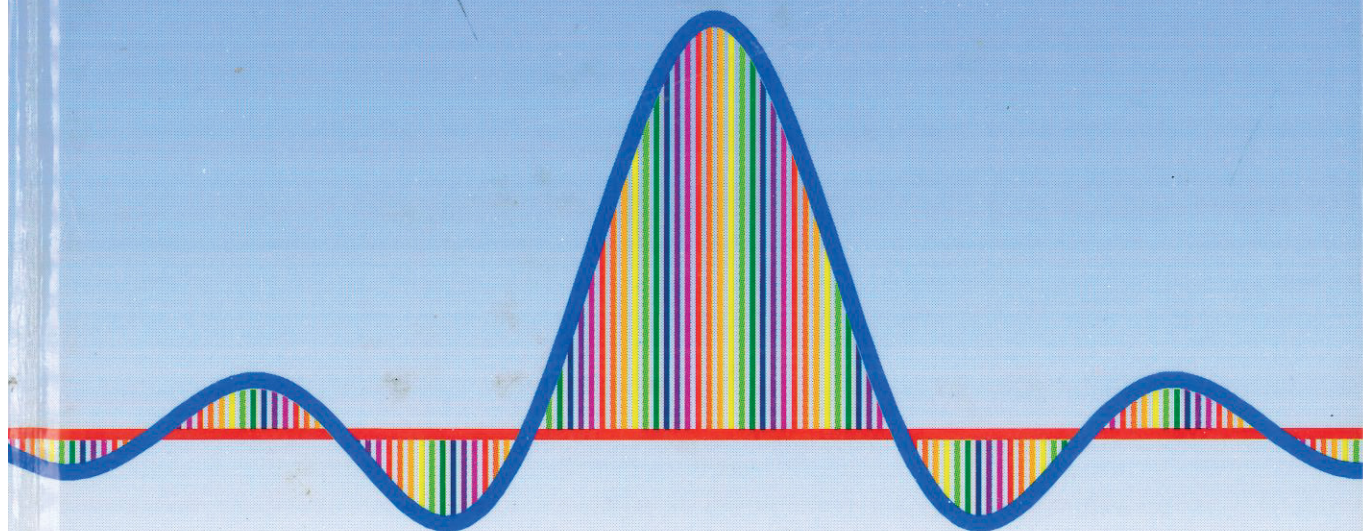


LINEAR SYSTEMS AND SIGNALS

B.P. LATHI



LINEAR SYSTEMS AND SIGNALS

LATHI

TK
5102.5
L29
1992

LINEAR SYSTEMS AND SIGNALS

Copyright ©1992 by Berkeley-Cambridge Press.

All rights reserved. Printed in the United States of America. No part of this book may be used or reproduced in any form or by any means without written permission, except in the case of brief quotations embodied in critical articles and reviews. For information address:

Berkeley-Cambridge Press
P.O. Box 947, Carmichael, CA. 95609-0947



Library of congress cataloging in publication data

Lathi, B. P. (Bhagawandas Pannalal)

Linear Systems and Signals.

Includes bibliographical references and index.

1. Signal processing—Mathematics. 2. System analysis
3. Linear time-invariant systems. 4. Digital filters (Mathematics)

I. Title.

TK5102.5.L29 1992 621.382'2—dc20 91-75492

ISBN 0-941413-34-9

Cover and Graphic Design: Mike Alie

engn TK
5102.5
L29
1992

APM 3600

Contents

Preface	xi
PART I: INTRODUCTION	
B Background	3*
B.1 Complex Numbers	3
B.1-1 A Historical Note	3
B.1-2 Algebra of Complex Numbers	6
B.2 Sinusoids	16
B.2-1 Addition of Sinusoids	19
B.2-2 Sinusoids in Terms of Exponentials: Euler's Formula	22
B.2-3 Generalized Sinusoids: Complex Frequency	23
B.3 Sketching Signals	25
B.3-1 Monotonic Exponentials	25
B.3-2 Exponentially Varying Sinusoid	26
B.3-3 Amplitude Modulated Sinusoid	28
B.4 Some Useful Signal Operations	29
B.4-1 Time-Shifting	29
B.4-2 Time-Scaling	30
B.4-3 Time-Inversion	32
B.4-4 Combined Operations	33
B.5 Mathematical Description of a Signal from Its Sketch	34
B.6 Even and Odd Functions	37
B.6-1 Some Properties of Even and Odd Functions	38
B.6-2 Even and Odd Components of a Signal	38
B.7 Cramer's Rule	39
B.8 Partial fraction expansion	41
B.8-1 Method of Clearing Fractions	42
B.8-2 Heaviside "Cover-up" Method	43
B.8-3 Mixture of Clearing Fractions and Heaviside "Cover-up"	48
B.8-4 Partial Fraction Expansion of Improper $F(x)$ with $m = n$	50
B.8-5 Modified Partial Fractions	51
B.9 Vectors and Matrices	51
B.9-1 Some Definitions and Properties	53
B.9-2 Matrix Algebra	54
B.9-3 Derivatives and Integrals of a Matrix	59

B.9-4	Characteristic Equation of a Matrix: Cayley-Hamilton Theorem	60
B.9-5	Computation of an Exponential and a Power of a Matrix	62
B.10	Miscellaneous	64
B.10-1	L'Hôpital's Rule	64
B.10-2	Taylor and Maclaurin Series	64
B.10-3	Power Series Expansion	64
B.10-4	Sums	64
B.10-5	Complex Numbers	65
B.10-6	Trigonometric Identities	65
B.10-7	Indefinite Integrals	66
B.10-8	Differentiation Table	67
B.10-9	Some Useful Constants	67
B.10-10	Solution of Quadratic and Cubic Equations	67
• 1	Introduction to Systems	69
1.1	Systems	69
1.2	Classification of Systems	72
1.2-1	Linear and Nonlinear Systems	72
1.2-2	Time-Invariant and Time-varying Parameter Systems	76
1.2-3	Instantaneous and Dynamic Systems	77
1.2-4	Causal and Noncausal Systems	77
1.2-5	Lumped-Parameter and Distributed Parameter Systems	80
1.2-6	Continuous-Time and Discrete-Time Systems	80
1.2-7	Analog and Digital Systems	81
1.3	System Model: Input-output Description	82
1.3-1	Electrical Systems	83
1.3-2	Mechanical Systems	85
1.3-3	Electromechanical Systems	89
1.4	Simultaneous Differential Equations	90
1.5	Internal and External Description of a System	92
1.5-1	Large Systems	95
1.6	State-Space Description of a System	96
1.7	Summary	98

PART II: TIME-DOMAIN ANALYSIS OF LTI SYSTEMS

• 2	Time-Domain Analysis: Continuous-Time Systems	105
2.1	Introduction	105
2.2	System Response to Internal Conditions: Zero-Input response	107
2.2-1	Some Insights Into the Zero-Input Behavior of a System	118
2.3	Unit Impulse Function	120
2.3-1	The Unit Impulse response $h(t)$	123
2.4	System Response to External Input: Zero-State Response	127
2.4-1	The Convolution Integral	130

2.4-2	Graphical Understanding of Convolution	138
2.4-3	Why Convolution? An Intuitive Explanation of System response	149
2.4-4	Some Reflections on the Use of Impulse Function	150
2.5	Numerical Convolution	150
2.6	System Stability	156
2.6-1	System Response to Bounded Inputs	159
2.7	Intuitive Insights In System Behavior	161
2.7-1	Dependence of System Behavior on Characteristic Modes	161
2.7-2	Response-Time of a System: The System Time-Constant	163
2.7-3	Time-Constant and Rise-Time of a System	164
2.7-4	Time-Constant and Filtering	165
2.7-5	Time-Constant and Pulse Dispersion (Spreading)	166
2.7-6	Time-Constant and Rate of Information transmission	167
2.7-7	Resonance Phenomenon	167
2.8	Classical Solution of Differential equations	170
2.8-1	Forced response: Method of Undetermined Coefficients	170
2.9	Appendix 2.1: Determining The Impulse Response	177
2.10	Summary	179

• 3 Time-Domain Analysis: Discrete-Time Systems 189

3.1	Discrete-Time Systems	189
3.2	Discrete-Time System Equations	195
3.2-1	Initial Conditions and Iterative solution of Difference Equations	196
3.3	System Response to Internal Conditions: Zero-Input Response	200
3.4	Unit Impulse Response $h[k]$	208
3.4-1	Determining Impulse Response $h[k]$	209
3.5	System Response to External Input: Zero-State Response	213
3.5-1	Graphical Procedure for Convolution Sum	220
3.6	System Stability	223
3.6-1	System Response to Bounded Inputs	229
3.7	Intuitive insights In System Behavior	230
3.8	Classical Solution of Linear Difference Equations	230
3.9	Appendix 2.1: Determining Impulse Response	235
3.10	Summary	237

PART III: FREQUENCY-DOMAIN (TRANSFORM) ANALYSIS OF LTI SYSTEMS

4 Continuous-Time Systems: Laplace Transform Analysis 249

4.1	The Laplace Transform	249
4.1-1	Finding Inverse Transform	255
4.2	Some Properties of the Laplace Transform	265
4.3	Transform Solution of Differential and Integro-Differential Equations	274
4.3-1	The Zero-State Response of LTIC Systems: The Transfer Function	278

4.3-2	Relationship Between $h(t)$ and $H(s)$	281
4.3-3	Frequency-Domain Interpretation of the Transfer function	282
4.3-4	Physical Interpretation of the Laplace Transform	284
4.4	Analysis of Electrical Networks: Transformed Network Method	285
4.4-1	Analysis of Active Circuits	294
4.4-2	Initial and Final Values	297
4.5	Block Diagrams	298
4.6	System Realization	303
4.6-1	Canonical or Direct-Form Realization	304
4.6-2	Cascade and Parallel Realizations	309
4.6-3	System Realization Using Operational Amplifiers	312
4.7	Frequency-Response of an LTIC System	315
4.7-1	Dependence of Frequency Response on Poles and Zeros of $H(s)$	322
4.8	Bilateral Laplace Transform	329
4.8-1	Linear System Analysis Using Bilateral Transform	335
4.9	Appendix 4.1: Second Canonical Realization	337
4.10	Summary	339
5	Discrete-Time Systems: z-Transform Analysis	353
5.1	The z-Transform	353
5.1-1	Finding the Inverse Transform	357
5.2	Some Properties of the z-Transform	363
5.3	z-Transform solution of Difference equations	368
5.3-1	Zero-State Response of LTID Systems: The Transfer Function	372
5.3-2	Relationship Between $h[k]$ and $H[z]$	375
5.3-3	Frequency-Domain Interpretation of the Transfer Function	376
5.3-4	Physical Interpretation of the z -Transform	376
5.4	System Realization	377
5.5	Frequency Response of Discrete-Time Systems	380
5.5-1	Periodic Nature of Frequency Response	385
5.5-2	Aliasing and Sampling Rate	386
5.5-3	Dependence of Frequency Response on Poles and Zeros of $H[z]$	388
5.6	Connection Between the z-Transform and Laplace Transform	396
5.7	Bilateral z-Transform	399
5.7-1	Analysis of LTID Systems Using Bilateral z -Transform	402
5.8	Summary	404

PART IV: SIGNAL ANALYSIS

6	Continuous-Time Signal Analysis: The Fourier Series	415
6.1	Representation of Periodic Signals by Trigonometric Fourier Series	415
6.1-1	The Fourier Spectrum	421
6.1-2	Effect of Symmetry	429
6.1-3	Determining the Fundamental Frequency and Period	431

6.1-4	Role of Amplitude and Phase Spectra in Wave Shaping	432
6.2	Exponential Fourier Series	438
6.2-1	Exponential Fourier Spectra	442
6.2-2	Why Use exponentials?	448
6.3	Alternate View of Fourier Representation: Signal-Vector Analogy	449
6.3-1	Vector Comparison	449
6.3-2	Signal Comparison	451
6.3-3	Orthogonal Vector Space	453
6.3-4	Orthogonal Signal Space	455
6.3-5	Gibbs Phenomenon	461
6.3-6	Some Examples of Orthogonal functions	464
6.4	Summary	467
7	Continuous-Time Signal Analysis: The Fourier Transform	474
7.1	Nonperiodic Signal Representation By Fourier Integral	474
7.2	Physical Appreciation of the Fourier Transform	480
7.2-1	Connection Between the Fourier and Laplace Transforms	482
7.3	Transform of Some Useful Functions	484
7.4	Some Properties of the Fourier Transform	490
7.4-1	Symmetry of the Direct and Inverse Transform Operation: Time-Frequency Duality	492
7.4-2	Duality Property	493
7.4-3	Scaling Property	494
7.4-4	Time-Shifting Property	496
7.4-5	Frequency-shifting Property	498
7.4-6	Convolution	502
7.4-7	Time-Differentiation and Integration	503
7.5	LTI System Analysis by Fourier Transform	506
7.6	Signal Distortion During Transmission	508
7.7	Ideal and Practical Filters	511
7.8	Thinking in Time-Domain and Frequency-Domain: Two-Dimensional View of Signals and Systems	514
7.9	Signal Energy	515
7.10	Data Truncation: Window Functions	518
7.11	Summary	522
8	Sampling	532
8.1	The Sampling Theorem	532
8.1-1	Practical Difficulties in Signal Reconstruction	534
8.1-2	Some Applications of the Sampling Theorem	538
8.2	Dual of the Sampling Theorem: Spectral Sampling	542
8.3	Numerical Computation of Fourier Transform: The Discrete Fourier Transform (DFT)	543
8.3-1	Some Properties of DFT	552

8.3-2 Fast Fourier Transform (FFT)	557
8.4 Summary	559
9 Analysis of Discrete-Time Signals	565
9.1 Discrete-Time Periodic Signals	565
9.1-1 Representation of Periodic Signals: Discrete-Time Fourier Series	566
9.2 Nonperiodic Signals: The Discrete-Time Fourier Transform (DTFT)	574
9.2-1 Relationship of the DTFT to Z -transform	581
9.3 Properties of DTFT	582
9.4 LTID System analysis by DTFT	583
9.5 Relationships Among Various Transforms	585
9.6 Derivation of the Z -Transform Pair	588
9.7 Summary	589
 PART V: STATE-SPACE ANALYSIS	
10 State-Space Analysis	595
10.1 Introduction	595
10.2 Systematic Procedure for Determining State Equations	598
10.2-1 Electrical Circuits	599
10.2-2 State Equations From Transfer Function	601
10.3 Solution of State Equations	610
10.3-1 Laplace Transform Solution of State Equations	610
10.3-2 Time-Domain solution of State Equations	615
10.4 Linear Transformation of State Vector	622
10.4-1 Diagonalization of Matrix A	626
10.5 Controllability and Observability	630
10.5-1 Inadequacy of transfer function Description	635
10.6 State-Space Analysis of Discrete-Time Systems	636
10.6-1 Solution in State-Space	638
10.6-2 Z -Transform Solution	641
10.7 Summary	642
 Supplementary Reading	 649
 Index	 651

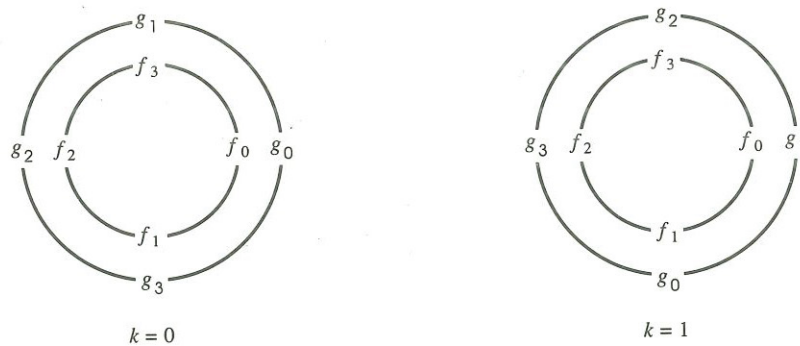


Fig. 8.12 Graphical picture of cyclic convolution.

Applications of DFT

The DFT is useful not only in the computation of direct and inverse Fourier transforms, but also in other applications such as convolution, correlation, and filtering. Use of the efficient FFT algorithm discussed in Sec. 8.3-2 makes it particularly appealing.

Convolution

Let $f(t)$ and $g(t)$ be the two signals to be convolved. In general, these signals may have different time duration. To convolve them by using their samples, they must be sampled at the same rate (not below the Nyquist rate of either signal). If f_k ($0 \leq k \leq N_1 - 1$) and g_k ($0 \leq k \leq N_2 - 1$) are the corresponding discrete sequences representing these samples, and if

$$c(t) = f(t) * g(t)$$

then†

$$c_k = f_k * g_k$$

in which c_k is the discrete sequence corresponding to $c(t)$; that is, $c_k = Tc(kT)$.

Because of the width property of the convolution, c_k exists for $0 \leq k \leq N_1 + N_2 - 1$. To be able to use the DFT technique of cyclic convolution, we must make sure that the cyclic convolution will yield the same result as the linear convolution. This means that the resulting signal of the cyclic convolution must have the same length ($N_1 + N_2 - 1$) as that of the signal resulting from linear convolution. This can be brought about by adding $N_2 - 1$ dummy samples of zero value to f_k and $N_1 - 1$ dummy samples of zero value to g_k (zero padding). This changes the length of both f_k and g_k to be $N_1 + N_2 - 1$. We can now use the FFT to find the convolution $f_k * g_k$ in three steps, as follows:

1. Find the DFTs F_r and G_r corresponding to f_k and g_k .
2. Multiply F_r by G_r .
3. Find the IDFT of $F_r G_r$. In Chapter 2 we computed $f(t) * h(t)$ numerically for signals, shown in Fig. 2.24a and 2.24b. The reader is encouraged to perform this convolution using the DFT (or FFT).

†This follows from the fact that $f_k = Tf(kT)$, $g_k = Tg(kT)$, and $c_k = Tc(kT)$. From Eq. (2.79), $c(kT) = Tf(kT) * Tg(kT)$. Therefore $c_k = f_k * g_k$.

Filtering

We generally think of filtering in terms of some hardware-oriented solution (namely, building a circuit with *RLC* components and op amps). However, filtering also has a software-oriented solution [a computer algorithm that yields the filtered output $y(t)$ for a given input $f(t)$]. This can be conveniently accomplished by using the DFT. If $f(t)$ is the signal to be filtered, then F_r , the DFT of f_k , is found. The spectrum F_r is then shaped (filtered) as desired by multiplying F_r by H_r , where H_r are the samples of the filter transfer function $H(\omega)$ [$H_r = H(r\omega_0)$]. Finally, we take the IDFT of $F_r H_r$ to obtain the filtered output y_k [$y_k = \frac{T_0}{N_0} y(kT)$].

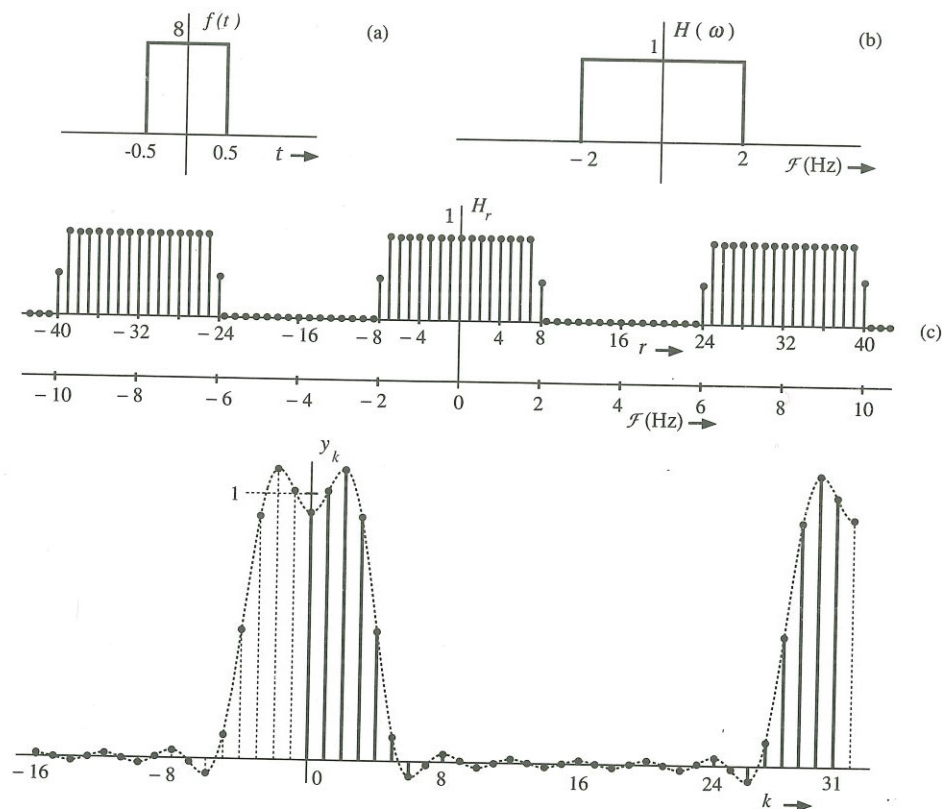


Fig. 8.13 Filtering $f(t)$ through $H(\omega)$.

Consider, for example, the problem of filtering the gate function $f(t)$ in Fig. 8.13a through an ideal lowpass filter $H(\omega)$, as shown in Fig. 8.13b. We have already found the 32 point DFT of $f(t)$ (see Fig. 8.11d). Next we multiply F_r by H_r . To compute H_r , we remember that in computing the 32 point DFT of $f(t)$, we have used $\mathcal{F}_0 = \frac{1}{4}$. Because F_r is 32-periodic, H_r must also be 32-periodic with samples separated by $\frac{1}{4}$ Hz. This means that H_r must be repeated every 8 Hz or 16π rad/sec (see Fig. 8.13c). This gives the 32 samples of H_r over $(0 \leq \omega \leq 16\pi)$ as follows:

No.	fk	Fr	Hr	FrHr	yk
0	1.0	8.000 E0	1.0	8.000 E0	9.285 E-1
1	1.0	7.179 E0	1.0	7.179 E0	1.009 E0
2	1.0	5.027 E0	1.0	5.027 E0	1.090 E0
3	1.0	2.331 E0	1.0	2.331 E0	9.123 E-1
4	0.5	0.000 E0	1.0	0.000 E0	4.847 E-1
5	0.0	-1.323 E0	1.0	-1.323 E0	8.884 E-2
6	0.0	-1.497 E0	1.0	-1.497 E0	-5.698 E-2
7	0.0	-8.616 E-1	1.0	-8.616 E-1	-1.383 E-2
8	0.0	0.000 E0	0.5	0.000 E0	2.933 E-2
9	0.0	5.803 E-1	0.0	0.000 E0	4.837 E-3
10	0.0	6.682 E-1	0.0	0.000 E0	-1.966 E-2
11	0.0	3.778 E-1	0.0	0.000 E0	-2.156 E-3
12	0.0	0.000 E0	0.0	0.000 E0	1.534 E-2
13	0.0	-2.145 E-1	0.0	0.000 E0	9.828 E-4
14	0.0	-1.989 E-1	0.0	0.000 E0	-1.338 E-2
15	0.0	-6.964 E-2	0.0	0.000 E0	-2.876 E-4
16	0.0	0.000 E0	0.0	0.000 E0	1.280 E-2
17	0.0	-6.964 E-2	0.0	0.000 E0	-2.876 E-4
18	0.0	-1.989 E-1	0.0	0.000 E0	-1.338 E-2
19	0.0	-2.145 E-1	0.0	0.000 E0	9.828 E-4
20	0.0	0.000 E0	0.0	0.000 E0	1.534 E-2
21	0.0	3.778 E-1	0.0	0.000 E0	-2.156 E-3
22	0.0	6.682 E-1	0.0	0.000 E0	-1.966 E-2
23	0.0	5.803 E-1	0.0	0.000 E0	4.837 E-3
24	0.0	0.000 E0	0.5	0.000 E0	2.933 E-2
25	0.0	-8.616 E-1	1.0	-8.616 E-1	-1.383 E-2
26	0.0	-1.497 E0	1.0	-1.497 E0	-5.698 E-2
27	0.0	-1.323 E0	1.0	-1.323 E0	8.884 E-2
28	0.5	0.000 E0	1.0	0.000 E0	4.847 E-1
29	1.0	2.331 E0	1.0	2.331 E0	9.123 E-1
30	1.0	5.027 E0	1.0	5.027 E0	1.090 E0
31	1.0	7.179 E0	1.0	7.179 E0	1.009 E0

Table 8.1

$$H_r = \begin{cases} 1 & 0 \leq r \leq 7 \text{ and } 25 \leq r \leq 31 \\ 0 & 9 \leq r \leq 23 \\ 0.5 & r = 8, 24 \end{cases}$$

We multiply F_r by H_r and take the IDFT. The resulting output signal is shown in Fig. 8.13d. Table 8.1 shows a printout of f_k , F_r , H_r , Y_r , and y_k .

⊙ Computer Example C8.2

Filtering can be done on a computer in software, much like it can be done in hardware (e.g. using op-amps, resistors, and capacitors). However, in software is often much easier to implement elaborate filters, as we only need to program their equations. We shall look filtering a square wave. After we compute the DFT, we will filter it with an ideal low pass filter. We do this by multiplying F_r by H_r , where H_r is the discrete Fourier transform of the ideal low pass filter.

```
% This routine generates a square wave as a Fourier series.
ftt=0; % Initialize sum of harmonics to zero.
ft=1/2; % Input dc term.
N = 50; % Select the number of sinusoids to use in the sum.
T = 0.1; % Select a sampling rate.
% This loop calculates the Fourier series for a square wave.
for t=0:256
    for n=0:N
        ftn=(-1)^n*(1/(2*n+1))*cos((2*n+1)*T*t);
        ftt=ftt+ftn;
    end
    Ftn=(2/pi)*ftt;
```

```

ft=[ft Ftn]; % This is the approximation of a square wave at each time.
ftt=0;      % Reset the sum of the harmonics.
end
t=0:1:30;   % Recreate the time base.
plot(t(1:length(ft)),ft)
Fr=fft(ft);
f = (1/T)/256*(0:127); % Create a frequency axis.
w = 2*pi*f; % Create a radian frequency axis.
magFr=abs(Fr); % Calculate the magnitude of FFT.
plot(f,magFr(1:128)) % Plot the FFT.
% This section defines the ideal low pass filter.
Hr=[ones((1/6)*length(Fr),1);zeros((5/6)*length(Fr),1)];
plot(f,Hr(1:128))
% This section multiplies the frequency responses of the signal and the
% filter.
Yr=Fr.*Hr'; % This is the FFT of the output of the filter.
plot(f,abs(Yr(1:128)))
% This section converts the output into time domain.
y=ifft(Yr); % This is the time domain output of the filter.
plot(t(1:length(y)),y) % Plot the filter output.
end

```

8.3-2 The Fast Fourier Transform (FFT)

The number of computations required in performing the DFT was dramatically reduced by an algorithm developed by Tukey and Cooley in 1965.⁴ This algorithm, known as the **Fast Fourier transform** (FFT), reduces the number of computations from something on the order of N_0^2 to $N_0 \log N_0$. To compute one sample F_r from Eq. (8.12a), we require N_0 complex multiplications and $N_0 - 1$ complex additions. To compute N_0 such values (F_r for $r = 0, 1, \dots, N_0 - 1$), we require a total of N_0^2 complex multiplications and $N_0(N_0 - 1)$ complex additions. For a large N_0 , this can be prohibitively time-consuming, even for a very high-speed computer.

Although there are many variations of the original Tukey-Cooley algorithm, these can be grouped into two basic types: **decimation-in-time** and **decimation-in-frequency**. The algorithm is simplified if we choose N_0 to be a power of 2, although this is not essential. To simplify the notation, let

$$W_{N_0} = e^{-j2\pi/N_0} = e^{-j\Omega_0} \quad (8.28)$$

so that

$$F_r = \sum_{k=0}^{N_0-1} f_k W_{N_0}^{kr} \quad 0 \leq r \leq N_0 - 1 \quad (8.29a)$$

and

$$f_k = \frac{1}{N_0} \sum_{r=0}^{N_0-1} F_r W_{N_0}^{-kr} \quad 0 \leq k \leq N_0 - 1 \quad (8.29b)$$

The Decimation-in-Time Algorithm

Here we divide the N_0 -point data sequence f_k into two $(\frac{N_0}{2})$ -point sequences consisting of even- and odd-numbered samples respectively, as follows:

$$\underbrace{f_0, f_2, f_4, \dots, f_{N_0-2}}_{\text{sequence } g_k} \quad \underbrace{f_1, f_3, f_5, \dots, f_{N_0-1}}_{\text{sequence } h_k}$$

Then from Eq. (8.29a),

$$F_r = \sum_{k=0}^{\frac{N_0}{2}-1} f_{2k} W_{N_0}^{2kr} + \sum_{k=0}^{\frac{N_0}{2}-1} f_{2k+1} W_{N_0}^{(2k+1)r} \quad (8.30)$$

Also, since

$$W_{\frac{N_0}{2}} = W_{N_0}^2 \quad (8.31)$$

we have

$$\begin{aligned} F_r &= \sum_{k=0}^{\frac{N_0}{2}-1} f_{2k} W_{\frac{N_0}{2}}^{kr} + W_{N_0}^r \sum_{k=0}^{\frac{N_0}{2}-1} f_{2k+1} W_{\frac{N_0}{2}}^{kr} \\ &= G_r + W_{N_0}^r H_r \quad 0 \leq r \leq N_0 - 1 \end{aligned} \quad (8.32)$$

in which G_r and H_r are the $(\frac{N_0}{2})$ -point DFTs of the even- and odd-numbered sequences, g_k and h_k , respectively. Also, G_r and H_r , being the $(\frac{N_0}{2})$ -point DFTs, are $(\frac{N_0}{2})$ -periodic. Hence

$$\begin{aligned} G_{r+(\frac{N_0}{2})} &= G_r \\ H_{r+(\frac{N_0}{2})} &= H_r \end{aligned} \quad (8.33)$$

Moreover,

$$W_{N_0}^{r+(\frac{N_0}{2})} = W_{\frac{N_0}{2}}^r W_{N_0}^r = e^{-j\pi} W_{N_0}^r = -W_{N_0}^r \quad (8.34)$$

From Eqs. (8.32), (8.33), and (8.34), we obtain

$$F_{r+(\frac{N_0}{2})} = G_r - W_{N_0}^r H_r \quad (8.35)$$

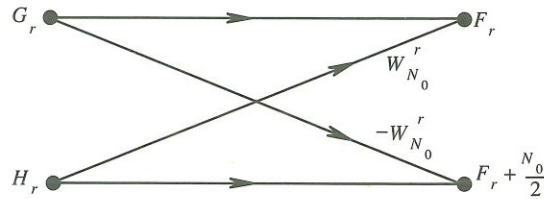


Fig. 8.14 Butterfly.

This property can be used to reduce the number of computations. We can, for example, compute F_r using Eq. (8.32) for the first $(\frac{N_0}{2})$ points ($0 \leq n \leq \frac{N_0}{2} - 1$) and using Eq. (8.35) for the last $\frac{N_0}{2}$ points:

$$F_r = G_r + W_{N_0}^r H_r \quad 0 \leq r \leq \frac{N_0}{2} - 1 \quad (8.36a)$$

$$F_{r+(\frac{N_0}{2})} = G_r - W_{N_0}^r H_r \quad 0 \leq r \leq \frac{N_0}{2} - 1 \quad (8.36b)$$

Thus an N_0 -point DFT can be computed by combining the two $(\frac{N_0}{2})$ -point DFTs, as in Eq. (8.36). These equations can be represented conveniently by the **signal flow** graph shown in Fig. 8.14. This structure is known as a **butterfly**. Figure 8.15a shows the implementation of Eqs. (8.33) for the case of $N_0 = 8$.

The next step is to compute the $(\frac{N_0}{2})$ -point DFTs G_r and H_r . We repeat the same procedure by dividing g_k and h_k into two $(\frac{N_0}{4})$ -point sequences corresponding to the even- and odd-numbered samples. Then we continue this process until we reach the one-point DFT. Figures 8.15a, 8.15b, and 8.15c show these steps for the case of $N_0 = 8$. Figure 8.15c shows that the two-point DFTs require no multiplication.

To count the number of computations required in the first step, assume that G_r and H_r are known. From Eqs. (8.36), it is clear that to compute all the N_0 points of the F_r , we require N_0 complex additions and $\frac{N_0}{2}$ complex multiplications† (corresponding to $W_{N_0}^r H_r$).

In the second step, to compute the $(\frac{N_0}{2})$ -point DFT G_r from the $(\frac{N_0}{4})$ -point DFT, we require $\frac{N_0}{2}$ complex additions and $\frac{N_0}{4}$ complex multiplications. We require an equal number of computations for H_r . Hence, in the second step, there are N_0 complex additions and $\frac{N_0}{2}$ complex multiplications. The number of computations required remains the same in each step. Since a total of $\log_2 N_0$ steps is needed to arrive at a one-point DFT, we require, conservatively, a total of $N_0 \log_2 N_0$ complex additions and $(\frac{N_0}{2}) \log_2 N_0$ complex multiplications, to compute the N_0 -point DFT.

The procedure for obtaining IDFT is identical to that used to obtain the DFT except that $W_{N_0} = e^{j(2\pi/N_0)}$ instead of $e^{-j(2\pi/N_0)}$ (in addition to the multiplier $1/N_0$). Another FFT algorithm, the **decimation-in-frequency** algorithm, is similar to the decimation-in-time algorithm. The only difference is that instead of dividing f_k into two sequences of even- and odd-numbered samples, we divide f_k into two sequences formed by the first $\frac{N_0}{2}$ and the last $\frac{N_0}{2}$ digits, proceeding in the same way until a single-point DFT is reached in $\log_2 N_0$ steps. The total number of computations in this algorithm is the same as that in the decimation-in-time algorithm.

8.4 SUMMARY

A signal bandlimited to B Hz can be reconstructed exactly from its samples if the sampling rate $R > 2B$ Hz (the sampling theorem). Such a reconstruction,

†Actually, $\frac{N_0}{2}$ is a conservative figure because some multiplications corresponding to the cases in which $W_{N_0}^r = 1, j$, etc., are eliminated.

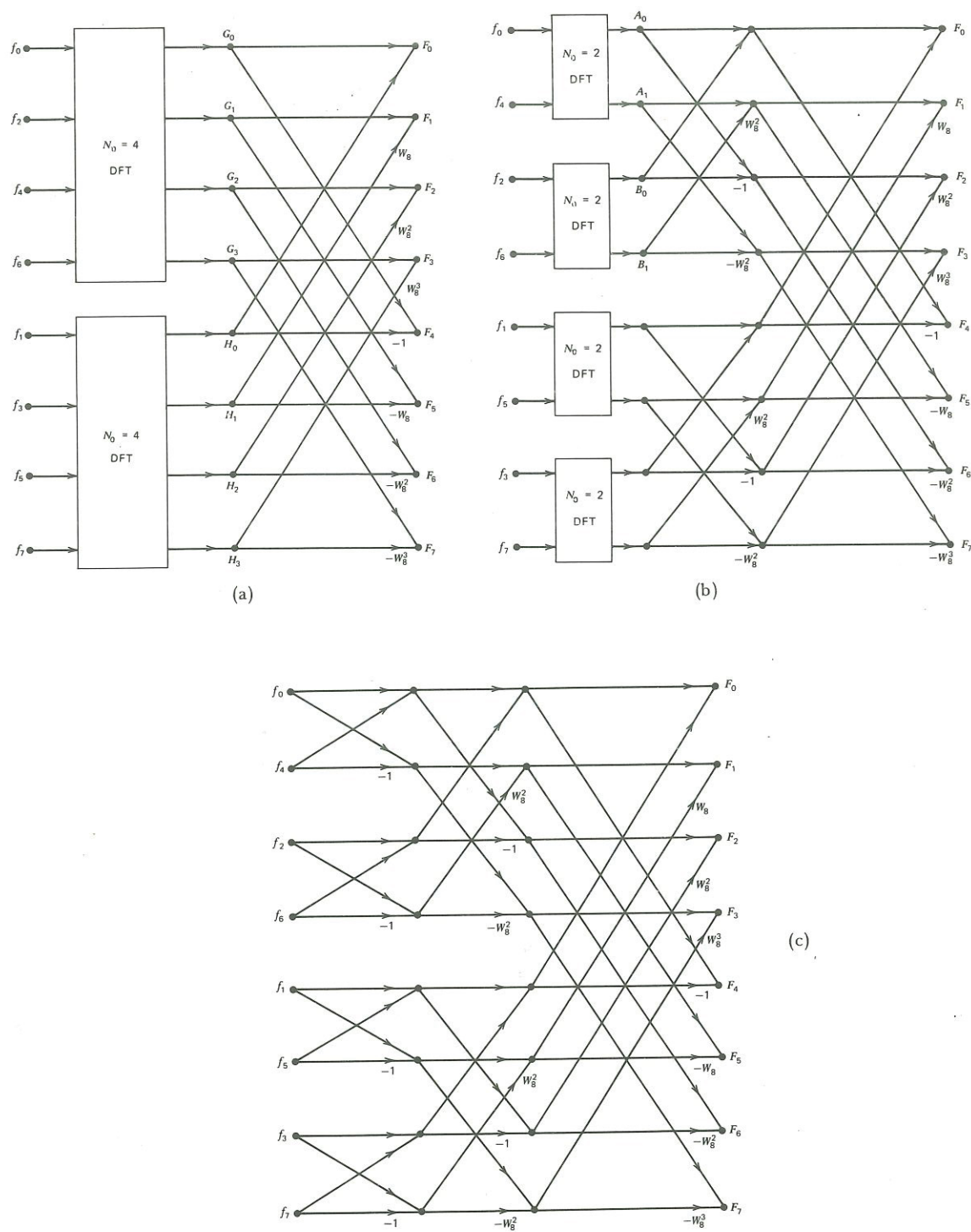


Fig. 8.15 Successive steps in 8-point FFT.

although possible theoretically, poses practical problems such as the need for filters with zero gain over a band (or bands) of frequencies. Such filters are unrealizable, or realizable with infinite time delay. Therefore, in practice, there is always an error in reconstructing a signal from its samples. Moreover, practical signals are not bandlimited, which causes an additional error (aliasing error) in signal reconstruction from its samples. Aliasing error can be eliminated by bandlimiting a signal to its effective bandwidth.

The sampling theorem is very important in signal analysis, processing, and transmission because it allows us to replace a continuous-time signal by a discrete sequence of numbers. Processing a continuous-time signal is therefore equivalent to processing a discrete sequence of numbers. This leads us directly into the area of digital filtering (discrete-time systems). In the field of communication, the transmission of a continuous-time message reduces to the transmission of a sequence of numbers. This opens doors to many new techniques of communicating continuous-time signals by pulse trains.

The dual of the sampling theorem states that for a signal time-limited to τ seconds, its spectrum $F(\omega)$ can be reconstructed from the samples of $F(\omega)$ taken at a uniform intervals not greater than $1/\tau$ Hz. In other words the spectrum should be sampled at a rate not less than τ samples/Hz.

To compute the direct or inverse Fourier transform numerically, we need relationship between the samples of $f(t)$ and $F(\omega)$. The sampling theorem and its dual provide such a quantitative relationship in the form of discrete Fourier transform (DFT). The DFT computations are greatly facilitated by fast Fourier transform (FFT) algorithm, which reduces the number of computations from something of the order of N_0^2 to $N_0 \log N_0$.

REFERENCES

1. Linden, D.A., "A Discussion of Sampling Theorem," *Proc. IRE*, vol. 47, pp 1219-1226, July 1959.
2. Bracewell, R.N., *The Fourier Transform and Its Applications*, 2nd revised ed., McGraw-Hill, New York, 1986.
3. Bennett, W.R., *Introduction to Signal Transmission*, McGraw-Hill, New York, 1970.
4. Cooley, J.W., and J.W., Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series", *Mathematics of Computation*, Vol. 19, pp 297-301, April 1965.

PROBLEMS

- 8.1-1 Figure P8.1-1 shows Fourier spectra of signals $f_1(t)$ and $f_2(t)$. Determine the Nyquist sampling rates for signals $f_1(t)$, $f_2(t)$, $f_1^2(t)$, $f_2^2(t)$, and $f_1(t)f_2(t)$.
Hint: Use the frequency convolution and the width property of the convolution.
- 8.1-2 Determine the Nyquist sampling rate and the Nyquist sampling interval for the signals
(a) $\text{sinc}^2(100\pi t)$ (b) $\text{sinc}(100\pi t) + 3 \text{sinc}^2(60\pi t)$ (c) $\text{sinc}(50\pi t)\text{sinc}(100\pi t)$.
- 8.1-3 Prove that a signal cannot be simultaneously time-limited and bandlimited.

89035671833



b89035671833a

ISBN 0-941413-34-9