80509 LINEAR DIGITAL FILTERING I

PART I: Design and Implementation of Digital Filters

1) Structures for implementing digital filters.
2) FIR (finite impulse response) and IIR (infinite impulse response) filters.
3) How to study the filter performance with the aid of the transfer function: stability, frequency response.
4) Filter design process.
5) Requiremenst for the amplitude, phase, phase delay and group delay responses.
6) Various approximation criteria for meeting the given specifications,illustrated by typicalclassical solutions.

   These solutions will be considered in more details in Parts III and IV of these lecture notes.

- What to read for the examination ?:

1) What is a linear digital filter and how to analyse it ?: difference equation; transfer function; the role of frequency, amplitude, phase, phase delay, and group delay responses; filter stability.
2) Direct-form and cascade-form structures.

DESIGN AND IMPLEMENTATION OF DIGITAL FILTERS

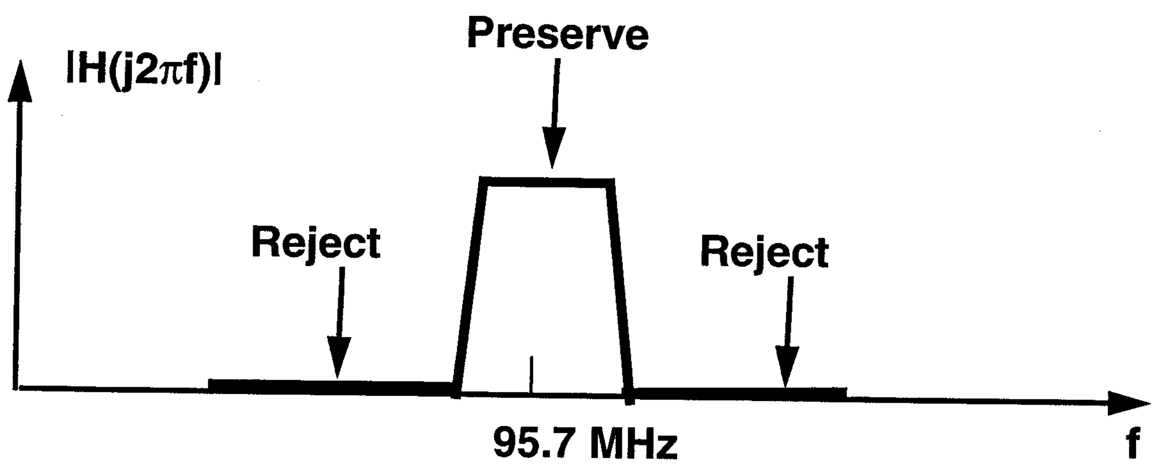
1. What is a digital filter?
2. Different filter types: Infinite impulse response (IIR) and finite impulse response (FIR) filters.
3. Various structures for the same transfer function.
4. Various implementation forms: computer program, signal processor, VLSI-circuit.
5. Filter design process: Given the specifications as well as the implementation form, find the tranfer function as well as a proper structure to fullfil the criteria as effectively as possible.
6. Synthesis of FIR filters.
7. Synthesis of IIR filters.
8. Finite wordlength effects: Scaling, output noise due to the multiplication roundoff errors, oscillations (limit cycle oscillations and overflow oscillations), effects of coefficient quantization.

The above list consists of the main topics of this course.

WHAT IS A FILTER?

- If you are willing to listen to Radio 957, then your radio set needs a filter which is able to pick up a channel which consists of frequencies in the vicinity of 95.7 MHz.
- This filter is a bandpass filter which preserves a frequency band in the vicinity of 95.7 MHz and rejects the other frequencies.

Requirements for the filter



WHAT IS A DIGITAL FILTER?

- **Answer:** The following difference equation:

$$y[n] = \sum_{k=0}^{M} a_k x[n-k] + \sum_{k=1}^{N} b_k y[n-k].$$

- Initial conditions: $y[-1] = y[-2] = \cdots = y[-N] = 0$ and $x[n] = 0$ for $n < 0$.
- Then,

$$y[0] = a_0 x[0]$$

$$y[1] = a_0 x[1] + a_1 x[0] + b_1 y[0]$$

$$y[2] = a_0 x[2] + a_1 x[1] + a_2 x[0] + b_1 y[1] + b_2 y[0].$$

- In general, for $r < \max\{N, M\}$

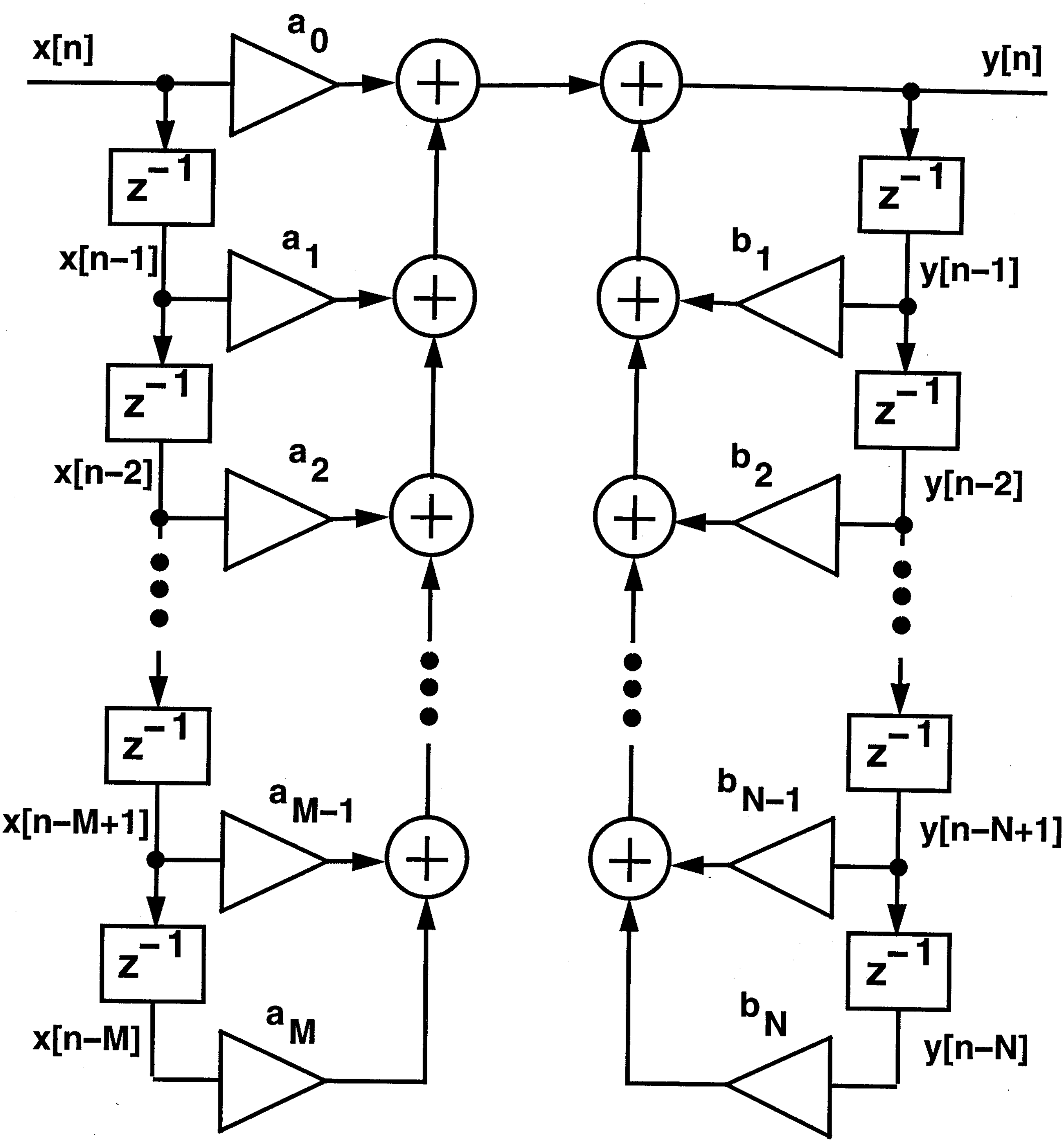
$$y[r] = \sum_{k=0}^{\min\{M,r\}} a_k x[r-k] + \sum_{k=1}^{\min\{N,r\}} b_k y[r-k]$$

and for $r \geq \max\{N, M\}$

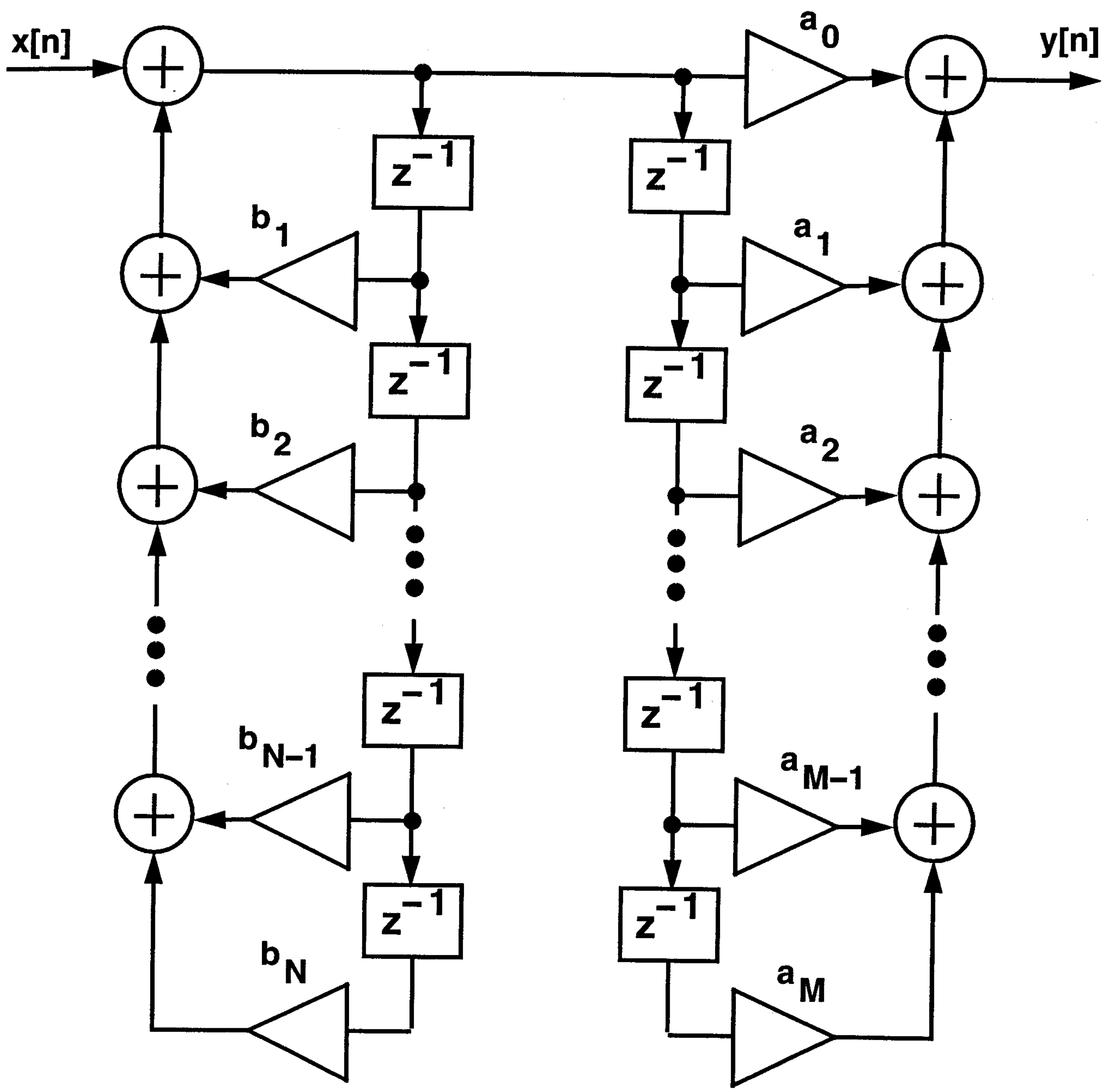
$$y[r] = \sum_{k=0}^{M} a_k x[n-r] + \sum_{k=1}^{N} b_k y[r-k].$$

- The following four pages give practical implementations for the above difference equation.

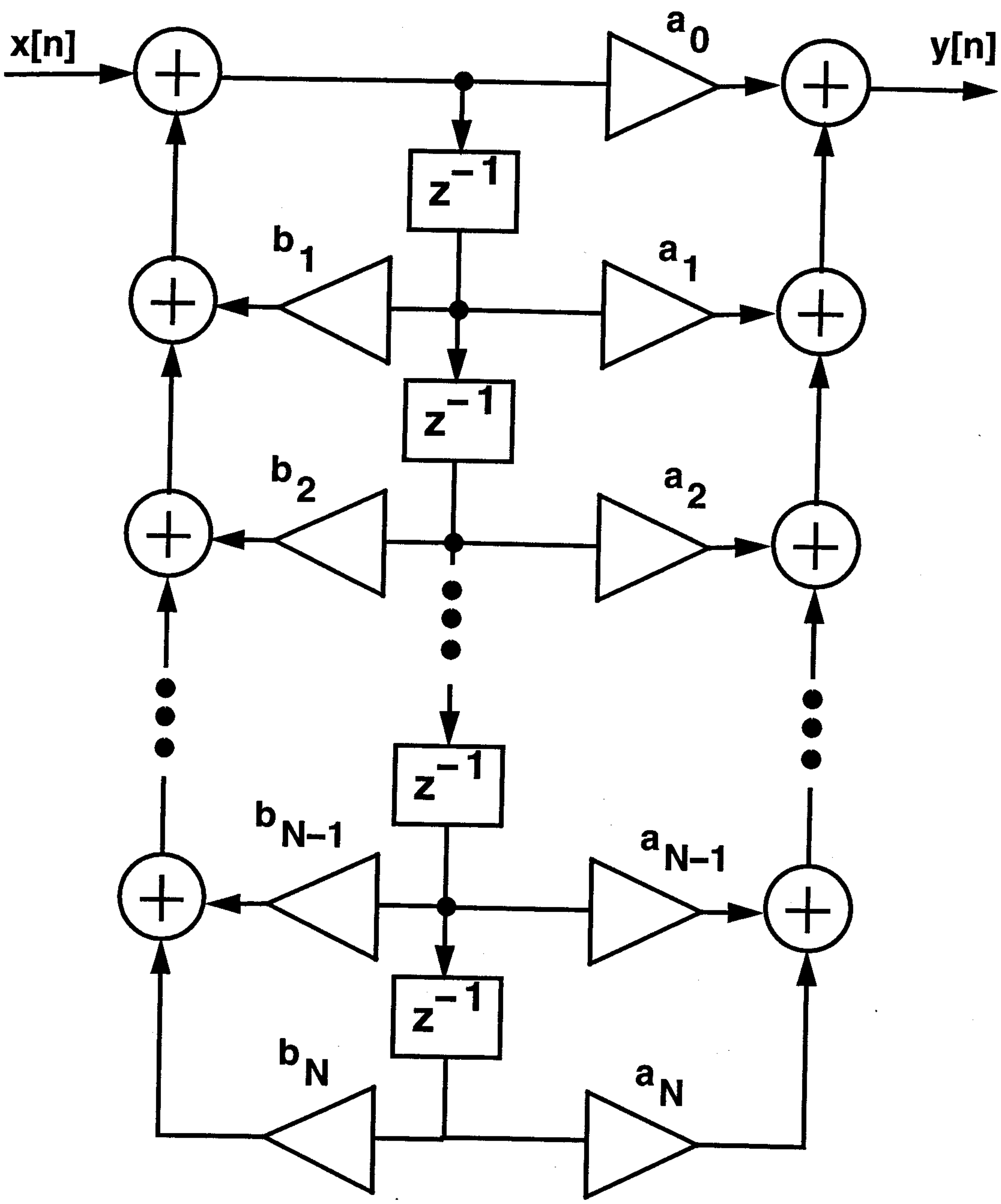
DIRECT-FORM I STRUCTURE FOR OUR DIFFERENCE EQUATION



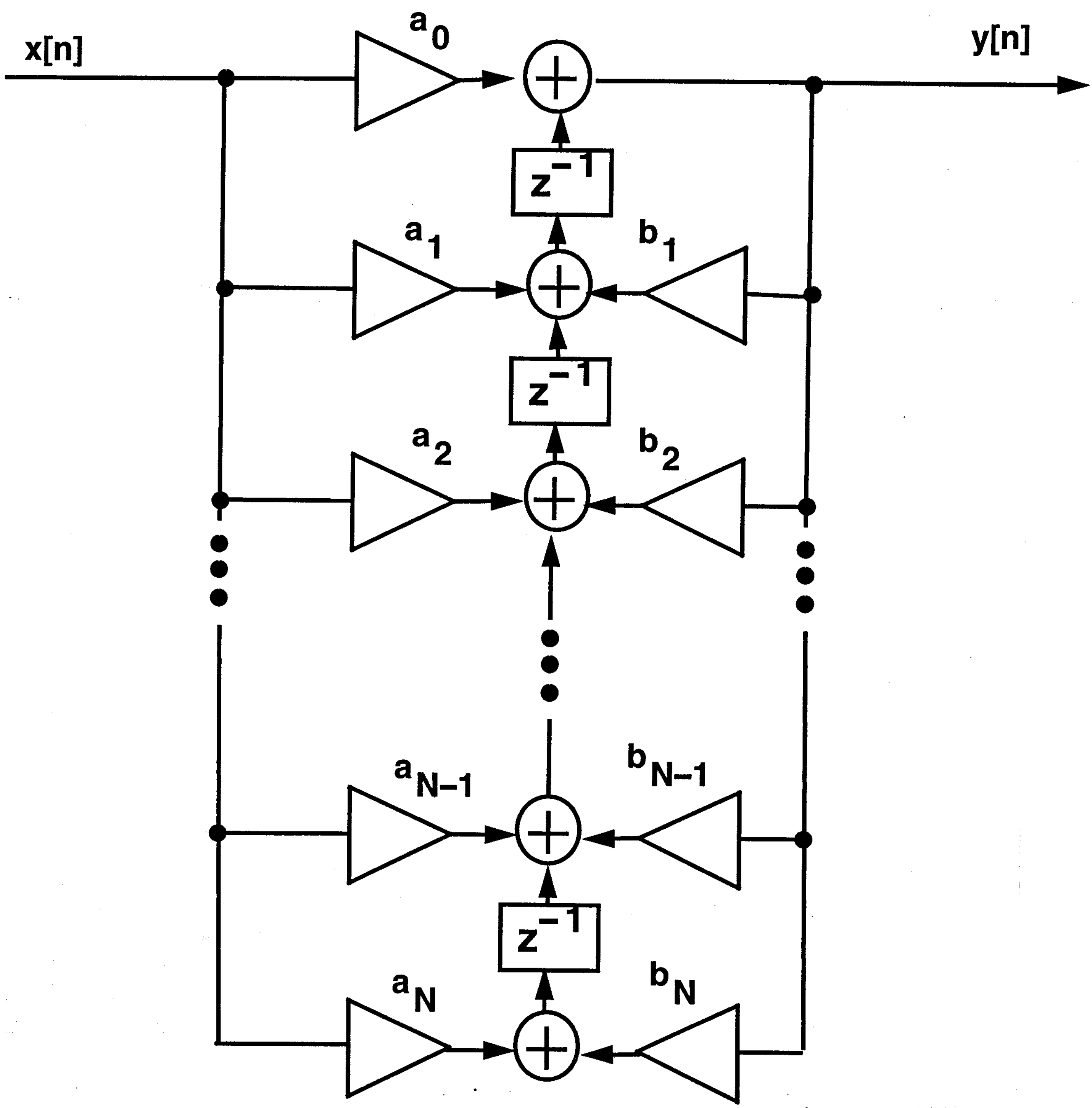
STRUCTURE RESULTING WHEN INTERCHANGING THE FEEDBACK AND FEEDFORWARD PARTS



DIRECT-FORM II STRUCTURE OBTAINED BY SHARING THE DELAY TERMS AND ASSUMING THAT $M = N$



STRUCTURE OBTAINED BY TRANSPOSING THE DIRECT-FORM II STRUCTURE

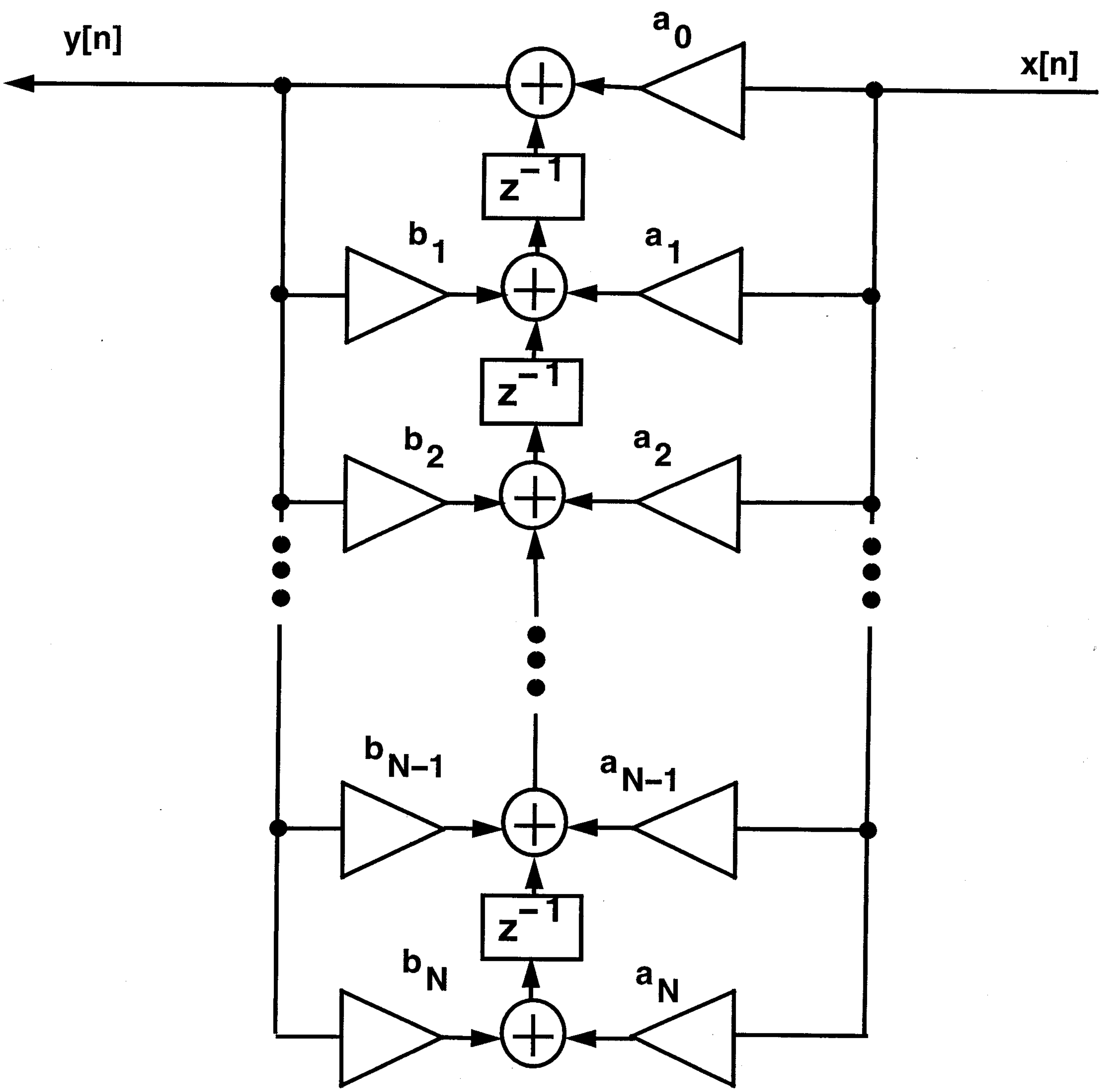


HOW TO GENERATE A TRANSPOSED STRUCTURE HAVING THE SAME TRANSFER FUNCTION

1) Reverse the directions of all arrows.
2) Interchange all branch nodes (black dots) and adders.
3) Interchange the output and input.

- When performing these operations to the structure of page 6, we arrive at the structure of page 9.
- Finally, the structure of page 7 is obtained by taking the mirror-image of the structure of page 9.

INTERMEDIATE STEP IN GENERATING THE STRUCTURE OF PAGE 7

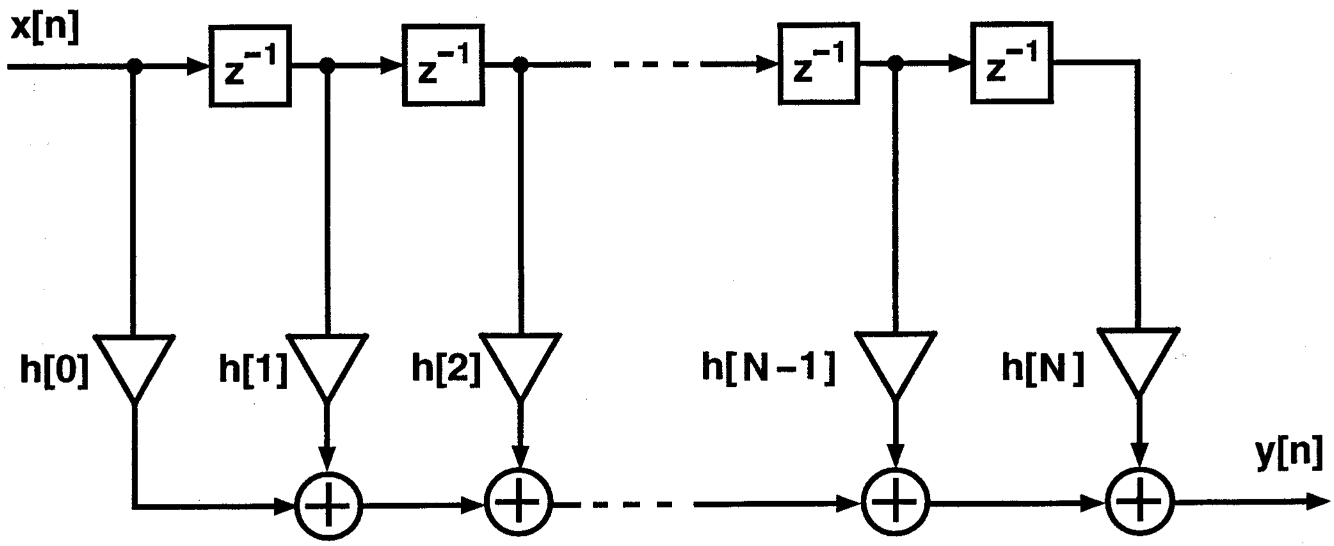


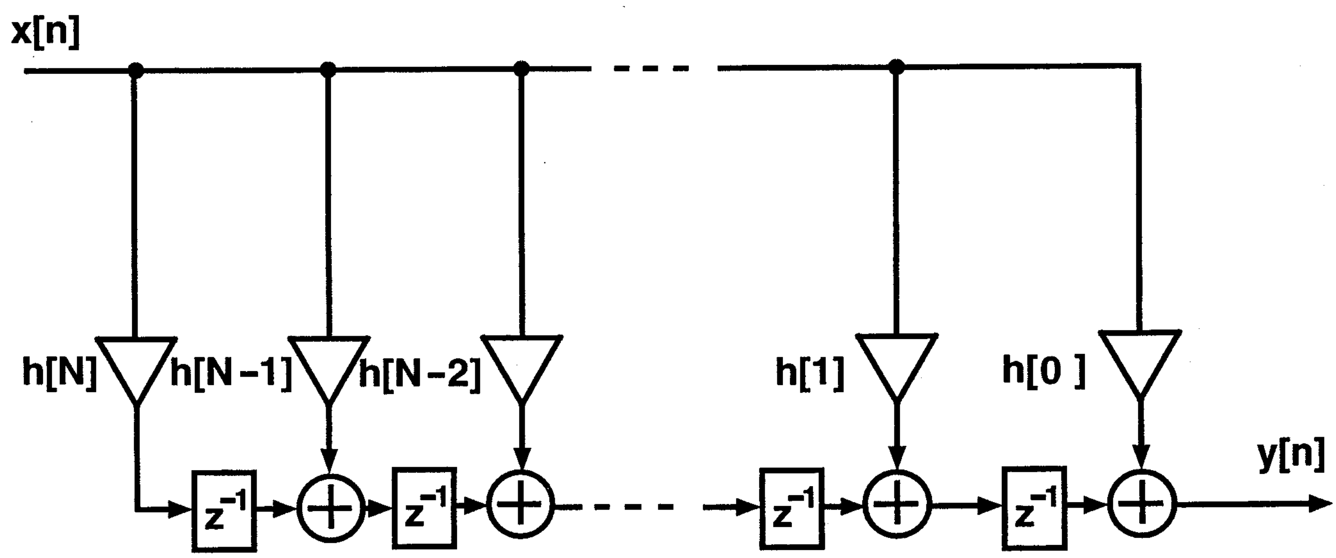
TWO BASIC DIGITAL FILTER TYPES

- For $N \geq 1$, the impulse response is of infinite duration $\Rightarrow$ These filters are called **infinite impulse response (IIR)** filters.
- For $N = 0$, the impulse response is of finite duration $\Rightarrow$ These filters are called **finite impulse response (FIR)** filters.
- For FIR filters, the difference equation is usually expressed in terms of the impulse response coefficients $h[n]$ as follows ($M = N$ and $a_k = h[k]$ for $k = 1, 2, \cdots, N$):

$$y[n] = \sum_{k=0}^{N} h[k]x[n-k].$$

TWO BASIC STRUCTURES FOR FIR FILTERS: FIRST DIRECT-FORM, THEN TRANSPOSED FORM





HOW TO STUDY THE PERFORMANCE OF A DIGITAL FILTER?

- **ANSWER:** In terms of the transfer function.
- The transfer function of a difference equation

$$y[n] = \sum_{k=0}^{M} a_k x[n-k] + \sum_{k=1}^{N} b_k y[n-k]$$

is

$$H(z) = Y(z)/X(z) = \frac{\sum_{k=0}^{M} a_k z^{-k}}{1 - \sum_{k=1}^{N} b_k z^{-k}}.$$

- Note that in the denominator of $H(z)$, the signs of the b_k's have been changed compared to the difference equation. In the direct-form realization, we use the signs of the difference equation!!
- Alternatively, $H(z)$ can be expressed as

$$H(z) = a_0 \frac{\prod_{k=0}^{M}(1 - c_k z^{-1})}{\prod_{k=1}^{N}(1 - d_k z^{-1})},$$

where d_k's and c_k's are the poles and zeros of $H(z)$, respectively.

- **FILTER STABILITY:** All the poles must lie inside the unit circle $|z| = 1$, that is, $|d_k| < 1$ for $k = 1, 2, \cdots, N$.

FREQUENCY-DOMAIN BEHAVIOR

- The response of our system with transfer function $H(z)$ to an exitation

$$x[n] = A\cos[n\omega_0 + \phi]$$

is given by

$$y[n] = A|H(e^{j\omega_0})|\cos[n\omega_0 + \phi + \arg\ H(e^{j\omega_0})]. \quad (A)$$

- The frequency response is thus obtained by evaluating $H(z)$ along the unit circle $z = e^{j\omega}$, that is,

$$H(e^{j\omega}) = H(z)_{|z\,=\,e^{j\omega}}.$$

- The complex-valued $H(e^{j\omega})$ is expressible in the forms (Re $\{z\}$ and Im $\{z\}$ stand for the real and imaginary parts of a complex number z):

$$\begin{aligned} H(e^{j\omega}) &= \mathrm{Re}\,\{H(e^{j\omega})\} + j\mathrm{Im}\,\{H(e^{j\omega})\} \\ &= |H(e^{j\omega})|e^{j\arg\ H(e^{j\omega})}. \end{aligned}$$

- Here,

$$|H(e^{j\omega})| = \sqrt{[\mathrm{Re}\,\{H(e^{j\omega})\}]^2 + [\mathrm{Im}\,\{H(e^{j\omega})\}]^2}$$

and

$$\arg\ H(e^{j\omega}) = \mathrm{atan2}(\mathrm{Im}\,\{H(e^{j\omega})\}, \mathrm{Re}\,\{H(e^{j\omega})\},$$

where

$$\text{atan2}(y,x) = \begin{cases} \tan^{-1}(y/x), & x \geq 0 \\ \pi + \tan^{-1}(y/x), & x < 0 \text{ and } y \geq 0 \\ -\pi + \tan^{-1}(y/x), & x < 0 \text{ and } y < 0. \end{cases}$$

- The above definition of arg $H(e^{j\omega})$ forced it to take values between $-\pi$ and π. In practice, it is desired to make this function continuous. In Matlab, this can be done by using the command 'unwrap'. As can be seen in the phase response plots of the filters later, the resulting arg $H(e^{j\omega})$ is continuous except for those angular frequencies, where the filter has a single zero on the unit circle. At these points, the is a jump of π upwards. Equally well we can use a jump of π downwards.
- A good compromize would be to use a jump upwards every second time and a jump downwards every second time. However, Matlab uses all the time jumps upwards.
- As seen from equation (A) on the previous page, $|H(e^{j\omega})|$ is the **amplitude response** of the filter and tells us the change caused by the filter for the oscillation amplitude of a sinusoidal signal of frequency ω.
- arg $H(e^{j\omega})$ is the **phase response** of the filter and tells us the change caused by the filter for the phase of a sinusoidal signal of frequency ω.

Alternative form for the output signal

- The output signal $y[n]$ of page 13 is also expressible as

$$y[n] = A|H(e^{j\omega_0})| \cos[(n - \tau_p(\omega_0))\omega_0 + \phi],$$

where

$$\tau_p(\omega) = -\arg\ H(e^{j\omega})/\omega$$

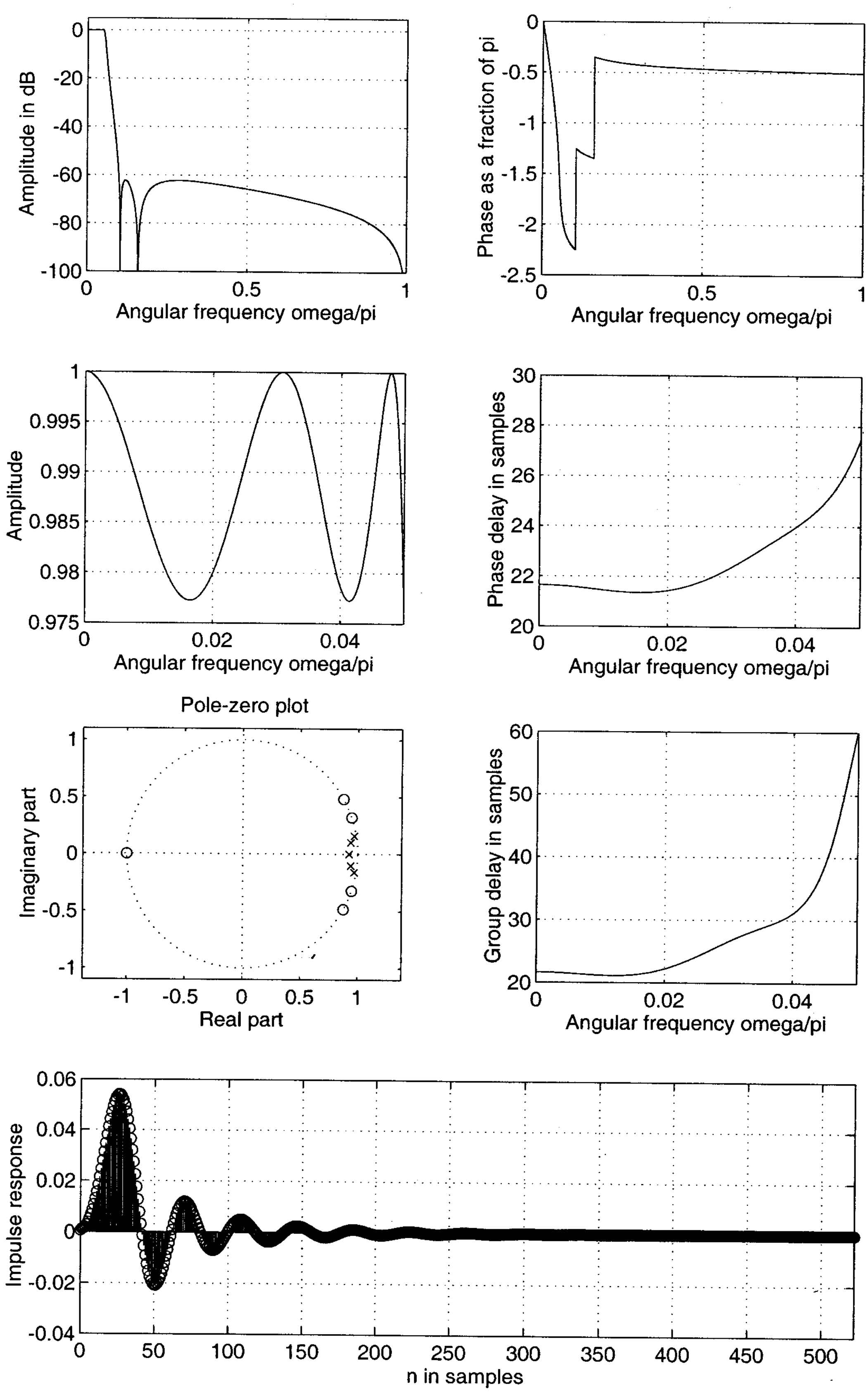
is the phase delay of the filter.

- The phase delay at $\omega = \omega_0$ gives directly the delay caused by the filter to a sinusoidal signal of frequency $\omega = \omega_0$.
- As an example we consider the effect of filtering using a fifth-order elliptic filter with characteristics shown on page 17.
- As test signals, we use $x[n] = \cos(n\omega_0)u(n)$, which starts at $n = 0$ and is zero before this time instant.
- For the first signal $\omega_0 = 0.04\pi$ so that its oscillation angular frequency is within the passband of the filter. As seen from page 17, for

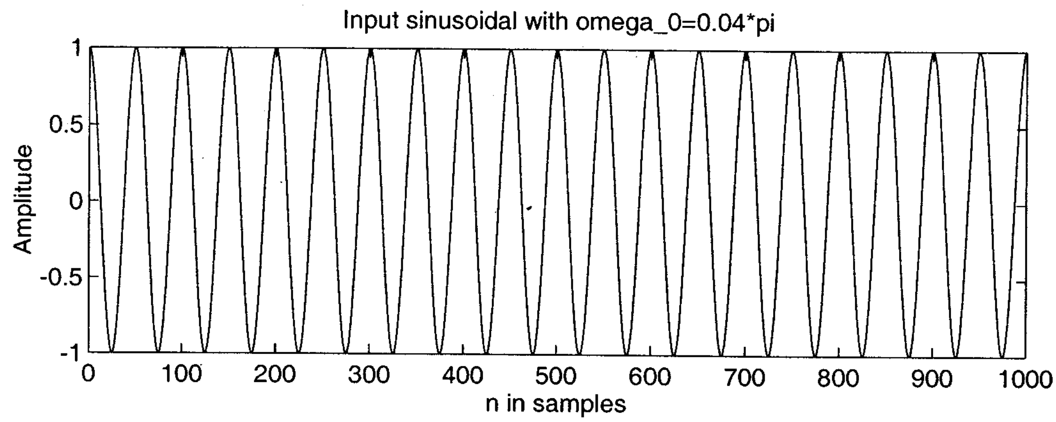
this frequency $|H(e^{j\omega_0})| = 0.9786$ and $\tau_p(\omega_0) = 23.98$ samples.

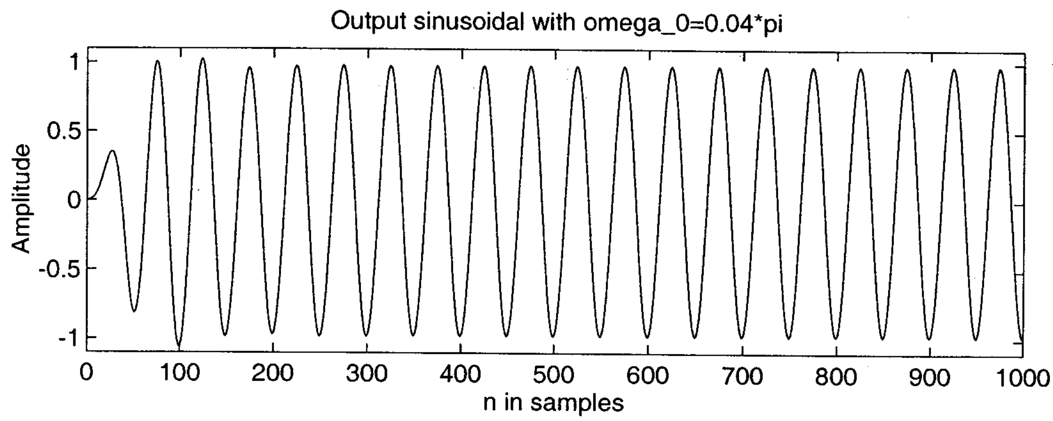
- As seen from page 18, the oscillation amplitude and the delay are the right ones after the transient part.
- Note that the signals are discrete, even though they have been plotted like for continuous-time signals.
- For the second test signal, $\omega_0 = 0.5\pi$ which is in the stopband of our filter and $|H(e^{j\omega_0})| = 0.00052$.
- As seen from page 18, after the transient part, this signal is attenuated to the desired level.
- Pages 19–27 give a matlab-file for plotting the responses of pages 17 and 18. Please study it. Later on, the same file is used as an example on how to generate cascade-form and parallel-form structures for the overall transfer function (pages 33, 34 and 38, 39).

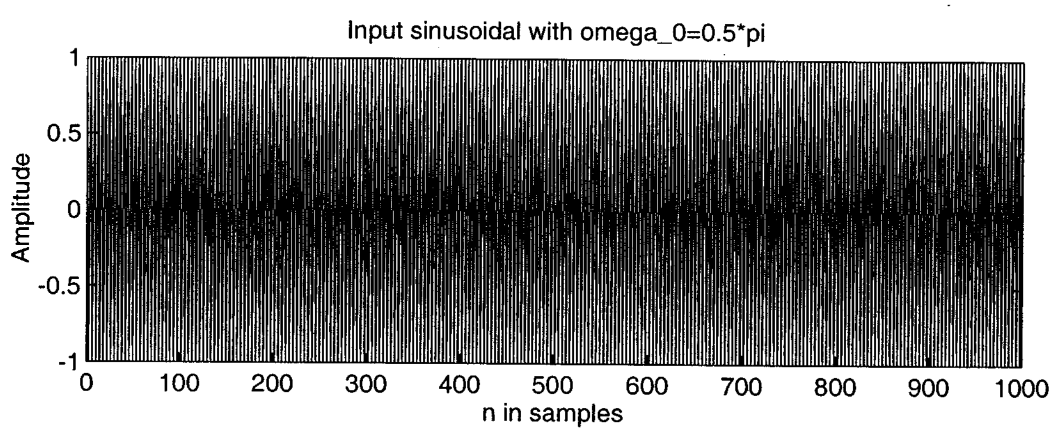
Responses for a fifth-order elliptic filter

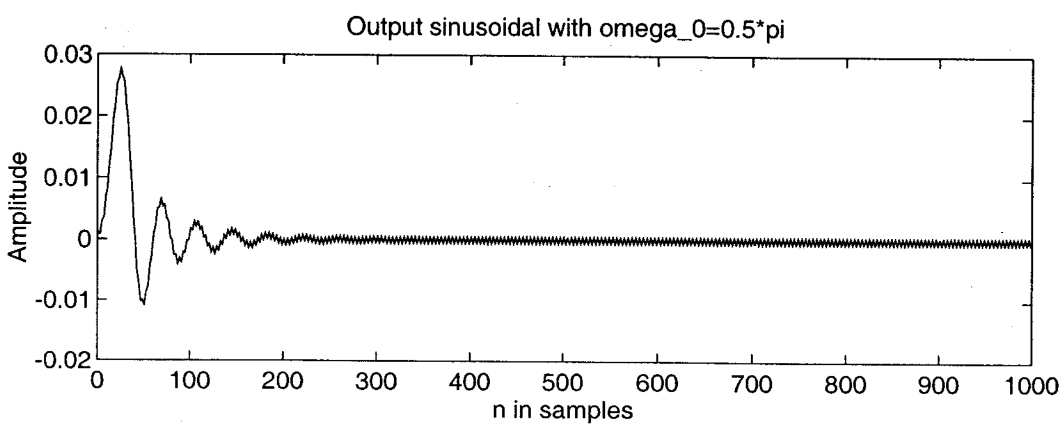


Filtering of two sinusoidal signals of frequencies $\omega_0 = 0.04\pi$ and $\omega_0 = 0.5\pi$









```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% The purpose of this matlab-file is
% four-fold. First, a fifth-order elliptic
% filter with omega_p=0.05pi, omega_s=0.1pi,
% A_p=0.2 dB, and A_s=62.273 dB is designed and
% several responses are plotted. Second, the
% effects of filtering are considered for two
% sinusoidals, one having the angular frequncy
% in the passband region and the second one in
% the stopband region. Third, it is shown how
% the filter shapes the Fourier transform of
% two pulses. Fourth, it is shown how
% the overall filter transfer function can be
% realized using a cascade-form and a parallel-
% form structure
% This file can be found in SUN's:
% ~ts/matlab/dsp/lueiir5.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
[N, wn] = ellipord(.05, .1, 0.2, 62.2734);
[B, A] = ellip(N,0.2,62.2734,wn);
%------------------------------------------
% The transfer function is of the form (A(1)=1)
% H(z)=N(z)/D(z), where
% N(z)=B(1)+B(2)z^(-1)+B(3)z^(-2)+B(4)z^(-3)+
% B(5)z^(-4)+B(6)z^(-5)
% and
% D(z)=A(1)+A(2)z^(-1)+A(3)z^(-2)+A(4)z^(-3)+
% A(5)z^(-4)+A(6)z^(-5).
%------------------------------------------
% Frequency response of the filter: e is the complex
% frequency response as a function of omega running
% from zero to pi. The number of grid points is
% 8*1024+1
%------------------------------------------
[e omega]=freqz(B,A,8*1024);
%--------------------------------------------
% Generate the phase response and make it continuous
% except for those angular frequency points where the
% filter has a zero on the unit circle. At these points
% there is a jump of pi upwards. Matlab does this using
% first the 'angle' command and the the 'unwrap'
% command. It should be pointed out that equally well
% we can jump downwards by pi. A good selection would
% be to jump upwards every second time and downwards
% every second time. Matlab is jumping upwards each
```

```
% time.
%---------------------------------------------
ang=angle(e);
ang=unwrap(ang);
%---------------------------------------------
% Generate the phase delay.
%---------------------------------------------
for i=1:length(e)
phad(i)=-ang(i)/omega(i);end
figure(1)
subplot(2,2,3)
%------------------------------
% Amplitude response in the passband
%------------------------------
plot(omega/pi,(abs(e)));grid;axis([0 .05 .975 1]);
ylabel('Amplitude');
xlabel('Angular frequency omega/pi');
subplot(2,2,1)
%------------------------------
% Amplitude response in dB
%------------------------------
plot(omega/pi,20*log10(abs(e)));grid;
axis([0 1 -100 5]);
ylabel('Amplitude in dB');
xlabel('Angular frequency omega/pi');
subplot(2,2,2)
%------------------------------
% Phase response
%------------------------------
plot(omega/pi,ang/pi);grid;
ylabel('Phase as a fraction of pi');
axis([0 1 -2.5 0]);
xlabel('Angular frequency omega/pi')
subplot(2,2,4)
%------------------------------
% Phase delay
%------------------------------
plot(omega/pi,phad);grid;axis([0 .05 20. 30.]);
ylabel('Phase delay in samples');
xlabel('Angular frequency omega/pi');
figure(2)
subplot(2,2,1)
%------------------------------
% Pole-zero plot
%------------------------------
zplane(B,A);title('Pole-zero plot');
subplot(2,2,2)
```

```
%--------------------------------
% Group delay
%--------------------------------
grpdelay(B,A,8*1024);grid;
axis([0 .05 20. 60.]);
ylabel('Group delay in samples');grid;
xlabel('Angular frequency omega/pi');
subplot(2,1,2)
%--------------------------------
% Impulse response
%--------------------------------
impz(B,A);grid;
ylabel('Impulse response');
xlabel('n in samples')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% Time to consider two sinusoidals. For the
% first one, the angular frequency 0.04*pi is in
% the passband region, whereas for the second one,
% 0.5*pi is in the stopband region.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
Nin = 1001;          %number of samples
n = 0:Nin-1;         %index vector
%----------------------------------------
% sinusoidals
%----------------------------------------
 in1=sin(.04*pi*n);
 in2=sin(0.5*pi*n);
%------------------------------------------
% Filter these signals
%------------------------------------------
 out1 = filter(B,A,in1);
 out2 = filter(B,A,in2);
%------------------------------------------
% Plot the input and output signals
%------------------------------------------
figure(3)
subplot(2,1,1);plot(n,in1);
title('Input sinusoidal with omega_0=0.04*pi');
ylabel('Amplitude');
xlabel('n in samples')
subplot(2,1,2);plot(n,out1);
axis([0 1000 -1.1 1.1]);
title('Output sinusoidal with omega_0=0.04*pi');
ylabel('Amplitude');
xlabel('n in samples')
```

```
figure(4)
subplot(2,1,1);plot(n,in2);
title('Input sinusoidal with omega_0=0.5*pi');
ylabel('Amplitude');
xlabel('n in samples')
subplot(2,1,2);plot(n,out2);
title('Output sinusoidal with omega_0=0.5*pi');
ylabel('Amplitude');
xlabel('n in samples')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% Time to consider two simple pulses. For the
% first one, x(0)=x(1)=x(2)=x(3)=x(4)=1/5 and
% for the second one, x(0)=x(2)=x(4)=1 and
% x(1)=x(3)=-1.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
Nin = 1001;          %number of samples
n = 0:Nin-1;         %index vector
%-----------------------------------------
% pulses
%-----------------------------------------
 in1=zeros(size(n));
 for k=1:5
   in1(k)=1/5;
 end
 in2=in1;
 in2(2)=-in1(2);
 in2(4)=-in1(4);
%-------------------------------------------
% Filter these signals
%-------------------------------------------
 out1 = filter(B,A,in1);
 out2 = filter(B,A,in2);
%-------------------------------------------
% Plot the input and output signals as well as
% their spectra
%-------------------------------------------
figure(5)
subplot(2,1,1);impz(in1(1:201))
title('Input pulse x(0)=x(1)=x(2)=x(3)=x(4)=1/5');
ylabel('Amplitude');
xlabel('n in samples')
subplot(2,1,2);impz(out1(1:201))
title('Output pulse');
ylabel('Amplitude');
xlabel('n in samples')
```

```
[H1,w]=freqz(in1,1,2^11);
[H2,w]=freqz(out1,1,2^11);
figure(6)
subplot(2,1,1);plot(w/pi,abs(H1));axis([0 1 0 1]);
title('Amplitude spectrum for the input pulse');
grid;
ylabel('Amplitude');xlabel('Angular frequency omega/pi')
subplot(2,1,2);plot(w/pi,abs(H2));axis([0 1 0 1]);
grid;
title('Amplitude spectrum for the output pulse');
ylabel('Amplitude');xlabel('Angular frequency omega/pi')
figure(7)
subplot(2,1,1);impz(in2(1:201))
title('Input pulse x(0)=x(2)=x(4)=1/5, x(1)=x(3)= -1/5');
ylabel('Amplitude');
xlabel('n in samples')
subplot(2,1,2);impz(out2(1:201))
title('Output pulse');
ylabel('Amplitude');
xlabel('n in samples')
[H1,w]=freqz(in2,1,2^11);
[H2,w]=freqz(out2,1,2^11);
figure(8)
subplot(2,1,1);plot(w/pi,abs(H1));axis([0 1 0 1]);
title('Amplitude spectrum for the input pulse');
grid;
ylabel('Amplitude');xlabel('Angular frequency omega/pi')
subplot(2,1,2);plot(w/pi,abs(H2));axis([0 1 0 .2]);
grid;
title('Amplitude spectrum for the output pulse');
ylabel('Amplitude');xlabel('Angular frequency omega/pi')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% Coefficients for the cascade-form structure
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%
%-----------------------------------------
% For the direct-form filter
%-----------------------------------------
disp('Direct-Form Filter:')
disp('------------------')
disp('Coefficients for the numerator polynomial')
B
disp('Coefficients for the denominator polynomial')
A
pause;
```

```
%----------------------------------------
% Poles and zeros
%----------------------------------------
pol=roots(A);
%---------------------------------------------
% Sort the poles in such a manner that the radii
% are in the ascending order. This means that pol(1)
% contains the real pole, pol(2) and pol(3) the
% the innermost complex conjugate pole pair (smaller
% angle), and pol(4) and pol(5) the outermost complex
% conjugate pole pair (larger angle).
%---------------------------------------------
pol=sort(pol);
%---------------------------------------------
% Sort the zeros in such a manner that the real parts
% are in the ascending order. This means that zer(1)
% contains the real pole at z=-1, pol(2) and pol(3) the
% the  complex conjugate zero pair having the larger
% angle, and pol(4) and pol(5) the zero pair having
% the smaller angle. See page 17 in the lecture notes.
%---------------------------------------------
zer=roots(B);
[Y,I]=sort(real(zer));
zer=zer(I);
%---------------------------------------------
% The overall filter consists of a constant, one
% first-order block and two second-order blocks.
% The first order block realizes the real pole and
% zero. The complex conjugate pole pair with a
% smaller angle and the zero pair on the unit circle
% with a larger angle are included in the first
% second-order block. The remaining pole and zero
% pairs are included in the second second-order
% block.
%---------------------------------------------
% First order block: (1+a*z^(-1))/(1+b*z^(-1)) with
% a=-zer(1) and b=-pol(1); this is expressed in the
% form (ar(1)+ar(2)*z^(-1))/(br(1)+br(2)*z^(-1)).
% Note that in the actual realization to be considered
% later in these lecture notes, we realize ar(2)=a and
% -br(2)=-b. b is in the feedback loop. Therefore, we
% realize -b instead of b. The sign of the
% coefficients in the denominator of H(z) are always
% changed in the realization!!
%---------------------------------------------
disp('Cascade-Form Realization')
disp('----------------------')
```

```
disp('First-order block')
ar(1)=1;ar(2)=-zer(1);
br(1)=1;br(2)=-pol(1);
disp('Numerator first, then denominator')
ar
br
%-----------------------------------------------
% Second-order blocks: N_k(z)/D_k(z) k=1,2, where
% N_k(z)=1-2r_k*cos(phi_k)*z^(-1)+(r_k)^2*z^(-2) and
% D_k(z)=1-2R_k*cos(Phi_k)*z^(-1)+(R_k)^2*z^(-2).
% Here the zero pair is located at z=r_k*e^(+-j*phi_k)
% and the pole pair at z=R_k*e^(+-j*Phi_k).
% Equally well the zero pair is located at
% z=Real{zero}+-j*Imag(zero) and the pole pair at
% z=Real{pole}+-j*Imag(pole). In this case,
% -2r_k*cos(phi_k)=-2*Real{zero), (r_k)^2=
% [Real{zero}]^2+[Imag{zero}]^2,
% -2R_k*cos(Phi_k)=-2*Real{pero), and (R_k)^2=
% [Real{pole}]^2+[Imag{pole}]^2.
% N_k(z) and D_k(z) are given below in the forms
% N_k(z)=ak(1)+ak(2)*z^(-1)+ak(3)*z^(-2) and
% D_k(z)=bk(1)+bk(2)*z^(-1)+bk(3)*z^(-2)
% Note that bk(1)=1 and we implement in practice
% -bk(2) and -bk(3)
%-----------------------------------------------
disp('Second-order blocks:')
a1(1)=1;a1(2)=-2*real(zer(2));
a1(3)=real(zer(2))^2+imag(zer(2))^2;
b1(1)=1;b1(2)=-2*real(pol(2));
b1(3)=real(pol(2))^2+imag(pol(2))^2;
a2(1)=1;a2(2)=-2*real(zer(4));
a2(3)=real(zer(4))^2+imag(zer(4))^2;
b2(1)=1;b2(2)=-2*real(pol(4));
b2(3)=real(pol(4))^2+imag(pol(4))^2;
disp('First block: numerator and then denominator')
a1
b1
disp('Second block: numerator and then denominator')
a2
b2
%-----------------------------------------------
% What is left is the constant to make the passband
% maximum for the amplitude response equal to unity
% It is directly the constant in the numerator in the
% direct form realization.
%-----------------------------------------------
disp('constant')
```

```
B(1)
pause;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% Coefficients for the parallel-form structure
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%
[R,P,K] = residuez(B,A);
%-----------------------------------------------
% Using the command shown below, we can express the
% overall transfer function in the form (partial fraction
% expansion) H(z)=K(1)+R(1)/[1-P(1)z^(-1)]+
% R(2)/[1-P(2)z^(-1)]+R(3)/[1-P(3)z^(-1)]+
% R(4)/[1-P(4)z^(-1)]+R(5)/[1-P(5)z^(-1)]
% Here, P(k) for k=1,2,3,4,5 are the poles of our
% filter. Note that B contains the numerator coefficients
% and A the denominator coefficients of our filter
%-----------------------------------------------
%
[R,P,K] = residuez(B,A);
%
%-----------------------------------------------
% We know now that we have one real pole at z=P(k) with
% the real-valued A(k). However, we don't know the proper
% value of k. Furthermore, we know that there are two
% complex congugate pole pairs. For both pairs, the
% corresponding R's are also complex conjugates.
%-----------------------------------------------
%
[Y,I]=sort(P);
P=P(I);
R=R(I);
%
%-------------------------------------------------
% The above command sorts the poles in such a manner that
% the radii are in the ascending order. This means that
% P(1) contains the real pole. Futhermore, R(1) is the
% the corresponding R. P(2) and P(3) the the innermost
% complex conjugate pole pair (smaller angle) with R(2)
% and R(3) being the corresponding R's. P(4) and P(5)
% are the outermost complex conjugate pole pair (larger
% angle) with R(4) and R(5) being the corresponding R's.
% See page 17 in the lecture notes.
%----------------------------------------------
% Now we are ready to find out the coefficients of the
% parallel-form structure considered on pages 33 and
```

```
% 34 in the lecture notes.
% We express H(z)=K(1)+H_r1(z)+H_1(z)+H_2(z). Here
% H_r1(z)=R(1)/[1-P(1)z^(-1)] is a first-order block
% realizing the real pole at z=P(1).
%-----------------------------------------------
disp('Parallel-Form Realization')
disp('-----------------------')
disp('constant')
K
disp('First-order block:')
disp('Numerator first, then denominator')
cr=real(R(1));
dr(1)=1;dr(2)=-P(1);
cr
dr
%-----------------------------------------------
% H_1(z)=R(2)/[1-P(2)z^(-1)]+R(3)/[1-P(3)z^(-1)]=
% N_1(z)/D_2(z),
% where
% N_1(z)={R(2)[1-P(3)z^(-1)]+R(3)[1-P(2)z^(-1)]
% and
% D_1(z)=[1-P(2)z^(-1)][1-P(3)z^(-1)].
% Here, P(2) and P(3) are a complex conjugate pole pair
% with a smaller angle. Therefore, D_1(z) is the same as
% for the first second-order block in the cascade-form
% realization. R(2) and R(3) are a complex conjugate pair.
% Therefore, N_1(z) is of the form a_0+a_1z^(-1).
% H_2(z) contains the remaining pole pair and in formed
% in the same manner. In the following, we express H_k(z)
% =[ck(1)+ck(2)z^(-1)]/[dk(1)+dk(2)z^(-1)+dk(2)z^(-2)].
%-----------------------------------------------
disp('Second-order blocks:')
d1=b1;d2=b2;
c1=conv(R(2),[1 -P(3)]);
c2=conv(R(4),[1 -P(5)]);
c1=2*real(c1);
c2=2*real(c2);
disp('First block: numerator and then denominator')
c1
d1
disp('Second block: numerator and then denominator')
c2
d2
```

FOURIER TRANSFORMS OF THE INPUT AND OUTPUT SIGNALS

- For signals that are not periodic, we can utilize the fact that the Fourier transforms of the input and output signals are related through

$$Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega}),$$

  where $X(e^{j\omega})$ and $Y(e^{j\omega})$ are the Fourier transforms of the input and output signals, respectively.

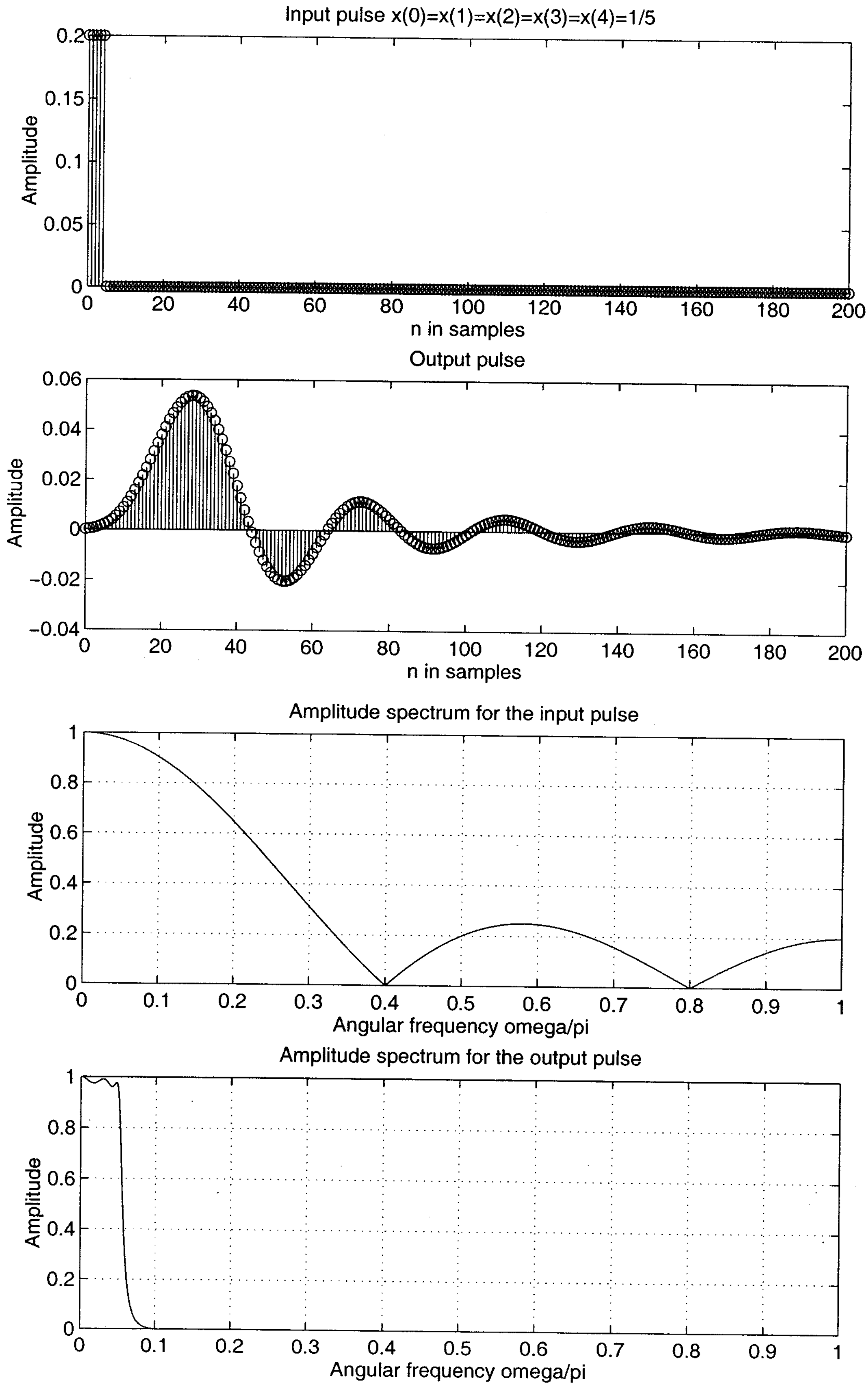
- As examples, we consider filtering two input signals with the aid of the filter considered previously.

- For the first signal, $x[n] = 1/5$ for $n = 0, 1, \cdots, 4$ and zero otherwise.

- The following page shows the input and output signals as well as their amplitude spectra $|X(e^{j\omega})|$ and $|Y(e^{j\omega})|$.

- It is seen that the filter makes the output signal much smoother preserving the frequencies in the passband region.

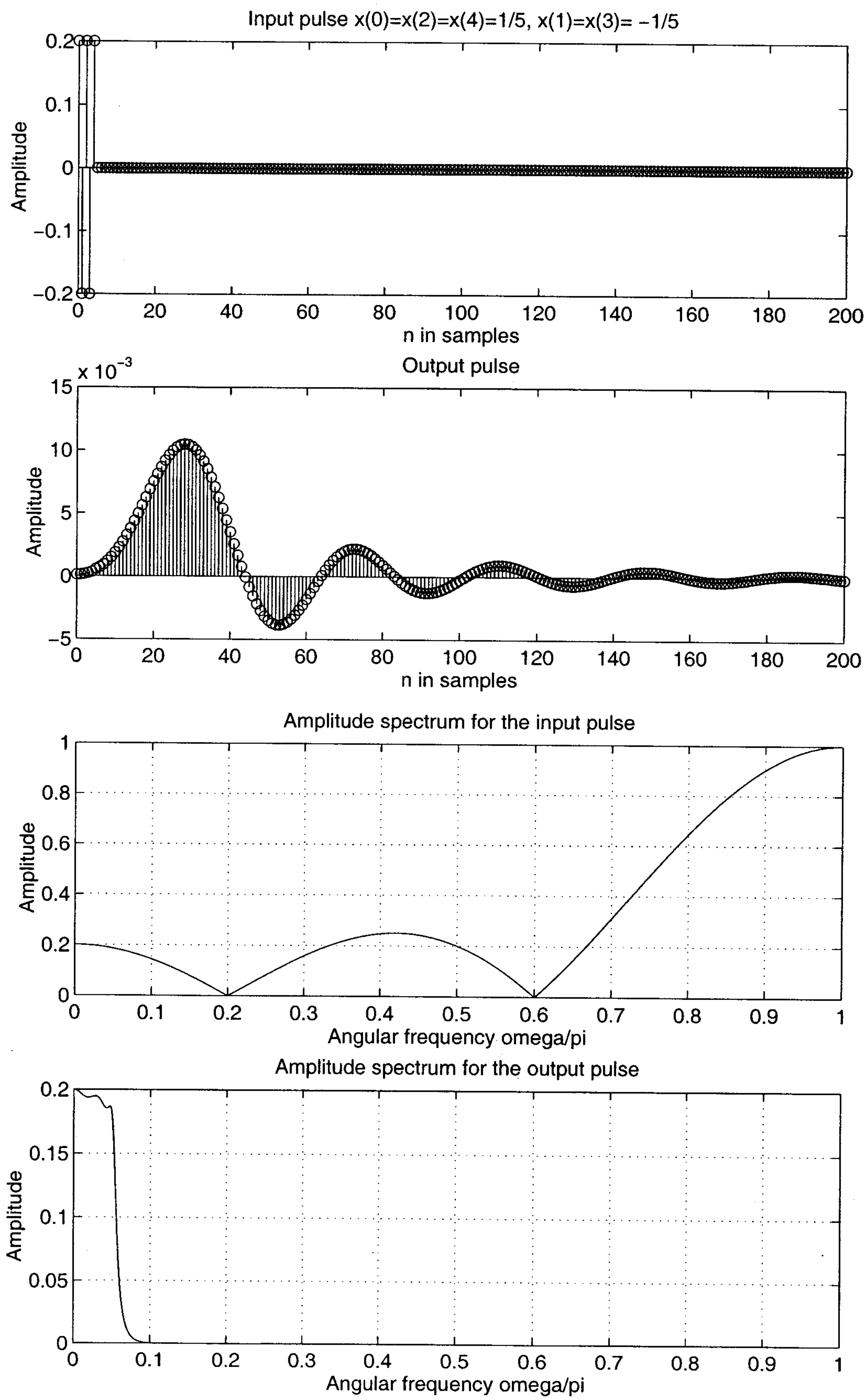
FILTERING OF AN INPUT SIGNAL $x[n] = 1/5$ for $n = 0, 1, \cdots, 4$



FILTERING OF AN INPUT SIGNAL $x[n] = 1/5$ for $n = 0, 2, 4$, $x[n] = -1/5$ for $n = 1, 3$

- For the second signal, $x[n] = 1/5$ for $n = 0, 2, 4$, $x[n] = -1/5$ for $n = 1, 3$, and zero otherwise.
- From the following page, it is seen that most of the energy of $|X(e^{j\omega})|$ is at high frequencies.
- Since this signal is filtered with a lowpass filter, the output signal has very low values.
- In the frequency domain, the output signal contains small energy at low frequencies.

FILTERING OF AN INPUT SIGNAL $x[n] = 1/5$ for $n = 0, 2, 4$, $x[n] = -1/5$ for $n = 1, 3$



VARIOUS FILTER STRUCTURES

- The same filter transfer function is implementable using several structures.
- In the sequel, several structures are introduced even though our course concentrates on a few structures.
- The purpose is to make the reader aware of the fact in practical implementations using either VLSI circuits or signal processors it is very crucial to select a proper filter structure.
- Furthermore, it is desired to show how many alternatives there are to realize the same transfer function.
- Please take just a short look at different alternatives.
- Some of the following structures are considered in more details in courses "Digital Linear Filtering II" and "System Level DSP Algorithms".

CASCADE-FORM STRUCTURES

- Consider a transfer function

$$H(z) = \frac{\sum_{k=0}^{N} a_k z^{-k}}{1 - \sum_{k=1}^{N} b_k z^{-k}}.$$

- This $H(z)$ can be expressed as

$$H(z) = a_0 \frac{\prod_{k=1}^{N}(1 - c_k z^{-1})}{\prod_{k=1}^{N}(1 - d_k z^{-1})}.$$

- Assuming that there are K_1 real zeros and poles, denoted by α_k and β_k, and K_2 complex conjugate zero and pole pairs, denoted by $r_k e^{\pm j\phi_k}$ and $R_k e^{\pm j\Phi_k}$, $H(z)$ can be rewritten as

$$H(z) = a_0 [\prod_{k=1}^{K_1} H_k^{(1)}(z)][\prod_{k=1}^{K_2} H_k^{(2)}(z)],$$

where

$$H_k^{(1)}(z) = \frac{1 + a_{1k}^{(1)} z^{-1}}{1 - b_{1k}^{(1)} z^{-1}}$$

and

$$H_k^{(2)}(z) = \frac{1 + a_{1k}^{(2)} z^{-1} + a_{2k}^{(2)} z^{-2}}{1 - b_{1k}^{(2)} z^{-1} - b_{2k}^{(2)} z^{-2}}$$

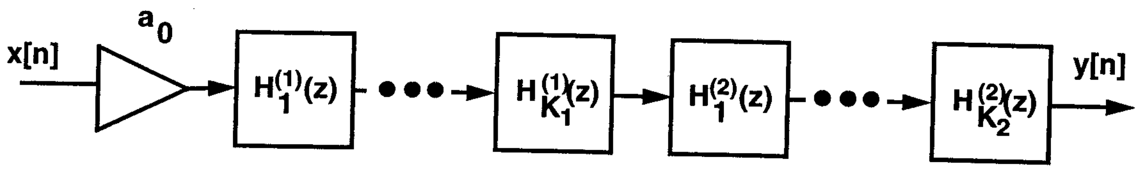
with

$$a_{1k}^{(1)} = -\alpha_k, \quad b_{1k}^{(1)} = \beta_k$$

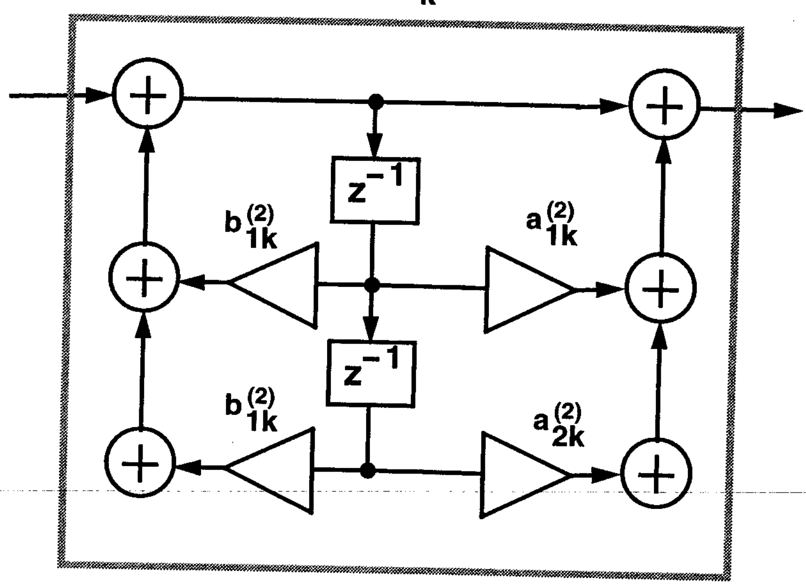
$$a_{1k}^{(2)} = -2r_k \cos\phi_k, \quad a_{2k}^{(2)} = r_k^2$$

$$b_{1k}^{(2)} = 2R_k \cos\Phi_k, \quad b_{2k}^{(2)} = -R_k^2.$$

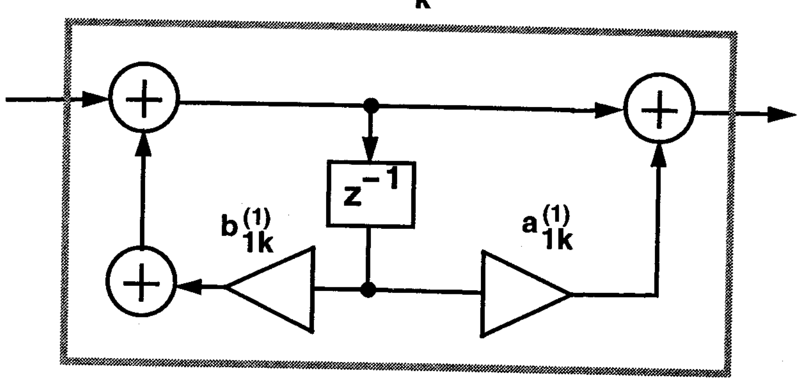
RESULTING STRUCTURE



H(2)k(z)



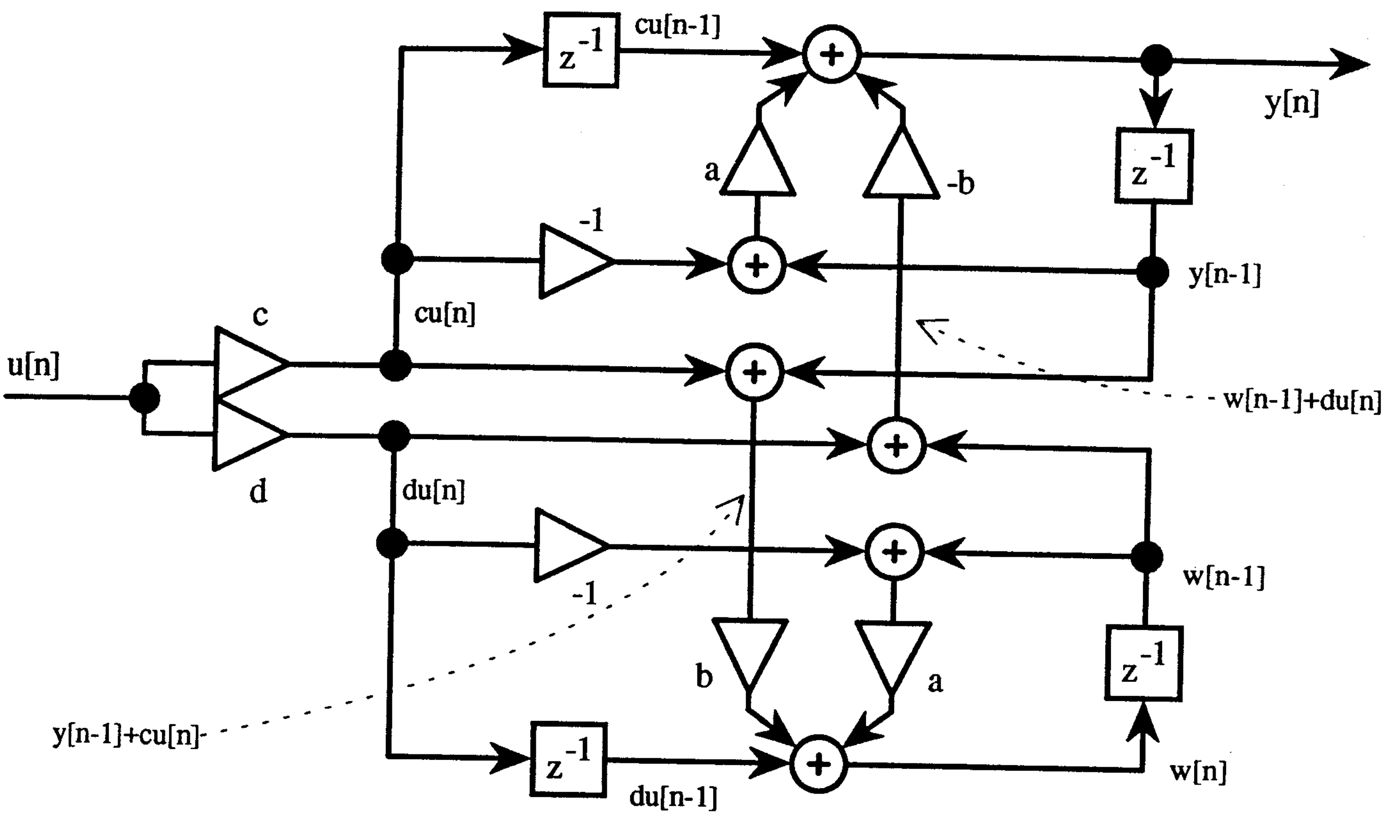
H(1)k(z)

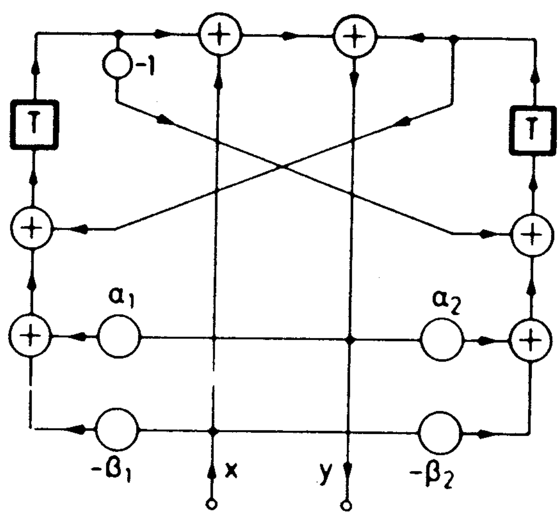


Generation of the cascade-form structure using a matlab file

- The matlab file of pages 19–27 shows how for a fifth-order elliptic filter a cascade-form structure can be generated.
- Please study the file carefully, especially the matlab commands.
- The same file can also be used for generating the parallel-form structure of pages 38 and 39.

OTHER SECOND-ORDER BLOCKS: FIRST MODIFIED COUPLED-FORM, THEN WAVE DIGITAL BLOCK





$$\alpha_1 = (B_2 - B_1 - 1)/2$$

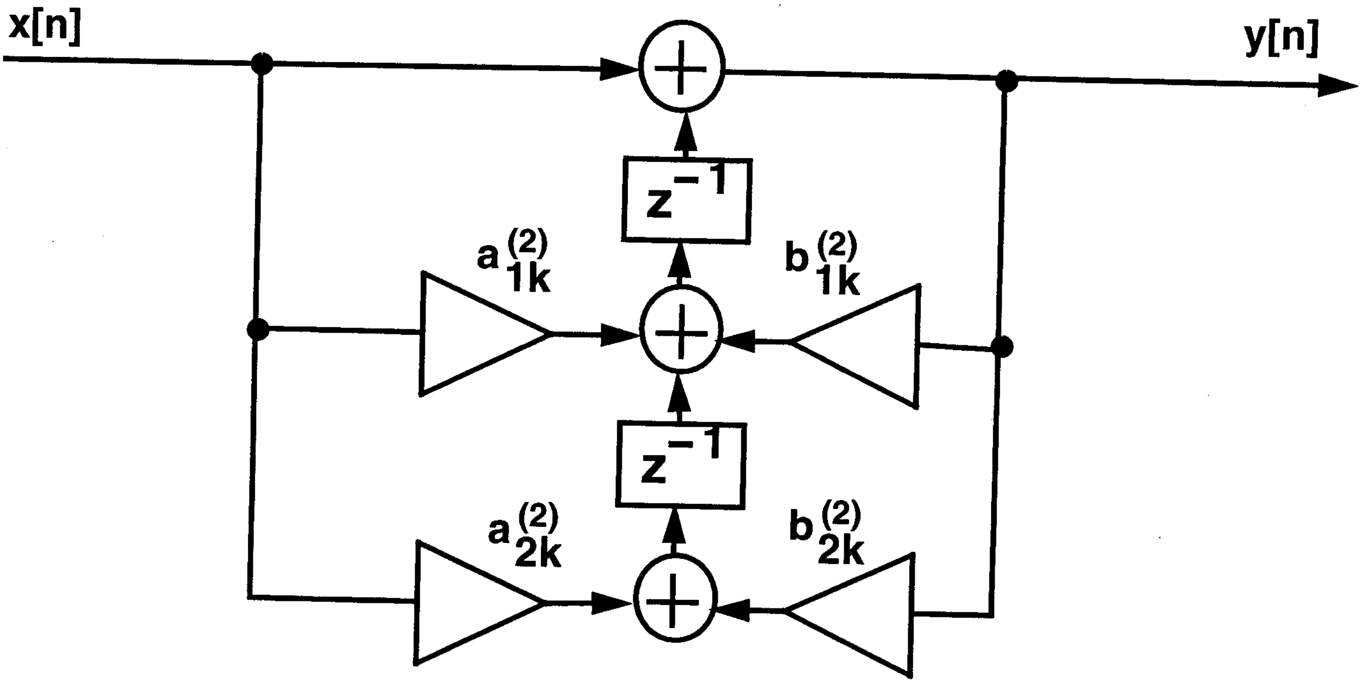
$$\alpha_2 = (1 - B_1 - B_2)/2$$

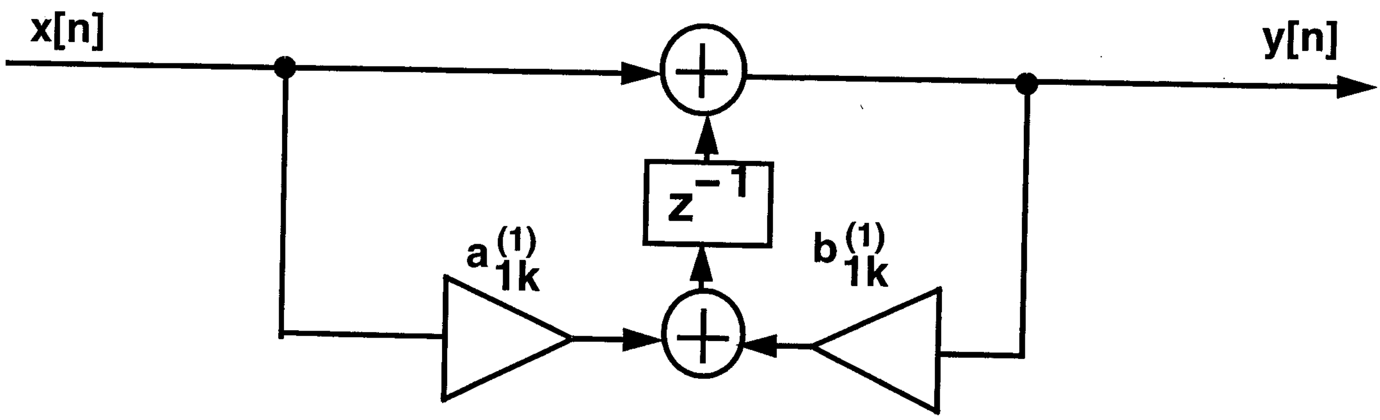
$$\beta_1 = (A_2 - A_1 - A_0)/2A_0$$

$$\beta_2 = (A_0 - A_2 - A_1)/2A_0$$

Fig. 64. Generally usable second-order section for which all stability problems (except that under looped conditions) can be solved in the same way as for WDFs.

USEFUL SECOND-ORDER AND FIRST-OTHER SECTIONS: TRANSPOSED DIRECT-FORM II BLOCKS





PARALLEL FORM STRUCTURES

- Using the partial fraction expansion, $H(z)$ is expressible as

$$H(z) = B_0 + \sum_{k=0}^{N} \frac{A_k}{1 - d_k z^{-1}}.$$

- Assuming that there are K_1 real poles, denoted by β_k, and the corresponding A_k is denoted by β_k as well as there are K_2 complex pole pairs, denoted by $R_k e^{\pm j\Phi_k}$ and the corresponding A_k's are denoted by $r_k e^{\pm j\phi_k}$, $H(z)$ can be rewritten as

$$H(z) = B_0 + \sum_{k=1}^{K_1} H_k^{(1)}(z) + \sum_{k=1}^{K_2} H_k^{(2)}(z)],$$

where

$$B_0 = -a_N / b_N$$

$$H_k^{(1)}(z) = \frac{a_{0k}^{(1)}}{1 - b_{1k}^{(1)} z^{-1}}$$

and

$$H_k^{(2)}(z) = \frac{a_{0k}^{(2)} + a_{1k}^{(2)} z^{-1}}{1 - b_{1k}^{(2)} z^{-1} - b_{2k}^{(2)} z^{-2}}$$

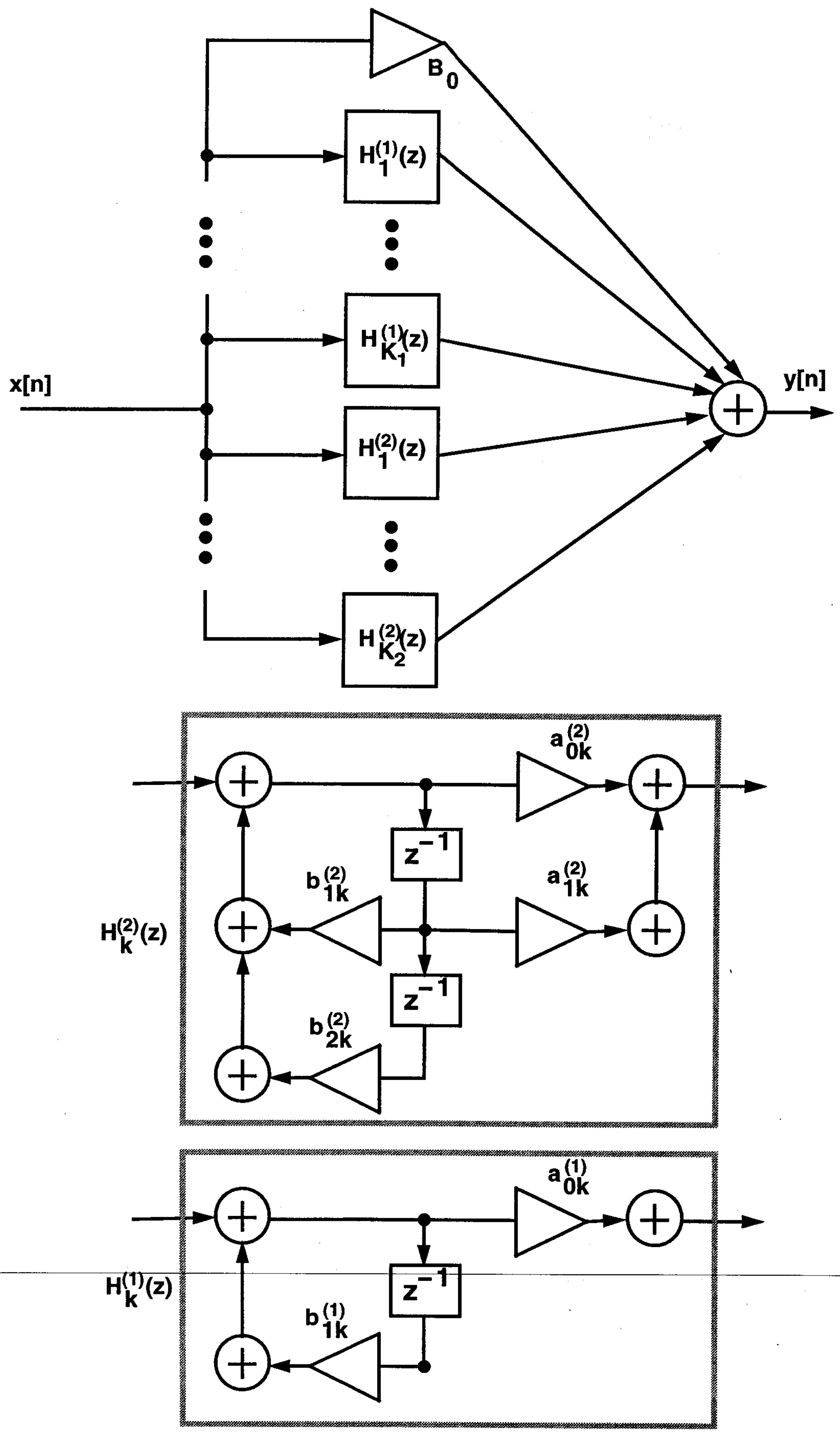
with

$$a_{1k}^{(1)} = \alpha_k, \quad b_{1k}^{(1)} = \beta_k$$

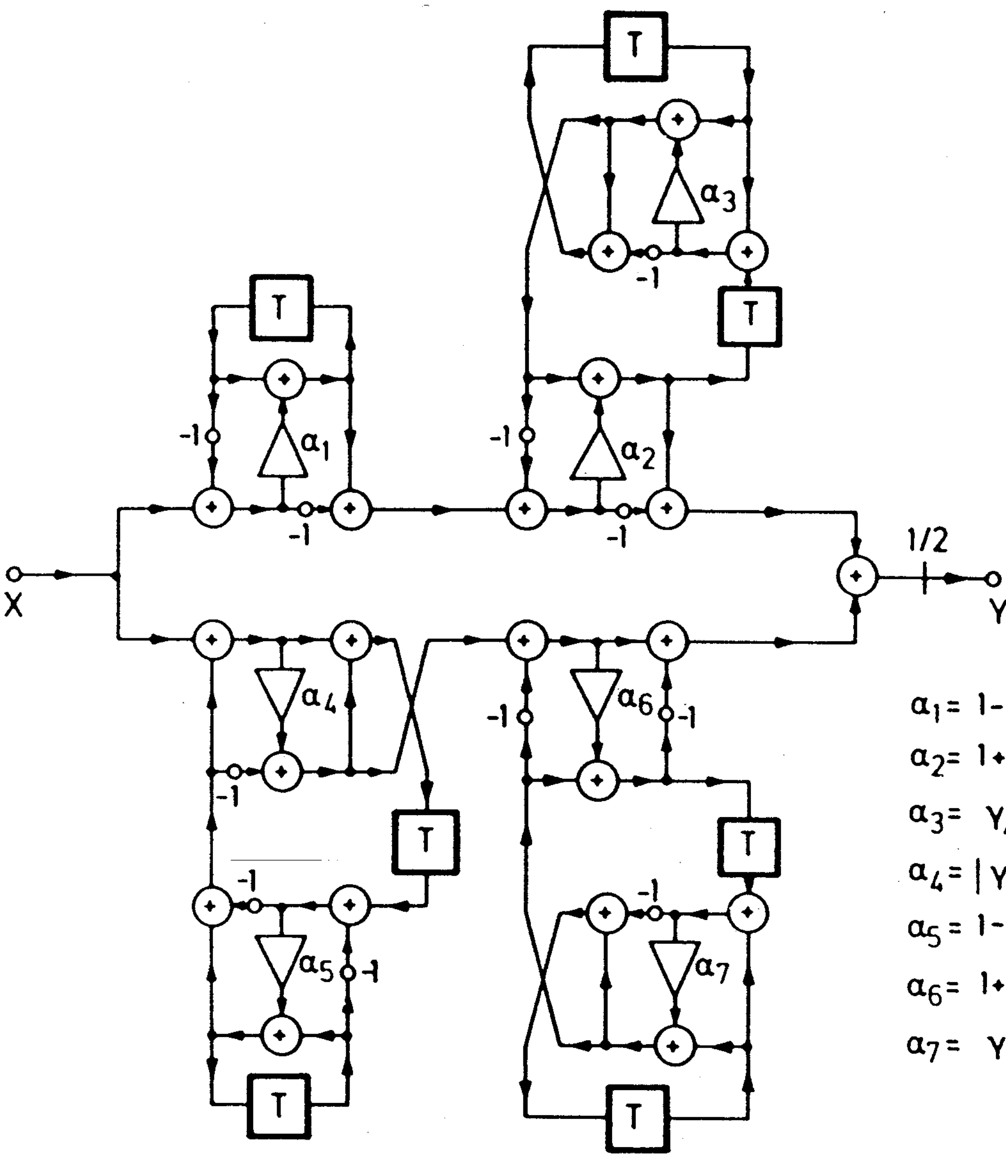
$$a_{0k}^{(2)} = 2r_k \cos \phi_k, \quad a_{1k}^{(2)} = -2r_k R_k \cos(\phi_k - \Phi_k)$$

$$b_{1k}^{(2)} = 2R_k \cos \Phi_k, \quad b_{2k}^{(2)} = -R_k^2.$$

RESULTING STRUCTURE



WAVE LATTICE FILTERS: PARALLEL CONNECTION OF TWO ALLPASS SECTIONS



$\alpha_1 = 1 - \gamma_0 = 0.487\,1$

$\alpha_2 = 1 + \gamma_3 = 0.331\,3$

$\alpha_3 = \gamma_4 = 0.334\,2$

$\alpha_4 = |\gamma_1| = 0.404\,4$

$\alpha_5 = 1 - \gamma_2 = 0.392\,2$

$\alpha_6 = 1 + \gamma_5 = 0.103\,8$

$\alpha_7 = \gamma_6 = 0.206\,7$

WAVE LADDER FILTERS

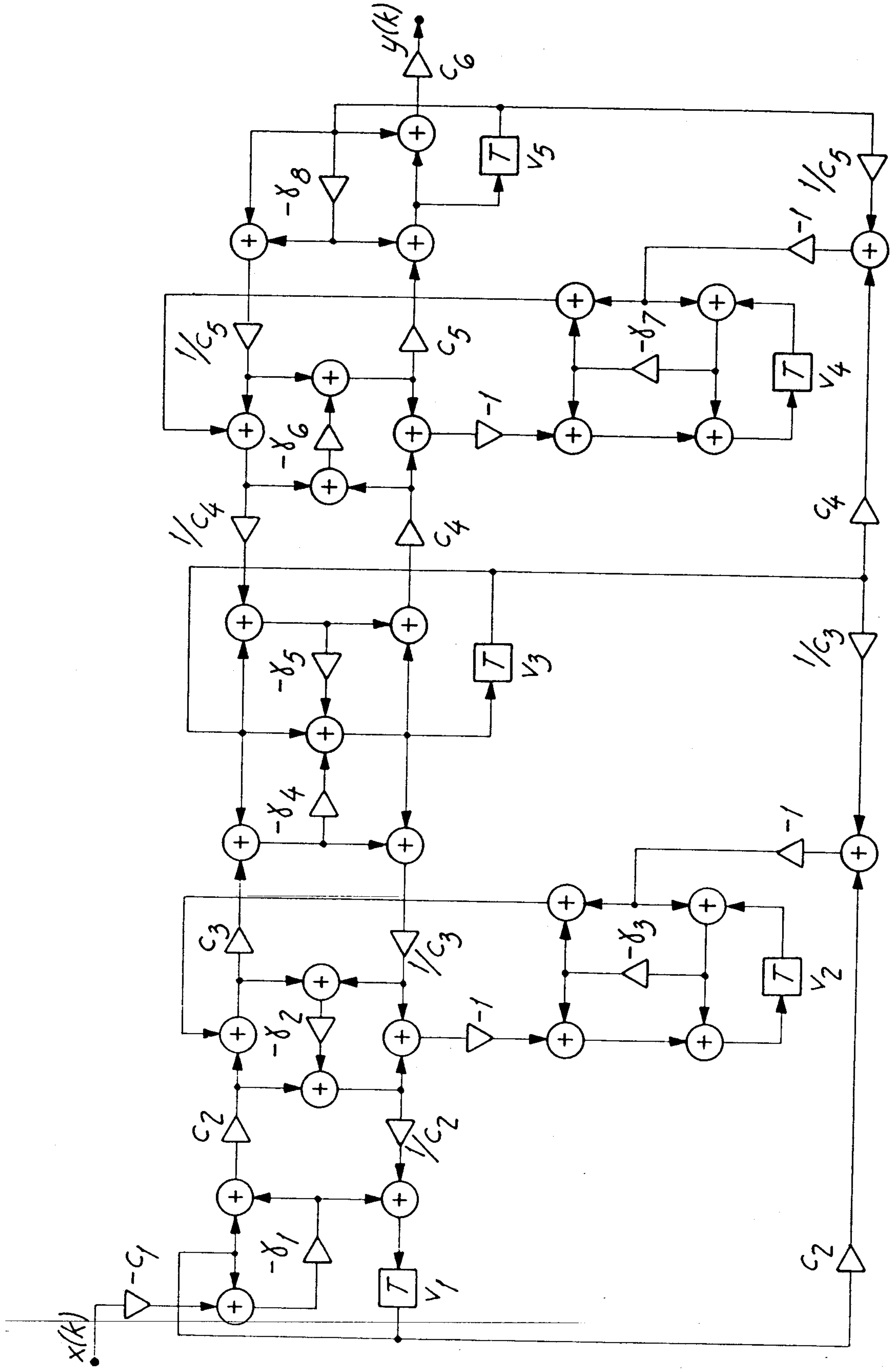


Fig. 4.6 Wave-flow diagram for the canonic wave digital ladder filter.

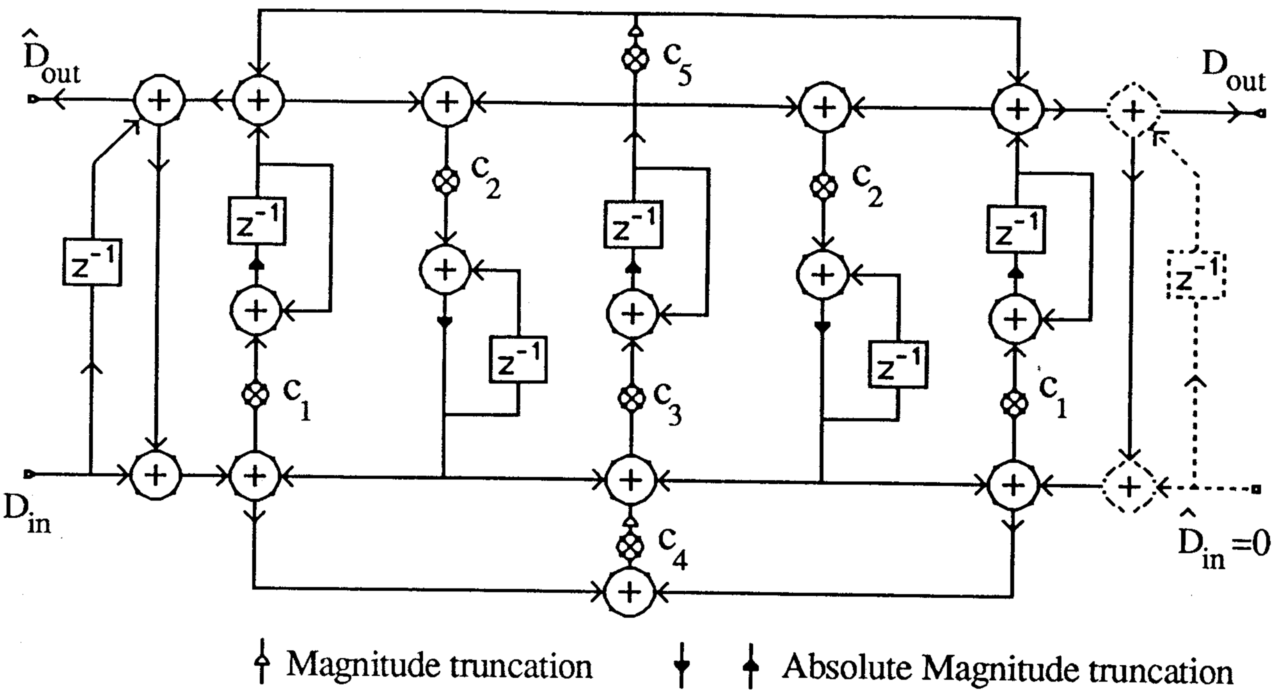


Fig.7. Symmetric LDI ladder structure for realizing a fifth-order filter. For the elliptic filter, all the coefficients are used. For Example 1 in Section IV, $c_3 = c_4 = 0$, and for Example 2, $c_4 = 0$. Magnitude and absolute magnitude trunctions are defined as quantization operations Q( ) characterized by $|Q(x)| \le |x|$ and $|Q(x)| < |x|$, respectively. In practice, the parts given using a dashed line are not used.

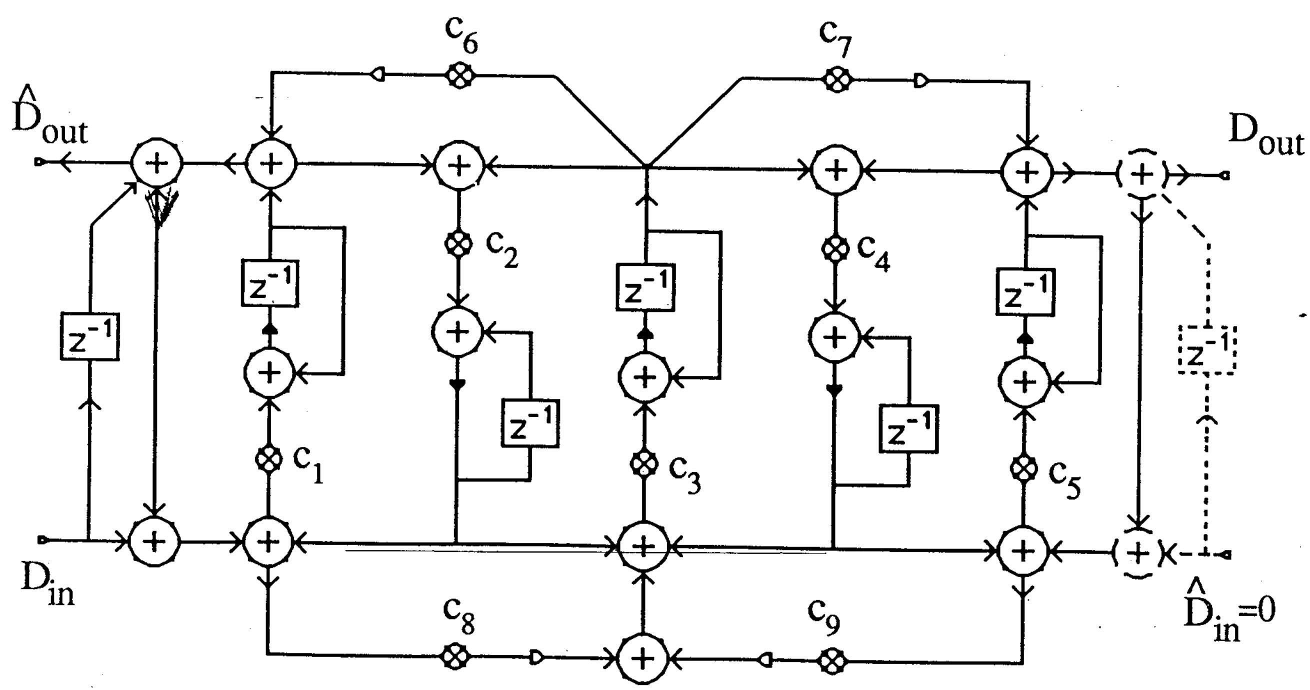


Fig. 8. Nonsymmetric LDI ladder structure for realizing a fifth-order elliptic filter.

STATE-SPACE STRUCTURES

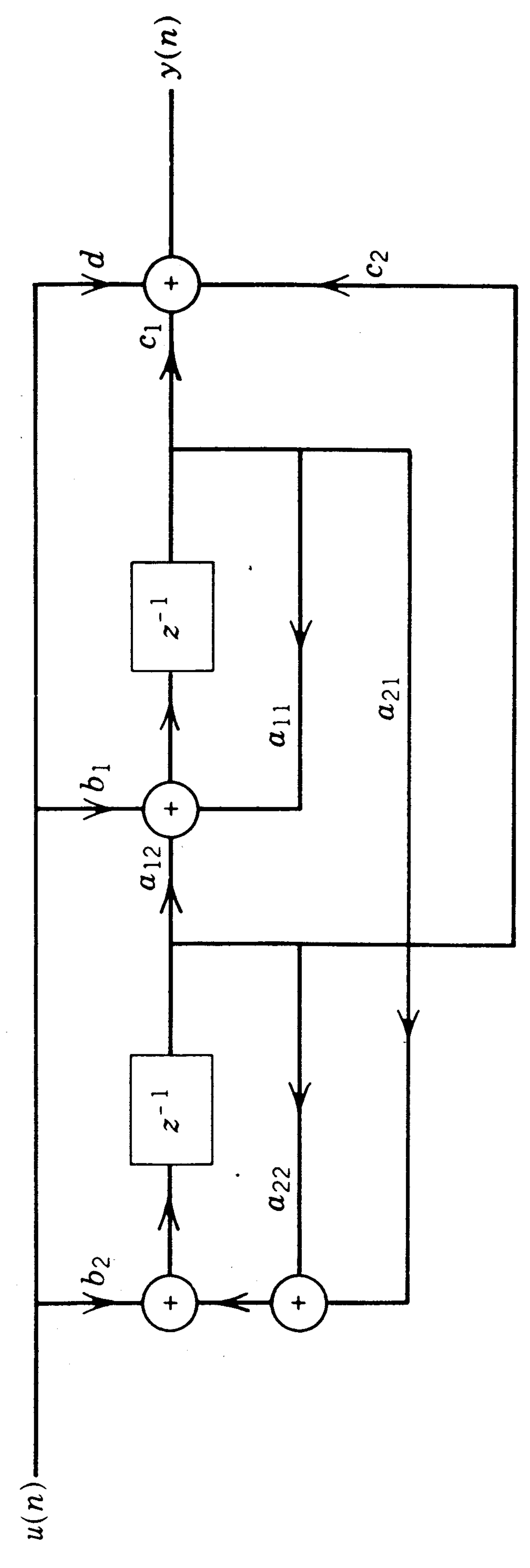
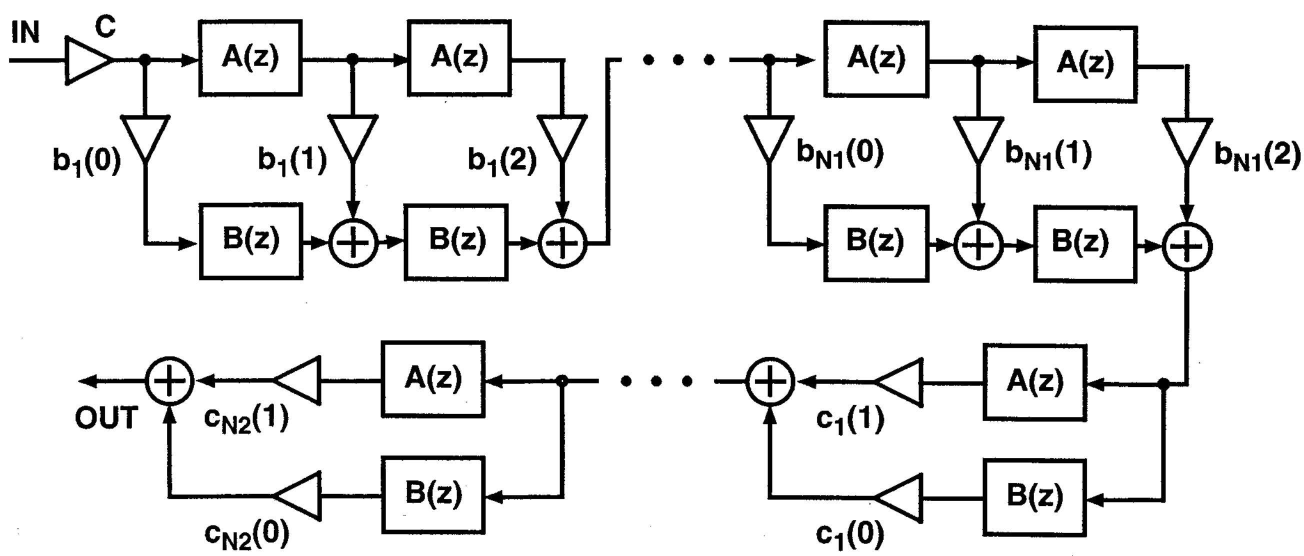


FIGURE 8.7 State-Variable Structure.

SPECIAL STRUCTURES: ALL COEFFIECIENT VALUES CAN BE EXPRESSED IN THE FORM $\pm 2^{-P_1} \pm 2^{-P_2} \pm 2^{-P_3}$

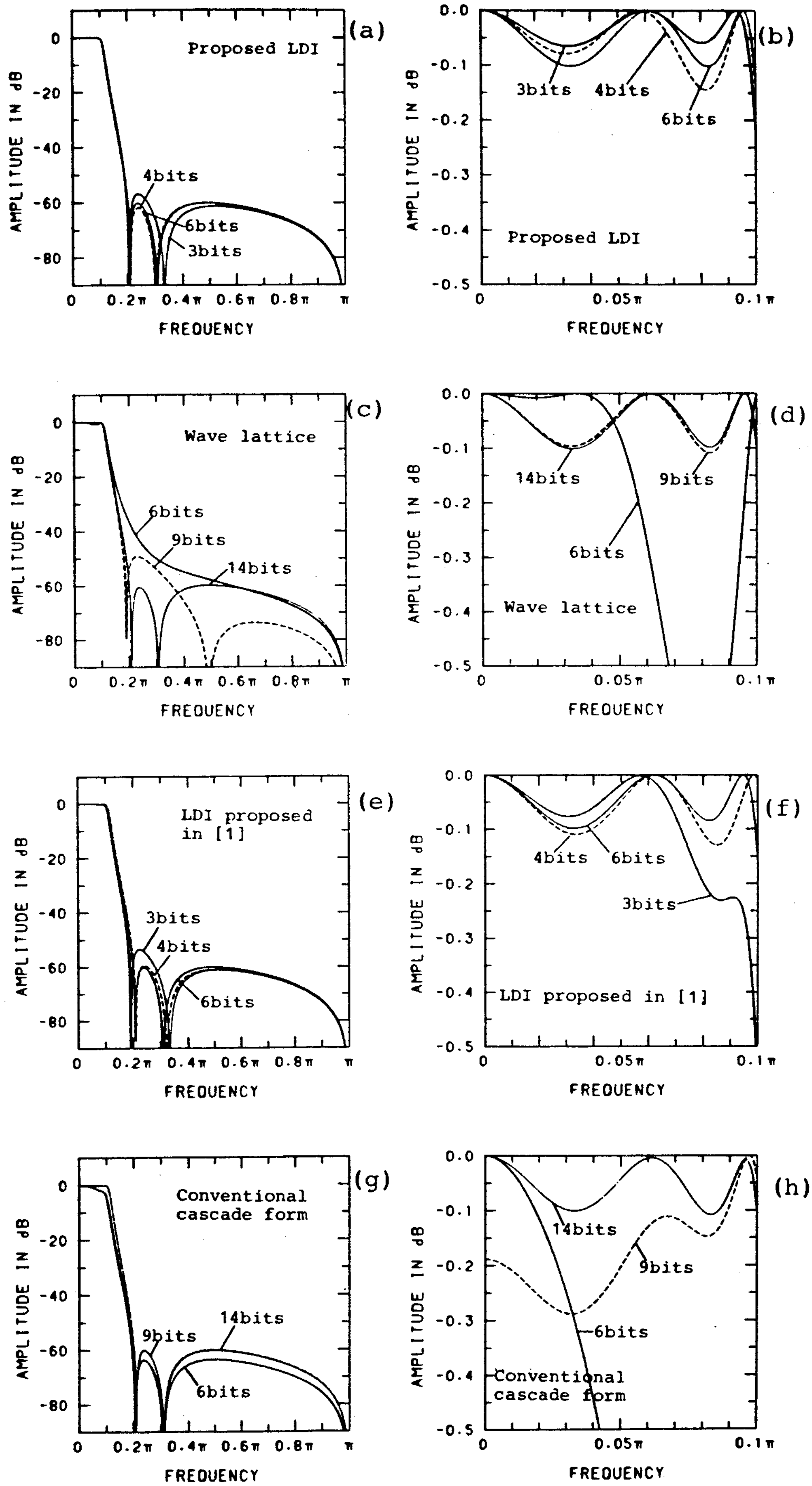
- Multiplier-free filters since all the coefficient values can be implemented by shift and add and/or subtract operations.
- These filters are attractive in VLSI implementations since there is no need to use a costly multiplier element.
- Structure for IIR filters, where $A(z)$ and $B(z)$ are allpass filters (to be considered in the course "System Level DSP Algorithms").



SELECTION AMONG DIFFERENT FILTER STRUCTURES

- Finite wordlength effects, such as output noise variance due to multiplication roundoff errors and limit cycle oscillations, determine the data wordlength in internal calculations to keep these effects under the given limits.
- Coefficient sensitivity determines the number of bits required to keep the response within the given criteria.
- Computer program: Direct-form filter with double precision arithmetic are generally the most effective.
- Signal processors: the length of the code is important.
- VLSI-circuits: the small silicon area, low power consumption, and high achievable sampling rate are important; Multiplierless filters are preferred since the general multiplier element is costly.

EXAMPLE: COEFFICIENT SENSITIVITY FOR SOME FILTER STRUCTURES



IIR FILTER STRUCTURES TO BE STUDIED IN THIS COURSE

- Direct-form structures
- Cascade-form structures with direct-form II blocks or transposed direct-form II blocks
- Parallel connections of two allpass filters are very effective and will be considered in details in the course "Digital Linear Filtering II". Lattice wave digital filters of page 40 are examples of these structures.
- It should be pointed out that the parallel form structures of pages 38 and 39 are not good at all; they are just of academic interest.
- Students interested in other structures please contact the lecturer.

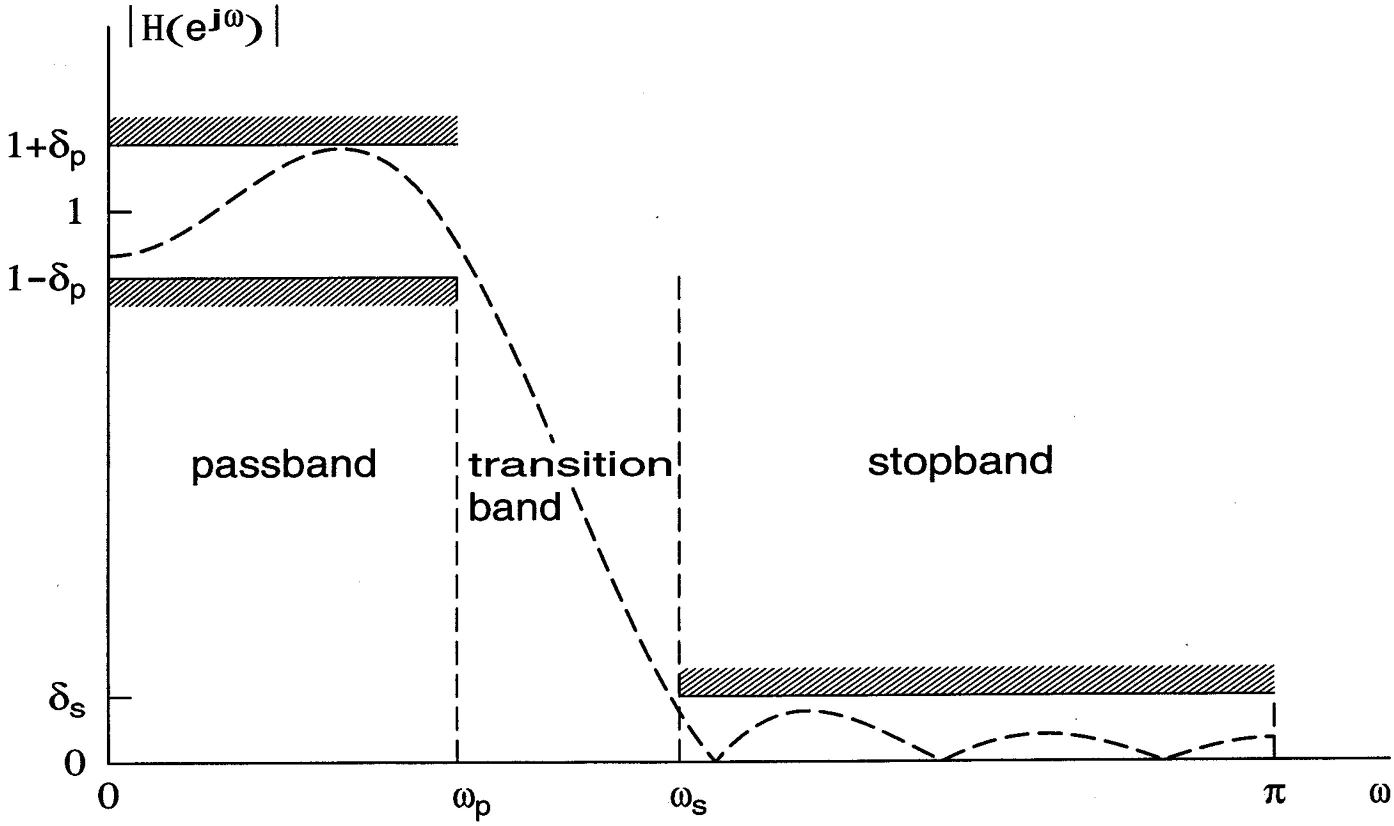
DIGITAL FILTER DESIGN PROCESS

Digital filter design involves usually the following basic steps:

1. Determine a desired response or a set of desired responses (e.g., a desired magnitude response and/or a desired phase response)
2. Select a class of filters for approximating the desired response(s) (e.g., linear-phase FIR filters or IIR filters being implementable as a parallel connection of two allpass filters).
3. Establish a criterion of 'goodness' for the response(s) of a filter in the selected class compared to the desired response(s).
4. Develop a method for finding the best member in the filter class.
5. Synthesize the best filter using a proper structure and a proper implementation form, e.g. using a computer program, a signal processor, or a VLSI chip.
6. Analyze the filter performance (especially, the finite-wordlegth effects are of great importance).

AMPLITUDE SPECIFICATIONS

Conventional lowpass specifications



$$1 - \delta_p \leq |H(e^{j\omega})| \leq 1 + \delta_p \quad \text{for} \quad \omega \in [0, \omega_p]$$

$$|H(e^{j\omega})| \leq \delta_s \quad \text{for} \quad \omega \in [\omega_s, \pi]$$

- ω is related to the "real frequency" f via

$$\omega = 2\pi f / f_s,$$

where f_s is the sampling frequency.

- If $f_s = 20$ kHz and the band edges are 4 kHz and 5 kHz, then $\omega_p = 0.4\pi$ and $\omega_s = 0.5\pi$.

- Sometimes the edges are given in terms of the *normalized frequency* defined by

$$f_{norm} = f/f_s.$$

- In the above example, the normalized passband and stopband edges are 0.2 and 0.25.
- Usually, the amplitudes of allowable ripples are given logarithmically (i.e. in decibels) in terms of the maximum passband variation and the minimum stopband attenuation, which are given by

$$A_p = 20\log_{10}(\frac{1+\delta_p}{1-\delta_p}) \quad \text{dB}$$

and

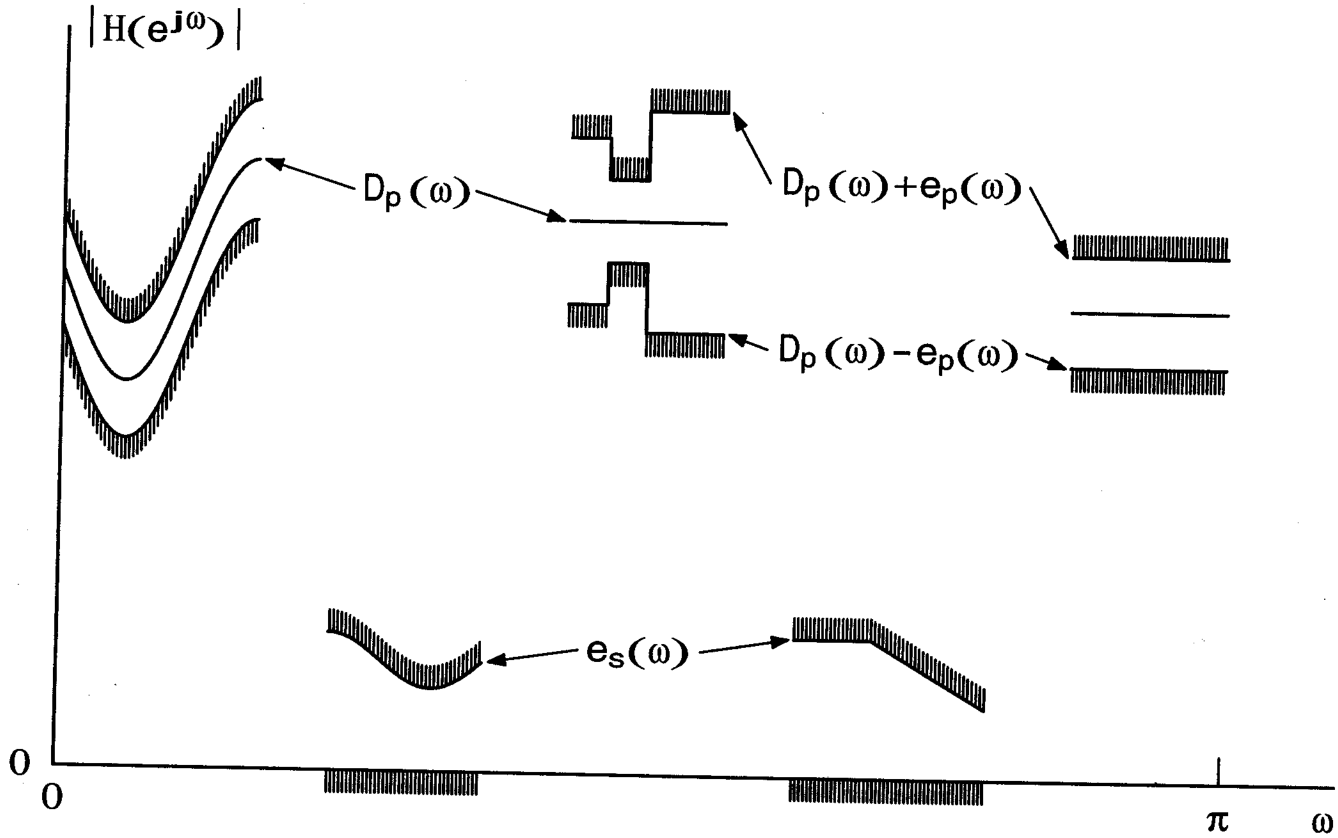
$$A_s = -20\log_{10}(\delta_s) \quad \text{dB},$$

respectively.

- Note that both of these quantities are positive.

Example: $\delta_p = 0.01$, $\delta_s = 0.001 \Rightarrow A_p = 0.17$ dB and $A_s = 60$ dB.

GENERAL AMPLITUDE SPECIFICATIONS



- The criteria can be written as:

$$D_p(\omega) - e_p(\omega) \leq |H(e^{j\omega})| \leq D_p(\omega) + e_p(\omega) \quad \text{for} \quad \omega \in X_p$$

$$|H(e^{j\omega})| \leq e_s(\omega) \quad \text{for} \quad \omega \in X_s,$$

- X_p and X_s are the passband and stopband regions.
- $e_p(\omega)$ is the permissible deviation from the desired passband response $D_p(\omega)$.
- $e_s(\omega)$ is the allowable deviation from zero in the stopband region.

ALTERNATIVE FORM

- Using the substitutions

$$e_p(\omega) = 1/W_p(\omega), \quad e_s(\omega) = 1/W_s(\omega),$$

the specifications of the previous transparency are expressible in the forms:

$$-1/W_p(\omega) \le [|H(e^{j\omega})| - D_p(\omega)] \le 1/W_p(\omega) \text{ for } \omega \in X_p$$

$$|H(e^{j\omega})| \le 1/W_s(\omega) \quad \text{for} \quad \omega \in X_s,$$

$$-1 \le W_p(\omega)[|H(e^{j\omega})| - D_p(\omega)] \le 1 \quad \text{for} \quad \omega \in X_p$$

$$W_s(\omega)|H(e^{j\omega})| \le 1 \quad \text{for} \quad \omega \in X_s,$$

or

$$|W_p(\omega)[|H(e^{j\omega})| - D_p(\omega)]| \le 1 \quad \text{for} \quad \omega \in X_p$$

$$|W_s(\omega)|H(e^{j\omega})|| \le 1 \quad \text{for} \quad \omega \in X_s.$$

DESIRED FORM

- Finally, these criteria can be combined to give the following form which is useful in many filter design techniques:

$$|E(\omega)| \leq \widehat{\epsilon} \quad \text{for} \quad \omega \in X = X_p \bigcup X_s,$$

where

$$E(\omega) = W(\omega)[|H(e^{j\omega})| - D(\omega)]$$

with

$$\widehat{\epsilon} = 1,$$

$$D(\omega) = \begin{cases} D_p(\omega) & \text{for } \omega \in X_p \\ 0 & \text{for } \omega \in X_s \end{cases}$$

and

$$W(\omega) = \begin{cases} W_p(\omega) & \text{for } \omega \in X_p \\ W_s(\omega) & \text{for } \omega \in X_s. \end{cases}$$

- $D(\omega)$ and $W(\omega)$ are called the *desired function* and the *weighting function*, respectively, and $E(\omega)$ is the *weighted error function.*

EXAMPLE

- In the bandpass case, the criteria are usually stated as

$$1 - \delta_p \leq |H(e^{j\omega}) \leq 1 + \delta_p \quad \text{for} \quad \omega \in [\omega_{p1}, \omega_{p2}]$$

$$-\delta_s \leq |H(e^{j\omega})| \leq \delta_s \quad \text{for} \quad \omega \in [0, \omega_{s1}] \bigcup [\omega_{s2}, \pi].$$

- These specifications can be written in the above form using

$$X = [0, \omega_{s1}] \bigcup [\omega_{p1}, \omega_{p2}] \bigcup [\omega_{s2}, \pi]$$

$$D(\omega) = \begin{cases} 1 & \text{for } \omega \in [\omega_{p1}, \omega_{p2}] \\ 0 & \text{for } \omega \in [0, \omega_{s1}] \bigcup [\omega_{s2}, \pi] \end{cases}$$

$$W(\omega) = \begin{cases} 1/\delta_p & \text{for } \omega \in [\omega_{p1}, \omega_{p2}] \\ 1/\delta_s & \text{for } \omega \in [0, \omega_{s1}] \bigcup [\omega_{s2}, \pi] \end{cases}$$

and

$$\widehat{\epsilon} = 1.$$

PHASE APPROXIMATIONS

- In some applications, it is necessary to preserve the shape of the input signal.
- This is achieved if the phase response $\arg H(e^{j\omega})$ approximates in the passband $[0, \omega_p]$ the linear curve

$$\phi(\omega) = -\tau_0 \omega + \tau_1,$$

where τ_0 and τ_1 can be freely chosen.
- The criteria are usually given in terms of the *group delay* response

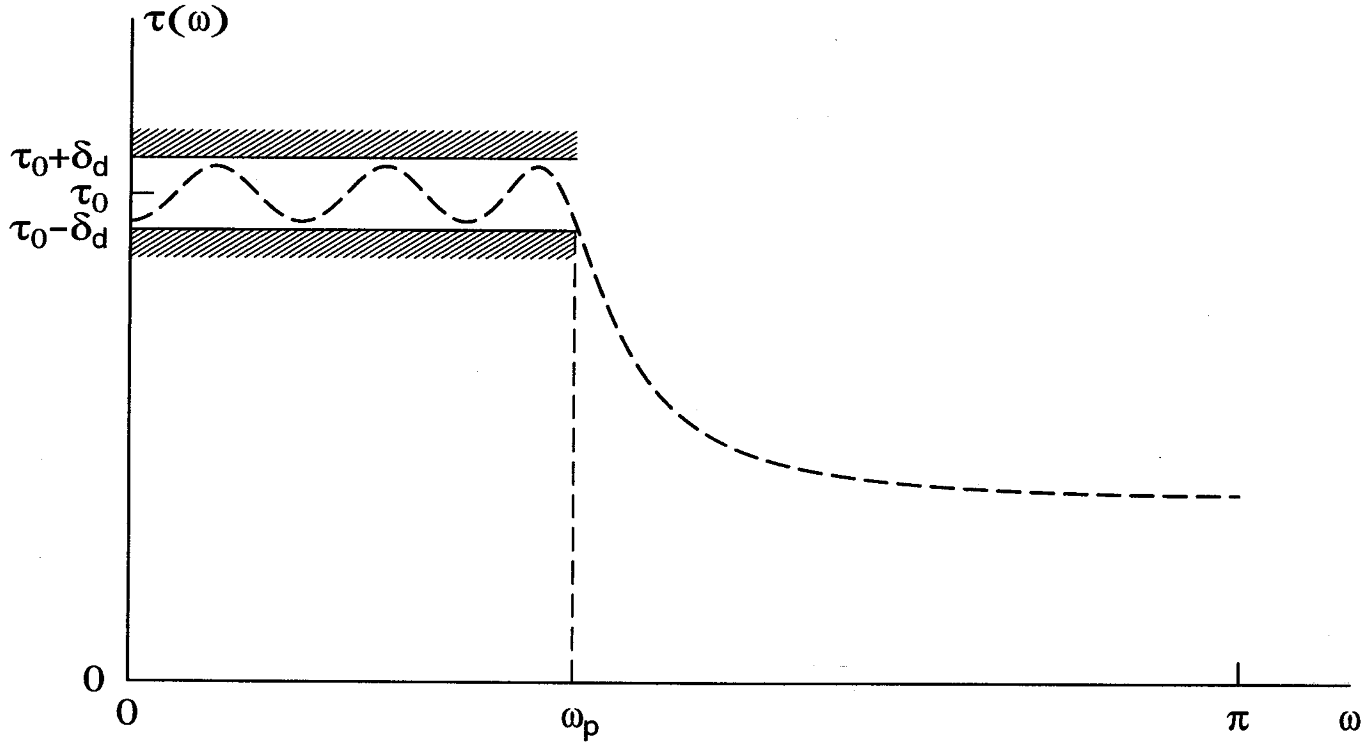
$$\tau_g(\omega) = -\frac{d \arg H(e^{j\omega})}{d\omega}$$

or *phase delay* response

$$\tau_p(\omega) = -\frac{\arg H(e^{j\omega})}{\omega}.$$

- These responses have simpler representation forms and are easier to interpret.
- $\tau_p(\omega_0)$ gives directly the delay caused by the filter for a sinusoidal signal of frequency ω_0.

- If the input signal is periodic or approximately periodic, as an electrocardiagram signal, then the phase delay is required to approximate a constant τ_0 with the given tolerance δ_d as shown in the following figure.



- Since the delay of all components is approximately equal, the signal shape is preserved.
- If the signal is not periodic, then, instead of the phase delay, the group delay can be used.
- Note that for a constant phase delay, τ_1 is forced to be zero, whereas for a constant group delay, τ_1 may take any value.

Filtering of an EGC-signal

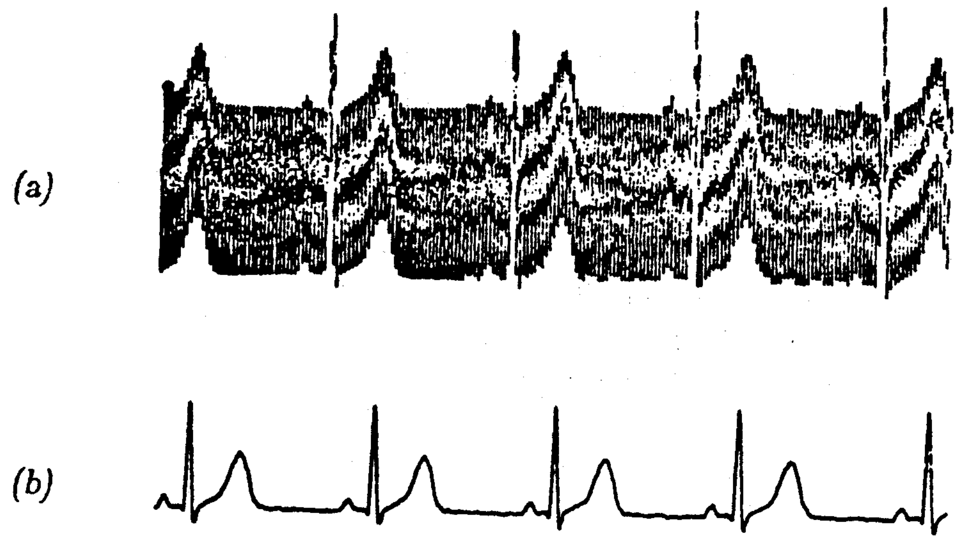


Fig. 10.2. *Performance of an implemented high-pass notch filter when the ECG signal is contaminated by the 50 Hz line frequency interference (heart rate 60 beats/min). (a) Original ECG signal. (b) Filtered ECG signal.*

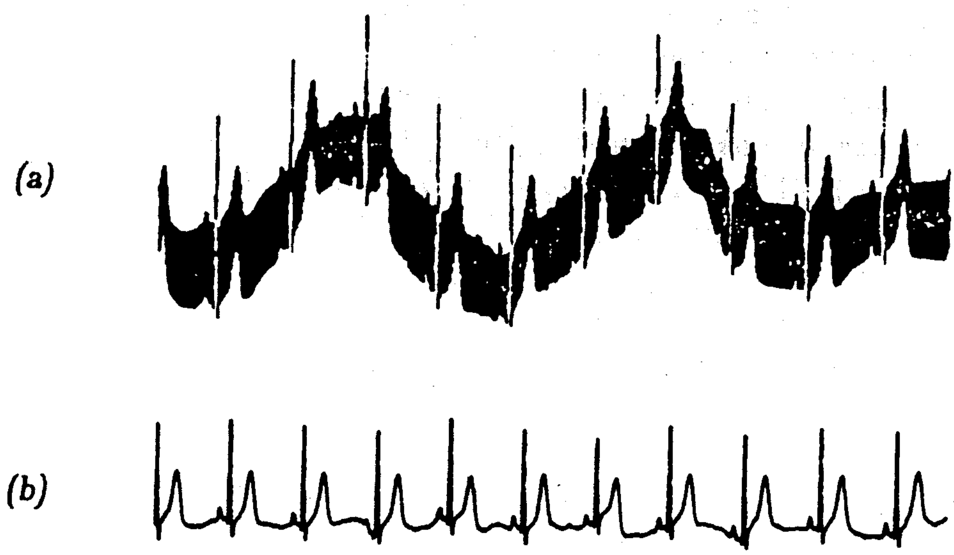


Fig. 10.3. *Elimination of the baseline drift (heart rate 60 beats/min). (a) Original ECG signal. (b) Filtered ECG signal.*

Response of the linear phase filter in use for $f_s = 200$ Hz

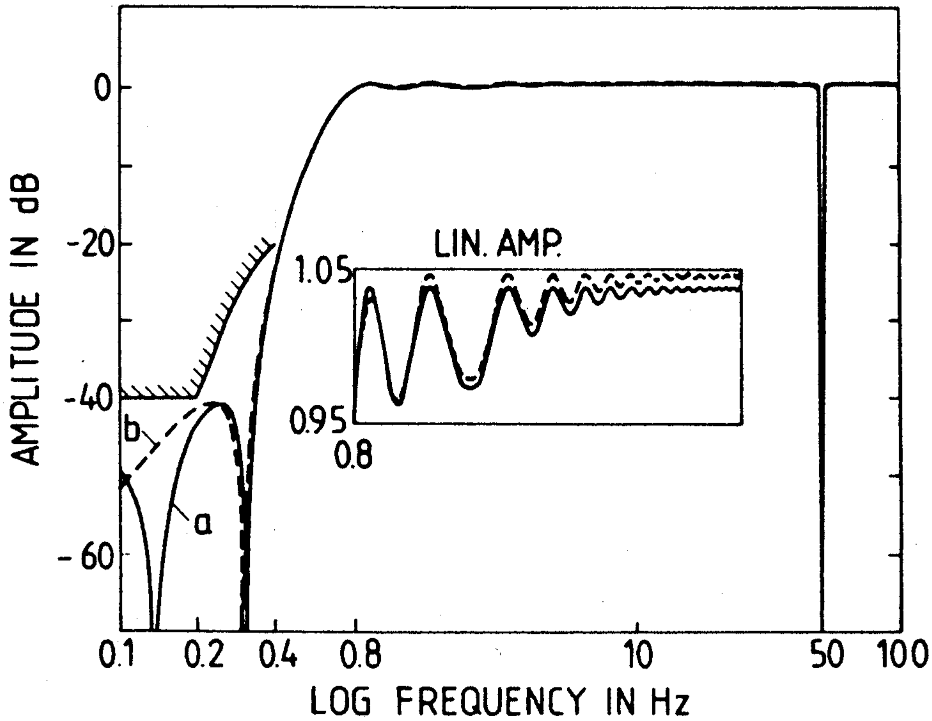


Fig. 8.3. *Amplitude response of the notch filter with infinite precision coefficients (a) and when the coefficients are quantized to 7 bits (b).*

COMMENTS

- If the periodic signal

$$x(n) = \sum_{k=1}^{K} A_k \cos(n(k\omega_0) + \phi_k)$$

is in the frequency range $\omega_1 \leq \omega \leq \omega_2$, the wave-form is presevered if in this range

  - $|H(e^{j\omega})| \approx 1$
  - $\arg H(e^{j\omega}) \approx -\tau_0\omega + r2\pi$ with r being the integer.

- Since each component of $x(n)$ is periodic with periodicity equal to 2π, the output signal is approximately

$$\begin{aligned} y(n) =& \sum_{k=1}^{K} A_k \cos(n(k\omega_0) + \phi_k - k\tau_0\omega_0 + r2\pi) \\ =& \sum_{k=1}^{K} A_k \cos(n(k\omega_0) + \phi_k - k\tau_0\omega_0) \\ =& \sum_{k=1}^{K} A_k \cos((n-\tau_0)(k\omega_0) + \phi_k) \\ =& x(n-\tau_0), \end{aligned}$$

that is, $y(n)$ is a delayed version of $x(n)$.

- If there is an additional phase shift of π or $-\pi$, then $y(n) = -x(n - \tau_0)$ so that in addition to the delay the sign of the periodic signal changes.

GENERAL PHASE SPECIFICATIONS

- The general specifications for the group or phase delay can be stated in terms of the weighted error function as

$$|E_\tau(\omega)| \leq \epsilon_\tau \quad \text{for} \quad \omega \in X_p$$

where

$$E_\tau(\omega) = W_\tau(\omega)[\tau(\omega) - D_\tau(\omega) - \tau_0].$$

- As an example, we consider the group delay equalization of an elliptic filter with the aid of an allpass filter.
- Some characteristics of our example fourth-order elliptic filter are shown on page 65. For this filter, the passband and stopband edges are located at $\omega = 0.5\pi$ and $\omega = 0.6\pi$, respectively. The passband and stopband ripples are 0.5 dB and 32 dB, respectively.
- It is seen that the group delay is far away from a constant in the passband and is monotonously increasing from one sample to nine samples as ω varies from 0 to 0.5π.

- In order to keep the amplitude response the same and to improve the group delay response, we cascade our elliptic filter with an eigth-order allpass filter with the transfer function of the form

$$H_{\text{all}}(z) = N_{\text{all}}(z)/D_{\text{all}}(z),$$

where

$$\begin{aligned} N_{\text{all}}(z) =& a_8 + a_7 z^{-1} + a_6 z^{-2} + a_5 z^{-3} + a_4 z^{-4} \\ &+ a_3 z^{-5} + a_2 z^{-6} + a_1 z^{-7} + z^{-8} \end{aligned}$$

and

$$\begin{aligned} D_{\text{all}}(z) =& 1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4} \\ &+ a_5 z^{-5} + a_6 z^{-6} + a_7 z^{-7} + a_8 z^{-8} \end{aligned}$$

- From the above equation, it is seen that the coefficients of the numerator and denominator are reversed versions of each other. This guarantees that the amplitude response is identically equal to unity and only the delay response (phase response) is changing.
- Furthermore, if the filter has a pole pair at $z = re^{\pm j\phi}$ [a real pole at $z = r$], then it has a zero pair at $z = (1/r)e^{\pm j\phi}$ [a real zero at $z = 1/r$].

- These filters are considered in more details in the end of Part II of these lecture notes.
- When cascading our elliptic filter with an allpass filter the group delay is given by

$$\tau_g(\omega) = \tau_g^{(1)}(\omega) + \tau_g^{(2)}(\omega),$$

where $\tau_g^{(1)}(\omega)$ and $\tau_g^{(2)}(\omega)$ are the group delay responses of the elliptic filter and the allpass delay equalizer, respectively.

- The coefficients of the allpass filter as well as τ_0, the passband average of $\tau_g(\omega)$, are desired to be optimized to minimize

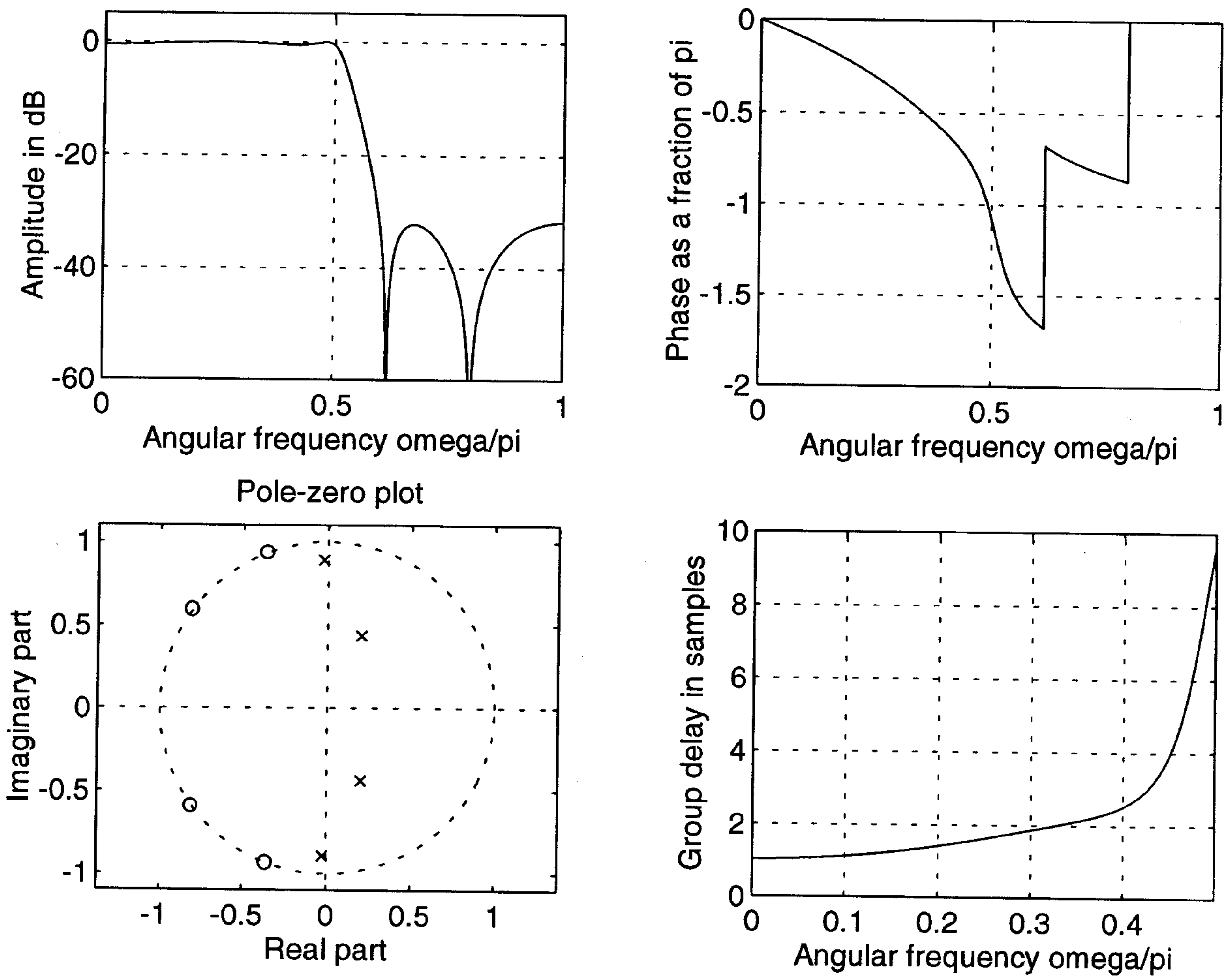
$$\epsilon = \max_{\omega\in[0,\,0.5\pi]} \left|\tau_g^{(1)}(\omega) + \tau_g^{(2)}(\omega) - \tau_0\right|,$$

that is, the maximum absolute deviation of the overall group delay response from the average value τ_0 in the interval $[0,\ 0.5\pi]$ is minimized.

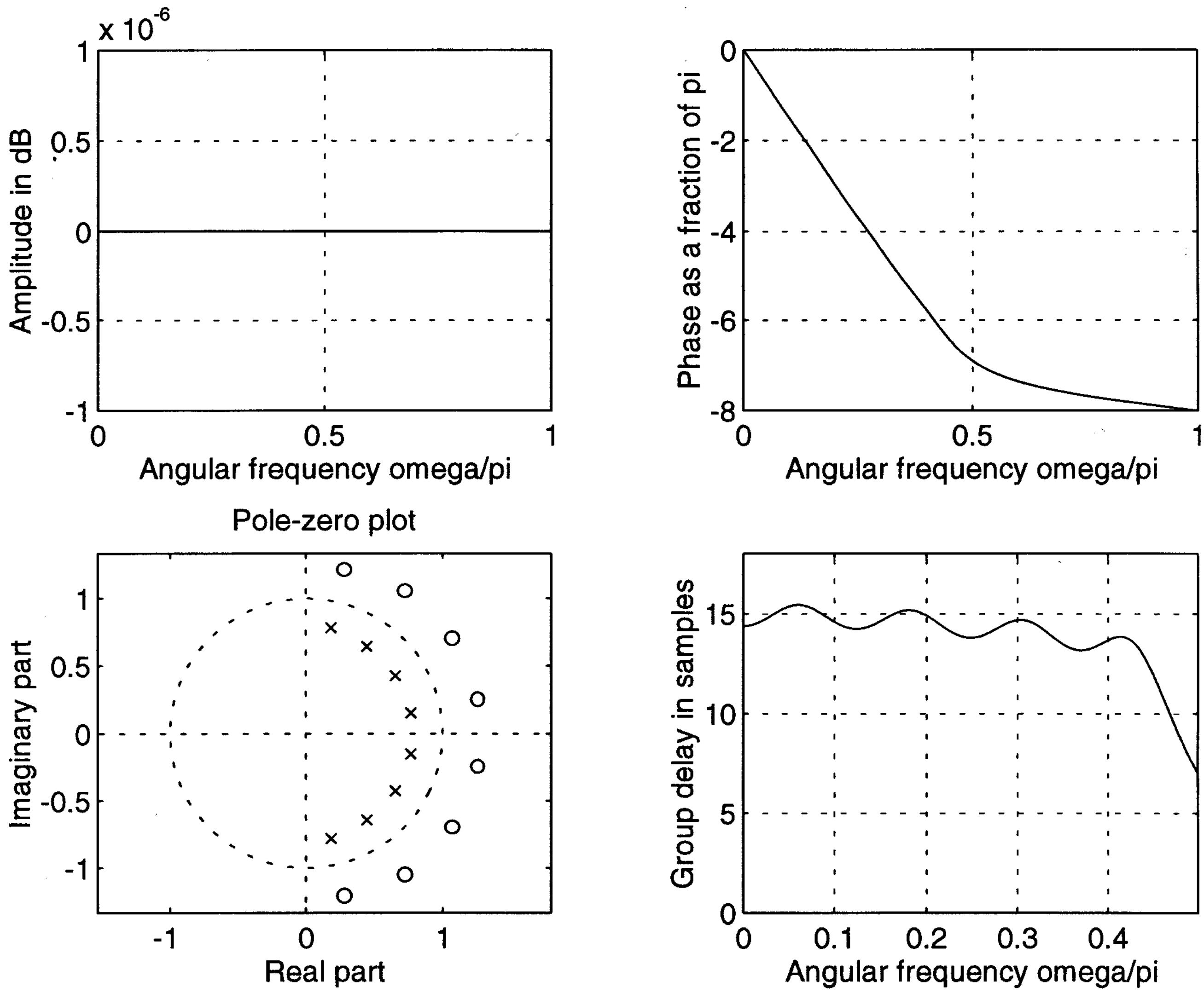
- Some characteristics of the optimized allpass filter are shown on page 66.

- Page 67 shows some of the responses of the cascaded filter, whereas page 68 shows the group delays of the elliptic filter, the allpass filter, and the overall filter.
- Note that the group delay is an additive response and the overall group delay achieves higher values than that of the elliptic filter, increasing the overall delay.
- Finally, it should be noted that the above approach of first designing a frequency-selective filter and then improving the group delay with the aid of an allpass filter is not the best approach.
- Filters with lower order and better performances can be designed by more sophisticated techniques to be considered in the course "Digital Linear Filtering II".

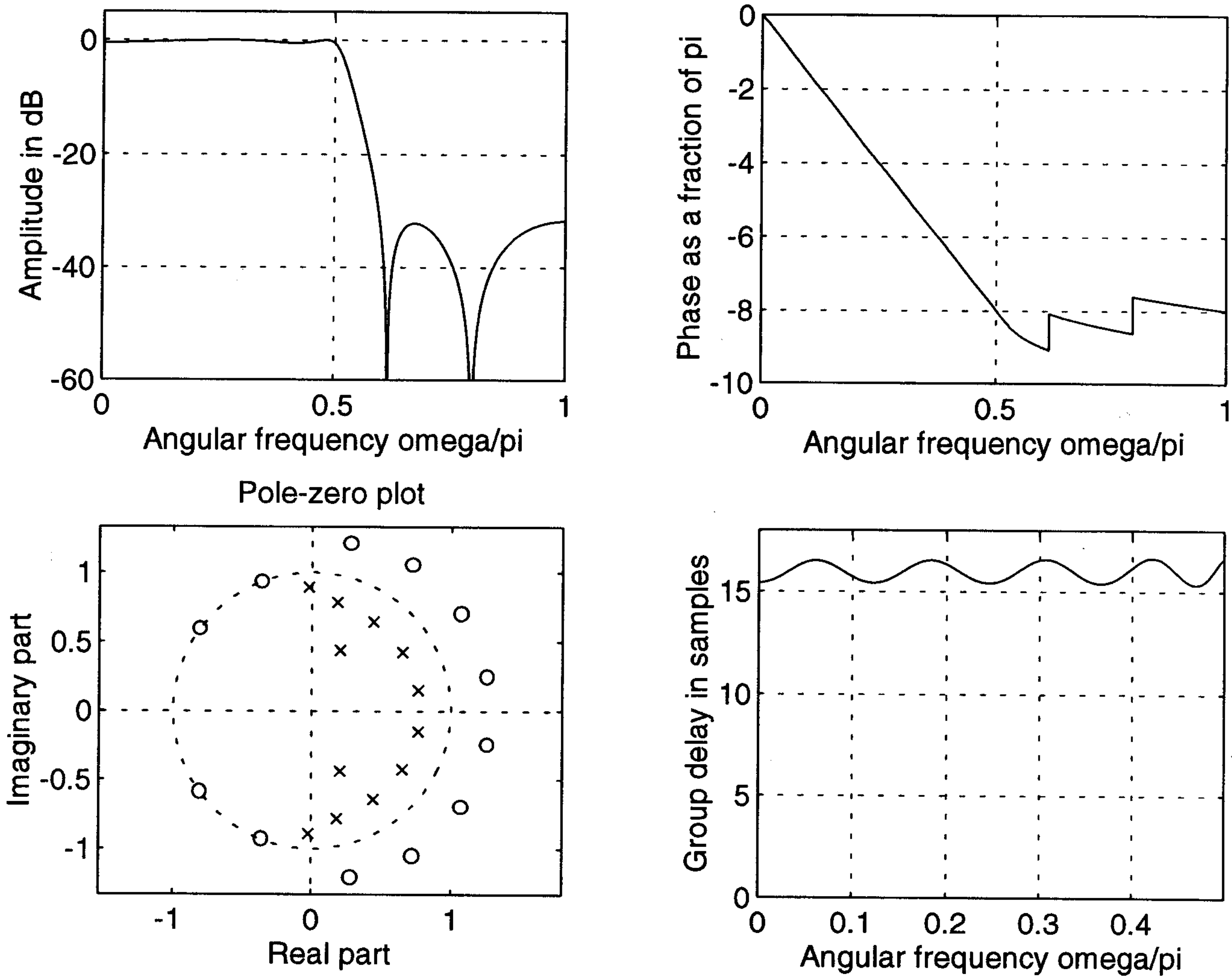
Some Characteristics of A Fourth-Order Elliptic Filter



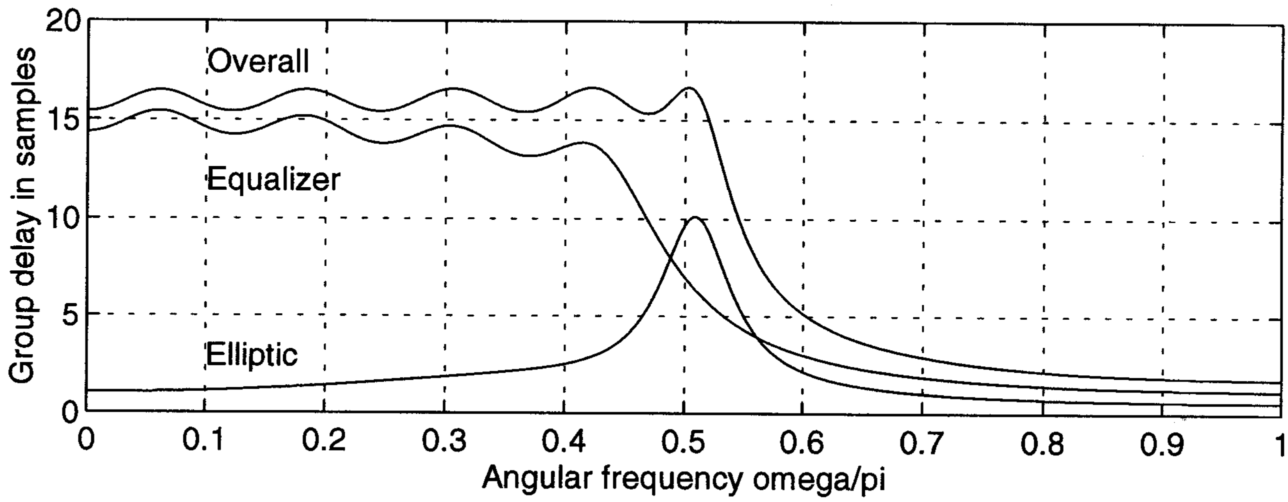
Some Characteristics of An Optimized Eigth-Order Allpass Filter

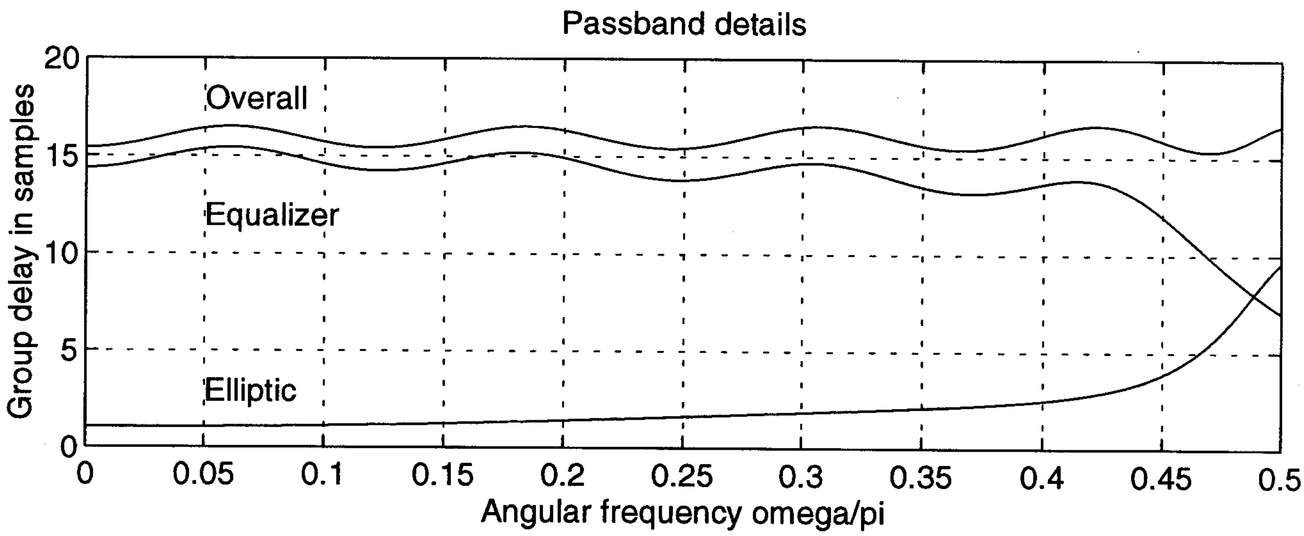


Some Characteristics of A Cascade of the Elliptic Filter and the Allpass Filter



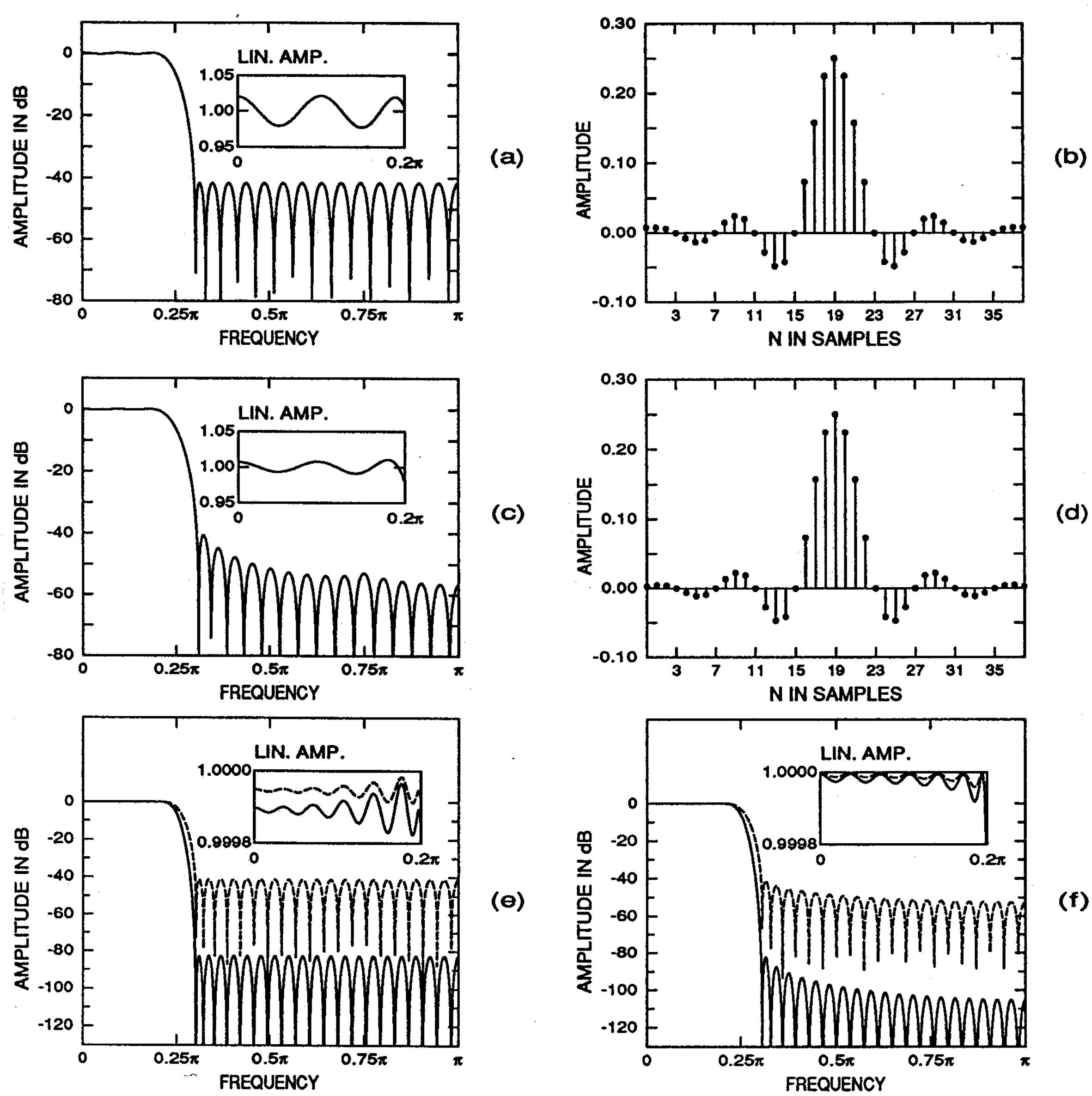
Group Delay responses for the Elliptic filter, the Allpass Filter, and Their Cascade



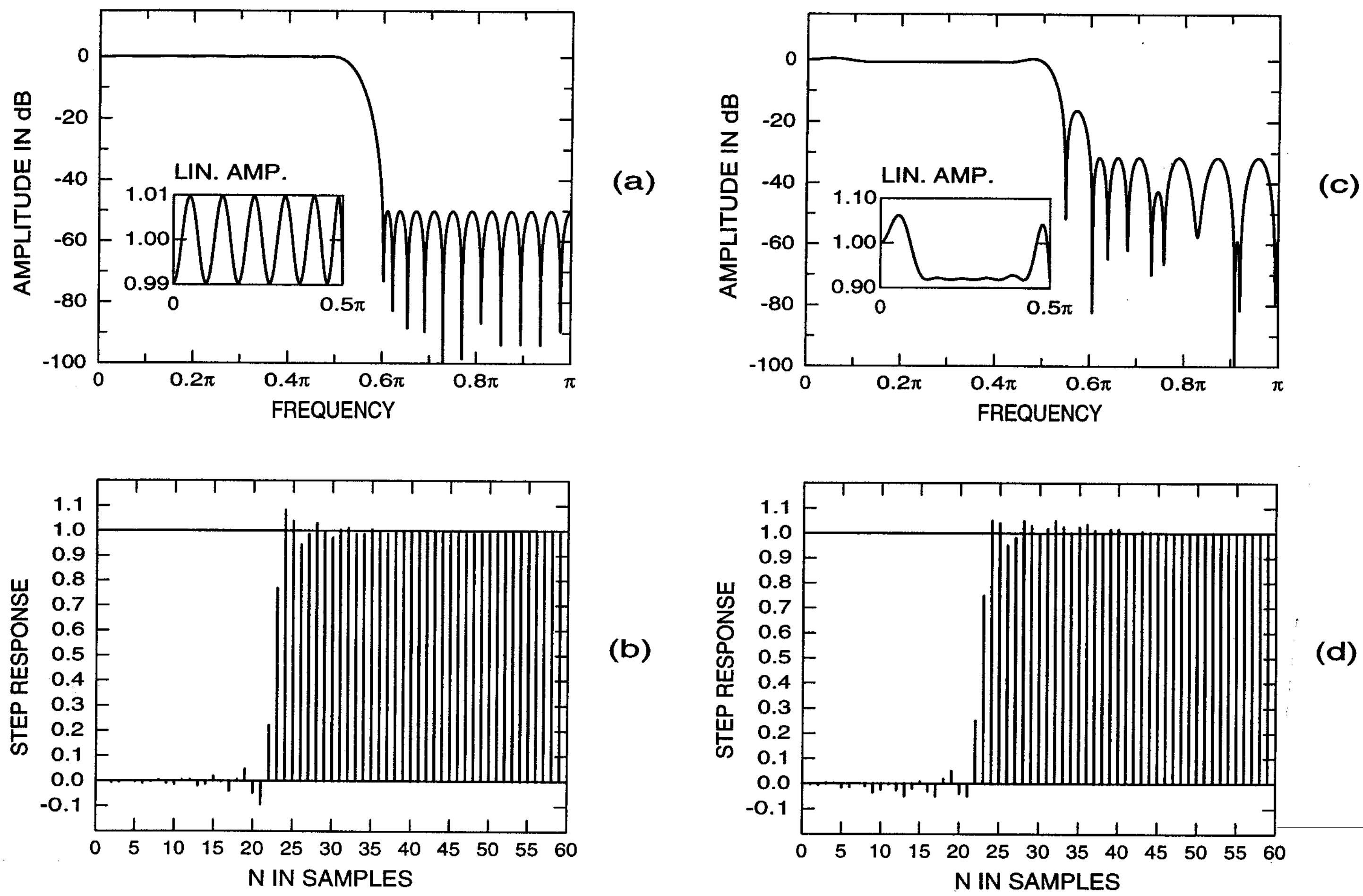


TIME-DOMAIN CONDITIONS

- In some cases, it is desired to optimize the frequency-domain behavior of the filter subject to the given time-domain conditions.
- Nyquist or N-th band filters: every N-th impulse response value is restricted to be zero except for the central sample of value $1/N$.



- In some applications, the overshoot of the step response of a digital filter, optimized only in the frequency domain, is too large.
- The filter has to be reoptimized with constraints on the ripple of the step response.



APPROXIMATION CRITERIA

- Three different error measures are normally used in the approximation theory and also in designing digital filters.
- For intruduction purposes, consider the following simple example:
- It is desired to approximate the desired curve $d(x) = x^2$ by a line $h(x) = a+bx$ for $0 \leq x \leq 1$.

MINIMAX ERROR DESIGNS

- In this case, the problem is to find a and b of $h(x) = a + bx$ in such a way that the peak absolute value of the following weighted error function

$$E(x) = W(x)[h(x) - d(x)]$$

on $[0,\ 1]$ is minimized. Here, $W(x)$ is a weighting function and it must be positive on $[0,\ 1]$.

- In order word, the problem is to find a and b to minimize the quantity

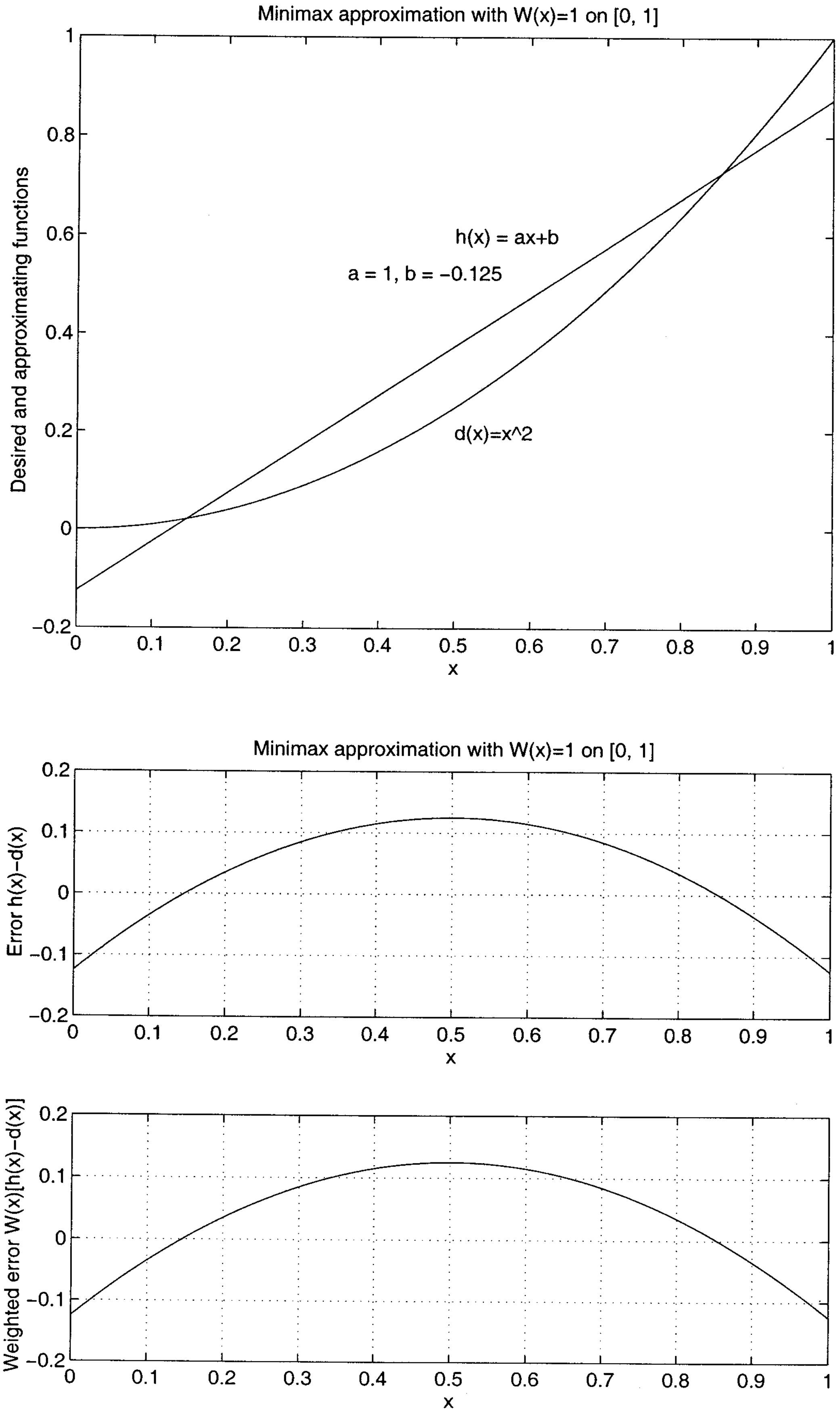
$$\epsilon = \max_{x \in [0,\ 1]} |E(x)|.$$

- We consider two cases. For both cases, $W(x) = 1$ on $[0.3,\ 1]$. For the first and second cases, $W(x) = 1$ and $W(x) = 10$ on $[0,\ 0.3]$, respectively.

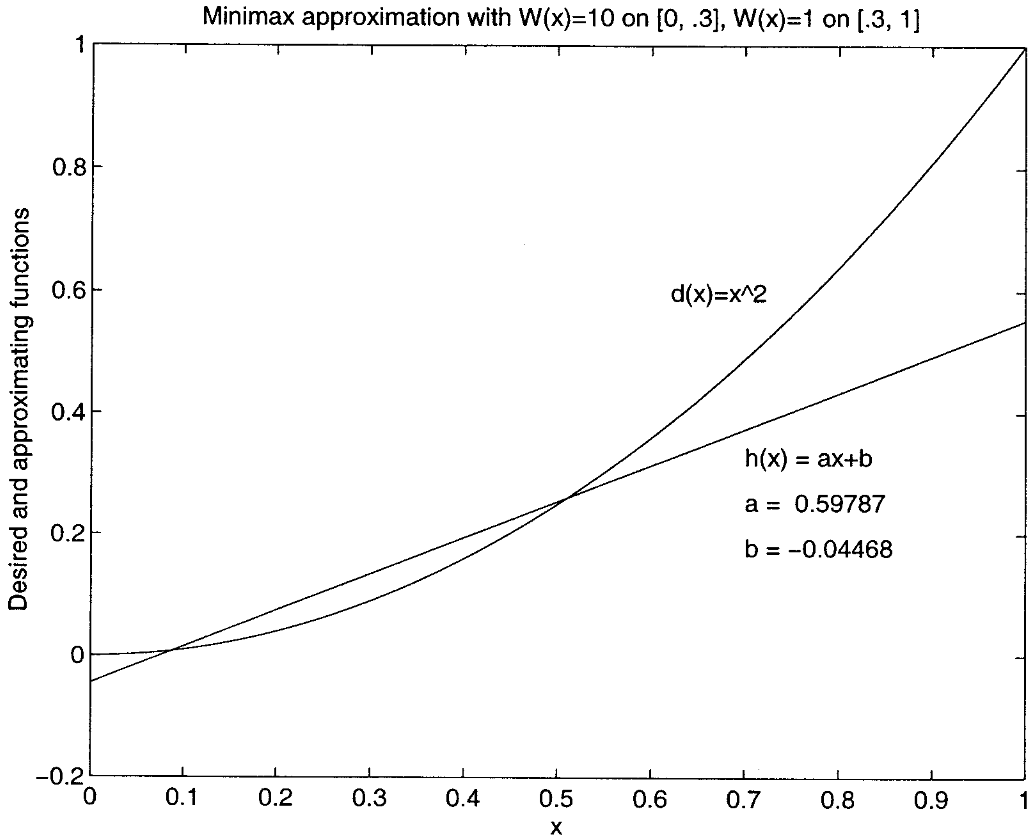
- Pages 74 and 75 give the optimum solutions to these cases.

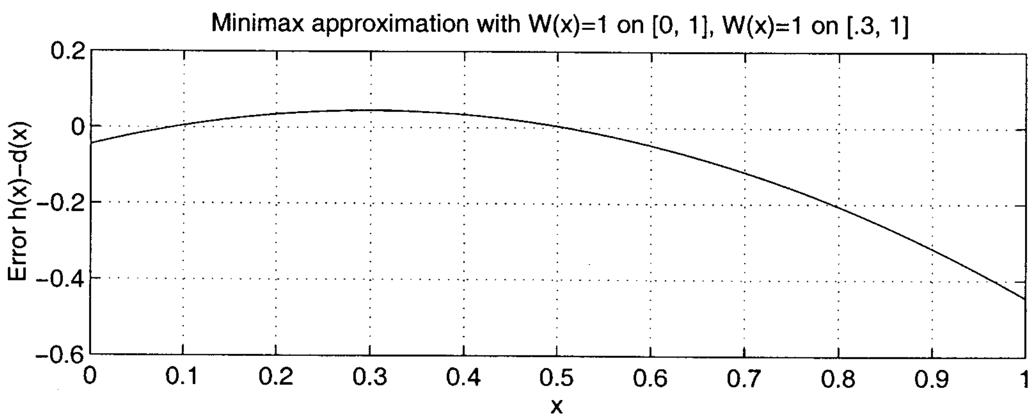
- It is seen that in both cases $E(x)$ achieves the peak absolute value at three points in such a way that the sign of $E(x)$ alternates. This is one more than the number of unknowns (a and b).
- In the first case, the peak absolute value of $E(x)$ is 0.125 and is achieved at $x = 0$, $x = 1/2$, and $x = 1$.
- In the second case, the peak absolute value of $E(x)$ is 0.4468 and is achieved at $x = 0$, $x = 0.3$, and $x = 1$.
- In the first case, the weighted error and the actual error $h(x) - d(x)$ are the same. In the second case, because of the higher weighting on $[0. 0.3]$, the absolute values of $h(x) - d(x)$ are significantly smaller in this interval.

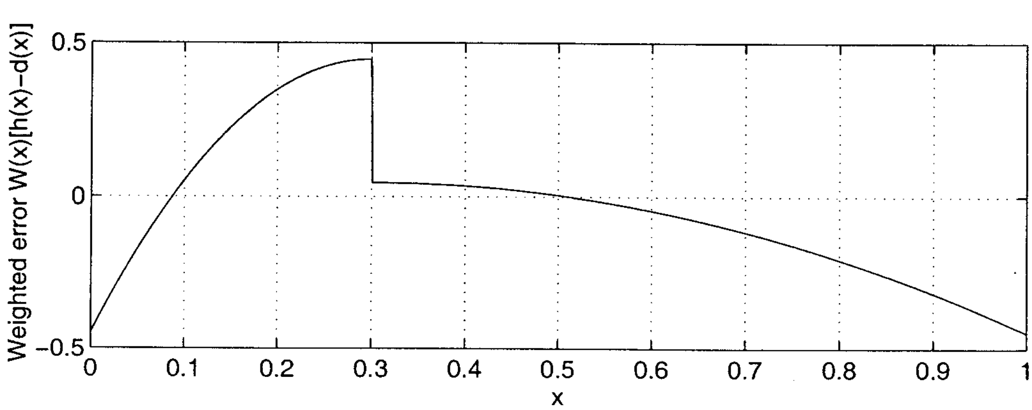
MINIMAX ERROR DESIGN: $W(x) = 1$ on [0, 1]



MINIMAX ERROR DESIGN: $W(x) = 10$ on $[0,\ 0.3]$, $W(x) = 1$ on $[0.3,\ 1]$







LEAST-SQUARED ERROR DESIGNS

- Another error measure is L_p-norm. In this case, the problem is to find a and b to minimze the quantity

$$E_p = \int_0^1 [W(x)[h(x) - d(x)]]^p d\omega,$$

where p is a positive even integer.

- For $p = 2$, the quantity to be minimized is

$$E_2 = \int_0^1 [W(x)[h(x) - d(x)]]^2 d\omega$$

and the optimum solution is called the least-squared error solution.

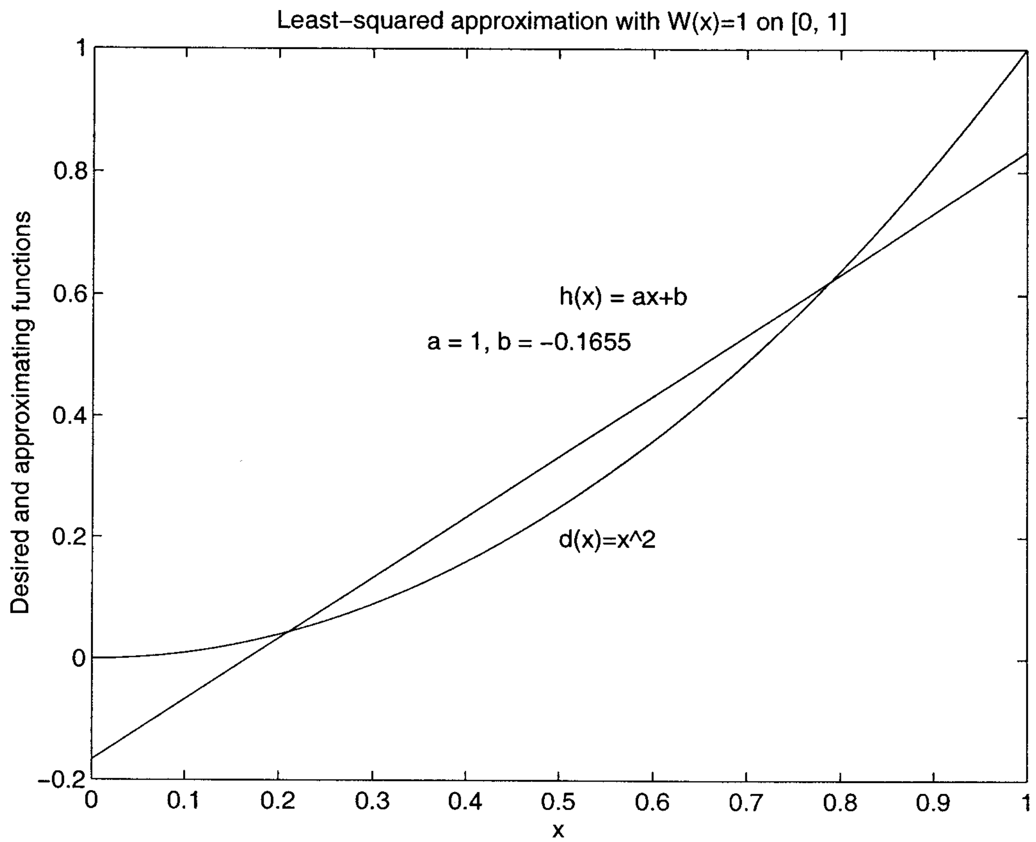
- We again consider two cases. For both cases, $W(x) = 1$ on $[0.3,\ 1]$. For the first and second cases, $W(x) = 1$ and $W(x) = 10$ on $[0,\ 0.3]$, respectively.

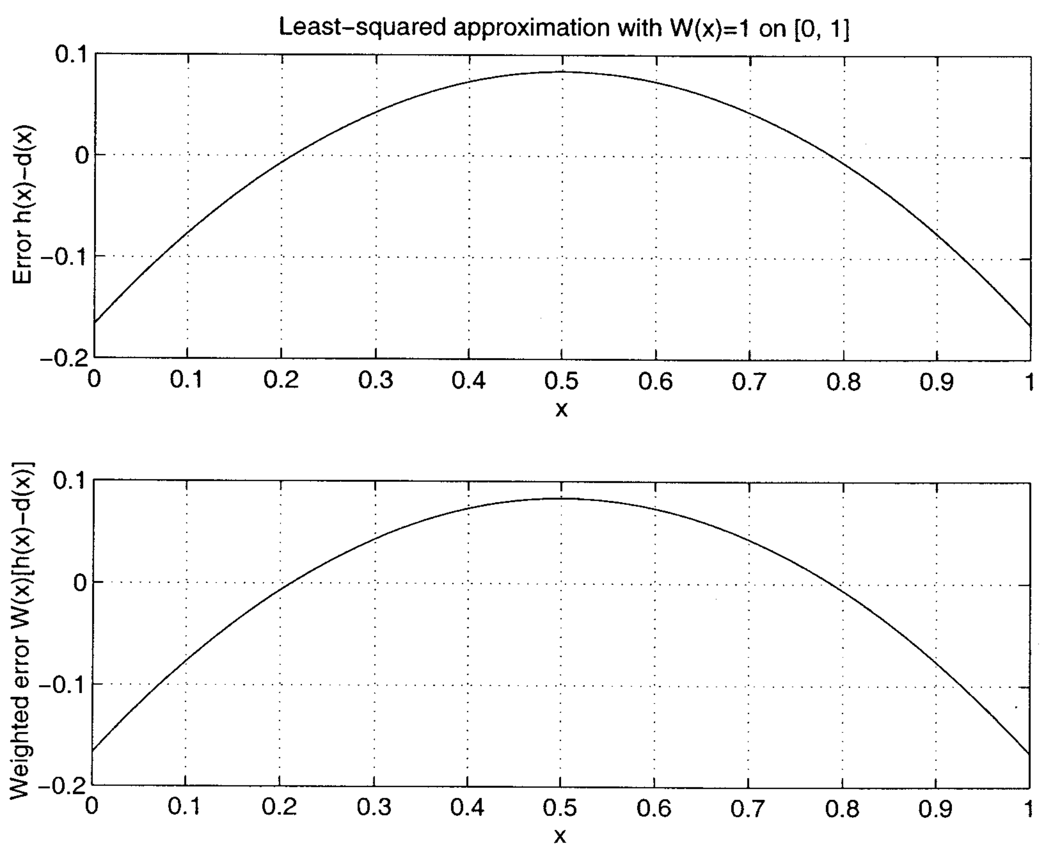
- Pages 78 and 79 give the optimum solutions to these cases.

- It is seen that in both cases, the absolute values of $E(x)$ are larger near the edges of the interval.

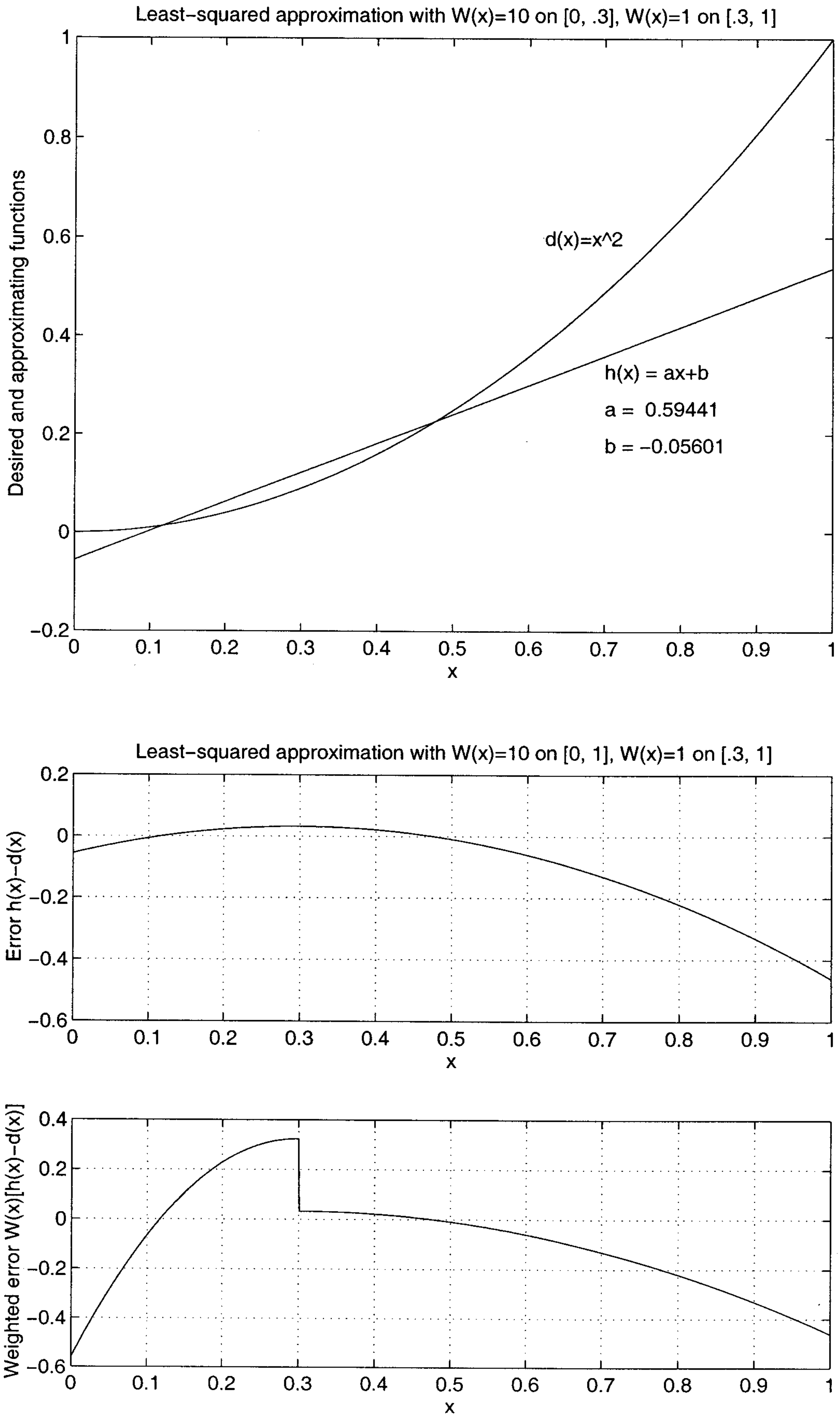
- In the first case, the weighted error and the actual error $h(x) - d(x)$ are again the same. In the second case, because of the higher weighting on $[0,\ 0.3]$, the absolute values of $h(x) - d(x)$ are significantly smaller in this interval.

LEAST-SQUARED ERROR DESIGN: $W(x) = 1$ on $[0,\ 1]$



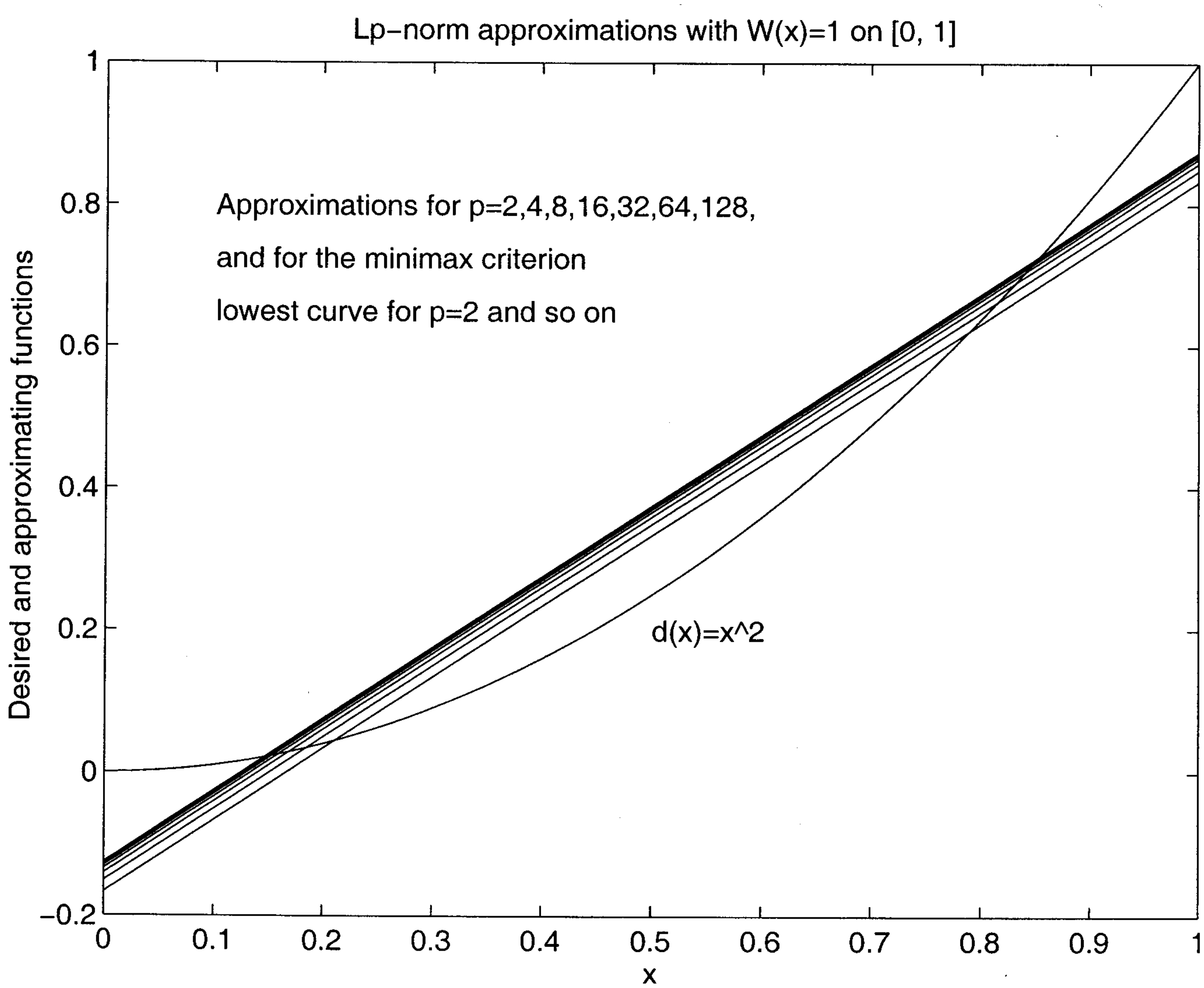


LEAST-SQUARED ERROR DESIGN: $W(x) = 10$ on $[0,\ 0.3]$, $W(x) = 1$ on $[0.3,\ 1]$



L_p-NORM DESIGNS

- It is characteristics of the L_p-norm designs that as $p \mapsto \infty$, the solution approaches the minimax solution.
- This is illustrated for the case $W(x) \equiv 1$ on [0, 1] in the figure below, where L_p-norm designs have been generated for $p = 2, 4, 8, 16, 32, 64, 128,$ and 256. As seen from the figure, the corresponding $h(x)$ approaches the minimax design, as is desired.



MAXIMALLY-FLAT APPROXIMATIONS

- In this case, one point $x = x_0$ is selected in the interval [0, 1] and the approximating function is determined such that

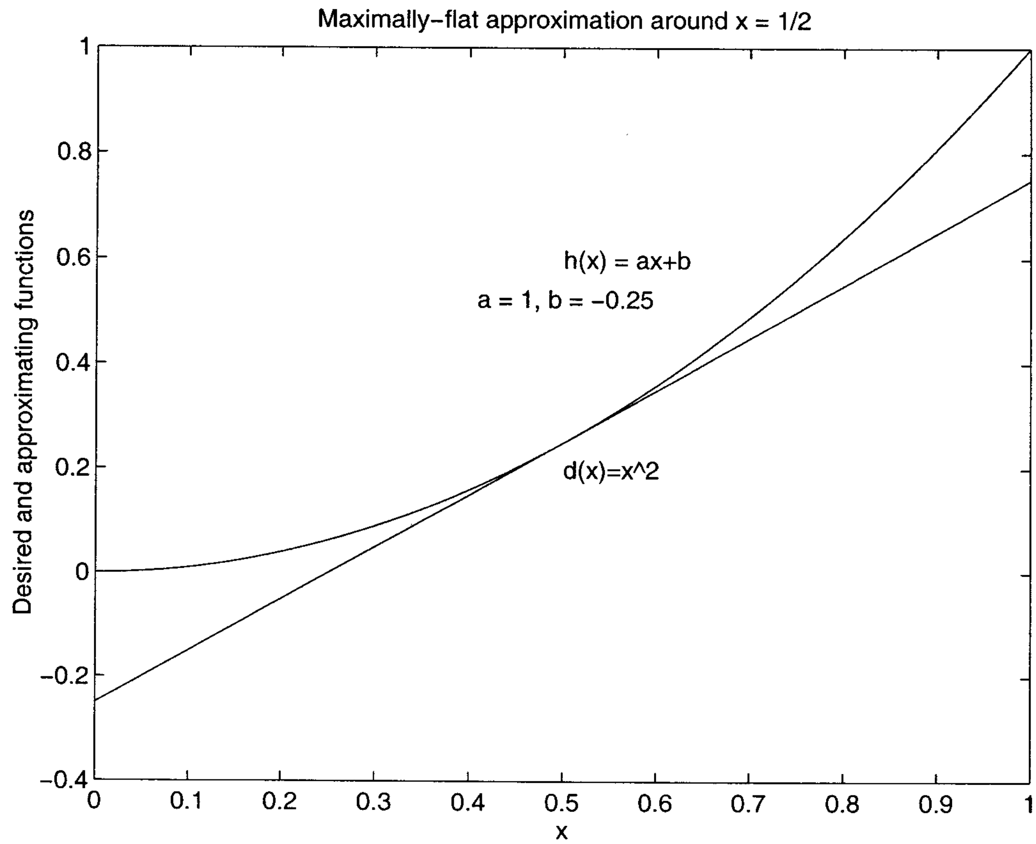
$$h(x_0) = d(x_0)$$

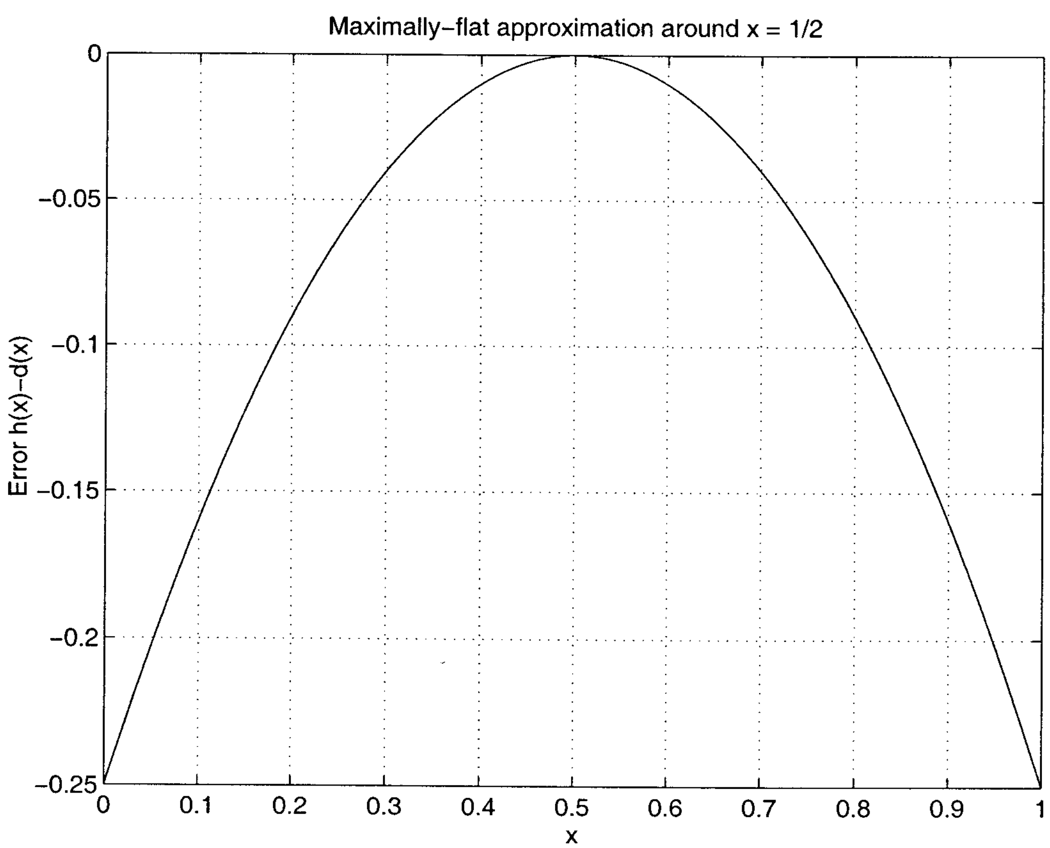
  and

$$\frac{dh(x)}{dx}\Big|_{x=x_0} = \frac{dd(x)}{dx}\Big|_{x=x_0}.$$

- For our problem, we have only two unknowns. Therefore, only the first derivative is needed to completely specify the approximating function. For more unknowns, higher order derivatives are used until the approximating function is uniquely determined.
- Let us select $x_0 = 1/2$. In this case ($d(x) = x^2$), it is required that $h(1/2) = a/2 + b = d(1/2) = 1/4$ and $h'(1/2) = a = d'(1/2) = 1$.
- Therefore, $a = 1$ and $b = -1/4$.
- The following page shows the resulting $h(x)$ as well as the error function $h(x) - d(x)$.
- As seen from the figure, the error is very small in the vicinity of the point $x = x_0 = 1/2$.

MAXIMALLY-FLAT SOLUTION





APPROXIMATION CRITERIA FOR DESIGNING DIGITAL FILTERS

- The above-mentioned approximation criteria are normally used in designing digital filter.

Minimax Error Designs

- Some applications require that the transfer function coefficients be optimized to minimize the maximum error between the approximating response and the given desired response.
- The solution minimizing this error function is called a *minimax* or *Chebyshev approximation*.
- In the case of weighted error function $E(\omega)$, the quantity to be minimized is the peak absolute value of $E(\omega)$ on X, i.e., the quantity

$$\epsilon = \max_{\omega \in X} |E(\omega)|.$$

NORMAL CASE

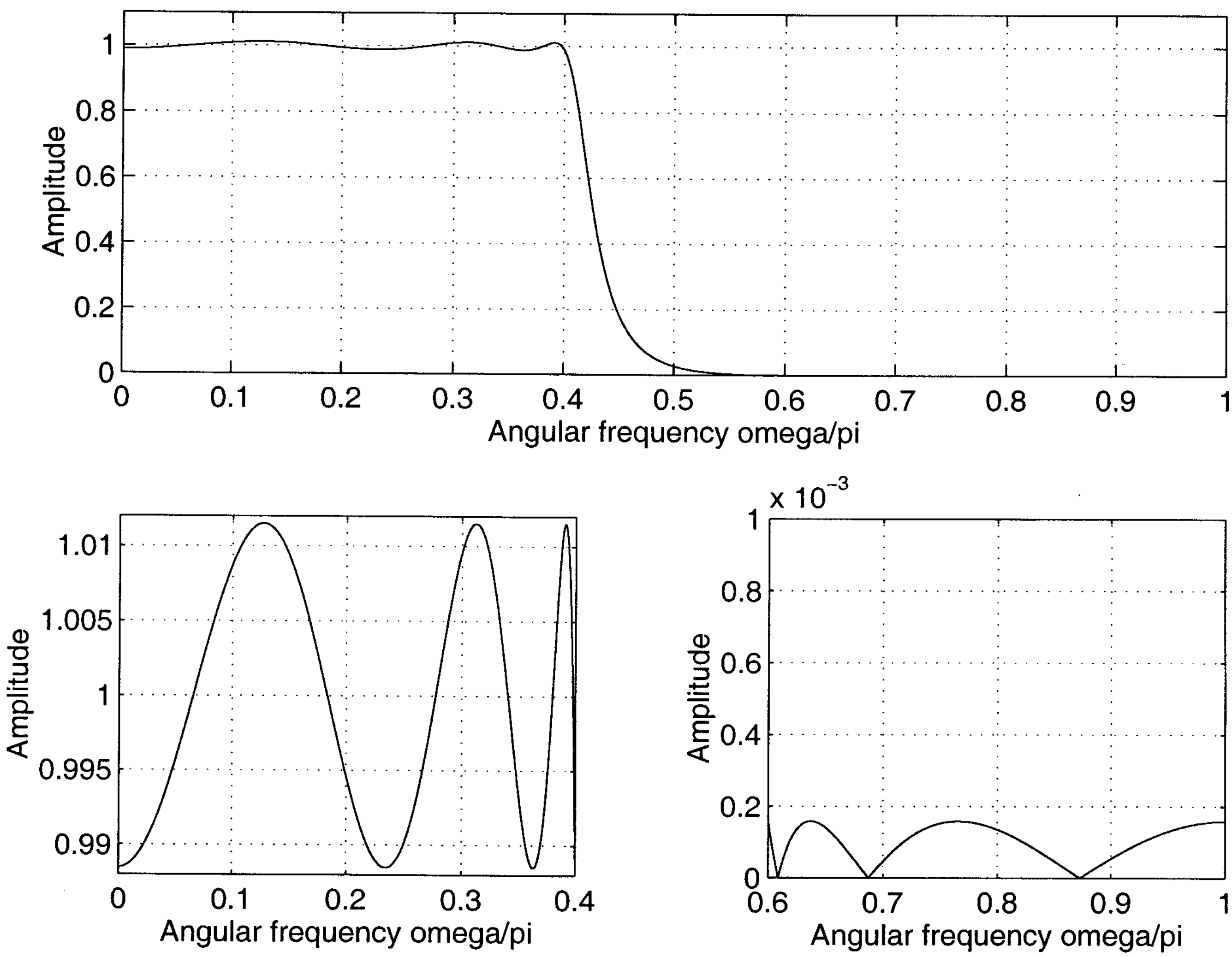
- The maximum allowable value of ϵ is specified.

I: Find the minimum order of a filter required to meet the given criteria

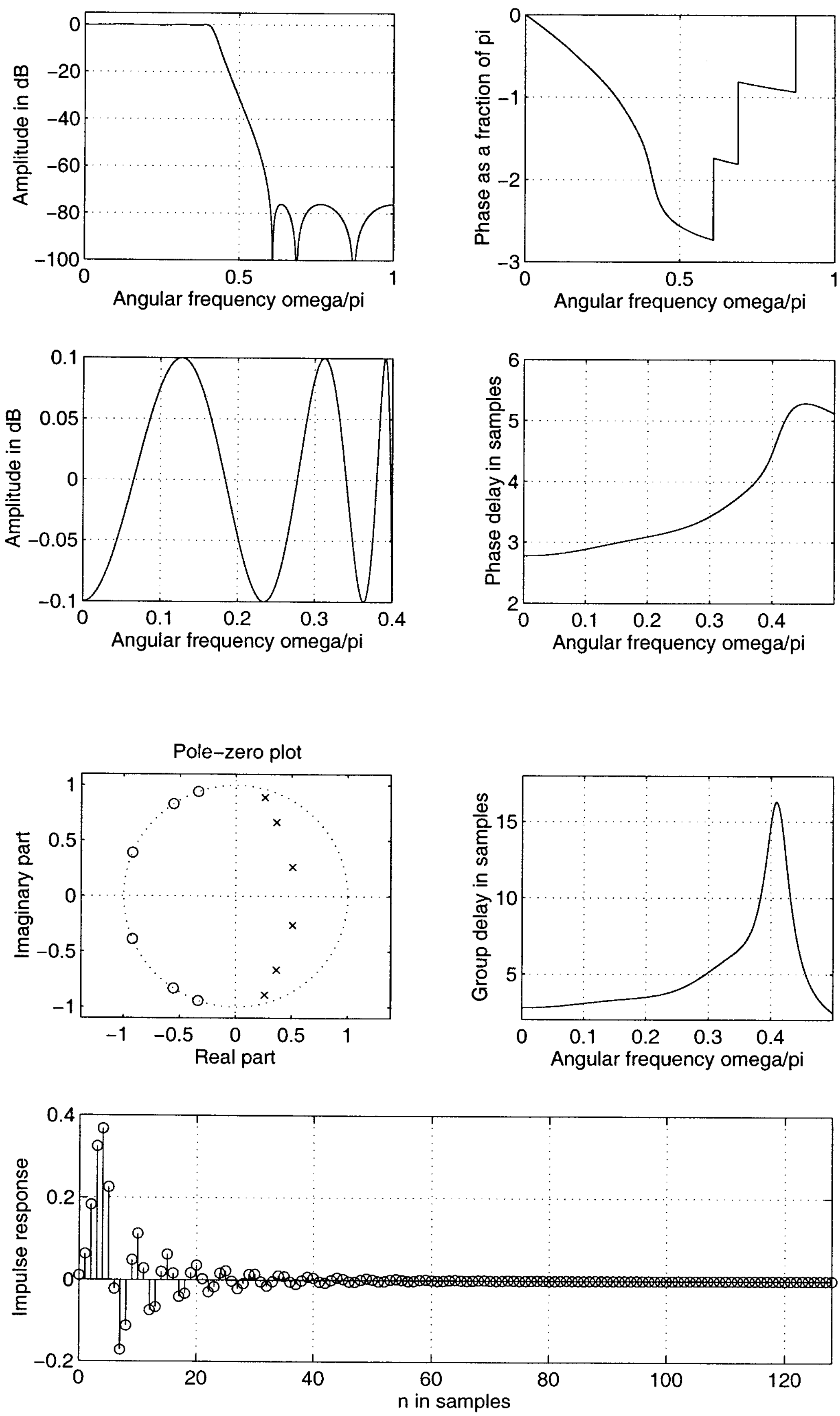
II: Optimize the coefficients of a minimum-order transfer function to minimize ϵ.

- Examples of minimax solutions are elliptic (Cauer) IIR filters (see pages 85 and 86) and equiripple linear-phase FIR filtes (see page 87).
- Note that for our example IIR filter, the passband amplitude oscillates around unity.
- In some cases, the maximum of the passband amplitude response of an IIR filter is restricted to be unity.
- Note that in the case of pages 85 and 86, the minimax criterion is used such that given the maximum allowable passband variation, the filter coefficients are optimized to maximize the stopband attenuation (these filters will be considered in more details in Part IV).
- The design of the FIR filter of page 87 will be considered in Part III.

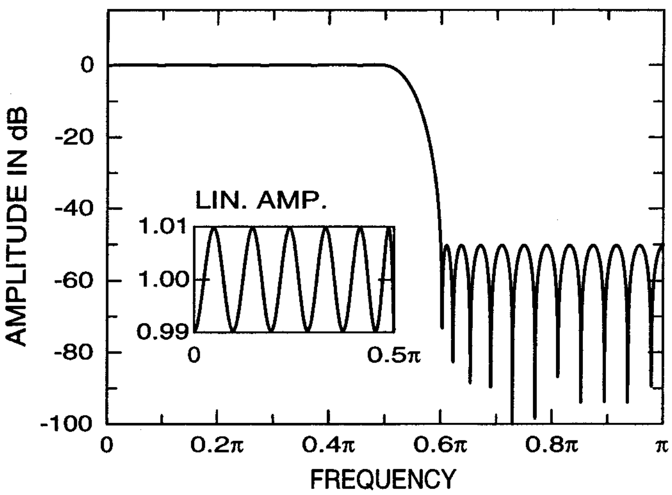
SIXTH-ORDER ELLIPTIC IIR FILTER: $\omega_p = 0.4\pi$, $\omega_s = 0.6\pi$, $A_p = 0.2$ dB, $A_s \geq 60$ dB.

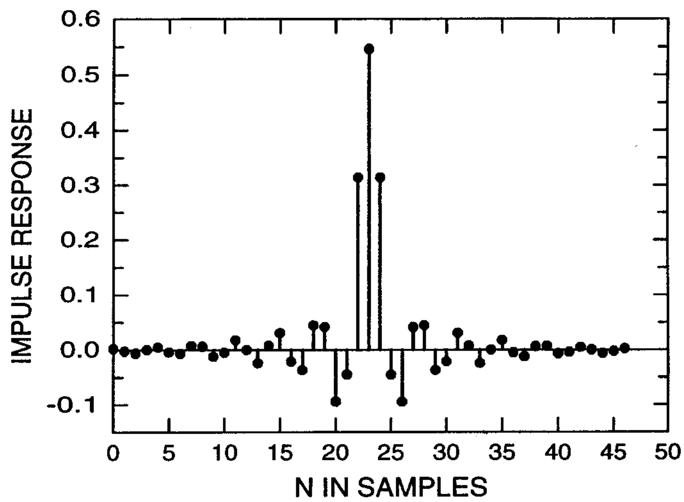


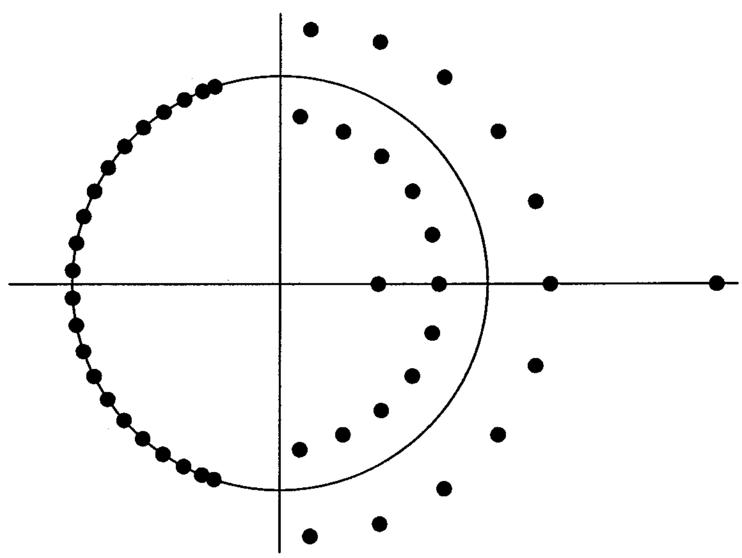
MORE DETAILS



MINIMAX FIR FILTER OF ORDER 46: $\omega_p = 0.5\pi$, $\omega_s = 0.6\pi$, $\delta_p = 0.01$, $\delta_s = 0.00316$







LEAST-SQUARED ERROR DESIGNS

- L_p-norm: It is desired to minimize the function

$$E_p = \int_X [W(\omega)[|H(e^{j\omega})| - D(\omega)]]^p d\omega,$$

where p is a positive even integer.

- It can be shown that as $p \mapsto \infty$, the solution minimizing the above quantity approaches the minimax solution.

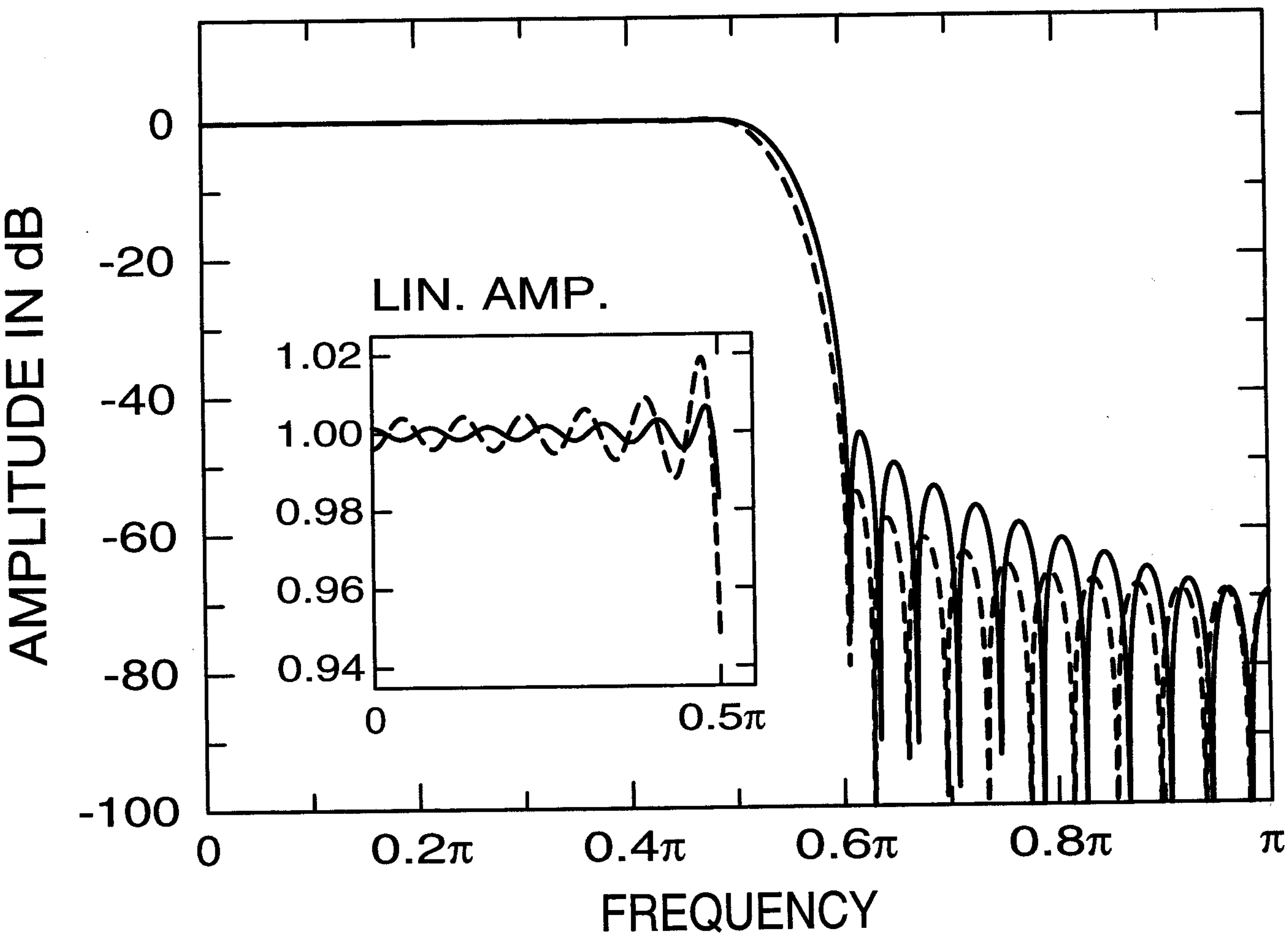
- This fact is exploited in some IIR filter design methods. For FIR filters, L_p-error designs are of little practical use since there are efficient algorithms directly available for designing in the minimax sense FIR filters with arbitrary specifications.

- The exception is the L_2-error or *least-squared error* designs, which can be found very effectively. In this case, the quantity to be minimized is

$$E_2 = \int_X [W(\omega)[|H(e^{j\omega})| - D(\omega)]]^2 d\omega,$$

where X is the union of the passbands and stopbands.

LEAST-SQUARED-ERROR FIR FILTERS OF ORDER 46: $\omega_p = 0.5\pi$, $\omega_s = 0.6\pi$; Stopband weightings are 1 and 10



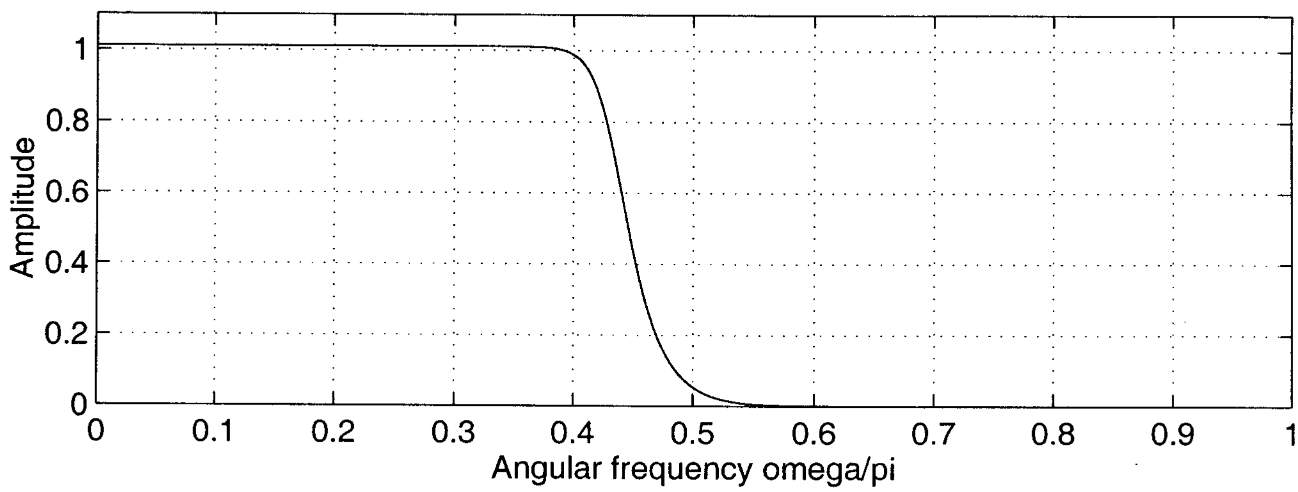
Maximally-Flat Approximations

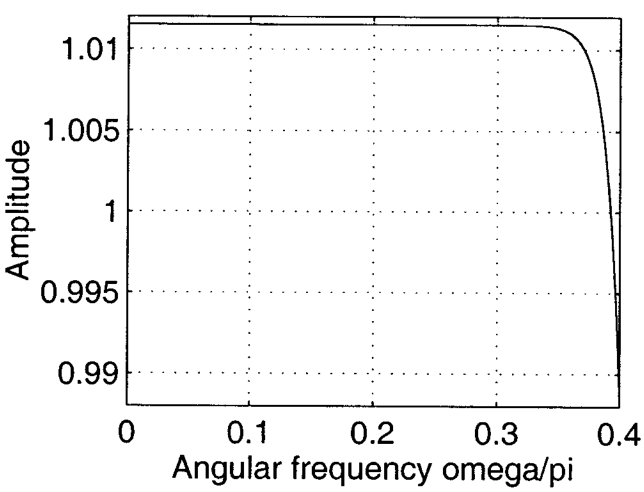
- The approximating response is obtained based on a Taylor series approximation to the desired response at a certain frequency point and the solution is called a *maximally flat approximation.*

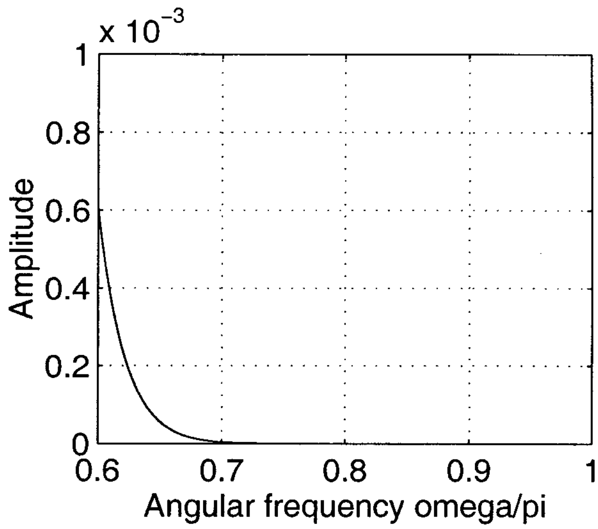
- In some cases, such as in designing maximally-flat (Butterworth) IIR filters, there are two points, one in the passband and one in the stopband, where a Taylor series approximation is applied. See pages 91 and 92.

- For designing linear-phase FIR filters, a similar approach can be used. See page 93.

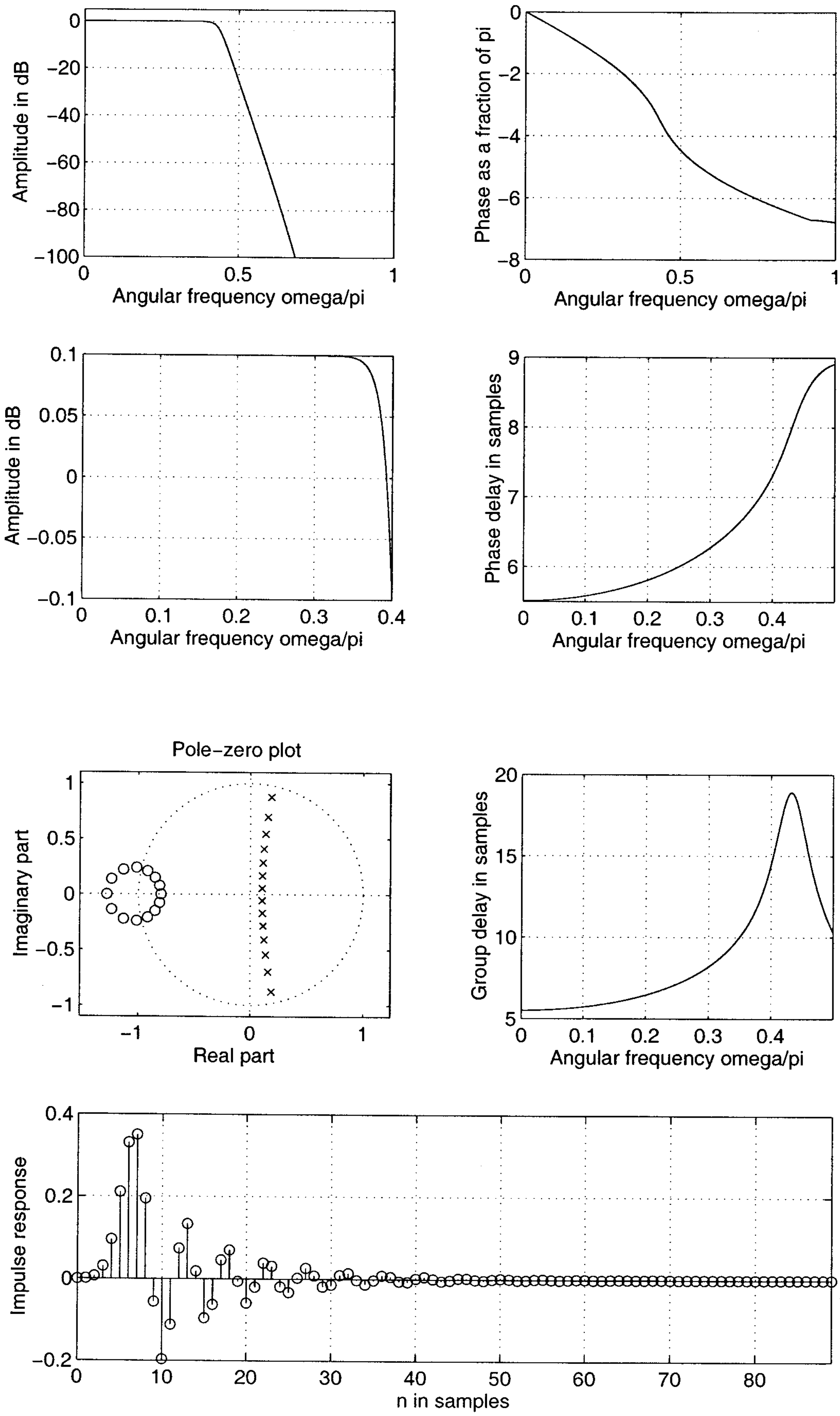
FOURTEENTH-ORDER BUTTERWORTH IIR FILTER: $\omega_p = 0.4\pi$, $\omega_s = 0.6\pi$, $A_p = 0.2$ dB, $A_s \geq 60$ dB.





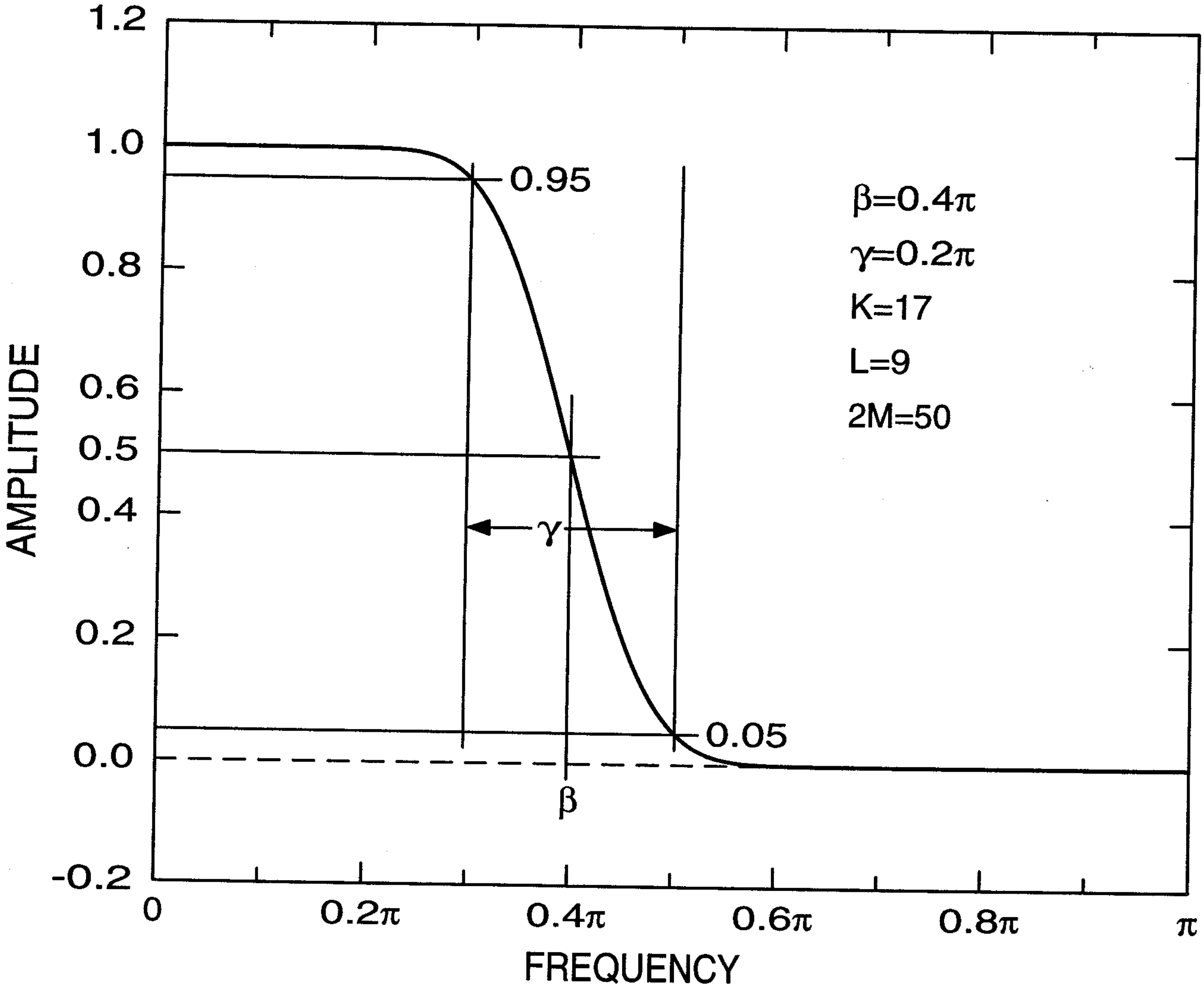


MORE DETAILS



MAXIMALLY-FLAT FIR FILTER OF ORDER

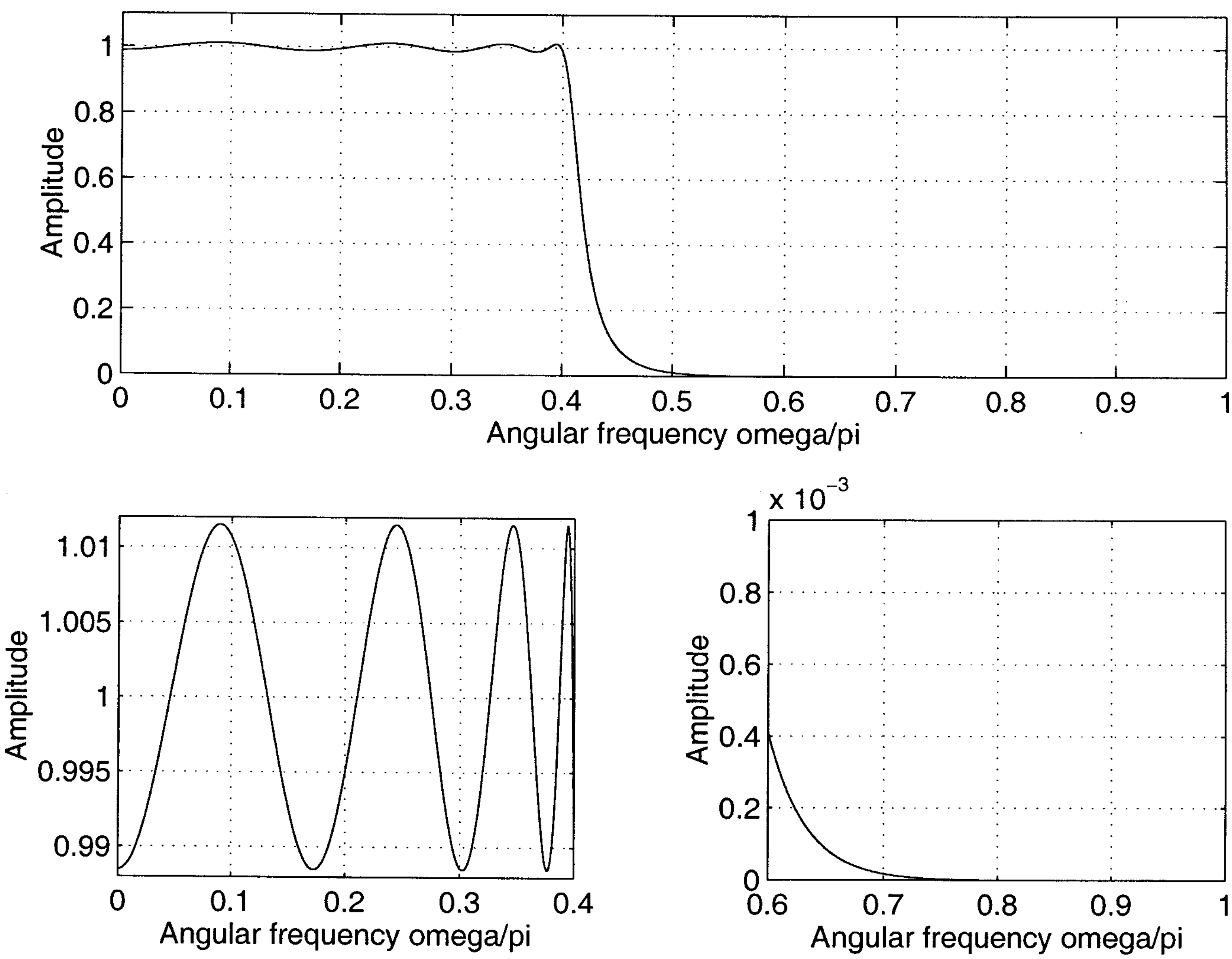
$N = 50$: $\omega_p = 0.3\pi$, $\omega_s = 0.5\pi$



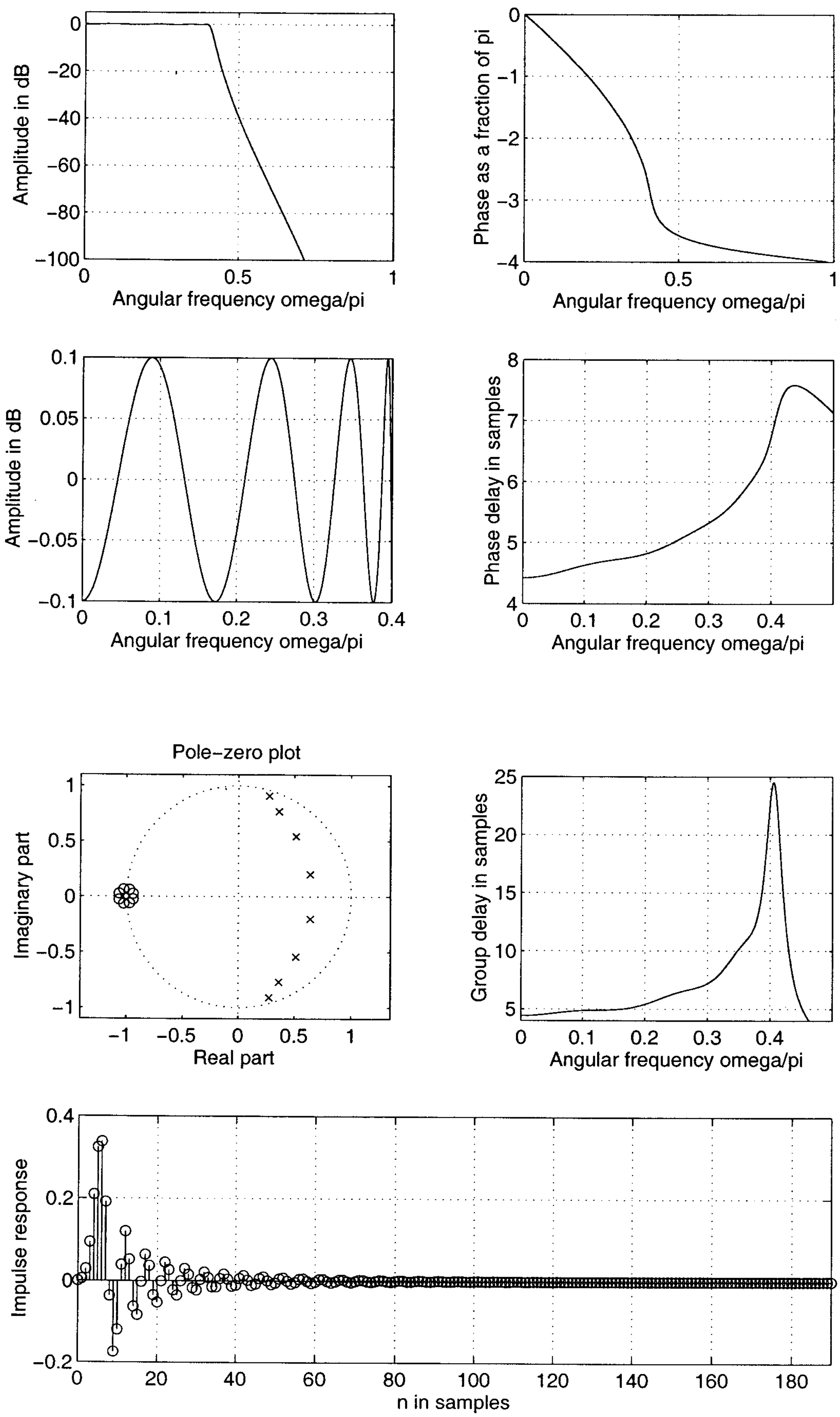
OTHER CRITERIA

- Most of the methods developed for designing digital filters use one of the above approximation criteria.
- In some synthesis techniques, a combination of these criteria is used. For instance, in the case of Chebyshev IIR filters, a Chebyshev approximation in the passband and a maximally flat approximation in the stopband are used. See pages 95 and 96.
- In the case of inverse Chebyshev filters, a maximally flat approximation in the passband and a Chebyshev approximation in the stopband are used. See pages 97 and 98.
- There exist also several simple filter design techniques which do not use directly the above criteria at all.
- A typical example of such methods is the design of FIR filters using windows, where the Fourier series of an ideal filter is first truncated and then smoothed using a window function.

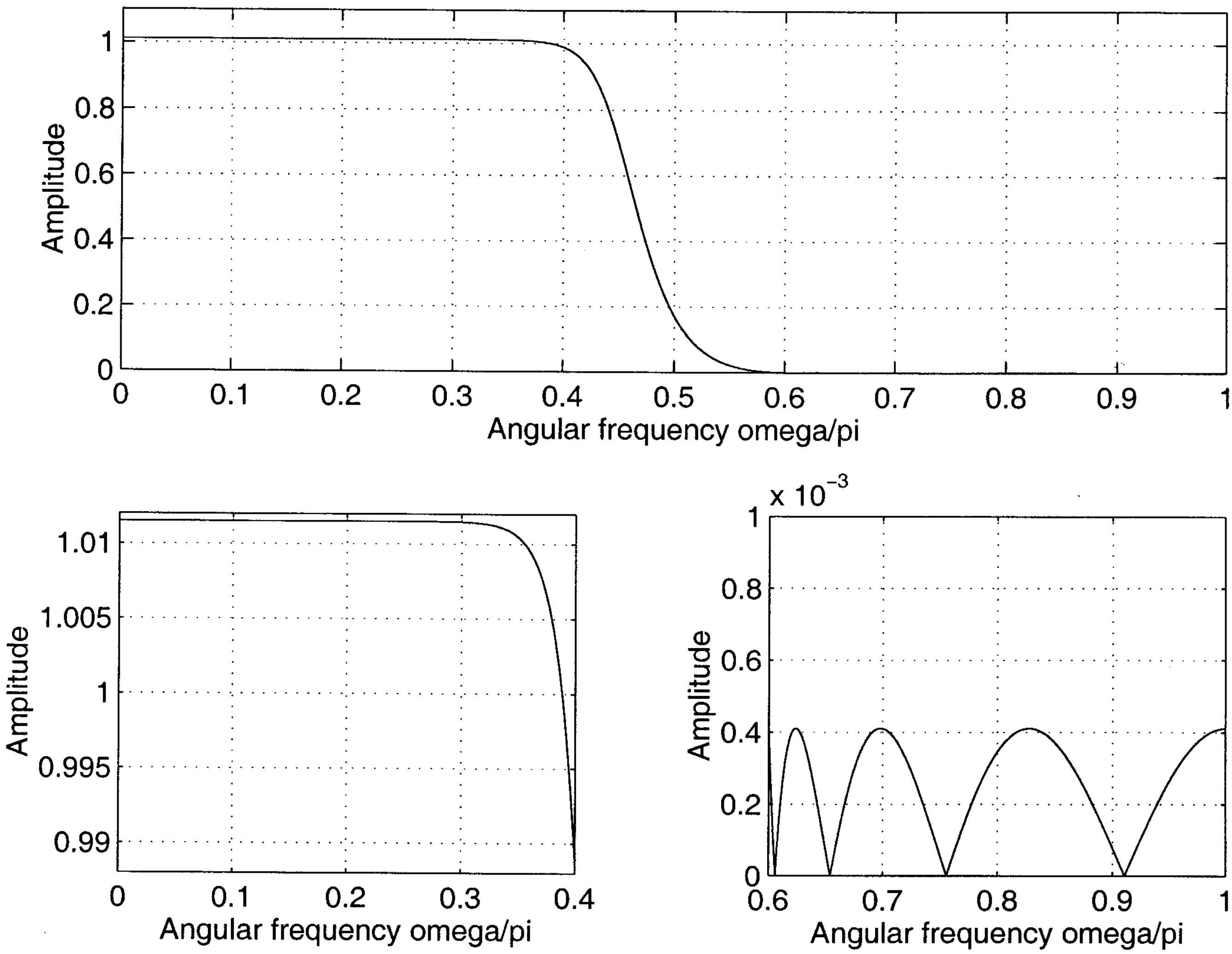
EIGHTH-ORDER CHEBYSHEV TYPE I IIR FILTER: $\omega_p = 0.4\pi$, $\omega_s = 0.6\pi$, $A_p = 0.2$ dB, $A_s \geq 60$ dB.



MORE DETAILS



EIGHTH-ORDER CHEBYSHEV TYPE II IIR FILTER: $\omega_p = 0.4\pi$, $\omega_s = 0.6\pi$, $A_p = 0.2$ dB, $A_s \geq 60$ dB.



MORE DETAILS