Chapter 8

The Z-Transform

Consider an input $x[n] = z^n, n \in \mathbb{Z}$ to an LTI system with impulse response $h[n]$. Then

$$y[n] = \sum_{k=-\infty}^{+\infty} h[k]z^{n-k} = z^n \underbrace{\sum_{k=-\infty}^{+\infty} h[k]z^{-k}}_{H(z)} = H(z)z^n. \tag{8.1}$$

Therefore, z^n is an eigenfunction of an LTI system with eigenvalue $H(z)$. As before, a natural question to ask is whether the summation exists. In the case of the summation defined as the Z-transform

$$X(z) = \sum_{n=-\infty}^{+\infty} x[n]z^{-n}, \tag{8.2}$$

the answer can be found by representing $z = re^{j\omega}$, with $r = |z|$. Hence

$$X(z) = X\left(re^{j\omega}\right) = \sum_{n=-\infty}^{+\infty} \left(x[n]r^{-n}\right) e^{-j\omega n} = \sum_{n=-\infty}^{+\infty} \breve{x}[n]e^{-j\omega n},$$

where $\breve{x}[n] = x[n]r^{-n}$. Therefore, we can see that the Z-transform is nothing but the Fourier transform of a scaled sequence, i.e.

$$X(z) = X\left(re^{j\omega}\right) = \mathcal{F}\left\{x[n]r^{-n}\right\}. \tag{8.3}$$

Hence

$$X(z)\Big|_{z=e^{j\omega}} = X\left(e^{j\omega}\right) = \mathcal{F}\left\{x[n]\right\}. \tag{8.4}$$

Now it is clear that for a Z-transform to exist, we would need

$$\sum_{n=-\infty}^{+\infty} \left|x[n]r^{-n}\right| < \infty, \tag{8.5}$$

which for appropriately chosen r would be possible. This already tells us that by choosing r appropriately we can always properly define the Z-transform. The range of values of z for which the Z-transform exists is called the *region-of-convergence*.

Example 8.1 *Let* $x[n] = a^n u[n]$. *Now*

$$X(z) = \sum_{n=-\infty}^{+\infty} a^n u[n] z^{-n} = \sum_{n=0}^{\infty} a^n z^{-n} = \sum_{n=0}^{\infty} \left(az^{-1}\right)^n.$$

Therefore this needs $\left|az^{-1}\right| < 1$ *or* $|z| > |a|$, *hence*

$$X(z) = \frac{1}{1 - az^{-1}} \qquad \textit{for} \quad |z| > |a|. \tag{8.6}$$

Thus the Z-transform is well defined for any $a \in \mathbb{C}$.

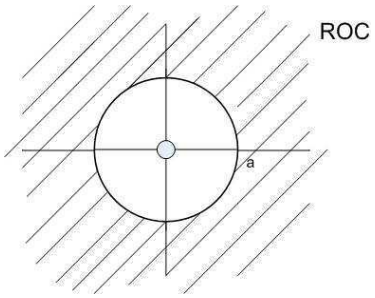


Figure 8.1: Pole-zero plot and region of convergence for Example 8.1.

Example 8.2 *Let* $x[n] = -a^n u[-n-1]$. *Now*

$$X(z) = -\sum_{n=-\infty}^{+\infty} a^n u[-n-1] z^{-n} = \sum_{n=-\infty}^{-1} -a^n z^{-n} = -\sum_{n=1}^{\infty} \left(a^{-1}z\right)^n$$

$$= 1 - \frac{1}{1-a^{-1}z} = \frac{1}{1-az^{-1}} \qquad for \quad \left|a^{-1}z\right| < 1 \; or \; |z| < |a|. \tag{8.7}$$

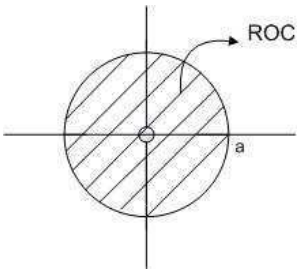


Figure 8.2: Pole-zero plot and region of convergence for Example 8.2.

Example 8.3 *Let* $x[n] = \left(\frac{1}{3}\right)^n u[n] - 2^n u[-n-1]$. *Now*

$$\begin{aligned} X(z) &= \sum_{n=0}^{\infty} \left(\frac{1}{3}z^{-1}\right)^n - \sum_{n=1}^{\infty} \left(2^{-1}z\right)^n \\ &= \frac{1}{1-\frac{1}{3}z^{-1}} - \left\{\frac{1}{1-2^{-1}z} - 1\right\} \qquad for \quad \left|\frac{1}{3}z^{-1}\right| < 1 \;\; and \;\; \left|2^{-1}z\right| < 1 \\ &= \frac{1}{1-\frac{1}{3}z^{-1}} - \left\{\frac{2^{-1}z}{1-2^{-1}z}\right\} \qquad for \quad |z| > \frac{1}{3} \;\; and \;\; |z| < 2 \\ &= \frac{1}{1-\frac{1}{3}z^{-1}} + \frac{1}{1-2z^{-1}} \qquad \frac{1}{3} < |z| < 2 \end{aligned}$$

8.1 Region of Convergence for the Z-transform

The region of convergence of a Z-transform consists of a ring in the Z-plane centered in the origin. This follows from the fact that the ROC consists of those values of $z = re^{j\omega}$ for which $x[n]r^{-n}$ has a Fourier transform. Since the convergence *only* depends on $r = |z|$, it is clear that if a specific value of z is in the ROC, then *all* values of z with the same

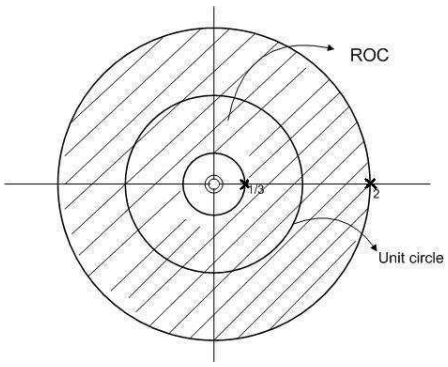


Figure 8.3: Pole-zero plot and region of convergence for Example 8.3.

magnitude are also in the ROC. Hence this guarantees that the ROC must be in concentric rings. Clearly we have the following properties for the ROC.

Property 1 The ROC is a ring or disk in the Z-plane centered at the origin, i.e.

$$0 \leq r_R \leq |z| < r_L < \infty$$

.

Property 2 The DTFT exists if and only if the ROC includes the unit circle ($|z| = 1$).

Property 3 The ROC cannot contain any poles.

Property 4 If $x[n]$ is a *finite-length* sequence, then the ROC is the entire Z-plane with the *possible* exception of $z = 0$ or $z = \infty$.

Property 5 If $x[n]$ is a right-sided sequence, i.e. $x[n] = 0$ for $n < N_1 < \infty$, for some $N_1 \in \mathbb{Z}$, then the ROC extends outwards from the *outermost* finite pole in $X(z)$.

Property 6 If $x[n]$ is a left-sided sequence, i.e. $x[n] = 0$ for $n > N_2 > -\infty$, for some $N_2 \in \mathbb{Z}$, then the ROC extends inwards from the *innermost* finite pole in $X(z)$.

Property 7 A two-sided sequence is an infinite-duration sequence that is neither right-sided nor left-sided. If $x[n]$ is a two-sided sequence, then the ROC will consist of a ring in the Z-plane bounded on the interior and exterior by a pole and does not contain any poles.

Property 8 The ROC must be a connected region.

8.2 The Inverse Z-Transform

By considering

$$X\left(re^{j\omega}\right) = \mathcal{F}\left\{x[n]r^{-n}\right\},$$

one can obtain

$$x[n]r^{-n} = \mathcal{F}^{-1}\left\{X\left(re^{j\omega}\right)\right\},$$

or

$$x[n]r^{-n} = \frac{1}{2\pi}\int_{-\pi}^{\pi} X\left(re^{j\omega}\right)e^{j\omega n}\,\mathrm{d}\omega,$$

or

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X\left(re^{j\omega}\right)\left(re^{j\omega}\right)^{n}\,\mathrm{d}\omega. \tag{8.8}$$

That is, we can recover $x[n]$ from integrating its Z-transform along a contour $z = re^{j\omega}$ in its ROC, by fixing r and varying ω from $-\pi$ to π. With $z = re^{j\omega}$, for fixed r we see that $\mathrm{d}z = jre^{j\omega}\,\mathrm{d}\omega = jz\,\mathrm{d}\omega$ or $\mathrm{d}\omega = \frac{\mathrm{d}z}{jz}$.
Since the integral in (8.8) is over a 2π interval of ω, it corresponds to a traversal around the circle $|z| = r$. Consequently

$$x[n] = \frac{1}{2\pi j}\oint X(z)z^{n-1}\,\mathrm{d}z, \tag{8.9}$$

where the symbol $\circlearrowleft$ denotes integration around a counter-clockwise closed circular contour centered at the origin and with radius of r. The value of r can be chosen such that $|z| = r$ is in the ROC.

8.3 Partial Fraction Expansion

Consider a Z-transform of the form of a rational function, *i.e.,*

$$X(z) = \frac{\sum_{k=0}^{M-1} b_k z^{-k}}{\sum_{k=0}^{N-1} a_k z^{-k}} = \frac{b_0\prod_{k=1}^{M-1}\left(1 - c_k z^{-1}\right)}{a_0\prod_{k=1}^{N-1}\left(1 - d_k z^{-1}\right)} \tag{8.10}$$

where $a_k, b_k, c_k, d_k \in \mathbb{C}$. If $M < N$ and all poles are simple, i.e., $d_k \neq d_j$ for $k \neq j$, we can do a partial fraction expansion to get

$$X(z) = \sum_{k=0}^{N-1}\frac{A_k}{1 - d_k z^{-1}}, \tag{8.11}$$

where

$$A_k = \left(1 - d_k z^{-1}\right) X(z)\Big|_{z=d_k}. \tag{8.12}$$

If $M \geq N$, we can obtain the right form by first doing a long division of the numerator by the denumerator. This gives a general form

$$X(z) = \sum_{r=0}^{M-N} B_r z^{-r} + \sum_{k=0}^{N-1} \frac{A_k}{1 - d_k z^{-1}}, \tag{8.13}$$

where the B_r's are obtained from the long division and the A_k's are as before.

If $X(Z)$ has a pole of order s at d_i and all other poles are simple we get

$$X(z) = \sum_{r=0}^{M-N} B_r z^{-r} + \sum_{k=0, k\neq i}^{N-1} \frac{A_k}{1 - d_k z^{-1}} + \sum_{m=1}^{s} \frac{C_m}{(1 - d_i z^{-1})^m}, \tag{8.14}$$

where

$$C_m = \frac{1}{(s-m)!(-d_i)^{s-m}} \left\{ \frac{\mathrm{d}^{s-m}}{\mathrm{d}w^{s-m}} \left[(1 - d_i w)^s X\left(w^{-1}\right)\right] \right\}_{w=d_i^{-1}} \tag{8.15}$$

and the A_k's and B_r's are obtained as before.

Using (8.14) and the ROC, we can invert the Z-transform using the inspection method, i.e.

$$a^n u[n] \quad \overset{Z}{\longleftrightarrow} \quad \frac{1}{1 - az^{-1}}, \qquad |z| > |a|, \tag{8.16}$$

$$-a^n u[-n-1] \quad \overset{Z}{\longleftrightarrow} \quad \frac{1}{1 - az^{-1}}, \qquad |z| < |a|. \tag{8.17}$$

Example 8.4 (Z-Transform) **(a)** *Let* $W(z) = \frac{1 - \frac{1}{2}z^{-1}}{1 - z^{-1} + z^{-2}}$ *which is well-defined in its region of convergence* $R_W : |z| > 1$. *Find* $w[n]$, *the inverse z-transform of this function.*

(b) *Find the inverse z-transform of* $X(z) = \ln(1 + z^{-2})$ *for* $|z| > 1$.
Hint: *Differentiate* $X(z)$ *with respect to* z.

(c) *Find the z-transform of* $y[n] = 2^n n u[-n-2]$. *Determine the region of convergence.*
Solution:

(a) In order to find the inverse z-transform of $W(z)$, we have to find its partial fraction expansion. First, we need to find the roots of the polynomial in the denominator.

$$1 - z^{-1} + z^{-2} = 0 \quad or \quad z^2 - z + 1 = 0 \Longrightarrow z = \frac{1 \pm \sqrt{1-4}}{2} = \frac{1 \pm \sqrt{3}i}{2} = e^{\pm \frac{\pi}{3} i}$$

So we have

$$W(z) = \frac{1 - \frac{1}{2}z^{-1}}{(1 - e^{\frac{\pi}{3}i}z^{-1})(1 - e^{-\frac{\pi}{3}i}z^{-1})} = \frac{\alpha}{1 - e^{\frac{\pi}{3}i}z^{-1}} + \frac{\beta}{1 - e^{-\frac{\pi}{3}i}z^{-1}}$$

where

$$\begin{aligned} \alpha &= W(z)(1 - e^{\frac{\pi}{3}i}z^{-1})\Big|_{z=e^{\frac{\pi}{3}i}} = \frac{1-\frac{1}{2}e^{-\frac{\pi}{3}i}}{1-e^{-\frac{2\pi}{3}i}} = \frac{1}{2} \\ \beta &= W(z)(1 - e^{-\frac{\pi}{3}i}z^{-1})\Big|_{z=e^{-\frac{\pi}{3}i}} = \frac{1-\frac{1}{2}e^{\frac{\pi}{3}i}}{1-e^{\frac{2\pi}{3}i}} = \frac{1}{2} \end{aligned}$$

Now for the ROC : $|z| > 1$, we have

$$\begin{aligned} w[n] &= \alpha(e^{\frac{\pi}{3}i})^n u[n] + \beta(e^{-\frac{\pi}{3}i})^n u[n] \\ &= \frac{e^{\frac{\pi n}{3}i} + e^{\frac{-\pi n}{3}i}}{2} u[n] = \cos(\frac{\pi}{3}n)u[n] \end{aligned}$$

Remark: *In general, if we have a transfer function $H(z) = \frac{1-az^{-1}}{1-bz^{-1}+cz^{-2}}$, with $a, b, c \in \mathbb{R}$ and $c \geq 0$ where the discriminant of the denominator is negative,* i.e., *$\Delta = b^2 - 4c < 0$ and $|z| > \sqrt{c}$, we can use the following scheme to find the inverse z-transform. Let $p = re^{i\theta}$ and p^* be the roots of the denominator polynomial. It is easy to show that $r^2 = c$ and $\theta = \cos^{-1}\frac{b}{2\sqrt{c}}$. We have*

$$H(z) = \frac{\alpha}{1 - pz^{-1}} + \frac{\beta}{1 - p^*z^{-1}}$$

where

$$\begin{aligned} \alpha &= \frac{a-p}{p^*-p} = \frac{1}{2} + \frac{(a - r\cos\theta)}{2r\sin\theta}i \\ \beta &= \frac{a-p^*}{p-p^*} = \frac{1}{2} + \frac{-(a - r\cos\theta)}{2r\sin\theta}i \end{aligned}$$

and

$$\begin{aligned} h[n] &= \alpha p^n u[n] + \beta p^{*n} u[n] \\ &= \left[\frac{(re^{i\theta})^n + (re^{-i\theta})^n}{2} + \frac{(r\cos\theta - a)}{r\sin\theta} \frac{(re^{i\theta})^n - (re^{-i\theta})^n}{2i} \right] u[n] \\ &= \left[\cos(\theta n) + \frac{(r\cos\theta - a)}{r\sin\theta} \sin(\theta n) \right] u[n]. \end{aligned}$$

(b) Let $x[n]$ be the inverse z-transform of $X(z)$. Define $y[n] = nx[n]$. According to the properties of the z-transform, we have

$$\begin{aligned} Y(z) &= \mathcal{Z}\{y[n]\} = -z\frac{d}{dz}X(z) = -z\frac{-2z^{-3}}{1+z^{-2}} = \frac{2z^{-2}}{1+z^{-2}} \\ &= 2 - \frac{2}{1+z^{-2}} = 2 - \frac{1}{1+iz^{-1}} - \frac{1}{1-iz^{-1}}. \end{aligned}$$

Therefore for $|z| > 1$,

$$\begin{aligned} y[n] &= \mathcal{Z}^{-1}\{Y(z)\} = 2\delta[n] - i^n u[n] - (-i)^n u[n] \\ &= 2\delta[n] + \begin{cases} -2(-1)^k & \text{if } n = 2k,\ k \in \mathbb{Z}_{\geq 0} \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} -2(-1)^{\frac{n}{2}} u[n-2] & \text{if } n = \text{even} \\ 0 & \text{if } n = \text{odd} \end{cases} \end{aligned}$$

Thus

$$x[n] = \frac{1}{n}y[n] = \begin{cases} \frac{-2(-1)^{\frac{n}{2}}}{n} u[n-2] & \text{if } n = \text{even} \\ 0 & \text{if } n = \text{odd} \end{cases}$$

(c) $y[n] = 2^n n u[-n-2]$, we can define $x[n] = 2^n u[-n-2]$. Then,

$$\begin{aligned} X(z) &= \sum_{n=-\infty}^{\infty} x[n]z^{-n} = \sum_{n=-\infty}^{\infty} 2^n u[-n-2]z^{-n} \\ &= \sum_{n=-\infty}^{-2} 2^n z^{-n} = \sum_{n=-\infty}^{-2} (\frac{z}{2})^{-n} \\ &= \sum_{m=2}^{\infty} (\frac{z}{2})^m = \frac{(\frac{z}{2})^2}{1-\frac{z}{2}} \end{aligned}$$

where the summation is well-defined for $|z| < 2$. Now, we have

$$\begin{aligned} Y(z) &= -z\frac{d}{dz}X(z) = -z\frac{\frac{1}{2}z - \frac{1}{8}z^2}{1 - z + \frac{1}{4}z^2} \\ &= \frac{1}{2}\frac{1 - 4z^{-1}}{z^{-1} - 4z^{-2} + 4z^{-3}} \end{aligned}$$

where the ROC is the set of all $z \in \mathbb{C}$ in which $X(z)$ is well-defined, which is $|z| < 2$.

8.4 Z-Transform Properties

1. **Linearity**:

$$a_1x_1[n] + a_2x_2[n] \overset{Z}{\longleftrightarrow} a_1X_1(z) + a_2X_2(z), \qquad \text{ROC contains } R_{x_1} \cap R_{x_2}$$

2. **Time-shifting**:

$$x[n - n_0] \overset{Z}{\longleftrightarrow} z^{-n_0}X(z), \qquad \text{ROC} = R_x$$

(except for the possible addition or deletion of $z = 0$ or $z = \infty$)

3. **Multiplication by exponential sequence**:

$$z_0^n x[n] \overset{Z}{\longleftrightarrow} X\left(z/z_0\right), \qquad \text{ROC} = |z_0|R_x$$

4. **Differentiation of $X(z)$**:

$$nx[n] \overset{Z}{\longleftrightarrow} -z\frac{\mathrm{d}X(z)}{\mathrm{d}z}, \qquad \text{ROC} = R_x$$

Example: Inverse of a non-rational Z-transform

$$X(z) = \log\left(1 + az^{-1}\right), \qquad |z| > |a|.$$

$$\frac{\mathrm{d}X(z)}{\mathrm{d}z} = \frac{1}{1 + az^{-1}}.\left(-az^{-2}\right).$$

Hence

$$-z\frac{\mathrm{d}X(z)}{\mathrm{d}z} = \frac{az^{-1}}{1 + az^{-1}} \overset{Z^{-1}}{\longleftrightarrow} a(-a)^{n-1}u[n-1] \qquad |z| > |a|.$$

Hence $nx[n] = a(-a)^{n-1}u[n-1]$, or $x[n] = \frac{a}{n}(-a)^{n-1}u[n-1]$. Hence

$$(-1)^{n-1}\frac{a^n}{n}u[n-1] \overset{Z}{\longleftrightarrow} \log\left(1+az^{-1}\right)$$

5. **Conjugation of a complex sequence**:

$$x^*[n] \overset{Z}{\longleftrightarrow} X^*(z^*) \qquad \text{ROC} = R_x$$

6. **Time-reversal**:

$$x[-n] \overset{Z}{\longleftrightarrow} X\left(\frac{1}{z}\right), \qquad \text{ROC} = \frac{1}{R_x}$$

7. **Convolution of sequences**:

$$x_1[n] * x_2[n] \overset{Z}{\longleftrightarrow} X_1(z)X_2(z), \qquad \text{ROC contains } R_{x_1} \cap R_{x_2}$$

8. **Initial value theorem**: If $x[n]$ is zero for $n < 0$, (i.e. if $x[n]$ is causal), then

$$x[0] = \lim_{z\to\infty} X(z)$$

Example 8.5 (Interleaving)

Let $x[n]$ and $y[n]$, $n \in \mathbb{Z}$, be sequences with respective Z-transforms $X(z)$ and $Y(z)$. Now consider a third sequence $u[n]$ that is constructed by interleaving $x[n]$ and $y[n]$. This means that $u[2l] = x[l]$ and $u[2l+1] = y[l]$, $l \in \mathbb{Z}$.

(a) Express the Z-transform of $u[n]$ in terms of $X(z)$ and $Y(z)$.

The ROC of $X(z)$ is $0.64 \le |z| \le 4$ and the ROC of $Y(z)$ is $0.25 \le |z| \le 9$.

(b) What is the ROC of $U(z)$?

Solution:

(a)

$$\begin{aligned} U(z) &= \sum_{n=-\infty}^{\infty} u[n]z^{-n} \\ &= \sum_{l=-\infty}^{\infty} u[2l]z^{-2l} + \sum_{l=-\infty}^{\infty} u[2l+1]z^{-2l-1} \\ &= \sum_{l=-\infty}^{\infty} x[l](z^2)^{-l} + z^{-1}\sum_{l=-\infty}^{\infty} y[l](z^2)^{-l} \\ &= X(z^2) + z^{-1}Y(z^2). \end{aligned}$$

(b) If z_0 is a pole of $X(z)$, then $\pm z_0^{\frac{1}{2}}$ will be poles of $X(z^2)$. This means

$$\begin{aligned} \mathrm{ROC}\{X(z^2)\}: &\quad 0.8 \leq |z| \leq 2, \\ \mathrm{ROC}\{Y(z^2)\}: &\quad 0.5 \leq |z| \leq 3. \end{aligned}$$

Multiplying $Y(z^2)$ by z^{-1} might add a pole at 0, which is outside the region of convergence of $Y(z^2)$. So $\mathrm{ROC}\{z^{-1}Y(z^2)\} = \mathrm{ROC}\{Y(z^2)\}$.

For $U(z)$ to exist, we need both terms in the sum to exist, so we take the intersection of the ROC of the individual terms. This gives

$$\mathrm{ROC}\{U(z)\}: \quad 0.8 < |z| < 2.$$

Example 8.6 (Region of Convergence) *Consider*

$$H(z) = \frac{1 - 4z^{-1} + 5z^{-2}}{(z^{-1} + 2)(1 + 6z^{-1} + 13z^{-2})}$$

as the transfer function of an LTI system.

(a) *What are the poles and zeros of $H(z)$?*

(b) *How many different regions of convergence can be assigned to $H(z)$? Determine them.*

(c) *For each ROC you have found in part (b), take the inverse z-transform and find the impulse response of the system in time domain.*

(d) *For each impulse response check the causality and stability of the system.*

(e) *In each case determine whether the impulse response is finite-length, right-sided, left-sided, or two-sided?*

Solution:

(a) The poles of the transfer function are the roots of $(z^{-1} + 2)(1 + 6z^{-1} + 13z^{-2}) = 0$, which can be computed as

- $z^{-1} + 2 = 0 \quad \Longrightarrow \quad p_1 = -\frac{1}{2}$
- $1 + 6z^{-1} + 13z^{-2} = 0 \quad or \quad z^2 + 6z + 13 = 0 \quad \Longrightarrow \quad p_{2,3} = -3 \pm \sqrt{9 - 13} = -3 \pm 2i$

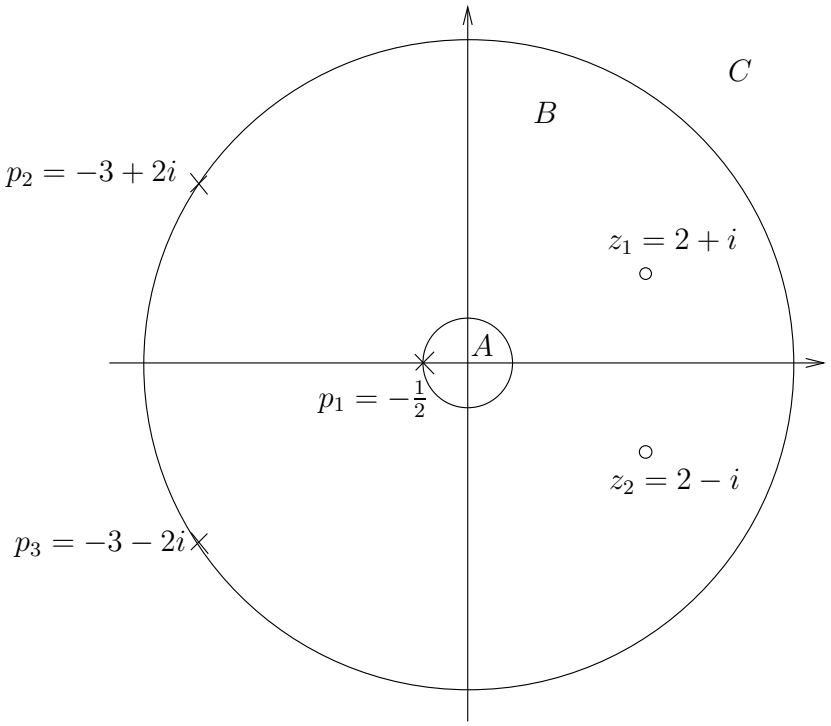


Figure 8.4: The zero-pole plot and three different regions of convergence.

We can also find the zeros of the transfer function by finding the roots of the nominator polynomial.

$$1 - 4z^{-1} + 5z^{-2} = 0 \quad or \quad z^2 - 4z + 5 = 0 \quad \Longrightarrow \quad z_{1,2} = 2 \pm \sqrt{4-5} = 2 \pm i$$

(b) Region of convergence has to be a continuous ring of the point of the $\mathbb{C}$-plane which does not contain any pole, i.e., if $a = re^{i\omega} \in ROC$, then all the points with the same magnitude ($|z| = r$) are also in the ROC. Therefore, as it can be seen in Fig. 8.4, there are three possible regions of convergence for the transfer function.

$$A = \{z : |z| < \frac{1}{2}\}, \quad B = \{z : \frac{1}{2} < |z| < \sqrt{13}\}, \quad C = \{z : |z| > \sqrt{13}\}. \quad (8.18)$$

(c) In order to find the inverse z-transform of $H(z)$ we need to find the partial fraction

expansion for this.

$$H(z) = \frac{1 - 4z^{-1} + 5z^{-2}}{(z^{-1}+2)(1+6z^{-1}+13z^{-2})} = \frac{\alpha}{z^{-1}+2} + \frac{\beta}{1-(-3+2i)z^{-1}} + \frac{\gamma}{1-(-3-2i)z^{-1}}$$

There two ways to find α, β and γ. In the following we will explain both of them.

- *First way: Solving system of linear equations:*

$$\begin{aligned}\frac{1 - 4z^{-1} + 5z^{-2}}{(z^{-1}+2)(1+6z^{-1}+13z^{-2})} &= \frac{\alpha}{z^{-1}+2} + \frac{\beta}{1-(-3+2i)z^{-1}} + \frac{\gamma}{1-(-3-2i)z^{-1}} \\ &= \frac{(\alpha+2\beta+2\gamma) + (6\alpha+(7+4i)\beta+(7-4i)\gamma)z^{-1} + (13\alpha+(3+2i)\beta+(3-2i)\gamma)z^{-2}}{(z^{-1}+2)(1+6z^{-1}+13z^{-2})}\end{aligned}$$

Thus,

$$\begin{cases} \alpha+2\beta+2\gamma = 1 \\ 6\alpha+(7+4i)\beta+(7-4i)\gamma = -4 \\ 13\alpha+(3+2i)\beta+(3-2i)\gamma = 5 \end{cases} \implies \begin{cases} \alpha = 58/82 \\ \beta = (6+95i)/82 \\ \gamma = (6-95i)/82. \end{cases}$$

- *Second way: evaluating the function at its poles*

$$\alpha = H(z)(z^{-1}+2)\Big|_{z=-\frac{1}{2}} = \frac{1-4z^{-1}+5z^{-2}}{1+6z^{-1}+13z^{-2}}\Big|_{z=-\frac{1}{2}} = \frac{29}{41}$$

and

$$\beta = H(z)(1-(-3+2i)z^{-1})\Big|_{z=-3+2i} = \frac{1-4z^{-1}+5z^{-2}}{(z^{-1}+2)(1-(-3-2i)z^{-1})}\Big|_{z=-3+2i} = \frac{6+95i}{82}$$

$$\gamma = H(z)(1-(-3-2i)z^{-1})\Big|_{z=-3-2i} = \frac{1-4z^{-1}+5z^{-2}}{(z^{-1}+2)(1-(-3+2i)z^{-1})}\Big|_{z=-3-2i} = \frac{6-95i}{82}$$

Therefore we have

$$H(z) = \frac{29}{82}\frac{1}{1+\frac{1}{2}z^{-1}} + \frac{6+95i}{82}\frac{1}{1-(-3+2i)z^{-1}} + \frac{6-95i}{82}\frac{1}{1-(-3-2i)z^{-1}}$$

Now we are ready to compute the inverse z-transform for each ROC. In the following computation we use $\phi = \tan^{-1}\frac{-2}{3}$

- *A:*

$$\begin{aligned}
h_A[n] &= -\frac{29}{82}(\frac{1}{2})^n u[-n-1] - \frac{6+95i}{82}(-3+2i)^n u[-n-1] - \frac{6-95i}{82}(-3-2i)^n u[-n-1] \\
&= -\left[\frac{29}{82}\frac{1}{2^n} + \frac{6+95i}{82}(\sqrt{13}e^{i\phi})^n + \frac{6-95i}{82}(\sqrt{13}e^{-i\phi})^n\right] u[-n-1] \\
&= -\left[\frac{29}{82}\frac{1}{2^n} + \frac{6}{41}\sqrt{13}^n \frac{e^{i\phi n}+e^{-i\phi n}}{2} - \frac{95}{41}\sqrt{13}^n \frac{e^{i\phi n}-e^{-i\phi n}}{2i}\right] u[-n-1] \\
&= -\left[\frac{29}{82}\frac{1}{2^n} + \frac{6}{41}\sqrt{13}^n \cos(\phi n) - \frac{95}{41}\sqrt{13}^n \sin(\phi n)\right] u[-n-1]
\end{aligned}$$

- *B:*

$$\begin{aligned}
h_B[n] &= \frac{29}{82}(\frac{1}{2})^n u[n] - \frac{6+95i}{82}(-3+2i)^n u[-n-1] - \frac{6-95i}{82}(-3-2i)^n u[-n-1] \\
&= \frac{29}{82}\frac{1}{2^n}u[n] - \left[\frac{6+95i}{82}(\sqrt{13}e^{i\phi})^n + \frac{6-95i}{82}(\sqrt{13}e^{-i\phi})^n\right] u[-n-1] \\
&= \frac{29}{82}\frac{1}{2^n}u[n] - \left[\frac{6}{41}\sqrt{13}^{-n} \frac{e^{i\phi n}+e^{-i\phi n}}{2} - \frac{95}{41}\sqrt{13}^{-n} \frac{e^{i\phi n}-e^{-i\phi n}}{2i}\right] u[-n-1] \\
&= \frac{29}{82}\frac{1}{2^n}u[n] - \left[\frac{6}{41}\sqrt{13}^n \cos(\phi n) - \frac{95}{41}\sqrt{13}^n \sin(\phi n)\right] u[-n-1]
\end{aligned}$$

- *C:*

$$\begin{aligned}
h_C[n] &= \frac{29}{82}(\frac{1}{2})^n u[n] + \frac{6+95i}{82}(-3+2i)^n u[n] + \frac{6-95i}{82}(-3-2i)^n u[n] \\
&= \left[\frac{29}{82}\frac{1}{2^n} + \frac{6+95i}{82}(\sqrt{13}e^{i\phi})^n + \frac{6-95i}{82}(\sqrt{13}e^{-i\phi})^n\right] u[n] \\
&= \left[\frac{29}{82}\frac{1}{2^n} + \frac{6}{41}\sqrt{13}^n \frac{e^{i\phi n}+e^{-i\phi n}}{2} - \frac{95}{41}\sqrt{13}^n \frac{e^{i\phi n}-e^{-i\phi n}}{2i}\right] u[n] \\
&= \left[\frac{29}{82}\frac{1}{2^n} + \frac{6}{41}\sqrt{13}^n \cos(\phi n) - \frac{95}{41}\sqrt{13}^n \sin(\phi n)\right] u[n]
\end{aligned}$$

(d) Causality: Since $h_A[n]$ and $h_B[n]$ have some terms in form of $u[-n-1]$, clearly they are not causal. All the terms in $h_C[n]$ are in the form $u[n]$ and it just depend on the future. So, $h_C[n]$ is a causal system.

Stability: We know that a sequence is stable if and only if the ROC of its z-transform contains the unit circle. According to the regions of convergences found in part (b),

the only stable system is $h_B[n]$. We can also check it in time domain:

$$\begin{aligned}
\sum_{n=-\infty}^{\infty} |h_B[n]| &= \sum_{n=-\infty}^{-1} \left| \frac{6}{41}\sqrt{13}^n \cos(\phi n) - \frac{95}{41}\sqrt{13}^n \sin(\phi n) \right| + \sum_{n=0}^{\infty} \left| \frac{29}{82}\frac{1}{2^n} \right| \\
&\leq \sum_{n=-\infty}^{-1} \left| \frac{6}{41}\sqrt{13}^n \cos(\phi n) \right| + \sum_{n=-\infty}^{-1} \left| \frac{95}{41}\sqrt{13}^n \sin(\phi n) \right| + \sum_{n=0}^{\infty} \left| \frac{29}{82}\frac{1}{2^n} \right| \\
&\leq \frac{6}{41} \sum_{n=-\infty}^{-1} \left| \sqrt{13}^n \right| + \frac{95}{41} \sum_{n=-\infty}^{-1} \left| \sqrt{13}^n \right| + \frac{29}{82} \sum_{n=0}^{\infty} \left| \frac{1}{2^n} \right| \\
&= \frac{101}{41}\frac{1}{\sqrt{13}-1} + \frac{29}{41} < \infty
\end{aligned}$$

We can also check that $\sum_{n=-\infty}^{\infty} |h_A[n]| = \infty$ and $\sum_{n=-\infty}^{\infty} |h_C[n]| = \infty$, but it is a bit long and more complicated.

(e) We can determine this property either by looking at the impulse responce or by considering the ROC.

- *Impulse response: All the terms in h_A are in form $u[-n-1]$ and so it is left-sided. $h_B[n]$ has both the terms of form $u[-n-1]$ and $u[n]$ and it is two-sided sequence. Finally, only terms of the form $u[n]$ contribute in $h_C[n]$ and so it is right-sided sequence.*
- *ROC: Region A is inside a circle, so the corresponding sequence should be left-sided. B is a ring and therefore should correspond to a two-sided sequence. Region C is outside of a circle and the inverse transform corresponds to this region would be a right-sided sequence.*

8.5 Analysis and Characterization of LTI Systems Using Z-Transform

The Z-transform plays a particularly important role in the analysis and representation of discrete-time LTI systems. From the convolution property, for an LTI system with impulse response $h[n] \overset{Z}{\longleftrightarrow} H(z)$ and an input $x[n] \overset{Z}{\longleftrightarrow} X(z)$, we have,

$$y[n] = (h * x)[n] = h[n] * x[n] \overset{Z}{\longleftrightarrow} H(z)X(z), \tag{8.19}$$

where X(z), Y(z) and H(z) are the Z-transforms and the ROC of $Y(z)$ is

$$R_y = R_x \cap R_h. \tag{8.20}$$

$H(z)$ is referred to as the system function or transfer function of the system.

8.5.1 Causality

A causal LTI system has an impulse response that is $h[n] = 0$ for $n < 0$ and therefore is right-sided. Hence the ROC of $H(z)$ is the exterior of a circle on the Z-plane.

Since for a causal LTI system,

$$H(z) = \sum_{n=0}^{\infty} h[n]z^{-n} \tag{8.21}$$

does *not* include any term with positive power of z, the ROC includes infinity. Hence a discrete LTI system is causal if and only if the ROC of its system function is exterior of a circle and includes infinity.

8.5.2 Stability

We know that an LTI system is BIBO stable if and only if its impulse response is absolutely summable. This implies that its DTFT exists, which in turn means that the unit circle is in the ROC of the system function. Therefore, we have the following:

An LTI system is stable if and only if the ROC of its system function $H(z)$ includes the unit circle.

Combining causality and stability, we see that a causal LTI system with rational system function $H(z)$ is stable if and only if all its poles of $H(z)$ lie inside the unit circle.

8.5.3 LTI Systems and Linear Constant-Coefficient Difference Equations

If

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

then

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

Hence

$$Y(z)\left[1 - \sum_{k=1}^{N-1} a_k z^{-k}\right] = X(z) \sum_{k=0}^{M-1} b_k z^{-k}$$

or

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

Therefore the system function for such systems are easy to write. This form of $H(z) = \frac{Y(z)}{X(z)}$ is also called the transfer function of a linear constant-coefficient difference equation system.

8.6 Problems

Problem 8.1 *Consider the system shown in Fig. 8.5,*

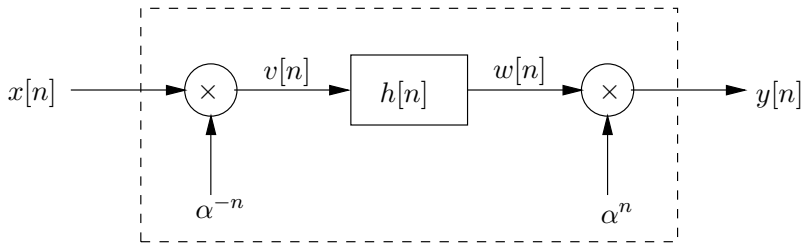


Figure 8.5: System with multiplier.

where $h[n]$ is the impulse response of the LTI system and $H(z)$ exists for

$$0 < r_{\min} < |z| < r_{\max} < \infty.$$

(a) Can the LTI system with impulse response $h[n]$ be BIBO stable? If so, determine constraints on $r_{\min}$ and $r_{\max}$, otherwise explain why.

(b) Is the system with input $x[n]$ and output $v[n]$ linear? Is it time-invariant?

(c) Is the overall system (with input $x[n]$ and $y[n]$) LTI? If so, find the impulse response of the system, otherwise give an example to show your claim.

(d) Can the overall system be BIBO stable? If so determine the constraints on α, $r_{\min}$, and $r_{\max}$, otherwise explain why.

Problem 8.2 (DFT and Z-transform) *Let us consider a sequence $x(n)$ having z-transform $X(z)$. If the sequence has finite duration of length N or less, it can be recovered from its N-point DFT. Hence its z-transform is uniquely determined by its N-point DFT. Show that*

$$X(z) = \sum_{n=0}^{N-1} x(n) z^{-n} = \frac{1 - z^{-N}}{N} \sum_{k=0}^{N-1} \frac{X(k)}{1 - e^{j2\pi k/N} z^{-1}}.$$

Problem 8.3 (Minimum Phase System) *A system is called* minimum phase *if the system and its inverse are causal and stable. A rational transfer function will be minimum phase if and only if all its zeros and poles are inside the unit circle. For example* $H_1(z) = \frac{1-az^{-1}}{1-bz^{-1}}$, *where* $a = \frac{1}{3}e^{\frac{3\pi}{4}i}$ *and* $b = \frac{1}{2}e^{\frac{\pi}{3}i}$, *is a minimum phase system.*

However, there exist other transfer functions such as

$$H_2(z) = \frac{z^{-1} - a^*}{1 - bz^{-1}}, \quad H_2(z) = \frac{1 - az^{-1}}{z^{-1} - b^*}, \quad \text{and } H_3(z) = \frac{z^{-1} - a^*}{z^{-1} - b^*}$$

which have the same magnitude as $H_1(z)$ *on the unit circle. In this problem we investigate two important properties of minimum phase systems which identify the minimum phase system among all systems with the same magnitude.*

(a) [MINIMUM GROUP-DELAY] *Let* $H(z) = \frac{(1-cz^{-1})}{(1-dz^{-1})}$ *where* $c = |c|e^{i\theta}$, $d = |d|e^{i\phi}$, $|c| > 1$ *and* $|d| < 1$. *Rewrite* $H(z) = H_{\min}(z)H_{ap}(z)$ *where*

- *All zeros and poles of* $H_{\min}(z)$ *are inside the unit circle,*
- $H_{ap}(z)$ *be a causal all-pass filter, i.e.,* $H(e^{j\omega}) = 1, \forall\omega$.

(i) Replace z *in* $H_{ap}(z)$ *by* $e^{i\omega}$ *and find the group-delay expression for* $H_{ap}(z)$ *in terms of* $|c|, |d|, \theta, \phi, \omega$ *and show that it is positive for any* ω.

(ii) Write the group-delay expression for $H(z)$ *in terms of the group-delay of* $H_{\min}(z)$ *and* $H_{ap}(z)$. *Compare the group-delay of* $H(z)$ *and* $H_{\min}(z)$.

(b) [MINIMUM ENERGY DELAY] *Let* $H_{\min}(z)$ *be a minimum phase system which has a zero at* α. *We can write* $H_{\min}(z) = Q(z)(1 - \alpha z^{-1})$, *where* $Q(z)$ *is also minimum phase. Now consider another system with transfer function* $H(z)$ *such that* $|H(z)| = |H_{\min}(z)|$ *and* $H(z)$ *has a zero at* $1/\alpha^*$ *instead of* α.

(i) Compare $\sum_{n=0}^{\infty} |h_{\min}[n]|^2$ *to* $\sum_{n=0}^{\infty} |h[n]|^2$.

(ii) Express $H(z)$ *in terms of* $Q(z)$.

(iii) Express $h_{\min}[n]$ *and* $h[n]$ *in terms of* $q[n]$ *and* α.

(iv) Write an expression for

$$\sum_{n=0}^{m} |h_{\min}[n]|^2 - \sum_{n=0}^{m} |h[n]|^2$$

and simplify it to find an expression in terms of $q[m]$ *and* α.

(v) *Compare* $\sum_{n=0}^{m} |h_{\min}[n]|^2$ *to* $\sum_{n=0}^{m} |h[n]|^2$ *and conclude that the minimum phase system has the minimum energy-delay among all the systems with the same magnitude response.*

Problem 8.4 *Let* $x[n]$ *be a discrete-time sequence and* $X(z)$ *its corresponding z-transform with appropriate ROC.*

(a) *Prove that the following relation holds:*

$$nx[n] \overset{Z}{\longleftrightarrow} -z\frac{d}{dz}X(z).$$

(b) *Using (a), show that*

$$(n+1)\alpha^n u[n] \overset{Z}{\longleftrightarrow} \frac{1}{(1-\alpha z^{-1})^2}, \qquad |z| > |\alpha|.$$

(c) *Suppose that the above expression corresponds to the impulse response of an LTI system. What can you say about the causality of such a system? About its stability?*

(d) *Let* $\alpha = 0.8$*, what is the spectral behavior of the corresponding filter? What if* $\alpha = -0.8$*?*

Problem 8.5 *Consider a causal discrete system represented by the following difference equation:*

$$y[n] - 3.25y[n-1] + 0.75y[n-2] = x[n-1] + 3x[n-2].$$

(a) *Compute the transfer function and check the stability of this system both analytically and graphically.*

(b) *If the input signal is* $x[n] = \delta[n] - 3\delta[n-1]$*, compute the z-transform of the output signal and discuss the stability.*

(c) *Take an arbitrary input signal that does not cancel the unstable pole of the transfer function and repeat b).*

Problem 8.6 *Consider two two-sided sequences* $h[n]$ *and* $g[n]$ *and consider a third sequence* $x[n]$ *which is built by interleaving the values of* $h[n]$ *and* $g[n]$*:*

$$x[n] = \ldots, h[-3], g[-3], h[-2], g[-2], h[-1], g[-1], h[0], g[0], h[1], g[1], h[2], g[2], h[3], g[3], \ldots$$

with $x[0] = h[0]$*.*

(a) *Express the z-transform of* $x[n]$ *in terms of the z-transforms of* $h[n]$ *and* $g[n]$*.*

(b) *Assume that the ROC of* $H(z)$ *is* $0.64 < |z| < 4$ *and that the ROC of* $G(z)$ *is* $0.25 < |z| < 9$*. What is the ROC of* $X(z)$*?*

Chapter 9

Filters and Filter Design

We have already shown that, from a mathematical point of view, a linear time-invariant system is completely characterized by its impulse response. While any absolutely summable impulse response defines a stable LTI system, the computation of an output sample for the system might require an infinite number of operations; this is for instance the case of the ideal filters in section 7.7.1. In practice, of course, we are interest in *realizable* systems, i.e. systems which can be implemented with a finite number of operations. It is immediate to see that any FIR filter belongs to this category, but we have also seen in 7.4.2 that there exist IIR systems (whose impulse response is an infinite sequence) which can still be implemented with a finite amount of computation and storage. It turns out that the most general class of such realizable discrete-time systems is described by *constant-coefficient difference equations*. The general concept of filter design usually starts with a given set of specifications, which in all but a handful of cases are expressed in terms of a desired frequency response; the design problem is solved by finding the appropriate coefficients for a suitable difference equation which implements the filter. We will show that realizable filters possess a transfer function which is a ratio of polynomials in the complex variable z^{-1}; as a consequence, filter design can be cast in terms of a polynomial optimization procedure for a given error measure. Finally, the structure of difference equation defines an explicit operational procedure for computing the filter's output values; by arranging the terms of the equation in different ways, we can arrive at different algorithmic structures for the implementation of digital filters.

9.1 Realizable Filters: General Properties

We will start our discussion with a general analysis of constant-coefficient difference equations and associated transfer functions, from which the essential properties of linear filters

(including stability) are easily derived.

9.1.1 Difference Equations & Initial Conditions

In its most general form, a constant-coefficient difference equation defines a relationship between an input signal $x[n]$ and an output signal $y[n]$ as

$$\sum_{k=0}^{N-1} a_k y[n-k] = \sum_{k=0}^{M-1} b_k x[n-k]; \tag{9.1}$$

in the rest of these notes we will restrict ourselves to the case in which all the coefficients a_k and b_k are real. Usually, it is $a_0 = 1$, so that the above equation can easily be rearranged as:

$$y[n] = \sum_{k=0}^{M-1} b_k x[n-k] - \sum_{k=1}^{N-1} a_k y[n-k]; \tag{9.2}$$

Clearly, the above relation defines each output sample $y[n]$ as a linear combination of past and present input values and past output values. However, it is easy to see that if $a_{N-1} \neq 0$ we can for instance rearrange (9.1) as

$$y[n-N+1] = \sum_{k=0}^{M-1} b'_k x[n-k] - \sum_{k=0}^{N-2} a'_k y[n-k];$$

where $a'_k = a_k/a_{N-1}$ and $b'_k = b_k/a_{N-1}$. With the change of variable $m = n - N + 1$, this becomes

$$y[m] = \sum_{k=N-M}^{N-1} b'_k x[m+k] - \sum_{k=1}^{N-1} a'_k y[m+k]; \tag{9.3}$$

which shows that the difference equation can be computed in the other way as well, namely by expressing $y[m]$ as a linear combination of *future* values of input and output. It is rather intuitive that the first approach defines a causal behavior, while the second approach is anticausal. The main point is that, contrary to the differential equations used in the characterization of continuous-time systems, difference equation can be used directly to translate the transformation operated by the system into an *explicit algorithmic form*. To see this, and to gain a lot of insight on the properties of difference equations, it may be useful to consider a possible implementation of the system in (9.2), here written in C:

```
extern double a[N];     // The a's coefficients
extern double b[M];     // The b's coefficients

static double x[M];     // Delay line for x
static double y[N];     // Delay line for y

double GetOutput(double input)
{
  int k;
  for (k = N-1; k > 0; k--)     // Shift delay line for x
    x[k] = x[k-1];
  x[0] = input;                 // new input value x[n]

  for (k = M-1; k > 0; k--)     // Shift delay line for x
    y[k] = y[k-1];

  double y = 0;
  for (k = 0; k < M; k++)
    y += b[k] * x[k];
  for (k = 1; k < M; k++)
    y -= a[k] * y[k];           // New value for y[n];
  y[0] = y;                     // Store in delay line

  return y;
}
```

It is immediate to verify that

1. the above routine realizes the difference equation in (9.2)
2. the storage required is $(N + M)$
3. each output sample is obtained via $(N + M - 1)$ multiplications and additions
4. the transformation is causal

If we try to compile and run the above routine, however, we immediately run into an *initialization* problem: the first time (actually, the first $\max(N, M-1)$ times) we call the function, the delay lines which hold past values of $x[n]$ and $y[n]$ will contain undefined values. Most likely, the compiler will notice this condition and will print a warning message signaling that the static arrays have not been properly initialized. We are back to the problem of setting the initial conditions of the system. *The choice which guarantees linearity and time invariance is called the* zero initial conditions *and corresponds to setting the delay lines to zero before starting the algorithm.* This choice implies that the system

response to the zero sequence is the zero sequence and, in this way, linearity and time invariance can be proven as in section 7.4.2.

9.1.2 Transfer Functions

The best way to analyze the properties of the system implemented by (9.1) is to apply the z-transform to both sides of the equation; we obtain:

$$Y(z) = X(z) \sum_{n=0}^{M-1} b_n z^{-n} - Y(z) \sum_{n=1}^{N-1} a_n z^{-n} \tag{9.4}$$

The above equation can be rearranged as

$$Y(z) = H(z)X(z) \tag{9.5}$$

where $H(z)$ is the *transfer function* of the system and is given by

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_{M-1} z^{-(M-1)}}{1 + a_1 z^{-1} + \ldots + a_{N-1} z^{-(N-1)}} \tag{9.6}$$

Such a transfer function is called a *rational transfer function* and is the ratio of two polynomials in z^{-1}; note that the degree of the polynomial at the numerator is $M-1$ and that of the denumerator is $N-1$. As such, it can be written in factored form as

$$H(z) = b_0 \frac{\prod_{n=1}^{M-1} (1 - z_n z^{-1})}{\prod_{n=1}^{N-1} (1 - p_n z^{-1})} \tag{9.7}$$

where the z_n are called the *zeros* of the filter and p_n are called the *poles*; the zeros are the roots of the numerator of the transfer function while the poles are the roots of the denominator. Clearly, if $z_i = p_k$ for some i and k (i.e. if a pole and a zero coincide) the corresponding first-order factors will cancel each other out and the degrees of numerator and denominator are both decreased by one. In the following, we will assume that such factors have been already removed and that the numerator and denominator polynomials of a given rational transfer function are coprime.

Recall that the roots of a polynomial with real-valued coefficients are either real or they occur in complex-conjugate pairs; a pair of complex-conjugate roots translates to a second-order term with real coefficients:

$$(1 - az^{-1})(1 - a^* z^{-1}) = 1 - 2\mathrm{Re}\{a\} z^{-1} + |a|^2 z^{-2} \tag{9.8}$$

As a consequence, the transfer function can be factored in the product of first- and second-order terms in which the coefficients are all strictly real; namely:

$$H(z) = b_0 \frac{\prod_{n=1}^{M_r}(1 - z_n z^{-1}) \prod_{n=M_r+1}^{M_c}(1 - 2\mathrm{Re}\{z_n\}z^{-1} + |z_n|^2 z^{-2})}{\prod_{n=1}^{N_r}(1 - p_n z^{-1}) \prod_{n=N_r+1}^{N_c}(1 - 2\mathrm{Re}\{p_n\}z^{-1} + |p_n|^2 z^{-2})} \tag{9.9}$$

where M_r is the number of real zeros, M_c is the number of complex-conjugate zeros and $M_r + 2M_c = M$ (with the same holding for the poles representation, i.e. $N_r + 2N_c = N$).

9.1.3 Stability Analysis

If we consider equation (9.5), $Y(z) = H(z)X(z)$, it is clear that $H(z)$ is the z-transform of the filter's impulse response. Therefore, the BIBO stability of a digital filter as described by (9.1) is easily inferred from the properties of the z-transform: *for a filter to be stable the ROC of $H(z)$ must contain the unit circle* because this guarantees the absolute summability of $h[n]$. Since the ROC is determined by the location of the poles of the transform, the above condition translates to the following:

- **For causal filters** the ROC of $H(z)$ is a region on the complex plane extending *outwards*, and therefore a necessary and sufficient condition for stability is that *all the poles of $H(z)$ are inside the unit circle.*
- **For anticausal filters** the ROC of $H(z)$ is a region on the complex plane extending *inwards*, and therefore a necessary and sufficient condition for stability is that *all the poles of $H(z)$ are outside the unit circle.*

9.2 Filter Design - Introduction

As we have seen, a realizable filter is completely described by its rational transfer function; designing a filter corresponds to determining the coefficients of the transfer function with respect to the desired filter characteristics. For an FIR filter of length M, there are M coefficients that have to be determined, and they correspond directly to the filter's impulse response. In a similar way, an IIR filter with a numerator of degree $M - 1$ and a denominator of degree $N - 1$ has $M + N - 1$ coefficients to determine (since we always assume $a_0 = 1$). The main questions are the following:

- How do we choose the filter's coefficients in order to obtain the desired filtering characteristics?

- What are the criteria to measure the quality of the obtained filter?
- What is the best algorithmic structure (software or hardware) to implement a given digital filter?

The first two questions are optimization problems in a parameter space of dimension $M + N - 1$ with a given optimality criterion (for instance, minimum square error or minimax). The last question is an algorithmic design problem subject to constraints of computational speed, storage and precision, which we will analyze in detail at the end of the chapter.

9.2.1 FIR versus IIR

Filter design has a long and noble history in the analog domain: a linear electronic network can be described in terms of a differential equation linking, for instance, the voltage as a function of time at the input of the network to the voltage at the output. The arrangement of the capacitors, inductances and resistors in the network determine the form of the differential equation, while their values determine its coefficients. A fundamental difference between an analog filter and a digital filter is that the transformation from input to output is almost always considered *instantaneous* (i.e., the propagation effects along the circuit are neglected). In digital filters, on the other hand, the delay is always explicit and is actually the fundamental building block in a processing system. Because of the physical properties of capacitors, which are ubiquitous in analog filters, the transfer function of an analog filter (expressed in terms of its Laplace transform) is "similar" to the transfer function of an IIR filter, in the sense that it contains both poles and zeros. In a sense, IIR filters can be considered the discrete-time counterpart of classic analog filters. FIR filters, on the other hand, are the flagship of digital signal processing; while one could conceive of an analog equivalent to an FIR, its realization would require the use of analog delay lines, which are costly and impractical components to manufacture. In a digital signal processing scenario, on the other hand, the designer can freely choose between two lines of attack with respect to a filtering problem, IIR or FIR, and therefore it is important to highlight advantages and disadvantages of each.

FIR Filters. The main advantages of FIR filters can be summarized as follows:

- ✓ Unconditional stability;
- ✓ Precise control of the phase response and, in particular, exact linear phase;
- ✓ Optimal algorithmic design procedures;
- ✓ Robustness with respect to finite numerical precision hardware

while their disadvantages are mainly:

× Longer input-output delay

× Higher computational cost with respect to IIR solutions

IIR Filters. IIR filters are often an afterthought in the context of digital signal processing in the sense that they are designed by mimicking established design procedures in the analog domain; their appeal lies mostly in their compact formulation: for a given computational cost, i.e., for a given number of operations per input sample, they can offer a much better magnitude response than an equivalent FIR filter. Furthermore, there are a few fundamental processing tasks (such as DC removal, as we will see later) which are the natural domain of IIR filters. The drawbacks of IIR filter, however, mirror in the negative the advantages of FIR filters. The main advantages of IIR filters can be summarized as follows:

✓ Lower computational cost with respect to an FIR filter with similar behavior

✓ Shorter input-output delay

✓ Compact representation

while their disadvantages are mainly:

× Stability is not guaranteed;

× Phase response is difficult to control;

× Design is complex in the general case;

× Sensitive to numerical precision.

For these reasons, in these notes and in the course we will concentrate mostly on the FIR design problem and we will tackle the design of IIR filters mostly for some specific processing tasks that are often encountered in practice.

9.2.2 Filter Specifications & Tradeoffs

A set of filter specifications represents the guidelines for application-oriented filter design. Real-world filters are designed with a variety of practical requirements in mind, most of which are conflicting. One such requirement, for instance, is to obtain a low "computational price" for the filtering operation; this cost is obviously proportional to the number of coefficients in the filters, as we have seen in the introduction, but it also depends heavily on

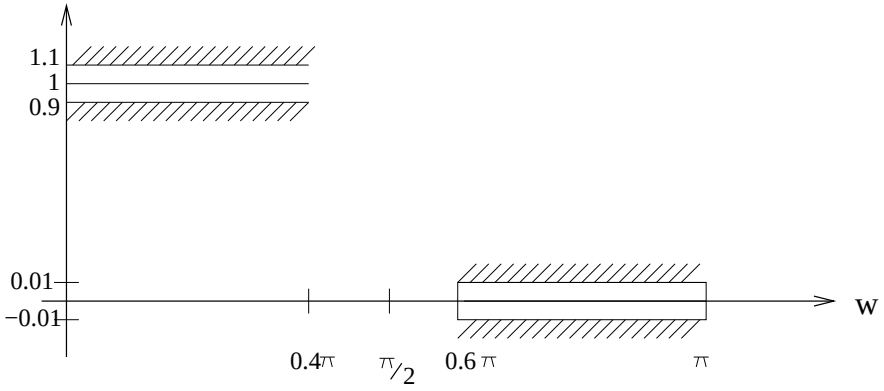


Figure 9.1: Filter specifications

the underlying hardware architecture. The tradeoffs between disparate requirements such as cost, precision or numerical stability are very subtle and not altogether obvious; the art of the digital filter designer, although probably less dazzling than the art of the *analog* filter designer, is to determine the best design strategy for a given practical problem.

In this chapter we certainly won't consider all the variables that enter a filter design scenario; the starting point, however, is almost always a set of *filter specifications* in the frequency domain. These are best illustrated by example: suppose our goal is to design a half-band lowpass filter, i.e., a lowpass filter with cutoff frequency $\pi/2$. The practical contraints to consider are the following:

- **Filter Type.** Whether we design an FIR or an IIR filter depends on a variety of factors specific to the practical application. This decision, however, is the first to be made since the filter specifications follow from it. A closely related design choice determines the maximum filter order which we can afford.

- **Transition band.** We should know by now (and we shall see again shortly) that we cannot obtain an arbitrarily sharp transition band in a realizable filter. Therefore, we must be willing to allow for the existence of a *transition band* from passband to stopband; suppose we estimate that its width can be up to 20% of the total bandwidth: since the cutoff is supposed to be at 0.5π, the transition band will thus extend from 0.4π to 0.6π.

- **Tolerances.** Similarly, we cannot simply impose a strict value of 1 for the passband and a value of 0 for the stopband, but we must allow for *tolerances*; suppose after examining the problem for which we are designing the filter we decide we can afford a 10% error in the passband and a 1% error in the stopband. Note that, in filter

design parlance, the attenuation in the stopband is frequently expressed on a decibel logarithmic scale:

$$A_{\text{dB}} = 20\log_{10}(\delta_s),$$

where δ_s is the maximum tolerated error in the stopband. In the previous example, we are thus requiring an attenuation of 40 dB.

These specifications can be represented graphically as in Figure 9.1; the filter design problem consists now in finding the minimum size FIR or IIR filter that fulfills the required specifications.

9.3 FIR Filter Design

In this section we will explore two fundamental strategies for FIR filter design: the window method and the minimax (or Parks-McClellan) method. Both methods seek to minimize the error between a desired (and often ideal) filter transfer function and the transfer function of the designed filter; they differ in the error measure used in the minimization. The window method is completely straightforward and is often used for quick designs. The minimax method, on the other hand, is the procedure of choice for accurate, optimal filters. Both methods will be illustrated with respect to the design of a lowpass filter.

9.3.1 FIR Filter Design by Windowing

Consider the problem of designing a lowpass filter with cutoff frequency w_c: with no further specifications (i.e., with no declared tolerance or approximations) the only solution is simply the inverse Fourier transform of the desired transfer function. The resulting impulse response $h[n]$ is of course the usual sinc function which we saw in section 7.7.1:

$$\begin{aligned} h[n] &= \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\omega})e^{j\omega n}d\omega \\ &= \frac{1}{2\pi}\int_{-w_c}^{w_c} e^{j\omega n}d\omega \\ &= \frac{1}{2\pi jn}\left[e^{j\omega_c n} - e^{-j\omega_c n}\right] \\ &= \frac{\sin(\omega_c n)}{\pi n} \\ &= \frac{\omega_c}{\pi}\,\text{sinc}(\frac{\omega_c}{\pi}n) \end{aligned}$$

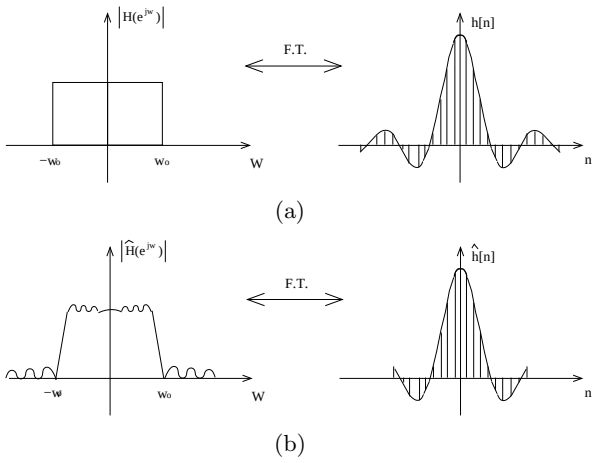


Figure 9.2: a) Ideal filter. b) Approximated filter.

The resulting filter, however, is an ideal filter and it cannot be represented by a rational transfer function with a finite number of coefficients. Our only hope is that by somehow relaxing the design constraints we can arrive at a realizable approximation $\hat{H}(e^{j\omega})$ of the ideal filter $H(e^{j\omega})$. Let us start by considering an approximated FIR filter obtained by simply truncating the original impulse response:

$$\hat{h}[n] = \begin{cases} h[n] & -N \leq n \leq N \\ 0 & \text{otherwise} \end{cases} \tag{9.10}$$

This is a $(2N+1)$-tap FIR filter; Figure 9.2(a) shows the ideal filter and Figure 9.2(b) shows the approximated filter, with their corresponding Fourier transforms (the reasons for their shapes will be clear later). The approximation we just created was obtained in a sort of "intuitive" way; we will now show, however, that it actually satisfies a very precise approximation criterion, namely the minimization of the mean square error (MSE) between the original and approximated filters. Let us denote this error by E_2, that is:

$$E_2 = \int_{-\pi}^{\pi} |H(e^{j\omega}) - \hat{H}(e^{j\omega})|^2 dw.$$

Therefore one optimization problem for an FIR filter of length M (where $M = 2N+1$) could be

$$\begin{aligned} \text{minimize} \quad & \left\| H\left(e^{j\omega}\right) - \hat{H}\left(e^{j\omega}\right) \right\|_2^2 \\ \text{s.t.} \quad & \\ & \hat{H}\left(e^{j\omega}\right) = \textstyle\sum_{n=-N}^{N} \hat{h}[n] e^{-j\omega n}. \end{aligned} \tag{9.11}$$

We can apply Parseval's theorem (see (5.44)) to obtain the equivalent expression in the discrete-time domain:

$$E_2 = 2\pi \sum_{n\in\mathcal{Z}} |h[n] - \hat{h}[n]|^2$$

If we now recall that $\hat{h}[n] = 0$ for $|n| > N$, we have

$$E_2 = 2\pi \left[\sum_{n=-N}^{N} |h[n] - \hat{h}[n]|^2 + \sum_{n=N+1}^{\infty} |h[n]|^2 + \sum_{n=-\infty}^{-N-1} |h[n]|^2 \right].$$

Obviously the last two terms are nonnegative and independent of $\hat{h}[n]$. Therefore, the minimization of E_2 with respect to $\hat{h}[n]$ is equivalent to the minimization of the first term only, and this is easily obtained by letting

$$\hat{h}[n] = h[n] \quad \text{for } n = -N, \ldots, N$$

If we look at what we are doing, it is apparent that we are trying to approximate a Fourier sum with a finite number of terms (we are actually working backwards, since the Fourier coefficients are the discrete-time filter values and the approximated function is a frequency response). Since the approximated function is discontinuous in ω_0 we will incur the Gibbs phenomenon, which explains the non-negligible ripples around the transition point. Also, because of the finite number of terms, the transition from passband to stopband will be less sharp; this is clearly apparent in Figure 9.2.

The Window Concept. Another way to look at the resulting approximation is to express $\hat{h}[n]$ as:

$$h[n] = h[n]w[n], \tag{9.12}$$

where $w[n]$ is a rectangular window of length $(2N + 1)$ taps centered at index zero:

$$w[n] = \text{rect}(n/N) = \begin{cases} 1 & -N \le n \le N \\ 0 & \text{otherwise} \end{cases}. \tag{9.13}$$

We know from the modulation theorem in (7.24) that the Fourier transform of (9.12) is

$$\hat{H}(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) W(e^{j(\omega-\theta)}) d\theta,$$

where $W(e^{j\omega})$ is the Fourier transform of the window $w[n]$:

$$W(e^{j\omega}) = \sum_{n=-N}^{N} e^{-j\omega n} = \frac{\sin\left(\omega\left(N + \frac{1}{2}\right)\right)}{\sin(\omega/2)} \tag{9.14}$$

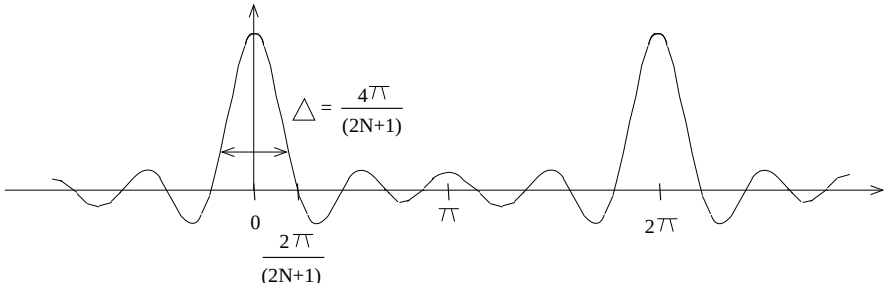


Figure 9.3: Approximated filter Fourier transform

An example of $W(e^{j\omega})$ for $N = 6$ is shown in Figure 9.3. By visual inspection, we can determine the following facts:

- The first zero crossing of $W(e^{j\omega})$ occurs at $\omega = 2\pi/(2N + 1)$
- The width of the main lobe of the magnitude response is $\Delta = 4\pi(2N + 1)$
- The magnitude response shows the presence of *sidelobes*, an oscillatory effect around the main lobe.

Therefore, the windowing operation on the ideal impulse response, i.e., the circular convolution of the ideal frequency response with $W(e^{jw})$, produces two main effects:

1. The sharp transition from passband to stopband is smoothed by the convolution with the main lobe of width Δ.
2. Ripples appear both in the stopband and the passband due to the convolution with the sidelobes.

These effects are shown in Figure 9.4. It appears that the sharpness of the transition band and the size of the ripples are dependent on the shape of the window's Fourier transform; indeed, by carefully designing the shape of the windowing sequence we can

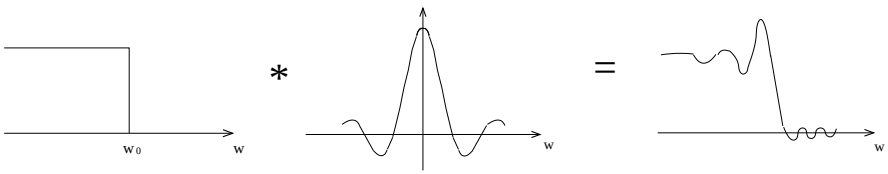


Figure 9.4: Convolution of the ideal filter with the window.

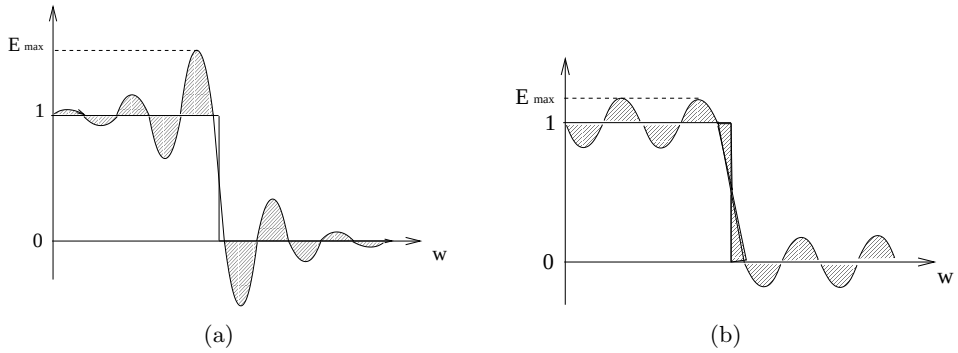


Figure 9.5: a) Minimum square error solution. b) Minimax solution.

trade mainlobe width for sidelobe amplitude and obtain a more controlled behavior in the frequency response of the approximation filter.

Although the MSE minimization procedure can lead to perfectly usable filters, its drawback is the lack of control of the maximum error in the frequency response with respect to the ideal filter. A more suitable approximation criterion may therefore be the *minimax* criterion, where we aim to explicitly minimize the *maximum* approximation error over the entire frequency support; this will be explained in detail in the next section. We can already say, however, that while the minimum square error is an integral criterion, the minimax is a pointwise criterion; or, more mathematically, that the MSE and the minimax are respectively $L_2([-\pi,\pi])$- and $L_\infty([-\pi,\pi])$-norm minimizations for the error function $E(\omega) = \hat{H}(e^{j\omega}) - H(e^{j\omega})$. Figure 9.5 illustrates the typical result of applying both criteria to the ideal lowpass problem. As it can be seen, the minimum square and minimax solutions are very different.

More General Windows. We have seen in the previous discussion the fundamental principles of FIR filter design by windowing. The starting point is the desired frequency response $H(e^{j\omega})$, from which the ideal impulse response $h[n]$ is obtained by the usual DTFT inversion formula

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

While the analytical evaluation of the above integral may be difficult or impossible in the general case, for frequency responses $H(e^{j\omega})$ which are *piecewise linear*, the computation of $h[n]$ can be carried out in an exact (albeit not trivial) way; the result will be a linear combination of modulated sinc and sinc-squared sequences[1]. The FIR approximation is

[1]For more details one can analyze the Matlab `fir1` function.

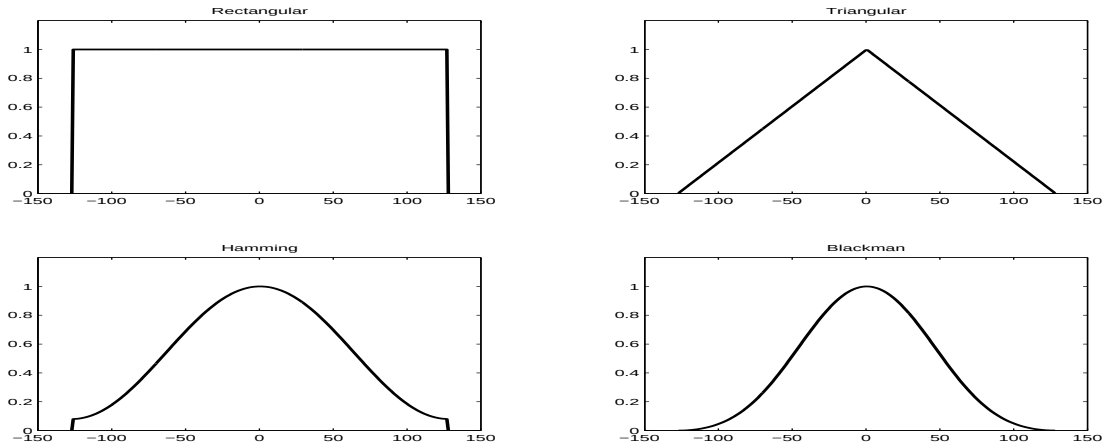


Figure 9.6: Some common windows in the time domain for $N = 127$ (total support 256 samples).

then obtained by applying a finite-length window $w[n]$ to the ideal impulse response:

$$\hat{h}[n] = w[n]h[n]$$

Since the frequency response of the FIR approximation is the circular convolution (in the frequency domain) of the desired response with the Fourier transform of the window, the window itself should be designed with the following goals in mind:

1. the window should be short, as to minimize the length of the FIR and therefore its computational cost
2. the spectrum of the window should be concentrated in frequency around zero as to minimize the "smearing" of the original frequency response; in other words, the window's main lobe should be as narrow as possible (it is clear that for $W(e^{j\omega}) = \delta(\omega)$ the resulting frequency response is identical to the original)
3. the unavoidable sidelobes of the window's spectrum should be small as to minimize the rippling effect in the resulting frequency response

It is clear that the first two requirements are openly in conflict; indeed, the width of the main lobe Δ is inversely proportional to the length of the window (we have seen, for instance, that for the rectangular window $\Delta = 4\pi/M$, with M the length of the filter). The second and third requirements are also in conflict, although the relationship between

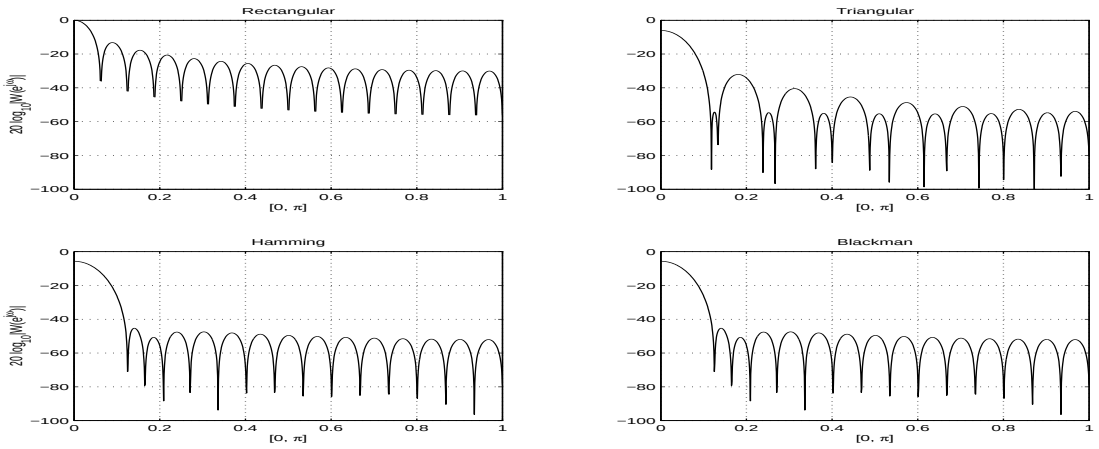


Figure 9.7: Some common windows in the frequency domain (magnitude in dBs).

mainlobe width and sidelobe amplitude is not straightforward and can be considered a design parameter. In the frequency response, reduction of the sidelobe amplitude implies that the Gibbs phenomenon is decreased, but at the price of an enlargement of the filter's transition band. For example, using a triangular window instead of a rectangular window strongly attenuates the ripples but, as a consequence, the transition band almost doubles.

There is a large body of literature concerning the design of windows; the most commonly used windows are those which admit a simple representation in closed form; amongst those we can name the triangular window, the Hamming window, and the Blackman window, which are plotted in Figure 9.6. For example, the zero-centered $(2N + 1)$-point Hamming window is defined as:

$$w(n) = 0.54 - 0.46\cos(2\pi(n + N)/2N), \quad |n| \leq N - 1$$

while the Blackman window is defined as:

$$w(n) = 0.42 - 0.5\cos(2\pi(n + N)/2N) + 0.08\cos(4\pi(n + N)/2N), \quad |n| \leq N - 1$$

Finally, it is worth mentioning the existence of the Kaiser window, in which a user definable parameter β is used to "tune" the mainlobe-sidelobe tradeoff independently of the window length.

9.3.2 Minimax FIR Filter Design

As we saw in the opening example, FIR filter design by windowing minimizes the overall mean square error between the desired frequency response and the actual response of the filter. Since this might lead to a very large error at frequencies near the transition band, we will now consider a different approach, namely the design by minimax optimization. This technique minimizes the maximum allowable error in the filter's magnitude response, both in the passband and in the stopband. Optimality in the minimax sense requires therefore the explicit stating of a set of *tolerances* in the prototypical frequency response, in the form of design specifications as seen in section 9.2.2. Before tackling the design procedure itself, we will need a series of intermediate results.

Generalized Linear Phase. In section 7.6.2 we introduced the concept of linear phase; a filter with linear phase response is particularly desirable since the phase response translates to just a time delay (possibly fractional) and we can concentrate on the magnitude response only. We also introduced the notion of group delay and showed that linear phase corresponds to constant group delay. Clearly the converse is not true: a frequency response of the type

$$H(e^{j\omega}) = |H(e^{j\omega})|e^{-j\omega d+j\alpha}$$

has constant group delay but differs from a linear phase system by a constant phase factor $e^{j\alpha}$. We will call this type of phase response *generalized linear phase*. Important cases are those for which $\alpha = 0$ (strictly linear phase) and $\alpha = \pi/2$ (generalized linear phase used in differentiators).

FIR Filter Types. Consider a causal, M-tap FIR filter with impulse response $h[n]$, $n = 0, 1, \ldots, M-1$; in the following we will be interested in filters whose impulse response is *symmetric or antisymmetric around its "midpoint"*. If the number of taps is odd, the midpoint of the impulse response coincides with the center tap $h[(M-1)/2]$; if the number of taps is even, on the other hand, the midpoint is still at $(M-1)/2$ but this value does not coincide with a tap since it is located "right in between" taps $h[M/2-1]$ and $h[M/2]$. Symmetric and antisymmetric FIR filters are important since their frequency response has generalized linear phase. The delay introduced by these filters is equal to $(M-1)/2$ samples; clearly this is an integer delay if M is odd, and it is fractional (half a sample more) if M is even. There are clearly four different possibilities for linear phase FIR impulse responses, which are listed here with their corresponding generalized linear phase parameters :

The generalized linear phase of (anti)symmetric FIR filters is easily shown. Consider for instance a Type I filter, and define $C = (M-1)/2$, the location of the center tap; we

Filter Type	Number of Taps	Symmetry	Delay	Phase Factor
Type I	odd	symmetric	integer	$\alpha = 0$
Type II	even	symmetric	fractional	$\alpha = 0$
Type III	odd	antisymmetric	integer	$\alpha = \pi/2$
Type IV	even	antisymmetric	fractional	$\alpha = \pi/2$

can compute the transfer function of the shifted impulse response $h_d[n] = h[n+C]$, which is now symmetric around zero. i.e. $h_d[-n] = h_d[n]$:

$$\begin{aligned} H_d(z) &= \sum_{n=-C}^{C} h_d[n] z^{-n} \\ &= h_d[0] + \sum_{n=-C}^{-1} h_d[n] z^{-n} + \sum_{n=1}^{C} h_d[n] z^{-n} \\ &= h_d[0] + \sum_{n=1}^{C} h_d[n](z^n + z^{-n}) \end{aligned} \tag{9.15}$$

By undoing the time shift we obtain the original Type I transfer function:

$$H(z) = z^{-\frac{M-1}{2}} H_d(z). \tag{9.16}$$

On the unit circle we have

$$\begin{aligned} H_d(e^{j\omega}) &= h_d[0] + \sum_{n=1}^{C} h_d[n](e^{j\omega n} + e^{-j\omega n}) \\ &= h_d[0] + 2\sum_{n=1}^{C} h_d[n] \cos n\omega \end{aligned} \tag{9.17}$$

which is a *real* function; the original Type I frequency response is obtained from (9.16)

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

which is clearly linear phase with delay $d = (M-1)/2$ and $\alpha = 0$. The generalized linear phase of the other three FIR types can be shown in exactly the same way.

Zero Locations. The symmetric structures of the four types of FIR filters impose some constraints on the locations of the zeros of the transfer function. Consider again a Type I filter; from (9.15) it is easy to see that $H_d(z^{-1}) = H_d(z)$; by using (9.16) we therefore have

$$\begin{cases} H(z) = z^{-\frac{M-1}{2}} H_d(z) \\ H(z^{-1}) = z^{\frac{M-1}{2}} H_d(z) \end{cases}$$

which leads to:

$$H(z^{-1}) = z^{M-1} H(z). \tag{9.18}$$

It is easy to show that the above relation is also valid for Type II filters, while for Type III and Type IV (antisymmetric filters) we have:

$$H(z^{-1}) = -z^{M-1} H(z). \tag{9.19}$$

These relations mean that if z_0 is a zero of a linear phase FIR, then so is z_0^{-1}. This result, coupled with the usual fact that all complex zeros come in conjugate pairs, implies that if z_0 is a zero of $H(z)$ then:

- If $z_0 = \rho \in \mathbb{R}$ then $(\rho, 1/\rho)$ are zeros.
- If $z_0 = \rho e^{j\theta}$ then $(\rho e^{j\theta}, (1/\rho)e^{j\theta}, \rho e^{-j\theta}, (1/\rho)e^{-j\theta})$ are zeros.

Consider now equation (9.18) again; if we set $z = -1$ we have

$$H(-1) = (-1)^{M-1} H(-1); \tag{9.20}$$

for Type II filters, $M-1$ is an odd number, which leads to the conclusion that $H(-1) = 0$; in other words, Type II filters *must* have a zero at $\omega = \pi$. Similar results can be demonstrated for the other filter types, and they are summarized as such:

Filter Type	Relation	Constraint on Zeros
Type I	$H(z^{-1}) = z^{M-1} H(z)$	No constraints
Type II	$H(z^{-1}) = z^{M-1} H(z)$	Zero at $z = -1$ (i.e. $\omega = \pi$)
Type III	$H(z^{-1}) = -z^{M-1} H(z)$	Zeros at $z = \pm 1$ (i.e. at $\omega = \pi$, $\omega = 0$)
Type IV	$H(z^{-1}) = -z^{M-1} H(z)$	Zero at $z = 1$ (i.e. $\omega = 0$)

These constraints are important in the choice of the filter type for a given set of specifications. Type II and Type III filters are not suitable in the design of highpass filters, for instance; similarly, Type III and Type IV filters are not suitable in in the design of lowpass filters.

Chebyshev Polynomials. Chebyshev polynomials are a family of orthogonal polynomials $\{T_k(x)\}_{k\in\mathbb{N}}$ which have, amongst others, the following interesting property:

$$\cos n\omega = T_n(\cos\omega); \tag{9.21}$$

in other words, the cosine of an integer multiple of an angle ω can be expressed as a polynomial in the variable $\cos\omega$. The first few Chebyshev polynomials are:

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x \\ T_4(x) &= 8x^4 - 8x^2 + 1 \end{aligned}$$

and, in general, they can be derived from the recursion formula

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x). \tag{9.22}$$

From the above table it is easy to see that we can write, for instance

$$\cos(3\omega) = 4\cos^3\omega - 3\cos\omega$$

The interest in Chebyshev polynomials comes from the fact that the zero-centered frequency response of a linear phase FIR can be expressed as a linear combination of cosine functions, as we have seen in detail for Type I filters in (9.17). By using Chebyshev polynomials we can rewrite such a response as just one big polynomial in the variable $\cos\omega$. Let us consider an explicit example for a length-7, Type I filter with nonzero coefficients $h[n] = [d\ c\ b\ a\ b\ c\ d]$; we have:

$$H_d(e^{j\omega}) = a + 2b\cos\omega + 2c\cos 2\omega + 2d\cos 3\omega$$

and by using the first four Chebyshev polynomials we can write:

$$\begin{aligned} H_d(e^{j\omega}) &= a + 2b\cos\omega + 2c(2\cos^2\omega - 1) + 2d(4\cos^3\omega - 3\cos\omega) \\ &= (a - 2c) + (2b - 6d)\cos\omega + 4c\cos^2\omega + 8d\cos^3\omega. \end{aligned} \tag{9.23}$$

In this case, $H_d(e^{j\omega})$ turns out to be a third degree polynomial in the variable $\cos\omega$. This is the case for any Type I filter, for which we can always write

$$H_d(e^{j\omega}) = \sum_{k=0}^{(M-1)/2} c_k \cos^k\omega \tag{9.24}$$

$$= P(x)|_{x=\cos\omega}. \tag{9.25}$$

where $P(x)$ is a polynomial of degree $(M-1)/2$ whose coefficients c_k are derived as linear combinations of the original filter coefficients a_k; we showed an example of this in (9.23). For the other types of linear phase FIR, a similar representation can be obtained after a few trigonometric manipulations. The general expression is:

$$\begin{aligned} H_d(e^{j\omega}) &= f(\omega)\sum_{k=0}^{L} c_k \cos^k \omega \\ &= f(\omega)P(x)|_{x=\cos\omega}; \end{aligned}$$

where the c_k's are still linear combinations of the original filter coefficients and where $f(\omega)$ is a weighting trigonometric function. Both $f(\omega)$ and the polynomial degree K vary as a function of the filter type[2]. In the following sections, however, we will concentrate only on the design of Type I filters, so these details will be overlooked; in practice, since the design is always carried out using numerical packages, the appropriate formulation for the filter expression is taken care of automatically.

Polynomial Optimization. Back to the filter design problem, we know that the FIR filters are (generalized) linear phase, so we can concentrate on the real frequency response of the zero-centered filter, which is represented by the trigonometric polynomial (9.25). Also, since the impulse response is real and symmetric, the aforementioned real frequency response is also symmetric around $\omega = 0$. Therefore the filter design procedure can be carried out only for values of ω over the interval $[0, \pi]$, with the other half of the spectrum obtained by symmetry. For these values of ω, the variable $x = \cos\omega$ is mapped onto the interval $[1, -1]$ and the mapping is invertible. Therefore, *the filter design problem becomes a problem of polynomial approximation over an interval.*

To illustrate the procedure by example, consider once more the set of filter specifications in Figure 9.1 and suppose we decide to use a Type I filter. Recall that we required the prototype filter to be lowpass, with a transition band from $\omega_p = 0.4\pi$ to $\omega_s = 0.6\pi$; we further stated that the tolerances for the realized filter's magnitude must not exceed 10% in the passband and 1% in the stopband. This implies that the maximum magnitude error between the prototype filter and the FIR filter response $H(e^{j\omega})$ must not exceed $\delta_p = 0.1$ in the interval $[0, \omega_p]$ and must not exceed $\delta_s = 0.01$ in the interval $[\omega_s, \pi]$. We

[2]For the sake of completeness, here is a summary of the details:

Filter Type	L	$f(\omega)$
Type I	$(M-1)/2$	1
Type II	$(M-2)/2$	$\cos(\omega/2)$
Type III	$(M-3)/2$	$\sin(\omega)$
Type IV	$(M-2)/2$	$\sin(\omega/2)$

can formulate this as follows: the frequency response of the desired filter is:

$$H_D(e^{j\omega}) = \begin{cases} 1 & \omega \in [0, \omega_p] \\ 0 & \omega \in [\omega_s, \pi] \end{cases} \tag{9.26}$$

(note that $H_D(e^{j\omega})$ is not specified in the transition band). Since the tolerances on passband and stopband are different, they can be expressed in terms of a weighting function $W(\omega)$ such that the tolerance on the error is constant over the two bands:

$$W(\omega) = \begin{cases} 1 & \omega \in [0, \omega_p] \\ \delta_p/\delta_s & \omega \in [\omega_s, \pi] \end{cases} \tag{9.27}$$

The design problem can now be reformulated as follows by defining an error function

$$E(\omega) = W(\omega)\left[H_d\left(e^{j\omega}\right) - H_D\left(e^{j\omega}\right)\right]. \tag{9.28}$$

Then the optimization problem becomes

$$\min_{\hat{\mathbf{h}}} \left\{ \max_{\omega \in F} |E(\omega)| \right\}, \tag{9.29}$$

where F is the closed subset of $0 \leq \omega \leq \pi$ such that

$$F = \{[0, \omega_p] \cup [\omega_s, \pi]\} = I_p \cup I_s \quad with \quad I_p = [0, \omega_p], I_s = [\omega_s, \pi], \tag{9.30}$$

and the question now is to find the coefficients for $h[n]$ (their number M and their values) which minimize the above error. If we have a feasible solution to (9.29) such that

$$\max_{\omega \in F} |E(\omega)| \leq \delta_p, \tag{9.31}$$

then we know that for $\omega \in I_p = [0, \omega_p]$,

$$\left| H_{des}\left(e^{j\omega}\right) - \hat{H}\left(e^{j\omega}\right) \right| \leq \delta_p, \tag{9.32}$$

and for $\omega \in I_s = [\omega_s, \pi]$, we have

$$\left| H_{des}\left(e^{j\omega}\right) - \hat{H}\left(e^{j\omega}\right) \right| \leq \delta_s. \tag{9.33}$$

Hence a feasible solution satisfying (9.31) is what is needed for a design. This optimization framework allows us to check if we can meet the desired specification. Note that we leave the transition band unconstrained (i.e., it doesn't affect the minimization of the error). Thus we can define

$$\tilde{W}(\omega) = \begin{cases} 1 & \omega \in [0, \omega_p] \\ \delta_p/\delta_s & \omega \in [\omega_s, \pi] \\ 0 & \omega \in (\omega_p, \omega_s), \end{cases} \tag{9.34}$$

and write

$$\tilde{E}(\omega) = \tilde{W}(\omega)\left[H_d\left(e^{j\omega}\right) - H_D\left(e^{j\omega}\right)\right], \tag{9.35}$$

with the feasibility check of

$$\|\tilde{E}(\omega)\|_\infty \triangleq \max_\omega |\tilde{E}(\omega)| \leq \delta_p. \tag{9.36}$$

Since

$$\|\tilde{E}(\omega)\|_\infty = \max_\omega |\tilde{E}(\omega)| = \max_{\omega \in I_p \cup I_s} |\tilde{E}(\omega)|,$$

this gives the same conditions as (9.32) and (9.33).

Therefore, the optimization problem we have is a $\|\cdot\|_\infty$ norm minimization instead of $\|\cdot\|_2$ norm minimization we had from before in (9.11). This philosophy of viewing the different filter design criteria as just optimization problems is a useful viewpoint which can be generalized to other design criteria. That is, (9.29) is just

$$\min_{\hat{H}(e^{j\omega})} \|\tilde{E}(\omega)\|_\infty.$$

The next step is to use (9.25) to reformulate the above expression as a polynomial optimization problem. To do so we replace the frequency response $H_d(e^{j\omega})$ with its polynomial equivalent and set $x = \cos\omega$; the passband interval $[0, \omega_p]$ and the stopband interval $[\omega_s, \pi]$ are mapped into the intervals for x:

$$\begin{aligned} I_p &= [\cos\omega_p, 1] \\ I_s &= [-1, \cos\omega_s], \end{aligned}$$

respectively; similarly, the desired response becomes:

$$D(x) = \begin{cases} 1 & \omega \in I_p \\ 0 & \omega \in I_s \end{cases} \tag{9.37}$$

and the weighting function becomes:

$$W(x) = \begin{cases} 1 & \omega \in I_p \\ \delta_p/\delta_s & \omega \in I_s \end{cases} \tag{9.38}$$

The new set of specifications are shown in Figure 9.8. Within this polynomial formulation, the optimization problem becomes:

$$\max_{x \in I_p \cup I_s} \{W(x)|P(x) - D(x)|\} = \max\{|E(x)|\} \leq \delta_p \tag{9.39}$$

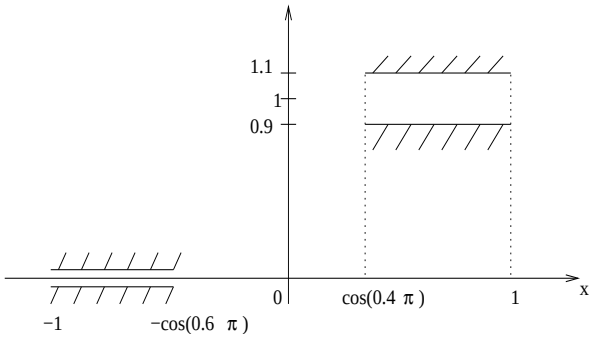


Figure 9.8: The filter specifications as in Figure 9.1 formulated here in terms of polynomial approximation, i.e. $x = \cos\omega$ for $\omega \in [0, \pi]$.

where $P(x)$ is the polynomial representation of the FIR frequency response as in (9.25).

Alternation Theorem:

The optimization problem stated by (9.39) can be solved by using the following theorem:

Theorem 9.1 *Consider a set* $\{I_k\}$ *of* closed, *disjoint intervals on the real axis and their union* $I = \cup_k I_k$. *Consider further:*

- *a polynomial* $P(x)$ *of degree* L, $P(x) = \sum_{n=0}^{L} a_n x^n$
- *a desired function* $D(x)$, *continuous over* I
- *a positive weighting function* $W(x)$

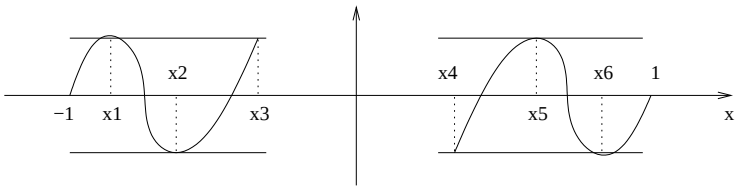


Figure 9.9: Equiripple error in passband and stopband

Consider now the approximation error function

$$E(x) = W(x)[D(x) - P(x)]$$

and the associated maximum approximation error over the set of closed intervals

$$E_{\max} = \max_{x \in I}\{|E(x)|\}$$

Then $P(x)$ is the unique *order-L polynomial which minimizes $E_{\max}$ if and only if there exist* at least $L + 2$ *successive values x_i in I such that $|E(x_i)| = E_{\max}$ and*

$$E(x_i) = -E(x_{i+1}).$$

In other words, the error function must have at least $L + 2$ alternations between its maximum and minimum values. Such a function is called equiripple.

Back to our lowpass filter example, assume we are trying to design a 9-tap optimal filter. This theorem tells us that if we found a polynomial $P(x)$ of degree 4 such that the error function (9.39) over I_p and I_s looks as in Figure 9.9 (6 alternations), then the polynomial would be the *optimal* and *unique* solution. Note that the extremal points (i.e. the values of the error function at the edges of the optimization intervals) *do* count in the number of alternations since the intervals I_k are closed.

The above theorem may seem a bit far-fetched since it does not tell us how to find the coefficients but it only gives us a test to verify their optimality. This test, however, is at the core of an *iterative* algorithm which refines the polynomial from an initial guess until the optimality condition is met. Before considering the optimization procedure more in detail, we will state without formal proof three consequences of the alternation theorem as it applies to the design of Type I lowpass filters:

- The minimum number of alternations for an optimal M-tap lowpass filter is $L + 2$, with $L = (M - 1)/2$; this is the result of the alternation theorem. The *maximum* number of alternation, however, is $L + 3$; filters with $L + 3$ alternation are called extraripple filters.

- Alternations always take place at $x = \cos\omega_p$ and $x = \cos\omega_s$ (i.e. at $\omega = \omega_p$ and $\omega = \omega_s$.

- If the error function has a local maximum or minimum, its absolute value at the extremum must be equal to $E_{\max}$ except possibly in $x = 0$ or $x = 1$. In other words, all local maxima and minima of the frequency response must be alternations except in $\omega = 0$ or $\omega = \pi$.

- If the filter is extraripple, the extra alternation occurs at either $\omega = 0$ or $\omega = \pi$.

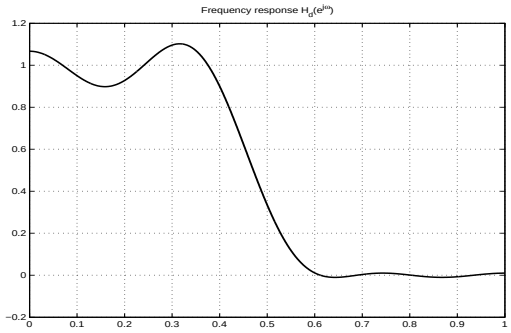


Figure 9.10: An optimal 13-tap Type I filter which does not meet the error specifications.

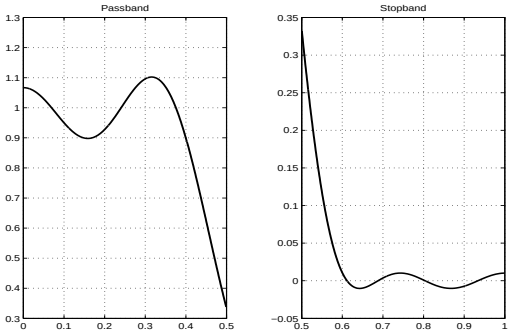


Figure 9.11: Details of passband and stopband of the frequency response in Figure 9.10.

Optimization Procedure. Finally, by putting all the elements together, we are ready to state an algorithmic optimization procedure for the design of optimal minimax FIR filters; this procedure is usually called the Parks-McClellan algorithm. Remember, we are trying to determine a polynomial $P(x)$ such that the approximation error in (9.39) is equiripple; for this, we need to determine both the degree of the polynomial and its coefficients. For a given degree L, for which the resulting filter will have $2L + 1$ taps, the L coefficients are found by an iterative procedure which successively refines an initial guess for the $L + 2$ alternation points x_i until the error is equiripple[3]. After the iteration has converged, we need to check that the corresponding $E_{\max}$ satisfies the upper bound

[3]Details about this crucial optimization step can be found in the bibliographic references. While a thorough discussion of the algorithm is beyond the scope of these notes, suffice it to say that at each iteration the new set of candidate extremal points is obtained by exchanging the old set with the ordinates of the current local maxima. This trick is also known as the Remez exchange algorithm and that is why, in Matlab, the Parks-McClellan algorithm is named `remez`.

imposed by the specifications; when this is not the case, the degree of the polynomial (and therefore the length of the filter) must be increased and the procedure must be restarted. Once the conditions on the error are satisfied, the filter coefficients can be obtained by inverting the Chebyshev expansion.

As a final note, an initial guess for the number of taps can be obtained using the empirical formula by Kaiser; for an M-tap FIR $h[n]$, $n = 0, \ldots, M-1$:

$$M \simeq \frac{-10\log_{10}(\delta_p\delta_s) - 13}{2.324\Omega} + 1$$

where δ_p is the passband tolerance, δ_s is the stopband tolerance and $\Omega = \omega_s - \omega_p$ is the width of the transition band.

The final design. We will now summarize the design steps for the specifications in Figure 9.1. We will use a Type I FIR. We start by using Kaiser's formula to obtain an estimate of the number of taps: since $\delta_p\delta_s = 10^{-3}$ and $\Omega = 0.2\pi$, we obtain $M = 12.6$ which we will round up to 13 taps. At this point we can use any numerical package for filter design to run the Parks-McClellan algorithm. In Matlab this would be

```
[h, err] = remez(12, [0 0.4 0.6 1], [1 1 0 0], [1 10]);
```

The resulting frequency response is plotted in Figure 9.10; please note that we are plotting the frequency responses of the zero-centered filter $h_d[n]$, which is a real function of ω. We can verify that the filter has indeed $(M-1)/2 = 6$ alternation by looking at a blowup picture of the passband and the stopband, as in Figure 9.11. The maximum error as returned by Matlab is however 0.102 which is larger than what our specifications called for, i.e. 0.1. We are thus forced to increase the number of taps; since we are using a Type I filter, the next choice is $M = 15$. Again, the error turns out to be larger than 0.1, since in this case we have $E_{\max} = 0.1006$. The next choice, $M = 17$, finally yields an error $E_{\max} = 0.05$, which exceeds the specifications by a factor of 2. It's the designer's choice to decide whether the computational gains of a shorter filter ($M = 15$) outweigh the small excess error. The impulse response and the frequency response of the 17-tap filter are plotted in Figure 9.12.

Other Types of Filters. The Parks-McClellan optimal FIR design procedure can be made to work for arbitrary filter types as well, such as highpass and passband of course but also for more sophisticated frequency responses. The constraints imposed by the zero locations as we saw on page 216 determine the type of filter to use; once the desired response $H_D(e^{j\omega})$ is expressed as a trigonometric function, the optimization algorithm can take its course. For arbitrary frequency responses, however, the fact that the transition bands are left unconstrained may lead to unacceptable peaks which render the filter useless. In these

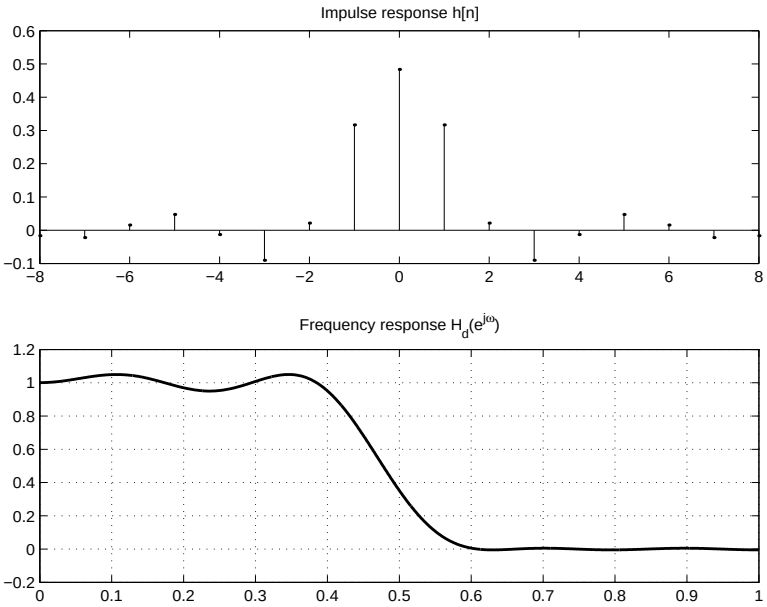


Figure 9.12: The 17-tap filter meeting the specifications.

cases, visual inspection of the obtained response is mandatory and experimentation with different filter lengths and tolerance may improve the final result.

Example 9.1

In the window method for filter design, one multiplies the desired impulse response $h[n]$ *(potentially of infinite length) with a window* $w[n]$ *of finite length* M *to obtain an approximation of the desired response* i.e., $h_{approx}[n] = h\left[n - \frac{M-1}{2}\right] w[n]$. *The Hamming window is defined as*

$$w[n] = \begin{cases} 0.54 - 0.46\cos\frac{2\pi n}{M-1} & \text{for } 0 \le n \le M-1 \\ 0 & \text{otherwise} \end{cases}$$

where M *is the number of taps (see Fig. 9.13).*

(a) We will use the window method with a Hamming window to design a $M = 21$*-tap differentiator. The frequency response of a differentiator is* $H(e^{j\omega}) = j\omega$ *for* $-\pi \le$

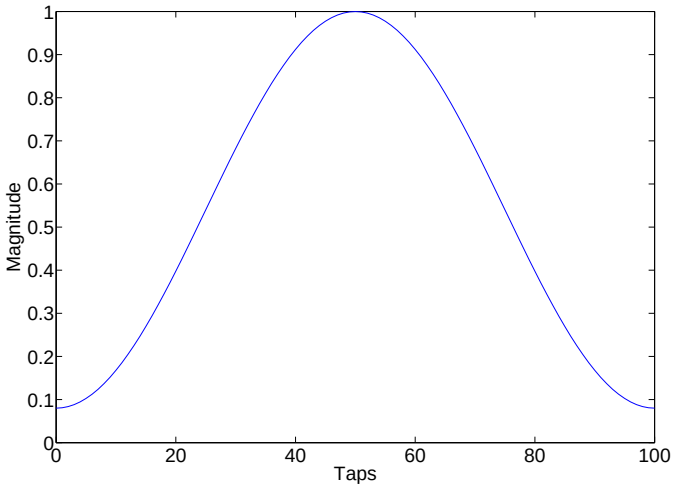


Figure 9.13: A Hamming window of length $M = 101$

$\omega \leq \pi$. *Give an expression for the impulse response* $h_{approx}[n]$. *Provide a single plot with* $h_{approx}[n]$, $w[n]$ *and* $h[n]$ *(Hint: you can superimpose plots using the* `hold on` *command in matlab* e.g. `plot([1 2 3],[3 2 1],'r:');hold on;plot([1 2 3],[3 3 3],'g-');plot([1 2 3],[4 5 29],'b*')`*)*

(b) Provide matlab plots of the phase and amplitude of the frequency response $H_{approx}(e^{j\omega})$, *as well as plots of the phase and amplitude of the desired frequency response* $H(e^{j\omega})$, *if possible combine the amplitude plots on one figure and the phase plots on one figure (Use matlab to compute the DTFT* $H_{approx}(e^{j\omega})$, *plot* $H(e^{j\omega})$ *for a large number of values of* ω*). Comment the result.*

Solution:

(a) We first need to compute $h[n]$ *for* $n \neq 0$*:*

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

For $n = 0$,

$$h[0] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\omega})e^{0}d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} j\omega d\omega = 0$$

Hence,

$$\begin{aligned} h_{approx}[n] &= h[n - \tfrac{M-1}{2}]w[n] \\ &= (0.54 - 0.46\cos\tfrac{2\pi n}{M-1})(\tfrac{\cos(\pi(n-\frac{M-1}{2}))}{n-\frac{M-1}{2}}) \\ &= 0.54\tfrac{\cos(\pi(n-\frac{M-1}{2}))}{n-\frac{M-1}{2}} - 0.46\tfrac{\cos\frac{2\pi n}{M-1}\cos(\pi(n-\frac{M-1}{2}))}{n-\frac{M-1}{2}} \end{aligned}$$

In Fig. 9.14 we show $h[n - \frac{M-1}{2}]$, $w[n]$ *and* $h_{approx}[n]$.

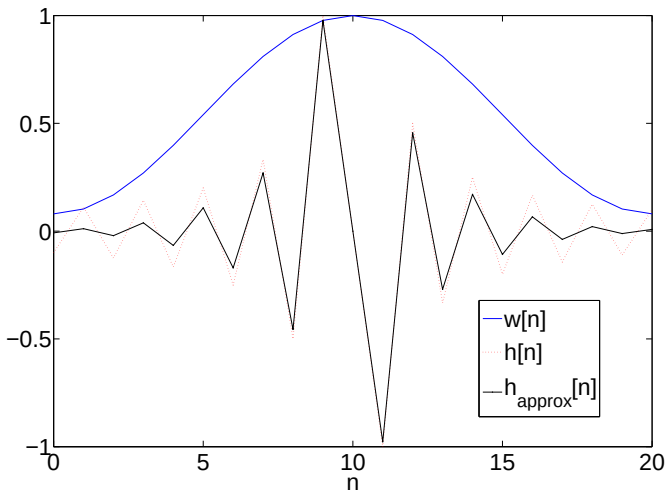


Figure 9.14: $h[n - \frac{M-1}{2}]$, $w[n]$ and $h_{approx}[n]$

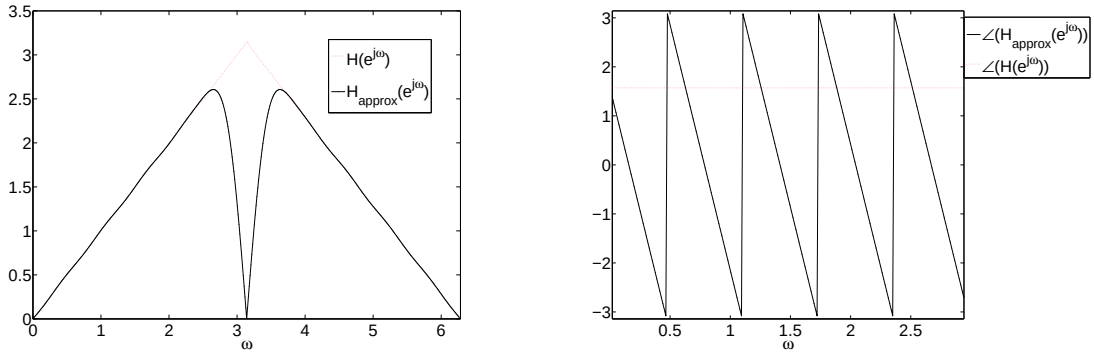
(b) In Fig. 9.15, we show the amplitude and frequency responses of $H_{approx}(e^{j\omega})$ *and* $H(e^{j\omega})$.

Example 9.2 (Gibbs Phenomenon) *In this exercise we will demonstrate the Gibbs phenomenon through rectangular windowing.*

Suppose we want to design a lowpass filter with a cut-off frequency of $\pi/2$, i.e. *we have a desired frequency response*

$$H_{\text{des}}\left(e^{j\omega}\right) = \begin{cases} 1 & -\frac{\pi}{2} \le \omega \le \frac{\pi}{2} \\ 0 & \textit{elsewhere.} \end{cases}$$

Let $h_{\text{des}}[n]$ *be the corresponding impulse response.*



(a) Amplitude response (b) Phase response

Figure 9.15: Amplitude and phase responses of $H_{approx}(e^{j\omega})$ and $H(e^{j\omega})$

We want to create a $2N+1$ *taps filter that represents the desired response as close as possible. We do this by applying a rectangular window* $w[n]$ *that is defined by*

$$w[n] = \begin{cases} 1 & -N \leq n \leq N \\ 0 & \textit{elsewhere.} \end{cases}$$

The resulting filter is given by

$$\hat{h}[n] = h_{\text{des}}[n]w[n]. \tag{9.40}$$

The goal of this exercise is to see the difference between $\left|\hat{H}\left(e^{j\omega}\right)\right|$ *and* $\left|H_{\text{des}}\left(e^{j\omega}\right)\right|$.

(a) Give the desired impulse response $h_{\text{des}}[n]$.

(b) Let $N = 10$. *Use the* `fft` *function in MATLAB to plot* 1000 *points of* $\left|\hat{H}_{\text{des}}\left(e^{j\omega}\right)\right|$ *in the interval* $0.4\pi \leq \omega \leq 0.6\pi$.

(c) Repeat Part (b) for $N = 100$ *and* $N = 200$. *Give a printout of the plot only.*

(d) How does the maximum of $\left|\hat{H}\left(e^{j\omega}\right)\right|$ *depend on* N*?*

(e) We have seen in class that applying a rectangular window corresponds to the optimal solution according to some optimization criteria/constraints. Formulate the complete optimization problem that has Equation (9.40) as a solution.

Solution:

(a) As seen in class

$$h_{\text{des}}[n] = \frac{1}{2}\text{sinc}\left(\frac{n}{2}\right).$$

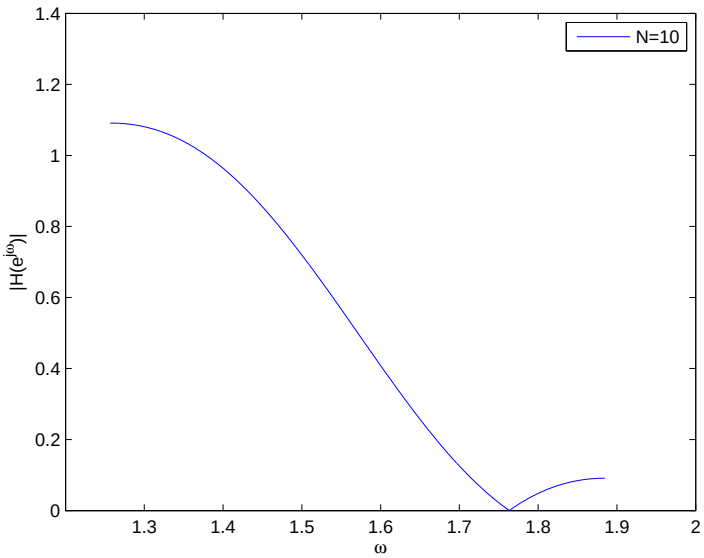
(b) We want 10^3 *points in the interval* $0.4\pi \leq \omega \leq 0.6\pi$. *This interval has length* 0.2π. *This means that we need* 10^4 *points in the entire interval. We take a DFT with* 10^4 *points.*

```
>> N = 10;
>> h_des = .5*sinc([-N:N]/2);
>> H_des = fft(h_des,1e4);
```

The interval $0.4\pi \leq \omega \leq 0.6\pi$ *corresponds to elements* $2001 \ldots 3000$ *of the DFT.*

```
>> w = linspace(0.4*pi,0.6*pi,1e3);
>> plot(w,abs(H_des(2001:3000)));
>> xlabel('\omega');
>> ylabel('|H(e^{j\omega})|');
>> legend('N=10')
```

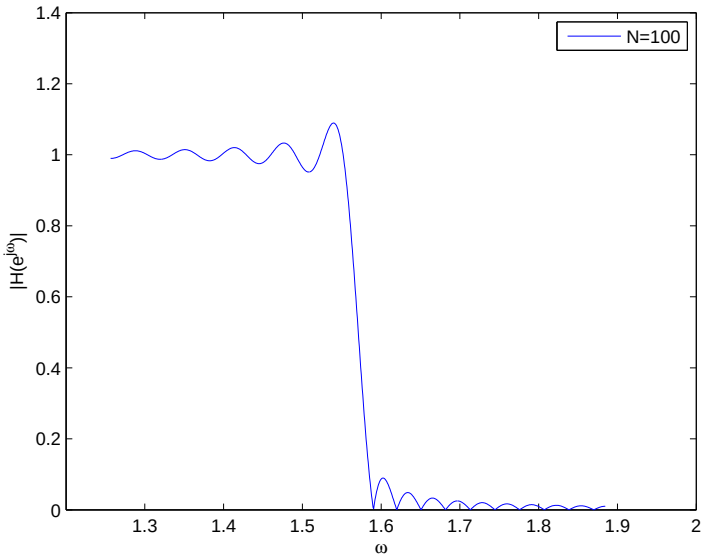
This gives us



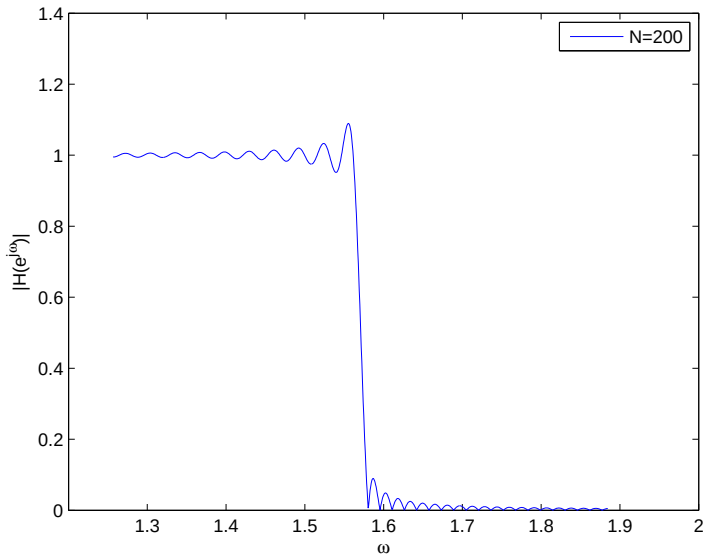
(c) For $N = 100$ we have

```
>> N = 100;
>> h_des = .5*sinc([-N:N]/2);
>> H_des = fft(h_des,1e4);
>> w = linspace(0.4*pi,0.6*pi,1e3);
>> plot(w,abs(H_des(2001:3000)));
>> xlabel('\omega');
>> ylabel('|H(e^{j\omega})|');
>> legend('N=100')
```

which gives



For $N = 200$ we give the plot only:



(d) The maximum does not depend on N, it is always around 1.09.

(e)

$$\begin{aligned} &\textit{minimize} && \left\| \hat{H}\left(e^{j\omega}\right) - H\left(e^{j\omega}\right) \right\|_2^2, \\ &\textit{subject to} && \hat{H}\left(e^{j\omega}\right) = \sum_{n=-N}^{N} \hat{h}[n] e^{-j\omega n}, \end{aligned}$$

where $\|\cdot\|_2^2$ *is the* $L_2[-\pi, \pi]$ *norm as defined in class.*

We see from part (d) that for increasing N *the approximation gets better in the mean-square sense, but that the maximum error remains about* 9%.

9.4 IIR Filter Design

As we mentioned in the introductory remarks, no optimal procedure exists for the design of IIR filters. The fundamental reason is that the optimization of the coefficients of a rational transfer function is a highly nonlinear problem and no satisfactory algorithm has yet been developed for the task. This, coupled with the impossibility of obtaining a linear phase response with an IIR[4] makes the design of IIR filter a much less formal art. Here we will concentrate on some basic IIR filters which are very simple and which are commonly used in practice and we will briefly illustrate the basic principles behind more general IIR design techniques.

9.4.1 All-Time Classics

There are a few tried-and-true applications in which simple IIR structures are the design of choice. These filters are so simple and so well behaved that they are a fundamental tool in the arsenal of any signal processing engineer.

DC Removal and Mean Estimation. The DC component of a signal is its mean value; a signal with zero mean is also called an AC signal. This nomenclature comes from electrical circuit parlance: DC is shorthand for *direct current*, while AC stands for *alternating current*; you might be familiar with these terms in relation to the current provided by a battery (constant and hence DC) and the current available from a mains socket (alternating at 50 or 60 Hz and therefore AC).

For a given sequence $x[n]$, one can always write

$$x[n] = x_{\mathrm{AC}}[n] + x_{\mathrm{DC}}$$

where x_{DC} is the mean of the sequence values. Please note that:

- The DC value of a finite-support signal is the value of its Fourier transform at $\omega = 0$, times the length of the signal's support
- The DC value of an infinite-support signal must be zero for the signal to be absolutely summable.

In most signal processing applications, where the input signal comes from an acquisition device (such as a sampler, a soundcard and so on), it is important to remove the DC component; this is because the DC offset is often a random offset caused by ground mismatches between the acquisition device and the associated hardware.

[4]There actually is a theorem which states that an infinite impulse response with linear phase is not realizable.

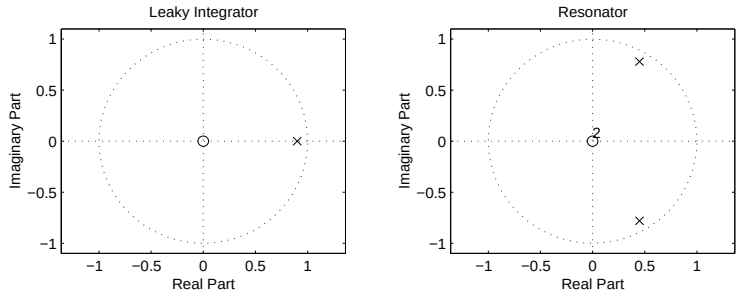


Figure 9.16: Pole-zero plots for the leaky integrator and the simple resonator.

For finite-length signals, computation of the mean is straightforward since it involves a finite number of operations. In most cases, however, we do not want to wait until the end of the signal before we try to remove its mean; what we need is some way to perform DC removal *on line*. The approach is therefore to obtain at each instant an *estimate* of the DC component from the past signal values, with the assumption that the estimate converges to the real mean of the signal. In order to obtain such an estimate, i.e. in order to obtain the average value of the past input samples, both approaches detailed in section 7.4 are of course valid (i.e. the Moving Average and the Leaky Integrator filters) . We have seen, however, that the leaky integrator provides a superior cost/benefit tradeoff and therefore the output of a leaky integrator with λ very close to one (usually 10^{-3}) is the estimator of choice for the DC component of a signal. The closer λ is to one, the more accurate the estimation; the speed of convergence of the estimate however becomes slower and slower as $\lambda \to 1$. This can be easily seen from the group delay at $\omega = 0$, which is

$$\mathrm{grd}\{H(1)\} = \frac{\lambda}{1-\lambda}$$

Resonator and Notch Filter Let's look again at how the leaky integrator works. Consider its Z-transform

$$H(z) = \frac{1-\lambda}{1-\lambda z^{-1}}$$

and notice that what we really want the filter to do is to extract the zero-frequency component (i.e. the frequency component that does not oscillate, i.e. the DC component). To do so, we placed a pole near $z = 1$, which of course corresponds to $z = e^{j\omega}$ for $\omega = 0$. Since the magnitude response of the filter will exhibit a peak near a pole, and since the peak will be higher the closer the pole is to the unit circle, we are in fact amplifying the zero-frequency component; this is apparent from the plot of the filter's frequency response

in Figure 7.9. The numerator, $1 - \lambda$, is chosen such that the magnitude of the filter at $\omega = 0$ is one; the net result is that the zero-frequency component will pass unmodified while all the other frequencies will be attenuated. The value of a filter's magnitude at a given frequency is often called the *gain*.

The very same approach can now be used to extract a signal component at *any* frequency. We will use a pole whose magnitude is still close to one (i.e. a pole near the unit circle) but whose phase is that of the frequency we want to extract. We will then choose a numerator so that the magnitude is unity at the frequency of interest. The one extra detail is that, since we want a real-valued filter, we will have to place a complex conjugate pole as well. The resulting filter is called a resonator and a typical pole-zero plot is shown in Figure 9.16. The Z-transform of a resonator at frequency ω_0 is therefore determined by the pole $p = \lambda e^{j\omega_0}$ and by its conjugate:

$$H(z) = \frac{G_0}{(1 - pz^{-1})(1 - p^*z^{-1})} = \frac{G_0}{1 - (2\lambda\cos\omega_0)z^{-1} + \lambda^2 z^{-2}} \tag{9.41}$$

The numerator value G_0 is computed so that the filter's gain at $\pm\omega_0$ is one; since in this case $|H(e^{j\omega_0})| = |H(e^{-j\omega_0})|$, we have

$$G_0 = (1 - \lambda)\sqrt{1 + \lambda^2 - 2\lambda\cos 2\omega_0}.$$

The magnitude and phase of a resonator with $\lambda = 0.9$ and $\omega_0 = \pi/3$ are shown in Figure 9.17.

A simple variant on the basic resonator can be obtained by considering the fact that the resonator is just a bandpass filter with a very narrow passband. As for all passband filters, we can therefore place a zero at $z = \pm 1$ and sharpen its midband frequency response. The corresponding Z-transform is now

$$H(z) = G_1 \frac{1 - z^{-2}}{1 - (2\lambda\cos\omega_0)z^{-1} + \lambda^2 z^{-2}}$$

with

$$G_1 = \frac{G_0}{\sqrt{2(1 - \cos 2\omega_0)}}$$

The corresponding magnitude response is shown in Figure 9.18.

9.4.2 IIR Design by Bilinear Transformation

As we mentioned in the introduction, analog filter design techniques give rise to analog filters whose transfer function (i.e. the Laplace transform of the continuous-time impulse response) is formally similar to the rational Z-transforms obtained from a constant-coefficient

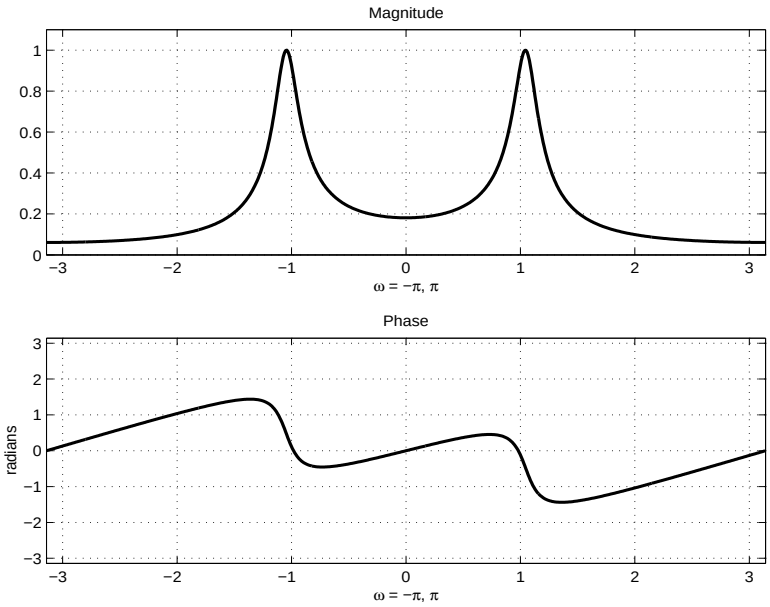


Figure 9.17: Frequency response of the simple resonator.

difference equation. This suggests associating a discrete-time IIR filter to the filter prototypes obtained in the continuous-time. While the details of analog filter design are outside the scope of these notes, it is important to mention that the techniques involved have an established and proven tradition; tabulated coefficients are readily available to determine the exact characteristics of an analog filter and the values of the electronic components which need to be employed. This is in stark contrast with the lack of an optimized IIR design procedure in the discrete-time domain.

The Analog Prototype. We will illustrate the design procedure by example. There are three fundamental families of (passive) analog lowpass filters: Butterworth, Chebyshev and Elliptic filters with the simplest type being Butterworth filters. Since we are going to illustrate the IIR design procedure by example, we will concentrate on this filter family.

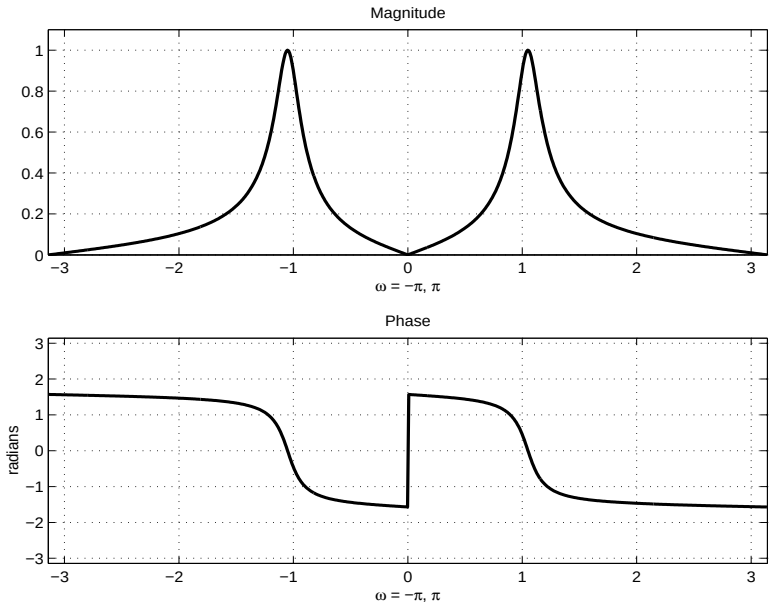


Figure 9.18: Frequency response of the modified resonator.

A *normalized* Butterworth filter of order N has an all-pole transfer function of the form

$$\begin{aligned} H(s) &= \frac{1}{(s-s_0)(s-s_1)\dots(s-s_{N-1})} \\ &= \left[\prod_{k=0}^{N-1}(s-s_k)\right]^{-1} \qquad (9.42) \\ &= \left[1+\sum_{k=1}^{N} a_k s^k\right]^{-1} \qquad (9.43) \end{aligned}$$

The values of the poles are derived by imposing that the square magnitude of the frequency response (i.e the magnitude of the continuous-time Fourier transform) must be of the form

$$|H(j\Omega)|^2 = \frac{1}{1+\Omega^{2N}} \qquad (9.44)$$

which is plotted in Figure 9.19 for different values of N. The most important feature of this magnitude function is that it is monotonic. It is immediate to see that the squared magnitude response is equal to $1/2$ at $\Omega = 1$, which is the cutoff frequency of the *normalized*

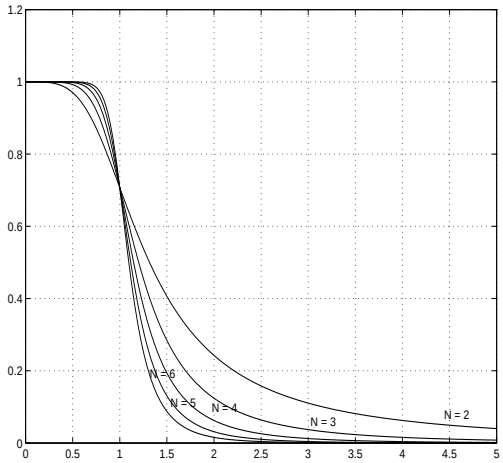


Figure 9.19: Magnitude of $H(j\Omega)$ for Butterworth filters of increasing order.

Butterworth. Since the impulse response must be real (we are designing an analog filter after all) we have that $H(-s) = H^*(s)$ and therefore (9.44) for $s = j\Omega$ translates to

$$|H(s)|^2 = H(s)H(-s) = \frac{1}{1 + (s/j)^{2N}}$$

which, in turn, gives the implicit locations of the poles as $s^{2N} = -j^{2N} = (-1)^{N-1}$. By solving for s, we have finally

$$s_k = e^{j\frac{\pi}{N}(k+\frac{N-1}{2})}, \quad k = 0, 1, \ldots, 2N-1$$

The poles, as it appears, are regularly distributed around the unit circle in the s-plane and their angular spacing is π/N; to obtain a stable and causal filter of order N, we just need to select the N poles in the left half of the s-plane (this is of course a standard result of System Theory). At this point we could plug these values back in (9.42) and obtain the transfer function coefficients in (9.43); in reality this is not how it's done, since tables exist which directly provide not only the a_k's for all practical values of N, but also the values of the electronic components necessary to build the filter. Clearly, Butterworth design is extremely straightforward from a practical point of view. As a last comment, note that if a different cutoff frequency Ω_c is desired, one just needs to replace s_k by $\Omega_c s_k$ which, in turn, corresponds to scaling each a_k by Ω_c^k.

Bilinear Transformation. The analog design part of a Butterworth filter is just a simple table lookup and we now want to transfer the properties of the filter to the discrete-time domain. The idea is to transform the transfer function of a continuous time filter into a transfer function for a discrete-time filter; this can be achieved by replacing the variable s in $H(s)$ by a suitable function of the variable z. The function in question has to satisfy certain properties and, above all, it has to preserve the stability of the resulting filter. A common mapping function is the *bilinear transformation*:

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

which is invertible via:

$$z = \frac{1 + s/2}{1 - s/2}$$

It can be verified that:

- The "frequency" axis $j\Omega$ in the s-plane is mapped onto the unit circle in the z-plane. This preserves the overall characteristic of the frequency response
- The left half of the s-plane is mapped *inside* the unit circle in the z-plane. This preserves the filter's stability.

The mapping linking the continuous frequency axis $j\Omega$ in the s plane to the periodic frequency axis $e^{j\omega}$ in the z-plane is given by:

$$\omega = 2\arctan(\Omega/2) \tag{9.45}$$

and conversely:

$$\Omega = 2\tan(\omega/2) \tag{9.46}$$

This represents a non-linear compression of the frequency axis, and therefore care must be taken in designing the filter specifications. An example of bilinear transformation is represented graphically in Figure 9.20.

A Design Example. Given a set of discrete-time specifications such as those in Figure 9.1, the design of a Butterworth digital filter involves the following steps: the starting point is the square-magnitude expression for the non-normalized filter

$$|H_c(j\Omega)|^2 = \frac{1}{1 + (\Omega/\Omega_c)^{2N}}.$$

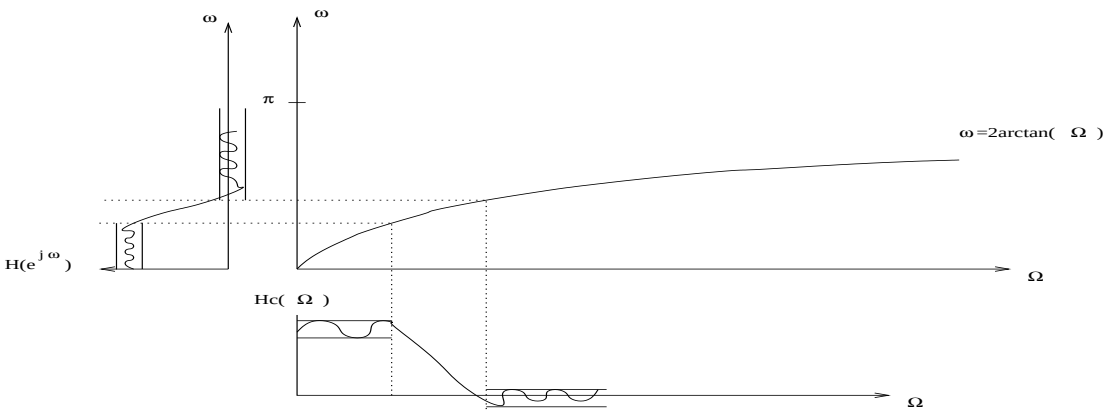


Figure 9.20: The mapping between Ω and ω operated by the bilinear transformation.

- Translate of the specifications in ω to specifications in Ω via (9.46):

$$\begin{aligned} \Omega_p &= 2\tan\omega_p \\ \Omega_s &= 2\tan\omega_s \end{aligned}$$

- Set the value of the square magnitude equal to the tolerance values (the monotonicity of the magnitude guarantees that the tolerance is satisfied if the conditions are imposed at the band edges)

$$\begin{aligned} |H_c(j\Omega_p)|^2 &= \delta_p^2 \\ |H_c(j\Omega_s)|^2 &= \delta_s^2 \end{aligned}$$

- Solve the above system of equations for N (the filter order) and Ω_c
- Find the normalized filter coefficients for the order N from a table
- Scale the coefficients by Ω_c^k
- Build the transfer function $H(s)$
- Apply the bilinear transformation to obtain $H(z)$

Numerical examples can be found in the bibliography. As a last remark, note that the Matlab command `butter` can be used to design digital Butterworth filters.

9.5 Filter Structures

We have seen in Section 9.1.1 a practical implementation of a rational transfer function. That was just one particular way of translating equation (9.1) into a working structure; it served well as an illustration but the design choices one can make are many, and we will now approach the design problem from a more general point of view.

It is easy to see, from inspection, that the basic building block which enter the recipe for a real-world filter are:

1. An addition operator for sequence values,
   implementing $y[n] = x_1[n] + x_2[n]$ (Fig. 9.21(a)).
2. A scalar multiplication operator,
   implementing $y[n] = ax[n]$ (Fig. 9.21(b)).
3. A unit delay operator,
   implementing $y[n] = x[n-1]$ (Fig. 9.21(c)).

By properly combining these elements and by exploiting the different possible factorization of a filter's rational transfer function, we can arrive at a variety of different working implementations of a filter.

9.5.1 FIR Filter Structures

In the Z-transform representation of an FIR transfer function as in (9.6), all the denominator coefficients a_n are zero; we have therefore

$$H(z) = b_0 + b_1 z^{-1} + \ldots + b_{M-1} z^{-(M-1)}$$

where, of course, the coefficients correspond to the nonzero values of the impulse response $h[n]$, i.e. $b_n = h[n]$. Using the constitutive elements outlined above, we can immediately draw a block diagram of an FIR filter as in Figure 9.22. In practice, however, additions will be distributed as shown in Figure 9.23; this kind of implementation is called a *transversal filter*. If the filter taps are all real, we can also consider the factored form of $H(z)$ as

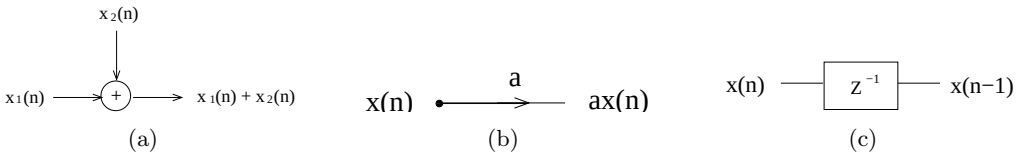


Figure 9.21: Constitutive elements for a filter: (a) Addition, (b) Multiplication, (c) Delay.

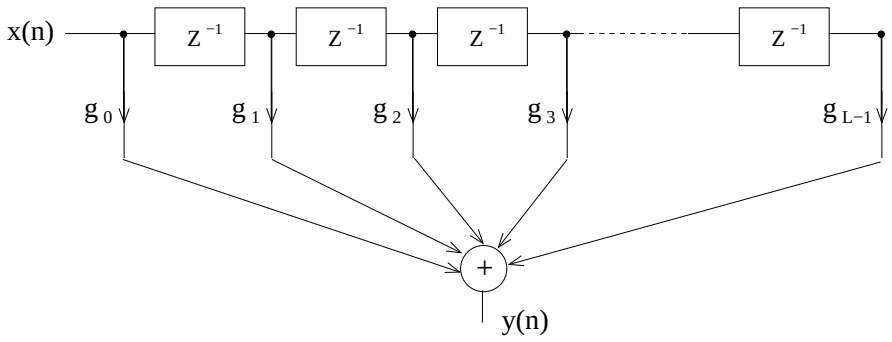


Figure 9.22: Direct FIR implementation.

in (9.9), i.e.

$$H(z) = b_0 \prod_{n=1}^{M_r} (1 - z_n z^{-1}) \prod_{n=1}^{M_c} (1 - \mathrm{Re}\{z_n\} z^{-1} + |z_n|^2 z^{-2})$$

where $M_r + 2M_c = M$. From this representation of the transfer function we can obtain an alternative structure for the FIR called *cascade*, which is shown in Figure 9.24. This cascade form is very important for IIR filters as well, as we will see later. Special optimizations of the FIR structures can be obtained in the case of symmetric and antisymmetric filters; these are considered in the exercises.

9.5.2 IIR filters structures

For IIR filter, both the a_n's and the b_n's in (9.6) are nonzero. One possible implementation based on the direct form of the transfer function is given in Figure 9.25. This implementation is called *Direct Form I* and it is immediate to see that the C-code implementation at the beginning of the chapter realizes a Direct Form I algorithm.

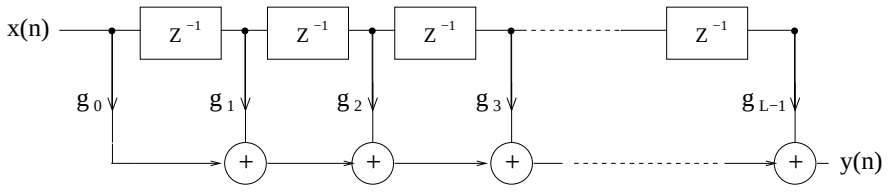


Figure 9.23: Transversal FIR implementation

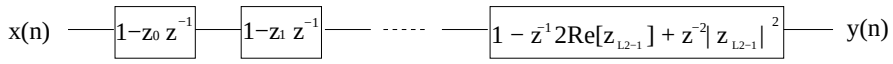


Figure 9.24: Cascade form of a filter.

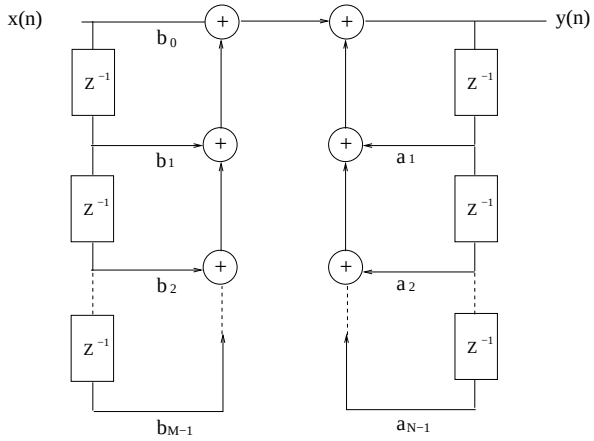


Figure 9.25: Direct Form implementation of an IIR filter.

By the commutative properties of the Z-transform, we can invert the order of computation to turn the Direct Form I structure into the structure shown in Figure 9.26(a); we can then combine the parallel delays together to obtain the structure in Figure 9.26(b). Here, for simplicity, we have assumed $N = M$ but obviously we can set some a_n's or b_n's to zero if this is not the case. This implementation is called *Direct Form II*; its obvious advantage is the reduced number of the required delay elements (hence of memory storage). A particular case, important for what follows, is the second order filter:

$$H(z) = \frac{1 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}}$$

which gives rise to the second order section displayed in Figure 9.27.

Again, for a real valued filter, we can consider the factored form of $H(z)$ as in (9.9). If we combine the complex conjugate poles and zeros, and group the real poles and zeros in twos, we can create a modular structure composed of second order sections. For instance, Figure 9.28 represents a 6^{th} order system.

There are still more possible implementations. For example, if we consider the partial fraction expansion of $H(z)$, we can rewrite the transfer function as the sum

$$H(z) = \sum_n D_n z^{-n} + \sum_n \frac{A_n}{1 - p_n z^{-1}} + \sum_n \frac{B_n + C_n z^{-1}}{(1 - p_n z^{-1})(1 - p_n^* z^{-1})}. \tag{9.47}$$

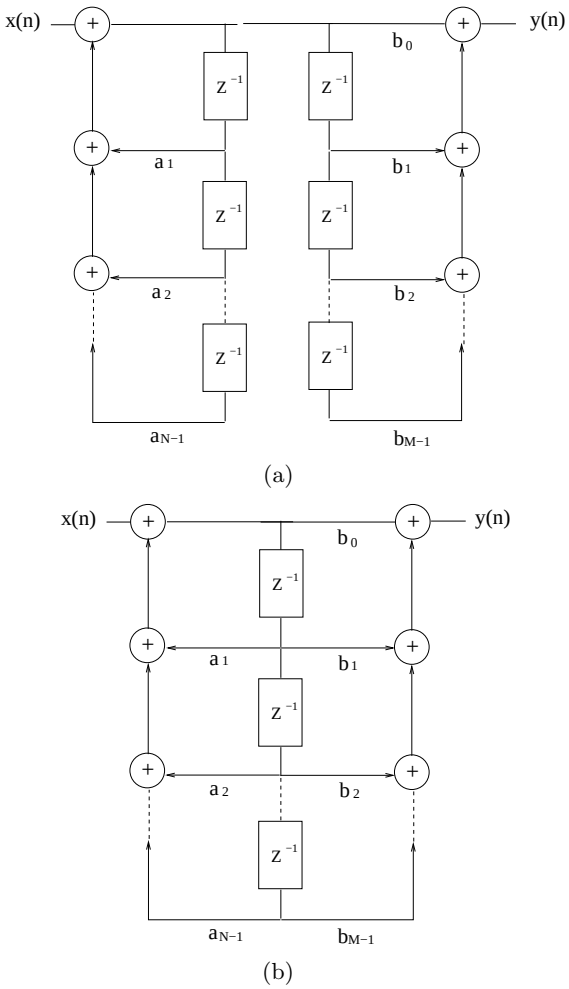


Figure 9.26: IIR filter structures: (a) Direct form I with inverted order. (b) Direct form II.

This generates a parallel structure of filters, whose outputs are summed together. The first branch corresponds to the first sum and it is an FIR filter; a further set of branches are associated to each term in the second sum, each one of them a first order IIR; the last set of branches is a collection of second order sections, one for each term of the third sum.

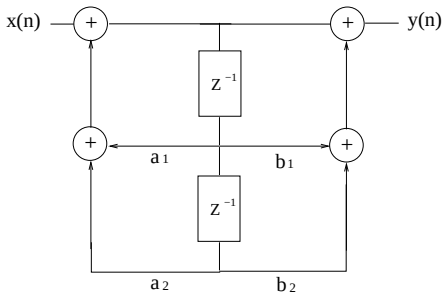


Figure 9.27: Direct Form II implementation for a 2^{nd} order filter with $b_0 = 1$

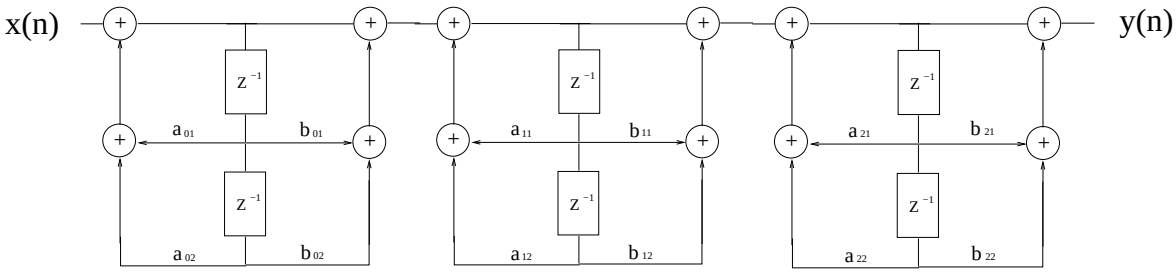


Figure 9.28: 6^{th} order filter implementation.

9.5.3 Some Remarks on Numerical Stability

A very important issue with digital filters is their numerical behavior for a given implementation. Two key questions are:

- Assume the computations are made with (basically) infinite precision but that the filter coefficients are represented internally with finite precision. How good is the resulting filter? Is it still stable?

- If computations are also made with finite precision arithmetic (which implies rounding and truncation error), what is the resulting numerical behavior of the system?

One important difference is that, in the first case, the system is at least guaranteed to be linear; in the second case, however, we can have non-linear effects such as overflows and limit cycles.

Precision and computational issues are very hard to analyze. Here, we will just note that the direct form implementation is more sensible to precision errors than the cascade form, which is why the cascade form is usually preferred in practice. Also, alternative

filter structures such as the *lattice* are designed to provide robustness for systems with low numerical precision, albeit at a higher computational cost.

9.6 Problems

Problem 9.1 *Consider the following set of complex numbers*

$$z_k = e^{j\pi(1-2^{-k})} \qquad k = 1, 2, \ldots, M$$

For $M = 4$,

1. *Plot z_k, $k = 1, 2, 3, 4$, on the complex plane.*
2. *Consider an FIR whose transfer function $H(z)$ has the following zeros:*

   $$\{z_1, z_2, z_1^*, z_2^*, -1\}$$

   and write out explicitly the expression for $H(z)$.
3. *How many nonzero taps will the impulse response $h[n]$ have at most?*
4. *Sketch the magnitude of $H(e^{j\omega})$.*
5. *What can you say about this filter:*
   (a) *What FIR type is it? (I, II, etc.)*
   (b) *Is it lowpass, bandpass, highpass?*
   (c) *Is it equiripple?*
   (d) *Is this a "good" filter? (By "good" we mean a filter which is close to 1 in the passband, close to zero in the stopband and which has a narrow transition band).*

Problem 9.2 *(Transfer functions, zeros and poles) Figure 9.29 shows the zeros and poles of three different filters with the unit circle for reference. Each zero is represented with an 'o' and each pole with an 'x' on the plot. Multiple zeros and poles are indicated by the multiplicity number shown to the upper right of the zero or pole.*

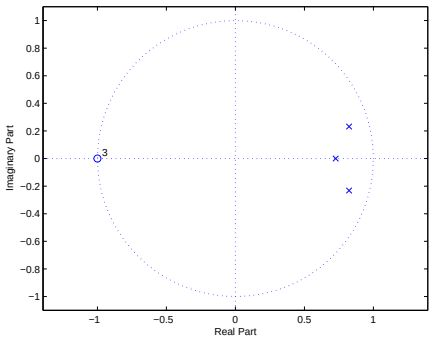
Sketch the magnitude of each frequency response and determine the type of filter.

Problem 9.3 *We consider a causal system with transfer function*

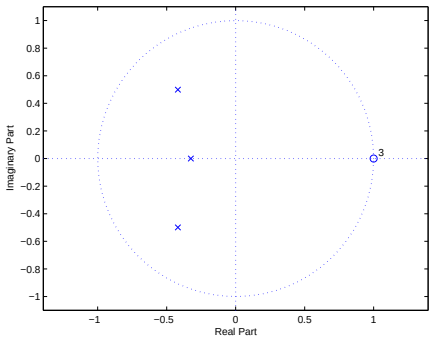
$$H(z) = \frac{1 - cz^{-1}}{1 - dz^{-1}}, \qquad |z| > \frac{1}{2}, \tag{9.48}$$

where $c = \frac{1}{2}e^{j(\phi+\pi)}$ and $d = \frac{1}{2}e^{j\phi}$. The variable ϕ is a parameter to the system. In this exercise we will analyze the behavior of this system as a function of ϕ.

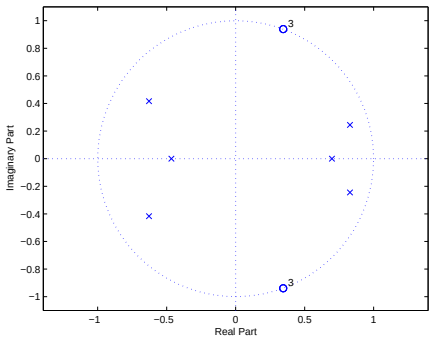
We start with an analysis in the Z-domain.



(a) Diagram 1



(b) Diagram 2



(c) Diagram 3

Figure 9.29: Zeros and Poles Diagrams

(a) Give the poles and zeros of $H(z)$ *as a function of* ϕ*. Give a pole-zero plot of* $H(z)$ *for* $\phi = 0$ *and* $\phi = \pi$*.*

In the Fourier domain we will consider the magnitude response only. In the analysis of LTI systems it is often convenient to consider the log-magnitude response $20\log_{10}\left|H(e^{j\omega})\right|$*.*

(b) Let $y[n]$ *be the output of the system at input* $x[n]$*. Express* $20\log_{10}\left|Y(e^{j\omega})\right|$ *in terms of* $20\log_{10}\left|X(e^{j\omega})\right|$ *and* $20\log_{10}\left|H(e^{j\omega})\right|$*.*

(c) Show that for a generic r *and* θ *the following holds:*

$$20\log_{10}\left|1 - re^{j\theta}\right| = 10\log_{10}\left(1 + r^2 - 2r\cos(\theta)\right). \tag{9.49}$$

(d) Derive the general expression for $20\log_{10}\left|H(e^{j\omega})\right|$ *for* $H(z)$ *given in (9.48). How does the system behave at* $\omega = \phi$ *and* $\omega = \phi + \pi$*?*

(e) Let $\phi = 0$*. In MATLAB, create a plot of* $20\log_{10}\left|H(e^{j\omega})\right|$*. Is this system all-pass, low-pass, high-pass or band-pass?*

(f) Let $\phi = \pi$*. In MATLAB, create a plot of* $20\log_{10}\left|H(e^{j\omega})\right|$*. Is this system all-pass, low-pass, high-pass or band-pass?*

Finally an analysis in the time-domain.

(g) For arbitrary ϕ*, determine the impulse repsonse of the system.*

(h) Let $\phi = 0$*. Based on the impulse response only, what can you say about the behaviour of the system at a unit step input? (You can think of the unit step function as a low frequency signal)*

(i) Let $\phi = \pi$*. Based on the impulse response only, what can you say about the behaviour of the system at a unit step input.*

Problem 9.4 *For this exercise, you first need to download the file* `santa_corrupt.wav` *from the course website. Load the file into matlab* $[data, fs] = wavread('santa\_corrupt.wav')$*. Listen to the file using* $soundsc(data, fs)$*.*

(a) Using matlab, provide a plot of the amplitude of the DFT of the sound sequence. After having listened to the sound sequence and looked at the frequency response, you should realize that the sequence was corrupted by high frequency noise!

(b) It is now up to you to design a 101 tap filter to de-noise the sequence. You are free to use either the windowing method or the Parks-McLellan algorithm. At the end, you should provide us a plot of the frequency response of the de-noised sequence, as well as a plot of the frequency and time response of the de-noising filter you designed. Further, provide a ***short*** *explanation of the procedure you followed. (Hint: do not try to recover the original sequence exactly, rather try to get rid of high frequencies).*