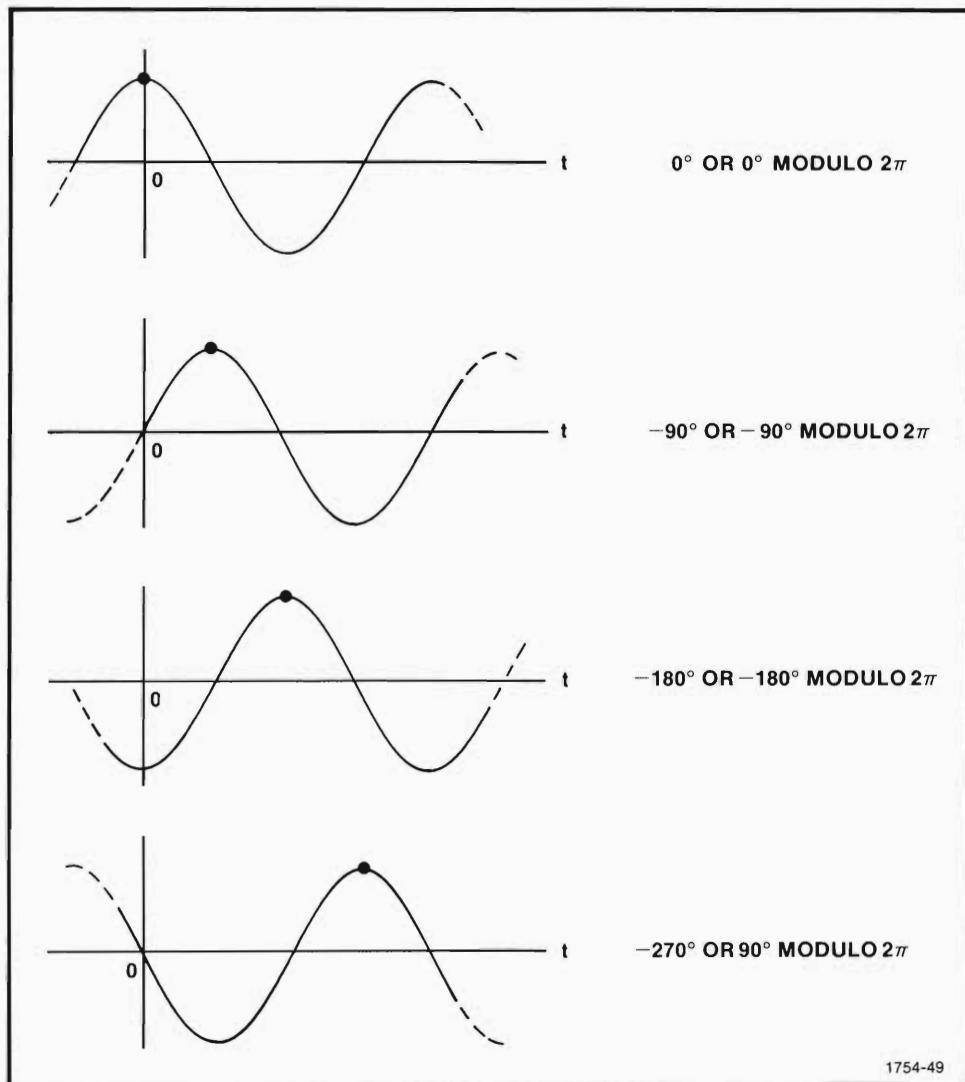
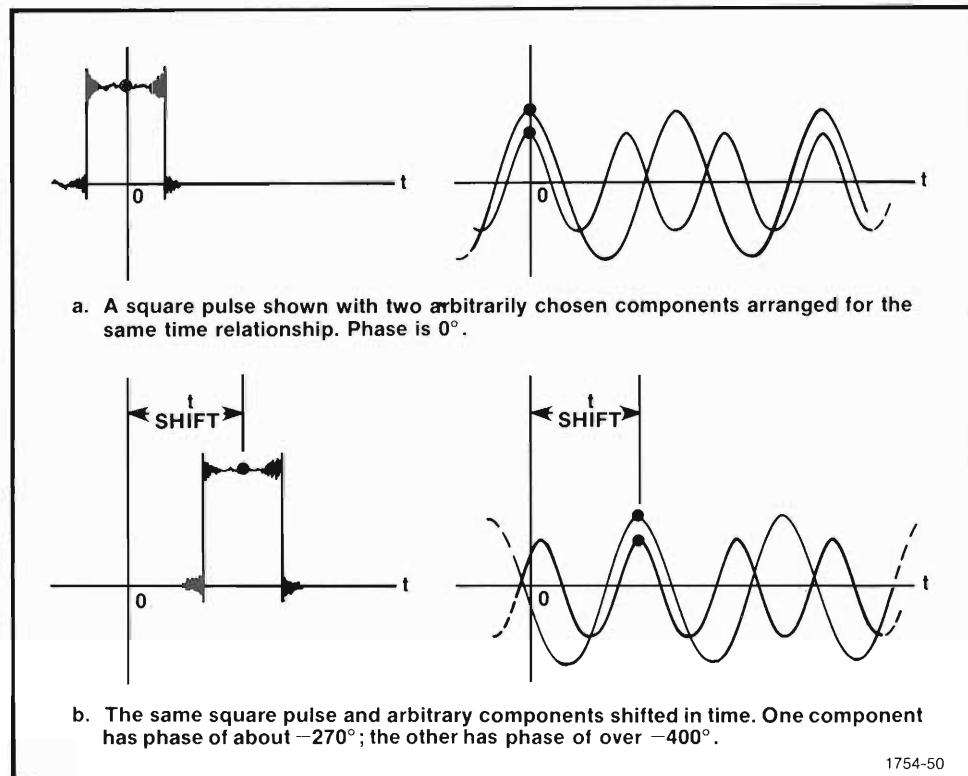


Fig. 4-7. Specifying phase depends upon your reference. If the reference is fixed to a specific point on the shifted waveform, then phase related to time zero can exceed the $\pm\pi$ range.

In terms of analyzing waveforms made up of many sinusoids, the phase of each frequency component is determined by the choice of zero time. In Fig. 4-8a, a pulse is shown arranged with time zero so that the pulse is an even function of time. The pulse is symmetric about time zero; therefore, it's made up solely of cosine terms. A heavy dot marks a reference point fixed to the pulse. The same reference is marked on two sinusoidal components in Fig. 4-8a that have arbitrarily been pulled from the pulse.

Now, as the pulse is shifted in time (Fig. 4-8b), each frequency component undergoes the same time shift. The amplitude and time relationship of each component to the others is unchanged, for they must all still add up in the same manner to form the pulse. However, in terms of phase relation to time zero, each component must vary individually. This is because each component is of a different period, but each is time shifted by the same amount. And phase can be expressed as the ratio of time shift to the component's period (phase = -360° shift/period). As indicated by the two arbitrary

The FFT



1754-50

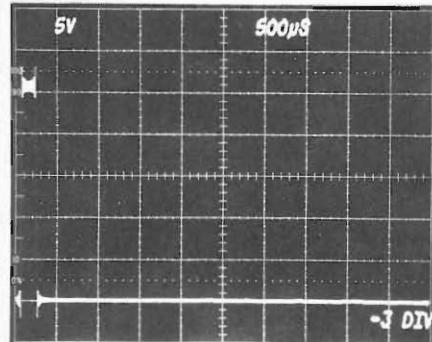
Fig. 4-8. The same time shift corresponds to different phase shifts for different frequencies.

components in Fig. 4-8b, a higher frequency component, because of its shorter period, gains phase at a greater rate than one of lower frequency.

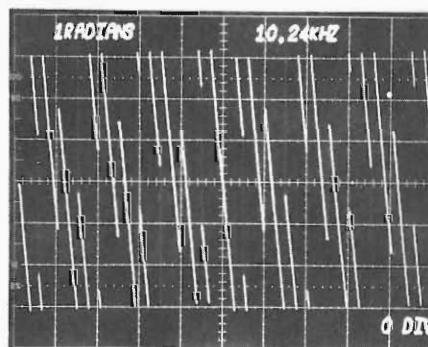
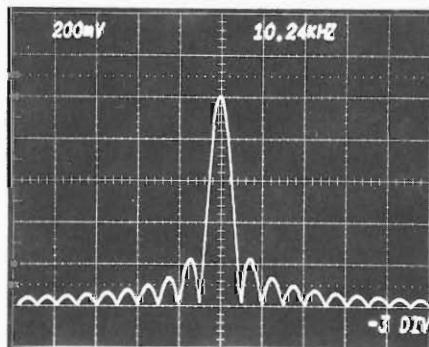
The pulse shown previously in Fig. 4-6 provides a good example for illustrating the difference between modulo 2π phase displays and continuous phase displays. This pulse is shown again in Fig. 4-9a with its position shifted slightly in time. Since time zero is at the left edge of the display, the additional shift in Fig. 4-9a should be reflected in the frequency domain as an increase in each component's phase.

Fig. 4-9b shows the frequency-domain magnitude and modulo 2π phase for the pulse in Fig. 4-9a. This phase display is certainly different from that in Fig. 4-6, but notice that it's still confined to a range of $\pm\pi$ radians. Fig. 4-9c shows the same frequency domain as Fig. 4-9b; however, notice that phase exceeds the $\pm\pi$ range. Here, phase is shown in a continuous format, not interrupted and repeated every $\pm\pi$. Even more interesting is the fact that the phase in Fig. 4-6, even though it was computed modulo 2π , would look no different computed as continuous phase. That's because of the pulse's location relative to time zero in Fig. 4-6—the time shift, or delay, there is not great enough to cause phase to exceed $\pm\pi$. However, with the increased time shift in Fig. 4-9a, phase does exceed the $\pm\pi$ range. (Continuous phase displays can be obtained with WDI TEK BASIC by entering an FFT statement of the form: FFT A,B,C,POLAR, θ .)

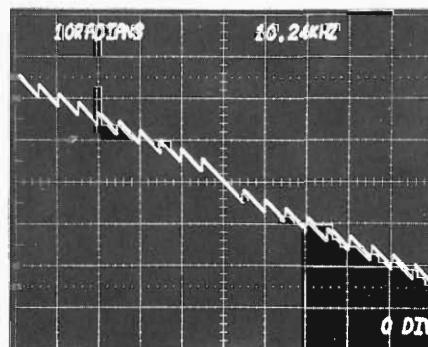
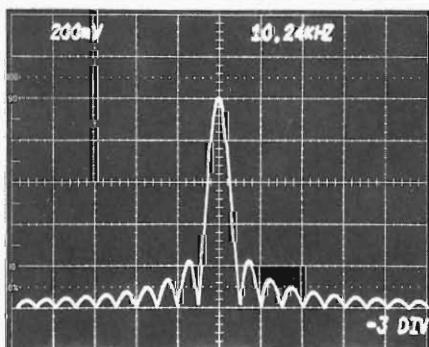
The FFT



a. A square pulse delayed slightly from time zero.



b. The frequency domain in terms of magnitude and modulo 2π phase.



c. The frequency domain in terms of magnitude and continuous phase.

1754-51

Fig. 4-9. Modulo 2π phase and continuous phase are two different ways of looking at the same thing.

The FFT

The important thing to realize here is that modulo 2π phase and continuous phase are nothing more than two means of presenting the same information. Continuous phase presents the effects of time shifting (advance or delay) directly— 360° of phase is represented as 360° of phase. On the other hand, modulo 2π phase views time shifting in the sense that a sinusoid with 0° phase (a cosine wave) appears no different than one shifted 360° . Thus, phases of 360° , 720° , ... are all represented as 0° in modulo 2π phase. This same idea applies to all phases in excess of $\pm\pi$ radians. They can be represented in the $\pm\pi$ range without changing the appearance of the sinusoid around time zero.

As a final note, there is another interesting thing that should be pointed out in Fig. 4-6 and 4-9. This isn't something new, but rather something in the manner of reviewing and reinforcing some theory covered in Section 2. Look at the pulses in Fig. 4-6a and 4-9a. The only difference between them is their location in time. Now look at the frequency-domain magnitudes and phases in Fig. 4-6b and c and 4-9b. Notice that there's absolutely no difference in the two magnitude displays. The phase displays, however, are markedly different! This points out again the Fourier transform property that time shifting only affects phase angle.

RECOGNIZING THE REALITIES OF DIGITAL FOURIER ANALYSIS

When an analysis technique is based heavily on theory, it is natural to let theory bias interpretations of the results. This is exactly as it should be, as long as theory is followed completely and applied precisely. Too often, though, results are interpreted in light of only the more general concepts. A rush to fill in the big picture leaves the details behind, and only a rough sketch is the result. And, too, there is often a tendency to categorize things—to place them under neat labels allowing easy prediction of results. For example, a 10 ohm resistor is only a 10 ohm resistor for convenience. If fits neatly into Ohm's law, $E=IxR$, but in reality "I" may be a high-frequency current and "R" a wire-wound resistor. Then the voltage drop, E, must be interpreted in light of the whole theory of conduction—capacitances, inductances, and thermal effects must be considered, too.

Failing to be thorough and precise in using theory to predict or interpret results causes analysis errors. Sometimes the errors are actually in the results, but often the errors are simply misinterpretations of the results.

In the case of the FFT algorithm, theory is applied thoroughly and precisely to the analysis of digitized waveforms. For the data supplied, the results are what they should be. And, when the results aren't what we think they should be, it is most often a case of incomplete interpretation. Some feature of the waveform, maybe even a subtle one, has been overlooked. Or possibly some of the properties of windowing and digitizing have been overlooked or their implications not fully understood. Whatever the case, most FFT errors are easily explained or corrected by careful attention to detail.

The FFT

There are two major classes of details that you should be aware of when you interpret FFT results. The first class of details concern the waveform itself. What is being transformed? The second class of details concern what happens to the waveform in preparing it for transformation by the DFT or the FFT algorithm. What are the effects of changing an analog waveform to digital data?

Let's look at these two classes of details and explore their significance in terms of the FFT.

What Is Being Transformed?

Square Waves May Not Be Square Waves. When we look at a real-life waveform, we can often predict at least some of its frequency-domain features before actually transforming it to the frequency domain. For example, if it's a repetitive waveform, its period fixes the fundamental frequency. And, if it's nonsinusoidal, we know frequencies other than the fundamental will be present. Many of these may be harmonics, multiples of the fundamental.

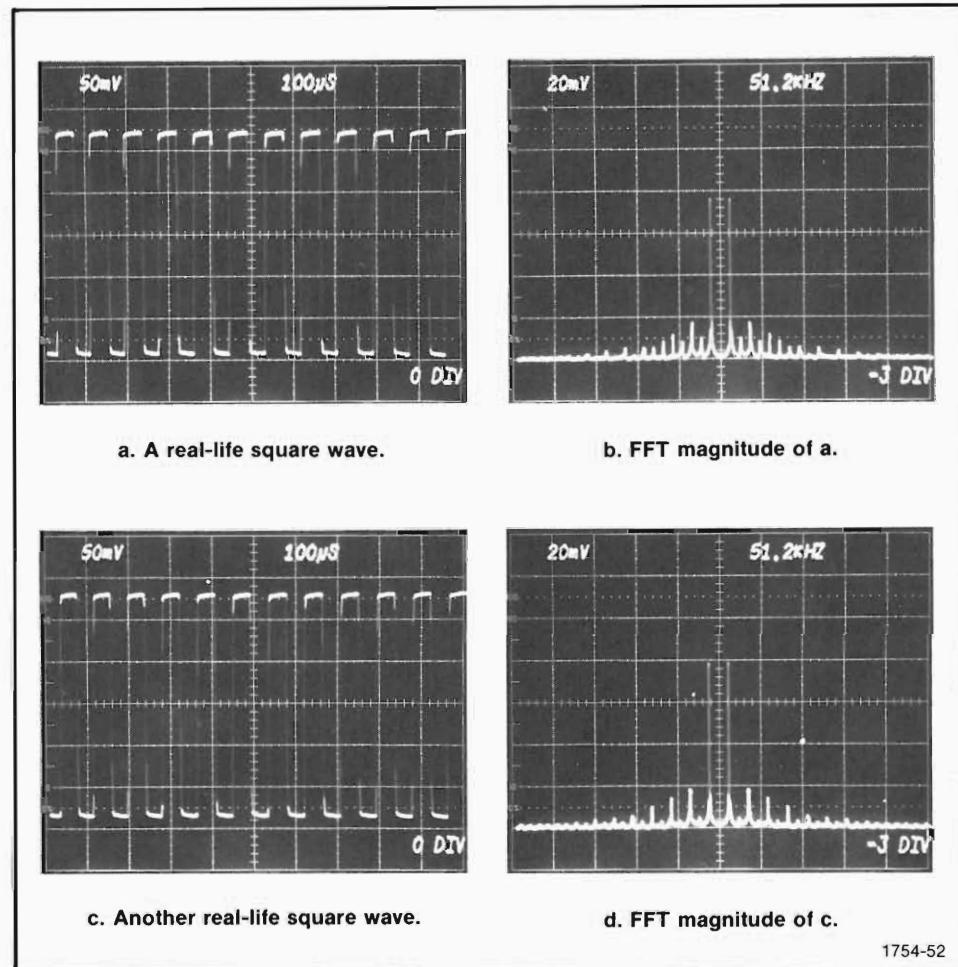
Of course, if the waveform looks a good deal like a standard waveform, much more can be predicted. For example, if it looks like a square wave, we can go to Table 2-1 in Section 2 and see what frequency-domain components the Fourier series defines. From this, we should expect to see the fundamental and the odd harmonics when the waveform is transformed. And, because we know the Fourier series for a square wave, we can also say something about the expected amplitudes of these harmonics.

However, when the waveform is transformed, we shouldn't be surprised if the results don't exactly match our predictions. In the case of a waveform that looks like a square wave, predictions based on the Fourier series for an ideal square wave should not be taken too seriously. There simply are no ideal square waves in real life. Instantaneous rise and fall times, perfect symmetry, and absolutely steady amplitudes, all of the things that make an ideal square wave, aren't fully attainable in real-life waveforms. The distortions of real life, no matter how small, do affect the frequency domain.

Look at the waveform in Fig. 4-10a for example. It looks like a square wave. We could make some predictions based on the Fourier series for an ideal square wave. But look at the waveform again. How close does it really come to being an ideal square wave?

If you look close enough, you can see that the waveform in Fig. 4-10a is really not an ideal square wave. This is confirmed by looking at its frequency-domain magnitude, obtained by the FFT, in Fig. 4-10b. There the fundamental frequency (closest to the center and greatest in amplitude) has a frequency of about 12 kHz. With care, the third harmonic can be picked out at 36 kHz, the fifth harmonic at 60 kHz, and so on. But, in doing this, you'll notice that there are significant components between these odd harmonics. And, in fact, these extra components are even harmonics. If the waveform in

The FFT



1754-52

Fig. 4-10. Barely perceptible differences in time-domain waveforms can often be seen as significant differences in the frequency domain.

Fig. 4-10a was an ideal square wave, these even harmonics wouldn't exist. However, they do exist because the waveform in Fig. 4-10a is not symmetric. The duration of the positive-going portions is slightly longer than the duration of the negative-going portions.

Now look at the waveform in Fig. 4-10c. Except for a slight change in amplitude, it really doesn't look too much different than the waveform in Fig. 4-10a. But look at the difference in the frequency-domain magnitude (Fig. 4-10d). The odd harmonics are now much clearer; the even harmonics have dropped significantly. The even harmonics are still visible, however, and their presence indicates that the waveform in Fig. 4-10c is still not quite symmetric.

The whole thrust of Fig. 4-10 is that real-life waveforms may look close to ideal, but they still cannot always be considered ideal. For the purposes of general description, it is convenient to call the waveforms in Fig. 4-10 square waves. For the purposes of Fourier analysis, however, you must be very precise in how you define a waveform. If you are not precise, if you predict results on the basis of the ideal instead of the real, you

The FFT

might be surprised by the FFT results. And if you are ever surprised by the frequency-domain results, go back and look closely at the original time-domain waveform. Ask yourself: "What, exactly, is being transformed?"

Time-Domain Noise Transforms to Frequency-Domain Noise. Noise is a constantly present physical phenomenon. It has a multitude of sources that combine to produce an ambient condition of interference. And, in the case of generating and acquiring waveforms, the effects of additive noise cannot be discounted.

Quite often, proper shielding can reduce the effects of noise, but there are practical limits to shielding. There is always a certain amount of leakage, and where low-level signals are concerned, this leakage may contribute significantly to signal degradation. Also, there are those cases where noise is added to the signal before shielding can be applied. Consider radar and sonar transmissions. The transmitters and receivers can be shielded, but little can be done to prevent noise addition over the transmission path.

When a noisy signal is acquired for Fourier analysis, Fig. 4-11a for example, you must consider the noise as part of the signal. Often, noise is additive, and the acquired signal is the sum of the noise-free signal and the noise. In terms of the FFT, you should expect to see the transform of the noise-free signal summed with the transform of the noise (Fig. 4-11b). This is in keeping with the linearity property of the Fourier transform.

What constitutes an intolerable level of noise depends upon what you're looking for. If you're only looking for the general shape of things, then quite a bit of noise can be tolerated. In Fig. 4-11a and b, for example, the general shape of the time-domain waveform and its frequency-domain magnitude is discernable. On the other hand, this same level of noise is intolerable when more precise information is desired. Then something like the waveforms in Fig. 4-11c and d are needed.

Noise is still present in Fig. 4-11c and d, but its level with respect to the signal is certainly much more tolerable. In the case of transient events, such improvements in signal-to-noise ratio must be achieved through shielding. However, when the signal is repetitive and the noise is random, signal averaging can improve the signal-to-noise ratio substantially. In the case of Fig. 4-11, the time-domain waveform is a single pulse from a train of pulses, and the improvement shown in Fig. 4-11c was obtained by signal averaging. When the averaged waveform is transformed to the frequency domain, the same type of improvement is seen there (Fig. 4-11d).

The technique of signal averaging is quite straightforward. A repetitive signal is acquired a number of times, and each acquisition is added to the last. Then the sum is divided by the number of acquisitions. The result is an average of the acquired signals in the manner indicated by Fig. 4-12.

Now, since ambient noise tends to be random and to have a zero mean, the contribution of noise to the average is reduced. And, since the signal of interest is repetitive, averaging strengthens its contribution. As more acquisitions are averaged,

The FFT

the noise contribution is further reduced and the signal further reinforced. For truly mean-zero noise, the improvement in signal-to-noise ratio for M averages is \sqrt{M} . When M is expressed as a power of two, the improvement corresponds to 3 dB per power of two. For example, $2^7=128$ averages corresponds to a 7×3 dB = 21 dB improvement in the signal-to-noise ratio. If more improvement is needed, the number of averages (M) is increased.

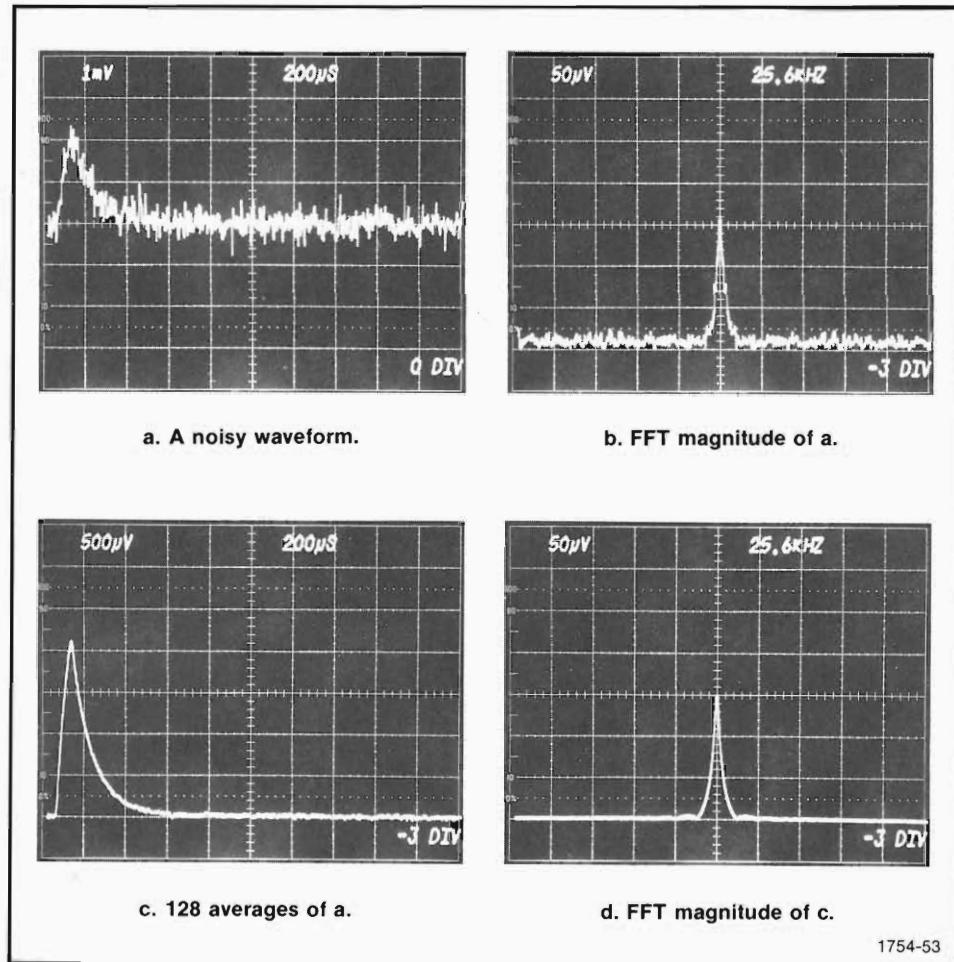


Fig. 4-11. Noise adds uncertainty to both the time domain and frequency domain. This uncertainty can be reduced by proper shielding and, in the case of repetitive signals, signal averaging.

The FFT

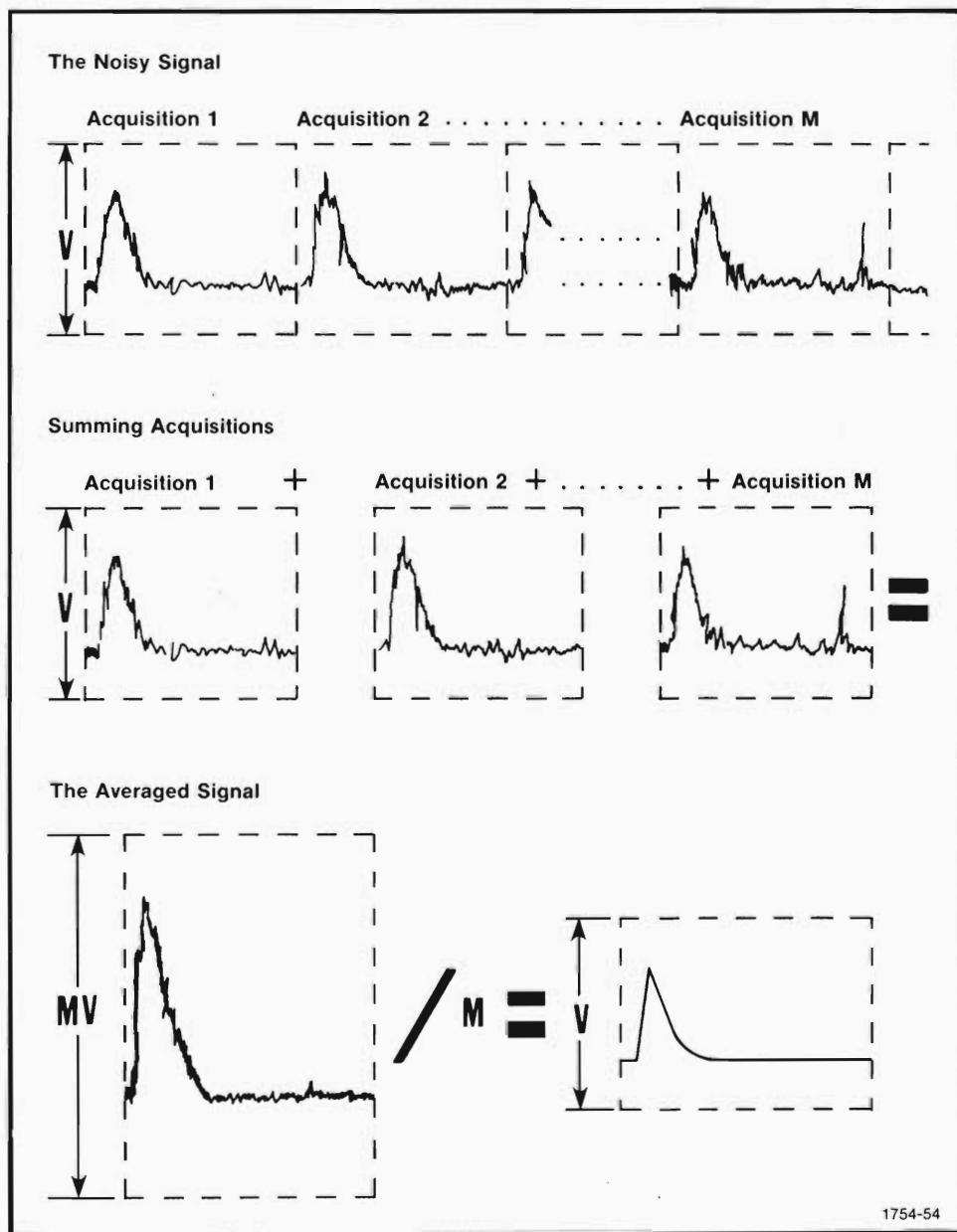


Fig. 4-12. Signal averaging is a straightforward technique for pulling repetitive waveforms out of random, mean-zero noise.

The FFT

What Are the Effects of Analog-to-Digital Conversion?

When an analog waveform is to be transformed to the frequency domain by the FFT, it must first be converted to a digital representation. The steps in this conversion include acquiring the waveform, sampling it through a data window, and finally converting the samples to digital words representing the waveform. If the FFT results are interpreted strictly from the viewpoint of the digitized samples, the results are exact. However, it is more likely that you'll want to interpret the results from the viewpoint of the analog waveform. Then the changes that the waveform undergoes in analog-to-digital conversion must be considered in the interpretation.

Conversion Noise. For the most part, noise generated by the analog-to-digital converter can be virtually eliminated through proper design procedures. There are two major noise sources, however, that are more closely process related than hardware related. These are time jitter during acquisition and quantizing error during digitizing.

The first of these two, time jitter, occurs when the analog waveform activates a level trigger that gates the sampling window on. For various reasons, subsequent windows of the same waveform may be triggered at slightly different points on the waveform. This results in the waveform losing horizontal (time) stability in the window. This jittering back and forth is sometimes seen on oscilloscope displays (Fig. 4-13a) when the oscilloscope trigger level is set at an unstable point on the waveform or waveform variations (noise for example) cause the trigger point to shift on the waveform.

For single-shot events or sampling techniques where the waveform is digitized in a single window, trigger jitter isn't generally a problem. A slight time shift in the window simply translates to the same time shift in each sample location on the waveform. However, if individual samples become jittered within the window, a problem does exist for single-shot events. This is best handled through careful digitizer design.

Time jitter can also be a problem, when repetitive waveforms are being acquired and digitized from samples taken over a number of windows. Then the effect of jitter is to cause some samples to be shifted on the waveform with respect to other samples. This causes the stored waveform to take on the appearance shown in Fig. 4-13b.

Notice in Fig. 4-13b how the sampling error from jitter resembles the random noise shown in Fig. 4-11a. And, if Fig. 4-13b were transformed to the frequency domain, the effect would be the same as that caused by additive noise. Although the amplitude of variations due to jitter are related to signal slope (dv/dt), the variations are often random in occurrence, like additive noise, and apt to have a zero mean. Because of these characteristics, signal averaging can be used to reduce the effects of time jitter. This is shown in Fig. 4-13b and c, where 512 averages were used to gain the signal-to-noise improvement of Fig. 4-13c.

There is another type of conversion noise that you should know about, too. This is quantizing noise, also referred to as quantizing error. Like general additive noise and

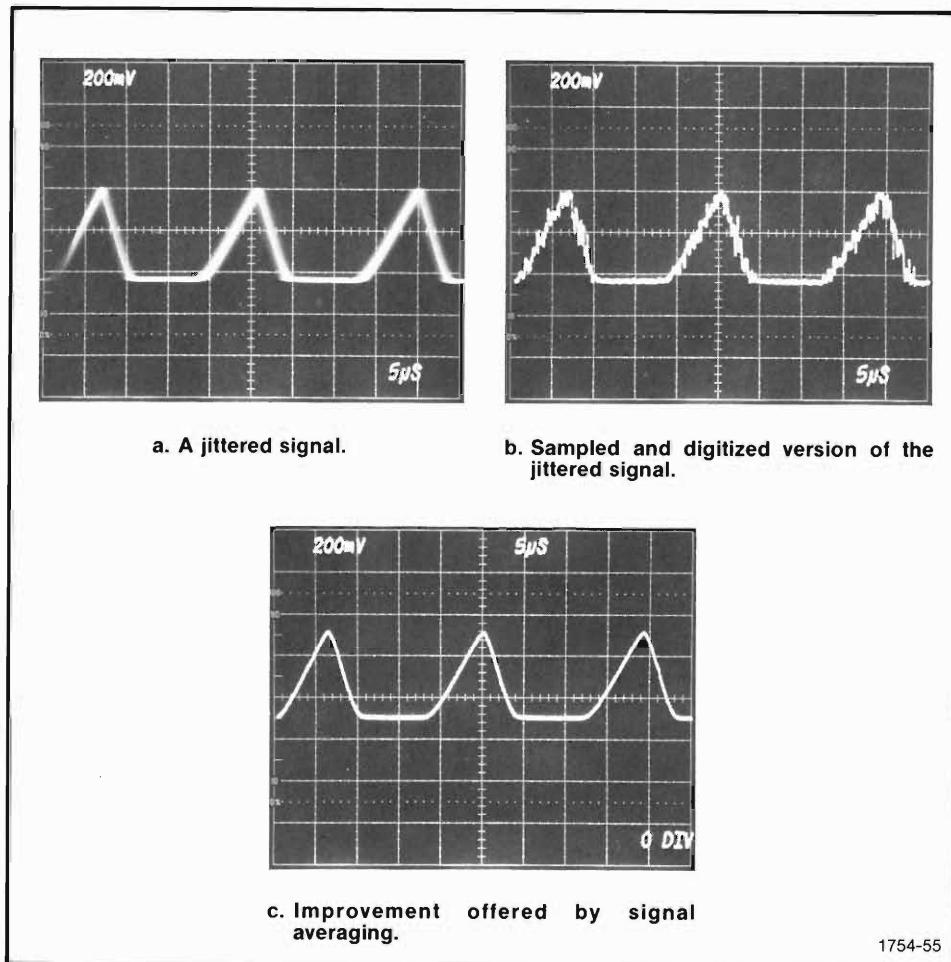


Fig. 4-13. Time jitter often has a random, noise-like effect that can be reduced by signal averaging.

time jitter, quantizing noise tends to be random and mean zero. Because of this, it too can be reduced under certain circumstances by signal averaging.

To understand the source of quantizing noise, let's consider digitizing a ramp portion from a waveform being acquired with the DPO. This is shown in Fig. 4-14.

The DPO takes 512 samples along the horizontal axis of the acquired waveform. Only a few of these horizontal sample points are indicated in Fig. 4-14. The digitizer used by the DPO is a 10-bit digitizer. That means that there are $2^{10}=1024$ digital levels available at each sample location for expressing the digital value of the sample. These vertical levels are also indicated in Fig. 4-14. They, along with the horizontal sample points, can be considered to compose a grid work, the points of which represent the allowable values for describing the waveform.

There are ramp portions from two waveforms shown on the digitizing grid in Fig. 4-14. Notice that the top ramp has values falling exactly on the available digital levels. When this ramp is sampled and digitized, the digital representation exactly matches the

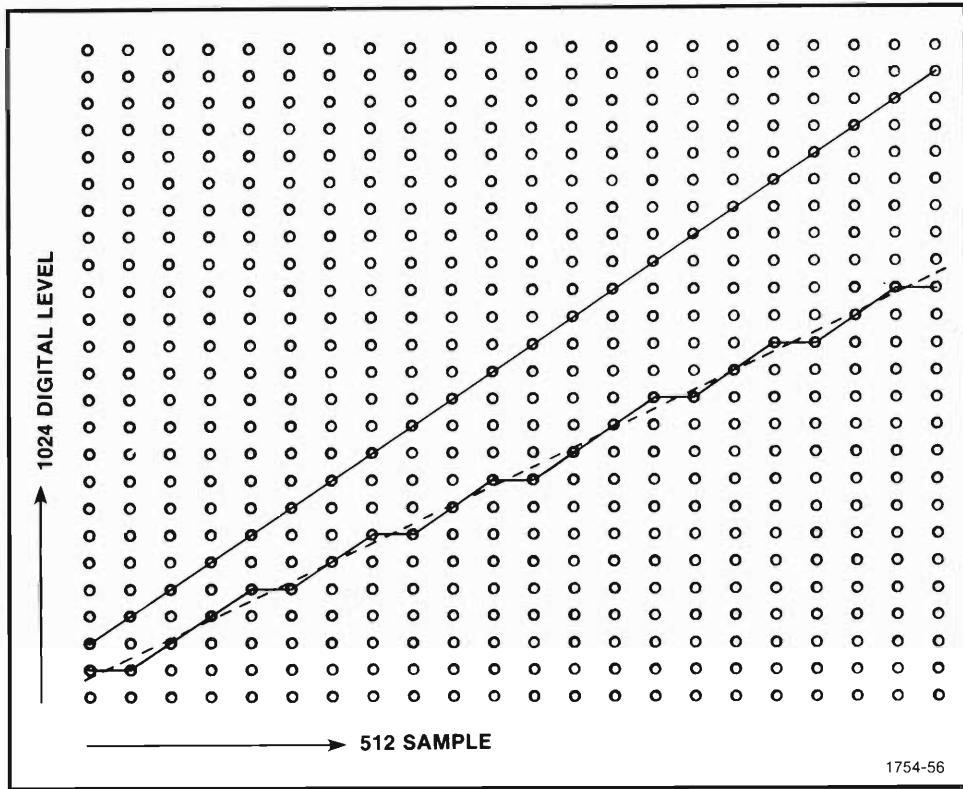


Fig. 4-14. Quantizing error is inherent in digitizing. The top ramp lies on the digital levels. The bottom ramp (dotted) lies between digital levels, and its digital description (solid line) suffers quantizing error.

ramp. Now, look at the bottom ramp in Fig. 4-14 (dotted line). In a number of places, its samples fall between the allowable digital levels. When it is digitized, the digital representation of this ramp follows the erratic path shown by the solid line. The slight deviations from the dotted line are quantizing error and are inherent in any digitizer of finite word length. The amount of this error at any sample point is no more than half of a digital level and occurs randomly over the block of waveform samples. In short, quantizing error is noise-like and is therefore referred to in many instances as "quantizing noise".

Since quantizing noise doesn't exceed half a digital level, it is generally low-level compared to other types of noise. However, it can become high-level noise. Consider acquiring a waveform so it ranges over only 100 of the available 1024 digital levels of a 10-bit converter. The quantizing error is then half a digital level out of the 100 levels used. Now consider acquiring the waveform so it ranges over all 1024 levels. Then the error from quantizing noise becomes half a digital level out of the 1024 levels used. A more than ten times reduction in the relative effect of quantizing noise is achieved in this case by simply exercising the full range of the analog-to-digital converter. With the DPO and R7912 Transient Digitizer, this is done by adjusting the sensitivity of the vertical plug-in so the instrument display area is filled by the waveform. This causes a major portion of the digitizer range to be used.

The FFT

Even if the full range of the analog-to-digital converter is used, there's still a possibility of half a digital level of quantizing noise. Increasing digitizer word length is one way to further decrease the amplitude of this noise relative to the signal. For example, a 16-bit digitizer is finer grained than a 10-bit digitizer, but there are practical limits to digitizer word length.

Another effective way of reducing quantizing noise levels on repetitive waveforms is to signal average, in the presence of two or more levels of noise, the output of the 10-bit digitizer into a storage array made up of 16-bit words. (The mathematics are also done with 16-bit words.) This averages random noise and quantizing error in the least significant bits of the 10-bit converter output to random variations in the least significant bits of the array's 16-bit words. The result is finer resolution than that implied by the 10-bit limitation of the hardware.

You should expect noise to be present at some level in all cases of digital signal processing. The noise may have become part of the signal before it was acquired (ambient, additive noise) or it may become part of the signal during analog-to-digital conversion (jitter, quantizing, and other sources). In most cases, the noise level can be reduced significantly. But noise can never be completely removed. Therefore, noise is always present in the FFT results, too. With care, the noise level with respect to other components can be kept small enough to be considered zero. However, when further processing is done (computing phase from the real and imaginary frequency domain for example) don't forget that low-level noise components can make significant contributions to the results.

Windowing Can Cause Leakage. Let's begin talking about leakage by considering a pure cosine wave and its FFT magnitude. We'll consider a cosine wave because it has zero phase (zero imaginary part), and we can concentrate on just the FFT magnitude for what needs to be said about leakage.

This pure cosine wave is shown in Fig. 4-15a. Its array elements were generated by a computer program rather than being obtained by acquiring and digitizing a waveform from a signal generator. This mathematically generated waveform is preferred for this demonstration because each waveform element can be precisely controlled, at least within the computational limits of computer word length. In the case of Fig. 4-15a, the cosine wave has been generated so that exactly 10 cycles appear over the 10 second window length. Also, the cosine wave has an exact amplitude of one volt.

The FFT magnitude of the cosine wave is shown in Fig. 4-15b. Its major parameters, taken from the array of computed Fourier coefficients, are printed out on the CRT photo. Notice that they describe exactly what we would expect for the cosine wave in Fig. 4-15a. The FFT magnitude indicates a cosine wave existing at exactly one Hertz with an amplitude of one volt. (Remember, half the energy is in the positive frequency domain and half in the negative frequency domain.)

Now let's change the frequency of the cosine wave just a little. Let's increase it so exactly 10.5 cycles appear in the 10 second window (1.05 Hz). This new cosine wave is shown in Fig. 4-15c and has an amplitude of exactly one volt. Its FFT magnitude is

The FFT

shown in Fig. 4-15d. Again, the major parameters of the FFT magnitude are printed out at the bottom of the CRT photo. They indicate that the cosine wave has an amplitude of 0.6508 volts and a frequency of 1.1 Hertz. What happened to our one volt, 1.05 Hertz cosine wave?

Leakage is what happened. Compare Figs. 4-15b and d. Fig. 4-15b is the line spectrum we expect for a cosine wave, but Fig. 4-15d is not. Fig. 4-15d is more like something you'd expect from transforming the cosine wave over a short interval with the Fourier integral. In fact, that is in essence what has been done in both cases. The effect of the $\sin x/x$ frequency-domain function for the rectangular data window is, however, very apparent in Fig. 4-15d. There the data window has caused noticeable widening at the bases of the two spectral components. This widening of the primary component can be thought of as leakage of primary power into adjacent components.

But why should Fig. 4-15d be any different than Fig. 4-15b? After all, both magnitude spectra are for cosine waves of the same amplitude and very nearly the same

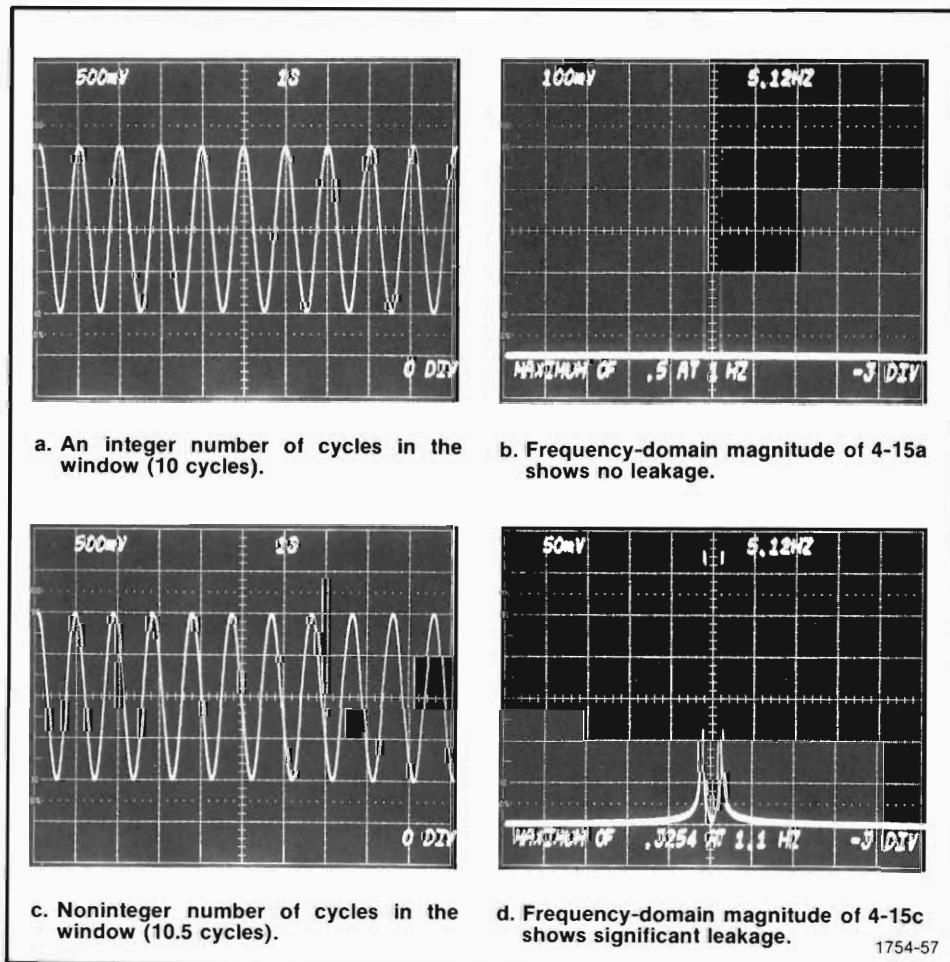


Fig. 4-15. A look at leakage.

The FFT

frequency. Why should the discrete transform treat them differently? In answer, an integer number of cosine cycles is acquired within the window in the one case; in the other case, a noninteger number of cycles is acquired.

To understand the effect of this on leakage, let's look closely at what happens when each cosine wave is prepared for transformation by the DFT via the FFT. This is best done through Figs. 4-16 and 4-17. In both of these figures, cosine waves and sampling rates of lower frequency are used for illustrative convenience. Positional relationships to the window are maintained, however, so the same concepts apply.

Beginning in Fig. 4-16a, a cosine wave is shown in both the time domain and frequency domain. This is the ideal. The cosine wave is exact, and its amplitude and frequency are reflected exactly in the frequency-domain line spectrum. (Since phase is zero, a phase diagram isn't included.)

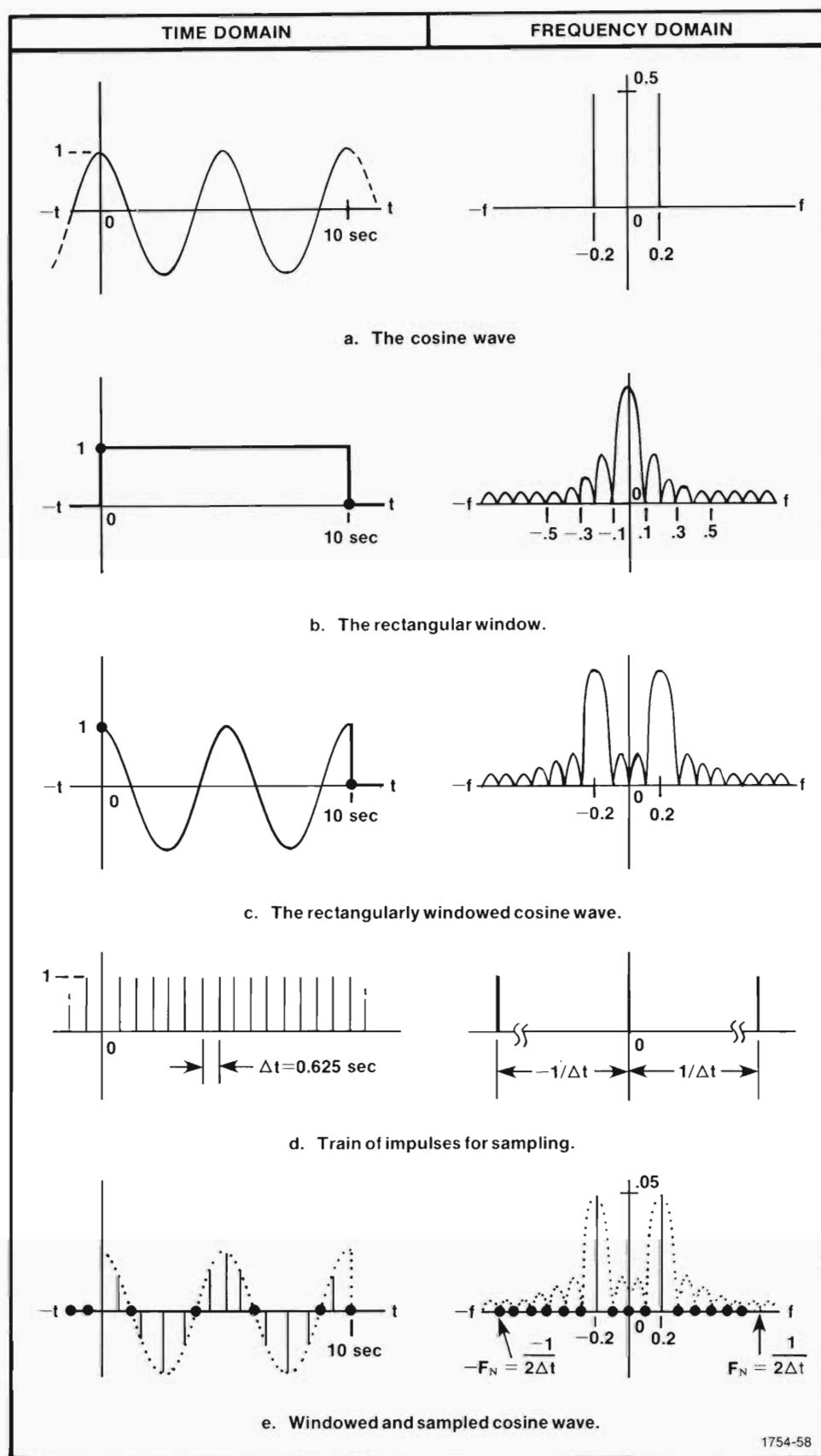
The rectangular window corresponding to the time-domain display areas in Fig. 4-15a and c is shown in Fig. 4-16b. It is defined in the time domain such that it is zero for all negative time and goes to a value of unity at time zero. It remains at unity amplitude from time zero out to 10 seconds. There it falls instantaneously so that its defined amplitude at 10 seconds is zero. In the frequency domain, the magnitude of the rectangular window is the familiar $\sin x/x$ of a square pulse. Because of the time width of the window, the magnitude diagram shows a zero at every $1/10$ seconds = 0.1 Hertz. The phase for this particular window is nonzero; however, its combination with other zero phase waveforms in this example make phase consideration unnecessary.

The first step of analog-to-digital conversion, whether it is done through hardware acquisition or through mathematical waveform generation, is shown in Fig. 4-16c. This is a limiting of our view of the waveform to a finite time interval, a window. In the time domain, this is mathematically equivalent to multiplying the rectangular window in Fig. 4-16b with the cosine wave in Fig. 4-16a. In the frequency domain, this time-domain multiplication corresponds to convolving the frequency-domain magnitudes in Figs. 4-16a and b. (Since the cosine wave has zero phase, its frequency domain convolution with the window results in a zero phase function.) Notice that the maximums of the two $\sin x/x$ major lobes in Fig. 4-16c reside directly over the 0.2 Hertz frequency of the cosine wave. Because of the pulse nature of the window, however, the frequency domain of the windowed cosine wave has now become a continuous spectrum instead of a line spectrum.

The windowed cosine wave in Fig. 4-16c still hasn't been converted to a digital form yet. It's still an unsampled, analog waveform.

To finish the conversion, we must limit our view of the waveform to specific points within the window's unity amplitude duration. This limiting to specific points is referred to as sampling and can be represented as a train of impulses similar to that shown in Fig. 4-16d. The impulses in this train define the discrete time-domain points to be considered in the transformation.

The FFT



1754-58

Fig. 4-16. Process of transforming a cosine wave having an integer number of cycles in the window.

The FFT

For ease of illustration, Fig. 4-16d shows only 16 samples occurring within the time span of the rectangular window. Each of these samples is separated by a Δt of 0.625 seconds. This is substantially fewer samples than used in Fig. 4-15. The window there included 512 samples at a spacing of $\Delta t=0.0195312$ seconds; however, this example doesn't require such resolution.

Fig. 4-16d also shows the frequency-domain magnitude corresponding to the impulse train (phase is zero). Notice that there is an impulse at zero Hertz and impulses occurring every $\pm 1/\Delta t$ Hertz from there. For the current discussion, the only impulse of concern is the one occurring at zero Hertz. The other frequency-domain impulses play a significant part in later discussion of the assumed periodicity of the DFT and FFT.

Time domain sampling of the windowed waveform corresponds mathematically to multiplying the windowed waveform in Fig. 4-16c by the impulse train in Fig. 4-16d. In the frequency domain, this corresponds to the convolution of the windowed waveform's frequency domain and the frequency domain of the impulse train. The results of this sampling are shown in Fig. 4-16e, where the waveform envelopes are suggested by dotted lines.

The time-domain samples in Fig. 4-16e are the waveform amplitudes that are expressed digitally and transformed to the frequency domain by the FFT. There are 16 samples shown in the time-domain of Fig. 4-16e. The first valid sample is considered to be at time zero since that is where the window becomes defined as having a value of one. The last valid sample is located 15 samples from this. The next sample, the 17th sample including time zero, is not valid since it occurs at the right edge of the window, which is zero by definition.

Transforming these 16 time-domain samples to the frequency domain results in 16 complex samples of the frequency domain. The arrangement of these samples on the envelope of the frequency-domain magnitude is shown in Fig. 4-16e. They span a frequency range of $1/\Delta t=1.6$ Hertz including zero Hertz. Seven samples are shown in the positive frequency domain, one is shown at zero Hertz, and eight are shown in the negative frequency domain. The eighth sample in the negative frequency domain is referred to as the Nyquist frequency (F_N) and is equal to $1/2\Delta t$. The Nyquist frequency is the highest frequency that can be defined by the $1/\Delta t$ sampling rate.

From the standpoint of leakage error, the location of the frequency-domain samples relative to the envelope of the frequency-domain magnitude is most important. Notice in Fig. 4-16e that the maximum of the two envelope peaks correspond exactly to sample points at ± 0.2 Hertz. This is the exact frequency of the cosine wave in Fig. 4-16a. Also, notice that the remaining samples fall on the zeros of the magnitude envelope. The result is, in essence, the exact line spectrum for the cosine wave.

Leakage doesn't appear in Fig. 4-16e because an integer number of cosine wave cycles is enclosed in the sampling window. This means that the cosine wave in Fig. 4-16 is harmonically related to the window length. And, with the sample arrangement shown, the cosine wave's frequency falls exactly on a sample point. All other sample points

The FFT

occur at zeros of the magnitude envelope. If the same things were illustrated with the 512 samples used in Fig. 4-15, the same effect would be seen. The only difference then would be that the frequency range covered by the transform would be 51.2 Hertz. The spacing of the frequency samples, however, would still be at 0.1 Hertz in either direction from zero Hertz.

Okay. Now we know why leakage isn't seen in Fig. 4-15b. It's simply because the repetitive waveform is harmonically related to the window length; that is, there's an integer number of cycles in the window.

But what about Fig. 4-15d? It represents a noninteger number of cycles in the window. How does this bring leakage into the results? To answer this, think about what would happen if the frequency of the repetitive waveform was such that it falls between frequency-domain sample points. Can you imagine what happens then? Fig. 4-17 will help you see this better.

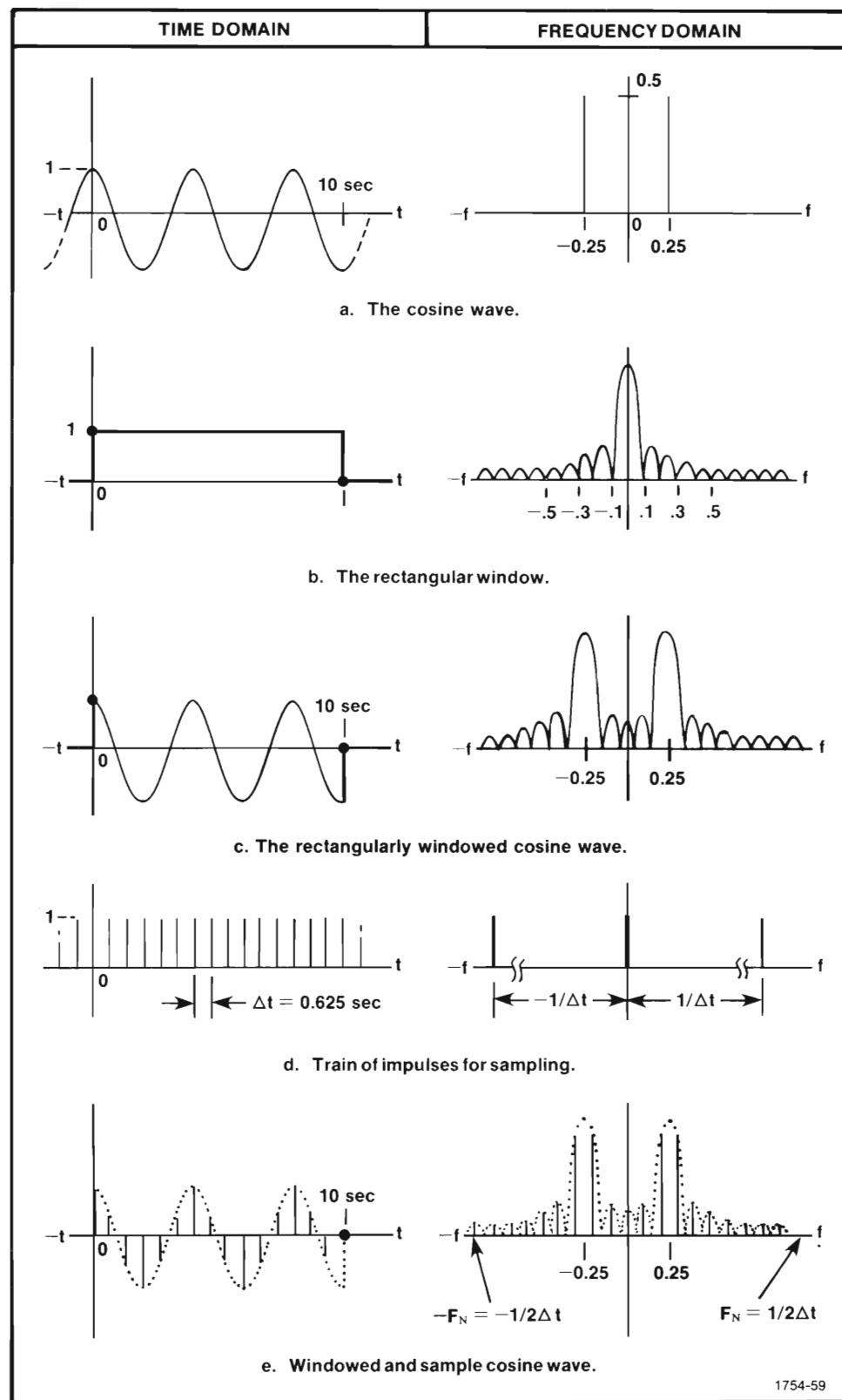
Fig. 4-17 shows exactly the same steps shown in Fig. 4-16. This time, though, the windowing, sampling, and transformation process is carried out on a cosine wave having a frequency of 0.25 Hertz instead of 0.2 Hertz. This cosine wave and its frequency-domain line spectrum are shown in Fig. 4-17a. Following it, in Fig. 4-17b, are the time-domain window and its frequency-domain magnitude. Notice that it is the same 10-second window used in Fig. 4-16. And, as was done in Fig. 4-16, the cosine wave and the window are multiplied. The product of the two is shown in Fig. 4-17c. Notice there that the window includes a noninteger number of cosine cycles (2.5 cycles). The same step in Fig. 4-16 included an integer number of cycles (2 cycles).

Following the product of the window and the cosine wave, the sampling train is shown in Fig. 4-17d. Again, nothing has changed from Fig. 4-16. The sampling train has exactly the same position and spacing between sample points. And it is applied to the windowed waveform in the same manner as described for Fig. 4-16.

Fig. 4-17e is where the effect of changing the cosine wave's frequency really becomes obvious. In particular, notice the position of the magnitude envelope relative to the samples. Notice how the two maximums of the magnitude envelope fall between the 0.2 and 0.3 Hertz samples. This causes two samples to fall on either side of the major lobe maximums. Also, notice that all of the remaining frequency-domain samples now fall at the peaks of the side lobes instead of at the zeros of the magnitude envelope. This is "leakage error." It takes power from components existing in the continuous waveform and gives power to frequency components that don't exist in the continuous waveform. Because of this leakage error, the discrete transform of the cosine wave is no longer the expected line spectrum. However, it should be pointed out here that the discrete transform is what it should be for the time-domain samples provided in Fig. 4-17e.

The same holds true for Fig. 4-15d. There is leakage error there, too, if you choose to interpret the frequency domain in terms of the line spectrum for a periodic waveform. But, if you interpret the results from a digital viewpoint, they are correct for the 512 time-domain samples provided in Fig. 4-15c.

The FFT



1754-59

Fig. 4-17. Process of transforming a cosine wave having a noninteger number of cycles in the window.

The FFT

The leakage error seen in Fig. 4-15d and 4-17e comes about because the period of the cosine wave is not harmonically related to the window. There is a noninteger number of cycles acquired in the window. In both examples, the leakage can be removed by adjusting the window length to include an integer number of cycles.

Figs. 4-16 and 4-17 point out some of the basic properties of leakage. There are some other things that should be pointed out too.

First of all, leakage is not restricted to the frequency-domain magnitude. In dealing with waveforms in general, leakage error, when it occurs, affects the imaginary part and real part of the frequency domain to the same degree. And when converting to the polar form, leakage is carried through the conversion to affect magnitude and phase. Of course if phase is zero, then it is trivial to consider the effect of leakage on phase.

Another thing you should realize is that leakage has the same relative effect on the harmonics of a nonsinusoidal, repetitive waveform as it has on the fundamental of that

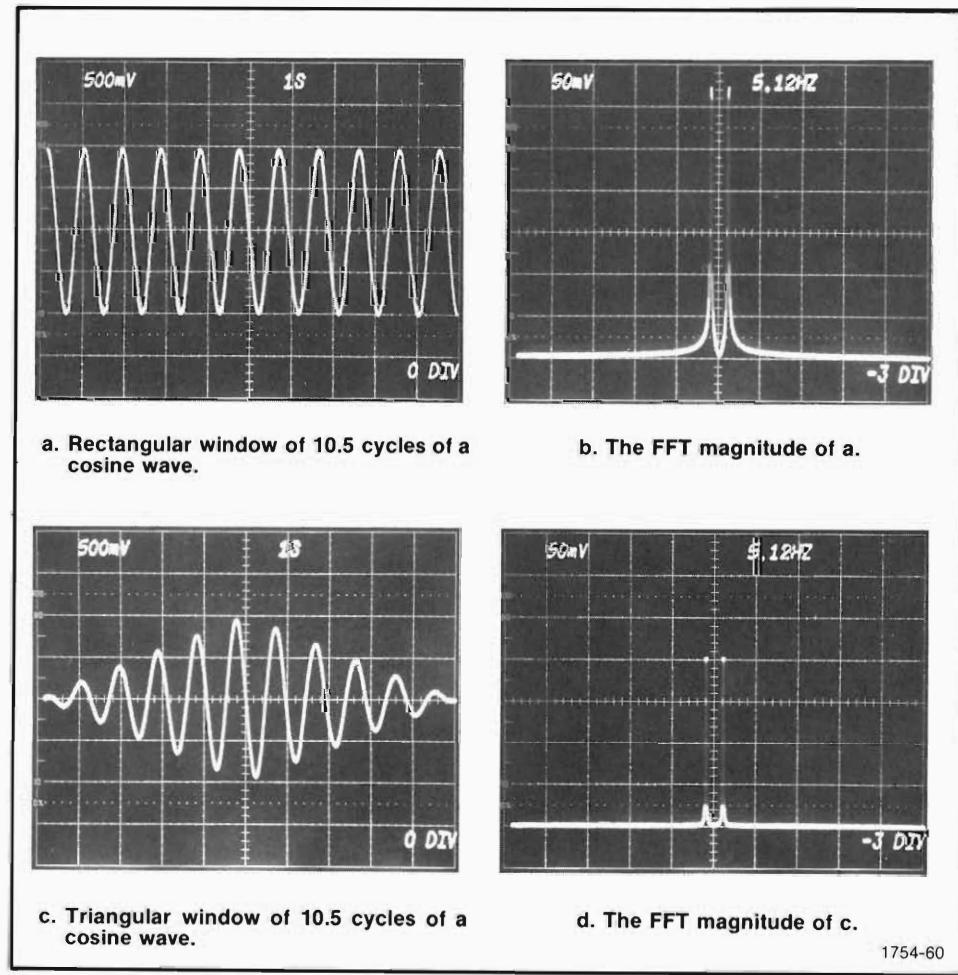


Fig. 4-18. Window shape determines the degree of widening (main lobe) and rate of decay (side lobe size) of a spectral component.

The FFT

waveform. This follows from the fact that the degree of leakage is related to the harmonic relationship of the fundamental to the window length. And in turn, the harmonics see the same relationship to the window through their relation to the fundamental. For example, if the waveform is acquired so that an integer number of cycles are in the window, then each of its harmonics will also have an integer number of cycles in the window.

With regard to nonrepetitive waveforms (pulses), leakage is not a problem as long as the pulse is fully contained within the window. In other words, the pulse must rise from zero and return to zero within the window. If you acquire the pulse so part of it is outside the window, then you have failed to fully define the pulse and should expect inconsistencies in the frequency domain.

As a final note on leakage, the form that leakage takes depends on the form of the window. The leakage shown thus far has been of a form uniquely tied to a rectangular data window. By changing the shape of the window, as shown in Fig. 4-18, you can change the shape of leakage. The properties of various window shapes and some guidelines for using them are given later in Section 5.

Assumed Periodicity—A Case of Even Becoming Odd. In order to demonstrate the periodicity assumed by the FFT, let's return to the cosine waves used to introduce the topic of leakage. In one case, exactly 10 cycles of the cosine wave fill the window; in the other case, the window is filled by 10.5 cycles.

Now let's use the FFT to transform the 10 cycles of cosine wave to the frequency domain. This is shown in Fig. 4-19a. Notice that this time, however, the frequency domain is shown in rectangular form instead of the polar form used in discussing leakage.

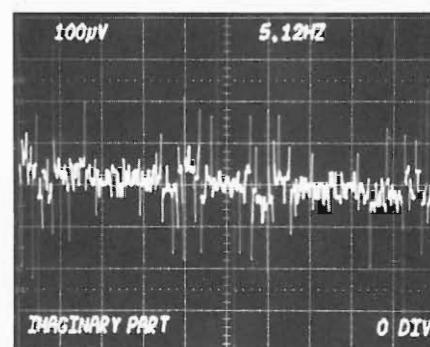
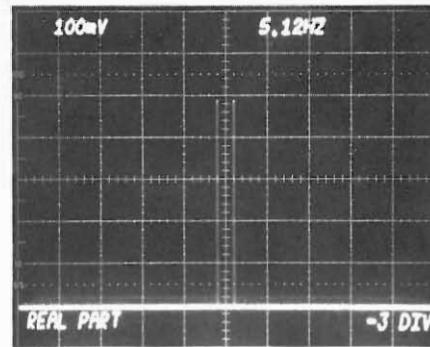
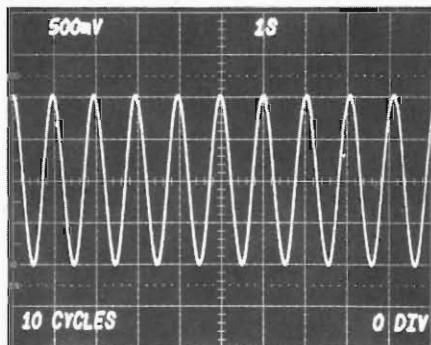
The frequency domain in Fig. 4-19a is what we should expect from theory. Since a cosine wave is an even function, its frequency domain function is a real and even function. This is shown in Fig. 4-19a by the nonzero real part and the effectively zero imaginary part.

Now let's use the FFT to transform the 10.5 cycles of cosine wave. Since we're still dealing with a cosine wave—an even function in theory—something very similar to Fig. 4-19a might be expected. But this isn't the case! In fact, the FFT results for 10.5 cycles (Fig. 4-19b) come out exactly the opposite. The real part is effectively zero, and the imaginary part is nonzero and an odd function. Instead of being the frequency domain for an even function, Fig. 4-19b is the frequency domain for an odd function.

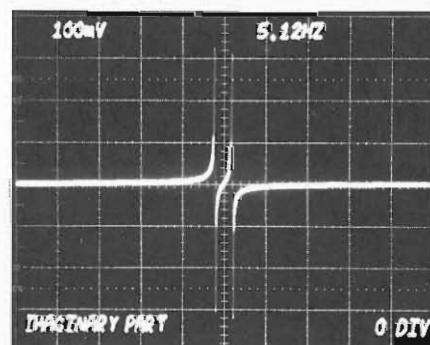
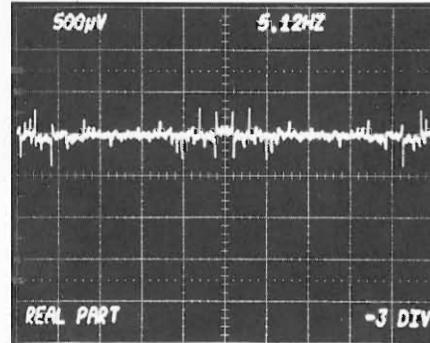
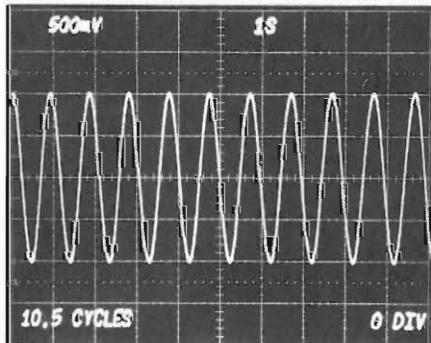
The paradox in Fig. 4-19 is easily explained if we think of the waveforms there as being continued beyond the windowed edges. But we can't totally ignore the window either, so imagine the waveforms being continued by repeating the window. In other words, duplicate the windowed cosine waves in Fig. 4-19. Then lay these duplicates out end to end from either side of the original window so that the information in each window repeats with a period equal to the window length. This is illustrated more clearly

The FFT

- a. 10 cycles of a cosine wave appear as an even function—FFT frequency domain has a nonzero real part and an effectively zero imaginary part.



- b. 10.5 cycles of a cosine wave appear as an odd function—FFT frequency domain has an effectively zero real part and a nonzero imaginary part.



1754-61

Fig. 4-19. Because of assumed periodicity in the FFT, an even function can appear to be an odd function.

The FFT

in Fig. 4-20 where the original window is marked by solid lines. The repeated windows are marked with dotted lines.

Notice in Fig. 4-20a how the windows containing exactly 10 cycles of the waveform allow the waveform to continue on in a cosinusoidal fashion. This is because the window edges fall at the same relative points on the waveform. Now notice how this doesn't happen for the 10.5 cycles of cosine wave (or any noninteger number of cycles). The window edges in Fig. 4-20b don't fall on the same relative points. Therefore, repeating the window results in an odd function of time for the case of 10.5 cycles. For the case of any other noninteger number of cycles (10.25 cycles for example), repeating the window results in a function that is neither even nor odd but is the sum of even and odd parts.

Fig. 4-20 illustrates the assumed periodicity of the FFT. This periodicity isn't necessarily related to the windowed waveform, but is instead directly related to the window. The window of data is assumed to repeat periodically. Exactly how this comes about is illustrated in Fig. 4-21.

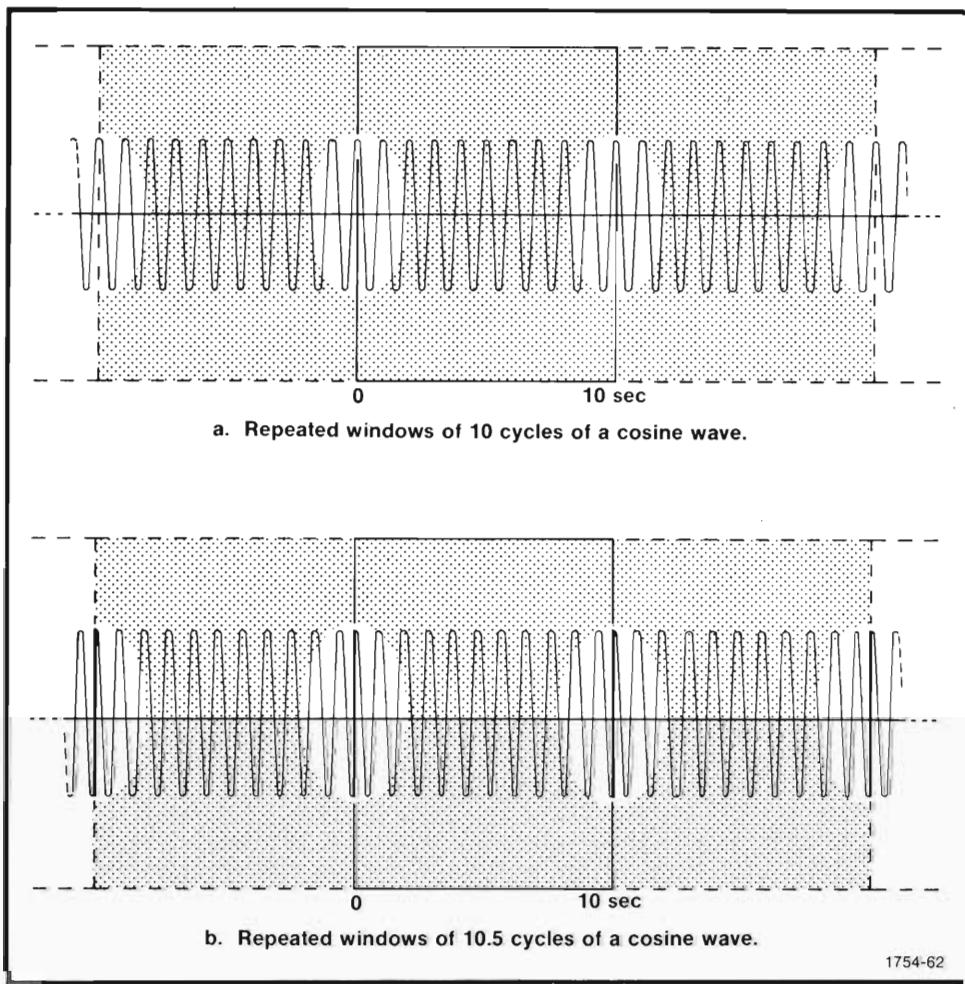


Fig. 4-20. The FFT assumes periodicity of the window.

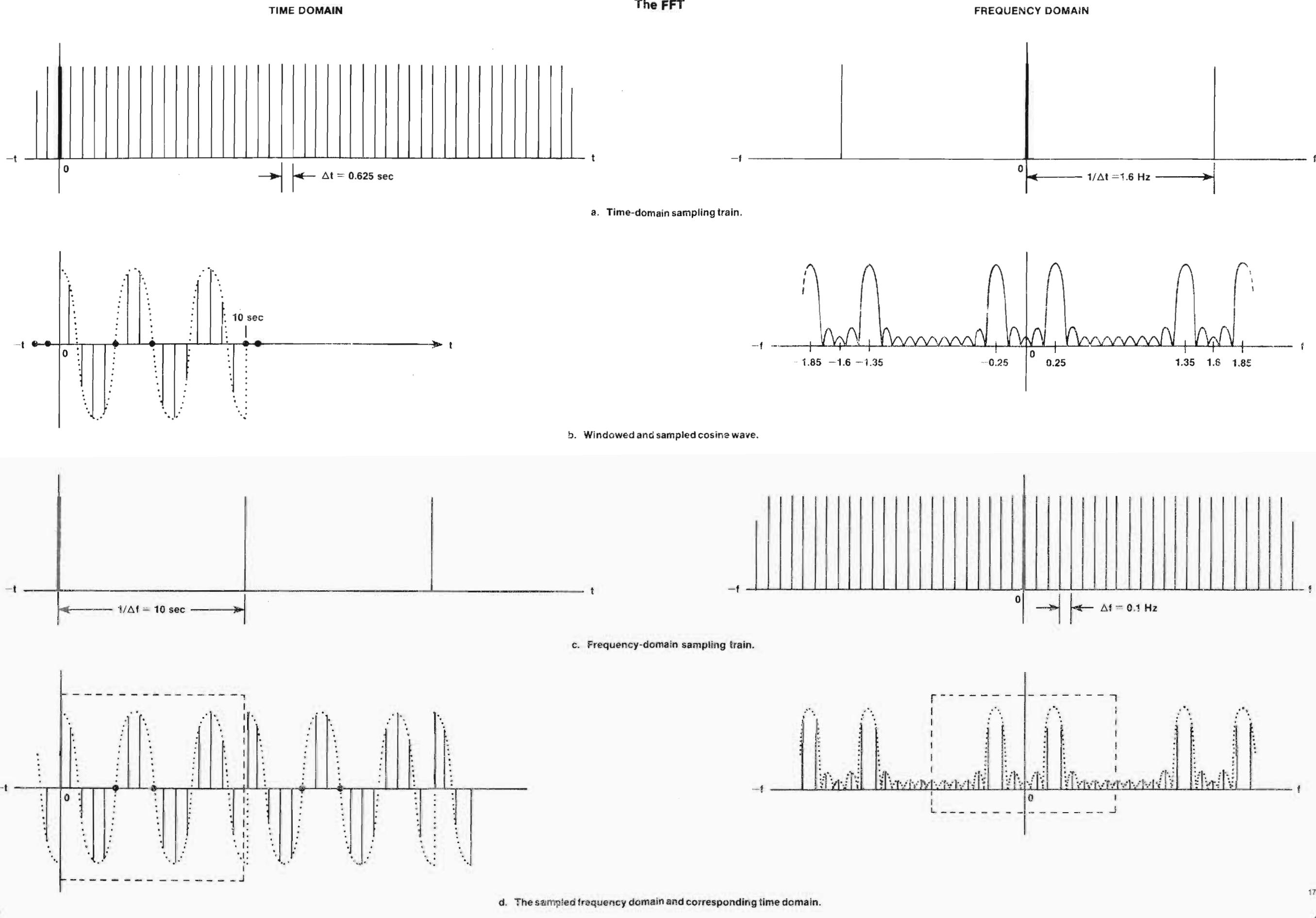


Fig. 4-21. FFT periodicity is tied in with sampling.

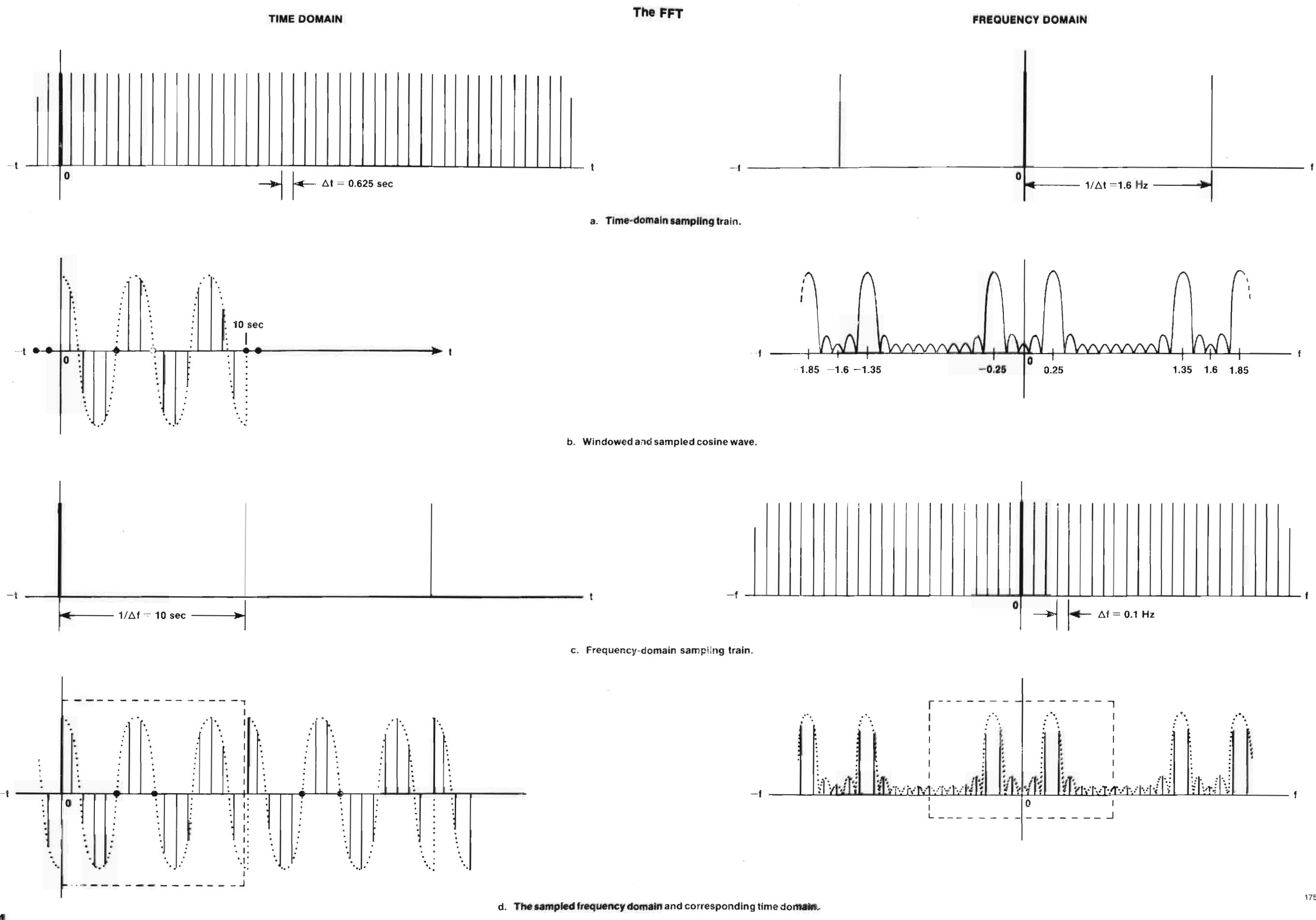
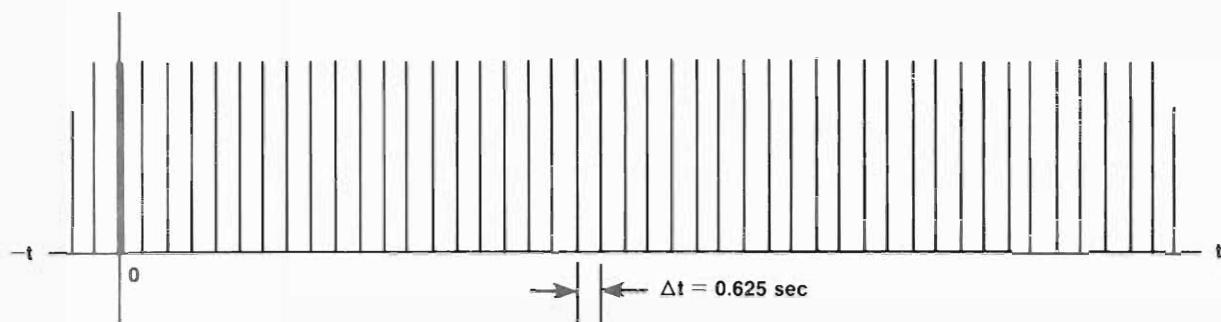


Fig. 4-21. FFT periodicity is tied in with sampling.

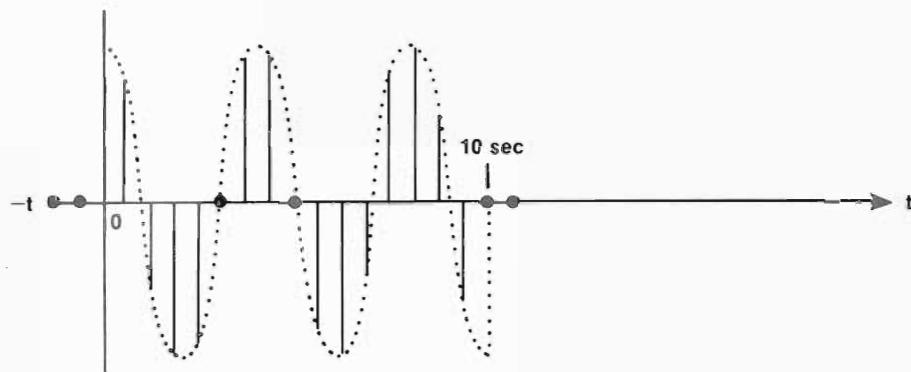


TIME DOMAIN

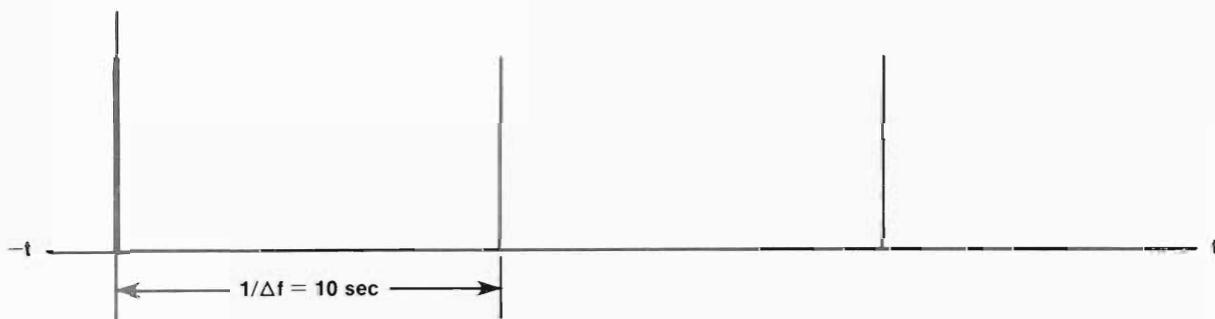
The



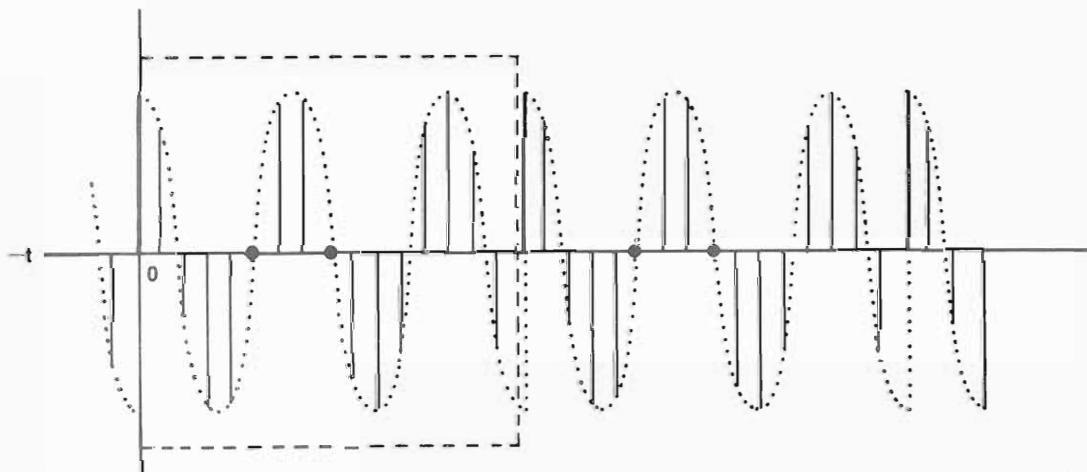
a. Time-domain



b. Windowed an



c. Frequency-c



d. The sampled frequency do

The FFT

Fig. 4-21 begins at a point corresponding to Fig. 4-17d. That is, a cosine wave has been windowed so that 2.5 cycles are contained in the window. Fig. 4-21a shows the time-domain impulse train used for sampling and the frequency-domain magnitude of this sampling train. The Δt spaced impulses in the time domain and the corresponding $1/\Delta t$ impulses in the frequency domain extend in both directions to infinity.

In Fig. 4-21b, the effects of time-domain sampling are shown. In the time domain, the windowed cosine wave of Fig. 4-17c is multiplied by the sampling train. This corresponds to convolution in the frequency domain. When the windowed cosine wave's frequency-domain magnitude is convolved with the $1/\Delta t$ spaced impulses in the frequency domain, the result is a repetition of the windowed cosine wave's frequency-domain magnitude at every $1/\Delta t$ Hertz.

Since the FFT computes only discrete frequency-domain points corresponding to the time-domain samples, the FFT can be equated to frequency-domain sampling. Such a sampling train, with an impulse spacing of $\Delta f=1/N\Delta t=0.1$ Hertz, is shown in the frequency domain of Fig. 4-21c. The time-domain function corresponding to this frequency-domain impulse train is also shown in Fig. 4-21c. It also is a train of impulses with a time spacing of $1/\Delta f=10$ seconds.

Fig. 4-21d shows the results of frequency-domain sampling. The sampling is done by multiplying the frequency-domain sampling train of Fig. 4-21c with the frequency-domain magnitude in Fig. 4-21b. This multiplication corresponds to convolution in the time domain. Thus, the windowed and sampled cosine wave in Fig. 4-21b is convolved with the $1/\Delta f$ impulses in Fig. 4-21c. The result of this is repetition of the windowed and sampled cosine wave in the manner shown in Fig. 4-21d.

The actual data taken by windowing and sampling and the data computed by the FFT are indicated in Fig. 4-21d by dotted lines. These data blocks are what actually fill the software arrays and what make up the instrument displays. (In practice, however, more samples are usually used for each array instead of the 16 shown.) The effective repetition of these data blocks comes about through sampling the original waveforms. How the assumed periodicity of the FFT affects the periodicity of the original waveform depends upon how the samples are arranged on the original waveform. For the case where an integer number of cycles are acquired in the window, the assumed periodicity of the FFT is harmonically related to the waveform's period. If a noninteger number of cycles is acquired, the waveform period isn't harmonically related to the assumed periodicity and the situation of Fig. 4-19b occurs. In either case, the FFT results are correct, considering the effects of windowing and sampling. The error comes in when digital results are interpreted strictly from an analog point of view.

As a final note, assumed periodicity is independent of the waveform being acquired. Periodic repetition of the sampling window always occurs. Thus, an acquired and sampled nonperiodic waveform (a pulse) is also subject to assumed periodicity in the manner of Fig. 4-21. However, if the pulse is acquired so it is completely defined within the sampling window, the actual time domain samples and computed Fourier coefficients will be a correct representation of the pulse. The assumed periodicity simply causes the information to appear duplicated beyond the window edges.

The FFT

Aliasing—A Case of Incomplete Definition. Before any waveform can undergo digital signal processing, the waveform must be windowed and sampled. The rate at which the waveform is sampled determines how well it is defined, how accurate the discrete representation is.

The rule governing proper sampling is referred to as the Nyquist sampling theorem. *This theorem states that the sampling rate must be at least twice the frequency of the highest frequency component in the waveform being sampled.* In other words, there must be at least two samples per cycle for any frequency component you wish to define. If there are less—if the sampling rate is less than twice the highest frequency present—then aliasing occurs.

You can verify the Nyquist sampling theorem through a very simple experiment. It consists of sampling sinusoids of increasing frequency while maintaining a constant sampling rate. At some point, the sinusoid's frequency becomes such that samples occur at less than two per cycle. Then aliasing error can be seen. This is most easily observed in the frequency domain, where the spectral components move out to the edge of the display as frequency is increased. When the Nyquist frequency is reached (the frequency where aliasing begins), the spectral components will have moved to the edges of the display. As the waveform frequency is further increased, the components fold around the edges of the FFT magnitude display and move back to lower frequency areas. This is aliasing. It is the representation of a high-frequency component by a lower frequency component. The key points of this experiment are shown in Fig. 4-22.

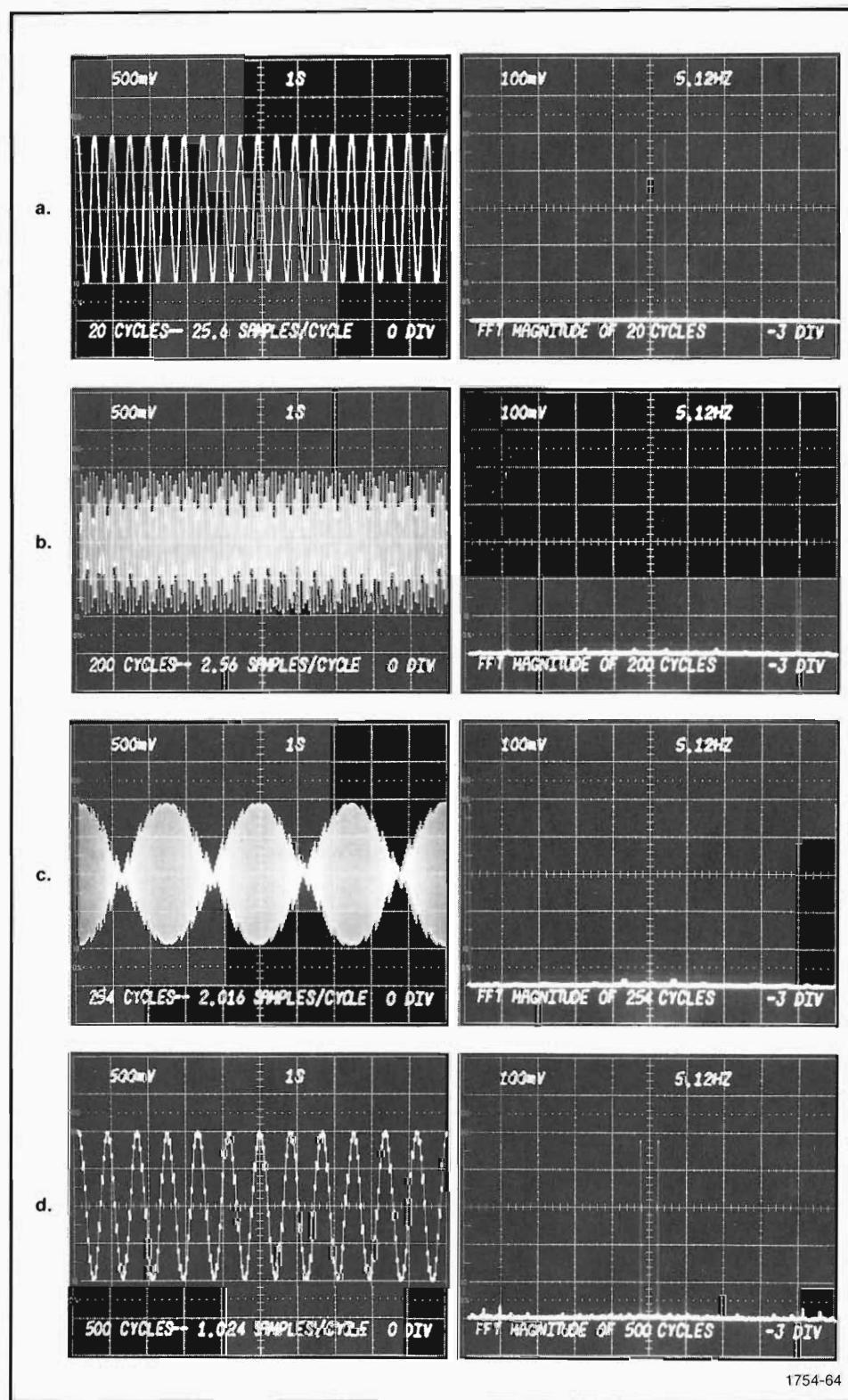
In Fig. 4-22a, 20 cycles of a cosine wave are included in the window. For 512 samples in the window, this gives about 25 samples per cycle, which is more than ample for defining the cosine wave. The FFT magnitude for this 25-samples-per-cycle condition is shown in Fig. 4-22a. It is what would normally be expected for the sampled cosine wave.

Now, in Fig. 4-22b, the frequency is increased so that 200 cycles appear in the window. This means there are a little over two samples per cycle. (In practice it is best to use at least three cycles per sample on the highest frequency component to be certain of avoiding aliasing.) As can be seen by the time-domain representation, it is a little difficult to visualize the cosine wave when only 2.56 samples per cycle are used. Nevertheless, the samples do define the cosine wave. It is correctly indicated by the spectral components near the first and ninth divisions in the FFT magnitude display.

Another increase in frequency brings us, in Fig. 4-22c, to a condition of 2.016 samples per cycle. The spectral components in the FFT magnitude have moved nearly to the edges of the display. We are near the limit stated by the Nyquist sampling theorem.

A large jump in frequency from that in Fig. 4-22c puts us well into the region of aliasing. This is shown in Fig. 4-22d, where 500 cycles have been sampled 1.024 times per cycle. Instead of seeing 500 cycles, though, the samples seem to have outlined about 12 cycles of a lower frequency cosine wave. And in the frequency domain, the

The FFT



1754-64

Fig. 4-22. The sampling rate must provide at least two samples on each cycle of the highest frequency component. Aliasing occurs if there are less than two samples per cycle.

The FFT

spectral components have moved back in from the display edges to indicate this low-frequency alias.

Where did the high frequency in Fig. 4-22d go? Well, it really resides somewhere beyond the edges of the magnitude display. How did its alias in the lower frequency region come about? That can be answered by looking at Fig. 4-23.

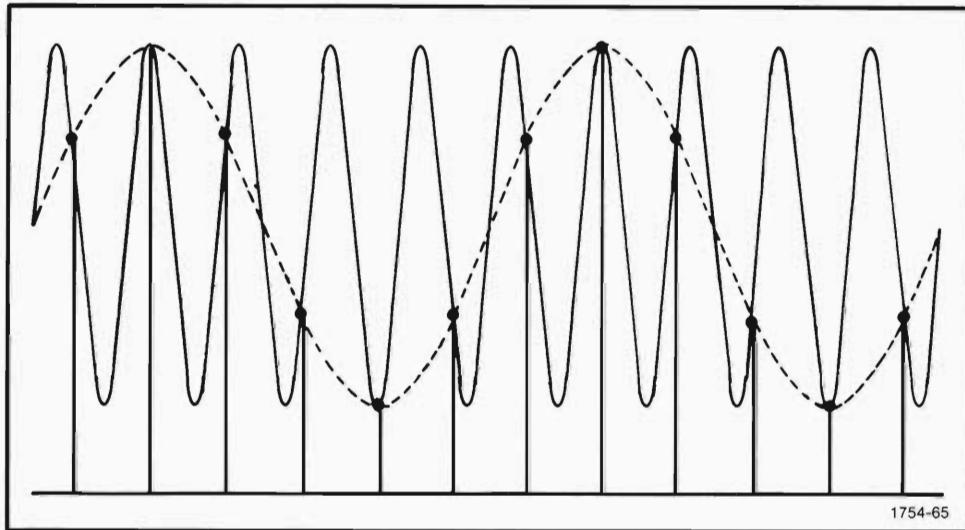


Fig. 4-23. Insufficient sampling of a high-frequency component results in a low-frequency alias.

In Fig. 4-23, ten cycles of a sinusoid are indicated by a solid line. Let's assume that these ten cycles represent a high-frequency component, say 100 kHz, of a waveform that is being sampled. Let's further assume that the heavy dots on the ten cycles represent the amplitude samples relative to the analog-to-digital reference, shown as a solid horizontal line. Notice that there are twelve samples for the ten cycles of the sinusoid. Since it is a 100 kHz sinusoid and it is sampled 1.2 times per cycle, it follows that the sampling frequency is 120 kHz. We know from past discussion that this sampling rate is too low for the 100 kHz component. The Nyquist frequency is 120 kHz/2 or 60 kHz, and as a result we should expect aliasing.

The dotted sinusoid in Fig. 4-23 is the low-frequency alias resulting from sampling the 100 kHz component at too low of a rate. Notice how this alias passes through each of the sample points. A little further inspection shows that, for the assumed sampling rate, the dotted sine wave has a frequency of 20 kHz.

Let's look at the frequencies associated with Fig. 4-23 again. There is a pattern there associated with aliasing. First, we assumed a high-frequency component of 100 kHz. From the sample arrangement on this 100 kHz component, we determined that the sampling rate is 120 kHz. And since the Nyquist frequency is one half the

The FFT

sampling rate, anything above 60 kHz becomes aliased. For our example, the Nyquist frequency is 40 kHz below the 100 kHz component. We also noted that the sample points describe a low-frequency sinusoid. This low-frequency sinusoid, the alias, has a frequency of 20 kHz—it occurs 40 kHz below the Nyquist frequency. Do you see the pattern?

When a sampled waveform has frequency components above the Nyquist or folding frequency, those components are folded about the Nyquist frequency into the frequency domain below. To aid in visualizing this folding action, refer to Fig. 4-24.

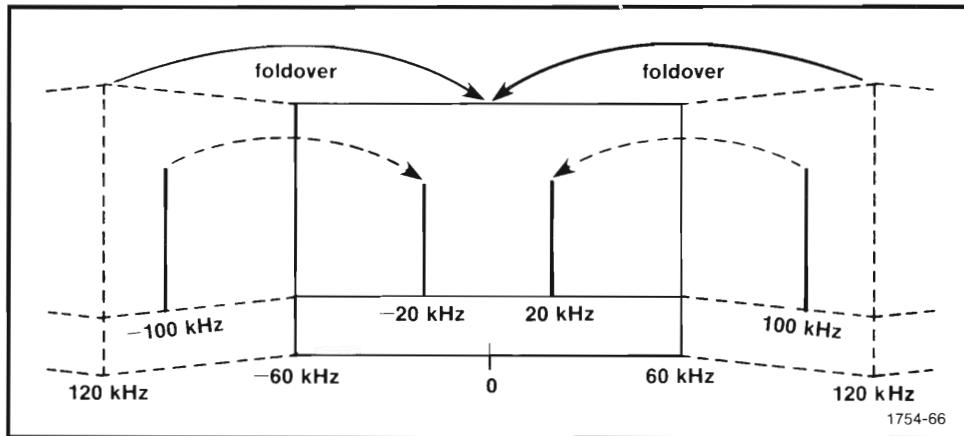


Fig. 4-24. When the sampling rate is 60 kHz, a 100 kHz component is folded down to become a 20 kHz alias.

Another demonstration of aliasing is shown in Fig. 4-25. In this case, an ideal square wave is constructed in a waveform array of the Digital Processing Oscilloscope. This was done by the following DPO TEK BASIC program:

```
100 LET A=.6283  
105 INTEGRATE A,PA  
110 LET B=COS(N*A)  
115 LET B=SGN(B):LET VB$="V":LET HB$="S"  
120 LET SB=1E-3:LET PB=B:STOP
```

The value of N in line 105 determines the number of cycles generated in the data window. If N is set to 20, then 20 cycles will be generated. The square wave is of unity amplitude, but you can change its amplitude by multiplying the B array by a constant. The horizontal scale factor is under the control of SB in line 120. In this example, SB is set to one millisecond per division, but you can set it to any value you wish. Fig. 4-25 was obtained by computing the FFT magnitude of the square wave generated by the example program in lines 100-120.

In Fig. 4-25, the Nyquist frequency is at the left and right sides of the display. The nonaliased components are marked with the numbers one through six. All of the other

The FFT

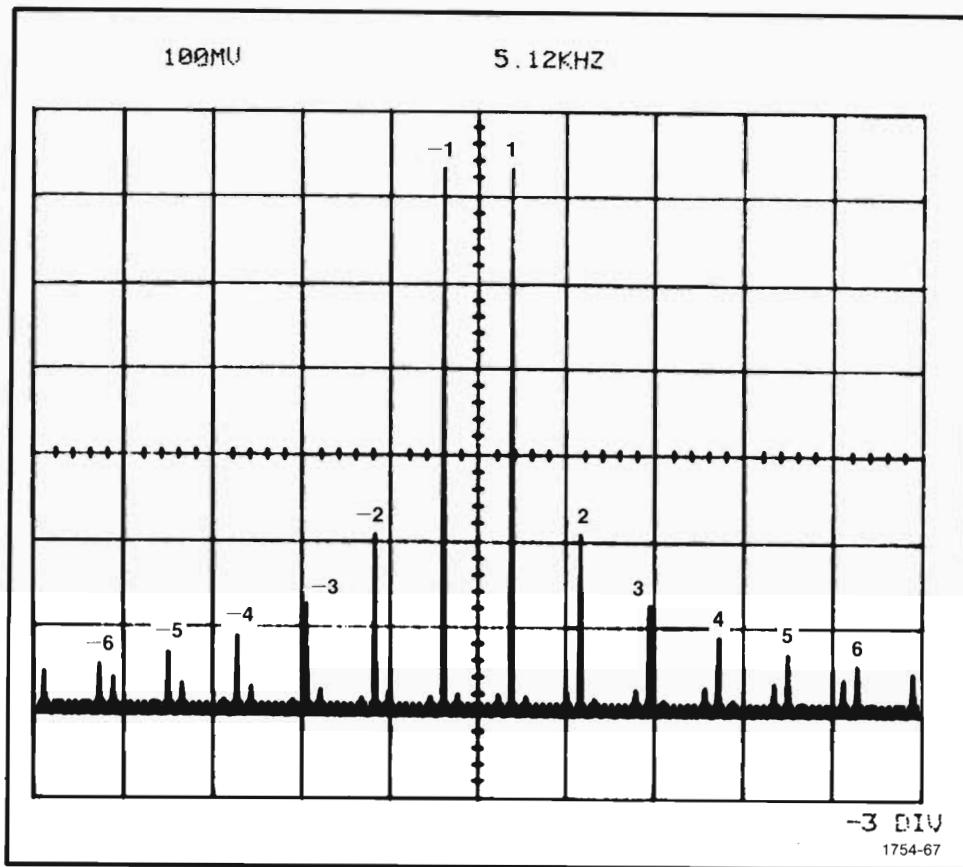


Fig. 4-25. FFT magnitude of an ideal square wave. The fundamental (1) and the harmonics numbered 2,3,4,5, and 6 are not aliases. All others are aliases.

components are aliases of higher frequencies, frequencies that actually exist beyond the edges of the display area. They have been folded about the edges by aliasing and proceed in lower frequencies to zero Hertz (center of the display). At zero Hertz, they fold again and move out to the edges, where they fold again. This goes on as long as there are aliases to be folded.

In general, you can distinguish aliases by knowing something about the waveform you have sampled. If it is periodic, you know that it has harmonic content, and you can find its fundamental from the period. Thus, you can predict where the harmonics should be. And, in general, the aliased components will appear between those harmonics.

In the case of nonperiodic waveforms, where the frequency domain appears to duplicate a continuous spectrum instead of a line spectrum, you can be assured that aliasing has occurred if the spectrum hasn't dropped to a low level before the frequency display edges. Actually pointing out aliases, however, is difficult since all frequencies are present in a pulse's spectrum.

The FFT

Aliasing is inevitable when analog waveforms are windowed and sampled. Remember, the data window itself has a spectrum that is infinite in extent. It is impossible to have a sampling rate that assures two samples per cycle of the highest component. In most cases, however, the high-frequency portion of a waveform's spectrum drops to what can be considered to be an insignificant level. Then all you need to do is select a sampling rate such that several samples per cycle occur for the highest frequency of significance. Any aliasing that occurs then will be below the level of consideration or at least will not be resolvable within the limits of computer word length and the dynamic range of the acquisition instruments.

You will run into some cases where waveforms are not high-frequency limited. The ideal square wave whose spectrum is shown in Fig. 4-25 is one example. Waveforms with very fast rise times and responses from high-pass filters are not band limited, and it is difficult to escape aliasing by adjusting sampling rate. In these cases, aliasing can be prevented by filtering the waveform before it is sampled. Filters used for doing this are referred to as antialiasing filters and are designed to limit the high-frequency content of a waveform to a known and acceptable cutoff frequency.

For most FFT applications, aliasing isn't a problem. It can usually be avoided. When it cannot be avoided, it is often sufficient to just recognize the aliased components and either ignore them or deal with them on an individual basis. And, if this is not acceptable, antialiasing filters can be used to band limit the signal to the desired cutoff frequency.

Other Sources of Error

Noise, leakage, periodicity, and aliasing are all things that are related to implementing the FFT in general. They really aren't FFT errors, though, but are truthfully errors associated with the analog waveform and how it is acquired and sampled.

Besides these general types of errors, there are possibilities for other errors related to the specific hardware or techniques used to acquire data. For example, sampling and digitizing can be done by hand from CRT photos of waveforms or by using various video techniques. In such approaches, optical errors may become important. The electron optics of the CRT and the lens optics of camera systems may add geometric distortion to the acquired waveform. Naturally this distortion becomes part of the data that the FFT works on and may have a significant effect on the frequency domain. There are, however, various software approaches that provide trace prediction and geometry correction.

Each approach to acquisition and sampling has specific errors unique to that approach. It is impossible to cover all of these approaches here and to list the possible errors. And, for the most part, such an extensive listing is unnecessary. Complete and comprehensive measurement systems do recognize these errors and attempt to reduce them to insignificance through proper hardware design and correction algorithms in the software.



SECTION 5

A GUIDE TO USING THE FFT

Up to this point, a lot of Fourier theory has been covered and some general principals for applying it have been pointed out along the way. Now is a good time to go back and summarize this theory in terms of the FFT and to list some specific guidelines for using the FFT. In addition to this recap, Section 5 ends by condensing some windowing information into a foldout. This foldout is perforated for easy removal and can be used as a quick-reference wall chart in your work area.

Some Important Properties of the FFT

For the most part, the important properties of the FFT are also the important properties of the Fourier integral. Most of these are detailed in Section 2; however, it is worthwhile to briefly list them again with some additions pertaining to the FFT. These properties are:

1. **The FFT Has an Inverse.** Any sampled waveform that is transformed to the frequency domain by the FFT algorithm can be transformed back to the time domain by basically the same algorithm.
2. **Even Functions Have Real Parts Only.** If a function is windowed so that repeated windows form a function that is a mirror image (symmetric such that $x(t)=x(-t)$) about time zero, then the windowed function is an even function. Its frequency domain is real and also even.
3. **Odd Functions Have Imaginary Parts Only.** If a function is windowed so that repeated windows form a function that is an inverted mirror image ($x(t)=-x(-t)$) about the window edge (time zero), then the windowed function is an odd function. Its frequency domain is imaginary and odd.
4. **Arbitrary Functions Are the Sum of Even and Odd Parts.** Any function may be expressed as being even or odd or the sum of even and odd parts.
5. **The FFT Is a Linear Transform.** This is a property whereby two or more waveforms can be summed in the time domain to give a third function, and the frequency domain of this new function is the sum of the frequency domains of the original functions.
6. **Time Scaling Affects Frequency Scaling.** A time-scale expansion corresponds to a frequency-scale compression, and a time-scale compression corresponds to a frequency-scale expansion. The effect of time scaling on the frequency-domain amplitude depends upon the type of waveform acquired (periodic or nonperiodic) and its relation to the window.

The FFT

7. **Frequency Scaling Affects Time Scaling.** A frequency-scale expansion corresponds to a time-scale compression, and frequency-scale compression corresponds to a time-scale expansion.

8. **Time Shifting Affects Phase Only.** Shifting a waveform within the window changes the real and imaginary parts of the frequency domain in such a manner that the square root of the sum of the squares (the magnitude) remains constant. The ratio of the imaginary part to the real part varies, however, and affects phase.

9. **Frequency Shifting Causes Time-Domain Modulation.** Shifting a frequency-domain function by plus and minus F causes a sinusoid of frequency F to be modulated in the time domain by the time-domain function corresponding to the frequency-domain function before shifting.

10. **The Convolution Property.** Multiplication of two time-domain waveforms corresponds to convolution of each waveform's frequency-domain function. Conversely, forming the complex product of two frequency-domain functions corresponds to convolving the associated time-domain waveforms. Thus, time-domain convolution is quickly performed with the FFT by transforming the waveforms to be convolved to the frequency domain, forming the complex product of these FFT results, and inverse transforming the product back to the time domain.

11. **The Correlation Property.** Correlating two time-domain waveforms corresponds to conjugating the frequency-domain function for one of the waveforms and then multiplying this by the frequency-domain function of the other waveform.

12. **The FFT Assumes Periodicity in All Cases.** The FFT assumes that the windowed data repeats with a period equal to the window time. Thus, there are many assumed windows extending to either side of the physical window, and each is an exact duplicate of the physical one.

Some Guidelines for Improving FFT Results

By themselves, the properties of the FFT may seem to make a rather dull list. Some or all of them may, at first glance, seem academic and of little usefulness in actually applying the FFT. Experience, however, will reveal exactly the opposite to be true. The following list of guidelines for using the FFT is derived from FFT's properties. Their usefulness and validity depend directly on the properties of the FFT.

Signal Average to Remove Additive Noise. Noise by itself is a function. It, too, has a frequency-domain counterpart. And by the property of linearity, noise that is added to a signal in the time domain is also added to the signal's frequency-domain function.

It is another property of noise that it is generally random and, thus, has an average value that tends to zero in the long term. Because of this, repetitive signals can be

The FFT

acquired a number of times and each acquisition can be summed and an average formed that reduces the level of additive noise. The signal-to-noise ratio improvement gained by this signal averaging is proportional to the square root of the number of waveforms averaged. When the number of averages is expressed as a power of two, this corresponds to a 3-dB improvement for each power of two averages. For example, 128 averages can be expressed as 2^7 and corresponds to a $7 \times 3 \text{ dB} = 21 \text{ dB}$ improvement where truly mean-zero noise is involved.

Whenever waveforms can be repetitively acquired, they should be signal averaged. How many times they should be averaged is a question of how noisy they are. As a general rule, you should signal average any repetitive signal at least 32 times. This reduces low-level noise and any noise added by time jitter or quantizing error. If the waveform is moderately noisy, 128 to 512 averages should provide sufficient improvement. If the waveform is practically buried in noise, many more averages may be required for the desired improvement.

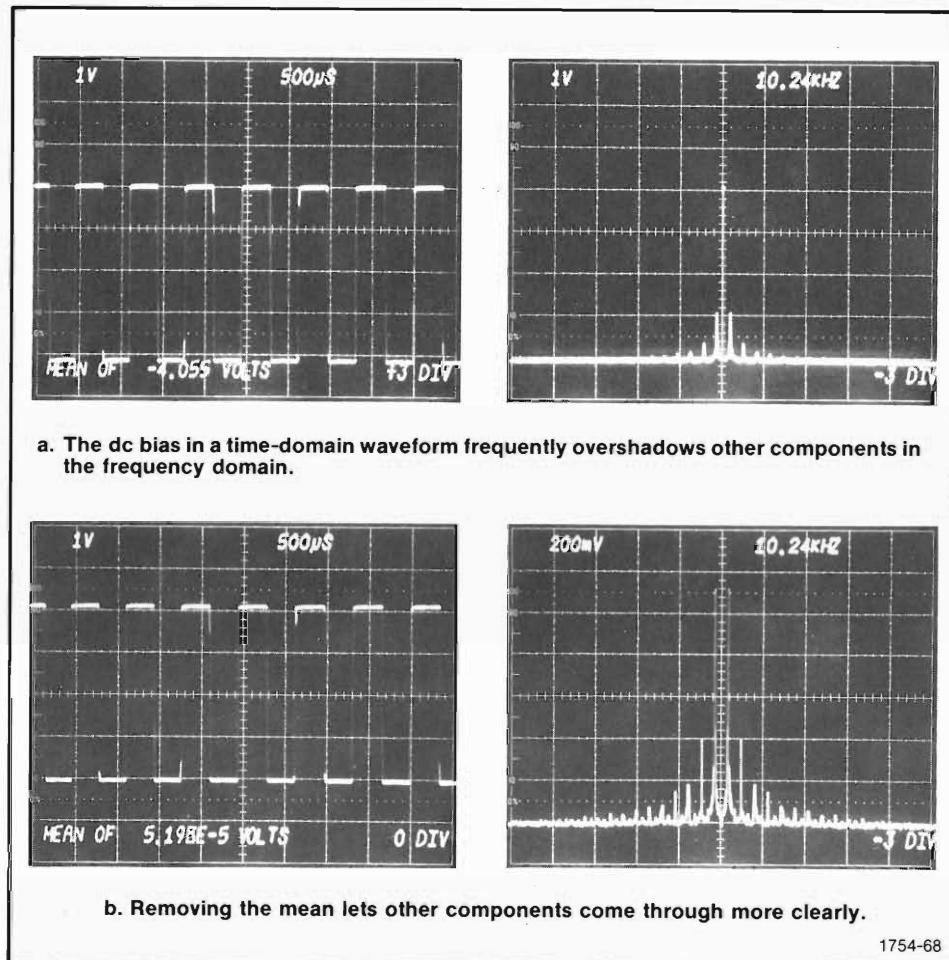
Removing the Mean Often Improves Amplitude Resolution. Many waveforms are such that their mean value is nonzero: they have a dc component. Sometimes waveforms are purposely acquired with a dc bias or inadvertently acquired with a dc bias. The later case frequently occurs when ground reference information isn't properly supplied with the waveform. And finally, the average value of a waveform may be nonzero simply because of the way it is windowed. For example, a pure sine wave has a zero mean over its period. But if the sine wave is considered for a time not equal to its period or not equal to an integer multiple of its period, its mean value is nonzero. Thus, analog waveforms that are normally thought of as having a zero mean will not have a zero mean if the acquisition window contains a noninteger number of cycles.

Whatever the source of the mean value of the waveform samples, the mean is an added component of the waveform. By the linearity property, it is added in both the time domain and the frequency domain. In some cases, it is so high in the frequency domain that it overshadows other waveform components. (Such a case is shown in Fig. 5-1a.) This can be avoided by computing the mean value of the stored waveform and then subtracting that mean value from the waveform samples. Removing the mean from Fig. 5-1a provides the results shown in Fig. 5-1b.

In DPO TEK BASIC and WDI TEK BASIC software, removing the mean is done very simply with one statement. For example, a waveform stored in array A will have its mean removed by entering and executing the following statement.

LET A=A-MEA(A)

The FFT



1754-68

Fig. 5-1. Removing the mean before transforming a waveform often improves frequency-domain resolution.

Always Look at Waveforms from the FFT's Point of View. No matter what type of signal is acquired, the FFT assumes that the data is repeated at every window length. The effect of this is shown in Fig. 5-2, where signals of several types are shown in analog form and then shown with the assumed periodicity of the FFT. The actual segment associated with the window is blocked in solid lines. The assumed repetitions are blocked in dotted lines.

In Fig. 5-2a, the case of periodic signals is shown. Notice that assumed periodicity doesn't change the waveform when the window edges fall on the same relative points of the waveform (integer number of cycles in the window). When the window edges fall at different points on the waveform, however, the assumed periodicity doesn't coincide with the analog waveform. This is the case where a noninteger number of cycles occur in the window, and leakage will smear the frequency-domain information.

Fig. 5-2b shows a nonperiodic waveform of a transient nature. Here, the assumed periodicity causes the transient to appear to be repeated with a period equal to the window length. If we take this periodic point of view, the FFT results describe the

The FFT

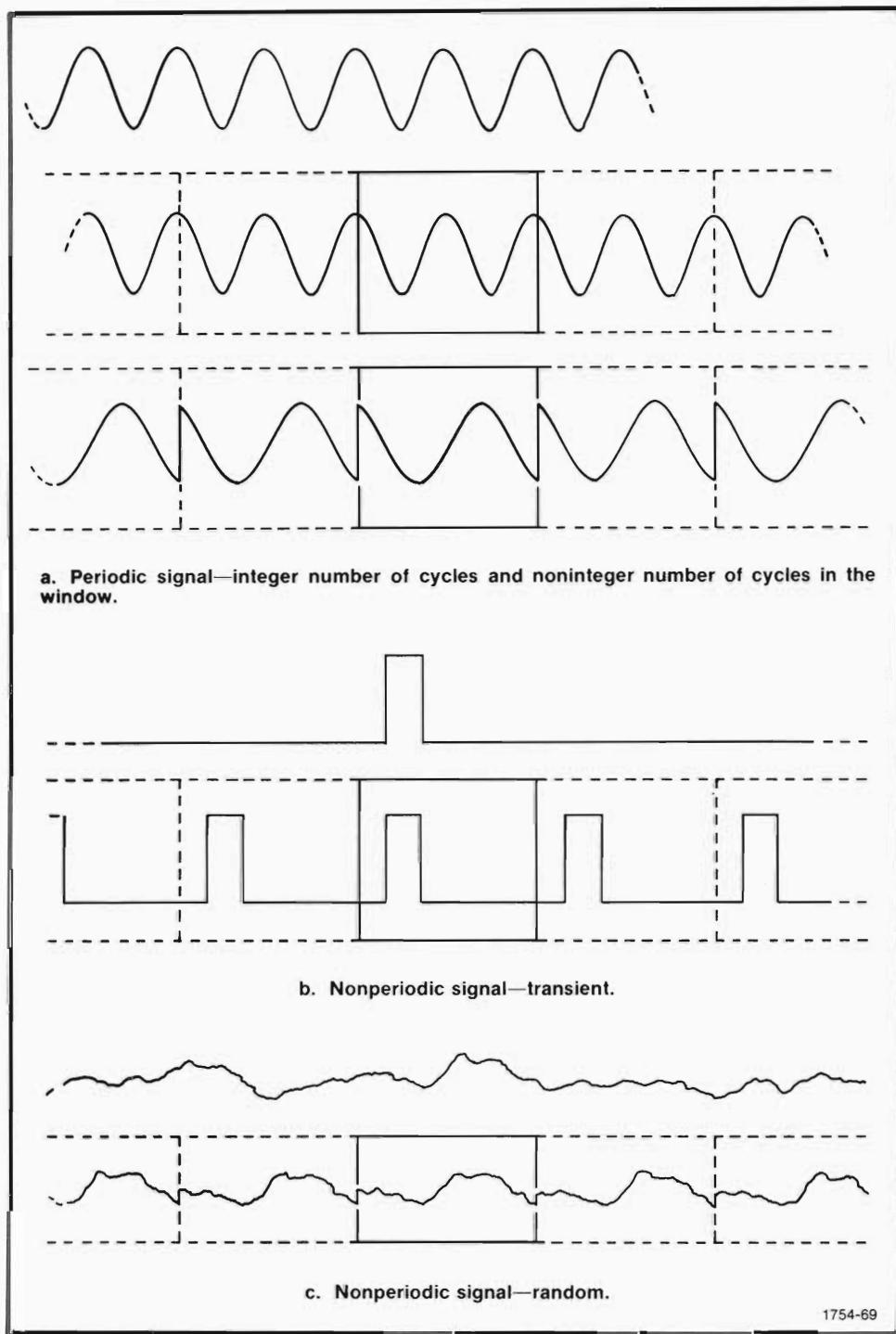


Fig. 5-2. The type of waveform doesn't change the FFT's point of view. Periodicity is always assumed. However, you can change your point of view, according to the type of signal, in interpreting FFT results.

1754-69

The FFT

discrete frequencies of the repeated pulse's line spectrum. On the other hand, if we choose the nonperiodic point of view, the FFT results represent estimates of a few discrete points on the continuous envelope of the energy spectrum for the single pulse. In either case, the FFT results themselves do not change. The numbers are the same. The only thing that changes is our interpretation of these numbers.

Still another type of waveform is shown in Fig. 5-2c. This is a random type of waveform and is subject to assumed periodicity in the same manner as any other waveform. Here again, we can take either point of view described for Fig. 5-2b. In both cases, the FFT results for the acquired interval of random data are the same. In no case, however, can we say anything about the data not acquired in the window. We could assume that it is zero outside the window (pulse), or that the same data exists outside the window as exists inside the window (periodic), or we can make no assumptions at all and just work on the data in the window. Again, these assumptions do not change the results; they just bias our interpretation of the results.

Use Sample Rates Greater than Twice the Highest Frequency. The Nyquist frequency, f_N , determines the highest frequency component of a waveform that can be defined by sampling. The Nyquist frequency is determined by the sampling rate and is given by $f_N = f_s/2 = 1/2\Delta t$, where f_s is the sampling rate and is equal to the reciprocal of the sample interval, Δt . This works out so that a component at the Nyquist frequency is sampled twice over its period. A component less than the Nyquist frequency is sampled more than twice on each cycle, and one greater than the Nyquist frequency is sampled less than twice per cycle.

Since it takes at least two points per cycle to uniquely define a sinusoid of given amplitude and frequency, any components existing below or at the Nyquist frequency are correctly defined. A component above the Nyquist frequency will have less than two samples per cycle and will be redefined as a low-frequency alias. For example, sampling a 100 kHz component when the Nyquist frequency is 60 kHz results in the 100 kHz component being aliased. The alias falls below the Nyquist frequency by the amount that the original component exceeds the Nyquist frequency. For the case of 100 kHz component and 60 kHz Nyquist frequency, the alias falls 40 kHz below the Nyquist frequency.

In practice, it is impossible to guarantee absolutely stable sampling intervals. Therefore, a margin of safety is prudent. Generally, you should sample so that three or more samples occur for each cycle of the highest expected frequency in a waveform. Your sampling rate should be greater than twice the highest significant frequency present in the waveform being sampled.

Sometimes, you won't know what the highest frequency in a waveform is. Or in some cases, your sampling rate is constrained by other considerations and aliasing can't be avoided. In these cases, aliases can often be identified and eliminated or ignored. The following are some suggestions for determining if aliasing has occurred and what components are aliases:

The FFT

1. If the frequency-domain display has significant components near the display edges or does not appear to go to zero before the display edges and remain at zero, then aliasing has probably occurred. These conditions are illustrated in Fig. 5-3.

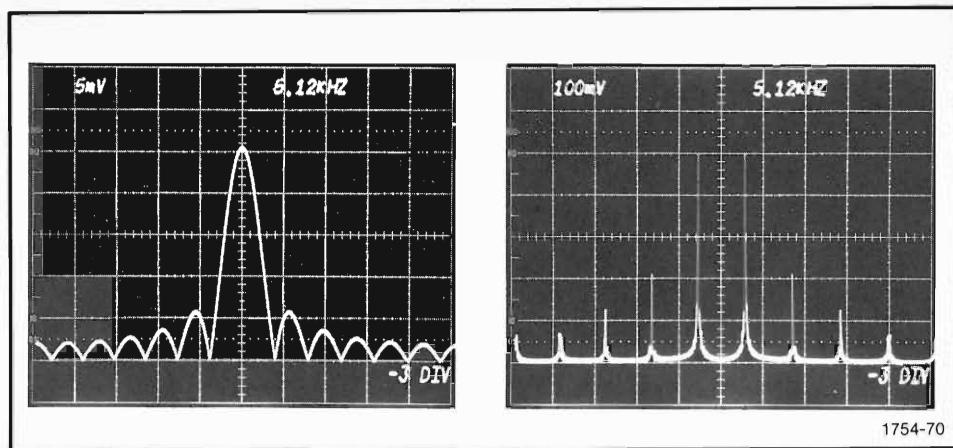


Fig. 5-3. Significant components at or near the display edges indicate that aliasing has probably occurred.

2. If aliases can't be identified and significant components are near the right side of the display, it is possible that the aliases have folded back on top of and added to valid components. In a line spectrum, this occurs when the display edges fall on a harmonic or midway between harmonics. These possibilities are illustrated in Fig. 5-4.

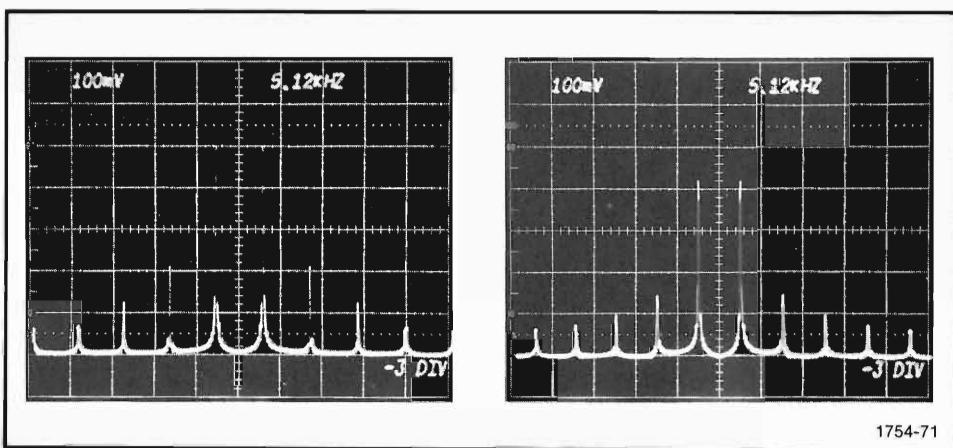


Fig. 5-4. There are a number of cases where aliases aren't obvious because they are added to valid components.

The FFT

3. If the frequency domain appears to be a line spectrum and harmonic spacing isn't even or some harmonics seem to increase in magnitude with increasing frequency, then aliasing should be suspected. In Fig. 5-5, some uneven spacings are indicated and the increasing envelope of aliases is shown by the dotted line. All of the components below the dotted line are aliases.

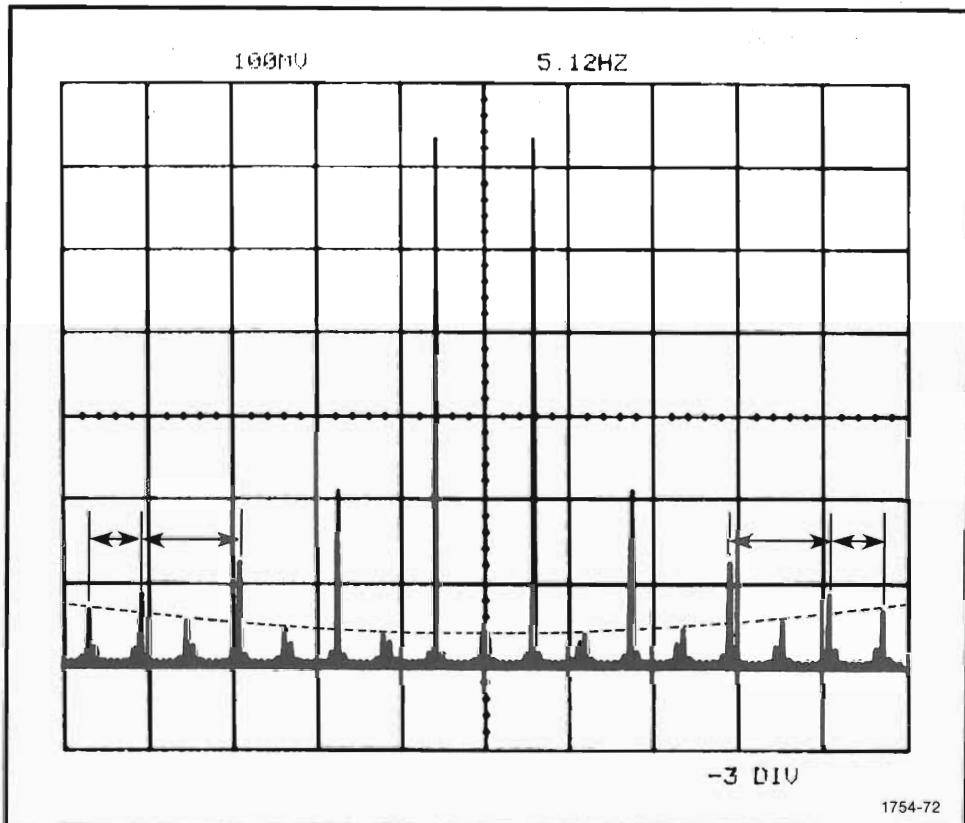


Fig. 5-5. Component spacing and trends in component amplitudes often aid in identifying aliases.

4. Low-level spectral components residing between higher-level components in a line spectrum are not always aliases. If these components are located exactly half way between adjacent, higher-level components, they may be valid harmonics. In the case of a nonsymmetric square wave, the degree of nonsymmetry is indicated by the amplitudes of the even harmonics. This is shown in Fig. 5-6.

The FFT

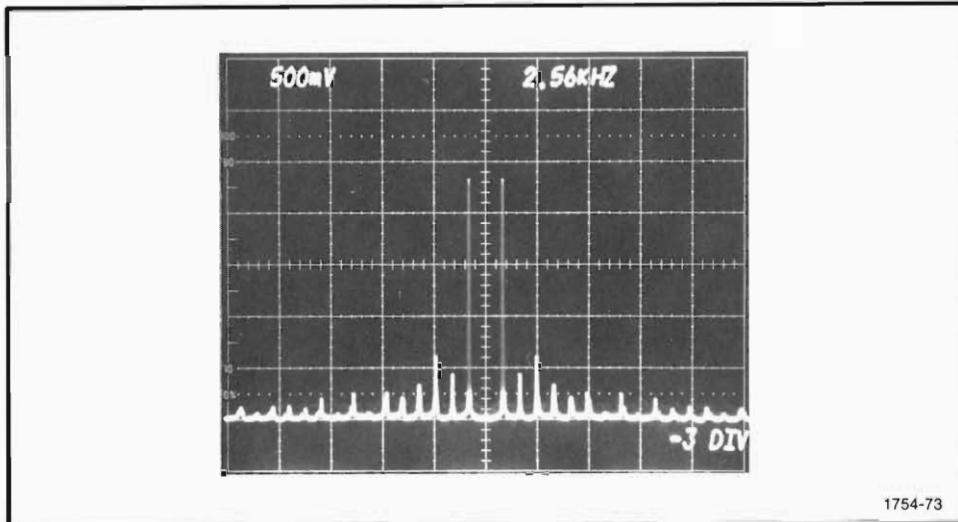


Fig. 5-6. Don't mistake harmonic distortion for aliases. In the above CRT photo, aliases are at an almost insignificant level. Even harmonics, however, are at very significant levels.

Phase is Only Valid for Existing Components. Phase is generally computed from the ratio of the imaginary part to the real part of the frequency domain. Quite often, low-level noise between existing components in the real and imaginary parts can give rise to significant phase components where frequency components don't effectively exist.

In interpreting computed phase diagrams, remember that they are only valid at the points where magnitude components are considered to exist.

Remove Delay to Reduce Phase. From the time-shifting property of the Fourier integral, we know what shifting a waveform relative to time zero affects frequency-domain phase, but not magnitude. The amount that a waveform is shifted positively in time is referred to as delay. And the more delay there is, the greater the phase.

The same thing holds true for the FFT. However, in terms of the FFT, it is more direct to think of delay as affecting both the real and imaginary parts of the frequency domain. Adding or removing delay causes the real and imaginary parts to change. Their change relative to each other is such that the magnitude ($\sqrt{Im^2 + Re^2}$) remains unchanged. On the other hand, the ratio of the imaginary to the real part does change with delay; therefore, phase ($\arctan Im/Re$) changes with delay.

By shifting a waveform in the window to reduce delay, the ratio of Im/Re can be reduced. In some cases, symmetries can be taken advantage of so that the imaginary part goes to zero. This corresponds to producing an even function of time and is illustrated in Fig. 5-7.

Fig. 5-7a shows a band-limited square pulse that is delayed from time zero. The nonzero real and imaginary parts of its frequency domain are also shown in Fig. 5-7a.

The FFT

The frequency-domain magnitude computed from these is the familiar $\sin x/x$ magnitude function for a square pulse. Also, since the real and imaginary parts are nonzero, their ratio will result in a nonzero phase function.

In Fig. 5-7b, the same pulse has had delay removed so that its midpoint coincides with time zero. Half of the pulse is at the left window edge, and the other half is at the right window edge. This is the data arrangement for an even function of time. It is made possible by taking advantage of the assumed periodicity of the FFT so that repeating the window in negative time completes the left half of the pulse.

As would be expected for an even function of time, the frequency domain in Fig. 5-7b is a real and even function of frequency. The imaginary part, except for digital noise, is zero.

The frequency-domain magnitude computed from the real and imaginary parts in Fig. 5-7b will match that computed for Fig. 5-7a. The magnitude is not changed by time shifting. Phase, however, does change. It will be zero if we consider the imaginary part to be absolutely zero. But since the imaginary part is only effectively zero compared to the real part, we should not expect actual calculations to result in zero phase. The digital noise will cause computed phase to be nonzero at some points.

Shifting a waveform in the window to reduce delay can be done during acquisition by choosing an appropriate trigger level. It can also be done after acquisition by using a program to shift data in the waveform array. In WDI TEK BASIC software, a delay argument is provided as an option in the FFT statement. It lets you specify how many sample intervals of delay are to be removed from the data array before the FFT algorithm is executed.

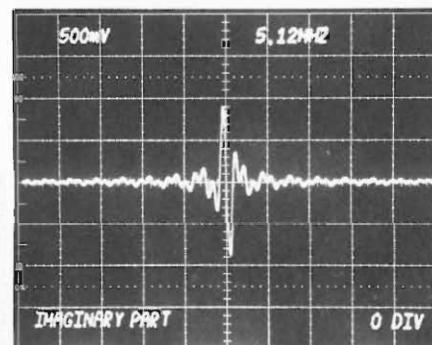
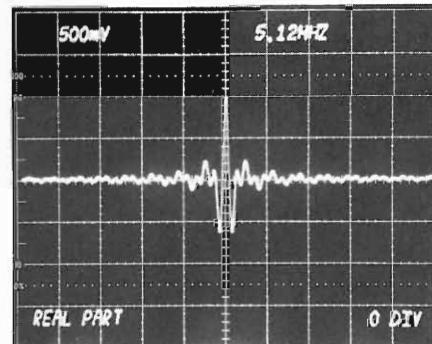
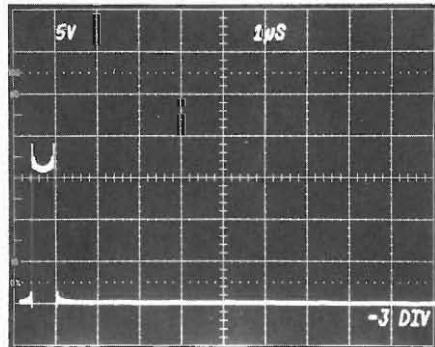
Changing Sample Rates for Different Resolutions. The rate at which you sample a waveform determines how well that waveform becomes defined in either the time or frequency domain. And, in general, increasing definition in the time domain causes a decrease in frequency-domain definition. This comes from the reciprocal relationship between the time-domain sample spacing, Δt , and the corresponding frequency-domain sample spacing, Δf . If you decrease Δt for more time resolution, then Δf increases for less frequency resolution, assuming the number of samples in the time window remains constant.

The effect of changing sample rates ($1/\Delta t$ and $1/\Delta f$) is shown in Fig. 5-8. There, the number of samples in each display is constant at 512. The sample rates, however, are changed by distributing the 512 samples over different time and frequency intervals.

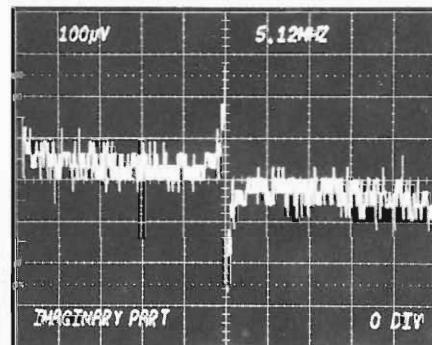
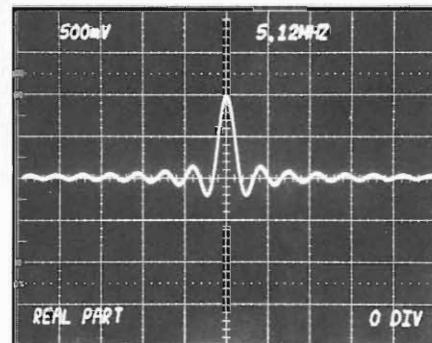
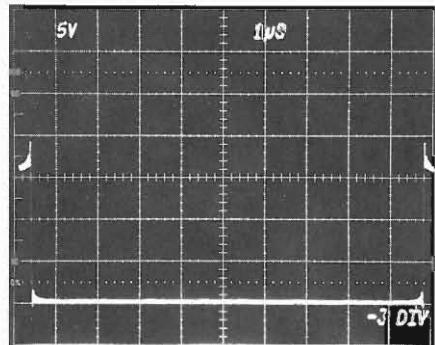
In Fig. 5-8a, a time-domain waveform is acquired in the window so that there are many samples over each of its cycles. This is the kind of resolution you would like for studying signals in the time domain. You can see the details of the waveform, and because of a small Δt , you can accurately determine its time-domain parameters. But this also causes the frequency-domain window, Fig. 5-8b, to span more area than

The FFT

a. Square pulse with about 0.7 μ s of delay.



b. The same square pulse with no delay.



1754-74

Fig. 5-7. Time shifting causes the imaginary and real parts of the frequency domain to vary. Magnitude, however, remains constant while phase reflects the time shift.

The FFT

needed. As you can see in Fig. 5-8b, the useful frequency-domain information is concentrated at the middle of the display. The rest of the display contains essentially useless information.

The opposite situation is shown in Fig. 5-8c and d. In Fig. 5-8c, the same waveform has been acquired so that there are fewer time samples per cycle. This sacrifices time resolution, but look what happens to the frequency domain in Fig. 5-8d. The useful frequency-domain information has spread out to cover most of the display area, and you can easily pick out the different frequency components and their amplitudes. This is what you want for studying signals in the frequency domain.

The choice of sample rate depends upon what you are looking for. If you are after time-domain information only, you'll want to sample for best definition of the waveform, as shown in Fig. 5-8a. If your interest lies in the frequency domain, the approach of Fig. 5-8c is better—be careful of aliasing, though. Generally, your interest will lie in both domains, and a compromise between these two extremes is necessary.

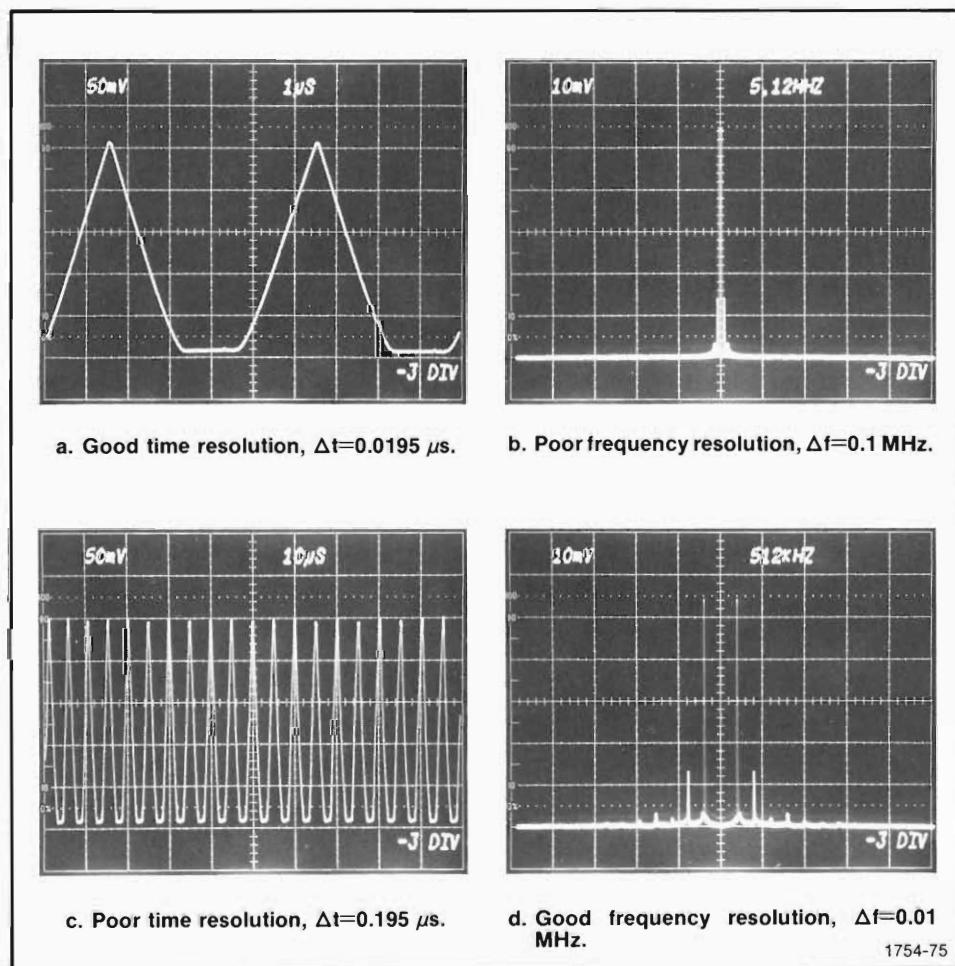


Fig. 5-8. Changing the sample rate changes the resolution.

The FFT

Change Windows to Change Leakage. Leakage is not a universal problem. It does not affect transient data as long as the transient is fully contained in the window. Leakage only occurs when the FFT is used to estimate the discrete line spectra associated with periodic and almost periodic signals. (Almost periodic signals have line spectra, but the components are not harmonically related.) The actual source of leakage, as was described in Section 4, is the window used in acquiring the waveform. The amount of leakage depends upon the window shape and how the waveform fits into the window.

For the simplest case, let's consider a periodic waveform in a rectangular acquisition window. If the waveform is acquired so that an integer number of cycles are in the window, leakage won't occur.

It's difficult, however, to obtain an exact integer number of cycles in an acquisition window. There are just too many variables involved in the physical process. So leakage is inevitable where periodic waveforms are transformed just as they are acquired. But leakage can be avoided or at least controlled by changing the window to fit the data or modify it to a better form.

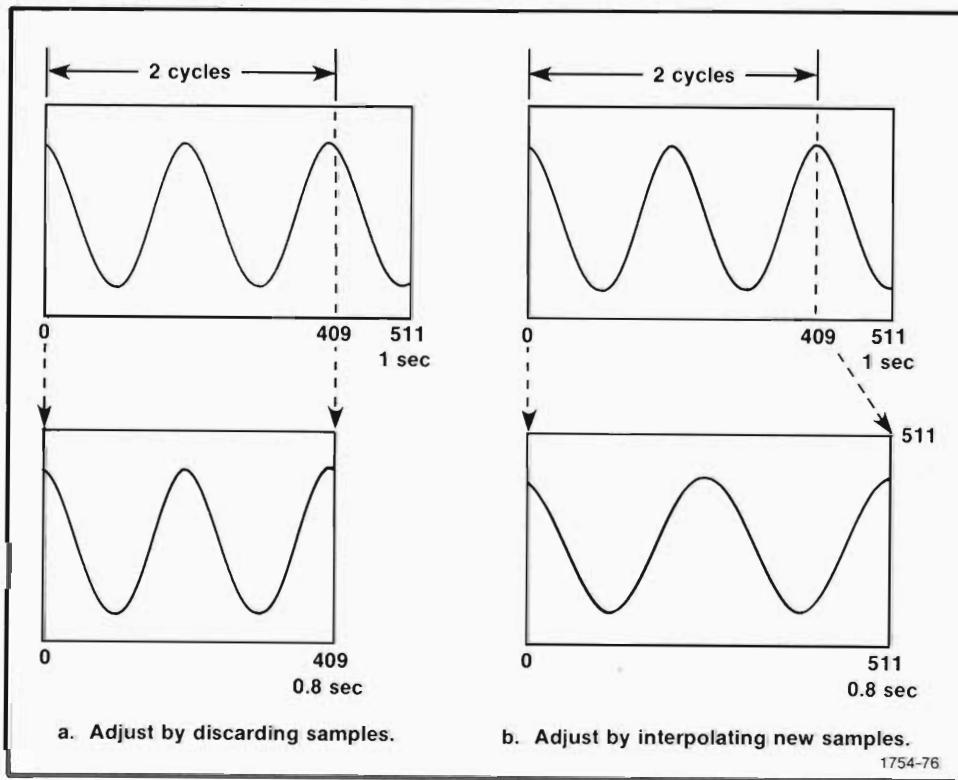


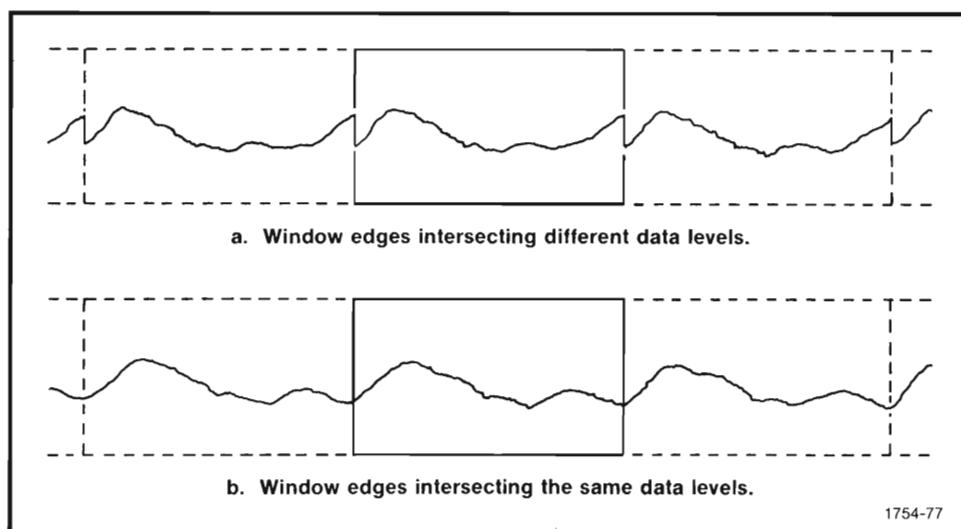
Fig. 5-9. When possible, adjust window length to include an integer number of cycles.

The FFT

For example, if about 2.5 cycles of a waveform are acquired, then we can disregard the samples on the offending half cycle and just transform the samples on the desired two cycles. In effect, the data window (as opposed to the acquisition window) is shortened in the manner of Fig. 5-9a. In that illustration, the two cycles are covered by the first 410 out of 512 samples, and leakage can be avoided by doing a 410-point FFT on just the two cycles. The usefulness of this technique, however, is limited if the FFT routine is of a type constrained to specific numbers of samples. For example, a power of two algorithm can only transform records of $2, 4, 8, 16, 32, 64, 128, 256, 512, \dots, 2^n$ samples. A 410-point FFT cannot be done with a power of two algorithm.

Fig. 5-9b shows a different approach to shortening the window. Although more complex, it has the advantage of retaining the number of samples needed by a particular FFT algorithm. What is done is: The time span of the integer number of cycles is determined first. This is shown at 0.8 seconds in Fig. 5-9b. Then you compute the sample spacing needed to place just those cycles in the data window without changing the number of samples needed by the FFT algorithm. For Fig. 5-9b, this is $\Delta t = 0.8$ seconds/512 = 1.56 milliseconds. These computed sample locations probably won't match the locations where waveform samples were actually taken. In Fig. 5-9b, the actual samples are located at every $\Delta t = 1$ second/512 = 1.95 milliseconds as opposed to the desired 1.56 milliseconds. So the next step is to use the actual sample values and locations to interpolate what the sample values should be for the new sample locations. Linear interpolation is the simplest way of doing this, but other methods may prove more accurate. The interpolates you compute will fill the data window with exactly an integer number of waveform cycles, and leakage is prevented in the transform to the frequency domain.

Since almost periodic data doesn't have a definable period, the techniques of Fig. 5-9 are not wholly applicable to almost periodic data. They can, however, be applied to make the data at least begin and end at the same level (see Fig. 5-10). This prevents jump



1754-77

Fig. 5-10. Remove jump discontinuities by making the data begin and end at the same level.

The FFT

discontinuities at the window edges. Doing this may not entirely eliminate leakage, but it certainly helps to reduce it.

Another way to reduce discontinuities at the window edges and reduce leakage to a tolerable level, is to taper the rectangular window. In short, get rid of the abrupt edges of the rectangular window. Make them fall off smoothly to zero. You can do this by multiplying the acquired data with a window function. Probably the simplest window function is the triangular pulse, and its effect on leakage has already been demonstrated in Fig. 4-18 of Section 4. Some other window shapes that have varying effects on leakage are shown in Table 5-1.

For the most part, the window shapes in Table 5-1 are explained by their names. The rectangular window, for example, is a square pulse and corresponds to a standard acquisition window. The triangular window is a triangular pulse, and the half cycle sine is the positive going portion of the sine wave. To get the cosine window and extended cosine bell, a $0.5(1 - \cos X)$ function is used in the manner of Fig. 5-11. The remaining windows in Table 5-1 are combinations of other windows. For example, the Hamming window is a 92% cosine added to an 8% pedestal, and the Parzen window is the rescaled convolution of two triangular windows.

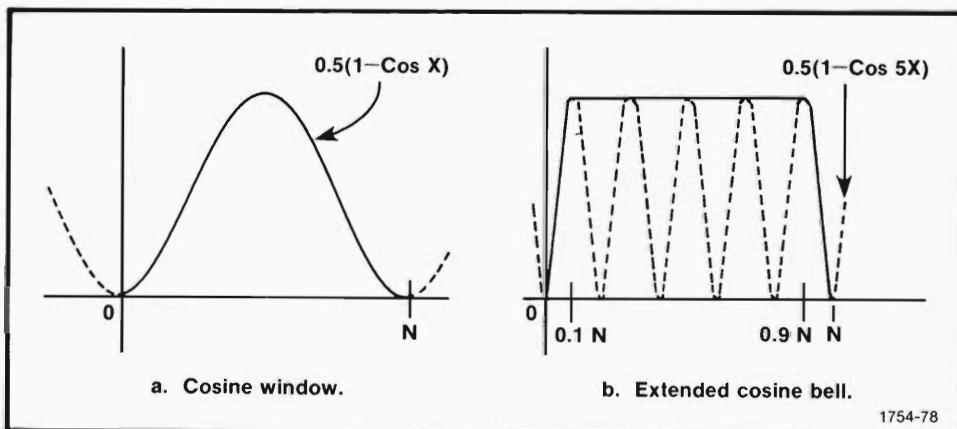


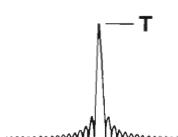
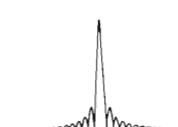
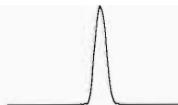
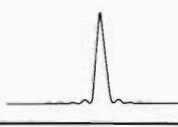
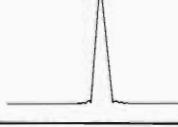
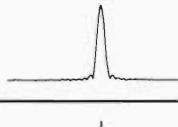
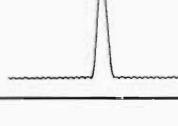
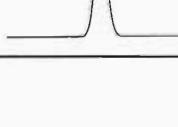
Fig. 5-11. Generating the cosine and extended cosine bell windows.

To give you a basis for comparing leakage from various windows Table 5-1 contains the normalized frequency-domain magnitudes of each window. Some specific parameters describing these magnitudes are also given in Table 5-1. These values were computed from software generated windows and may vary slightly from theoretical values.

In the first column of parameters listed in Table 5-1, the peak magnitude of each window is compared to that of the rectangular window. In the second column, the amplitude of the highest side lobe is given in decibels referenced to the major lobe peak. The 3-dB bandwidth of the major lobe is given in the third column. These bandwidth

The FFT

TABLE 5-1
Some Common Data Windows and Their Frequency-Domain Parameters

Unity Amplitude Window	Shape Equation	Frequency Domain Magnitude	Major Lobe Height	Highest Side Lobe (dB)	Bandwidth (3 dB)	Theoretical Roll-Off (dB/Octave)
Rectangle	$A=1$ for $t=0$ to T		T	-13.2	0.86β	6
Extended Cosine Bell	$A=0.5(1-\cos 2\pi 5t/T)$ for $t=0$ to $T/10$ and $t=9T/10$ to T $A=1$ for $t=T/10$ to $9T/10$		$0.9 T$	-13.5	0.95β	18 (beyond 5β)
Half Cycle Sine	$A=\sin 2\pi 0.5t/T$ for $t=0$ to T		$0.64 T$	-22.4	1.15β	12
Triangle	$A=2t/T$ for $t=0$ to $T/2$ $A=-2t/T + 2$ for $t=T/2$ to T		$0.5 T$	-26.7	1.27β	12
Cosine (Hanning)	$A=0.5(1-\cos 2\pi t/T)$ for $t=0$ to T		$0.5 T$	-31.6	1.39β	18
Half Cycle Sine ³	$A=\sin^3 2\pi 0.5t/T$ for $t=0$ to T		$0.42 T$	-39.5	1.61β	24
Hamming	$A=0.08 + 0.46(1-\cos 2\pi t/T)$ for $t=0$ to T		$0.54 T$	-41.9	1.26β (Beyond 5β)	6
Cosine ²	$A=(0.5(1-\cos 2\pi t/T))^2$ for $t=0$ to T		$0.36 T$	-46.9	1.79β	30
Parzen	$A=1-6(2t/T-1)^2+6 2t/T-1 ^3$ for $t=T/4$ to $3T/4$ $A=2(1- 2t/T-1)^3$ for $t=0$ to $T/4$ and $t=3T/4$ to T		$0.37 T$	-53.2	1.81β	24

1754-79

The FFT

values are normalized to β , the reciprocal of the window's time duration. The last column of parameters gives the theoretical rate of decay (roll-off) of the side lobes.

In general, the lower the side lobes, the less leakage or skirts you will see in the frequency-domain of the windowed data. However, lowering the side lobes also results in more energy being concentrated into widening the major lobe. In Table 5-1, you'll notice that the windows are listed in order of decreasing side-lobe level. As a result, they are also listed in order of increasing bandwidth. The exception to this is the Hamming window, which has a comparatively narrow major lobe for its side-lobe level.

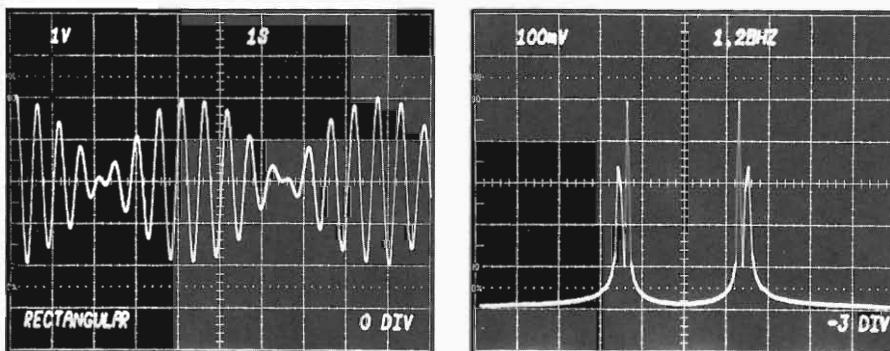
In terms of line spectra, resolution is decreased as bandwidth increases. In other words, equal-amplitude, adjacent frequencies become more difficult to distinguish. On the other hand, as the side lobes decrease, selectivity increases. This means you have increased ability to distinguish adjacent frequency components of unequal amplitudes. This is further demonstrated in Fig. 5-12.

Notice in Table 5-1 and Fig. 5-12 that the major lobe magnitude decreases substantially for various window shapes. This is understandable since each of the unity amplitude windows has less area (energy) than the unity amplitude rectangle. As a result, the frequency-domain magnitude of a windowed waveform decreases according to the window's energy. You can compensate for this by using window amplitudes greater than unity.

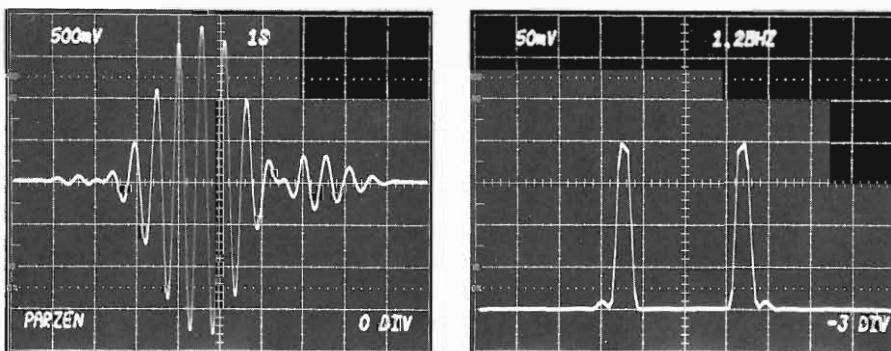
When should windowing be used or when should it not be used? If windowing is needed, which windowing function should be used? The answers to these questions depend upon what you are looking for. If a waveform has adjacent components of nearly equal amplitude, you may want to leave the data in the rectangular window. The increased major lobe width of another window shape may cause the two adjacent components to leak into each other and appear as one. On the other hand, if there is a small component near a large component, a low-side-lobe window will decrease leakage around the large component and make the small component more visible. Ultimately, selecting a window is a compromise between needed side lobe reduction and the allowable increase in major lobe width.

The use of windowing and the choice of windows require some prior knowledge of the signal to be windowed. You have to know what you want out of the frequency domain. And to a degree, you must know what the frequency domain has to offer. It's much like using light filters to improve the quality of photographs. There are some clear cut cases where filters will improve the picture. Then, there are many other cases that require experimentation before the fine details are arrived at.

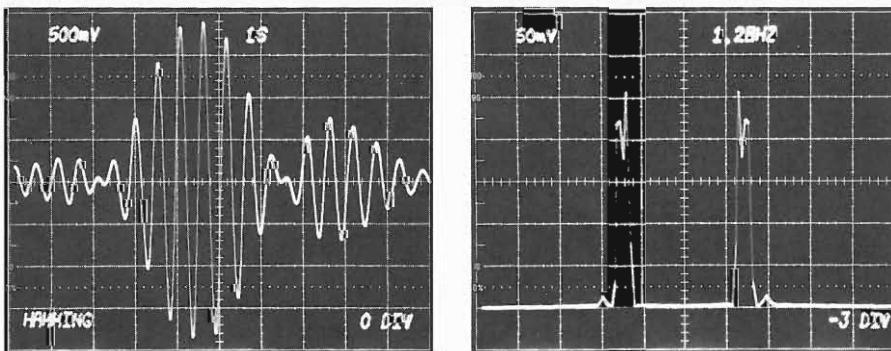
The FFT



a. An almost periodic waveform in the Rectangular acquisition window. The FFT magnitude (4x expanded for detail) shows closely adjacent, nearly equal components; one has substantial leakage. Also, a small wrinkle at two divisions from center hints at a possible third component.



b. Multiplying the waveform by a Hamming window reduces side-lobe leakage and reveals a third low-level component in the FFT magnitude (4x expanded for detail).



c. A Parzen window offers more side lobe reduction, but the increased bandwidth of the major lobe causes the two nearly equal components to merge completely into each other.

1754-80

Fig. 5-12. Windowing is a tradeoff between major lobe bandwidth and side lobe reduction. For this particular waveform, the Hamming window offers the best compromise.

SECTION 6

A BRIEF LOOK AT SOME FFT APPLICATIONS

Astronomy, physics, chemistry, statistics, biomedicine, electronics, mechanics, and a host of other related and unrelated fields have areas of study that can and do benefit from the FFT. This is because the FFT is not a discipline-related technique. It is a broad technique of mathematical analysis. Wherever things vibrate, pump, pulse, bubble, burst, or in any other way change with time, there are possible applications for the FFT.

If you are familiar with spectrum analyzers, think of all the places they are used and the literature covering their uses. The FFT is applicable in all of these areas and more—it is only limited by your ability to provide it with the proper data. Given a phenomenon: if you can acquire and sample it, you can FFT it. Or, if you can't acquire it, maybe you can simulate it. In either case, the FFT gives you the complex frequency domain, a domain where many difficult time-domain techniques become greatly simplified.

To even briefly discuss all of the application possibilities of the FFT would be a considerable task. In lieu of this, let's just look briefly at a few representative examples. Perhaps these will suggest further applications in your specific field of interest.

Distortion Analysis

We have already discussed one type of distortion several times. This is the distortion related to square wave symmetry, where amplitudes of even harmonics indicate the degree of distortion. There are other types of distortion that are probably of more widespread interest.

One of these is harmonic distortion. Percents of harmonic distortion for various harmonics are often quoted when amplifiers and transmission systems are being discussed. Total harmonic distortion, the sum of the harmonic distortions, is also often quoted.

Testing for harmonic distortion is a relatively straightforward operation. A sinusoid (a pure test tone) is fed into the network to be tested. Generally the level of this sinusoid is set to produce the maximum rated output of the circuit. The frequency of the tone depends on the particular system being tested, with 1000 Hertz being a common test frequency for audio circuits.

The FFT

The output caused by the test tone will contain any harmonic distortion caused by the network under test. If it is appreciable, the distortion can be seen by transforming the output signal to the frequency domain and looking at the frequency-domain magnitude. If the network causes harmonic distortion, the frequency-domain magnitude will have frequency components that are harmonically related to the test tone. These harmonics are distortion; they are frequencies in the output that are not present in the input. The percent of harmonic distortion can be determined in the manner shown in Fig. 6-1.

Another type of distortion occurs when two test tones are fed simultaneously into a network. Within the network, the signals tend to modulate each other and produce sum and difference frequencies. The production of these sum and difference frequencies (side bands) is called intermodulation distortion and is related to amplifier linearity. Determining intermodulation distortion from the frequency-domain magnitude of a network's output is shown in Fig. 6-2.

Of course, any harmonic analysis made via the FFT must be made with due consideration for leakage. You may need to preprocess the data to ensure an integral number of cycles, or you may want to window the data to reduce frequency-domain leakage.

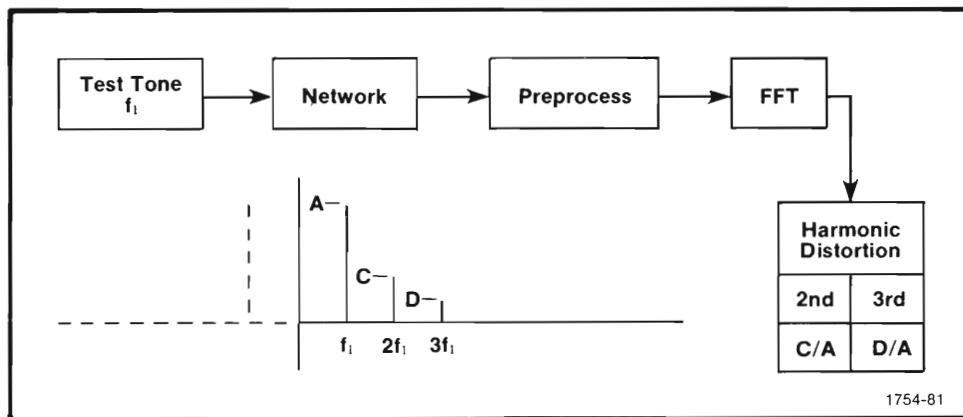


Fig. 6-1. Measuring harmonic distortion.

The FFT

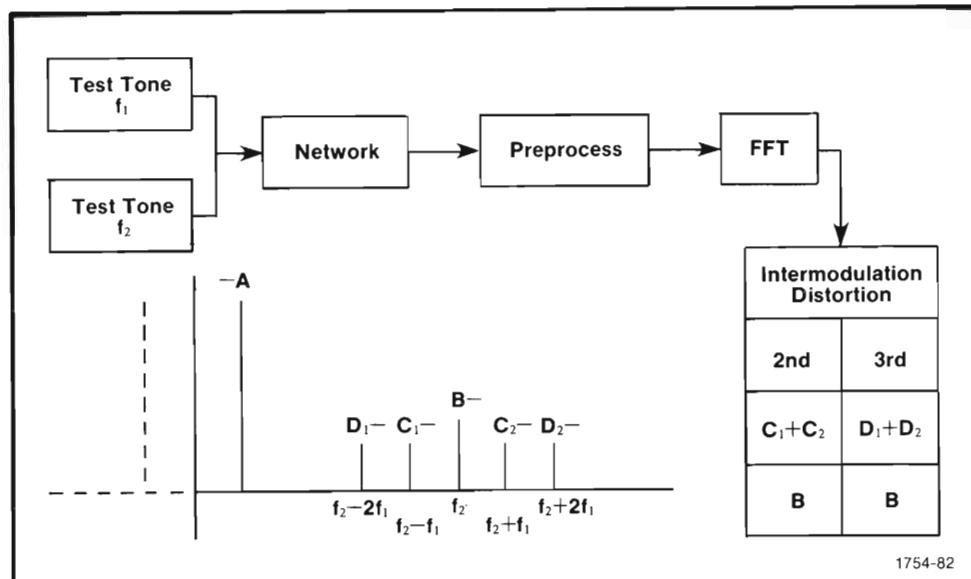


Fig. 6-2. Measuring Intermodulation distortion.

Vibration and Mechanical Signature Analysis

Motors, pumps, compressors, rollers, and other assorted rotating machinery vibrate when they are in operation. When a machine "vibrates too much," it becomes obvious that "the darn thing is shot." This is the simplest type of vibration analysis. But by the time the results are in, a worn ten-dollar bearing may have cost thousands of dollars in further damage and unplanned down time.

A better approach is to monitor machinery vibrations through strategically placed displacement, velocity, and acceleration transducers. Periodic readings of vibration levels can then be taken, and foreboding trends can often be spotted long before the vibrations reach a level that a machine operator would consider suspicious.

Still, vibration level alone cannot tell the whole story. A component can be defective and failing or may have already failed without affecting vibration level. Yet, those defect or failure vibrations are there, and they often become quite visible when the machine's total vibration waveshape is transformed to the frequency domain. Imbalances, misalignments, and bearing instabilities all add their components to the spectrum.

A vibration spectrum is referred to as a mechanical signature, and the determination of information from it is referred to as signature analysis. Generally, mechanical signatures are obtained when a machine is placed in operation or a

The FFT

standard signature is obtained from a known good machine. Then mechanical signatures taken at later dates can be compared to the standard and significant changes in signature components noted. Often, too, specific signature components can be matched to specific mechanical parts. Thus, not only can defects or impending failure be predicted, but the defective part can also be pinpointed. Some examples of how failures or defects might be spotted are illustrated in Fig. 6-3.

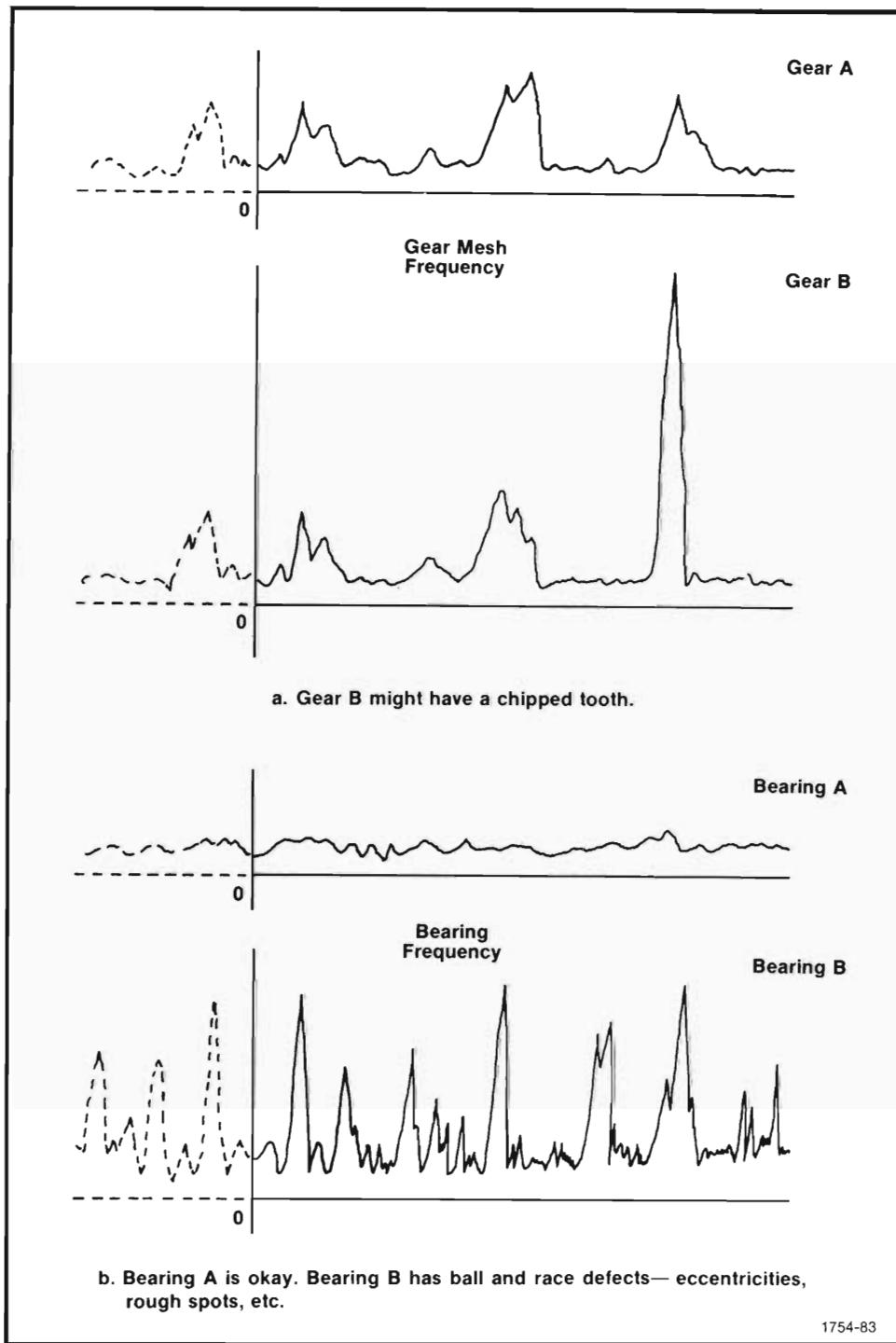


Fig. 6-3. Signature analysis can point out mechanical deficiencies.

The FFT

Frequency Response Estimation (Transfer Functions)

A large part of scientific and engineering work centers around systems. These systems can be simple or complicated. They can be mechanical, electrical, or biological. But whatever they may be, the main issues usually are: "What does it do? How does it react to this stimulus? What would happen if...?"

Often, these issues can be settled by actually testing the system. Just as often, though, testing the system under the conditions in question may be impractical, or there may be an element of danger involved. Then you would rather predict than test or at least get some reassuring predictions before actual testing.

What you need is a means of completely characterizing the system. You need to know how it will respond to each frequency component of an arbitrary input signal.

How a system reacts at every frequency is called the *system frequency response*. This frequency response is often expressed as an amplitude-frequency plot and a phase-frequency plot or, where possible, a set of equations describing these plots. For a linear, time-invariant system, the frequency response completely characterizes the system. If the system is nonlinear or time varying, a frequency response plot can characterize the system for some specific operating conditions.

There are a variety of methods for obtaining frequency responses. The simplest is to feed sinusoids at various frequencies, one at a time, into the system; then the amplitude and phase changes at the output are plotted. Swept frequency oscillators and various detection schemes are common means of obtaining frequency response plots. A spectrum analyzer with a tracking generator is another standard approach. Typically, however, standard instrumentation is limited to providing only an amplitude-frequency plot. Also, swept frequency oscillators and tracking generators have limited frequency ranges. These limitations result in incomplete characterization of the system—only an amplitude-frequency plot is obtained for a limited frequency band. Depending upon your analysis job, you may need more than this.

Another point to consider is that sinusoids are not practical test signals for some situations. This is particularly true for many mechanical and geological systems. Consider oil exploration for example. Geological surveys are routinely conducted by exploding small charges on the earth's surface. Responses, picked up by geophones placed at strategic locations, can be used to characterize subsurface structure.

A sharp concentration of energy, such as a geologist's test charge, is generally referred to as an impulse. How a system reacts to an impulse is termed the *impulse response*, and the impulse response is the time-domain equivalent of the frequency response.

If you Fourier transform or FFT a system's impulse response, you get the system frequency response. This is often referred to as the system's *transfer function*; however, a transfer function is, strictly speaking, the Laplace transform of the impulse response. The transfer function and frequency response can be considered to be the same, though, as long as the system is assumed to be at steady state before time zero.

The FFT

In the theoretical sense, an impulse is a function of infinite amplitude, zero width, and unity area. Its frequency domain has unity amplitude at every frequency. In practical terms, an impulse has a finite amplitude (great enough to elicit a response, but small enough to avoid damaging the system) and a nonzero width. The width must be much less than the expected response time of the system.

All of this can be likened to a Chinese gong—the strike of the mallet is the impulse, and the vibration of the gong is the impulse response. If the gong is struck with too much force, it is driven through the wall and destroyed. If it's struck too lightly, the response is minimal and you hear nothing. If you strike it forcefully but let the mallet rest on the gong, it sounds mushy. Only when it is struck sharply and with moderate force does it respond with its clear, characteristic ringing.

There are also cases where an impulse is not an appropriate test signal. In testing electronic networks, for example, it is often easier to apply a unit step function to the input. Then the voltage at the output is referred to as the *step response*. This is related to the impulse response by the fact that the derivative of a step is an impulse. And, for a linear, time-invariant system, the derivative of the step response gives the impulse response. And again, the frequency response is the FFT of the impulse response.

There are some cases, too, where none of the standard test signals are appropriate—sinusoids, swept frequencies, impulses, and steps are out of the question. A communications network might be a good example of this.

Suppose the network has already been installed and is in use; you don't want to interrupt it for frequency response testing. For cases such as this, the operating signals must be your test signals. Their relationship to the frequency response is shown in Fig. 6-4.

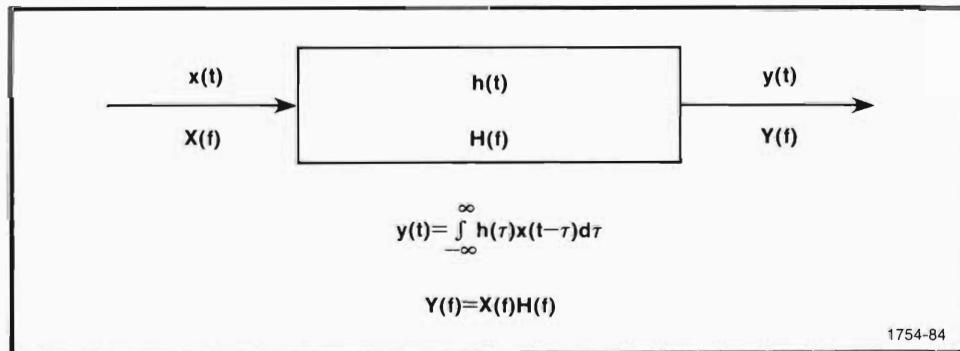


Fig. 6-4. The characterizing parameters of a linear, time-invariant system.

The FFT

6-4, where $x(t)$ is the input signal, $h(t)$ is the impulse response, and $y(t)$ is the output signal. The simplest relationship exists when these terms are looked at in the frequency domain. Then the frequency domain of the output, $Y(f)$, is equal to the product of the frequency response, $H(f)$, and the input frequency domain, $X(f)$. The frequency response can be determined by acquiring $x(t)$ and the corresponding $y(t)$ and fast Fourier transforming both to the frequency domain. $H(f)$ is then found by dividing $Y(f)$ by $X(f)$. Caution is advised in this operation since there is a potential divide-by-zero situation in computing $Y(f)/X(f)$. A divide-by-zero situation can generally be avoided, however, by placing data checks and branches in your program prior to the point where division takes place.

In fact, a general word of caution for finding frequency responses and using the other analysis techniques discussed in this section is probably appropriate here. First of all, it is best to say that the frequency response has been *estimated*. This is because digital techniques are used to represent an analog phenomenon. Even if the digital estimate is exact, it is only exact for the sample points. Anything that happens between sample points must be speculated from the points to either side. Also, physical systems do change. They change with time—they age—and they usually are only linear in their normal operating regions, and then not always exactly linear. So, even with the best data acquisition, the most accurate digitizing, and the most precise computations, you should still treat your results as estimates. Maybe A.N. Whitehead, a prominent mathematician and philosopher of the early 1900's, said it best with:

"There is no more common error than to assume that, because prolonged and accurate mathematical calculations have been made, the application of the result to some fact of nature is absolutely certain."

Convolution

Once you have obtained either the impulse response or the frequency response of a system, you have the system completely characterized. Unfortunately, however, it's usually difficult to tell exactly how a system is going to react to an input waveshape by just looking at the impulse response or the frequency response.

In order to predict the system's output waveshape for a given input waveshape, you need to solve the convolution integral. This is the integral relationship shown in Fig. 6-4. It states that the system's output signal, $y(t)$, is the convolution of the impulse response, $h(t)$, and the input signal, $x(t)$. The τ used in the integral of Fig. 6-4 is just a dummy time variable that facilitates time shifting in the convolution operation.

The integral, as shown in Fig. 6-4, can be evaluated in a fairly straightforward manner by using digital techniques. This, however, can be time consuming. It's much quicker to perform convolution by taking advantage of the fact that convolution in the time domain corresponds to multiplication in the frequency domain. Thus, convolution is performed by transforming the input signal to the frequency domain, via the FFT, to get $X(f)$. Then the impulse response is transformed to the frequency domain to get the frequency response, $H(f)$. The product of these two complex quantities is then formed

The FFT

to give $Y(f)=X(f)H(f)$. $Y(f)$ is the frequency-domain function for the system output caused by inputting $x(t)$. The time-domain function for the system output is simply obtained by inverse Fourier transforming $Y(f)$ to get $y(t)$. By knowing the frequency response of a system, the above technique can be used to predict the system's output waveshape for any input waveshape.

There are some cases, however, where you already know the output waveshape, and you'd like to know what input waveshape caused it. This, too, can be estimated as long as you know the $H(f)$ of the system.

The input waveshape can be determined by evaluating $X(f)=Y(f)/H(f)$ for $X(f)$ and then taking the inverse transform of $X(f)$ to get $x(t)$. This operation is often referred to as deconvolution, and there is need for caution here. Specifically, the complex division of the $Y(f)$ data arrays by the $H(f)$ data arrays may result in a divide-by-zero situation at some data points. You'll need to take programming steps to avoid these divide-by-zero possibilities. Also, the deconvolution procedure is sensitive to noise, and noise components in the data being deconvolved may become greatly amplified. Usable deconvolution results generally cannot be obtained without first applying systematic filtering.

As a final note regarding convolution and deconvolution with the FFT, there are several standard definitions that may be used in the algorithms. These generally differ only in scaling constants, so the results you get from convolution may differ from expected results by a multiplicative constant. You might have to do some amplitude rescaling. The waveshape, however, will be correct.

Correlation

These days, just about everyone has a feeling for what correlation means. Thanks to news media, politicians, and various special interest groups, correlation has almost become a household word. We're all probably familiar with headlines and statements that go something like this: "Surgeon General Correlates Smoking and Cancer"..."there is a high degree of correlation between education and income"... So we've come to equate correlation with terms like association, cause and effect, and similarity. These are our common usage definitions of correlation, and to a large degree, they are good intuitive definitions for the mathematical operation of correlation.

The mathematical definition of correlation goes like this:

$$r(\tau)=\lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^{T} x(t)y(t+\tau)dt,$$

where $r(\tau)$ is the correlation function formed by summing the lagged products of two waveforms, $x(t)$ and $y(t)$. τ is the time lag between $x(t)$ and $y(t)$. Functionally, correlation can be thought of as a matching up of waveform components or a similarity test between waveforms. The equivalent hardware definition in Fig. 6-5 might make this operation easier to visualize.

The FFT

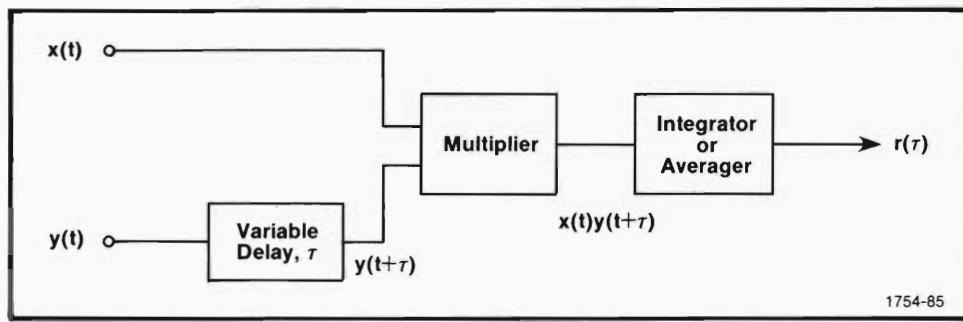


Fig. 6-5. Hardware implementation of correlation.

In terms of digital signal processing, correlation is greatly simplified by using the FFT. The two waveforms to be correlated, $x(t)$ and $y(t)$, are transformed to the frequency domain. Following this, one term is conjugated, and then the complex product is formed to give $R(f)=X(f)Y^*(f)=(X^*(f)Y(f))^*$. Here, * is used to denote conjugation. The final step is inverse transforming $R(f)$ back to the time domain to get $r(\tau)$.

Depending on the waveforms used, two types of correlation can be done. If the two waveforms are the same, $x(t)=y(t)$, their correlation is referred to as autocorrelation. If the two waveforms are different, $x(t)\neq y(t)$, their correlation is referred to as cross correlation. Let's take a brief look at each of these operations and some possible applications.

Autocorrelation. It is useful to note that the autocorrelation function of a periodic signal is periodic. Also, the autocorrelation function of a nonperiodic signal is nonperiodic. These two things are demonstrated in Fig. 6-6a and b, where a sine wave is autocorrelated and random noise is autocorrelated.

The autocorrelation functions in Fig. 6-6 are arranged so that zero time lag is at center screen. Positive time lag is to the right, and negative lag is to the left. In the case of the sine wave's autocorrelation function, maximum correlation occurs at lag zero (center screen). This is where the sine wave is exactly overlaid by itself (a perfect match). Maximum correlation also occurs for the sine wave at every lag equal to the sine wave's period; however, the autocorrelation function in Fig. 6-6a appears triangularly windowed. This apparent triangular windowing occurs because extra arrays of zero's are appended to the waveform arrays, in the manner of Fig. 6-7, before correlation. These appended zeros prevent errors from cyclic correlation, but make the sine wave appear to be pulsed instead of continuous. This pulsing causes a nearly triangular envelope for the correlation function.

Notice in Fig. 6-6b that the autocorrelation function for noise is large at lag zero and very small for all other lags. An exact match (perfect correlation) is obtained only when the noise exactly overlays itself (lag zero). For other lags, there is little or no match and the noise is said to be uncorrelated.

Fig. 6-6c shows the autocorrelation function for what appears to be random noise. From the periodicity of the autocorrelation function, however, it is obvious that a

The FFT

periodic signal is buried in the noise. Thus, autocorrelation is a useful tool for detecting the presence of periodic signals buried in noise. Biomedical studies, astronomy, and tone control systems are a few possible application areas for autocorrelation detection techniques.

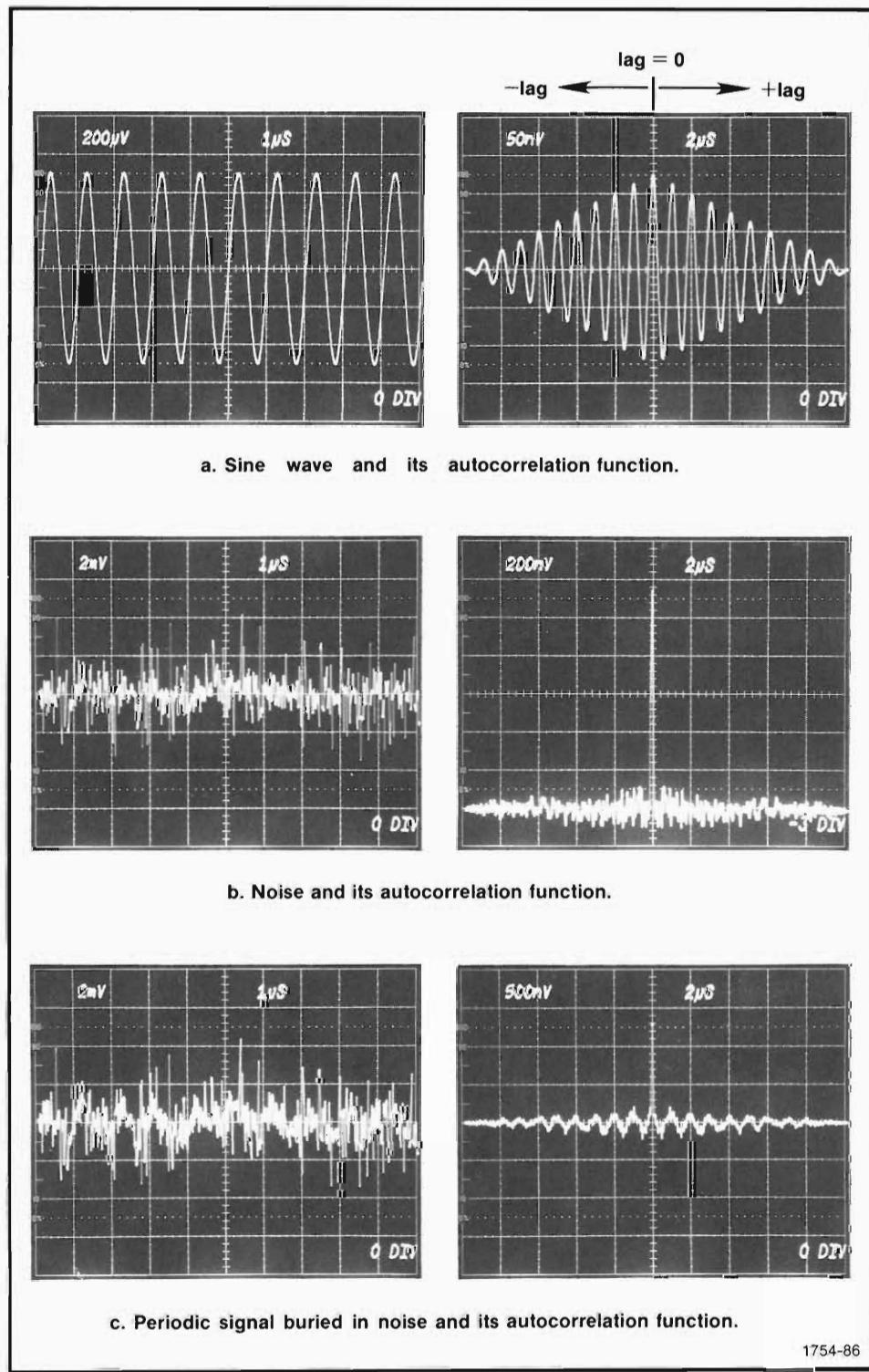
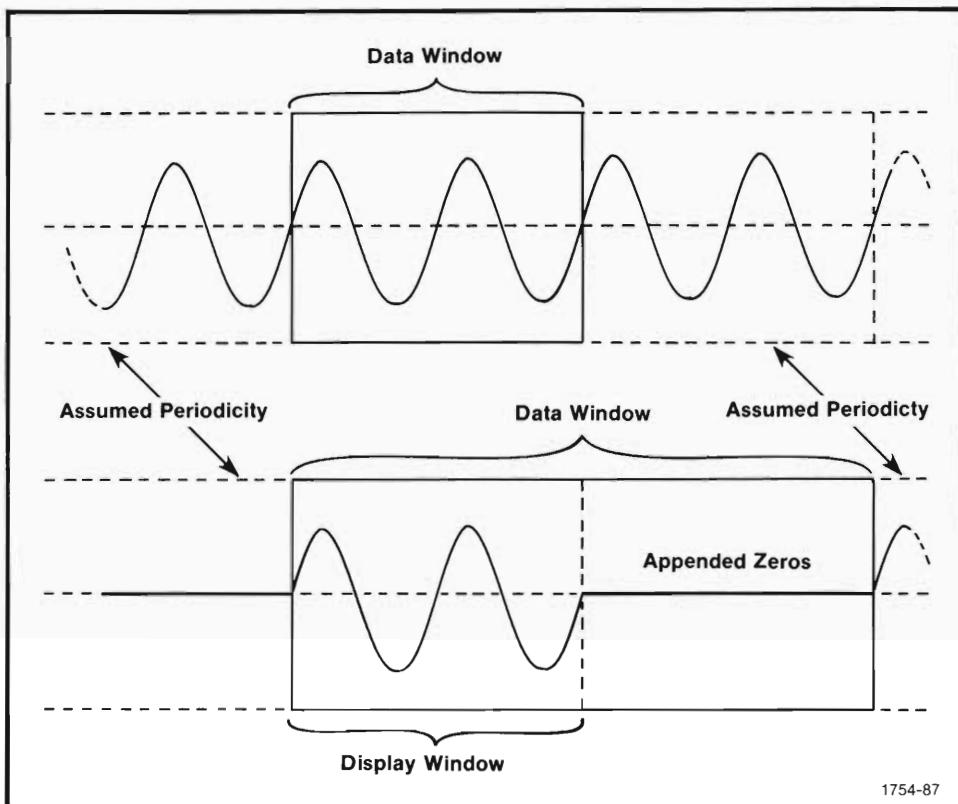


Fig. 6-6. Autocorrelation detects periodic signals buried in noise.

The FFT



1754-87

Fig. 6-7. Appending an array of zeros prevents cyclic correlation with the FFT. This, however, gives the waveform the appearance of being gated and results in a nearly triangular envelope for periodic correlation functions.

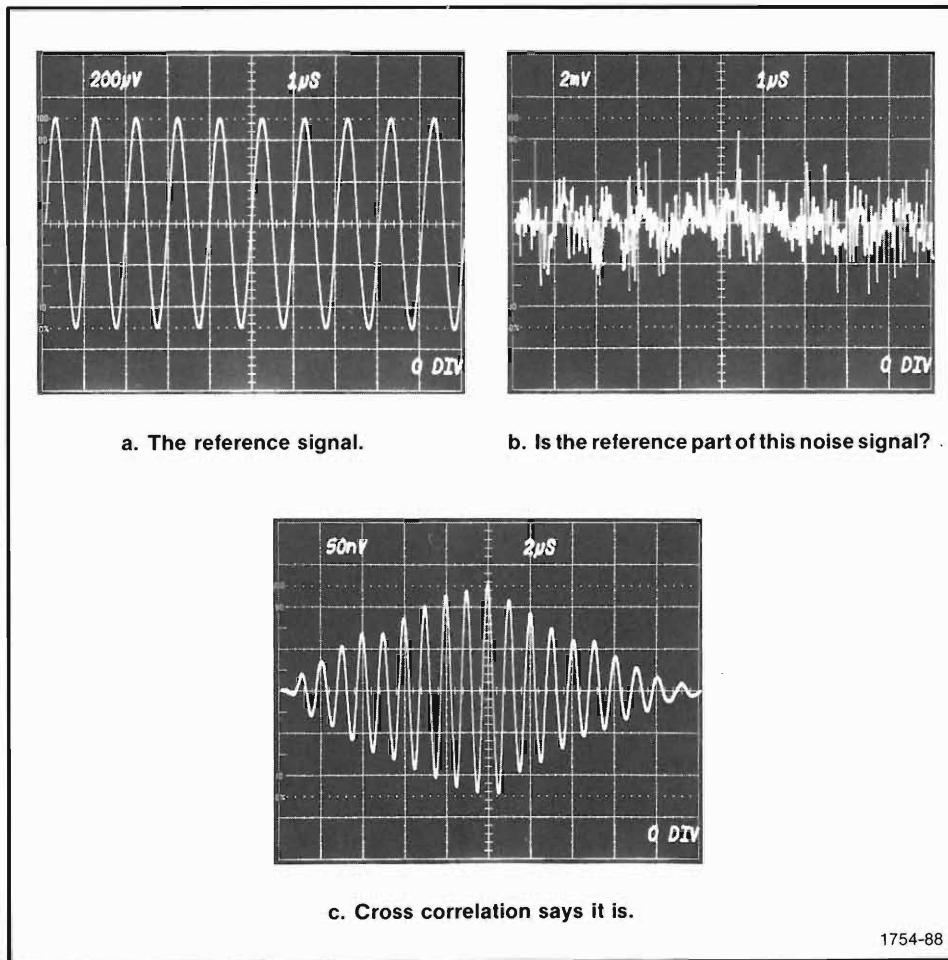
Cross Correlation. In autocorrelation, a signal is multiplied by delayed versions of itself. The process of cross correlation differs only in that two signals are used; one is multiplied by delayed versions of the other. The resulting cross-correlation function contains only those frequency components common to both waveforms.

To see the usefulness of this, let's return to the example of detecting a signal buried in noise. Suppose you're receiving signals that are obscured by noise, but you know the type of signal you are looking for. This is often the case in radar, sonar, and tone control, where the transmitted signal is well defined but the received signal is buried in noise. This is demonstrated in Fig. 6-8, where a sine wave is buried in noise. Notice in Fig. 6-8 that there are no noise components in the cross-correlation function—this is because noise is not common to the signals being correlated.

Speaking of noise, noise is usually a nuisance in most measurement situations. But, if you are interested in getting an approximation of a linear system's impulse response, noise is a useful test signal. All you need to do is drive the system input with a wideband noise source and cross correlate this input with the resulting system output. If the test is conducted carefully, the cross-correlation results will approximate the shape of the system impulse response.

The FFT

Besides detecting signals buried in noise and approximating impulse responses, cross correlation finds many applications where delays must be measured. Time delay is an important parameter in studying path diversity problems, using echo ranging techniques, or characterizing transmission systems. With cross correlation, the best match between a transmitted and received signal is found. This best match causes a maximum in the cross-correlation function, and the distance from the maximum to the lag zero point gives the delay between the two signals. The basics of this concept are illustrated in Fig. 6-9.



1754-88

Fig. 6-8. If you know the waveform you are looking for, cross correlation can help you find it.

The FFT

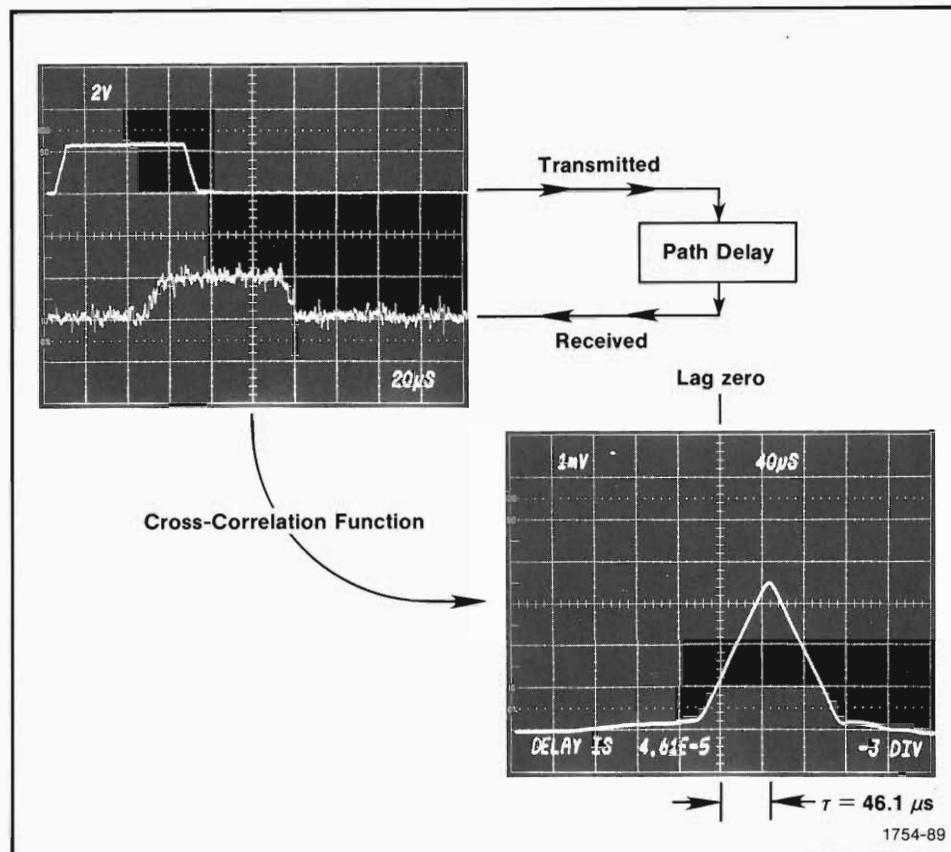


Fig. 6-9. The location of maximum cross correlation indicates the time delay between waveforms.

Power Spectra

As a final note, it should be mentioned that the FFT of an autocorrelation function results in what is called a power spectrum.

The FFT of an autocorrelation function gives what is generally referred to as the auto spectrum. Also, referred to as power spectral density or PSD, auto spectra are widely used in vibration analyses.

When a cross-correlation function is fast Fourier transformed to the frequency domain, the result is referred to as a cross spectrum. Cross spectra contain the magnitude products and phase differences of the frequency components that are common to the signals involved in the cross correlation. Like auto spectra, cross spectra are frequently used in vibration analyses too.



PART III

MATHEMATICS OF THE FFT

This part of the manual probably will neither add nor detract from your ability to use the FFT.

So why bother with PART III?

Well, first of all, many people have an intrinsic curiosity about what makes things work. For those people, PART III should satisfy some of that curiosity.

More importantly, however, PART III gives you a basis for communicating with others about the FFT. If you need to discuss various algorithms with a programmer, it will be helpful to be familiar with signal flow diagrams. Also, knowing a little about at least one FFT algorithm is helpful when reading literature on other algorithms and techniques. And lastly, as you become more expert in using the FFT, you'll find more people coming to you with questions about using the FFT and how the algorithm works. PART III will help you answer some of those questions.

SECTION 7

AN ALGORITHM FOR COMPUTING THE DFT

The FFT is not a single algorithm for computing the discrete Fourier transform. There are several algorithms that provide a basic time advantage over the N^2 operations required for straightforward evaluation of the DFT. But, because these algorithms are faster than the N^2 approach, they are all lumped under the heading of FFT.

Different FFT algorithms have been developed because different people would like to operate on different types of data, with different types of machines, while exploiting particular properties of the data or machine being used. One particular property of data is the number of samples. There are algorithms for operating on N samples where N is equal to two raised to an integer power. Others are designed to work with N equal to the product of several integers. For this discussion, however, we'll pick the class of algorithms designed for $N=2^k$, where k is an integer. The $N=2^k$ algorithms, also known as "power of two algorithms," are more straightforward and relatively faster executing than the more general algorithms.

The FFT

Frequently encountered FFTs are typically based on either the Cooley-Tukey algorithm or the Sande-Tukey algorithm. These two algorithms differ primarily in their organizational approach. The Cooley-Tukey algorithm takes an approach referred to as decimation in time, and the Sande-Tukey algorithm uses a decimation-in-frequency approach. This latter approach, decimation in frequency, is the approach we'll be looking at with the power-of-two, Sande-Tukey algorithm for computing the DFT.

The Sande-Tukey Algorithm for Computing the DFT

To begin, let's recall the expression for the DFT. This expression is discussed in Section 3 and was given as

$$X_d(k) = \frac{1}{N} \sum_{n=0}^{N-1} x_o(n) e^{-j2\pi kn/N}$$

For notational convenience, let's restate this as

$$A(n) = \sum_{t=0}^{N-1} x_o(t) W^{-nt}$$

for $n=0,1,\dots,N-1$. The time-domain data is given by $x_o(0), x_o(1), \dots, x_o(N-1)$, and W is equal to

$$e^{-j2\pi/N}$$

Since the $1/N$ term preceding the summation sign in $X_d(k)$ is simply a scaling term, it is omitted for the sake of simplifying the expressions.

Computing the FFT of $x_o(t)$ consists of $\log_2 N = M$ stages. Each stage requires pairs of computations of the form

$$\begin{aligned} x_{m+1}(r) &= x_m(r) + x_m(s) \\ \text{and} \quad x_{m+1}(s) &= (x_m(r) - x_m(s)) W^{-p} \end{aligned}$$

for specified integers r, s, p , between 0 and $N-1$ and m between 0 and $M-1$. The results at the end of the m^{th} computational stage are denoted by $x_m(t)$, where $t=0,1,\dots,N-1$. Also, the example algorithm is an in-place algorithm. This means that the current results replace the previous results; thus, $x_{m+1}(t)$ overwrites $x_m(t)$ in going from stage m to stage $m+1$. The subscript "m" merely defines a sequence of arrays that define the contents of an associated storage location at the end of the m^{th} stage.

To gain an idea of how the computations take place, let's use a 16-point FFT as an example. A superficial look will be enough to see the essence of the algorithm. If you are interested in actually implementing an FFT algorithm, several of the references listed in the Bibliography offer more detailed explanations. The discussions here will give you a general idea of the algorithm and some of the terms used in describing FFT algorithms.

The FFT

The organization of the FFT computations is shown in Fig. 7-1. Notice that the data are gathered into groups at each stage, and as each stage is passed, the groups are broken into smaller groups. This goes on until the transformation is complete, with one datum per group. This total operation is referred to as decimation in frequency. A decimation-in-time algorithm is organized in exactly the opposite manner: the computations proceed into larger and larger groups.

The data elements within each group of the decimation-in-frequency algorithm are computed from pairs of corresponding elements in the preceding groups. This is indicated in Fig. 7-1 and shown with more detail in the flow diagram of Fig. 7-2.

Each of the heavy dots in Fig. 7-2 represents a point of computation. Although these dots are spacially separated in Fig. 7-2, they are one and the same point or memory location in an in-place computation.

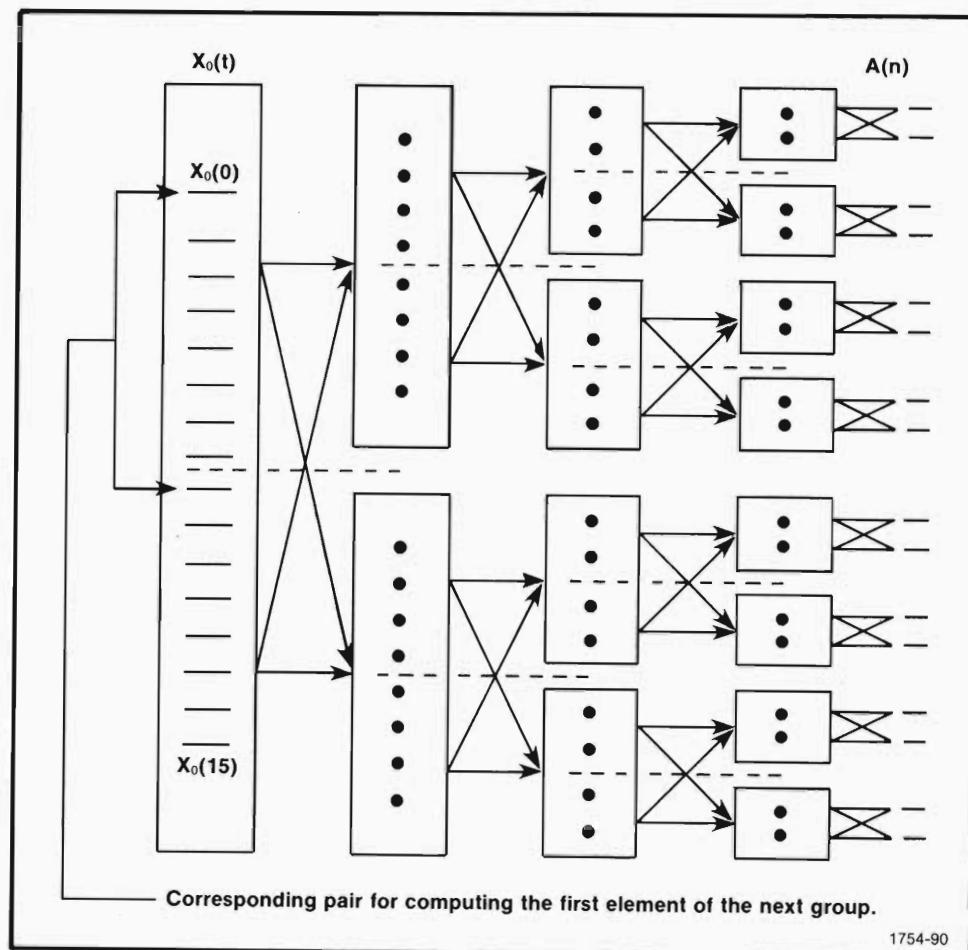


Fig. 7-1. The general organization of a decimation-in-frequency algorithm.

The FFT

As was mentioned earlier, the computations take place in pairs and are of the form

$$x_{m+1}(r) = x_m(r) + x_m(s)$$

and

$$x_{m+1}(s) = (x_m(r) - x_m(s))W^{-p}$$

An example of such a computation pair is shown in Fig. 7-2 at the x_1 stage. There, $x_0(7)$ and $x_0(15)$ are added to obtain $x_1(7)$. Also, at the next node, $x_0(0)$ is multiplied by W^0 and added to the product of $x_0(8)$ and $-W^0$ to obtain $x_1(8)$.

Notice in Fig. 7-2 that the operations are grouped at each stage according to whether or not the elements must be multiplied by the W^p factor. This organization efficiency is further augmented by taking advantage of sine and cosine symmetries to obtain the "twiddle factors" indicated by W^0, W^{-1} , etc. Depending on the particular FFT algorithm, these twiddle factors may be individually generated at each stage, or they may be called from a previously generated table.

After the final stage of the algorithm has been passed through, $x_o(t)$ will have been transformed to $A(n)$ —the time-domain data has been transformed to frequency-domain data. As you can see from the final stage in Fig. 7-2, $A(n)$ occurs in scrambled order. The order of these Fourier coefficients can be unscrambled by a process referred to as bit reversal. If we re-express $A(0)$ through $A(15)$ in binary code, the correct location of each frequency component can be determined by reversing or flipping the address bits. This is shown in Table 7-1 and is the final step of the transformation.

Table 7-1

Bit Reversal Places the Fourier Coefficients into Correct Order

A(0)	= A(0000)	bit reversal	A(0000)	= A(0)
A(8)	A(1000)		A(0001)	A(1)
A(4)	A(0100)		A(0010)	A(2)
A(12)	A(1100)		A(0011)	A(3)
.	.		.	.
.	.		.	.
A(11)	A(1011)		A(1101)	A(13)
A(7)	A(0111)		A(1110)	A(14)
A(15)	A(1111)		A(1111)	A(15)

Admittedly, this discussion of the FFT algorithm has been brief. But then its purpose is only to instill a small measure of familiarity with the algorithm. If you would like to study the Sande-Turkey algorithm, or any other algorithm, in greater depth, the following bibliography contains a number of good sources for algorithm information.

The FFT

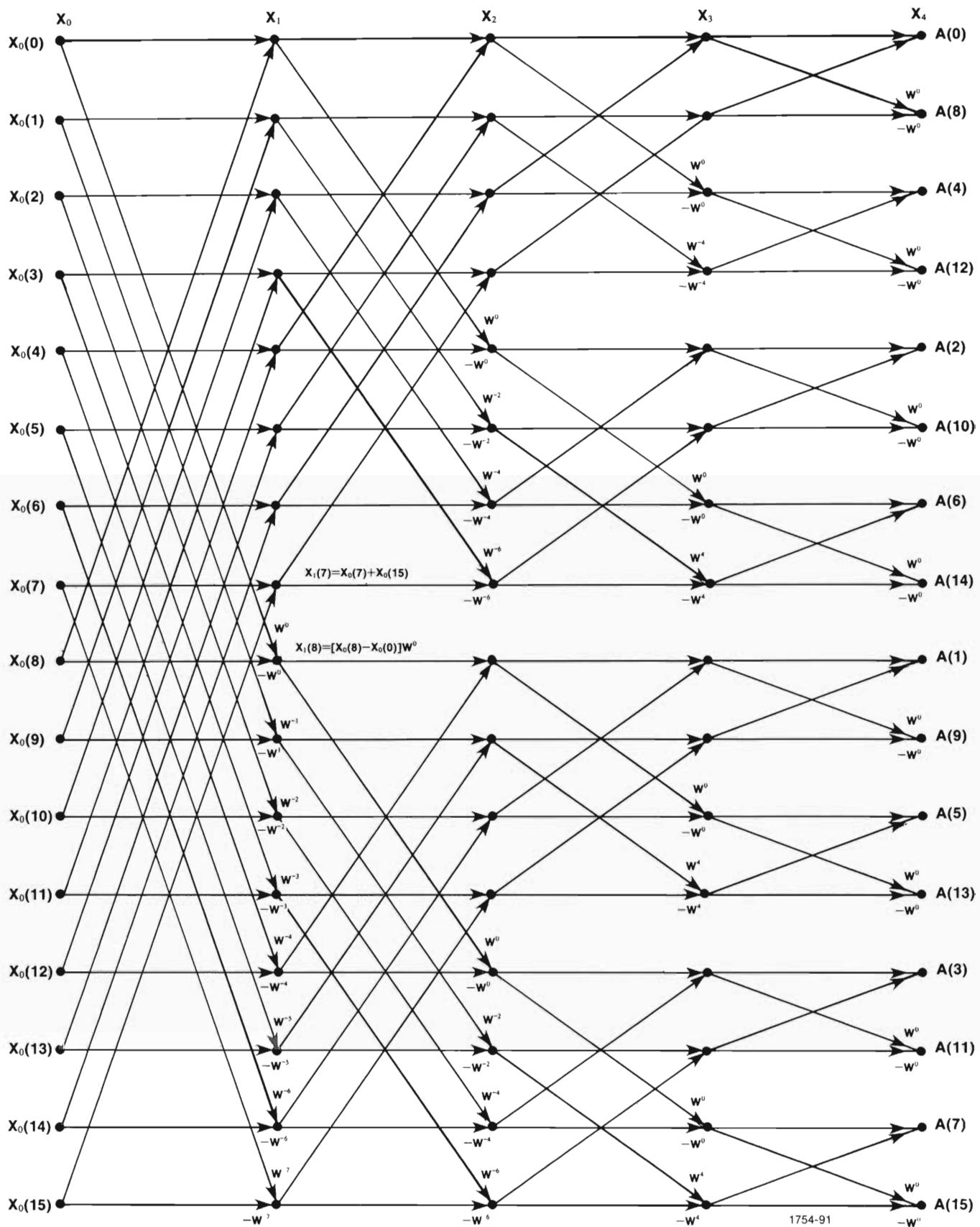
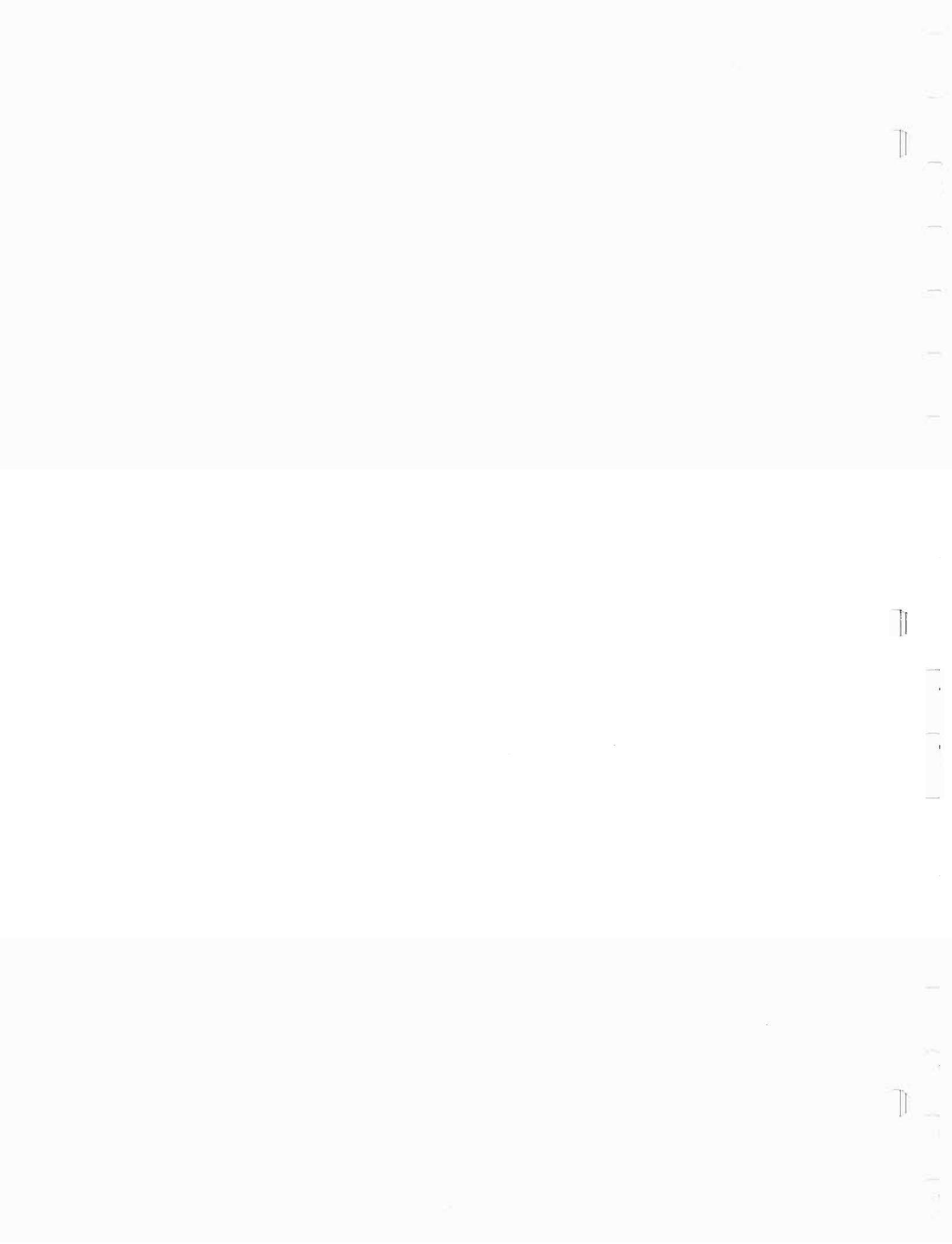


Fig. 7-2. Flow diagram for a 16-point, decimation-in-frequency FFT algorithm.



BIBLIOGRAPHY

Bendat, J.S., and A.G. Piersol, *Random Data: Analysis and Measurement Procedures*. New York: Wiley, 1971.

Bergland, G.D., "A Guided Tour of the Fast Fourier Transform," *IEEE Spectrum*, July 1969, pp. 41-52.

Blackman, R.B., and J.W. Tukey, *The Measurement of Power Spectra*. New York: Dover Publications, 1958.

Bracewell, R., *The Fourier Transform and Its Applications*. New York: McGraw-Hill, 1965.

Brigham, E.O., *The Fast Fourier Transform*. Englewood Cliffs: Prentice-Hall, 1974.

Brigham, E.O., and R.E. Morrow, "The Fast Fourier Transform," *IEEE Spectrum*, Dec. 1967, pp. 63-70.

Chirlian, P.M., *Basic Network Theory*. New York: McGraw-Hill, 1969.

Cooley, J.W., and J.W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, Vol. 19, April 1965, pp. 297-301.

Gold, B., and C.M. Rader, *Digital Processing of Signals*. New York: McGraw, 1969.

Griffiths, J.W.R., P.L. Stocklin, and C. Van Schooneveld, eds., *Signal Processing*, Proceedings of the NATO Advanced Study Institute on Signal Processing. New York: Academic Press, 1973.

IEEE Transactions on Audio and Electroacoustics, Special issue on the fast Fourier transform and its application to digital filtering and spectral analysis, Vol. AU-15, No. 2, June 1967.

IEEE Transactions on Audio and Electroacoustics, Special issue on the fast Fourier transform, Vol. AU-17, No. 2, June 1969.

Jenkins, G.M., and D.G. Watts, *Spectral Analysis and Its Applications*. San Francisco: Holden-Day, 1968.

Lange, F.H., *Correlation Techniques*. London: Iliffe Books Ltd., 1967. Published in the U.S.A. by D. Van Nostrand Company, Princeton, N.J.

The FFT

Oppenheim, A.V., and R.W. Schafer, *Digital Signal Processing*, Englewood Cliffs: Prentice-Hall, 1975.

Otnes, R.K., and L. Enochson, *Digital Time Series Analysis*. New York: Wiley, 1972.

Papoulis, A., *The Fourier Integral and Its Applications*. New York: McGraw-Hall, 1962.

Proceedings of the IEEE, Special issue on digital signal processing, Vol. 63, No. 4, April 1975.

Rabiner, L.R., and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs: Prentice-Hall, 1975.

Rabiner, L.R., et al, "Terminology in Digital Signal Processing," *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-20, Dec. 1972, pp. 322-337.

Ramirez, R.W., "Fast Fourier Transform Makes Correlation Simpler," *Electronics*, June 26, 1975, pp. 98-103.

Ramirez, R.W., "The Fast Fourier Transform's Errors Are Predictable, Therefore Manageable," *Electronics*, June 13, 1974, pp. 96-102.

Roth, P.R. "Effective Measurements Using Digital Signal Processing," *IEEE Spectrum*, April 1971, pp. 62-70.

Schafer, R.W., and L.R. Rabiner, "A Digital Signal Processing Approach to Interpolation," *Proceedings of the IEEE*, Vol. 61, June 1973, pp. 692-702.

Van Valkenburg, M.E., *Network Analysis*, 2nd ed. Englewood Cliffs: Prentice-Hall, 1964

INDEX

A

additive noise 4-16
 advance 1-6, 1-8
 aliasing 4-36, 5-6
 almost periodic data 5-14
 analog-to-digital conversion 3-3, 4-19,
 4-24
 antialiasing filters 4-41
 arrays 4-2
 autocorrelation 6-9
 auto spectrum 6-13
 averaging 4-16, 4-22, 5-2

B

band limiting 4-41
 bandwidth 5-15
 bit reversal 7-4
 Bunsen, Robert W. 2-5

C

complex numbers 2-21
 complex signals 2-17, 4-4
 continuous phase 1-6, 4-6, 4-9
 continuous spectrum 2-13, 4-24
 conversion noise 4-19
 convolution 2-32, 4-24, 5-2, 6-7
 Cooley, J.W. 2-10
 Cooley-Tukey algorithm 3-11, 4-1, 7-2
 correlation 5-2, 6-8
 cosine squared window 5-16
 cosine wave 2-31, 3-2
 cosine window (Hanning) 5-16
 counter, frequency 1-3
 cross correlation 6-11
 cross spectrum 6-13
 CRT photos 1-5

D

Danielson 3-9
 dc component 3-6, 5-3
 decimation in frequency 7-2
 decimation in time 7-2
 deconvolution 6-8
 delay 1-6, 1-8, 2-27, 4-6, 5-9, 6-12
 DFT 3-1, 3-5, 7-1
 digital viewpoint 4-1
 Dirichlet conditions 2-2, 2-12
 Dirichlet, P.G.L. 2-2
 discontinuities 2-2, 2-3, 2-7, 5-14
 discrete Fourier transform (DFT) 2-38,
 3-1, 3-5, 7-1
 discrete spectrum 2-5, 4-6
 distortion 5-9, 6-1
 DPO TEK BASIC 4-2, 4-5
 duty factor 2-13

E

Euler, Leonard 2-2
 even functions 2-23, 2-34, 4-30, 5-1, 5-10
 existence of Fourier series 2-2
 extended cosine bell window 5-15, 5-16

F

fast Fourier transform (FFT) 2-38, 3-1,
 3-9, 7-1
 FFT 2-38, 3-1, 3-9, 7-1
 FFT statement 4-3
 flow diagram, FFT 7-5
 folding frequency 4-39
 Fourier coefficients 3-6, 3-7, 4-5
 Fourier integral 2-1, 2-10, 2-29, 2-37
 Fourier, J.B.J. 2-1, 2-2
 Fourier series 2-1, 2-4, 2-8, 2-29, 4-9
 existence 2-2
 exponential form 2-11

The FFT

Fourier transform 2-12, 2-29, 2-33
Fraunhofer, Joseph 2-5
frequency counter 1-3
frequency decimation 7-2
frequency-domain diagrams 2-15
frequency plane 1-6, 1-7
frequency response 6-5
frequency scaling 2-34, 5-2
frequency shifting 2-37, 5-2
full-wave rectifier 2-9
fundamental frequency 2-6, 4-14

G

Galileo 1-1, 1-2
Garwin, R.L. 3-10
geological survey 6-5
Gibb's phenomenon 2-6

H

half-cycle sine cubed window 5-16
half-cycle sine window 5-16
half-wave rectifier 2-9
Hamming window 5-15, 5-16
Hanning window (cosine) 5-16
hardware FFT 3-11
harmonic frequency 2-6, 2-11, 4-14, 5-7,
 6-1
Huygens, Christiaan 1-1

I

IFT statement 4-5
imaginary part 2-20, 2-34, 4-4
impulse response 6-5, 6-6, 6-11
impulse train 3-3, 4-26, 4-35
in-place algorithm 7-2, 7-3
intermodulation distortion 6-2
interpolation 5-13
inverse DFT 3-5
inverse FFT 4-5, 5-1
inverse Fourier transform 2-12, 2-33

J

jitter 4-19

K

Kirchhoff, Gustav 2-5

L

Lanczos 3-9
leakage 4-22, 5-4, 5-13
Lebesgue, Henri 2-2
line spectrum 2-5, 4-5, 4-6, 4-8, 4-24
linearity 6-2
linearity property 2-34, 4-16, 5-1
low-level components 4-8

M

magnitude 2-20, 4-6
magnitude diagram 1-6, 1-7, 2-2, 2-16
mean value 5-3
mechanical vibration 6-3
modulo 2π phase 1-6, 4-6, 4-11

N

negative frequency 2-15, 2-17
Newton 2-5
noise 4-7, 4-16, 4-19, 5-2, 6-10, 6-11
nonperiodic waveform 2-12, 2-27, 4-30,
 4-35, 5-4
Nyquist frequency 4-5, 4-26, 4-36, 5-6
Nyquist sampling theorem 4-36

O

odd functions 2-24, 2-34, 4-30, 5-1
off-line operation 3-12
overshoot 2-7

The FFT

P

Parzen window 5-15, 5-16
pendulum 1-1, 1-2
periodicity 2-2, 2-27, 4-30, 5-4
phase 2-20, 4-6, 4-10, 5-9
phase diagram 1-6, 1-7, 1-9, 2-2, 2-16
polar form, 2-20, 4-6
power spectrum 6-13
power spectral density (PSD) 6-13
PSD 6-13
pulse train 2-9, 2-13

Q

quantizing error 4-19

R

readout, CRT 1-5
real part 2-20, 2-34, 4-4
real-time results 3-12
real-valued data 4-3
rectangular form 2-20, 4-4
rectangular window 2-29, 3-3, 4-23, 5-16
rectifier 2-9
resolution 4-22, 5-10, 5-17
Riemann, Bernhard 2-2
ringing 2-8
round off 3-9, 4-5
Runge, C. 3-9

S

sampling 3-2, 4-24, 4-35, 5-6, 5-10
sampling interval 3-5, 3-6, 4-26, 5-6, 5-10
Sande-Tukey algorithm 4-1, 7-2
sawtooth wave 2-9, 2-24
selectivity 5-17
side lobes 5-15
signal averaging 4-16, 4-22, 5-2
signal-to-noise ratio 4-16, 5-3
signature analysis 6-3
sin x/x function 2-19, 4-23
sine wave 1-7
skirts 5-17

software FFT 3-11
spectral diagrams 2-5
spectrum analyzer 1-13, 4-6
square pulse 2-19, 2-22, 2-26, 5-10
square wave 2-3, 2-4, 2-9, 4-14, 4-39
step response 6-6
summing frequency components 1-10,
2-26, 2-34
symmetry 2-23, 2-24, 4-14
system frequency response 6-5

T

TEK BASIC 4-2
Thomas, L.H. 3-9
time base 1-4
time decimation 7-2
time history 1-4
time jitter 4-19
time plane 1-6, 1-7
time scaling 2-34, 5-2
time shifting 2-27, 2-36, 4-10, 4-13
total harmonic distortion 6-1
transfer function 6-5
triangular wave 2-9
triangular window 4-29, 5-15, 5-16
truncation 2-6
Tukey, J.W. 2-10
twiddle factors 7-4

V

vibration analysis 6-3
viewpoint 2-27, 4-1, 4-5, 4-8, 5-4

W

waveform arrays 4-2
waveform space 1-6, 1-7
WDI TEK BASIC 4-2, 4-5
Whitehead, A.N. 6-7
windowing 2-29, 3-2, 4-22, 4-26, 5-13,
5-16

The FFT

Z

zero reference 1-5

YOUR COMMENTS COUNT

The Manual Writers at Tektronix, Inc. are interested in what you think about this manual, how you use it, and changes you might like to see in future manuals. Any queries regarding this manual will be answered personally.

What did you find that was:

interesting? _____

frustrating? _____

helpful? _____

confusing? _____

Is there anything you would like to see added to or deleted from this manual? _____

What is your major application area for this product? _____

Have you found any interesting applications, operating hints, or software routines which you would like to share with us? _____

* * * *

Name: _____ Position: _____

Company: _____ Department: _____

Street: _____

City: _____ State: _____ Zip: _____

FIRST CLASS
PERMIT NO. 61
BEAVERTON, OREGON

BUSINESS REPLY MAIL

No postage necessary if mailed in the United States

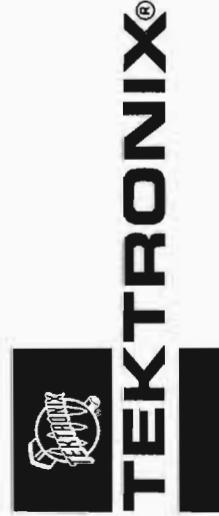
Postage will be paid by

TEKTRONIX, INC.

P.O. Box 500

Beaverton, Oregon 97005

ATTN: SPS Manuals 58-157



Some Common Data Windows and Their Frequency-Domain Parameters

Unity Amplitude Window	Shape Equation	Frequency Domain Magnitude	Major Lobe Height (dB)	Highest Side Lobe (dB)	Band-width (3 dB) (dB)	Theoretical Roll-Off (dB/Octave)
Rectangle			T	-13.2	0.86β	6
Extended Cosine Bell	$A=0.5(1-\cos 2\pi 5t/T)$ for $t=0$ to $T/10$ and $t=9T/10$ to T $A=1$ for $t=T/10$ to $9T/10$		0.9 T	-13.5	0.95β	18 (beyond 5β)
Half Cycle Sine	$A=\sin 2\pi 0.5t/T$ for $t=0$ to T		0.64 T	-22.4	1.15β	12
Triangle	$A=2t/T$ for $t=0$ to $T/2$ $A=-2t/T + 2$ for $t=T/2$ to T		0.5 T	-26.7	1.27β	12
Cosine (Hanning)	$A=0.5(1-\cos 2\pi t/T)$ for $t=0$ to T		0.5 T	-31.6	1.39β	18
Half Cycle Sine ³	$A=\sin^3 2\pi 0.5t/T$ for $t=0$ to T		0.42 T	-39.5	1.61β	24
Hamming	$A=0.08 + 0.46(1-\cos 2\pi t/T)$ for $t=0$ to T		0.54 T	-41.9	1.26β	6 (Beyond 5β)
Cosine ²	$A=(0.5(1-\cos 2\pi t/T))^2$ for $t=0$ to T		0.36 T	-46.9	1.79β	30
Parzen	$A=1-6(2t/T-1)^2+6 2t/T-1 ^3$ for $t=T/4$ to $3T/4$ $A=2(1- 2t/T-1 ^3)$ for $t=0$ to $T/4$ and $t=3T/4$ to T		0.37 T	-53.2	1.81β	24

On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform

FREDRIC J. HARRIS, MEMBER, IEEE

Abstract—This paper makes available a concise review of data windows and their effect on the detection of harmonic signals in the presence of broad-band noise, and in the presence of nearby strong harmonic interference. We also call attention to a number of common errors in the application of windows when used with the fast Fourier transform. This paper includes a comprehensive catalog of data windows along with their significant performance parameters from which the different windows can be compared. Finally, an example demonstrates the use and value of windows to resolve closely spaced harmonic signals characterized by large differences in amplitude.

I. INTRODUCTION

HERE IS MUCH signal processing devoted to detection and estimation. Detection is the task of determining if a specific signal set is present in an observation, while estimation is the task of obtaining the values of the parameters describing the signal. Often the signal is complicated or is corrupted by interfering signals or noise. To facilitate the detection and estimation of signal sets, the observation is decomposed by a basis set which spans the signal space [1]. For many problems of engineering interest, the class of signals being sought are periodic which leads quite naturally to a decomposition by a basis consisting of simple periodic functions, the sines and cosines. The classic Fourier transform is the mechanism by which we are able to perform this decomposition.

By necessity, every observed signal we process must be of finite extent. The extent may be adjustable and selectable, but it must be finite. Processing a finite-duration observation imposes interesting and interacting considerations on the harmonic analysis. These considerations include detectability of tones in the presence of nearby strong tones, resolvability of similar-strength nearby tones, resolvability of shifting tones, and biases in estimating the parameters of any of the aforementioned signals.

For practicality, the data we process are N uniformly spaced samples of the observed signal. For convenience, N is highly composite, and we will assume N is even. The harmonic estimates we obtain through the discrete Fourier transform (DFT) are N uniformly spaced samples of the associated periodic spectra. This approach is elegant and attractive when the processing scheme is cast as a spectral decomposition in an N -dimensional orthogonal vector space [2]. Unfortunately, in many practical situations, to obtain meaningful results this elegance must be compromised. One such

Manuscript received September 10, 1976; revised April 11, 1977 and September 1, 1977. This work was supported by Naval Undersea Center (now Naval Ocean Systems Center) Independent Exploratory Development Funds.

The author is with the Naval Ocean Systems Center, San Diego, CA, and the Department of Electrical Engineering, School of Engineering, San Diego State University, San Diego, CA 92182.

compromise consists of applying windows to the sampled data set, or equivalently, smoothing the spectral samples.

The two operations to which we subject the data are sampling and windowing. These operations can be performed in either order. Sampling is well understood, windowing is less so, and sampled windows for DFT's significantly less so! We will address the interacting considerations of window selection in harmonic analysis and examine the special considerations related to sampled windows for DFT's.

II. HARMONIC ANALYSIS OF FINITE-EXTENT DATA AND THE DFT

Harmonic analysis of finite-extent data entails the projection of the observed signal on a basis set spanning the observation interval [1], [3]. Anticipating the next paragraph, we define T seconds as a convenient time interval and NT seconds as the observation interval. The sines and cosines with periods equal to an integer submultiple of NT seconds form an orthogonal basis set for continuous signals extending over NT seconds. These are defined as

$$\left. \begin{array}{l} \cos \left[\frac{2\pi}{NT} kt \right] \\ \sin \left[\frac{2\pi}{NT} kt \right] \end{array} \right\} \quad \begin{array}{l} k = 0, 1, \dots, N-1, N, N+1, \dots \\ 0 \leq t < NT. \end{array} \quad (1)$$

We observe that by defining a basis set over an ordered index k , we are defining the spectrum over a line (called the frequency axis) from which we draw the concepts of bandwidth and of frequencies close to and far from a given frequency (which is related to resolution).

For sampled signals, the basis set spanning the interval of NT seconds is identical with the sequences obtained by uniform samples of the corresponding continuous spanning set up to the index $N/2$,

$$\left. \begin{array}{l} \cos \left[\frac{2\pi}{NT} knT \right] = \cos \left[\frac{2\pi}{N} kn \right] \\ \sin \left[\frac{2\pi}{NT} knT \right] = \sin \left[\frac{2\pi}{N} kn \right] \end{array} \right\} \quad \begin{array}{l} k = 0, 1, \dots, N/2 \\ n = 0, 1, \dots, N-1. \end{array} \quad (2)$$

We note here that the trigonometric functions are unique in that uniformly spaced samples (over an integer number of periods) form orthogonal sequences. Arbitrary orthogonal functions, similarly sampled, do not form orthogonal sequences. We also note that an interval of length NT seconds is not the same as the interval covered by N samples separated by intervals of T seconds. This is easily understood when we

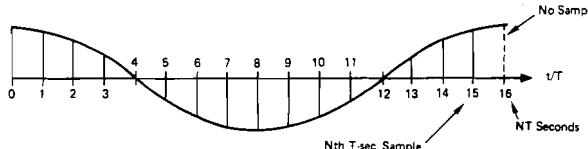


Fig. 1. N samples of an even function taken over an NT second interval.

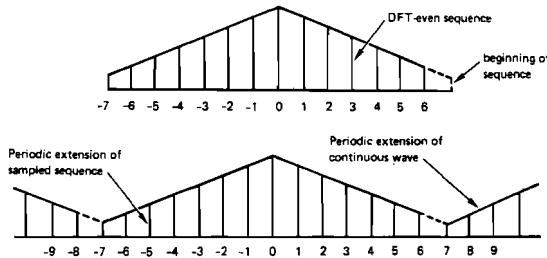


Fig. 2. Even sequence under DFT and periodic extension of sequence under DFT.

realize that the interval over which the samples are taken is closed on the left and is open on the right (i.e., [—]). Fig. 1 demonstrates this by sampling a function which is even about its midpoint and of duration NT seconds.

Since the DFT essentially considers sequences to be periodic, we can consider the missing end point to be the beginning of the next period of the periodic extension of this sequence. In fact, under the periodic extension, the next sample (at 16 s in Fig. 1.) is indistinguishable from the sample at zero seconds.

This apparent lack of symmetry due to the missing (but implied) end point is a source of confusion in sampled window design. This can be traced to the early work related to convergence factors for the partial sums of the Fourier series. The partial sums (or the finite Fourier transform) always include an odd number of points and exhibit even symmetry about the origin. Hence much of the literature and many software libraries incorporate windows designed with true even symmetry rather than the implied symmetry with the missing end point!

We must remember for DFT processing of sampled data that even symmetry means that the projection upon the sampled sine sequences is identically zero; it does not mean a matching left and right data point about the midpoint. To distinguish this symmetry from conventional evenness we will refer to it as DFT-even (i.e., a conventional even sequence with the right-end point removed). Another example of DFT-even symmetry is presented in Fig. 2 as samples of a periodically extended triangle wave.

If we evaluate a DFT-even sequence via a finite Fourier transform (by treating the $+N/2$ point as a zero-value point), the resultant continuous periodic function exhibits a non zero imaginary component. The DFT of the same sequence is a set of samples of the finite Fourier transform, yet these samples exhibit an imaginary component equal to zero. Why the disparity? We must remember that the missing end point under the DFT symmetry contributes an imaginary sinusoidal component of period $2\pi/(N/2)$ to the finite transform (corresponding to the odd component at sequence position $N/2$). The sampling positions of the DFT are at the multiples of $2\pi/N$, which, of course, correspond to the zeros of the imaginary sinusoidal component. An example of this fortuitous sampling is shown in Fig. 3. Notice the sequence $f(n)$,

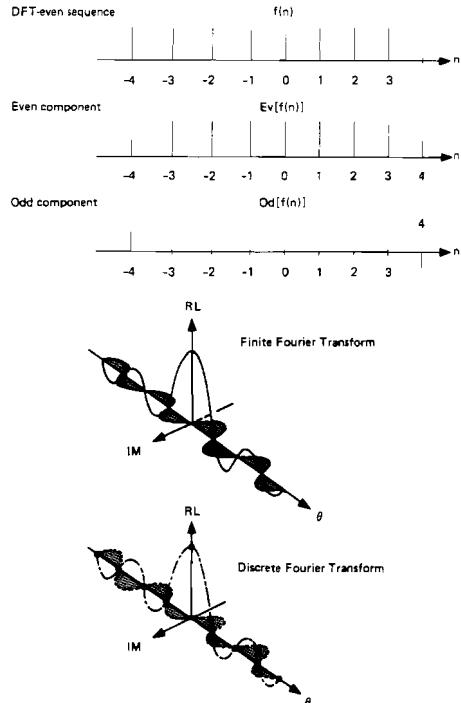


Fig. 3. DFT sampling of finite Fourier transform of a DFT even sequence.

is decomposed into its even and odd parts, with the odd part supplying the imaginary sine component in the finite transform.

III. SPECTRAL LEAKAGE

The selection of a finite-time interval of NT seconds and of the orthogonal trigonometric basis (continuous or sampled) over this interval leads to an interesting peculiarity of the spectral expansion. From the continuum of possible frequencies, only those which coincide with the basis will project onto a single basis vector; all other frequencies will exhibit non zero projections on the entire basis set. This is often referred to as spectral leakage and is the result of processing finite-duration records. Although the amount of leakage is influenced by the sampling period, leakage is not caused by the sampling.

An intuitive approach to leakage is the understanding that signals with frequencies other than those of the basis set are not periodic in the observation window. The periodic extension of a signal not commensurate with its natural period exhibits discontinuities at the boundaries of the observation. The discontinuities are responsible for spectral contributions (or leakage) over the entire basis set. The forms of this discontinuity are demonstrated in Fig. 4.

Windows are weighting functions applied to data to reduce the spectral leakage associated with finite observation intervals. From one viewpoint, the window is applied to data (as a multiplicative weighting) to reduce the order of the discontinuity at the boundary of the periodic extension. This is accomplished by matching as many orders of derivative (of the weighted data) as possible at the boundary. The easiest way to achieve this matching is by setting the value of these derivatives to zero or near to zero. Thus windowed data are smoothly brought to zero at the boundaries so that the periodic extension of the data is continuous in many orders of derivative.

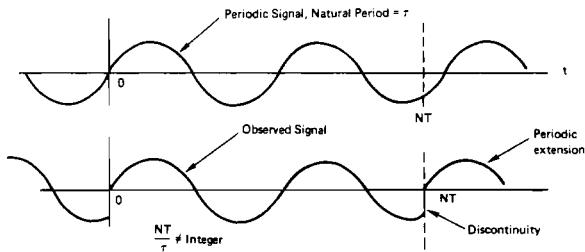


Fig. 4. Periodic extension of sinusoid not periodic in observation interval.

From another viewpoint, the window is multiplicatively applied to the basis set so that a signal of arbitrary frequency will exhibit a significant projection only on those basis vectors having a frequency close to the signal frequency. Of course both viewpoints lead to identical results. We can gain insight into window design by occasionally switching between these viewpoints.

IV. WINDOWS AND FIGURES OF MERIT

Windows are used in harmonic analysis to reduce the undesirable effects related to spectral leakage. Windows impact on many attributes of a harmonic processor; these include detectability, resolution, dynamic range, confidence, and ease of implementation. We would like to identify the major parameters that will allow performance comparisons between different windows. We can best identify these parameters by examining the effects on harmonic analysis of a window.

An essentially bandlimited signal $f(t)$ with Fourier transform $F(\omega)$ can be described by the uniformly sampled data set $f(nT)$. This data set defines the periodically extended spectrum $F^T(\omega)$ by its Fourier series expansion as identified as

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) \exp(-j\omega t) dt \quad (3a)$$

$$F^T(\omega) = \sum_{n=-\infty}^{+\infty} f(nT) \exp(-j\omega nT) \quad (3b)$$

$$f(t) = \int_{-\pi/T}^{+\pi/T} F^T(\omega) \exp(+j\omega t) d\omega / 2\pi \quad (3c)$$

$$|F(\omega)| = 0, \quad |\omega| \geq \frac{1}{2}[2\pi/T]$$

and where

$$F^T(\omega) = F(\omega), \quad |\omega| \leq \frac{1}{2}[2\pi/T].$$

For (real-world) machine processing, the data must be of finite extent, and the summation of (3b) can only be performed as a finite approximation as indicated as

$$F_a(\omega) = \sum_{n=-N/2}^{+N/2} f(nT) \exp(-j\omega nT), \quad N \text{ even} \quad (4a)$$

$$F_b(\omega) = \sum_{n=-N/2}^{(N/2)-1} f(nT) \exp(-j\omega nT), \quad N \text{ even} \quad (4b)$$

$$F_c(\omega_k) = \sum_{n=-N/2}^{(N/2)-1} f(nT) \exp(-j\omega_k nT), \quad N \text{ even} \quad (4c)$$

$$F_d(\omega_k) = \sum_{n=0}^{N-1} f(nT) \exp(-j\omega_k nT), \quad N \text{ even} \quad (4d)$$

where

$$\omega_k = \frac{2\pi}{NT} k, \quad \text{and } k = 0, 1, \dots, N-1.$$

We recognize (4a) as the finite Fourier transform, a summation addressed for the convenience of its even symmetry. Equation (4b) is the finite Fourier transform with the right-end point deleted, and (4c) is the DFT sampling of (4b). Of course for actual processing, we desire (for counting purposes in algorithms) that the index start at zero. We accomplish this by shifting the starting point of the data $N/2$ positions, changing (4c) to (4d). Equation (4d) is the forward DFT. The $N/2$ shift will affect only the phase angles of the transform, so for the convenience of symmetry we will address the windows as being centered at the origin. We also identify this convenience as a major source of window misapplication. The shift of $N/2$ points and its resultant phase shift is often overlooked or is improperly handled in the definition of the window when used with the DFT. This is particularly so when the windowing is performed as a spectral convolution. See the discussion on the Hanning window under the $\cos^\alpha(X)$ windows.

The question now posed is, to what extent is the finite summation of (4b) a meaningful approximation of the infinite summation of (3b)? In fact, we address the question for a more general case of an arbitrary window applied to the time function (or series) as presented in

$$F_w(\omega) = \sum_{n=-\infty}^{+\infty} w(nT) f(nT) \exp(-j\omega nT) \quad (5)$$

where

$$w(nT) = 0, \quad |n| > \frac{N}{2}, \quad N \text{ even}$$

and

$$w(nT) = w(-nT), \quad n \neq \frac{N}{2}, \quad w\left(\frac{N}{2}T\right) = 0.$$

Let us now examine the effects of the window on our spectral estimates. Equation (5) shows that the transform $F_w(\omega)$ is the transform of a product. As indicated in the following equation, this is equivalent to the convolution of the two corresponding transforms (see Appendix):

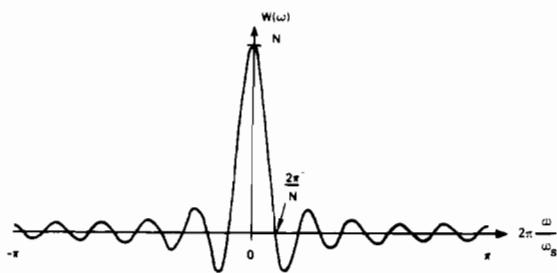
$$F_w(\omega) = \int_{-\infty}^{+\infty} F(x) W(\omega - x) dx / 2\pi \quad (6)$$

or

$$F_w(\omega) = F(\omega) * W(\omega).$$

Equation (6) is the key to the effects of processing finite-extent data. The equation can be interpreted in two equivalent ways, which will be more easily visualized with the aid of an example. The example we choose is the sampled rectangle window; $w(nT) = 1.0$. We know $W(\omega)$ is the Dirichlet kernel [4] presented as

$$W(\omega) = \exp\left(+j\frac{\omega T}{2}\right) \frac{\sin\left[\frac{N}{2}\omega T\right]}{\sin\left[\frac{1}{2}\omega T\right]}. \quad (7)$$

Fig. 5. Dirichlet kernel for N point sequence.

Except for the linear phase shift term (which will change due to the $N/2$ point shift for realizability), a single period of the transform has the form indicated in Fig. 5. The observation concerning (6) is that the value of $F_w(\omega)$ at a particular ω , say $\omega = \omega_0$, is the sum of all of the spectral contributions at each ω weighted by the window centered at ω_0 and measured at ω (see Fig. 6).

A. Equivalent Noise Bandwidth

From Fig. 6, we observe that the amplitude of the harmonic estimate at a given frequency is biased by the accumulated broad-band noise included in the bandwidth of the window. In this sense, the window behaves as a filter, gathering contributions for its estimate over its bandwidth. For the harmonic detection problem, we desire to minimize this accumulated noise signal, and we accomplish this with small-bandwidth windows. A convenient measure of this bandwidth is the equivalent noise bandwidth (ENBW) of the window. This is the width of a rectangle filter with the same peak power gain that would accumulate the same noise power (see Fig. 7).

The accumulated noise power of the window is defined as

$$\text{Noise Power} = N_0 \int_{-\pi/T}^{+\pi/T} |W(\omega)|^2 d\omega / 2\pi \quad (8)$$

where N_0 is the noise power per unit bandwidth. Parseval's theorem allows (8) to be computed by

$$\text{Noise Power} = \frac{N_0}{T} \sum_n w^2(nT). \quad (9)$$

The peak power gain of the window occurs at $\omega = 0$, the zero frequency power gain, and is defined by

$$\text{Peak Signal Gain} \doteq W(0) = \sum_n w(nT) \quad (10a)$$

$$\text{Peak Power Gain} = W^2(0) = \left[\sum_n w(nT) \right]^2. \quad (10b)$$

Thus the ENBW (normalized by N_0/T , the noise power per bin) is given in the following equation and is tabulated for the windows of this report in Table I

$$\text{ENBW} = \frac{\sum_n w^2(nT)}{\left[\sum_n w(nT) \right]^2}. \quad (11)$$

B. Processing Gain

A concept closely allied to ENBW is processing gain (PG) and processing loss (PL) of a windowed transform. We can think of the DFT as a bank of matched filters, where each

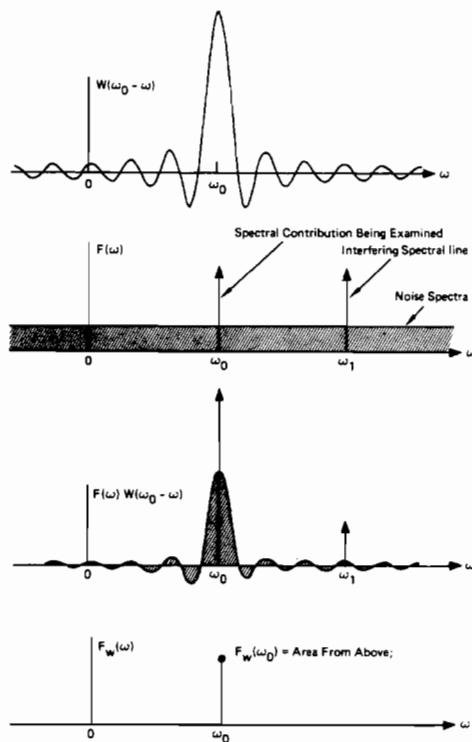


Fig. 6. Graphical interpretation of equation (6). Window visualized as a spectral filter.

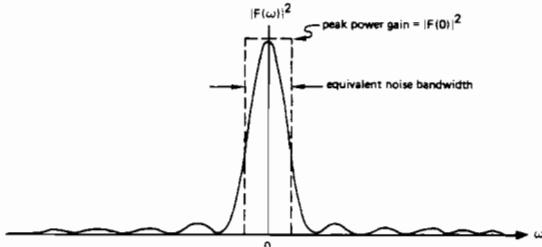


Fig. 7. Equivalent noise bandwidth of window.

filter is matched to one of the complex sinusoidal sequences of the basis set [3]. From this perspective, we can examine the PG (sometimes called the coherent gain) of the filter, and we can examine the PL due to the window having reduced the data to zero values near the boundaries. Let the input sampled sequence be defined by (12):

$$f(nT) = A \exp(+j\omega_k nT) + q(nT) \quad (12)$$

where $q(nT)$ is a white-noise sequence with variance σ_q^2 . Then the signal component of the windowed spectrum (the matched filter output) is presented in

$$\begin{aligned} F(\omega_k)|_{\text{signal}} &= \sum_n w(nT) A \exp(+j\omega_k nT) \exp(-j\omega_k nT) \\ &= A \sum_n w(nT). \end{aligned} \quad (13)$$

We see that the noiseless measurement (the expected value of the noisy measurement) is proportional to the input amplitude A . The proportionality factor is the sum of the window terms, which is in fact the dc signal gain of the window. For a rectangle window this factor is N , the number of terms in the window. For any other window, the gain is reduced due to the window smoothly going to zero near the boundaries. This

TABLE I
WINDOWS AND FIGURES OF MERIT

WINDOW	HIGHEST SIDE- LOBE LEVEL (dB)	SIDE- LOBE FALL- OFF (dB/OCT)	COHERENT GAIN	EQUIV. NOISE BW (BINS)	3.0-dB BW (BINS)	SCALLOP LOSS (dB)	WORST CASE PROCESS LOSS (dB)	6.0-dB BW (BINS)	OVERLAP CORRELATION (PCNT)		
	75% OL	50% OL									
RECTANGLE	-13	-6	1.00	1.00	0.89	3.92	3.92	1.21	75.0	50.0	
TRIANGLE	-27	-12	0.50	1.33	1.28	1.82	3.07	1.78	71.9	25.0	
COS ² (X) HANNING	$\alpha = 1.0$ $\alpha = 2.0$ $\alpha = 3.0$ $\alpha = 4.0$	-23 -32 -39 -47	-12 -18 -24 -30	0.64 0.50 0.42 0.38	1.23 1.50 1.73 1.94	1.20 1.44 1.66 1.86	2.10 1.42 1.08 0.86	3.01 3.18 3.47 3.75	1.65 2.00 2.32 2.59	75.5 65.9 56.7 48.6	31.8 16.7 8.5 4.3
HAMMING	-43	-6	0.54	1.36	1.30	1.78	3.10	1.81	70.7	23.5	
RIESZ	-21	-12	0.67	1.20	1.16	2.22	3.01	1.59	76.5	34.4	
RIEMANN	-26	-12	0.59	1.30	1.26	1.89	3.03	1.74	73.4	27.4	
DE LA VALLE- POUSSIN	-53	-24	0.38	1.92	1.82	0.90	3.72	2.55	49.3	5.0	
TUKEY	$\alpha = 0.25$ $\alpha = 0.50$ $\alpha = 0.75$	-14 -15 -19	-18 -18 -18	0.88 0.75 0.63	1.10 1.22 1.36	1.01 1.15 1.31	2.96 2.24 1.73	3.39 3.11 3.07	1.38 1.57 1.80	74.1 72.7 70.5	44.4 36.4 25.1
BOHMAN	-46	-24	0.41	1.79	1.71	1.02	3.54	2.38	54.5	7.4	
POISSON	$\alpha = 2.0$ $\alpha = 3.0$ $\alpha = 4.0$	-19 -24 -31	-6 -6 -6	0.44 0.32 0.25	1.30 1.65 2.08	1.21 1.45 1.75	2.09 1.46 1.03	3.23 3.64 4.21	1.69 2.08 2.58	69.9 54.8 40.4	27.8 15.1 7.4
HANNING- POISSON	$\alpha = 0.5$ $\alpha = 1.0$ $\alpha = 2.0$	-35 -39 NONE	-18 -18 -18	0.43 0.38 0.29	1.61 1.73 2.02	1.54 1.64 1.87	1.26 1.11 0.87	3.33 3.50 3.94	2.14 2.30 2.65	61.3 56.0 44.6	12.6 9.2 4.7
CAUCHY	$\alpha = 3.0$ $\alpha = 4.0$ $\alpha = 5.0$	-31 -35 -30	-6 -6 -6	0.42 0.33 0.28	1.48 1.76 2.06	1.34 1.50 1.68	1.71 1.36 1.13	3.40 3.83 4.28	1.90 2.20 2.53	61.6 48.8 38.3	20.2 13.2 9.0
GAUSSIAN	$\alpha = 2.5$ $\alpha = 3.0$ $\alpha = 3.5$	-42 -55 -69	-6 -6 -6	0.51 0.43 0.37	1.39 1.64 1.90	1.33 1.55 1.79	1.69 1.25 0.94	3.14 3.40 3.73	1.86 2.18 2.52	67.7 57.5 47.2	20.0 10.6 4.9
DOLPH- CHEBYSHEV	$\alpha = 2.5$ $\alpha = 3.0$ $\alpha = 3.5$ $\alpha = 4.0$	-50 -60 -70 -80	0 0 0 0	0.53 0.48 0.45 0.42	1.39 1.51 1.62 1.73	1.33 1.44 1.55 1.65	1.70 1.44 1.25 1.10	3.12 3.23 3.35 3.48	1.85 2.01 2.17 2.31	69.6 64.7 60.2 55.9	22.3 16.3 11.9 8.7
KAISER- BESSEL	$\alpha = 2.0$ $\alpha = 2.5$ $\alpha = 3.0$ $\alpha = 3.5$	-46 -57 -69 -82	-6 -6 -6 -6	0.49 0.44 0.40 0.37	1.50 1.65 1.80 1.93	1.43 1.57 1.71 1.83	1.46 1.20 1.02 0.89	3.20 3.38 3.56 3.74	1.99 2.20 2.39 2.57	65.7 59.5 53.9 48.8	16.9 11.2 7.4 4.8
BARCILON- TEMES	$\alpha = 3.0$ $\alpha = 3.5$ $\alpha = 4.0$	-53 -58 -68	-6 -6 -6	0.47 0.43 0.41	1.56 1.67 1.77	1.49 1.59 1.69	1.34 1.18 1.05	3.27 3.40 3.52	2.07 2.23 2.36	63.0 58.6 54.4	14.2 10.4 7.6
EXACT BLACKMAN	-51	-6	0.46	1.57	1.52	1.33	3.29	2.13	62.7	14.0	
BLACKMAN	-58	-18	0.42	1.73	1.68	1.10	3.47	2.35	56.7	9.0	
MINIMUM 3-SAMPLE BLACKMAN-HARRIS	-67	-6	0.42	1.71	1.66	1.13	3.45	1.81	57.2	9.6	
* MINIMUM 4-SAMPLE BLACKMAN-HARRIS	-92	-6	0.36	2.00	1.90	0.83	3.85	2.72	46.0	3.8	
* 61 dB 3-SAMPLE BLACKMAN-HARRIS	-61	-6	0.45	1.61	1.56	1.27	3.34	2.19	61.0	12.6	
74 dB 4-SAMPLE BLACKMAN-HARRIS	-74	-6	0.40	1.79	1.74	1.03	3.56	2.44	53.9	7.4	
4-SAMPLE $\alpha = 3.0$ KAISER-BESSEL	-69	-6	0.40	1.80	1.74	1.02	3.56	2.44	53.9	7.4	

*REFERENCE POINTS FOR DATA ON FIGURE 12 – NO FIGURES TO MATCH THESE WINDOWS.

reduction in proportionality factor is important as it represents a known bias on spectral amplitudes. Coherent power gain, the square of coherent gain, is occasionally the parameter listed in the literature. Coherent gain (the summation of (13)) normalized by its maximum value N is listed in Table I.

The incoherent component of the windowed transform is given by

$$F(\omega_k)|_{\text{noise}} = \sum_n w(nT) q(nT) \exp(-j\omega_k nT) \quad (14a)$$

and the incoherent power (the mean-square value of this component where $E\{\cdot\}$ is the expectation operator) is given by

$$\begin{aligned} E\{|F(\omega_k)|_{\text{noise}}|^2\} &= \sum_n \sum_m w(nT) w(mT) E\{q(nT) q^*(mT)\} \\ &\quad \cdot \exp(-j\omega_k nT) \exp(+j\omega_k mT) \\ &= \sigma_q^2 \sum_n w^2(nT). \end{aligned} \quad (14b)$$

Notice the incoherent power gain is the sum of the squares of the window terms, and the coherent power gain is the square of the sum of the window terms.

Finally, PG, which is defined as the ratio of output signal-to-noise ratio to input signal-to-noise ratio, is given by

$$\begin{aligned} \text{PG} &= \frac{S_o/N_o}{S_i/N_i} = \frac{A^2 \left[\sum_n w(nT) \right]^2 / \sigma_q^2 \sum_n w^2(nT)}{A^2 / \sigma_q^2} \\ &\quad \cdot \frac{\left[\sum_n w(nT) \right]^2}{\sum_n w^2(nT)}. \end{aligned} \quad (15)$$

Notice PG is the reciprocal of the normalized ENBW. Thus large ENBW suggests a reduced processing gain. This is reasonable, since an increased noise bandwidth permits additional noise to contribute to a spectral estimate.

C. Overlap Correlation

When the fast Fourier transform (FFT) is used to process long-time sequences a partition length N is first selected to establish the required spectral resolution of the analysis. Spectral resolution of the FFT is defined in (16) where Δf is the resolution, f_s is the sample frequency selected to satisfy the Nyquist criterion, and β is the coefficient reflecting the bandwidth increase due to the particular window selected. Note that $[f_s/N]$ is the minimum resolution of the FFT which we denote as the FFT bin width. The coefficient β is usually selected to be the ENBW in bins as listed in Table I

$$\Delta f = \beta \left(\frac{f_s}{N} \right). \quad (16)$$

If the window and the FFT are applied to nonoverlapping partitions of the sequence, as shown in Fig. 8, a significant part of the series is ignored due to the window's exhibiting small values near the boundaries. For instance, if the transform is being used to detect short-duration tone-like signals, the non overlapped analysis could miss the event if it occurred near the boundaries. To avoid this loss of data, the transforms are usually applied to the overlapped partition sequences as shown in Fig. 8. The overlap is almost always 50 or 75 percent. This overlap processing of course increases the work load to cover the total sequence length, but the rewards warrant the extra effort.

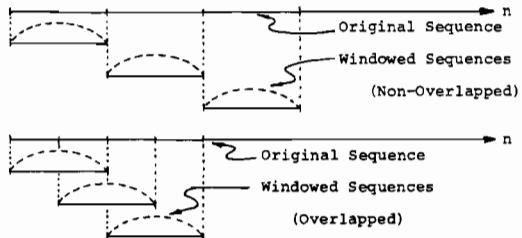


Fig. 8. Partition of sequences for nonoverlapped and for overlapped processing.

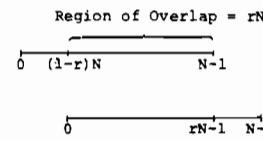


Fig. 9. Relationship between indices on overlapped intervals.

An important question related to overlapped processing is what is the degree of correlation of the random components in successive transforms? This correlation, as a function of fractional overlap r , is defined for a relatively flat noise spectrum over the window bandwidth by (17). Fig. 9 identifies how the indices of (17) relate to the overlap of the intervals. The correlation coefficient

$$c(r) = \frac{\left\{ \sum_{n=0}^{rN-1} (W(n) W(n + [1 - r]N)) \right\}}{\left\{ \sum_{n=0}^{N-1} W^2(n) \right\}} \quad (17)$$

is computed and tabulated in Table I. for each of the windows listed for 50- and 75-percent overlap.

Often in a spectral analysis, the squared magnitude of successive transforms are averaged to reduce the variance of the measurements [5]. We know of course that when we average K identically distributed independent measurements, the variance of the average is related to the individual variance of the measurements by

$$\frac{\sigma_{\text{Avg.}}^2}{\sigma_{\text{Meas.}}^2} = \frac{1}{K}. \quad (18)$$

Now we can ask what is the reduction in the variance when we average measurements which are correlated as they are for overlapped transforms? Welch [5] has supplied an answer to this question which we present here, for the special case of 50- and 75-percent overlap

$$\frac{\sigma_{\text{Avg.}}^2}{\sigma_{\text{Meas.}}^2} = \frac{1}{K} [1 + 2c^2(0.5)] - \frac{2}{K^2} [c^2(0.5)],$$

50 percent overlap

$$\begin{aligned} &= \frac{1}{K} [1 + 2c^2(0.75) + 2c^2(0.5) + 2c^2(0.25)] \\ &\quad - \frac{2}{K^2} [c^2(0.75) + 2c^2(0.5) + 3c^2(0.25)], \end{aligned}$$

75 percent overlap. (19)

The negative terms in (19) are the edge effects of the average and can be ignored if the number of terms K is larger than ten. For good windows, $c^2(0.25)$ is small compared to 1.0,

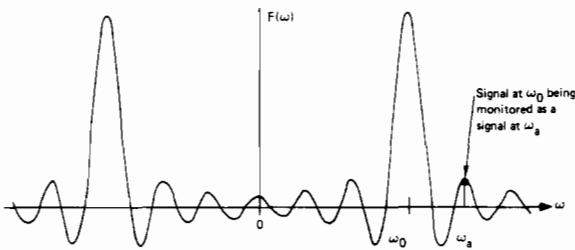


Fig. 10. Spectral leakage effect of window.

and can also be omitted from (19) with negligible error. For this reason, $c(0.25)$ was not listed in Table I. Note, that for good windows (see last paragraph of Section IV-F), transforms taken with 50-percent overlap are essentially independent.

D. Scalloping Loss

An important consideration related to minimum detectable signal is called scalloping loss or picket-fence effect. We have considered the windowed DFT as a bank of matched filters and have examined the processing gain and the reduction of this gain ascribable to the window for tones matched to the basis vectors. The basis vectors are tones with frequencies equal to multiples of f_s/N (with f_s being the sample frequency). These frequencies are sample points from the spectrum, and are normally referred to as DFT output points or as DFT bins. We now address the question, what is the additional loss in processing gain for a tone of frequency midway between two bin frequencies (that is, at frequencies $(k+1/2)f_s/N$)?

Returning to (13), with ω_k replaced by $\omega_{(k+1/2)}$, we determine the processing gain for this half-bin frequency shift as defined in

$$F(\omega_{(1/2)})|_{\text{signal}} = A \sum_n w(nT) \exp(-j\omega_{(1/2)}nT),$$

where $\omega_{(1/2)} = \frac{1}{2} \frac{\omega_s}{N} = \frac{\pi}{NT}$. (20a)

We also define the scalloping loss as the ratio of coherent gain for a tone located half a bin from a DFT sample point to the coherent gain for a tone located at a DFT sample point, as indicated in

$$\text{Scalloping Loss} = \frac{\left| \sum_n w(nT) \exp\left(-j\frac{\pi}{N}n\right) \right|}{\sum_n w(nT)} = \frac{\left| W\left(\frac{1}{2} \frac{\omega_s}{N}\right) \right|}{W(0)}. \quad (20b)$$

Scalloping loss represents the maximum reduction in PG due to signal frequency. This loss has been computed for the windows of this report and has been included in Table I.

E. Worst Case Processing Loss

We now make an interesting observation. We define worst case PL as the sum of maximum scalloping loss of a window and of PL due to that window (both in decibel). This number is the reduction of output signal-to-noise ratio as a result of windowing and of worst case frequency location. This of course is related to the minimum detectable tone in broad-band noise. It is interesting to note that the worst case loss is always between 3.0 and 4.3 dB. Windows with worst case PL exceeding 3.8 dB are very poor windows and should not

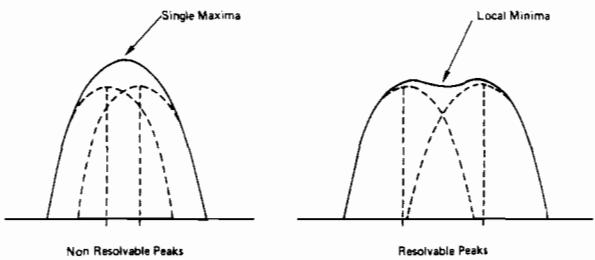


Fig. 11. Spectral resolution of nearby kernels.

be used. Additional comments on poor windows will be found in Section IV-G. We can conclude from the combined loss figures of Table I and from Fig. 12 that for the detection of single tones in broad-band noise, nearly any window (other than the rectangle) is as good as any other. The difference between the various windows is less than 1.0 dB and for good windows is less than 0.7 dB. The detection of tones in the presence of other tones is, however, quite another problem. Here the window does have a marked affect, as will be demonstrated shortly.

F. Spectral Leakage Revisited

Returning to (6) and to Fig. 6, we observe the spectral measurement is affected not only by the broadband noise spectrum, but also by the narrow-band spectrum which falls within the bandwidth of the window. In fact, a given spectral component say at $\omega = \omega_0$ will contribute output (or will be observed) at another frequency, say at $\omega = \omega_a$ according to the gain of the window centered at ω_0 and measured at ω_a . This is the effect normally referred to as spectral leakage and is demonstrated in Fig. 10 with the transform of a finite duration tone of frequency ω_0 .

This leakage causes a bias in the amplitude and the position of a harmonic estimate. Even for the case of a single real harmonic line (not at a DFT sample point), the leakage from the kernel on the negative-frequency axis biases the kernel on the positive-frequency line. This bias is most severe and most bothersome for the detection of small signals in the presence of nearby large signals. To reduce the effects of this bias, the window should exhibit low-amplitude sidelobes far from the central main lobe, and the transition to the low sidelobes should be very rapid. One indicator of how well a window suppresses leakage is the peak sidelobe level (relative to the main lobe); another is the asymptotic rate of falloff of these sidelobes. These indicators are listed in Table I.

G. Minimum Resolution Bandwidth

Fig. 11 suggests another criterion with which we should be concerned in the window selection process. Since the window imposes an effective bandwidth on the spectral line, we would be interested in the minimum separation between two equal-strength lines such that for arbitrary spectral locations their respective main lobes can be resolved. The classic criterion for this resolution is the width of the window at the half-power points (the 3.0-dB bandwidth). This criterion reflects the fact that two equal-strength main lobes separated in frequency by less than their 3.0-dB bandwidths will exhibit a single spectral peak and will not be resolved as two distinct lines. The problem with this criterion is that it does not work for the coherent addition we find in the DFT. The DFT output points are the coherent addition of the spectral components weighted through the window at a given frequency.

If two kernels are contributing to the coherent summation, the sum at the crossover point (nominally half-way between them) must be smaller than the individual peaks if the two peaks are to be resolved. Thus at the crossover points of the kernels, the gain from each kernel must be less than 0.5, or the crossover points must occur beyond the 6.0-dB points of the windows. Table I lists the 6.0-dB bandwidths of the various windows examined in this report. From the table, we see that the 6.0-dB bandwidth varies from 1.2 bins to 2.6 bins, where a bin is the fundamental frequency resolution ω_s/N . The 3.0-dB bandwidth does have utility as a performance indicator as shown in the next paragraph. Remember however, it is the 6.0-dB bandwidth which defines the resolution of the windowed DFT.

From Table I, we see that the noise bandwidth always exceeds the 3.0-dB bandwidth. The difference between the two, referenced to the 3.0-dB bandwidth, appears to be a sensitive indicator of overall window performance. We have observed that for all the good windows on the table, this indicator was found to be in the range of 4.0 to 5.5 percent. Those windows for which this ratio is outside that range either have a wide main lobe or a high sidelobe structure and, hence, are characterized by high processing loss or by poor two-tone detection capabilities. Those windows for which this ratio is inside the 4.0 to 5.5-percent range are found in the lower left corner of the performance comparison chart (Fig. 12), which is described next.

While Table I does list the common performance parameters of the windows examined in this report, the mass of numbers is not enlightening. We do realize that the sidelobe level (to reduce bias) and the worst case processing loss (to maximize detectability) are probably the most important parameters on the table. Fig. 12 shows the relative position of the windows as a function of these parameters. Windows residing in the lower left corner of the figure are the good-performing windows. They exhibit low-sidelobe levels and low worst case processing loss. We urge the reader to read Sections VI and VII; Fig. 12 presents a lot of information, but not the full story.

V. CLASSIC WINDOWS

We will now catalog some well-known (and some not well-known) windows. For each window we will comment on the justification for its use and identify its significant parameters. All the windows will be presented as even (about the origin) sequences with an odd number of points. To convert the window to DFT-even, the right end point will be discarded and the sequence will be shifted so that the left end point coincides with the origin. We will use normalized coordinates with sample period $T = 1.0$, so that ω is periodic in 2π and, hence, will be identified as θ . A DFT bin will be considered to extend between DFT sample points (multiples of $2\pi/N$) and have a width of $2\pi/N$.

A. Rectangle (Dirichlet) Window [6]

The rectangle window is unity over the observation interval, and can be thought of as a gating sequence applied to the data so that they are of finite extent. The window for a finite Fourier transform is defined as

$$w(n) = 1.0, \quad n = -\frac{N}{2}, \dots, -1, 0, 1, \dots, \frac{N}{2} \quad (21a)$$

and is shown in Fig. 13. The same window for a DFT is

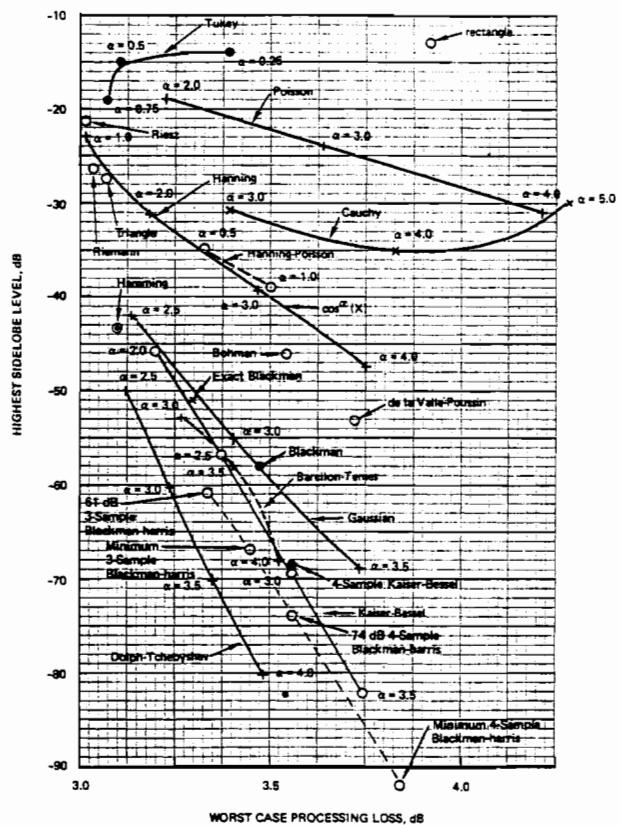


Fig. 12. Comparison of windows: sidelobe levels and worst case processing loss.

defined as

$$w(n) = 1.0, \quad n = 0, 1, \dots, N-1. \quad (21b)$$

The spectral window for the DFT window sequence is given in

$$W(\theta) = \exp \left(-j \frac{N-1}{2} \theta \right) \frac{\sin \left[\frac{N}{2} \theta \right]}{\sin \left[\frac{1}{2} \theta \right]}. \quad (21c)$$

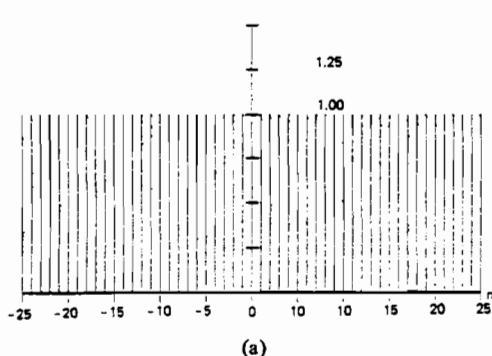
The transform of this window is seen to be the Dirichlet kernel, which exhibits a DFT main-lobe width (between zero crossings) of 2 bins and a first sidelobe level approximately 13 dB down from the main-lobe peak. The sidelobes fall off at 6.0 dB per octave, which is of course the expected rate for a function with a discontinuity. The parameters of the DFT window are listed in Table I.

With the rectangle window now defined, we can answer the question posed earlier: in what sense does the finite sum of (22a) approximate the infinite sum of (22b)?

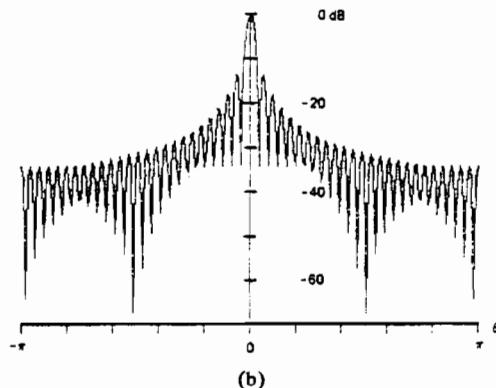
$$F(\theta) = \sum_{n=-N/2}^{+N/2} f(n) \exp(-jn\theta) \quad (22a)$$

$$F(\theta) = \sum_{n=-\infty}^{+\infty} f(n) \exp(-jn\theta). \quad (22b)$$

We observe the finite sum is the rectangle-windowed version of the infinite sum. We recognize that the infinite sum is the Fourier series expansion of some periodic function for which the $f(n)$'s are the Fourier series coefficients. We also recognize

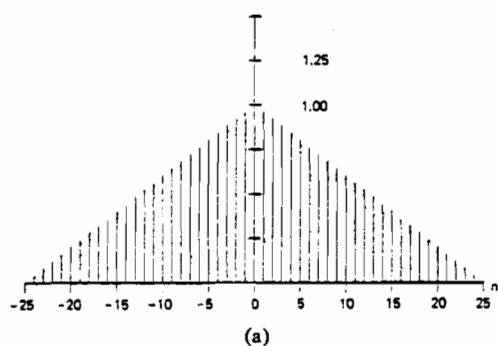


(a)

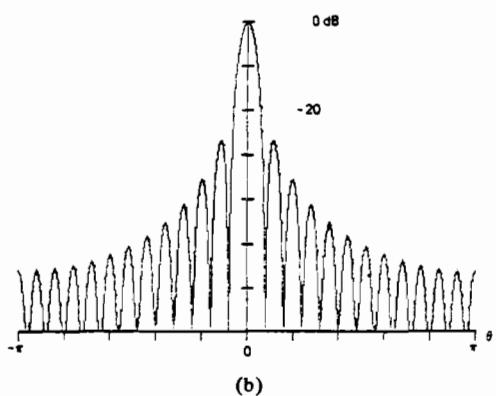


(b)

Fig. 13. (a) Rectangle window. (b) Log-magnitude of transform.



(a)



(b)

Fig. 14. (a) Triangle window. (b) Log-magnitude of transform.

that the finite sum is simply the partial sum of the series. From this viewpoint we can cast the question in terms of the convergence properties of the partial sums of Fourier series. From this work we know the partial sum is the least mean-square error approximation to the infinite sum.

We observe that mean square convergence is a convenient analytic concept, but it is not attractive for finite estimates or for numerical approximations. Mean-square estimates tend to oscillate about their means, and do not exhibit uniform convergence. (The approximation in a neighborhood of a point of continuity may get worse if more terms are added to the partial sum.) We normally observe this behavior near points of discontinuity as the ringing we call Gibbs phenomenon. It is this oscillatory behavior we are trying to control by the use of other windows.

B. Triangle (Fejer, Bartlet) Window [7]

The triangle window for a finite Fourier transform is defined as

$$W(n) = 1.0 - \frac{|n|}{N/2}, \quad n = -\frac{N}{2}, \dots, -1, 0, 1, \dots, \frac{N}{2} \quad (23a)$$

and is shown in Fig. 14. The same window for a DFT is defined as

$$W(n) = \begin{cases} \frac{n}{N/2}, & n = 0, 1, \dots, \frac{N}{2} \\ W(N-n), & n = \frac{N}{2}, \dots, N-1 \end{cases} \quad (23b)$$

and the spectral window corresponding to the DFT sequence is given in

$$W(\theta) = \frac{2}{N} \exp \left[-j \left(\frac{N}{2} - 1 \right) \theta \right] \left[\frac{\sin \left(\frac{N}{4} \theta \right)}{\sin \left(\frac{1}{2} \theta \right)} \right]^2. \quad (23c)$$

The transform of this window is seen to be the squared Dirichlet kernel. Its main-lobe width (between zero crossings) is twice that of the rectangle's and the first sidelobe level is approximately 26 dB down from the main-lobe peak, again, twice that of the rectangle's. The sidelobes fall off at -12 dB per octave, reflecting the discontinuity of the window residing in the first derivative (rather than in the function itself). The triangle is the simplest window which exhibits a nonnegative transform. This property can be realized by convolving any window (of half-extent) with itself. The resultant window's transform is the square of the original window's transform.

A window sequence derived by self-convolving a parent window contains approximately twice the number of samples as the parent window, hence corresponds to a trigonometric polynomial (its Z-transform) of approximately twice the order. (Convoluting two rectangles each of $N/2$ points will result in a triangle of $N + 1$ points when the zero end points are counted.) The transform of the window will now exhibit twice as many zeros as the parent transform (to account for the increased order of the associated trigonometric polynomial). But how has the transform applied these extra zeros available from the increased order polynomial? The self-

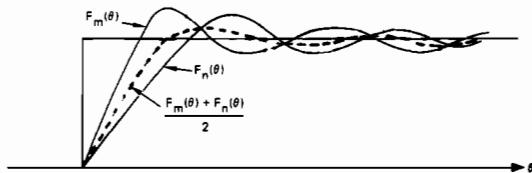


Fig. 15. Two partial sums and their average.

TABLE II
FEJER CONVERGENCE FACTORS AS AN AVERAGE TRANSFORM

$F_0(\theta)$	f_0								
$F_1(\theta)$	$f_{-1} \quad f_0 \quad f_{+1}$								
$F_2(\theta)$	$f_{-2} \quad f_{-1} \quad f_0 \quad f_{+1} \quad f_{+2}$								
$F_3(\theta)$	$f_{-3} \quad f_{-2} \quad f_{-1} \quad f_0 \quad f_{+1} \quad f_{+2} \quad f_{+3}$								
$F^4(\theta)$	$\frac{0}{4}f_{-4}$	$\frac{1}{4}f_{-3}$	$\frac{2}{4}f_{-2}$	$\frac{3}{4}f_{-1}$	$\frac{4}{4}f_0$	$\frac{3}{4}f_{+1}$	$\frac{2}{4}f_{+2}$	$\frac{1}{4}f_{+3}$	$\frac{0}{4}f_{+4}$

convolved window simply places repeated zeros at each location for which the parent transform had a zero. This, of course, not only sets the transform to zero at those points, but also sets the first derivative to zero at those points. If the intent of the increased order of polynomial is to hold down the sidelobe levels, then doubling up on the zeros is a wasteful tactic. The additional zeros might better be placed between the existing zeros (near the local peaks of the sidelobes) to hold down the sidelobes rather than at locations for which the transform is already equal to zero. In fact we will observe in subsequent windows that very few good windows exhibit repeated roots.

Backing up for a moment, it is interesting to examine the triangle window in terms of partial-sum convergence of Fourier series. Fejer observed that the partial sums of Fourier series were poor numerical approximations [8]. Fourier coefficients were easy to generate however, and he questioned if some simple modification of coefficients might lead to a new set with more desirable convergence properties. The oscillation of the partial sum, and the contraction of those oscillations as the order of the partial sum increased, suggested that an average of the partial sums would be a smoother function. Fig. 15 presents an expansion of two partial sums near a discontinuity. Notice the average of the two expansions is smoother than either. Continuing in this line of reasoning, an average expansion $F^N(\theta)$ might be defined by

$$F^N(\theta) = \frac{1}{N} [F_{N-1}(\theta) + F_{N-2}(\theta) + \cdots + F_0(\theta)] \quad (24)$$

where $F_M(\theta)$ is the M -term partial sum of the series. This is easily visualized in Table II, which lists the nonzero coefficients of the first four partial sums and their average summation. We see that the Fejer convergence factors applied to the Fourier series coefficients is, in fact, a triangle window. The averaging of partial sums is known as the method of Cesàro summability.

C. $\text{Cos}^\alpha(X)$ Windows

This is actually a family of windows dependent upon the parameter α , with α normally being an integer. Attractions of this family include the ease with which the terms can be generated, and the easily identified properties of the transform

of the cosine function. These properties are particularly attractive under the DFT. The window for a finite Fourier transform is defined as

$$w(n) = \cos^\alpha \left[\frac{n}{N} \pi \right], \quad n = -\frac{N}{2}, \dots, -1, 0, 1, \dots, \frac{N}{2} \quad (25a)$$

and for a DFT as

$$w(n) = \sin^\alpha \left[\frac{n}{N} \pi \right], \quad n = 0, 1, 2, \dots, N-1. \quad (25b)$$

Notice the effect due to the change of the origin. The most common values of α are the integers 1 through 4, with 2 being the most well known (as the Hanning window). This window is identified for values of α equal to 1 and 2 in (26a), (26b), (27a), and (27b), (the "a" for the finite transforms, the "b" for the DFT):

$$\alpha = 1.0 \text{ (cosine lobe)}$$

$$w(n) = \cos \left[\frac{n}{N} \pi \right], \quad n = -\frac{N}{2}, \dots, -1, 0, 1, \dots, \frac{N}{2} \quad (26a)$$

$$\alpha = 1.0 \text{ (sine lobe)}$$

$$w(n) = \sin \left[\frac{n}{N} \pi \right], \quad n = 0, 1, 2, \dots, N-1 \quad (26b)$$

$$\alpha = 2.0 \text{ (cosine squared, raised cosine, Hanning)}$$

$$\begin{aligned} w(n) &= \cos^2 \left[\frac{n}{N} \pi \right] \\ &= 0.5 \left[1.0 + \cos \left[\frac{2n}{N} \pi \right] \right], \\ &\quad n = -\frac{N}{2}, \dots, -1, 0, 1, \dots, \frac{N}{2} \end{aligned} \quad (27a)$$

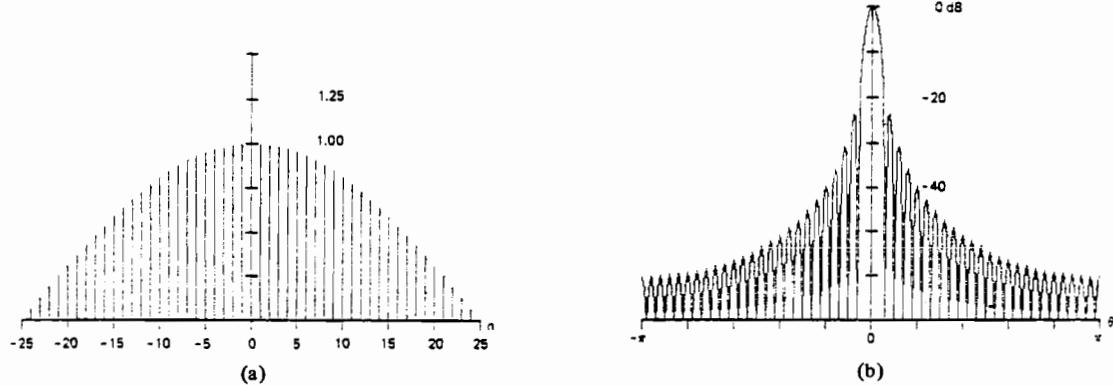
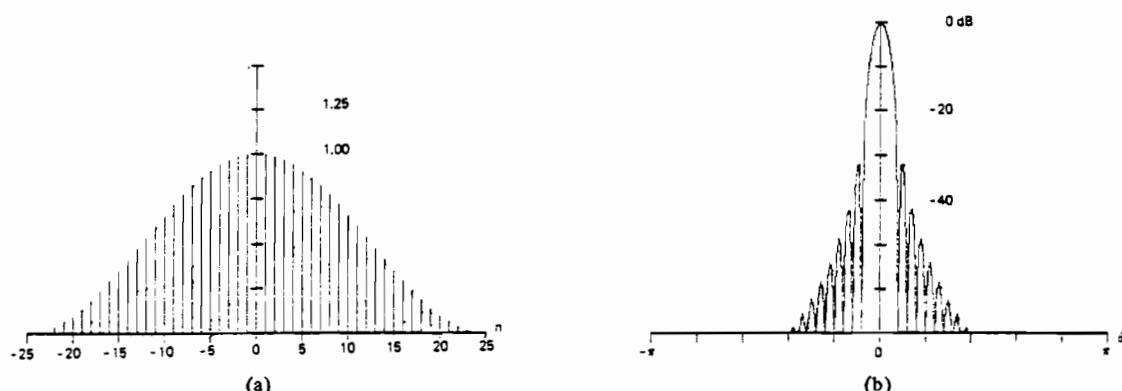
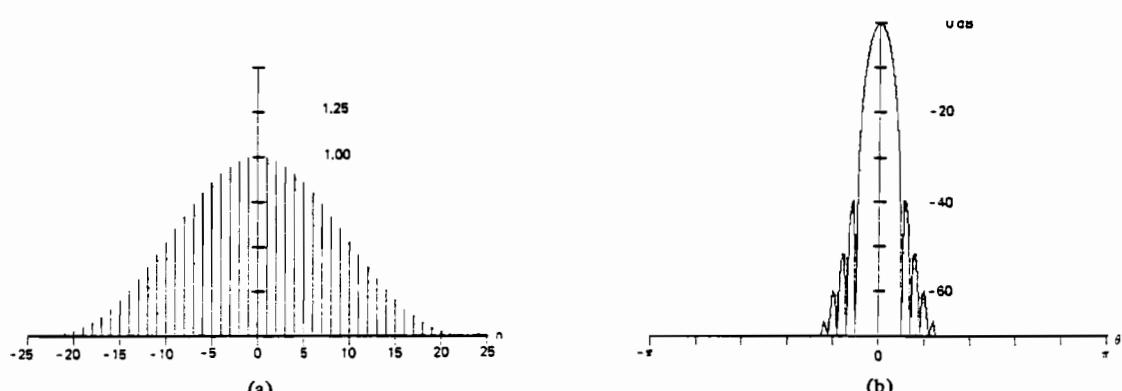
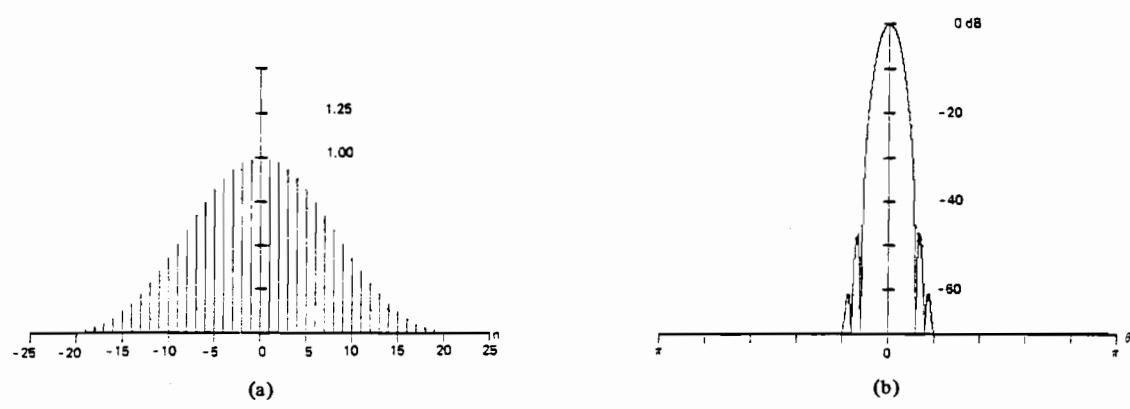
$$\alpha = 2.0 \text{ (sine squared, raised cosine, Hanning)}$$

$$\begin{aligned} w(n) &= \sin^2 \left[\frac{n}{N} \pi \right] \\ &= 0.5 \left[1.0 - \cos \left[\frac{2n}{N} \pi \right] \right], \quad n = 0, 1, 2, \dots, N-1. \end{aligned} \quad (27b)$$

The windows are shown for α integer values of 1 through 4 in Figs. 16 through 19. Notice as α becomes larger, the windows become smoother and the transform reflects this increased smoothness in decreased sidelobe level and faster falloff of the sidelobes, but with an increased width of the main lobe.

Of particular interest in this family, is the Hann window (after the Austrian meteorologist, Julius Von Hann)¹ [7]. Not only is this window continuous, but so is its first derivative. Since the discontinuity of this window resides in the second derivative, the transform falls off at $1/\omega^3$ or at -18 dB per octave. Let us closely examine the transform of this window. We will gain some interesting insight and learn of a clever application of the window under the DFT.

¹The correct name of this window is "Hann." The term "Hanning" is used in this report to reflect conventional usage. The derived term "Hann'd" is also widely used.

Fig. 16. (a) $\cos(n\pi/N)$ window. (b) Log-magnitude of transform.Fig. 17. (a) $\cos^2(n\pi/N)$ window. (b) Log-magnitude of transform.Fig. 18. (a) $\cos^3(n\pi/N)$ window. (b) Log-magnitude of transform.Fig. 19. (a) $\cos^4(n\pi/N)$ window. (b) Log-magnitude of transform.

The sampled Hanning window can be written as the sum of the sequences indicated in

$$w(n) = 0.5 + 0.5 \cos \left[\frac{2n}{N} \pi \right],$$

$$n = -\frac{N}{2}, \dots, -1, 0, 1, \dots, \frac{N}{2} - 1. \quad (28a)$$

Each sequence has the easily recognized DFT indicated in

$$W(\theta) = 0.5 D(\theta) + 0.25 \left[D \left(\theta - \frac{2\pi}{N} \right) + D \left(\theta + \frac{2\pi}{N} \right) \right] \quad (28b)$$

where

$$D(\theta) = \exp \left(+j \frac{\theta}{2} \right) \frac{\sin \left[\frac{N}{2} \theta \right]}{\sin \left[\frac{1}{2} \theta \right]}.$$

We recognize the Dirichlet kernel at the origin as the transform of the constant 0.5 samples and the pair of translated kernels as the transform of the single cycle of cosine samples. Note that the translated kernels are located on the first zeros of the center kernel, and are half the size of the center kernel. Also the sidelobes of the translated kernel are about half the size and are of opposite phase of the sidelobes of the central kernel. The summation of the three kernels' sidelobes being in phase opposition, tends to cancel the sidelobe structure. This cancelling summation is demonstrated in Fig. 20 which depicts the summation of the Dirichlet kernels (without the phase-shift terms).

The partial cancelling of the sidelobe structure suggests a constructive technique to define new windows. The most well-known of these are the Hamming and the Blackman windows which are presented in the next two sections.

For the special case of the DFT, the Hanning window is sampled at multiples of $2\pi/N$, which of course are the locations of the zeros of the central Dirichlet kernel. Thus only three nonzero samples are taken in the sampling process. The positions of these samples are at $-2\pi/N$, 0, and $+2\pi/N$. The value of the samples obtained from (28b) (including the phase factor $\exp(-j(N/2)\theta)$ to account for the $N/2$ shift) are $-\frac{1}{4}$, $+\frac{1}{2}$, $-\frac{1}{4}$, respectively. Note the minus signs. These results from the shift in the origin for the window. Without the shift, the phase term is missing and the coefficients are all positive $\frac{1}{4}$, $\frac{1}{2}$, $\frac{1}{4}$. These are incorrect for DFT processing, but they find their way into much of the literature and practice.

Rather than apply the window as a product in the time domain, we always have the option to apply it as a convolution in the frequency domain. The attraction of the Hanning window for this application is twofold; first, the window spectra is nonzero at only three data points, and second, the sample values are binary fractions, which can be implemented as right shifts. Thus the Hanning-windowed spectral points obtained from the rectangle-windowed spectral points are obtained as indicated in the following equation as two real adds and two binary shifts (to multiply by $\frac{1}{2}$):

$$F(k)|_{\text{Hanning}} = \frac{1}{2} [F(k) - \frac{1}{2} [F(k-1) + F(k+1)]]|_{\text{Rectangle}}. \quad (29)$$

Thus a Hanning window applied to a real transform of length N can be performed as N real multiplies on the time sequence

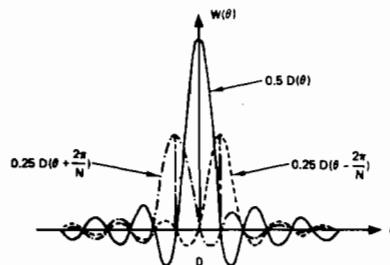


Fig. 20. Transform of Hanning window as a sum of three Dirichlet kernels.

or as $2N$ real adds and $2N$ binary shifts on the spectral data. One other mildly important consideration, if the window is to be applied to the time data, is that the samples of the window must be stored somewhere, which normally means additional memory or hardware. It so happens that the samples of the cosine for the Hanning window are already stored in the machine as the trig-table for the FFT; thus the Hanning window requires no additional storage.

D. Hamming Window [7]

The Hamming window can be thought of as a modified Hanning window. (Note the potential source of confusion in the similarities of the two names.) Referring back to Figs. 17 and 20, we note the inexact cancellation of the sidelobes from the summation of the three kernels. We can construct a window by adjusting the relative size of the kernels as indicated in the following to achieve a more desirable form of cancellation:

$$w(n) = \alpha + (1 - \alpha) \cos \left[\frac{2\pi}{N} n \right]$$

$$W(\theta) = \alpha D(\theta) + 0.5 (1 - \alpha) \left[D \left(\theta - \frac{2\pi}{N} \right) + D \left(\theta + \frac{2\pi}{N} \right) \right]. \quad (30a)$$

Perfect cancellation of the first sidelobe (at $\theta = 2.5[2\pi/N]$) occurs when $\alpha = 25/46$ ($\alpha \approx 0.543478261$). If α is selected as 0.54 (an approximation to 25/46), the new zero occurs at $\theta = 2.6[2\pi/N]$ and a marked improvement in sidelobe level is realized. For this value of α , the window is called the Hamming window and is identified by

$$w(n) = \begin{cases} 0.54 + 0.46 \cos \left[\frac{2\pi}{N} n \right], & n = -\frac{N}{2}, \dots, -1, 0, 1, \dots, \frac{N}{2} \\ 0.54 - 0.46 \cos \left[\frac{2\pi}{N} n \right], & n = 0, 1, 2, \dots, N-1. \end{cases} \quad (30b)$$

The coefficients of the Hamming window are nearly the set which achieve minimum sidelobe levels. If α is selected to be 0.53856 the sidelobe level is -43 dB and the resultant window is a special case of the Blackman-Harris windows presented in Section V-E. The Hamming window is shown in Fig. 21. Notice the deep attenuation at the missing sidelobe position. Note also that the small discontinuity at the boundary of the window has resulted in a $1/\omega$ (6.0 dB per octave) rate of falloff. The better sidelobe cancellation does result in a much lower initial sidelobe level of -42 dB. Table I lists the param-

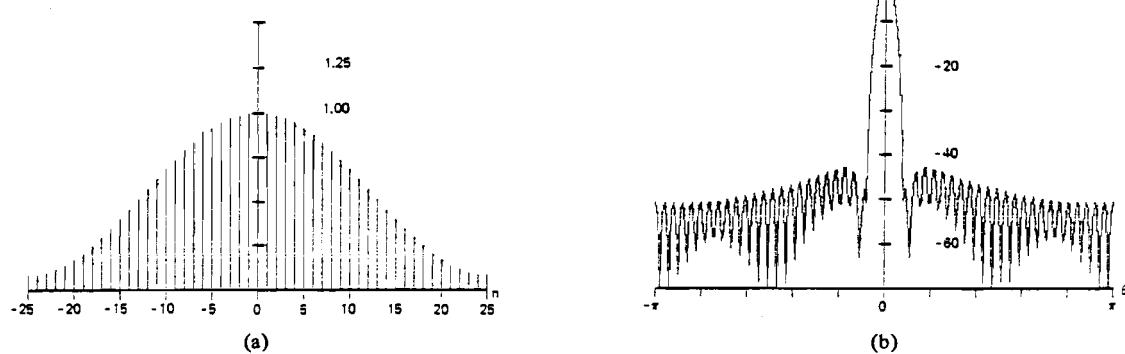


Fig. 21. (a) Hamming window. (b) Log-magnitude of Fourier transform.

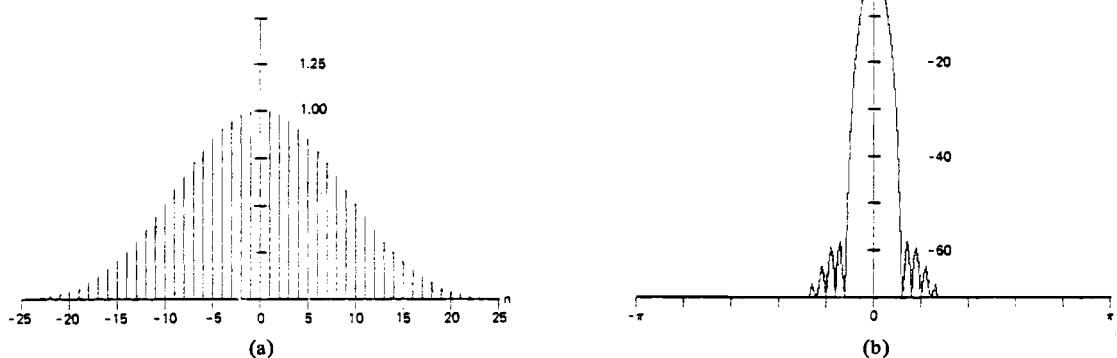


Fig. 22. (a) Blackman window. (b) Log-magnitude of transform.

eters of this window. Also note the loss of binary weighting; hence the need to perform multiplication to apply the weighting factors of the spectral convolution.

E. Blackman Window [7]

The Hamming and Hanning windows are examples of windows constructed as the summation of shifted Dirichlet kernels. This data window is defined for the finite Fourier transform in (31a) and for the DFT in (31b); equation (31c) is the resultant spectral window for the DFT given as a summation of the Dirichlet kernels $D(\theta)$ defined by $W(\theta)$ in (21c);

$$W(n) = \sum_{m=0}^{N/2} a_m \cos \left[\frac{2\pi}{N} mn \right], \quad n = -\frac{N}{2}, \dots, -1, 0, 1, \dots, \frac{N}{2} \quad (31a)$$

$$W(n) = \sum_{m=0}^{N/2} (-1)^m a_m \cos \left[\frac{2\pi}{N} mn \right], \quad n = 0, 1, \dots, N-1 \quad (31b)$$

$$W(\theta) = \sum_{m=0}^{N/2} (-1)^m \frac{a_m}{2} \left[D\left(\theta - \frac{2\pi}{N} m\right) + D\left(\theta + \frac{2\pi}{N} m\right) \right]. \quad (31c)$$

Subject to constraint

$$\sum_{m=0}^{N/2} a_m = 1.0.$$

We can see that the Hanning and the Hamming windows are

of this form with a_0 and a_1 being nonzero. We see that their spectral windows are summations of three-shifted kernels.

We can construct windows with any K nonzero coefficients and achieve a $(2K-1)$ summation of kernels. We recognize, however, that one way to achieve windows with a narrow main lobe is to restrict K to a small integer. Blackman examined this window for $K = 3$ and found the values of the nonzero coefficients which place zeros at $\theta = 3.5(2\pi/N)$ and at $\theta = 4.5(2\pi/N)$, the position of the third and the fourth sidelobes, respectively, of the central Dirichlet kernel. These exact values and their two place approximations are

$$a_0 = \frac{7938}{18608} \doteq 0.42659071 \simeq 0.42$$

$$a_1 = \frac{9240}{18608} \doteq 0.49656062 \simeq 0.50$$

$$a_2 = \frac{1430}{18608} \doteq 0.07684867 \simeq 0.08.$$

The window which uses these two place approximations is known as the Blackman window. When we describe this window with the "exact" coefficients we will refer to it as the exact Blackman window. The Blackman window is defined for the finite transform in the following equation and the window is shown in Fig. 22:

$$W(n) = 0.42 + 0.50 \cos \left[\frac{2\pi}{N} n \right] + 0.08 \cos \left[\frac{2\pi}{N} 2n \right], \\ n = -\frac{N}{2}, \dots, -1, 0, 1, \dots, \frac{N}{2}. \quad (32)$$

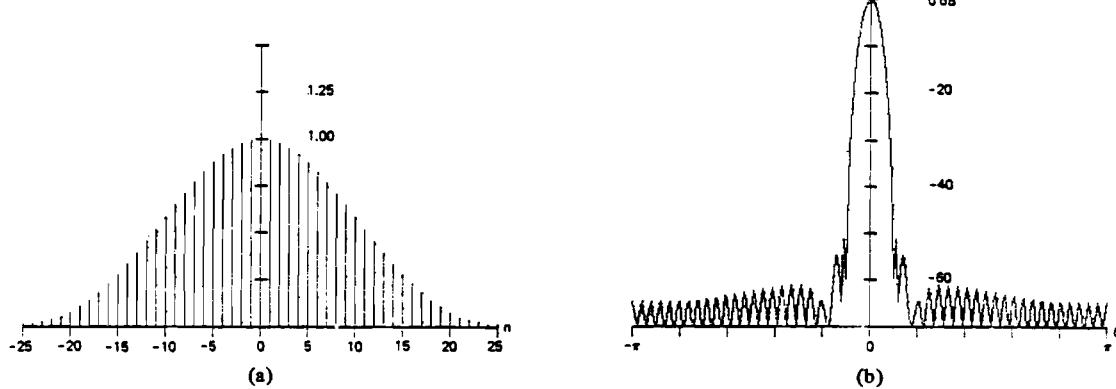


Fig. 23. (a) Exact Blackman window. (b) Log-magnitude of transform.

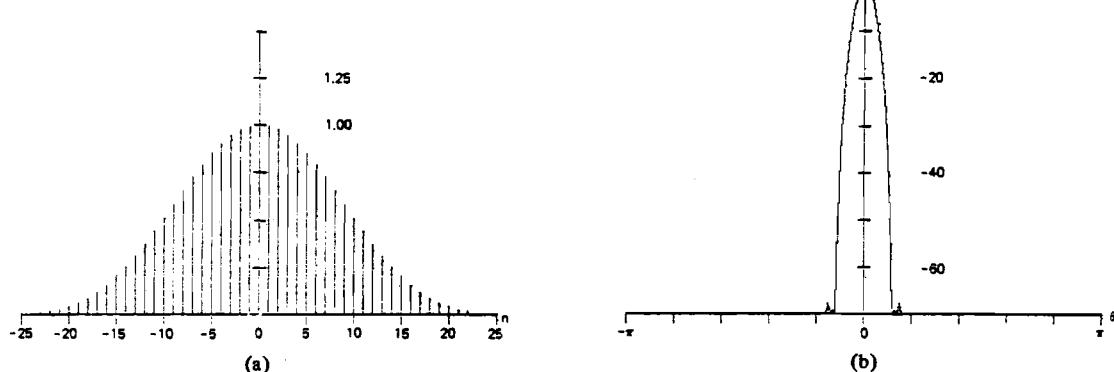


Fig. 24. (a) Minimum 3-term Blackman-Harris window. (b) Log-magnitude of transform.

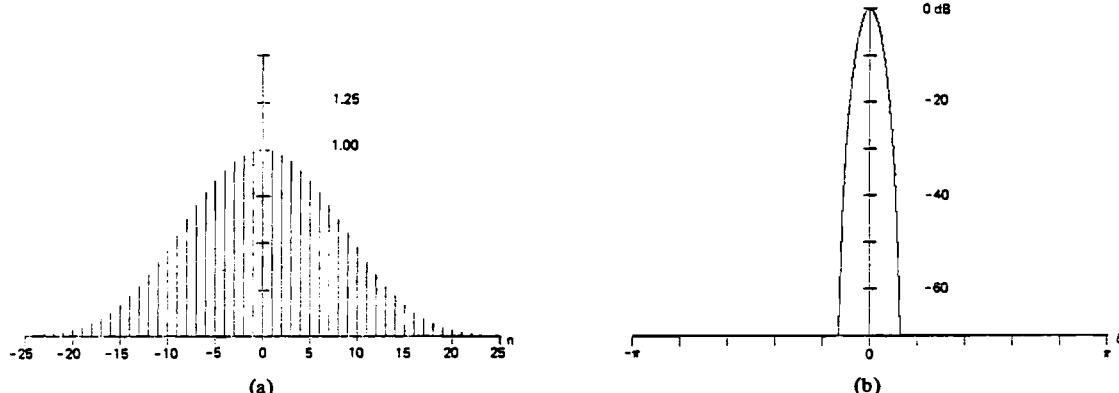


Fig. 25. (a) 4-term Blackman-Harris window. (b) Log-magnitude of transform.

The exact Blackman window is shown in Fig. 23. The sidelobe level is 51 dB down for the exact Blackman window and is 58 dB down for the Blackman window. As an observation, note that the coefficients of the Blackman window sum to zero ($0.42 - 0.50 + 0.08$) at the boundaries while the exact coefficients do not. Thus the Blackman window is continuous with a continuous first derivative at the boundary and falls off like $1/\omega^3$ or 18 dB per octave. The exact terms (like the Hamming window) have a discontinuity at the boundary and falls off like $1/\omega$ or 6 dB per octave. Table I lists the parameters of these two windows. Note that for this class of windows, the a_0 coefficient is the coherent gain of the window.

Using a gradient search technique [9], we have found the windows which for 3- and 4-nonzero terms achieve a minimum

sidelobe level. We have also constructed families of 3- and 4-term windows in which we trade main-lobe width for sidelobe level. We call this family the Blackman-Harris window. We have found that the minimum 3-term window can achieve a sidelobe level of -67 dB and that the minimum 4-term window can achieve a sidelobe level of -92 dB. These windows are defined for the DFT by

$$w(n) = a_0 - a_1 \cos\left(\frac{2\pi}{N} n\right) + a_2 \cos\left(\frac{2\pi}{N} 2n\right) - a_3 \cos\left(\frac{2\pi}{N} 3n\right), \\ n = 0, 1, 2, \dots, N-1. \quad (33)$$

The listed coefficients correspond to the minimum 3-term window which is presented in Fig. 24, another 3-term window

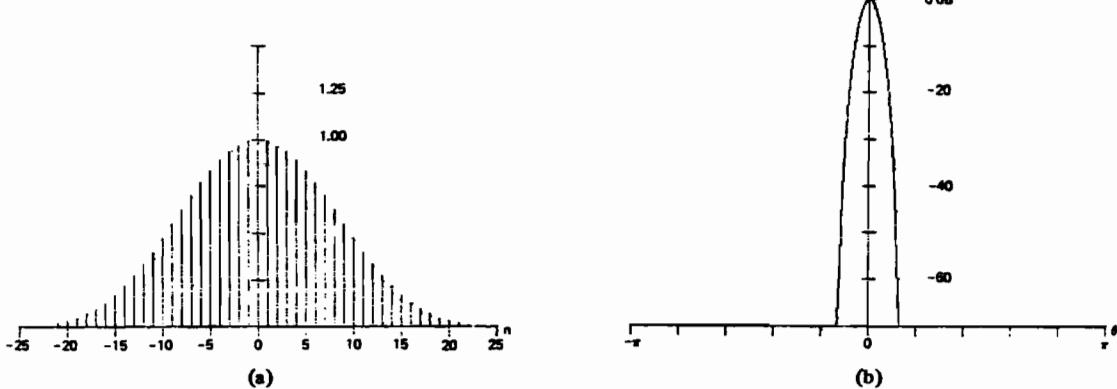


Fig. 26. (a) 4-sample Kaiser-Bessel window. (b) Log-magnitude of transform.

	3-Term (-67 dB)	3-Term (-61 dB)	4-Term (-92 dB)	4-Term (-74 dB)
a_0	0.42323	0.44959	0.35875	0.40217
a_1	0.49755	0.49364	0.48829	0.49703
a_2	0.07922	0.05677	0.14128	0.09392
a_3	---	---	0.01168	0.00183

(to establish another data point in Fig. 12), the minimum 4-term window (to also establish a data point in Fig. 12), and another 4-term window which is presented in Fig. 25. The particular 4-term window shown is one which performs well in a detection example described in Section VI (see Fig. 69). The parameters of these windows are listed in Table I. Note in particular where the Blackman and the Blackman-Harris windows reside in Fig. 12. They are surprisingly good windows for the small number of terms in their trigonometric series. Note, if we were to extend the line connecting the Blackman-Harris family it would intersect the Hamming window which, in Section V-D, we noted is nearly the minimum sidelobe level 2-term Blackman-Harris window.

We also mention that a good approximation to the Blackman-Harris 3- and 4-term windows can be obtained as scaled samples of the Kaiser-Bessel window's transform (see Section V-H). We have used this approximation to construct 4-term windows for adjustable bandwidth convolutional filters as reported in [10]. This approximation is defined as

$$\begin{aligned} b_m &= \frac{\sinh [\pi\sqrt{\alpha^2 - m^2}]}{\pi\sqrt{\alpha^2 - m^2}}, \quad m \leq \alpha, \quad 2 \leq \alpha \leq 4 \\ c &= b_0 + 2b_1 + 2b_2 + (2b_3) \\ a_0 &= \frac{b_0}{c} \quad a_m = 2 \frac{b_m}{c}, \quad m = 1, 2, (3). \end{aligned} \quad (34)$$

The 4 coefficients for this approximation when $\alpha = 3.0$ are $a_0 = 0.40243$, $a_1 = 0.49804$, $a_2 = 0.09831$, and $a_3 = 0.00122$. Notice how close these terms are to the selected 4-term Blackman-Harris (-74 dB) window. The window defined by these coefficients is shown in Fig. 26. Like the prototype from which it came (the Kaiser-Bessel with $\alpha = 3.0$), this window exhibits sidelobes just shy of -70 dB from the main lobe. On the scale shown, the two are indistinguishable. The parameters of this window are also listed in Table I and the window is entered in Fig. 12 as the "4-sample Kaiser-Bessel." It was these 3- and 4-sample Kaiser-Bessel prototype

windows (parameterized on α) which were the starting conditions for the gradient minimization which leads to the Blackman-Harris windows. The optimization starting with these coefficients has virtually no effect on the main-lobe characteristics but does drive down the sidelobes approximately 5 dB.

F. Constructed Windows

Numerous investigators have constructed windows as products, as sums, as sections, or as convolutions of simple functions and of other simple windows. These windows have been constructed for certain desirable features, not the least of which is the attraction of simple functions for generating the window terms. In general, the constructed windows tend not to be good windows, and occasionally are very bad windows. We have already examined some simple window constructions. The Fejer (Bartlett) window, for instance, is the convolution of two rectangle windows; the Hamming window is the sum of a rectangle and a Hanning window; and the $\cos^4(X)$ window is the product of two Hanning windows. We will now examine other constructed windows that have appeared in the literature. We will present them so they are available for comparison. Later we will examine windows constructed in accord with some criteria of optimality (see Sections V-G, H, I, and J). Each window is identified only for the finite Fourier transform. A simple shift of $N/2$ points and right end-point deletion will supply the DFT version. The significant figures of performance for these windows are also found in Table I.

1) *Riesz (Bochner, Parzen) Window* [11]: The Riesz window, identified as

$$w(n) = 1.0 - \left| \frac{n}{N/2} \right|^2, \quad 0 \leq |n| \leq \frac{N}{2} \quad (35)$$

is the simplest continuous polynomial window. It exhibits a discontinuous first derivative at the boundaries; hence its transform falls off like $1/\omega^2$. The window is shown in Fig. 27. The first sidelobe is -22 dB from the main lobe. This window is similar to the cosine lobe (26) as can be demonstrated by examining its Taylor series expansion.

2) *Riemann Window* [12]: The Riemann window, defined by

$$w(n) = \frac{\sin \left[\frac{n}{N} 2\pi \right]}{\left[\frac{n}{N} 2\pi \right]}, \quad 0 \leq |n| \leq \frac{N}{2} \quad (36)$$

is the central lobe of the SINC kernel. This window is con-

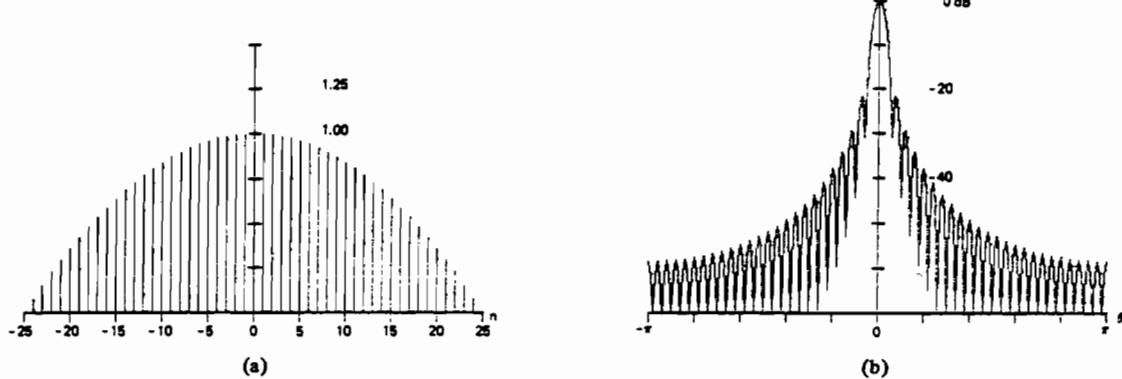


Fig. 27. (a) Riesz window. (b) Log-magnitude of transform.

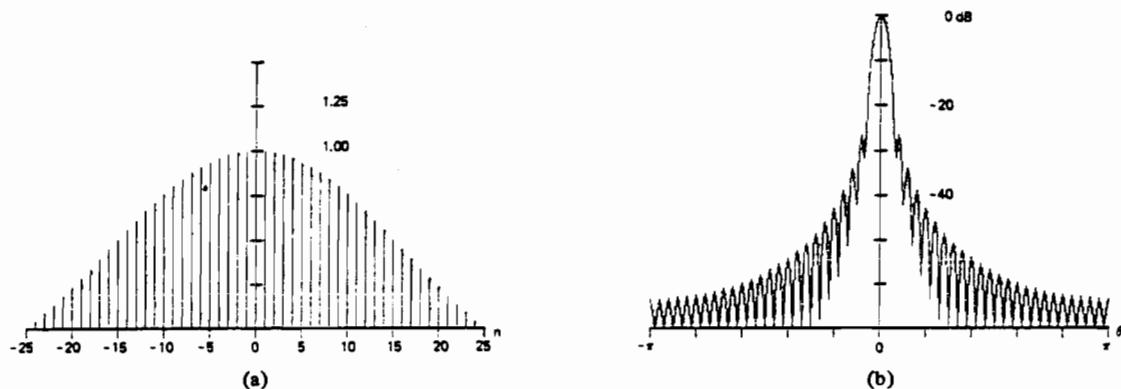


Fig. 28. (a) Riemann window. (b) Log-magnitude of transform.

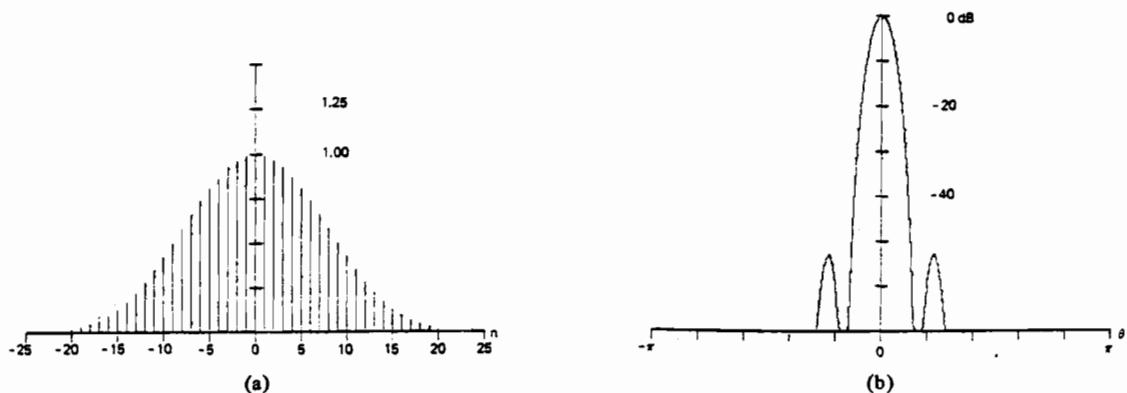


Fig. 29. (a) The de la Vallé-Poussin window. (b) Log-magnitude of transform.

tinuous, with a discontinuous first derivative at the boundary. It is similar to the Riesz and cosine lobe windows. The Riemann window is shown in Fig. 28.

3) *de la Vallé-Poussin (Jackson, Parzen) Window* [11]: The de la Vallé-Poussin window is a piecewise cubic curve obtained by self-convolving two triangles of half extent or four rectangles of one-fourth extent. It is defined as

$$w(n) = \begin{cases} 1.0 - 6 \left[\frac{n}{N/2} \right]^2 \left[1.0 - \frac{|n|}{N/2} \right], & 0 \leq |n| \leq \frac{N}{4} \\ 2 \left[1.0 - \frac{|n|}{N/2} \right]^3, & \frac{N}{4} \leq |n| \leq \frac{N}{2}. \end{cases} \quad (37)$$

The window is continuous up to its third derivative so that its sidelobes fall off like $1/\omega^4$. The window is shown in Fig. 29. Notice the trade off of main-lobe width for sidelobe level. Compare this with the rectangle and the triangle. It is a non-negative window by virtue of its self-convolution construction.

4) *Tukey Window* [13]: The Tukey window, often called the cosine-tapered window, is best imagined as a cosine lobe of width $(\alpha/2)N$ convolved with a rectangle window of width $(1.0 - \alpha/2)N$. Of course the resultant transform is the product of the two corresponding transforms. The window represents an attempt to smoothly set the data to zero at the boundaries while not significantly reducing the processing gain of the windowed transform. The window evolves from the rectangle to the Hanning window as the parameter α varies from zero to unity. The family of windows exhibits a confusing array of

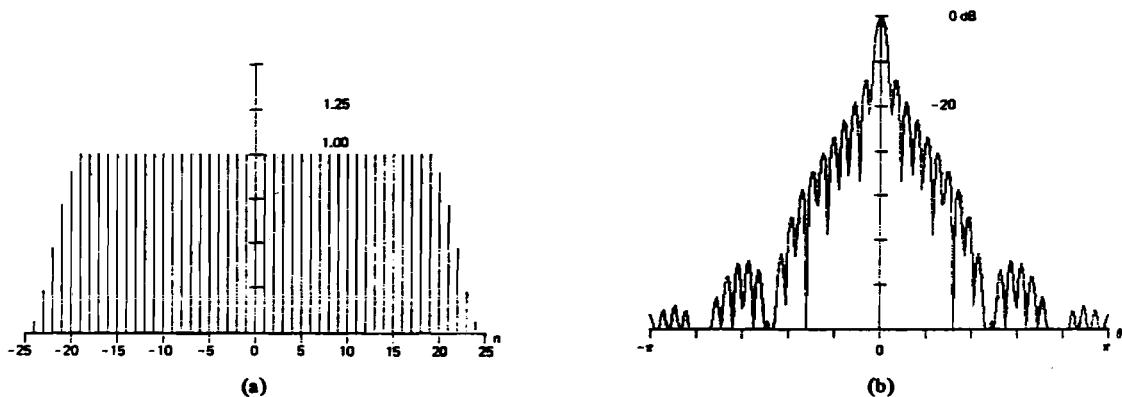


Fig. 30. (a) 25-percent cosine taper (Tukey) window. (b) Log-magnitude of transform.

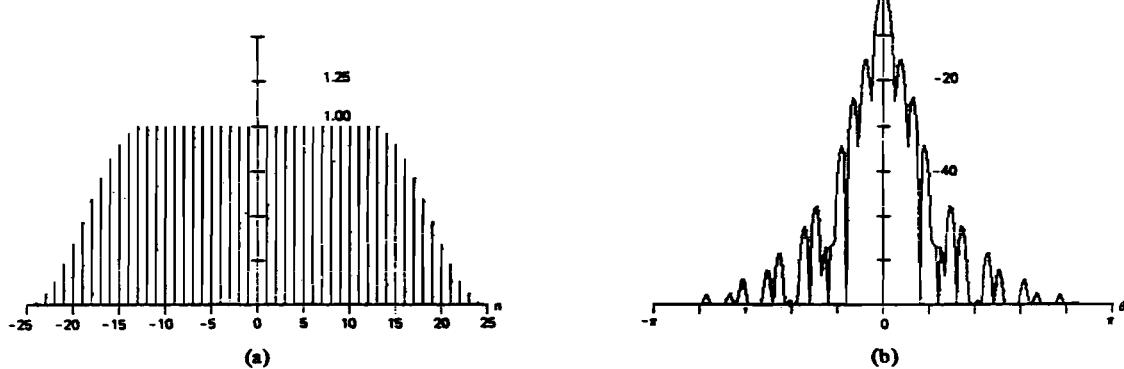


Fig. 31. (a) 50-percent cosine taper (Tukey) window. (b) Log-magnitude of transform.

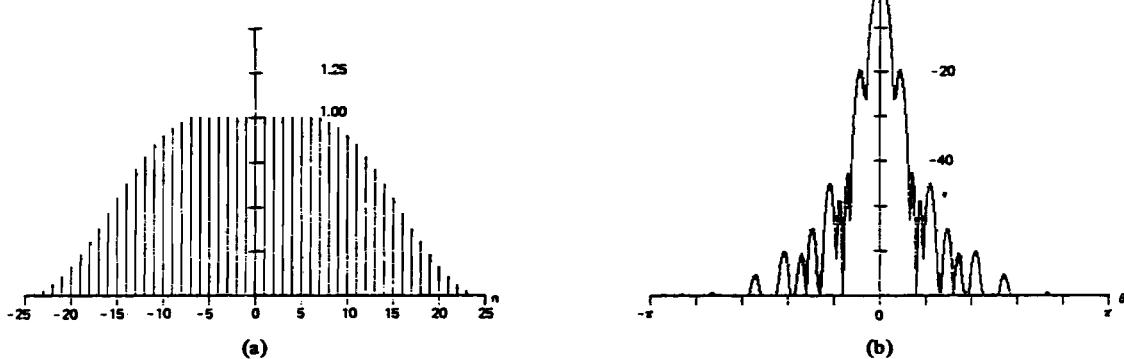


Fig. 32. (a) 75-percent cosine taper (Tukey) window. (b) Log-magnitude of transform.

sidelobe levels arising from the product of the two component transforms. The window is defined by

$$w(n) = \begin{cases} 1.0, & 0 \leq |n| \leq \alpha \frac{N}{2} \\ 0.5 \left[1.0 + \cos \left[\pi \frac{n - \alpha \frac{N}{2}}{2(1 - \alpha) \frac{N}{2}} \right] \right], & \alpha \frac{N}{2} \leq |n| \leq \frac{N}{2} \end{cases} \quad (38)$$

The window is shown in Figs. 30-32 for values of α equal to 0.25, 0.50, and 0.75, respectively.

5) *Bohman Window* [14]: The Bohman window is ob-

tained by the convolution of two half-duration cosine lobes (26a), thus its transform is the square of the cosine lobe's transform (see Fig. 16). In the time domain the window can be described as a product of a triangle window with a single cycle of a cosine with the same period and, then, a corrective term added to set the first derivative to zero at the boundary. Thus the second derivative is continuous, and the discontinuity resides in the third derivative. The transform falls off like $1/\omega^4$. The window is defined in the following and is shown in Fig. 33:

$$w(n) = \left[1.0 - \frac{|n|}{N/2} \right] \cos \left[\pi \frac{|n|}{N/2} \right] + \frac{1}{\pi} \sin \left[\pi \frac{|n|}{N/2} \right], \quad 0 \leq |n| \leq \frac{N}{2}. \quad (39)$$

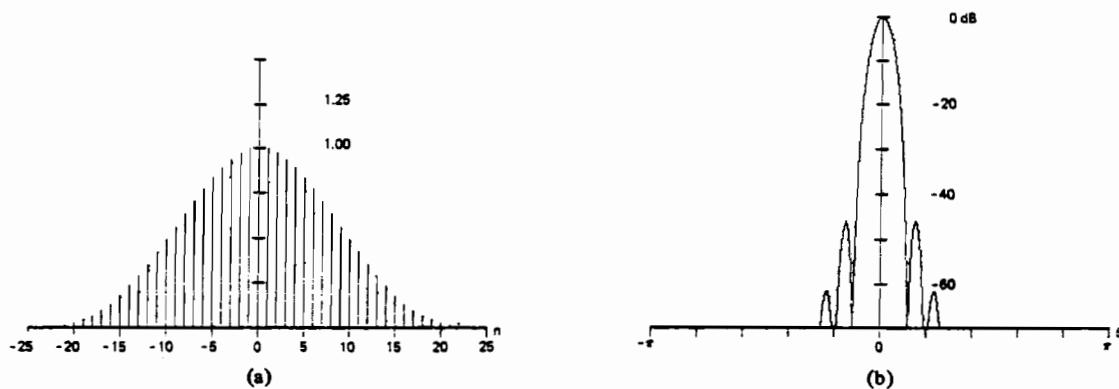
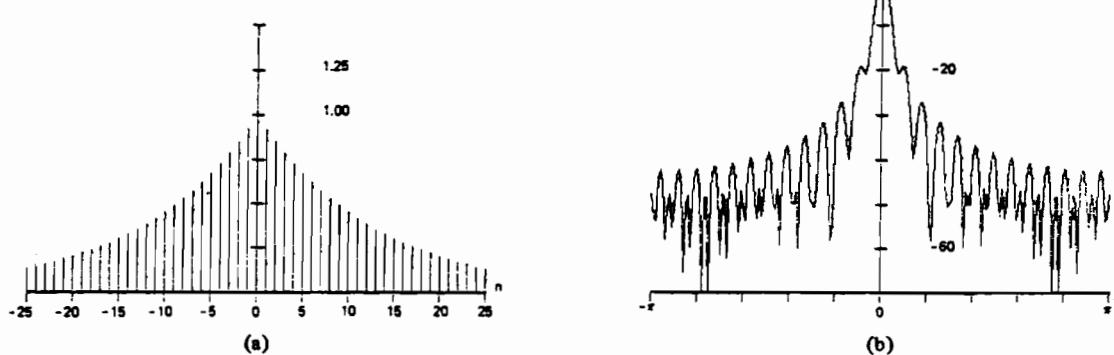
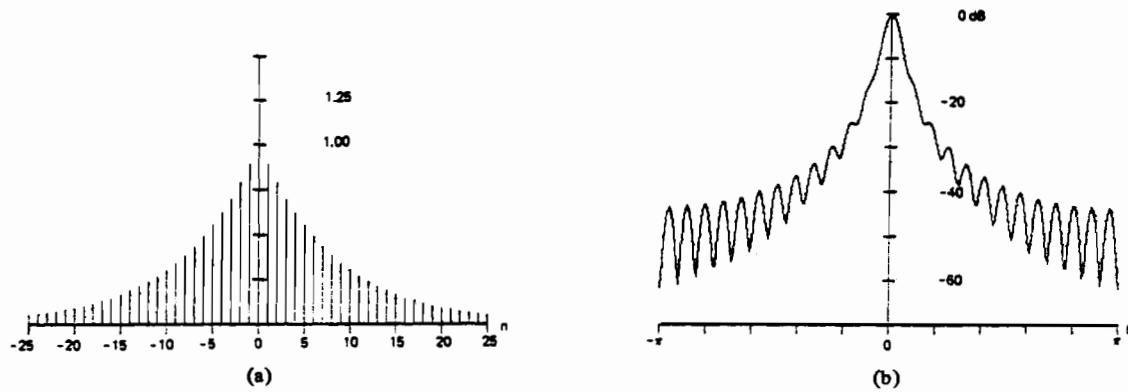


Fig. 33. (a) Bohman window. (b) Log-magnitude of transform.

Fig. 34. (a) Poisson window. (b) Log-magnitude of transform ($\alpha = 2.0$).Fig. 35. (a) Poisson window. (b) Log-magnitude of transform ($\alpha = 3.0$).

6) *Poisson Window [12]*: The Poisson window is a two-sided exponential defined by

$$w(n) = \exp\left(-\alpha \frac{|n|}{N/2}\right), \quad 0 \leq |n| \leq \frac{N}{2}. \quad (40)$$

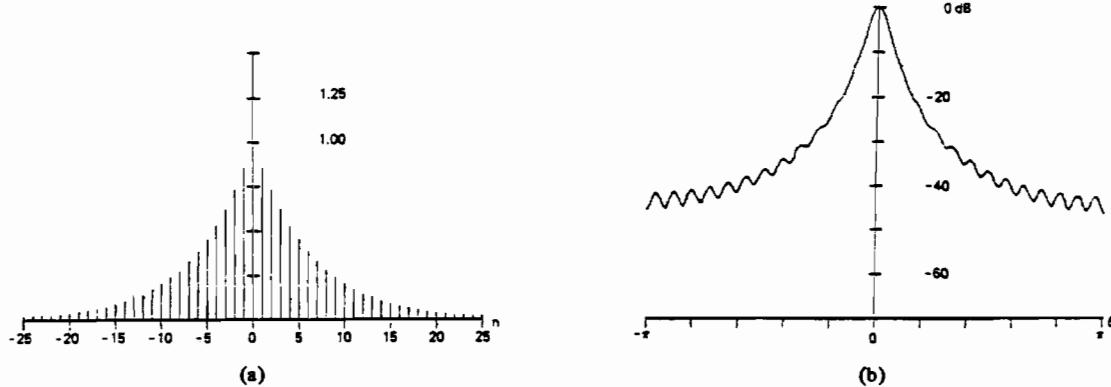
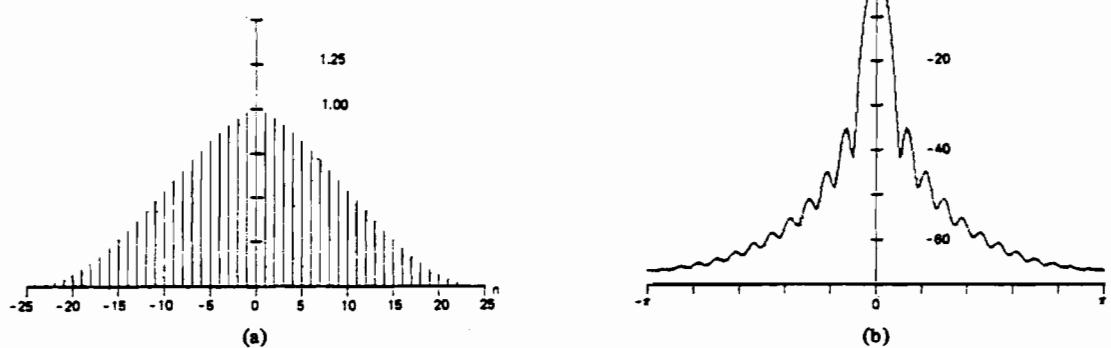
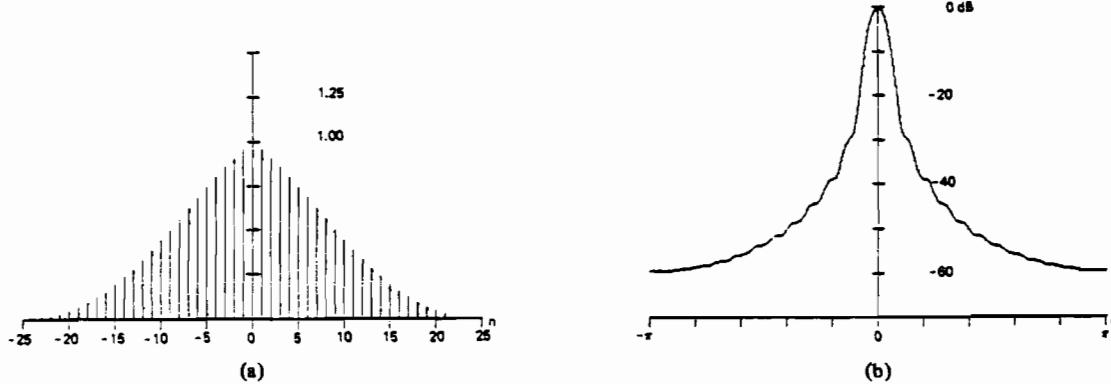
This is actually a family of windows parameterized on the variable α . Since it exhibits a discontinuity at the boundaries, the transform can fall off no faster than $1/\omega$. The window is shown in Figs. 34–36 for values of α equal to 2.0, 3.0, and 4.0, respectively. Notice as the discontinuity at the boundaries becomes smaller, the sidelobe structure merges into the asymptote. Also note the very wide main lobe; this will be

observed in Table I as a large equivalent noise bandwidth and as a large worst case processing loss.

7) *Hanning-Poisson Window*: The Hanning-Poisson window is constructed as the product of the Hanning and the Poisson windows. The family is defined by

$$w(n) = 0.5 \left[1.0 + \cos\left(\pi \frac{n}{N/2}\right) \right] \exp\left(-\alpha \frac{|n|}{N/2}\right), \quad 0 \leq |n| \leq \frac{N}{2}. \quad (41)$$

This window is similar to the Poisson window. The rate of sidelobe falloff is determined by the discontinuity in the first

Fig. 36. (a) Poisson window. (b) Log-magnitude of transform ($\alpha = 4.0$).Fig. 37. (a) Hanning-Poisson window. (b) Log-magnitude of transform ($\alpha = 0.5$).Fig. 38. (a) Hanning-Poisson window. (b) Log-magnitude of transform ($\alpha = 1.0$).

derivative at the origin and is $1/\omega^2$. Notice as α increases, forcing more of the exponential into the Hanning window, the zeros of the sidelobe structure disappear and the lobes merge into the asymptote. This window is shown in Figs. 37-39 for values of α equal to 0.5, 1.0, and 2.0, respectively. Again note the very large main-lobe width.

8) *Cauchy (Abel, Poisson) Window [15]*: The Cauchy window is a family parameterized on α and defined by

$$w(n) = \frac{1}{1 + \left[\alpha \frac{n}{N/2} \right]^2}, \quad 0 \leq |n| \leq \frac{N}{2}. \quad (42)$$

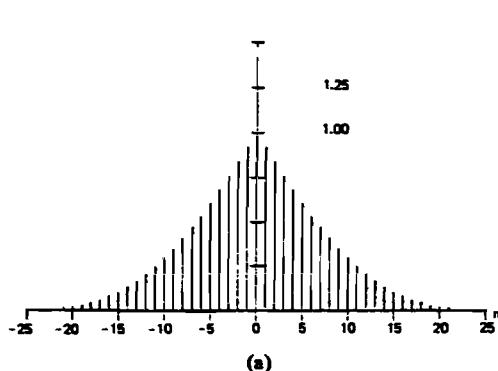
The window is shown in Figs. 40-42 for values of α equal to 3.0, 4.0, and 5.0, respectively. Note the transform of the

Cauchy window is a two-sided exponential (see Poisson windows), which when presented on a log-magnitude scale is essentially an isosceles triangle. This causes the window to exhibit a very wide main lobe and to have a large ENBW.

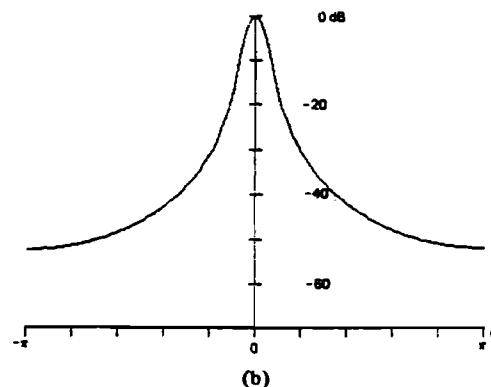
G. Gaussian or Weierstrass Window [15]

Windows are smooth positive functions with tall thin (i.e., concentrated) Fourier transforms. From the generalized uncertainty principle, we know we cannot simultaneously concentrate both a signal and its Fourier transform. If our measure of concentration is the mean-square time duration T and the mean-square bandwidth W , we know all functions satisfy the inequality of

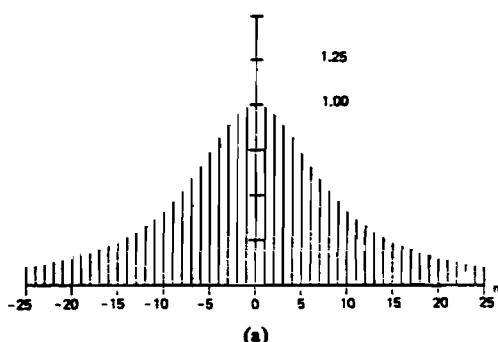
$$TW \geq \frac{1}{4\pi} \quad (43)$$



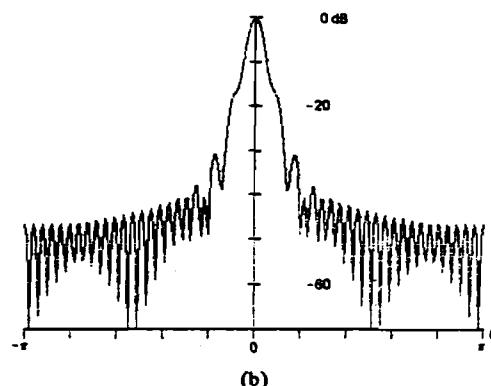
(a)



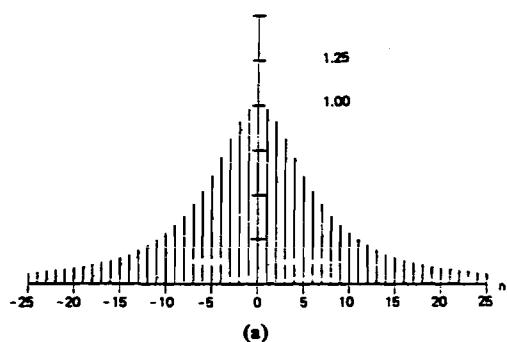
(b)

Fig. 39. (a) Hanning-Poisson window. (b) Log-magnitude of transform ($\alpha = 2.0$).

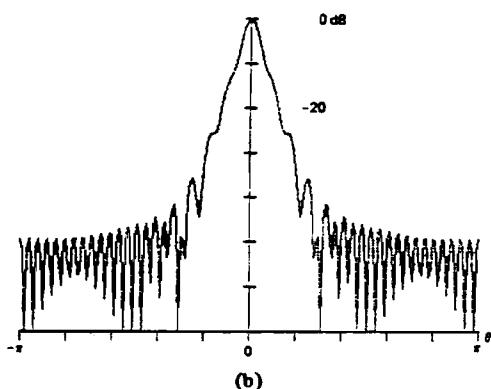
(a)



(b)

Fig. 40. (a) Cauchy window. (b) Log-magnitude of transform ($\alpha = 3.0$).

(a)



(b)

Fig. 41. (a) Cauchy window. (b) Log-magnitude of transform ($\alpha = 4.0$).

with equality being achieved only for the Gaussian pulse [16]. Thus the Gaussian pulse, characterized by minimum time-bandwidth product, is a reasonable candidate for a window. When we use the Gaussian pulse as a window we have to truncate or discard the tails. By restricting the pulse to be finite length, the window no longer is minimum time-bandwidth. If the truncation point is beyond the three-sigma point, the error should be small, and the window should be a good approximation to minimum time-bandwidth.

The Gaussian window is defined by

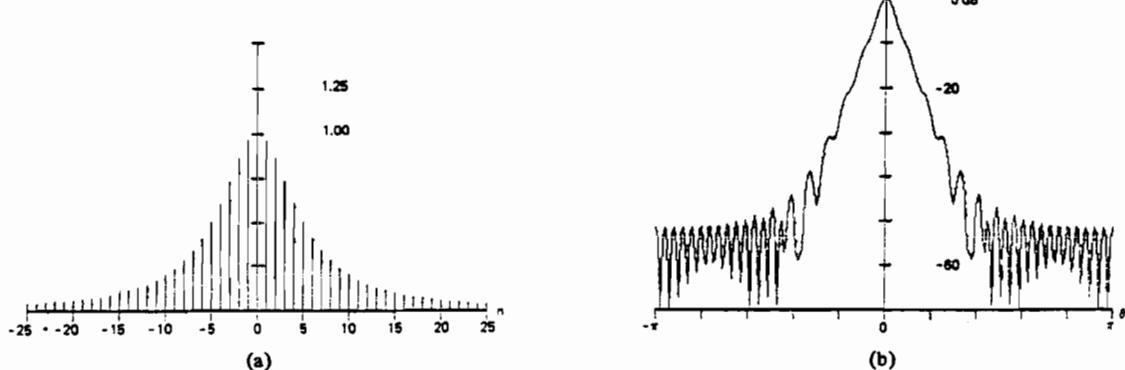
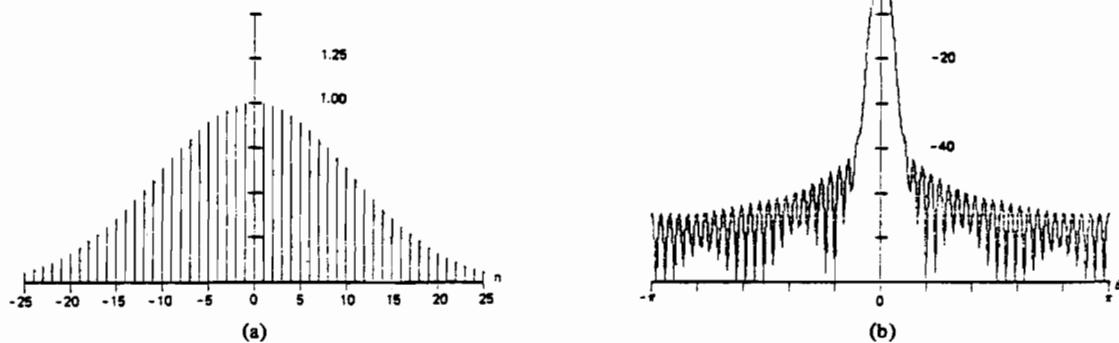
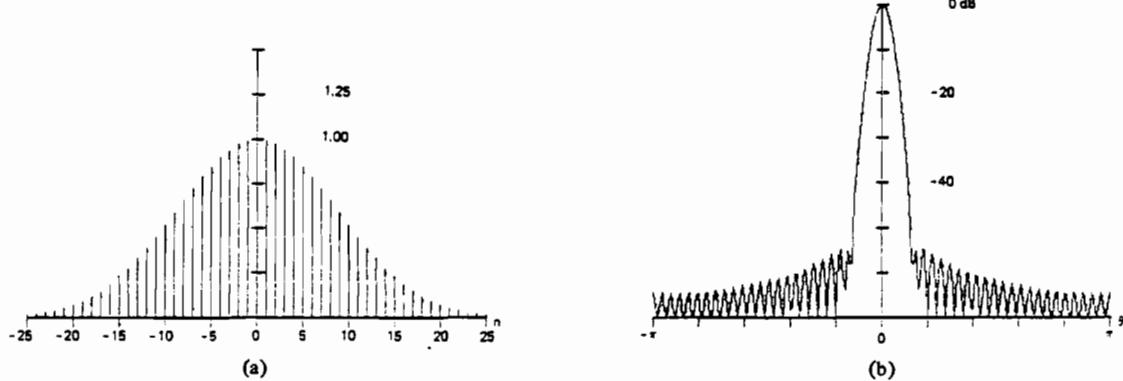
$$w(n) = \exp \left[-\frac{1}{2} \left[\alpha \frac{n}{N/2} \right]^2 \right]. \quad (44a)$$

The transform is the convolution of a Gaussian transform with

a Dirichlet kernel as indicated in

$$\begin{aligned} W(\theta) &= \frac{1}{2} \frac{\sqrt{2\pi}}{\alpha} \exp \left[-\frac{1}{2} \left[\frac{1}{\alpha} \theta \right]^2 \right] * D(\theta) \\ &\simeq \frac{N}{2} \frac{\sqrt{2\pi}}{\alpha} \exp \left[-\frac{1}{2} \left[\frac{1}{\alpha} \theta \right]^2 \right], \quad \text{for } \alpha > 2.5, \text{ and } \theta \text{ small.} \end{aligned} \quad (44b)$$

This window is parameterized on α , the reciprocal of the standard deviation, a measure of the width of its Fourier transform. Increased α will decrease with the width of the window and reduce the severity of the discontinuity at the boundaries. This will result in an increased width transform

Fig. 42. (a) Cauchy window. (b) Log-magnitude of transform ($a = 5.0$).Fig. 43. (a) Gaussian window. (b) Log-magnitude of transform ($a = 2.5$).Fig. 44. (a) Gaussian window. (b) Log-magnitude of transform ($a = 3.0$).

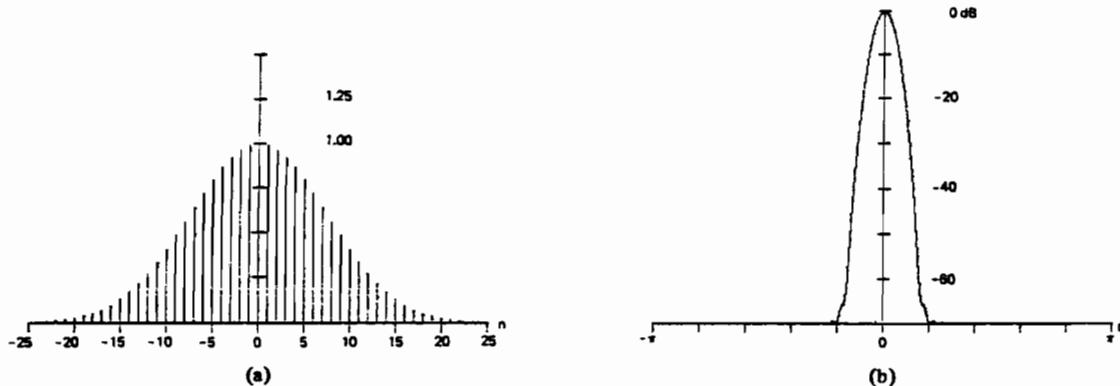
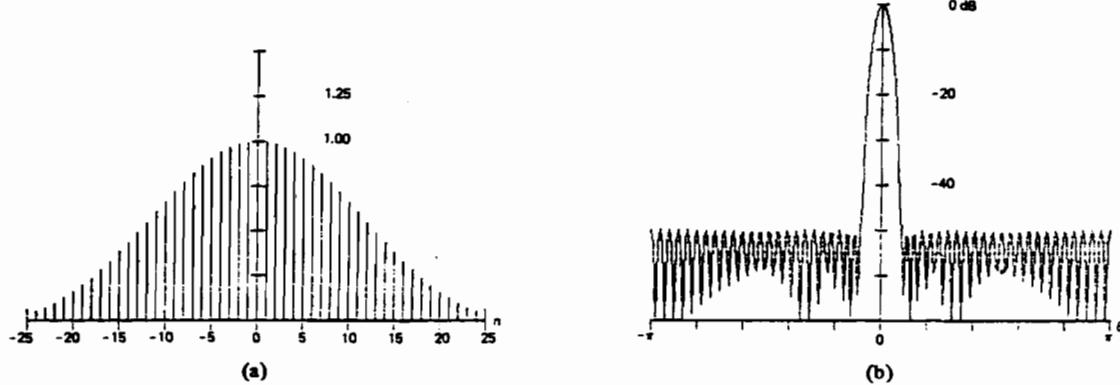
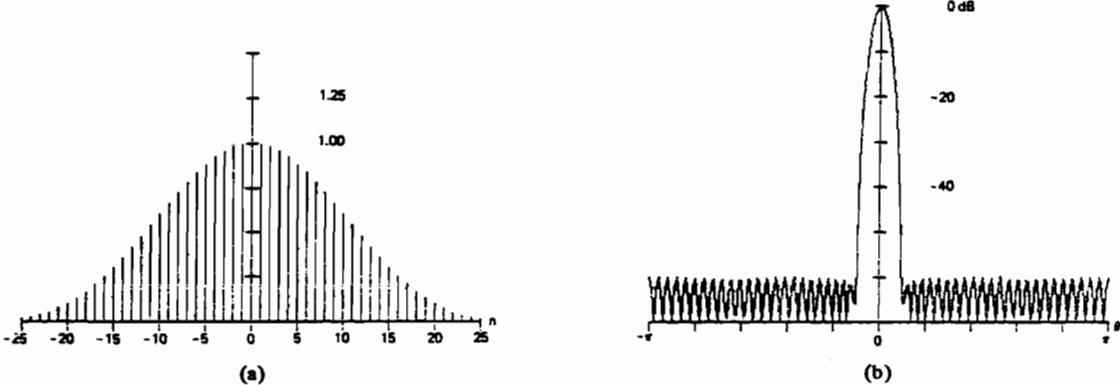
main lobe and decreased sidelobe levels. The window is presented in Figs. 43, 44, and 45 for values of α equal to 2.5, 3.0, and 3.5, respectively. Note the rapid drop-off rate of sidelobe level in the exchange of sidelobe level for main-lobe width. The figures of merit for this window are listed in Table I.

H. Dolph-Chebyshev Window [17]

Following the reasoning of the previous section, we seek a window which, for a known finite duration, in some sense exhibits a narrow bandwidth. We now take a lead from the antenna design people who have faced and solved a similar problem. The problem is to illuminate an antenna of finite

aperture to achieve a narrow main-lobe beam pattern while simultaneously restricting sidelobe response. (The antenna designer calls his weighting procedure *shading*.) The closed-form solution to the minimum main-lobe width for a given sidelobe level is the Dolph-Chebyshev window (shading). The continuous solution to the problem exhibits impulses at the boundaries which restricts continuous realizations to approximations (the Taylor approximation). The discrete or sampled window is not so restricted, and the solution can be implemented exactly.

The relation $T_n(X) = \cos(n\theta)$ describes a mapping between the n th-order Chebyshev (algebraic) polynomial and the n th-order trigonometric polynomial. The Dolph-Chebyshev

Fig. 45. (a) Gaussian window. (b) Log-magnitude of transform ($\alpha = 3.5$).Fig. 46. (a) Dolph-Chebyshev window. (b) Log-magnitude of transform ($\alpha = 2.5$).Fig. 47. (a) Dolph-Chebyshev window. (b) Log-magnitude of transform ($\alpha = 3.0$).

window is defined with this mapping in the following equation, in terms of uniformly spaced samples of the window's Fourier transform,

$$W(k) = (-1)^k \frac{\cos \left[N \cos^{-1} \left[\beta \cos \left(\pi \frac{k}{N} \right) \right] \right]}{\cosh [N \cosh^{-1}(\beta)]}, \quad 0 \leq |k| \leq N-1 \quad (45)$$

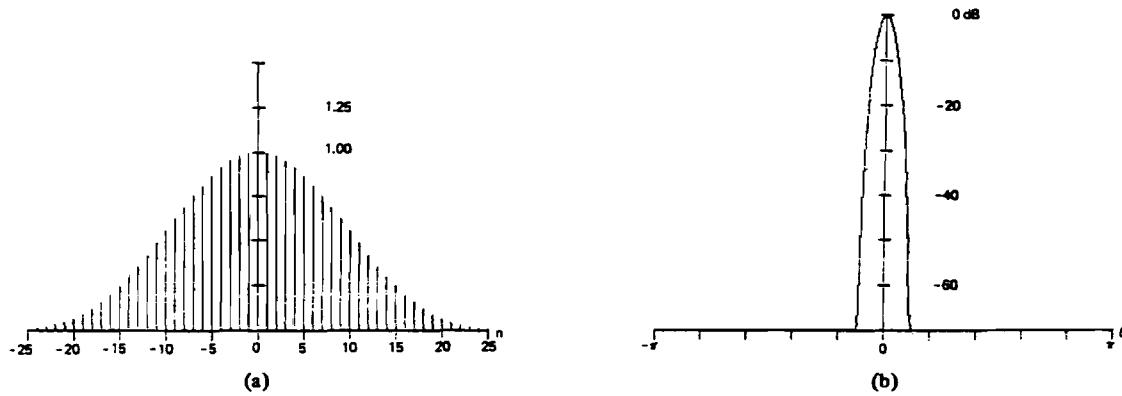
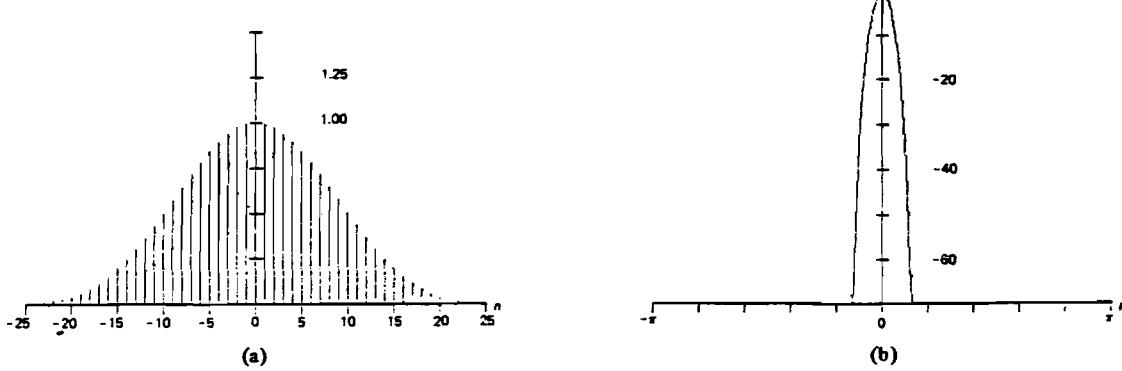
where

$$\beta = \cosh \left[\frac{1}{N} \cosh^{-1}(10^\alpha) \right]$$

and

$$\cos^{-1}(X) = \begin{cases} \frac{\pi}{2} - \tan^{-1}[X/\sqrt{1.0 - X^2}], & |X| \leq 1.0 \\ \ln[X + \sqrt{X^2 - 1.0}], & |X| \geq 1.0. \end{cases}$$

To obtain the corresponding window time samples $w(n)$, we simply perform a DFT on the samples $W(k)$ and then scale for unity peak amplitude. The parameter α represents the log of the ratio of main-lobe level to sidelobe level. Thus a value of α equal to 3.0 represents sidelobes 3.0 decades down from the main lobe, or sidelobes 60.0 dB below the main lobe. The $(-1)^k$ alternates the sign of successive transform samples to reflect the shifted origin in the time domain. The window is

Fig. 48. (a) Dolph-Chebyshev window. (b) Log-magnitude of transform ($\alpha = 3.5$).Fig. 49. (a) Dolph-Chebyshev window. (b) Log-magnitude of transform ($\alpha = 4.0$).

presented in Figs. 46-49 for values of α equal to 2.5, 3.0, 3.5, and 4.0, respectively. Note the uniformity of the sidelobe structure; almost sinusoidal! It is this uniform oscillation which is responsible for the impulses in the window.

I. Kaiser-Bessel Window [18]

Let us examine for a moment the optimality criteria of the last two sections. In Section V-G we sought the function with minimum time-bandwidth product. We know this to be the Gaussian. In Section V-H we sought the function with restricted time duration, which minimized the main-lobe width for a given sidelobe level. We now consider a similar problem. For a restricted energy, determine the function of restricted time duration T which maximizes the energy in the band of frequencies W . Slepian, Pollak, and Landau [19], [20] have determined this function as a family parameterized over the time-bandwidth product, the prolate-spheroidal wave functions of order zero. Kaiser has discovered a simple approximation to these functions in terms of the zero-order modified Bessel function of the first kind. The Kaiser-Bessel window is defined by

$$w(n) = \frac{I_0\left[\pi\alpha \sqrt{1.0 - \left(\frac{n}{N/2}\right)^2}\right]}{I_0[\pi\alpha]}, \quad 0 \leq |n| \leq \frac{N}{2} \quad (46a)$$

where

$$I_0(X) = \sum_{k=0}^{\infty} \left[\frac{\left(\frac{x}{2}\right)^k}{k!} \right]^2.$$

The parameter $\pi\alpha$ is half of the time-bandwidth product. The transform is approximately that of

$$W(\theta) = \frac{N}{I_0(\pi\alpha)} \frac{\sinh [\sqrt{\alpha^2\pi^2 - (N\theta/2)^2}]}{\sqrt{\alpha^2\pi^2 - (N\theta/2)^2}}. \quad (46b)$$

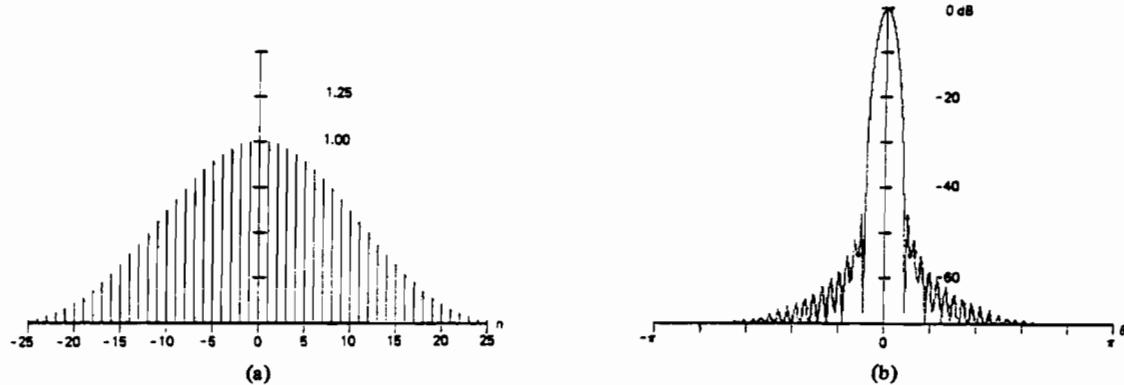
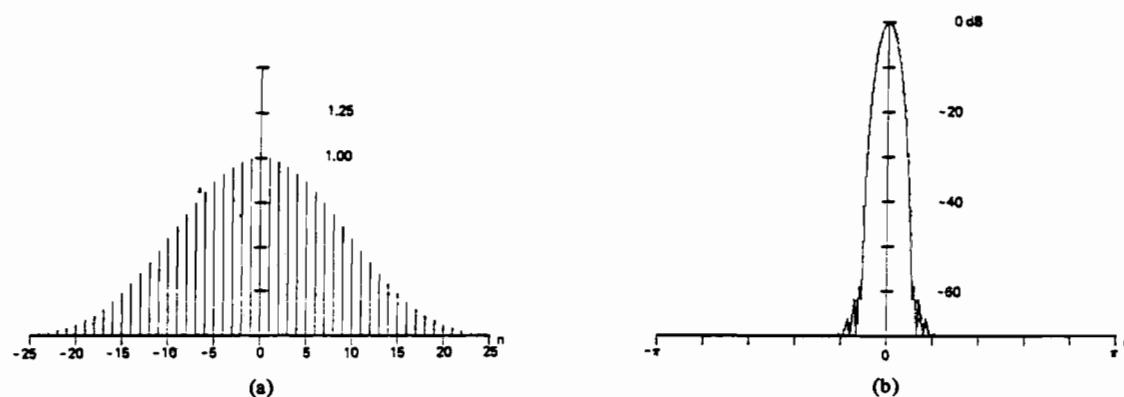
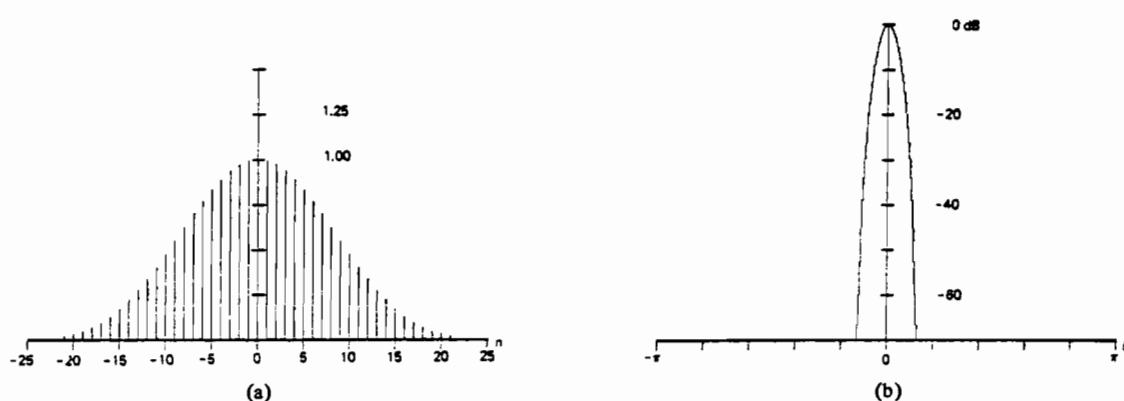
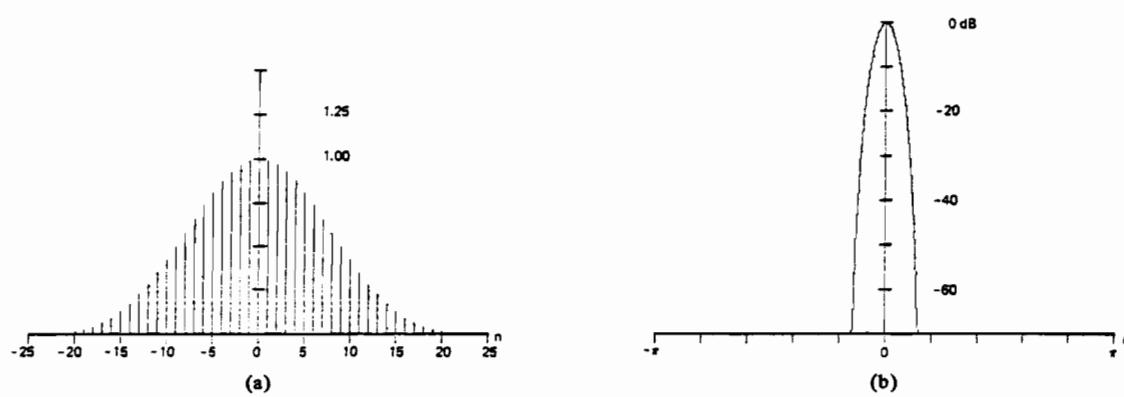
This window is presented in Figs. 50-53 for values of α equal to 2.0, 2.5, 3.0, and 3.5, respectively. Note the trade off between sidelobe level and main-lobe width.

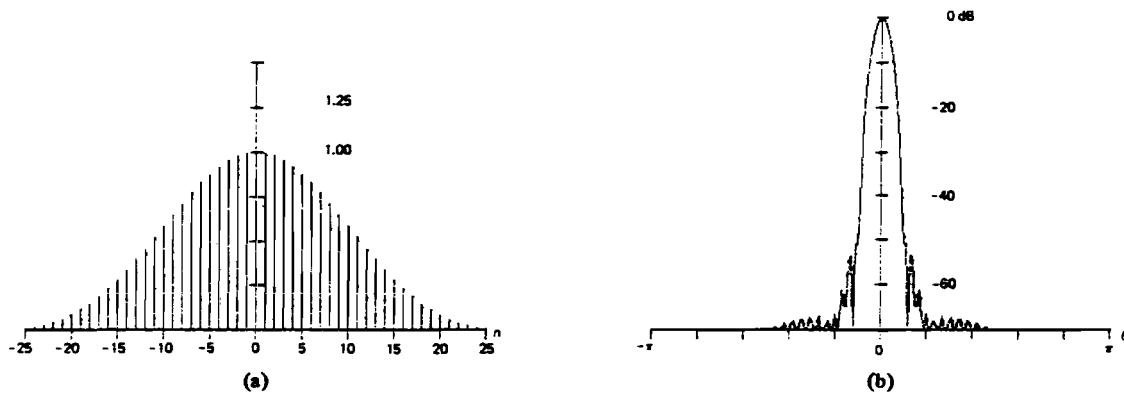
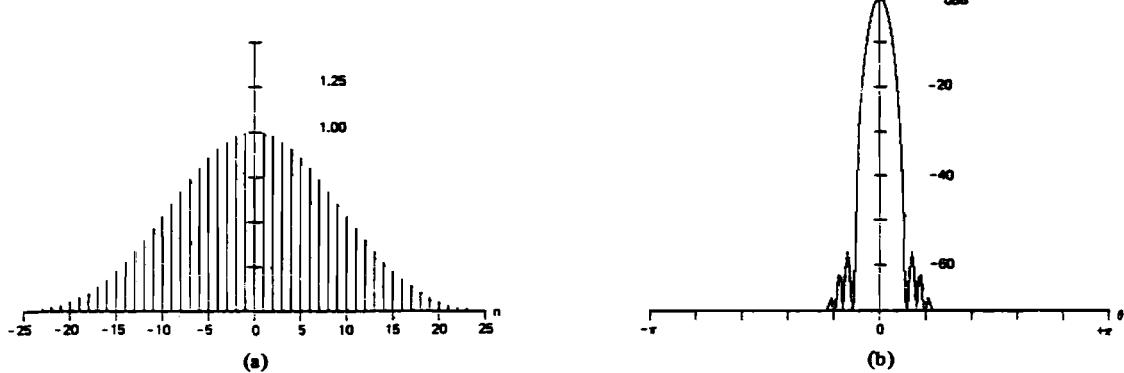
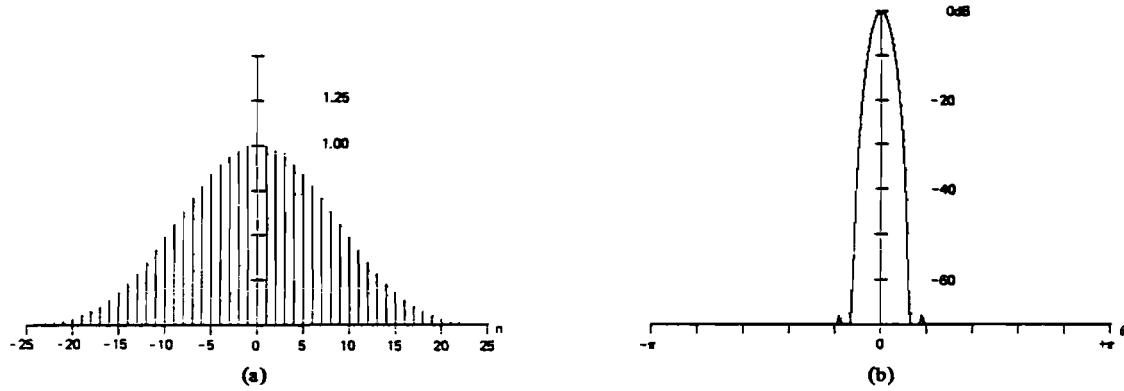
J. Barcilon-Temes Window [21]

We now examine the last criterion of optimality for a window. We have already described the Slepian, Pollak, and Landau criterion. Subject to the constraints of fixed energy and fixed duration, determine the function which maximizes the energy in the band of frequencies W . A related criterion, subject to the constraints of fixed area and fixed duration, is to determine the function which minimizes the energy (or the weighted energy) outside the band of frequencies W . This is a reasonable criterion since we recognize that the transform of a good window should minimize the energy it gathers from frequencies removed from its center frequency. Till now, we have been responding to this goal by maximizing the concentration of the transform at its main lobe.

A closed-form solution of the unweighted minimum-energy criterion has not been found. A solution defined as an expansion of prolate-spheroidal wave functions does exist and it is of the form shown in

$$H\left(\frac{\omega}{W}\right) = \sum_n \frac{\psi_{2n}(\pi\alpha, 0)}{1 - \lambda_{2n}} \psi_{2n}\left(\pi\alpha, \frac{\omega}{W}\right). \quad (47)$$

Fig. 50. (a) Kaiser-Bessel window. (b) Log-magnitude of transform ($\alpha = 2.0$).Fig. 51. (a) Kaiser-Bessel window. (b) Log-magnitude of transform ($\alpha = 2.5$).Fig. 52. (a) Kaiser-Bessel window. (b) Log-magnitude of transform ($\alpha = 3.0$).Fig. 53. (a) Kaiser-Bessel window. (b) Log-magnitude of transform ($\alpha = 3.5$).

Fig. 54. (a) Barcilon-Temes window. (b) Log-magnitude of transform ($\alpha = 3.0$).Fig. 55. (a) Barcilon-Temes window. (b) Log-magnitude of transform ($\alpha = 3.5$).Fig. 56. (a) Barcilon-Temes window. (b) Log-magnitude of transform ($\alpha = 4.0$).

Here the λ_{2n} is the eigenvalue corresponding to the associated prolate-spheroidal wave function $|\psi_{2n}(x, y)|$, and the $\pi\alpha$ is the selected half time-bandwidth product. The summation converges quite rapidly, and is often approximated by the first term or by the first two terms. The first term happens to be the solution of the Slepian, Pollak, and Landau problem, which we have already examined as the Kaiser-Bessel window.

A closed-form solution of a weighted minimum-energy criterion, presented in the following equation has been found by Barcilon and Temes:

$$\text{Minimize } \int_w^\infty |H(\omega)|^2 \frac{\omega}{\sqrt{\omega^2 - W^2}} d\omega. \quad (48)$$

This criterion is one which is a compromise between the Dolph-Chebyshev and the Kaiser-Bessel window criteria.

Like the Dolph-Chebyshev window, the Fourier transform is more easily defined, and the window time-samples are obtained by an inverse DFT and an appropriate scale factor. The transform samples are defined by

$$W(k) = (-1)^k \frac{A \cos [y(k)] + B \left[\frac{y(k)}{C} \sin [y(k)] \right]}{[C + AB] \left[\left[\frac{y(k)}{C} \right]^2 + 1.0 \right]} \quad (49)$$

where

$$A = \sinh (C) = \sqrt{10^{2\alpha} - 1}$$

$$B = \cosh (C) = 10^\alpha$$

$$C = \cosh^{-1}(10^\alpha)$$

$$\beta = \cosh \left[\frac{1}{N} C \right]$$

$$y(k) = N \cos^{-1} \left[\beta \cos \left(\pi \frac{k}{N} \right) \right].$$

(See also (45).) This window is presented in Figs. 54-56 for values of α equal to 3.0, 3.5, and 4.0, respectively. The main-lobe structure is practically indistinguishable from the Kaiser-Bessel main-lobe. The figures of merit listed on Table I suggest that for the same sidelobe level, this window does indeed reside between the Kaiser-Bessel and the Dolph-Chebyshev windows. It is interesting to examine Fig. 12 and note where this window is located with respect to the Kaiser-Bessel window; striking similarity in performance!

VI. HARMONIC ANALYSIS

We now describe a simple experiment which dramatically demonstrates the influence a window exerts on the detection of a weak spectral line in the presence of a strong nearby line. If two spectral lines reside in DFT bins, the rectangle window allows each to be identified with no interaction. To demonstrate this, consider the signal composed of two frequencies $10 f_s/N$ and $16 f_s/N$ (corresponding to the tenth and the sixteenth DFT bins) and of amplitudes 1.0 and 0.01 (40.0 dB separation), respectively. The power spectrum of this signal obtained by a DFT is shown in Fig. 57 as a linear interpolation between the DFT output points.

We now modify the signal slightly so that the larger signal resides midway between two DFT bins; in particular, at $10.5 f_s/N$. The smaller signal still resides in the sixteenth bin. The power spectrum of this signal is shown in Fig. 58. We note that the sidelobe structure of the larger signal has completely swamped the main lobe of the smaller signal. In fact, we know (see Fig. 13) that the sidelobe amplitude of the rectangle window at 5.5 bins from the center is only 25 dB down from the peak. Thus the second signal (5.5 bins away) could not be detected because it was more than 26 dB down, and hence, hidden by the sidelobe. (The 26 dB comes from the 25-dB sidelobe level minus the 3.9-dB processing loss of the window plus 3.0 dB for a high confidence detection.) We also note the obvious asymmetry around the main lobe centered at 10.5 bins. This is due to the coherent addition of the sidelobe structures of the pair of kernels located at the plus and minus 10.5 bin positions. We are observing the self-leakage between the positive and the negative frequencies. Fig. 59 is the power spectrum of the signal pair, modified so that the large-amplitude signal resides at the 10.25-bin position. Note the change in asymmetry of the main-lobe and the reduction in the sidelobe level. We still can not observe the second signal located at bin position 16.0.

We now apply different windows to the two-tone signal to demonstrate the difference in second-tone detectability. For some of the windows, the poorer resolution occurs when the large signal is at 10.0 bins rather than at 10.5 bins. We will always present the window with the large signal at the location corresponding to worst-case resolution.

The first window we apply is the triangle window (see Fig. 60). The sidelobes have fallen by a factor of two over the rectangle windows' lobes (e.g., the -35-dB level has fallen to -70 dB). The sidelobes of the larger signal have fallen to approximately -43 dB at the second signal so that it is barely

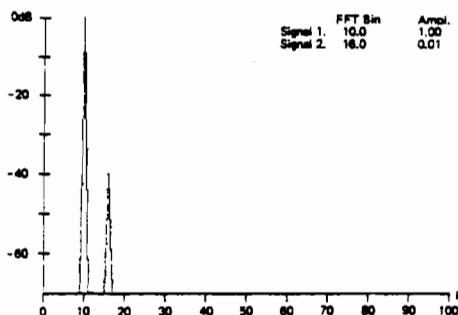


Fig. 57. Rectangle window.

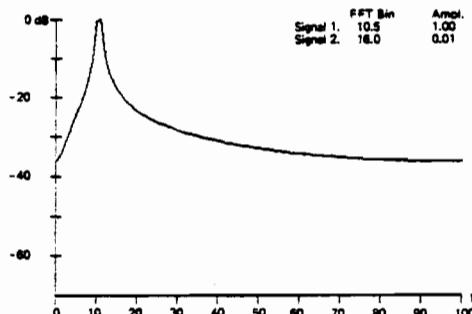


Fig. 58. Rectangle window.

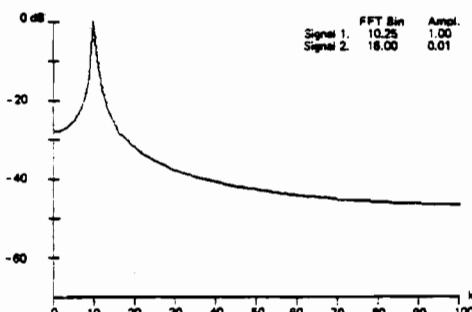


Fig. 59. Rectangle window.

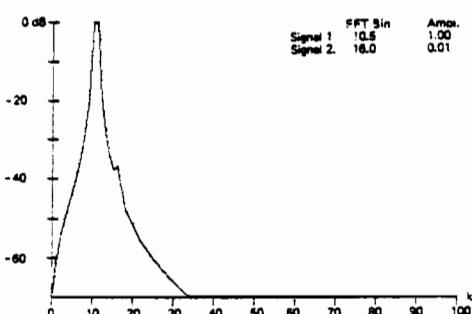


Fig. 60. Triangle window.

detectable. If there were any noise in the signal, the second tone would probably not have been detected.

The next windows we apply are the $\cos^\alpha(x)$ family. For the cosine lobe, $\alpha = 1.0$, shown in Fig. 61 we observe a phase cancellation in the sidelobe of the large signal located at the small signal position. This cannot be considered a detection. We also see the spectral leakage of the main lobe over the frequency axis. Signals below this leakage level would not be detected. With $\alpha = 2.0$ we have the Hanning window, which is

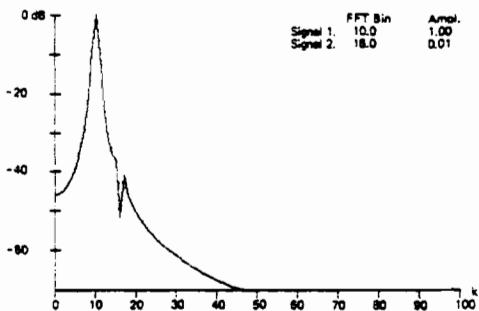
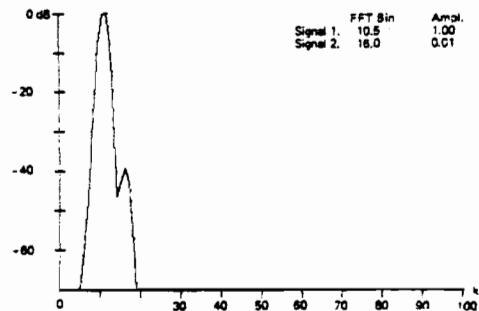
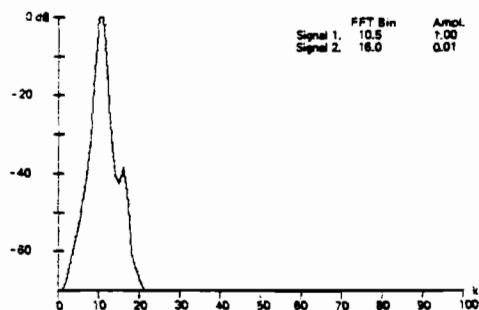
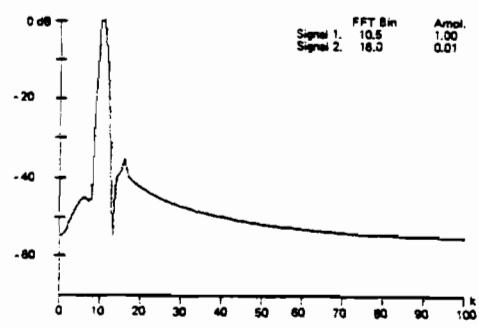
Fig. 61. $\cos(n\pi/N)$ window.Fig. 64. $\cos^4(n\pi/N)$ window.Fig. 62. $\cos^2(n\pi/N)$ window.

Fig. 65. Hamming window.

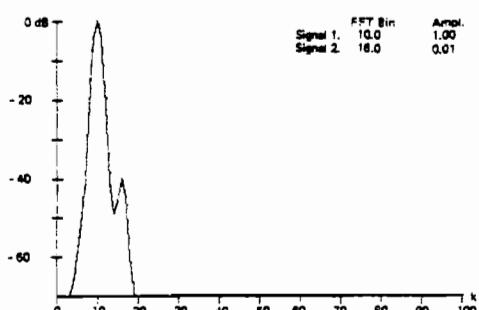
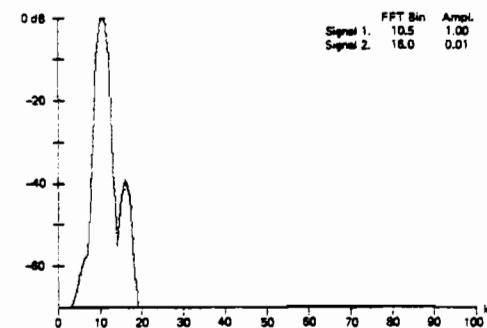
Fig. 63. $\cos^3(n\pi/N)$ window.

Fig. 66. Blackman window.

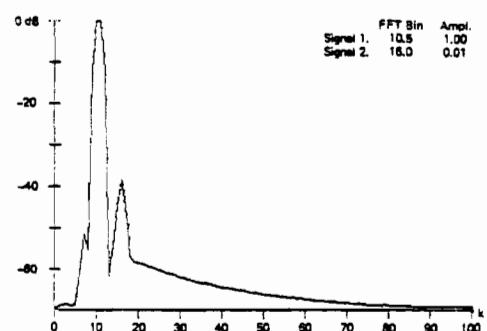


Fig. 67. Exact Blackman window.

presented in Fig. 62. We detect the second signal and observe a 3.0-dB null between the two lobes. This is still a marginal detection. For the $\cos^3(x)$ window presented in Fig. 63, we detect the second signal and observe a 9.0-dB null between the lobes. We also see the improved sidelobe response. Finally for the $\cos^4(x)$ window presented in Fig. 64, we detect the second signal and observe a 7.0-dB null between the lobes. Here we witness the reduced return for the trade between sidelobe level and main-lobe width. In obtaining further reduction in sidelobe level we have caused the increased main-lobe width to encroach upon the second signal.

We next apply the Hamming window and present the result in Fig. 65. Here we observe the second signal some 35 dB down, approximately 3.0 dB over the sidelobe response of the large signal. Here, too, we observe the phase cancellation and the leakage between the positive and the negative frequency components. Signals more than 50 dB down would not be detected in the presence of the larger signal.

The Blackman window is applied next and we see the results in Fig. 66. The presence of the smaller amplitude kernel is now very apparent. There is a 17-dB null between the two signals. The artifact at the base of the large-signal kernel is

the sidelobe structure of that kernel. Note the rapid rate of falloff of the sidelobe leakage has confined the artifacts to a small portion of the spectral line.

We next apply the exact Blackman coefficients and witness the results in Fig. 67. Again the second signal is well defined with a 24-dB null between the two kernels. The sidelobe structure of the larger kernel now extends over the entire spectral range. This leakage is not terribly severe as it is nearly

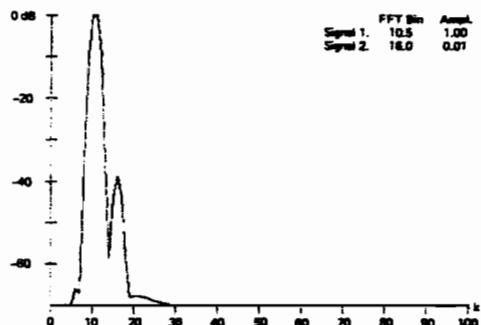


Fig. 68. Minimum 3-term Blackman-Harris window.

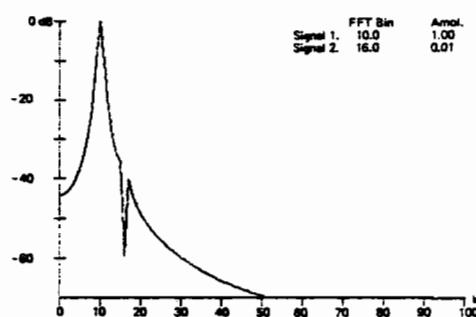


Fig. 71. Riesz window.

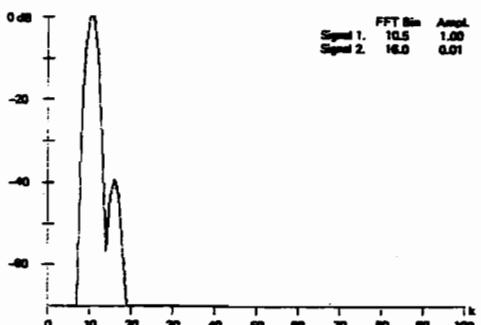


Fig. 69. 4-term Blackman-Harris window.

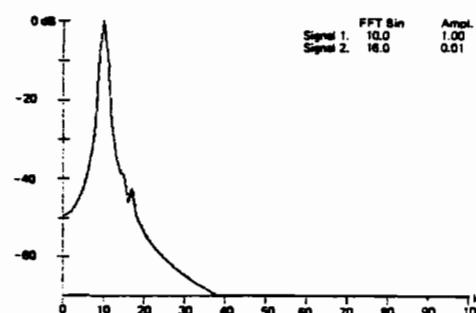


Fig. 72. Riemann window.

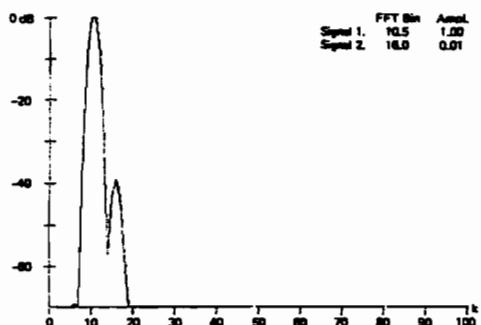


Fig. 70. 4-sample Kaiser-Bessel window.

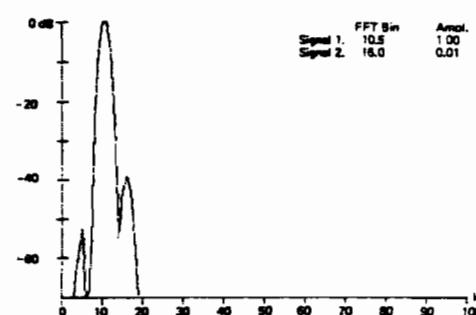


Fig. 73. de la Vallé-Poussin window.

60-dB down relative to the peak. There is another small artifact at 50-dB down on the low frequency side of the large kernel. This is definitely a single sidelobe of the large kernel. This artifact is essentially removed by the minimum 3-term Blackman-Harris window which we see in Fig. 68. The null between the two signal main lobes is slightly smaller, at approximately 20 dB.

Next the 4-term Blackman-Harris window is applied to the signal and we see the results in Fig. 69. The sidelobe structures are more than 70-dB down and as such are not observed on this scale. The two signal lobes are well defined with approximately a 19-dB null between them. Now we apply the 4-sample Kaiser-Bessel window to the signal and see the results in Fig. 70. We have essentially the same performance as with the 4-term Blackman-Harris window. The only observable difference on this scale is the small sidelobe artifact 68 dB down on the low frequency side of the large kernel. This group of Blackman-derived windows perform admirably well for their simplicity.

The Riesz window is the first of our constructed windows and is presented in Fig. 71. We have not detected the second signal but we do observe its effect as a 20.0-dB null due

to the phase cancellation of a sidelobe in the large signal's kernel.

The result of a Riemann window is presented in Fig. 72. Here, too, we have no detection of the second signal. We do have a small null due to phase cancellation at the second signal. We also have a large sidelobe response.

The next window, the de la Vallé-Poussin or the self-convolved triangle, is shown in Fig. 73. The second signal is easily found and the power spectrum exhibits a 16.0-dB null. An artifact of the window (its lower sidelobe) shows up, however, at the fifth DFT bin as a signal approximately 53.0 dB down. See Fig. 29.

The result of applying the Tukey family of windows is presented in Figs. 74-76. In Fig. 74 (the 25-percent taper) we see the lack of second-signal detection due to the high sidelobe structure of the dominant rectangle window. In Fig. 75 (the 50-percent taper) we observe a lack of second-signal detection, with the second signal actually filling in one of the nulls of the first signals' kernel. In Fig. 76 (the 76-percent taper) we witness a marginal detection in the still high sidelobes of the larger signal. This is still an unsatisfying window because of the artifacts.

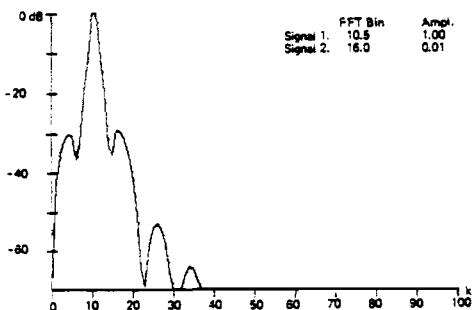


Fig. 74. Tukey (25-percent cosine taper) window.

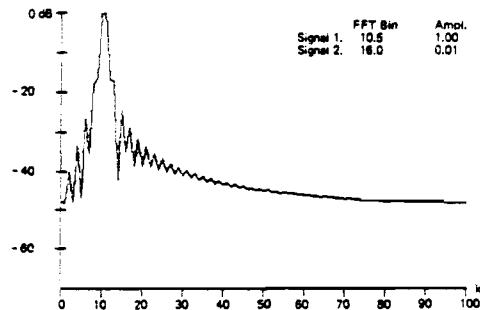
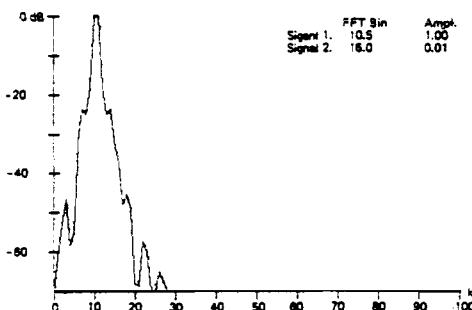
Fig. 78. Poisson window ($\alpha = 2.0$).

Fig. 75. Tukey (50-percent cosine taper) window.

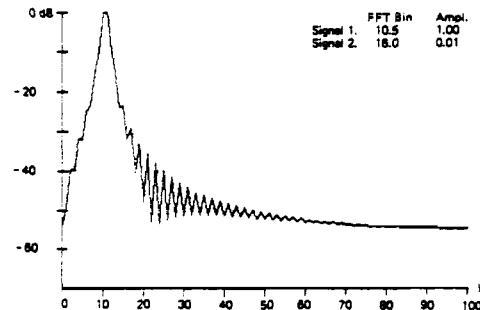
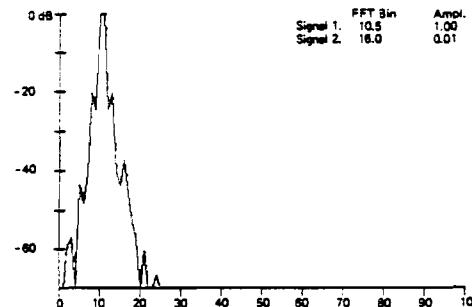
Fig. 79. Poisson window ($\alpha = 3.0$).

Fig. 76. Tukey (75-percent cosine taper) window.

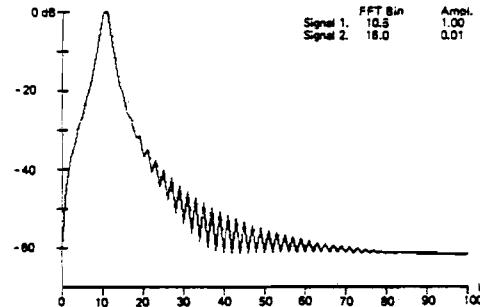
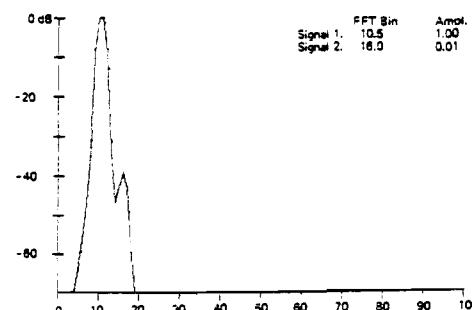
Fig. 80. Poisson window ($\alpha = 4.0$).

Fig. 77. Bohman window.

The Bohman construction window is applied and presented in Fig. 77. The second signal has been detected and the null between the two lobes is approximately 6.0 dB. This is not bad, but we can still do better. Note where the Bohman window resides in Fig. 12.

The result of applying the Poisson-window family is presented in Figs. 78-80. The second signal is not detected for any of the selected parameter values due to the high-sidelobe

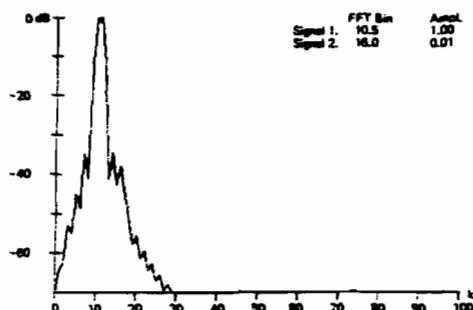
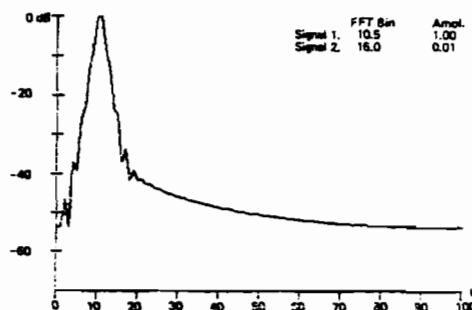
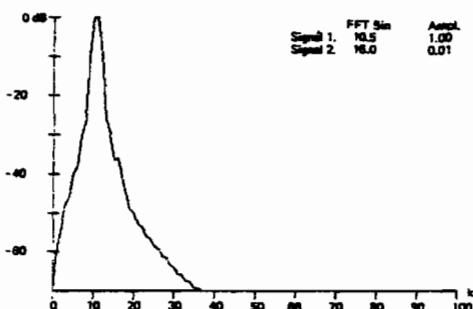
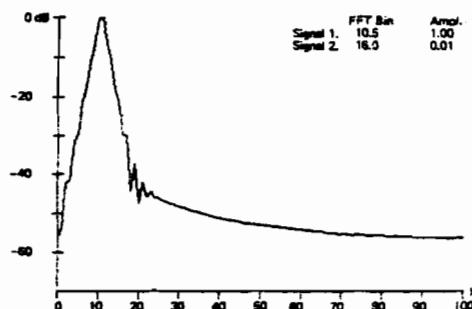
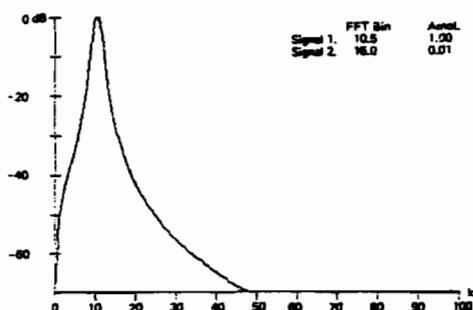
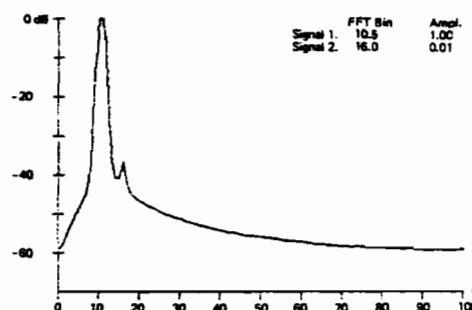
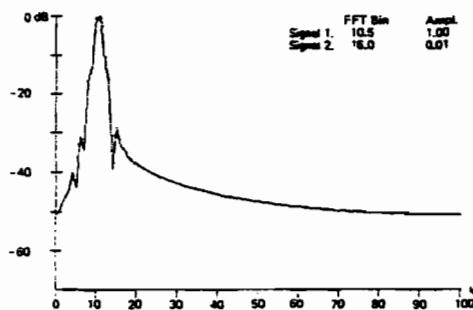
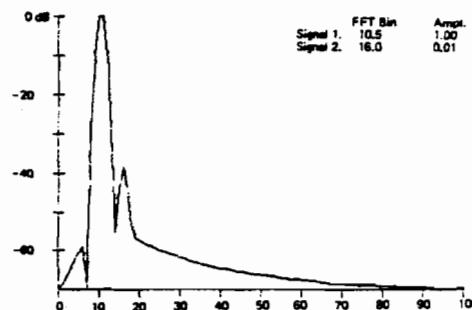
levels of the larger signal. We anticipated this poor performance in Table I by the large difference between the 3.0 dB and the ENBW.

The result of applying the Hanning-Poisson family of windows is presented in Figs. 81-83. Here, too, the second signal is either not detected in the presence of the high-sidelobe structure or the detection is bewildered by the artifacts.

The Cauchy-family windows have been applied and the results are presented in Figs. 84-86. Here too we have a lack of satisfactory detection of the second signal and the poor sidelobe response. This was predicted by the large difference between the 3.0 dB and the equivalent noise bandwidths as listed in Table I.

We now apply the Gaussian family of windows and present the results in Figs. 87-89. The second signal is detected in all three figures. We note as we further depress the sidelobe structure to enhance second-signal detection, the null deepens to approximately 16.0 dB and then becomes poorer as the main-lobe width increases and starts to overlap the lobe of the smaller signal.

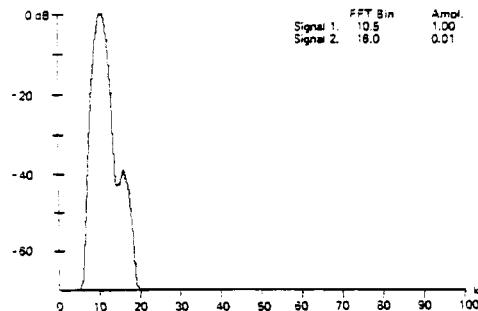
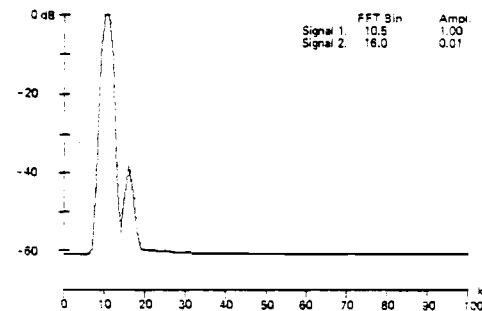
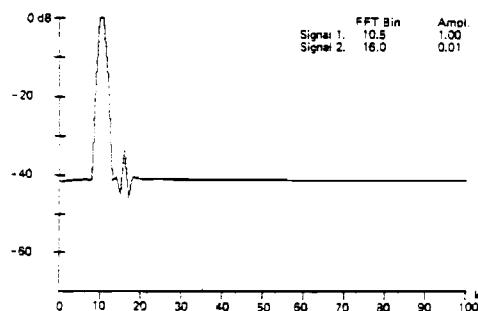
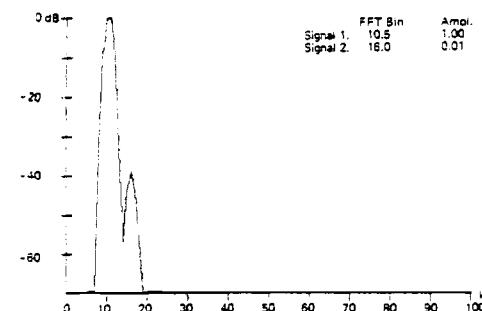
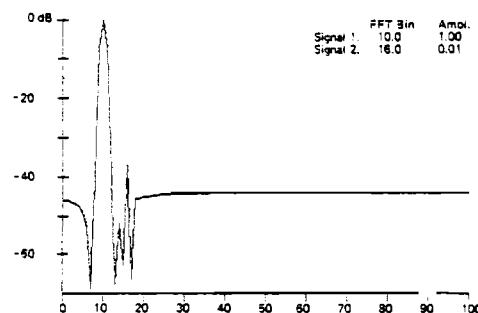
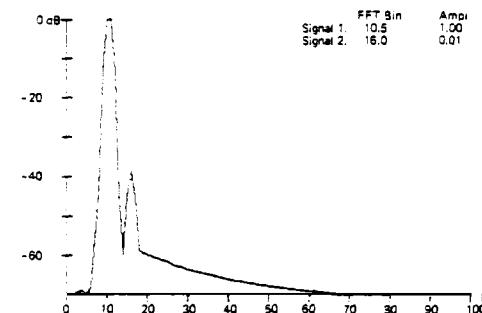
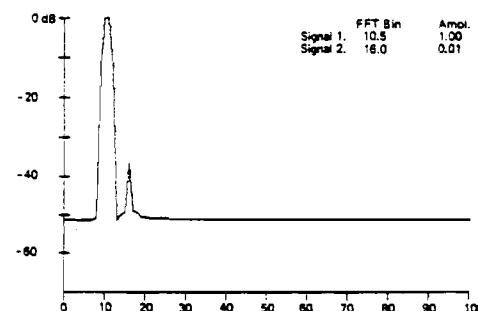
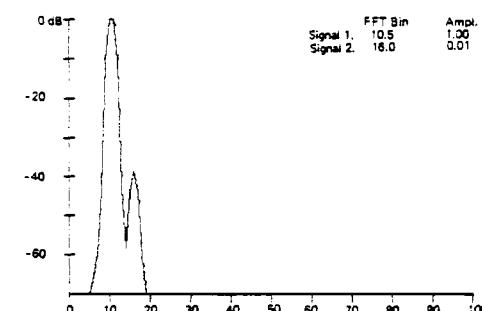
The Dolph-Chebyshev family of windows is presented in Figs. 90-94. We observe strong detection of the second signal

Fig. 81. Hanning-Poisson window ($\alpha = 0.5$).Fig. 85. Cauchy window ($\alpha = 4.0$).Fig. 82. Hanning-Poisson window ($\alpha = 1.0$).Fig. 86. Cauchy window ($\alpha = 5.0$).Fig. 83. Hanning-Poisson window ($\alpha = 2.0$).Fig. 87. Gaussian window ($\alpha = 2.5$).Fig. 84. Cauchy window ($\alpha = 3.0$).Fig. 88. Gaussian window ($\alpha = 3.0$).

in all cases, but it is distressing to see the uniformly high sidelobe structure. Here, we again see the coherent addition of the sidelobes from the positive and negative frequency kernels. Notice that the smaller signal is not 40-dB down now. What we are seeing is the scalloping loss of the large signals' main-lobe being sampled off of the peak and being referenced as zero dB. Figs. 90 and 91 demonstrate the sensitivity of the sidelobe coherent addition to main-lobe position. In Fig. 90 the larger signal is at bin 10.5; in Fig. 91 it is at bin 10.0.

Note the difference in phase cancellation near the base of the large signal. Fig. 93, the 70-dB-sidelobe window, exhibits an 18-dB null between the two main lobes but the sidelobes have added constructively (along with the scalloping loss) to the -62.0-dB level. In Fig. 94, we see the 80-dB sidelobe window exhibited sidelobes below the 70-dB level and still managed to hold the null between the two lobes to approximately 18.0 dB.

The Kaiser-Bessel family is presented in Figs. 95-98. Here,

Fig. 89. Gaussian window ($a = 3.5$).Fig. 93. Dolph-Chebyshev window ($a = 3.5$).Fig. 90. Dolph-Chebyshev window ($a = 2.5$).Fig. 94. Dolph-Chebyshev window ($a = 4.0$).Fig. 91. Dolph-Chebyshev window ($a = 2.5$).Fig. 95. Kaiser-Bessel window ($a = 2.0$).Fig. 92. Dolph-Chebyshev window ($a = 3.0$).Fig. 96. Kaiser-Bessel window ($a = 2.5$).

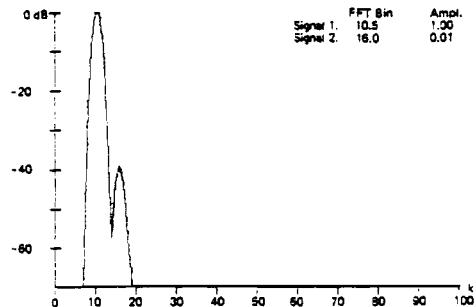
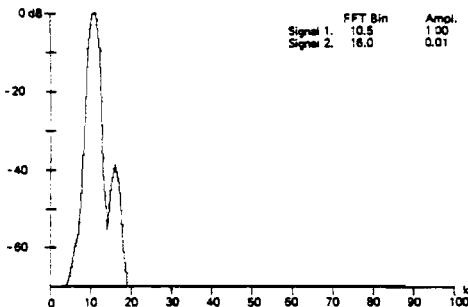
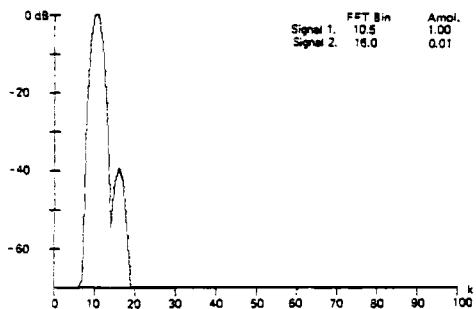
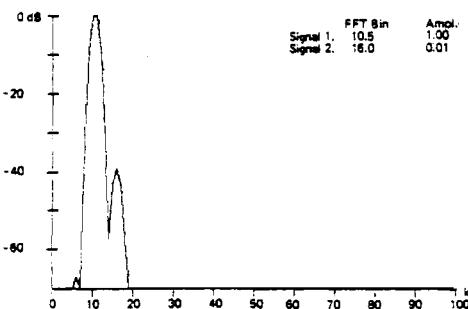
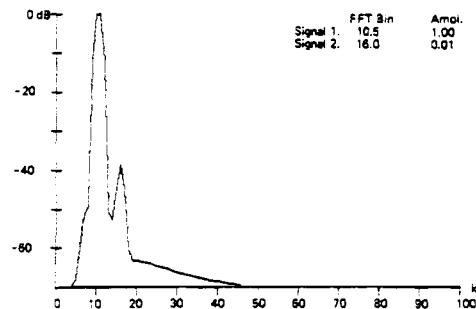
too, we have strong second-signal detection. Again, we see the effect of trading increased main-lobe width for decreased sidelobe level. The null between the two lobes reaches a maximum of 22.0 dB as the sidelobe structure falls and then becomes poorer with further sidelobe level improvement. Note that this window can maintain a 20.0-dB null between the two signal lobes and still hold the leakage to more than 70 dB down over the entire spectrum.

Figs. 99–101 present the performance of the Barcilon-Temes window. Note the strong detection of the second signal.

There are slight sidelobe artifacts. The window can maintain a 20.0-dB null between the two signal lobes. The performance of this window is slightly shy of that of the Kaiser-Bessel window, but the two are remarkably similar.

VII. CONCLUSIONS

We have examined some classic windows and some windows which satisfy some criteria of optimality. In particular, we have described their effects on the problem of general har-

Fig. 97. Kaiser-Bessel window ($a = 3.0$).Fig. 100. Barcilon-Temes window ($a = 3.5$).Fig. 98. Kaiser-Bessel window ($a = 3.5$).Fig. 101. Barcilon-Temes window ($a = 4.0$).Fig. 99. Barcilon-Temes window ($a = 3.0$).

monic analysis of tones in broadband noise and of tones in the presence of other tones. We have observed that when the DFT is used as a harmonic energy detector, the worst case processing loss due to the windows appears to be lower bounded by 3.0 dB and (for good windows) upper bounded near 3.75 dB. This suggests that the choice of particular windows has very little effect on worst case performance in DFT energy detection. We have concluded that a good performance indicator for the window is the difference between the equivalent noise bandwidth and the 3.0-dB bandwidth normalized by the 3.0-dB bandwidth. The windows which perform well (as indicated in Fig. 12) exhibit values for this ratio between 4.0 and 5.5 percent. The range of this ratio for the windows listed in Table I is 3.2 to 22.9 percent.

For multiple-tone detection via the DFT, the window employed does have a considerable effect. Maximum dynamic range of multitone detection requires the transform of the window to exhibit a highly concentrated central lobe with very-low sidelobe structure. We have demonstrated that many classic windows satisfy this criterion with varying

degrees of success and some not at all. We have demonstrated the optimal windows (Kaiser-Bessel, Dolph-Chebyshev, and Barcilon-Temes) and the Blackman-Harris windows perform best in detection of nearby tones of significantly different amplitudes. Also for the same dynamic range, the three optimal windows and the Blackman-Harris window are roughly equivalent with the Kaiser-Bessel and the Blackman-Harris, demonstrating minor performance advantages over the others. We note that while the Dolph-Chebyshev window appears to be the best window by virtue of its relative position in Fig. 12, the coherent addition of its constant-level sidelobes detracts from its performance in multi tone detection. Also the side-lobe structure of the Dolph-Chebyshev window exhibits extreme sensitivity to coefficient errors. This would affect its performance in machines operating with fixed-point arithmetic. This suggests that the Kaiser-Bessel or the Blackman-Harris window be declared the top performer. My preference is the Kaiser-Bessel window. Among other reasons, the coefficients are easy to generate and the trade-off of sidelobe level as a function of time-bandwidth product is fairly simple. For many applications, the author would recommend the 4-sample Blackman-Harris (or the 4-sample Kaiser-Bessel) window. These have the distinction of being defined by a few easily generated coefficients and of being able to be applied as a spectral convolution after the DFT.

We have called attention to a persistent error in the application of windows when performing convolution in the frequency domain, i.e., the omission of the alternating signs on the window sample spectrum to account for the shifted time origin. We have also identified and clarified a source of confusion concerning the evenness of windows under the DFT.

Finally, we comment that all of the conclusions presented about window performance in spectral analysis are also applicable to shading for array processing of spatial sampled data, including FFT beamforming.

APPENDIX
**THE EQUIVALENCE OF WINDOWING IN THE TIME
 DOMAIN TO CONVOLUTION IN THE FREQUENCY DOMAIN**

Let

$$f(t) = \int_{-\infty}^{+\infty} F(\omega) \exp(-j\omega t) d\omega / 2\pi$$

and

$$W(\omega) = \sum_{n=-N/2}^{+N/2} w(nT) \exp(+j\omega nT).$$

Then

$$F_w(\omega) = \sum_{n=-\infty}^{+\infty} w(nT) f(nT) \exp(+j\omega nT)$$

becomes

$$\begin{aligned} F_w(\omega) &= \sum_{n=-\infty}^{+\infty} w(nT) \int_{-\infty}^{+\infty} F(x) \exp(-jxnT) dx / 2\pi \\ &\quad \cdot \exp(+j\omega nT) \\ &= \int_{-\infty}^{+\infty} F(x) \sum_{n=-\infty}^{+\infty} w(nT) \exp(+j(\omega - x)nT) dx / 2\pi \\ &= \int_{-\infty}^{+\infty} F(x) \sum_{n=-N/2}^{+N/2} w(nT) \exp(+j(\omega - x)nT) dx / 2\pi \\ &= \int_{-\infty}^{+\infty} F(x) W(\omega - x) dx / 2\pi \end{aligned}$$

or

$$F_w(\omega) = F(\omega) * W(\omega).$$

REFERENCES

- [1] C. W. Helstrom, *Statistical Theory of Signal Detection*, 2nd ed. New York: Pergamon Press, 1968, Ch. IV, 4, pp. 124-130.
- [2] J. W. Cooley, P. A. Lewis, and P. D. Welch, "The finite Fourier transform," *IEEE Trans. Audio Electroacoust.*, vol. AU-17, pp. 77-85, June 1969.
- [3] J. W. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. New York: Wiley, 1965, ch. 4.3, pp. 223-228.

- [4] C. Lanczos, *Discourse on Fourier Series*. New York: Hafner Publishing Co., 1966, ch. 1, pp. 29-30.
- [5] P. D. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Trans. Audio Electroacoust.*, vol. AU-15, pp. 70-73, June 1967.
- [6] J. R. Rice, *The Approximation of Functions*, Vol. I. Reading, MA: Addison-Wesley, 1964, ch. 5.3, pp. 124-131.
- [7] R. B. Blackman and J. W. Tukey, *The Measurement of Power Spectra*. New York: Dover, 1958, appendix B.5, pp. 95-100.
- [8] L. Fejér, "Untersuchungen über Fouriersche Reihen," *Mat. Ann.*, 58, pp. 501-569, 1904.
- [9] L. R. Rabiner, B. Gold, and C. A. McGonegal, "An approach to the approximation problem for nonrecursive digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-18, pp. 83-106, June 1970.
- [10] F. J. Harris, "High-resolution spectral analysis with arbitrary spectral centers and adjustable spectral resolutions," *J. Comput. Elec. Eng.*, vol. 3, pp. 171-191, 1976.
- [11] E. Parzen, "Mathematical considerations in the estimation of spectra," *Technometrics*, vol. 3, no. 2, pp. 167-190, May 1961.
- [12] N. K. Bary, *A Treatise on Trigonometric Series*, Vol. I. New York: Macmillan, 1964, ch. I.53, pp. 149-150, ch. I.68, pp. 189-192.
- [13] J. W. Tukey, "An introduction to the calculations of numerical spectrum analysis," in *Spectral Analysis of Time Series*, B. Harris, Ed. New York: Wiley, 1967, pp. 25-46.
- [14] H. Bohman, "Approximate Fourier analysis of distribution functions," *Arkiv Före Matematik*, vol. 4, 1960, pp. 99-157.
- [15] N. I. Akhiezer, *Theory of Approximation*. New York: Ungar, 1956, ch. IV.64, pp. 118-120.
- [16] L. E. Franks, *Signal Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1969, ch. 6.1, pp. 136-137.
- [17] H. D. Helms, "Digital filters with equiripple or minimax responses," *IEEE Trans. Audio Electroacoust.*, vol. AU-19, pp. 87-94, Mar. 1971.
- [18] F. F. Kuo and J. F. Kaiser, *System Analysis by Digital Computer*. New York: Wiley, 1966, ch. 7, pp. 232-238.
- [19] D. Slepian and H. Pollak, "Prolate-spheroidal wave functions, Fourier analysis and uncertainty—I," *Bell Tel. Syst. J.*, vol. 40, pp. 43-64, Jan. 1961.
- [20] H. Landau and H. Pollak, "Prolate-spheroidal wave functions, Fourier analysis and uncertainty—II," *Bell Tel. Syst. J.*, vol. 40, pp. 65-84, Jan. 1961.
- [21] V. Barcilon and G. Temes, "Optimum impulse response and the Van Der Maas function," *IEEE Trans. Circuit Theory*, vol. CT-19, pp. 336-342, July 1972.

BIBLIOGRAPHY—ADDITIONAL GENERAL REFERENCES

- R. B. Blackman, *Data Smoothing and Prediction*. Reading, MA: Addison-Wesley, 1965.
- D. R. Brillinger, *Time Series Data Analysis and Theory*. New York: Holt, Rinehart, and Winston, 1975.
- D. Gingras, "Time series windows for improving discrete spectra estimation," Naval Undersea Research and Development Center, Rep. NUC TN-715, Apr. 1972.
- F. J. Harris, "Digital signal processing," Class notes, San Diego State Univ., 1971.
- G. M. Jenkins, "General considerations in the estimation of spectra," *Technometrics*, vol. 3, no. 2, pp. 133-166, May 1961.

