DESIGN OF FIR FILTERS USING IDENTICAL SUBFILTERS AS BASIC BUILDING BLOCKS

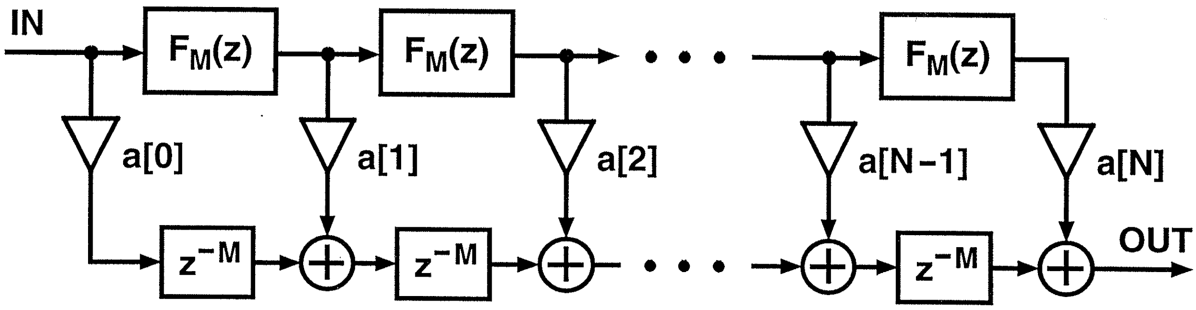
- The cost of implementation of an FIR filter can be reduced by designing it by interconnecting a number of identical subfilters with the aid of a few additional adders and multipliers.
- The main advantage of using identical copies of the same filter lies in the fact that with this approach it is relatively easy to synthesize selective FIR filters without general multipliers.
- This material is based on T. Saramäki, "Finite impulse reponse filter design", Chapter 4 in *Handbook for Digital Signal Processing*, edited by S. K. Mitra and J. F. Kaiser, John Wiley and Sons, New York, 1993, Section 4-12.
- This pile of lecture notes contains some additional material compared to the above-mentioned article.

Filter Structures

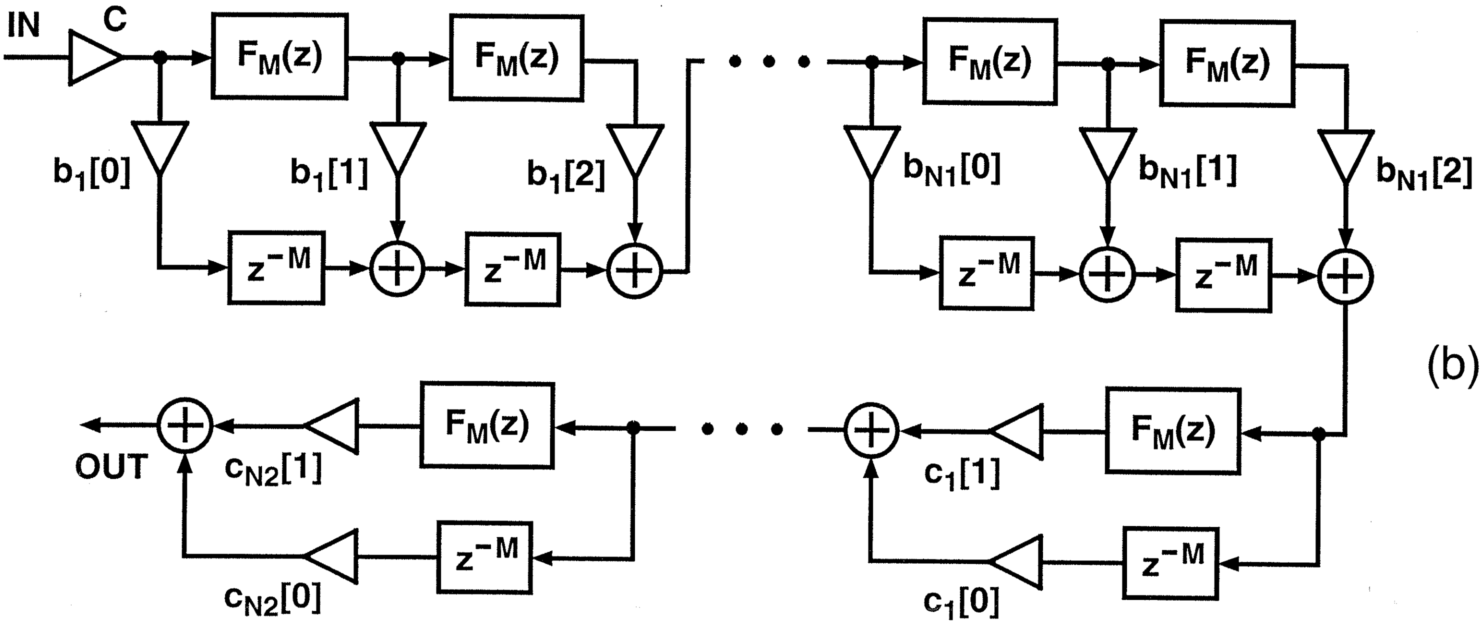
- The building block (subfilter) $F_M(z)$ is a Type I filter of order $2M$:

$$F_M(z) = \sum_{n=0}^{2M} f[n]z^{-n}, \quad f[2M-n] = f[n].$$

- The first structure is used for the design purposes and the second one for the implementation purposes.



(a)



(b)

Filter Frequency Response

- The use of the extra delay terms z^{-M} guarantees that the overall frequency response has linear phase and is expressible as

$$H(e^{j\omega}) = e^{-jNM\omega} H(\omega),$$

where

$$H(\omega) = \sum_{n=0}^{N} a[n][F_M(\omega)]^n$$

with

$$F_M(\omega) = f[M] + 2\sum_{n=0}^{M} f[M-n]\cos n\omega.$$

Conditions for the Tap Coefficients and Subfilter

- The tap coefficients $a[n]$ and $F_M(z)$ can be determined such that $H(\omega)$ meets

$$1 - \delta_p \leq H(\omega) \leq 1 + \delta_p \quad \text{for} \quad \omega \in X_p$$
$$-\delta_s \leq H(\omega) \leq \delta_s \quad \text{for} \quad \omega \in X_s.$$

- X_p and X_s may consist of several bands.

Simultaneous Specifications

- Since $H(\omega)$ can be obtained from the polynomial

$$P(x) = \sum_{n=0}^{N} a[n]x^n$$

using the substitution

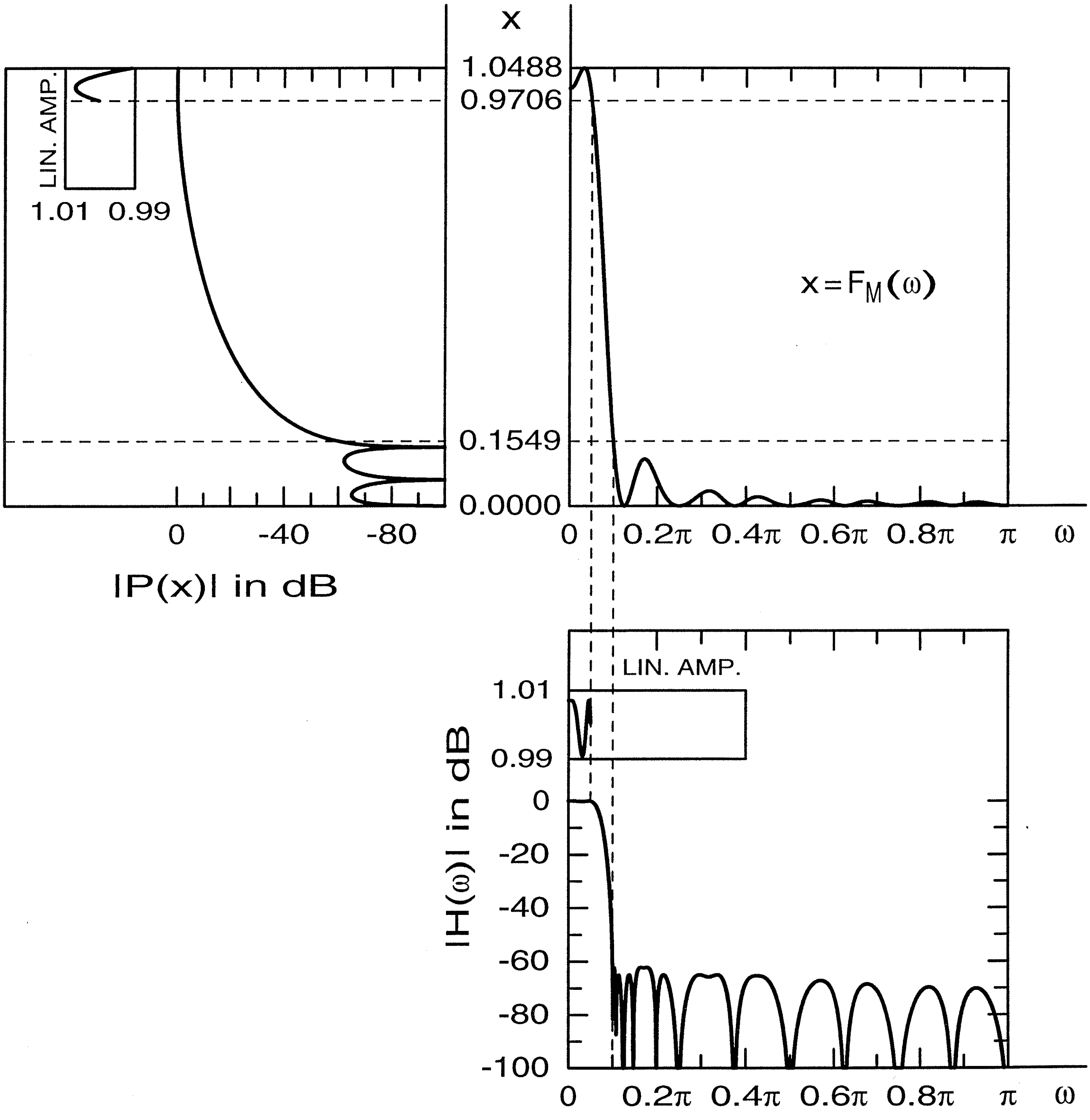
$$x = F_M(\omega),$$

the general simultaneous conditions for the $a[n]$'s and $F_M(z)$ can be stated as

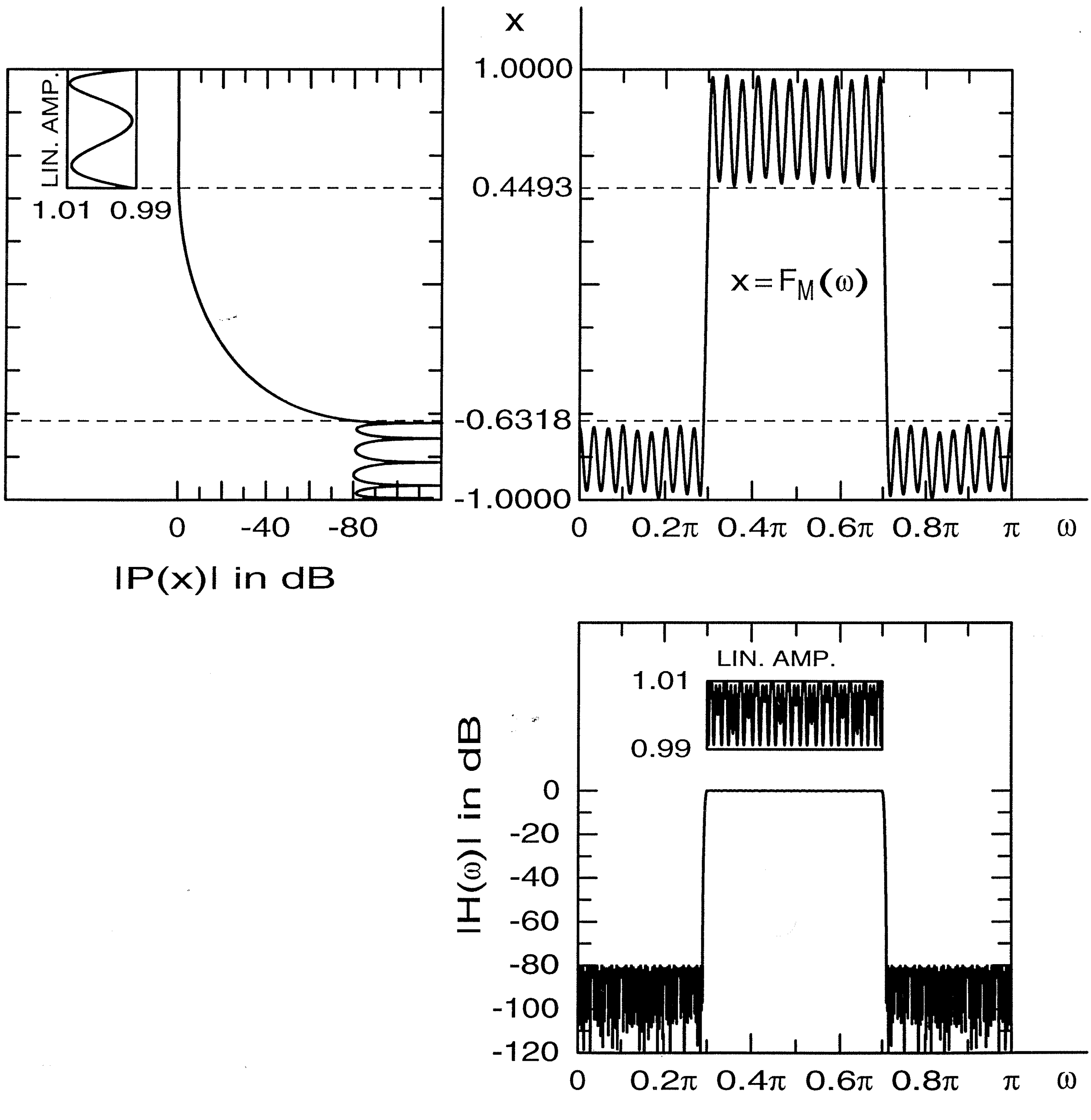
$$\begin{aligned} 1 - \delta_p \leq P(x) &\leq 1 + \delta_p \quad \text{for} \quad x_{p1} \leq x \leq x_{p2} \\ -\delta_s \leq P(x) &\leq \delta_s \quad \text{for} \quad x_{s1} \leq x \leq x_{s2} \\ x_{p1} \leq F_M(\omega) &\leq x_{p2} \quad \text{for} \quad \omega \in X_p \\ x_{s1} \leq F_M(\omega) &\leq x_{s2} \quad \text{for} \quad \omega \in X_s. \end{aligned}$$

- The figures in the following three transparencies exemplify these relations in three different cases to be considered in more details later.

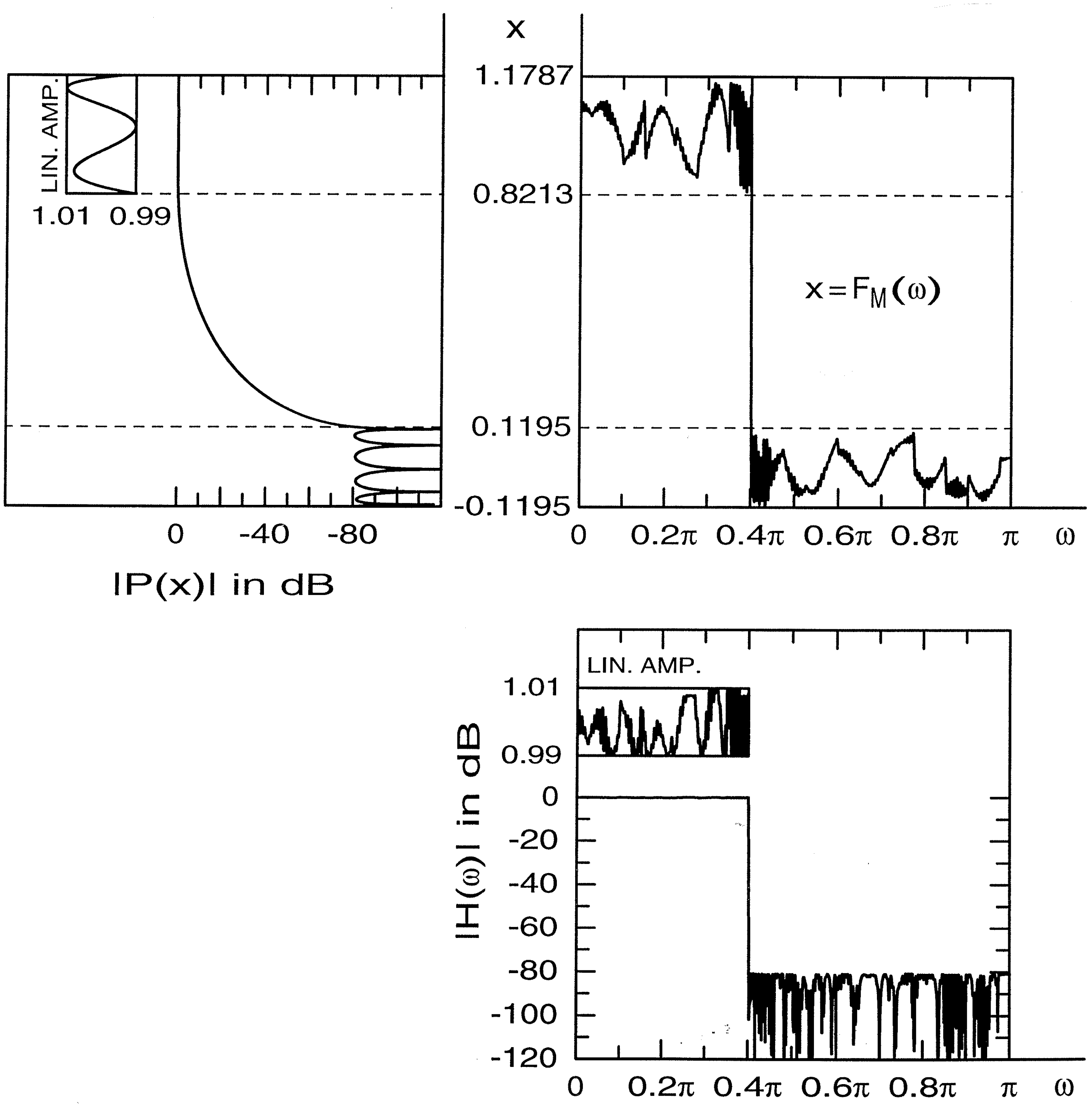
Example 1: Design of a composite filter using four prescribed subfilters. $x_{s1} = 0$, $x_{s2} = 0.1549$, $x_{p1} = 0.9706$, **and** $x_{p2} = 1.0488$ **are determined by the subfilter.** $\omega_p = 0.05\pi$, $\omega_s = 0.1\pi$, $\delta_p = 0.01$, **and** $\delta_s = 0.001$.



Example 2: Case A Simultaneous Specifications: $x_{s1} = -1$, $x_{p2} = 1$. $x_{s2} = -0.6318$, $x_{p1} = 0.4493$, ω_{p1}, $\omega_{p2} = 0.5\pi \pm 0.2\pi$, ω_{s1}, $\omega_{s2} = 0.5\pi \pm 0.21\pi$, $N = 8$ **subfilters,** $\delta_p = 0.01$, **and** $\delta_s = 0.0001$.



Example 3: Case B Simultaneous Specifications: $x_{p1},\ x_{p2} = 1 \pm \widehat{\delta}_p,\ x_{s1},\ x_{s2} = \widehat{\delta}_s$. $\widehat{\delta}_p = 0.1787$, $\widehat{\delta}_s = 0.1195$, $N = 8$ **subfilters.** $\omega_p = 0.4\pi$, $\omega_s = 0.402\pi$, $\delta_p = 0.01$, **and** $\delta_s = 0.0001$.



Interpretations

- If these conditions are satisfied, then $x = F_M(\omega)$ converts the desired passband behavior of $P(x)$ on $[x_{p1}, x_{p2}]$ (oscillations within the limits $1 \pm \delta_p$) onto the passband region X_p of the overall filter. Similarly, the desired stopband behavior of $P(x)$ on $[x_{s1}, x_{s2}]$ is converted onto the stopband region X_s.
- Hence, the amplitude values are preserved and only the argument axis is changed.
- Alternatively, $P(x)$ can be interpreted as an *amplitude change function*, which tells that if the subfilter response $F_M(\omega)$ achieves the value x_0, then the overall response $H(\omega)$ achives the value $P(x_0)$ without regard of the frequency.
- The passband and stopband regions of $F_M(\omega)$ and $H(\omega)$ are the same.
- All that happens is that the multiple use of the same subfilter reduces the large passband and stopband variations in $F_M(\omega)$ to small variations in $H(\omega)$.

Basic Problems

- *Problem I*: Given N, the number of subfilters, optimize the $a[n]$'s [or, equivalently, $P(x)$] and $F_M(z)$ to meet the given criteria with the minimum subfilter order $2M$.
- *Problem II*: Given $F_M(z)$, optimize the $a[n]$'s to meet the given criteria with the minimum value of N.
- In addition to these problems, $F_M(z)$ and $P(x)$ can be optimized to minimize N, the number of subfilters, for the given subfilter order $2M$.
- Also, the subfilter order can be minimized for the given N and the given values of the $a[n]$'s, like in the Kaiser-Hamming approach.

Three Different Simultaneous Specifications

- For Problem II, the parameters x_{p1}, x_{p2}, x_{s1}, and x_{s2} are fixed and determined by $F_M(z)$ (see Example 1 in transparency 6).
- For Problem I, these parameters are adjustable and their number can be can be reduced from four to two in the following two useful ways:
- Case A: $x_{s1} = -1$, $x_{p2} = 1$; x_{s2} and x_{s1} are adjustable.
- Case B: $x_{s1} = -\widehat{\delta}_s$, $x_{s2} = \widehat{\delta}_s$, $x_{p1} = 1 - \widehat{\delta}_p$, $x_{p2} = 1 + \widehat{\delta}_p$; $\widehat{\delta}_p$ and $\widehat{\delta}_s$ are adjustable.
- Case A is beneficial when the subfilter is a conventional direct-form design: the subfilter is automatically peak scaled with the maximum and minimum values of $F_M(\omega)$ being $+1$ and -1, respectively (see Example 2 in transparency 7).
- In Case B, the subfilter criteria are conventional with passband ripple of $\widehat{\delta}_p$ and stopband ripple of $\widehat{\delta}_p$ (see Example 3 in transparency 8).

Tap Coefficients for the Second Overall Structure

- The additional tap coefficients in the second overall structure of transparency 2 can be obtained by factoring the polynomial $P(x)$ into the second-order and first-order terms as

$$P(x) = C\prod_{k=1}^{N_1}[b_k[2]x^2+b_k[1]x+b_k[0]]\prod_{k=1}^{N_2}[c_k[1]x+c_k[0]],$$

where

$$2N_1 + N_2 = N.$$

- The advantages of this structure compared to the first one are that the extra delays z^{-M} can be shared with the subfilter $F_M(z)$ and its sensitivity to variations in the tap coefficients is lower.
- Later on, it will be shown how $P(x)$ can be designed directly in the above form.

Filter Optimization

- For the above problems, the design of $P(x)$ can be accomplished conveniently with the aid of an FIR filter using the substitution

$$x = \alpha \cos \Omega + \beta$$

in $P(x)$, yielding

$$G(\Omega) = P(\alpha \cos \Omega + \beta) = \sum_{n=0}^{N} g[n] \cos^n \Omega,$$

where

$$g[n] = \sum_{r=n}^{N} a[r] \binom{r}{n} \alpha^n \beta^{r-n}. \qquad (A)$$

- Being expressible as an N-th degree polynomial in $\cos \Omega$, $G(\Omega)$ is the zero-phase frequency response of a Type I linear-phase FIR filter of order $2N$ and can be designed using standard FIR filter design algorithms (see the lecture notes on Digital Filtering II, especially the handbook chapter).

- These algorithms give the impulse-response coefficients $\widehat{g}[n]$ of the corresponding filter.
- $G(\Omega)$ can be expressed as $G(\Omega) = \widehat{g}[N] + 2\sum_{n=1}^{N} \widehat{g}[N-n]\cos n\Omega$ which can be rewritten in the desired form using the identity $\cos n\Omega = T_n(\cos\Omega)$, where $T_n(x)$ is the n-th degree Chebyshev polynomial.
- On page 17 it will be shown how $P(x)$ can be generated based on the impulse response as well as the zero locations of the corresponding FIR filter.

- By selecting

$$\alpha = (x_{p2} - x_{s1})/2, \quad \beta = (x_{p2} + x_{s1})/2,$$

the x-plane regions $[x_{p1}, x_{p2}]$ and $[x_{s1}, x_{s2}]$ are mapped, respectively, onto the Ω-plane regions $[0, \Omega_p]$ and $[\Omega_s, \pi]$, where (see the figure in the following transparency)

$$\Omega_p = \cos^{-1}\left[\frac{2x_{p1} - x_{p2} - x_{s1}}{x_{p2} - x_{s1}}\right], \ \Omega_s = \cos^{-1}\left[\frac{2x_{s2} - x_{p2} - x_{s1}}{x_{p2} - x_{s1}}\right]$$

and the conditions for $P(x)$ can be expressed in terms of $G(\Omega)$ as

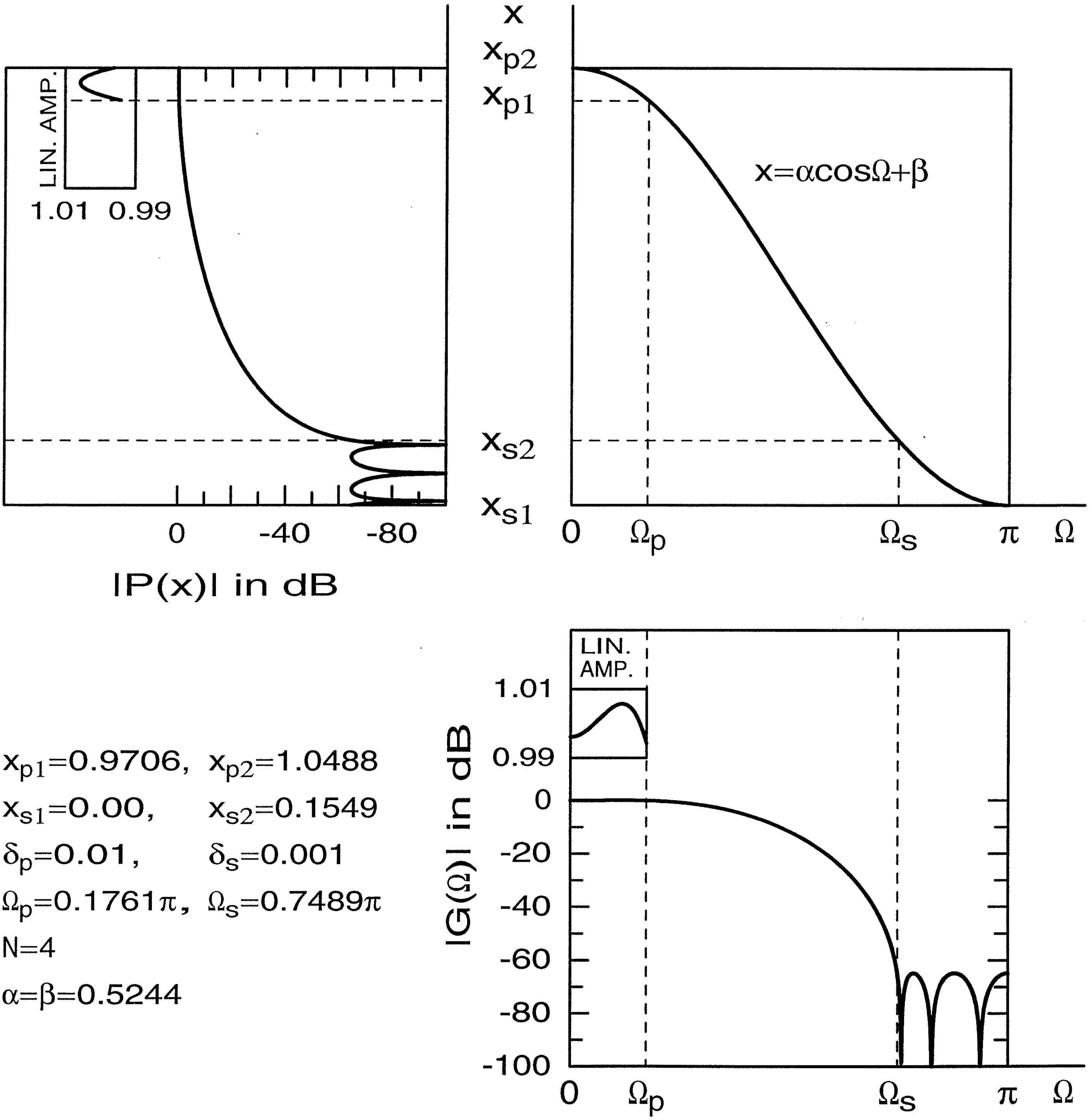
$$1 - \delta_p \le G(\Omega) \le 1 + \delta_p \quad \text{for} \quad 0 \le \Omega \le \Omega_p$$
$$-\delta_s \le G(\Omega) \le \delta_s \quad \text{for} \quad \Omega_s \le \Omega \le \pi.$$

- $G(\Omega)$ meeting these conventional lowpass specifications can then be converted back into the polynomial $P(x)$ using the substitution

$$\cos\Omega = [x - \beta]/\alpha.$$

- The resulting tap coefficients $a[n]$ can be determined from the $g[n]$'s according to Eq. (A).

Design of the polynomial $P(x)$ with the aid of an FIR filter response $G(\Omega)$ for the given x_{s1}, x_{s2}, x_{p1}, and x_{p2} and for the given δ_p and δ_s.



$x_{p1}=0.9706$, $x_{p2}=1.0488$

$x_{s1}=0.00$, $x_{s2}=0.1549$

$\delta_p=0.01$, $\delta_s=0.001$

$\Omega_p=0.1761\pi$, $\Omega_s=0.7489\pi$

$N=4$

$\alpha=\beta=0.5244$

How to Find $P(x)$ Conveniently in the Cascade Form

- Let the transfer function $G(z)$ satisfying the conditions given on page 15 be of the form

$$G(z) = \sum_{n=0}^{2N} \widehat{g}[n] z^{-n}, \quad \widehat{g}[2N-n] = \widehat{g}[n].$$

- This $G(z)$ is factorizable as

$$G(z) = \widehat{g}[0] G_1(z) G_2(z) G_3(z),$$

where

$$\begin{aligned} G_1(z) = \prod_{k=1}^{M_1} (1 &- [2(r_k + \frac{1}{r_k}) \cos\theta_k] z^{-1} \\ &+ [r_k^2 + \frac{1}{r_k^2} + 4\cos^2\theta_k] z^{-2} \\ &- [2(r_k + \frac{1}{r_k}) \cos\theta_k] z^{-3} + z^{-4}), \end{aligned}$$

$$G_2(z) = \prod_{k=1}^{M_2} (1 - [\widehat{r}_k + \frac{1}{\widehat{r}_k}] z^{-1} + z^{-2}),$$

$$G_3(z) = \prod_{k=1}^{M_3} (1 - [2\cos\widehat{\theta}_k] z^{-1} + z^{-2}).$$

- Here $2M_1 + M_2 + M_3 = N$.

- $G_1(z)$ contains M_1 zero quadruplets at $z = r_k e^{\pm j\theta_k}$ and $z = (1/r_k)e^{\pm j\theta_k}$ for $k = 1, 2, \cdots, M_1$.
- $G_2(z)$ contains M_2 reciprocal zero pairs on the real axis at $z = \widehat{r}_k, 1/\widehat{r}_k$ for $k = 1, 2, \cdots, M_2$.
- $G_3(z)$ contains M_3 zero pairs on the unit circle at $z = e^{\pm j\widehat{\theta}_k}$ for $k = 1, 2, \cdots, M_3$.
- The corresponding zero-phase frequency response is expressible as

$$G(\Omega) = \widehat{g}[0]G_1(\Omega)G_2(\Omega)G_3(\Omega),$$

where

$$G_1(\Omega) = \prod_{k=1}^{M_1}(2\cos 2\Omega - [4(r_k + \frac{1}{r_k})\cos\theta_k]\cos\Omega + [r_k^2 + \frac{1}{r_k^2} + 4\cos^2\theta_k]),$$

$$G_2(\Omega) = \prod_{k=1}^{M_2}(2\cos\Omega - [\widehat{r}_k + \frac{1}{\widehat{r}_k}]),$$

$$G_3(\Omega) = \prod_{k=1}^{M_3}(2\cos\Omega - [2\cos\widehat{\theta}_k]).$$

- Alternatively, $(2\cos 2\Omega = 4\cos^2\Omega - 2)$

$$G_1(\Omega) = 2^{2M_1}\prod_{k=1}^{M_1}(\cos^2\Omega + a_k\cos\Omega + b_k),$$

$$G_2(\Omega) = 2^{M_2} \prod_{k=1}^{M_2} (\cos\Omega + d_k),$$

and

$$G_3(\Omega) = 2^{M_3} \prod_{k=1}^{M_3} (\cos\Omega + e_k),$$

where

$$a_k = -(r_k + \frac{1}{r_k}) \cos\theta_k,$$

$$b_k = (r_k^2 + \frac{1}{r_k^2})/4 + \cos^2\theta_k - 1/2,$$

$$d_k = -(\widehat{r}_k + \frac{1}{\widehat{r}_k})/2,$$

and

$$e_k = -\cos\widehat{\theta}_k.$$

- Hence,

$$G(\Omega) = \widehat{C} \prod_{k=1}^{N_1} [\widehat{b}_k[2]\cos^2\Omega + \widehat{b}_k[1]\cos\Omega + \widehat{b}_k[0]] \times \prod_{k=1}^{N_2} [\widehat{c}_k[1]\cos\Omega + \widehat{c}_k[0]],$$

where

$$\widehat{C} = 2^N \widehat{g}[0],$$

$$N_1 = M_1,$$

$$N_2 = M_2 + M_3,$$

$$\widehat{b}_k[2] = 1 \quad \text{for} \quad k = 1, 2, \cdots, N_1,$$

$$\widehat{b}_k[1] = a_k \quad \text{for} \quad k = 1, 2, \cdots, N_1,$$

$$\widehat{b}_k[0] = b_k \quad \text{for} \quad k = 1, 2, \cdots, N_1,$$

$$\widehat{c}_k[1] = 1 \quad \text{for} \quad k = 1, 2, \cdots, N_2,$$

$$\widehat{c}_k[0] = d_k \quad \text{for} \quad k = 1, 2, \cdots, M_2,$$

and

$$\widehat{c}_k[0] = e_{k-M_2} \quad \text{for} \quad k = M_2 + 1, M_2 + 2, \cdots, N_2.$$

- The unscaled $P(x)$ obtained from $G(\Omega)$ using the substitution

$$\cos \Omega = [x - \beta]/\alpha$$

is then

$$P(x) = C \prod_{k=1}^{N_1} [b_k[2]x^2 + b_k[1]x + b_k[0]] \prod_{k=1}^{N_2} [c_k[1]x + c_k[0]],$$

where

$$C = \widehat{C}/\alpha^N$$

$$b_k[2] = 1 \quad \text{for} \quad k = 1, 2, \cdots, N_1,$$

$$b_k[1] = \alpha \widehat{b}_k[1] - 2\beta \quad \text{for} \quad k = 1, 2, \cdots, N_1,$$

$$b_k[0] = \alpha^2 \widehat{b}_k[0] - \beta\alpha\widehat{b}_k[1] + \beta^2 \quad \text{for} \quad k = 1, 2, \cdots, N_1,$$

$$c_k[1] = 1 \quad \text{for} \quad k = 1, 2, \cdots, N_2,$$

and

$$c_k[0] = \alpha\widehat{c}_k[0] - \beta \quad \text{for} \quad k = 1, 2, \cdots, N_2.$$

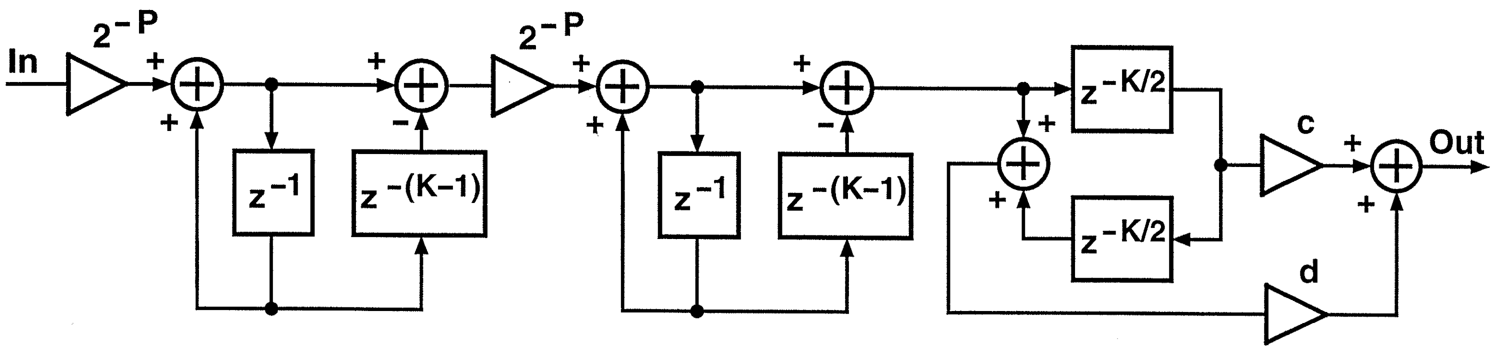
Example

- This example illustrates how multiplier-free filters can be designed by first determining a computationally efficient subfilter with higher ripple values than the required ones and then using the additional tap coefficients to reduce these ripples to the desired level (Problem II).
- Consider the specifications: $\omega_p = 0.05\pi$, $\omega_s = 0.1\pi$, $\delta_p = 0.01$, and $\delta_s = 0.001$.
- For narrowband cases of this kind, a particularly efficient subfilter transfer function is of the form:

$$F_M(z) = \left[2^{-P}\frac{1 - z^{-K}}{1 - z^{-1}}\right]^2 \left[cz^{-K/2} + d(1 + z^{-K})\right],$$

  where 2^{-P}, with P integer-valued, is a scaling multiplier and $M = 3K/2 - 1$.
- An efficient implementation of this transfer function is depicted in the following transparency.

An implementation of the proposed subfilter



- If modulo arithmetic (e.g., 1's or 2's complement arithmetic) and the worst-case scaling $2^{-P} \leq 1/K$ (corresponds to peak scaling in this case) are used, the outputs of the first two blocks in the above figure are correct even though internal overflows may occur.

- This implementation is very attractive as, in this case, the system does not need initial resetting and the effect of temporary miscalculations vanishes automatically from the output in a finite time.

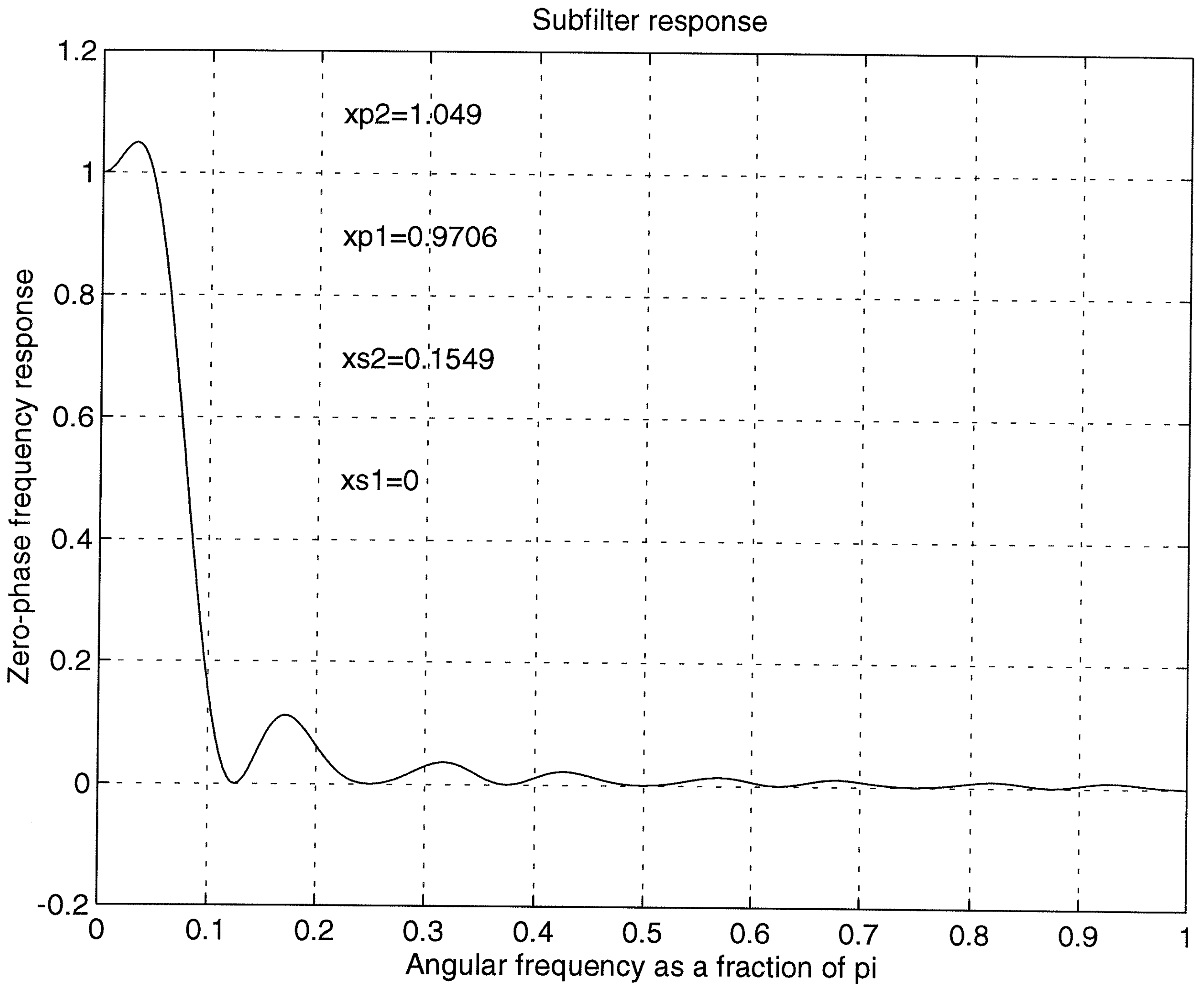
Optimization of the Additional Tap Coefficients

- By selecting $K = 16$, $P = 4$, $c = 2$, and $d = -2^{-1}$, the resulting subfilter requires no general multipliers and $F_M(\omega)$ varies within $x_{p1} = 0.9706$ and $x_{p2} = 1.0488$ on $[0, \omega_p]$ and within $x_{s1} = 0$ and $x_{s2} = 0.1549$ on $[\omega_s, \pi]$ (see Example 1 in transparency 6 as well as transparency 26).
- Using the above procedure, $\alpha = \beta = 0.5244$ and the edges of $G(\Omega)$ become $\Omega_p = 0.1761\pi$ and $\Omega_s = 0.7489\pi$ (see transparency 16).
- The minimum even-order $2N$ to meet the resulting criteria is 8 so that the required number of subfilters is $N = 4$. See transparency 27.
- The impulse-response coefficients for this prototype filter $G(z)$ are $\widehat{g}[0] = \widehat{g}[8] = -0.01875$, $\widehat{g}[1] = \widehat{g}[7] = -0.03892$, $\widehat{g}[2] = \widehat{g}[6] = 0.05492$, $\widehat{g}[3] = \widehat{g}[5] = 0.28811$, $\widehat{g}[4] = 0.42547$. See transparency 28.
- The zeros of this filter are located at $z = 0.3741$,

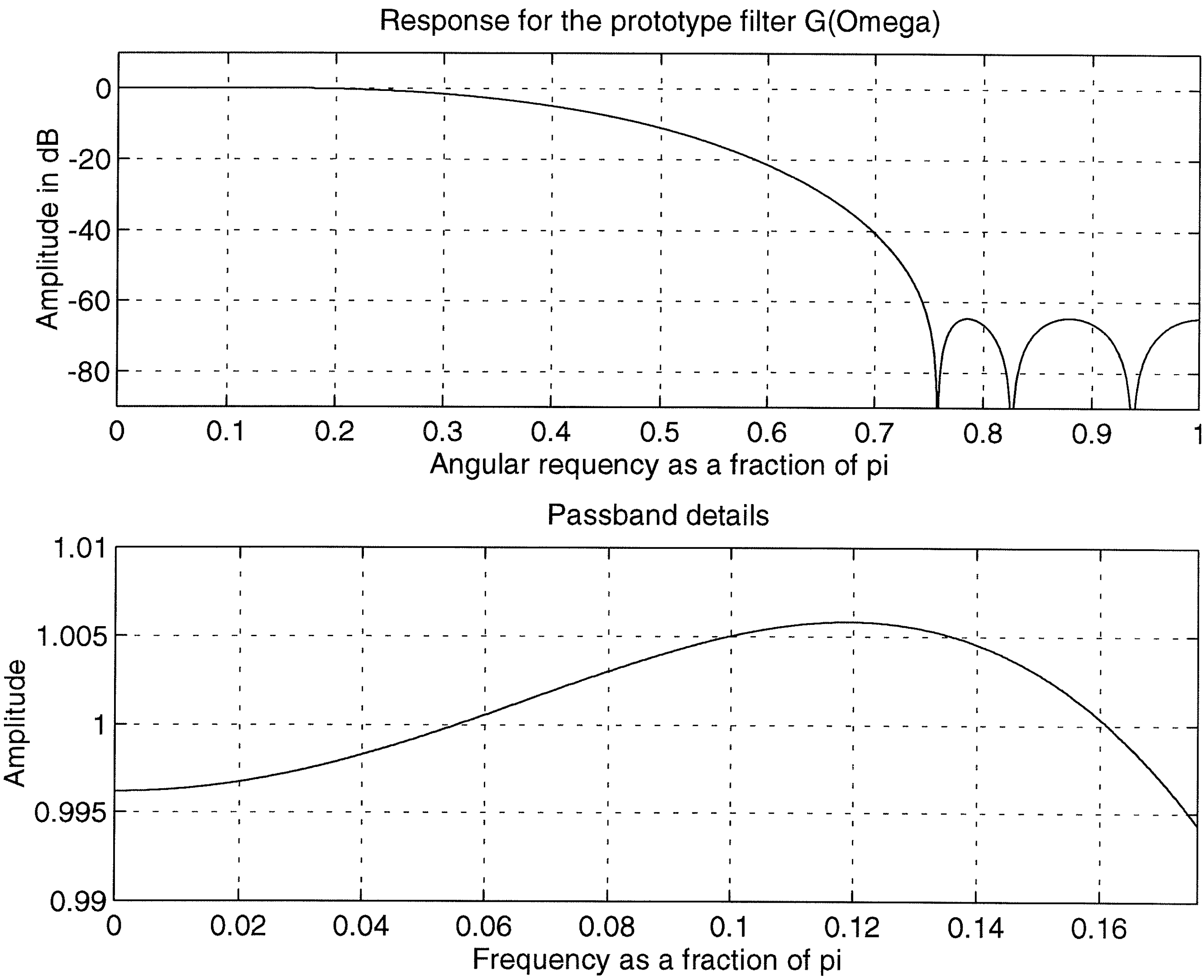
1/0.3741 (reciprocal zero pair on the unit circle), $z = e^{\pm 0.7582\pi}$, $z = e^{\pm 0.8265\pi}$, and $z = e^{\pm 0.9379\pi}$. See transparency 29.

- In this case, $N_1 = 0$, $N_2 = 4$, $\widehat{C} = -0.3001$, $\widehat{c}_1[0] = -1.5237$, $\widehat{c}_2[0] = 0.7251$, $\widehat{c}_2[0] = 0.8551$, and $\widehat{c}_1[0] = 0.9810$,
- The corresponding polynomial $P(x)$ contains thus only first-order sections.
- By fixing $c_k[1] = 1$ for $k = 1, 2, 3, 4$ to take the value of unity (like in the procedure described on pages 17–21), C and the coefficients $c_k[0] = 1$ for $k = 1, 2, 3, 4$ take the infinite-precision values shown in the table of transparency 30.

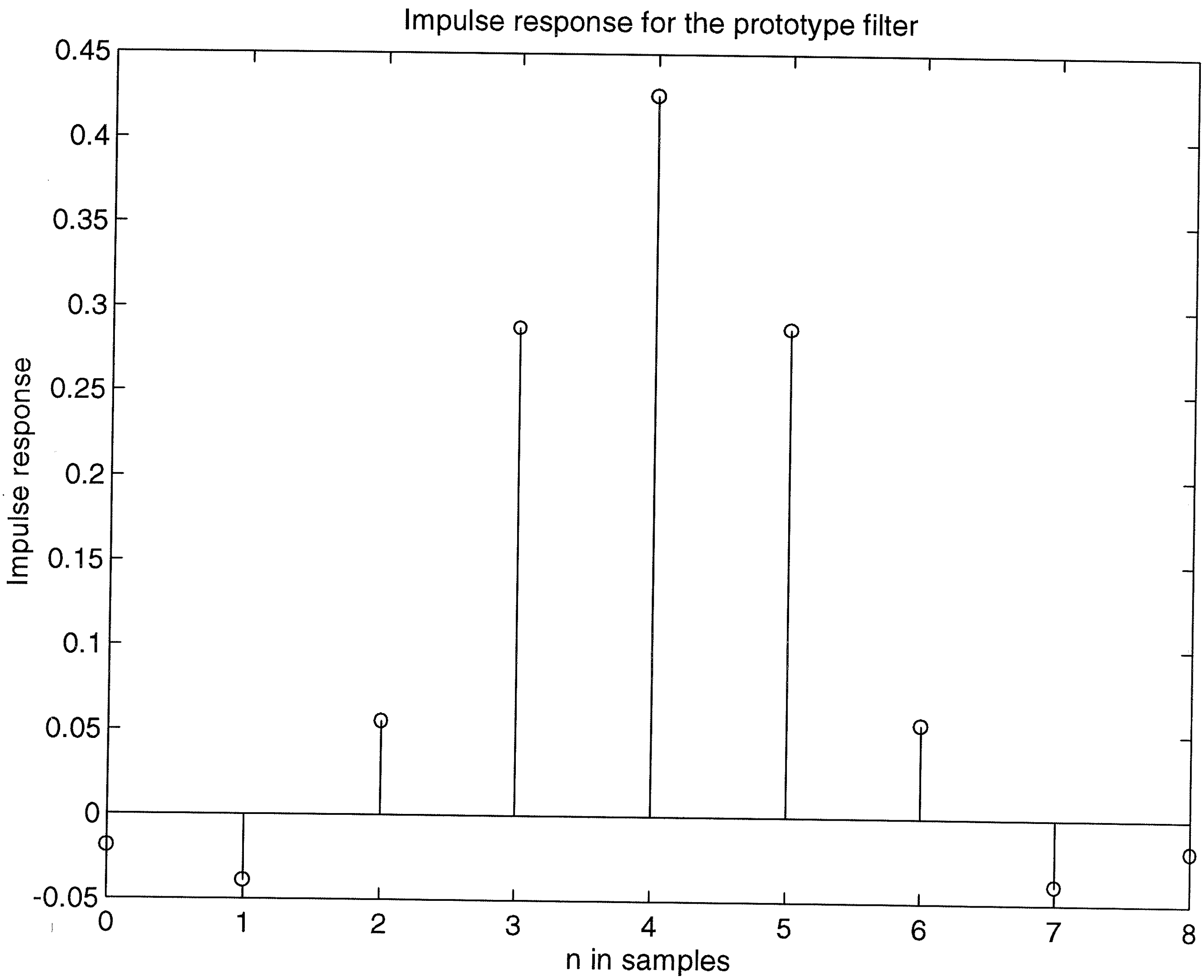
Zero-phase frequency response for the example subfilter



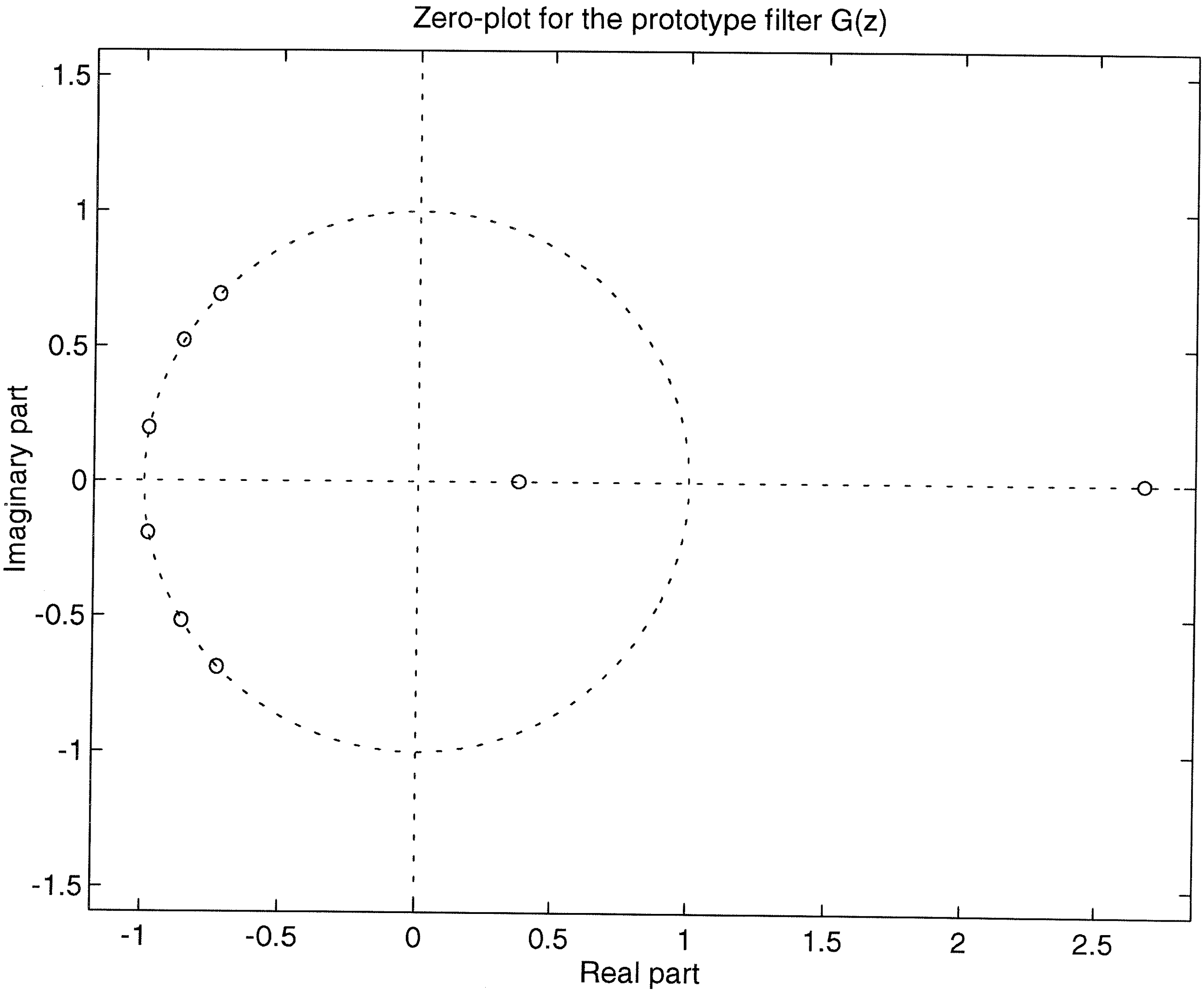
Amplitude response for the FIR prototype filter $G(z)$



Impulse response for the FIR prototype filter $G(z)$



Zero-plot for the FIR prototype filter $G(z)$



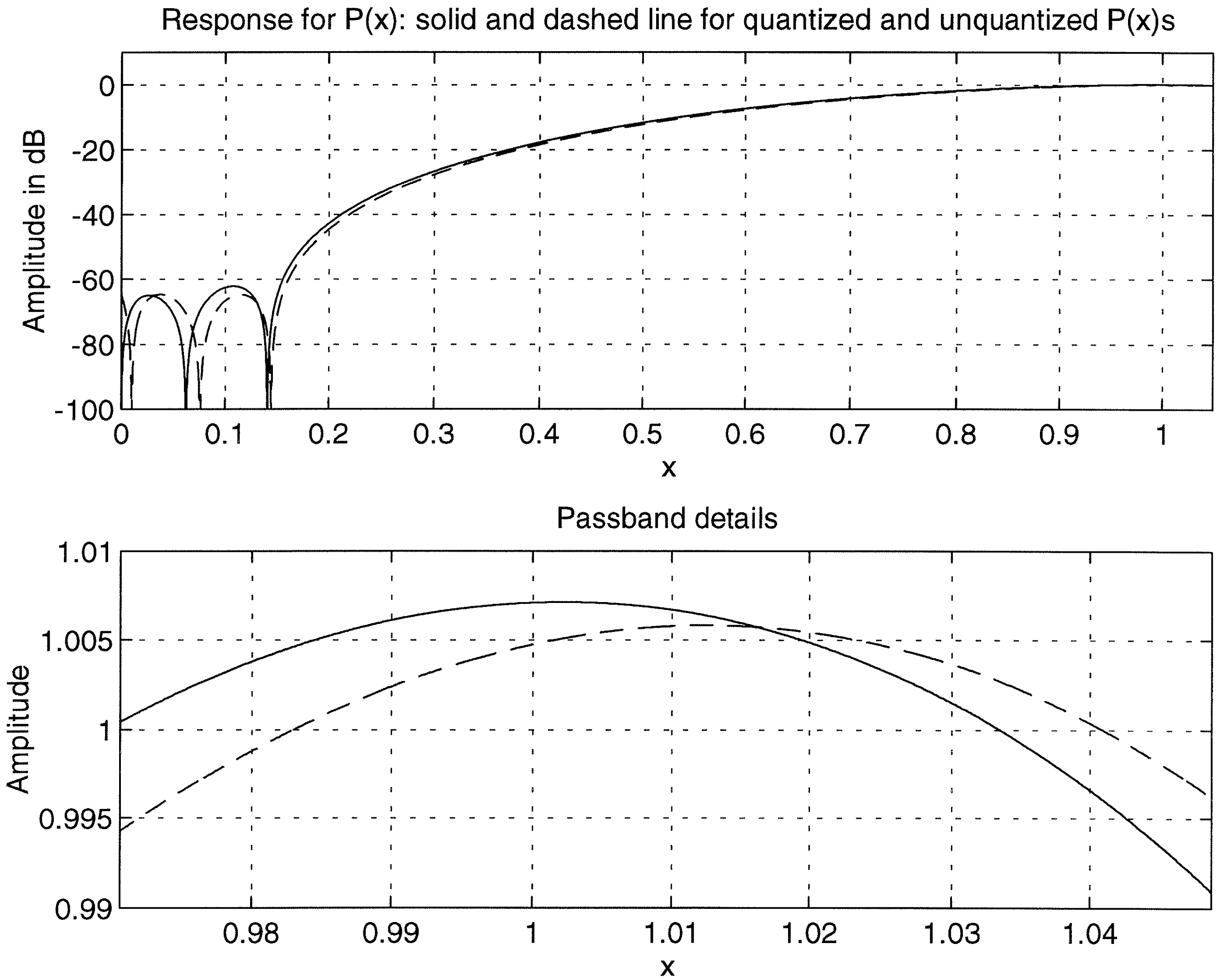
Infinite and Quantized Tap Coefficients

TABLE 4-11 Tap Coefficients for the Filter of Example 4.19

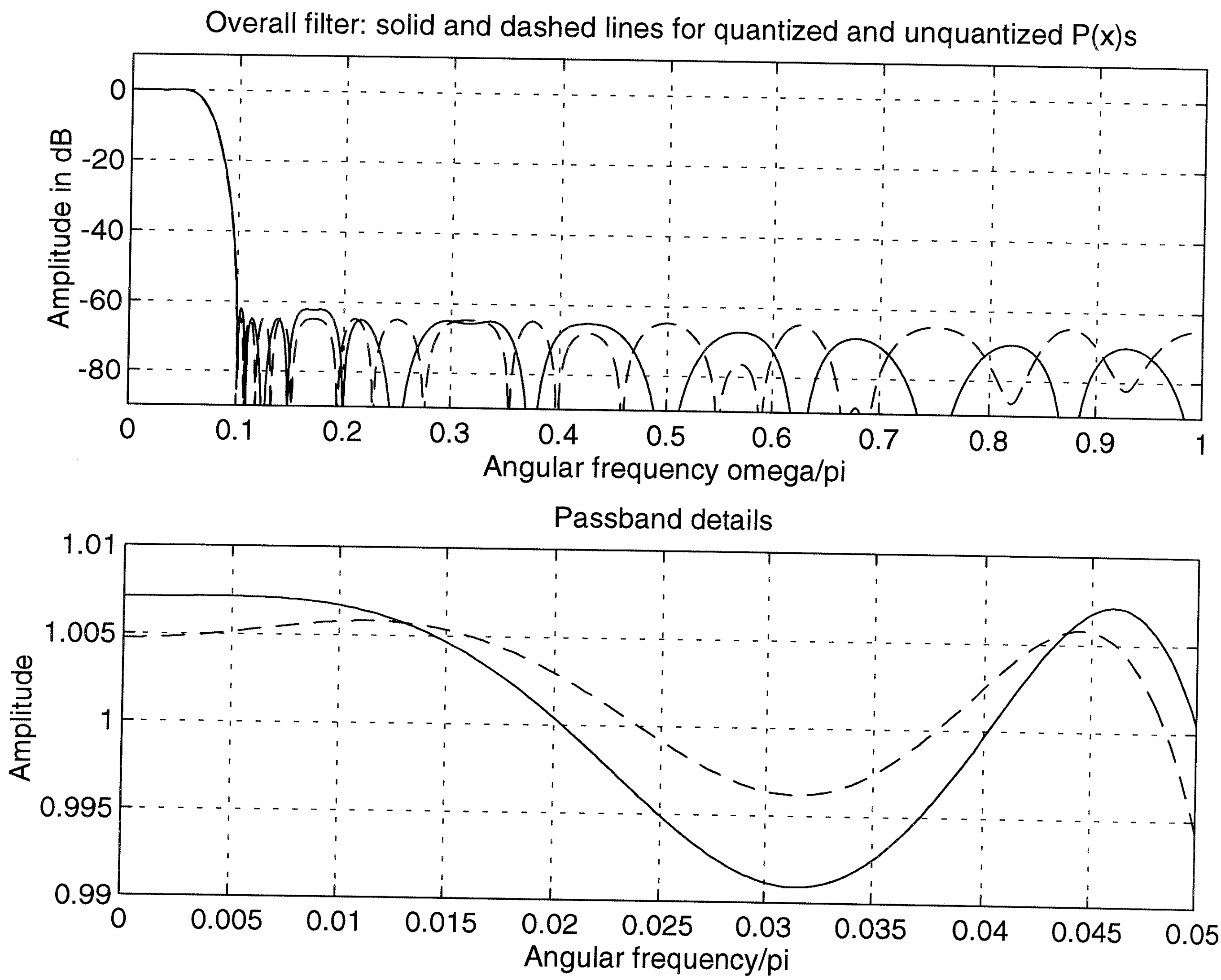
Infinite-Precision	Coefficients
$c_1[1] = 1$	$c_1[0] = -0.009995$
$c_2[1] = 1$	$c_2[0] = -0.075844$
$c_3[1] = 1$	$c_3[0] = -0.144123$
$c_4[1] = 1$	$c_4[0] = -1.323373$
$C = -3.967595$	
Quantized	**Coefficients**
$c_1[1] = 1$	$c_1[0] = 0$
$c_2[1] = 1$	$c_2[0] = -2^{-4}$
$c_3[1] = 1$	$c_3[0] = -2^{-3} - 2^{-6}$
$c_4[1] = 1$	$c_4[0] = -2^{0} - 2^{-2} - 2^{-4}$
$C = -2^{2}$	

- The given criteria are still met when these coefficients are quantized to the easily implementable values shown in the table of the previous transparency. See transparencies 32 and 33.
- The resulting composite filter requires no general multiplications, making it very useful for hardware or VLSI implementation.
- The responses of transparency 6 are for this overall design.
- In the following transparencies you can find a Matlab-file, called exa19.m, for performing the above synthesis (study it carefully and try it).

Responses for the quantized and unquantized $P(x)$. Stopband region is $[0,\ 0.1549]$ and the passband region is $[0.9706,\ 1.0488]$.



Responses for the overall filter with quantized and unquantized additional tap coefficients



```
%Matlab-file exa19.m for solving the problem
%4.19 in T. Saram"aki "Finite impulse response
%filter design" in Handbook for Digital Signal
%Processing, edited by S. K. Mitra and J. F. Kaiser,
%John Wiley & Sons, 1993.
%
%can be found in SUN's: ~ts/matlab/sldsp/exa19.m
%
%Identical FIR Subfilters
%
rrs=ones(1,16)/16;
a(1)=-1/2;a(17)=-1/2;
a(9)=2;
f=conv(rrs,rrs);
f=conv(f,a);
%
%Passband and stopband maxima and minima
%of the zero phase frequency response
%
[H,z]=zeroam(f,.0,.05,4000);
xp2=max(H);xp1=min(H);
[H,z]=zeroam(f,.1,1.,4000);
xs2=max(H);xs1=min(H);
[H,z]=zeroam(f,.0,1.,4000);
figure(1)
plot(z/pi,H);grid;title('Subfilter response');
ylabel('Zero-phase frequency response');
xlabel('Angular frequency as a fraction of pi');
text(.22,1.1,['xp2=',num2str(xp2)])
text(.22,.9,['xp1=',num2str(xp1)])
text(.22,.7,['xs2=',num2str(xs2)])
text(.22,.5,['xs1=',num2str(xs1)])
u(1)=.0,y(1)=0;
%Determine G(Omega); alpha and beta from
%equation (4.196), Omega_p and Omega_s
%from equation (4.197)
figure(2)
alpha=(xp2-xs1)/2;beta=(xp2+xs1)/2;
den=xp2-xs1;
Omegap=acos((2*xp1-xp2-xs1)/den)/pi;
Omegas=acos((2*xs2-xp2-xs1)/den)/pi;
plot(u,y);axis([0 100 0 100]);
```

```
text(20,80,['alpha=',num2str(alpha)])
text(20,60,['beta=',num2str(beta)])
text(20,40,['Omegap=',num2str(Omegap),'pi'])
text(20,20,['Omegas=',num2str(Omegas),'pi'])
%Minimum even order of G(z) to meet the resulting
%criteria is 8, that is, we need 4 subfilters
g=remez(8,[0 Omegap Omegas 1], [1 1 0 0],[1 10]);
[HH,z]=zeroam(g,.0,1.,4000);
figure(3);subplot(211)
plot(z/pi,20*log10(abs(HH)));axis([0 1 -90 10]); grid;
title('Response for the prototype filter G(Omega)');
xlabel('Angular requency as a fraction of pi');
ylabel('Amplitude in dB')
%Passband details
subplot(212);plot(z/pi,(HH)),grid;axis([0 Omegap .99 1.01]);
title('Passband details');
xlabel('Frequency as a fraction of pi'); ylabel('Amplitude')
figure(4)
impz(g);
title('Impulse response for the prototype filter');
xlabel('n in samples'); ylabel('Impulse response')
figure(5)
zplane(g);
title('Zero-plot for the prototype filter G(z)');
%Express G(z) in the form
%G(z)=g(0)[1+a1*z^(-1)+z^(-2)]*[1+a2*z^(-1)+z^(-2)]*
%[1+a3*z^(-1)+z^(-2)]*[1+a4*z^(-1)+z^(-2)]
%How to do this? g(0) is value of the impulse response
%of G(z) at n=0. In Matlab g(1). The zeros of G(z) are located at
%z=r, 1/r, with r_1=0.3741, and z=exp(+-theta_k)
%for k=2,3,4 with theta_2=0.8265pi, theta_2=0.8265pi,
%theta_4=0.8265pi,
%Then, a1=-(r+1/r), a_k=-2cos(theta_k), k=2,3,4
zer=roots(g);
[Y I]=sort(-real(zer));
zer=zer(I);
%gives the following zeros
%zer =
%
%   2.6733
%   0.3741
%  -0.7251 + 0.6886i
```

```
% -0.7251 - 0.6886i
% -0.8551 + 0.5185i
% -0.8551 - 0.5185i
% -0.9810 + 0.1938i
% -0.9810 - 0.1938i
r=zer(2);
a1=-(r+1/r);
theta2=acos(real(zer(3)))/pi;
a2=-2*cos(pi*theta2);
theta3=acos(real(zer(5)))/pi;
a3=-2*cos(pi*theta3);
theta4=acos(real(zer(7)))/pi;
a4=-2*cos(pi*theta4);
%Factorized zero-phase response G(Omega) is obtained
%by using the substitution 1+z^(-2)==2*cos(Omega), and
%z^(-1)==1. Note that the delay term is totally diregarded.
%We obtain G(Omega)=G0[cos(Omega)+G1]*[cos(Omega)+G2]*
%[cos(Omega)+G3]*[cos(Omega)+G4], where G0=8g(0) and
%G_k=a_k/2 for k=1,2,3,4.
G0=(2^4)*g(1); %-0.3001
G1=a1/2; % -1.5237
G2=a2/2; % 0.7251
G3=a3/2; % 0.8551
G4=a4/2; % 0.9810
%Now the desired P(x)=G([x-beta/alpha]) [equation (4.199)], giving
%P(x)=P0[x+P1]*[x+P2]*[x+P3]*[x+P4], where P0=G0/(alpha^4),
%Pk=alpha*Pk-beta for k=1,2,3,4.
P0=G0/(alpha^4); %-3.9674
P1=alpha*G1-beta; %-1.3234
P2=alpha*G2-beta; %-0.1441
P3=alpha*G3-beta; %-0.0760
P4=alpha*G4-beta; %-0.0099
%
%
figure(6)
plot(u,y);axis([0 100 0 100]);
text(3,95,'zeros of G(z) are located at z=0.3741, 1/0.3741')
text(3,87.5,'z=exp(+-j0.7582pi), z=exp(+-j0.8265pi),z=exp(+-
j0.9379pi)')
text(3,80,'G(z)=-0.0188*[1-3.0473z^(-1)+z^(-2)]*[1+1.4503z^(-
1)+z^(-2)]*')
text(13,72.5,'[1+1.7107z^(-1)+z^(-2)]*[1+1.9619z^(-1)+z^(-2)]')
```

```
text(3,60,'G(Omega)=-0.3001[cos(Omega)-
1.5237]*[cos(Omega)+0.7251]*')
text(13,52.5,'[cos(Omega)+0.8551]*[cos(Omega)+0.9810]')
text(3,45,'P(x)=-3.967595*[x-1.3234]*[x-0.1441]*')
text(13,37.5,'[x-0.0760]*[x-0.0099]')
text(3,30,'Qunatized P(x)=-2^2*[x-2^0-2^(-2)-2^(-4)]*[x-2^(-3)-2^(-
6)]*')
text(13,22.5,'[x-2^(-4)]*[x-0]')
x=(.0:.0001:1.0488);
h1=P0*(x+P1);h2=(x+P2);h3=(x+P3);h4=(x+P4);
hh1=h1.*h2;
hh1=hh1.*h3;
hh1=hh1.*h4;
%Simple additional tap coefficients
h1=-4*(x-1-2^(-2)-2^(-4));h2=(x-0);h3=(x-2^(-4));h4=(x-2^(-3)-2^(-6));
hh2=h1.*h2;
hh2=hh2.*h3;
hh2=hh2.*h4;
figure(7);
subplot(211)
plot(x,20*log10(abs(hh1)),'- -',x,20*log10(abs(hh2)));axis([xs1 xp2 -
100 10]);grid
title('Response for P(x): solid and dashed line for quantized and
unquantized P(x)s');
xlabel('x'); ylabel('Amplitude in dB')
%Passband details
subplot(212);plot(x,hh1,'- -',x,hh2),grid;axis([xp1 xp2 .99 1.01]);
title('Passband details');
xlabel('x'); ylabel('Amplitude')
%Overall responses
h1=P0*(H+P1);h2=(H+P2);h3=(H+P3);h4=(H+P4);
hh1=h1.*h2;
hh1=hh1.*h3;
hh1=hh1.*h4;
%Simple additional tap coefficients
h1=-4*(H-1-2^(-2)-2^(-4));h2=(H-0);h3=(H-2^(-4));h4=(H-2^(-3)-2^(-
6));
hh2=h1.*h2;
hh2=hh2.*h3;
hh2=hh2.*h4;
figure(8)
subplot(211)
```

```
plot(z/pi,20*log10(abs(hh1)),'- -',z/pi,20*log10(abs(hh2)));axis([0 1. -
100 10]);
axis([0 1 -90 10]);grid
title('Overall filter: solid and dashed lines for quantized and
unquantized P(x)s');
xlabel('Angular frequency omega/pi'); ylabel('Amplitude in dB')
%Passband details
subplot(212);plot(z/pi,hh1,'- -',z/pi,hh2),grid;axis([.0, .05 .99 1.01]);
title('Passband details');
xlabel('Angular frequency/pi'); ylabel('Amplitude')
```

Solution to Problem I

- Problem I can be solved by finding Ω_p and Ω_s for $G(\Omega)$ of the given even-order $2N$ in such a way that it meets

$$1 - \delta_p \leq G(\Omega) \leq 1 + \delta_p \quad \text{for} \quad 0 \leq \Omega \leq \Omega_p$$

$$-\delta_s \leq G(\Omega) \leq \delta_s \quad \text{for} \quad \Omega_s \leq \Omega \leq \pi.$$

and the subfilter criteria become as mild as possible so that they can be met by the minimum even order $2M$.

- For any $G(\Omega)$, the corresponding polynomial $P(x)$ is obtained using the substitution

$$\cos \Omega = [x - \beta]/\alpha,$$

where

$$\alpha = 1, \quad \beta = 0$$

for Case A and

$$\alpha = \frac{2}{2 + \cos \Omega_p - \cos \Omega_s}, \quad \beta = \frac{(\cos \Omega_s - 1)}{2 + \cos \Omega_p - \cos \Omega_s} \qquad (A)$$

for Case B.

- In Case A, the resulting passband and stopband region of $P(x)$ are $[x_{p1}, 1]$ and $[-1, x_{s2}]$, where

$$x_{s2} = \cos\Omega_s, \quad x_{p1} = \cos\Omega_p.$$

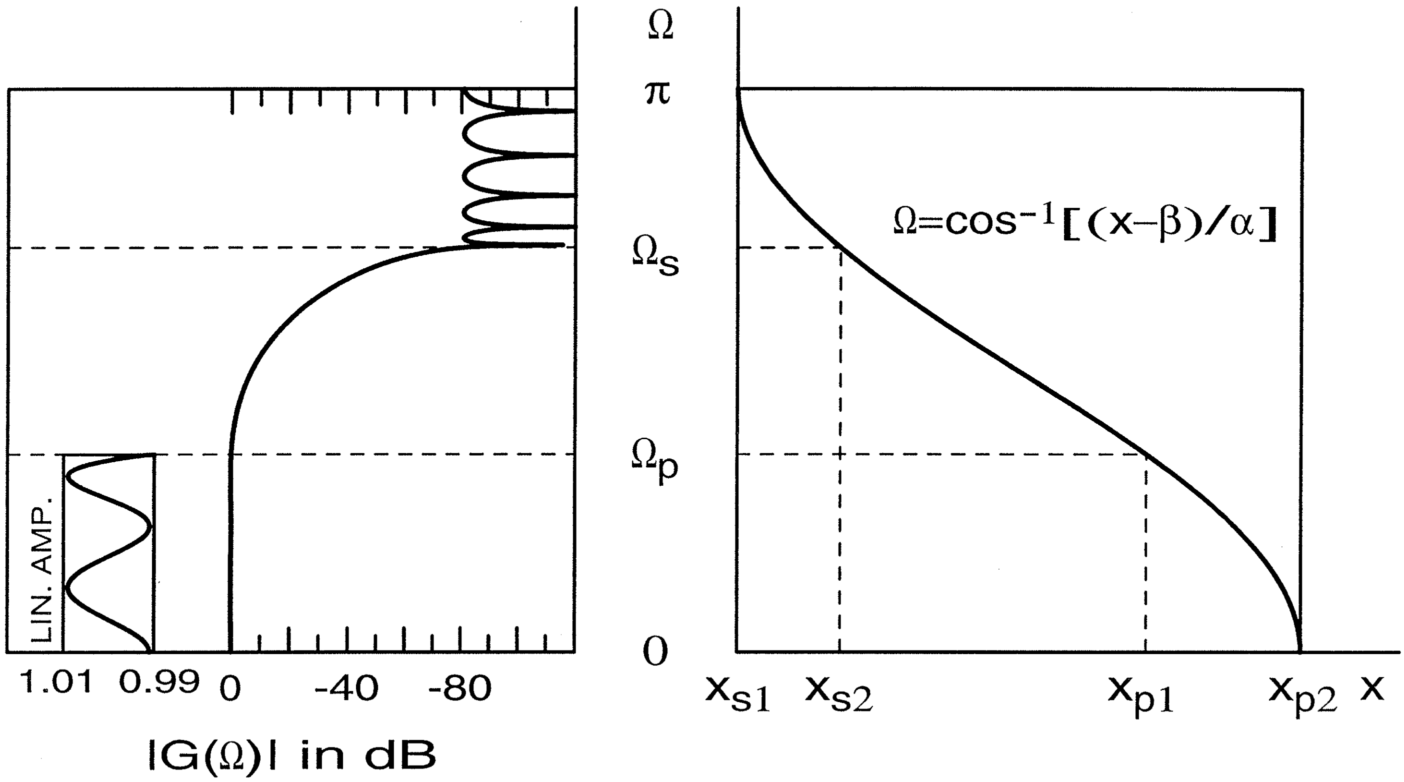
- In Case B, the corresponding regions are $[1 - \widehat{\delta}_p, 1 + \widehat{\delta}_p]$ and $[-\widehat{\delta}_s, \widehat{\delta}_s]$, where

$$\widehat{\delta}_p = \frac{1 - \cos\Omega_p}{2 + \cos\Omega_p - \cos\Omega_s}, \quad \widehat{\delta}_s = \frac{1 + \cos\Omega_s}{2 + \cos\Omega_p - \cos\Omega_s}.$$

- The figure in the following transparency exemplifies these relations.

- Note that $P(x)$ for Case B can be obtained from the Case A polynomial by replacing x by $[x - \alpha]/\beta$, where α and β are given by Eq. (A).

Relations of the Case A and Case B polynomials $P(x)$ to the best extraripple solution for the given values of δ_p and δ_s. $N = 8$, $\delta_p = 0.009$, $\delta_s = 0.0009$.



$\Omega_p = 0.3517\pi$, $\Omega_s = 0.7177\pi$

Case A:

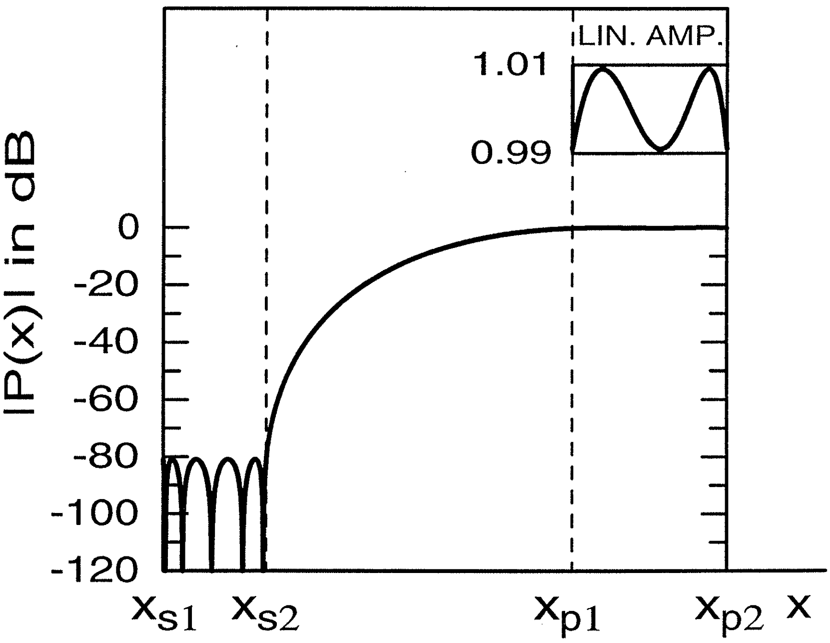
$x_{p1} = 0.4493$, $x_{p2} = 1.000$

$x_{s1} = -1.00$, $x_{s2} = -0.6318$

Case B:

$x_{p1}, x_{p2} = 1 \pm 0.1787$

$x_{s1}, x_{s2} = \pm 0.1195$



How to find the best solution fast?

- It has turned out that the mildest subfilter criteria are typically obtained at those values of Ω_p and Ω_s for which $G(\Omega)$ has an extraripple solution (one extrema more than required by the characterization theorem) for the specified values of δ_p and δ_s (see the lecture notes on Digital Filtering II, especially the handbook chapter).
- The best extraripple solution is the one for which Ω_p and $\pi - \Omega_s$ are the most equal.
- As an example, the figure in the previous transparency gives the best extraripple solution for $N = 8$, $\delta_p = 0.009$, and $\delta_s = 0.00009$ along with the corresponding polynomials $P(x)$ in Cases A and B.
- Note that the allowable passband and stopband variations for the subfilter are in both cases huge compared to those of the overall design.

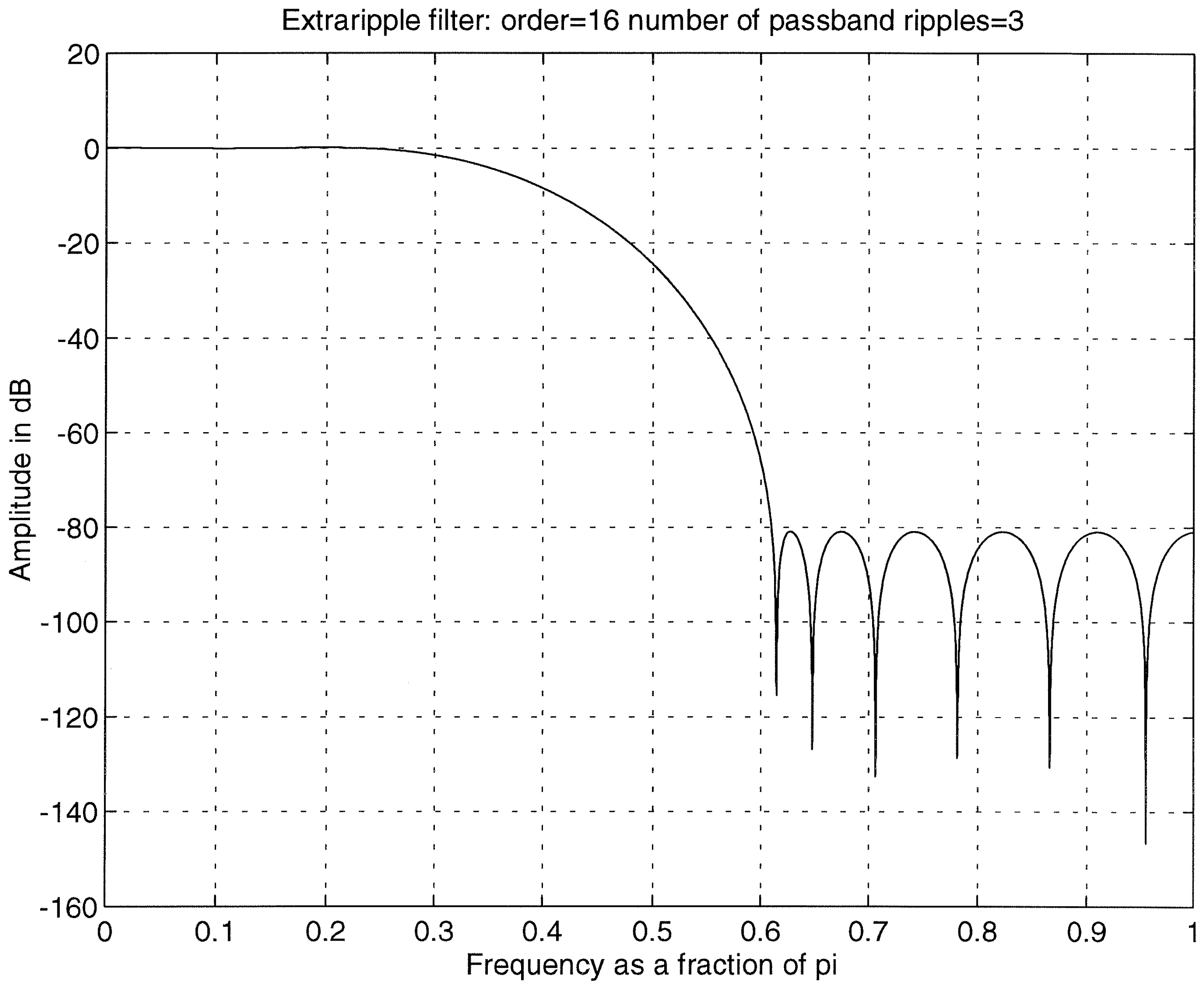
- The solutions of transparency 36 are used later as a starting point for synthesising multiplier-free filters for $\delta_p = 0.01$ and $\delta_s = 0.0001$.
- The desired extraripple solutions can be found directly using the algorithm of Hofstetter, Oppenheim, and Siegel.
- This algorithm can also be implemented by slightly modifying the Mclellan-Parks-Rabiner algorithm.
- Later on, we shall introduce a Matlab-file, called extralin.m, which automatically performs all the work for us.

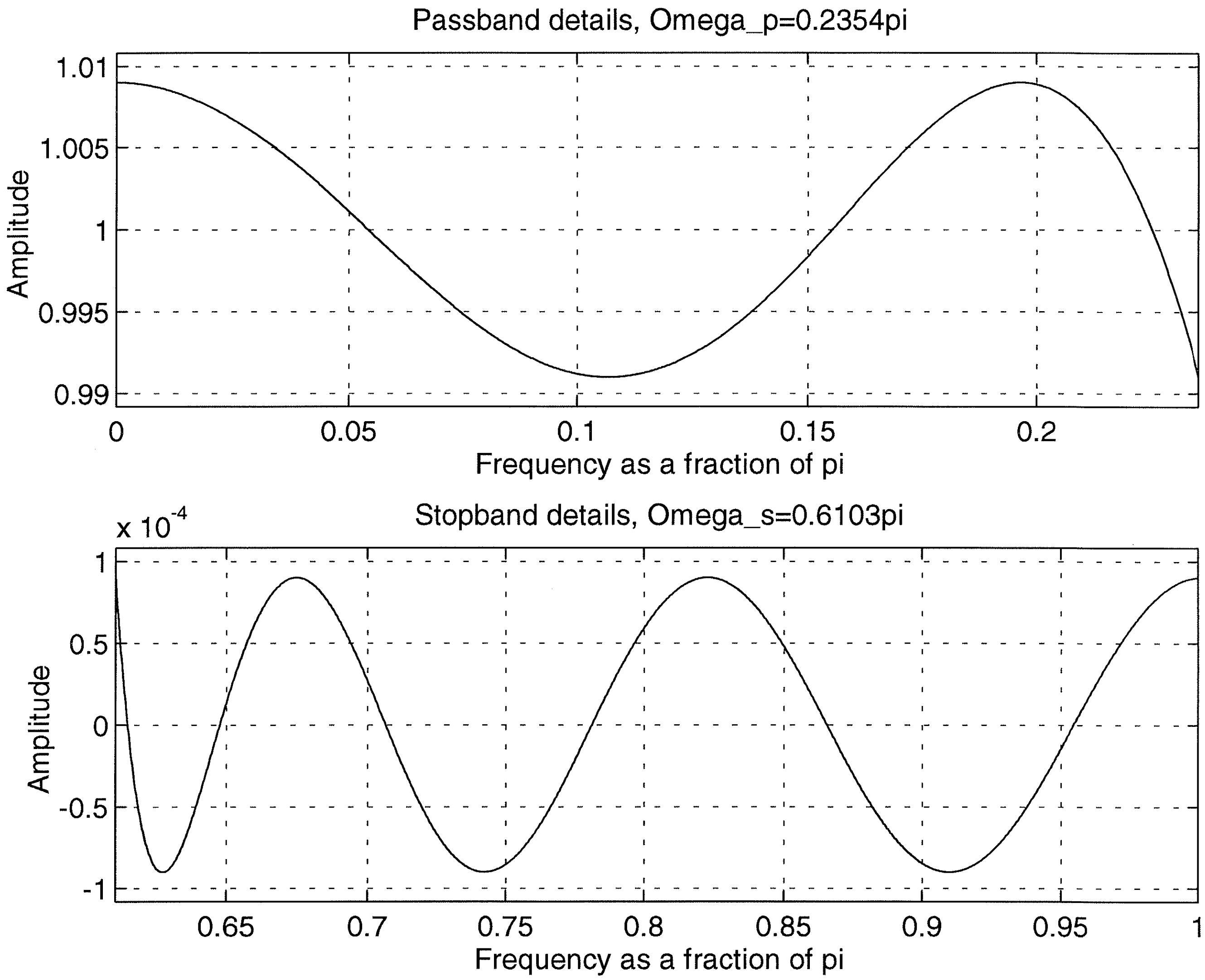
The best extraripple solution for $N = 8$, $\delta_p = 0.009$, and $\delta_s = 0.00009$

- In the following, there are three sets of 8 transparencies for different extraripple solutions.
- These solutions are characterized by the fact that they oscillate in both the passband and stopband just between the given limits ($1 \pm \delta_p$ in the passband and $\pm\delta_p$ in the stopband).
- For these filters, the passband edge Ω_p (the stopband edge Ω_s) is determined as the (first) last frequency point where the value $1 - \delta_p$ (δ_p) is achieved.
- When these edge points are included in the number of extremal points, these designs have one more extremal points than required by the characterization theorem of the best solution.
- When counting the number of passband ripples for the extraripple filters, the cutoff point Ω_p is not included.
- For each selection of the number of ripples in the

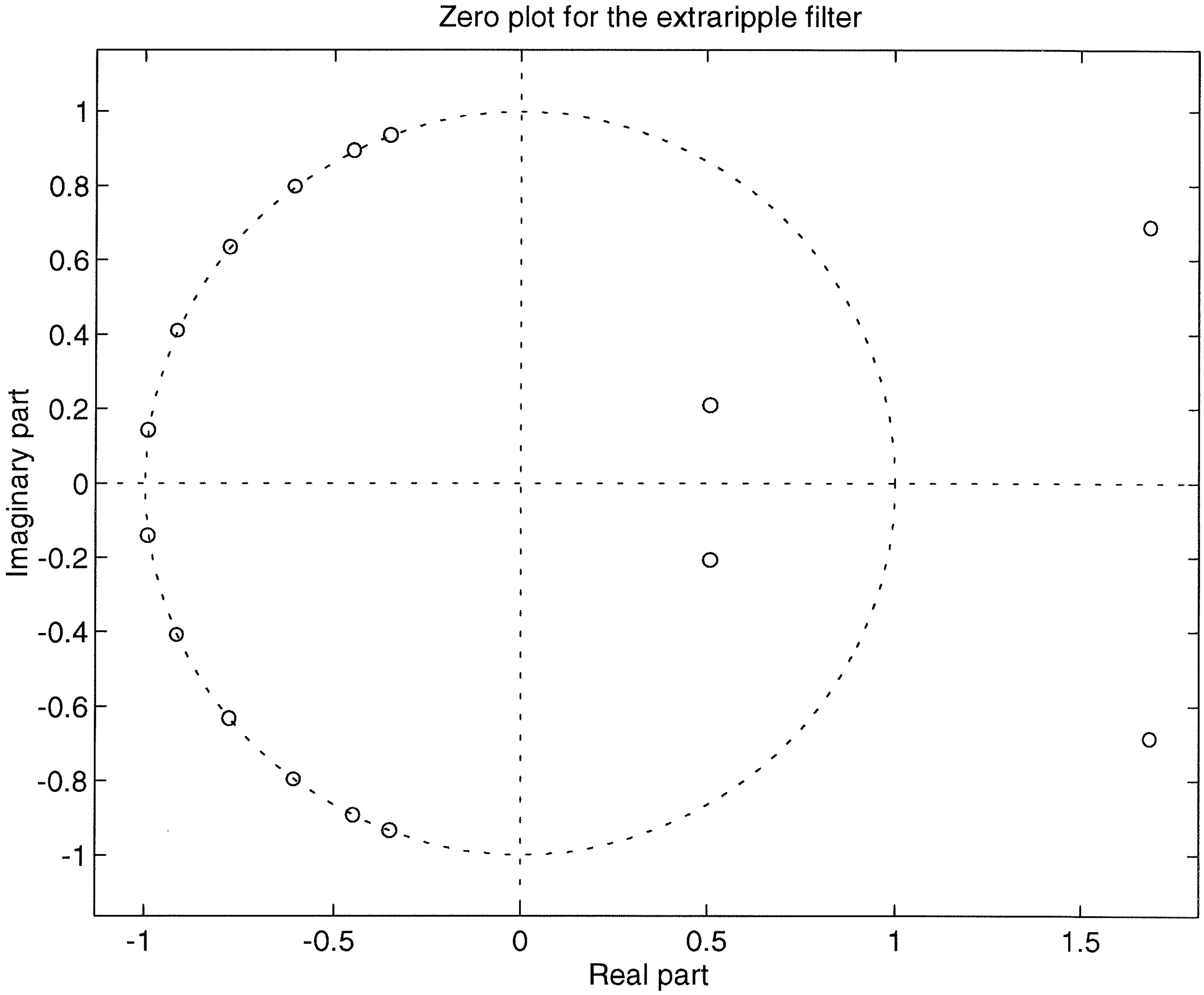
passband, there is a unique solution. The number of passband ripples can be selected between unity and N for a filter of order $2N$.

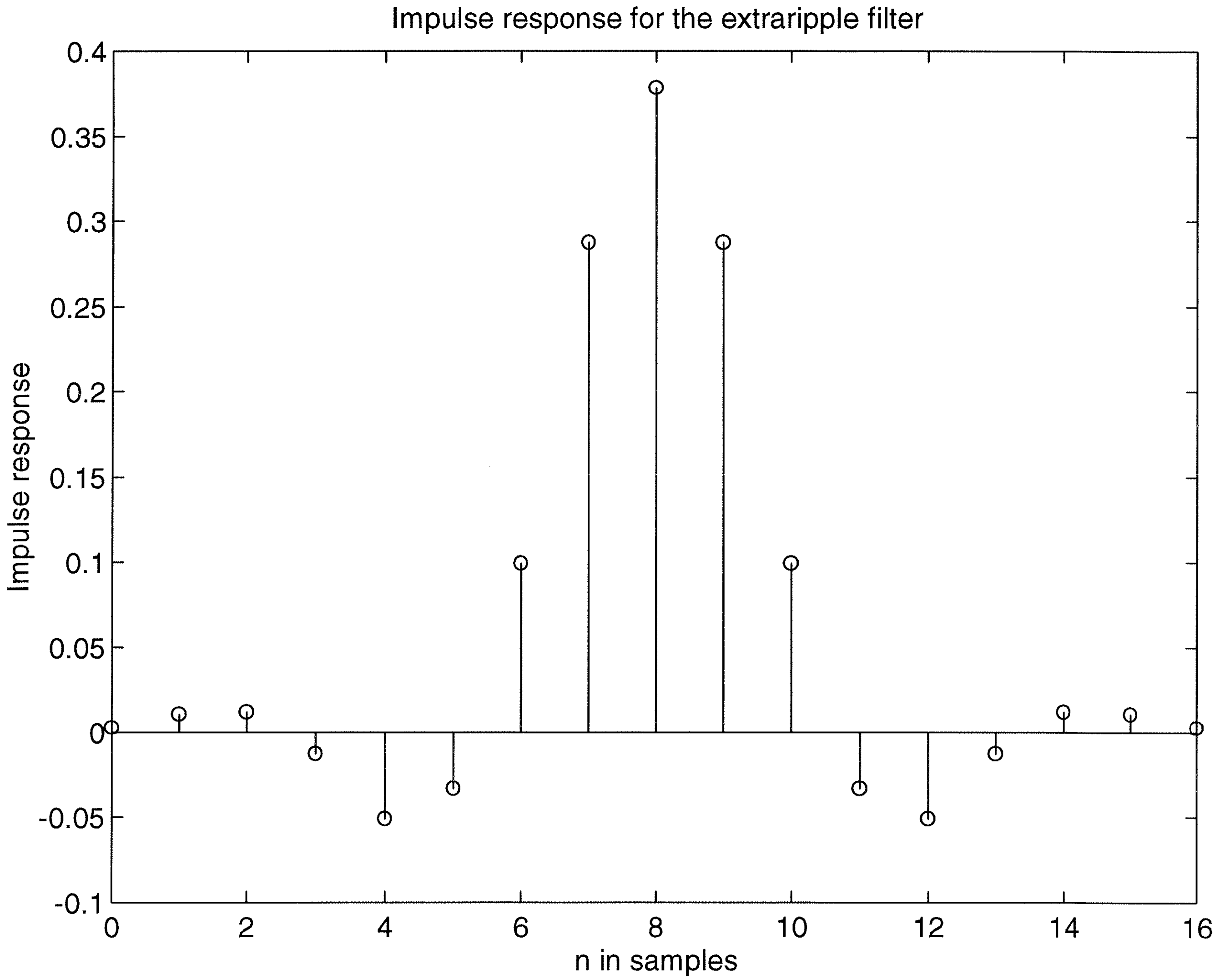
- The first, second, and third sets in the following are for the cases with three, four, and five passband ripples.
- For these cases, $\Omega_p = 0.2354\pi$ and $\Omega_s = 0.6103\pi$; $\Omega_p = 0.3571\pi$ and $\Omega_s = 0.7177\pi$; and $\Omega_p = 0.4646$ and $\Omega_s = 0.8167\pi$.
- Hence, for the case with four passband ripples, Ω_p and $\pi - \Omega_s$ are the most equal.


Extraripple filter: order=16 number of passband ripples=3
20
0
-20
-40
-60
-80
-100
-120
-140
-160
Amplitude in dB
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Frequency as a fraction of pi

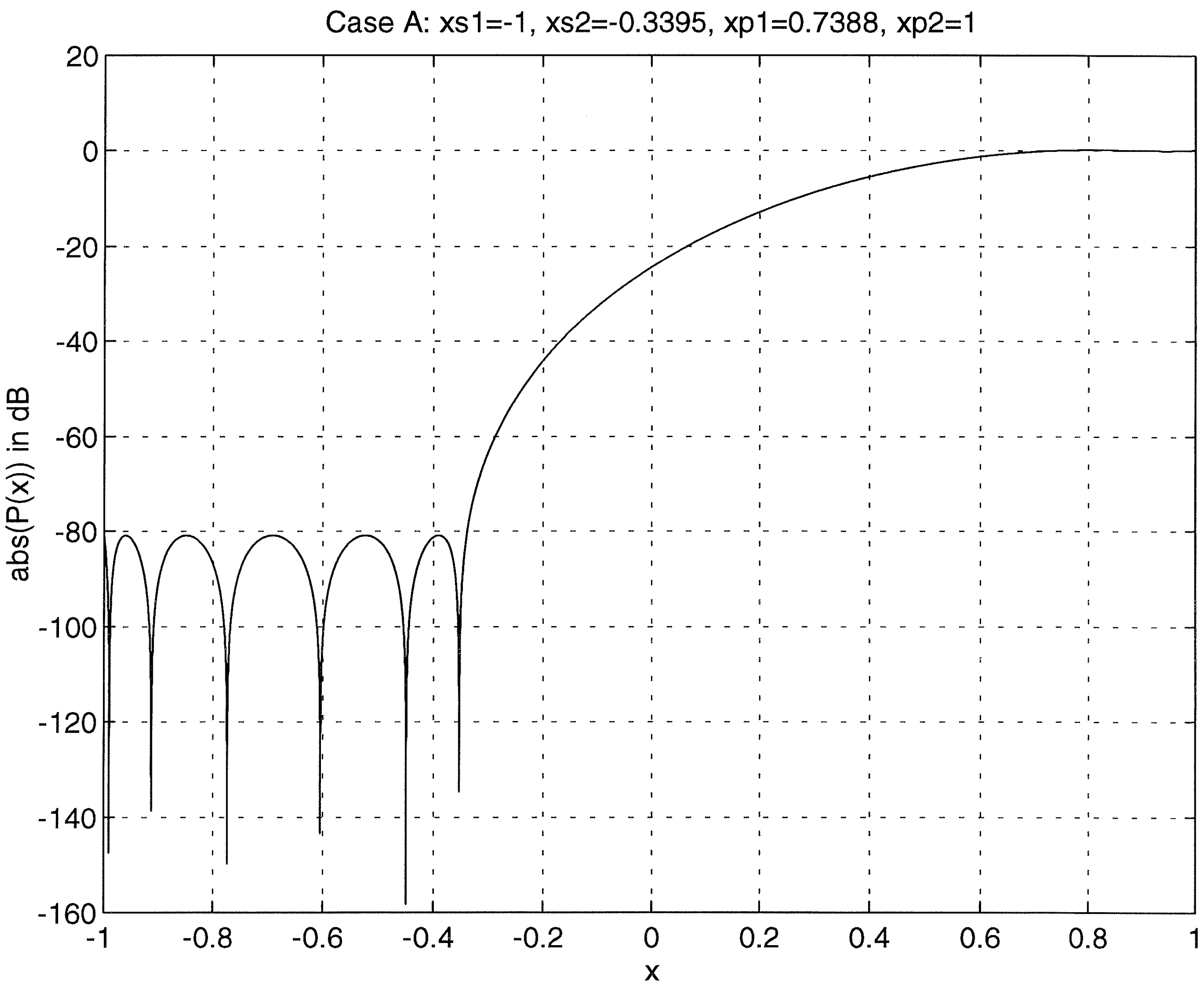


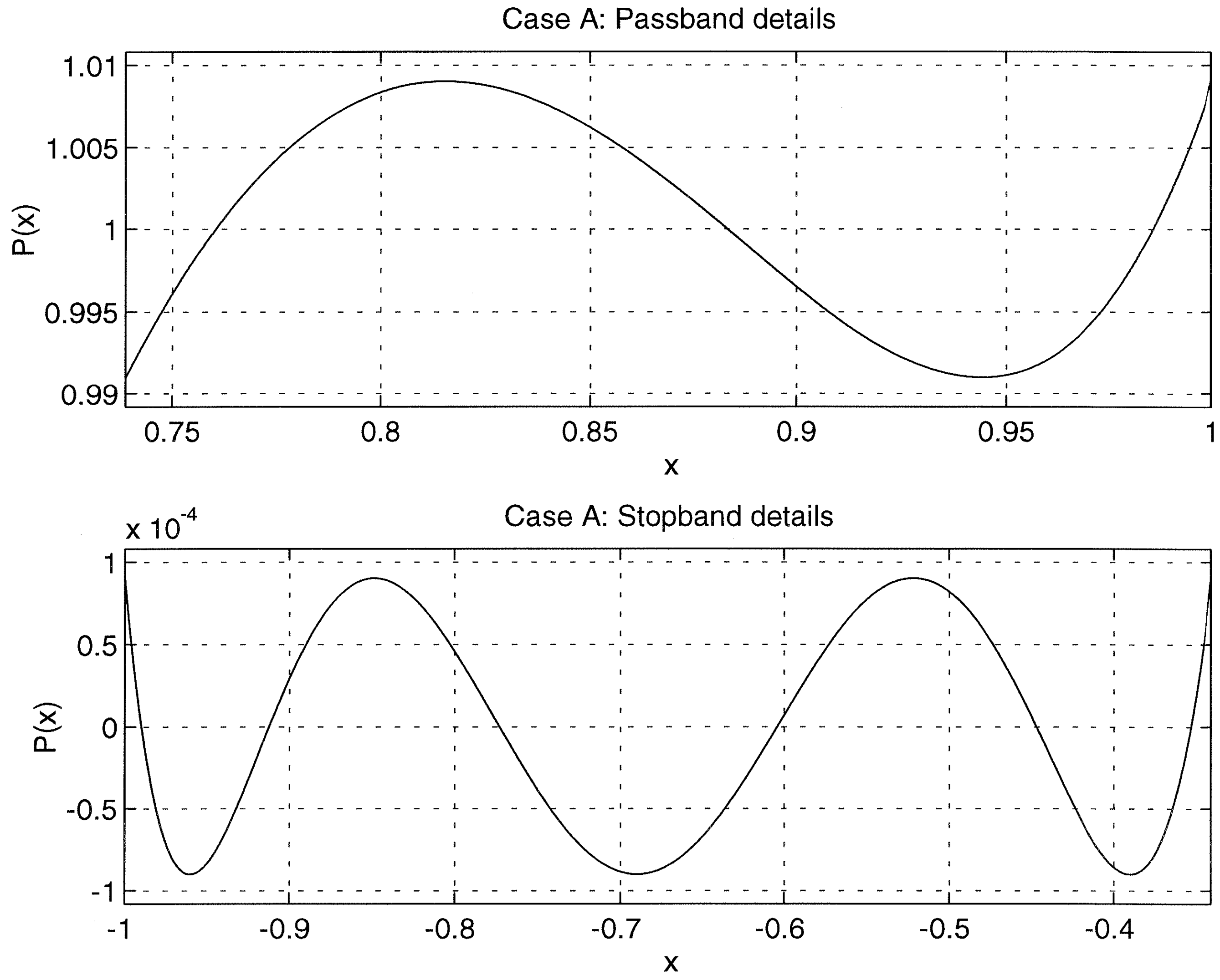
Passband details, Omega_p=0.2354pi
1.01
1.005
1
0.995
0.99
Amplitude
0
0.05
0.1
0.15
0.2
Frequency as a fraction of pi
Stopband details, Omega_s=0.6103pi
x 10^-4
1
0.5
0
-0.5
-1
Amplitude
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1
Frequency as a fraction of pi


Zero plot for the extraripple filter
Imaginary part
Real part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1
1.5

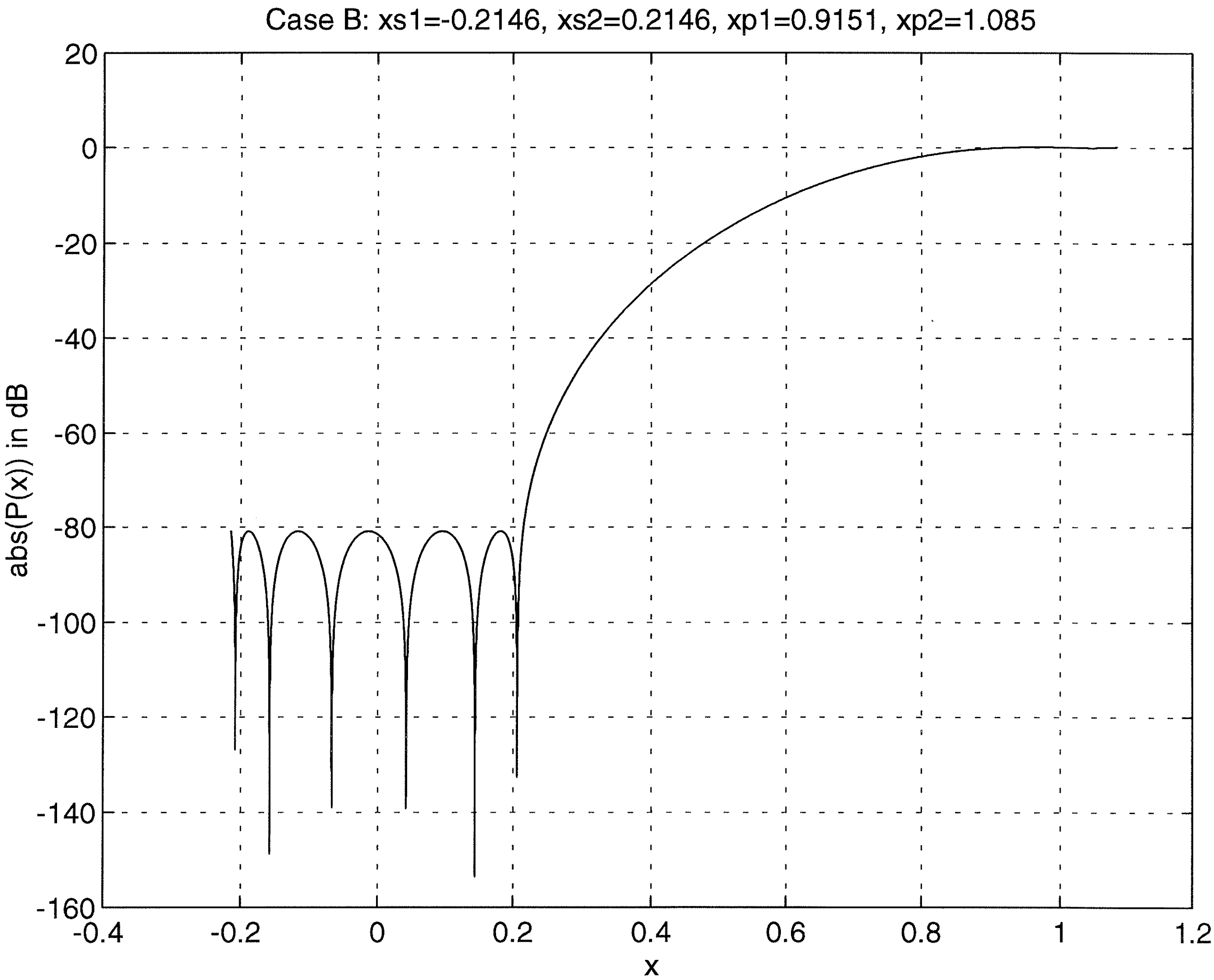


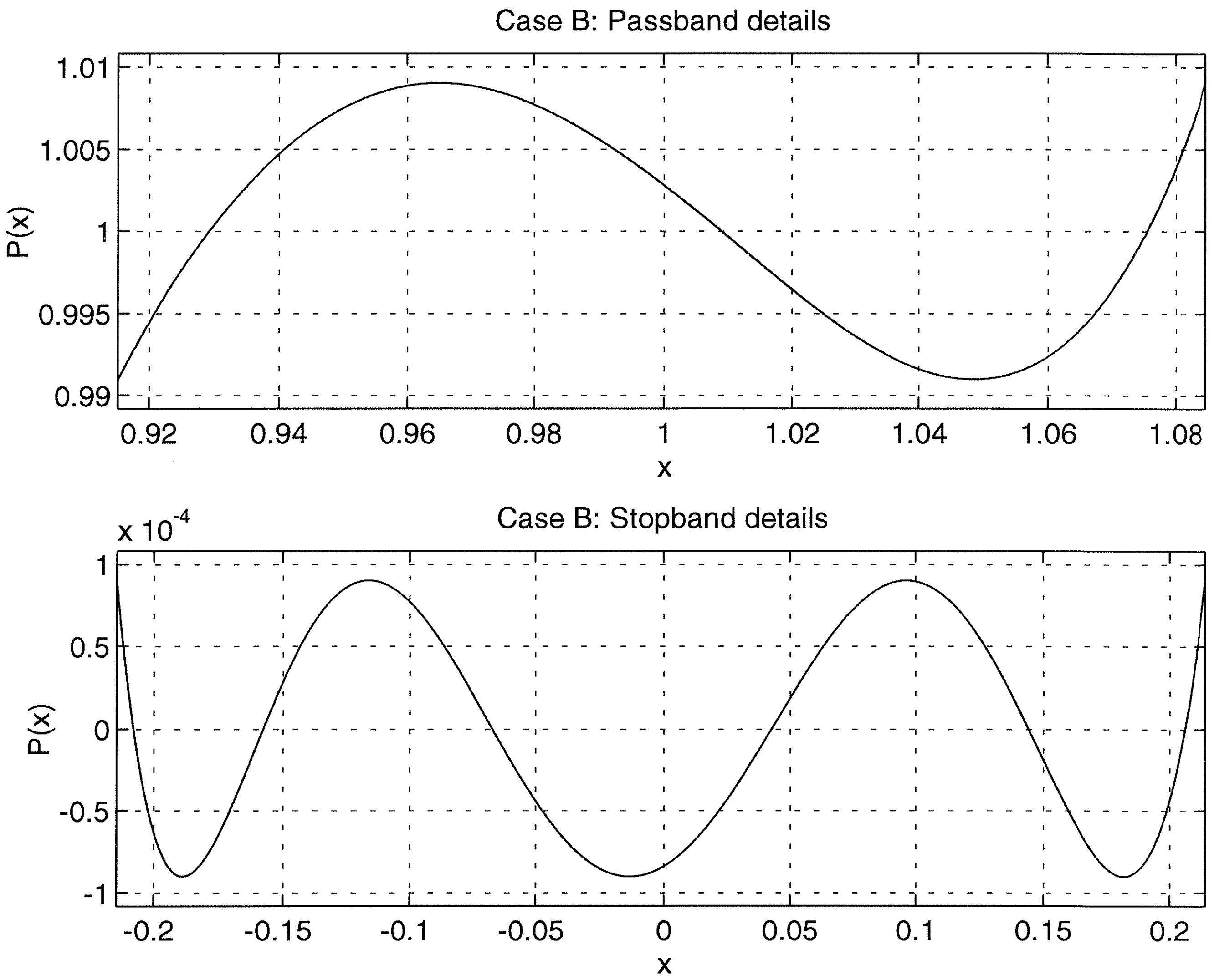
Impulse response for the extraripple filter
Impulse response
n in samples
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
0
2
4
6
8
10
12
14
16


Case A: xs1=-1, xs2=-0.3395, xp1=0.7388, xp2=1
abs(P(x)) in dB
20
0
-20
-40
-60
-80
-100
-120
-140
-160
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
x

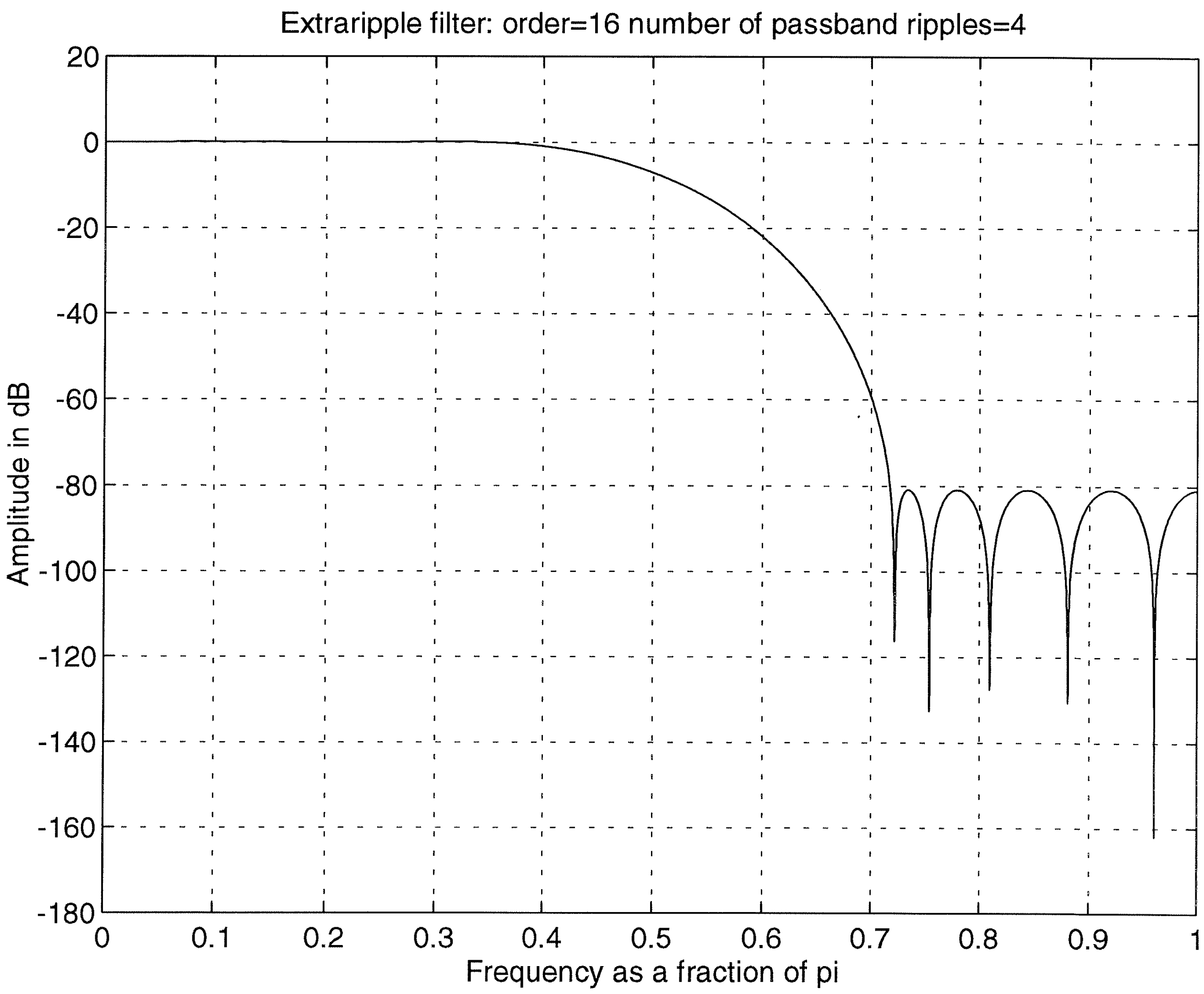


Case A: Passband details
1.01
1.005
1
0.995
0.99
P(x)
0.75
0.8
0.85
0.9
0.95
1
x
Case A: Stopband details
x 10-4
1
0.5
0
-0.5
-1
P(x)
-1
-0.9
-0.8
-0.7
-0.6
-0.5
-0.4
x

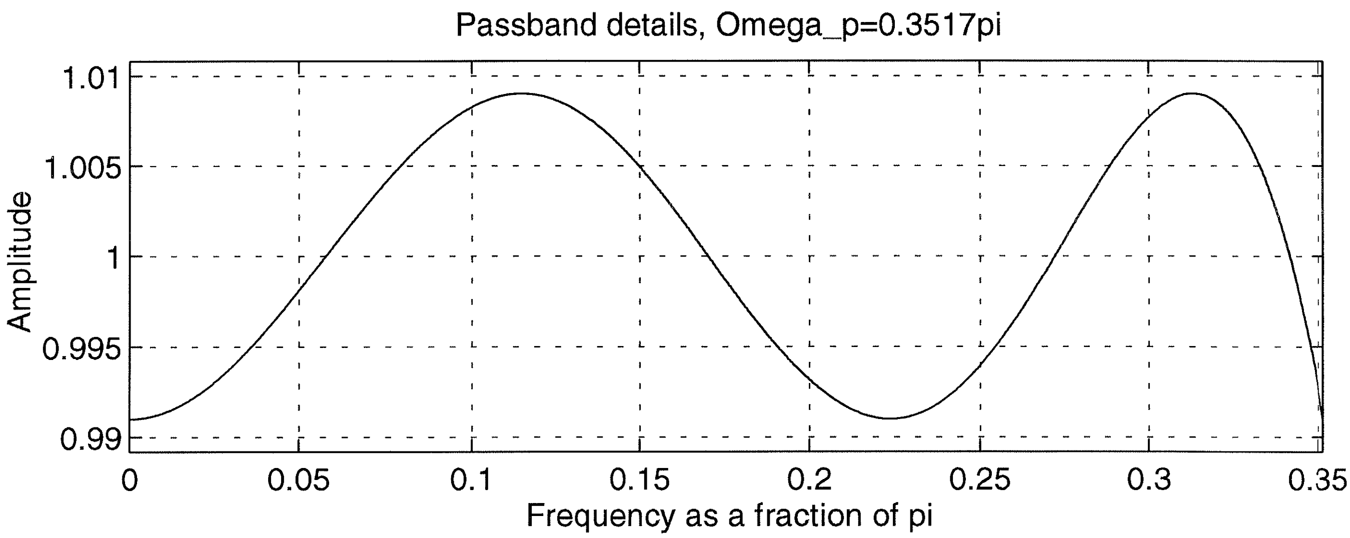

Case B: xs1=-0.2146, xs2=0.2146, xp1=0.9151, xp2=1.085
abs(P(x)) in dB
x
20
0
-20
-40
-60
-80
-100
-120
-140
-160
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
1.2



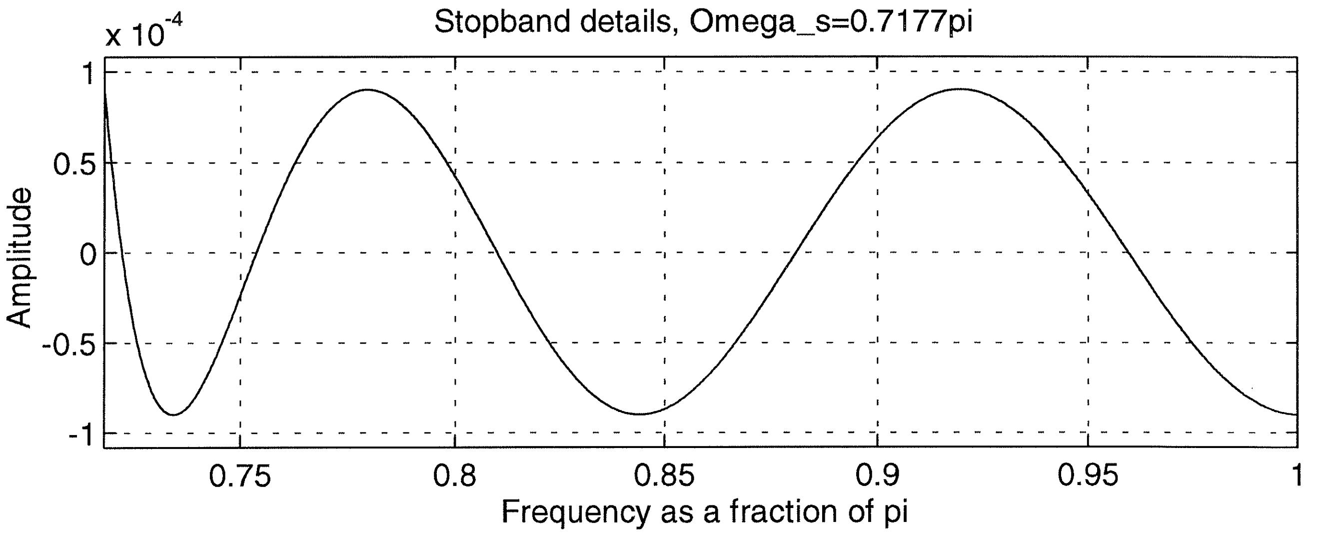
Case B: Passband details
P(x)
1.01
1.005
1
0.995
0.99
0.92
0.94
0.96
0.98
1
1.02
1.04
1.06
1.08
x
Case B: Stopband details
x 10^-4
P(x)
1
0.5
0
-0.5
-1
-0.2
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
0.2
x



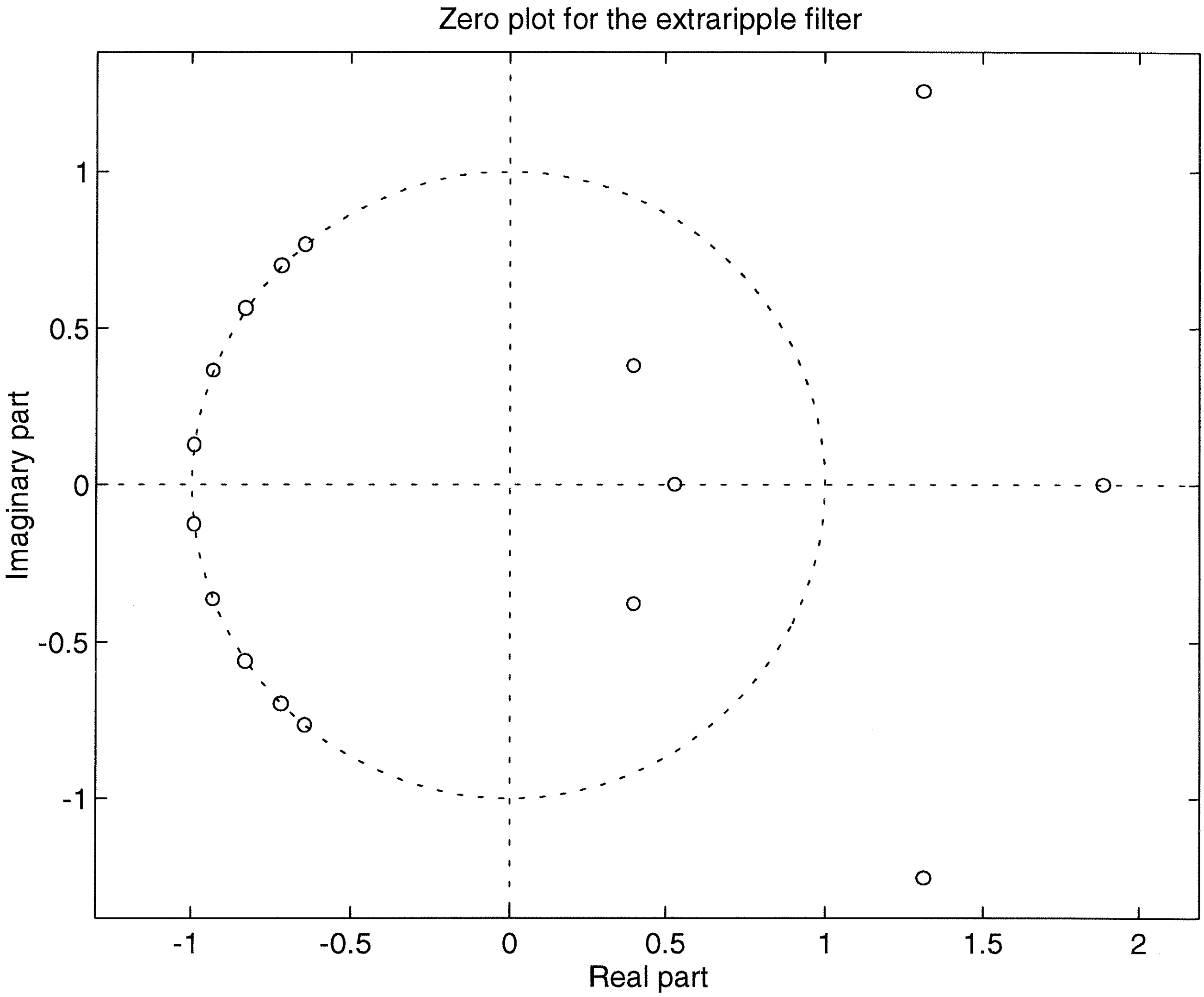
Extraripple filter: order=16 number of passband ripples=4
20
0
-20
-40
-60
-80
-100
-120
-140
-160
-180
Amplitude in dB
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Frequency as a fraction of pi

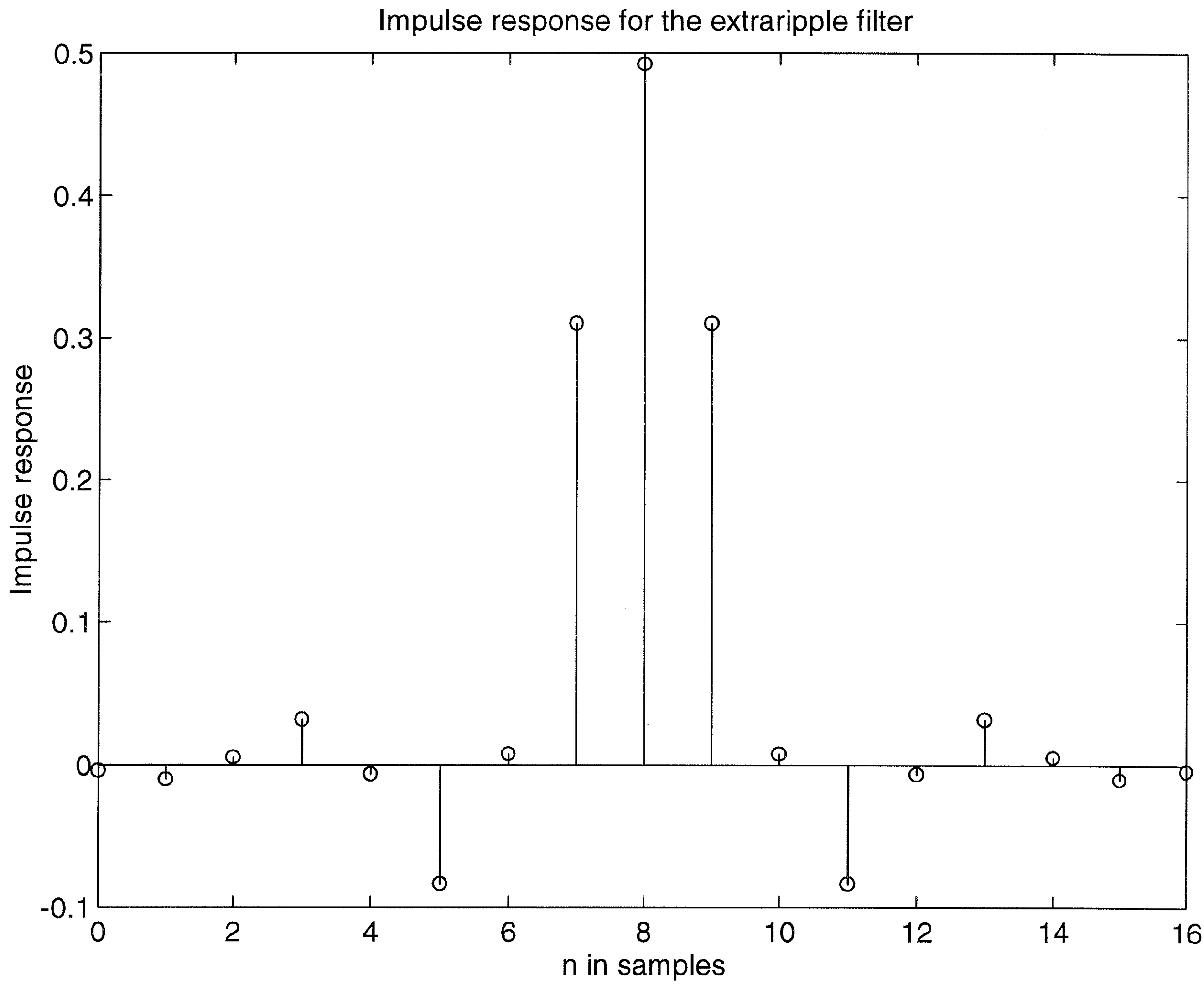


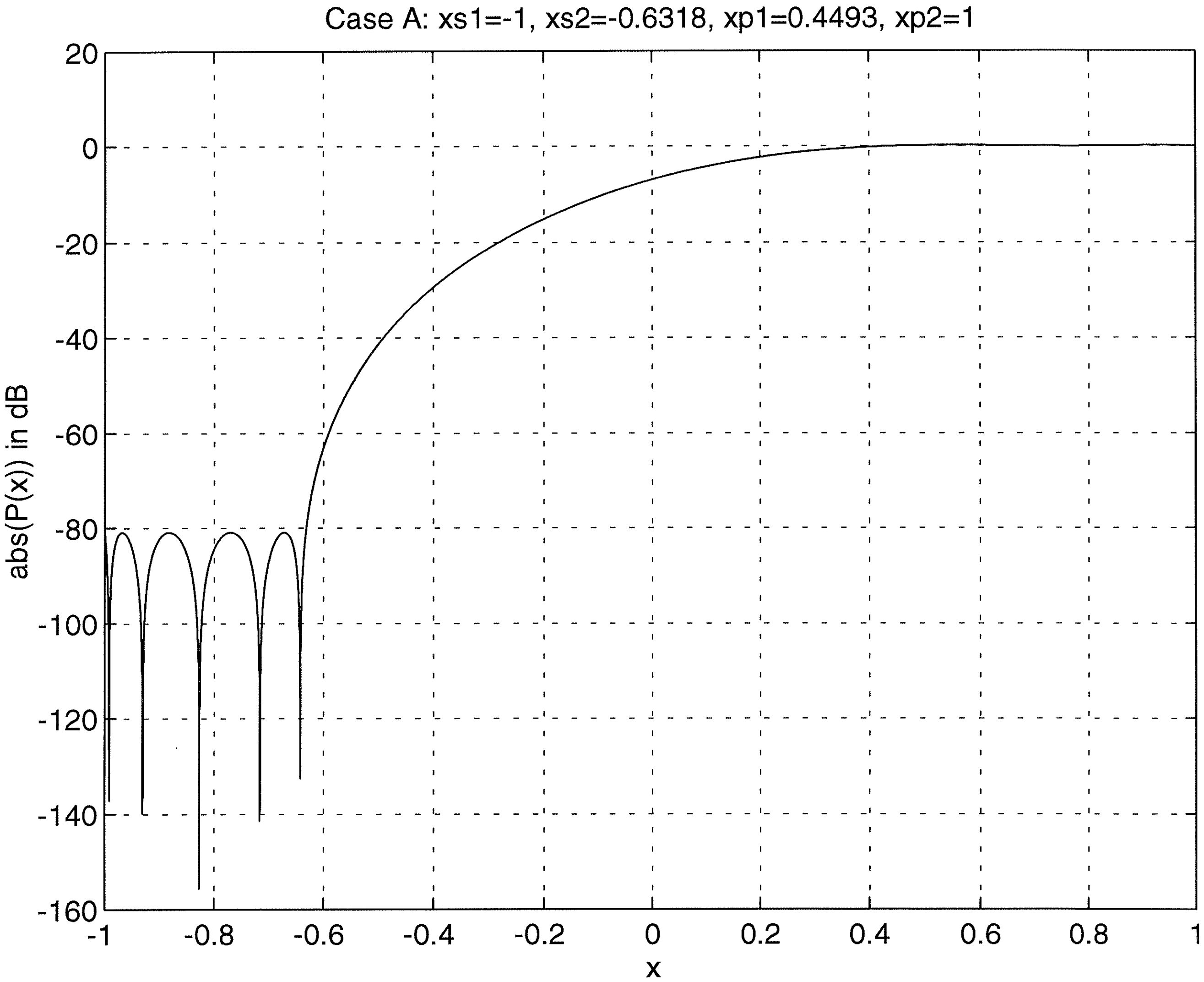
Passband details, Omega_p=0.3517pi
Amplitude
1.01
1.005
1
0.995
0.99
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
Frequency as a fraction of pi



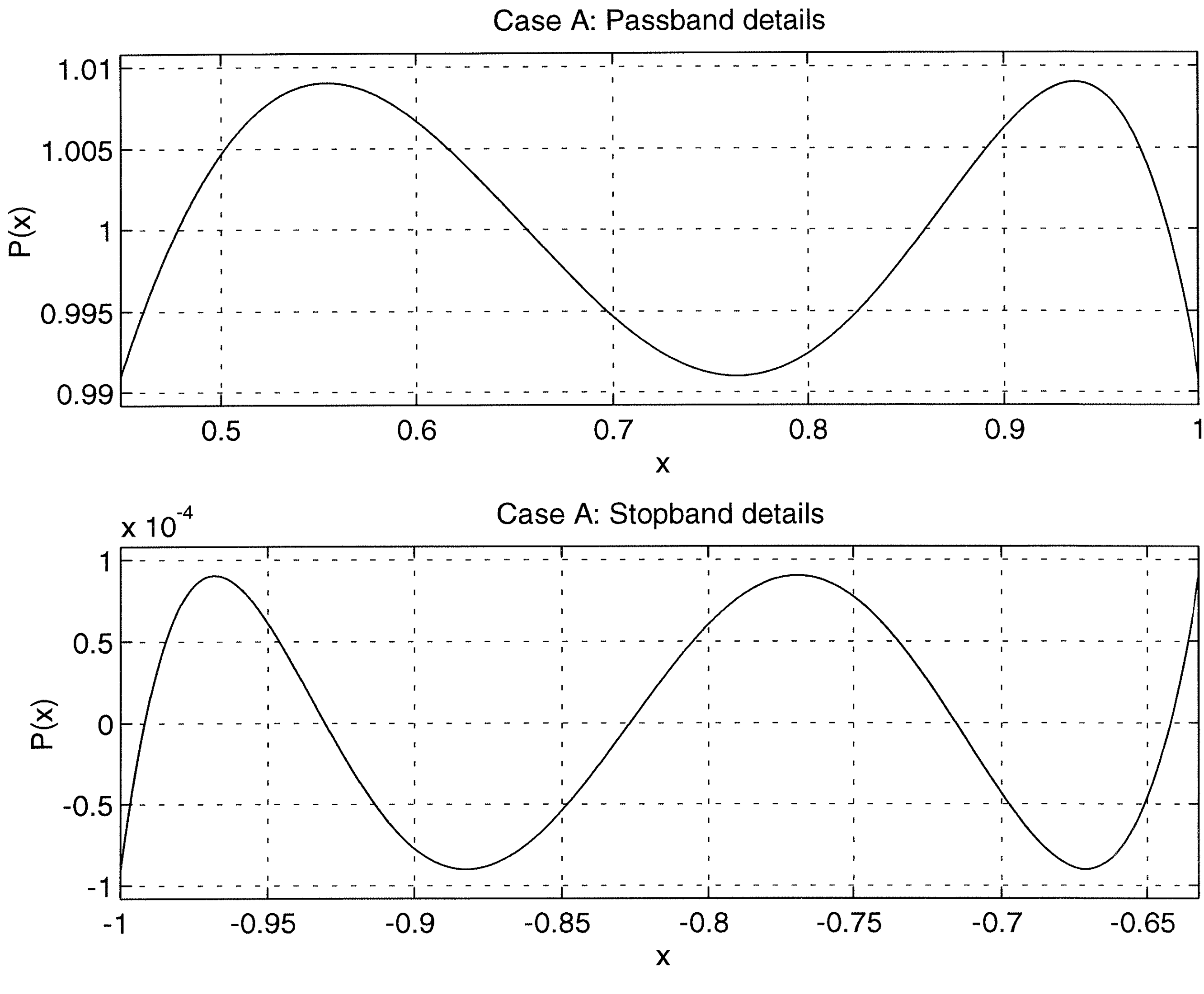
Stopband details, Omega_s=0.7177pi
x 10^-4
Amplitude
1
0.5
0
-0.5
-1
0.75
0.8
0.85
0.9
0.95
1
Frequency as a fraction of pi


Zero plot for the extraripple filter
Imaginary part
Real part
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
1.5
2

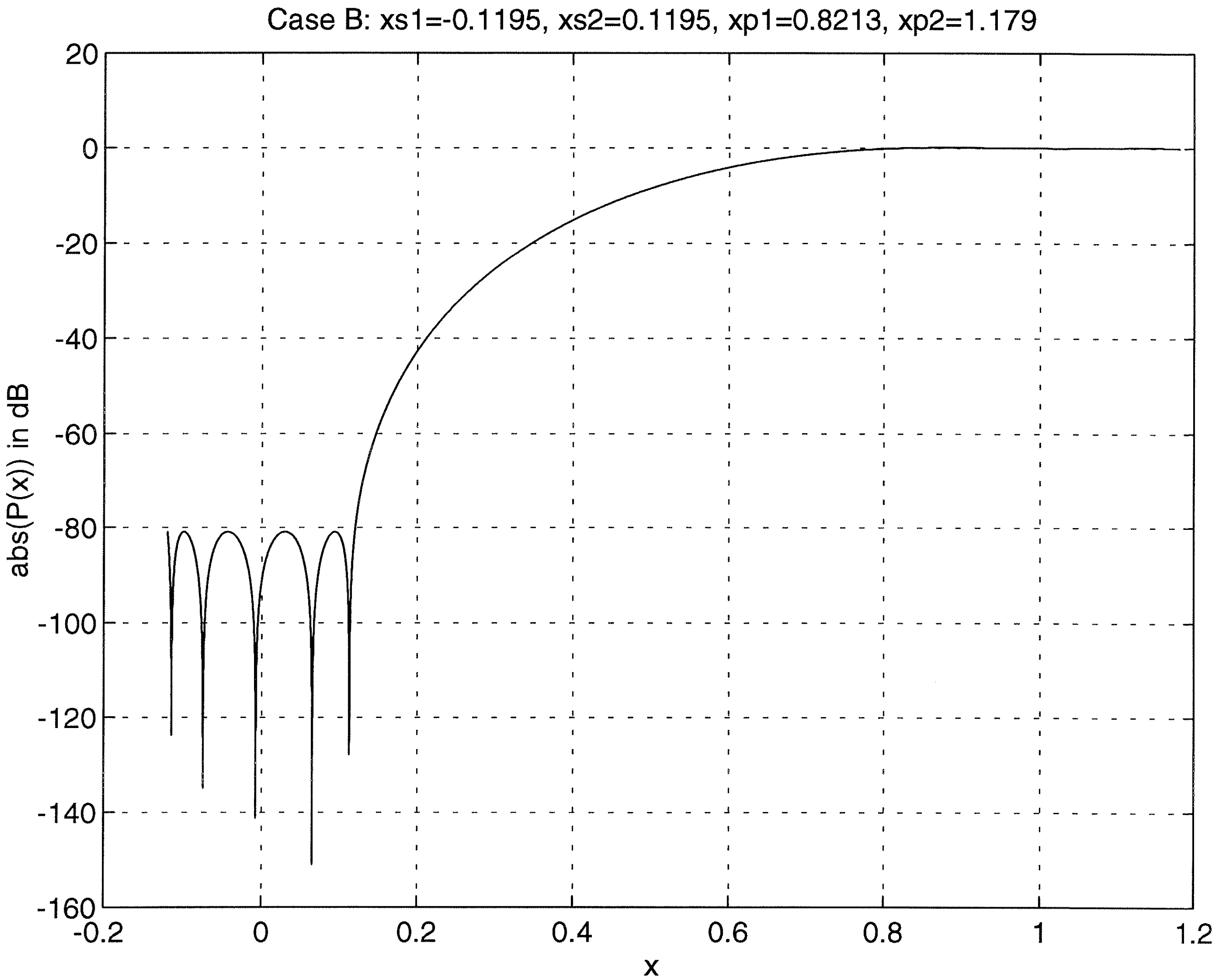

Impulse response for the extraripple filter
Impulse response
0.5
0.4
0.3
0.2
0.1
0
-0.1
0
2
4
6
8
10
12
14
16
n in samples



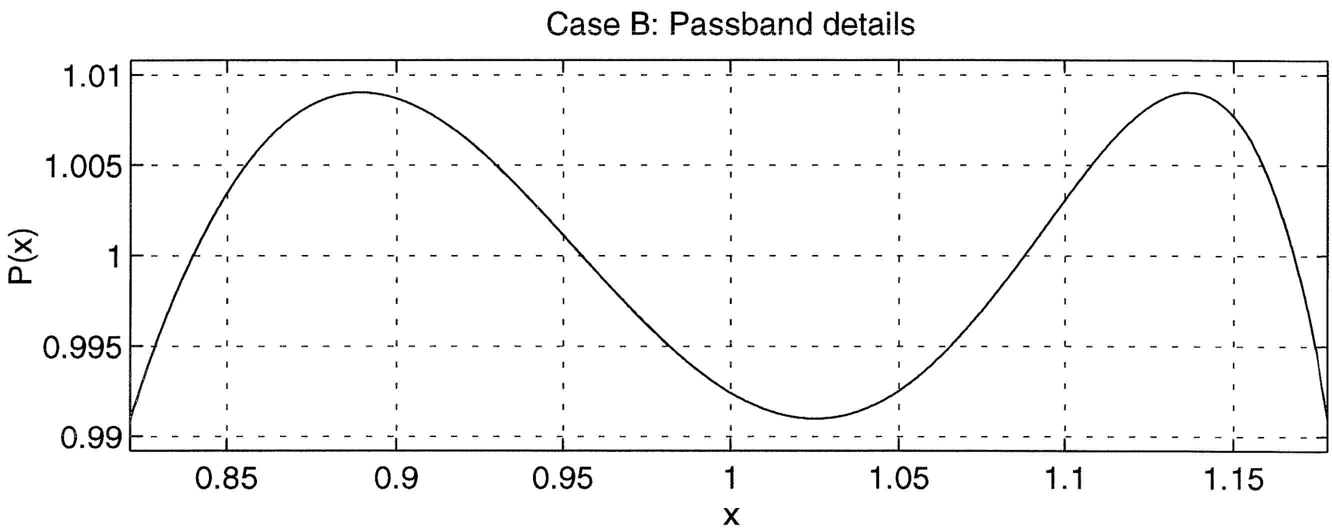
Case A: xs1=-1, xs2=-0.6318, xp1=0.4493, xp2=1
abs(P(x)) in dB
20
0
-20
-40
-60
-80
-100
-120
-140
-160
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
x

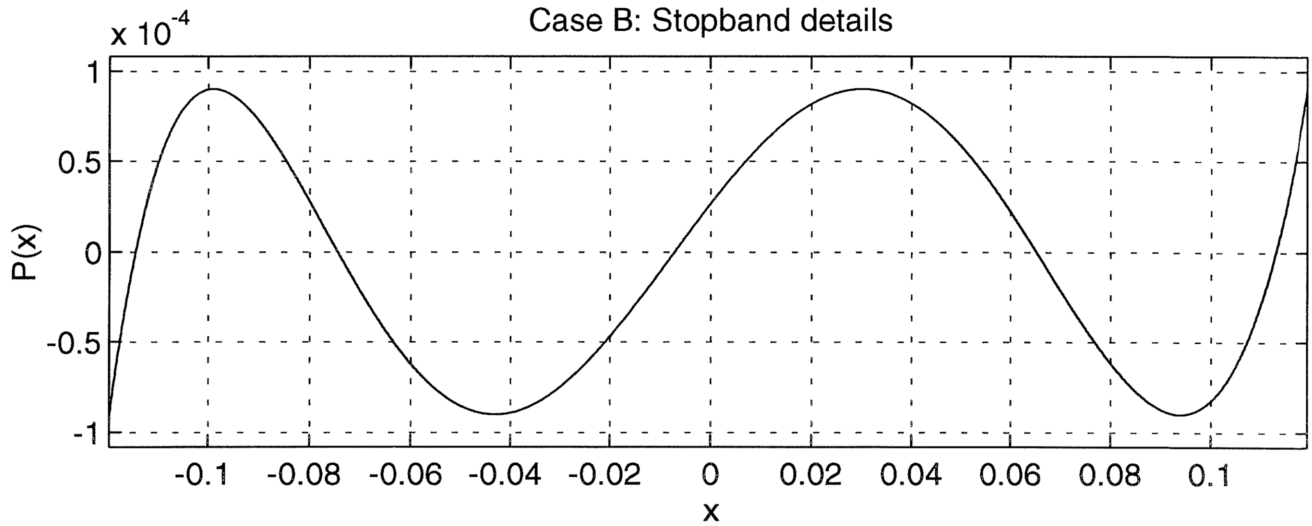


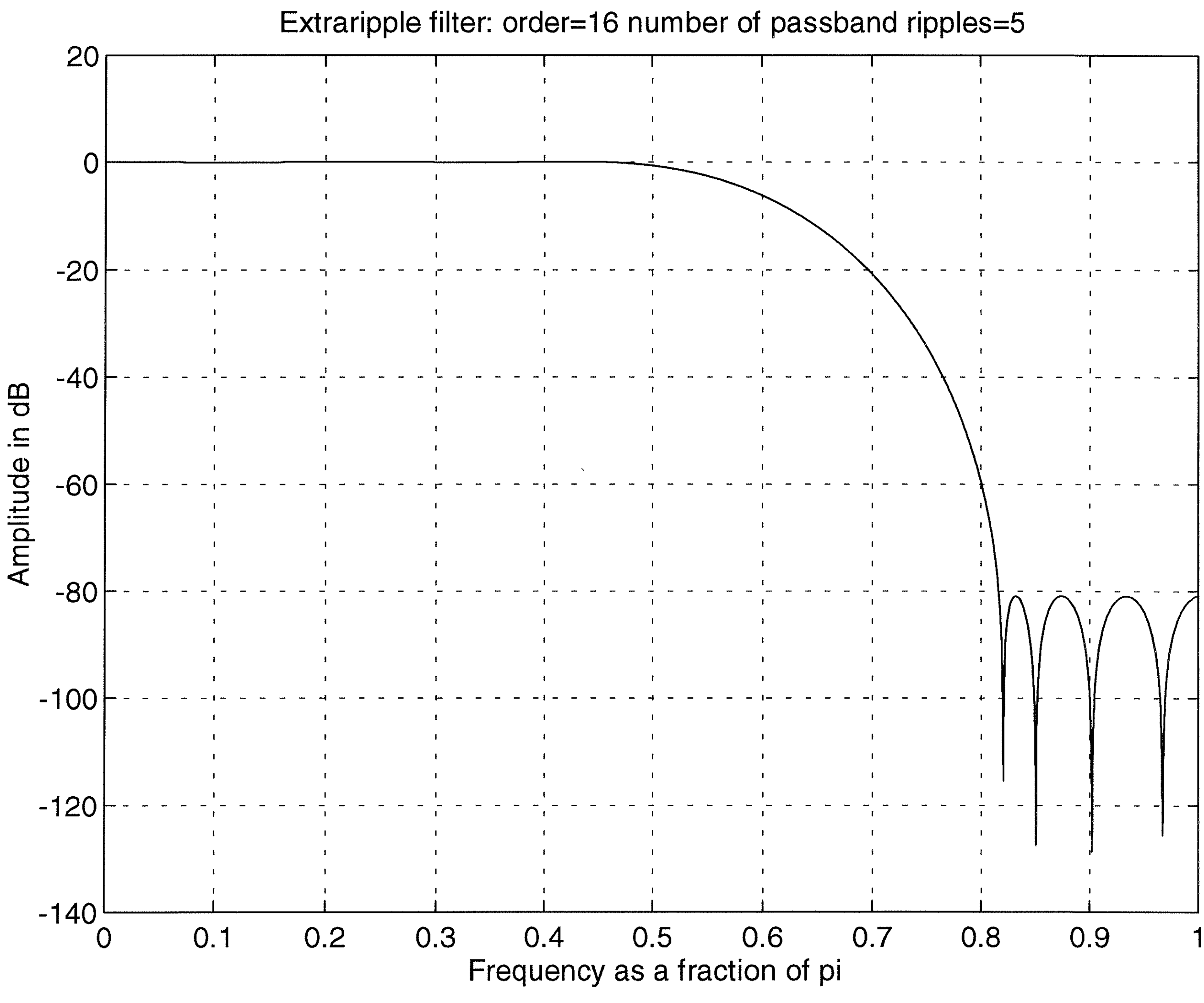
Case A: Passband details
1.01
1.005
1
0.995
0.99
P(x)
0.5
0.6
0.7
0.8
0.9
1
x
Case A: Stopband details
x 10^-4
1
0.5
0
-0.5
-1
P(x)
-1
-0.95
-0.9
-0.85
-0.8
-0.75
-0.7
-0.65
x

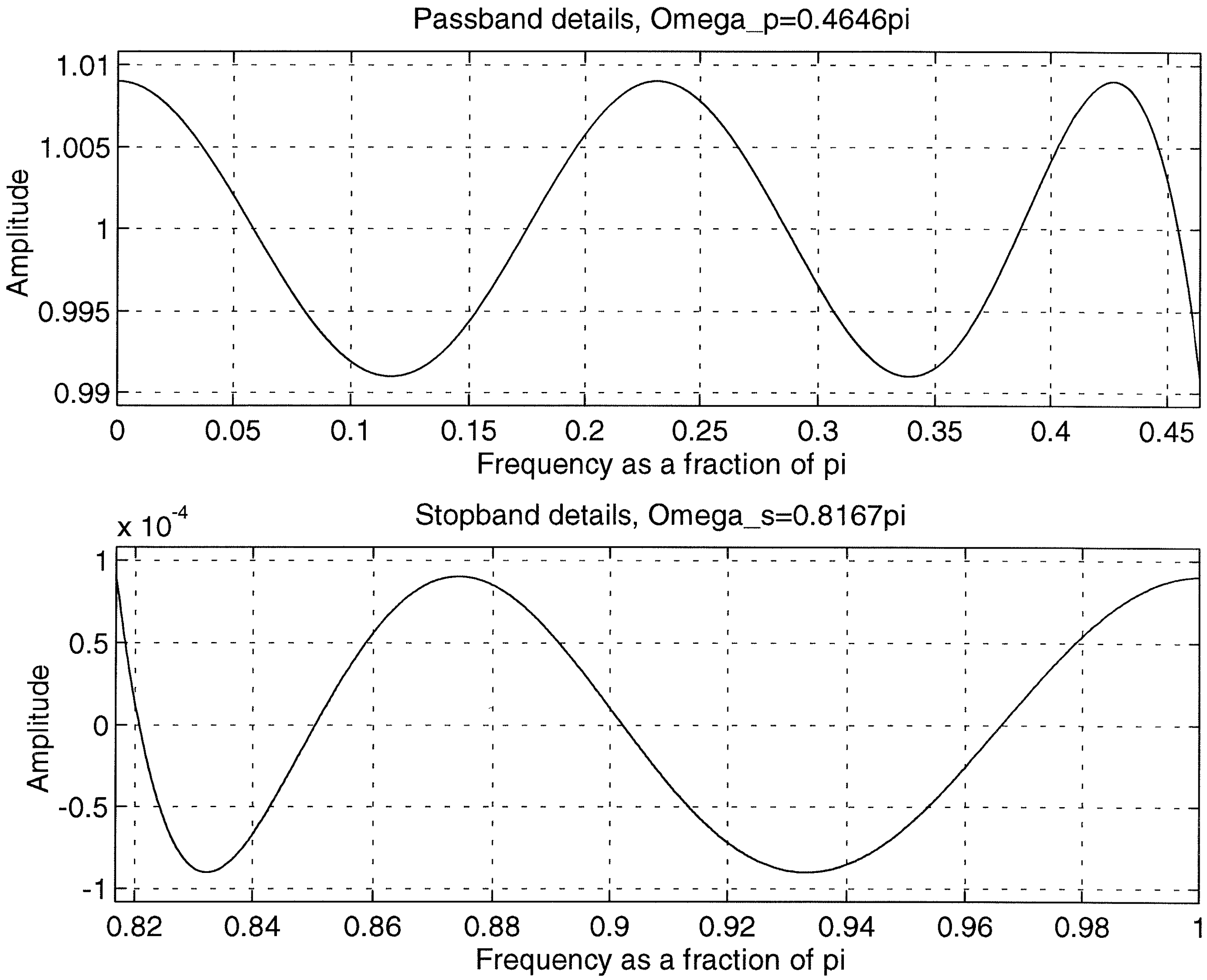


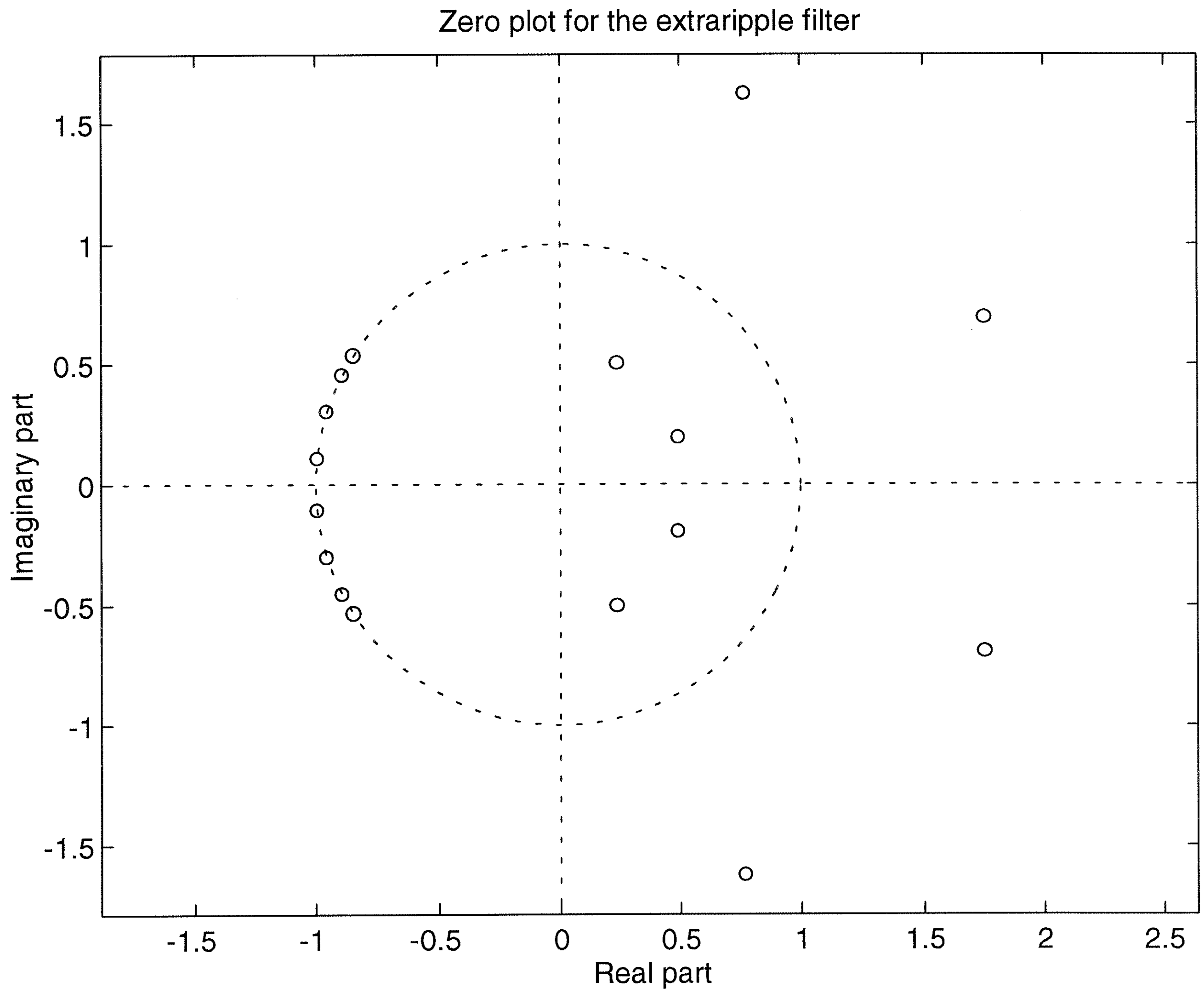
Case B: xs1=-0.1195, xs2=0.1195, xp1=0.8213, xp2=1.179
abs(P(x)) in dB
x
20
0
-20
-40
-60
-80
-100
-120
-140
-160
-0.2
0
0.2
0.4
0.6
0.8
1
1.2


Case B: Passband details
P(x)
1.01
1.005
1
0.995
0.99
0.85
0.9
0.95
1
1.05
1.1
1.15
x

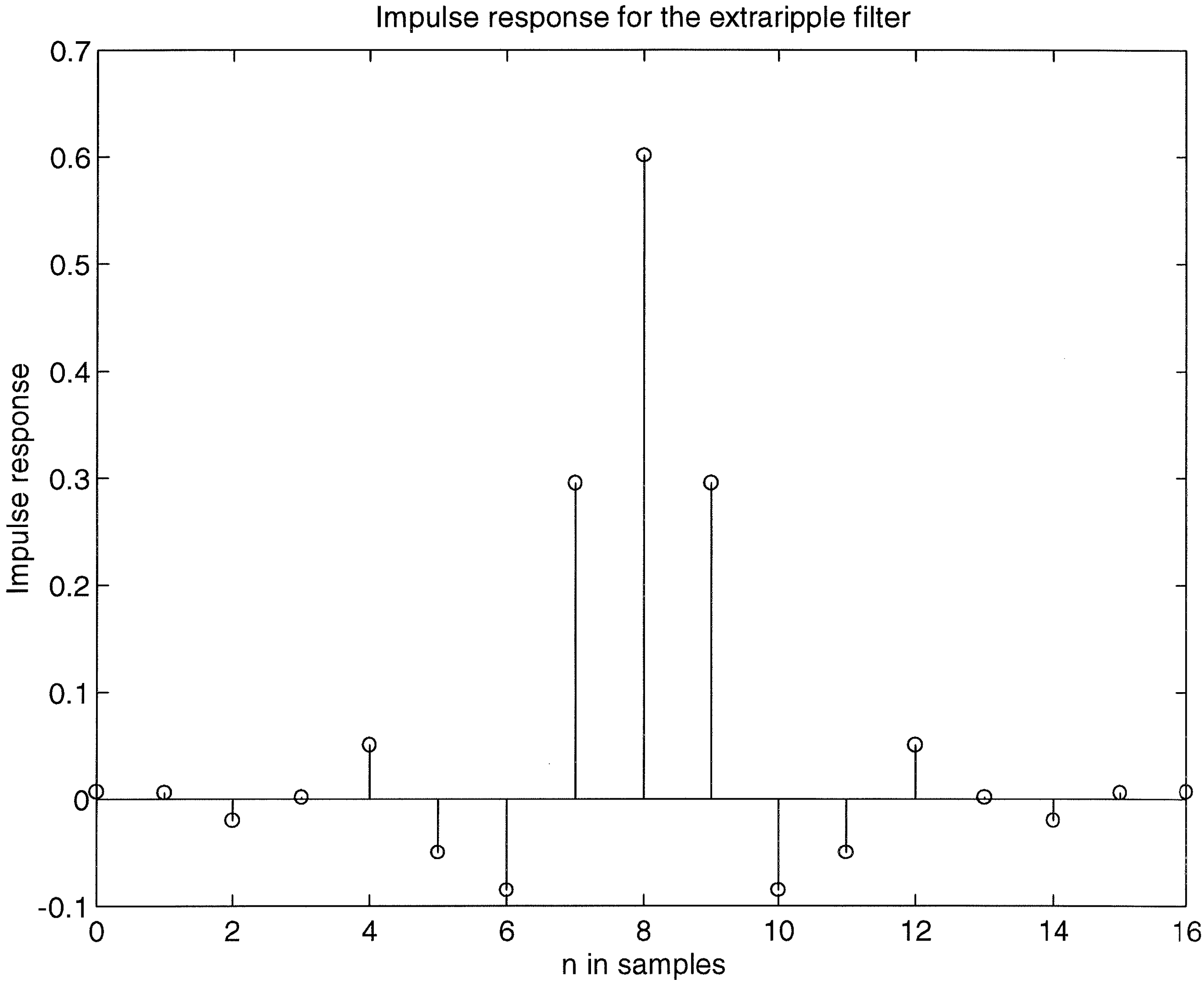

Case B: Stopband details
x 10^-4
P(x)
1
0.5
0
-0.5
-1
-0.1
-0.08
-0.06
-0.04
-0.02
0
0.02
0.04
0.06
0.08
0.1
x


Extraripple filter: order=16 number of passband ripples=5
20
0
-20
-40
-60
-80
-100
-120
-140
Amplitude in dB
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Frequency as a fraction of pi

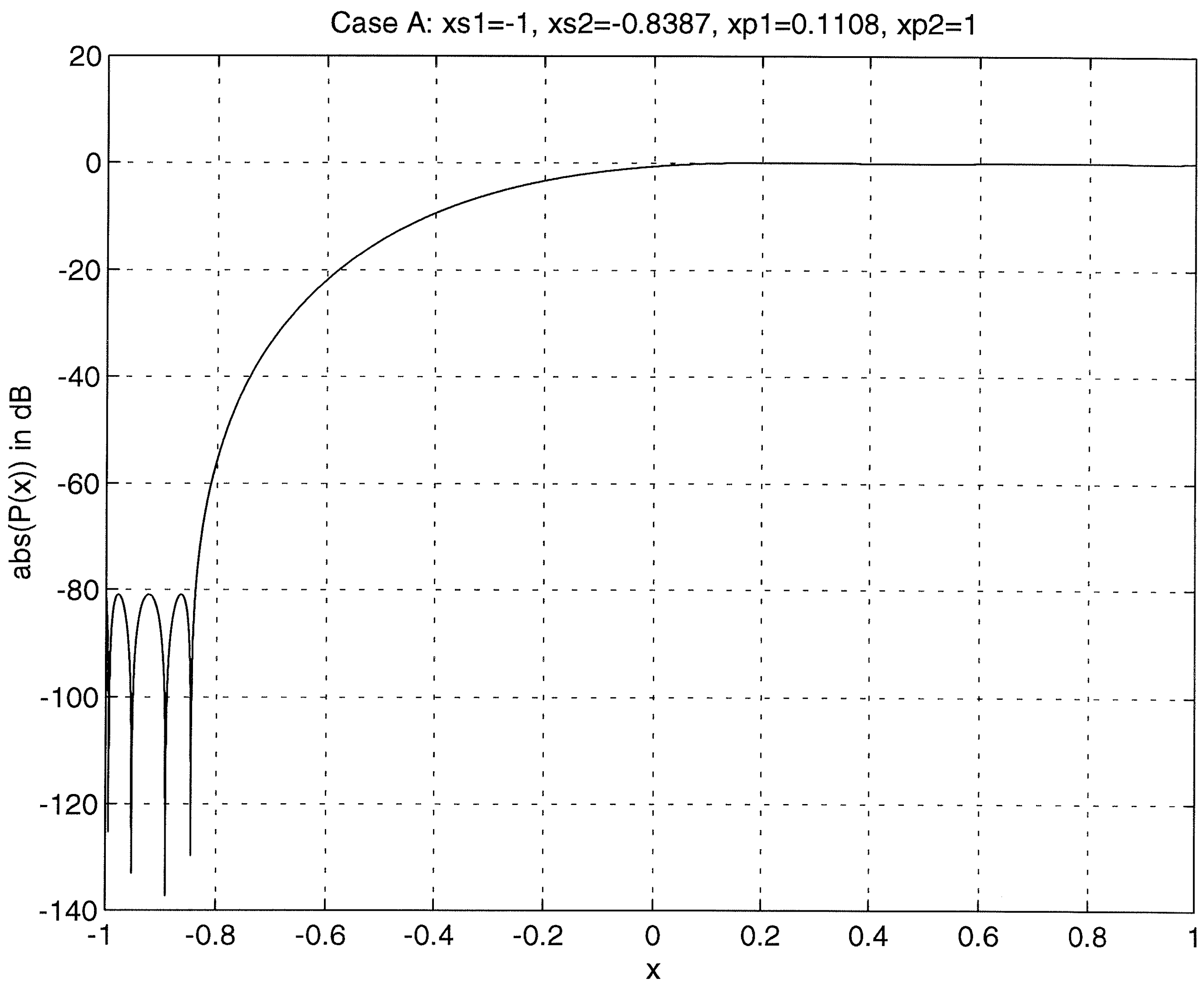

Passband details, Omega_p=0.4646pi
1.01
1.005
1
0.995
0.99
Amplitude
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
Frequency as a fraction of pi
Stopband details, Omega_s=0.8167pi
x 10^-4
1
0.5
0
-0.5
-1
Amplitude
0.82
0.84
0.86
0.88
0.9
0.92
0.94
0.96
0.98
1
Frequency as a fraction of pi

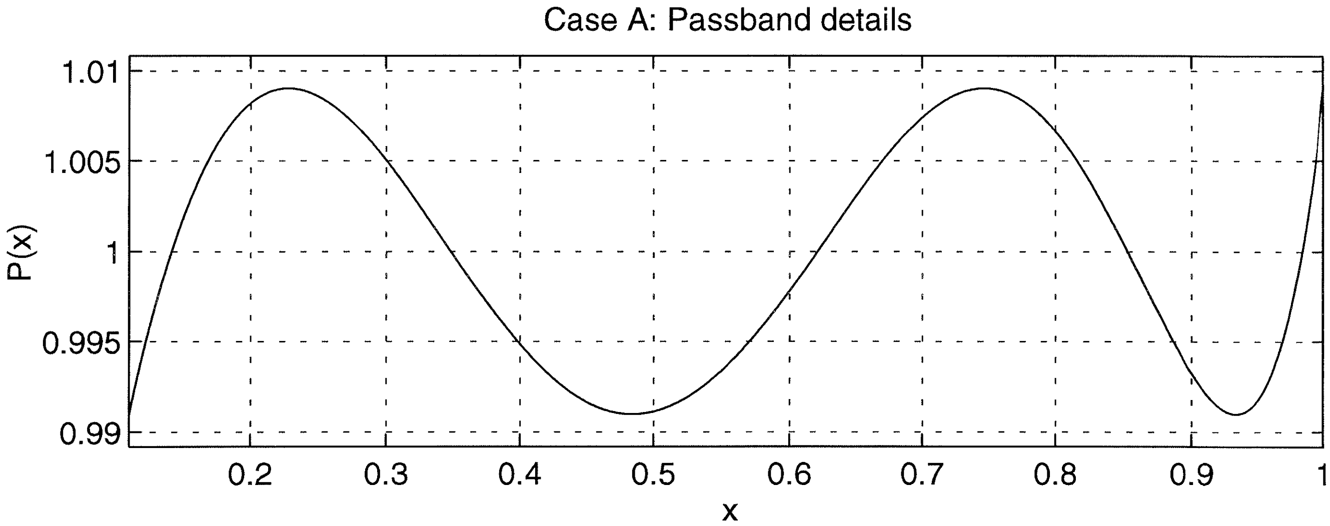


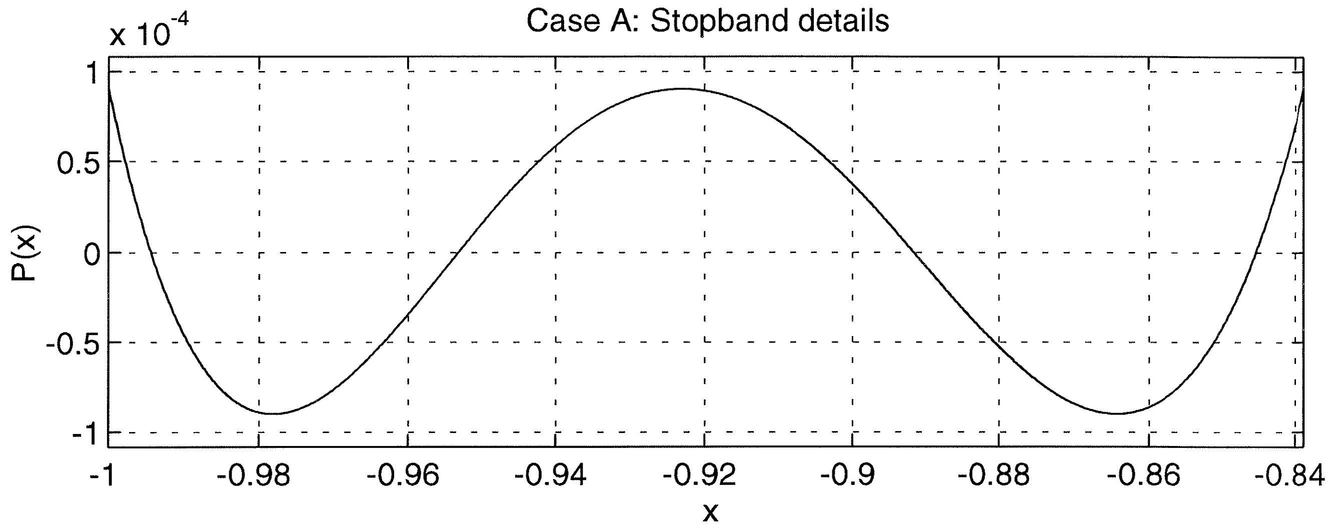
Zero plot for the extraripple filter
Imaginary part
Real part
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5

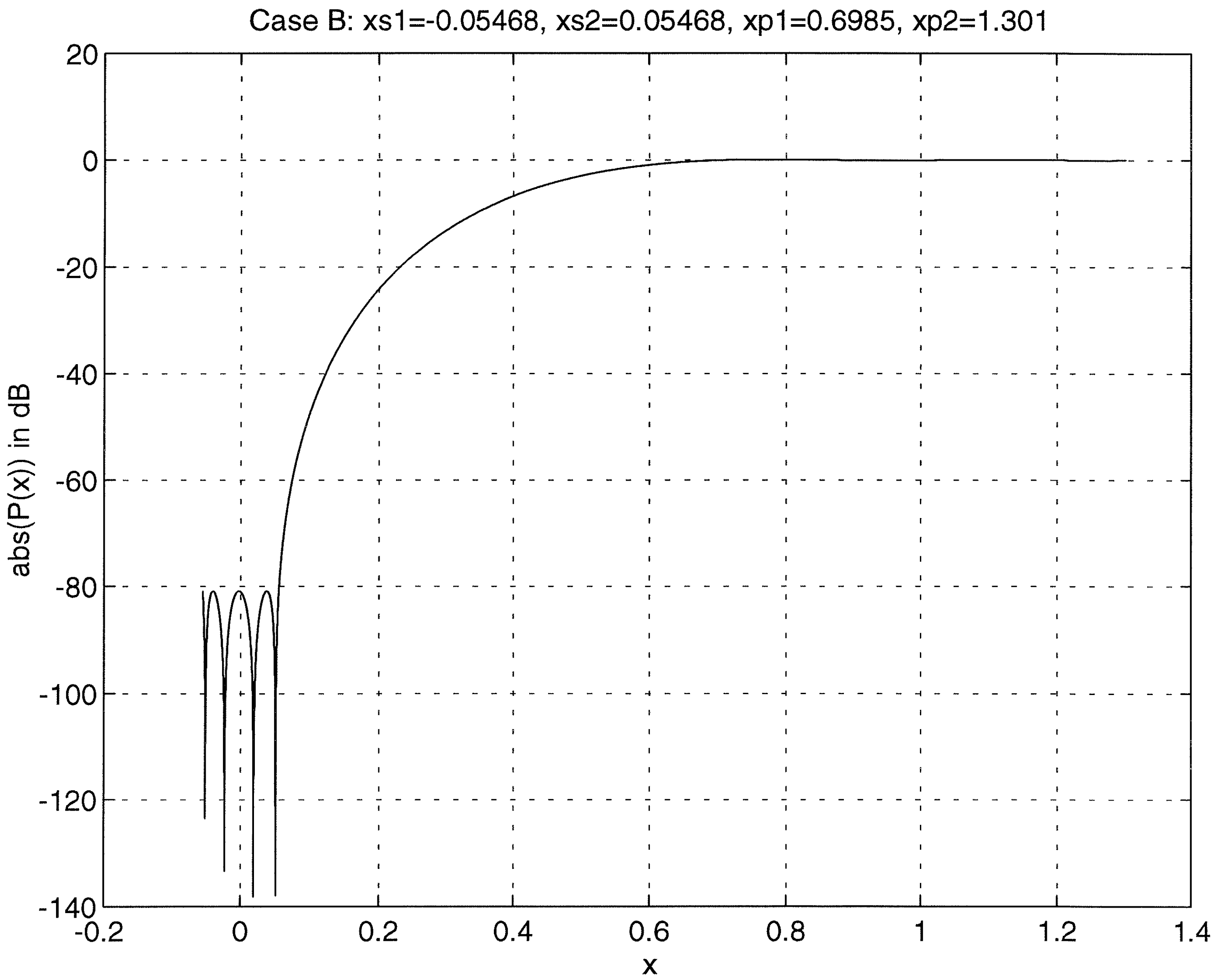


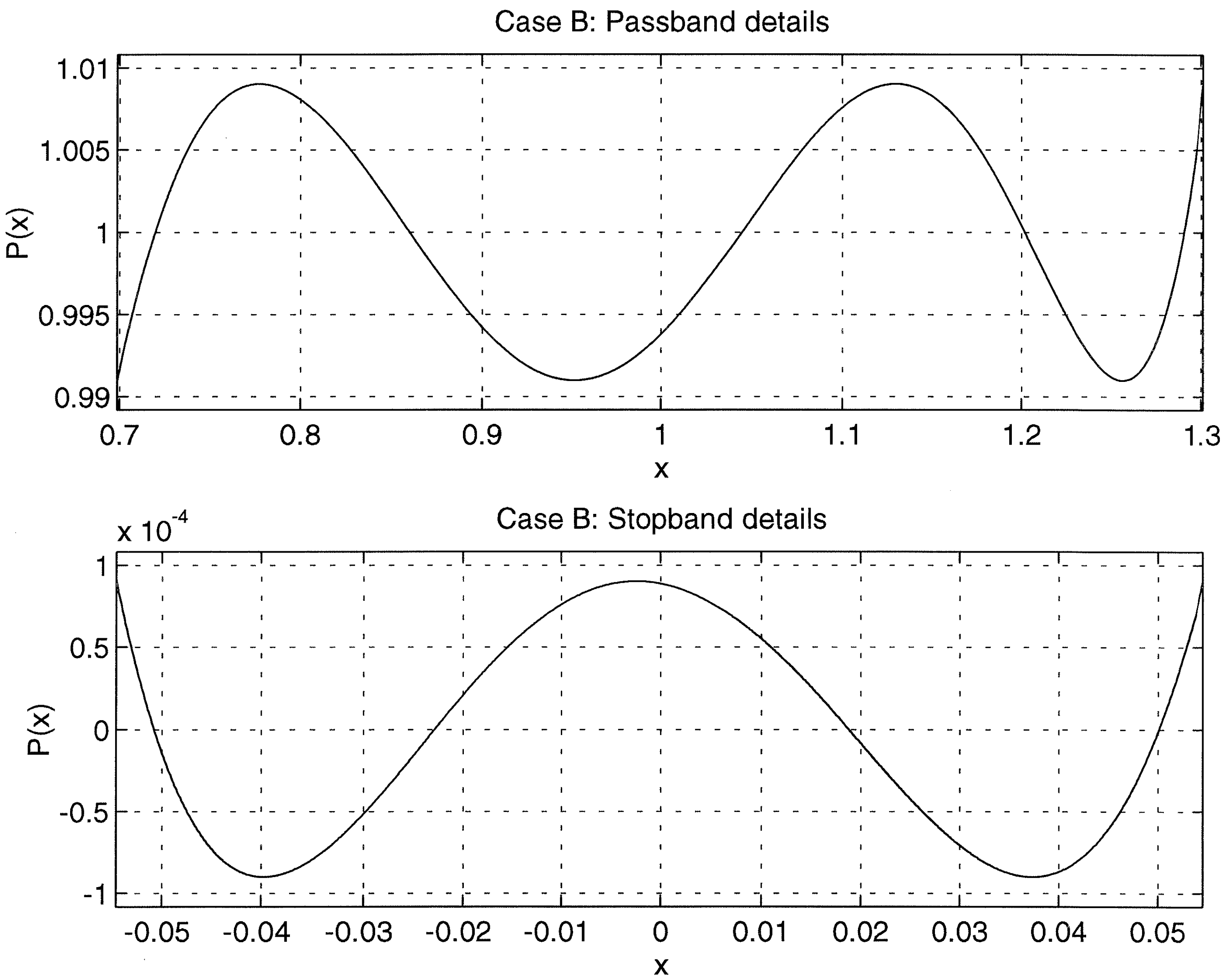
Impulse response for the extraripple filter
Impulse response
n in samples
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
0
2
4
6
8
10
12
14
16


Case A: xs1=-1, xs2=-0.8387, xp1=0.1108, xp2=1
abs(P(x)) in dB
x
20
0
-20
-40
-60
-80
-100
-120
-140
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


Case A: Passband details
1.01
1.005
1
0.995
0.99
P(x)
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x


Case A: Stopband details
x 10^-4
1
0.5
0
-0.5
-1
P(x)
-1
-0.98
-0.96
-0.94
-0.92
-0.9
-0.88
-0.86
-0.84
x


Case B: xs1=-0.05468, xs2=0.05468, xp1=0.6985, xp2=1.301
abs(P(x)) in dB
x
20
0
-20
-40
-60
-80
-100
-120
-140
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4


Case B: Passband details
P(x)
1.01
1.005
1
0.995
0.99
0.7
0.8
0.9
1
1.1
1.2
1.3
x
Case B: Stopband details
x 10^-4
P(x)
1
0.5
0
-0.5
-1
-0.05
-0.04
-0.03
-0.02
-0.01
0
0.01
0.02
0.03
0.04
0.05
x

Matlab-file extralin.m

- Given δ_p and δ_s, this routine automatically finds out the best extraripple solution.
- It gives for Cases A and B $P(x)$ in the form
$$P(x) = C\prod_{k=1}^{N_1}[b_k[2]x^2+b_k[1]x+b_k[0]]\prod_{k=1}^{N_2}[c_k[1]x+c_k[0]],$$
where the $b_k[2]$'s and $c_k[1]$'s are equal to unity.
- It gives also x_{s1}, x_{s2}, x_{p1}, and x_{p2}.
- The desired $P(x)$ is formed according to the discussion of pages 17–21 by using the values of α and β given on page 34.
- The above data is saved for further use in file linfir.
- There is also a file, called subfir.m, which reads linfir as well as the impulse response coefficients of the subfilter (hsub) and plots variuos responses.
- On page 56 there is an example on how to use subfir.m.
- You can find subfir.m and extralin.m in the following transparencies.

```
% Matlab m-file (extralin.m) for determining the best
% extraripple solution for minimizing the even sub-
% filter order for the given number of subfilters in
% the case where identical FIR subfilters are used
% for building the overall filter.
%
% See T. Saram"aki "Finite impulse response filter
% design" in Handbook for Digital Signal Processing,
% edited by S. K. Mitra and J. F. Kaiser, John Wiley
% & Sons, 1993, pages 256 - 271.
%
% This program determines the unquantized additional
% tap coefficients in Figure 4-58(b).
%
% The input data for this program are the passband
% and stopband ripples of the overall filter as well
% as the number of subfilters.
%
% This program finds first the best extraripple
% solution for these criteria. See the above-
% metioned article. Some additional information is
% included in the program
%
% Tapio Saram"aki 5.3.1997, e-mail: ts@cs.tut.fi
%
% This program uses a modified Remez-routine, called
% extrar.m, for determing the extraripple solutions.
% This program has been modified by Tapio Saram"aki.
%
% The routines be found in Sun's: ~ts/matlab/sldsp
%
% Quntization of the additional tap coefficients to
% two or three powers-of-two is not yet included in
% this routine.
%
% This routine explains how to get P(x) in the form
% of equation (4.205) in a simple manner than in the
% above-mentioned article.
%
% Initial values for the case considered in the
% above-mentioned article can be found by using
% dp=0.009,ds=0.00009,nsub=8. Please try!
```

```
%
% can be found in SUN's: ~ts/matlab/sldsp
%
 disp('Hi there')
 disp('I am an program for designing extraripple')
 disp('lowpass FIR filters')
 dp=input('Passband ripple= ');
 ds=input('Stopband ripple= ');
 disp('1 for fixing the nuber of passband ripples')
 itype=input('0 for automatic optimization: ')
 if itype==1
   ipass=input('Number of passband ripples= ');
 end
 nsub=input('Number of subfilters = ');
 nfilt=2*nsub;
 ipas=0;
 if itype==1 ipas=ipass-1;end
 diffe=1000;
 ll=0;
 while ll < 1
   ipas=ipas+1
   [hhh,cut]=...
   extrar(nfilt,[dp ds],ipas,[0 1],[1 1],1);
   cut=2*cut;
%
% Find the passband edge, that is, the frequency point
% where the value 1-dp is achieved
%
   xc=(cut(2)+cut(1))/2;
   xd=cut(2)-xc;
   amdes=1-dp;
%
% the edge is dsired to be determined with accuracy
% less than or equal to 10^(-12)*pi
%
    kk=ceil(log10((10^12/(cut(2)-cut(1))/1000))/2);
    for k=1:kk
      [AA,ww]=zeroam(hhh,xc-xd,xc+xd,1000);
      [Y,I]=sort(abs(AA-amdes));
      ww=ww(I);xc=ww(1)/pi;xd=xd/1000;
    end
    cc1=xc;
```

```
%
% Find the passband edge, that is, the frequency point
% where the value 1-dp is achieved
%
    xc=(cut(2)+cut(1))/2;
    xd=cut(2)-xc;
    amdes=ds;
    for k=1:kk
      [AA,ww]=zeroam(hhh,xc-xd,xc+xd,1000);
      [Y,I]=sort(abs(AA-amdes));
      ww=ww(I);xc=ww(1)/pi;xd=xd/1000;
    end
    cc2=xc;
    difc=1-cc2;diffe1=abs(difc-cc1);
    ll=1
    if diffe1 < diffe
      h=hhh;diffe=diffe1;ipass=ipas;ll=0;c1=cc1;
      c2=cc2;
    end
    if itype==1 ll=1; end
  end
 [H,f]=zeroam(h,.0,1.,4000);
 figure(1)
 plot(f/pi,20*log10(abs(H)));grid;
 title(['Extraripple filter: order=',...
 num2str(nfilt),' number of passband ripples=',...
 num2str(ipass)]);
 xlabel('Frequency as a fraction of pi');
 ylabel('Amplitude in dB')
 figure(2)
 subplot(211)
 plot(f/pi,H);grid;axis([0 c1 1-1.2*dp 1+1.2*dp]);
 title(['Passband details, Omega_p=', num2str(c1),'pi']);
 xlabel('Frequency as a fraction of pi');
 ylabel('Amplitude ');
 subplot(212)
 plot(f/pi,H);grid;axis([c2 1 -1.2*ds 1.2*ds]);
 title(['Stopband details, Omega_s=', num2str(c2),'pi']);
 xlabel('Frequency as a fraction of pi');
 ylabel('Amplitude ');
 figure(3)
 zplane(h);title('Zero plot for the extraripple filter')
```

```
  figure(4)
  impz(h)
  title('Impulse response for the extraripple filter');
  xlabel('n in samples');ylabel('Impulse response')
  g=h;
%
% Form
G(Omega)=gconst*G_1(Omega)*G_2(Omega)*G_3(Omega),
% Here, gconst=2^(nsub)*h(1). G_1(Omega)=cos(Omega)+gnr(1)
% with gnr(1)=(-r-1/r)/2 if g contains a reciprocal zero
% pair at z=r,1/r.
% G_2(Omega)=[cos^2(Omega)+gnq1(1)*cos(Omega)+qng2(1)]...
% [cos^2(Omega)+gnq1(nq)*cos(Omega)+qng2(nq)] if g contains
% nq zero quadruplets at z=r(k)*exp(+-jphi(k),
% (1/r(k))*exp(+-jphi(k) for k=1,2,...,nq.
% Knowing the zero quadruplet z1,z2,z3,z4 for the kth term,
% we first form gk(z)=poly(z1,z2,z3,z4)=
% [1+z^(-4)]+a[z^(-1)+z^(-3)]+bz^(-3).
% Then, the corresponding term in G_2(Omega) is
% [cos^2(Omega)+(a/2)*cos(Omega)+(b-2)/4].
% G_3(Omega)=[cos(Omega)+gnc(1)]...[cos(Omega)+gnc(nc)]
% if g contains nc zero pairs on the unit circle at the
% angular frequencies Omega=Omega_k for k=1,2,...,nc.
% Here, gnc(k)=-cos(Omega_k)
%
  q=roots(h);q=sort(q);
  gconst=g(1)*2^(nsub);
  nq=floor((ipass-1)/2); %number of zero quadruplets
  nr=ipass-1-2*nq; %number of reciprocal zero pairs (1 or 0)
  nc=nsub-ipass+1; %number of unit-circle zero pairs
%
% For possible reciprocal zero pair on the real axis
%
  if nr==1 gnr(1)=(-q(1)-1/q(1))/2; end
%
% For zero quadruplets
%
  if nq > 0
    for k=1:nq
      clear d;
      d(1)=q(nr+2*k-1);d(2)=q(nr+2*k);d(3)=1/d(1);
      d(4)=1/d(2);
```

```
    f=poly(d);
    gnq1(k)=real(f(2)/2);gnq2(k)=real((f(3)-2)/4);
   end
 end
%
% For zero pairs on the unit circle
%
 mm=nr+2*nq
 for k=1:nc
   clear d;
   d(1)=q(mm+2*k-1);d(2)=q(mm+2*k);
   f=poly(d);
   gnc(k)=f(2)/2;
 end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Case A: x=cos(Omega), P(x) is formed like in equation
% (4.205); D=gconst, N_2=nr+nc, gamma_k's are gnr(1)
% (nr=1) and gnc(k) for k=1,2,...,nc. alpha(k)=gnq1(k)
% and beta(k)= gnq2(k) for k=1,2,.., nq; N1=nq
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Edges for P(x)
%
 xp2=1;xp1=cos(pi*c1);xs2=cos(pi*c2);xs1=-1;
%
% Evaluate P(x) in the range [-1, 1)
 x=(-1:.0001:1);
 PA=gconst*ones(size(x));
 if nr==1 p=x+gnr(1); PA=p.*PA;end
 if nq> 0
   for k=1:nq
     p=x.*x+gnq1(k)*x+gnq2(k);
     PA=p.*PA;
   end
 end
 for k=1:nc
   p=x+gnc(k);
   PA=p.*PA;
 end
 figure(5)
```

```
  plot(x,20*log10(abs(PA)));grid;
  title(['Case A: xs1=',num2str(xs1),', xs2='...
  ,num2str(xs2),', xp1=',num2str(xp1),', xp2='...
  ,num2str(xp2)])
  ylabel('abs(P(x)) in dB'); xlabel('x');
  figure(6)
  subplot(211)
  plot(x,PA);axis([xp1 xp2  1-1.2*dp 1+1.2*dp]);grid;
  title(['Case A: Passband details'])
  ylabel('P(x)'); xlabel('x');
  subplot(212)
  plot(x,PA);axis([xs1 xs2  -1.2*ds 1.2*ds]);grid;
  title(['Case A: Stopband details'])
  ylabel('P(x)'); xlabel('x');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%
% Case B: P(x)=P'((x-beta)/alpha)), where P'(x) is the
% polynomial in Case A. P(x) is generated in the form of
% equation (4.205) by applying the transformation to the
% first-order and second-order sections of P'(x)
% separately.
% [x+d] is mapped to (1/alpha)[x+(alpha*d-beta)].
% [x^2+d1*x+d2] is mapped to (1/alpha^2)[x^2+dd1*x+dd2],
% where dd1=-2*beta+alpha*d1 and dd2=beta^2-
% beta*alpha*d1+alpha^2*d2. The terms before the
% parenthesis are included in D.
% D=gconstb, N_2=nr+nc, gamma_k's are gnrb(1) (nr=1) and
% gncb(k) for k=1,2,...,nc. alpha(k)=gnq1b(k) and beta(k)=
% gnq2b(k) for k=1,2,.., nq; N1=nq
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Determine alpha, beta, bardp, and bards according to
% equations (4.202) and (4.204)
%
 aa=2+cos(c1*pi)-cos(pi*c2);
 alpha=2/aa;beta=(1-cos(pi*c2))/aa;
 bardp=(1-cos(pi*c1))/aa;
 bards=(1+cos(pi*c2))/aa;
%
% Determine D
%
```

```
 gconstb=gconst/(alpha^(nsub));
%
% First-order section corresponding to a possible reciprocal
% zero pair on the real axis
%
 if nr==1 gnrb(1)=alpha*gnr(1)-beta; end
%
% Second-order sections corresponding to the zero
% quadruplets
%
 if nq > 0
   for k=1:nq
     gnq1b(k)=-2*beta+alpha*gnq1(k);
     gnq2b(k)=beta*beta-beta*alpha*gnq1(k)...
             +alpha*alpha*gnq2(k);
   end
 end
%
% First-order sections corresponding to the zero pairs on
% the unit circle
%
 for k=1:nc
   gncb(k)=alpha*gnc(k)-beta;
 end
%
% Edges for P(x)
%
 xp2b=1+bardp;xp1b=1-bardp;xs2b=bards;xs1b=-bards;
%
% Evaluate P(x) in the interval [xs1b, xp2b]
%
 x=(xs1b:.0001:xp2b);
 PA=gconstb*ones(size(x));
 if nr==1 p=x+gnrb(1); PA=p.*PA;end
 if nq> 0
   for k=1:nq
     p=x.*x+gnq1b(k)*x+gnq2b(k);
     PA=p.*PA;
   end
 end
 for k=1:nc
   p=x+gncb(k);
```

```
  PA=p.*PA;
end
figure(7)
plot(x,20*log10(abs(PA)));grid;
title(['Case B: xs1=',num2str(xs1b),...
', xs2=',num2str(xs2b),', xp1=',num2str(xp1b),...
', xp2=',num2str(xp2b)])
ylabel('abs(P(x)) in dB'); xlabel('x');
figure(8)
subplot(211)
plot(x,PA);axis([xp1b xp2b  1-1.2*dp 1+1.2*dp]);grid;
title(['Case B: Passband details'])
ylabel('P(x)'); xlabel('x');
subplot(212)
plot(x,PA);axis([xs1b xs2b  -1.2*ds 1.2*ds]);grid;
title(['Case B: Stopband details'])
ylabel('P(x)'); xlabel('x');
disp('Form P(x)')
disp('unscaled tap coefficients')
itype=input('1 for Case A and 2 for Case B')
if itype==1
  xxs1=xs1;xxs2=xs2;xxp1=xp1;xxp2=xp2;
  D=gconst;
  N2=nc;
  if nr==1 N2=1+nc; gamma(1)=gnr(1);end
  if nc > 0
    for k=1:nc
      gamma(nr+k)=gnc(k);
    end
  end
  if nq > 0
    N1=nq;
    for k=1:nq
      alph(k)=gnq1(k);
      bet(k)=gnq2(k);
    end
  end
end
if itype==2
  xxs1=xs1b;xxs2=xs2b;xxp1=xp1b;xxp2=xp2b;
  D=gconstb;
  N2=nc;
```

```
    if nr==1 N2=1+nc; gamma(1)=gnrb(1);end
    if nc > 0
      for k=1:nc
        gamma(nr+k)=gncb(k);
      end
    end
    if nq > 0
      N1=nq
      for k=1:nq
        alph(k)=gnq1b(k);
        bet(k)=gnq2b(k);
      end
    end
  end
  N2=nr+nc;
  disp('Print P(x)=D(x^2+alph(1)x^2+bet(1))*...')
  disp('*(x^2+alph(N_1)x^2+bet(N_1))*(x+gamma(1))*...')
  disp('*(x+gamma(N2)')
  N1
  N2
  D
  alph
  bet
  gamma
  figure(9)
  bb(1)=xxs1;bb(2)=xxs2;bb(3)=xxp1;bb(4)=xxp2;
  bb(5)=D;bb(6)=N1;bb(7)=N2;
  z1(1)=0;z2(1)=0;
  plot(z1,z2);axis([-100 0 -100 0]);
  step=-100*2/(2*(N1+N2+3)+1);
  if itype==1
title('Unscaled Additional Tap Coefficients in Case A')
  end
  if itype==2
title('Unscaled Additional Tap Coefficients in Case B')
  end
  text(-90, step, [num2str(N1), ' second-order sections']);
  text(-90, 2*step, [num2str(N2), ' first-order sections']);
  text(-90, 3*step, ['C = ', num2str(D,7)]);
  if N1 > 0
    for k=1:N1
      b1=1;
```

```
      b2=alph(k);
      b3=bet(k);
      bb(length(bb)+1)=b1;
      bb(length(bb)+1)=b2;
      bb(length(bb)+1)=b3;
      l=3+k
  text(-90,l*step,...
  ['b',num2str(k),'(',num2str(2),') = ',num2str(b1,7),...
  ', b',num2str(k),'(',num2str(1),') = ',num2str(b2,7),...
  ', b',num2str(k),'(',num2str(0),') = ',num2str(b3,7),...
  ]);
    end
  end
  if N2 > 0
    for k=1:N2
      b1=1;
      b2=gamma(k);
      bb(length(bb)+1)=b1;
      bb(length(bb)+1)=b2;
      l=3+N1+k
  text(-90,l*step,...
  ['c',num2str(k),'(',num2str(1),') = ',num2str(b1,7),...
  ', c',num2str(k),'(',num2str(0),') = ',num2str(b2,7),...
  ]);
    end
  end
hui=rot90(rot90(rot90(bb)));
save linfir hui -ascii -double
```

```
% Matlab-file subfir.m for analysing filters
% constructed using  identical FIR subfilters
% as building blocks. The file extralin.m generates
% linfir containg the coefficients of P(x) as well
% as its edges
% can be found in SUN's: ~ts/matlab/sldsp
clear all
close all
load linfir;
bb=linfir;
  xs1=bb(1);xs2=bb(2);xp1=bb(3);xp2=bb(4);N1=bb(6);
  N2=bb(7);C=bb(5);le=7;
  if N1>0;
    for k=1:N1
      le=le+1;b(k,3)=bb(le);
      le=le+1;b(k,2)=bb(le);
      le=le+1;b(k,1)=bb(le);
    end
  end
  if N2>0
    for k=1:N2
      le=le+1;c(k,2)=bb(le);
      le=le+1;c(k,1)=bb(le);
    end
  end
%
% load the subfilter
%
  load hsub
  hsub=rot90(hsub);
%
% Subfilter response
%
  [H,z]=zeroam(hsub,.0,1.,10000);
  figure(1)
  plot(z/pi,H);grid;title('Subfilter');
  ylabel('Zero-phase frequency response');
  xlabel('Angular frequency as a fraction of pi');
%
%Passband and stopband minimum and maximum as well as
%the overall response for P(x) and for the composite
%filter
```

```
%
 xp=(xp1:.0001:xp2);
 xs=(xs1:.0001:xs2);
 xx=(xs1:.0001:xp2);
 Pp=C*ones(size(xp));
 Ps=C*ones(size(xs));
 P=C*ones(size(xx));
 Pove=C*ones(size(H));
 if N1 > 0
  for k=1:N1
   cc3=b(k,3);
   cc2=b(k,2);
   cc1=b(k,1);
   pp=cc3*xp.*xp+cc2*xp+cc1;
   ps=cc3*xs.*xs+cc2*xs+cc1;
   p=cc3*xx.*xx+cc2*xx+cc1;
   pove=cc3*H.*H+cc2*H+cc1;
   Pp=Pp.*pp;
   Ps=Ps.*ps;
   P=P.*p;
   Pove=Pove.*pove;
  end
 end
 if N2 > 0
  for k=1:N2
   cc2=c(k,2);
   cc1=c(k,1);
   pp=cc2*xp+cc1;
   ps=cc2*xs+cc1;
   p=cc2*xx+cc1;
   pove=cc2*H+cc1;
   Pp=Pp.*pp;
   Ps=Ps.*ps;
   P=P.*p;
   Pove=Pove.*pove;
  end
 end
 apmin=min(Pp);
 apmax=max(Pp);
 asmax=max(Ps);
 asmin=min(Ps);
 ass=max(abs(Ps));
```

```
figure(2)
AP=1.5*20*log10(abs(apmax));
AS=1.5*20*log10(ass);
ave=(apmax+apmin)/2;dp=apmax-ave;
APA=ave+1.2*dp;API=ave-1.2*dp;
plot(xx,20*log10(abs(P)));axis([xs1 xp2 AS AP]);
grid;
title(['P(x): xs1=',num2str(xs1),', xs2='...
,num2str(xs2),', xp1=',num2str(xp1),', xp2='...
,num2str(xp2)])
ylabel('abs(P(x)) in dB'); xlabel('x');
figure(3)
subplot(211)
plot(xp,Pp);axis([xp1 xp2  API APA]);
grid;
title(['Passband details'])
ylabel('P(x)'); xlabel('x');
subplot(212)
plot(xs,Ps);axis([xs1 xs2  1.2*asmin 1.2*asmax]);grid;
title(['Stopband details'])
ylabel('P(x)'); xlabel('x');
figure(4)
plot(z/pi,20*log10(abs(Pove)));axis([0 1 AS AP]);
;grid;
title(['Overall filter']);
ylabel('Amplitude in dB');
xlabel('Angular frequency as a fraction of pi');
figure(5)
subplot(211)
plot(z/pi,Pove);axis([0 1  API APA]);
grid;
title(['Passband details'])
ylabel('Zero-phase response');
xlabel('Angular frequency as a fraction of pi');
subplot(212)
plot(z/pi,Pove);axis([0 1  1.2*asmin 1.2*asmax]);grid;
title(['Stopband details'])
ylabel('Zero-phase response');
xlabel('Angular frequency as a fraction of pi');
```

Example: $\omega_p = 0.4\pi$, $\omega_s = 0.402\pi$, $\delta_p = 0.01$, **and** $\delta_s = 0.0001$.

- The following table gives the minimum subfilter orders $2M$ for various values of N, the number of subfilters, along with the subfilter specifications in Cases A and B as well as the number of distinct coefficients $N + M + 2$, and the overall filter order, $2MN$.
- $N = 1$ corresponds to the direct-form design.

TABLE 4-12 Data for Filters Synthesized Using Identical Subfilters

Number of Subfilters	Subfilter Order	$\cos \Omega_p$ / $\cos \Omega_s$	$\overline{\delta}_p$ / $\overline{\delta}_s$	Number of Distinct Coefficients	Overall Filter Order
$N = 1$	3138			1570	3138
$N = 2$	2056	0.98038	0.005000	1032	4112
		−0.94450	0.014142		
$N = 4$	1046	0.79100	0.057687	529	4184
		−0.83208	0.046348		
$N = 6$	692	0.58445	0.125781	354	4152
		−0.71930	0.084963		
$N = 8$	514	0.43774	0.183502	267	4112
		−0.62629	0.121968		
$N = 10$	408	0.33818	0.228955	216	4080
		−0.55241	0.154843		
$N = 15$	268	0.29826	0.266590	151	4020
		−0.33403	0.252999		
$N = 20$	200	0.28324	0.288587	122	4000
		−0.20044	0.321925		
$N = 30$	132	0.17262	0.355611	98	3960
		−0.15404	0.363594		
$N = 40$	98	0.11975	0.391903	91	3920
		−0.12635	0.388965		
$N = 50$	78	0.08953	0.414360	91	3900
		−0.10775	0.406070		

- It is interesting to observe that the overall filter order for all the cases is approximately 1.3 times (within 1.24–1.33) that of the direct-form design.
- If all the identical subfilters are implemented separately, then the overall multiplication rate per sample, $N(M+1)+N+1$, is higher than that of the direct-form equivalent.
- However, the structures of transparency 2 become advantageous if all the subfilters are implemented using a single subfilter by applying multiplexing.
- Since the subfilter order can be reduced to any value by increasing the number of subfilters, it has the potential of being realized by a fast short convolution algorithm or implemented using an integrated FIR filter chip.

Design of FIR Filters Without General Multipliers

- Using the second overall structure of transarency 2, it is relatively easy to design high-order filters without general multipliers.
- Filters of this kind are very attractive in VLSI implementation where a general multiplier is very costly.
- These filters can be designed in two steps.
- In the first step, the additional tap coefficients of the second structure are quantized to values which are simple combinations of powers-of-two.
- The second step then involves designing the subfilter in such a way that there are no general multipliers.

- It is relatively easy to get such a subfilter without time-consuming optimization since the ripple values of the subfilter are very large and, consequently, large coefficient quantization errors are allowed.
- The rule of thumb for direct rounding of FIR filter coefficients is that if the allowed quantization error is made double, one bit is saved.
- Also the order of the subfilter is significantly reduced compared to the order of the overall filter.
- Another rule of thumb for direct rounding is that if there are two filters with the same allowable quantization error and the order of the first filter is one fourth that of the second filter, then the first filter requires one bit less.

Performing the First Step

- To allow some quantization error for the tap coefficients, $G(\Omega)$ of the given order $2N$ is first designed to be the best extraripple solution for the passband and stopband ripples of $0.8\delta_p \cdots 0.9\delta_p$ and $0.8\delta_s \cdots 0.9\delta_s$.
- This $G(\Omega)$ is then converted to $P(x)$ according to the previous discussion and the passband and stopband regions of $P(x)$, $[x_{p1}, x_{p2}]$ and $[x_{s1}, x_{s2}]$, are located.
- In both Case A and Case B, the resulting $P(x)$ can be factored in the form

$$P(x) = D \prod_{k=1}^{N_1} (x^2 + \alpha_k x + \beta_k) \prod_{k=1}^{N_2} (x + \gamma_k).$$

- A very straightforward technique to arrive at simple tap coefficients is based on expressing the coefficients of the second- and first-order terms as

$$\alpha_k = b_k[1]/b_k[2], \quad \beta_k = b_k[0]/b_k[2] \qquad (A)$$

$$\gamma_r = c_k[0]/c_k[1].$$

- The resulting $P(x)$ can be written in the following form corresponding to the second structure:

$$P(x) = C \prod_{k=1}^{N_1} (b_k[2]x^2 + b_k[1]x + b_k[0]) \prod_{k=1}^{N_2} (c_k[1]x + c_k[0]).$$

- If the coefficients of a second-order term are desired to be quantized to two powers-of-two values, i.e. values of the form $\pm 2^{-P_1} \pm 2^{-P_2}$, then a simple technique is to first set $b_k[2]$ to take all possible two powers-of-two values within 1/2 and 1.

- Then, for each value of $b_k[2]$, the remaining coefficient values $b_k[1]$ and $b_k[0]$ are determined from Eq. (A) and quantized to the closest two powers-of-two values. Finally, those values which provide the closest approximations to α_k and β_k are selected.

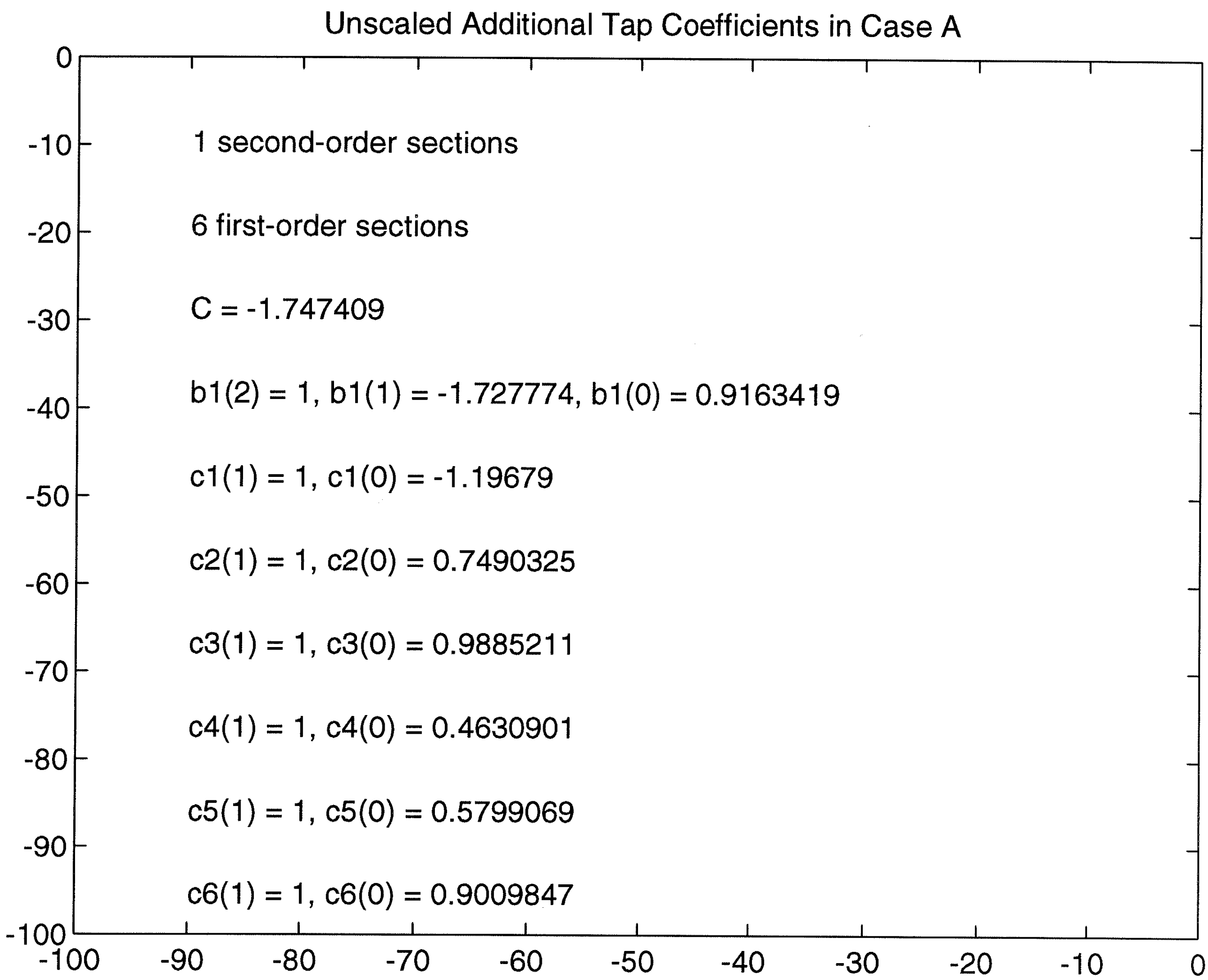
- Quantized values for the $c_k[1]$'s and $c_k[0]$'s can be found in the same manner.
- Then, C is selected such that the average of $P(x)$ in the passband region $[x_{p1}, x_{p2}]$ is unity and it is checked whether $P(x)$ is within the limits $1 \pm \delta_p$ in the passband region $[x_{p1}, x_{p2}]$ and within the limits $\pm\delta_s$ in the stopband region $[x_{s1}, x_{s2}]$.
- If not, some of the coefficients require three powers-of-two representations.
- What remains is to design a multiplier-free subfilter such that its zero-phase frequency response $F_M(\omega)$ stays within x_{p1} and x_{p2} in the passband region and within x_{s1} and x_{s2} in the stopband region.

Example: $\delta_p = 0.01$ and $\delta_s = 0.0001$ and $N = 8$ subfilters are used.

- The figure of transparency 36 gives the best extraripple solution of $G(\Omega)$ for the ripple values $0.9\delta_p$ and $0.9\delta_s$ as well as the corresponding polynomial $P(x)$ in both Case A and Case B together with its passband and stopband regions.
- The following two transparencies give the unquantized coefficient values of polynomials $P(x)$ for both Cases A and B.
- In both cases, $P(x)$ still meets the given criteria when the additional tap coefficents are quantized, using the above procedure, to the values shown in the table of transparency 52.
- In Case A, it is required that $F_M(\omega)$ stays within the limits 0.4493 and 1 in the passband(s) and within the limits -1 and -0.6318 in the stopband(s).
- In Case B, the required passband and stopband ripples are $\widehat{\delta_p} = 0.1787$ and $\widehat{\delta_s} = 0.1195$, respectively.

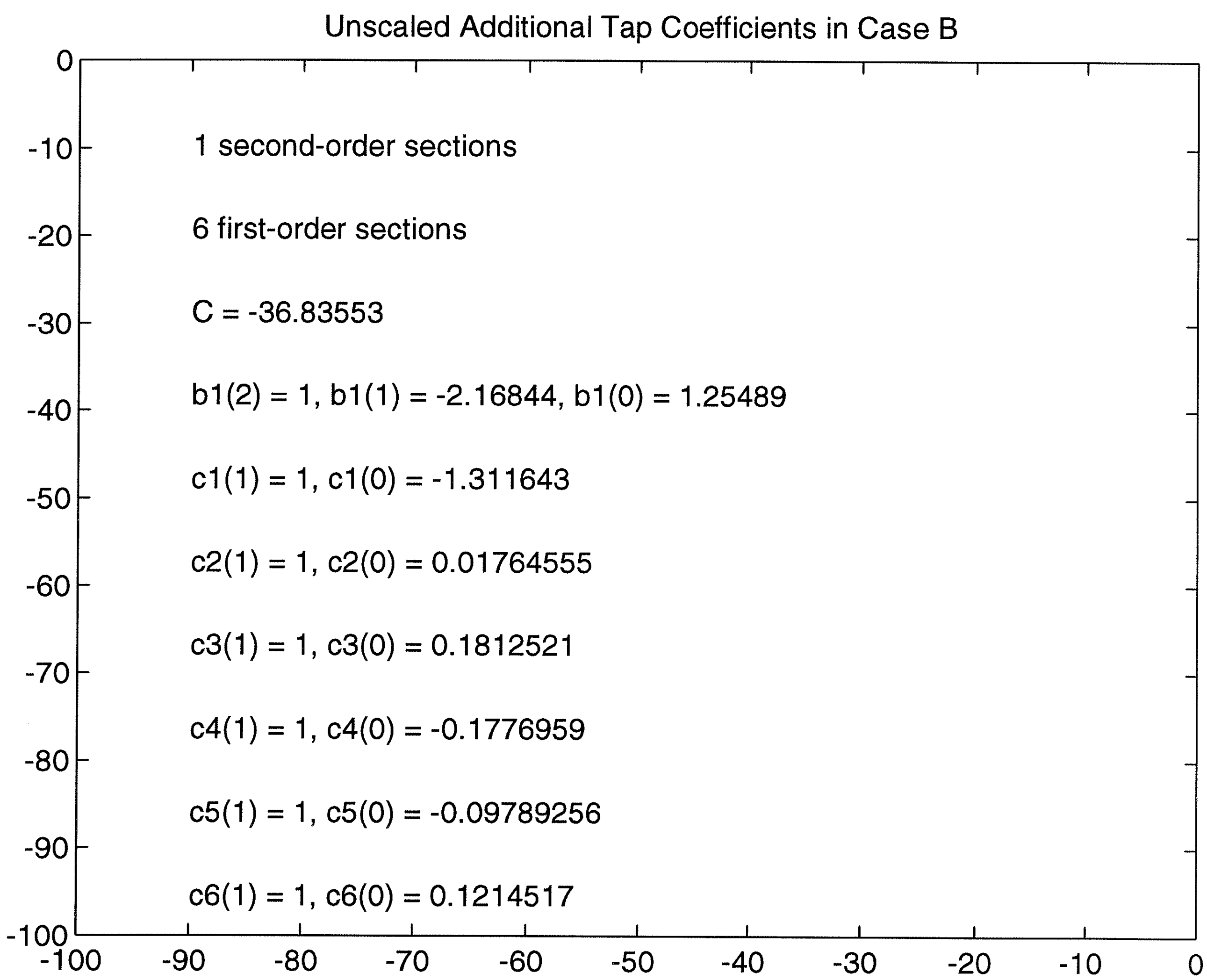
Unquantized Tap Coefficients of $P(x)$ in Case A: $\alpha_k \equiv b_k(1)$, $\beta_k \equiv b_k(0)$, $\gamma_k \equiv c_k(0)$

- The corresponding $P(x)$ can be found after page 39 in the case of the extraripple solution with four passband extrema.



Unquantized Tap Coefficients of $P(x)$ in Case B: $\alpha_k \equiv b_k(1)$, $\beta_k \equiv b_k(0)$, $\gamma_k \equiv c_k(0)$

- The corresponding $P(x)$ can be found after page 39 in the case of the extraripple solution with four passband extrema.



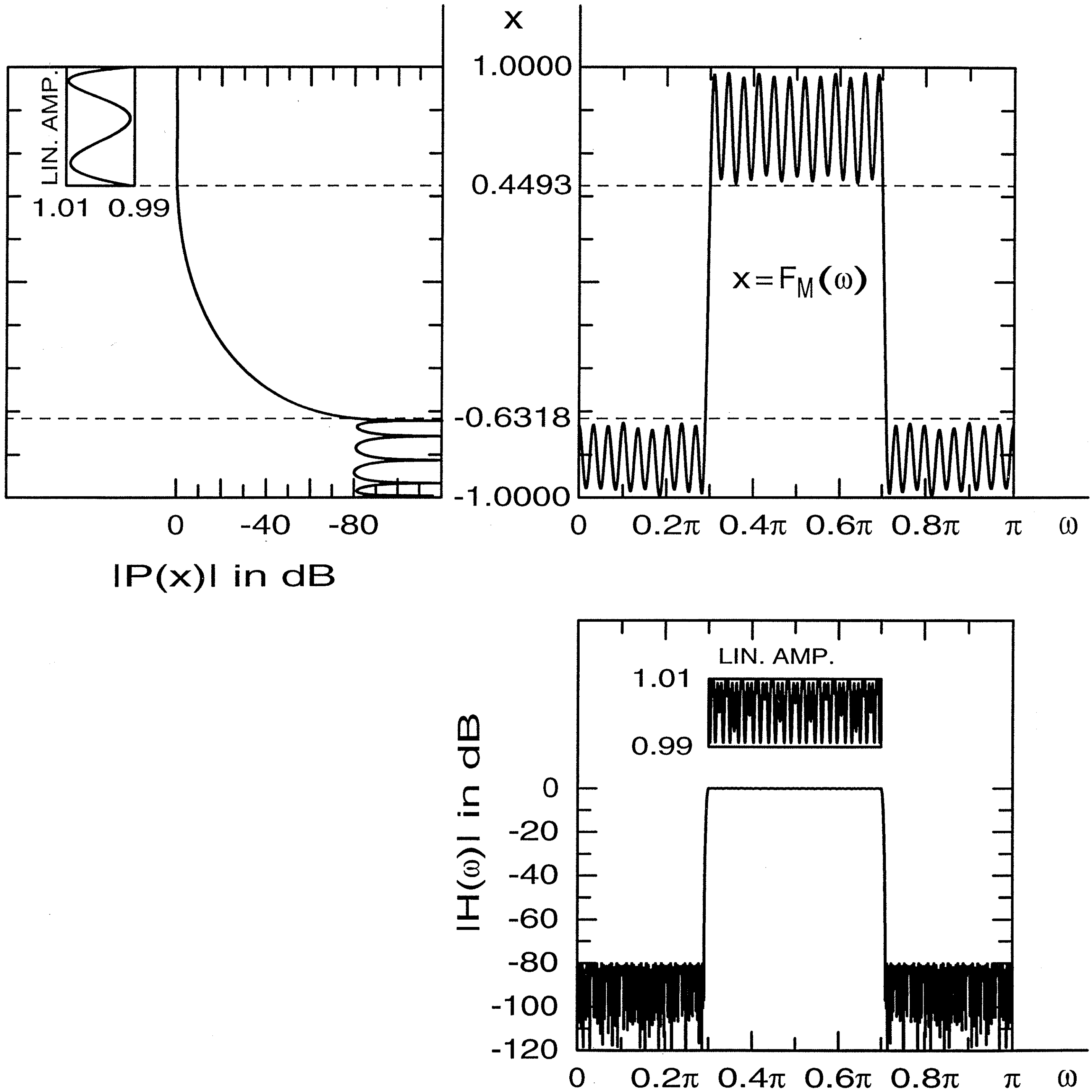
Quantized Tap Coefficients

	Case A	
$b_1[2] = 2^{-1} + 2^{-6}$	$b_1[1] = -2^0 + 2^{-3} - 2^{-7}$	$b_1[0] = 2^{-1} - 2^{-6} - 2^{-7}$
$c_1[1] = 2^{-1} + 2^{-2}$	$c_1[0] = -2^0 + 2^{-4} + 2^{-5}$	
$c_2[1] = 2^0$	$c_2[0] = 2^0 - 2^{-7}$	
$c_3[1] = 2^0 - 2^{-4}$	$c_3[0] = 2^0 - 2^{-3}$	
$c_4[1] = 2^{-1} + 2^{-3}$	$c_4[0] = 2^{-1} + 2^{-6}$	
$c_5[1] = 2^0 - 2^{-3}$	$c_5[0] = 2^{-1} + 2^{-3}$	
$c_6[1] = 2^0 - 2^{-3}$	$c_6[0] = 2^{-1} + 2^{-4}$	
$C = -2^2 - 2^1 - 2^{-1} - 2^{-5}$		
	Case B	
$b_1[2] = 2^0 - 2^{-3} + 2^{-6}$	$b_1[1] = -2^1 + 2^{-4} + 2^{-8}$	$b_1[0] = 2^0 + 2^{-3} - 2^{-8}$
$c_1[1] = 2^{-1} + 2^{-4} + 2^{-7}$	$c_1[0] = -2^{-1} - 2^{-2}$	
$c_2[1] = 2^{-1} + 2^{-6}$	$c_2[0] = 2^{-4} - 2^{-8}$	
$c_3[1] = 2^{-1} + 2^{-3}$	$c_3[0] = 2^{-5} + 2^{-6}$	
$c_4[1] = 2^0$	$c_4[0] = 2^{-7}$	
$c_5[1] = 2^0 - 2^{-5}$	$c_5[0] = -2^{-4}$	
$c_6[1] = 2^0 - 2^{-5}$	$c_6[0] = -2^{-3} + 2^{-6}$	
$C = -2^8 + 2^4 + 2^3 - 2^0$		

Case A Bandpass Filter with $\omega_{p1}, \omega_{p2} = 0.5\pi \pm 0.2\pi$ and $\omega_{s1}, \omega_{s2} = 0.5\pi \pm 0.21\pi$

- In Case A, it is required that $F_M(\omega)$ stays within the limits 0.4493 and 1 in the passbands and within the limits -1 and -0.6318 in the stopbands (see the following transparency).
- The minimum even subfilter order to meet the criteria is 112.
- If the subfilter order is increased to 120, then the given criteria are still met when direct rounding is used to quantize the coefficients to the closest two powers-of-two values in 8-bit representations (see the following transparency).
- These values are shown in the table of transparency 55.

Responses for the Case A Bandpass Filter



Coefficients for Case A Bandpass Subfilters

TABLE 4-14 Quantized Coefficients[a] for a Bandpass Subfilter of Order 120

$f[0] = 8 \times 2^{-8}$	$f[2] = 20 \times 2^{-8}$	$f[4] = -15 \times 2^{-8}$	$f[6] = 6 \times 2^{-8}$	$f[8] = 3 \times 2^{-8}$
$f[10] = -4 \times 2^{-8}$	$f[12] = -2 \times 2^{-8}$	$f[14] = 5 \times 2^{-8}$	$f[16] = -1 \times 2^{-8}$	$f[18] = -5 \times 2^{-8}$
$f[20] = 4 \times 2^{-8}$	$f[22] = 4 \times 2^{-8}$	$f[24] = -6 \times 2^{-8}$	$f[26] = -1 \times 2^{-8}$	$f[28] = 7 \times 2^{-8}$
$f[30] = -4 \times 2^{-8}$	$f[32] = -6 \times 2^{-8}$	$f[34] = 8 \times 2^{-8}$	$f[36] = 3 \times 2^{-8}$	$f[38] = -12 \times 2^{-8}$
$f[40] = 4 \times 2^{-8}$	$f[42] = 12 \times 2^{-8}$	$f[44] = -12 \times 2^{-8}$	$f[46] = -7 \times 2^{-8}$	$f[48] = 20 \times 2^{-8}$
$f[50] = -4 \times 2^{-8}$	$f[52] = -28 \times 2^{-8}$	$f[54] = 28 \times 2^{-8}$	$f[56] = 34 \times 2^{-8}$	$f[58] = -120 \times 2^{-8}$
$f[60] = -48 \times 2^{-8}$				

[a] $f[n]$ is zero for n odd.

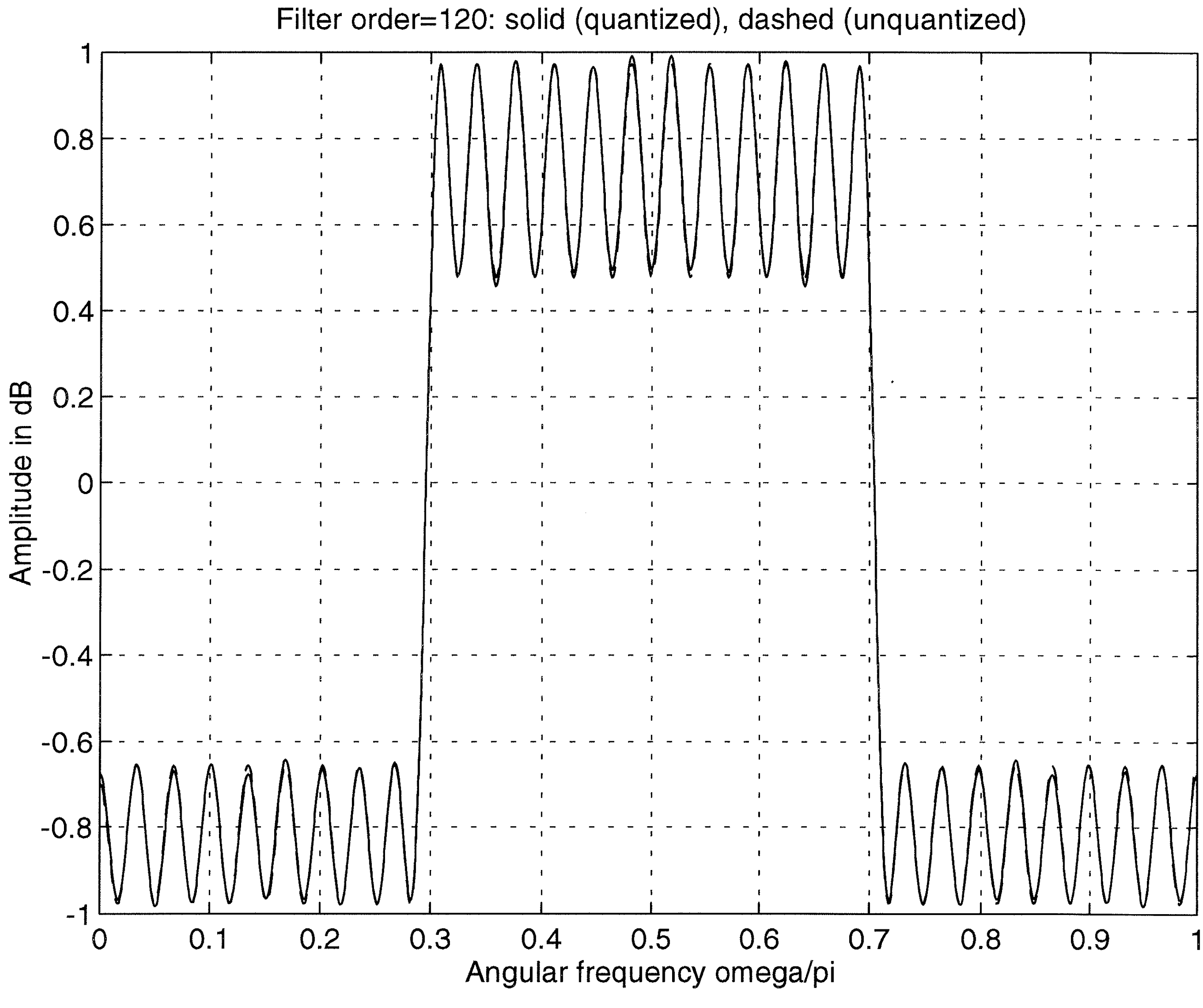
TABLE 4-15 Quantized Coefficients[a] for a Bandpass Subfilter of Order 136

$f[0] = -2 \times 2^{-6}$	$f[2] = 5 \times 2^{-6}$	$f[4] = 0$	$f[6] = 0$	$f[8] = 1 \times 2^{-6}$
$f[10] = 0$	$f[12] = -1 \times 2^{-6}$	$f[14] = 0$	$f[16] = 1 \times 2^{-6}$	$f[18] = -1 \times 2^{-6}$
$f[20] = -1 \times 2^{-6}$	$f[22] = 1 \times 2^{-6}$	$f[24] = 0$	$f[26] = -1 \times 2^{-6}$	$f[28] = 1 \times 2^{-6}$
$f[30] = 1 \times 2^{-6}$	$f[32] = -2 \times 2^{-6}$	$f[34] = 0$	$f[36] = 2 \times 2^{-6}$	$f[38] = -1 \times 2^{-6}$
$f[40] = -2 \times 2^{-6}$	$f[42] = 2 \times 2^{-6}$	$f[44] = 1 \times 2^{-6}$	$f[46] = -3 \times 2^{-6}$	$f[48] = 1 \times 2^{-6}$
$f[50] = 3 \times 2^{-6}$	$f[52] = -3 \times 2^{-6}$	$f[54] = -2 \times 2^{-6}$	$f[56] = 5 \times 2^{-6}$	$f[58] = -1 \times 2^{-6}$
$f[60] = -7 \times 2^{-6}$	$f[62] = 7 \times 2^{-6}$	$f[64] = 8 \times 2^{-6}$	$f[66] = -30 \times 2^{-6}$	$f[68] = -12 \times 2^{-6}$

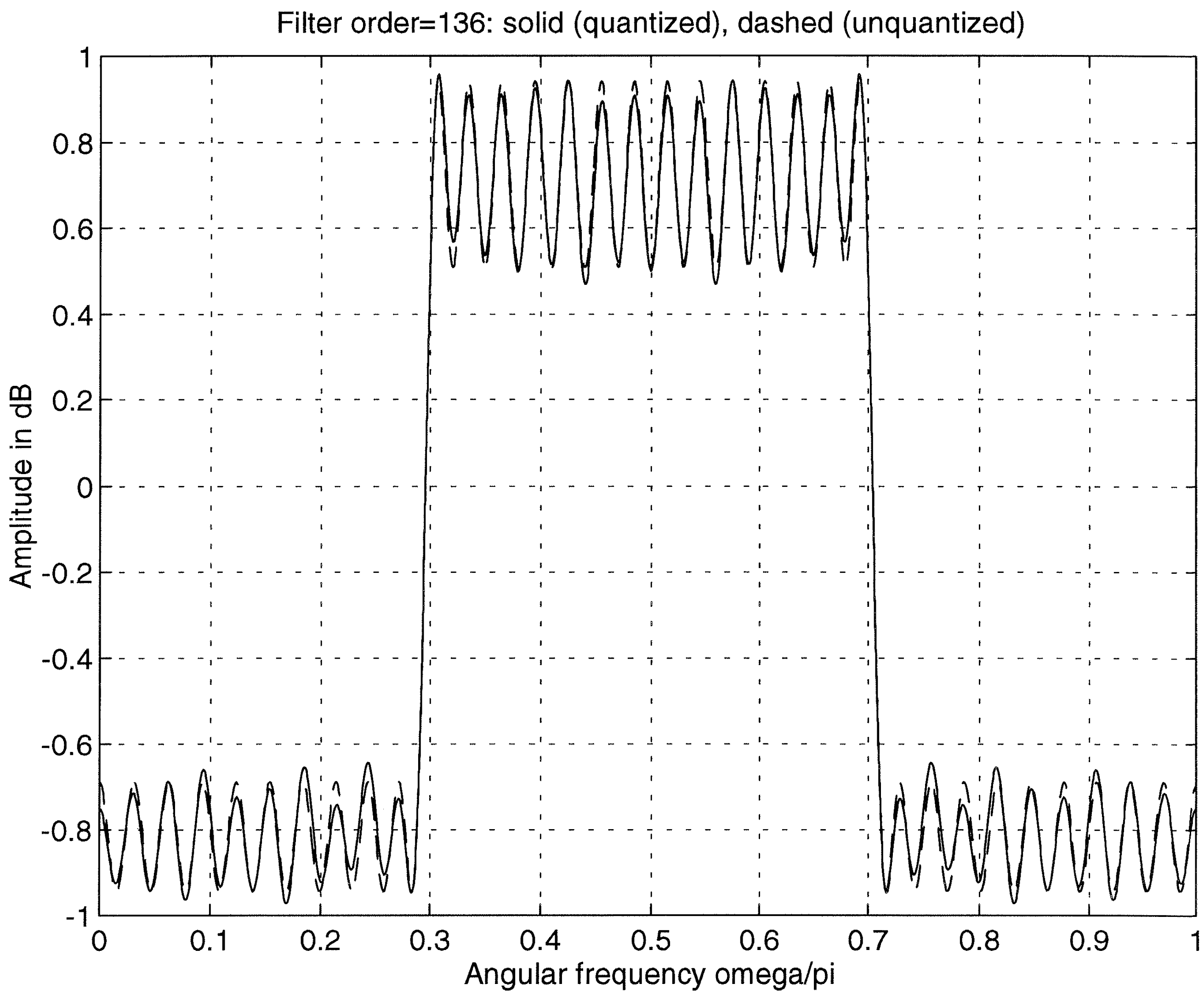
[a] $f[n]$ is zero for n odd.

- The overall filter order is 960, whereas the minimum order of an equivalent conventional direct-form design is 636.
- The price paid for getting a multiplier-free design is thus a fifty percent increase in the filter order.
- If the subfilter order is increased to 136, then with direct rounding we end up with the very simple six-bit coefficient values of the table of the previous transparency.
- The following transparencies give a Matlab-file for designing the above-mentioned bandpass filters as well as the responses for the quantized and unquantized filters.
- Please use extralin.m with passband ripple equal to 0.009, stopband ripple equal to 0.0009, and the number of subfilters equal to 8. Select the Case A. Use then firband.m and finally subfir.m. Check the responses obtained by subfir.m

```
% Matlab-file firband.m for designing
% quantized bandpass filters
% can be found in SUN's: ~ts/matlab/sldsp
f=[0. .29 .3 .7 .71 1.];
m=[-0.8159 -0.8159 0.7247 0.7247 -0.8159 -0.8159];
w=[1.55 1 1.55];
h120=remez(120, f, m, w);
w=[1.7 1 1.7];
h136=remez(136, f, m, w);
hs120=round(h120*2^8)/(2^8)
hs136=round(h136*2^6)/(2^6)
figure(1)
[H,W]=zeroam(h120,.0,1.,2000);
[H1,W1]=zeroam(hs120,.0,1.,2000);
plot(W/pi,H,'- -',W/pi,H1);axis([0 1 -1 1]);grid;
ylabel('Amplitude in dB'); xlabel('Angular frequency omega/pi');
title('Filter order=120: solid (quantized), dashed (unquantized)')
figure(2)
[H,W]=zeroam(h136,.0,1.,2000);
[H1,W1]=zeroam(hs136,.0,1.,2000);
plot(W/pi,H,'- -',W/pi,H1);axis([0 1 -1 1]);grid;
ylabel('Amplitude in dB'); xlabel('Angular frequency omega/pi');
title('Filter order=136: solid (quantized), dashed (unquantized)')
hui=rot90(hs120);
save hsub hui -ascii -double
```



Filter order=120: solid (quantized), dashed (unquantized)
Amplitude in dB
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Angular frequency omega/pi


Filter order=136: solid (quantized), dashed (unquantized)
Amplitude in dB
Angular frequency omega/pi
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Case B Lowpass Filter with $\omega_p = 0.4\pi$ and $\omega_s = 0.402\pi$.

- In Case B, the required passband and stopband ripples are $\widehat{\delta}_p = 0.1787$ and $\widehat{\delta}_s = 0.1195$, respectively.
- Using the frequency-response masking approach, the given lowpass filter criteria are met by a subfilter of the form $F_M(z) = F(z^L)G_1(z) + [z^{-N_F L/2} - F(z^L)]G_2(z)$, where $L = 16$, the order of $F(z)$ is 40, and the orders of $G_1(z)$ and $G_2(z)$ are 22 and 30, respectively.
- This filter has been slightly overdesigned such that direct rounding can be used to quantize the filter coefficients to the 6-bit values shown in the table of the following transparency.
- Only one coefficient ($g_2[14] = 19 \cdot 2^{-6}$) requires a three powers-of-two representation.

Coefficient Values for the Case B Lowpass Subfilter Designed Using the Frequency-Reponse Masking Approach

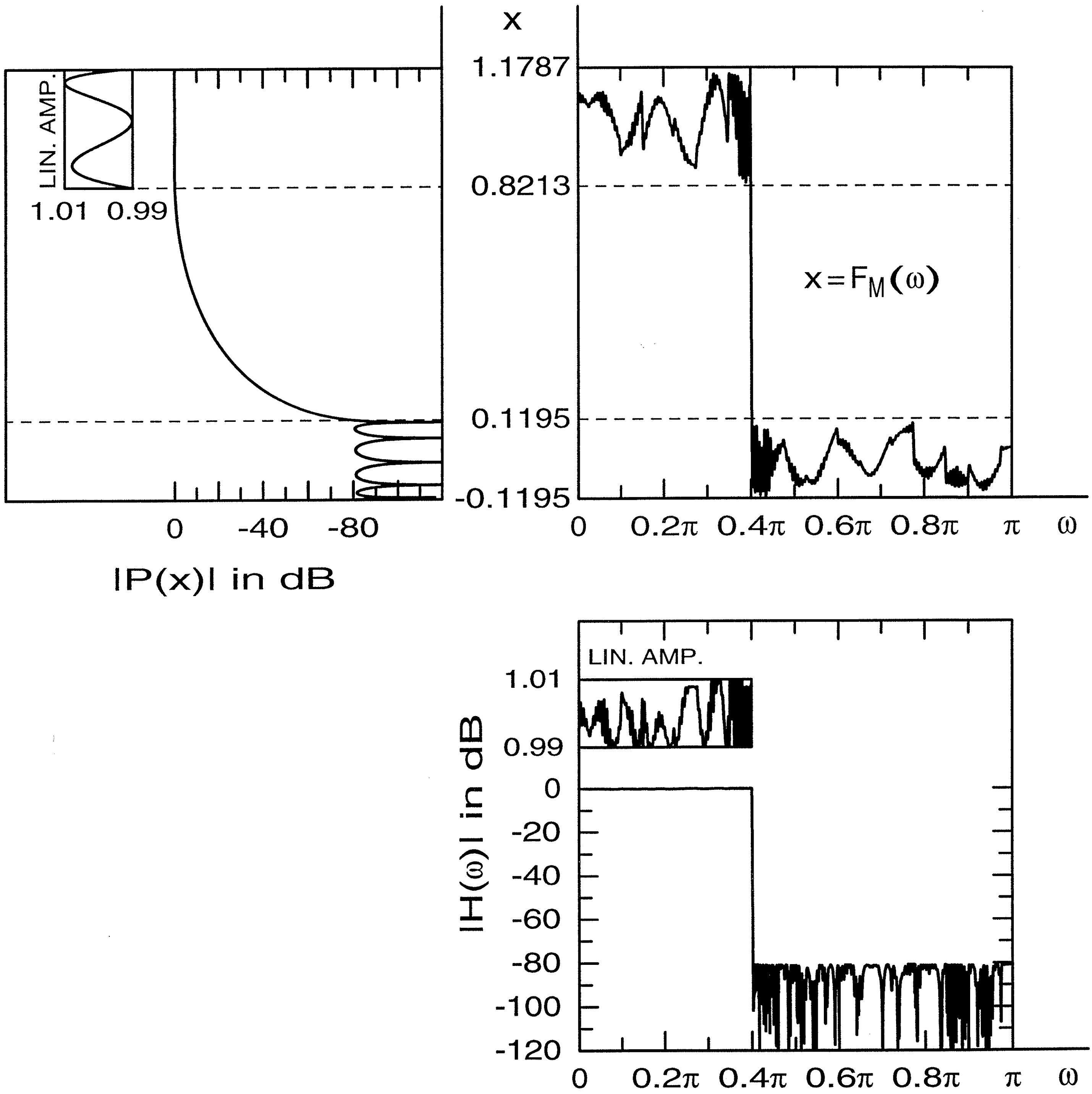
TABLE 4-16 Quantized Coefficients for a Subfilter Designed Using the Frequency-Response Masking Approach

$f[0] = 2 \times 2^{-6}$	$f[1] = -4 \times 2^{-6}$	$f[2] = -3 \times 2^{-6}$	$f[3] = -2 \times 2^{-6}$	$f[4] = 1 \times 2^{-6}$
$f[5] = 0$	$f[6] = -1 \times 2^{-6}$	$f[7] = -2 \times 2^{-6}$	$f[8] = 0$	$f[9] = 2 \times 2^{-6}$
$f[10] = 1 \times 2^{-6}$	$f[11] = -2 \times 2^{-6}$	$f[12] = -3 \times 2^{-6}$	$f[13] = 1 \times 2^{-6}$	$f[14] = 4 \times 2^{-6}$
$f[15] = 1 \times 2^{-6}$	$f[16] = -5 \times 2^{-6}$	$f[17] = -6 \times 2^{-6}$	$f[18] = 5 \times 2^{-6}$	$f[19] = 17 \times 2^{-6}$
$f[20] = 28 \times 2^{-6}$				
$g_1[0] = 2 \times 2^{-6}$	$g_1[1] = 3 \times 2^{-6}$	$g_1[2] = -1 \times 2^{-6}$	$g_1[3] = -2 \times 2^{-6}$	$g_1[4] = -1 \times 2^{-6}$
$g_1[5] = 3 \times 2^{-6}$	$g_1[6] = 2 \times 2^{-6}$	$g_1[7] = -4 \times 2^{-6}$	$g_1[8] = -5 \times 2^{-6}$	$g_1[9] = 4 \times 2^{-6}$
$g_1[10] = 20 \times 2^{-6}$	$g_1[11] = 28 \times 2^{-6}$			
$g_2[0] = -3 \times 2^{-6}$	$g_2[1] = -1 \times 2^{-6}$	$g_2[2] = 1 \times 2^{-6}$	$g_2[3] = 2 \times 2^{-6}$	$g_2[4] = 1 \times 2^{-6}$
$g_2[5] = -1 \times 2^{-6}$	$g_2[6] = -2 \times 2^{-6}$	$g_2[7] = 0$	$g_2[8] = 3 \times 2^{-6}$	$g_2[9] = 3 \times 2^{-6}$
$g_2[10] = -1 \times 2^{-6}$	$g_2[11] = -5 \times 2^{-6}$	$g_2[12] = -2 \times 2^{-6}$	$g_2[13] = 7 \times 2^{-6}$	$g_2[14] = 19 \times 2^{-6}$
$g_2[15] = 24 \times 2^{-6}$				

Comments

- Note that no optimization has been used in finding these coefficient values. The overall filter order is 70 percent higher than that of a direct-form equivalent (5360 compared to 3138).
- The filter responses are given in the following transparency.
- In the above, direct rounding has been used for quantizing the subfilter coefficients.
- Another technique, leading to better results, is to use mixed integer linear programming as proposed by Lim.

Responses for the Case B Lowpass Filter



Responses for the Case B Lowpass Filter