MATLAB PROGRAMS

MATLAB Programs

12.1 INTRODUCTION

MATLAB is an interactive, matrix-based system for scientific and engineering numeric computation and visualization. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notations. MATLAB has extensive facilities for displaying vectors and matrices as graphs as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. The name MATLAB stands for MATrix LABoratory since its basic data element is a matrix (array).

12.2 MATLAB VARIABLES—SCALARS, VECTORS, AND MATRICES

MATLAB stores variables in the form of matrices which are MN, where M is the number of rows and N the number of columns. A 1×1 matrix is a scalar, a $1 \times N$ matrix is a row vector, and an $M \times 1$ matrix is a column vector. All elements of a matrix can be real or complex numbers; $\sqrt{-1}$ can be written as either `'i'` or `'j'` provided they are not redefined by the user. A matrix is written with a square bracket `'[ ]'` with spaces separating adjacent columns and semicolons separating adjacent rows. For example, consider the following assignments of the variable `x`:

Real scalar $\gg$ `x = 5`

Complex scalar $\gg$ `x = 5+10j` (or $\gg$ `x = 5+10i`)

Row vector $\gg$ `x = [1 2 3]` (or `x = [1, 2, 3]`)

Column vector $\gg$ `x = [1; 2; 3]`

3×3 matrix $\gg$ `x = [1 2 3; 4 5 6; 7 8 9]`

There are a few notes of caution. Complex elements of a matrix should not be typed with spaces, i.e., `'-1+2j'` is fine as a matrix element, `'-1+2j'` is not. Also, `'-1+2j'` is interpreted correctly, whereas `'-1+j2'` is not (MATLAB interprets the `'j2'` as the name of a variable. You can always write `'-1+j*2'`.

12.2.1 Complex Number Operations

Some of the important operations on complex numbers are given below:

Complex scalar `>> x = 3+4j`

Real part of x `>> real(x)` ⇒ 3

Imaginary part of x `>> imag(x)` ⇒ 4

Magnitude of x `>> abs(x)` ⇒ 5

Angle of x `>> angle(x)` ⇒ 0.9273

Complex conjugate of x `>> conj(x)` ⇒ 3-4i

12.2.2 Generating Vectors

Vectors can be generated using the `':'` command. For example, to generate a vector `'x'` that takes on the values 0 to 10 in increments of 0.5, type the following which would generate a 1×21 matrix:

```
>> x = [0:0.5:10];
```

Other commands used to generate vectors include `'linspace'`, which generates a vector by specifying the first and last number and the number of equally spaced entries between the first and the last number, and `'logspace'`, which is the same except that entries are spaced logarithmically between the first and the last entry.

12.2.3 Accessing Vector Elements

Elements of a matrix are accessed by specifying the row and column. For example, in the matrix specified by `A = [1 2 3; 4 5 6; 7 8 9]`, the element in the first row and third column can be accessed by

`>> x = A(1,3)` which yields 3

The entire second row can be accessed by

`>> y = A(2,:)` which yields `[4 5 6]`

where the `':'` means 'take all the entries in the column'. A submatrix of A consisting of rows 1 and 2 and all the three columns is specified by

`>> z = A(1:2,1:3)` which yields `[1 2 3; 4 5 6]`

12.3 MATRIX OPERATIONS

MATLAB contains a number of arithmetic, relational, and logical operations on matrices.

12.3.1 Arithmetic Matrix Operations

The basic arithmetic operations on matrices (and of course on scalars, which are special cases of matrices) are

+	addition	\	left division
-	subtraction	^	exponentiation (power)
*	multiplication	'	conjugate transpose
/	right division		

An error message occurs if the sizes of matrices are incompatible for the operation. Division is defined as follows: The solution to $A * x = b$ is `x = A\b` and the solution to $x * A = b$ is `x = b/A`, provided A is invertible and all the matrices are compatible. Addition and subtraction involve element-by-element arithmetic operations, whereas matrix multiplication and division do not. However, MATLAB provides for element-by-element operations as well by prepending a `'.'` before the operator as shown below:

.*	multiplication	.^	exponentiation (power)
./	right division	.'	transpose (unconjugated)
.\	left division		

The difference between matrix multiplication and element-by-element multiplication can be seen in the following example:

```
≫ A = [1 2; 3 4]
A =
 1 2
 3 4
≫ B = A*A
B =
  7 10
 15 22
≫ C = A.*A
C =
 1  4
 9 16
```

12.3.2 Relational Operations

The basic relational operations are

<	less than	>=	greater than or equal to
<=	less than or equal to	==	equal to
>	greater than	~=	not equal to

These are element-by-element operations that return a matrix of ones (1 = true) and zeros (0 = false). Be careful of the distinction between `'='` and `'=='`.

12.4 FLOW CONTROL OPERATIONS

MATLAB contains the usual set of flow control structures, e.g., `for`, `while`, and `if`, plus the logical operators, e.g., & (and), | (or), and ~ (not).

12.5 MATH FUNCTIONS

MATLAB comes with a large number of built-in functions that operate on matrices on an element-by-element basis. These include:

`sin`	sine
`cos`	cosine
`tan`	tangent
`asin`	inverse sine
`acos`	inverse cosine
`atan`	inverse tangent
`exp`	exponential
`log`	natural logarithm
`log10`	common logarithm
`sqrt`	square root
`abs`	absolute value
`sign`	signum

12.6 SIMPLE PLOTTING COMMANDS

The simple two-dimensional plotting commands include

`plot`	plot in linear coordinates as a continuous function
`stem`	plot in linear coordinates as discrete samples
`loglog`	logarithmic x and y axes
`semilogx`	linear y and logarithmic x axes
`semilogy`	linear x and logarithmic y axes
`bar`	bar graph
`errorbar`	error bar graph
`hist`	histogram
`polar`	polar coordinates

SOLVED EXAMPLES

Generation of Basic Signals

Example 12.1 Write a program to generate the following signals:

(a) Unit step function $u(t)$
(b) Unit impulse function $\delta(t)$
(c) A rectangular pulse of width 2
(d) Unit ramp function $r(t)$

Solution

```
%***** Generation of Basic Signals ***
clc;clear all;close all;
t=-5:0.005:5;
%**** Unit Step Function ********
u=0.5*(sign(t)+1);
%**** Unit Impulse Function ****
impls=0.5*(sign(t)+1)-0.5*(sign(t-
0.005)+1);
```

```
%****Rectangular Pulse*******
rectpulse=(sign(t)+1)-(sign(t-2)+1);
%***Ramp Signal *******
tt=0:50;
rmp=0.5*tt.*(sign(tt)+1);
figure;subplot(411);
plot(t,u);AXIS([-5 5 0 1.5]);
title('Unit Step Function');
subplot(412);
plot(t,impls);AXIS([-5 5 0 1.5]);
title('Impulse Function');
subplot(413);
plot(t,rectpulse);AXIS([-5 5 0 2.5]);
title('Rectangular Pulse');
subplot(414);plot(tt,rmp);
xlabel('Time-->');ylabel('Amplitude--
>');
title('Unit Ramp Function');
```

The result is shown in Fig. 12.1.

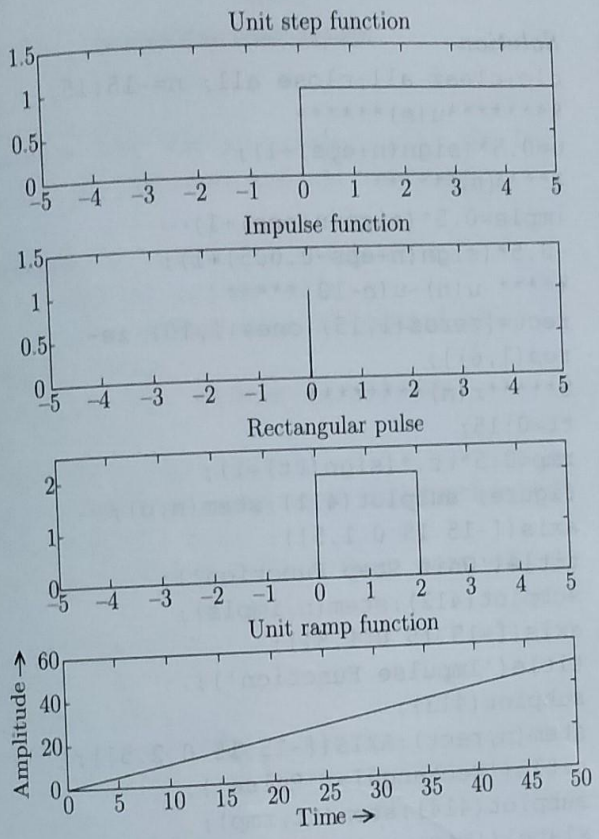

Fig. 12.1 Continuous-time basic signals

Example 12.2 Visualizing operations on independent variables. Write a program to generate the following signals:

(a) $u(t-1)$

(b) $u(2t+1)$

(c) $\delta(t-2.5)$

(d) $r(t-3)$

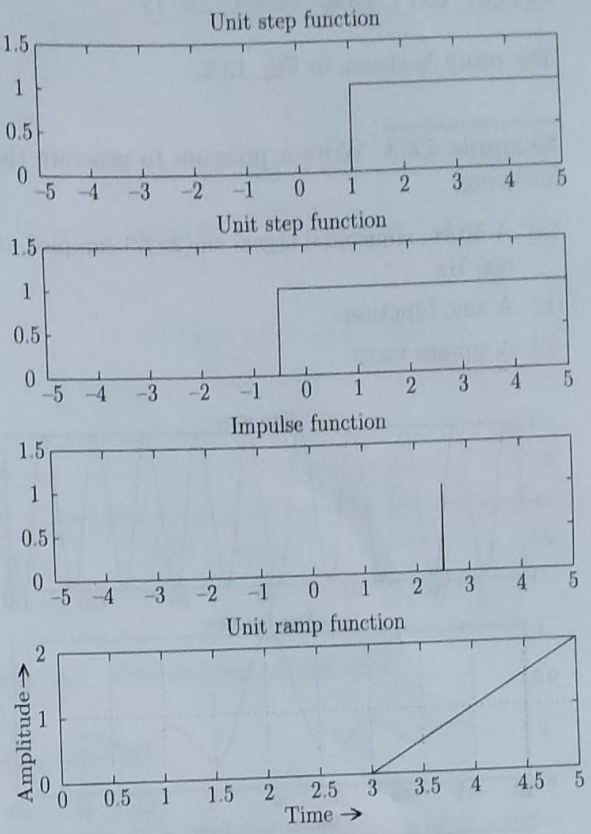

Fig. 12.2 Operations on independent variables

Solution

```
clc;clear all;close all;
t=-5:0.005:5;
u1=0.5*(sign(t-1)+1);%u(t-1)
u2=0.5*(sign(2*t+1)+1); %u(2t+1)
%**δ(t-2.5)****
ims=0.5*(sign(t-2.5)+1)...
-0.5*(sign(t-2.5-0.005)+1);
%*** Ramp Signal **********
tt=0:5; rmp=0.5*(tt-3).*(sign(tt-
3)+1);
figure; subplot(411);
plot(t,u1); axis([-5 5 0 1.5]);
title('Unit Step Function');
```

```
subplot(412);plot(t,u2);
axis([-5 5 0 1.5]);
title('Unit Step Function');
subplot(413);plot(t,ims);
axis([-5 5 0 1.5]);
title('Impulse Function');
subplot(414);plot(tt,rmp);grid on;
xlabel('Time-->');ylabel('Amplitude--
>');
title('Unit Ramp Function');
```

The result is shown in Fig. 12.2.

Example 12.3 Write a program to generate the following:

(a) A 50 Hz sinusoidal signal $\sin(2\pi ft)$ sampled at 600 Hz

(b) A sinc function

(c) A square wave

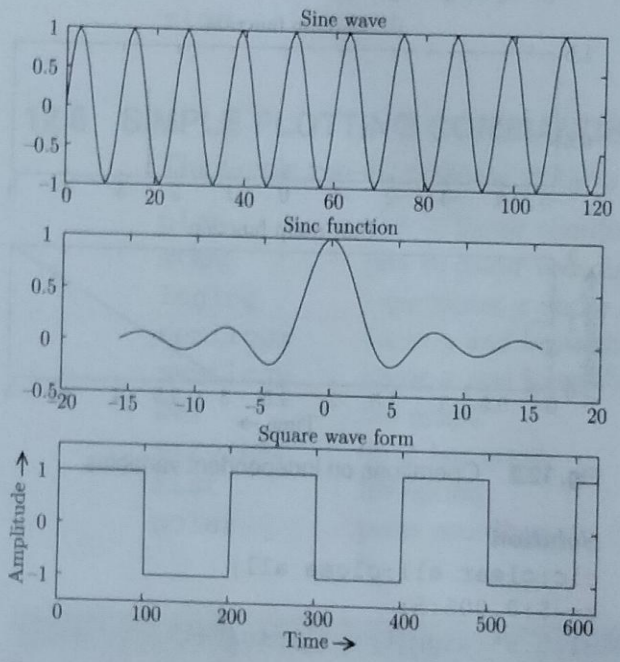

Fig. 12.3 Basic signals

Solution

```
%*** Sinusoidal Signal ****
clc;clear all;close all;
N=120;n=0:N-1;f=50;Ts=1/600;
x=sin(2*pi*f*n*Ts);
%***Sinc Function *****
t=-5*pi:pi/50:5*pi;
sincpulse=sin(t)./t;
%*** Square Wave ****
tt= 0:.0001:.0625;t1=1:626;
y = square(2*pi*50*tt);
figure; subplot(311);plot(n,x);
title('Sine Wave');
subplot(312);plot(t,sincpulse);
title('Sinc Function'); subplot(313);
plot(t1,y);axis([0 625 -1.5 1.5])
xlabel('Time-->');ylabel('Amplitude--
>');
title('Square Waveform');
```

The result is shown in Fig. 12.3

Example 12.4 Write a program to generate the following discrete-time signals:

(a) Unit step function $u(n)$

(b) Unit impulse function or unit sample function $\delta(n)$

(c) A rectangular pulse of width 10

(d) Unit ramp function $r(n)$

Solution

```
clc;clear all;close all; n=-15:15;
%*******u(n)******
u=0.5*(sign(n+eps)+1);
%***δ(n)******
impls=0.5*(sign(n+eps)+1)...
-0.5*(sign(n+eps-0.005)+1);
%**** u(n)-u(n-10)*****
rect=[zeros(1,15) ones(1,10) ze-
ros(1,6)];
%*****r(n)********
tt=0:15;
rmp=0.5*tt.*(sign(tt)+1);
figure; subplot(411);stem(n,u);
axis([-15 15 0 1.5]);
title('Unit Step Function');
subplot(412);stem(n,impls);
axis([-15 15 0 1.5]);
title('Impulse Function');
subplot(413);
stem(n,rect);AXIS([-15 15 0 2.5]);
title('Rectangular Pulse');
subplot(414);stem(tt,rmp);
xlabel('Time-->');ylabel('Amplitude--
>');
title('Unit Ramp Function');
```

The result is shown in Fig. 12.4.

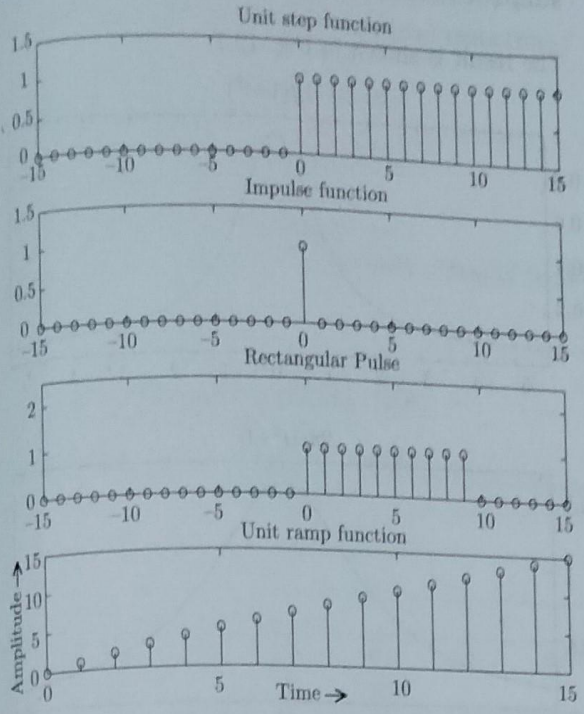

Fig. 12.4 Discrete-time basic signals

Example 12.5 Let $x(n) = u(n) - u(n-10)$. Decompose $x(n)$ into even and odd components.

Solution

```
clc;clear all;close all;
n=-10:15; x=[zeros(1,10) ones(1,10)
zeros(1,6)];
m=-fliplr(n); m1=min([m,n]);m2=
max([m,n]);
m=m1:m2; nm=n(1)-m(1);
n1=1:length(n); x1=zeros(1,length(m));
x1(n1+nm)=x; x=x1;
xe=0.5*(x+fliplr(x));% Even part of
x(n)
x0=0.5*(x-fliplr(x));% Odd part of
x(n)
figure; t=-15:15;
subplot(311);stem(t,x);
axis([-15 15 0 1.5]);title('Original
Signal');
subplot(312);stem(t,xe);
axis([-15 15 0 1.5]);title('Even
Part')
subplot(313);stem(t,x0);
axis([-15 15 -1 1]);
xlabel('Time-->');ylabel('Amplitude--
>');
title('Odd part');
```

The result is shown in Fig. 12.5.

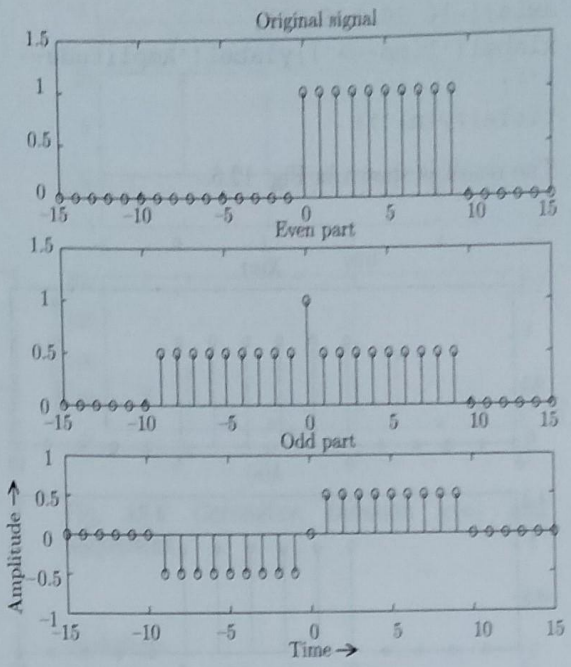

Fig. 12.5 Even and odd parts of $x(n)$

Convolution

Example 12.6 Let $x(n) = h(n) = u(n) - u(n-6)$. Write a program to determine $y(n) = x(n) * h(n)$.

Solution

```
clc;clear all;close all;
nx=-5:10;nh=-5:10;
x=[zeros(1,5) ones(1,6) zeros(1,5)];
h=[zeros(1,5) ones(1,6) zeros(1,5)];
Nx=length(x); Nh=length(h);
Ny=Nx+Nh-1;
newx=[x zeros(1,Ny-Nx)];
newh=[h zeros(1,Ny-Nh)];
for n=0:Ny-1
sum=0;
for k=0:n
sum=sum+newx(k+1)*newh(n-k+1);
```

```
end
y(n+1)=sum;
end
n1=min(nx)+min(nh):max(nx)+max(nh);
figure; subplot(311);stem(nx,x);
axis([-5 10 0 1.5]);title('x(n)');
subplot(312);stem(nh,h);
axis([-5 10 0 1.5]);title('h(n)');
subplot(313);stem(n1,y);
axis([-10 20 0 6.5]);
xlabel('Time-->');ylabel('Amplitude-->');
title('y(n)');
```

The result is shown in Fig. 12.6.

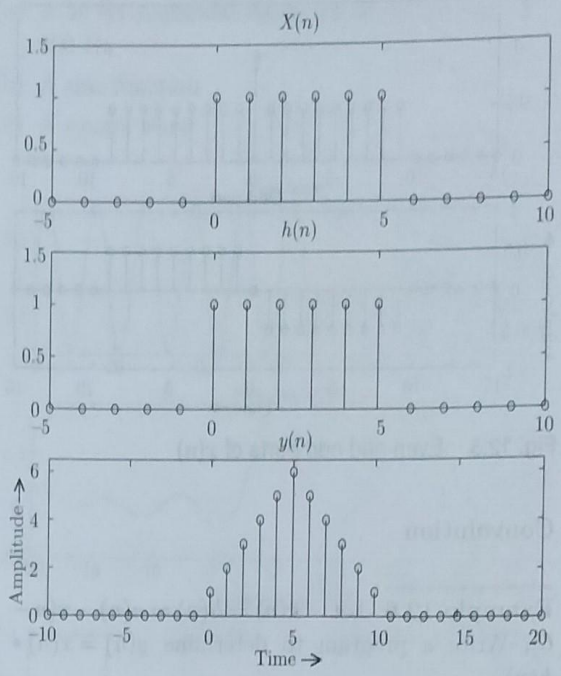

Fig. 12.6 Convolution of a rectangular signal with itself

Example 12.7 Determine the convolution of $x(t) = \dfrac{1}{1+t^2}$ with itself.

Solution

```
clc; clear all;close all;
syms t tau%for symbolic calculation
x=1/(1+t^2);
%***Convolution Integral**
z=int(subs(x,tau)*subs(x,t-tau),tau,-inf,inf);
z=simplify(z);
figure; subplot(211); ezplot(x);
subplot(212);ezplot(z);
```

The result is shown in Fig. 12.7.

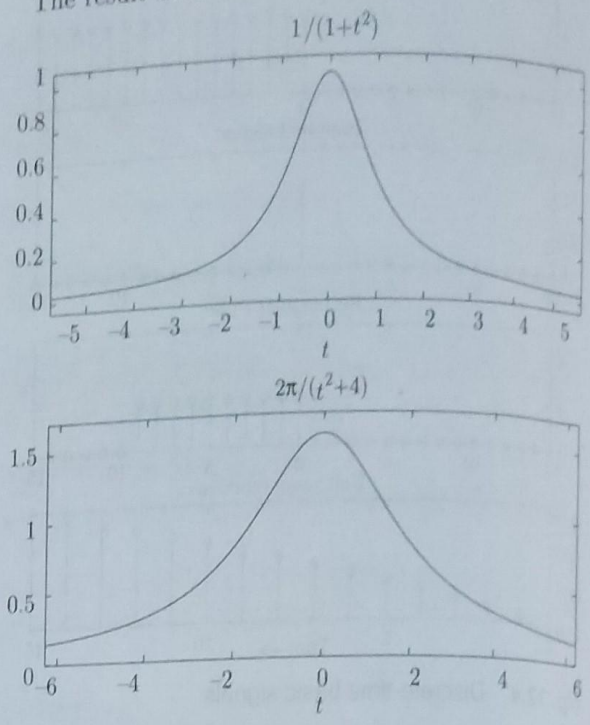

Fig. 12.7 Convolution of $x(t) = \dfrac{1}{1+t^2}$ with itself

Correlation

Example 12.8 Compute the crosscorrelation between the following two sequences:

$$x(n) = \{1, 2, \underset{\uparrow}{-1}, 3\}$$

$$h(n) = \{\underset{\uparrow}{3}, -2, 1, 4\}$$

Solution

```
clc;clear all;close all;
x=[1 2 -1 3];nx=-2:1;
h=[3 -2 1 4];nh=0:3;
hh=fliplr(h);
Nx=length(x);
Nh=length(h);
Ny=Nx+Nh-1;
newx=[x zeros(1,Ny-Nx)];
newh=[hh zeros(1,Ny-Nh)];
for n=0:Ny-1
sum=0;
for k=0:n
sum=sum+newx(k+1)*newh(n-k+1);
```

```
end
y(n+1)=sum;
end
n1=min(nx)+min(nh):max(nx)+max(nh);
figure;
subplot(311);stem(nx,x);
title('x(n)');
subplot(312); stem(nh,h);
title('h(n)');
subplot(313);stem(n1,y);
xlabel('Time-->');ylabel('Amplitude--
>');
title('y(n)');
```

The result is shown in Fig. 12.8.

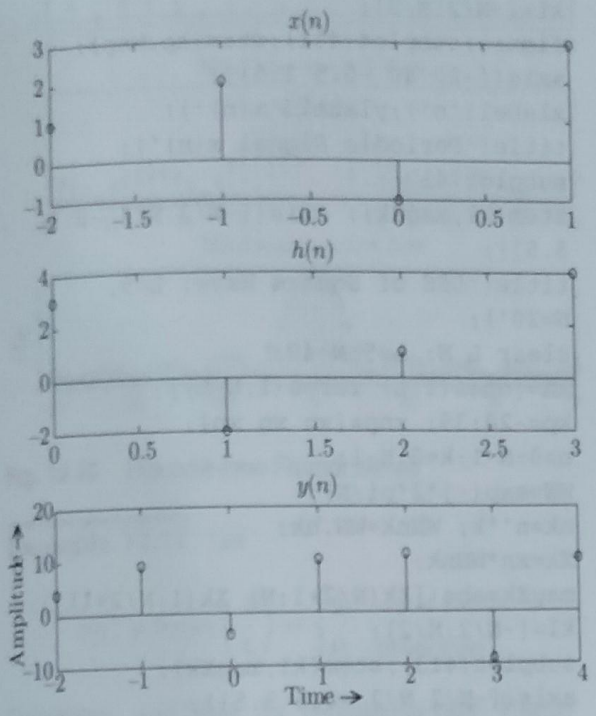

Fig. 12.8 Correlation between $x(n)$ and $x(n)$ sequences

Example 12.9 Let

$$x(n) = \{3, 11, 7, \underset{\uparrow}{0}, -1, 4, 2\}$$

be a prototype sequence, and let $y(n)$ be its noise-corrupted-and-shifted version

$$y(n) = x(n-2) + w(n)$$

where $w(n)$ is a Gaussian noise sequence with zero mean and variance 1. Compute the crosscorrelation between $y(n)$ and $x(n)$.

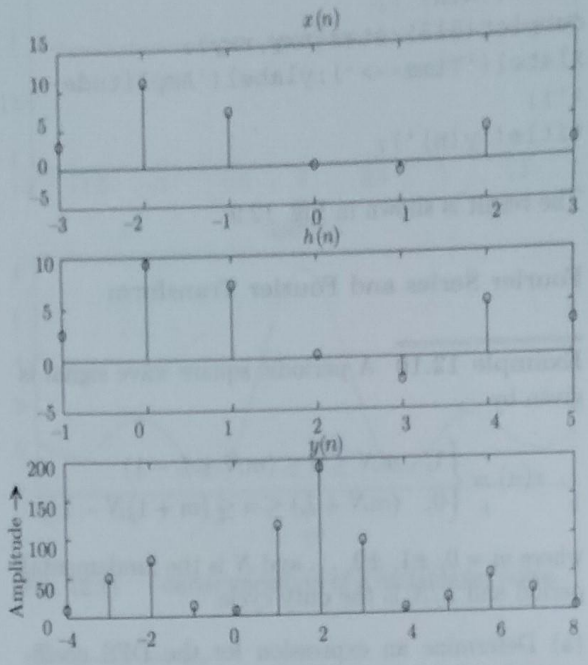

Fig. 12.9 Correlation between $x(n)$ and $y(n)$ sequences

Solution

```
clc;clear all;close all;
x=[3 11 7 0 -1 4 2];nx=-3:3;
xn2=[3 11 7 0 -1 4 2];ny=nx+2;%x(n-2)
%***Gaussian Noise***
w=randn(1,length(xn2));
%**y(n)=x(n-2)+w(n)**
y=xn2+w;
yy=fliplr(y);
Nx=length(x);
Ny=length(y);
Nxy=Nx+Ny-1;
newx=[x zeros(1,Nxy-Nx)];
newy=[yy zeros(1,Nxy-Ny)];
for n=0:Nxy-1
sum=0;
for k=0:n
sum=sum+newx(k+1)*newy(n-k+1);
end
rxy(n+1)=sum;
end
nxy=min(nx)+min(ny):max(nx)+max(ny);
figure;
```

```
subplot(311);stem(nx,x);
title('x(n)');
subplot(312);stem(ny,y);
title('h(n)');
subplot(313);stem(nxy,rxy);
xlabel('Time-->');ylabel('Amplitude-->');
title('y(n)');
```

The result is shown in Fig. 12.9.

Fourier Series and Fourier Transform

Example 12.10 A periodic square wave signal is given by

$$x(n) = \begin{cases} 1, & mN \le n \le (mN + L - 1) \\ 0, & (mN + L) \le n \le (m+1)N - 1 \end{cases}$$

where $m = 0, \pm 1, \pm 2, \ldots$ and N is the fundamental period and L/N is the duty cycle.

(a) Determine an expression for the DFS coefficients $|X_k|$ in terms of L and N.

(b) Plot the magnitude $|X_k|$ for $L = 5$, $N = 20$; $L = 5$, $N = 40$; $L = 9$, $N = 60$.

(c) Comment on the results.

Solution

By definition $\omega_0 = 2\pi/N$

$$X_k = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j\omega_0 kn}$$

$$= \frac{1}{N}\sum_{n=0}^{L-1} e^{-j\frac{2\pi}{N}kn}$$

$$= \begin{cases} \frac{1}{N}L, & k = 0, \pm N, \pm 2N, \ldots \\ \frac{1}{N}\frac{1-e^{-j2\pi kL/N}}{1-e^{-j2\pi k/N}}, & \text{otherwise} \end{cases}$$

The last expression can be simplified to

$$\frac{1}{N}\frac{1-e^{-j2\pi kL/N}}{1-e^{-j2\pi k/N}}$$

$$= \frac{1}{N}\frac{e^{-j\pi kL/N}}{e^{-j\pi k/N}}\frac{e^{j\pi kL/N} - e^{-j\pi kL/N}}{e^{j\pi k/N} - e^{-j\pi k/N}}$$

$$= \frac{1}{N}e^{-j\pi k(L-1)/N}\frac{\sin(\pi kL/N)}{\sin(\pi k/N)}$$

The magnitude of X_k is given by

$$|X_k|, = \begin{cases} \frac{1}{N}L, & k = 0, \pm N, \pm 2N, \ldots \\ \left|\frac{1}{N}\frac{\sin(\pi kL/N)}{\sin(\pi k/N)}\right|, & \text{otherwise} \end{cases}$$

```
clc;clear all;close all;
L=5;N=20;
xn=[ones(1,L) zeros(1,N-L)];
kp=-20:39;
xnp=[xn xn xn];
n=0:N-1;k=0:N-1;
WN=exp(-j*2*pi/N);
nk=n'*k; WNnk=WN.^nk; Xk=xn*WNnk;
magXk=abs([Xk(N/2+1:N) Xk(1:N/2+1)]);
k1=[-N/2:N/2];
figure;subplot(411);stem(kp,xnp);
axis([-20 40 -0.5 1.5]);
xlabel('n');ylabel('x(n)');
title('Periodic Signal x(n)');
subplot(412);
stem(k1,magXk); axis([-N/2 N/2 -0.5
5.5]);
title('DFS of Square Wave: L=5,
N=20');
clear L N; L=5;N=40;
xn=[ones(1,L) zeros(1,N-L)];
kp=-20:39; xnp=[xn xn xn];
n=0:N-1;k=0:N-1;
WN=exp(-j*2*pi/N);
nk=n'*k; WNnk=WN.^nk;
Xk=xn*WNnk;
magXk=abs([Xk(N/2+1:N) Xk(1:N/2+1)]);
k1=[-N/2:N/2];
subplot(413);stem(k1,magXk);
axis([-N/2 N/2 -0.5 5.5]);
title('DFS of Square Wave:
L=5, N=40');
clear L N; L=9;N=60;
xn=[ones(1,L) zeros(1,N-L)];
kp=-20:39; xnp=[xn xn xn];
n=0:N-1;k=0:N-1;
WN=exp(-j*2*pi/N);
nk=n'*k; WNnk=WN.^nk;
Xk=xn*WNnk;
magXk=abs([Xk(N/2+1:N)
Xk(1:N/2+1)]);
k1=[-N/2:N/2];
subplot(414);stem(k1,magXk);
axis([-N/2 N/2 -0.5 9.5]);
```

```
xlabel('k');ylabel('|Xk|');
title('DFS of square wave: L=9,
N=60');
```

The result is shown in Fig. 12.10.

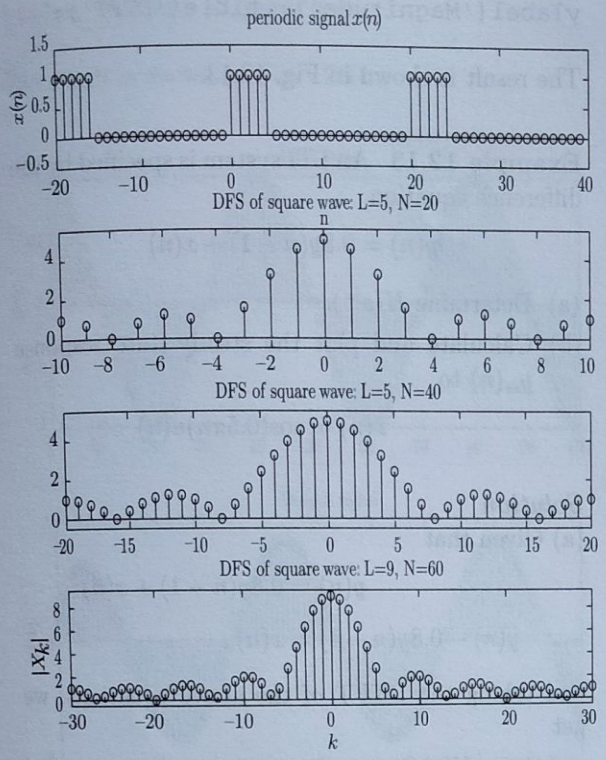

Fig. 12.10 Discrete-time Fourier series

Example 12.11 Let

$$x(t) = 2\,\text{rect}\left(\frac{t}{4}\right) = \begin{cases} 2, & -2 < t < 2 \\ 0, & \text{otherwise} \end{cases}$$

Determine and plot its continuous-time Fourier transform.

Solution

```
clc; clear all;close all;
syms t omega%for symbolic calculation
x=2; expw=exp(-j*omega *t);
%***Fourier Transform**
z=int(x*expw,omega,-2,2);
z=simplify(z);
figure(1);
subplot(211); ezplot('2', [-2 2]);
subplot(212) ezplot(z);
```

The result is shown in Fig. 12.11.

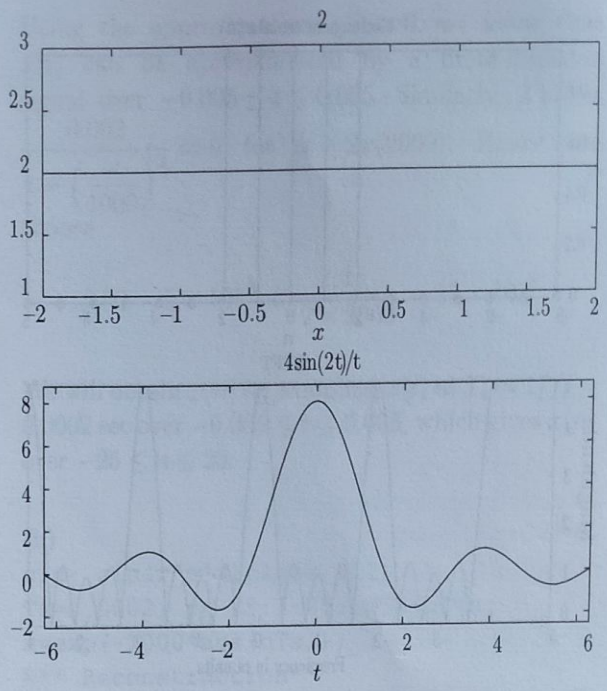

Fig. 12.11 Fourier transform of a rectangular pulse

Example 12.12 Determine and plot the discrete-time Fourier transform (DTFT) of the following:
(a) A rectangular pulse given by

$$x(n) = \begin{cases} 1, & -2 \le n \le 2 \\ 0, & \text{otherwise} \end{cases}$$

(b) A sinusoidal signal

$$x(n) = \cos\left(\frac{\pi n}{2}\right), \quad 0 \le n \le 100$$

Solution

(a)

```
clc; clear all;close all;
n=-2:2;x=ones(1,5); k=-400:400;
w=(pi/100)*k;
X=x*(exp(-j*pi/100)).^(n'*k);
magX=abs(X);
figure;subplot(211);
stem(-8:8, [zeros(1,6) x zeros(1,6)]);
xlabel('n');ylabel('x');
title('Rectangle Signal x(n)')
subplot(212); plot(w/pi,magX);
xlabel('Frequency in pi Units');
ylabel('Magnitude'); title('DTFT');
```

The result is shown in Fig. 12.12.

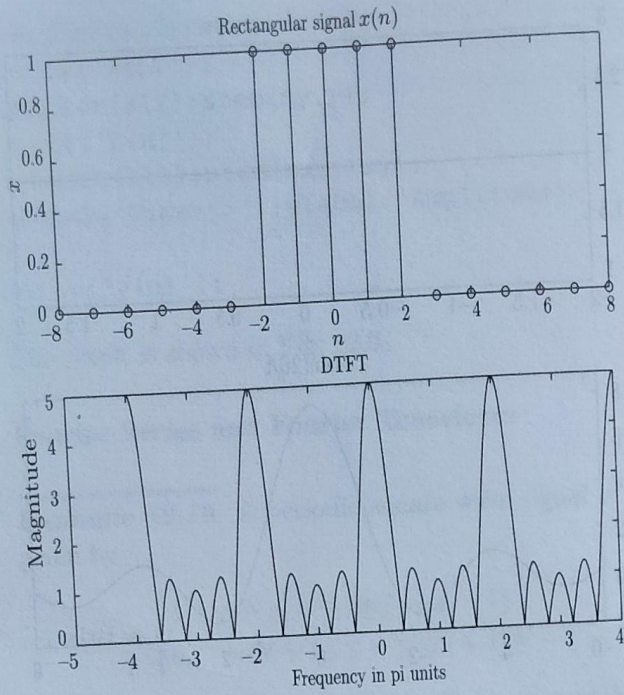

Fig. 12.12 DTFT of a rectangular pulse

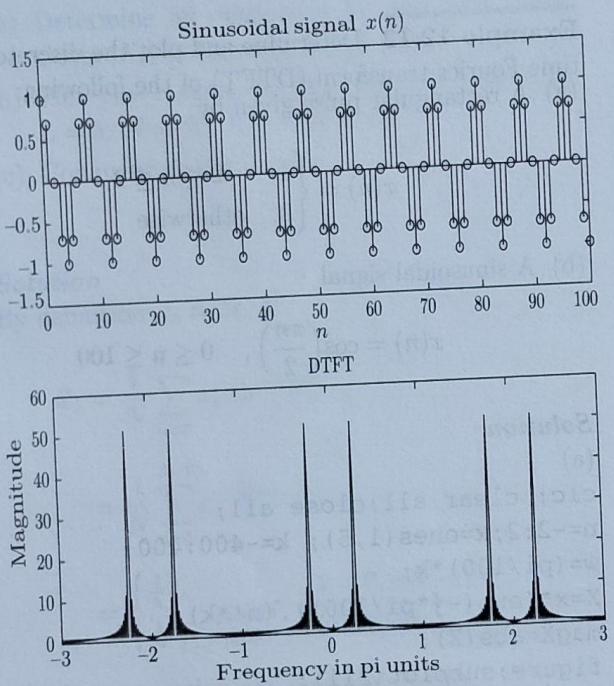

Fig. 12.13 DTFT of a sinusoidal signal

(b)

```
clc; clear all;close all;
n=0:100;x=cos(pi*n/4);
k=-300:300;w=(pi/100)*k;
X=x*(exp(-j*pi/100)).^(n'*k);
magX=abs(X);
figure;subplot(211);
stem(n,x);axis([0 100 -1.5 1.5]);
xlabel('n');ylabel('x');
title('Sinusoidal Signal x(n)');
subplot(212); plot(w/pi,magX);
xlabel('Frequency in pi Units');
ylabel('Magnitude'); title('DTFT');
```

The result is shown in Fig. 12.13.

Example 12.13 An LTI system is specified by the difference equation

$$y(n) = 0.8y(n-1) + x(n)$$

(a) Determine $H(e^{j\omega})$.

(b) Calculate and plot the steady-state response $y_{ss}(n)$ to

$$x(n) = \cos(0.5\pi n)u(n)$$

Solution

(a) Given that

$$y(n) = 0.8y(n-1) + x(n)$$

$$y(n) - 0.8y(n-1) = x(n)$$

and taking the DTFT of the above equations, we get

$$\frac{Y(e^{j\omega})}{X(e^{j\omega})} = H(e^{j\omega}) = \frac{1}{1 - 0.8e^{-j\omega}}$$

$$= \frac{e^{j\omega}}{e^{j\omega} - 0.8}$$

(b) Given that the input is $x(n) = \cos(0.5\pi n)$ with frequency $\omega_0 = 0.05\pi$,

$$H(e^{j\omega})\Big|_{\omega_0 = 0.05\pi} = \frac{1}{1 - 0.8e^{-j0.05\pi}}$$

$$= 4.092e^{-j0.5377}$$

Therefore, the response to a periodic signal is

$$y_{ss}(n) = 4.0928\cos(0.05\pi n - 0.5377)$$

$$= 4.0928\cos[0.05\pi(n - 3.42)]$$

This means that at the output the sinusoid is scaled by 4.0928 and shifted by 3.42 samples.

```
clc; clear all;close all;
b=1,a=[1 -0.8];
n=[0:100];x=cos(0.05*pi*n);
```

```
y=filter(b,a,x);
subplot(211);stem(n,x);
xlabel('n');ylabel('x(n)');
title('Input Signal');
subplot(212);stem(n,y);
xlabel('n');ylabel('y(n)');
title('Output Signal');
```

The result is shown in Fig. 12.14.

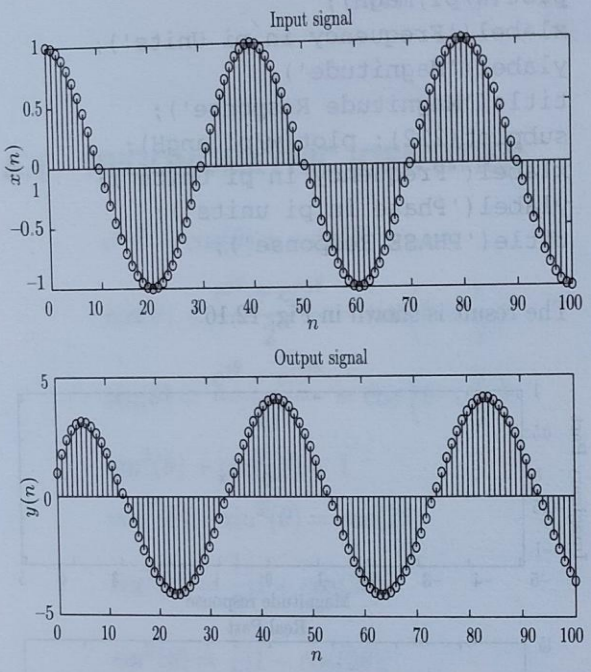

Fig. 12.14 Steady-state response

Sampling Theorem

Example 12.14 Let $x(t) = e^{-1000|t|}$.

(a) Sample $x(t)$ at $f_s = 5000$ samples/sec to obtain $x(n)$.

(b) From the samples $x(n)$, reconstruct $x(t)$.

Solution

(a) We know that

$$e^{-a|t|} \longleftrightarrow \frac{2a}{a^2+\omega^2}$$

$$e^{-1000|t|} \longleftrightarrow \frac{2 \times 1000}{1000^2+\omega^2}$$

$$\longleftrightarrow \frac{0.002}{1+\left(\frac{\omega}{1000}\right)^2}$$

Using the approximation $e^{-5} \approx 0$, we know that $x(t)$ can be approximated by a finite-duration signal over $-0.005 \le t \le 0.005$. Similarly, $X(\omega) = \dfrac{0.002}{1+\left(\frac{\omega}{1000}\right)^2} \approx 0$ for $\omega \ge 2\pi(2000)$. Hence, we choose

$$\Delta(t) = 5 \times 10^{-5} \ll \frac{1}{2 \times 2000} = 25 \times 10^{-5}$$

We will obtain $x(n)$ by sampling $x(t)$ at $T_s = 1/f_s = 0.0002$ sec over $-0.005 \le t \le 0.005$, which gives $x(n)$ over $-25 \le n \le 25$.

(b)

```
clc; clear all;close all;
Ts=0.0002; n=-25:1:25; nTs=n*Ts;
x=exp(-1000*abs(nTs));
%** Reconstruction**
Dt=0.0002;t=-0.005:Dt:0.005; fs=5000;
xt=x*sinc(fs*(ones(length(n),1)*t...
-nTs'*ones(1,length(t))));
figure;;stem(n,x);
hold on; plot(-25:25,xt);
xlabel('Time');ylabel('x(t)');
title('Reconstructed Signal');
```

The result is shown in Fig. 12.15.

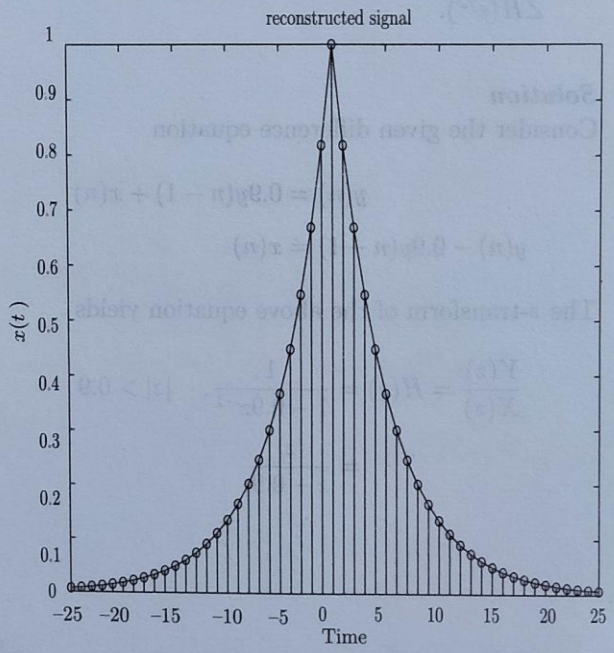

Fig. 12.15 Sampling and interpolation

Laplace Transform

Example 12.15 Using MATLAB's symbolic math toolbox, determine the Laplace transform of the following signals:

(a) $x_1(t) = te^{-at}u(t)$

(b) $x_2(t) = [\sin(at) + \cos(bt)](u(t)$

Solution

```
clc; clear all;close all;
syms a b t
x1t=t*exp(-a*t);
x2t=sin(a*t)+cos(b*t);
X1s=laplace(x1t);
X2s=laplace(x2t);
X2s=simplify(X2s);
disp(X1s);disp(X2s);
```

z-Transform

Example 12.16 Given a causal system

$$y(n) = 0.9y(n-1) + x(n)$$

(a) Find $H(z)$ and sketch its pole-zero plot.

(b) Plot the frequency response $|H(e^{j\omega})|$ and $\angle H(e^{j\omega})$.

Solution

Consider the given difference equation

$$y(n) = 0.9y(n-1) + x(n)$$

$$y(n) - 0.9y(n-1) = x(n)$$

The z-transform of the above equation yields

$$\frac{Y(z)}{X(z)} = H(z) = \frac{1}{1 - 0.9z^{-1}}, \quad |z| > 0.9$$

$$= \frac{z}{z - 0.9}$$

The system has one pole at $z = 0.9$ and one zero at the origin.

```
clc; clear all;close all;
b=[1];a=[1 -0.9];
zplane(b,a);
w=[-200:1:200]*pi/100;
H=freqz(b,a,w);
magH=abs(H);angH=angle(H);
figure; subplot(211);
plot(w/pi,magH);
xlabel('Frequency in pi Units');
ylabel('Magnitude');
title('Magnitude Response');
subplot(212); plot(w/pi,angH);
xlabel('Frequency in pi Units');
ylabel('Phase in pi units');
title('PHASE Response');
```

The result is shown in Fig. 12.16.

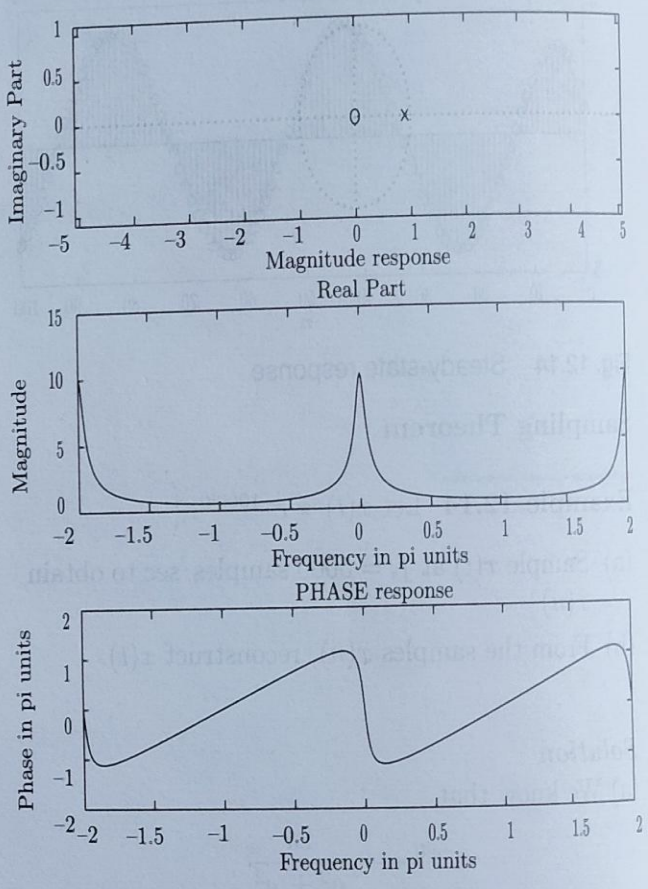

Fig. 12.16 Pole-zero plot and frequency response