2

Signal Extraction Basics

2.1 Introduction

One of the most common tasks in signal processing is to extract a desired signal, say x_n, from an observed signal:

$$y_n = x_n + v_n \tag{2.1.1}$$

where v_n is an undesired component. The nature of v_n depends on the application. For example, it could be a white noise signal, which is typical of the background noise picked up during the measurement process, or it could be any other signal—not necessarily measurement noise—that must be separated from x_n.

The desired signal x_n often represents a smooth *trend* that conveys useful information about the underlying dynamics of the evolving time series. Trend extraction is carried out routinely on financial, business, census, climatic, and other applications.

An estimate, $\hat{x}_n$, of the desired signal x_n is obtained by processing the observed signal y_n through a processor designed according to some optimization criterion. There exist a large variety of signal extraction methods, most of them based on a least-squares minimization criterion, falling into two basic classes: (a) model-based parametric methods, such as those based on Wiener and Kalman filtering, and (b) non-parametric methods based on a variety of approaches, such as local polynomial modeling, exponential smoothing, splines, regularization filters, wavelets, and SVD-based methods. Some of the non-parametric methods (exponential smoothing, splines, regularization filters) can also be cast in a state-space Kalman filtering form.

We discuss the Wiener and Kalman approaches in chapters 11 and 13, and the SVD-based methods in chapter 15. In this chapter, we concentrate primarily on non-parametric methods.

We consider also the problem of "de-seasonalizing" a time series, that is, estimating and removing a periodic component. Many physical and financial time series have a natural periodicity built into them, such as daily, monthly, quarterly, yearly. The observed signal can be decomposed into three components: a periodic (or nearly periodic) seasonal part s_n, a smooth trend t_n, and a residual irregular part v_n that typically represents noise,

$$y_n = s_n + t_n + v_n \tag{2.1.2}$$

In such cases, the signal processing task is to determine both the trend and the seasonal components, t_n and s_n. Often, economic data are available only after they have been de-seasonalized, that is, after the seasonal part s_n has been removed. Further processing of the de-seasonalized trend, t_n, can provide additional information such as identifying business cycles. Moreover, modeling of the trend can be used for forecasting purposes.

The particular methods of smoothing, trend extraction, and seasonal decomposition that we consider in this and the next few chapters are:

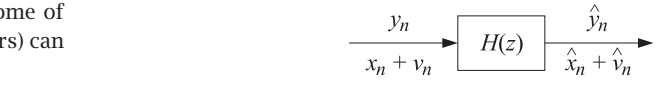
- local polynomial smoothing filters (Savitzky-Golay filters) — Chap. 3
- minimum-roughness filters (Henderson filters) — Chap. 4
- local polynomial modeling and LOESS — Chap. 5
- exponential smoothing — Chap. 6
- smoothing splines — Chap. 7
- regularization filters (Whittaker-Henderson, Hodrick-Prescott) — Chap. 8
- wavelet denoising — Chap. 10
- seasonal decomposition (classical, moving average, census X-11) — Chap. 9
- bandpass and other filters in business and finance — Chap. 8

2.2 Noise Reduction and Signal Enhancement

A standard method of extracting the desired signal x_n from y_n is to design an appropriate filter $H(z)$ that removes the noise component v_n and at the same time lets x_n go through unchanged. It is useful to view the design specifications and operation of such filter both in the time and frequency domains. Using linearity, we can express the output signal due to the input of Eq. (2.1.1) in the form:

$$\hat{y}_n = \hat{x}_n + \hat{v}_n \tag{2.2.1}$$

where $\hat{x}_n$ is the output due to x_n and $\hat{v}_n$ the output due to v_n. The two design conditions for the filter are that $\hat{x}_n$ be as similar to x_n as possible and that $\hat{v}_n$ be as small as possible; that is, ideally we require:†



$$\boxed{\begin{aligned}\hat{x}_n &= x_n \\ \hat{v}_n &= 0\end{aligned}} \tag{2.2.2}$$

In general, these conditions cannot be satisfied simultaneously. To determine when they can be satisfied, we may express them in the frequency domain in terms of the corresponding frequency spectra as follows: $\hat{X}(\omega) = X(\omega)$ and $\hat{V}(\omega) = 0$.

Applying the filtering equation $\hat{Y}(\omega) = H(\omega)Y(\omega)$ separately to the signal and noise components, we have the conditions:

$$\begin{aligned}\hat{X}(\omega) &= H(\omega)X(\omega) = X(\omega) \\ \hat{V}(\omega) &= H(\omega)V(\omega) = 0\end{aligned} \tag{2.2.3}$$

† An overall delay in the recovered signal is often acceptable, that is, $\hat{x}_n = x_{n-D}$.

The first requires that $H(\omega) = 1$ at all ω at which the signal spectrum is nonzero, $X(\omega) \neq 0$. The second requires that $H(\omega) = 0$ at all ω for which the noise spectrum is nonzero, $V(\omega) \neq 0$.

These two conditions can be met simultaneously only if the signal and noise spectra do not overlap, as shown in Fig. 2.2.1.[‡] In such cases, the filter $H(\omega)$ must have a *passband* that coincides with the signal band, and a *stopband* that coincides with the noise band. The filter removes the noise spectrum and leaves the signal spectrum unchanged.

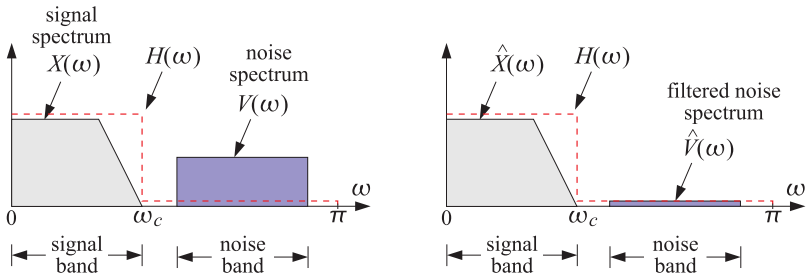


Fig. 2.2.1 Signal and noise spectra before and after filtering.

If the signal and noise spectra overlap, as is the typical case in practice, the above conditions cannot be satisfied simultaneously. In such cases, we must compromise between the two design conditions and trade off one for the other. Depending on the application, we may decide to design the filter to remove as much noise as possible, but at the expense of distorting the desired signal. Alternatively, we may decide to leave the desired signal as undistorted as possible, but at the expense of having some noise in the output.

The latter alternative is depicted in Fig. 2.2.2 where a low-frequency signal x_n exists in the presence of a broadband noise component, such as white noise, having a flat spectrum extending over the entire[1] Nyquist interval, $-\pi \leq \omega \leq \pi$.

The filter $H(\omega)$ is chosen to be an ideal lowpass filter with passband covering the signal bandwidth, say $0 \leq \omega \leq \omega_c$. The noise energy in the filter's stopband $\omega_c \leq \omega \leq \pi$ is removed completely by the filter, thus reducing the strength (i.e., the rms value) of the noise. The spectrum of the desired signal is not affected by the filter, but neither is the portion of the noise spectrum that falls within the signal band. Thus, some noise will survive the filtering process.

A measure of the amount of noise reduction achieved by a filter is given by the *noise gain*, or *noise reduction ratio* (NRR) of the filter, defined in Eq. (1.12.16), which is valid for white noise input signals. Denoting the input and output mean-square noise values by $\sigma^2 = E[v_n^2]$ and $\hat{\sigma}^2 = E[\hat{v}_n^2]$, we have:

$$\mathcal{R} = \frac{\hat{\sigma}^2}{\sigma^2} = \frac{1}{2\pi}\int_{-\pi}^{\pi} |H(\omega)|^2 \, d\omega = \sum_n h_n^2 \tag{2.2.4}$$

[‡]Here, ω is in units of radians per sample, i.e., $\omega = 2\pi f/f_s$, with f in Hz, and f_s is the sampling rate.

[1]For discrete-time signals, the spectra are periodic in ω with period 2π, or in f with period f_s.

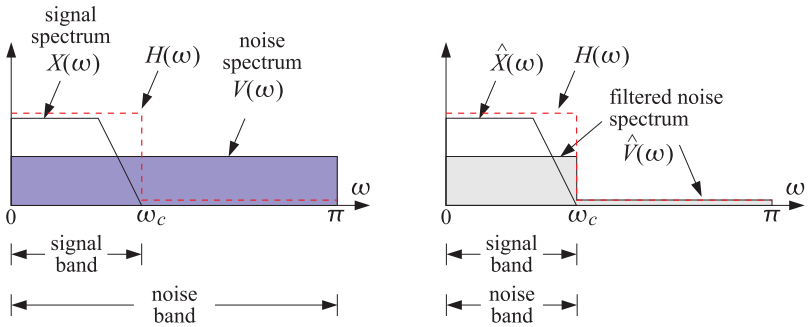


Fig. 2.2.2 Signal enhancement filter with partial noise reduction.

For the case of an ideal lowpass filter, with frequency and impulse responses given by [29],

$$H(\omega) = \begin{cases} 1, & \text{if } |\omega| \leq \omega_c \\ 0, & \text{if } \omega_c \leq |\omega| \leq \pi \end{cases} \qquad \text{and} \qquad h_n = \frac{\sin(\omega_c n)}{\pi n}, \quad -\infty < n < \infty \tag{2.2.5}$$

the integration range in Eq. (2.2.4) collapses to the filter's passband, that is, $-\omega_c \leq \omega \leq \omega_c$, and over this range the value of $H(\omega)$ is unity, giving:

$$\mathcal{R} = \frac{\hat{\sigma}^2}{\sigma^2} = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} 1 \cdot d\omega = \frac{2\omega_c}{2\pi} = \frac{\omega_c}{\pi} \tag{2.2.6}$$

Thus, the NRR is the proportion of the signal bandwidth with respect to the Nyquist interval. The same conclusion also holds when the desired signal is a high-frequency or a mid-frequency signal. For example, if the signal spectrum extends only over the mid-frequency band $\omega_a \leq |\omega| \leq \omega_b$, then $H(\omega)$ can be designed to be unity over this band and zero otherwise. A similar calculation yields in this case:

$$\mathcal{R} = \frac{\hat{\sigma}^2}{\sigma^2} = \frac{\omega_b - \omega_a}{\pi} \tag{2.2.7}$$

The noise reduction/signal enhancement capability of a filter can also be expressed in terms of the signal-to-noise ratio. The SNRs at the input and output of the filter are defined in terms of the mean-square values as:

$$SNR_{\text{in}} = \frac{E[x_n^2]}{E[v_n^2]}, \qquad SNR_{\text{out}} = \frac{E[\hat{x}_n^2]}{E[\hat{v}_n^2]}$$

Therefore, the relative *improvement* in the SNR introduced by the filter will be:

$$\frac{SNR_{\text{out}}}{SNR_{\text{in}}} = \frac{E[\hat{x}_n^2]}{E[\hat{v}_n^2]} \cdot \frac{E[v_n^2]}{E[x_n^2]} = \frac{1}{\mathcal{R}} \cdot \frac{E[\hat{x}_n^2]}{E[x_n^2]}$$

If the desired signal is not changed by the filter, $\hat{x}_n = x_n$, then

$$\frac{SNR_{\text{out}}}{SNR_{\text{in}}} = \frac{1}{\mathcal{R}} \tag{2.2.8}$$

Thus, minimizing the noise reduction ratio is equivalent to maximizing the signal-to-noise ratio at the filter's output.

The NRRs computed in Eqs. (2.2.6) or (2.2.7) give the *maximum* noise reductions achievable with *ideal* lowpass or bandpass filters that do not distort the desired signal. Such ideal filters are not realizable because they have double-sided impulse responses with infinite anticausal tails. Thus, in practice, we must use *realizable* approximations to the ideal filters, such as FIR filters, or causal IIR filters. The realizable filters may meet the two design goals approximately, for example, by minimizing the NRR subject to certain constraints that help sustain the signal passband. Examples of this approach are discussed in Sections 2.3, 2.4, and generalized in Sections 3.1 and 4.2.

The use of realizable filters introduces two further design issues that must be dealt with in practice: One is the *transient response* of the filter and the other, the amount of *delay* introduced into the output. The more closely a filter approximates the sharp transition characteristics of an ideal response, the closer to the unit circle its poles get, and the longer its transient response becomes. Stated differently, maximum noise reduction, approaching the ideal limit (2.2.6), can be achieved only at the expense of introducing long transients in the output.

The issue of the delay introduced into the output has to do with the steady-state response of the filter. After steady-state has set in, different frequency components of an input signal suffer different amounts of delay, as determined by the phase delay $d(\omega) = -\arg H(\omega)/\omega$ of the filter [29].

In particular, if the filter has *linear phase*, then it causes an overall delay in the output. Indeed, assuming that the filter has nearly unity magnitude, $|H(\omega)| \simeq 1$, over its passband (i.e., the signal band) and is zero over the stopband, and assuming a constant phase delay $d(\omega) = D$, we have for the frequency response

$$H(\omega) = |H(\omega)|e^{-j\omega d(\omega)} \simeq e^{-j\omega D}$$

over the passband, and we find for the filtered version of the desired signal:

$$\hat{x}_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} \hat{X}(\omega)e^{j\omega n}\,d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\omega)X(\omega)e^{j\omega n}\,d\omega$$
$$= \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} X(\omega)e^{j\omega(n-D)}\,d\omega = x(n-D)$$

the last equation following from the inverse DTFT of the desired signal:

$$x_n = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} X(\omega)e^{j\omega n}\,d\omega$$

Many smoothing filters used in practice (e.g., see Chapters 3 and 4) are double-sided filters, h_n, $-M \le n \le M$, with a symmetric impulse response, $h_n = h_{-n}$, and therefore, they introduce no delay in the output ($D = 0$). On the other hand, if such filters are made causal by a delay ($D = M$), then they will introduce a delay in the output. Such delays are of concern in some applications such as monitoring and filtering real-time data in the financial markets.

Next, we consider some noise reduction examples based on simple filters, calculate the corresponding noise reduction ratios, discuss the tradeoff between transient response and noise reduction, and present some simulation examples.

2.3 First-Order Exponential Smoother

It is desired to extract a constant signal $x_n = s$ from the noisy measured signal

$$y_n = x_n + v_n = s + v_n$$

where v_n is zero-mean white Gaussian noise of variance σ_v^2. To this end, the following IIR lowpass filter may be used, where $b = 1 - a$,

$$H(z) = \frac{b}{1 - az^{-1}}, \quad H(\omega) = \frac{b}{1 - ae^{-j\omega}}, \quad |H(\omega)|^2 = \frac{b^2}{1 - 2a\cos\omega + a^2} \tag{2.3.1}$$

where the parameter a is restricted to the range $0 < a < 1$. Because the desired signal x_n is constant in time, the signal band will be just the DC frequency $\omega = 0$. We require therefore that the filter have unity gain at DC. This is guaranteed by the above choice of the parameter b, that is, we have at $\omega = 0$, or equivalently at $z = 1$,

$$H(z)\,\big|_{z=1} = \frac{b}{1-a} = 1$$

The NRR can be calculated from Eq. (2.2.4) by summing the impulse response squared. Here, $h_n = ba^n u_n$, therefore, using the geometric series, we find

$$\mathcal{R} = \frac{\hat{\sigma}^2}{\sigma^2} = \sum_n h_n^2 = b^2 \sum_{n=0}^{\infty} a^{2n} = \frac{b^2}{1-a^2} = \frac{(1-a)^2}{1-a^2} = \frac{1-a}{1+a} \tag{2.3.2}$$

The filter's magnitude response, pole-zero pattern, and the corresponding input and output noise spectra are shown in Fig. 2.3.1. The shaded area under the $|H(\omega)|^2$ curve (including its negative-frequency portion) is equal as the NRR computed above.

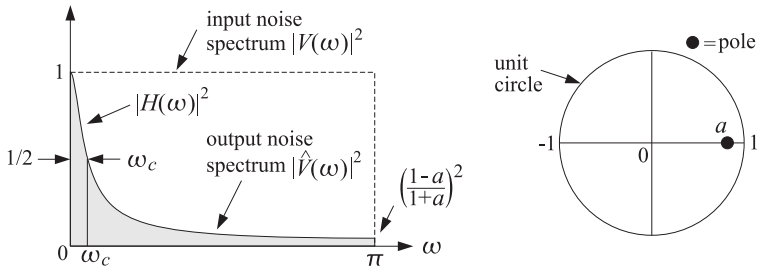


Fig. 2.3.1 Lowpass exponential smoothing filter.

The NRR is always less than unity because a is restricted to $0 < a < 1$. To achieve high noise reduction, a must be chosen near one. But, then the filter's effective time constant will become large:†

$$n_{\text{eff}} = \frac{\ln\epsilon}{\ln a} \to \infty \qquad \text{as} \qquad a \to 1$$

†The values $\epsilon = 0.01$ and $\epsilon = 0.001$ correspond to the so-called 40-dB and 60-dB time constants [30].

The filter's 3-dB cutoff frequency ω_c can be calculated by requiring that $|H(\omega_c)|^2$ drops by $1/2$, that is,

$$|H(\omega_c)|^2 = \frac{b^2}{1 - 2a\cos\omega_c + a^2} = \frac{1}{2}$$

which can be solved to give $\cos\omega_c = 1 - (1-a)^2/2a$. If a is near one, $a \lesssim 1$, we can use the approximation $\cos x \simeq 1 - x^2/2$ and solve for ω_c approximately:[†]

$$\omega_c \simeq 1 - a$$

This shows that as $a \to 1$, the filter becomes a narrower lowpass filter, removing more noise from the input, but at the expense of increasing the time constant.

The tradeoff between noise reduction and speed of response is illustrated in Fig. 2.3.2, where 200 samples of a simulated noisy signal y_n were filtered using the difference equation of the filter, that is, replacing $b = 1 - a$

$$y_n = s + v_n\,, \quad \hat{x}_n = a\hat{x}_{n-1} + (1-a)y_n \tag{2.3.3}$$

and initialized at $\hat{x}_{-1} = 0$. The value of the constant was $s = 5$, and the input noise variance, $\sigma_v^2 = 1$. The random signal v_n was generated by the built-in MATLAB function `randn`. The figure on the left corresponds to $a = 0.90$, which has a 40-dB time constant, NRR, and SNR improvement in dB:

$$n_{\text{eff}} = \frac{\ln(0.01)}{\ln(0.90)} = 44\,, \qquad \mathcal{R} = \frac{1-0.90}{1+0.90} = \frac{1}{19}\,, \qquad 10\log_{10}\left(\frac{1}{\mathcal{R}}\right) = 12.8\text{ dB}$$

The right figure has $a = 0.98$, with a longer time constant of $n_{\text{eff}} = 228$, a smaller $\mathcal{R} = 1/99$, and bigger SNR improvement, $10\log_{10}(1/\mathcal{R}) = 20$ dB.

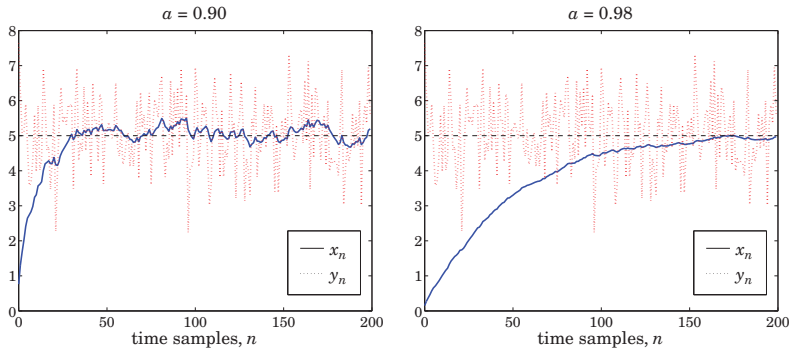


Fig. 2.3.2 Noisy input and smoothed output.

[†]The full 3-dB width of the interval $[-\omega_c, \omega_c]$ is $2\omega_c = 2(1-a)$. This is a special case of a more general result [30] that the 3-dB width due to a filter pole with radius r near the unit circle, $r \lesssim 1$, is given by $\Delta\omega = 2(1-r)$.

To understand how this filter works in the time domain and manages to reduce the noise, we rewrite the difference equation (2.3.3) in its convolutional form:

$$\hat{x}_n = b\sum_{m=0}^{n} a^m y_{n-m} = b\left(y_n + ay_{n-1} + a^2 y_{n-2} + \cdots + a^n y_0\right)$$

The sum represents a weighted average of all the past samples up to the present time instant. As a result, the rapid fluctuations of the noise component v_n are averaged out. The closer a is to 1, the more equal weighting the terms get, and the more effective the averaging of the noise. The exponential weighting de-emphasizes the older samples and causes the sum to behave as though it had effectively a finite number of terms, thus, safeguarding the mean-square value of $\hat{x}_n$ from diverging (see, for example, Sec. 1.15.) Because of the exponential weighting, this filter is also called an *exponential smoother*.

This filter can be applied to the smoothing of *any* low-frequency signal, not just constants. One must make sure that the bandwidth of the desired signal x_n is *narrower* than the filter's lowpass width ω_c, so that the filter will not remove any of the higher frequencies present in x_n.

The exponential smoother is a standard tool in many applications requiring the smoothing of data in signal processing, statistics, economics, physics, and chemistry. It is also widely used in forecasting applications, for example in inventory control, where the quantity $\hat{x}_n$ is interpreted as the one-step ahead forecast. More precisely, the *forecasting filter* and its I/O difference equation are given by:

$$H_f(z) = z^{-1}H(z) = \frac{bz^{-1}}{1 - az^{-1}}\,, \quad F_{n+1} = aF_n + (1-a)y_n \tag{2.3.4}$$

where F_{n+1} is the predicted value of x_{n+1} based on the available data y_n up to time n.

We discuss the exponential smoother further in Sec. 6.1, where we rederive it from an optimization criterion and generalize it to higher orders.

A slight variation of Eq. (2.3.1) which improves the NRR without affecting the speed of response can be derived by adding a zero in the transfer function at $z = -1$ or equivalently, at $\omega = \pi$. The resulting first-order filter will be:

$$H(z) = \frac{b(1+z^{-1})}{1 - az^{-1}} \qquad \Rightarrow \qquad |H(\omega)|^2 = \frac{2b^2(1+\cos\omega)}{1 - 2a\cos\omega + a^2} \tag{2.3.5}$$

where b is fixed by requiring unity gain at DC:

$$H(z)\big|_{z=1} = \frac{2b}{1-a} = 1 \qquad \Rightarrow \qquad b = \frac{1-a}{2}$$

The zero at $\omega = \pi$ suppresses the high-frequency portion of the input noise spectrum even more than the filter of Eq. (2.3.1), thus, resulting in smaller NRR for the same value of a. The impulse response of this filter can be computed using partial fractions:

$$H(z) = \frac{b(1+z^{-1})}{1 - az^{-1}} = A_0 + \frac{A_1}{1 - az^{-1}}\,, \quad \text{where} \quad A_0 = -\frac{b}{a}, \qquad A_1 = \frac{b(1+a)}{a}$$

Therefore, its (causal) impulse response will be:

$$h_n = A_0\delta(n) + A_1 a^n u(n)$$

Note, in particular, that $h_0 = A_0 + A_1 = b$. It follows that

$$\mathcal{R} = \sum_{n=0}^{\infty} h_n^2 = h_0^2 + \sum_{n=1}^{\infty} h_n^2 = b^2 + A_1^2 \frac{a^2}{1-a^2} = \frac{1-a}{2}$$

This is slightly smaller than that of Eq. (2.3.2), because of the inequality:

$$\frac{1-a}{2} < \frac{1-a}{1+a}$$

The 3-dB cutoff frequency can be calculated easily in this example. We have

$$|H(\omega_c)|^2 = \frac{2b^2(1+\cos\omega_c)}{1-2a\cos\omega_c + a^2} = \frac{1}{2}$$

which can be solved for ω_c in terms of a:

$$\cos\omega_c = \frac{2a}{1+a^2} \quad \Leftrightarrow \quad \tan\left(\frac{\omega_c}{2}\right) = \frac{1-a}{1+a} \tag{2.3.6}$$

Conversely, we can solve for a in terms of ω_c:

$$a = \frac{1-\sin\omega_c}{\cos\omega_c} = \frac{1-\tan(\omega_c/2)}{1+\tan(\omega_c/2)} \tag{2.3.7}$$

It is easily checked that the condition $0 < a < 1$ requires that $\omega_c < \pi/2$. We note also that the substitution $z \to -z$ changes the filter into a highpass one.

Such simple first-order lowpass or highpass filters with easily controllable widths are useful in many applications, such as the low- and high-frequency shelving filters of audio equalizers [30].

2.4 FIR Averaging Filters

The problem of extracting a constant or a low-frequency signal x_n from the noisy signal $y_n = x_n + v_n$ can also be approached with FIR filters. Consider, for example, the third-order filter:

$$H(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3}$$

The condition that the constant signal x_n go through the filter unchanged is the condition that the filter have unity gain at DC, which gives the constraint among the filter weights:

$$H(z)\big|_{z=1} = h_0 + h_1 + h_2 + h_3 = 1 \tag{2.4.1}$$

The NRR of this filter will be simply:

$$\mathcal{R} = \sum_n h_n^2 = h_0^2 + h_1^2 + h_2^2 + h_3^2 \tag{2.4.2}$$

The *optimum* third-order FIR filter will be the one that *minimizes* this NRR, subject to the lowpass constraint (2.4.1). To solve this minimization problem, we introduce a Lagrange multiplier λ and incorporate the constraint (2.4.1) into the performance index:

$$\mathcal{J} = \mathcal{R} + \lambda\left(1 - \sum_{n=0}^{3} h_n\right) = \sum_{n=0}^{3} h_n^2 + \lambda\left(1 - \sum_{n=0}^{3} h_n\right) \tag{2.4.3}$$

The minimization can be carried out easily by setting the partial derivatives of $\mathcal{J}$ to zero and solving for the h's:

$$\frac{\partial\mathcal{J}}{\partial h_n} = 2h_n - \lambda = 0 \quad \Rightarrow \quad h_n = \frac{\lambda}{2}, \quad n = 0,1,2,3$$

Thus, all four h's are equal, $h_0 = h_1 = h_2 = h_3 = \lambda/2$. The constraint (2.4.1) then fixes the value of λ to be $1/2$ and we find the optimum weights:

$$h_0 = h_1 = h_2 = h_3 = \frac{1}{4}$$

and the minimized NRR becomes:

$$\mathcal{R}_{\min} = \left(\frac{1}{4}\right)^2 + \left(\frac{1}{4}\right)^2 + \left(\frac{1}{4}\right)^2 + \left(\frac{1}{4}\right)^2 = 4\left(\frac{1}{4}\right)^2 = \frac{1}{4}$$

The I/O equation for this optimum smoothing filter becomes:

$$\hat{x}_n = \frac{1}{4}(y_n + y_{n-1} + y_{n-3} + y_{n-3})$$

More generally, the optimum length-N FIR filter with unity DC gain and minimum NRR is the filter with equal weights:

$$H(z) = \frac{1}{N}\left[1 + z^{-1} + z^{-2} + \cdots + z^{-(N-1)}\right] \tag{2.4.4}$$

and I/O equation:

$$\hat{x}_n = \frac{1}{N}(y_n + y_{n-1} + \cdots + y_{n-N+1}) \tag{2.4.5}$$

with minimized NRR:

$$\mathcal{R} = h_0^2 + h_1^2 + \cdots + h_{N-1}^2 = N \cdot \left(\frac{1}{N}\right)^2 = \frac{1}{N} \tag{2.4.6}$$

Thus, by choosing N large enough, the NRR can be made as small as desired. Again, as the NRR decreases, the filter's time constant ($n_{\text{eff}} = N$) increases.

How does the FIR smoother compare with the IIR smoother of Eq. (2.3.1)? First, we note the IIR smoother is very simple computationally, requiring only 2 MACs[†] per output sample, whereas the FIR requires N MACs.

Second, the FIR smoother typically performs better in terms of both the NRR and the transient response, in the sense that for the same NRR value, the FIR smoother has shorter time constant, and for the same time constant, it has a smaller NRR. We illustrate these remarks below.

Given a time constant $n_{\text{eff}} = \ln\epsilon/\ln a$ for an IIR smoother, the "equivalent" FIR smoother should be chosen to have the same length $N = n_{\text{eff}}$, thus,

$$\boxed{N = \frac{\ln\epsilon}{\ln a}, \quad a = \epsilon^{1/N}} \tag{2.4.7}$$

[†] multiplication-accumulations

For example, if $a = 0.90$ and $\epsilon = 0.01$, then $N = n_{\text{eff}} = 44$. But then, the NRR of the FIR smoother will be $\mathcal{R} = 1/N = 1/44$, which is better than that of the IIR filter, $\mathcal{R} = (1-a)/(1+a) = 1/19$. This case is illustrated in the left graph of Fig. 2.4.1, where the FIR output was computed by Eq. (2.4.5) with $N = 44$ for the same noisy input of Fig. 2.3.2. The IIR output is the same as in that figure.

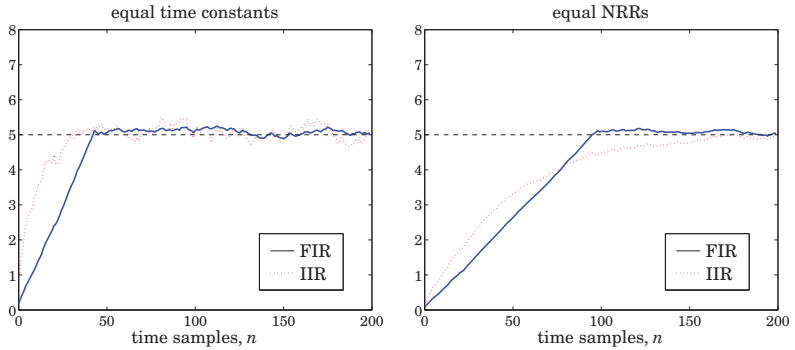


Fig. 2.4.1 Comparison of FIR and IIR smoothing filters.

Similarly, if an IIR smoother achieves a certain NRR value, the "equivalent" FIR filter with the same NRR should have length N such that:

$$\mathcal{R} = \frac{1-a}{1+a} = \frac{1}{N} \quad \Rightarrow \quad \boxed{N = \frac{1+a}{1-a}, \quad a = \frac{N-1}{N+1}} \tag{2.4.8}$$

For example, if $a = 0.98$, then we get $N = 99$, which is much shorter than the IIR time constant $n_{\text{eff}} = 228$ computed with $\epsilon = 0.01$. The right graph of Fig. 2.4.1 illustrates this case, where the FIR output was computed by Eq. (2.4.5) with $N = 99$.

An approximate relationship between the IIR time constant n_{eff} and N can be derived in this case as follows. Using the small-x approximation $\ln\big((1+x)/(1-x)\big) \simeq 2x$, we have for large N:

$$\ln(1/a) = \ln\left(\frac{1+(1/N)}{1-(1/N)}\right) \simeq \frac{2}{N}$$

It follows that

$$n_{\text{eff}} = \frac{\ln(1/\epsilon)}{\ln(1/a)} \simeq N\,\frac{1}{2}\ln\left(\frac{1}{\epsilon}\right)$$

Typically, the factor $(\ln(1/\epsilon)/2)$ is greater than one, resulting in a longer IIR time constant n_{eff} than N. For example, we have:

$$\begin{aligned}
n_{\text{eff}} &= 1.15\,N, \quad \text{if } \epsilon = 10^{-1} && \text{(10\% time constant)}\\
n_{\text{eff}} &= 1.50\,N, \quad \text{if } \epsilon = 5\cdot 10^{-2} && \text{(5\% time constant)}\\
n_{\text{eff}} &= 2.30\,N, \quad \text{if } \epsilon = 10^{-2} && \text{(1\% or 40-dB time constant)}\\
n_{\text{eff}} &= 3.45\,N, \quad \text{if } \epsilon = 10^{-3} && \text{(0.1\% or 60-dB time constant)}
\end{aligned}$$

Finally, we note that a further advantage of the FIR smoother is that it is a *linear phase* filter. Indeed, using the finite geometric series formula, we can write the transfer

function of Eq. (2.4.5) in the form:

$$H(z) = \frac{1}{N}\big(1 + z^{-1} + z^{-2} + \cdots + z^{-(N-1)}\big) = \frac{1}{N}\,\frac{1-z^{-N}}{1-z^{-1}} \tag{2.4.9}$$

Setting $z = e^{j\omega}$, we obtain the frequency response:

$$\boxed{H(\omega) = \frac{1}{N}\,\frac{1-e^{-jN\omega}}{1-e^{-j\omega}} = \frac{1}{N}\,\frac{\sin(N\omega/2)}{\sin(\omega/2)}\,e^{-j\omega(N-1)/2}} \tag{2.4.10}$$

which has a linear phase response. The transfer function (2.4.9) has zeros at the Nth roots of unity, except at $z = 1$, that is,

$$z_k = e^{j\omega_k}, \quad \omega_k = \frac{2\pi k}{N}, \quad k = 1, 2, \ldots, N-1$$

The zeros are distributed equally around the unit circle and tend to suppress the noise spectrum along the Nyquist interval, except at $z = 1$ where there is a pole/zero cancellation and we have $H(z) = 1$.

Fig. 2.4.2 shows the magnitude and phase response of $H(\omega)$ for $N = 16$. Note that the phase response is *piece-wise linear* with slope $(N-1)/2$. It exhibits 180^{o} jumps at $\omega = \omega_k$, where the factor $\sin(N\omega/2)/\sin(\omega/2)$ changes algebraic sign.

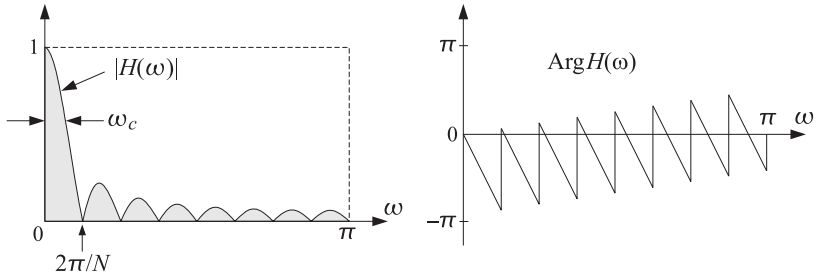


Fig. 2.4.2 Magnitude and phase responses of FIR smoother, for $N = 16$.

The 3-dB cutoff frequency of the filter is somewhat less than half the base of the mainlobe, that is,

$$\boxed{\omega_c = \frac{0.886\pi}{N}} \tag{2.4.11}$$

It corresponds to a drop of the magnitude response squared by a factor of $1/2$. Indeed, setting $\omega/2 = \omega_c/2 = 0.443\pi/N$ in (2.4.10), we have

$$\left|\frac{1}{N}\,\frac{\sin(N\,0.443\pi/N)}{\sin(0.443\pi/N)}\right|^2 \simeq \left|\frac{1}{N}\,\frac{\sin(0.443\pi)}{(0.443\pi/N)}\right|^2 = \left|\frac{\sin(0.443\pi)}{0.443\pi}\right|^2 \simeq \frac{1}{2}$$

where we used the approximation $\sin(\pi/2N) \simeq \pi/2N$, for large N. In decibels, we have $-20\log_{10}\big(\sin(0.443\pi)/0.443\pi\big) = 3.01$ dB, hence, the name "3-dB frequency."

Like its IIR counterpart of Eq. (2.3.1), the FIR averaging filter (2.4.5) can be applied to any low-frequency signal x_n—not just a constant signal. The averaging of the N successive samples in Eq. (2.4.5) tends to smooth out the highly fluctuating noise component v_n, while it leaves the slowly varying component x_n almost unchanged.

However, if x_n is not so slowly varying, the filter will also tend to average out these variations, especially when the averaging operation (2.4.5) reaches across many time samples when N is large. In the frequency domain, the same conclusion follows by noting that as N increases, the filter's cutoff frequency ω_c decreases, thus removing more and more of the higher frequencies that might be present in the desired signal.

Thus, there is a limit to the applicability of this type of smoothing filter: Its length must be chosen to be large enough to reduce the noise, but not so large as to start distorting the desired signal by smoothing it too much.

A rough quantitative criterion for the selection of the length N is as follows. If it is known that the desired signal x_n contains significant frequencies up to a maximum frequency, say $\omega_{\max}$, then we may choose N such that $\omega_{\max} \leq \omega_c = 0.886\pi/N$, which gives $N \leq 0.886\pi/\omega_{\max}$.

The FIR averaging filter can also be implemented in a *recursive form* based on the summed version of the transfer function (2.4.9). For example, the direct-form realization of $H(z)$ is described by the I/O difference equation:

$$\hat{x}_n = \hat{x}_{n-1} + \frac{1}{N}[y_n - y_{n-N}] \tag{2.4.12}$$

Because of the pole-zero cancellation implicit in (2.4.12) such implementation is prone to roundoff accumulation errors and instabilities, and therefore, not recommended for continuous real-time processing even though it is efficient computationally.

The FIR smoothing filter will be considered in further detail in Sec. 3.1, generalized to local polynomial smoothing filters that minimize the NRR subject to additional linear constraints on the filter weights. In Sec. 4.2, it is generalized to minimum-roughness filters that minimize a filtered version of the NRR subject to similar constraints.

Like the IIR smoother, the FIR smoother and its generalizations are widely used in many data analysis applications. It is also useful in de-seasonalizing applications, where if N is chosen to be the seasonal period, the filter's Nth root of unity zeros coincide with the harmonics of the seasonal component so that the filter will extract the smooth trend while eliminating the seasonal part.

2.5 Problems

2.1 Show that the z-domain transformation, $z \to -z$, maps a lowpass filter into a highpass one. Show that under this transformation, the impulse response of the lowpass filter h_n gets mapped into $(-1)^n h_n$.

2.2 Given the real-valued impulse response h_n of a lowpass filter, show that the filter with the complex-valued impulse response $e^{j\omega_0 n}h_n$ defines a bandpass filter centered at ω_0. What sort of filter is defined by the real-valued impulse response $\cos(\omega_0 n)h_n$? Explain how the previous problem is a special case of this problem.

2.3 *Highpass Signal Extraction.* Design a first-order IIR filter to extract the high-frequency $x_n = (-1)^n s$ from the noisy signal

$$y_n = x_n + v_n = (-1)^n s + v_n$$

where s is a constant amplitude and v_n is zero-mean, white Gaussian noise with variance σ_v^2. Start by converting the two lowpass filters given in Sec. 2.3 into highpass filters. For each of the resulting filters, plot the corresponding magnitude response and calculate the NRR in terms of the pole parameter a.

For the values of the parameters $s = 2$ and $a = 0.99$, compute 200 samples of the signal y_n and process it through your filters and plot the output. Discuss the transient effect vs. the signal extraction ability of the filters.

2.4 *Bandpass Signal Extraction.* A noisy sinusoid of frequency $f_0 = 500$ Hz is sampled at a rate of $f_s = 10$ kHz:

$$y_n = x_n + v_n = \cos(\omega_0 n) + v_n$$

where $\omega_0 = 2\pi f_0/f_s$ and v_n is a zero-mean, unit-variance, white Gaussian noise signal. The sinusoid can be extracted by a bandpass resonator-like filter of the form:

$$H(z) = \frac{G}{(1 - Re^{j\omega_0}z^{-1})(1 - Re^{-j\omega_0}z^{-1})} = \frac{G}{1 - 2R\cos\omega_0\, z^{-1} + R^2 z^{-2}}$$

Its poles are at $z = Re^{\pm j\omega_0}$ with $0 < R < 1$. For R near unity, the 3-dB width of this filter is given approximately by $\Delta\omega = 2(1 - R)$.

Fix the gain factor G by requiring that the filter have unity gain at ω_0, that is, $|H(\omega_0)| = 1$. Then, show that the NRR of this filter is given by:

$$\mathcal{R} = \sum_{n=0}^{\infty} h_n^2 = \frac{(1 - R)(1 + R^2)(1 - 2R\cos(2\omega_0) + R^2)}{(1 + R)(1 - 2R^2\cos(2\omega_0) + R^4)}$$

For the values of the parameters $R = 0.99$ and $\omega_0 = 0.1\pi$, plot the magnitude response of this filter and indicate on the graph its 3-dB width. Calculate the corresponding NRR.

Then, calculate and plot 300 samples of the noisy signal y_n, and process it through the filter. On a separate graph, plot the resulting estimate $\hat{x}_n$ together with the desired signal x_n.

Discuss the signal extraction capability of this filter vs. the transient effects vs. the delay shift introduced by the filter's phase delay $d(\omega) = -\text{Arg}\, H(\omega)/\omega$, and calculate the amount of delay $d(\omega_0)$ at ω_0 and indicate it on the graph.