4

Minimum Roughness Filters

4.1 Weighted Local Polynomial Filters

The design of the LPSM filters was based on a least-squares criterion, such as (3.2.2), where all error terms were equally weighted within the filter's window:

$$\mathcal{J} = \sum_{m=-M}^{M} e_m^2 = \sum_{m=-M}^{M} (y_m - \hat{y}_m)^2 = \sum_{m=-M}^{M} \left(y_m - \sum_{i=0}^{d} c_i m^i \right)^2 = \min$$

This can be generalized by using unequal positive weights, w_m, $-M \le m \le M$:

$$\mathcal{J} = \sum_{m=-M}^{M} w_m e_m^2 = \sum_{m=-M}^{M} w_m \left(y_m - \sum_{i=0}^{d} c_i m^i \right)^2 = \min \tag{4.1.1}$$

Introducing the diagonal matrix $W = \mathrm{diag}([w_{-M}, \dots, w_0, \dots, w_M])$, we may write Eq. (4.1.1) compactly as:

$$\mathcal{J} = \mathbf{e}^T W \mathbf{e} = (\mathbf{y} - S\mathbf{c})^T W (\mathbf{y} - S\mathbf{c}) = \min \tag{4.1.2}$$

where $\mathbf{y}, S, \mathbf{c}$ have the same meaning as in Eqs. (3.2.26)–(3.2.30). Differentiating with respect to $\mathbf{c}$ gives the orthogonality and normal equations:

$$S^T W \mathbf{e} = S^T W (\mathbf{y} - S\mathbf{c}) = 0 \quad \Leftrightarrow \quad (S^T W S)\mathbf{c} = S^T W \mathbf{y} \tag{4.1.3}$$

with solution for $\mathbf{c}$ and the estimate $\hat{\mathbf{y}} = S\mathbf{c}$:

$$\begin{aligned} \mathbf{c} &= (S^T W S)^{-1} S^T W \mathbf{y} = G^T \mathbf{y} \\ \hat{\mathbf{y}} &= S\mathbf{c} = S(S^T W S)^{-1} S^T W \mathbf{y} = B^T \mathbf{y} \end{aligned} \tag{4.1.4}$$

where we defined

$$\boxed{\begin{aligned} G &= WS(S^T W S)^{-1} \\ B &= GS^T = WS(S^T W S)^{-1} S^T \end{aligned}} \tag{4.1.5}$$

The matrix B satisfies the following properties:

$$\begin{aligned} S^T B &= S^T \\ B^T &= W^{-1} B W \\ BWB^T &= BW = WB^T \end{aligned} \tag{4.1.6}$$

The first implies the usual polynomial-preserving moment constraints $S^T \mathbf{b}_m = \mathbf{u}_m$, for $-M \le m \le M$, where $\mathbf{b}_m$ is the mth column of B. The second shows that B is no longer symmetric, and the third may be used to simplify the minimized value of the performance index. Indeed, using the orthogonality property, we obtain:

$$\mathcal{J}_{\min} = \mathbf{e}^T W \mathbf{e} = \mathbf{y}^T W \mathbf{y} - \mathbf{y}^T B W \mathbf{y} - \mathbf{y}^T W B^T \mathbf{y} + \mathbf{y}^T B W B^T \mathbf{y} = \mathbf{y}^T W \mathbf{y} - \mathbf{y}^T B W \mathbf{y}$$

A fourth property follows if we assume that the weights w_m are symmetric about their middle, $w_m = w_{-m}$, or more generally if W is assumed to be positive-definite, symmetric, and centro-symmetric, which implies that it remains invariant under reversal of its rows and its columns. The centro-symmetric property can be stated concisely as $JW = WJ$, where J is the column-reversing matrix consisting of ones along its anti-diagonal, that is, the reverse of a column vector is $\mathbf{b}^R = J\mathbf{b}$. Under this assumption on W, it can be shown that B is also centro-symmetric:

$$JB = BJ \quad \Rightarrow \quad \mathbf{b}_m^R = \mathbf{b}_{-m}, \quad -M \le m \le M \tag{4.1.7}$$

This can be derived by noting that reversing the basis vector $\mathbf{s}_i$ simply multiplies it by the phase factor $(-1)^i$, so that $JS = S\Omega$, where Ω is the diagonal matrix of phase factors $(-1)^i$, $i = 0, 1, \dots, d$. This then implies Eq. (4.1.7). Similarly one can show that $JG = G\Omega$, so that the reverse of each differentiation filter is $\mathbf{g}_i^R = (-1)^i \mathbf{g}_i$.

The filtering equations (3.2.33) and (3.2.34) retain their form. Among the possible weighting matrices W, we are interested in those such that the polynomial fitting problem (4.1.2) has an equivalent characterization as the minimization of the NRR subject to the polynomial-preserving constraints $S^T \mathbf{b}_m = \mathbf{u}_m$. To this end, we consider the constrained minimization of a generalized or "prefiltered" NRR:

$$\boxed{\mathcal{R} = \mathbf{b}^T V \mathbf{b} = \min, \quad \text{subject to} \quad S^T \mathbf{b} = \mathbf{u}} \tag{4.1.8}$$

for a given $(d+1)$-dimensional vector $\mathbf{u}$. The $N \times N$ matrix V, where $N = 2M+1$, is assumed to be strictly positive-definite, symmetric, and Toeplitz. We may write component-wise:

$$\mathcal{R} = \sum_{n,m=-M}^{M} b(n) V_{n-m} b(m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |B(\omega)|^2 V(\omega)\, d\omega \tag{4.1.9}$$

where we set $V_{nm} = V_{n-m}$ because of the Toeplitz property, and introduced the corresponding DTFTs:

$$B(\omega) = \sum_{n=-M}^{M} b(n) e^{-j\omega n}, \quad V(\omega) = \sum_{k=-\infty}^{\infty} V_k e^{-j\omega k} \tag{4.1.10}$$

One way to guarantee a positive-definite V is to take $V(\omega)$ to be the power spectrum of a given filter, say, $D(\omega)$, that is, choose $V(\omega)=|D(\omega)|^2$, so that $\mathcal{R}$ will be the ordinary NRR of the cascaded filter $F(\omega)=D(\omega)B(\omega)$ or $F(z)=D(z)B(z)$:

$$\mathcal{R}=\frac{1}{2\pi}\int_{-\pi}^{\pi}|B(\omega)|^2V(\omega)d\omega=\frac{1}{2\pi}\int_{-\pi}^{\pi}|B(\omega)D(\omega)|^2d\omega \tag{4.1.11}$$

The minimum-R_s or minimum-roughness filters discussed in Sec. 4.2 correspond to the choice $D(z)=(1-z^{-1})^s$, for some integer s. For a general V and $\mathbf{u}$, the solution of the problem (4.1.8) is obtained by introducing a Lagrange multiplier vector $\boldsymbol{\lambda}$:

$$\mathcal{J}=\mathbf{b}^TV\mathbf{b}+2\boldsymbol{\lambda}^T(\mathbf{u}-S^T\mathbf{b})=\min$$

leading to the solution:

$$\begin{aligned}\boldsymbol{\lambda}&=(S^TV^{-1}S)^{-1}\mathbf{u}\\ \mathbf{b}&=V^{-1}S\boldsymbol{\lambda}=V^{-1}S(S^TV^{-1}S)^{-1}\mathbf{u}\end{aligned} \tag{4.1.12}$$

If we choose $\mathbf{u}_m=[1,m,m^2,\dots,m^d]^T$ as the constraint vectors and put together the resulting solutions as the columns of a matrix B, then,

$$B=[\cdots\mathbf{b}_m\cdots]=V^{-1}S(S^TV^{-1}S)^{-1}[\cdots\mathbf{u}_m\cdots]$$

or, because $S^T=[\cdots\mathbf{u}_m\cdots]$,

$$\boxed{B=V^{-1}S(S^TV^{-1}S)^{-1}S^T} \tag{4.1.13}$$

This solution appears to be different from the solution (4.1.5) of the least-squares problem, $B=WS(S^TWS)^{-1}S^T$. Can the two solutions be the same? The trivial choice $V=W=I$ corresponds to the LPSM filters. The choice $V=W^{-1}$ is not acceptable because with V assumed Toeplitz, and W assumed diagonal, it would imply that all the weights are equal, which is again the LPSM case. A condition that guarantees the equivalence is the following [123,99]:

$$VWS=SC\quad\Rightarrow\quad WS=V^{-1}SC \tag{4.1.14}$$

where C is an invertible $(d+1)\times(d+1)$ matrix. Indeed, then $S^TWS=S^TV^{-1}SC$, and,

$$G=WS(S^TWS)^{-1}=V^{-1}S(S^TV^{-1}S)^{-1} \tag{4.1.15}$$

so that

$$B=WS(S^TWS)^{-1}S^T=V^{-1}S(S^TV^{-1}S)^{-1}S^T \tag{4.1.16}$$

For the minimum-R_s filters, the particular choices for W,V do indeed satisfy condition (4.1.14) with an *upper-triangular* matrix C. With the equivalence of the polynomial-fitting and minimum-NRR approaches at hand, we can also derive the corresponding predictive/interpolating differentiation filters. Choosing $\mathbf{u}=\mathcal{D}^i\mathbf{u}_t$ as the constraint vector in (4.1.12), we obtain,

$$\mathbf{b}_t^{(i)}=V^{-1}S(S^TV^{-1}S)^{-1}\mathcal{D}^i\mathbf{u}_t=WS(S^TWS)^{-1}\mathcal{D}^i\mathbf{u}_t \tag{4.1.17}$$

and at the sample values $t=m$, $-M\le m\le M$, or, at $\mathbf{u}_t=\mathbf{u}_m$, we obtain the differentiation matrix having the $\mathbf{b}_m^{(i)}$ as columns, $B^{(i)}=[\cdots\mathbf{b}_m^{(i)}\cdots]$:

$$B^{(i)}=WS(S^TWS)^{-1}\mathcal{D}^iS^T=V^{-1}S(S^TV^{-1}S)^{-1}\mathcal{D}^iS^T \tag{4.1.18}$$

Computationally, it is best to orthogonalize the basis S. Let $W=U^TU$ be the Cholesky factorization of the positive-definite symmetric matrix W, where U is an $N\times N$ upper-triangular factor. Then, performing the QR-factorization on the $N\times(d+1)$ matrix US, the above computations become:

$$\begin{aligned}&W=U^TU\\ &US=Q_0R_0\,,\quad\text{with }\ Q_0^TQ_0=I\,,\quad R_0=(d+1)\times(d+1)\ \text{upper-triangular}\\ &B=U^TQ_0Q_0^TU^{-T}\\ &B^{(i)}=U^TQ_0(R_0^{-T}\mathcal{D}^iR_0^T)Q_0^TU^{-T}\\ &\mathbf{b}_t^{(i)}=U^TQ_0R_0^{-T}\mathcal{D}^i\mathbf{u}_t\end{aligned} \tag{4.1.19}$$

The MATLAB functions `lpsm`, `lpdiff`, `lpinterp` have the weighting matrix W as an additional input, which if omitted defaults to $W=I$. They implement Eqs. (4.1.19) and their full usage is:

```
[B,G] = lpsm(N,d,W);
B = lpdiff(N,d,i,W);
b = lpinterp(N,d,t,i,W);
```

The factorizations in Eq. (4.1.19) lead naturally to a related implementation in terms of discrete polynomials that are orthogonal with respect to the weighted inner product:

$$\mathbf{a}^TW\mathbf{b}=\sum_{m=-M}^{M}w_m\,a(m)b(m) \tag{4.1.20}$$

Such polynomials may be constructed from the monomials $s_i(m)=m^i$, $i=0,1,\dots,d$ via Gram-Schmidt orthogonalization applied with respect to the above inner product. The result of orthogonalizing the basis $S=[\mathbf{s}_0,\mathbf{s}_1,\dots,\mathbf{s}_d]$ is $Q=[\mathbf{q}_0,\mathbf{q}_1,\dots,\mathbf{q}_d]$ whose columns $q_i(m)$ are polynomials of order i in the variable m that are mutually orthogonal, that is, up to an overall normalization:

$$\mathbf{q}_i^TW\mathbf{q}_j=\delta_{ij}D_i\,,\quad i,j=0,1,\dots,d\quad\Rightarrow\quad Q^TWQ=D \tag{4.1.21}$$

where $D=\mathrm{diag}([D_0,D_1,\dots,D_d])$ is the diagonal matrix of the (positive) normalization factors D_i. These factors can be selected to be unity if so desired. For the minimum-roughness filters, these polynomials are special cases of the Hahn orthogonal polynomials, whose properties are discussed in Sec. 4.3. For unity weights $w_m=1$, the polynomials reduce to the discrete Chebyshev/Gram polynomials.

Numerically, these polynomials can be constructed from the factorization (4.1.19). Since D is positive-definite, we may define $D^{1/2}=\mathrm{diag}([D_0^{1/2},D_1^{1/2},\dots,D_d^{1/2}])$ to be its square root. Then we construct Q,R in terms of the factors U,Q_0,R_0:

$$Q=U^{-1}Q_0D^{1/2}\,,\quad R=D^{-1/2}R_0 \tag{4.1.22}$$

where R is still upper-triangular. Then, we have $Q^TWQ = D$ and

$$QR = U^{-1}Q_0D^{1/2}D^{-1/2}R_0 = U^{-1}Q_0R_0 = U^{-1}US = S$$

which is equivalent to the Gram-Schmidt orthogonalization of the basis S, and leads to the following equivalent representation of Eq. (4.1.19):

$$\begin{aligned}
&S = QR\,, \quad \text{with } Q^TWQ = D\,, \quad R = (d+1)\times(d+1) \text{ upper-triangular}\\
&B = WQD^{-1}Q^T\\
&B^{(i)} = WQD^{-1}(R^{-T}\mathcal{D}^iR^T)Q^T\\
&\mathbf{b}_t^{(i)} = WQD^{-1}R^{-T}\mathcal{D}^i\mathbf{u}_t
\end{aligned} \tag{4.1.23}$$

Since $Q = [\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_d]$, the matrix B can be expressed as,

$$B = WQD^{-1}Q^T = W\sum_{r=0}^{d} D_r^{-1}\mathbf{q}_r\mathbf{q}_r^T \tag{4.1.24}$$

and for diagonal W, we have component-wise:

$$\boxed{b_m(k) = B_{km} = w_k\sum_{r=0}^{d}\frac{q_r(k)q_r(m)}{D_r}} \qquad -M \le m, k \le M \tag{4.1.25}$$

The sum in (4.1.25) can be simplified further using the Christoffel-Darboux identity discussed in Sec. 4.3. The polynomial predictive interpolation filters $\mathbf{b}_t^{(i)}$ can also be expressed in a similar summation form:

$$\boxed{b_t^{(i)}(k) = w_k\sum_{r=0}^{d}\frac{q_r(k)q_r^{(i)}(t)}{D_r}} \tag{4.1.26}$$

where $q_r^{(i)}(t)$ is the ith derivative of the polynomial $q_r(t)$ obtained from $q_r(m)$ by replacing the discrete variable m by t. This can be justified as follows. The mth rows of the matrices S and Q are the $(d+1)$-dimensional vectors:

$$\begin{aligned}
\mathbf{u}_m^T &= [s_0(m), s_1(m), \dots, s_d(m)] = [1, m, \dots, m^d]\\
\mathbf{p}_m^T &= [q_0(m), q_1(m), \dots, q_d(m)]
\end{aligned} \tag{4.1.27}$$

and since $S = QR$, they are related by $\mathbf{u}_m^T = \mathbf{p}_m^TR$. Replacing m by t preserves this relationship, so that $\mathbf{u}_t^T = \mathbf{p}_t^TR$, or,

$$\mathbf{u}_t = R^T\mathbf{p}_t\,, \quad \text{where } \mathbf{p}_t = [q_0(t), q_1(t), \dots, q_d(t)]^T \tag{4.1.28}$$

Differentiating i times, we obtain

$$\mathcal{D}^i\mathbf{u}_t = \mathbf{u}_t^{(i)} = R^T\mathbf{p}_t^{(i)} \quad \Rightarrow \quad \mathbf{p}_t^{(i)} = R^{-T}\mathcal{D}^i\mathbf{u}_t \tag{4.1.29}$$

and therefore $\mathbf{b}_t^{(i)}$ from Eq. (4.1.23) can be written in the following form, which implies Eq. (4.1.26):

$$\boxed{\mathbf{b}_t^{(i)} = WQD^{-1}\mathbf{p}_t^{(i)}} \tag{4.1.30}$$

As in the case of the LPSM filters, for the special case $d = N-1$, the interpolation filters correspond to Lagrange interpolation. In this case Q becomes an invertible $N\times N$ matrix satisfying the weighted unitarity property $Q^TWQ = D$, which implies

$$Q^{-1} = D^{-1}Q^TW \tag{4.1.31}$$

from which we obtain the *completeness* property:

$$\boxed{QD^{-1}Q^T = W^{-1}} \tag{4.1.32}$$

which shows that $B = I$. Similarly, using $WQD^{-1} = Q^{-T}$, we obtain from (4.1.23) the usual Lagrange interpolation polynomials:

$$\mathbf{b}_t = WQD^{-1}R^{-T}\mathbf{u}_t = Q^{-T}R^{-T}\mathbf{u}_t = S^{-T}\mathbf{u}_t \tag{4.1.33}$$

With $d = N-1$, the matrix Q is an orthogonal basis for the full space $\mathbb{R}^N$. One of the applications of Eq. (4.1.31) is the representation of signals, such as images or speech in terms of *orthogonal-polynomial moments* [137–150].

Given an N-dimensional signal block $\mathbf{y}$, such as a row in a scanned image, we define the N-dimensional vector of moments with respect to the polynomials Q,

$$\boldsymbol{\mu} = D^{-1}Q^TW\mathbf{y} \quad \Rightarrow \quad \mu_r = \frac{1}{D_r}\sum_{n=-M}^{M} q_r(n)w_ny_n\,, \quad r = 0, 1, \dots, N-1 \tag{4.1.34}$$

Because of Eq. (4.1.31), we have $\boldsymbol{\mu} = Q^{-1}\mathbf{y}$, which allows the reconstruction of $\mathbf{y}$ from its moments:

$$\mathbf{y} = Q\boldsymbol{\mu} \quad \Rightarrow \quad y_n = \sum_{r=0}^{N-1} q_r(n)\mu_r\,, \quad -M \le n \le M \tag{4.1.35}$$

4.2 Henderson Filters

All the results of the previous section find a concrete realization in the minimum-R_s filters that we discuss here. Consider the order-s backward difference filter and its impulse response defined by:

$$D_s(z) = (1 - z^{-1})^s \quad \Leftrightarrow \quad d_s(k) = (-1)^k\binom{s}{k}\,, \quad k = 0, 1, \dots, s \tag{4.2.1}$$

This follows from the binomial expansion:

$$(1 - z^{-1})^s = \sum_{k=0}^{s}(-1)^k\binom{s}{k}z^{-k} \tag{4.2.2}$$

The operation of the filter $D_s(z)$ on a signal f_n, with output g_n, is usually denoted in terms of the backward difference operator $\nabla f_n = f_n - f_{n-1}$ as follows:

$$g_n = \nabla^s f_n = \sum_{k=0}^{s} d_s(k) f_{n-k} = \sum_{k=0}^{s} (-1)^k \binom{s}{k} f_{n-k} \tag{4.2.3}$$

If the signal f_n is restricted over the range $-M \le n \le M$, then because $0 \le k \le s$ and $-M \le n-k \le M$, the above equation can be written in the more precise form:

$$g_n = \nabla^s f_n = \sum_{k=\max(0,n-M)}^{\min(s,n+M)} (-1)^k \binom{s}{k} f_{n-k}\,, \quad -M \le n \le M+s \tag{4.2.4}$$

Eq. (4.2.4) gives the full convolutional output $g_n = (d_s * f)_n$, while (4.2.3) is the corresponding steady-state output, obtained by restricting the output index n to the range $-M+s \le n \le M$. Defining the $(N+s)$-dimensional output vector $\mathbf{g}$ and N-dimensional input vector $\mathbf{f}$, where $N = 2M+1$,

$$\mathbf{g} = [g_{-M}, \dots, g_M, \dots, g_{M+s}]^T\,, \quad \mathbf{f} = [f_{-M}, \dots, f_M]^T\,,$$

we may write the full filtering equation (4.2.4) in matrix form:

$$\mathbf{g} = D_s \mathbf{f} \tag{4.2.5}$$

where D_s is the full $(N+s)\times N$ convolutional matrix of the filter $d_s(k)$ defined by its matrix elements:

$$(D_s)_{nm} = d_s(n-m)\,, \quad -M \le n \le M+s\,, \quad -M \le m \le M \tag{4.2.6}$$

and subject to the restriction that only the values $0 \le n-m \le s$ will result in a non-zero matrix element. The MATLAB functions `binom` and `diffmat` allow the calculation of the binomial coefficients $d_s(k)$ and the convolution matrix D_s:

```
d = binom(s,k);      % binomial coefficients d_s(k)
D = diffmat(s,N);    % (N+s)×N difference convolution matrix
```

For example, the convolution matrix for $N = 7$ and $s = 3$ is:

$$D_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 3 & -3 & 1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -3 & 1 & 0 & 0 & 0 \\ 0 & -1 & 3 & -3 & 1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -3 & 1 & 0 \\ 0 & 0 & 0 & -1 & 3 & -3 & 1 \\ 0 & 0 & 0 & 0 & -1 & 3 & -3 \\ 0 & 0 & 0 & 0 & 0 & -1 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

The function `diffmat` is simply a call to `convmat`:


```
D = convmat(binom(s),N);
```

A minimum-R_s filter $B(z)$ is defined to minimize the NRR of the cascaded filter $F(z) = D_s(z)B(z)$ subject to the $d+1$ linear constraints $S^T\mathbf{b} = \mathbf{u}$, for a given constraint vector $\mathbf{u}$, where $\mathbf{b}$ denotes the impulse response of $B(z)$ assumed to be double-sided, that is, b_n, $-M \le n \le M$.

The actual smoothing of data is carried out by the filter $B(z)$ itself, whereas the filter $F(z)$ is used to design $B(z)$. This is depicted in Fig. 4.2.1 in which the filtered output is $\hat{x}_n$, and the output of $F(z)$ is the differenced signal $\nabla^s \hat{x}_n$ whose mean-square value may be taken as a measure of smoothness to be minimized.

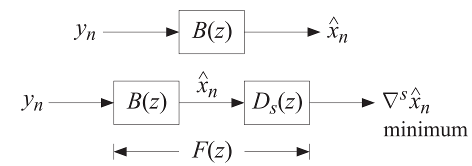

Fig. 4.2.1 Design and smoothing by minimum-R_s filter.

Letting $f_n = \nabla^s b_n$ be the impulse response of the filter $F(z)$, or in matrix form $\mathbf{f} = D_s \mathbf{b}$, the corresponding cascaded NRR will be:

$$\mathcal{R}_s = \mathbf{f}^T\mathbf{f} = \sum_{n=-M}^{M+s} f_n^2 = \sum_{n=-M}^{M+s} (\nabla^s b_n)^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |D_s(\omega)B(\omega)|^2\, d\omega$$

Since $\mathbf{f}^T\mathbf{f} = \mathbf{b}^T(D_s^T D_s)\mathbf{b}$, we can state the design condition of the minimum-R_s filters as

$$\boxed{\mathcal{R}_s = \sum_{n=-M}^{M+s} (\nabla^s b_n)^2 = \mathbf{b}^T(D_s^T D_s)\mathbf{b} = \min\,, \quad \text{subject to } \ S^T\mathbf{b} = \mathbf{u}} \tag{4.2.7}$$

This has exactly the same form as Eq. (4.1.8) with $V = D_s^T D_s$. The minimization of $\mathcal{R}_s$ justifies the name "minimum-R_s " filters. The minimum-R_0 LPSM filters of Sec. 3.7 correspond to $s = 0$. In the actuarial literature, the following criterion is used instead, which differs from $\mathcal{R}_s$ by a normalization factor:

$$R_s = \frac{\mathbf{b}^T(D_s^T D_s)\mathbf{b}}{\mathbf{d}_s^T\mathbf{d}_s} = \frac{\mathcal{R}_s}{\mathbf{d}_s^T\mathbf{d}_s} = \min \tag{4.2.8}$$

where R_s is referred as the "smoothing coefficient", $\mathbf{d}_s$ is the impulse response vector of the filter $D_s(z)$, and $\mathbf{d}_s^T\mathbf{d}_s$ is the NRR of $D_s(z)$. Using a binomial identity (a special case of (4.2.13) for $k = 0$), we have,

$$\mathbf{d}_s^T\mathbf{d}_s = \sum_{m=0}^{s} d_s^2(m) = \sum_{m=0}^{s} \binom{s}{m}^2 = \binom{2s}{s} \tag{4.2.9}$$

The criterion (4.2.7) provides a measure of smoothness. To see this, let $\hat{x}_n$ be the result of filtering an arbitrary stationary signal y_n through the filter $B(z)$. If $S_{yy}(\omega)$ is the power spectrum of y_n , then the power spectra of the filtered output $\hat{x}_n$ and of the differenced output $\nabla^s\hat{x}_n$ will be $|B(\omega)|^2 S_{yy}(\omega)$ and $|D_s(\omega)B(\omega)|^2 S_{yy}(\omega)$, respectively. Therefore, the mean-square value of $\nabla^s\hat{x}_n$ will be:

$$E\big[(\nabla^s\hat{x}_n)^2\big] = \frac{1}{2\pi}\int_{-\pi}^{\pi}|D_s(\omega)B(\omega)|^2 S_{yy}(\omega)\,d\omega \tag{4.2.10}$$

If y_n is white noise of variance σ^2, or if we assume that $S_{yy}(\omega)$ is bounded from above by a constant, such as $S_{yy}(\omega)\le\sigma^2$, then we obtain:

$$E\big[(\nabla^s\hat{x}_n)^2\big] \le \frac{1}{2\pi}\int_{-\pi}^{\pi}|D_s(\omega)B(\omega)|^2 \sigma^2\,d\omega = \mathcal{R}_s\sigma^2 \tag{4.2.11}$$

For white noise, $S_{yy}(\omega)=\sigma^2$, Eq. (4.2.11) becomes an equality. Thus, minimizing $\mathcal{R}_s$ will minimize $E\big[(\nabla^s\hat{x}_n)^2\big]$ and tend to result in a smoother filtered signal $\hat{x}_n$. This property justifies the term "minimum-roughness" filters.

The choice $s=2$ is preferred in smoothing financial and business-cycle data, and is used also by the related method of the Whittaker-Henderson or Hodrick-Prescott filter. The choice $s=3$ is standard in the actuarial literature. The choice $s=4$ is not common but it was used by De Forest [65–68] who was the first to formulate and solve the minimum-R_s problem in 1871. Others, like Hardy and Henderson have considered the minimum-R_3 problem, while Sheppard [76] solved the minimum-R_s problem in general.

Henderson [79] was the first to show the *equivalence* between the NRR minimization problem (4.2.7) with $V=D_s^T D_s$ and the weighted least-squares polynomial fitting problem (4.1.1) using the so-called Henderson weights w_m. Therefore, the minimum-R_s filters are often referred to as *Henderson filters*. They are used widely in seasonal-adjustment, census, and business-cycle extraction applications. We discuss this equivalence next, following essentially Henderson's method.

The elements of the $N\times N$ matrix $V=D_s^T D_s$ are $(D_s^T D_s)_{nm}=V_{nm}=V_{n-m}$, where V_k is the autocorrelation function of the power spectrum $V(\omega)=|D_s(\omega)|^2$. Working in the z-domain, we have the spectral density:

$$V(z)=D_s(z)D_s(z^{-1})=(1-z^{-1})^s(1-z)^s=(-1)^s z^s(1-z^{-1})^{2s} \tag{4.2.12}$$

which shows that $V(z)$ effectively acts as the $(2s)$-difference operation ∇^{2s}. Taking inverse z-transforms of both sides of (4.2.12), we obtain:

$$V_k = \sum_{m=\max(0,k)}^{\min(s,k+s)} d_s(m)d_s(m-k) = (-1)^s d_{2s}(k+s)\,,\quad -s\le k\le s \tag{4.2.13}$$

or, explicitly in terms of the definition of d_s:

$$V_k = (-1)^k\sum_{m=\max(0,k)}^{\min(s,k+s)}\binom{s}{m}\binom{s}{m-k} = (-1)^k\binom{2s}{s+k},\quad -s\le k\le s \tag{4.2.14}$$

or,

$$V_k = (-1)^k\frac{(2s)!}{(s+k)!\,(s-k)!}\,,\quad -s\le k\le s \tag{4.2.15}$$

The V matrix is a banded Toeplitz matrix with bandwidth $\pm s$, whose central row or central column consist of the numbers V_k, $-s\le k\le s$, with V_0 positioned at the center of the matrix. As an example,

$$V = D_3^T D_3 = \begin{bmatrix} 20 & -15 & 6 & -1 & 0 & 0 & 0\\ -15 & 20 & -15 & 6 & -1 & 0 & 0\\ 6 & -15 & 20 & -15 & 6 & -1 & 0\\ -1 & 6 & -15 & 20 & -15 & 6 & -1\\ 0 & -1 & 6 & -15 & 20 & -15 & 6\\ 0 & 0 & -1 & 6 & -15 & 20 & -15\\ 0 & 0 & 0 & -1 & 6 & -15 & 20 \end{bmatrix}$$

with central column or central row:

$$V_k = \{-1,\ 6,\ -15,\ 20,\ -15,\ 6,\ -1\}\quad\text{for}\quad k=\{-2,-1,0,1,2\}$$

To understand the action of V as the difference operator ∇^{2s}, let $\mathbf{f}$ be an N dimensional vector indexed for $-M\le m\le M$, and form the output N-dimensional vector:

$$\mathbf{g}=V\mathbf{f}\quad\Rightarrow\quad g_n=\sum_{m=-M}^{M}V_{n-m}f_m\,,\quad -M\le n\le M \tag{4.2.16}$$

where $n-m$ is further restricted such that $-s\le n-m\le s$. Next, consider an extended version of $\mathbf{f}$ obtained by padding s zeros in front and s zeros at the end, so that the extended vector $\mathbf{f}^{\text{ext}}$ will be indexed over, $-(M+s)\le m\le(M+s)$:

$$\mathbf{f}^{\text{ext}} = [\underbrace{0,\dots,0}_{s}, f_{-M},\dots,f_0,\dots,f_M,\underbrace{0,\dots,0}_{s}]^T$$

Then, the summation in Eq. (4.2.16) can be extended as,

$$g_n=\sum_{m=-M-s}^{M+s}V_{n-m}f_m^{\text{ext}}\,,\quad -M\le n\le M \tag{4.2.17}$$

But because of the restriction $-s\le n-m\le s$, the above summation can be restricted to be over $n-s\le m\le n+s$, which is a subrange of the range $-(M+s)\le m\le(M+s)$ because we assumed $-M\le n\le M$. Thus, we may write:

$$g_n=\sum_{m=-n-s}^{n+s}V_{n-m}f_m^{\text{ext}}\,,\quad -M\le n\le M$$

or, changing to $k=n-m$,

$$g_n=\sum_{k=-s}^{s}V_k f_{n-k}^{\text{ext}} = (-1)^s\sum_{k=-s}^{s}d_{2s}(s+k)f_{n-k}^{\text{ext}} = (-1)^s\sum_{i=0}^{2s}d_{2s}(i)f_{n+s-i}^{\text{ext}} \tag{4.2.18}$$

but that is precisely the ∇^{2s} operator:

$$g_n=(-1)^s\nabla^{2s}f_{n+s}^{\text{ext}}\,,\quad -M\le n\le M \tag{4.2.19}$$

If f_m^{ext} is a polynomial of degree $(2s+i)$, then the $(2s)$-differencing operation will result into a polynomial of degree i. Suppose that we start with the weighted monomial:

$$f_m = w_m m^i\,, \quad -M \le m \le M \tag{4.2.20}$$

where the weighting function w_m is itself a polynomial of degree $2s$, then in order for the extended vector f_m^{ext} to vanish over $M < |m| \le M+s$, the function w_m must have zeros at these points, that is,

$$w_m = 0\,, \quad \text{for } \; m = \pm(M+1), \pm(M+2), \dots, \pm(M+s)$$

This condition fixes w_m uniquely, up to a normalization constant:

$$\boxed{w_m = \prod_{i=1}^{s}\left[(M+i)^2 - m^2\right]} \qquad \text{(Henderson weights)} \tag{4.2.21}$$

These are called *Henderson weights.* Because the extended signal f_m^{ext} is a polynomial of degree $(2s+i)$, it follows that the signal g_n will be a polynomial of degree i.

Defining the $N\times N$ diagonal matrix $W = \text{diag}\big([w_{-M}, \dots, w_0, \dots, w_M]\big)$, we can write (4.2.20) vectorially in terms of the monomial basis vector $\mathbf{s}_i$ as $\mathbf{f} = W\mathbf{s}_i$. We showed that the matrix operation $\mathbf{g} = V\mathbf{f} = VW\mathbf{s}_i$ results into a polynomial of degree i, which therefore can be expanded as a linear combination of the monomials $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_i$ up to order i, that is,

$$VW\mathbf{s}_i = \sum_{j=0}^{i} \mathbf{s}_j C_{ji} \tag{4.2.22}$$

for appropriate coefficients C_{ji}, which may thought of as the matrix elements of an upper-triangular matrix. Applying this result to each basis vector of $S = [\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_d]$ up to order d, it follows that

$$VWS = SC\,, \quad C = (d+1)\times(d+1) \text{ upper-triangular} \tag{4.2.23}$$

But, this is exactly the condition (4.1.14). Thus, we have shown the equivalence of the NRR minimization problem (4.2.7) with $V = D_s^T D_s$ and the weighted least-squares polynomial fitting problem (4.1.1) with the Henderson weights w_m. The rest of the results of Sec. 4.1 then carry through unchanged.

The MATLAB function `lprs` implements the design. It constructs the W matrix from the Henderson weights and passes it into the function `lpsm`:

```
[B,G] = lprs(N,d,s);        % local polynomial minimum-R_s filters
```

The Henderson weights w_m, $-M \le m \le M$ are calculated by the function `hend`:

```
w = hend(N,s);        % Henderson weights
```

In the next section, we derive closed-form expressions for the Henderson filters using Hahn orthogonal polynomials. Analytical expressions can also be derived working with

the non-orthogonal monomial basis S. It follows from $B = WS(S^TWS)^{-1}S^T$ that the kth component of the mth filter will be:

$$b_m(k) = B_{km} = w_k \sum_{i,j=0}^{d} k^i m^j \Phi_{ij} = w_k\, \mathbf{u}_k^T \Phi \mathbf{u}_m \tag{4.2.24}$$

where $\mathbf{u}_k = [1, k, k^2, \dots, k^d]^T$ and Φ is the inverse of the Hankel matrix $F = S^TWS$ whose matrix elements are the weighted inner products:

$$F_{ij} = (S^TWS)_{ij} = \mathbf{s}_i^T W \mathbf{s}_j = \sum_{m=-M}^{M} w_m m^{i+j} \equiv F_{i+j}\,, \quad i,j = 0,1,\dots,d \tag{4.2.25}$$

Except for the factor w_k and the different values of Φ_{ij} the expressions are similar to those of the LPSM filters of Sec. 3.3. The matrix Φ has a similar checkerboard structure. For example, we have for the commonly used case $d = 3$ and $s = 3$:

$$b_m(k) = w_k\,[1,\ k,\ k^2,\ k^3] \begin{bmatrix} \Phi_{00} & 0 & \Phi_{02} & 0 \\ 0 & \Phi_{11} & 0 & \Phi_{13} \\ \Phi_{20} & 0 & \Phi_{22} & 0 \\ 0 & \Phi_{31} & 0 & \Phi_{33} \end{bmatrix} \begin{bmatrix} 1 \\ m \\ m^2 \\ m^3 \end{bmatrix} \tag{4.2.26}$$

where

$$w_k = \left[(M+1)^2 - k^2\right]\left[(M+2)^2 - k^2\right]\left[(M+3)^2 - k^2\right] \tag{4.2.27}$$

and

$$F = \begin{bmatrix} F_0 & 0 & F_2 & 0 \\ 0 & F_2 & 0 & F_4 \\ F_2 & 0 & F_4 & 0 \\ 0 & F_4 & 0 & F_6 \end{bmatrix} \quad \Rightarrow \quad \Phi = F^{-1} = \begin{bmatrix} \Phi_{00} & 0 & \Phi_{02} & 0 \\ 0 & \Phi_{11} & 0 & \Phi_{13} \\ \Phi_{20} & 0 & \Phi_{22} & 0 \\ 0 & \Phi_{31} & 0 & \Phi_{33} \end{bmatrix}$$

where we obtain from the checkerboard submatrices:

$$\begin{bmatrix} \Phi_{00} & \Phi_{02} \\ \Phi_{20} & \Phi_{22} \end{bmatrix} = \begin{bmatrix} F_0 & F_2 \\ F_2 & F_4 \end{bmatrix}^{-1}, \quad \begin{bmatrix} \Phi_{11} & \Phi_{13} \\ \Phi_{31} & \Phi_{33} \end{bmatrix} = \begin{bmatrix} F_2 & F_4 \\ F_4 & F_6 \end{bmatrix}^{-1} \tag{4.2.28}$$

The corresponding F-factors for $s = 3$ are:

$$F_0 = \frac{2}{35}(2M+7)(2M+5)(2M+3)(2M+1)(M+3)(M+2)(M+1)$$

$$F_2 = \frac{1}{9}M(M+4)F_0$$

$$F_4 = \frac{1}{11}(3M^2 + 12M - 4)F_2$$

$$F_6 = \frac{1}{143}(15M^4 + 120M^3 + 180M^2 - 240M + 68)F_2$$

which give rise to the matrix elements of Φ:

$$\begin{aligned}
\Phi_{00} &= 315\,(3M^2+12M-4)\,/\,D_1\\
\Phi_{02} &= -3465\,/\,D_1\\
\Phi_{22} &= 31185\,/\,D_1\\
\Phi_{11} &= 1155\,(15M^4+120M^3+180M^2-240M+68)\,/\,D_2\\
\Phi_{13} &= -15015\,(3M^2+12M-4)\,/\,D_2\\
\Phi_{33} &= 165165\,/\,D_2
\end{aligned}$$

with the denominator factors:

$$\begin{aligned}
D_1 &= 8\,(2M+9)\,(2M+7)\,(2M+5)\,(2M+3)\,(M+3)\,(M+2)\,(M+1)\,(4M^2-1)\\
D_2 &= 8M\,(M-1)\,(M+4)\,(M+5)\,D_1
\end{aligned}$$

In particular, setting $m = 0$ we find the central filter $b_0(k)$, which for the case $d = 3$ and $s = 3$, is referred to as "Henderson's ideal formula:"

$$b_0(k) = w_k\,(\Phi_{00} + k^2\Phi_{02})$$

or, with $w_k = \big[(M+1)^2-k^2\big]\big[(M+2)^2-k^2\big]\big[(M+3)^2-k^2\big]$:

$$b_0(k) = \frac{315\,(3M^2+12M-4-11k^2)\,w_k}{8\,(2M+9)\,(2M+7)\,(2M+5)\,(2M+3)\,(M+3)\,(M+2)\,(M+1)\,(4M^2-1)} \tag{4.2.29}$$

The corresponding predictive/interpolating differentiation filters $b_t^{(i)}(k)$ are given by a similar expression:

$$b_t^{(i)}(k) = w_k \mathbf{u}_k^T \Phi \mathcal{D}^i \mathbf{u}_t \tag{4.2.30}$$

or, explicitly, for the $d = s = 3$ case and differentiation order $i = 0, 1, 2, 3$:

$$b_t^{(i)}(k) = w_k\,[1,\ k,\ k^2,\ k^3]\begin{bmatrix}\Phi_{00} & 0 & \Phi_{02} & 0\\ 0 & \Phi_{11} & 0 & \Phi_{13}\\ \Phi_{20} & 0 & \Phi_{22} & 0\\ 0 & \Phi_{31} & 0 & \Phi_{33}\end{bmatrix}\begin{bmatrix}0&0&0&0\\1&0&0&0\\0&2&0&0\\0&0&3&0\end{bmatrix}^i\begin{bmatrix}1\\ t\\ t^2\\ t^3\end{bmatrix} \tag{4.2.31}$$

Example 4.2.1: *USD/Euro exchange rate.* Consider four methods of smoothing the USD/Euro foreign exchange rate for the years 1999-08. The monthly data are available from the web site: `http://research.stlouisfed.org/fred2/series/EXUSEU`

The upper-left graph in Fig. 4.2.2 shows the smoothing by an LPSM filter of length $N = 19$ and polynomial order $d = 3$. In the upper-right graph a minimum-R_s Henderson filter was used with $N = 19$, $d = 3$, and smoothness order $s = 3$.

The middle-left graph uses the SVD signal enhancement method with embedding order $M = 8$ and rank $r = 2$.

The middle-right graph uses the Whittaker-Henderson, or Hodrick-Prescott filter with smoothing parameter $\lambda = 100$ and smoothness order $s = 3$.

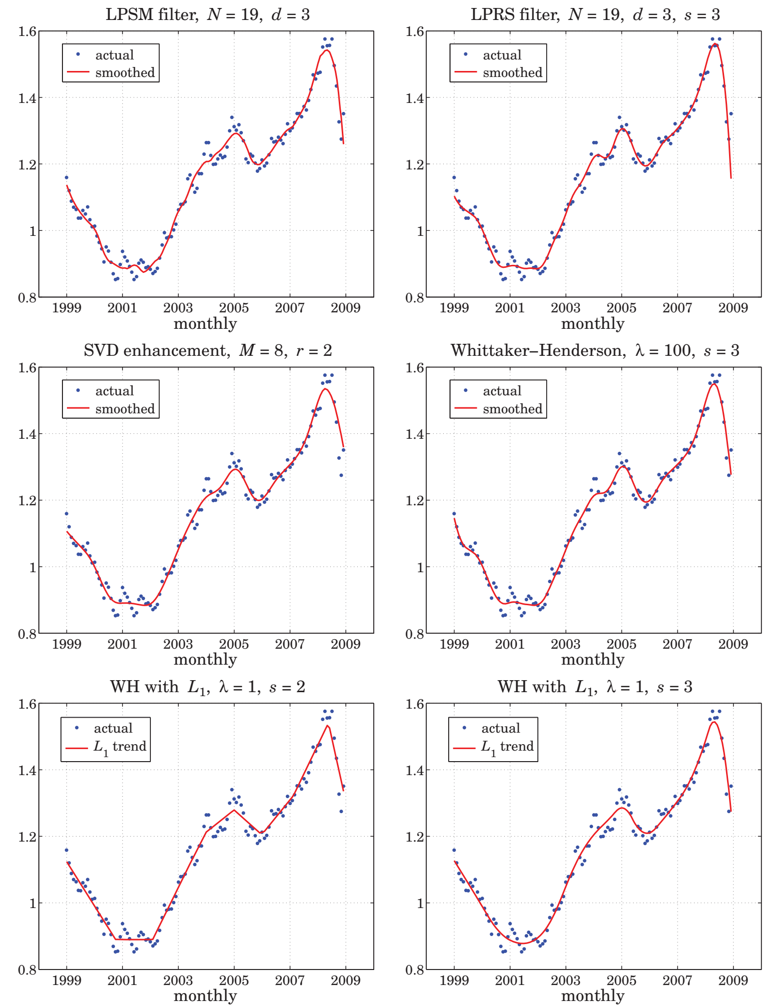

Fig. 4.2.2 Smoothing of USD/Euro exchange rate.

The lower left and right graphs use the Whittaker-Henderson regularization filter with the L_1 criterion with differentiation orders $s = 2$ and $s = 3$ and smoothing parameter $\lambda = 1$, implemented with the CVX package.[†]. The $s = 2$ case represents the smoothed signal in piece-wise linear form. The L_1 case is discussed further in Sec. 8.7.

The following MATLAB code illustrates the generation of the four graphs:

```
Y = loadfile('exuseu.dat');                % data file available in the OSP toolbox
```

[†] `http://cvxr.com/cvx/`

```
y = Y(:,4); t = taxis(y,12,1999);        % extract signal y_n from data file
                                         % the function taxis defines the t-axis
N=19; d=3; x1 = lpfilt(lpsm(N,d),y);     % LPSM filter
s=3; x2 = lpfilt(lprs(N,d,s),y);         % LPRS filter
M=8; r=2; x3 = svdenh(y,M,r);            % SVD enhancement
la=100; s=3; x4 = whsm(y,la,s);          % Whittaker-Henderson

s = 2; la = 1; N = length(y);            % Whittaker-Henderson with L1 criterion
D = diff(eye(N),s);                      % for x6, use s = 3

cvx_begin                                % use CVX package to solve the L1 problem
    variable x5(N)
    minimize( sum_square(y-x5) + la * norm(D*x5,1) )
cvx_end

figure; plot(t,y,'.', t,x1,'-'); figure; plot(t,y,'.', t,x2,'-');
figure; plot(t,y,'.', t,x3,'-'); figure; plot(t,y,'.', t,x4,'-');
figure; plot(t,y,'.', t,x5,'-'); figure; plot(t,y,'.', t,x6,'-');
```

All methods have comparable performance and can handle the end-point problem. □

The computational procedures implemented into the function `lprs` were outlined in Eq. (4.1.19). The related orthogonalized basis Q defined in Eq. (4.1.23) will be realized in terms of the Hahn orthogonal polynomials.

A direct consequence of upper-triangular nature of the matrix C in Eq. (4.2.23) is that the basis Q becomes an *eigenvector basis* for the matrix VW [123,99]. To see this, substitute $S = QR$ into (4.2.23),

$$VWQR = QRC \quad \Rightarrow \quad VWQ = Q\Lambda\,, \quad \Lambda = RCR^{-1} \tag{4.2.32}$$

Multiplying both sides by Q^TW and using the property $Q^TWQ = D$, we obtain:

$$Q^TWVWQ = Q^TWQ\Lambda = D\Lambda \tag{4.2.33}$$

Because R and C are both upper-triangular, so will be Λ and $D\Lambda$. But the left-hand side of (4.2.33) is a symmetric matrix, and so must be the right-hand side $D\Lambda$. This requires that $D\Lambda$ and hence Λ be a diagonal matrix, e.g., $\Lambda = \mathrm{diag}([\lambda_0, \lambda_1, \dots, \lambda_d])$. This means that the rth column of Q is an eigenvector:

$$VW\mathbf{q}_r = \lambda_r \mathbf{q}_r\,, \quad r = 0, 1, \dots, d \tag{4.2.34}$$

Choosing $d = N-1$ would produce all the eigenvectors of VW. In this case, we have $Q^{-1} = D^{-1}Q^TW$ and we obtain the decomposition:

$$VW = Q\Lambda Q^{-1} = Q\Lambda D^{-1}Q^TW \quad \Rightarrow \quad V = Q(\Lambda D^{-1})Q^T$$

We also find for the inverse of $V = D_s^TD_s$:

$$V^{-1} = WQ\Lambda^{-1}D^{-1}Q^TW$$

There exist [93–95] similar and efficient ways to calculate $V^{-1} = (D_s^TD_s)^{-1}$. The eigenvalues λ_r can be shown to be [123]:

$$\lambda_r = \frac{(2s+r)!}{r!} = \prod_{i=1}^{2s}(r+i)\,, \quad r = 0, 1, \dots, d \tag{4.2.35}$$

As we see in the next section, the rth column $q_r(n)$ of Q is a Hahn polynomial of degree r in n, and hence $W\mathbf{q}_r$, or component-wise, $w_nq_r(n)$, will be a polynomial of degree $2s+r$. Moreover, because of the zeros of w_n, the polynomial $f_n = w_nq_r(n)$ can be extended to be over the range $-M-s \le n \le M+s$. Using the same reasoning as in Eq. (4.2.19), it follows that (4.2.34) can be written as

$$(-1)^s\,\nabla^{2s}f_{n+s}^{\mathrm{ext}} = \lambda_r q_r(n)\,, \quad -M \le n \le M$$

Since this is valid as an identity in n, it is enough to match the highest powers of n from both sides, that is, n^r. Thus, on the two sides we have

$$f_{n+s}^{\mathrm{ext}} = w_{n+s}q_r(n+s) = \underbrace{(-1)^s[(n+s)^{2s} + \cdots]}_{w_{n+s}}\,\underbrace{[a_{rr}(n+s)^r + \cdots]}_{q_r(n+s)}\,, \quad \text{or,}$$

$$(-1)^sf_{n+s}^{\mathrm{ext}} = a_{rr}n^{2s+r} + \cdots\,, \quad \text{and also,} \quad q_r(n) = a_{rr}n^r + \cdots$$

where a_{rr} is the highest coefficient of $q_r(n)$ and the dots indicate lower powers of n. Dropping the a_{rr} constant, the eigenvector condition then becomes:

$$\nabla^{2s}[n^{2s+r} + \cdots] = \lambda_r[n^r + \cdots]$$

Each operation of ∇ on n^i lowers the power by one, that is, $\nabla(n^i) = i\,n^{i-1} + \cdots$, $\nabla^2(n^i) = i(i-1)n^{i-2} + \cdots$, $\nabla^3(n^i) = i(i-1)(i-2)n^{i-3} + \cdots$, etc. Thus, we have:

$$\nabla^{2s}[n^{2s+r} + \cdots] = (2s+r)(2s+r-1)(2s+r-2)\cdots(r+1)\,n^r + \cdots$$

which yields Eq. (4.2.35).

4.3 Hahn Orthogonal Polynomials

Starting with Chebyshev [104], the discrete Chebyshev/Gram polynomials have been used repeatedly in the least-squares polynomial fitting problem, LPSM filter design, and other applications [104–151]. Bromba and Ziegler [123] were the first to establish a similar connection between the Hahn orthogonal polynomials and the minimum-R_s problem. For a review of the Hahn polynomials, see Karlin and McGregor [113].

The Hahn polynomials $Q_r(x)$ of a discrete variable $x = 0, 1, 2, \dots, N-1$ and orders $r \le N-1$ satisfy a weighted orthogonality property of the form:

$$\sum_{x=0}^{N-1} w(x)Q_r(x)Q_m(x) = D_r\delta_{rm}\,, \quad r, m = 0, 1, \dots, N-1$$

where the weighting function $w(x)$ depends on two parameters α, β and is defined up to a normalization constant as follows:

$$w(x) = \frac{(\alpha+x)!}{x!}\cdot\frac{(\beta+N-1-x)!}{(N-1-x)!}\,, \quad x = 0, 1, \dots, N-1 \tag{4.3.1}$$

The length N can be even or odd, but here we will consider only the odd case and set as usual $N = 2M+1$. The interval $[0, N-1]$ can be mapped onto the symmetric interval $[-M, M]$ by making the change of variables $x = n + M$, with $-M \le n \le M$. Then, the weighting function becomes,

$$w(n) = \frac{(\alpha + M + n)!}{(M + n)!} \cdot \frac{(\beta + M - n)!}{(M - n)!}, \quad -M \le n \le M \tag{4.3.2}$$

Defining $q_r(n) = Q_r(x)\big|_{x=n+M}$, the orthogonality property now reads:

$$\boxed{\sum_{n=-M}^{M} w(n) q_r(n) q_m(n) = D_r \delta_{rm}}, \quad r, m = 0, 1, \dots, N-1 \tag{4.3.3}$$

The minimum-R_s problem corresponds to the particular choice $\alpha = \beta = s$. In this case, the weighting function $w(n)$ reduces to the Henderson weights of Eq. (4.2.21):

$$w(n) = \frac{(s + M + n)!}{(M + n)!} \cdot \frac{(s + M - n)!}{(M - n)!} = \prod_{i=1}^{s} (M + n + i) \cdot \prod_{i=1}^{s} (M - n + i), \quad \text{or,}$$

$$w(n) = \prod_{i=1}^{s} \left[(M + i)^2 - n^2\right], \quad -M \le n \le M \tag{4.3.4}$$

For $s = 0$, the weights reduce to $w(n) = 1$ corresponding to the discrete Chebyshev/Gram polynomials. Because the weights are unity, the Chebyshev/Gram polynomials can be regarded as discrete-time versions of the Legendre polynomials. In fact, they tend to the latter in the limit $N \to \infty$ [133]. Similarly, the Hahn polynomials may be regarded as discrete versions of the Jacobi polynomials. At the opposite limit, $s \to \infty$, the Hahn polynomials tend to the *Krawtchouk* polynomials [133], which are discrete versions of the Hermite polynomials [130]. We review Krawtchouk polynomials and their application to the design of maximally flat filters in Sec. 4.4.

In general, the Hahn polynomials are given in terms of the hypergeometric function ${}_3F_2(a_1, a_2, a_3; b_1, b_2; z)$. For $\alpha = \beta = s$, they take the following explicit form:

$$\boxed{q_r(n) = Q_r(x) = \sum_{k=0}^{r} a_{rk} x^{[k]}\Big|_{x=n+M} = \sum_{k=0}^{r} a_{rk} (n + M)^{[k]}}, \quad -M \le n \le M \tag{4.3.5}$$

where $x^{[k]}$ denotes the falling-factorial power,

$$x^{[k]} = x(x-1)\cdots(x-k+1) = \frac{x!}{(x-k)!} = \frac{\Gamma(x+1)}{\Gamma(x-k+1)} \tag{4.3.6}$$

The polynomial coefficients are:

$$\boxed{a_{rk} = (-1)^k \prod_{m=1}^{k} \left[\frac{(r - m + 1)(2s + r + m)}{(N - m)(s + m)m}\right]}, \quad k = 0, 1, \dots, r \tag{4.3.7}$$

where $a_{r0} = 1$. Expanding the product we have:

$$a_{rk} = \frac{(-1)^k r(r-1)\cdots(r-k+1) \cdot (2s + r + 1)(2s + r + 2)\cdots(2s + r + k)}{(N-1)(N-2)\cdots(N-k) \cdot (s+1)(s+2)\cdots(s+k) \cdot k!} \tag{4.3.8}$$

The polynomials satisfy the symmetry property ,

$$q_r(-n) = (-1)^r q_r(n) \tag{4.3.9}$$

The orthogonality property (4.3.3) is satisfied with the following values of D_r:

$$\boxed{D_r = \frac{(s!)^2}{(2M)!} \cdot \frac{r!\,(2M - r)!}{(2M)!} \cdot \frac{(2s + r + 1)(2s + r + 2)\cdots(2s + r + N)}{2s + 2r + 1}} \tag{4.3.10}$$

For minimum-R_s filter design with polynomial order $d \le N - 1$, only polynomials up to order d are needed, that is, $q_r(n)$, $r = 0, 1, \dots, d$. Arranging these as the columns of the $N \times (d+1)$ matrix $Q = [\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_d]$, the orthogonality property can be expressed as $Q^T W Q = D$, where $D = \text{diag}\big([D_0, D_1, \dots, D_d]\big)$.

The relationship to the monomial basis $S = [\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_d]$ is through an upper-triangular invertible matrix R, that is, $S = QR$. This can be justified by noting that the power series of $q_r(n)$ in n is a linear combination of the monomials $s_i(n) = n^i$ for $i = 0, 1, \dots, r$. In fact, R can be easily constructed from the Hahn coefficients a_{rk} and the Stirling numbers.

Thus, the construction of the minimum-R_s filters outlined in Eq. (4.1.23) is explicitly realized by the Hahn polynomial basis matrix Q:

$$B = W Q D^{-1} Q^T \tag{4.3.11}$$

or, component-wise,

$$b_m(n) = B_{nm} = w(n) \sum_{r=0}^{d} \frac{q_r(n) q_r(m)}{D_r}, \quad -M \le n, m \le M \tag{4.3.12}$$

A more direct derivation of (4.3.11) is to perform the local polynomial fit in the Q-basis. The desired degree-d polynomial can be expanded in the linear combination:

$$\hat{y}_m = \sum_{i=0}^{d} c_i m^i = \sum_{r=0}^{d} a_r q_r(m) \quad \Rightarrow \quad \hat{\mathbf{y}} = S\mathbf{c} = Q\mathbf{a}$$

Then, minimize the weighted performance index with respect to $\mathbf{a}$:

$$\mathcal{J} = (\mathbf{y} - Q\mathbf{a})^T W (\mathbf{y} - Q\mathbf{a}) = \min$$

Using the condition $Q^T W Q = D$, the solution leads to the same B:

$$\mathbf{a} = D^{-1} Q^T W \mathbf{y} \quad \Rightarrow \quad \hat{\mathbf{y}} = Q\mathbf{a} = Q D^{-1} Q^T W \mathbf{y} = B^T \mathbf{y} \tag{4.3.13}$$

The computation of the basis Q is facilitated by the following MATLAB functions. We note first that the falling factorial powers are related to ordinary powers by the *Stirling numbers* of the first and second kind:

$$x^{[k]} = \sum_{i=0}^{k} S_1(k, i) x^i \quad \Leftrightarrow \quad x^k = \sum_{i=0}^{k} S_2(k, i) x^{[i]} \tag{4.3.14}$$

These numbers may be arranged into lower-triangular matrices S_1 and S_2, which are inverses of each other. For example, we have for $k = 0, 1, 2, 3$:

$$\begin{bmatrix} x^{[0]} \\ x^{[1]} \\ x^{[2]} \\ x^{[3]} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 2 & -3 & 1 \end{bmatrix} \begin{bmatrix} x^0 \\ x^1 \\ x^2 \\ x^3 \end{bmatrix} \quad \Leftrightarrow \quad \begin{bmatrix} x^0 \\ x^1 \\ x^2 \\ x^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} x^{[0]} \\ x^{[1]} \\ x^{[2]} \\ x^{[3]} \end{bmatrix}$$

$$S_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 2 & -3 & 1 \end{bmatrix}, \qquad S_2 = S_1^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 3 & 1 \end{bmatrix}$$

The MATLAB function `stirling` generates these matrices up to a desired order:

```
S = stirling(d,kind);     % Stirling numbers up to order d of kind = 1,2
```

A polynomial can be expressed in falling factorial powers or in ordinary powers. The corresponding coefficient vectors are related by the Stirling numbers:

$$P(x) = \sum_{k=0}^{d} a_k x^{[k]} = \sum_{i=0}^{d} c_i x^i \quad \Rightarrow \quad \mathbf{c} = S_1^T \mathbf{a}\,, \quad \mathbf{a} = S_2^T \mathbf{c}$$

The function `polval` allows the evaluation of a polynomial in falling (or rising) factorial powers or in ordinary powers at any vector of x values:

```
P = polval(a,z,type);     % polynomial evaluation in factorial powers
```

The functions `hahncoeff`, `hahnpol`, and `hahnbasis` allow the calculation of the Hahn coefficients (4.3.7), the evaluation of the polynomial $Q_r(x)$ at any vector of x's, and the construction of the Hahn basis $Q = [\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_d]$:

```
[a,c] = hahncoeff(N,r,s);      % Hahn polynomial coefficients a_rk
Q = hahnpol(N,r,s,x);          % evaluate Hahn polynomial Q_r(x)
[Q,D,L] = hahnbasis(N,d,s);    % Hahn basis Q = [q_0,q_1,...,q_d]
```

Like all orthogonal polynomials, the Hahn polynomials satisfy a three-term recurrence relation of the form:

$$nq_r(n) = \alpha_r q_{r+1}(n) + \beta_r q_r(n) + \gamma_r q_{r-1}(n) \tag{4.3.15}$$

that starts with $r = 0$ and $q_{-1}(n) = 0$ and ends at $r = N-2$. The recurrence relation is a direct consequence of the property (which follows from (4.3.3)) that the order-r polynomial $q_r(n)$ is orthogonal to *every* polynomial of degree strictly less than r. Let us denote the weighted inner product by

$$(a, b) = \sum_{n=-M}^{M} w(n) a(n) b(n) \tag{4.3.16}$$

Then, since the polynomial $nq_r(n)$ has degree $r+1$, it can be expanded as a linear combination of the polynomials $q_i(n)$ up to degree $r+1$:

$$nq_r(n) = \sum_{i=0}^{r+1} c_i q_i(n)$$

The coefficients are determined using the orthogonality property by

$$(nq_r, q_i) = \sum_{j=0}^{r+1} c_j (q_j, q_i) = \sum_{j=0}^{r+1} c_j D_i \delta_{ij} = D_i c_i \quad \Rightarrow \quad c_i = \frac{(nq_r, q_i)}{D_i} \tag{4.3.17}$$

This implies that $c_i = 0$ for $i \le r-2$, therefore, only the terms $i = r+1, r, r-1$ will survive, which is the recurrence relation. Indeed, we note that $(nq_r, q_i) = (q_r, nq_i)$ and that $nq_i(n)$ has degree $(i+1)$. Therefore, as long as $i+1 < r$, or, $i \le r-2$, this inner product will be zero. It follows from (4.3.17) that:

$$\alpha_r = \frac{(nq_r, q_{r+1})}{D_{r+1}}, \quad \beta_r = \frac{(nq_r, q_r)}{D_r}, \quad \gamma_r = \frac{(nq_r, q_{r-1})}{D_{r-1}} \tag{4.3.18}$$

Because the weights $w(n)$ are symmetric, $w(n) = w(-n)$, and the polynomials satisfy, $q_r(-n) = (-1)^r q_r(n)$, it follows immediately that $\beta_r = 0$. The coefficient γ_r can be related to α_{r-1} by noting that

$$\alpha_{r-1} = \frac{(nq_{r-1}, q_r)}{D_r} = \frac{(nq_r, q_{r-1})}{D_r} \quad \Rightarrow \quad (nq_r, q_{r-1}) = D_r \alpha_{r-1}\,, \quad \text{and hence,}$$

$$\gamma_r = \frac{(nq_r, q_{r-1})}{D_{r-1}} = \frac{D_r \alpha_{r-1}}{D_{r-1}} \tag{4.3.19}$$

Moreover, α_r is related to the leading coefficients a_{rr} of the $q_r(n)$ polynomial. From the definition (4.3.5), we can write

$$q_r(n) = a_{rr} n^r + p_{r-1}(n)\,, \quad q_{r+1}(n) = a_{r+1,r+1} n^{r+1} + p_r(n)$$

where $p_{r-1}(n)$ and $p_r(n)$ are polynomials of degree $r-1$ and r, respectively. Since $D_{r+1} = (q_{r+1}, q_{r+1})$, we have,

$$\alpha_r = \frac{(nq_r, q_{r+1})}{(q_{r+1}, q_{r+1})} = \frac{(a_{rr} n^{r+1} + np_{r-1},\, q_{r+1})}{(a_{r+1,r+1} n^{r+1} + p_r,\, q_{r+1})} = \frac{a_{rr}(n^{r+1},\, q_{r+1})}{a_{r+1,r+1}(n^{r+1},\, q_{r+1})} = \frac{a_{rr}}{a_{r+1,r+1}}$$

where we used the orthogonality of $q_{r+1}(n)$ with $np_{r-1}(n)$ and $p_r(n)$, both of which have order r. Thus,

$$\alpha_r = \frac{a_{rr}}{a_{r+1,r+1}} \tag{4.3.20}$$

Using Eqs. (4.3.7) and (4.3.10), the expressions for α_r and γ_r simplify into:

$$\alpha_r = -\frac{(2M - r)(2s + r + 1)}{2(2s + 2r + 1)}, \quad \gamma_r = -\frac{r(2M + 2s + r + 1)}{2(2s + 2r + 1)} \tag{4.3.21}$$

These satisfy the constraint $\alpha_r + \gamma_r = -M$, which follows from the recurrence relation and the conditions $q_r(-M) = a_{r0} = 1$ for all r. Next, we derive the Christoffel-Darboux identity which allows the simplification of the sum in (4.3.12). Setting $\beta_r = 0$, replacing $\gamma_r = \alpha_{r-1}D_r/D_{r-1}$ and dividing by D_r, the recurrence relation reads:

$$\frac{nq_r(n)}{D_r} = \frac{\alpha_r}{D_r}q_{r+1}(n) + \frac{\alpha_{r-1}}{D_{r-1}}q_{r-1}(n) \tag{4.3.22}$$

Multiplying by $q_r(m)$, interchanging the roles of n, m, and subtracting, we obtain:

$$\begin{aligned}
\frac{nq_r(n)q_r(m)}{D_r} &= \frac{\alpha_r}{D_r}q_{r+1}(n)q_r(m) + \frac{\alpha_{r-1}}{D_{r-1}}q_{r-1}(n)q_r(m)\\
\frac{mq_r(m)q_r(n)}{D_r} &= \frac{\alpha_r}{D_r}q_{r+1}(m)q_r(n) + \frac{\alpha_{r-1}}{D_{r-1}}q_{r-1}(m)q_r(n)\\
\frac{(n-m)q_r(n)q_r(m)}{D_r} &= \frac{\alpha_r}{D_r}\big[q_{r+1}(n)q_r(m) - q_r(n)q_{r+1}(m)\big] -\\
&\quad - \frac{\alpha_{r-1}}{D_{r-1}}\big[q_r(n)q_{r-1}(m) - q_{r-1}(n)q_r(m)\big]
\end{aligned}$$

Summing up over r, and using $q_{-1}(n) = 0$, the successive terms on the right-hand side cancel except for the last one, resulting in the Christoffel-Darboux identity:

$$(n-m)\sum_{r=0}^{d}\frac{q_r(n)q_r(m)}{D_r} = \frac{\alpha_d}{D_d}\big[q_{d+1}(n)q_d(m) - q_d(n)q_{d+1}(m)\big]\,, \quad \text{or,}$$

$$\boxed{\sum_{r=0}^{d}\frac{q_r(n)q_r(m)}{D_r} = \frac{\alpha_d}{D_d}\,\frac{q_{d+1}(n)q_d(m) - q_d(n)q_{d+1}(m)}{n-m}} \tag{4.3.23}$$

Using this identity into the filter equations (4.3.12), we find

$$\boxed{b_m(n) = w(n)\,\frac{\alpha_d}{D_d}\,\frac{q_{d+1}(n)q_d(m) - q_d(n)q_{d+1}(m)}{n-m}} \tag{4.3.24}$$

This is valid for $-M \le n, m \le M$ and for orders $0 \le d \le N-2$. At $n = m$, the numerator vanishes, so that the numerator and denominator have a common factor $n - m$, which cancels resulting in a polynomial of degree d in n and m. In particular, the central Henderson filters are:

$$\boxed{b_0(n) = w(n)\,\frac{\alpha_d}{D_d}\,\frac{q_{d+1}(n)q_d(0) - q_d(n)q_{d+1}(0)}{n}} \tag{4.3.25}$$

where either $q_d(0)$ or $q_{d+1}(0)$ is zero depending on whether d is odd or even. In fact for the two successive values $d = 2r$ and $d = 2r+1$, while the asymmetric filters $b_m(n)$ are different, the central filters are the same and given by:

$$b_0(n) = \frac{\alpha_{2r}}{D_{2r}}q_{2r}(0)\,\frac{q_{2r+1}(n)}{n} = -\frac{\alpha_{2r+1}}{D_{2r+1}}q_{2r+2}(0)\,\frac{q_{2r+1}(n)}{n} \tag{4.3.26}$$

the equality of the coefficients following by setting $d = 2r+1$ and $n = 0$ in Eq. (4.3.22).

Next, we derive explicit formulas for some specific cases. The first few Hahn polynomials of orders $d = 0, 1, 2, 3, 4, 5$ and arbitrary M and s are, for $-M \le n \le M$:

$$\begin{aligned}
q_0(n) &= 1\\
q_1(n) &= -\frac{n}{M}\\
q_2(n) &= \frac{(2s+3)n^2 - M(M+s+1)}{M(2M-1)(s+1)}\\
q_3(n) &= -\frac{(2s+5)n^3 - \big[3M^2 + (s+1)(3M-1)\big]n}{M(M-1)(2M-1)(s+1)}\\
q_4(n) &= \frac{(2s+5)(2s+7)n^4 - (2s+5)\big(6M^2 + 6(s+1)M - 4s-5\big)n^2}{M(M-1)(2M-1)(2M-3)(s+1)(s+2)}\\
&\quad + \frac{3M(M-1)(s+M+1)(s+M+2)}{M(M-1)(2M-1)(2M-3)(s+1)(s+2)}\\
q_5(n) &= -\frac{(2s+7)(2s+9)n^5 - 5(2s+7)\big(2M^2 + 2(s+1)M - 2s-3\big)n^3}{M(M-1)(M-2)(2M-1)(2M-3)(s+1)(s+2)}\\
&\quad - \frac{\big[15M^4 + 30(s+1)M^3 + 5(3s^3+s-7)M^2 - (s+1)(s+2)(25M-6)\big]n}{M(M-1)(M-2)(2M-1)(2M-3)(s+1)(s+2)}
\end{aligned} \tag{4.3.27}$$

They are normalized such that $q_r(-M) = 1$. Setting $s = 0$, we obtain the corresponding *discrete Chebyshev/Gram* polynomials:

$$\begin{aligned}
q_0(n) &= 1\\
q_1(n) &= -\frac{n}{M}\\
q_2(n) &= \frac{3n^2 - M(M+1)}{M(2M-1)}\\
q_3(n) &= -\frac{5n^3 - (3M^2+3M-1)n}{M(M-1)(2M-1)}\\
q_4(n) &= \frac{35n^4 - 5(6M^2+6M-5)n^2 + 3M(M^2-1)(M+2)}{2M(M-1)(2M-1)(2M-3)}\\
q_5(n) &= -\frac{63n^5 - 35(2M^2+2M-3)n^3 + (15M^4+30M^3-35M^2-50M+12)n}{2M(M-1)(M-2)(2M-1)(2M-3)}
\end{aligned} \tag{4.3.28}$$

The central Henderson filters for the cases $d = 0, 1$, $d = 2, 3$, and $d = 4, 5$ are as follows for general M and s. For $d = 0, 1$:

$$b_0(n) = \frac{(2s+1)!\,(2M)!}{(s!)^2\,(2M+2s+1)!}\,w(n) \tag{4.3.29}$$

where $w(n)$ is given by Eq. (4.3.4). For $d = 2, 3$, we have:

$$b_0(n) = \frac{(M+s+1)(2s+3)!\,(2M)!\big(3M^2 + (s+1)(3M-1) - (2s+5)n^2\big)}{(2M-1)(s!)^2\,(2M+2s+3)!}\,w(n) \tag{4.3.30}$$

This generalizes Henderson's ideal formula (4.2.29) to arbitrary s. For $s = 1, 2$, it simplifies into:

$$s=1,\quad b_0(n) = \frac{15\,(3M^2+6M-2-7n^2)\,w_1(n)}{2\,(M+1)\,(2M+3)\,(2M+5)\,(4M^2-1)}$$

$$s=2,\quad b_0(n) = \frac{105\,(M^2+3M-1-3n^2)\,w_2(n)}{2\,(M+1)\,(M+2)\,(2M+3)\,(2M+5)\,(2M+7)\,(4M^2-1)}$$

where $w_1(n)$ and $w_2(n)$ correspond to (4.3.4) with $s = 1$ and $s = 2$. The case $s = 0$ is, of course, the same as Eq. (3.3.17). For the case $d = 4, 5$, we find:

$$\begin{aligned} b_0(n) = {}& \frac{(M+s+1)\,(M+s+2)\,(2s+5)!\,(2M)!}{2\,(2M-1)\,(2M-3)\,\big((s+2)!\big)^2\,(2M+2s+5)!}\cdot w(n)\cdot \\ &\cdot\Big[(2s+7)\,(2s+9)\,n^4 - 5\,(2s+7)\,\big(2M^2+2\,(s+1)M-2s-3\big)\,n^2 + \\ &+ 15M^4 + 30\,(s+1)M^3 + 5\,(3s^2+s-7)M^2 + (s+1)\,(s+2)\,(25M-6)\Big] \end{aligned} \tag{4.3.31}$$

Eqs. (4.3.29)–(4.3.31), as well as the case $d = 6, 7$, have been implemented into the MATLAB function `lprs2`, with usage:

```
b0 = lprs2(N,d,s);
```
% exact forms of the Henderson filters $b_0(n)$ for $0 \le d \le 6$

The asymmetric interpolation filters $b_t(n)$ can be obtained by replacing the discrete variable m by t in Eqs. (4.3.12) and (4.3.24):

$$b_t(n) = w(n)\sum_{r=0}^{d}\frac{q_r(n)\,q_r(t)}{D_r} = w(n)\,\frac{\alpha_d}{D_d}\,\frac{q_{d+1}(n)\,q_d(t)-q_d(n)\,q_{d+1}(t)}{n-t} \tag{4.3.32}$$

Some specific cases are as follows. For $d = 0$, we have:

$$b_t(n) = \frac{(2s+1)!\,(2M)!}{(s!)^2\,(2M+2s+1)!}\,w(n) \tag{4.3.33}$$

For $d = 1$,

$$b_t(n) = \frac{4\,(2s+1)!\,(2M-1)!}{(s!)^2\,(2M+2s+2)!}\,w(n)\,\big[M^2+(s+1)M+(2s+3)\,nt\big] \tag{4.3.34}$$

For $d = 2$:

$$\begin{aligned} b_t(n) = {}& \frac{4\,(2s+1)!\,(2M-1)!}{(s!)^2\,(2M+2s+2)!}\,w(n)\Big[M\,(M+s+1)\,\big[3M^2+3\,(s+1)M-s-1\big] \\ &+ (s+1)\,(2M-1)\,(2M+2s+3)\,nt - M\,(M+s+1)\,(2s+5)\,(n^2+t^2) \\ &+ (s+1)\,(2M-1)\,(2M+2s+3)\,n^2t^2\Big] \end{aligned} \tag{4.3.35}$$

The corresponding predictive differentiation filters are obtained by differentiating with respect to t.

The above closed-form expressions were obtained with the following simple Maple procedures that define the Hahn coefficients a_{rk}, the Hahn polynomials $q_r(n)$ and their norms D_r, and the interpolation filters $b_t(n)$:


```
factpow := proc(x,k) product((x-m), m=0..k-1); end proc;

a := proc(M,r,s,k)
        (-1)^k * product((r-m+1)*(2*s+r+m)/(2*M+1-m)/(s+m)/m, m=1..k);
     end proc;

Q := proc(M,r,s,n) if r=0 then 1; else
        sum(a(M,r,s,k)*factpow(n+M,k), k=0..r);
     end if; end proc;

Dr := proc(M,r,s) GAMMA(s+1)^2 * GAMMA(r+1) * GAMMA(2*M+1-r)
        * product(2*s+r+i, i=1..(2*M+1)) / GAMMA(2*M+1)^2 / (2*s+2*r+1);
      end proc;

B := proc(M,d,s,n,t)
        sum(Q(M,r,s,n)/Dr(M,r,s)*Q(M,r,s,t), r=0..d);
     end proc;
```

where `factpow` defines the falling-factorial powers, and it is understood that the result from the procedure `B(M,d,s,n,t)` must be multiplied by the Henderson weights $w(n)$.

There are other useful choices for the weighting function $w(n)$, such as binomial, which are similar to gaussian weights and lead to the Krawtchouk orthogonal polynomials, or exponentially decaying $w(n) = \lambda^n$, with $n \ge 0$ and $0 < \lambda < 1$, leading to the discrete Laguerre polynomials [135,136] and exponential smoothers. However, these choices do not have an equivalent minimum-NRR characterization. Even so, the smoothing filters are efficiently computed in the orthogonal polynomial basis by:

$$B = WS(S^TWS)^{-1}S^T = WQD^{-1}Q^T\,,\quad Q^TWQ = D \tag{4.3.36}$$

4.4 *Maximally-Flat Filters and Krawtchouk Polynomials*

Greville [84] has shown that in the limit $s \to \infty$ the minimum-R_s filters tend to maximally flat FIR filters that satisfy the usual flatness constraints at dc, that is, $B^{(i)}(\omega)\big|_{\omega=0} = \delta(i)$, for $i = 0, 1, \dots, d$, but also have monotonically decreasing magnitude responses and satisfy $(2M-d)$ additional flatness constraints at the Nyquist frequency, $\omega = \pi$. They are identical to the well-known maximally flat filters introduced by Herrmann [174]. Bromba and Ziegler [123,178] have shown that their impulse responses are given in terms of the Krawtchouk orthogonal polynomials [109,130,133]. Meer and Weiss [140] have derived the corresponding differentiation filters based on the Krawtchouk polynomials for application to images. Here, we look briefly at these properties.

The Krawtchouk polynomials are characterized by a parameter p such that $0 < p < 1$ and are defined over the symmetric interval $-M \le n \le M$ by [133]

$$\bar{q}_r(n) = \sum_{k=0}^{r}\frac{(-1)^k r\,(r-1)\cdots(r-k+1)\,p^{-k}}{(N-1)\,(N-2)\cdots(N-k)\cdot k!}\,(n+M)^{[k]} \tag{4.4.1}$$

where $N = 2M+1$ and $r = 0, 1, \dots, N-1$. They satisfy the orthogonality property,

$$\sum_{n=-M}^{M}\bar{w}(n)\,\bar{q}_r(n)\,\bar{q}_m(n) = \bar{D}_r\delta_{rm} \tag{4.4.2}$$

with the following binomial weighting function and norms, where $q = 1 - p$:

$$\begin{aligned} \bar{w}(n) &= \binom{2M}{M+n} p^{M+n} q^{M-n} = \frac{(2M)!}{2^{2M}(M+n)!\,(M-n)!}\, p^{M+n} q^{M-n} \\ \bar{D}_r &= \frac{r!\,(2M-r)!}{(2M)!}\,\frac{q^r}{p^r} \end{aligned} \tag{4.4.3}$$

In the limit $s \to \infty$, the Hahn polynomials tend to the special Krawtchouk polynomials with the parameter $p = q = 1/2$. To see this, we note that the Hahn polynomials are normalized such that $q_r(-M) = 1$, and we expect that they would have a straightforward limit as $s \to \infty$. Indeed, it is evident that the limit of the Hahn coefficients (4.3.8) is

$$\bar{a}_{rk} = \lim_{s\to\infty} a_{rk} = \frac{(-1)^k r(r-1)\cdots(r-k+1)\cdot 2^k}{(N-1)(N-2)\cdots(N-k)\cdot k!} \tag{4.4.4}$$

and therefore, the Hahn polynomials will tend to

$$\bar{q}_r(n) = \sum_{k=0}^{r} \frac{(-1)^k r(r-1)\cdots(r-k+1)\cdot 2^k}{(N-1)(N-2)\cdots(N-k)\cdot k!}\,(n+M)^{[k]} \tag{4.4.5}$$

which are recognized as a special case of (4.4.1) with $p = 1/2$. The Henderson weights (4.3.4) and norms (4.3.10) diverge as $s \to \infty$, but we may normalize them by a common factor, such as $s^{2M}(s!)^2$, so that they will converge. The limits of the rescaled weights and norms are:

$$\begin{aligned} \bar{w}(n) &= \lim_{s\to\infty}\left[\frac{(2M)!\,w(n)}{2^{2M}s^{2M}(s!)^2}\right] = \lim_{s\to\infty}\left[\frac{(2M)!\,(s+M+n)!\,(s+M-n)!}{2^{2M}s^{2M}(s!)^2\,(M+n)!\,(M-n)!}\right] \\ \bar{D}_r &= \lim_{s\to\infty}\left[\frac{(2M)!\,D_r}{2^{2M}s^{2M}(s!)^2}\right] \\ &= \lim_{s\to\infty}\left[\frac{r!\,(2M-r)!}{(2M)!}\cdot\frac{(2s+r+1)(2s+r+2)\cdots(2s+r+N)}{2^{2M}s^{2M}(2s+2r+1)}\right] \end{aligned}$$

They are easily seen to lead to Eqs. (4.4.3) with $p = 1/2$, that is,

$$\begin{aligned} \bar{w}(n) &= \frac{1}{2^{2M}}\binom{2M}{M+n} = \frac{(2M)!}{2^{2M}(M+n)!\,(M-n)!} \\ \bar{D}_r &= \frac{r!\,(2M-r)!}{(2M)!} \end{aligned} \tag{4.4.6}$$

The first few of the Krawtchouk polynomials are:

$$\begin{aligned} \bar{q}_0(n) &= 1 \\ \bar{q}_1(n) &= -\frac{n}{M} \\ \bar{q}_2(n) &= \frac{2n^2 - M}{M(2M-1)} \\ \bar{q}_3(n) &= -\frac{2n^3 - (3M-1)n}{M(M-1)(2M-1)} \\ \bar{q}_4(n) &= \frac{4n^4 - (12M-8)n^2 + 3M(M-1)}{M(M-1)(2M-1)(2M-3)} \\ \bar{q}_5(n) &= -\frac{4n^5 - 20(M-1)n^3 + (15M^2-25M+6)n}{M(M-1)(M-2)(2M-1)(2M-3)} \end{aligned} \tag{4.4.7}$$

These polynomials satisfy the three-term recurrence relation:

$$n\bar{q}_r(n) = \bar{\alpha}_r\bar{q}_{r+1}(n) + \bar{\gamma}_r\bar{q}_{r-1}(n)\,, \quad \bar{\alpha}_r = -\frac{2M-r}{2}\,, \quad \bar{\gamma}_r = -\frac{r}{2} \tag{4.4.8}$$

with the coefficients $\bar{\alpha}_r, \bar{\gamma}_r$ obtained from Eq. (4.3.21) in the limit $s \to \infty$. The three-term relations lead to the usual Christoffel-Darboux identity from which we may obtain the asymmetric predictive filters:

$$\bar{b}_t(n) = \bar{w}(n)\sum_{r=0}^{d}\frac{\bar{q}_r(n)\bar{q}_r(t)}{\bar{D}_r} = \bar{w}(n)\,\frac{\bar{\alpha}_d}{\bar{D}_d}\,\frac{\bar{q}_{d+1}(n)\bar{q}_d(t) - \bar{q}_d(n)\bar{q}_{d+1}(t)}{n-t} \tag{4.4.9}$$

Differentiation with respect to t gives the corresponding predictive differentiation filters. Some examples are as follows. For $d = 0$ and $d = 1$, we have, respectively

$$\bar{b}_t(n) = \bar{w}(n)\,, \quad \bar{b}_t(n) = \bar{w}(n)\,\frac{2nt+M}{M} \tag{4.4.10}$$

For $d = 2$, the smoothing and first-order differentiation filters are:

$$\begin{aligned} \bar{b}_t(n) &= \bar{w}(n)\,\frac{4n^2t^2 - 2M(n^2+t^2) + 2(2M-1)nt + M(3M-1)}{M(2M-1)} \\ \dot{\bar{b}}_t(n) &= \bar{w}(n)\,\frac{2(2M-1)n - 4Mt + 8n^2t}{M(2M-1)} \end{aligned} \tag{4.4.11}$$

and setting $t = 0$, the central filters simplify into:

$$\bar{b}_0(n) = \bar{w}(n)\,\frac{3M-1-2n^2}{2M-1}\,, \quad \dot{\bar{b}}_0(n) = \bar{w}(n)\,\frac{2n}{M} \tag{4.4.12}$$

For $d = 3$, we have:

$$\begin{aligned} \bar{b}_t(n) = \frac{\bar{w}(n)}{3M(M-1)(2M-1)}\Big[&8n^3t^3 - 4(3M-1)(n^3t + nt^3) + 12(M-1)n^2t^2 \\ &- 6M(M-1)(n^2+t^2) + (30M^2-30M+8)nt - 3M(M-1)(3M-1)\Big] \end{aligned} \tag{4.4.13}$$

As expected, setting $t = 0$ produces the same result as the $d = 2$ case. Numerically, the smoothing and differentiation filters can be calculated by passing the Krawtchouk weights $\bar{w}(n)$ into the functions `lpsm`, `lpdiff`, and `lpinterp`:

```
W  = diag(hend(N,inf));    % Krawtchouk weights
B  = lpsm(N,d,W);          % smoothing filters
Bi = lpdiff(N,d,i,W);      % i-th derivative filters
b  = lpinterp(N,d,t,i,W);  % interpolation filters b_t
```

The function $\text{hend}(N, s)$, with $s = \infty$, calculates the Krawtchouk weights of Eq. (4.4.6). In turn, the filter matrices B or $B^{(i)}$ may be passed into the filtering function `lpfilt`. Alternatively, one can call `lprs` with $s = \infty$:

```
B = lprs(N,d,inf);    % LPRS with Krawtchouk weights, maximally-flat filters
```

It is well-known [84,174–187] that the maximally-flat FIR filters of length $N = 2M+1$ and polynomial order $d = 2r + 1$ have frequency responses given by the following equivalent expressions:

$$\begin{aligned} B_0(\omega) &= \sum_{i=0}^{r}\binom{M}{i}x^i(1-x)^{M-i} = 1 - \sum_{i=r+1}^{M}\binom{M}{i}x^i(1-x)^{M-i} \\ &= (1-x)^{M-r}\sum_{i=0}^{r}\binom{M-r+i-1}{i}x^i\,, \qquad \text{where } x = \sin^2\left(\frac{\omega}{2}\right) \end{aligned} \tag{4.4.14}$$

Near $\omega \simeq 0$ and near $\omega \simeq \pi$, the second and third expressions have the following expansions that exhibit the desired flatness constraints [123]:

$$\begin{aligned} \omega \simeq 0 \quad &\Rightarrow \quad B_0(\omega) \simeq 1 - (\text{const.})\,\omega^{2r+2} = 1 - (\text{const.})\,\omega^{d+1} \\ \omega \simeq \pi \quad &\Rightarrow \quad B_0(\omega) \simeq (\text{const.})\,(\omega-\pi)^{2M-2r} = (\text{const.})\,(\omega-\pi)^{2M-d+1} \end{aligned} \tag{4.4.15}$$

The first implies the flatness constraints at dc, $B_0^{(i)}(0) = \delta(i)$, for $i = 0, 1, \dots, d$, and the second, the flatness constraints at Nyquist, $B_0^{(i)}(\pi) = 0$, for $i = 0, 1, \dots, 2M-d$.

Example 4.4.1: For $d = 2$ or $r = 1$, the z-transform of $b_0(n)$ in Eq. (4.4.12) can be calculated explicitly resulting in:

$$B_0(z) = \left[\frac{(1+z^{-1})(1+z)}{4}\right]^{M-1}\frac{1}{4}\big[2(M+1) - (M-1)(z+z^{-1})\big]$$

With $z = e^{j\omega}$ we may write

$$x = \sin^2\left(\frac{\omega}{2}\right) = \frac{(1-z^{-1})(1-z)}{4} = \frac{2-z-z^{-1}}{4} \quad \Rightarrow \quad \frac{z+z^{-1}}{4} = \frac{1}{2} - x$$

$$1 - x = \cos^2\left(\frac{\omega}{2}\right) = \frac{(1+z^{-1})(1+z)}{4}$$

Thus, we may express $B_0(z)$ in terms of the variable x:

$$B_0(z) = (1-x)^{M-1}\big[1 + (M-1)x\big]$$

which corresponds to Eq. (4.4.14) for $r = 1$. □

Example 4.4.2: Fig. 4.4.1 shows the frequency responses $B_0(\omega)$ for the values $N = 13$, $r = 2$, ($d = 4, 5$), and the smoothness parameter values: $s = 3$, $s = 6$, $s = 9$, and $s = \infty$.

Because $b_0(n)$ is symmetric about $n = 0$, the quantities $B_0(\omega)$ are real-valued. In the limit $s \to \infty$, the response becomes positive and monotonically decreasing. The following MATLAB code illustrates the generation of the bottom two graphs and verifies Eq. (4.4.14):

```
N=13; r=2; d = 2*r+1; M = floor(N/2);

B = lprs(N,d,9);   b9   = B(:,M+1);          % LPRS filter with s = 9
B = lprs(N,d,inf); binf = B(:,M+1);          % LPRS with Krawtchouk weights

f = linspace(0,1,1001); w = pi*f; x = sin(w/2).^2;
B9   = real(exp(j*w*M) .* freqz(b9,1,w));       % frequency responses
Binf = real(exp(j*w*M) .* freqz(binf,1,w));

Bth = 0;
for i=0:r,
   Bth = Bth + nchoosek(M,i) * x.^i .* (1-x).^(M-i);   % Eq. (4.4.14)
end

norm(Bth-Binf)                        % compare Eq. (4.4.14) with output of LPSM

figure; plot(f,B9); figure; plot(f,Binf);
```

The calls to `lprs` and `lpsm` return the full smoothing matrices B from which the central column $\mathbf{b}_0$ is extracted.

The frequency response function `freqz` expects its filter argument to be causal. The factor $e^{j\omega M}$ compensates for that, corresponding to a time-advance by M units. □

Finally, we note that the Krawtchouk binomial weighting function $\bar{w}(n)$ tends to a gaussian for large M, which is a consequence of the De Moivre-Laplace theorem,

$$\bar{w}(n) = \frac{(2M)!}{2^{2M}(M+n)!\,(M-n)!} \simeq \frac{1}{\sqrt{\pi M}}\,e^{-n^2/M}\,, \quad -M \le n \le M \tag{4.4.16}$$

In fact, the two sides of (4.4.16) are virtually indistinguishable for $M \ge 10$.

4.5 Missing Data and Outliers

The presence of outliers in the observed signal can cause large distortions in the smoothed signal. The left graph of Fig. 4.5.1 shows what can happen. The two vertical lines indicate the region in which there are four strong outliers, which cause the smoothed curve to deviate drastically from the desired signal.

One possible solution [53,165] is to ignore the outliers and estimate the smoothed values from the surrounding available samples using a filter window that spans the outlier region. The same procedure can be used if some data samples are missing. Once the outliers or missing values have been interpolated, one can apply the weighted LPSM filters as usual. The right graph in Fig. 4.5.1 shows the four adjusted interpolated samples. The resulting smoothed signal now estimates the desired signal more accurately.

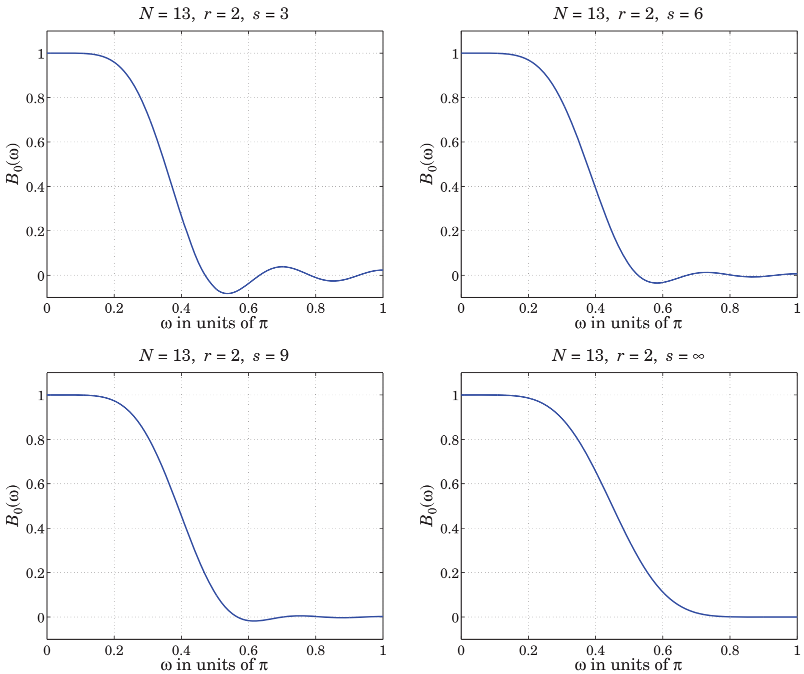

Fig. 4.4.1 Frequency responses of minimum-R_s and maximally-flat filters.

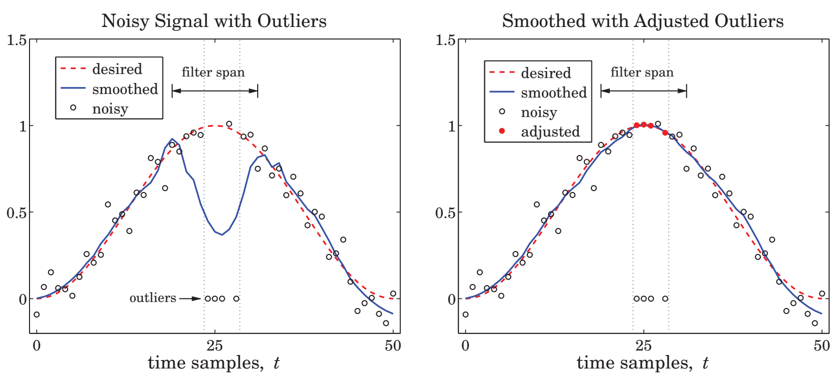

Fig. 4.5.1 Smoothing with missing data or outliers.

This solution can be implemented by replacing the outliers or the missing data by zeros (or, any other values), and assign zero weights to them in the least-squares polynomial fitting problem.

Given a long observed signal y_n, $n = 0, 1, \dots, L-1$, let us assume that in the vicinity of $n = n_0$ there is an outlier or missing sample at the time instant n_0+m, where m lies in the interval $-M \le m \le M$, as shown in Fig. 4.5.2. Several outliers or missing data may be present, not necessarily adjacent to each other, each being characterized by a similar index m.

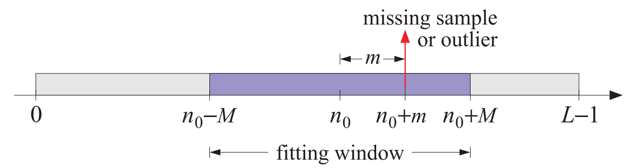

Fig. 4.5.2 Missing sample or outlier and the data window used for estimating it.

The outlier samples y_{n_0+m} can be replaced by zeros and their estimated values, $\hat{y}_{n_0+m}$, can be calculated from the surrounding samples using a filter of length $N = 2M+1$. The corresponding least-squares polynomial-fitting problem is defined by

$$\mathcal{J} = \sum_{m=-M}^{M} p_m w_m \left(y_{n_0+m} - \sum_{i=0}^{d} c_i m^i \right)^2 = \min \tag{4.5.1}$$

where w_m are the usual Henderson weights and the p_m are zero at the indices for the missing data, and unity otherwise. Let $\mathbf{y} = [y_{n_0-M}, \dots, y_{n_0}, \dots, y_{n_0+M}]^T$, and denote by $W, \mathcal{P}$ the corresponding diagonal matrices of the weights w_m, p_m. Then, (4.5.1) reads:

$$\mathcal{J} = (\mathbf{y} - S\mathbf{c})^T \mathcal{P} W (\mathbf{y} - S\mathbf{c}) = \min, \tag{4.5.2}$$

leading to the orthogonality conditions and the solution for $\mathbf{c}$:

$$S^T W \mathcal{P} (\mathbf{y} - S\mathbf{c}) = 0 \quad \Rightarrow \quad \mathbf{c} = (S^T \mathcal{P} W S)^{-1} S^T W \mathcal{P} \mathbf{y} \tag{4.5.3}$$

where we assumed that $S^T \mathcal{P} W S$ is invertible.† The estimated samples will be:

$$\hat{\mathbf{y}} = S\mathbf{c} = S(S^T \mathcal{P} W S)^{-1} S^T W \mathcal{P} \mathbf{y} = B^T \mathbf{y} \tag{4.5.4}$$

with the filter matrix,

$$B = \mathcal{P} W S (S^T \mathcal{P} W S)^{-1} S^T \tag{4.5.5}$$

We note that $\mathcal{P}$ is a projection matrix ($\mathcal{P}^T = \mathcal{P}$ and $\mathcal{P}^2 = \mathcal{P}$) and commutes with W, $\mathcal{P}W = W\mathcal{P}$, because both are diagonal. Defining $\mathcal{Q} = I - \mathcal{P}$ to be the complementary projection matrix, the estimated signal can be decomposed in two parts: $\hat{\mathbf{y}} = \mathcal{P}\hat{\mathbf{y}} + \mathcal{Q}\hat{\mathbf{y}}$, with $\mathcal{Q}\hat{\mathbf{y}}$ being the part that contains the estimated missing values or adjusted outliers.

The quantity $\mathcal{P}\mathbf{y}$ represents the samples that are being used to make the estimates, whereas $\mathcal{Q}\mathbf{y}$ corresponds to the missing samples and can be set to zero or to an arbitrary vector $\mathcal{Q}\mathbf{y}_{\text{arb}}$, in other words, we may replace $\mathbf{y}$ by $\mathcal{P}\mathbf{y} + \mathcal{Q}\mathbf{y}_{\text{arb}}$ without affecting the solution of Eq. (4.5.4). This so because $\mathcal{P}(\mathcal{P}\mathbf{y} + \mathcal{Q}\mathbf{y}_{\text{arb}}) = \mathcal{P}\mathbf{y}$.

†This requires that the number of outliers within the data window be at most $N - d - 1$.

Once the estimated missing values have been obtained, we may replace $Q\mathbf{y}_{\text{arb}}$ by $\mathcal{Q}\hat{\mathbf{y}}$ and recompute the ordinary W-weighted least-squares estimate from the adjusted vector $\mathcal{P}\mathbf{y} + \mathcal{Q}\hat{\mathbf{y}}$. This produces the same $\hat{\mathbf{y}}$ as in (4.5.4). Indeed, one can show that,

$$\hat{\mathbf{y}} = S(S^T\mathcal{P}WS)^{-1}S^TW\mathcal{P}\mathbf{y} = S(S^TWS)^{-1}S^TW(\mathcal{P}\mathbf{y} + \mathcal{Q}\hat{\mathbf{y}}) \tag{4.5.6}$$

To see this, start with the orthogonality equation (4.5.3), and replace $\mathcal{P}\hat{\mathbf{y}} = \hat{\mathbf{y}} - \mathcal{Q}\hat{\mathbf{y}}$:

$$S^TW\mathcal{P}(\mathbf{y} - \hat{\mathbf{y}}) = 0 \quad\Rightarrow\quad S^TW\mathcal{P}\mathbf{y} = S^TW\mathcal{P}\hat{\mathbf{y}} = S^TW(\hat{\mathbf{y}} - \mathcal{Q}\hat{\mathbf{y}})\,, \quad \text{or,}$$

$$S^TW(\mathcal{P}\mathbf{y} + \mathcal{Q}\hat{\mathbf{y}}) = S^TW\hat{\mathbf{y}} = S^TWS(S^T\mathcal{P}WS)^{-1}W\mathcal{P}\mathbf{y}$$

from which Eq. (4.5.6) follows by multiplying both sides by $S(S^TWS)^{-1}$. The MATLAB function `lpmissing` implements the calculation of B in (4.5.5):

```
B = lpmissing(N,d,m,s);          % filter matrix for missing data
```

The following MATLAB code illustrates the generation of Fig. 4.5.1:

```
t = (0:50)'; x0 = (1-cos(2*pi*t/50))/2;      % desired signal

seed=2005; randn('state',seed);
y = x0 + 0.1 * randn(51,1);                  % noisy signal

n0 = 25; m = [-1 0 1 3];                     % four outlier indices relative to n0
y(n0+m+1) = 0;                               % four outlier or missing values

N= 13; d = 2; s = 0; M=(N-1)/2;              % filter specs
x = lpfilt(lprs(N,d,s),y);                   % distorted smoothed signal

B = lpmissing(N,d,m,s);                      % missing-data filter B

yhat = B'*y(n0-M+1:n0+M+1);                  % apply B to the block n0-M <= n <= n0+M
ynew = y; ynew(n0+m+1) = yhat(M+1+m);        % new signal with interpolated outlier values

xnew = lpfilt(lprs(N,d,s),ynew);             % recompute smoothed signal

figure; plot(t,x0,'--', t,y,'o', t,x,'-');          % left graph
figure; plot(t,x0,'--', t,y,'o', t,xnew,'-');       % right graph
hold on; plot(n0+m,yhat(M+1+m),'.');
```

The above method of introducing zero weights at the outlier locations can be automated and applied to the entire signal. Taking a cue from Cleveland's LOESS method [192] discussed in the next section, we may apply the following procedure.

Given a length-L signal y_n, $n = 0, 1, \dots, L-1$, with $L \geq N$, an LPSM or LPRS filter with design parameters N, d, s can be applied to y_n to get a preliminary estimate of the smoothed signal $\hat{x}_n$, and compute the error residuals $e_n = y_n - \hat{x}_n$, that is,

$$\begin{aligned} B &= \text{lprs}(N, d, s) \\ \hat{\mathbf{x}} &= \text{lpfilt}(B, \mathbf{y}) \\ \mathbf{e} &= \mathbf{y} - \hat{\mathbf{x}} \end{aligned} \tag{4.5.7}$$

From the error residual $\mathbf{e}$, one may compute a set of "robustness" weights r_n by using the median of $|e_n|$ as a normalization factor in the bisquare function:

$$\mu = \text{median}(|e_n|)\,, \quad r_n = W\left(\frac{e_n}{K\mu}\right), \quad n = 0, 1, \dots, L-1 \tag{4.5.8}$$

where K is a constant such as $K = 2$–6, and $W(u)$ is the bisquare function,

$$W(u) = \begin{cases} (1-u^2)^2, & \text{if } |u| \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{4.5.9}$$

If a residual e_n deviates too far from the median, that is, $|e_n| > K\mu$, then the robustness weight r_n is set to zero. A new estimate $\hat{x}_n$ can be calculated at each time n by defining the diagonal matrix P in terms of the robustness weights in the neighborhood of n, and then calculating the estimate using the c_0 component of the vector $\mathbf{c}$ in Eq. (4.5.3), that is,

$$\begin{aligned} P_n &= \text{diag}\big([r_{n-M}, \dots, r_n, \dots, r_{n+M}]\big) \\ \hat{x}_n &= c_0 = \mathbf{u}_0^T(S^TP_nWS)^{-1}S^TWP_n\mathbf{y}(n) \end{aligned} \tag{4.5.10}$$

where $\mathbf{u}_0 = [1, 0, \dots, 0]^T$ and $\mathbf{y}(n) = [y_{n-M}, \dots, y_n, \dots, y_{n+M}]^T$. Eq. (4.5.10) may be used for $M \leq n \leq L-1-M$. For $0 \leq n < M$ and $L-1-M < n \leq L-1$ the values of $\hat{x}_n$ can be obtained from the first M and last M outputs of $\hat{\mathbf{y}}$ in (4.5.4) applied to the first and last length-N data vectors and robustness weights:

$$\mathbf{y} = [y_0, y_1, \dots, y_{N-1}]^T\,, \quad P = \text{diag}\big([r_0, r_1, \dots, r_{N-1}]\big)$$

$$\mathbf{y} = [y_{L-N}, y_{L-N+1}, \dots, y_{L-1}]^T\,, \quad P = \text{diag}\big([r_{L-N}, r_{L-N+1}, \dots, r_{L-1}]\big)$$

From the new estimates $\hat{x}_n$, one can compute the new residuals $e_n = y_n - \hat{x}_n$, and repeat the procedure of Eqs. (4.5.8)–(4.5.10) a few more times. A total of 3–4 iterations is typically adequate. The MATLAB function `rlpfilt` implements the above steps:

```
[x,r] = rlpfilt(y,N,d,s,Nit)       % robust local polynomial filtering
```

Its outputs are the estimated signal $\hat{x}_n$ and the robustness weights r_n. The median scaling factor K is an additional optional input, which otherwise defaults to $K = 6$.

If the residuals e_n are gaussian-distributed with variance σ^2, then $\mu = 0.6745\sigma$. The default value $K = 6$ (Cleveland [192]) corresponds to allowing through 99.99 percent of the residuals. Other possible values are $K = \sqrt{6} = 2.44$ (Loader [224]) and $K = 4$ allowing respectively 90 and 99 percent of the values.

Fig. 4.5.3 shows the effect of increasing the number of robustness iterations. It is the same example as that in Fig. 4.5.1, but we have added another four outliers in the vicinity of $n = 10$. The upper-left graph corresponds to ordinary filtering without any robustness weights. One observes the successive improvement of the estimate as the number of iterations increases.

The following MATLAB code illustrates the generation of the lower-right graph. The signal y_n is generated exactly as in the previous example; the outlier values are then introduced around $n = 10$ and $n = 25$:

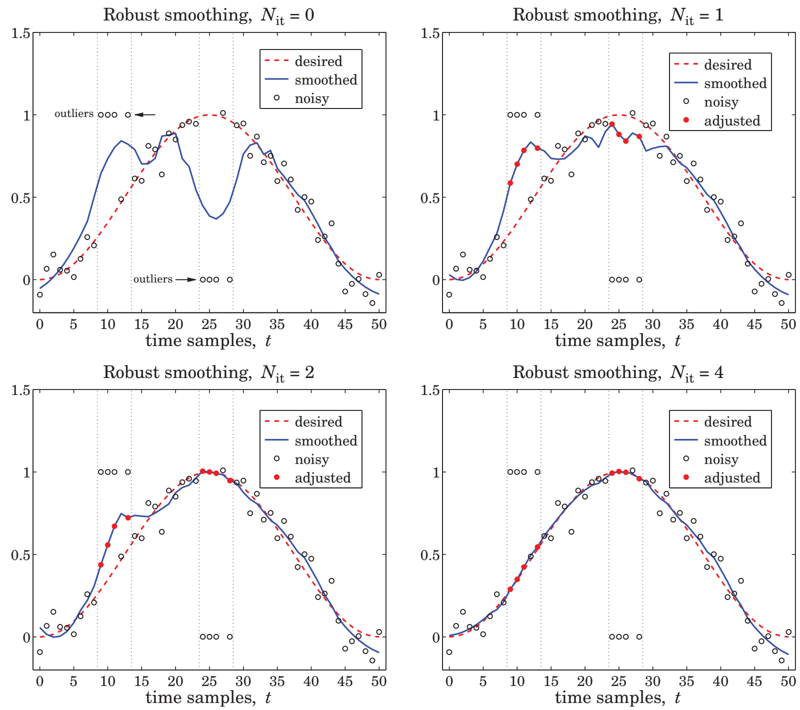

Fig. 4.5.3 Robust smoothing with outliers.

```
n1=10; n2=25; m = [-1 0 1 3];                    % outlier indices relative to n1 and n2
y(n1+m+1)=1;  y(n2+m+1)=0;                       % outlier values

Nit=4; K=4; x = rlpfilt(y,N,d,s,Nit,K);          % robust LP filtering

plot(t,x0,'--', t,y,'o', t,x,'-', n1+m,x(n1+m+1),'.', n2+m,x(n2+m+1),'.');
```

4.6 Problems

4.1 Using binomial identities, prove the equivalence of the three expressions in Eq. (4.4.14) for the maximally-flat filters. Then, show Eq. (4.4.15) and determine the proportionality constants indicated as (const.).

5

Local Polynomial Modeling

5.1 Weighted Local Polynomial Modeling

The methods of weighted least-squares local polynomial modeling and robust filtering can be generalized to unequally-spaced data in a straightforward fashion. Such methods provide enough flexibility to model a wide variety of data, including surfaces, and have been explored widely in recent years [188–231]. For equally-spaced data, the weighted performance index centered at time n was:

$$\mathcal{J}_n = \sum_{m=-M}^{M} \big(y_{n+m} - p(m)\big)^2 w(m) = \min\,, \quad p(m) = \sum_{r=0}^{d} c_i m^r \tag{5.1.1}$$

The value of the fitted polynomial $p(m)$ at $m = 0$ represents the smoothed estimate of y_n, that is, $\hat{x}_n = c_0 = p(0)$. Changing summation indices to $k = n + m$, Eq. (5.1.1) may be written in the form:

$$\mathcal{J}_n = \sum_{k=n-M}^{n+M} \big(y_k - p(k-n)\big)^2 w(k-n) = \min\,, \quad p(k-n) = \sum_{r=0}^{d} c_i (k-n)^r \tag{5.1.2}$$

For a set of N unequally-spaced observations $\{t_k, y(t_k)\}$, $k = 0, 1, \dots, N-1$, we wish to interpolate smoothly at some time instant t, not necessarily coinciding with one of the observation times t_k, but lying in the interval $t_0 \le t \le t_{N-1}$. A generalization of the performance index (5.1.2) is to introduce a t-dependent window bandwidth h_t, and use only the observations that lie within that window, $|t_k - t| \le h_t$, to perform the polynomial fit:

$$\mathcal{J}_t = \sum_{|t_k - t| \le h_t} \big(y(t_k) - p(t_k - t)\big)^2 w(t_k - t) = \min\,, \quad p(t_k - t) = \sum_{r=0}^{d} c_r (t_k - t)^r \tag{5.1.3}$$

The estimated/interpolated value at t will be $\hat{x}_t = c_0 = p(0)$, and the estimated first derivative, $\hat{\dot{x}}_t = c_1 = \dot{p}(0)$, and so on for the higher derivatives, with $r!\,c_r$ representing the rth derivative. As illustrated in Fig. 5.1.1, the fitted polynomial,

$$p(x - t) = \sum_{r=0}^{d} c_r (x - t)^r\,, \qquad t - h_t \le x \le t + h_t$$