2. Cascade Form

Expressing (9.5) as products of second-order polynomial system functions via factorization:

$$H(z) = \sum_{k=0}^{M} b_k z^{-k} = \prod_{k=1}^{M_c} \left(\beta_{0k} + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}\right) \qquad (9.9)$$

where $M_c = \lfloor (M+1)/2 \rfloor$ is the largest integer contained in $(M+1)/2$. Note that when M is odd, one of the $\{\beta_{2k}\}$ will be zero. Assuming that M is even, this implementation needs M storage elements, $3M/2$ multiplications and M additions, for computing each output value of $y[n]$.

Why second-order polynomial instead of first-order polynomial?

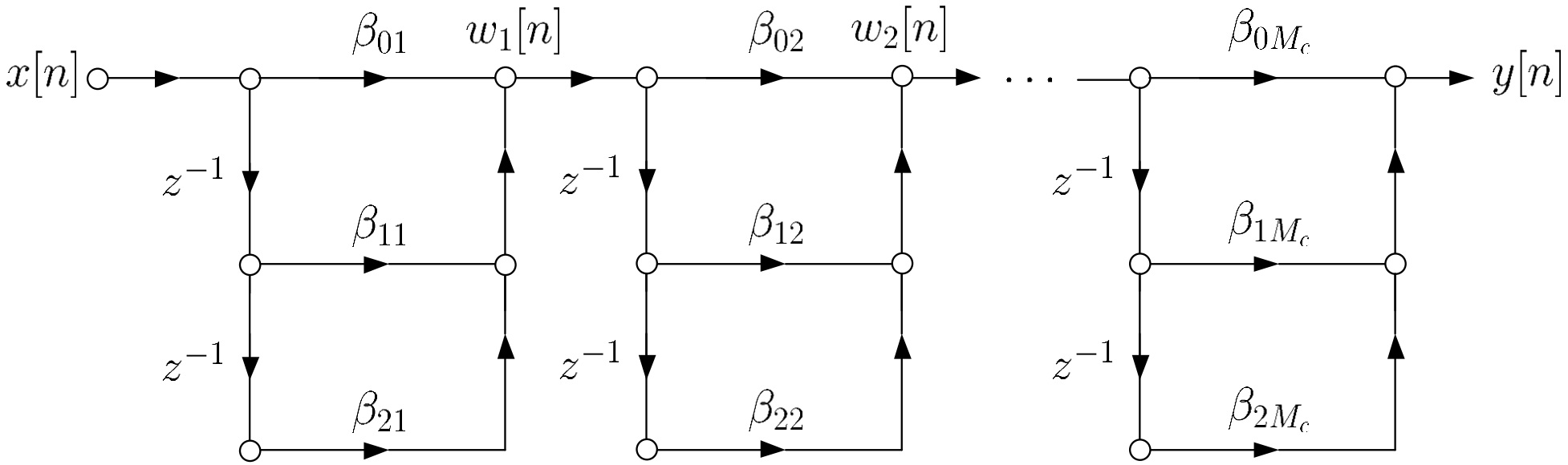


Fig.9.6: Cascade form of FIR filter

$$w_1[n] = \beta_{01}x[n] + \beta_{11}x[n-1] + \beta_{21}x[n-2] \qquad (9.10)$$

and

$$w_2[n] = \beta_{02}w_1[n] + \beta_{12}w_1[n-1] + \beta_{22}w_1[n-2] \qquad (9.11)$$

Taking z transform on (9.10) and (9.11):

$$\begin{aligned} W_1(z) &= \beta_{01}X(z) + \beta_{11}z^{-1}X(z) + \beta_{21}z^{-2}X(z) \\ &= \left(\beta_{01} + \beta_{11}z^{-1} + \beta_{21}z^{-2}\right) X(z) \end{aligned} \tag{9.12}$$

and

$$\begin{aligned} W_2(z) &= \beta_{02}W_1(z) + \beta_{12}z^{-1}W_1(z) + \beta_{22}z^{-2}W_1(z) \\ &= \left(\beta_{02} + \beta_{12}z^{-1} + \beta_{22}z^{-2}\right) W_1(z) \end{aligned} \tag{9.13}$$

Substituting (9.13) into (9.12) yields:

$$W_2(z) = \left(\beta_{01} + \beta_{11}z^{-1} + \beta_{21}z^{-2}\right)\left(\beta_{02} + \beta_{12}z^{-1} + \beta_{22}z^{-2}\right) X(z) \tag{9.14}$$

Extending (9.14) to $y[n]$, we finally get:

$$\begin{aligned} Y(z) &= X(z)\left(\beta_{01} + \beta_{11}z^{-1} + \beta_{21}z^{-2}\right)\cdots\left(\beta_{0M_c} + \beta_{1M_c}z^{-1} + \beta_{2M_c}z^{-2}\right) \\ &= X(z)\prod_{k=1}^{M_c}\left(\beta_{0k} + \beta_{1k}z^{-1} + \beta_{2k}z^{-2}\right) \end{aligned} \tag{9.15}$$

To save the computational complexity, we express (9.9) as:

$$H(z) = G \prod_{k=1}^{M_c} \left(1 + \beta'_{1k} z^{-1} + \beta'_{2k} z^{-2}\right) \qquad (9.16)$$

where $G = \beta_{01}\beta_{02}\cdots\beta_{0M_c}$, $\beta'_{1k} = \beta_{1k}/\beta_{0k}$ and $\beta'_{2k} = \beta_{2k}/\beta_{0k}$, $k = 1, 2, \cdots, M_c$. That is, all $\{\beta_{0k}\}$ are normalized to 1.

Assuming that M is even, (9.16) needs M delay elements, $(M+1)$ multiplications and M additions, for computing each output value of $y[n]$

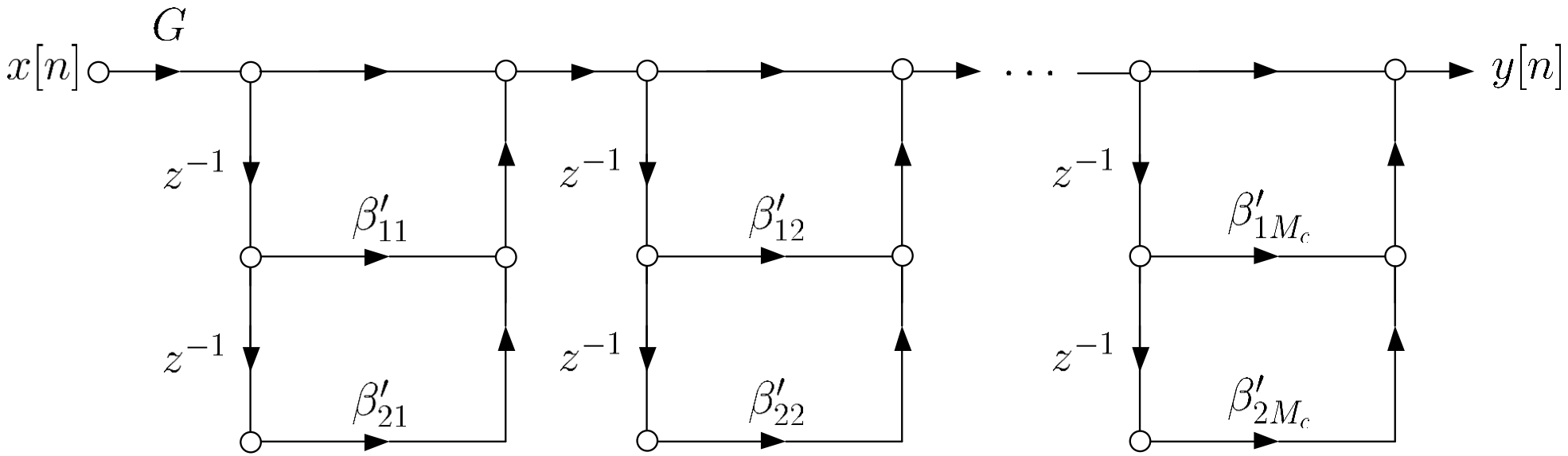


Fig.9.7: Cascade form of FIR filter with smaller complexity

Example 9.2

Draw the signal flow graph using the cascade form for the LTI system whose transfer function is:

$$H(z) = \sum_{k=0}^{4} z^{-k} = 1 + z^{-1} + z^{-2} + z^{-3} + z^{-4}$$

To factorize $H(z)$, we use the MATLAB command `roots([1 1 1 1 1])` to solve for the roots:

```
 0.3090 + 0.9511i
 0.3090 - 0.9511i
-0.8090 + 0.5878i
-0.8090 - 0.5878i
```

Hence $H(z)$ can be factorized as

$$H(z) = \left(1 - [0.309 + j0.9511]z^{-1}\right)\left(1 - [0.309 - j0.9511]z^{-1}\right) \times \left(1 - [-0.809 + j0.5878]z^{-1}\right)\left(1 - [-0.809 - j0.5878]z^{-1}\right)$$

Although it can be realized with first-order sections, complex coefficients are needed, which implies higher computational cost. To guarantee real-valued coefficients, we group the sections of complex conjugates together:

$$H(z) = \left(1 - 0.618z^{-1} + z^{-2}\right)\left(1 + 1.618z^{-1} + z^{-2}\right)$$
$$= \left(1 + 1.618z^{-1} + z^{-2}\right)\left(1 - 0.618z^{-1} + z^{-2}\right)$$

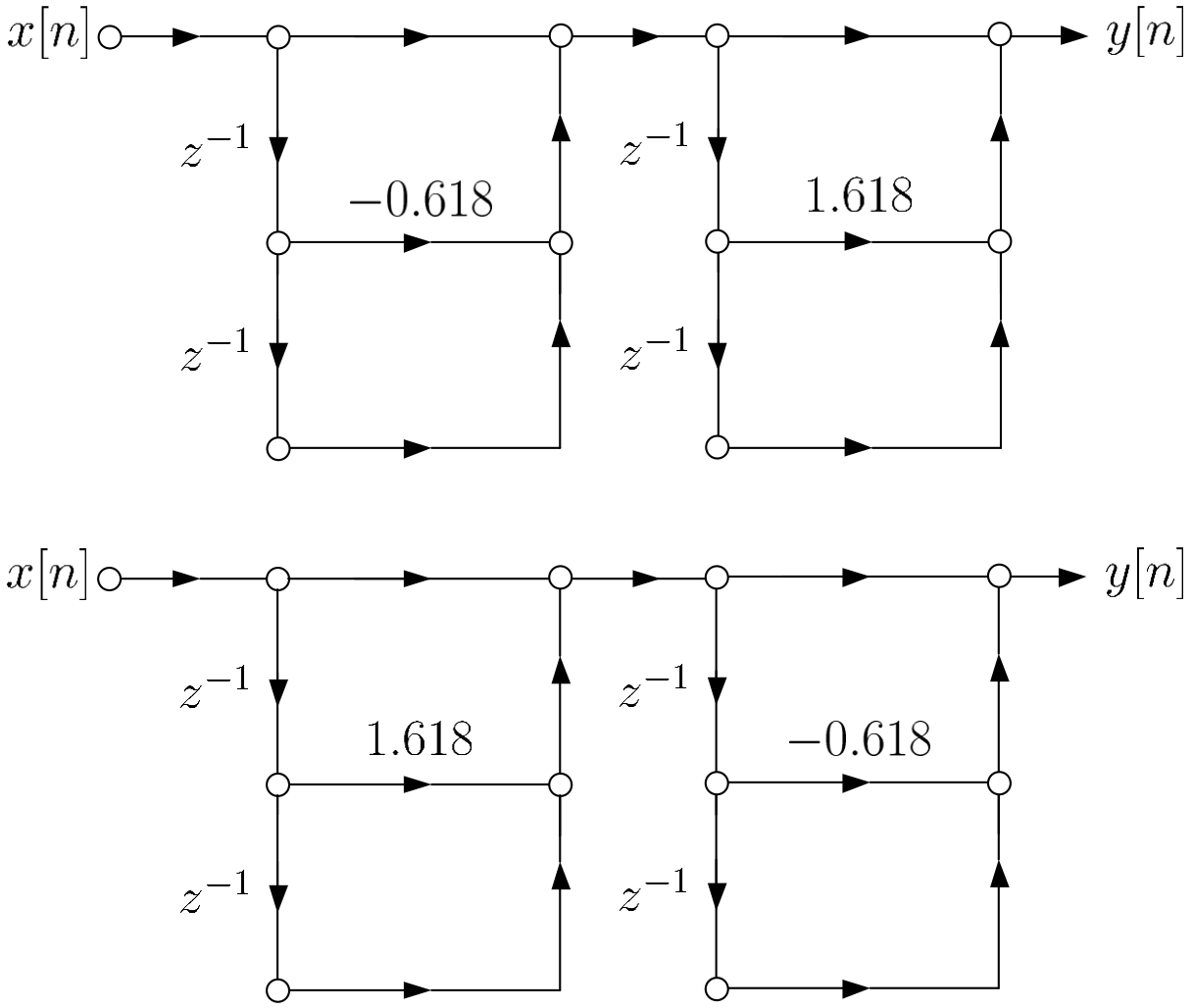


Fig.9.8: Two possible cascade forms for FIR filter

Structures for IIR Filter

When there is at least one pole in $H(z)$, it corresponds to an IIR filter. The corresponding transfer function is thus:

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \tag{9.17}$$

That is, IIR filter is the general form of any discrete-time LTI system.

1. Direct Form

It realizes (9.4) in an explicit manner via decomposing it into a pair of difference equations:

$$v[n] = \sum_{k=0}^{M} b_k x[n-k] \tag{9.18}$$

and

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + v[n] \tag{9.19}$$

The direct form can also be obtained by decomposing $H(z)$ into two transfer functions as:

$$H(z) = H_1(z) \cdot H_2(z) \tag{9.20}$$

where

$$H_1(z) = \sum_{k=0}^{M} b_k z^{-k} \tag{9.21}$$

and

$$H_2(z) = \frac{1}{1 + \sum_{k=1}^{N} a_k z^{-k}} \tag{9.22}$$

In the z transform domain, we have:

$$V(z) = X(z)H_1(z) = X(z)\sum_{k=0}^{M} b_k z^{-k} \tag{9.23}$$

and

$$Y(z) = V(z)H_2(z) = \frac{V(z)}{1 + \sum_{k=1}^{N} a_k z^{-k}} \tag{9.24}$$

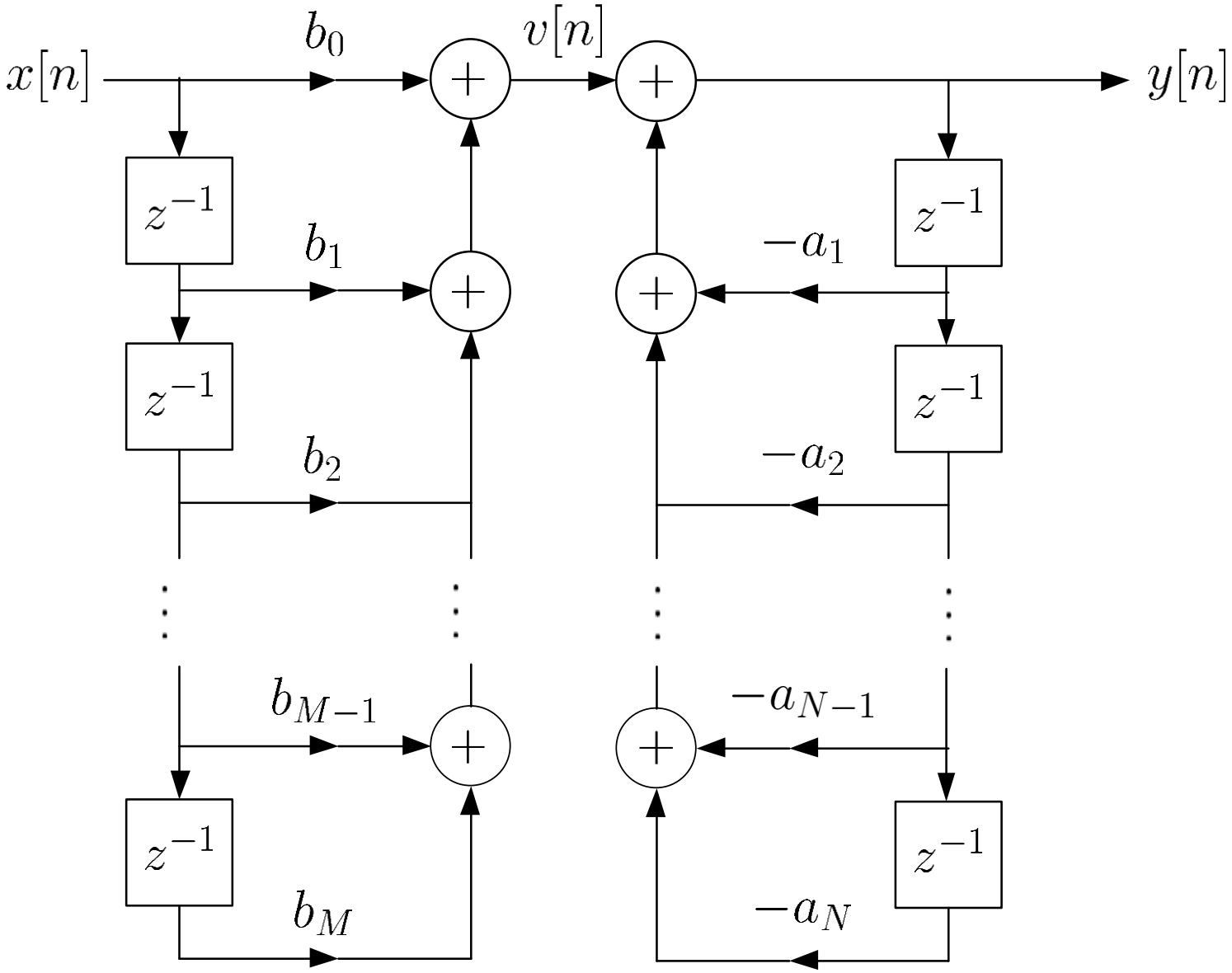


Fig.9.9: Direct form of IIR filter

The direct form implementation needs $(M+N)$ memory locations, $(M+N+1)$ multiplications and $(M+N)$ additions, for computing each output value of $y[n]$.

2. Canonic Form

On the other hand, we can first pass $x[n]$ through the filter $H_2(z)$ to produce an intermediate signal $w[n]$. The $w[n]$ is then passed through the system $H_1(z)$ to give $y[n]$:

$$W(z) = X(z)H_2(z) = \frac{X(z)}{1 + \sum_{k=1}^{N} a_k z^{-k}} \tag{9.25}$$

and

$$Y(z) = W(z)H_1(z) = W(z)\sum_{k=0}^{M} b_k z^{-k} \tag{9.26}$$

Applying inverse z transform, we get:

$$w[n] = -\sum_{k=1}^{N} a_k w[n-k] + x[n] \tag{9.27}$$

and

$$y[n] = \sum_{k=0}^{M} b_k w[n-k] \tag{9.28}$$

which can be considered as an alternative direct form.

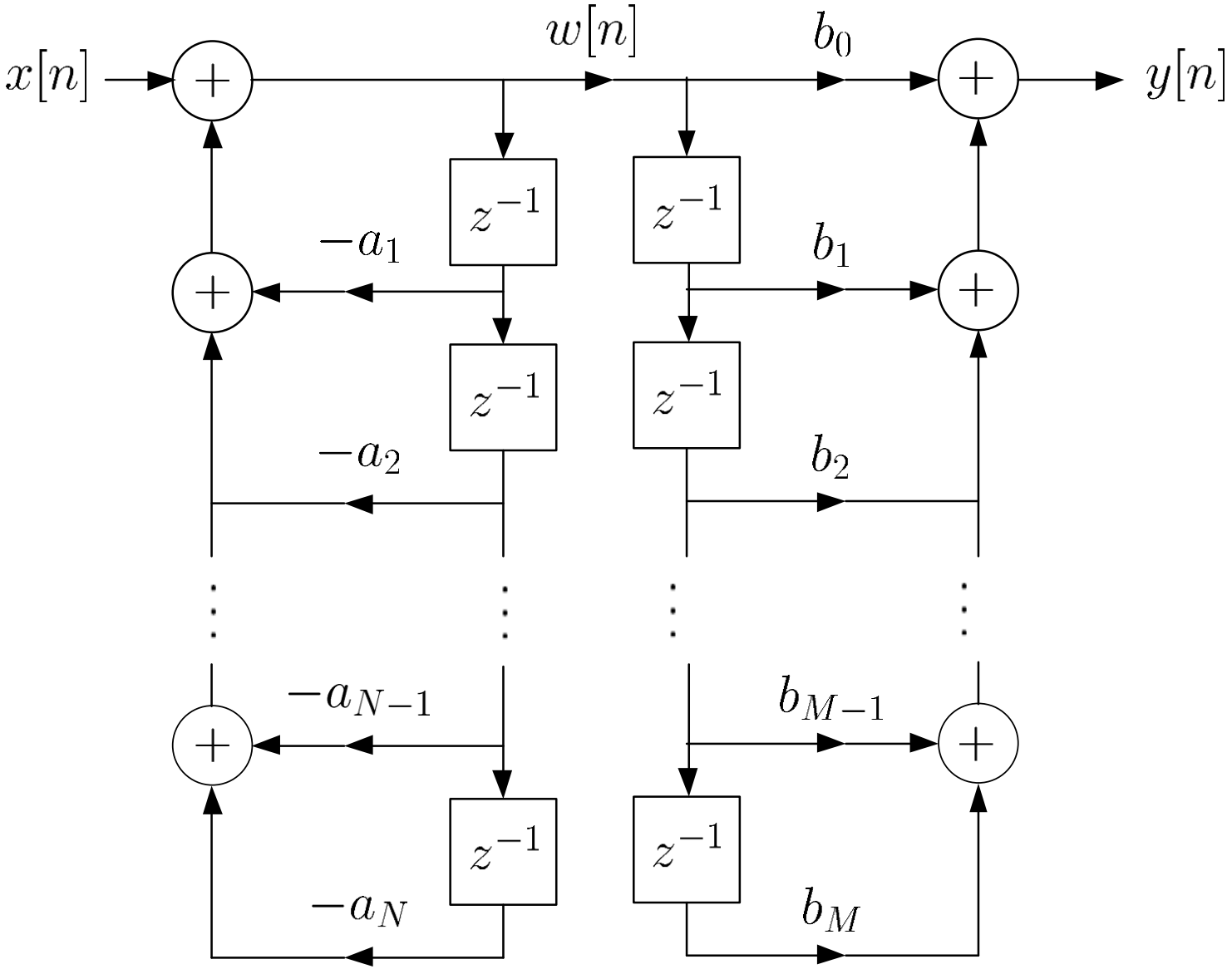


Fig.9.10: Alternative direct form of IIR filter

Assume $M = N$. Since the same signals $w[n], w[n-1], \cdots, w[n-N]$, are stored in the two chains of storage elements, they can be combined to reduce the memory requirement.

In general, the minimum number of delay elements required is $\max(M, N)$.

It is called canonic form because this implementation involves the minimum number of storages.

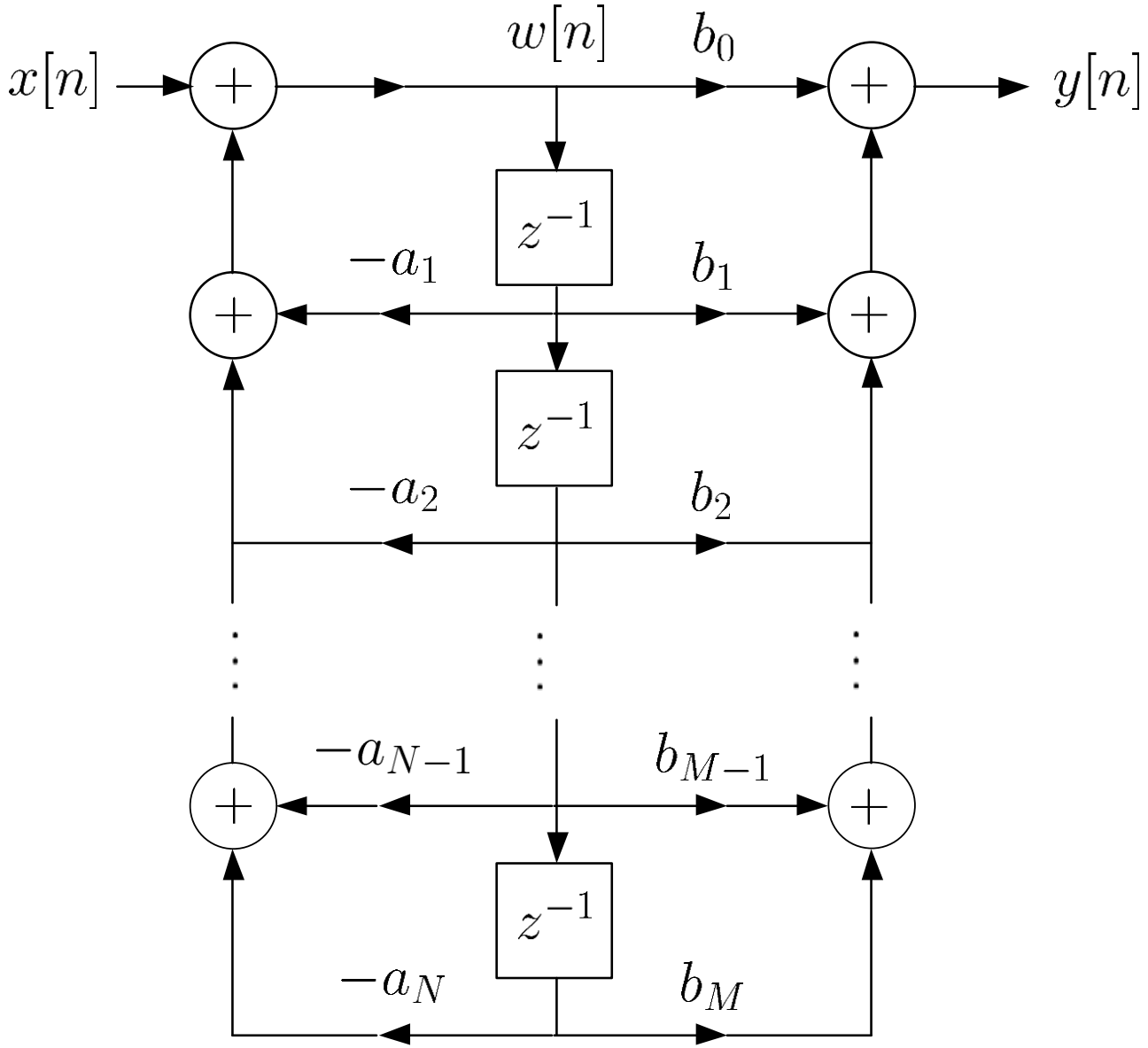


Fig.9.11: Canonic form of IIR filter

Example 9.3

Draw the block diagrams using the direct and canonic forms for the LTI system whose transfer function is:

$$H(z) = \frac{1 - 3z^{-1} + 2z^{-2}}{1 + 0.3z^{-1} - 0.1z^{-2}}$$

According to (9.18)-(9.19):

$$v[n] = x[n] - 3x[n-1] + 2x[n-2]$$
$$y[n] = -0.3y[n-1] + 0.1y[n-2] + v[n]$$

Based on (9.27)-(9.28):

$$w[n] = -0.3w[n-1] + 0.1w[n-2] + x[n]$$
$$y[n] = w[n] - 3w[n-1] + 2w[n-2]$$

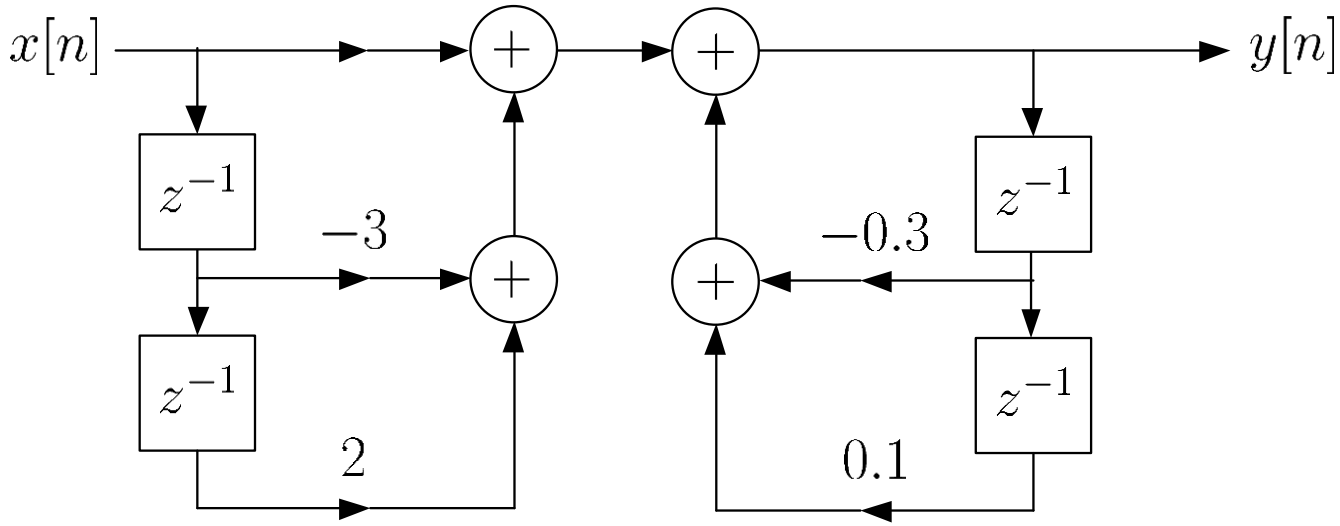


Fig.9.12: Direct form of second-order IIR filter

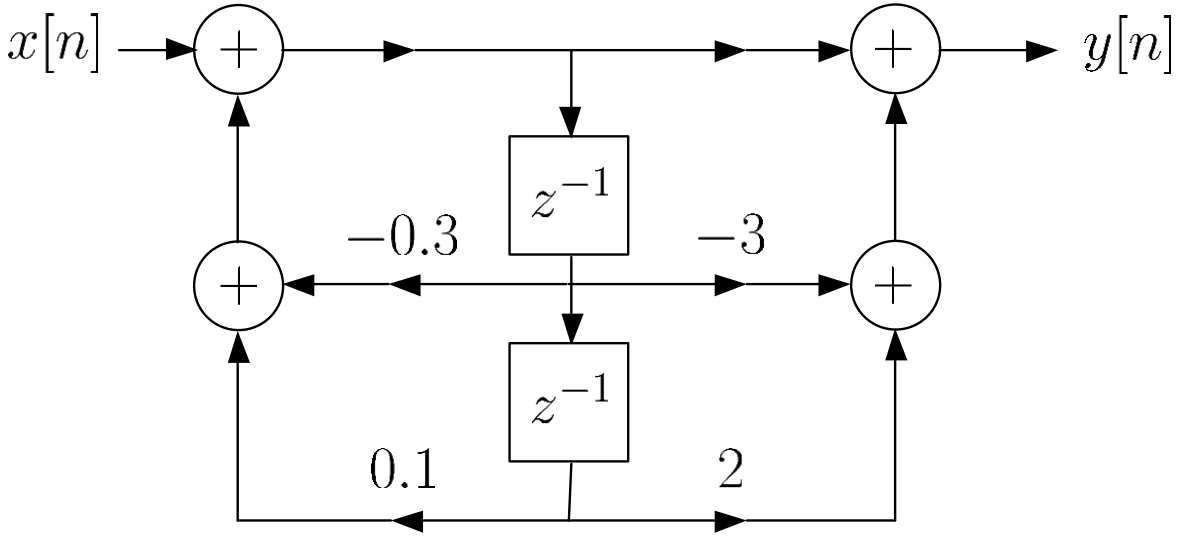


Fig.9.13: Canonic form of second-order IIR filter

3. Cascade Form

We factorize the numerator and denominator polynomials in terms of second-order polynomial system functions as:

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} = \prod_{k=1}^{N_c} \frac{\beta_{0k} + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \qquad (9.29)$$

Without loss of generality, it is assumed that $N \geq M$ so that $N_c = \lfloor (N+1)/2 \rfloor$. Note that when M or N is odd, one of the $\{\beta_{2k}\}$ or $\{\alpha_{2k}\}$ will be zero.

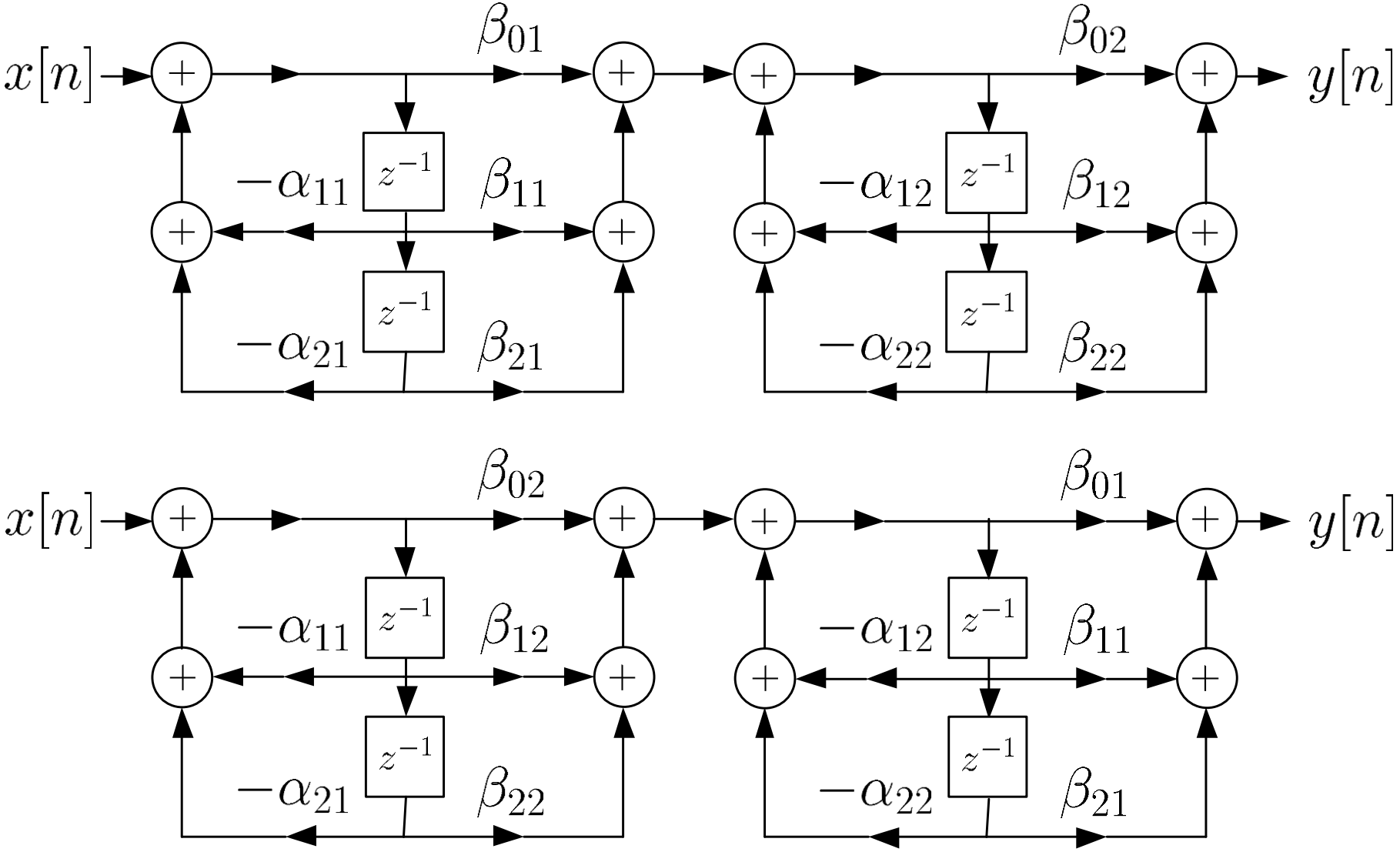
Each second-order subsystem

$$\frac{\beta_{0k} + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \tag{9.30}$$

can be realized in either the direct or canonic form. Nevertheless, the canonic form is preferred because it requires the minimum number of delay elements.

In IIR filter implementation, we can group the numerator and denominator of (9.30) in different ways, leading to different pole and zero combinations in each of the second-order sections.

For example, there are four possible cascade realizations for a fourth-order IIR filter with $M = N = 4$.


$x[n]$
β_{01}
β_{02}
$y[n]$
$-\alpha_{11}$
z^{-1}
β_{11}
$-\alpha_{12}$
z^{-1}
β_{12}
z^{-1}
z^{-1}
$-\alpha_{21}$
β_{21}
$-\alpha_{22}$
β_{22}
$x[n]$
β_{02}
β_{01}
$y[n]$
$-\alpha_{11}$
z^{-1}
β_{12}
$-\alpha_{12}$
z^{-1}
β_{11}
z^{-1}
z^{-1}
$-\alpha_{21}$
β_{22}
$-\alpha_{22}$
β_{21}

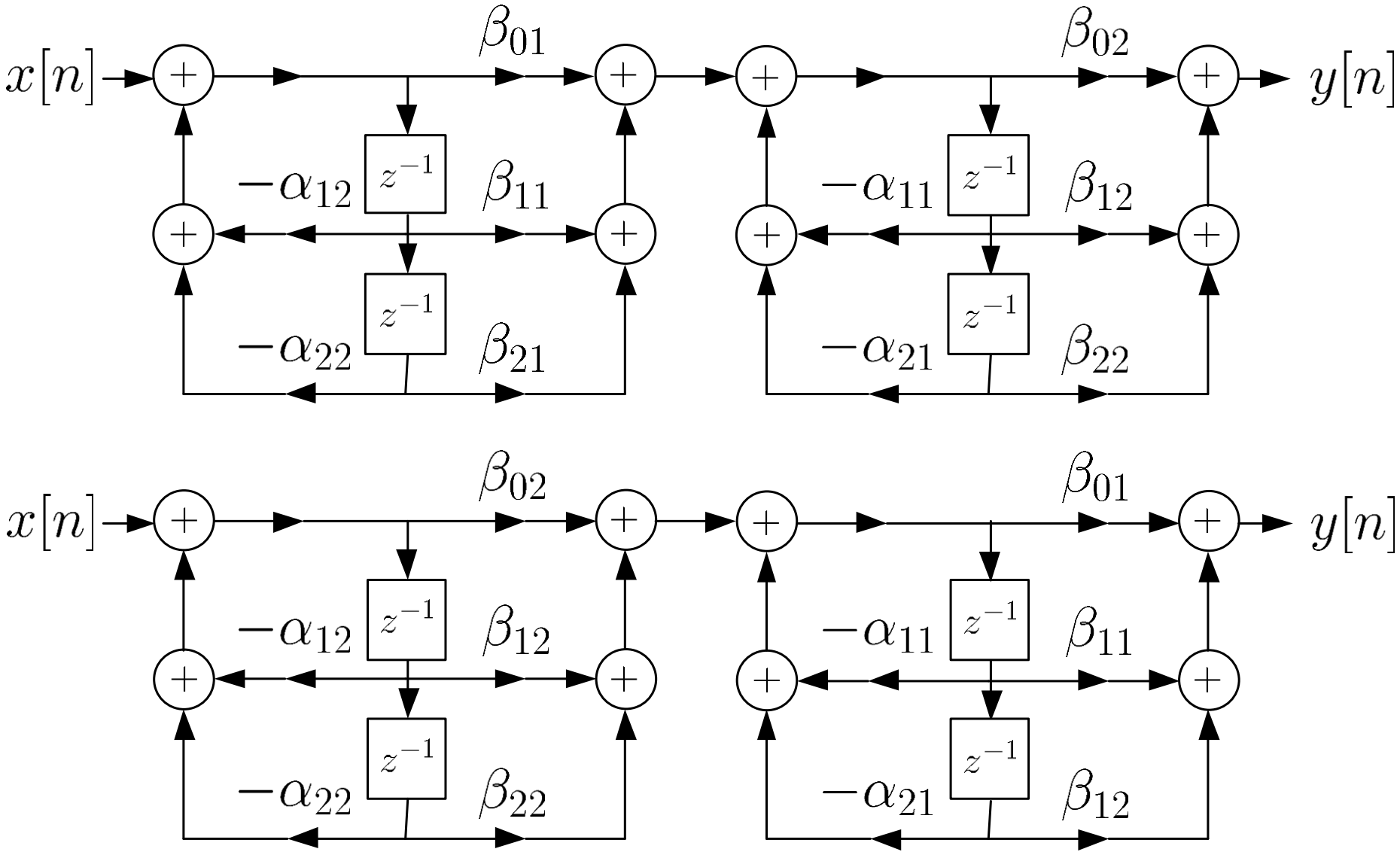


Fig.9.14: 4 possible cascade realizations for 4th-order IIR filter

To save the computational complexity, we express (9.29) as

$$H(z) = G \prod_{k=1}^{N_c} \frac{1 + \beta'_{1k} z^{-1} + \beta'_{2k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \qquad (9.31)$$

where $G = \beta_{01}\beta_{02} \cdots \beta_{0N_c}$, $\beta'_{1k} = \beta_{1k}/\beta_{0k}$ and $\beta'_{2k} = \beta_{2k}/\beta_{0k}$, $k = 1, 2, \cdots, N_c$.

Assuming that N is even with $N = M$, the cascade implementation of (9.31) needs N or $2N$ delay elements, $(2N + 1)$ multiplications and $2N$ additions, for computing each $y[n]$. That is, its memory and computational requirements are equal to those of the direct form.

Example 9.4

Draw the signal flow graph using the cascade form with first-order sections for the LTI system whose transfer function is:

$$H(z) = \frac{1 - 3z^{-1} + 2z^{-2}}{1 + 0.3z^{-1} - 0.1z^{-2}}$$

For each first-order section, canonic form is assumed.

Solving the quadratic equations of the numerator and denominator polynomials, we can factorize $H(z)$ as:

$$H(z) = \frac{\left(1 - 2z^{-1}\right)\left(1 - z^{-1}\right)}{\left(1 + 0.5z^{-1}\right)\left(1 - 0.2z^{-1}\right)}$$

There are four possible cascade forms for $H(z)$:

$$
\begin{aligned}
H(z) &= \frac{1-2z^{-1}}{1+0.5z^{-1}} \cdot \frac{1-z^{-1}}{1-0.2z^{-1}} \\
&= \frac{1-z^{-1}}{1+0.5z^{-1}} \cdot \frac{1-2z^{-1}}{1-0.2z^{-1}} \\
&= \frac{1-2z^{-1}}{1-0.2z^{-1}} \cdot \frac{1-z^{-1}}{1+0.5z^{-1}} \\
&= \frac{1-z^{-1}}{1-0.2z^{-1}} \cdot \frac{1-2z^{-1}}{1+0.5z^{-1}}
\end{aligned}
$$

Note that although all four realizations are equivalent for infinite precision, they may differ in actual implementation when finite-precision numbers are employed.

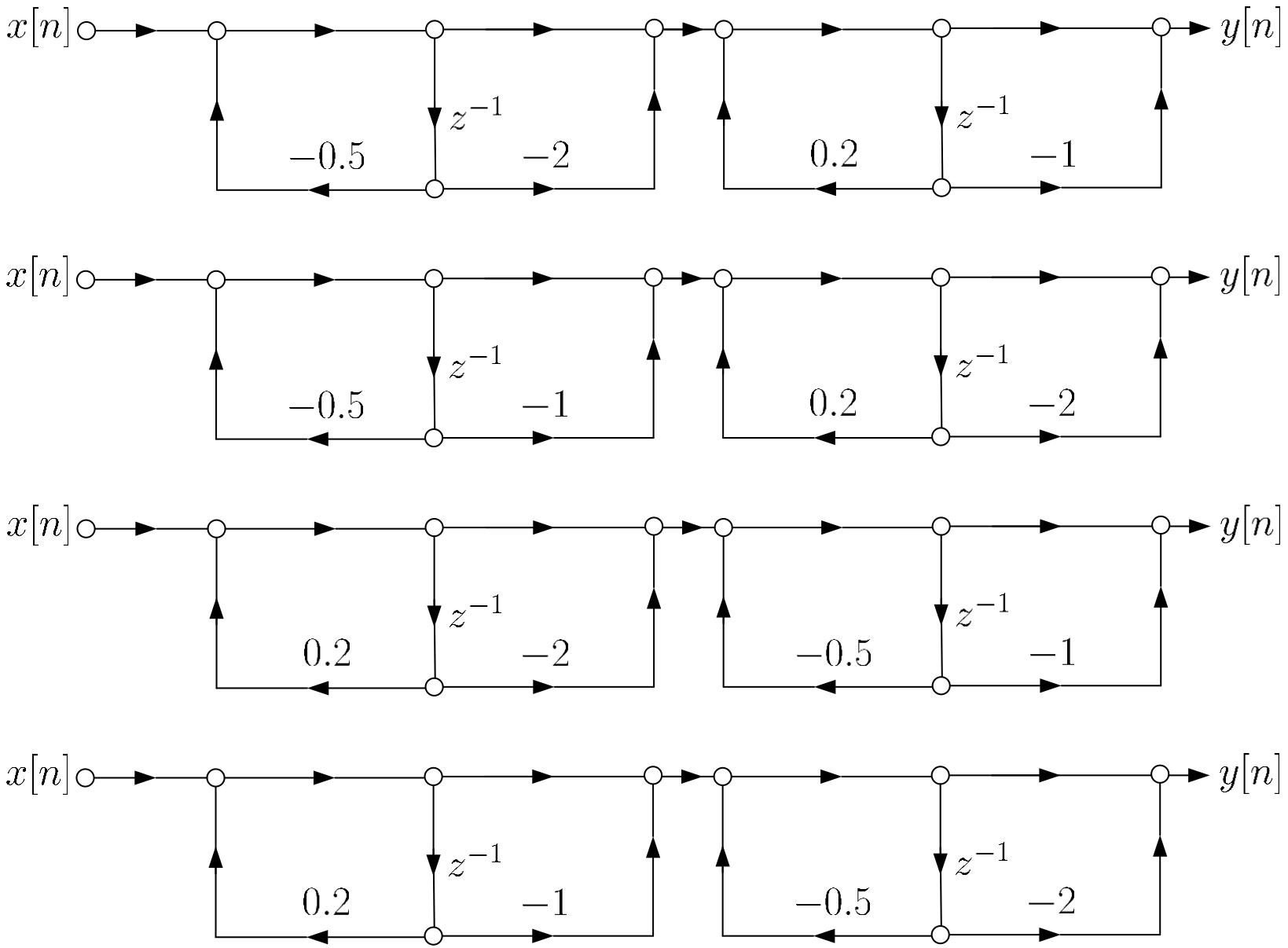


Fig.9.15: 4 possible cascade realizations with 1st-order sections

Example 9.5

Consider a LTI system whose transfer function $H(z)$ is:

$$H(z) = \frac{\left(1 - \frac{13}{10}z^{-1} + \frac{2}{5}z^{-2}\right)\left(1 - \frac{17}{12}z^{-1} + \frac{1}{2}z^{-2}\right)}{\left(1 - 3z^{-1} + 2z^{-2}\right)\left(1 - 7z^{-1} + 12z^{-2}\right)}$$

Find the number of possible combinations of cascade form with second-order sections. Determine if the system is stable or not.

Further factorizing $H(z)$ yields:

$$H(z) = \frac{\left(1 - \frac{1}{2}z^{-1}\right)\left(1 - \frac{2}{3}z^{-1}\right)\left(1 - \frac{3}{4}z^{-1}\right)\left(1 - \frac{4}{5}z^{-1}\right)}{\left(1 - z^{-1}\right)\left(1 - 2z^{-1}\right)\left(1 - 3z^{-1}\right)\left(1 - 4z^{-1}\right)}$$

For the numerator polynomials, there are 3 grouping possibilities in terms of second-order sections as follows:

$$\left(1-\frac{1}{2}z^{-1}\right)\left(1-\frac{2}{3}z^{-1}\right), \qquad \left(1-\frac{3}{4}z^{-1}\right)\left(1-\frac{4}{5}z^{-1}\right)$$
$$\left(1-\frac{1}{2}z^{-1}\right)\left(1-\frac{3}{4}z^{-1}\right), \qquad \left(1-\frac{2}{3}z^{-1}\right)\left(1-\frac{4}{5}z^{-1}\right)$$
$$\left(1-\frac{1}{2}z^{-1}\right)\left(1-\frac{4}{5}z^{-1}\right), \qquad \left(1-\frac{3}{4}z^{-1}\right)\left(1-\frac{2}{3}z^{-1}\right)$$

There are also 3 grouping possibilities for the denominator:

$$\left(1-z^{-1}\right)\left(1-2z^{-1}\right), \qquad \left(1-3z^{-1}\right)\left(1-4z^{-1}\right)$$
$$\left(1-z^{-1}\right)\left(1-3z^{-1}\right), \qquad \left(1-2z^{-1}\right)\left(1-4z^{-1}\right)$$
$$\left(1-z^{-1}\right)\left(1-4z^{-1}\right), \qquad \left(1-3z^{-1}\right)\left(1-2z^{-1}\right)$$

Each fourth-order IIR filter can be realized in four possible cascade forms.

As a result, the number of possible cascade form combinations with second-order sections for $H(z)$ is $3 \cdot 3 \cdot 4 = 36$.

When implementing a filter, causality is always assumed because it is impossible to realize a noncausal system where the output depends on future input. It is clear that the region of convergence (ROC) for causal $H(z)$ is $|z| > 4$. Since the ROC does not include the unit circle, the system is not stable.

In summary, a causal system is stable when all poles are inside the unit circle.

4. Parallel Form

The idea of the parallel form is similar to the partial fraction expansion of z transform:

$$H(z) = \sum_{l=0}^{M-N} B_l z^{-l} + \sum_{k=1}^{N_c} \frac{\gamma_{0k} + \gamma_{1k} z^{-1}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \qquad (9.32)$$

where $N_c = \lfloor (N+1)/2 \rfloor$. But now we use second-order sections in order to ensure all $\{\gamma_{0k}\}$, $\{\gamma_{1k}\}$, $\{\alpha_{1k}\}$ and $\{\alpha_{2k}\}$ are real.

Note that when $M < N$, the first summation term in (9.32) will not be included.

Example 9.6

Draw the block diagram using parallel form for a LTI system whose transfer function is:

$$H(z) = \frac{1 - 3z^{-1} + 2z^{-2}}{1 + 0.3z^{-1} - 0.1z^{-2}}$$

Following the long division as in Example 5.9, we obtain:

$$H(z) = -20 + \frac{21 + 3z^{-1}}{1 + 0.3z^{-1} - 0.1z^{-2}}$$

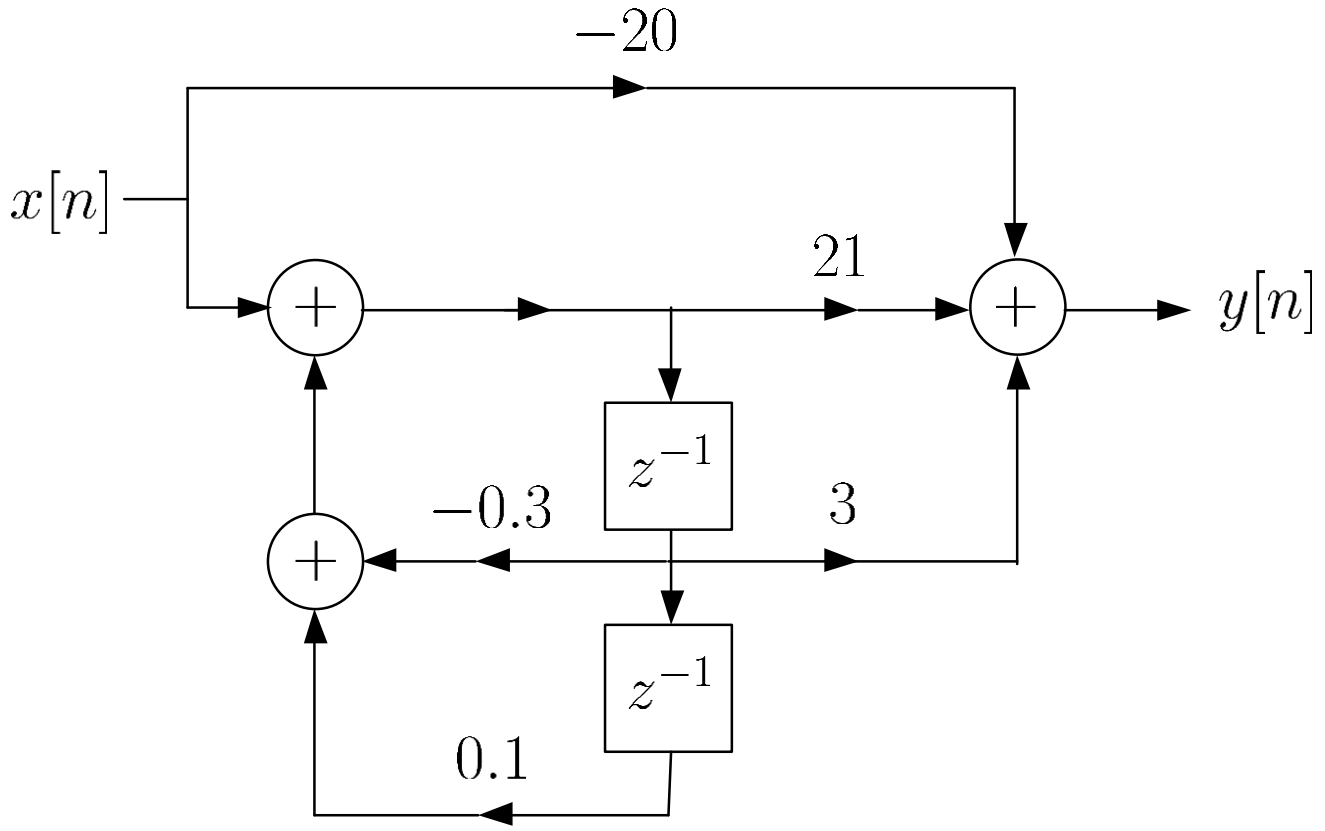


Fig.9.16: Parallel form with second-order section

As the poles of $H(z)$ are real, we can also express $H(z)$ in terms of first-order sections as:

$$H(z) = -20 + \frac{\frac{75}{7}}{1 + 0.5z^{-1}} + \frac{\frac{72}{7}}{1 - 0.2z^{-1}}$$

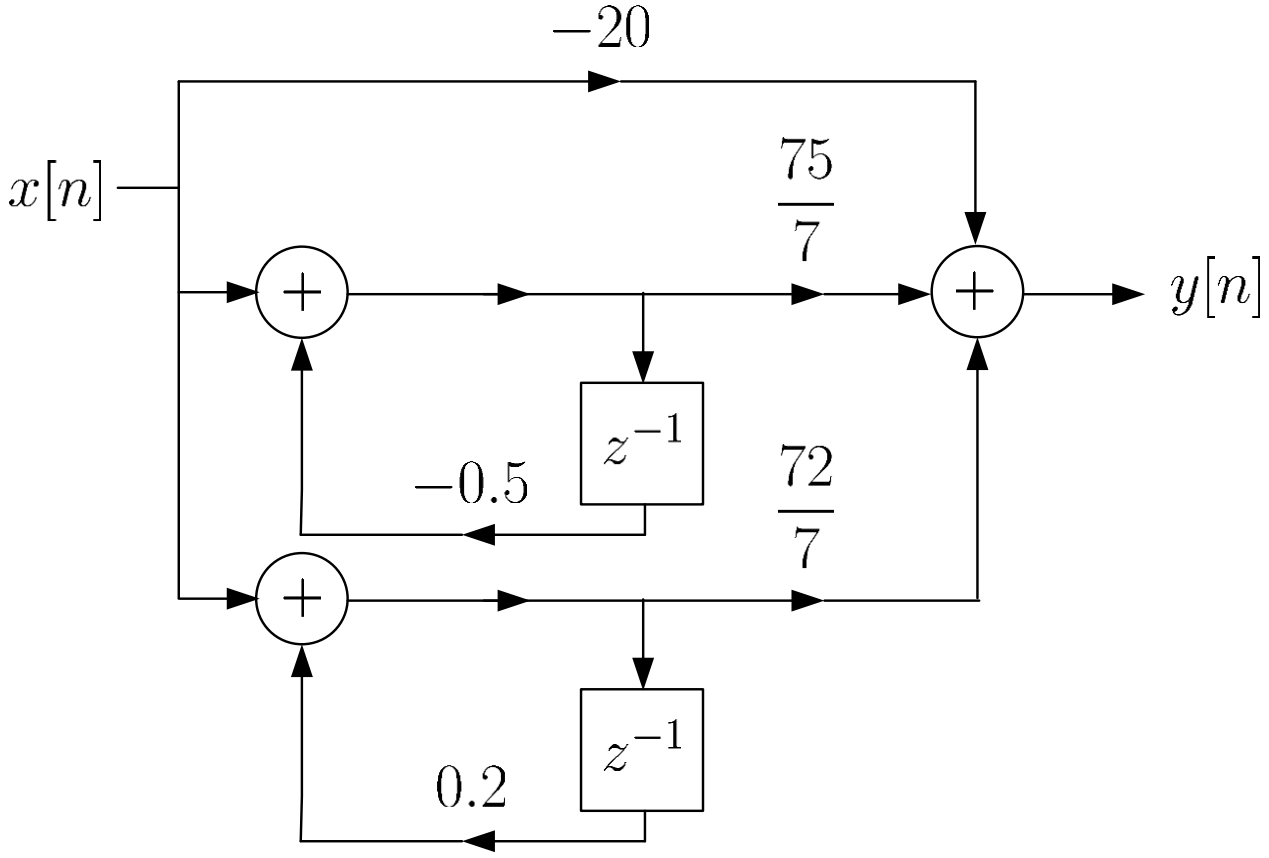


Fig.9.17: Parallel form with first-order sections

Example 9.7

Determine the transfer function $H(z)$ and the difference equation which relates $x[n]$ and $y[n]$ for

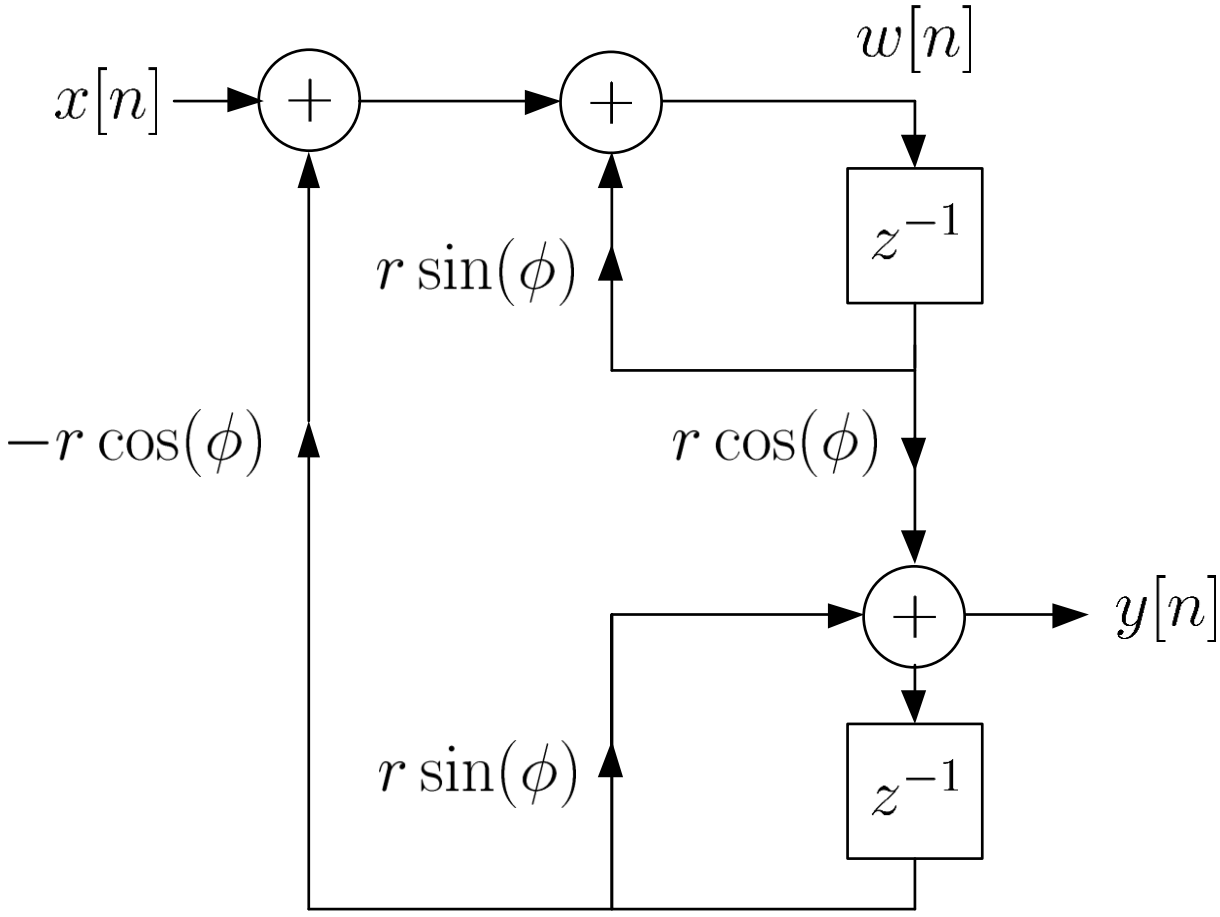


Fig.9.18: LTI system parameterized by r and ϕ

We first introduce an intermediate sequence $w[n]$ to relate $x[n]$ and $y[n]$. Then we can establish:

$$w[n] = w[n-1] \cdot r\sin(\phi) - y[n-1] \cdot r\cos(\phi) + x[n]$$

and

$$w[n-1] \cdot r\cos(\phi) + y[n-1] \cdot r\sin(\phi) = y[n]$$

Applying z transform yields:

$$W(z) = W(z)z^{-1}r\sin(\phi) - Y(z)z^{-1}r\cos(\phi) + X(z)$$

and

$$W(z)z^{-1}r\cos(\phi) + Y(z)z^{-1}r\sin(\phi) = Y(z)$$

From the second equation, we have:

$$W(z) = \frac{1 - z^{-1}r\sin(\phi)}{z^{-1}r\cos(\phi)}Y(z)$$

Substituting it into the first equation, we finally have:

$$\frac{1 - z^{-1} r \sin(\phi)}{z^{-1} r \cos(\phi)} Y(z) \left(1 - z^{-1} r \sin(\phi)\right) = -Y(z) z^{-1} r \cos(\phi) + X(z)$$
$$\Rightarrow Y(z) \left(1 - 2z^{-1} r \sin(\phi) + z^{-2} r^2\right) = X(z) z^{-1} r \cos(\phi)$$
$$\Rightarrow H(z) = \frac{Y(z)}{X(z)} = \frac{z^{-1} r \cos(\phi)}{1 - 2z^{-1} r \sin(\phi) + z^{-2} r^2}$$

Taking the inverse z transform gives:

$$y[n] - 2r \sin(\phi) y[n-1] + r^2 y[n-2] = r \cos(\phi) x[n-1]$$

Comparison of Different Structures

The major factors that affect our choice of a specific realization are computational complexity, memory requirement, and finite word-length effects. Assuming that M is even with $M = N$:

Structure	Multiplication	Addition	Register
Direct form	$M+1$	M	M
Cascade form	$M+1$	M	M

Table 9.1: FIR filter structure comparison

Structure	Multiplication	Addition	Register
Direct form	$2M+1$	$2M$	$2M$
Canonic form	$2M+1$	$2M$	M
Cascade form	$2M+1$	$2M$	M
Parallel form	$2M+1$	$3M/2+1$	M

Table 9.2: IIR filter structure comparison

Computations of the direct form can be reduced if the FIR filter coefficients are symmetric or anti-symmetric.

When the filter coefficients are expressed using infinite precision numbers, all realizations are same. However, in practice, they are processed in registers which have finite word-lengths. In the presence of quantization errors, the cascade and parallel realizations are more robust than the direct and canonic forms, that is, they have frequency responses closer to the desired responses.

FIR filters are less sensitive than IIR filters to finite word-length effects.

For a feasible system, it should be causal and stable.

In cascade and parallel realizations of IIR filters, system stability can be easily monitored by checking pole locations.

Chapter 10: FIR Filter Design

Chapter Intended Learning Outcomes:

(i) Understanding of the characteristics of linear-phase finite impulse response (FIR) filters

(ii) Ability to design linear-phase FIR filters according to predefined specifications using the window and frequency sampling methods

Steps in Digital Filter Design

1. Specification Determination

The first step is to obtain the filter specifications or requirements which are determined by the applications.

Suppose the requirement is a lowpass filter with cutoff frequency of ω_c to achieve a task of noise reduction

The filter specification can be described by discrete-time Fourier transform (DTFT) $H(e^{j\omega})$:

$$H(e^{j\omega}) = \begin{cases} 1, & |\omega| < \omega_c \\ 0, & -\pi < \omega < -\omega_c, \quad \omega_c < \omega < \pi \end{cases} \tag{10.1}$$

which specifies both the magnitude and phase

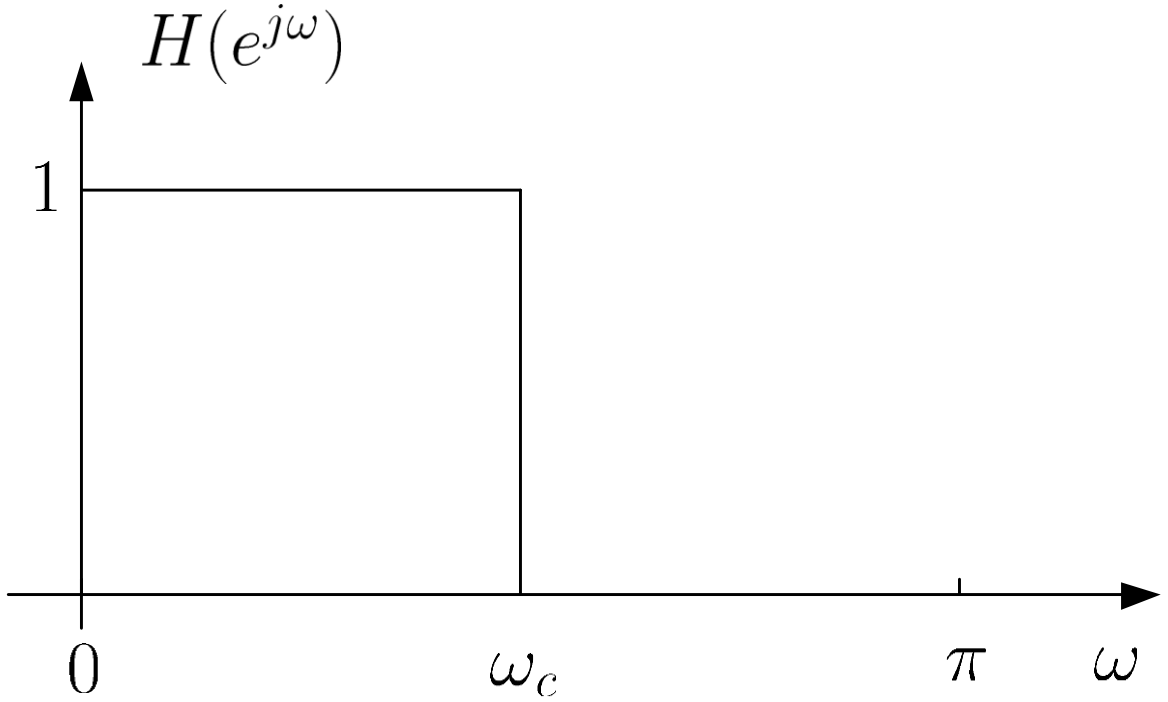


Fig.10.1: Ideal lowpass filter

- unity gain for the whole range of $\omega \in (0, \omega_c)$
- complete suppression for $\omega \in (\omega_c, \pi)$
- step change in frequency response at $\omega = \omega_c$

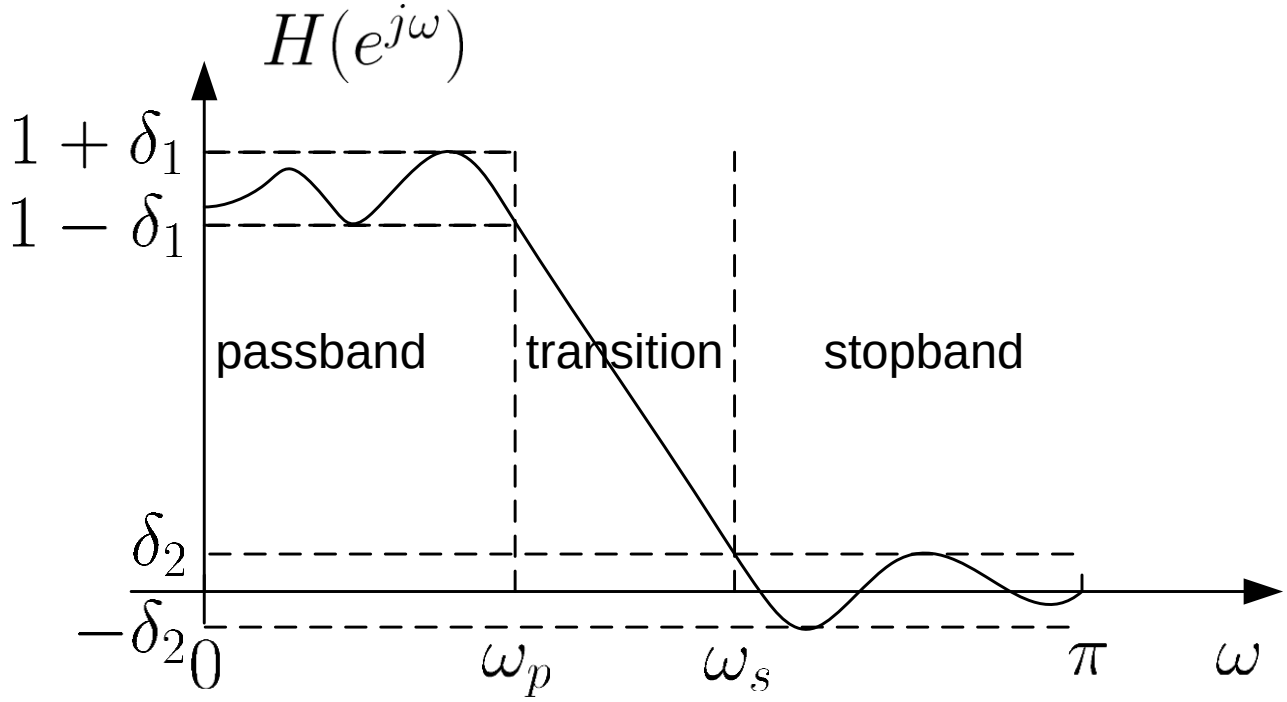


Fig.10.2: Practical $H(e^{j\omega})$

- passband corresponds to $\omega \in (0, \omega_p)$ where ω_p is the passband frequency and δ_1 is the passband ripple or tolerance which is the maximum allowable deviation from unity in this band

- stopband corresponds to $\omega \in (\omega_s, \pi)$ where ω_s is the stopband frequency and δ_2 is the stopband ripple or tolerance which is the maximum allowable deviation from zero in this band
- transition band corresponds to $\omega_c \in (\omega_p, \omega_s)$ where there are no restrictions on $H(e^{j\omega})$ in this band

2. Filter Response Calculation

We then use digital signal processing techniques to obtain a filter description in terms of transfer function $H(z)$ or impulse response $h[n]$ that fulfills the given specifications

3. Implementation

When $H(z)$ or $h[n]$ are known, the filter can then be realized in hardware or software according to a given structure

Advantages of FIR Filter

Transfer function of a causal FIR filter $H(z)$ with length N is:

$$H(z) = \sum_{n=0}^{N-1} h[n] z^{-n} \qquad (10.2)$$

where $h[n]$ is the finite-duration impulse response

- Phase response can be exactly linear which results in computation reduction and zero phase distortion. Note that when there is phase distortion, different frequency components of a signal will undergo different delays in the filtering process
- It is always stable because $\sum_{n=0}^{N-1} |h[n]| < \infty$ for finite N
- Less sensitive to finite word-length effects

- Efficient implementation via digital signal processors with multiply-and-add (MAC) instruction
- Existence of optimality theorem for FIR filter design

Linear-Phase FIR Filter

The phase response of a linear-phase filter is a linear function of the frequency ω:

$$\angle H(e^{j\omega}) = \phi - \tau\omega, \quad -\pi < \omega < \pi \tag{10.3}$$

where ϕ equals 0 or $\pi/2$ and τ is a constant which is function of the filter length.

Note that (10.3) is slightly different from that in previous chapters where the phase response complements with the magnitude response $|H(e^{j\omega})|$.

Here, the phase response is related to $H(e^{j\omega})$ by

$$H(e^{j\omega}) = H_r(e^{j\omega})e^{j\angle H(e^{j\omega})} \tag{10.4}$$

where $H_r(e^{j\omega})$ is real and is called the amplitude response.

Comparing

$$H(e^{j\omega}) = |H(e^{j\omega})|e^{j\angle H(e^{j\omega})} \tag{10.5}$$

where $|H(e^{j\omega})|$ is nonnegative, we see that when $H_r(e^{j\omega})$ is positive, the phase responses in (10.4) and (10.5) are identical but there is a phase difference of π for a negative $H_r(e^{j\omega})$.

Example 10.1

Determine the amplitude response and the corresponding phase response for a FIR filter with impulse response $h[n]$:

$$h[n] = u[n] - u[n-N]$$

Plot the spectra for $N = 10$. Compare the results with the magnitude response and the corresponding phase response.

Using Example 6.2:

$$H(e^{j\omega}) = \frac{\sin(\omega N/2)}{\sin(\omega/2)} \cdot e^{-j\omega(N-1)/2}$$

Hence the amplitude response is

$$H_r(e^{j\omega}) = \frac{\sin(\omega N/2)}{\sin(\omega/2)}$$

and the corresponding phase response is

$$\angle H(e^{j\omega}) = -\frac{N-1}{2}\omega$$

On the other hand, the magnitude response and the corresponding phase response are:

$$|H(e^{j\omega})| = \left|\frac{\sin(\omega N/2)}{\sin(\omega/2)}\right|$$

and

$$\angle(H(e^{j\omega})) = -\frac{\omega(N-1)}{2} + \angle\left(\frac{\sin(\omega N/2)}{\sin(\omega/2)}\right)$$

The MATLAB program is provided as `ex10_1.m`.

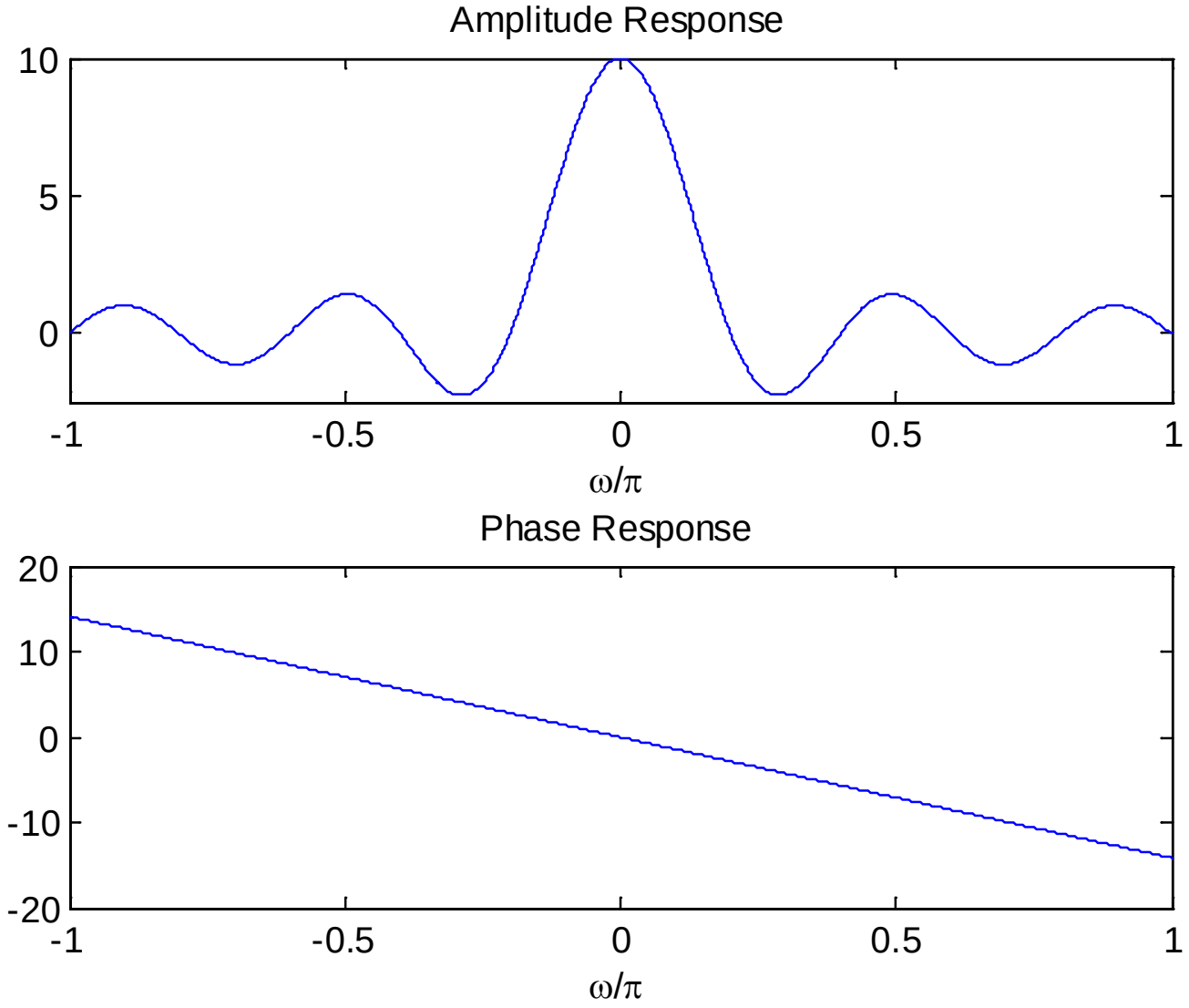


Fig.10.3: Amplitude and phase spectra

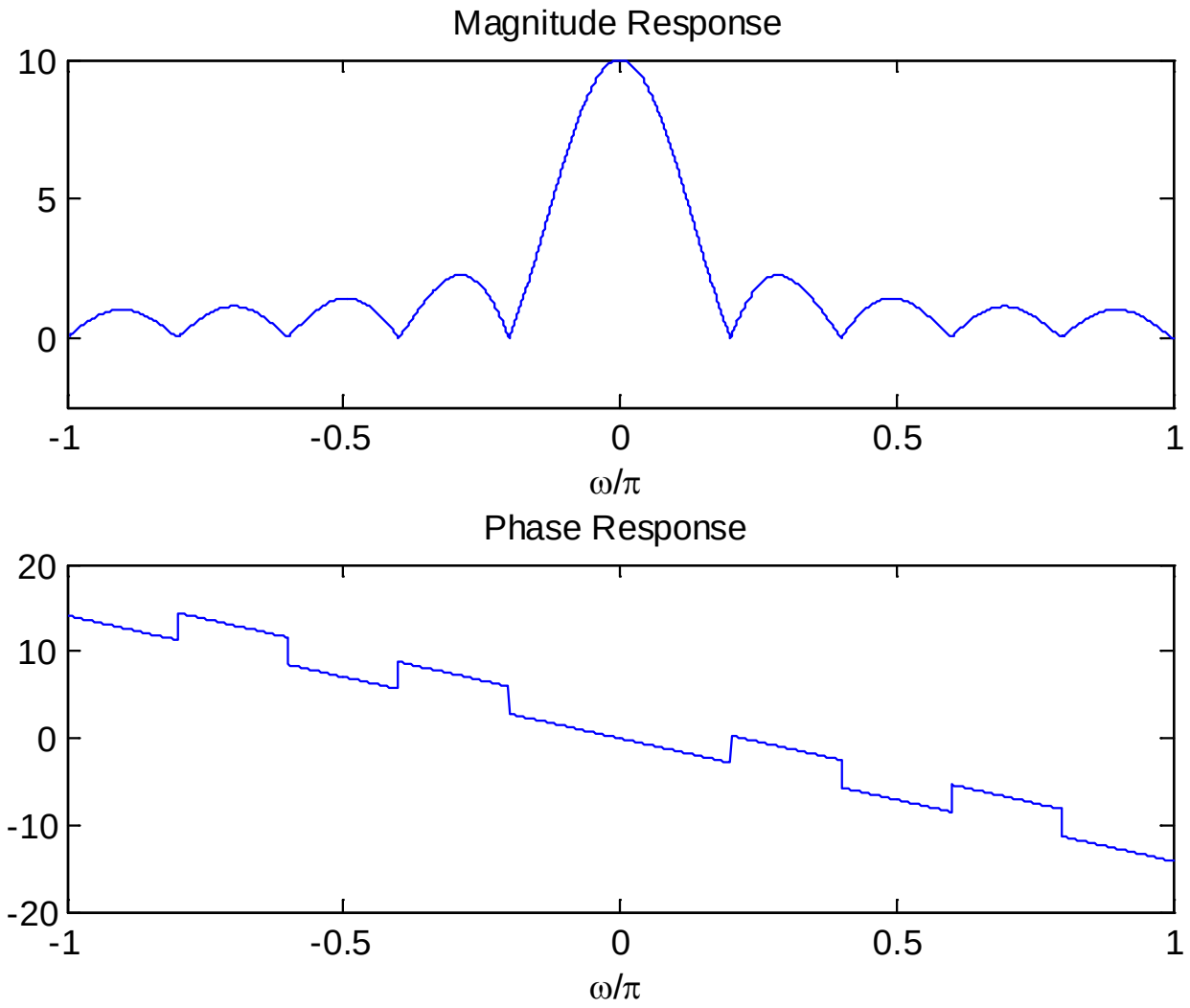


Fig.10.4: Magnitude and phase spectra

For a causal linear-phase FIR filter, the impulse response $h[n]$ is either symmetric or anti-symmetric:

$$h[n] = h[N-1-n], \quad 0 \le n \le N-1 \tag{10.6}$$

or

$$h[n] = -h[N-1-n], \quad 0 \le n \le N-1 \tag{10.7}$$

For example, when N is even in (10.7):

$$h[0] = -h[N-1],\ h[1] = -h[N-2],\ \cdots,\ h[N/2-1] = -h[N/2]$$

$$\begin{aligned} y[n] &= \sum_{k=0}^{N-1} h[k]x[n-k] \\ &= h[0]x[n] + h[1]x[n-1] + \cdots + h[N/2-1]x[n-N/2+1] \\ &\quad + h[N/2]x[n-N/2] + \cdots + h[N-1]x[n-N+1] \\ &= h[0]\left(x[n] - x[n-N+1]\right) + h[1]\left(x[n-1] - x[n-N+2]\right) + \cdots \\ &\quad + h[N/2-1]\left(x[n-N/2+1] - x[n-N/2]\right) \end{aligned} \tag{10.8}$$

How many multipliers and additions are needed?

Example 10.2

Draw the block diagram using the direct form with minimum number of multiplications for the FIR filter whose impulse response is:

$$h[n] = h[4-n], \quad 0 \le n \le 4$$

Following (10.8), we obtain:

$$\begin{aligned} y[n] &= \sum_{k=0}^{4} h[k]x[n-k] \\ &= h[0]\left(x[n] + x[n-4]\right) + h[1]\left(x[n-1] + x[n-3]\right) + h[2]x[n-2] \end{aligned}$$

The number of multiplications is reduced from 5 to 3 while the number of additions remains unchanged, which is 4.

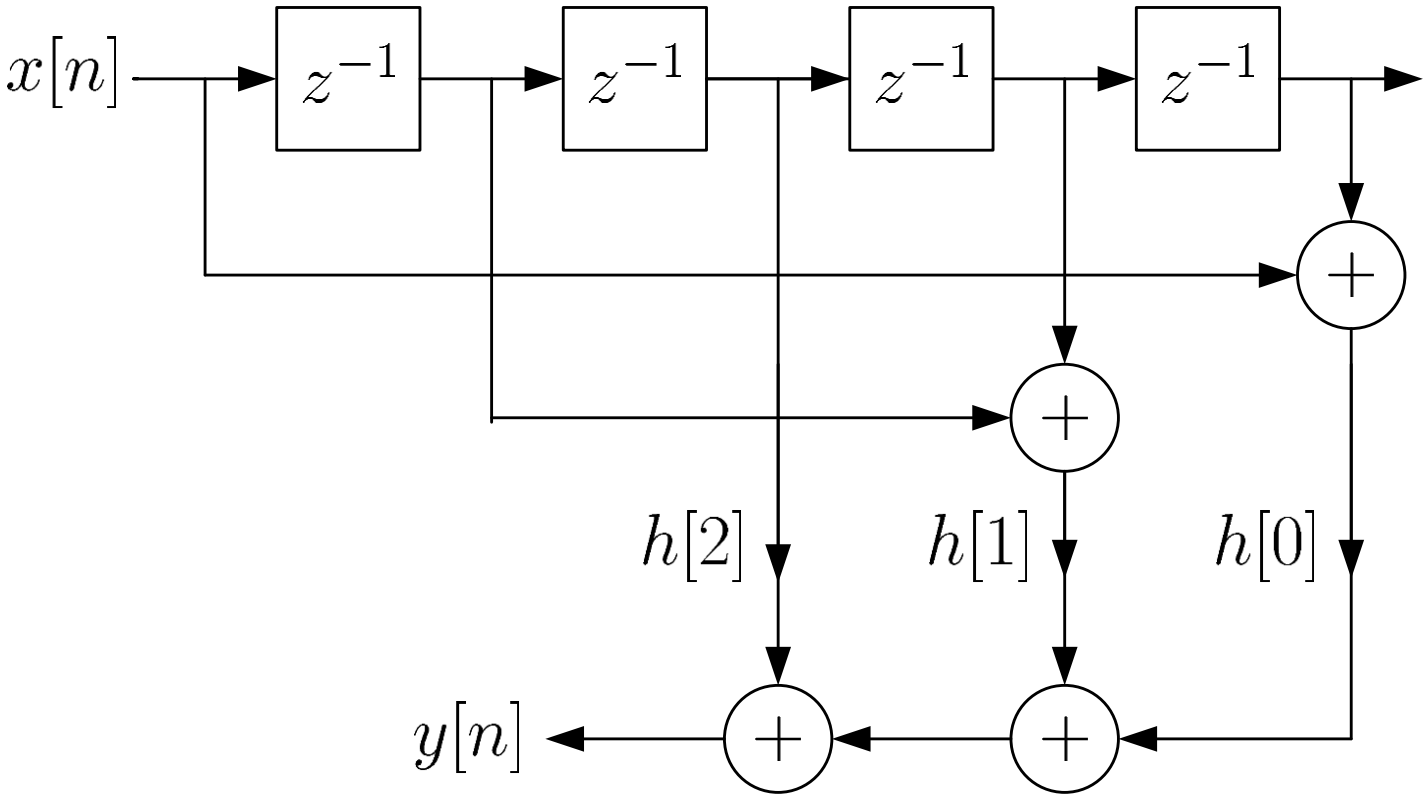


Fig.10.5: Block diagram for symmetric impulse response

There are four types of linear-phase FIR filters:

1. Symmetric Impulse Response with Odd N

Taking the DTFT of $h[n]$ yields

$$H(e^{j\omega}) = \left(2 \sum_{n=0}^{\frac{N-1}{2}-1} h[n] \cos\left(\left(\frac{N-1}{2} - n \right) \omega \right) + h[(N-1)/2] \right) e^{-j\omega(N-1)/2} \quad (10.9)$$

where $\phi = 0$ and $\tau = (N-1)/2$ is an integer.

Example 10.3

The impulse response $h[n]$ of a causal FIR filter satisfies

$$h[n] = h[N-1-n],\ 0 \leq n \leq N-1$$

with N is odd. Show that

$$\angle H(e^{j\omega}) = -\frac{N-1}{2}\omega$$

According to (10.6) with odd N, we have

$$\begin{aligned} h[0] &= h[N-1] \\ h[1] &= h[N-2] \\ &\cdots\cdots\cdots\cdots \\ h[(N-1)/2-1] &= h[(N-1)/2+1] \end{aligned}$$

Hence

$$
\begin{aligned}
H(e^{j\omega}) &= \sum_{n=0}^{N-1} h[n]e^{-j\omega n} \\
&= h[0] + \cdots + h[(N-1)/2-1]e^{-j\omega((N-1)/2-1)} + h[(N-1)/2]e^{-j\omega(N-1)/2} \\
&\quad + h[(N-1)/2+1]e^{-j\omega((N-1)/2+1)} + \cdots + h[N-1]e^{-j\omega(N-1)} \\
&= e^{-j\omega(N-1)/2}\Big(h[0]e^{j\omega(N-1)/2} + h[N-1]e^{-j\omega(N-1)/2} \\
&\quad + \cdots + h[(N-1)/2-1]e^{j\omega} + h[(N-1)/2+1]e^{-j\omega} + h[(N-1)/2]\Big) \\
&= e^{-j\omega(N-1)/2}\Big(2h[0]\cos(\omega(N-1)/2) + \cdots + 2h[(N-1)/2-1]\cos(\omega) \\
&\quad + h[(N-1)/2]\Big) \\
&= e^{-j\omega(N-1)/2}\left(2\sum_{n=0}^{\frac{N-1}{2}-1} h[n]\cos\left(\left(\frac{N-1}{2}-n\right)\omega\right) + h[(N-1)/2]\right)
\end{aligned}
$$

which validates (10.9) with $\angle H(e^{j\omega}) = -\omega(N-1)/2$.

2. Symmetric Impulse Response with Even N

The DTFT of $h[n]$ is:

$$H(e^{j\omega}) = 2\left(\sum_{n=0}^{\frac{N}{2}-1} h[n]\cos\left(\left(\frac{N-1}{2}-n\right)\omega\right)\right)e^{-j\omega(N-1)/2} \quad (10.10)$$

where $\phi = 0$ and $\tau = (N-1)/2$ is not an integer.

3. Anti-symmetric Impulse Response with Odd N

The DTFT of $h[n]$ is:

$$H(e^{j\omega}) = 2\left(\sum_{n=0}^{\frac{N-1}{2}-1} h[n]\sin\left(\left(\frac{N-1}{2}-n\right)\omega\right)\right)e^{j(\pi/2-\omega(N-1)/2)} \quad (10.11)$$

where $\phi = \pi/2$ and $\tau = (N-1)/2$ is an integer

Note that $h[(N-1)/2] = 0$ due to anti-symmetric property

4. Anti-symmetric Impulse Response with Even N

Taking the DTFT of $h[n]$ yields:

$$H(e^{j\omega}) = 2\left(\sum_{n=0}^{\frac{N}{2}-1} h[n]\sin\left(\left(\frac{N-1}{2}-n\right)\omega\right)\right)e^{j(\pi/2-\omega(N-1)/2)} \quad (10.12)$$

where $\phi = \pi/2$ and $\tau = (N-1)/2$ is not an integer.

Example 10.4

Consider an input sequence $x[n]$ of length 400 such that $x[n] = 0$ for $0 \leq n \leq 99$ and $200 \leq n \leq 299$, and it is a sinusoid with frequencies 0.03π and 0.08π for $100 \leq n \leq 199$ and $300 \leq n \leq 399$, respectively.

Examine the filter output $y[n]$ with the following two FIR systems:

(a) $h[n] = \{2, -0.9, -0.72, -0.58, -0.46, -0.37\}$ with $h[0] = 2$

(b) $h[n] = \{0.6, 0.9, -1.2, 0.9, 0.6\}$ with $h[0] = 0.6$

Are they linear phase filters?

The MATLAB program for this example is provided as `ex10_4.m`.

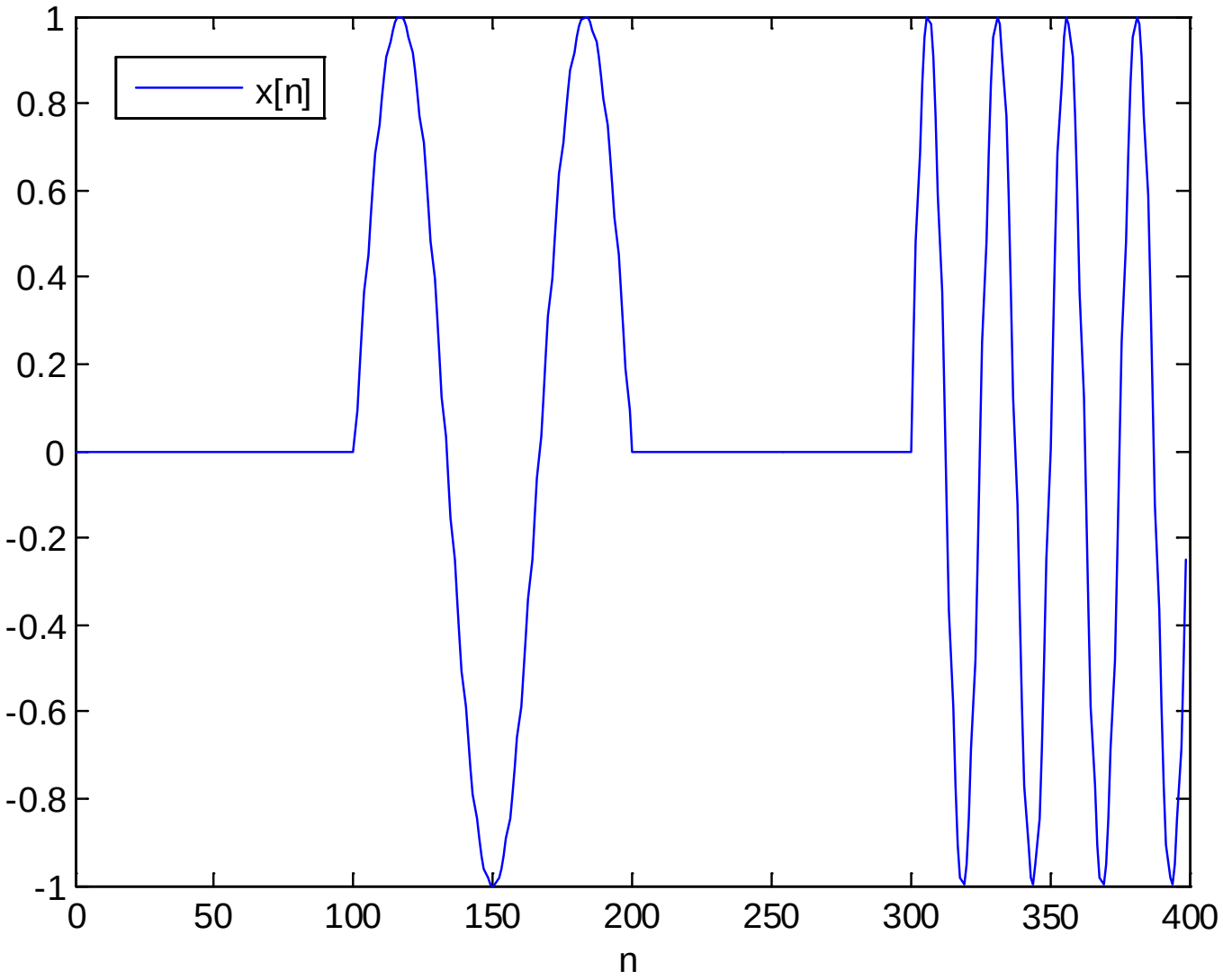


Fig.10.6: Pulsed sinusoid with two frequencies

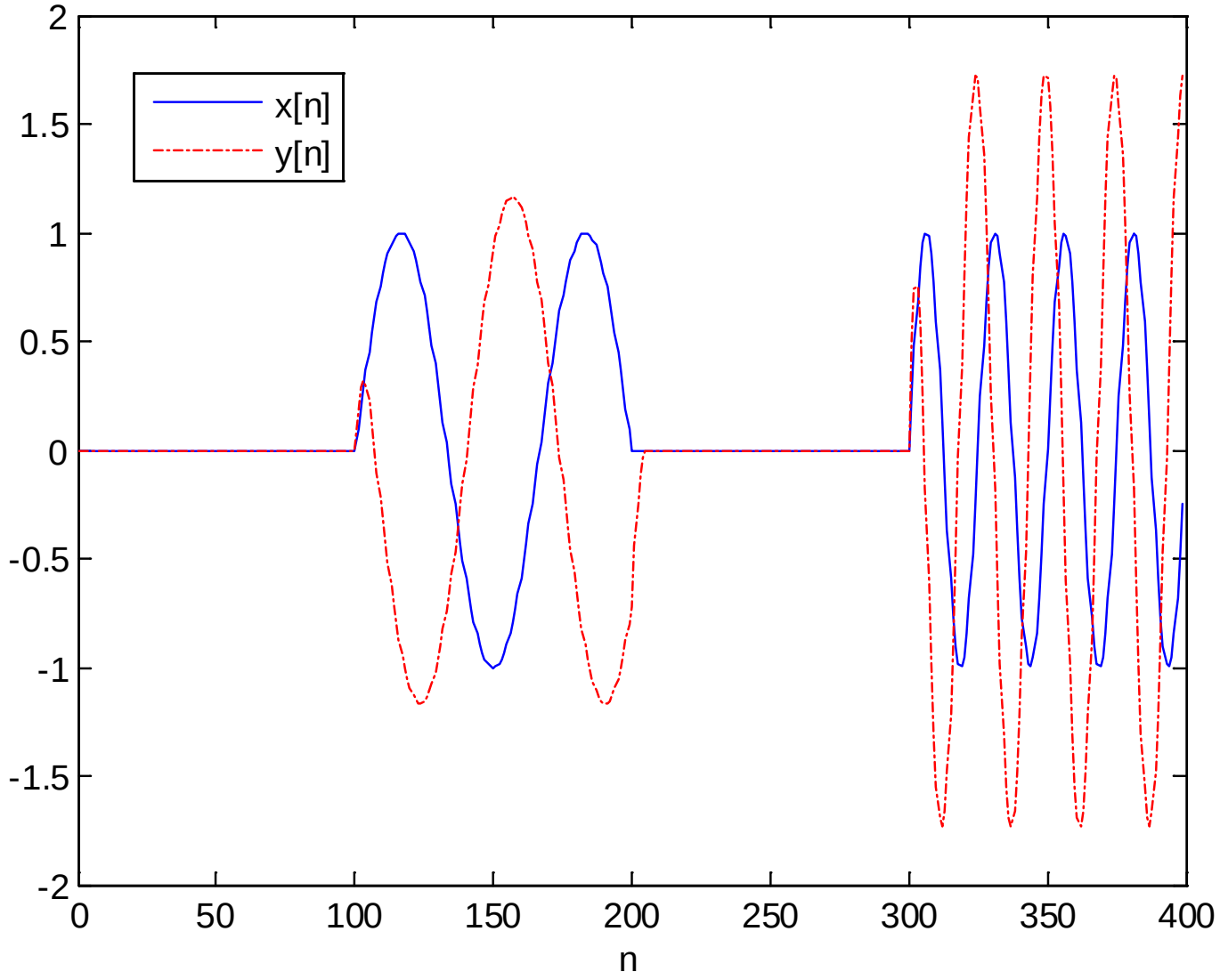


Fig.10.7: Filter output with nonlinear-phase filter

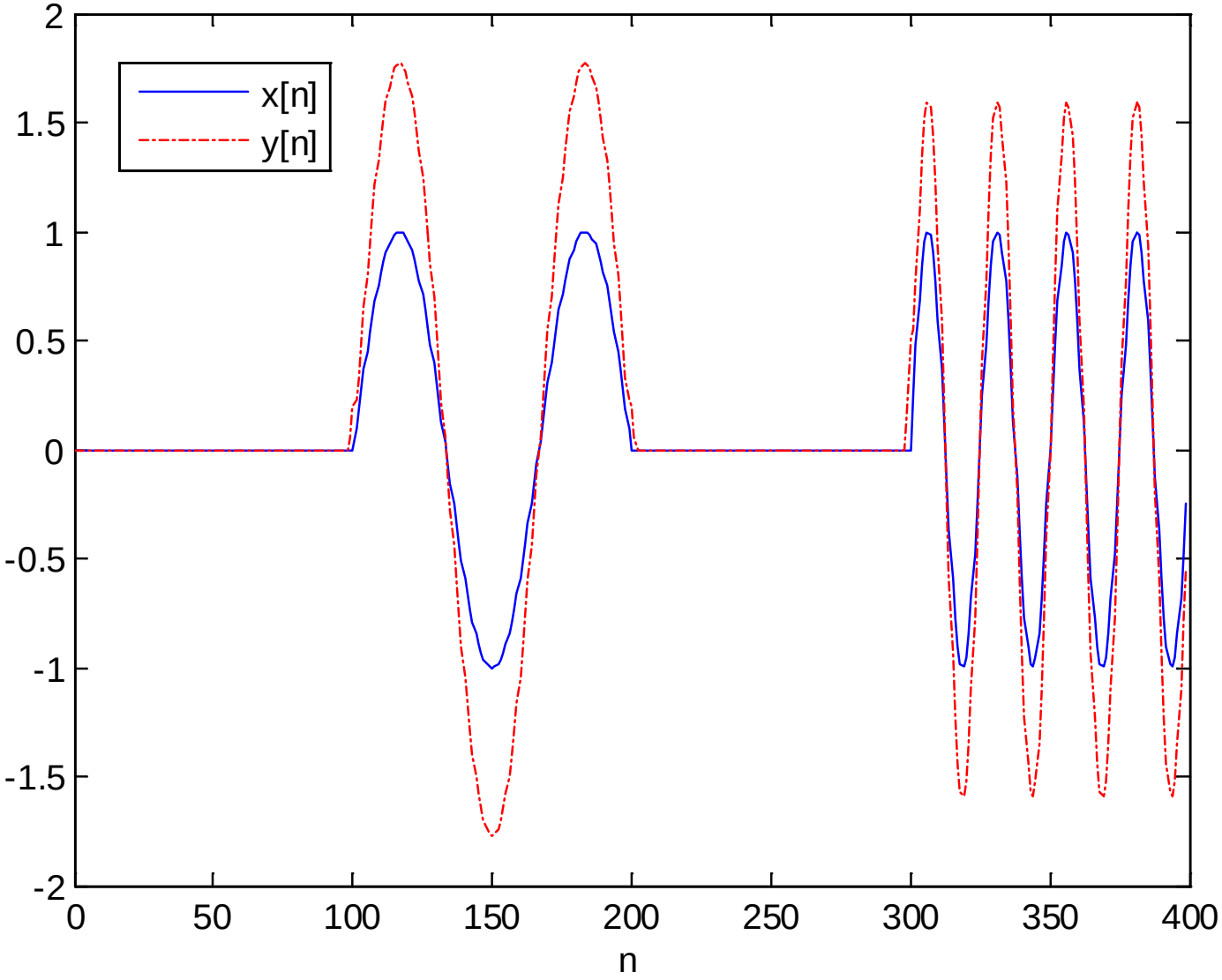


Fig.10.8: Filter output with linear-phase filter

Example 10.5

The impulse response of a causal linear-phase FIR $h[n]$ is:

$$h[n] = \{1/5, -1/4, 1/3, -1/2, 1, 0, -1, 1/2, -1/3, 1/4, -1/5\}$$

where $h[0] = 1/5$. Determine $H(e^{j\omega})$. Plot the amplitude and phase responses and then deduce the function of the filter. Find the expected filter output $y[n]$ when passing an input sequence of the form $x[n] = \sin(0.6n)$ through the filter.

As $N = 11$ is odd and $h[n]$ is anti-symmetric, we apply (10.11):

$$H(e^{j\omega}) = \left(2\sin(\omega) - \sin(2\omega) + \frac{2}{3}\sin(3\omega) - \frac{1}{2}\sin(4\omega) + \frac{2}{5}\sin(5\omega)\right) e^{j(\pi/2 - 5\omega)}$$

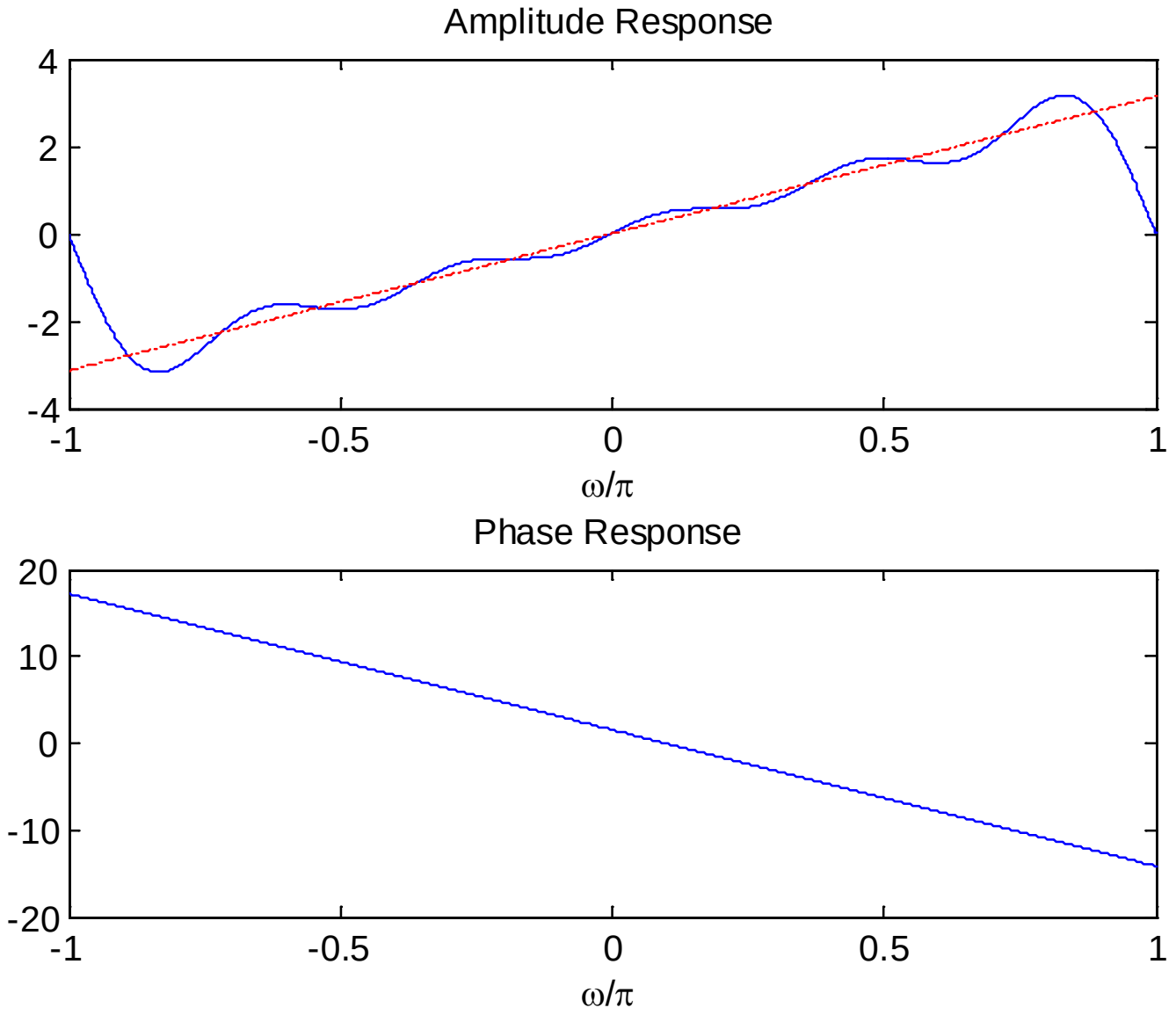


Fig.10.9: Amplitude and phase responses of differentiator

The amplitude response can be approximated as:

$$H_r(e^{j\omega}) = \left(2\sin(\omega) - \sin(2\omega) + \frac{2}{3}\sin(3\omega) - \frac{1}{2}\sin(4\omega) + \frac{2}{5}\sin(5\omega)\right) \approx \omega$$

According to the time-shifting property, the DTFT of $h[n]$ with a time advance of 5 samples is then:

$$\begin{aligned}&\left(2\sin(\omega) - \sin(2\omega) + \frac{2}{3}\sin(3\omega) - \frac{1}{2}\sin(4\omega) + \frac{2}{5}\sin(5\omega)\right) e^{j\pi/2} \\ &\quad \approx \omega e^{j\pi/2} = \omega\left[\cos\left(\frac{\pi}{2}\right) + j\sin\left(\frac{\pi}{2}\right)\right] = j\omega\end{aligned}$$

This means that the causal FIR filter has a frequency response approximately equals $j\omega$ with a delay of 5 samples.

The system with frequency response $j\omega$ is known as the differentiator. Recall:

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}d\omega$$

Differentiating both sides with respect to n yields:

$$\begin{aligned}\frac{dx[n]}{dn} &= \frac{d}{dn}\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}d\omega\right)\\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})\frac{de^{j\omega n}}{dn}d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[X(e^{j\omega})\cdot j\omega\right]e^{j\omega n}d\omega\end{aligned}$$

which confirms:

$$\frac{dx[n]}{dn} \leftrightarrow X(e^{j\omega}) \cdot j\omega$$

As a result, we expect the output $y[n]$ is close to:

$$y[n] = \frac{dx[n]}{dn} = 0.6\cos(0.6n)$$

with a delay of 5 samples

The MATLAB program is provided as `ex10_5.m`.

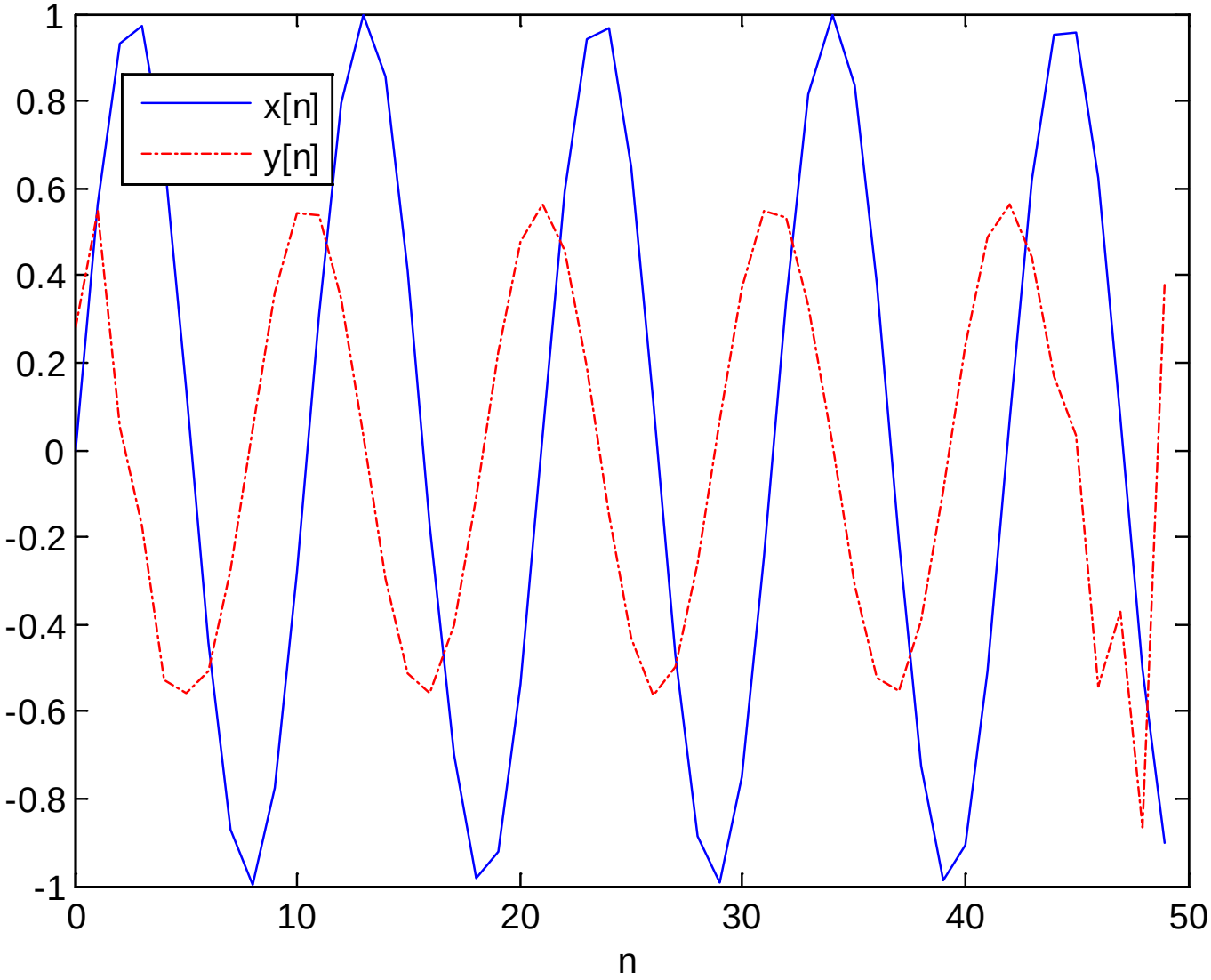


Fig.10.10: Differentiator output with sinusoidal input

Window Method

Based on directly approximating the desired frequency response $H_d(e^{j\omega})$

Ideal impulse response $h_d[n]$ is calculated from inverse DTFT:

$$h_d[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega, \quad n = \cdots, -1, 0, 1, 2, \cdots \qquad (10.13)$$

What are the problems in using (10.13)?

Example 10.6

Determine the impulse response $h_d[n]$ of an ideal lowpass filter with a cutoff frequency of ω_c. The DTFT of the filter is:

$$H_d(e^{j\omega}) = \begin{cases} 1, & |\omega| < \omega_c \\ 0, & -\pi < \omega < -\omega_c, \quad \omega_c < \omega < \pi \end{cases}$$

Using Example 6.3, $h_d[n]$ is:

$$h_d[n] = \frac{\sin(\omega_c n)}{\pi n} = \frac{\omega_c}{\pi}\text{sinc}\left(\frac{\omega_c n}{\pi}\right)$$

where

$$\text{sinc}(u) = \frac{\sin(\pi u)}{\pi u}$$

Ideal filter is noncausal and its impulse response is of infinite length

The basic idea of window method is to truncate $h_d[n]$ to obtain a linear-phase and causal FIR filter via 2 steps:

- Windowing
  - Extract a finite set of $h_d[n]$ with positive and negative time indexes because the impulse response is generally symmetric or anti-symmetric around $n = 0$, say, $\{h_d[-M], h_d[-M+1], \cdots, h_d[M]\}$ corresponding to a length of $(2M+1)$
  - Filter response is linear phase but is only an approximation of $H_d(e^{j\omega})$ due to the coefficient truncation
- Time Shifting
  - Delay $\{h_d[-M], h_d[-M+1], \cdots, h_d[M]\}$ by M samples to obtain $h[n]$ with $h[0] = h_d[-M]$ to achieve causality
  - Filter output is received after a delay of M samples

Example 10.7

Use the window method to design a linear-phase and causal FIR system with 7 coefficients to approximate an ideal lowpass filter whose cutoff frequency is $\omega_c = 0.1\pi$.

From Example 10.6:

$$h_d[n] = \frac{\sin(0.1\pi n)}{\pi n} = 0.1\text{sinc}\,(0.1n)$$

Via windowing, we extract the set of $\{h_d[-3], h_d[-2], \cdots, h_d[3]\}$. Notice that $h_d[n]$ is symmetric around at $n = 0$ as $h_d[n] = h_d[-n]$.

Their values are calculated as:

$$h_d[3] = h_d[-3] = \frac{\sin(0.1\pi \cdot 3)}{\pi \cdot 3} = 0.08584$$

$$h_d[2] = h_d[-2] = \frac{\sin(0.1\pi \cdot 2)}{\pi \cdot 2} = 0.09355$$

$$h_d[1] = h_d[-1] = \frac{\sin(0.1\pi)}{\pi} = 0.09836$$

$$h_d[0] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega\cdot 0}d\omega = \frac{1}{2\pi}\int_{-0.1\pi}^{0.1\pi} d\omega = 0.1$$

Via time-shifting of 3 samples, a causal filter is obtained as:

$$h[n] = h_d[n-3], \quad n = 0, 1, \cdots, 6:$$

or

$$h[n] = \{0.08584, 0.09355, 0.9836, 0.1, 0.9836, 0.09355, 0.08584\}$$

with $h[0] = 0.08584$.

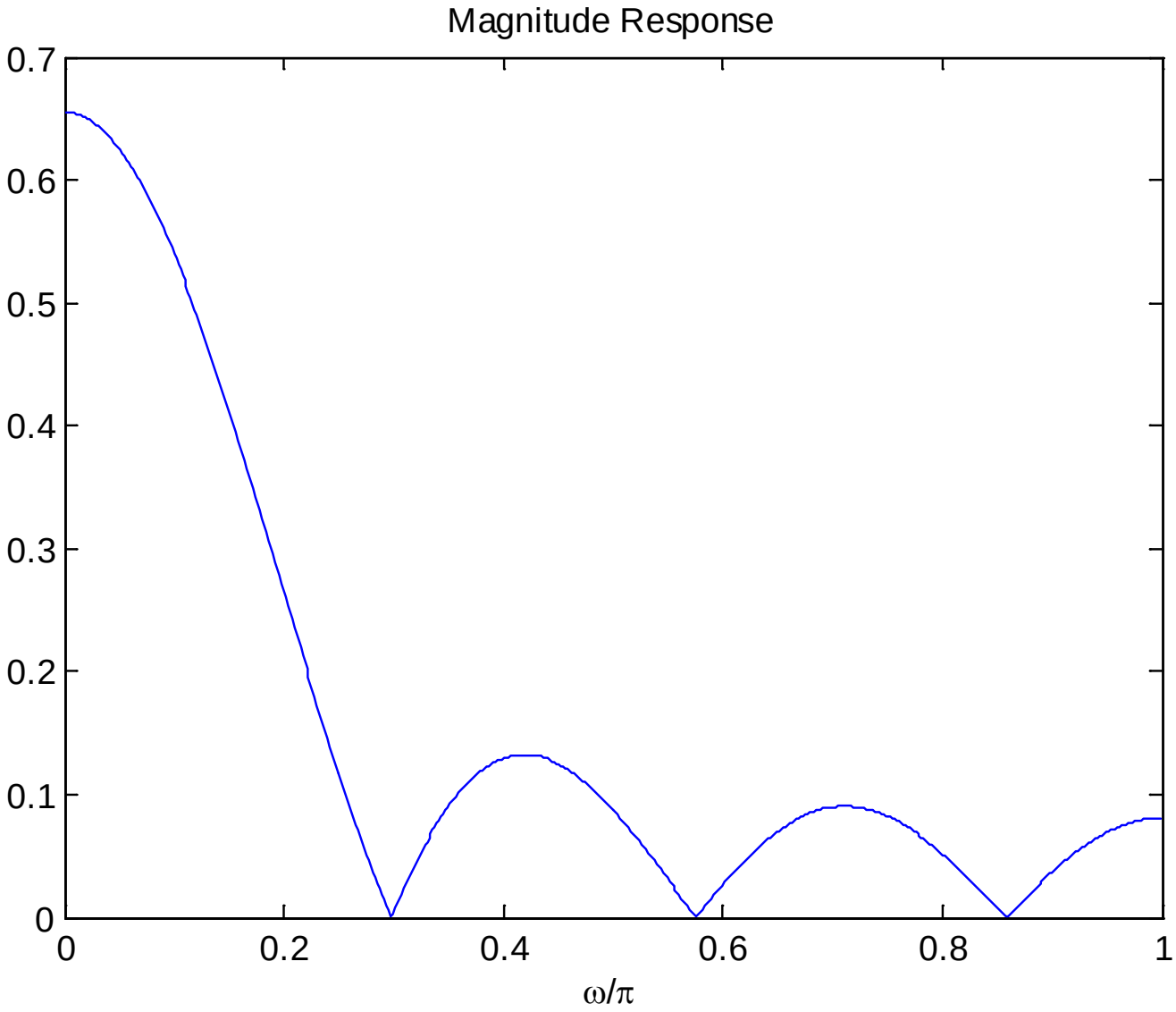


Fig.10.11: Magnitude response of lowpass filter with $N = 7$

Note that we can use the MATLAB command `fir1(6,0.1,boxcar(7),'noscale')` to produce $h[n]$

Alternatively, we can first perform the time shifting prior to windowing

As there should be a phase of $-\tau\omega$ where $\tau = (N-1)/2$ in the practical filter, we modify the desired frequency response as:

$$H_d(e^{j\omega}) = \begin{cases} 1 \cdot e^{-j\omega\tau}, & |\omega| < \omega_c \\ 0, & -\pi < \omega < -\omega_c, \quad \omega_c < \omega < \pi \end{cases}$$

Note that multiplying $e^{-j\omega\tau}$ in the frequency domain corresponds to a time shift of τ in the time domain.

The corresponding impulse response is

$$h_d[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{-j\omega\tau}e^{j\omega n}d\omega = \frac{\sin(\omega_c(n-\tau))}{\pi(n-\tau)}$$

As the filter length is N, we then perform windowing on $h_d[n]$:

$$h[n] = \begin{cases} h_d[n], & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

Substituting $N = 7$ and $\omega_c = 0.1\pi$ yields the same FIR impulse response.

Note that we can base on this approach to determine $h[n]$ even when N is an even integer

The truncation operation can be considered as multiplying $h_d[n]$ by a rectangular window function:

$$w[n] = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases}$$

That is,

$$h[n] = h_d[n] \cdot w[n]$$

Generally speaking, $w[n]$ is not restricted to be rectangular and it can be any symmetric function so that the resultant filter is linear phase

Example 10.8

Use the window method to design a linear-phase and causal FIR filter of length 101 such that the sampled version of a continuous-time sinusoid with frequency of 80 Hz can pass through it with negligible attenuation while the sampled signal corresponds to a tone of frequency 120 Hz will be suppressed. The sampling frequency is 1000 Hz.

Let the continuous-time sinusoids be

$$s_i(t) = \cos(\Omega_i t), \quad \Omega_1 = 80 \cdot 2\pi,\ \Omega_2 = 120 \cdot 2\pi$$

From (4.1), the corresponding discrete-time versions with sampling interval of $T = 1/1000$ s are

$$s_i[n] = s_i(nT) = \cos(\Omega_i n/1000)$$

which gives:

$$s_1[n] = \cos(0.16\pi n)$$

and

$$s_2[n] = \cos(0.24\pi n)$$

The frequencies are 0.16π and 0.24π in discrete-time domain.

In order to suppress $s_2[n]$ while keeping $s_1[n]$, we can use a lowpass filter with cutoff frequency of $\omega_c = 0.2\pi$, which is simply the average of the two discrete frequencies.

Using Example 10.7 with $N = 101$ and $\omega_c = 0.2\pi$, the required filter impulse response $h[n]$ is:

$$h[n] = \frac{\sin(0.2\pi(n-50))}{\pi(n-50)} = 0.2\text{sinc}(0.2(n-50)), \quad 0 \le n \le 100$$

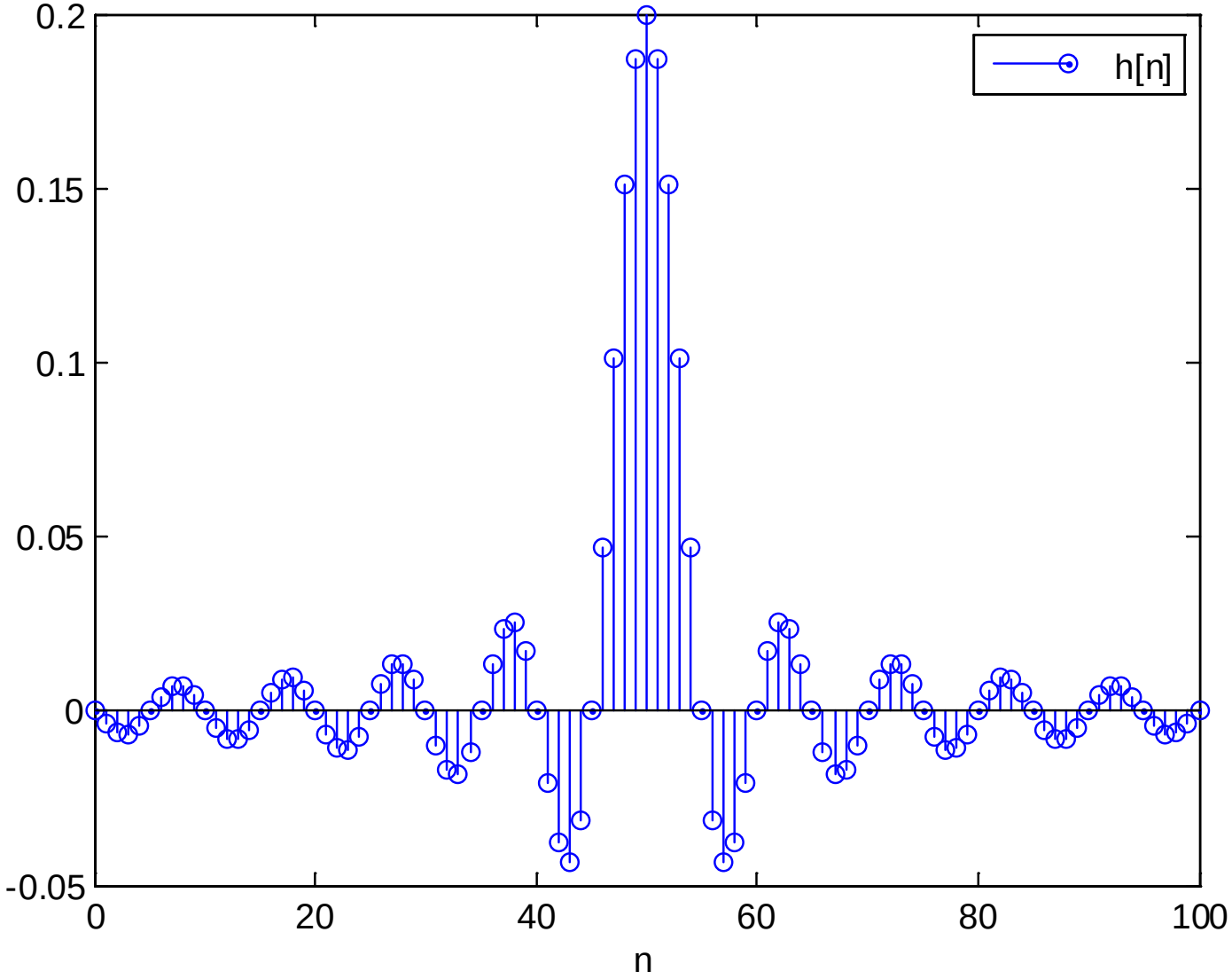


Fig.10.12: Impulse response of lowpass filter with $N = 101$

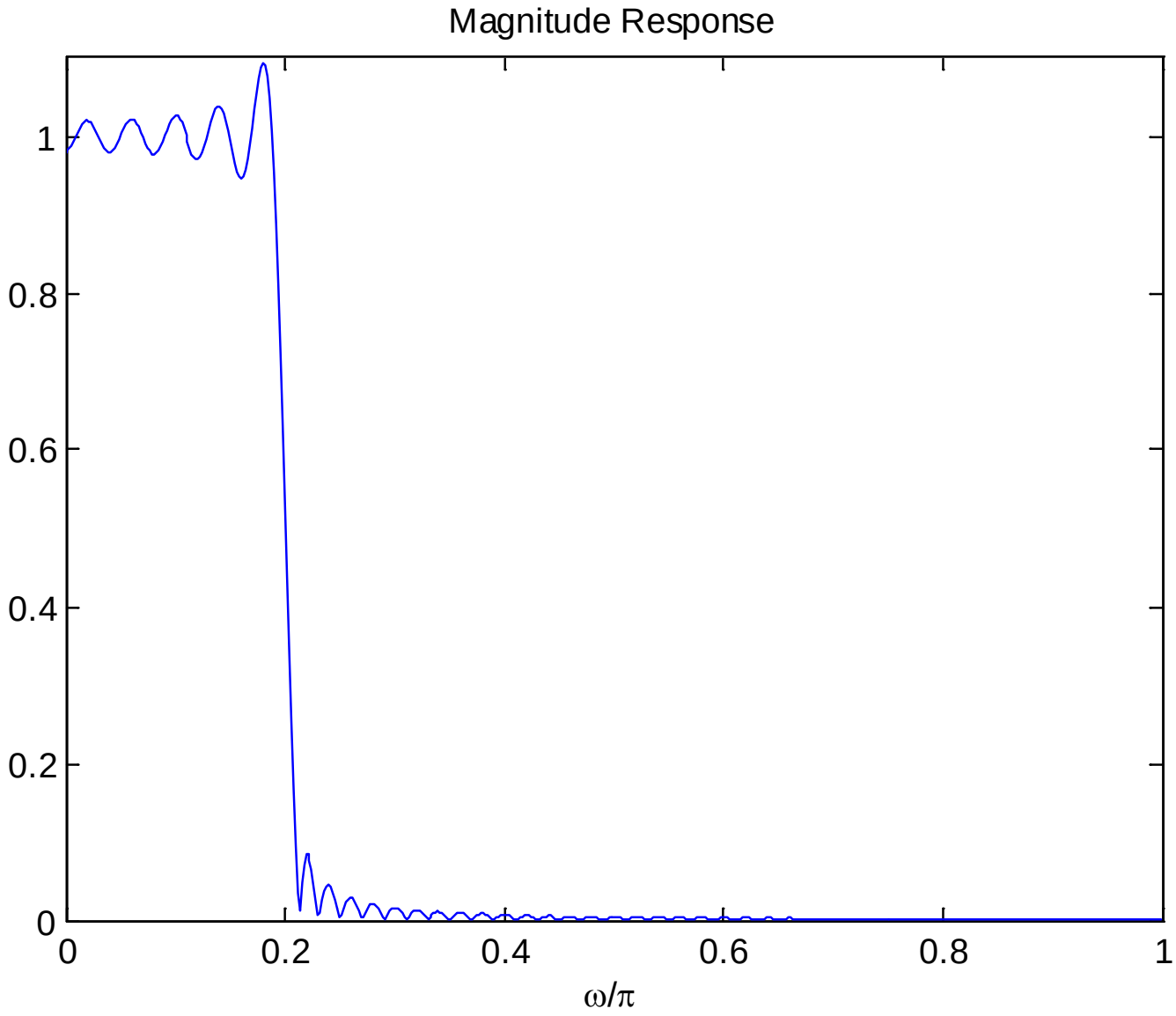


Fig.10.13: Magnitude response of lowpass filter with $N = 101$

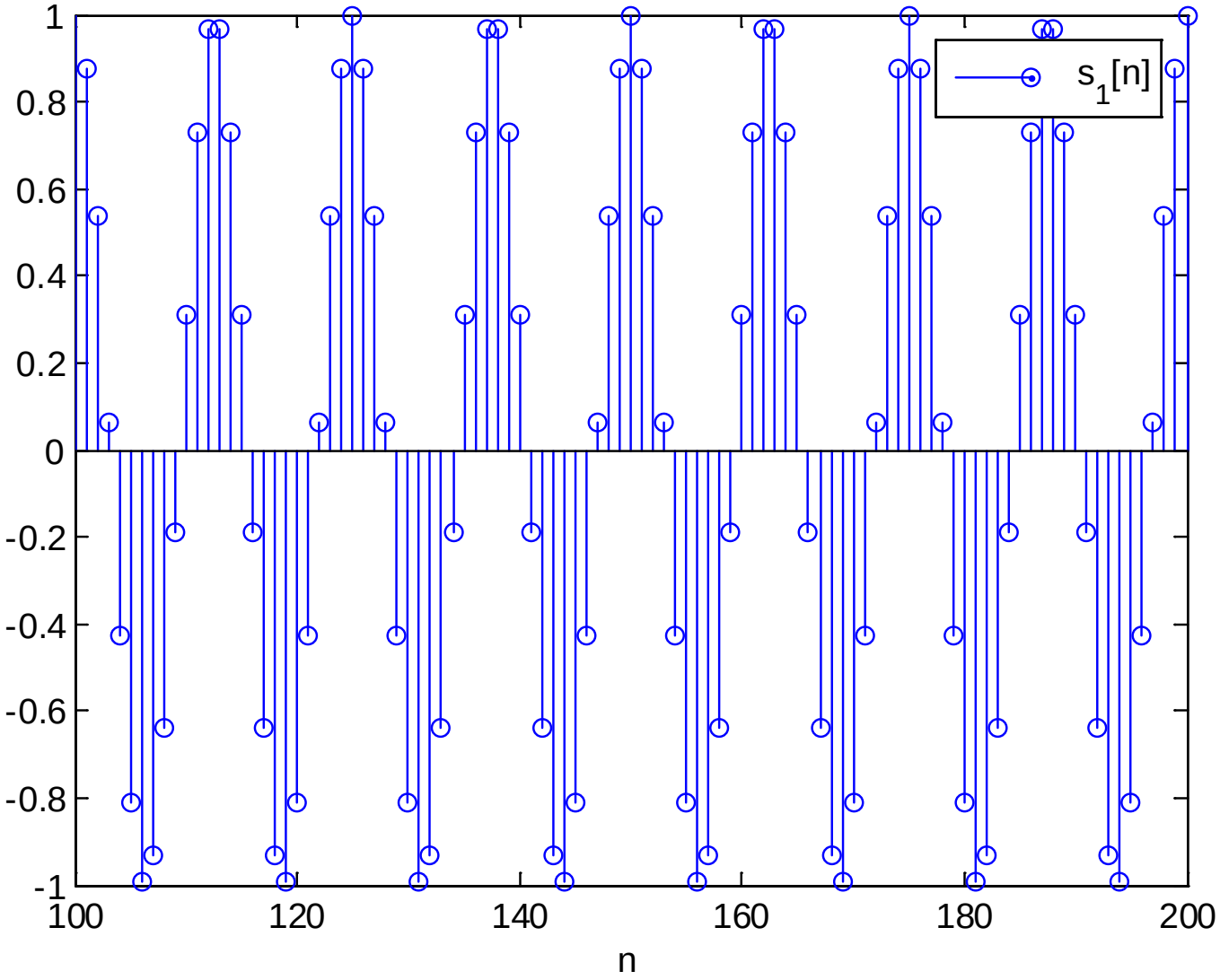


Fig.10.14: Discrete-time tone with frequency 0.16π

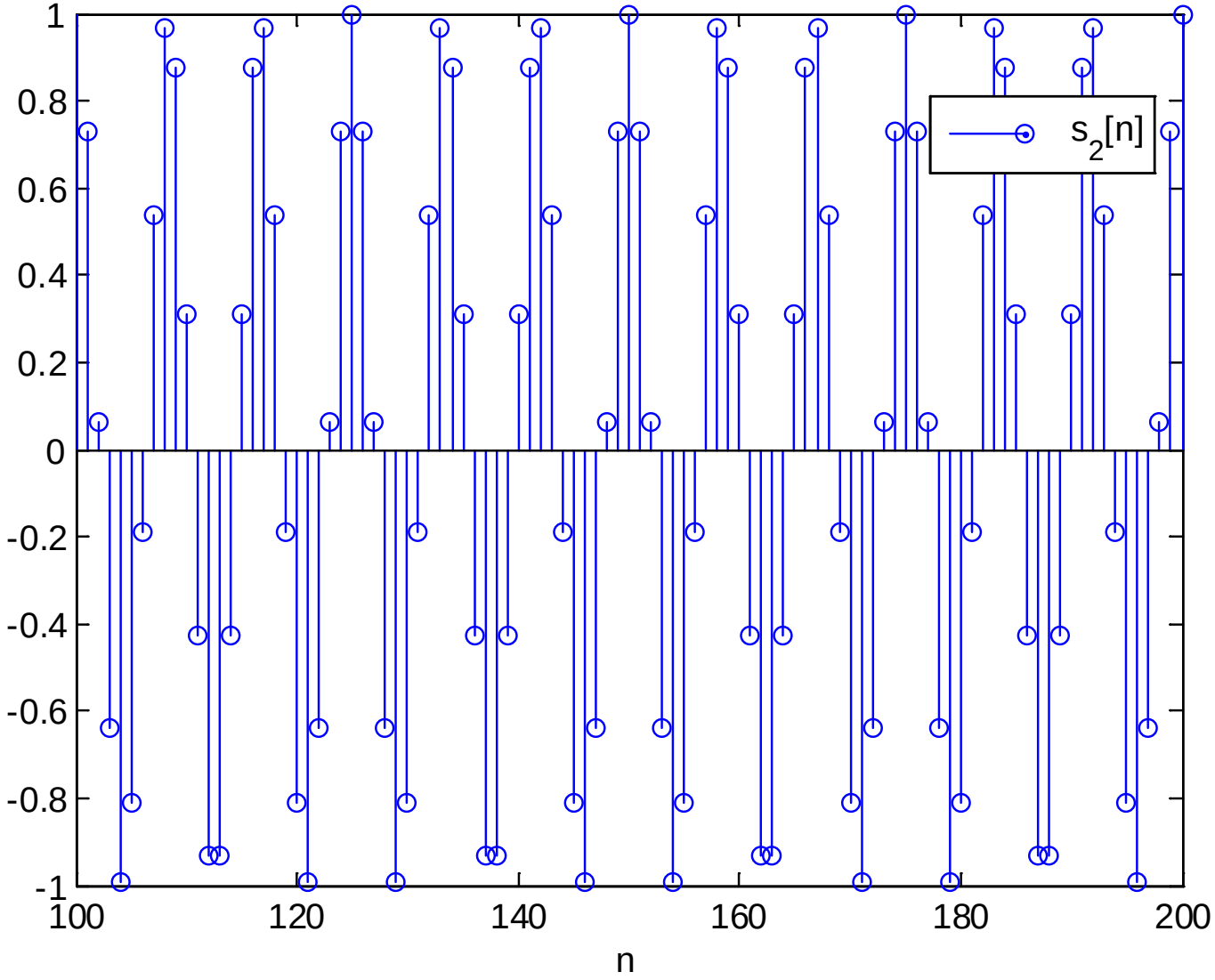


Fig.10.15: Discrete-time tone with frequency 0.24π

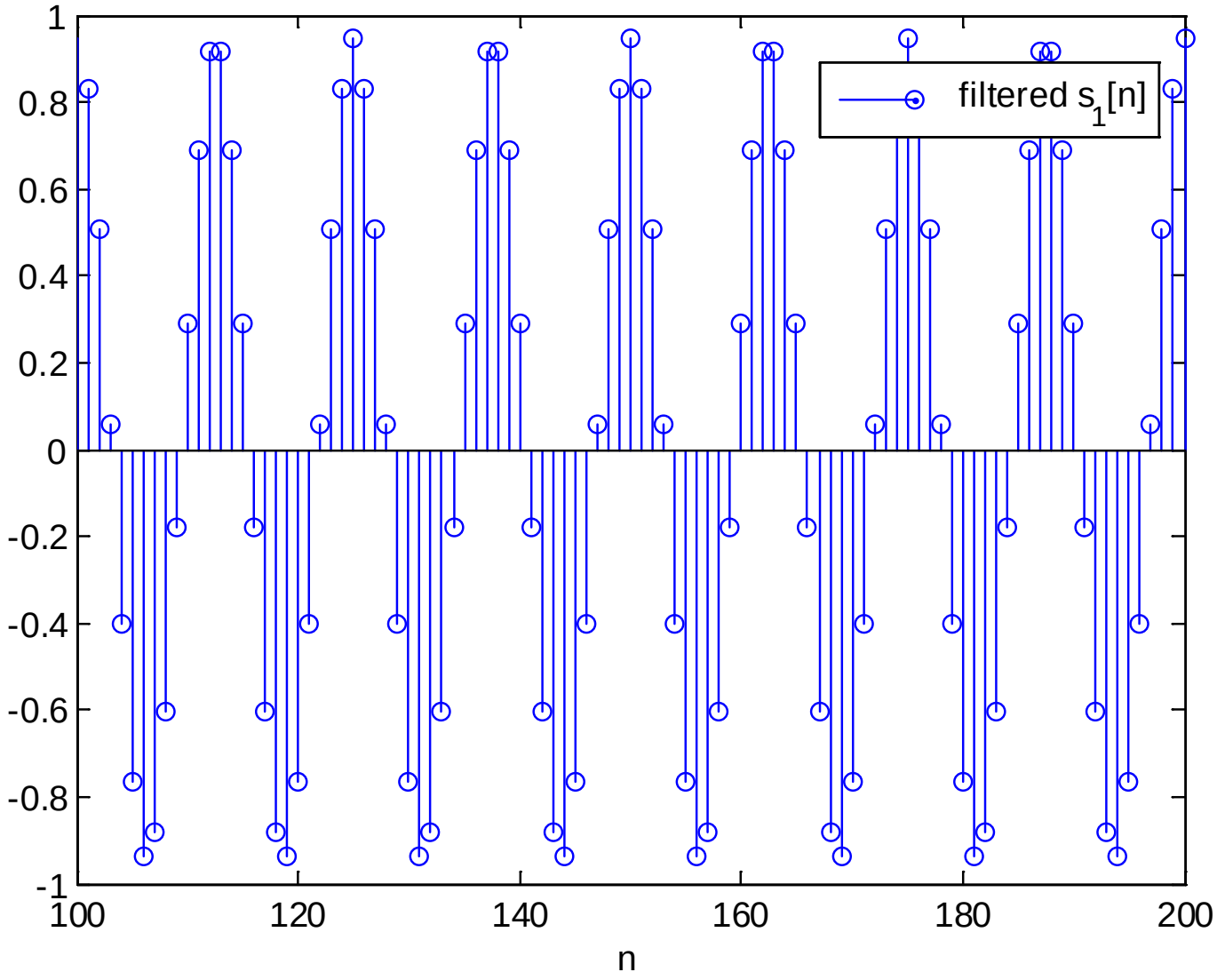


Fig.10.16: Filter output for frequency 0.16π

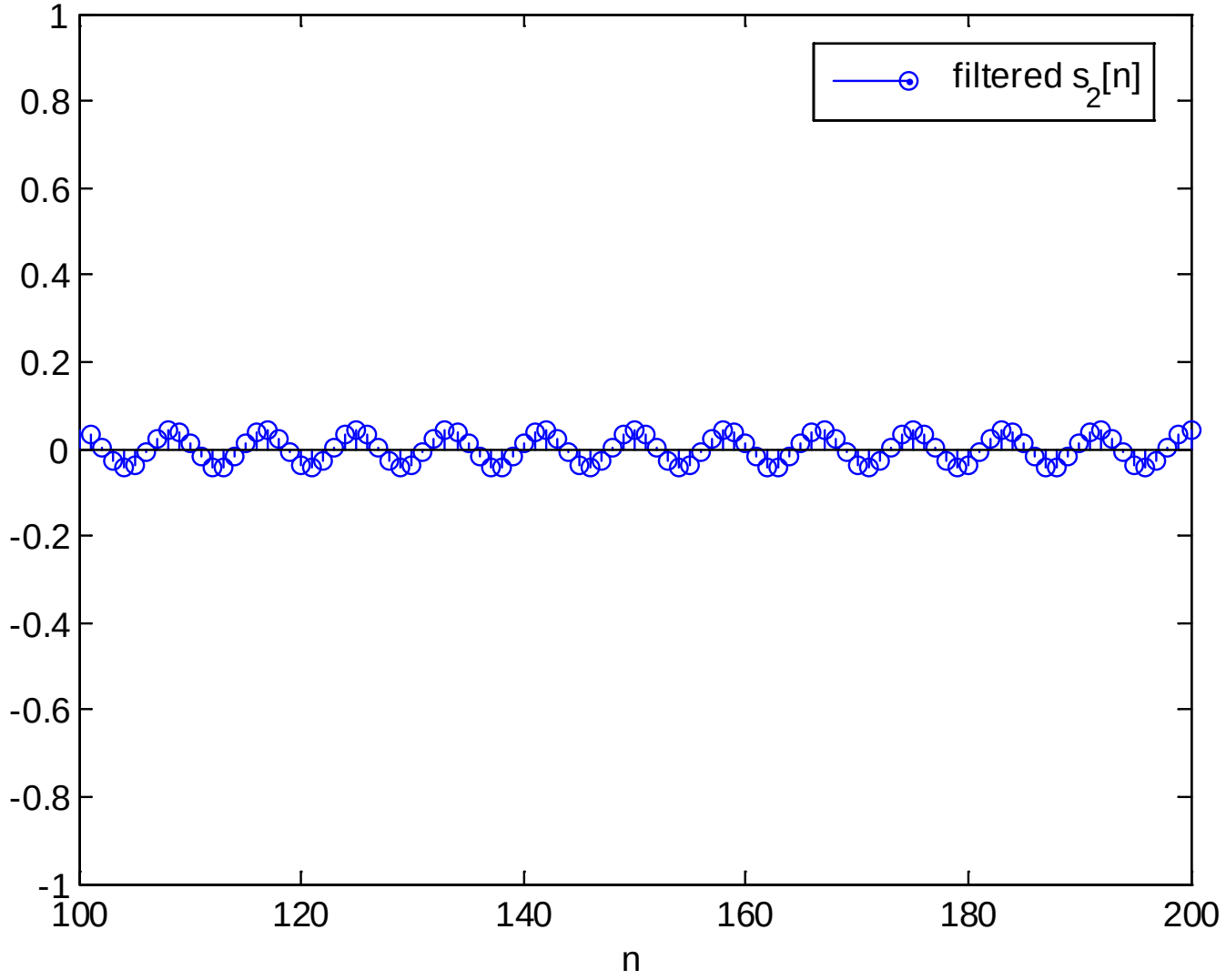


Fig.10.17: Filter output for frequency 0.24π

Analysis of Windowing

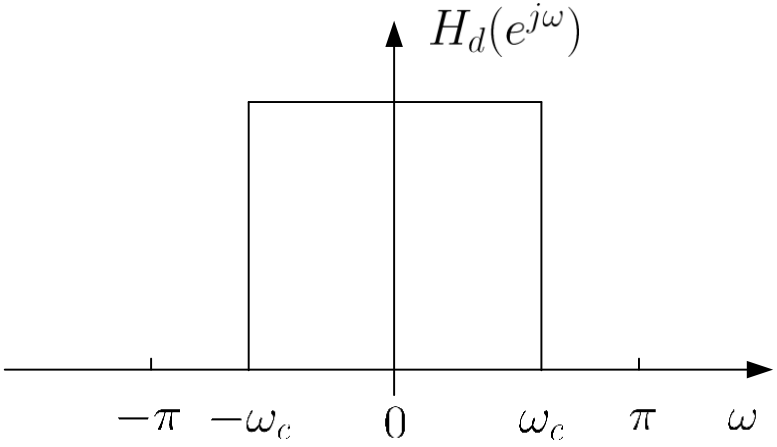
To incorporate passband and stopband frequencies and ripples in the FIR filter design, we need to study the windowing effect

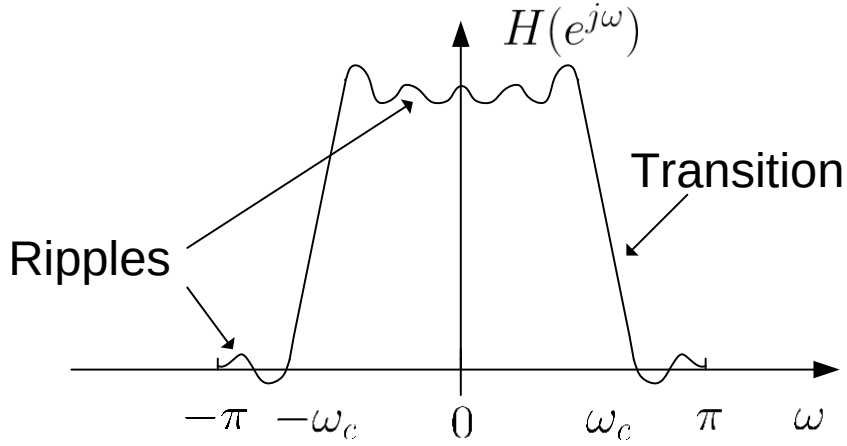
Recall

$$h[n] = h_d[n] \cdot w[n] \tag{10.14}$$

According to the multiplication property of (6.18):

$$H(e^{j\omega}) = H_d(e^{j\omega}) \tilde{\otimes} W(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\tau}) W(e^{j(\omega-\tau)}) d\tau \tag{10.15}$$





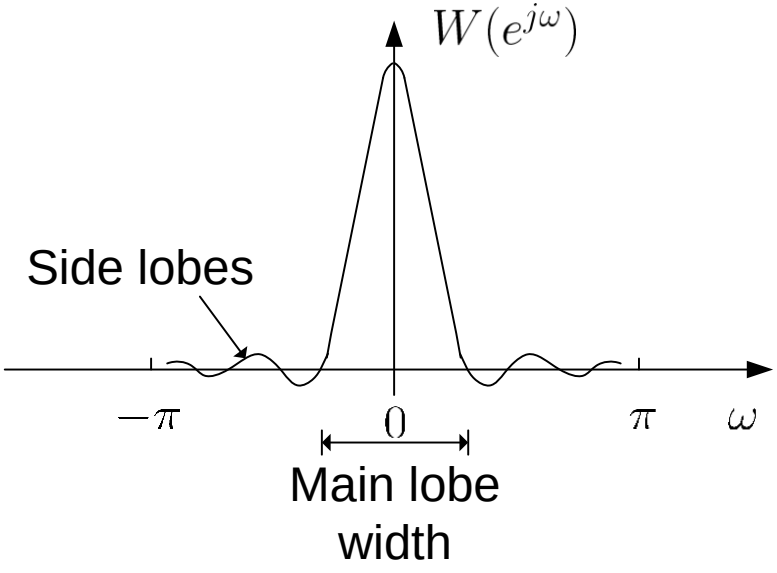


Fig.10.18: Illustration of $H(e^{j\omega}) = H_d(e^{j\omega}) \tilde{\otimes} W(e^{j\omega})$

$H(e^{j\omega})$ is a smeared version of ideal $H_d(e^{j\omega})$

$W(e^{j\omega})$ has a peaky main lobe and several side lobes of smaller magnitudes

Main lobe produces the transition band in $H(e^{j\omega})$. That is, the transition width is proportional to the main lobe width, which is inversely proportional to the filter length N

Side lobes are responsible to produce the ripples in the passband and stopband with $\delta_1 = \delta_2$

$H(e^{j\omega})$ approaches $H_d(e^{j\omega})$ when $W(e^{j\omega})$ has the smallest main lobe width and side lobe magnitude. That is, the ideal but not practical form of $W(e^{j\omega}) = \delta(\omega)$ gives $H(e^{j\omega}) = H_d(e^{j\omega})$

- Rectangular Window

$$w[n] = u[n] - u[n-N] \tag{10.16}$$

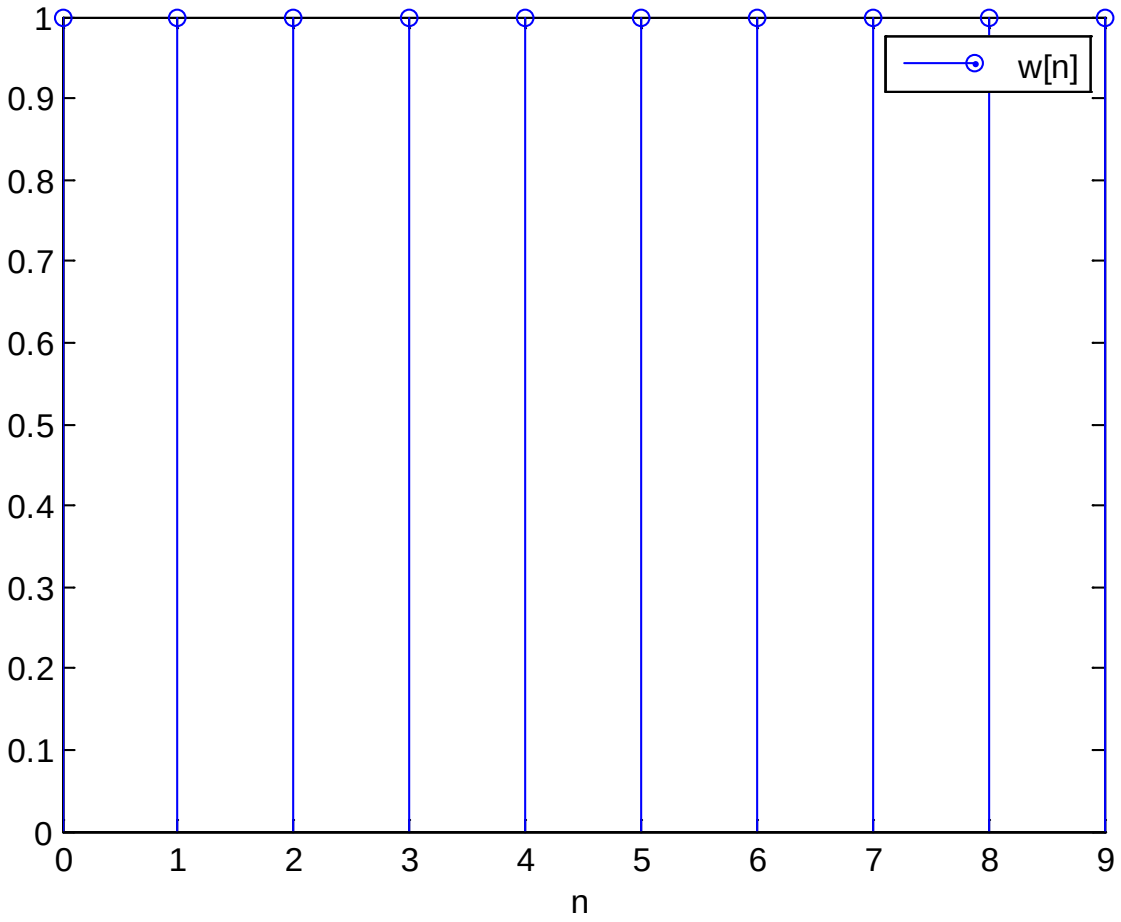


Fig.10.19: Rectangular window function for $N = 10$

From Example 10.1:

$$W(e^{j\omega}) = \frac{\sin(\omega N/2)}{\sin(\omega/2)} \cdot e^{-j\omega(N-1)/2} \qquad (10.17)$$

The main lobe width is

$$\Delta\omega_m = \frac{4\pi}{N} \qquad (10.18)$$

which is proportional to the filter transition width

That is, transition width decreases as the filter length N increases.

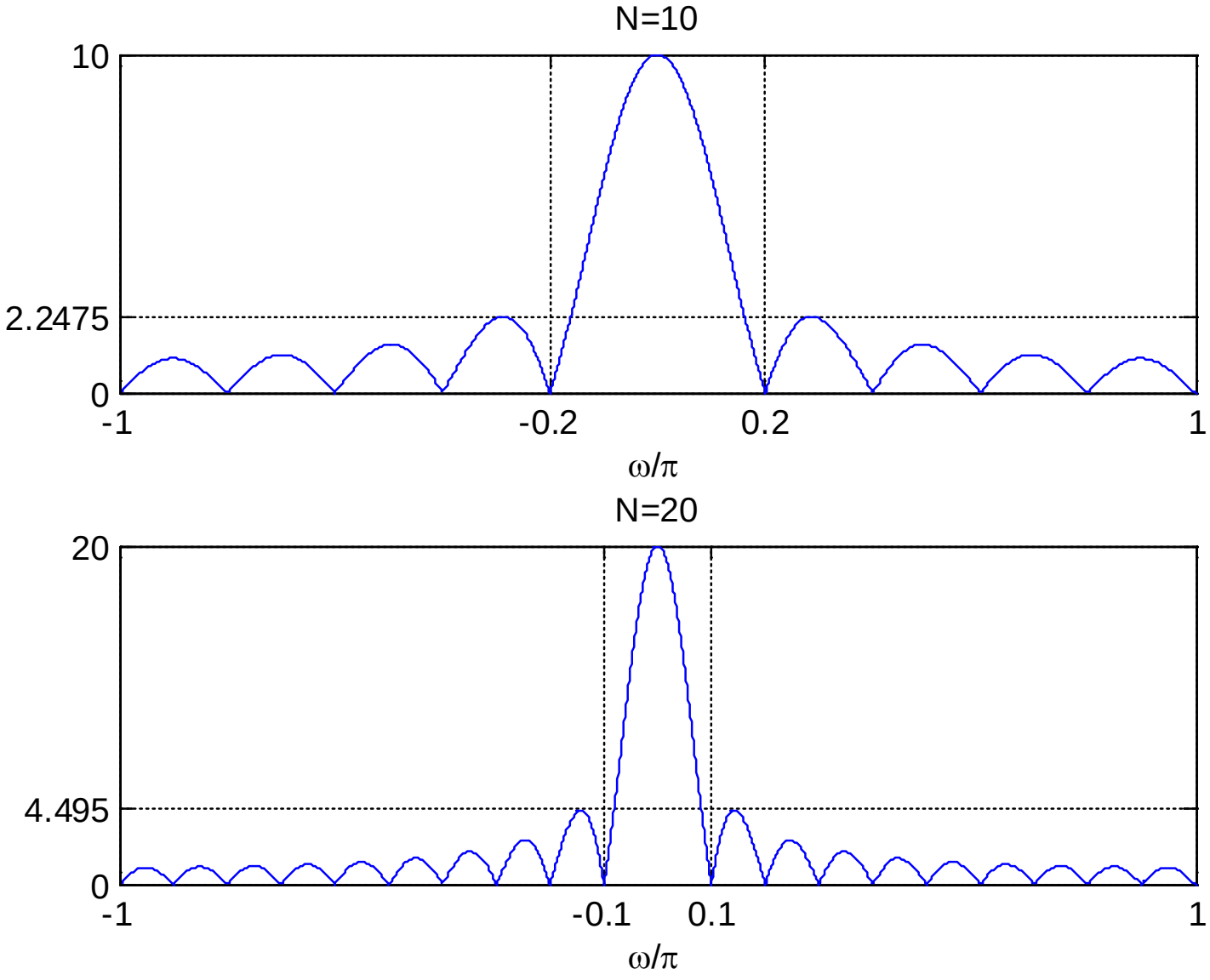


Fig.10.20: Magnitude responses of rectangular window for $N = 10$ & $N = 20$

Relative peak side lobe, r_p, is the ratio of the peak side lobe height to main lobe height in dB:

$$r_p = 20\log_{10}\left(\frac{2.2475}{10}\right) = 20\log_{10}\left(\frac{4.495}{20}\right) = -13 \text{ dB} \qquad (10.19)$$

A larger r_p implies ripples with bigger magnitudes

As N increases, widths of the main lobe and all side lobes decrease but their areas remain unchanged, meaning that the ripples cannot be reduced.

This is known as Gibbs phenomenon due to sudden change in transition from 0 to 1 and 1 to 0. This can be removed by tapering the window smoothly to zero at each end. However, the heights of the side lobes can be diminished but at the expense of wider main lobe and hence a wider transition.

- Bartlett Window

$$w[n] = \begin{cases} 1 - \dfrac{2\left|n - \dfrac{N-1}{2}\right|}{N-1}, & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases} \qquad (10.20)$$

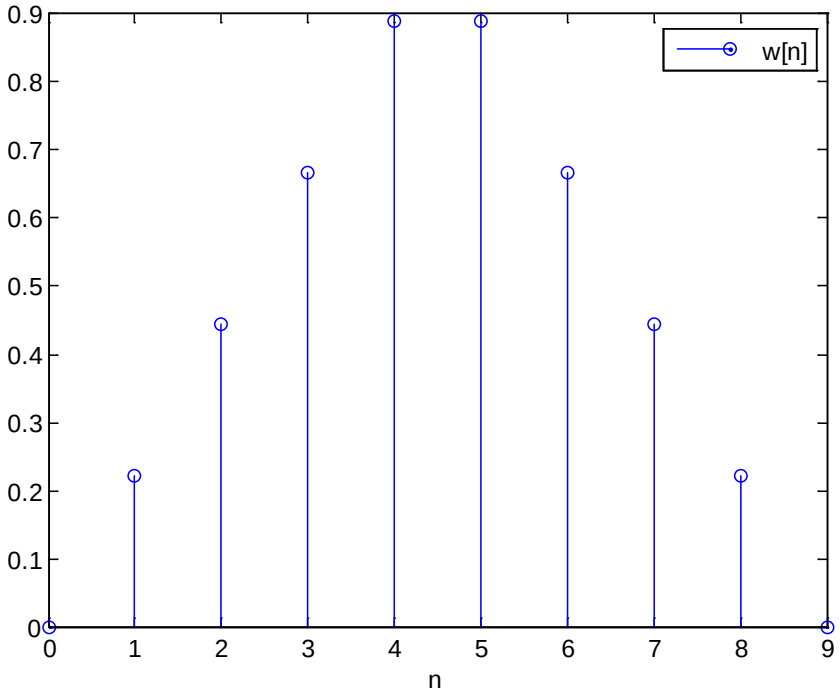


Fig.10.21: Bartlett window function for $N = 10$

- Hanning Window

$$w[n] = \begin{cases} \frac{1}{2}\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right), & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases} \qquad (10.21)$$

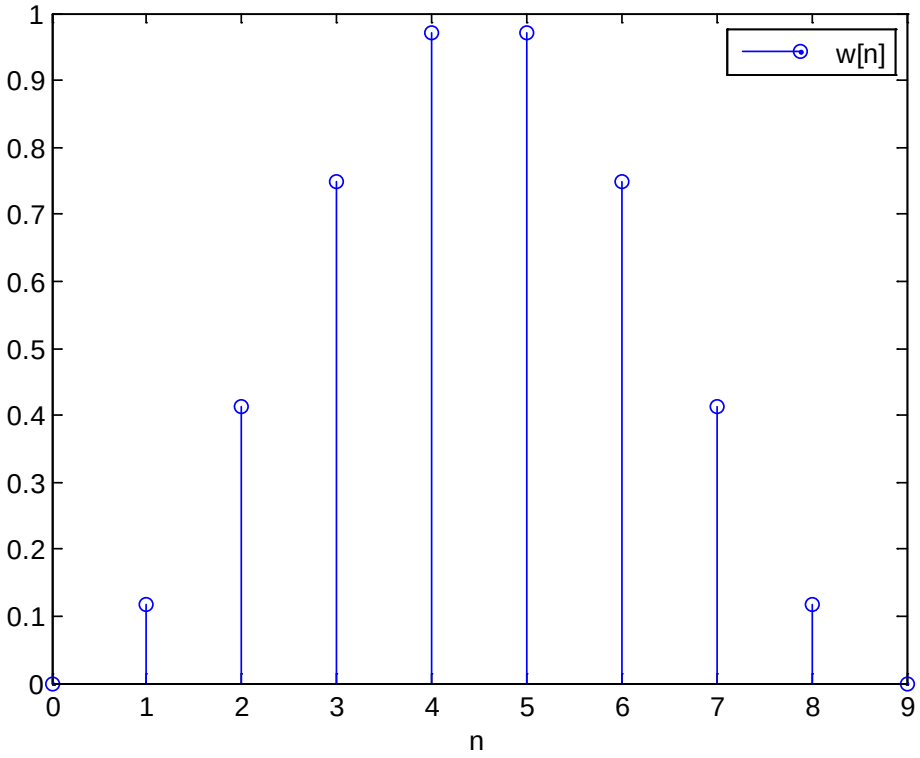


Fig.10.22: Hanning window function for $N = 10$

- Hamming Window

$$w[n] = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{N-1}\right), & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases} \qquad (10.22)$$

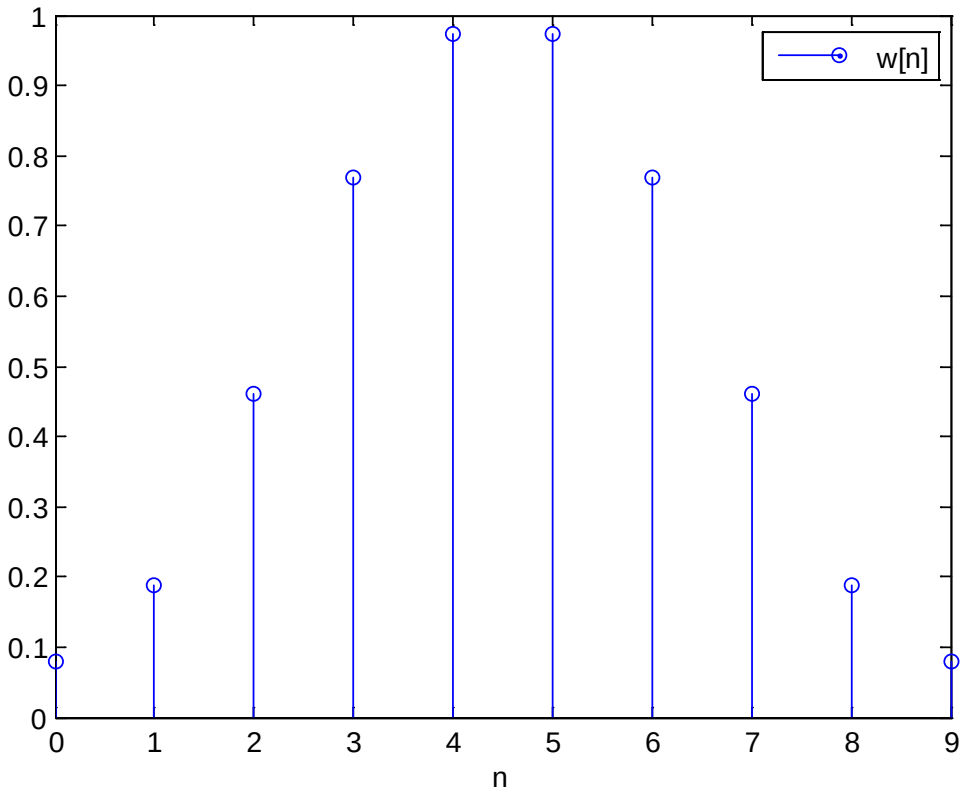


Fig.10.23: Hamming window function for $N = 10$

- Blackman Window

$$w[n] = \begin{cases} 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right), & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases} \quad (10.23)$$

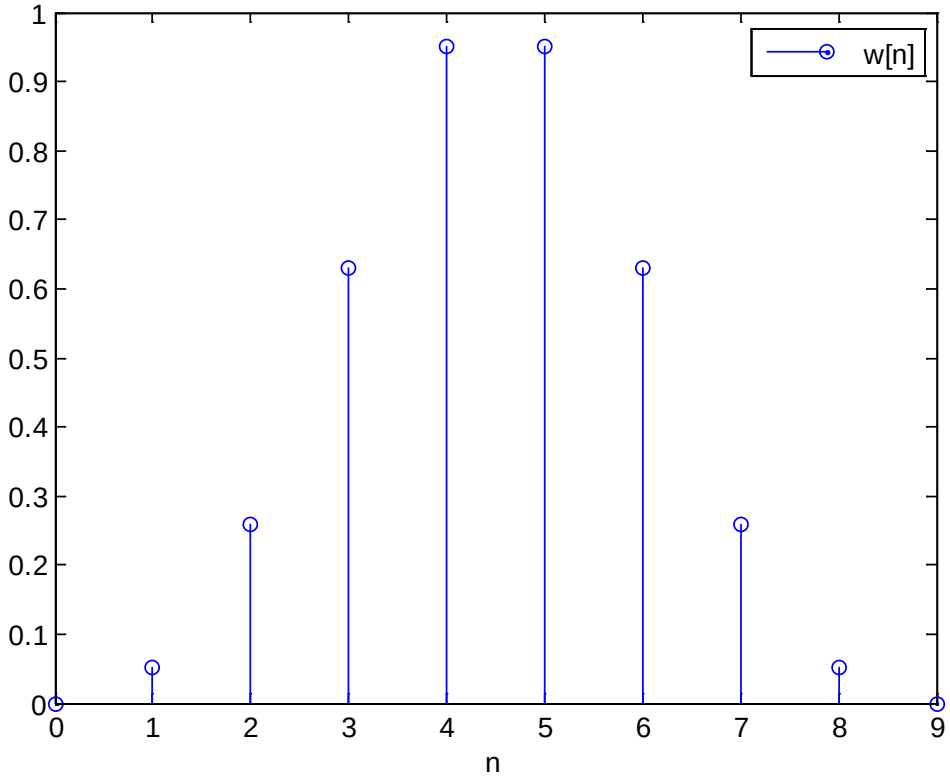


Fig.10.24: Blackman window function for $N = 10$

Note that by using $(N+2)$ terms for a filter length of N and then discarding the end points, we can avoid zero coefficients at both ends for Bartlett, Hanning and Blackman windows.

Window	$\Delta\omega_m$	r_p (dB)
Rectangular	$4\pi/N$	-13
Bartlett	$8\pi/N$	-27
Hanning	$8\pi/N$	-32
Hamming	$8\pi/N$	-43
Blackman	$12\pi/N$	-58

Table 10.1: Characteristics of window functions

As a result, there is a tradeoff between transition and ripple

Example 10.9

Use the window method to design a linear-phase and causal FIR system to approximate an ideal lowpass filter whose cutoff frequency is $\omega_c = 0.5\pi$. Try the rectangular and Bartlett window functions with filter lengths of 7, 21, 61 and 101, to investigate the frequency magnitude responses.

Using Example 10.7, $h[n]$ with the rectangular window is:

$$h[n] = \frac{\sin(\omega_c(n-\tau))}{\pi(n-\tau)}, \quad 0 \leq n \leq N-1$$

with N equals 7, 21, 61 and 101 and $\tau = (N-1)/2$.

Multiplying $h[n]$ by (10.20) yields the corresponding filter coefficients for the Bartlett window.

The MATLAB program is provided as `ex10_9.m`.

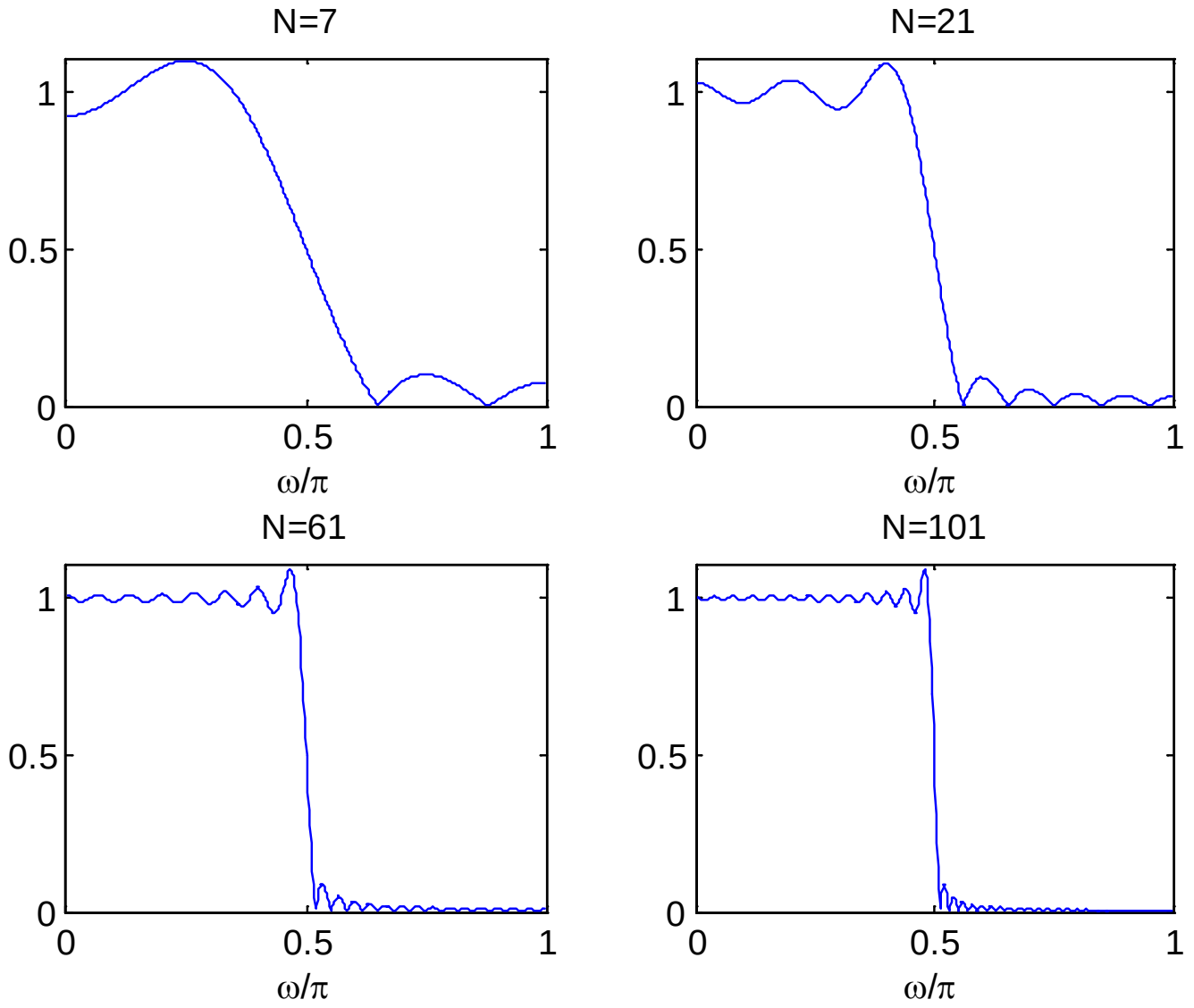


Fig.10.25: Magnitude responses with rectangular window at different N

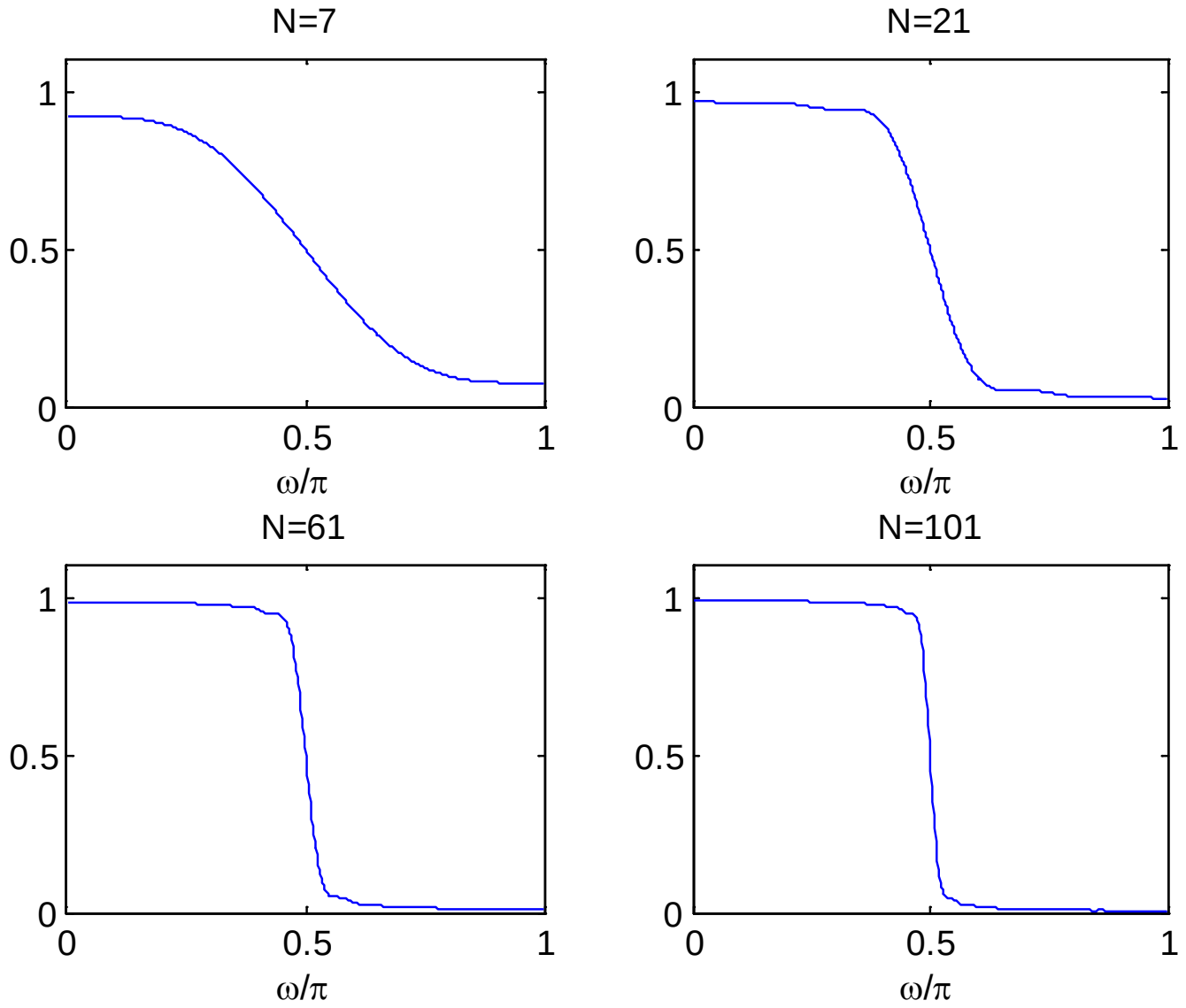


Fig.10.26: Magnitude responses with Bartlett window at different N

Example 10.10

Use the window method to find the impulse response of a linear-phase and causal FIR filter which approximates an ideal lowpass filter whose cutoff frequency is $\omega_c = 0.25\pi$. It is required that the filter is of fourth-order and Bartlett window is employed.

A fourth-order filter implies a filter length of $N = 5$. Using Example 10.7 with $N = 5$ and $\omega_c = 0.25\pi$:

$$h[n] = \frac{\sin(\omega_c(n-2))}{\pi(n-2)} \cdot w[n], \quad 0 \leq n \leq 4$$

To avoid zero coefficients, we set $N = 7$ in (10.20) and extract the middle 5 values to yield $w[n] = \{1/3, 2/3, 1, 2/3, 1/3\}$ with $w[0] = 1/3$.

As a result

$$h[n] = \{0.0531, 0.1501, 0.25, 0.1501, 0.0531\}$$

with $h[0] = 0.0531$

$H(z)$ is:

$$H(z) = 0.0531 + 0.1501z^{-1} + 0.25z^{-2} + 0.1501z^{-3} + 0.0531z^{-4}$$

The MATLAB command `fir1(4,0.25,triang(5),'noscale')` can also produce $h[n]$.

Note that if we use $N = 5$ in (10.20), $w[n] = \{0, 1/2, 1, 1/2, 0\}$ with $w[0] = 0$. The resultant impulse response will be $h[n] = \{0, 0.1125, 0.25, 0.1125, 0\}$ with $h[0] = 0$, indicating that the effective length is only 3

In window method, the ripple is:

$$\delta = \delta_1 = \delta_2$$

And the transition width is:

$$\Delta\omega = \omega_s - \omega_p$$

with

$$\omega_c = (\omega_p + \omega_s)/2$$

They are determined from $H(e^{j\omega}) = H_d(e^{j\omega})\tilde{\otimes}W(e^{j\omega})$ in (10.15), which is not straightforward for computation

Nevertheless, by noting that (10.15) corresponds to one of the forms in (10.9)-(10.12), $H_r(e^{j\omega})$ is easily obtained by extracting the amplitude

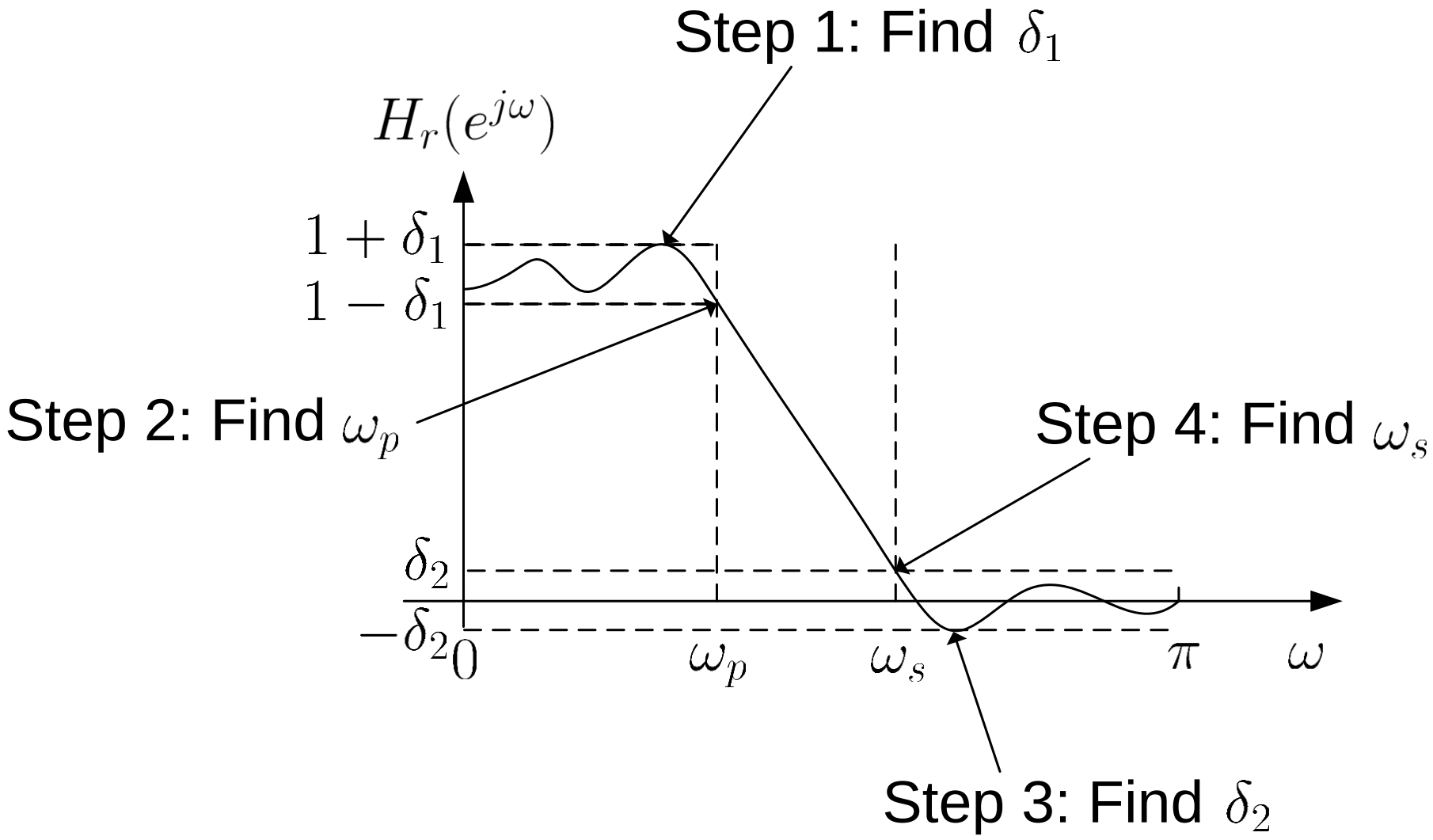


Fig.10.27: Steps to find passband and stopband ripples and frequencies

Example 10.11

Use the window method with the rectangular window to find the impulse response of a linear-phase and causal FIR filter which approximates an ideal lowpass filter whose cutoff frequency is $\omega_c = 0.5\pi$. It is required that the filter has a length of $N = 21$. Plot its amplitude response and then determine δ_1, δ_2, ω_p, ω_s and $\Delta\omega$.

Using Example 10.7 with $N = 21$ and $\omega_c = 0.5\pi$, $h[n]$ is:

$$h[n] = \frac{\sin(0.5\pi(n-10))}{\pi(n-10)}, \quad 0 \le n \le 20$$

As $h[n]$ is symmetric and N is odd, from (10.9):

$$H_r(e^{j\omega}) = 2\sum_{n=0}^{9} h[n]\cos\left((10-n)\,\omega\right) + h[10]$$

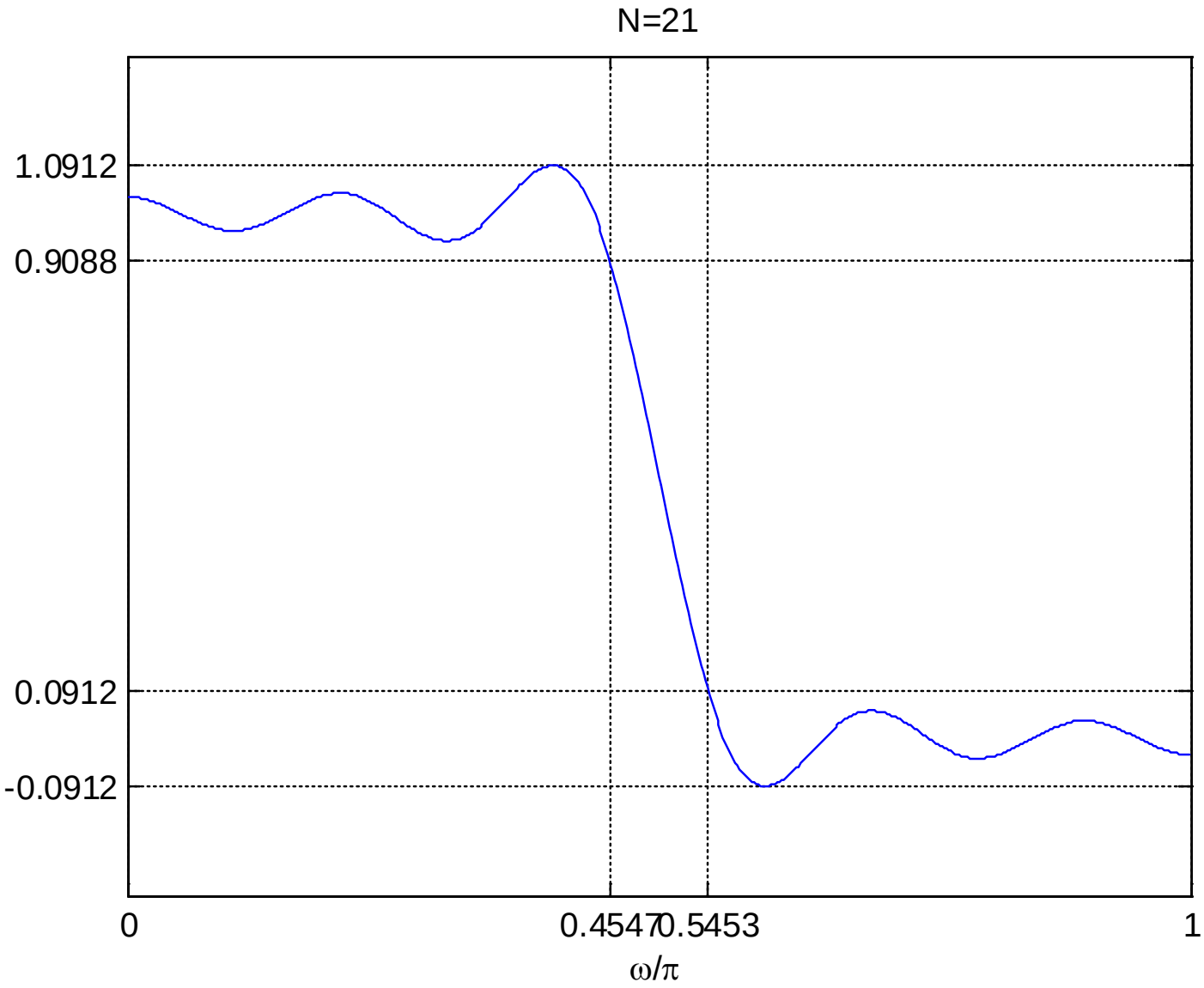


Fig.10.28: Measuring parameters from $H_r(e^{j\omega})$

We see that

$$\delta = \delta_1 = \delta_2 = 0.0912$$

Peak approximation error, e_p, which is the ripple in dB, is:

$$e_p = 20 \log_{10} \delta = 20 \log_{10} 0.0912 = -21 \text{ dB}$$

Furthermore

$$\omega_p = 0.4547\pi$$

and

$$\omega_s = 0.5453\pi$$

which gives

$$\Delta\omega = \omega_s - \omega_p = 0.2846$$

with

$$\omega_c = (\omega_p + \omega_s)/2 = 0.5\pi$$

The MATLAB program is provided as `ex10_11.m`.

Window	$\Delta\omega$	e_p (dB)
Rectangular	$1.8\pi/N$	-21
Bartlett	$6.1\pi/N$	-25
Hanning	$6.2\pi/N$	-44
Hamming	$6.6\pi/N$	-53
Blackman	$11\pi/N$	-74

Table 10.2: Transition width and ripple due to different window functions

Example 10.12

Use the window method to design a linear-phase and causal FIR filter which approximates an ideal lowpass filter whose cutoff frequency is $\omega_c = 0.5\pi$. The maximum allowable transition width is $\Delta\omega = 0.05\pi$ and the maximum allowable tolerance is $\delta = 0.005$.

The ripple of $\delta = 0.005$ corresponds to:

$$e_p = 20\log_{10}(0.005) = -46 \text{ dB}$$

From Table 10.2, Hamming and Blackman windows are the two candidates which can meet the ripple requirement

We choose the former because it involves a shorter filter length. The required length for the Hamming window is:

$$\Delta\omega = 0.05\pi = 6.6\pi/N \Rightarrow N = 132$$

Using Example 10.7 with $N = 132$ and $\omega_c = 0.5\pi$, the filter impulse response is:

$$h[n] = \frac{\sin(0.5\pi(n - 65.5))}{\pi(n - 65.5)} \cdot w[n], \quad 0 \leq n \leq 131$$

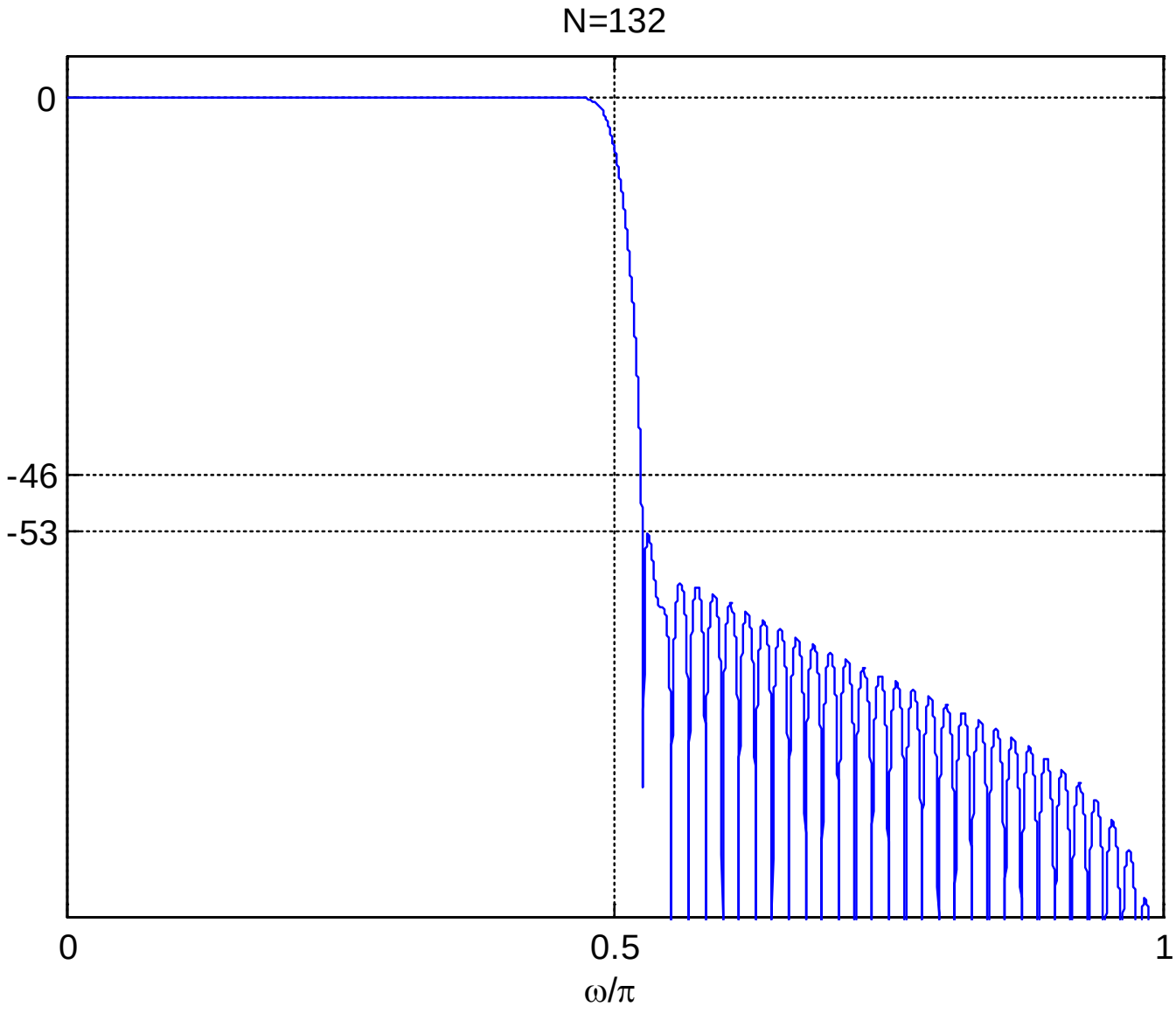


Fig.10.29: Magnitude response with Hamming window

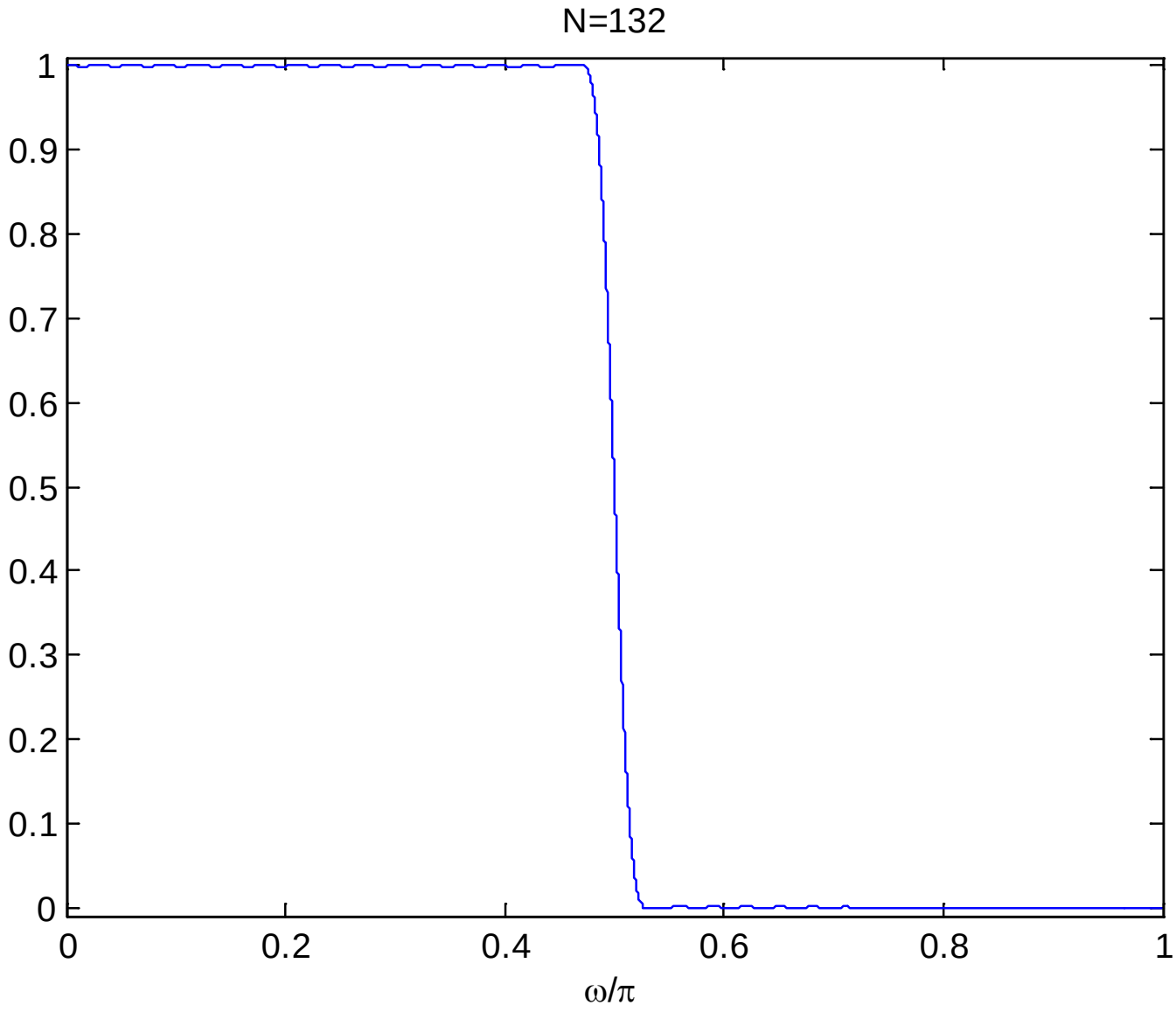


Fig.10.30: Amplitude response with Hamming window

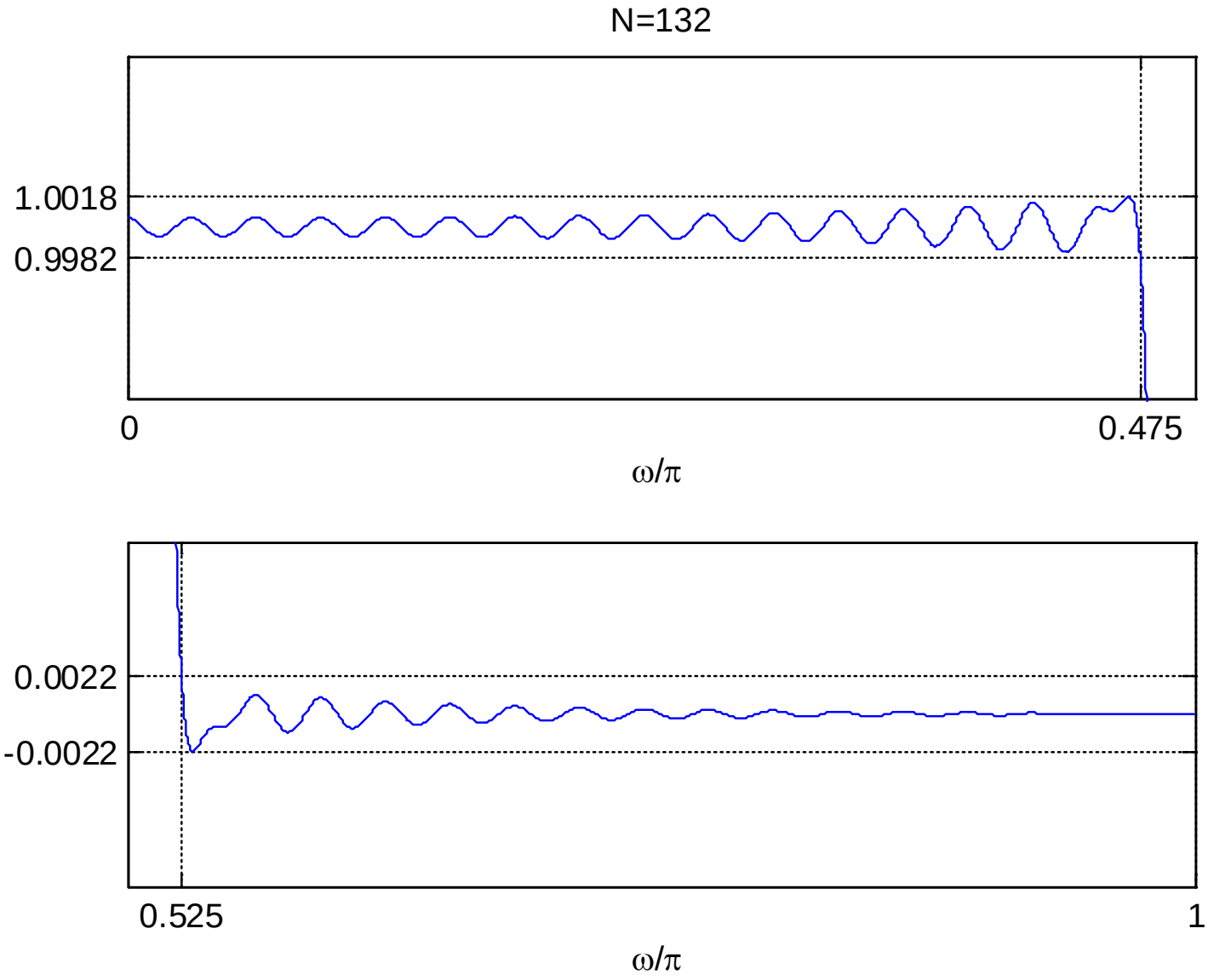


Fig.10.31: Zoomed amplitude responses with Hamming window

In summary:

$$e_p = -53 \text{ dB} \quad \text{or} \quad \delta \approx 0.002$$

and

$$\Delta\omega = 0.05\pi$$

with

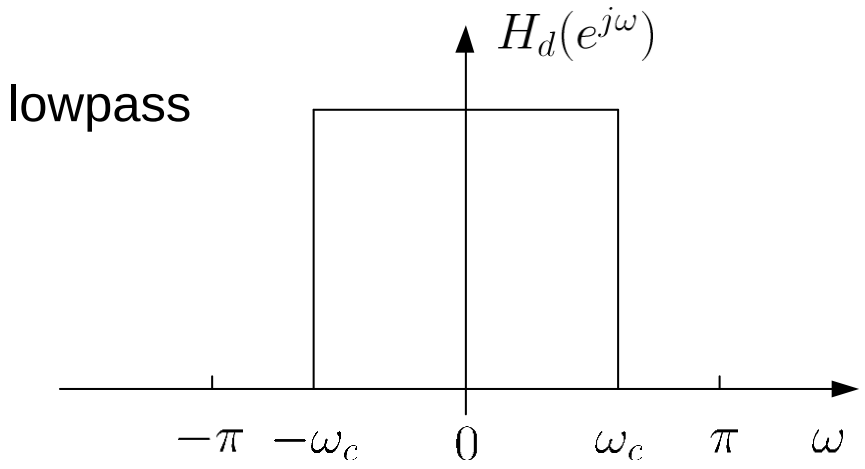
$$\omega_c = (\omega_p + \omega_s)/2 = 0.5\pi$$

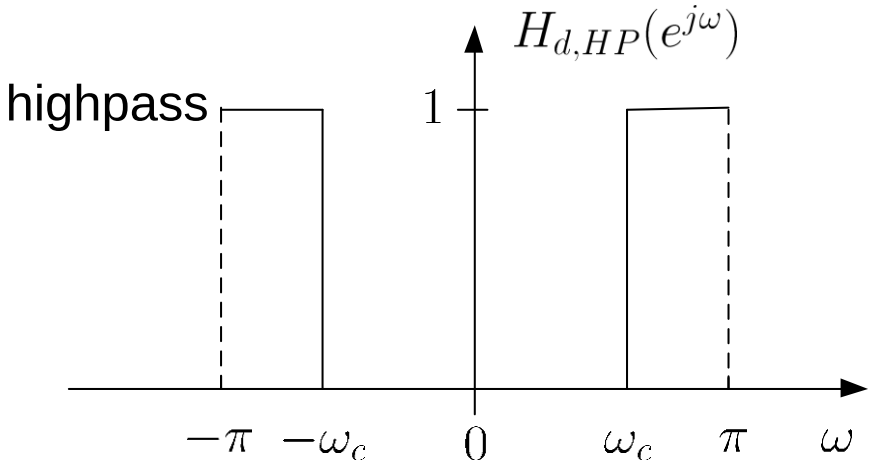
The designed filter meets the given specifications although it has a much smaller ripple than the desired value.

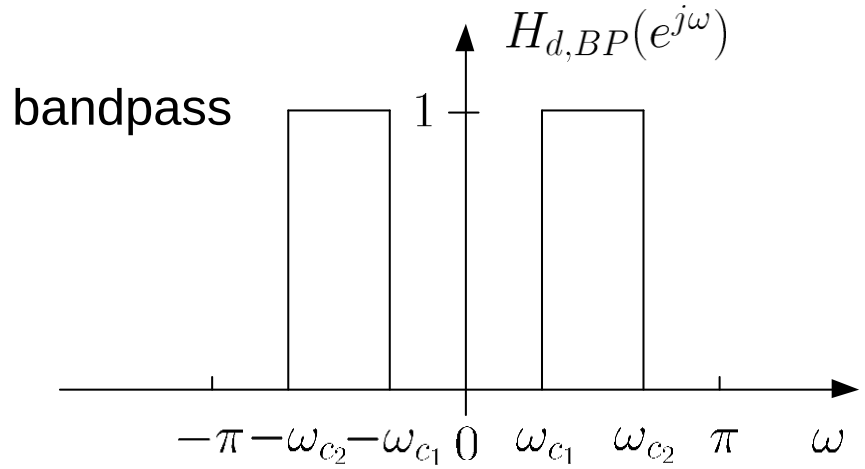
The MATLAB program is provided as `ex10_12.m`.

Extension to Typical Frequency Selective Filter Design

For typical frequency selective filters, namely, highpass, bandpass and bandstop filters, inverse DTFT can be used







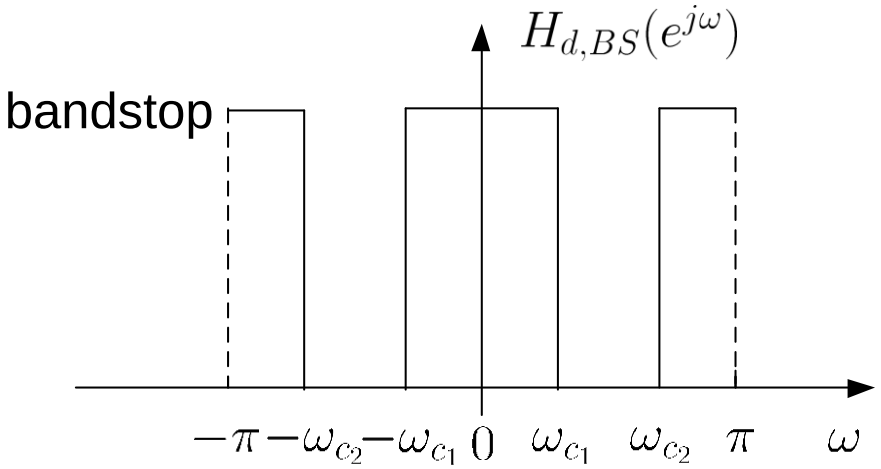


Fig.10.32: Typical frequency-selective filters

Nevertheless, we can utilize the lowpass filter to obtain impulse responses for $H_{d,HP}(e^{j\omega})$, $H_{d,BP}(e^{j\omega})$ and $H_{d,BS}(e^{j\omega})$

A highpass filter can be considered as subtraction of a lowpass filter from an allpass filter:

$$H_{d,HP}(e^{j\omega}) = 1 - H_d(e^{j\omega})$$

$$\Rightarrow h_{d,HP}[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j\omega n} d\omega - \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega n} d\omega = \delta[n] - \frac{\omega_c}{\pi}\text{sinc}\left(\frac{\omega_c n}{\pi}\right) \quad (10.24)$$

Assuming rectangular window:

$$h_{HP}[n] = \text{sinc}[n-\tau] - \frac{\omega_c}{\pi}\text{sinc}\left(\frac{\omega_c(n-\tau)}{\pi}\right) \quad (10.25)$$

where $\tau = (N-1)/2$

Example 10.13

Use the window method with the rectangular window to find the impulse response of a linear-phase and causal FIR filter which approximates an ideal highpass filter whose cutoff frequency is $\omega_c = 0.5\pi$. It is required that the filter has a length of $N = 21$.

According to (10.25) with $N = 21$ and $\omega_c = 0.5\pi$:

$$h_{HP}[n] = \delta[n - 10] - 0.5\text{sinc}\left(0.5(n - 10)\right), \quad 0 \leq N \leq 20$$

Note that we can also use the MATLAB command `fir1(20,0.5,'high',boxcar(21),'noscale')` to get the same result

The MATLAB program is provided as `ex10_13.m`.

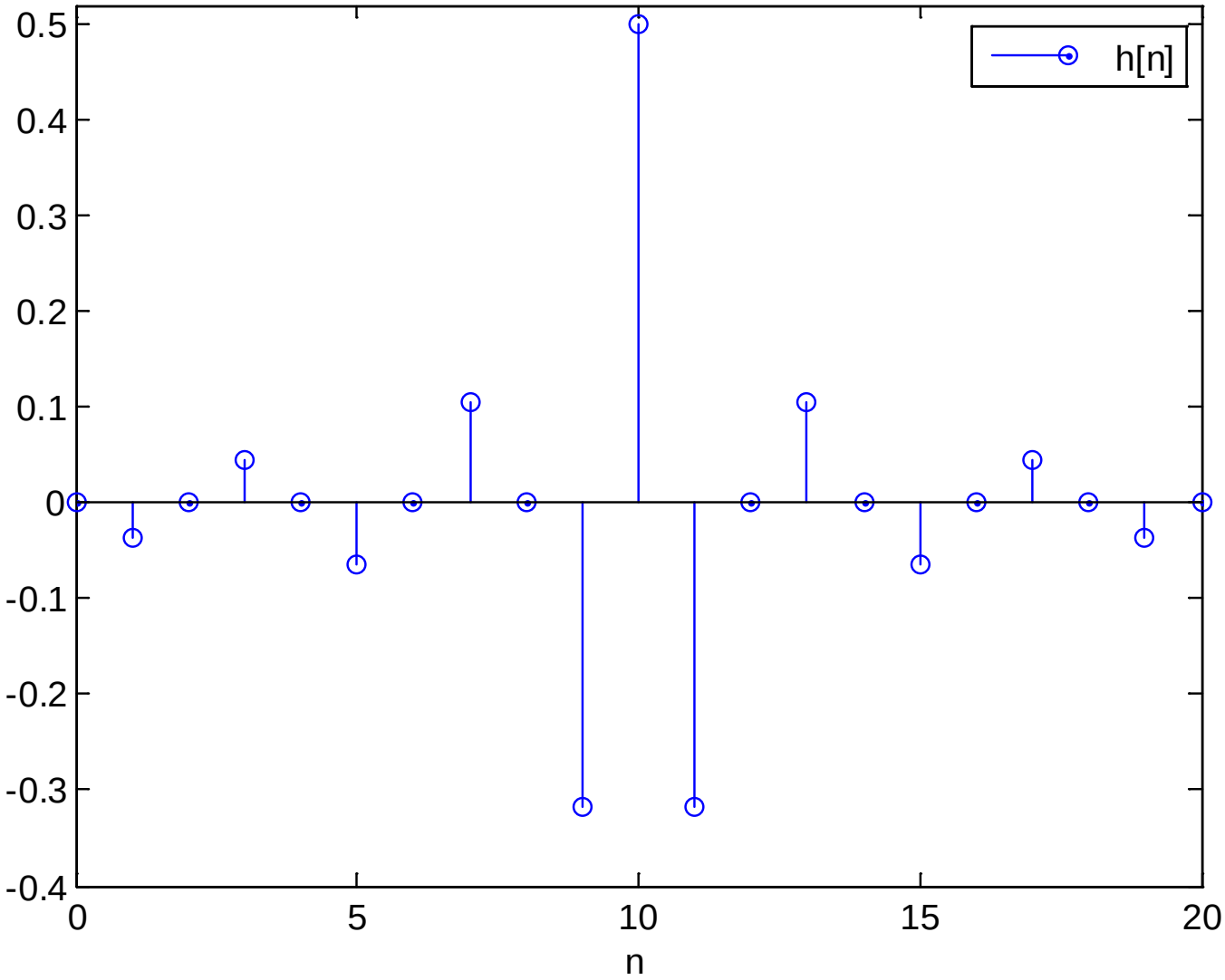


Fig.10.33: Impulse response of highpass filter with $N = 21$

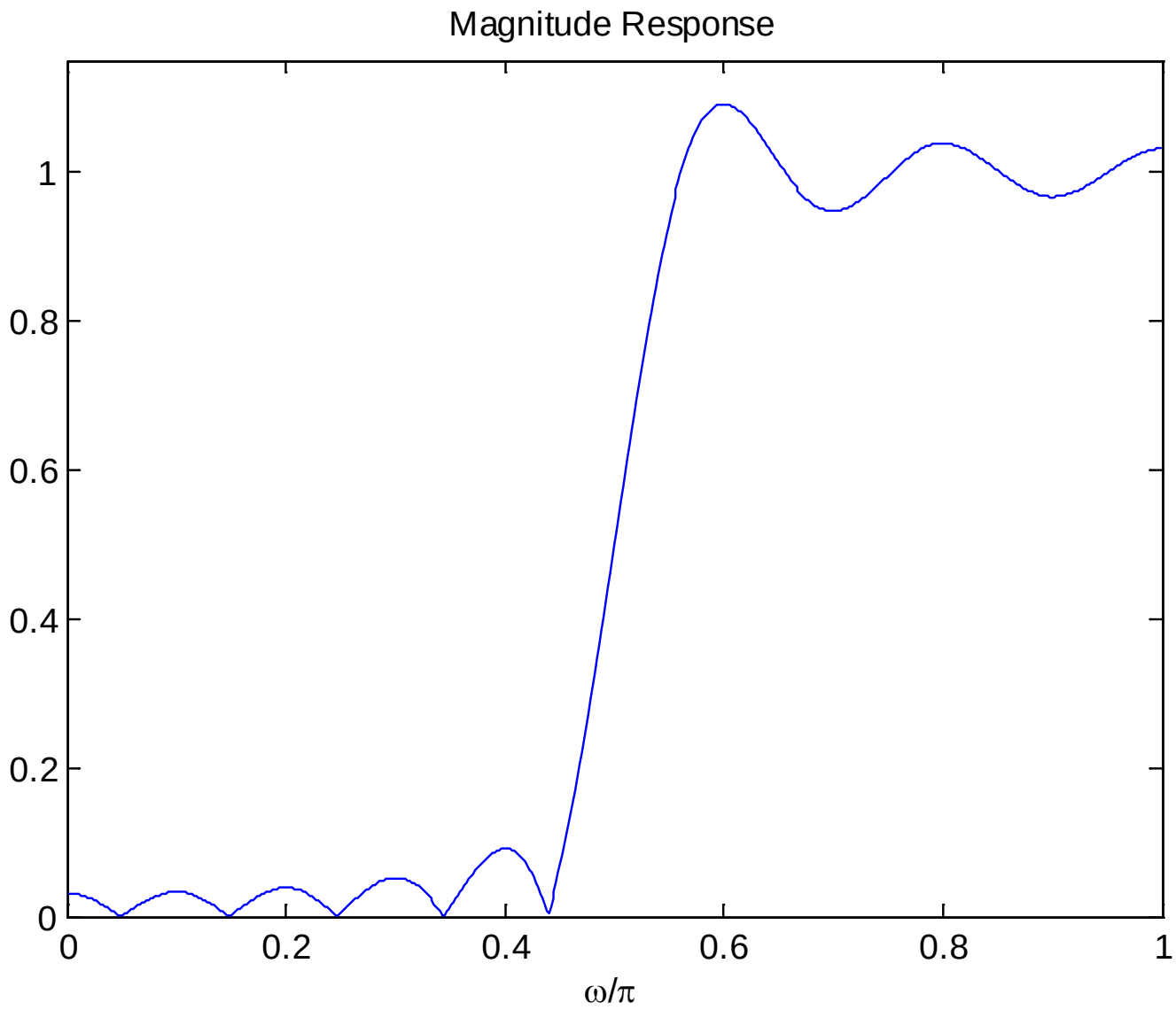


Fig.10.34: Magnitude response of highpass filter with $N = 21$

Similarly, an ideal bandpass filter can be designed from subtraction of two lowpass filters with cutoff frequencies ω_{c_1} and ω_{c_2}:

$$\begin{aligned} h_{d,BP}[n] &= \frac{1}{2\pi}\int_{-\omega_{c_2}}^{\omega_{c_2}} e^{j\omega n}d\omega - \frac{1}{2\pi}\int_{-\omega_{c_1}}^{\omega_{c_1}} e^{j\omega n}d\omega \\ &= \frac{\omega_{c_2}}{\pi}\text{sinc}\left(\frac{\omega_{c_2}n}{\pi}\right) - \frac{\omega_{c_1}}{\pi}\text{sinc}\left(\frac{\omega_{c_1}n}{\pi}\right) \end{aligned} \qquad (10.26)$$

An ideal bandstop filter impulse response can be obtained by subtracting a bandpass filter from an allpass filter:

$$h_{d,BS}[n] = \delta[n] - \frac{\omega_{c_2}}{\pi}\text{sinc}\left(\frac{\omega_{c_2}n}{\pi}\right) + \frac{\omega_{c_1}}{\pi}\text{sinc}\left(\frac{\omega_{c_1}n}{\pi}\right) \qquad (10.27)$$

Kaiser Window

A problem in typical window functions is that we cannot control the ripple δ

Kaiser window can control both δ and $\Delta\omega$ in a nearly optimal manner and it has the form of:

$$w[n] = \frac{I_0\left(\beta\sqrt{1-\left(1-\frac{2n}{N-1}\right)^2}\right)}{I_0(\beta)}, \quad 0 \le n \le N-1 \qquad (10.28)$$

where I_0 is the modified zero-order Bessel function. Apart from N, there is β which is responsible for the window shape

That is, by properly choosing N and β, the design specifications in terms of δ and $\Delta\omega$ can be precisely met

N is computed as:

$$N = \left\lceil \frac{-e_p - 8}{2.285\Delta\omega} \right\rceil + 1 \qquad (10.29)$$

where $e_p = 20\log_{10}\delta$ is the peak approximation error and $\lceil u \rceil$ rounds up u to the nearest integer

β is determined from:

$$\beta = \begin{cases} 0.1102(-e_p - 8.7), & -e_p > 50 \\ 0.5842(-e_p - 21)^{0.4} + 0.07886(-e_p - 21), & 21 \le -e_p \le 50 \\ 0, & -e_p < 21 \end{cases} \qquad (10.30)$$

Example 10.14

Use the window method with the Kaiser window to design a linear-phase and causal FIR filter which approximates an ideal lowpass filter whose cutoff frequency is $\omega_c = 0.5\pi$. The maximum allowable transition width is $\Delta\omega = 0.05\pi$ and the maximum allowable tolerance is $\delta = 0.005$.

Since

$$-e_p = -20\log_{10}(0.005) = 46$$

and $\Delta\omega = 0.05\pi$, we then have:

$$N = \left\lceil \frac{46 - 8}{2.285 \cdot 0.05\pi} \right\rceil + 1 = 107$$

and

$$\beta = 0.5842(46 - 21)^{0.4} + 0.07886(46 - 21) = 4.09$$

As a result, the filter impulse response is:

$$h[n] = \frac{\sin(0.5\pi(n-53))}{\pi(n-53)} \cdot w[n], \quad 0 \leq n \leq 106$$

where $w[n]$ is determined from (10.28).

In summary:

$$e_p = -46 \text{ dB} \text{ or } \delta \approx 0.005$$

$$\Delta\omega = 0.05\pi$$

with

$$\omega_c = (\omega_p + \omega_s)/2 = 0.5\pi$$

The designed filter exactly meets the given specifications

The MATLAB program is provided as `ex10_14.m`.

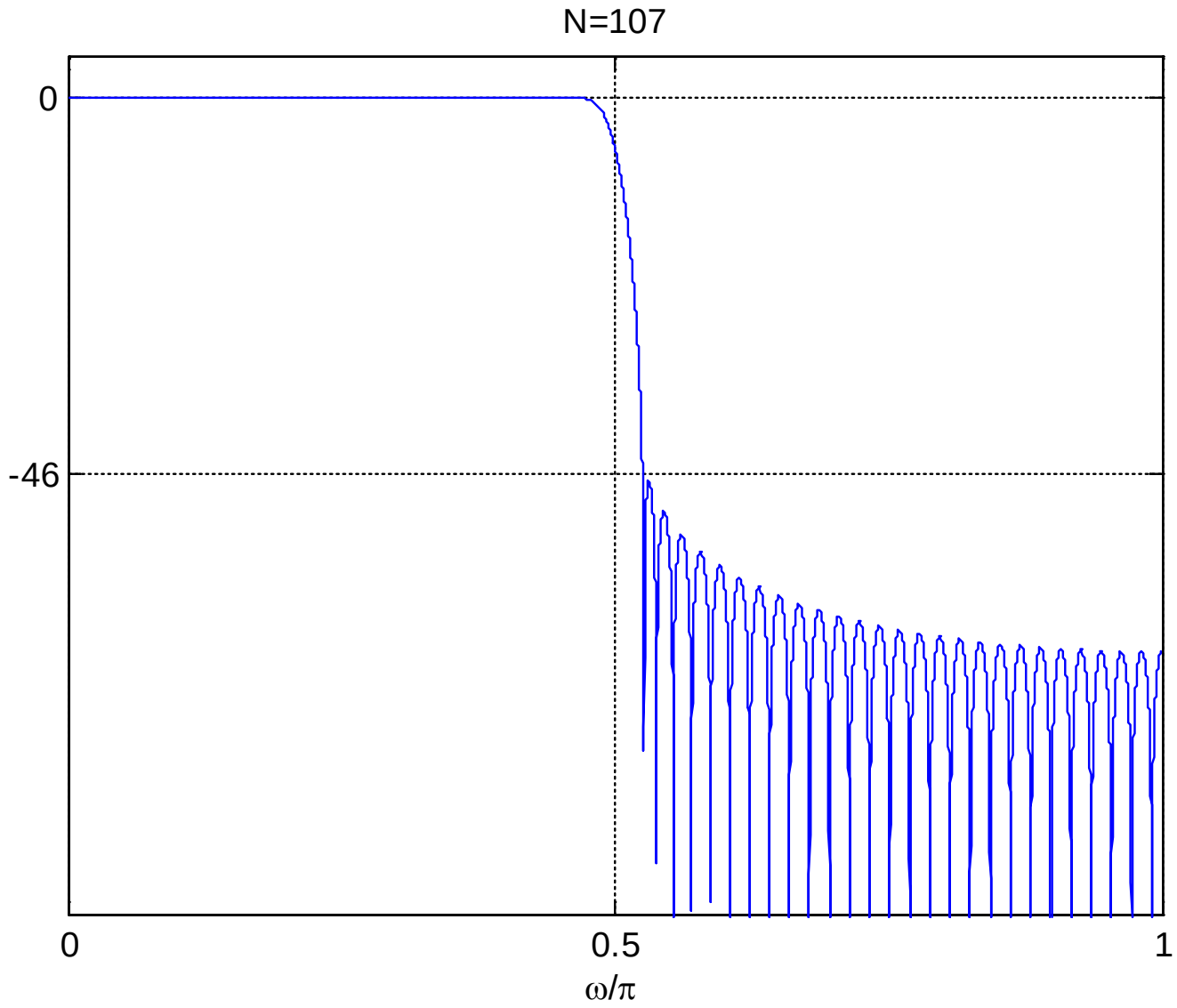


Fig.10.35: Magnitude response with Kaiser window

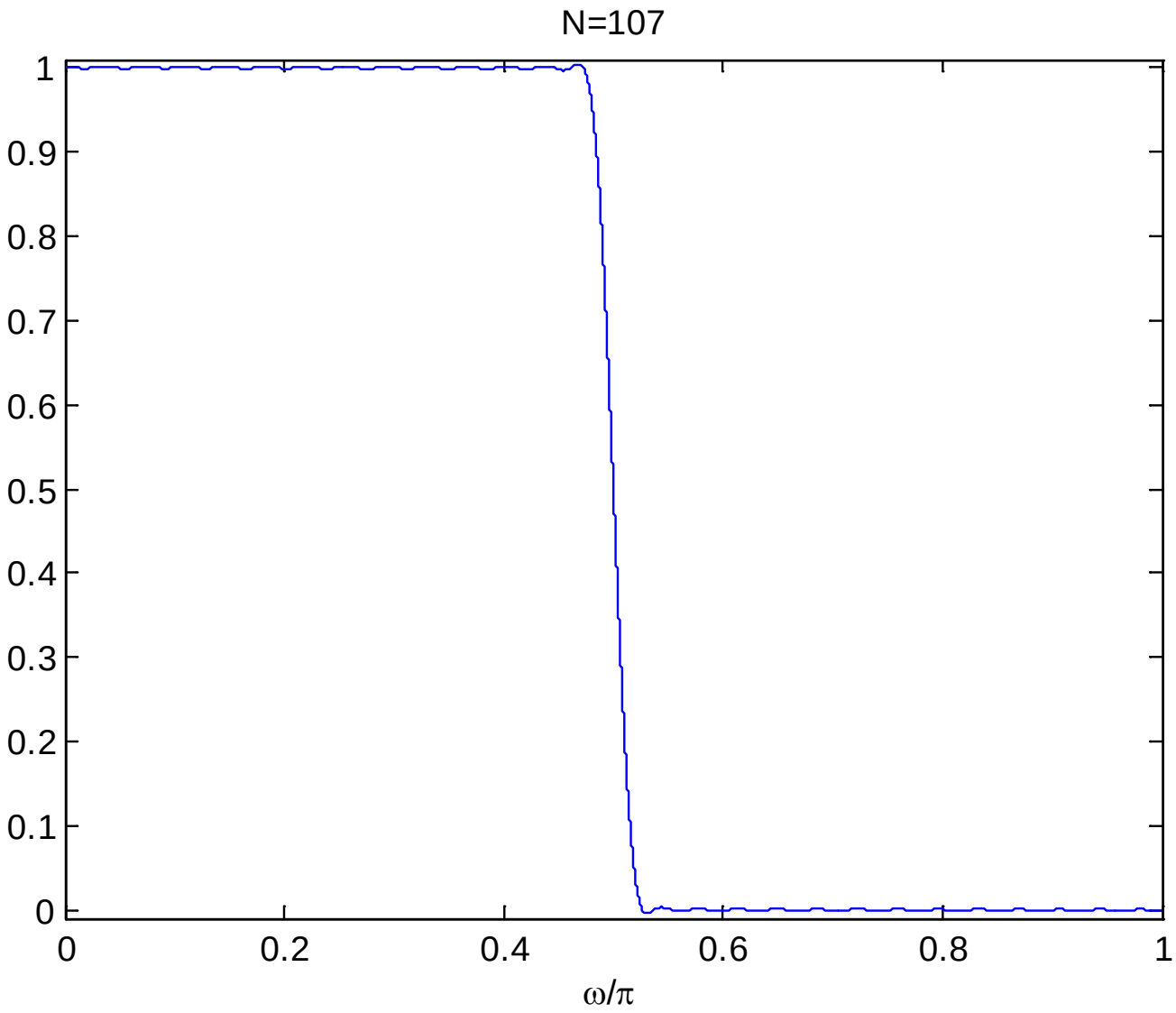


Fig.10.36: Amplitude response with Kaiser window

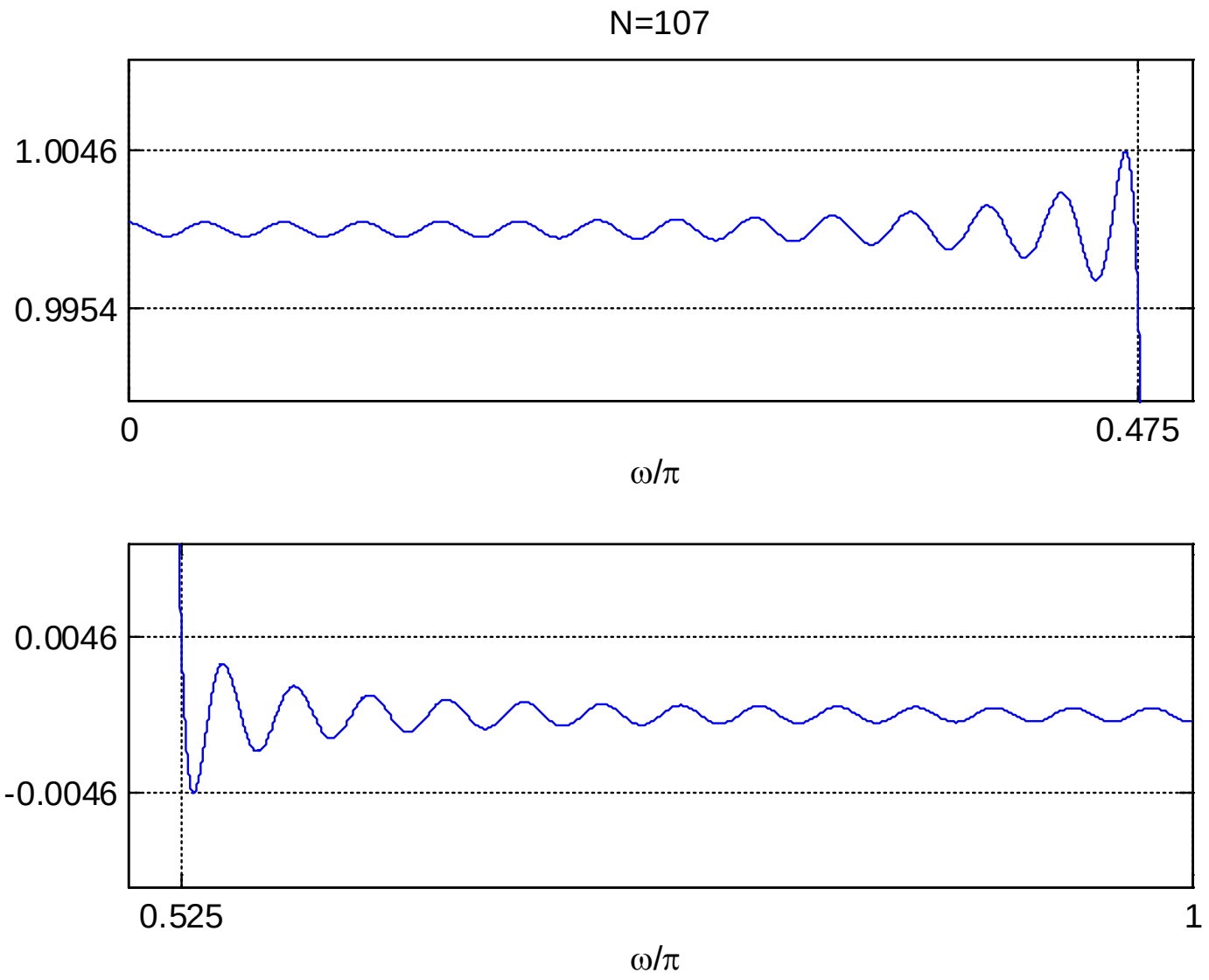


Fig.10.37: Zoomed amplitude responses with Kaiser window

Frequency Sampling Method

The basic idea is to utilize the discrete Fourier transform (DFT), which corresponds to samples of the desired frequency response $H_d(e^{j\omega})$, to produce $h[n]$

Recall:

$$H[k] = H_d(e^{j\omega})|_{\omega=\frac{2\pi k}{N}}, \quad k = 0, 1, \cdots, N-1 \qquad (10.31)$$

is equal to $H_d(e^{j\omega})$ sampled at N distinct frequencies between $\omega \in [0, 2\pi]$ with a uniform frequency spacing of $2\pi/N$

A causal and linear-phase filter is designed with 2 steps:

- Extract N uniformly-spaced samples from $H_d(e^{j\omega})$ in the frequency range of $\omega \in [0, 2\pi]$
- Compute $h[n]$ by taking the inverse DFT of $H[k]$:

$$h[n] = \frac{1}{N}\sum_{k=0}^{N-1} H[k]W_N^{-kn}, \quad W_N = e^{-j2\pi/N},\ 0 \le n \le N-1 \qquad (10.32)$$

Taking z transform of $h[n]$:

$$\begin{aligned} H(z) &= \sum_{n=0}^{N-1} h[n]z^{-n} \\ &= \sum_{n=0}^{N-1}\left(\frac{1}{N}\sum_{k=0}^{N-1} H[k]e^{j2\pi kn/N}\right) z^{-n} \\ &= \frac{1}{N}\sum_{k=0}^{N-1} H[k]\sum_{n=0}^{N-1}\left(e^{j(2\pi/N)k}z^{-1}\right)^n \\ &= \frac{1-z^{-N}}{N}\sum_{k=0}^{N-1}\frac{H[k]}{1-e^{j(2\pi/N)k}z^{-1}} \end{aligned} \qquad (10.33)$$

The filter frequency response is then:

$$H(e^{j\omega}) = \frac{1 - e^{-j\omega N}}{N} \sum_{k=0}^{N-1} \frac{H[k]}{1 - e^{j(2\pi/N)k} e^{-j\omega}} \qquad (10.34)$$

From (10.33) and (7.7) together with (10.31):

$$H(e^{j2\pi r/N}) = \frac{1}{N} \sum_{k=0}^{N-1} H[k] \sum_{n=0}^{N-1} e^{j(2\pi/N)(k-r)n} = H[r]$$

$$\Rightarrow H(e^{j\omega})|_{\omega=\frac{2\pi k}{N}} = H[k] = H_d(e^{j\omega})|_{\omega=\frac{2\pi k}{N}}, \quad k = 0, 1, \cdots, N-1 \qquad (10.35)$$

It is simple as only uniformly-spaced samples of the desired frequency response or the DFT coefficients are needed

At $\omega = (2\pi/N)k$, $k = 0, 1, \cdots, N-1$, $H(e^{j\omega})$ equals $H_d(e^{j\omega})$ but we cannot control the values of the remaining frequency points

It lacks flexibility in specifying the passband and stopband cutoff frequencies since placement of “1” & “0” & transition samples is constrained to integer multiples of $2\pi/N$

Example 10.15

Use the frequency sampling method to design a linear-phase and causal FIR filter with a length of $N = 7$ to approximate an ideal lowpass filter whose cutoff frequency is $\omega_c = 0.1\pi$.

From Example 10.7, the ideal frequency response with linear phase is

$$H_d(e^{j\omega}) = \begin{cases} 1 \cdot e^{-j\omega\tau}, & |\omega| < \omega_c \\ 0, & -\pi < \omega < -\omega_c, \quad \omega_c < \omega < \pi \end{cases}$$

where $\tau = (N-1)/2 = 3$ and $\omega_c = 0.1\pi$

We consider the frequency interval of $\omega \in [0, 2\pi]$:

$$H_d(e^{j\omega}) = \begin{cases} 1 \cdot e^{-j\omega\tau}, & 0 < \omega < \omega_c \\ 0, & \omega_c < \omega < 2\pi - \omega_c \\ 1 \cdot e^{-j(\omega - 2\pi)\tau}, & 2\pi - \omega_c < \omega < 2\pi \end{cases}$$

Extracting the values of $H_d(e^{j\omega})$ at $\omega = (2\pi/7)k$, $k = 0, 1, \cdots, 6$:

$$H[k] = \begin{cases} 1, & k = 0 \\ 0, & k = 1, 2, \cdots, 6 \end{cases}$$

Taking the inverse DFT:

$$h[n] = \frac{1}{7}, \quad n = 0, 1, \cdots, 6$$

The MATLAB program is provided as `ex10_15.m`.

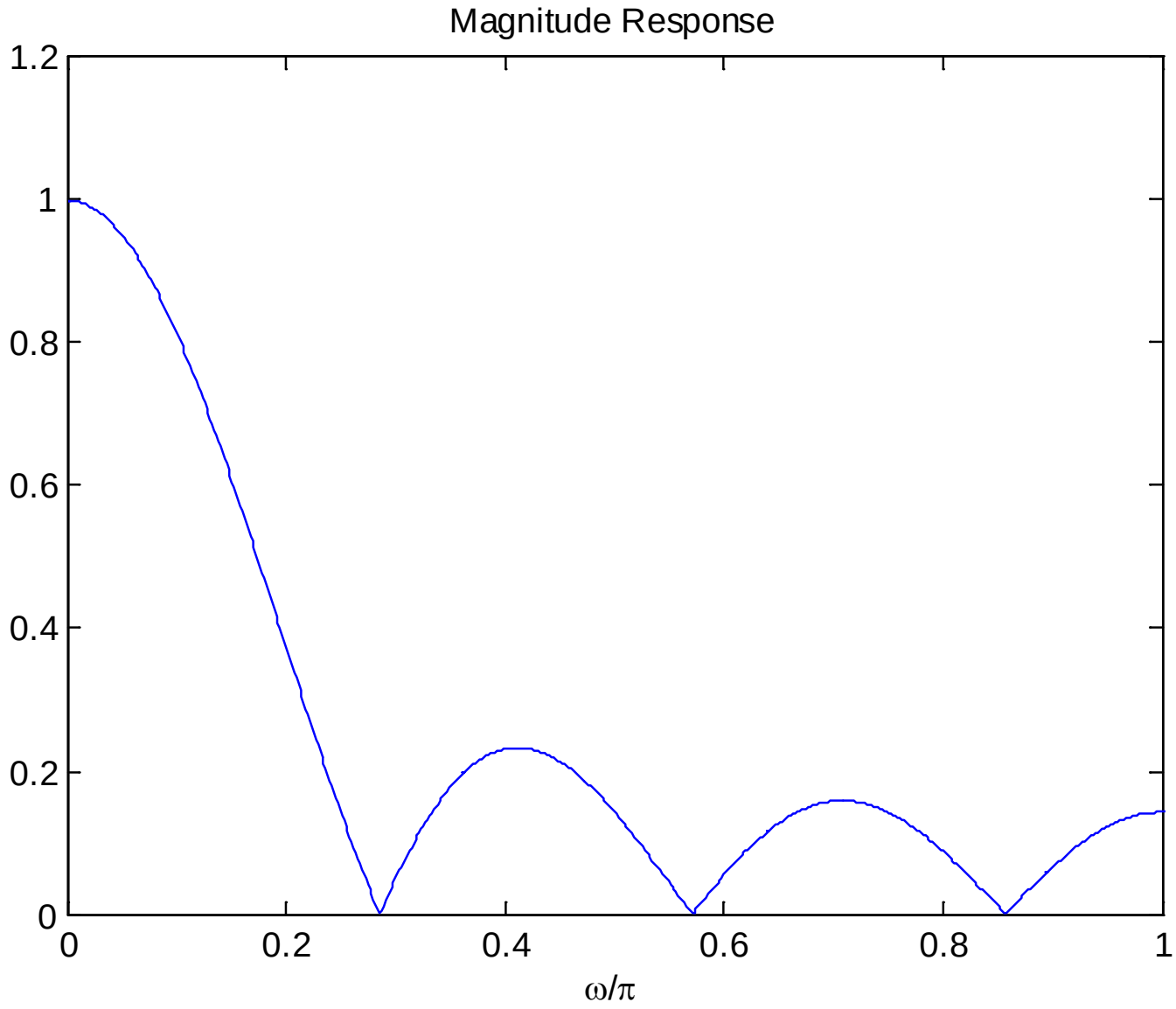


Fig.10.38: Magnitude response based on frequency sampling

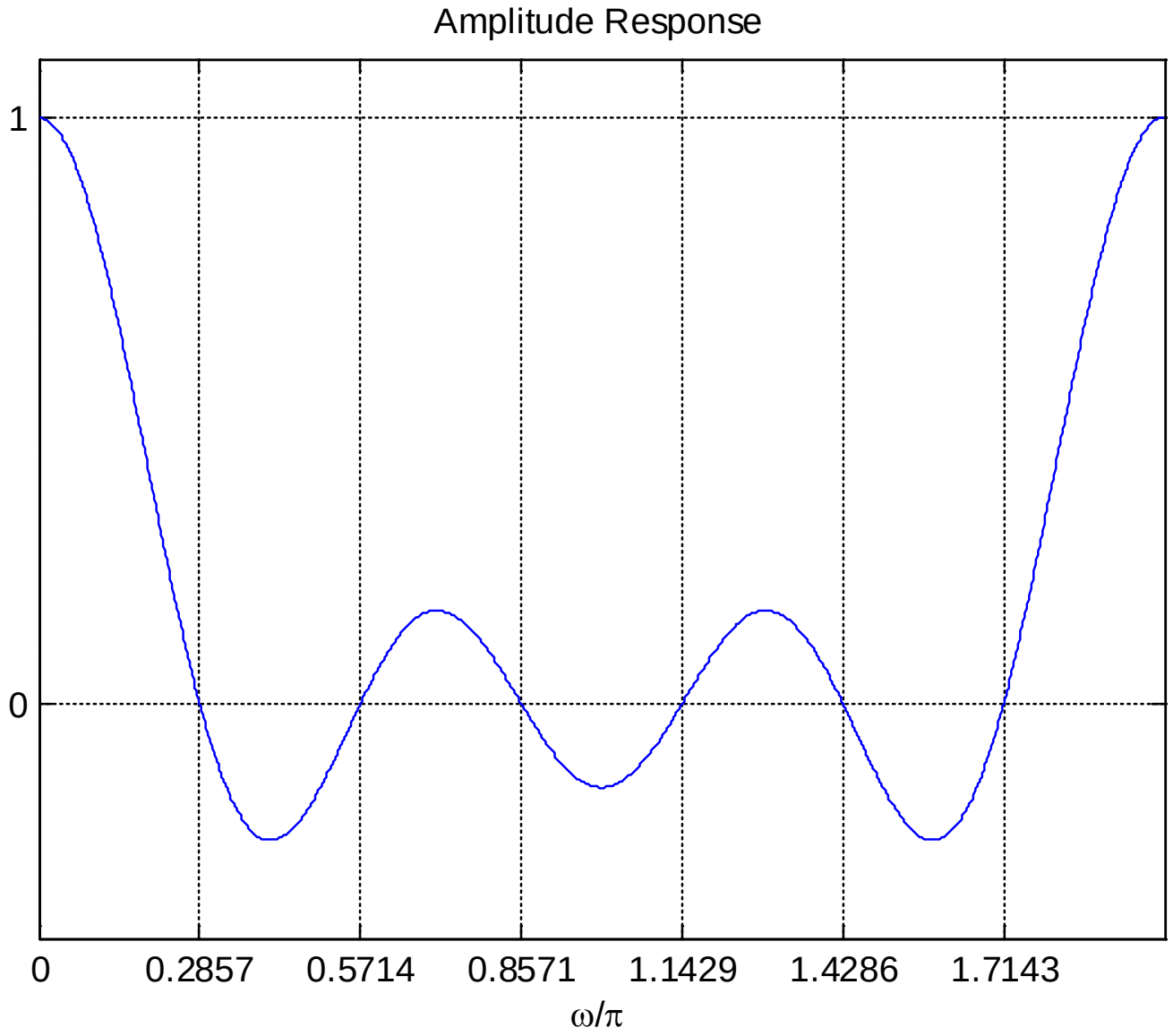


Fig.10.39: Amplitude response based on frequency sampling

Optimal Equiripple Method

The basic idea is to evenly distribute the ripples in both passband and stopband

- Required filter length will be shorter than that of the window method where its $H(e^{j\omega})$ exactly meets the passband or stopband ripple specification at one frequency and is superior to it at other frequencies in the band

- Allow $\delta_1 \neq \delta_2$

- Passband and stopband frequencies can be precisely specified although $\omega_p = \omega_c - \Delta\omega/2$ and $\omega_s = \omega_c + \Delta\omega/2$ are implicitly implied in the window method

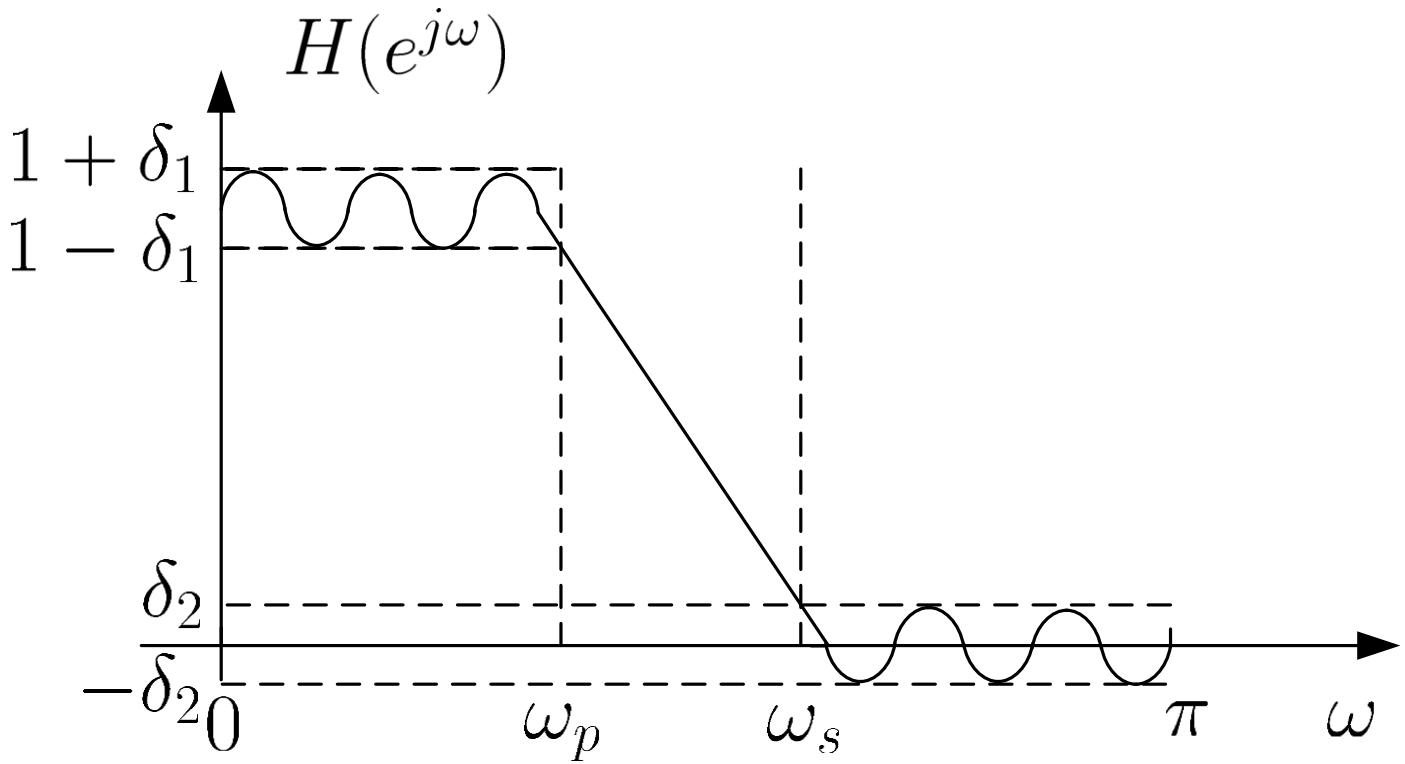


Fig.10.40: Illustration of optimal equiripple lowpass filter

Ripples are uniformly distributed such that $H(e^{j\omega})$ reaches its maximum deviations of δ_1 and δ_2 more than once

The impulse response $h[n]$ of optimal equiripple design is determined from:

$$h[n] = \min_{\{\tilde{h}[n]\}} \left(\max_{0 \le \omega \le \omega_p, \omega_s \le \omega \le \pi} |E(e^{j\omega})| \right), \quad n = 0, 1, \cdots, N-1 \tag{10.36}$$

where

$$E(e^{j\omega}) = W(e^{j\omega})(H_d(e^{j\omega}) - \tilde{H}(e^{j\omega})) \tag{10.37}$$

which corresponds to a minmax optimization problem

$E(e^{j\omega})$ is the frequency-domain error between the desired and actual responses weighted by $W(e^{j\omega})$

$W(e^{j\omega})$ is the weighting function incorporates all specification parameters, namely, δ_1, δ_2, ω_p and ω_s, into the design process

For example, in lowpass filter design, $W(e^{j\omega})$ has the form of:

$$W(e^{j\omega}) = \begin{cases} \dfrac{\delta_2}{\delta_1}, & 0 \leq \omega \leq \omega_p \\ 1, & \omega_s \leq \omega \leq \pi \end{cases} \tag{10.38}$$

When $\delta_1 > \delta_2$, there is a larger weighting at the stopband. On the other hand, $\delta_1 < \delta_2$ implies a larger weighting at the passband

To solve for the minmax problem of (10.36), we can make use of the Parks-McClellan algorithm which requires iterations. The corresponding MATLAB command is `firpm` where the filter length parameter N is also required.

We can employ

$$N = \left\lceil \frac{-10\log_{10}(\delta_1\delta_2) - 13}{2.324(\omega_s - \omega_p)} \right\rceil \tag{10.39}$$

to get its initial estimate and then compute $h[n]$. If the tolerance specifications are not met, we increment N until the maximum deviations are bounded by δ_1 and δ_2.

Example 10.16

Use the optimal equiripple method to design a linear-phase and causal FIR filter which approximates an ideal lowpass filter whose passband frequency is $\omega_p = 0.475\pi$ and stopband frequency is $\omega_p = 0.525\pi$. The maximum allowable tolerance is $\delta = \delta_1 = \delta_2 = 0.005$ in both passband and stopband.

According to (10.39), an initial value of N is computed as:

$$N = \left\lceil \frac{-10\log_{10}(0.005 \cdot 0.005) - 13}{2.324(0.525\pi - 0.425\pi)} \right\rceil = 91$$

Starting with $N = 91$ in the Parks-McClellan algorithm, we increment its value until $N = 96$ so that the tolerance specifications are met.

The MATLAB program is provided as `ex10_16.m`.

In summary:

$$e_p = -46 \text{ dB} \text{ or } \delta_1 = \delta_2 = 0.005$$

$$\omega_p = 0.475\pi, \ \omega_s = 0.525\pi$$

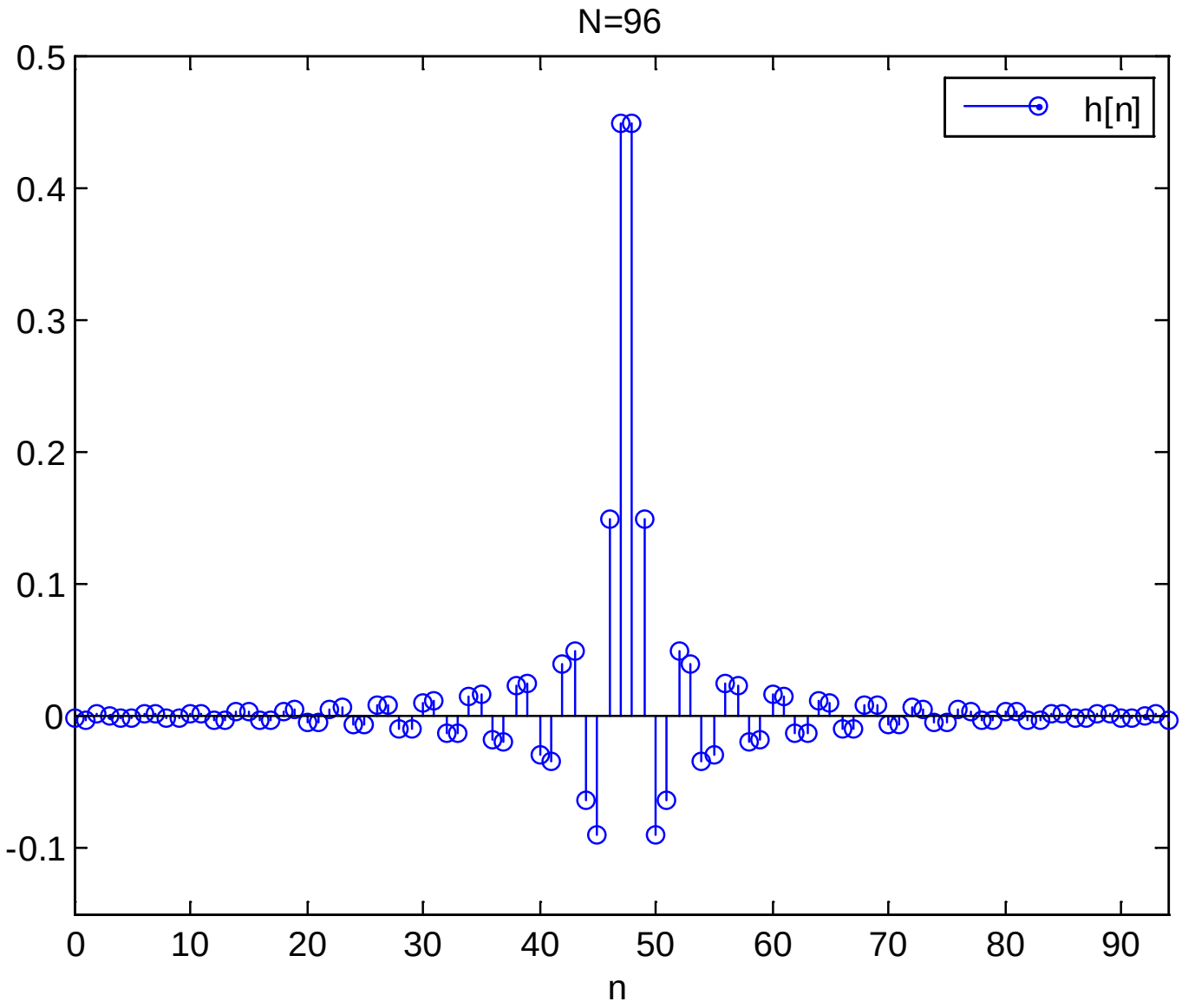


Fig.10.41: Impulse response of optimal equiripple lowpass filter

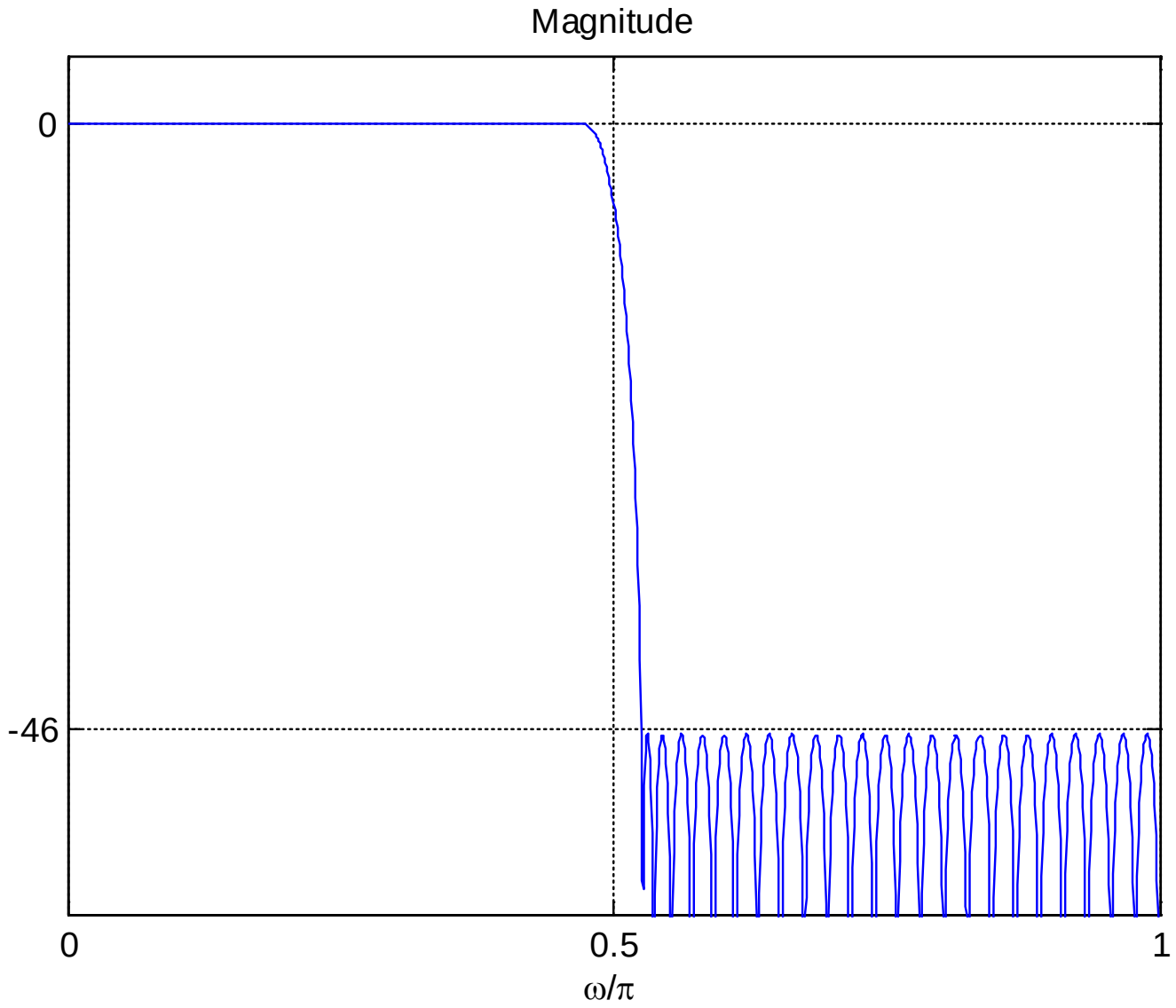


Fig.10.42: Magnitude response of optimal equiripple lowpass filter

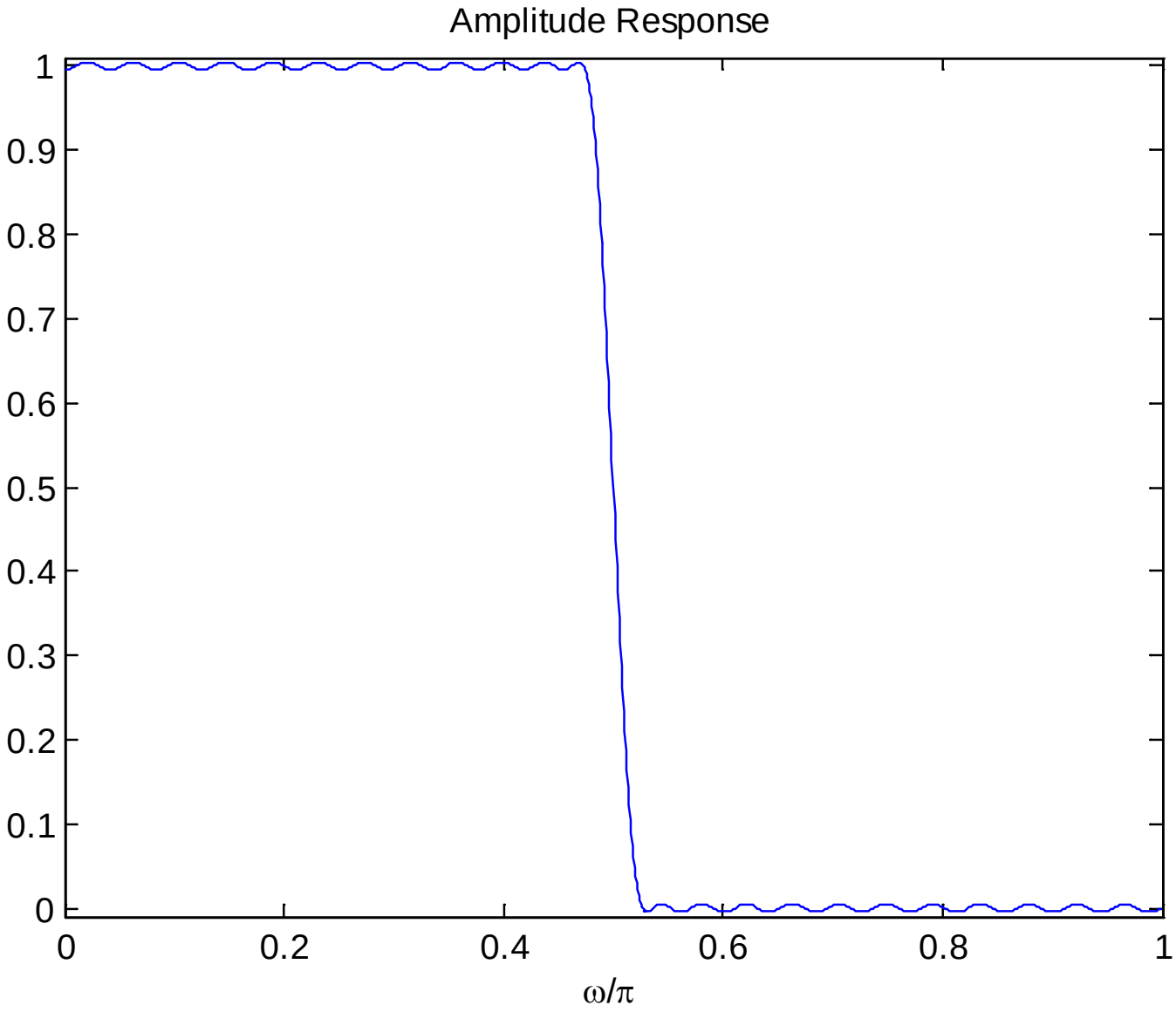


Fig.10.43: Amplitude response of optimal equiripple lowpass filter

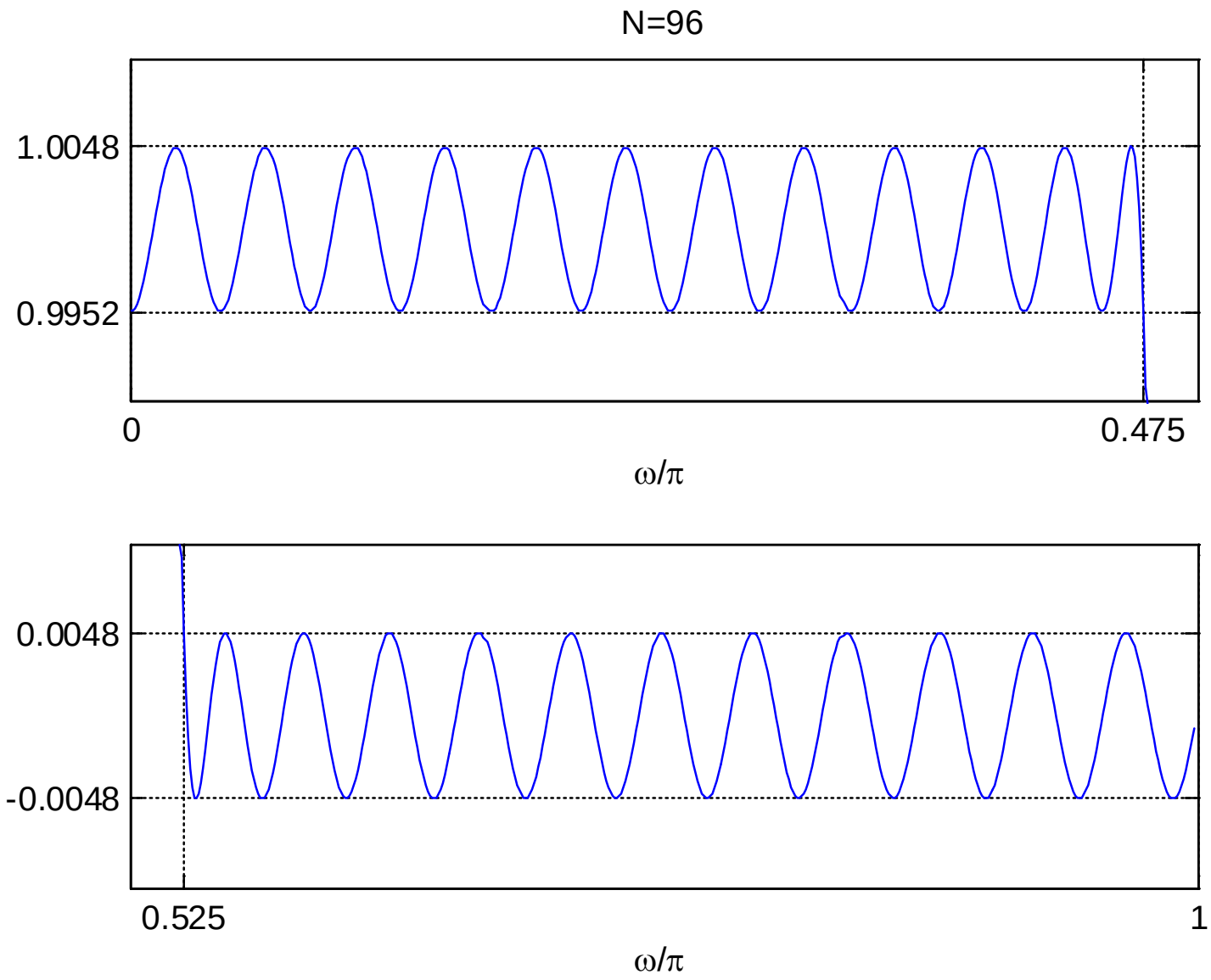


Fig.10.44: Zoomed amplitude responses of optimal equiripple lowpass filter

Chapter 11: IIR Filter Design

Chapter Intended Learning Outcomes:

(i) Ability to design analog Butterworth filters

(ii) Ability to design lowpass IIR filters according to predefined specifications based on analog filter theory and analog-to-digital filter transformation

(iii) Ability to construct frequency-selective IIR filters based on a lowpass IIR filter

Steps in Infinite Impulse Response Filter Design

The system transfer function of an IIR filter is:

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \qquad (11.1)$$

The task in IIR filter design is to find $\{a_k\}$ and $\{b_k\}$ such that $H(z)$ satisfies the given specifications.

Once $H(z)$ is computed, the filter can then be realized in hardware or software according to a direct, canonic, cascade or parallel form

We make use of the analog filter design to produce the required $H(z)$

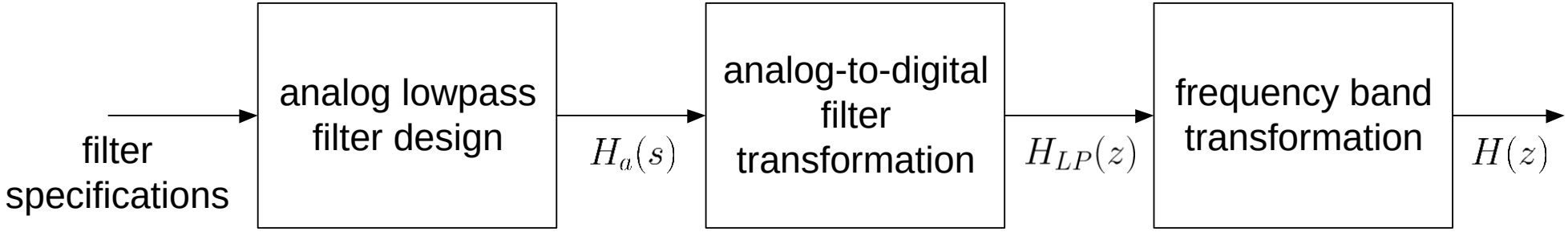


Fig.11.1: Steps in determining transfer function of IIR filter

Note that s is the Laplace transform parameter and substituting $s = j\Omega$ in $H_a(s)$ yields the Fourier transform of the filter, that is, $H_a(j\Omega)$

Main drawback is that there is no control over the phase response of $H(e^{j\omega})$, implying that the filter requirements can only be specified in terms of magnitude response

Butterworth Lowpass Filter Design

In analog lowpass filter design, we can only specify the magnitude of $H_a(j\Omega)$. Typically, we employ the magnitude square response, that is, $|H_a(j\Omega)|^2$:

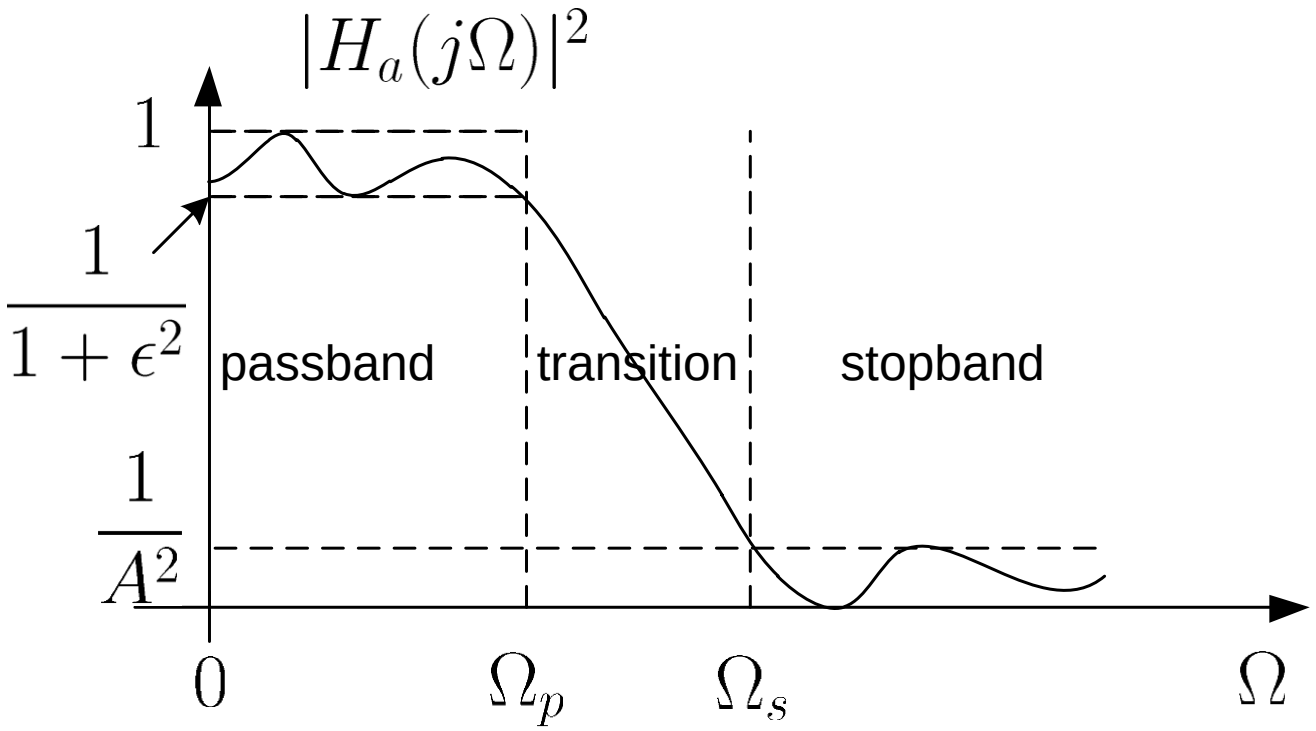


Fig.11.2: Specifications of analog lowpass filter

Passband corresponds to $\Omega \in [0, \Omega_p]$ where Ω_p is the passband frequency and ϵ is called the passband ripple

Stopband corresponds to $\Omega \in [\Omega_s, \infty)$ where Ω_s is the stopband frequency and A is called the stopband attenuation

Transition band corresponds to $\Omega \in [\Omega_p, \Omega_s]$

The specifications are represented as the two inequalities:

$$\frac{1}{1+\epsilon^2} \leq |H_a(j\Omega)|^2 \leq 1, \quad 0 \leq \Omega \leq \Omega_p \tag{11.2}$$

and

$$0 \leq |H_a(j\Omega)|^2 \leq \frac{1}{A^2}, \quad \Omega \geq \Omega_s \tag{11.3}$$

In particular, at $\Omega = \Omega_p$ and $\Omega = \Omega_s$, we have:

$$|H_a(j\Omega_p)|^2 = \frac{1}{1+\epsilon^2} \tag{11.4}$$

and

$$|H_a(j\Omega_s)|^2 = \frac{1}{A^2} \tag{11.5}$$

Apart from ϵ and A, it is also common to use their respective dB versions, denoted by R_p and A_s:

$$R_p = -10\log_{10}\left(\frac{1}{1+\epsilon^2}\right) \Rightarrow \epsilon = \sqrt{10^{R_p/10}-1} \tag{11.6}$$

and

$$A_s = -10\log_{10}\left(\frac{1}{A^2}\right) \Rightarrow A = 10^{A_s/20} \tag{11.7}$$

The magnitude square response of a Nth-order Butterworth lowpass filter is:

$$|H_a(j\Omega)|^2 = \frac{1}{1+\left(\frac{\Omega}{\Omega_c}\right)^{2N}} \tag{11.8}$$

The filter is characterized by Ω_c and N, which represent the cutoff frequency and filter order

- $|H_a(j\Omega)|^2 = 1$ at $\Omega = 0$ and $|H_a(j\Omega)|^2 = 0.5$ at $\Omega = \Omega_c$ for all N
- $|H_a(j\Omega)|^2$ is a monotonically decreasing function of frequency which indicates that there is no ripple
- filter shape is closer to the ideal response as N increases, although the filter with order of $N \to \infty$ is not realizable.

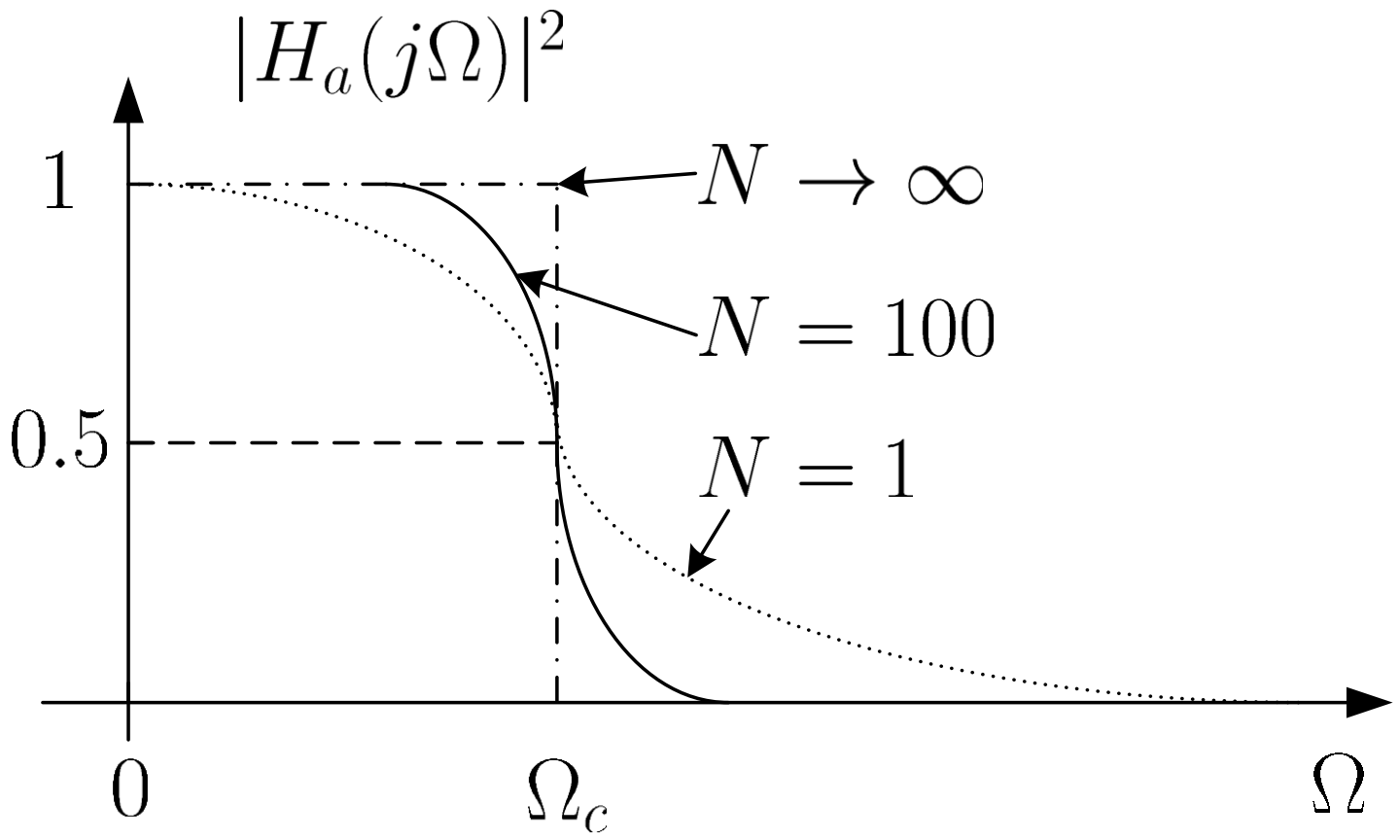


Fig.11.3: Magnitude square responses of Butterworth lowpass filter

To determine $H_a(s)$, we first make use of its relationship with $H_a(j\Omega)$:

$$H_a(s)\big|_{s=j\Omega} = H_a(j\Omega) \tag{11.9}$$

From (11.8)-(11.9), we obtain:

$$H_a(s)H_a(-s) = |H_a(j\Omega)|^2\Big|_{\Omega=s/j} = \frac{(j\Omega_c)^{2N}}{s^{2N} + (j\Omega_c)^{2N}} \tag{11.10}$$

The $2N$ poles of $H_a(s)H_a(-s)$, denoted by c_k, $k = 0, 1, \cdots, 2N-1$, are given as:

$$c_k = \begin{cases} \Omega_c e^{jk\pi/N}, & \text{odd } N \\ \Omega_c e^{jk\pi/N} \cdot e^{j\pi/(2N)}, & \text{even } N \end{cases} \tag{11.11}$$

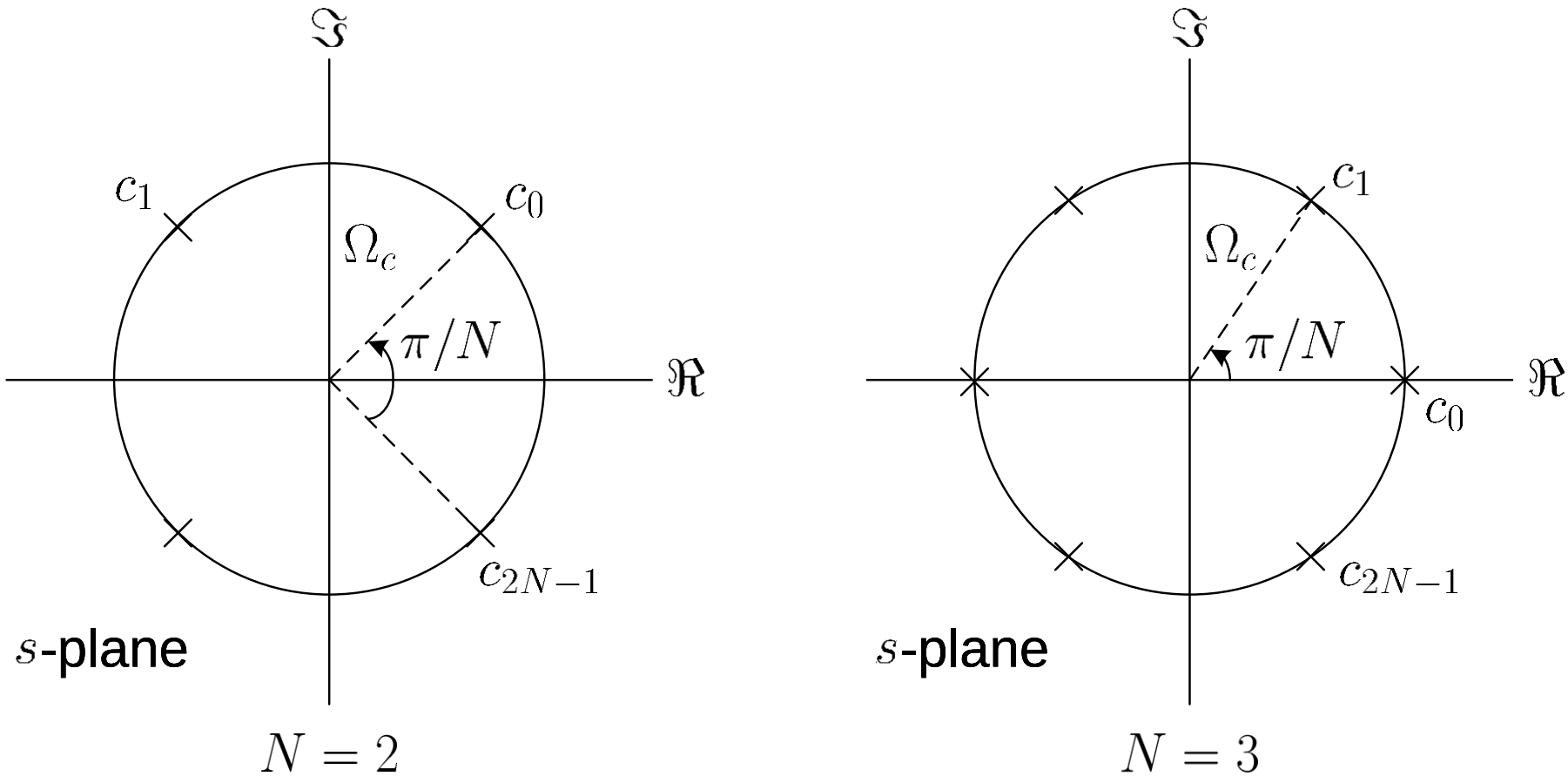


Fig.11.4: Poles of Butterworth lowpass filter

- $\{c_k\}$ are uniformly distributed on a circle of radius Ω_c with angular spacing of π/N in the s-plane

- poles are symmetrically located with respect to the imaginary axis

- there are two real-valued poles when N is odd

To extract $H_a(s)$ from (11.10), we utilize the knowledge that all poles of a stable and causal analog filter should be on the left half of the s-plane. As a result, $H_a(s)$ is:

$$H_a(s) = \frac{\Omega_c^N}{\prod\limits_{\Re\{c_k\}<0} (s - c_k)} \tag{11.12}$$

Example 11.1

The magnitude square response of a Butterworth lowpass filter has the form of:

$$|H_a(j\Omega)|^2 = \frac{1}{1 + 0.000001\Omega^6}$$

Determine the filter transfer function $H_a(s)$.

Expressing $|H_a(j\Omega)|^2$ as:

$$|H_a(j\Omega)|^2 = \frac{1}{1 + \left(\frac{\Omega}{10}\right)^{2\cdot 3}}$$

From (11.8), $\Omega_c = 10$ and $N = 3$

From (11.11):

$$c_k = 10e^{jk\pi/3}, \quad k = 2, 3, 4$$

Finally, we apply (11.12) to obtain:

$$\begin{aligned} H_a(s) &= \frac{\Omega_c^3}{(s - c_2)(s - c_3)(s - c_4)} \\ &= \frac{1000}{(s - 10e^{j2\pi/3})(s + 10)(s - 10e^{j4\pi/3})} \\ &= \frac{1000}{(s + 10)(s^2 + 10s + 100)} \end{aligned}$$

To find Ω_c and N given the passband and stopband requirements in terms of Ω_p, Ω_s, R_p and A_s, we exploit (11.4)-(11.5) together with (11.6)-(11.7) to obtain

$$-10\log_{10}\left(\frac{1}{1+\left(\frac{\Omega_p}{\Omega_c}\right)^{2N}}\right) = R_p \tag{11.13}$$

and

$$-10\log_{10}\left(\frac{1}{1+\left(\frac{\Omega_s}{\Omega_c}\right)^{2N}}\right) = A_s \tag{11.14}$$

Solving (11.13)-(11.14) and noting that N should be an integer, we get

$$N = \left\lceil \frac{\log_{10}\left[\left(10^{R_p/10}-1\right)/\left(10^{A_s/10}-1\right)\right]}{2\log_{10}(\Omega_p/\Omega_s)} \right\rceil \qquad (11.15)$$

where $\lceil u \rceil$ rounds up u to the nearest integer.

The Ω_c is then obtained from (11.13) or (11.14) so that the specification can be exactly met at Ω_p or Ω_s, respectively

From (11.13), Ω_c is computed as:

$$\Omega_c = \frac{\Omega_p}{\left(10^{R_p/10}-1\right)^{1/(2N)}} \qquad (11.16)$$

From (11.14), Ω_c is computed as:

$$\Omega_c = \frac{\Omega_s}{\left(10^{A_s/10} - 1\right)^{1/(2N)}} \tag{11.17}$$

As a result, the admissible range of Ω_c is:

$$\Omega_c \in \left[\frac{\Omega_p}{\left(10^{R_p/10} - 1\right)^{1/(2N)}}, \frac{\Omega_s}{\left(10^{A_s/10} - 1\right)^{1/(2N)}}\right] \tag{11.18}$$

Example 11.2

Determine the transfer function of a Butterworth lowpass filter whose magnitude requirements are $\Omega_p = 4\pi \ \text{rads}^{-1}$, $\Omega_s = 6\pi \ \text{rads}^{-1}$, $R_p = 8$ dB and $A_s = 16$ dB.

Employing (11.15) yields:

$$N = \left\lceil \frac{\log_{10}\left[\left(10^{8/10}-1\right)/\left(10^{16/10}-1\right)\right]}{2\log_{10}(4\pi/(6\pi))} \right\rceil = \lceil 2.45 \rceil = 3$$

Putting $N = 3$ in (11.18), the cutoff frequency is:

$$\Omega_c \in \left[\frac{4\pi}{\left(10^{8/10}-1\right)^{1/(2\cdot 3)}}, \frac{6\pi}{\left(10^{16/10}-1\right)^{1/(2\cdot 3)}} \right] = [9.5141, 10.2441]$$

For simplicity, we select $\Omega_c = 10$. Using Example 11.1, the filter transfer function $H_a(s)$ is:

$$H_a(s) = \frac{1000}{(s+10)(s^2+10s+100)} = \frac{1000}{s^3+20s^2+200s+1000}$$

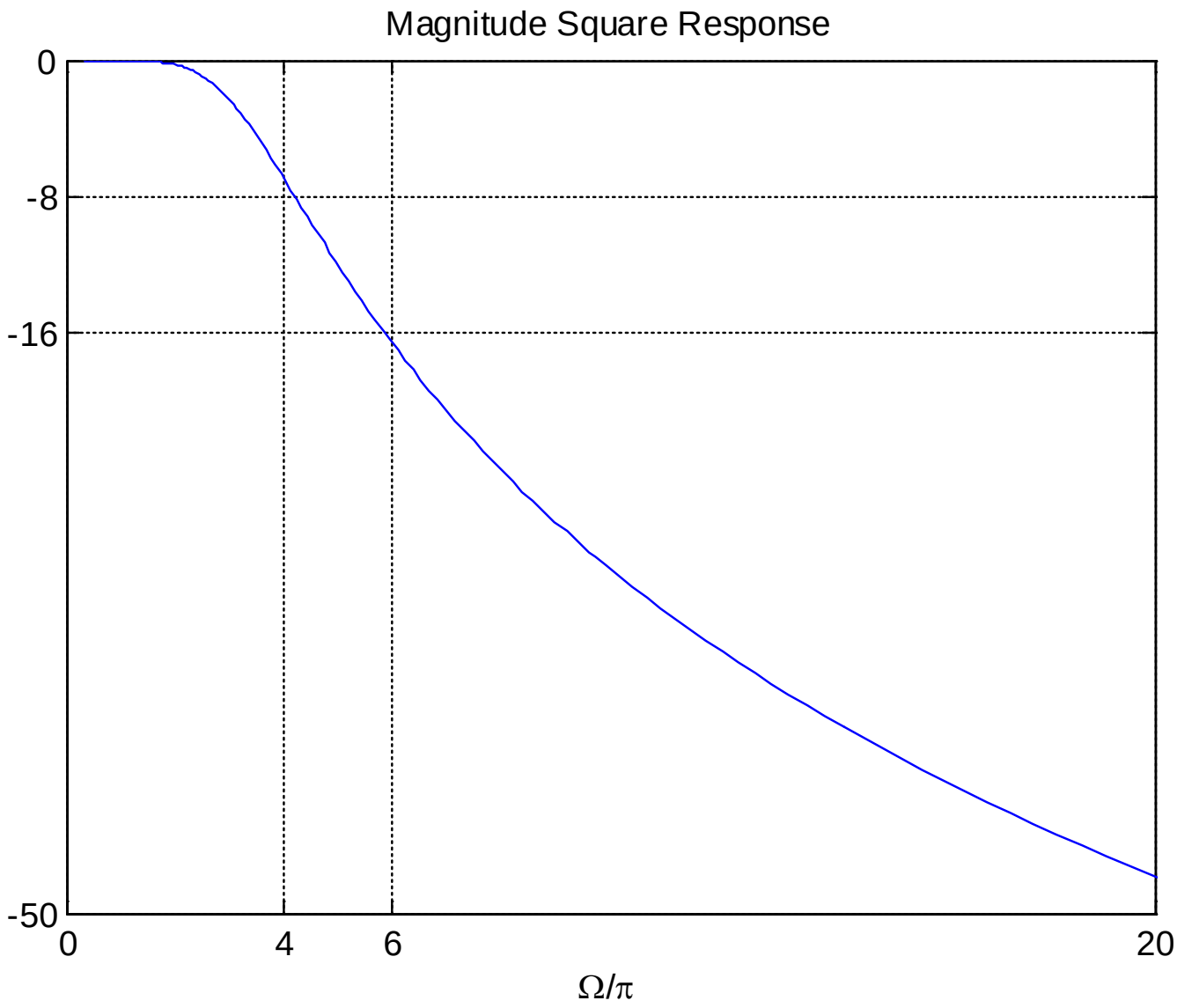


Fig.11.5: Magnitude square response of Butterworth lowpass filter

The MATLAB program is provided as `ex11_2.m` where the command `freqs`, which is analogous to `freqz`, is used to plot $|H_a(j\Omega)|^2$

Analog-to-Digital Filter Transformation

Typical methods include impulse invariance, bilinear transformation, backward difference approximation and matched-z transformation

Their common feature is that a stable analog filter will transform to a stable system with transfer function $H_{LP}(z)$.

Left half of s-plane maps into inside of unit circle in z-plane

Each has its pros and cons and thus optimal transformation does not exist

- Impulse Invariance

The idea is simply to sample impulse response of the analog filter $h_a(t)$ to obtain the digital lowpass filter impulse response $h_{LP}[n]$

The relationship between $h_{LP}[n]$ and $h_a(t)$ is

$$h_{LP}[n] = T \cdot h_a(nT), \quad n = \cdots -1, 0, 1, 2, \cdots \tag{11.19}$$

where T is the sampling interval

Why there is a scaling of *T* ?

With the use of (4.5) and (5.3)-(5.4), $H_{LP}(e^{j\omega})$ is:

$$H_{LP}(e^{j\omega}) = \sum_{k=-\infty}^{\infty} H_a\left(j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right) \qquad (11.20)$$

where the analog and digital frequencies are related as:

$$\omega = \Omega T \qquad (11.21)$$

The impulse response of the resultant IIR filter is similar to that of the analog filter

Aliasing due to the overlapping of $\{H_a\left(j\left(\omega - 2\pi k\right)/T\right)\}$ which are not bandlimited. However, $H_a(j\Omega)$ corresponds to a lowpass filter and thus aliasing effect is negligibly small.

To derive the IIR filter transfer function $H_{LP}(z)$ from $H_a(s)$, we first obtain the partial fraction expansion:

$$H_a(s) = \sum_{k=1}^{N} \frac{A_k}{s - s_k} \tag{11.22}$$

where $\{s_k\}$ are the poles on the left half of the s-plane

The inverse Laplace transform of (11.22) is given as:

$$h_a(t) = \begin{cases} \sum_{k=1}^{N} A_k e^{s_k t}, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{11.23}$$

Substituting (11.23) into (11.19), we have:

$$h_{LP}[n] = \sum_{k=1}^{N} TA_k e^{s_k nT} u[n] \tag{11.24}$$

The z transform of $h_{LP}[n]$ is:

$$H_{LP}(z) = \sum_{k=1}^{N} \frac{TA_k}{1 - e^{s_k T} z^{-1}} \tag{11.25}$$

Comparing (11.22) and (11.25), it is seen that a pole of $s = s_k$ in the s-plane transforms to a pole at $z = e^{s_k T}$ in the z-plane:

$$z = e^{sT} \tag{11.26}$$

Expressing $s = \sigma + j\Omega$:

$$z = e^{\sigma T} \cdot e^{j\Omega T} = e^{\sigma T} \cdot e^{j(\Omega + 2\pi k/T)T} \qquad (11.27)$$

where k is any integer, indicating a many-to-one mapping

Each infinite horizontal strip of width $2\pi/T$ maps into the entire z-plane

$\sigma = 0$ maps to $|z| = 1$, that is, $j\Omega$ axis in the s-plane transforms to the unit circle in the z-plane

$\sigma < 0$ maps to $|z| < 1$, stable $H_a(s)$ produces stable $H_{LP}(z)$

$\sigma > 0$ maps to $|z| > 1$, right half of the s-plane maps into the outside of the unit circle in the z-plane

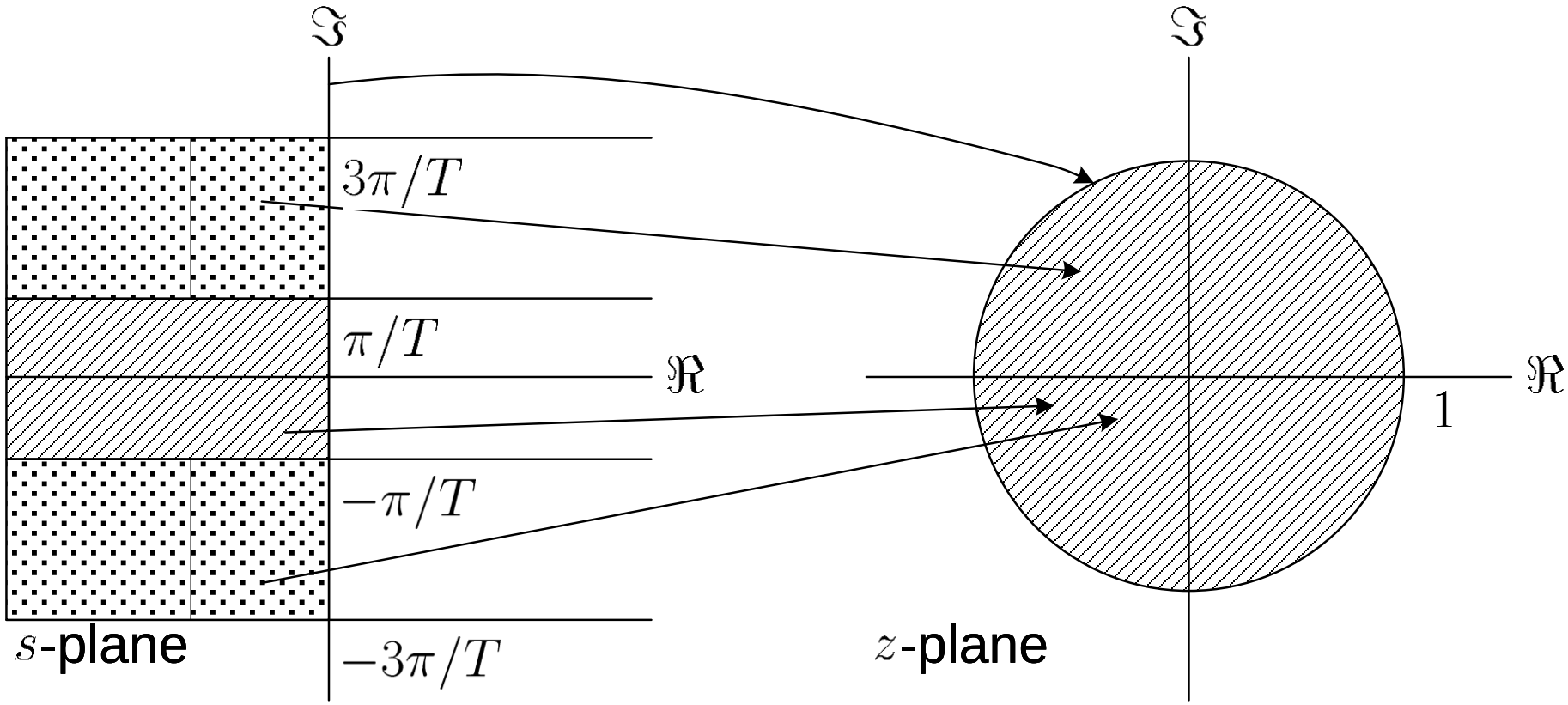


Fig.11.6: Mapping between s and z in impulse invariance method

Given the magnitude square response specifications of $H_{LP}(e^{j\omega})$ in terms of ω_p, ω_s, R_p and A_s, the design procedure for $H_{LP}(z)$ based on the impulse invariance method is summarized as the following steps:

(i) Select a value for the sampling interval T and then compute the passband and stopband frequencies for the analog lowpass filter according to $\Omega_p = \omega_p/T$ and $\Omega_s = \omega_s/T$

(ii) Design the analog Butterworth filter with transfer function $H_a(s)$ according to Ω_p, Ω_s, R_p and A_s

(iii) Perform partial fraction expansion on $H_a(s)$ as in (11.22)

(iv) Obtain $H_{LP}(z)$ using (11.25)

Example 11.3

The transfer function of an analog filter has the form of

$$H_a(s) = \frac{2s}{s^2 + 6s + 8}$$

Use impulse invariance method with sampling interval $T = 1$ to transform $H_a(s)$ to a digital filter transfer function $H(z)$.

Performing partial fraction expansion on $H_a(s)$:

$$H_a(s) = \frac{-2}{s+2} + \frac{4}{s+4}$$

Applying (11.25) with $T = 1$ yields

$$H(z) = \frac{-2}{1 - e^{-2}z^{-1}} + \frac{4}{1 - e^{-4}z^{-1}} = \frac{2 - 0.5047z^{-1}}{1 - 0.1537z^{-1} + 0.0025z^{-2}}$$

Example 11.4

Determine the transfer function $H_{LP}(z)$ of a digital lowpass filter whose magnitude requirements are $\omega_p = 0.4\pi$, $\omega_s = 0.6\pi$, $R_p = 8$ dB and $A_s = 16$ dB. Use the Butterworth lowpass filter and impulse invariance method in the design.

Selecting the sampling interval as $T = 0.1$, the analog frequency parameters are computed as:

$$\Omega_p = \frac{\omega_p}{T} = 4\pi$$

and

$$\Omega_s = \frac{\omega_s}{T} = 6\pi$$

Using Example 11.2, a Butterworth filter which meets the magnitude requirements are:

$$H_a(s) = \frac{1000}{(s+10)(s^2+10s+100)} = \frac{1000}{s^3+20s^2+200s+1000}$$

Performing partial fraction expansion on $H_a(s)$ with the use of the MATLAB command `residue`, we get

$$H_a(s) = \frac{10}{s+10} + \frac{-5-j2.8868}{s+5-j8.6603} + \frac{-5+j2.8868}{s+5+j8.6603}$$

Applying (11.25) with $T = 0.1$ yields

$$
\begin{aligned}
H_{LP}(z) &= \frac{0.1 \cdot 10}{1 - e^{-10 \cdot 0.1} z^{-1}} + \frac{0.1 \cdot (-5 - j2.8868)}{1 - e^{(-5+j8.6603) \cdot 0.1} z^{-1}} + \frac{0.1 \cdot (-5 + j2.8868)}{1 - e^{(-5-j8.6603) \cdot 0.1} z^{-1}} \\
&= \frac{1}{1 - 0.3679 z^{-1}} + \frac{-0.5 - j0.2887}{1 - (0.3929 + j0.4620) z^{-1}} \\
&\quad + \frac{-0.5 + j0.2887}{1 - (0.3929 - j0.4620) z^{-1}} \\
&= \frac{1}{1 - 0.3679 z^{-1}} + \frac{-1 + 0.6597 z^{-1}}{1 - 0.7859 z^{-1} + 0.3679 z^{-2}} \\
&= \frac{0.2417 z^{-1} + 0.1262 z^{-2}}{1 - 1.1538 z^{-1} + 0.6570 z^{-2} - 0.1354 z^{-3}}
\end{aligned}
$$

The MATLAB program is provided as `ex11_4.m`.

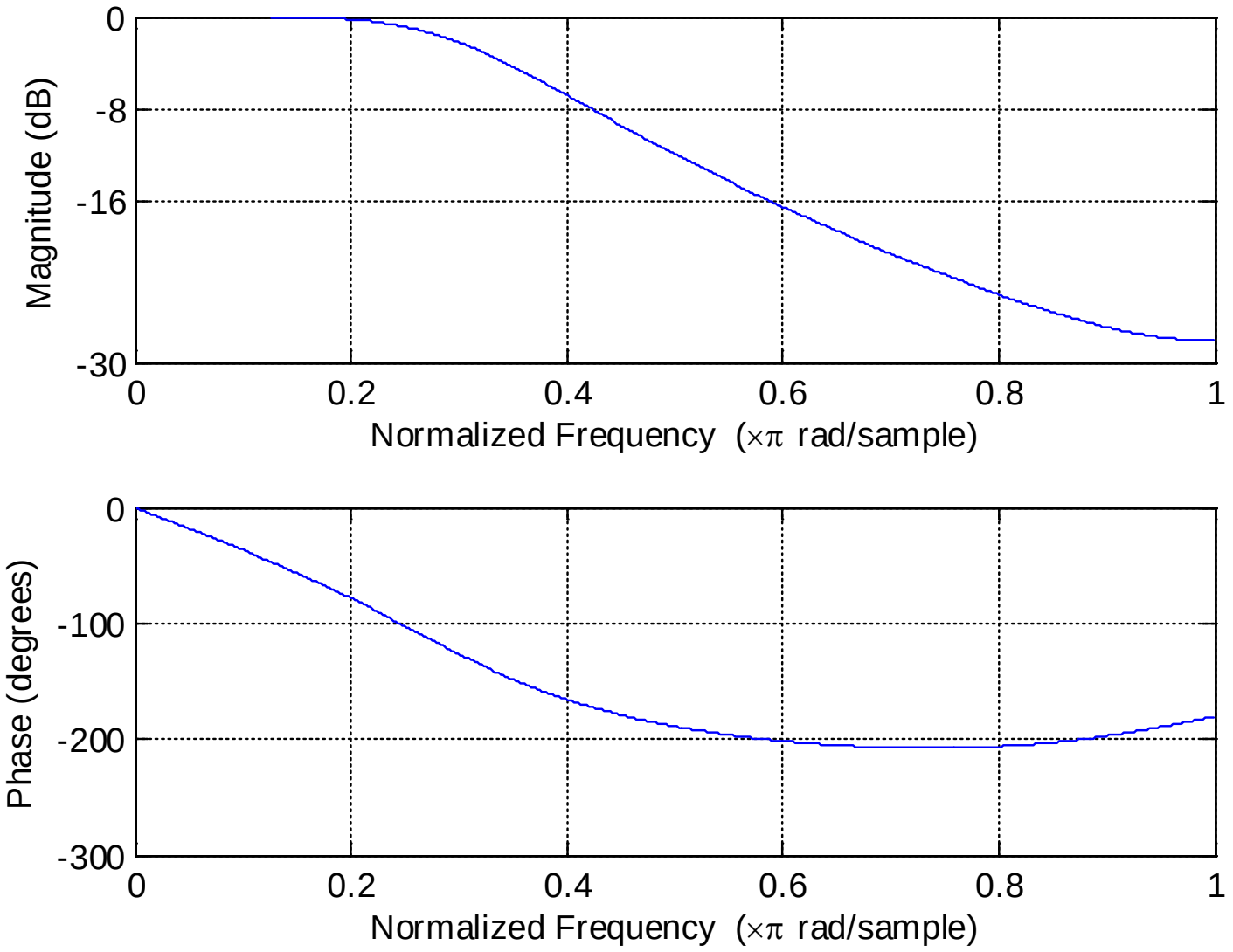


Fig.11.7: Magnitude and phase responses based on impulse invariance

- Bilinear Transformation

It is a conformal mapping that maps the $j\Omega$ axis of the s-plane into the unit circle of the z-plane only once, implying there is no aliasing problem as in the impulse invariance method

It is a one-to-one mapping

The relationship between s and z is:

$$s = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \Leftrightarrow z = \frac{1 + sT/2}{1 - sT/2} \tag{11.28}$$

Employing $s = \sigma + j\Omega$, z can be expressed as:

$$z = \frac{(1 + \sigma T/2) + j\Omega T/2}{(1 - \sigma T/2) - j\Omega T/2} \tag{11.29}$$

$\sigma = 0$ maps to $|z| = 1$, that is, $j\Omega$ axis in the s-plane transforms to the unit circle in the z-plane

$\sigma < 0$ maps to $|z| < 1$, stable $H_a(s)$ produces a stable $H_{LP}(z)$

$\sigma > 0$ maps to $|z| > 1$, right half of the s-plane maps into the outside of the unit circle in the z-plane

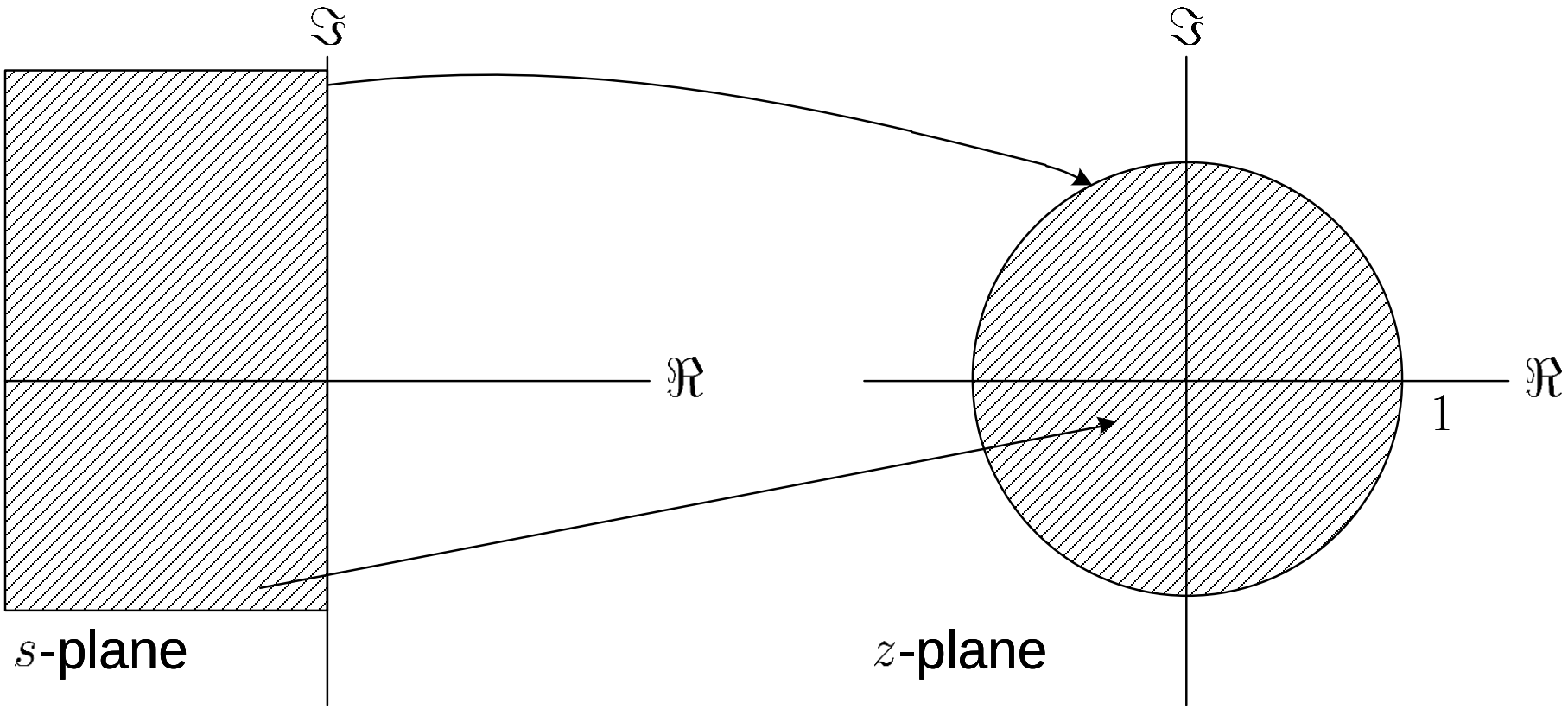


Fig.11.8: Mapping between s and z in bilinear transformation

Although aliasing is avoided, the drawback of the bilinear transformation is that there is no linear relationship between ω and Ω

Putting $z = e^{j\omega}$ and $s = j\Omega$ in (11.28), ω and Ω are related as:

$$j\Omega = \frac{2}{T} \cdot \frac{1 - e^{-j\omega}}{1 + e^{-j\omega}} = \frac{2}{T} \cdot \frac{e^{j\omega/2} - e^{-j\omega/2}}{e^{j\omega/2} + e^{-j\omega/2}} = j\frac{2}{T} \tan\left(\frac{\omega}{2}\right)$$

$$\Rightarrow \Omega = \frac{2}{T} \tan\left(\frac{\omega}{2}\right) \Leftrightarrow \omega = 2\tan^{-1}\left(\frac{\Omega T}{2}\right) \qquad (11.30)$$

Given the magnitude square response specifications of $H_{LP}(e^{j\omega})$ in terms of ω_p, ω_s, R_p and A_s, the design procedure for $H_{LP}(z)$ based on the bilinear transformation is summarized as the following steps:

(i) Select a value for T and then compute the passband and stopband frequencies for the analog lowpass filter according $\Omega_p = (2/T)\tan(\omega_p/2)$ and $\Omega_s = (2/T)\tan(\omega_s/2)$

(ii) Design the analog Butterworth filter with transfer function $H_a(s)$ according to Ω_p, Ω_s, R_p and A_s.

(iii) Obtain $H_{LP}(z)$ from $H_a(s)$ using the substitution of (11.28).

Example 11.5

The transfer function of an analog filter has the form of

$$H_a(s) = \frac{2s}{s^2 + 6s + 8}$$

Use the bilinear transformation with $T = 1$ to transform $H_a(s)$ to a digital filter with transfer function $H(z)$.

Applying (11.28) with $T = 1$ yields

$$H(z) = \frac{2 \cdot 2\dfrac{1 - z^{-1}}{1 + z^{-1}}}{\left(2\dfrac{1 - z^{-1}}{1 + z^{-1}}\right)^2 + 6 \cdot 2\dfrac{1 - z^{-1}}{1 + z^{-1}} + 8} = \frac{4 - 4z^{-2}}{15 + 14z^{-1} + 9z^{-2}}$$

Example 11.6

Determine the transfer function $H_{LP}(z)$ of a digital lowpass filter whose magnitude requirements are $\omega_p = 0.4\pi$, $\omega_s = 0.6\pi$, $R_p = 8$ dB and $A_s = 16$ dB. Use the Butterworth lowpass filter and bilinear transformation in the design.

Selecting $T = 0.1$, the analog frequency parameters are computed according to (11.30) as:

$$\Omega_p = \frac{2}{T}\tan\left(\frac{\omega_p}{2}\right) = 14.5309$$

and

$$\Omega_s = \frac{2}{T}\tan\left(\frac{\omega_s}{2}\right) = 27.5276$$

Employing (11.15) yields:

$$N = \left\lceil \frac{\log_{10}\left[\left(10^{8/10}-1\right)/\left(10^{16/10}-1\right)\right]}{2\log_{10}(14.5309/27.5276)} \right\rceil = \lceil 1.56 \rceil = 2$$

Putting $N = 2$ in (11.18), the cutoff frequency is:

$$\Omega_c \in \left[\frac{14.5309}{\left(10^{8/10}-1\right)^{1/(2\cdot 2)}}, \frac{27.5276}{\left(10^{16/10}-1\right)^{1/(2\cdot 2)}} \right] = [9.5725, 11.0289]$$

For simplicity, $\Omega_c = 10$ is employed.

Following (11.11)-(11.12):

$$H_a(s) = \frac{100}{s^2 + 14.1421s + 100}$$

Finally, we use (11.28) with $T = 0.1$ to yield

$$\begin{aligned} H_{LP}(z) &= \frac{100}{\left(\frac{2}{0.1} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}\right)^2 + 14.1421 \cdot \frac{2}{0.1} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} + 100} \\ &= \frac{1 + 2z^{-1} + z^{-2}}{7.8284 - 6z^{-1} + 2.1716z^{-2}} \end{aligned}$$

The MATLAB program is provided as `ex11_6.m`.

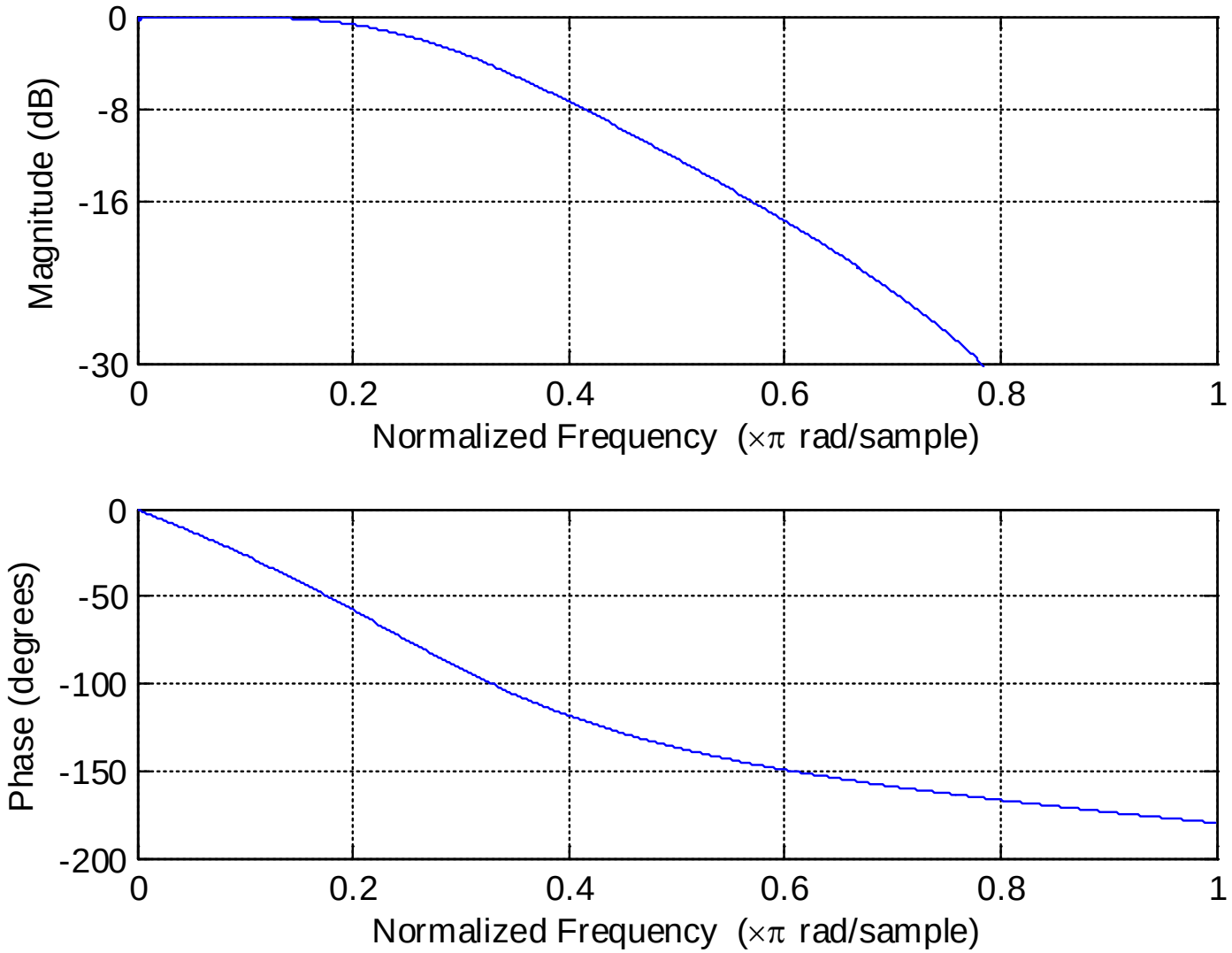


Fig.11.9: Magnitude and phase responses based on bilinear transformation

Frequency Band Transformation

The operations are similar to that of the bilinear transformation but now the mapping is performed only in the z-plane:

$$z_o^{-1} = T(z^{-1}) \tag{11.31}$$

where z_o and z correspond to the lowpass and resultant filters, respectively, and T denotes the transformation operator.

To ensure the transformed filter to be stable and causal, the unit circle and inside of the z_o-plane should map into those of the z-plane, respectively.

Filter Type	Transformation Operator	Design Parameter
Lowpass	$z_o^{-1} = \dfrac{z^{-1} - \alpha}{1 - \alpha z^{-1}}$	$\alpha = \dfrac{\sin\left(\dfrac{\omega_{c_o} - \omega_c}{2}\right)}{\sin\left(\dfrac{\omega_{c_o} + \omega_c}{2}\right)}$
Highpass	$z_o^{-1} = -\dfrac{z^{-1} + \alpha}{1 + \alpha z^{-1}}$	$\alpha = -\dfrac{\cos\left(\dfrac{\omega_{c_o} + \omega_c}{2}\right)}{\cos\left(\dfrac{\omega_{c_o} - \omega_c}{2}\right)}$
Bandpass	$z_o^{-1} = \dfrac{z^{-2} - \dfrac{2\alpha\beta}{\beta+1} z^{-1} + \dfrac{\beta-1}{\beta+1}}{\dfrac{\beta-1}{\beta+1} z^{-2} - \dfrac{2\alpha\beta}{\beta+1} z^{-1} + 1}$	$\alpha = \dfrac{\cos\left(\dfrac{\omega_{c_2} + \omega_{c_1}}{2}\right)}{\cos\left(\dfrac{\omega_{c_2} - \omega_{c_1}}{2}\right)}$ $\beta = \cot\left(\dfrac{\omega_{c_2} - \omega_{c_1}}{2}\right) \tan\left(\dfrac{\omega_{c_o}}{2}\right)$

Bandstop	$z_o^{-1} = \dfrac{z^{-2} - \dfrac{2\alpha}{1+\beta}z^{-1} + \dfrac{1-\beta}{1+\beta}}{\dfrac{1-\beta}{1+\beta}z^{-2} - \dfrac{2\alpha}{1+\beta}z^{-1} + 1}$	$\alpha = \dfrac{\cos\left(\dfrac{\omega_{c_2} + \omega_{c_1}}{2}\right)}{\cos\left(\dfrac{\omega_{c_2} - \omega_{c_1}}{2}\right)}$ $\beta = \cot\left(\dfrac{\omega_{c_2} - \omega_{c_1}}{2}\right)\tan\left(\dfrac{\omega_{c_o}}{2}\right)$

Table 11.1: Frequency band transformation operators

Example 11.7

Determine the transfer function $H(z)$ of a digital highpass filter whose magnitude requirements are $\omega_p = 0.6\pi$, $\omega_s = 0.4\pi$, $R_p = 8$ dB and $A_s = 16$ dB. Use the Butterworth lowpass filter and bilinear transformation in the design.

Using Example 11.6, the corresponding lowpass filter transfer function $H_{LP}(z_o)$ is:

$$H_{LP}(z_o) = \frac{1 + 2z_o^{-1} + z_o^{-2}}{7.8284 - 6z_o^{-1} + 2.1716z_o^{-2}}$$

Assigning the cutoff frequencies as the midpoints between the passband and stopband frequencies, we have

$$\omega_{c_o} = \omega_c = \frac{0.4\pi + 0.6\pi}{2} = 0.5\pi$$

With the use of Table 11.1, the corresponding value of α is:

$$\alpha = -\frac{\cos\left(\frac{\omega_{c_o} + \omega_c}{2}\right)}{\cos\left(\frac{\omega_{c_o} - \omega_c}{2}\right)} = -\frac{\cos(0.5\pi)}{\cos(0)} = 0$$

which gives the transformation operator:

$$z_o^{-1} = -\frac{z^{-1} + 0}{1 + 0 \cdot z^{-1}} = -z^{-1}$$

As a result, the digital highpass filter transfer function is:

$$H(z) = H_{LP}(z_o)|_{z_o^{-1} = -z^{-1}} = \frac{1 - 2z^{-1} + z^{-2}}{7.8284 + 6z^{-1} + 2.1716z^{-2}}$$

The MATLAB program is provided as `ex11_7.m`.

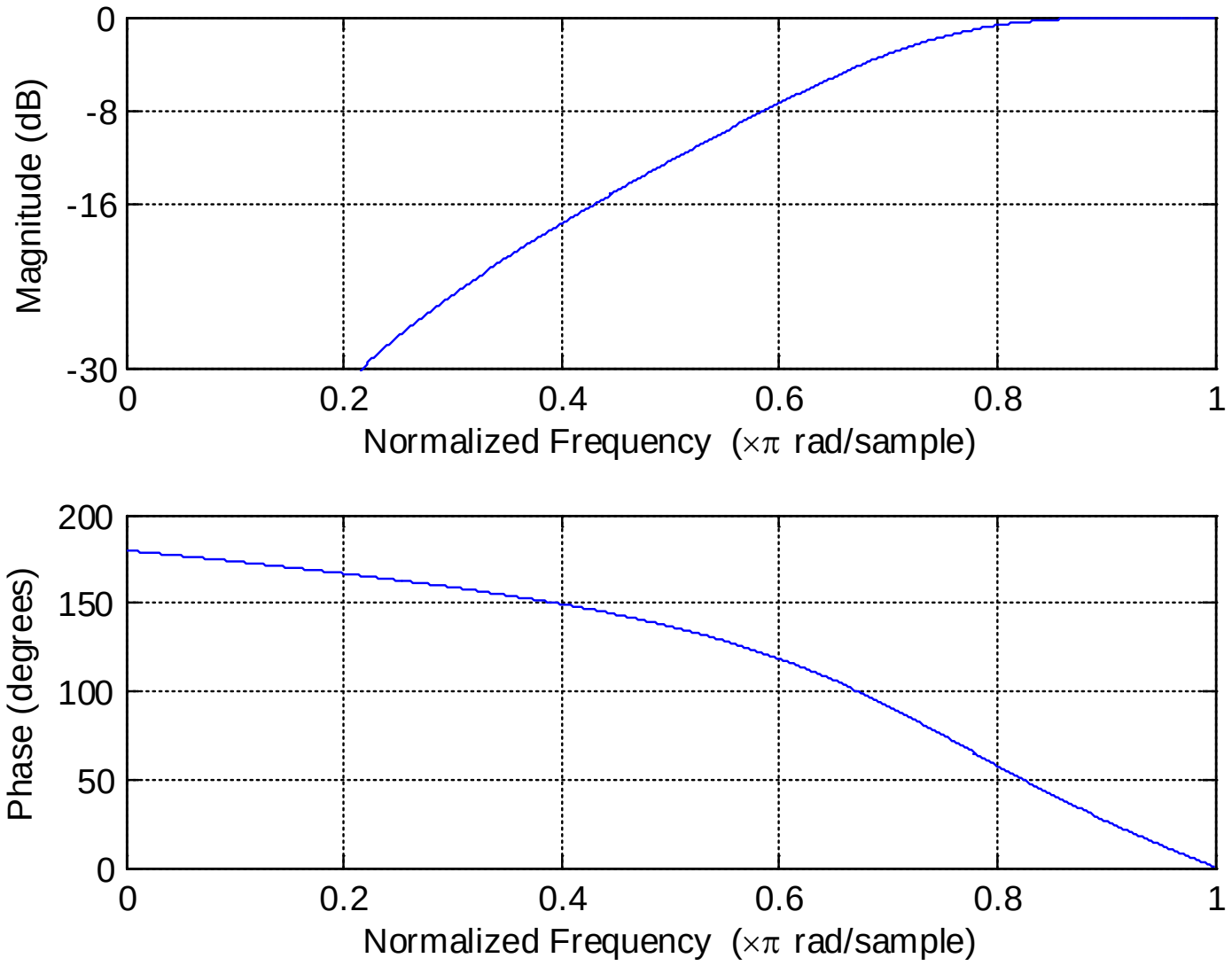


Fig.11.10: Magnitude and phase responses based on frequency band transformation

Chapter 12: Tools and Applications

Chapter Intended Learning Outcomes:

(i) Understanding the basic concepts of signal modeling, correlation, maximum likelihood estimation, least squares and iterative numerical methods

(ii) Ability to solve simple signal processing problems with the use of investigated tools

Tools

- Signal Modeling

1. White Noise

A random sequence $q[n]$ is white if

$$E\{q[n]\} = 0 \tag{12.1}$$

and

$$E\{q[n]q[m]\} = \begin{cases} \sigma_q^2, & m = n \\ 0, & \text{otherwise} \end{cases} \tag{12.2}$$

where E denotes the expectation operator and σ_q^2 is the power or variance of $q[n]$

The mean value of $q[n]$ is zero for all n

There is no correlation between $q[m]$ and $q[n]$ if $m \neq n$

The probability density function (PDF) is not specified, which means that $q[n]$ can be a uniform or Gaussian sequence

The MATLAB command to generate a white uniform number with $\sigma_q^2 = 1/12$ is `rand-0.5` and that of a white Gaussian random variable with $\sigma_q^2 = 1$ is `randn`

$q[n]$ is a wideband signal and has a flat spectrum in the mean sense

2. Sinusoidal Model

Many real-world signals in the areas such as radar, sonar and communications, can be well described as:

$$x[n] = \sum_{k=1}^{L} \alpha_k \cos(\omega_k n + \phi_k) + q[n] \qquad (12.3)$$

where α_k , $\omega_k \in (0, \pi)$ and $\phi_k \in [0, 2\pi)$ denote the amplitude, frequency and phase of the k-th sinusoid, respectively

3. Autoregressive Model

Autoregressive (AR) process has been used to represent many real-world signals such as speech and electroencephalography (EEG)

A M-th order AR process is:

$$x[n] = \sum_{k=1}^{M} a_k x[n-k] + q[n] \qquad (12.4)$$

where $\{a_k\}$ are the AR parameters and $q[n]$ is an additive white noise

Taking the z transform on (12.4), we see that the transfer function $H(z)$ of the system with input $q[n]$ and output $x[n]$ is:

$$H(z) = \frac{X(z)}{Q(z)} = \frac{1}{1 - \sum_{k=1}^{M} a_k z^{-k}} \qquad (12.5)$$

4. Moving Average Model

A moving average (MA) process of order N is:

$$x[n] = \sum_{k=0}^{N} b_k q[n-k] \tag{12.6}$$

where $q[n]$ is an additive white noise and $\{b_k\}$ are called the MA parameters

Taking the z transform on (12.6), we see that the transfer function $H(z)$ of the system with input $q[n]$ and output $x[n]$ is:

$$H(z) = \frac{X(z)}{Q(z)} = \sum_{k=0}^{N} b_k z^{-k} \tag{12.7}$$

and its finite-duration impulse response $h[n]$ is

$$h[n] = \begin{cases} b_n, & 0 \leq n \leq N \\ 0, & \text{otherwise} \end{cases} \tag{12.8}$$

5. Autoregressive Moving Average Model

The autoregressive moving average (ARMA) process generalizes the AR and MA models:

$$x[n] = \sum_{k=1}^{M} a_k x[n-k] + \sum_{k=0}^{N} b_k q[n-k] \tag{12.9}$$

where $\{a_k\}$ and $\{b_k\}$ are the ARMA parameters and $q[n]$ is an additive white noise

The system transfer function is:

$$H(z) = \frac{\sum_{k=0}^{N} b_k z^{-k}}{1 - \sum_{k=1}^{M} a_k z^{-k}} \qquad (12.10)$$

- Correlation

It provides a simple and useful measure for determining the similarity between two sequences

Autocorrelation measures the similarity of the same signal $x[n]$ at different time indices and it is defined as:

$$r_{xx}[l] = E\{x[n]x[n-l]\} \qquad (12.11)$$

where l is called the lag

If $|r_{xx}[l]|$ is large, $x[n]$ is similar to $x[n-l]$ while they are not similar for a small $|r_{xx}[l]|$

Two important properties for $r_{xx}[l]$ are:

$$r_{xx}[0] \geq |r_{xx}[l]|, \quad l \neq 0 \tag{12.12}$$

and

$$r_{xx}[l] = r_{xx}[-l] \tag{12.13}$$

When ergodicity holds, that is, the mean value is equal to the time average, $r_{xx}[l]$ is also expressed as:

$$r_{xx}[l] = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x[n]x[n-l] \tag{12.14}$$

An estimate of $r_{xx}[l]$ using a finite-length $x[n]$ for $0 \leq n \leq N-1$, denoted by $\hat{r}_{xx}[l]$, is:

$$\hat{r}_{xx}[l] = \hat{r}_{xx}[-l] = \frac{1}{N-l} \sum_{n=l}^{N-1} x[n]x[n-l], \quad 0 \leq l \leq N-1 \quad (12.15)$$

Cross-correlation is used for two different sequences $x[n]$ and $y[n]$:

$$r_{xy}[l] = E\{x[n]y[n-l]\} = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x[n]y[n-l] \quad (12.16)$$

and

$$\hat{r}_{xy}[l] = \frac{1}{N-l} \sum_{n=l}^{N-1} x[n]y[n-l], \quad 0 \leq l \leq N-1 \quad (12.17)$$

with $\hat{r}_{xy}[-l] = \hat{r}_{yx}[l]$, $l = 1, 2, \cdots, N-1$

Example 12.1

Use the MATLAB command `randn` to generate a zero-mean Gaussian sequence with length 10000 and power $\sigma_q^2 = 1$. Verify its whiteness.

We first use `q=randn(1,10000)` to generate the sequence and then use the MATLAB command `xcorr(q)` to compute $\hat{r}_{xx}[l]$

The MATLAB program is provided as `ex12_1.m`.

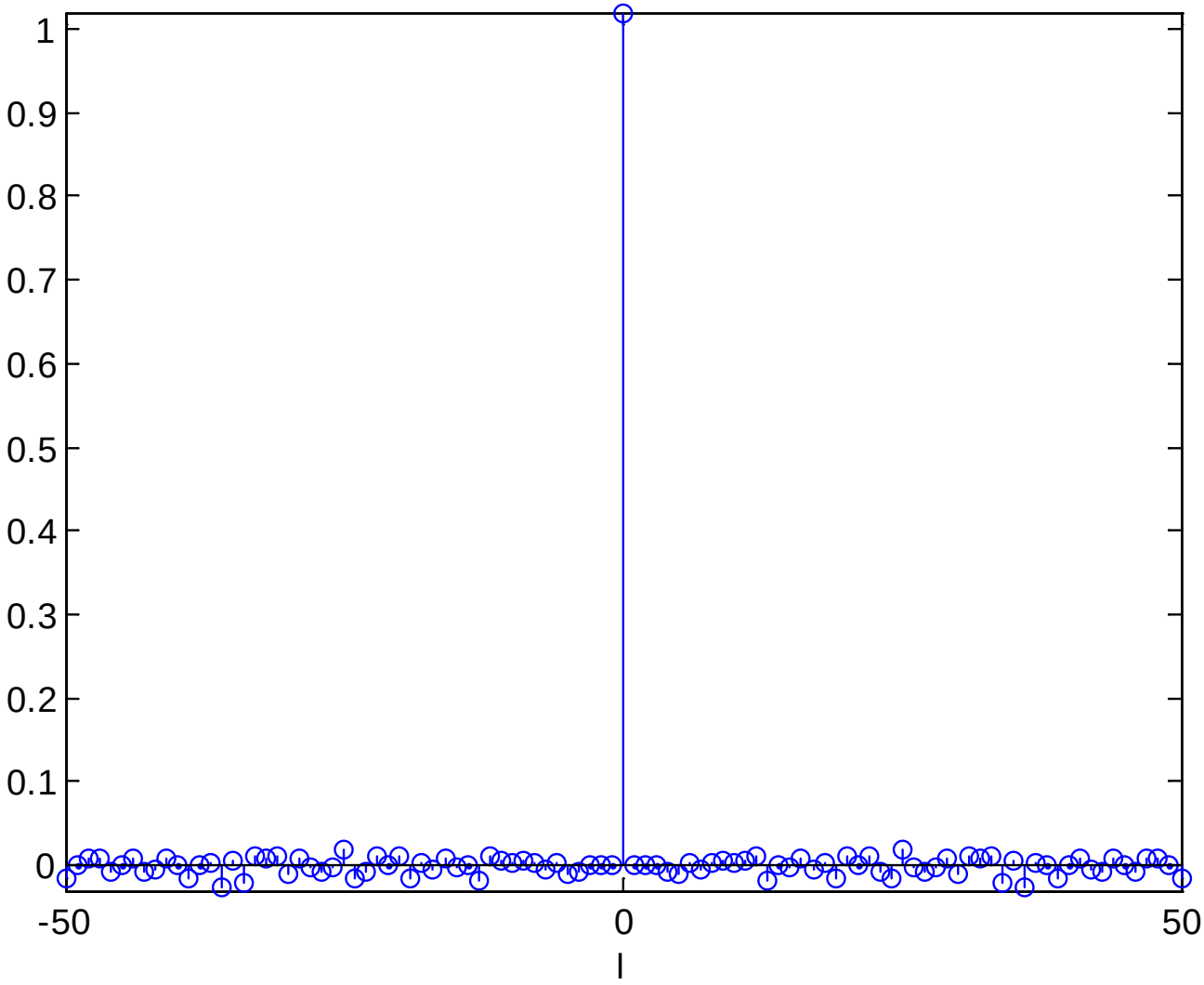


Fig. 12.1: Autocorrelation of white Gaussian sequence

- Maximum Likelihood Estimation

Suppose there is an observed sequence $x[n]$ which contains a deterministic signal in the presence of noise:

$$x[n] = f(\boldsymbol{\psi}, n) + q[n], \quad n = 0, 1, \cdots, N-1 \qquad (12.18)$$

where $f(\cdot)$ is the known function of the parameter vector $\boldsymbol{\psi}$ and $q[n]$ is the noise

The task is to find $\boldsymbol{\psi}$ from $x[n]$

The maximum likelihood approach aims to provide optimum estimation for $\boldsymbol{\psi}$ when the PDF of $q[n]$ is available

Assume $q[n]$ is zero-mean Gaussian variable with variance σ_q^2

Zero-mean implies that the mean value of $x[n]$ is:

$$E\{x[n]\} = f(\boldsymbol{\psi}, n) \tag{12.19}$$

The PDF of each scalar $x[n]$ parameterized by $\boldsymbol{\psi}$ is:

$$p(x[n]; \boldsymbol{\psi}) = \frac{1}{\sqrt{2\pi\sigma_q^2}} e^{-\frac{1}{2\sigma_q^2}\left(x[n] - f(\boldsymbol{\psi}, n)\right)^2} \tag{12.20}$$

Defining $\mathbf{x} = [x[0]\ x[1]\ \cdots\ x[N-1]]^T$, $\mathbf{q} = [q[0]\ q[1]\ \cdots\ q[N-1]]^T$ and $\mathbf{f}(\boldsymbol{\psi}) = [f(\boldsymbol{\psi}, 0)\ f(\boldsymbol{\psi}, 1)\ \cdots\ f(\boldsymbol{\psi}, N-1)]^T$, we extend (12.20) to $\mathbf{x}$:

$$p(\mathbf{x}; \boldsymbol{\psi}) = \frac{1}{(2\pi)^{N/2}|\mathbf{C}|^{1/2}} e^{-\frac{1}{2}\left(\mathbf{x} - \mathbf{f}(\boldsymbol{\psi})\right)^T \mathbf{C}^{-1}\left(\mathbf{x} - \mathbf{f}(\boldsymbol{\psi})\right)} \tag{12.21}$$

where $\mathbf{C}$ is called the covariance matrix:

$$\mathbf{C} = E\{(\mathbf{x} - \mathbf{f}(\boldsymbol{\psi}))(\mathbf{x} - \mathbf{f}(\boldsymbol{\psi}))^T\} = E\{\mathbf{q}\mathbf{q}^T\} \qquad (12.22)$$

The maximum likelihood estimate for $\boldsymbol{\psi}$ is

$$\hat{\boldsymbol{\psi}} = \arg\max_{\boldsymbol{\psi}} p(\mathbf{x}; \boldsymbol{\psi}) \qquad (12.23)$$

For white $q[n]$, according to (12.1)-(12.2), $\mathbf{C}$ is simply determined as:

$$\mathbf{C} = \sigma_q^2 \mathbf{I} \qquad (12.24)$$

where $\mathbf{I}$ is the identity matrix, resulting in $|\mathbf{C}| = \sigma_q^{2N}$ and $\mathbf{C}^{-1} = \sigma_q^{-2}\mathbf{I}$

As a result, the PDF of $\mathbf{x}$ is simplified as:

$$p(\mathbf{x};\boldsymbol{\psi}) = \frac{1}{(2\pi\sigma_q^2)^{N/2}} e^{-\frac{1}{2\sigma_q^2}\left(\mathbf{x}-\mathbf{f}(\boldsymbol{\psi})\right)^T\left(\mathbf{x}-\mathbf{f}(\boldsymbol{\psi})\right)} \qquad (12.25)$$

Maximizing $p(\mathbf{x};\boldsymbol{\psi})$ is equal to minimizing $\left(\mathbf{x}-\mathbf{f}(\boldsymbol{\psi})\right)^T\left(\mathbf{x}-\mathbf{f}(\boldsymbol{\psi})\right)$, the maximum likelihood estimate is also equal to:

$$\hat{\boldsymbol{\psi}} = \arg\min_{\boldsymbol{\psi}} \left[\left(\mathbf{x}-\mathbf{f}(\boldsymbol{\psi})\right)^T\left(\mathbf{x}-\mathbf{f}(\boldsymbol{\psi})\right)\right] \qquad (12.26)$$

- Least Squares

Compared with the maximum likelihood approach, it does not require the PDF information but its estimation performance may be inferior

The least squares estimate for $\boldsymbol{\psi}$ is:

$$\begin{aligned}\hat{\boldsymbol{\psi}} &= \arg\min_{\boldsymbol{\psi}} \left[(\mathbf{x} - \mathbf{f}(\boldsymbol{\psi}))^T (\mathbf{x} - \mathbf{f}(\boldsymbol{\psi})) \right] \\ &= \arg\min_{\boldsymbol{\psi}} \sum_{n=0}^{N-1} [x[n] - f(\boldsymbol{\psi}, n)]^2 \end{aligned} \quad (12.27)$$

That is, the maximum likelihood and least squares estimates are equivalent for white noise

Note that when $f(\boldsymbol{\psi}, n)$ is a linear function of $\boldsymbol{\psi}$, then $\hat{\boldsymbol{\psi}}$ can be simplified to a global closed-form solution

Example 12.2

Given N samples of $x[n]$ which has the form of:

$$x[n] = \alpha n + \beta + q[n], \quad n = 0, 1, \cdots, N-1$$

where α and β are the parameters of interest and $q[n]$ is an unknown zero-mean noise sequence. Determine the least squares estimates for α and β. This problem is also known as linear regression in statistics.

According to (12.27), the least squares cost function is:

$$J(\alpha, \beta) = \sum_{n=0}^{N-1} (x[n] - \alpha n - \beta)^2$$

which is quadratic with parameters α and β.

To find the least squares estimates, we differentiate $J(\alpha, \beta)$ with respect to α and β, and then set the resultant expressions to zero:

$$\left.\frac{\partial J(\alpha,\beta)}{\partial \alpha}\right|_{\alpha=\hat{\alpha},\beta=\hat{\beta}} = 2\sum_{n=0}^{N-1}\left(x[n]-\hat{\alpha}n-\hat{\beta}\right)(-n) = 0$$

$$\Rightarrow \sum_{n=0}^{N-1} nx[n] = \hat{\alpha}\sum_{n=0}^{N-1} n^2 + \hat{\beta}\sum_{n=0}^{N-1} n$$

$$\left.\frac{\partial J(\alpha,\beta)}{\partial \beta}\right|_{\alpha=\hat{\alpha},\beta=\hat{\beta}} = 2\sum_{n=0}^{N-1}\left(x[n]-\hat{\alpha}n-\hat{\beta}\right)(-1) = 0$$

$$\Rightarrow \sum_{n=0}^{N-1} x[n] = \hat{\alpha}\sum_{n=0}^{N-1} n + N\hat{\beta}$$

Hence:

$$\begin{bmatrix} \sum_{n=0}^{N-1} n^2 & \sum_{n=0}^{N-1} n \\ \sum_{n=0}^{N-1} n & N \end{bmatrix} \begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix} = \begin{bmatrix} \sum_{n=0}^{N-1} nx[n] \\ \sum_{n=0}^{N-1} x[n] \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix} = \begin{bmatrix} \sum_{n=0}^{N-1} n^2 & \sum_{n=0}^{N-1} n \\ \sum_{n=0}^{N-1} n & N \end{bmatrix}^{-1} \begin{bmatrix} \sum_{n=0}^{N-1} nx[n] \\ \sum_{n=0}^{N-1} x[n] \end{bmatrix}$$

which corresponds to a closed-form global solution.

The MATLAB program is provided as `ex12_2.m`

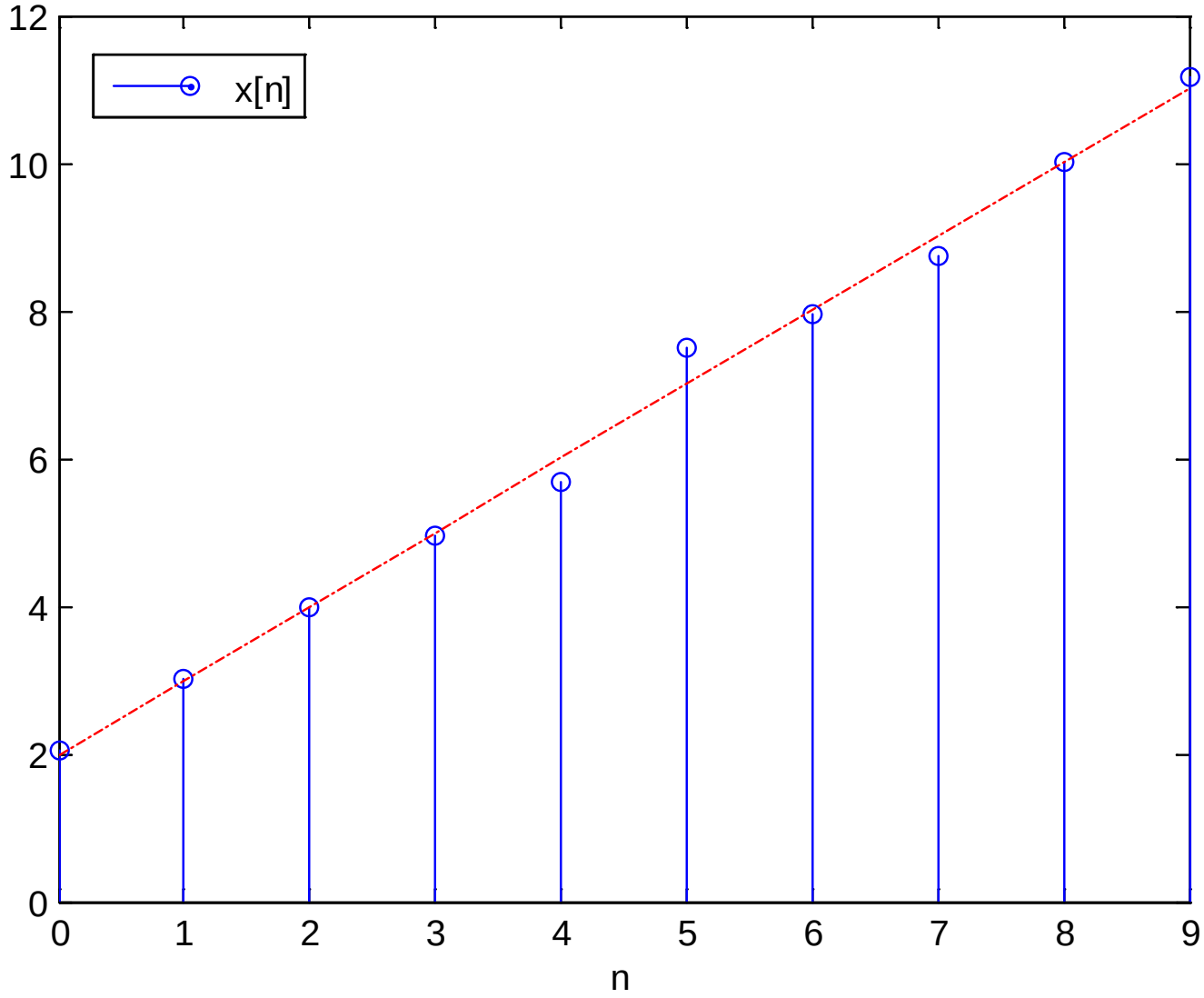


Fig.12.2: Linear regression with $\alpha = 1$, $\beta = 2$ and $N = 10$

- Iterative Numerical Methods

When the parameters of interest are not linear in the observed data, the corresponding cost functions based on the maximum likelihood and least squares methods contain local minima and maxima

If we have an initial estimate $\hat{\boldsymbol{\psi}}_0$ sufficiently close to the global optimum, $\hat{\boldsymbol{\psi}}$ can be obtained using iterative schemes:

Assuming minimization of $J(\boldsymbol{\psi})$, Netwon-Raphson method is

$$\hat{\boldsymbol{\psi}}_{k+1} = \hat{\boldsymbol{\psi}}_k - \mathbf{H}^{-1}(J(\hat{\boldsymbol{\psi}}_k))\boldsymbol{\nabla}(J(\hat{\boldsymbol{\psi}}_k)) \qquad (12.28)$$

where $\mathbf{H}(J(\hat{\boldsymbol{\psi}}_k))$ is the Hessian matrix and $\boldsymbol{\nabla}(J(\hat{\boldsymbol{\psi}}_k))$ is the gradient vector, computed at the k-th iteration estimate $\hat{\boldsymbol{\psi}}_k$.

Steepest descent method is

$$\hat{\boldsymbol{\psi}}_{k+1} = \hat{\boldsymbol{\psi}}_k - \mu \boldsymbol{\nabla}(J(\hat{\boldsymbol{\psi}}_k)) \tag{12.29}$$

where μ is a positive constant which controls the convergence rate and stability of the algorithm

Typical choices of stopping criteria include number of iterations and $|\hat{\boldsymbol{\psi}}_{k+1} - \hat{\boldsymbol{\psi}}_k| < \epsilon$ where ϵ is a sufficiently small positive constant

The Newton-Raphson algorithm provides fast convergence but matrix inverse is required

The steepest descent method is stable but its convergence rate is slow

Applications

- Signal Generation with Non-Integer Sample Delay

Given a discrete-time sequence $x[n]$, we can generate $x[n-\Delta]$ where Δ is not an integer, according to the time shifting property of discrete-time Fourier transform (DTFT) as follows. The DTFT transform pair for $x[n-\Delta]$ is:

$$x[n-\Delta] \leftrightarrow e^{-j\omega\Delta}X(e^{j\omega}) \tag{12.30}$$

The ideal frequency response for a non-integer sample delay is:

$$H_d(e^{j\omega}) = e^{-j\omega\Delta} \tag{12.31}$$

Applying the inverse DTFT, the ideal impulse response is:

$$h_d[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j\omega(n-\Delta)}d\omega = \text{sinc}(n-\Delta) \quad (12.32)$$

In theory, $x[n-\Delta]$ is computed as:

$$x[n-\Delta] = \sum_{k=-\infty}^{\infty} x[n-k]\text{sinc}\,(k-\Delta) \quad (12.33)$$

which aligns with Example 4.3.

We utilize time-shifting and truncation to obtain a practical solution:

$$y[n] = \sum_{k=0}^{2M} x[n-k]\mathrm{sinc}\,(k - M - \Delta) \qquad (12.34)$$

which corresponds to a causal system because only $x[n-2M], x[n-2M+1], \cdots, x[n]$, are required at time n

The value of M should be chosen sufficiently large to reduce the truncation error

The relationship between $y[n]$ and $x[n-\Delta]$ is:

$$y[n] \approx x[n - \Delta - M] \qquad (12.35)$$

That is, the desired time-shifted signal is obtained after a delay of M samples

Example 12.3

Design a causal finite impulse response (FIR) system with 10 coefficients to approximate a time-shifter whose input is $x[n]$ and output is $y[n] = x[n - \Delta]$ where $\Delta = 0.8$.

From (12.32), the ideal impulse response is:

$$h_d[n] = \text{sinc}(n - 0.8)$$

Investigating $h_d[n]$ for $-5 \leq n \leq 6$, we have $h_d[n] = \{-0.0323, 0.0390, -0.0492, 0.0668, -0.1039, 0.2339, 0.9355, -0.1559, 0.0850, -0.0585, 0.0445, -0.0360\}$ with $h_d[-5] = -0.0323$. The coefficients $h_d[-4], h_d[-3], \cdots, h_d[5]$, are chosen because they have the largest energy. Similar to (12.34), $y[n]$ is:

$$y[n] = \sum_{k=0}^{9} x[n-k]\text{sinc}\,(k - 4 - 0.8)$$

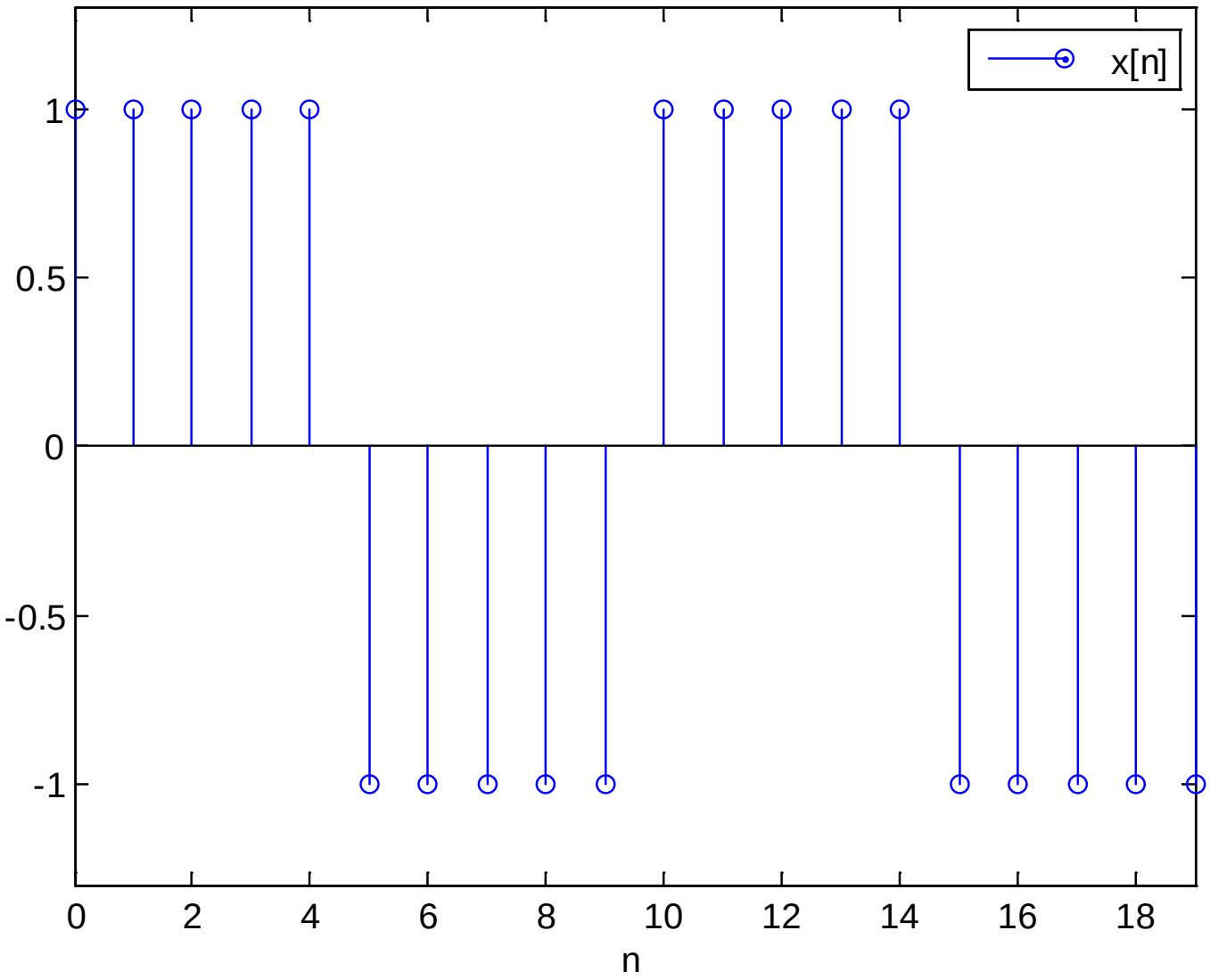


Fig.12.3: Square wave sequence

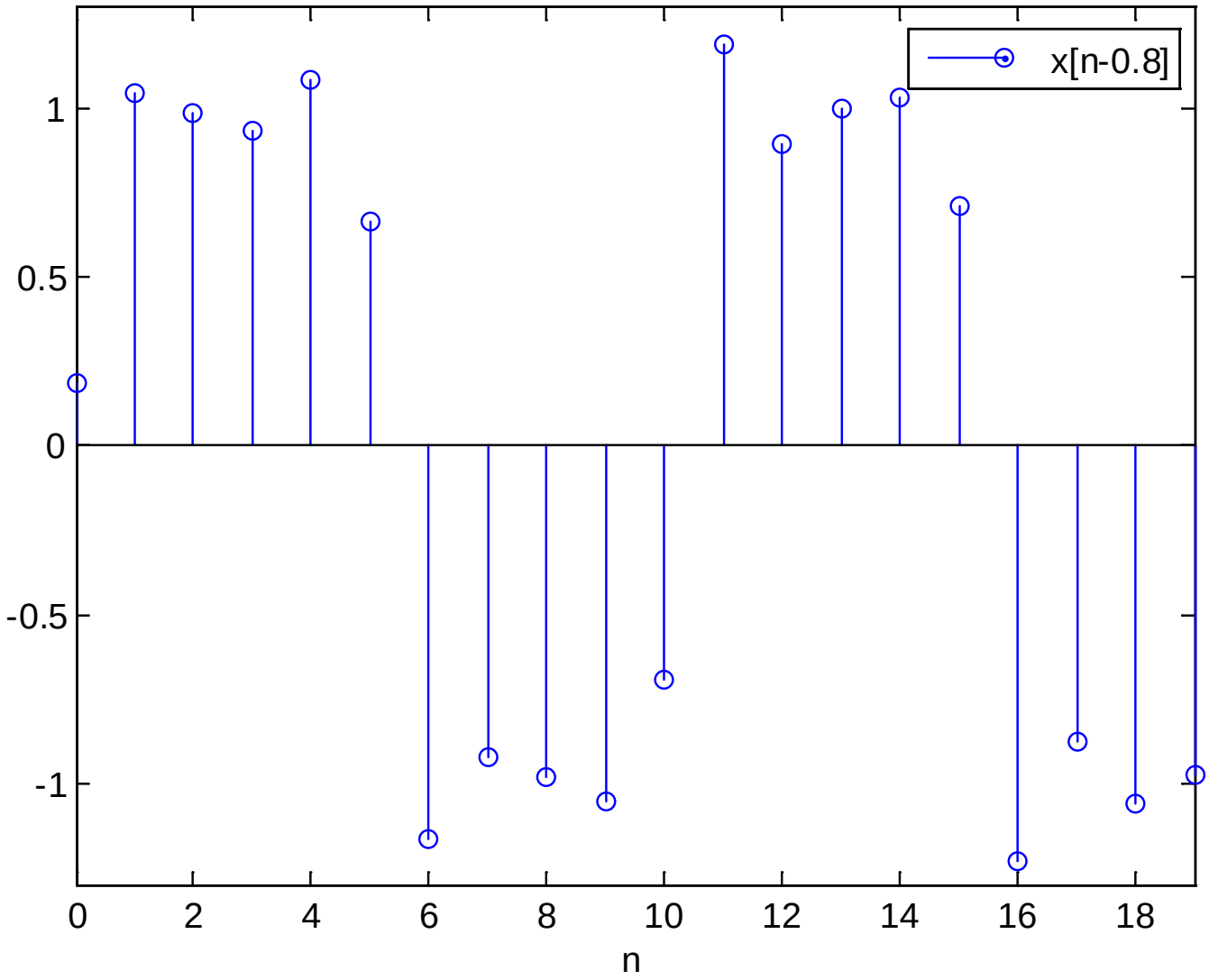
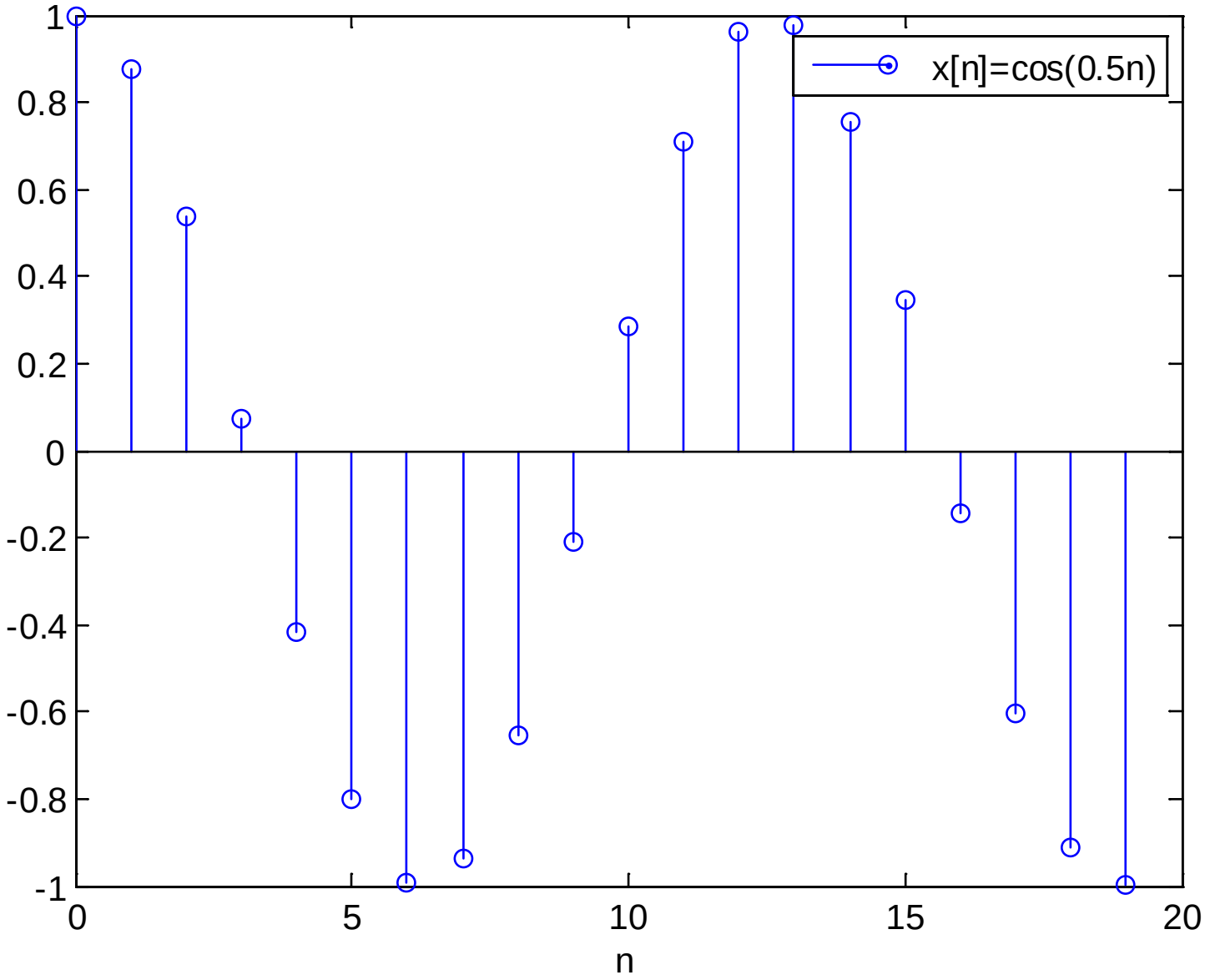
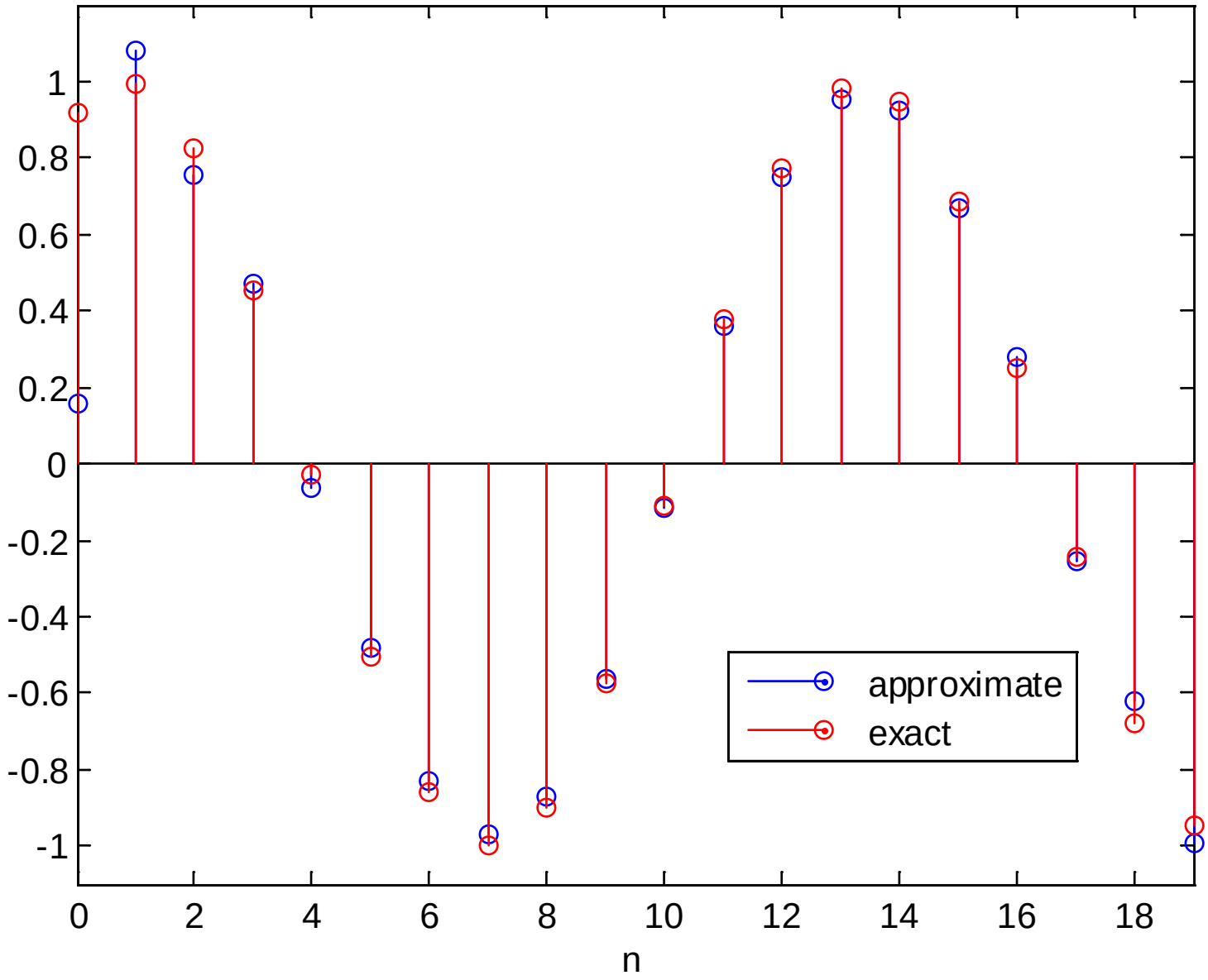


Fig.12.4: Time-shifted square wave sequence


x[n]=cos(0.5n)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5
10
15
20
n


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2
4
6
8
10
12
14
16
18
n
approximate
exact

The MATLAB program is provided as `ex12_3.m`

- Moving Average for Security Analysis

MA is one of the standard technical indicators for security analysis, which aims to identify the trend direction or define potential support and resistance levels. Basically, it is the average price of a security over a specified time period. The MA signal $y[n]$ for $x[n]$ in security analysis is:

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k] \tag{12.36}$$

(12.36) can be viewed as $x[n] \otimes h[n]$ where $h[n]$ is:

$$h[n] = \begin{cases} \frac{1}{N}, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \tag{12.37}$$

Example 12.4

Plot the 20-day MA for the Dow Jones Industrial Average (DJIA), which is a well-known stock market index, at the year of 2009.

According to (12.36), $N = 20$ and $x[n]$ corresponds to the close prices of the DJIA. Note that apart from the 252 trading days of 2009, we also need the last 19 close prices of 2008 for computation.

The MA curve is a smoothed and delayed version

The smoothing is resulted from the lowpass filtering process where the high frequency components in $x[n]$ are removed.

There is a delay because 19 previous samples are employed

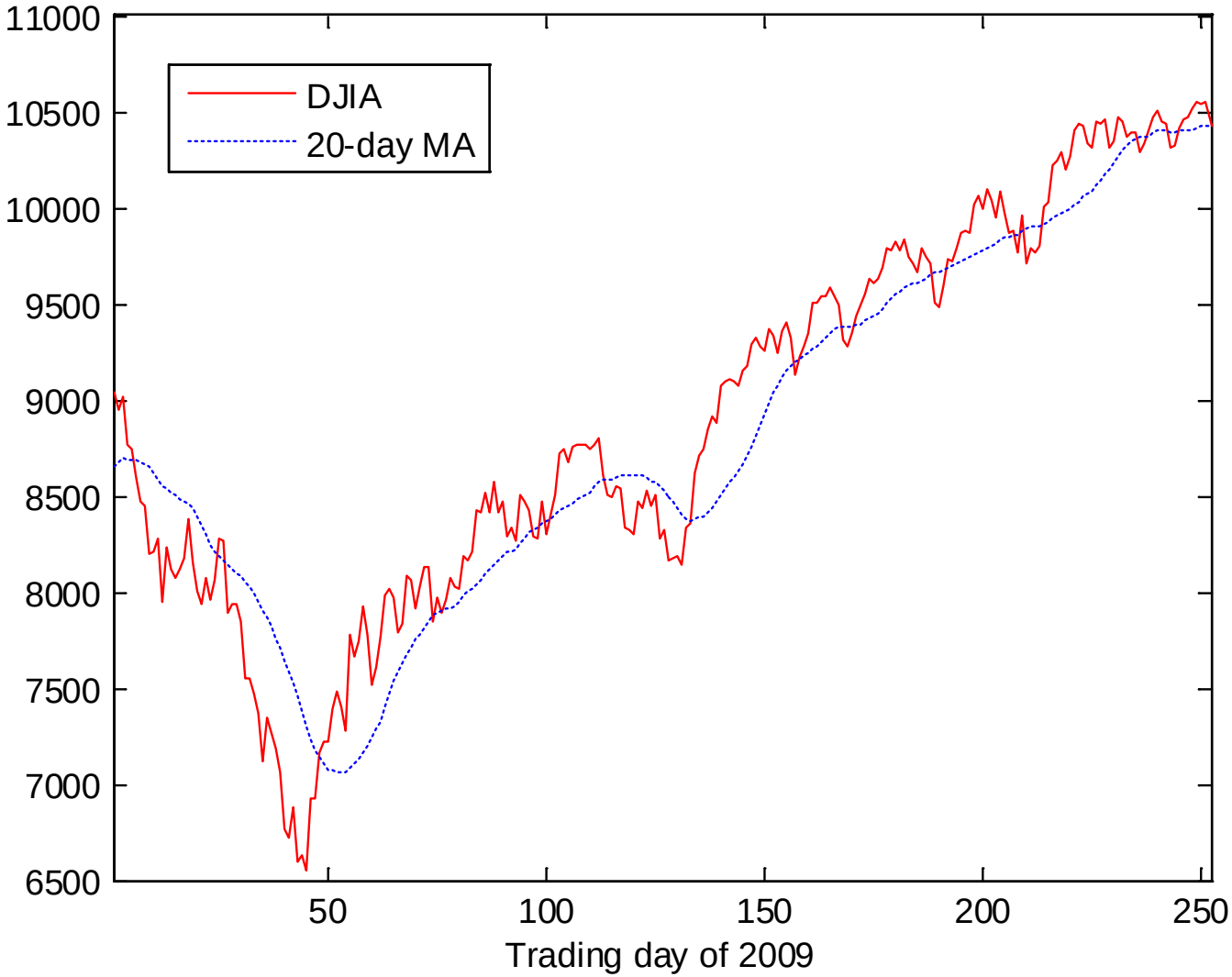


Fig.12.5: 20-day MA for Dow Jones Industrial Average at 2009

The MATLAB program is provided as `ex12_4.m`

- Frequency Shift-Keying Signal Detection

Frequency-shift keying (FSK) is a simple frequency modulation scheme in digital communications

The basic idea is to use $L = 2^v$ distinct frequencies, say, ω_i, $i = 0, 1, \cdots, L-1$, to represent a v-bit symbol for data transmission

The transmitted FSK signal is modeled as:

$$s_i[n] = \cos(\omega_i n), \quad i = 0, 1, \cdots, L-1, \quad n = 0, 1, \cdots, N-1 \quad (12.38)$$

where $\omega_l \neq \omega_k$ for $l \neq k$

Based on (12.16), it can be shown that $r_{s_i s_j}[l] = 0$ for all l when $i \neq j$, while $r_{s_i s_i}[0] \geq |r_{s_i s_i}[l]|$, $l \neq 0$, according to (12.12)

Suppose a noisy FSK signal $r[n]$ with unknown frequency is received

With the use of (12.15) and (12.17), the frequency index can be estimated as:

$$\hat{i} = \arg\max_i \sum_{n=0}^{N-1} r[n] s_i[n] \qquad (12.39)$$

Consider $L = 2$ such that ω_0 and ω_1 represent bits "0" and "1", respectively, bit "0" is detected if $\sum_{n=0}^{N-1} r[n]s_0[n] > \sum_{n=0}^{N-1} r[n]s_1[n]$. Otherwise, bit "1" is detected

- Spectral Analysis using Discrete Time Fourier Transform

As DTFT $X(e^{j\omega})$ shows the frequency components of its time-domain representation $x[n]$, we can find the signal frequency information by investigating the magnitude response $|X(e^{j\omega})|$.

Example 12.5

Determine the fundamental frequency of the speech segment in Fig.1.1

The sampling frequency is 22000 Hz. We use the MATLAB command `freqz` to plot the frequency magnitude response

There are three peaks at 245 Hz, 493 Hz and 740 Hz. As voiced speech is periodic, and $245 \times 2 \approx 493$ and $245 \times 3 \approx 740$, we deduce that the speech fundamental frequency is 245 Hz

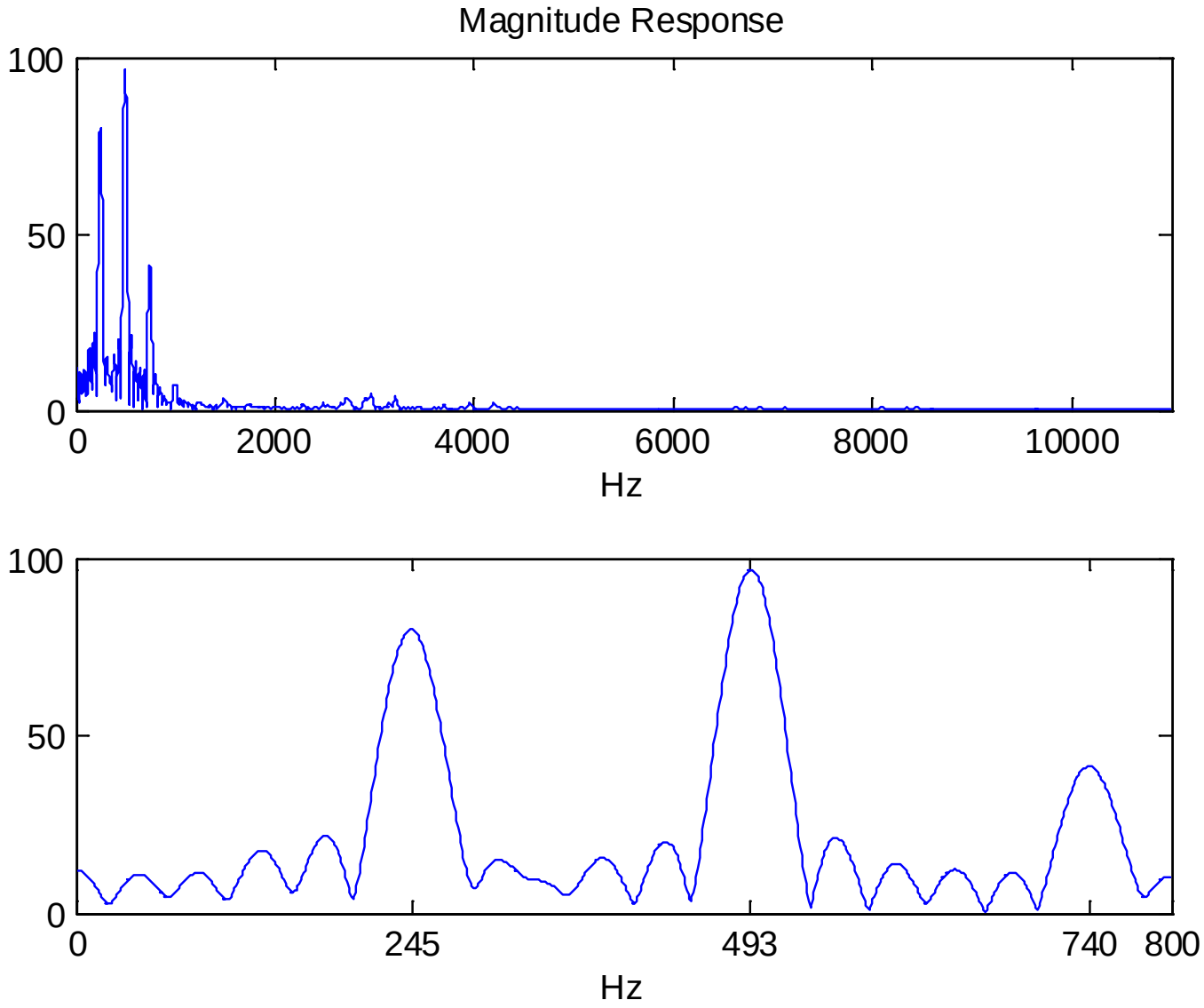


Fig.12.6: Magnitude plots for speech segment

The MATLAB program is provided as `ex12_5.m`

- Noise Reduction using Discrete Fourier Transform

Suppose we are given a noisy signal $r[n]$:

$$r[n] = s[n] + q[n], \quad n = 0, 1, \cdots, N-1 \tag{12.40}$$

where $s[n]$ is the signal of interest and $q[n]$ is the additive noise

The task is to extract $s[n]$ from $r[n]$

Suppose $s[n]$ is a narrowband signal while $q[n]$ is a wideband sequence, discrete Fourier transform (DFT) may be utilized to retrieve $s[n]$

First, we compute $R[k]$ from $r[n]$. An estimate of $S[k]$, denoted by $\hat{S}[k]$, is then obtained as:

$$\hat{S}[k] = \begin{cases} R[k], & \{\omega_k\} \text{ correspond to } s[n] \\ 0, & \text{otherwise} \end{cases} \tag{12.41}$$

where $\omega_k = 2\pi k/N$, $k = 0, 1, \cdots, N-1$. Finally, the estimate of $s[n]$ is given by the inverse DFT of $\hat{S}[k]$

Example 12.6

Given N samples of a noisy signal $r[n]$ of the form:

$$r[n] = s[n] + q[n], \quad n = 0, 1, \cdots, N-1$$

where $s[n]$ is a real sinusoid and $q[n]$ is a white noise with variance σ_q^2. Estimate $s[n]$ from $r[n]$.

A MATLAB simulation study with $s[n] = \sin(0.2n + 1)$, $\sigma_q^2 = 0.3$ and $N = 100$, is performed

The two peaks and the remaining coefficients roughly correspond to the narrowband $s[n]$ and wideband $q[n]$, respectively

As a result, we only keep the two peaks as $\hat{S}[k]$ and the estimate of $s[n]$

The MATLAB program is provided as `ex12_6.m`

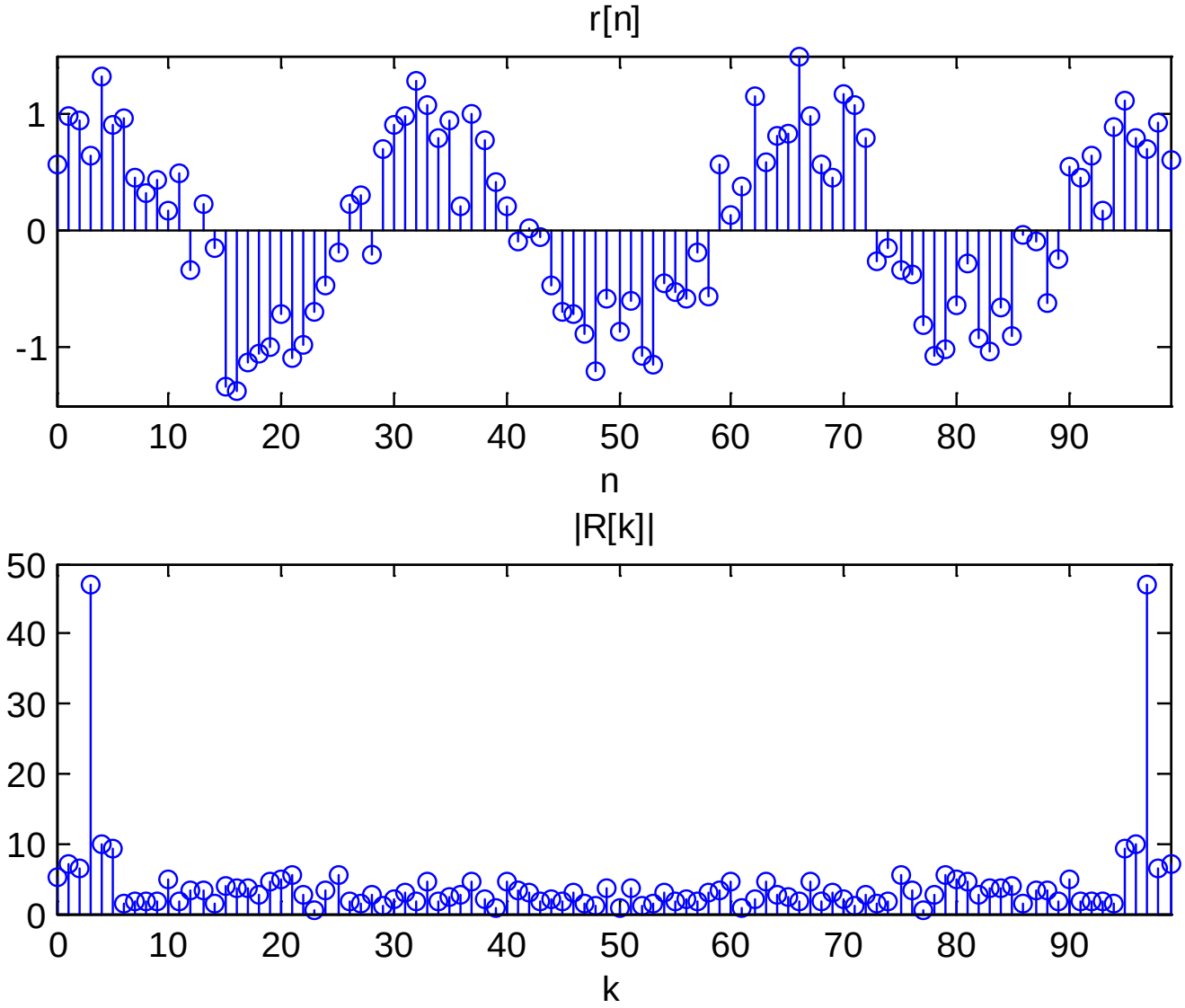


Fig.12.7: Plots for noisy sequence

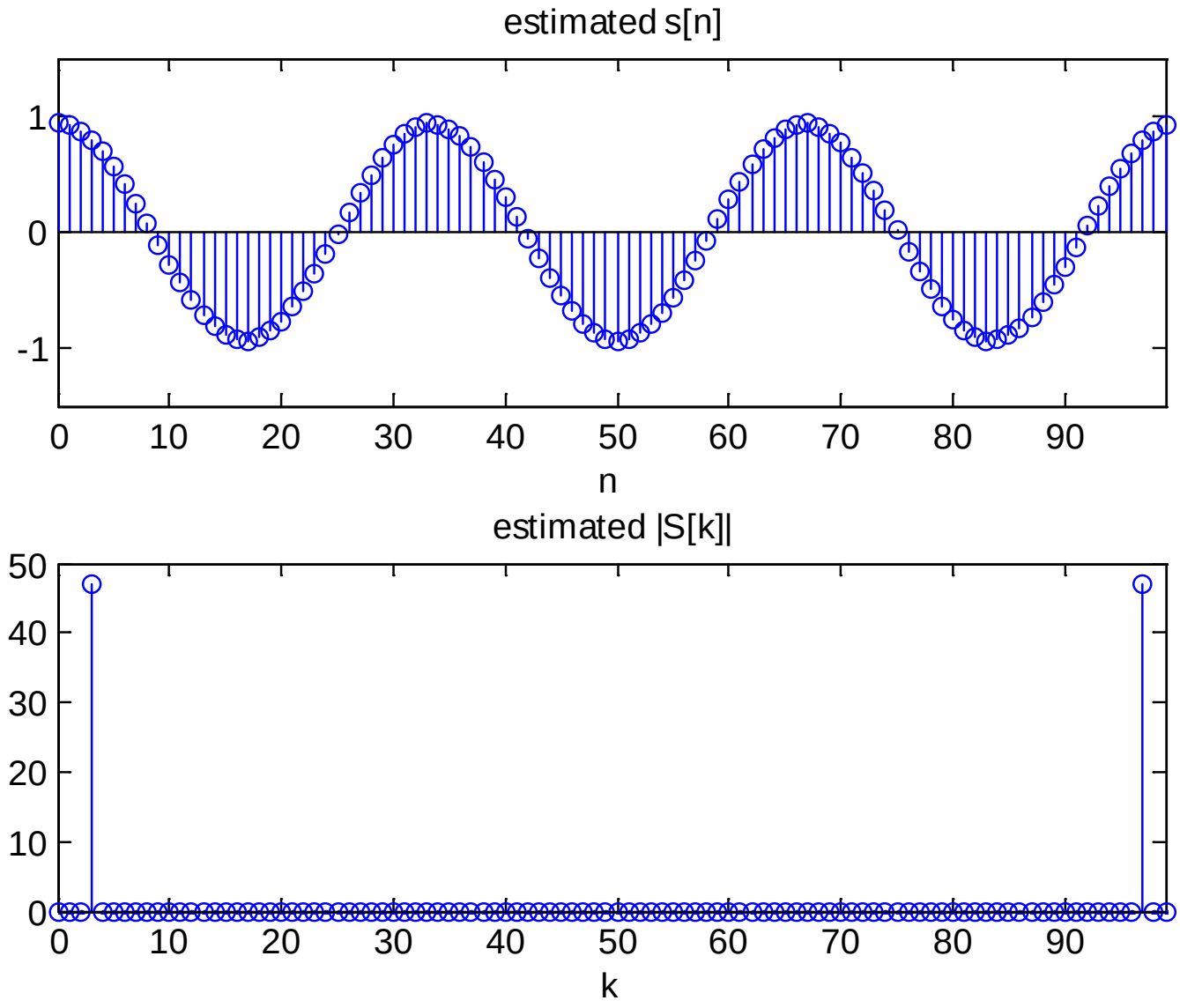


Fig.12.8: Plots for recovered signal

- Estimation of Constant in Noise

When $f(\boldsymbol{\psi}, n) = \alpha$, (12.18) becomes

$$x[n] = \alpha + q[n], \quad n = 0, 1, \cdots, N-1 \qquad (12.42)$$

Assume the noise $q[n]$ is a white Gaussian process with variance $\sigma_{q_n}^2$, $n = 0, 1, \cdots, N-1$

The noise covariance matrix is now equal to $\mathbf{C} = \text{diag}(\sigma_{q_0}^2, \sigma_{q_1}^2, \cdots, \sigma_{q_{N-1}}^2)$ which is a diagonal matrix of $\{\sigma_{q_n}^2\}$

The maximum likelihood estimate of α is

$$\hat{\alpha} = \arg\min_{\alpha} J(\alpha) \qquad (12.43)$$

where

$$J(\alpha) = \sum_{n=0}^{N-1} \frac{(x[n] - \alpha)^2}{\sigma_{q_n}^2} \qquad (12.44)$$

Differentiating $J(\alpha)$ with respect to α and then setting the resultant expression to zero, we get:

$$\left.\frac{dJ(\alpha)}{\partial \alpha}\right|_{\alpha=\hat{\alpha}} = 2\sum_{n=0}^{N-1} \frac{(x[n] - \hat{\alpha})(-1)}{\sigma_{q_n}^2} = 0 \Rightarrow \hat{\alpha} = \frac{\sum_{n=0}^{N-1} \sigma_{q_n}^{-2} x[n]}{\sum_{n=0}^{N-1} \sigma_{q_n}^{-2}} \qquad (12.45)$$

where each $x[n]$ is weighted by $1/\sigma_{q_n}^2$. That is, a smaller weight is used for $x[n]$ with a larger noise and vice versa

The least squares solution is obtained by substituting $\sigma_{q_n}^2 = 1$:

$$\hat{\alpha} = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \qquad (12.46)$$

which is simply the average value of $x[n]$

In terms of estimation performance, (12.45) is more accurate than (12.46) but the latter has the advantage that prior knowledge of the noise powers is not required

- Time Delay Estimation

The problem is to find the time-shift between two versions of the same signal

Radar ranging is a representative application

From Fig.1.3 and let the transmitted and received signals be $s[n]$ and $r[n]$, respectively, the time delay estimation problem is to find τ with the use of $s[n]$ and $r[n]$

The received signal $r[n]$ is:

$$r[n] = \alpha s[n - \tau] + q[n] \qquad (12.47)$$

where α is the attenuation parameter, τ is the round-trip propagation time, and $q[n]$ is a zero-mean noise.

For simplicity, here we assume that τ is an integer

We can use correlation to estimate τ:

$$\hat{\tau} = \arg\max_{\tau} \sum_{n} r[n]s[n-\tau] \tag{12.48}$$

Or using least squares approach where the cost function is:

$$J(\alpha, \tau) = \sum_{n} (r[n] - \alpha s[n-\tau])^2 \tag{12.49}$$

which contains two variables

Differentiate $J(\alpha, \tau)$ with respect to α and set the resultant expression to zero to find the least squares estimate of α in terms of τ, and then substitute it back to (12.49)

The least squares solution of τ can also be written as

$$\hat{\tau} = \arg\min_{\tau} \sum_{n} \left(r[n] - \frac{\sum_{k} r[k]s[k-\tau]}{\sum_{k} s^2[k-\tau]} s[n-\tau] \right)^2 \qquad (12.50)$$

- Sinusoidal Parameter Estimation

The problem is to find the amplitudes, frequencies and phases of tones embedded in noise

The simplest model corresponds to $L = 1$ in (12.3):

$$x[n] = \alpha \cos(\omega n + \phi) + q[n], \quad n = 0, 1, \cdots, N-1 \qquad (12.51)$$

Assume $q[n]$ is white Gaussian with variance σ_q^2

The maximum likelihood estimate is equal to the least squares solution:

$$\hat{\boldsymbol{\psi}} = \arg\min_{\boldsymbol{\psi}} J(\boldsymbol{\psi}), \quad \boldsymbol{\psi} = [\alpha\ \omega\ \phi]^T \tag{12.52}$$

where

$$J(\boldsymbol{\psi}) = \sum_{n=0}^{N-1} [x[n] - \alpha\cos(\omega n + \phi)]^2 \tag{12.53}$$

which corresponds to a multimodal function

With an initial estimate $\hat{\boldsymbol{\psi}}_0$ sufficiently close to the global minimum of (12.53), the steepest descent algorithm for finding $\boldsymbol{\psi}$ is:

$$\begin{bmatrix} \hat{\alpha}_{k+1} \\ \hat{\omega}_{k+1} \\ \hat{\phi}_{k+1} \end{bmatrix} = \begin{bmatrix} \hat{\alpha}_k \\ \hat{\omega}_k \\ \hat{\phi}_k \end{bmatrix} - \mu \begin{bmatrix} \dfrac{\partial J(\hat{\boldsymbol{\psi}}_k)}{\partial \hat{\alpha}_k} \\ \dfrac{\partial J(\hat{\boldsymbol{\psi}}_k)}{\partial \hat{\omega}_k} \\ \dfrac{\partial J(\hat{\boldsymbol{\psi}}_k)}{\partial \hat{\phi}_k} \end{bmatrix}$$

$$= \begin{bmatrix} \hat{\alpha}_k \\ \hat{\omega}_k \\ \hat{\phi}_k \end{bmatrix} - 2\mu \begin{bmatrix} -\sum_{n=0}^{N-1} \left[x[n] - \hat{\alpha}_k \cos(\hat{\omega}_k n + \hat{\phi}_k) \right] \cos(\hat{\omega}_k n + \hat{\phi}_k) \\ \sum_{n=0}^{N-1} \left[x[n] - \hat{\alpha}_k \cos(\hat{\omega}_k n + \hat{\phi}_k) \right] \sin(\hat{\omega}_k n + \hat{\phi}_k) n \\ \sum_{n=0}^{N-1} \left[x[n] - \hat{\alpha}_k \cos(\hat{\omega}_k n + \hat{\phi}_k) \right] \sin(\hat{\omega}_k n + \hat{\phi}_k) \end{bmatrix} \quad (12.54)$$