Operations on digital sound: digital filter

Øyvind Ryan

Jan 20, 2017

Digital filters

What we will define as *digital filters* is exemplified by the following procedure:

$$z_n = \frac{1}{4}(x_{n-1} + 2x_n + x_{n+1}), \quad \text{for } n = 0,\ 1,\ \ldots,\ N-1.$$

Matrices of filters

Assume that the input vector is periodic with period N, so that $x_{n+N} = x_n$. It is straightforward to show that the output vector $\boldsymbol{z}$ is also periodic with period N.

The filter is also clearly a linear transformation and may therefore be represented by an $N \times N$ matrix S that maps the vector $\boldsymbol{x} = (x_0, x_1, \ldots, x_{N-1})$ to the vector $\boldsymbol{z} = (z_0, z_1, \ldots, z_{N-1})$, i.e., we have $\boldsymbol{z} = S\boldsymbol{x}$.

The elements of S can be found by row as

$$S = \frac{1}{4}\begin{pmatrix} 2 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \\ 1 & 2 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 2 & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 2 \end{pmatrix}.$$

Circulant Toeplitz matrices

The matrix we just stated is called a circulant Toeplitz matrix. The general definition is as follows and may seem complicated, but is in fact quite straightforward:

Definition 3.1 An $N \times N$-matrix S is called a Toeplitz matrix if its elements are constant along each diagonal. More formally, $S_{k,l} = S_{k+s,l+s}$ for all nonnegative integers k, l, and s such that both $k+s$ and $l+s$ lie in the interval $[0, N-1]$. A Toeplitz matrix is said to be circulant if in addition

$$S_{(k+s) \bmod N,(l+s) \bmod N} = S_{k,l}$$

for all integers k, l in the interval $[0, N-1]$, and all s (Here mod denotes the remainder modulo N).

More general expression for a filter

$$z_n = \sum_k t_k x_{n-k}.$$

- $\boldsymbol{x}$ denotes the *input vector*.
- $\boldsymbol{z}$ the *output vector*.
- t_k denotes the *filter coefficients*.

Filter in Matlab

Assume that t_0, t_1, ..., t_{kmax} are the only non-zero filter coefficients.

```
z = zeros(1, N);
for n = kmax:(N-1)
    for k = 0:kmax
        z(n + 1) = z(n + 1) + t(k + 1)*x(n - k + 1);
    end
end
```

Filter in Python

```
z = zeros_like(x)
for n in range(kmax,N):
    for k in range(kmax + 1):
        z[n] += t[k]*x[n - k]
```

Filters as matrices, Proposition 3.2

Any operation defined by Equation (3.3) is a linear transformation which transforms a vector of period N to another of period N. It may therefore be represented by an $N \times N$ matrix S that maps the vector $\boldsymbol{x} = (x_0, x_1, \ldots, x_{N-1})$ to the vector $\boldsymbol{z} = (z_0, z_1, \ldots, z_{N-1})$, i.e., we have $\boldsymbol{z} = S\boldsymbol{x}$. Moreover, the matrix S is a circulant Toeplitz matrix, and the first column $\boldsymbol{s}$ of this matrix is given by

$$s_k = \begin{cases} t_k, & \text{if } 0 \leq k < N/2; \\ t_{k-N} & \text{if } N/2 \leq k \leq N-1. \end{cases}$$

In other words, the first column of S can be obtained by placing the coefficients in (3.3) with positive indices at the beginning of $\boldsymbol{s}$, and the coefficients with negative indices at the end of $\boldsymbol{s}$.

Compact notation for filters, Definition 3.3

Let k_{min}, k_{max} be the smallest and biggest index of a filter coefficient in Equation (3.3) so that $t_k \neq 0$ (if no such values exist, let $k_{\text{min}} = k_{\text{max}} = 0$), i.e.

$$z_n = \sum_{k=k_{min}}^{k_{max}} t_k x_{n-k}.$$

We will use the following compact notation for S:

$$S = \{t_{k_{min}}, \ldots, t_{-1}, \underline{t_0}, t_1, \ldots, t_{k_{max}}\}.$$

In other words, the entry with index 0 has been underlined, and only the nonzero t_k's are listed. k_{max} and k_{min} are also called the start and end indices of S. By the length of S, denoted $l(S)$, we mean the number $k_{max} - k_{min}$.

Convolution of vectors, Definition 3.4

By the *convolution* of two vectors $\boldsymbol{t} \in \mathbb{R}^M$ and $\boldsymbol{x} \in \mathbb{R}^N$ we mean the vector $\boldsymbol{t} * \boldsymbol{x} \in \mathbb{R}^{M+N-1}$ defined by

$$(\boldsymbol{t} * \boldsymbol{x})_n = \sum_k t_k x_{n-k},$$

where we only sum over k so that $0 \leq k < M$, $0 \leq n-k < N$.

Using convolution to compute filters Proposition 3.5

Assume that S is a filter on the form

$$S = \{t_{-L}, \ldots, \underline{t_0}, \ldots, t_L\}.$$

If $\boldsymbol{x} \in \mathbb{R}^N$, then $S\boldsymbol{x}$ can be computed as follows:

- Form the vector
  $\tilde{\boldsymbol{x}} = (x_{N-L}, \cdots, x_{N-1}, x_0, \cdots, x_{N-1}, x_0, \cdots, x_{L-1}) \in \mathbb{R}^{N+2L}$.
- Use the conv function to compute $\tilde{\boldsymbol{z}} = \boldsymbol{t} * \tilde{\boldsymbol{x}} \in \mathbb{R}^{M+N+2L-1}$.
- We have that $S\boldsymbol{x} = (\tilde{z}_{2L}, \ldots, \tilde{z}_{M+N-2})$.

Convolution and polynomials Proposition 3.6

Assume that $p(x) = a_N x^N + a_{N-1} x_{N-1} + \ldots, a_1 x + a_0$ and $q(x) = b_M x^M + b_{M-1} x_{M-1} + \ldots, b_1 x + b_0$ are polynomials of degree N and M respectively. Then the coefficients of the polynomial pq can be obtained by computing `conv(a,b)`.

Digital filters and vector frequency response, Definition 3.7

A linear transformation $S : \mathbb{R}^N \mapsto \mathbb{R}^N$ is a said to be a digital filter, or simply a filter, if, for any integer n in the range $0 \leq n \leq N-1$ there exists a value $\lambda_{S,n}$ so that

$$S(\phi_n) = \lambda_{S,n}\phi_n,$$

i.e., the N Fourier vectors are the eigenvectors of S. The vector of (eigen)values $\boldsymbol{\lambda}_S = (\lambda_{S,n})_{n=0}^{N-1}$ is often referred to as the *(vector) frequency response* of S.

The product of two filters is a filter, Corollary 3.8

The product of two digital filters is again a digital filter. Moreover, all digital filters commute, i.e. if S_1 and S_2 are digital filters, $S_1 S_2 = S_2 S_1$.

Time-invariance, Definition 3.9

Assume that S is a linear transformation from $\mathbb{R}^N$ to $\mathbb{R}^N$. Let $\boldsymbol{x}$ be input to S, and $\boldsymbol{y} = S\boldsymbol{x}$ the corresponding output. Let also $\boldsymbol{z}$, $\boldsymbol{w}$ be delays of $\boldsymbol{x}$, $\boldsymbol{y}$ with d elements (i.e. $z_n = x_{n-d}$, $w_n = y_{n-d}$). S is said to be *time-invariant* if, for any d and $\boldsymbol{x}$, $S\boldsymbol{z} = \boldsymbol{w}$ (i.e. S sends the delayed input vector to the delayed output vector).

Characterizations of digital filters, Theorem 3.10

The following are equivalent characterizations of a digital filter:

- $S = (F_N)^H D F_N$ for a diagonal matrix D, i.e. the Fourier basis is a basis of eigenvectors for S.
- S is a circulant Toeplitz matrix.
- S is linear and time-invariant.

Connection between frequency response and the matrix, Theorem 3.11

Any digital filter is uniquely characterized by the values in the first column of its matrix. Moreover, if $\boldsymbol{s}$ is the first column in S, the frequency response of S is given by

$$\boldsymbol{\lambda}_S = \mathrm{DFT}_N \boldsymbol{s}.$$

Conversely, if we know the frequency response $\boldsymbol{\lambda}_S$, the first column $\boldsymbol{s}$ of S is given by

$$\boldsymbol{s} = \mathrm{IDFT}_N \boldsymbol{\lambda}_S.$$

Connection between vector- and continuous frequency response, Theorem 3.14

The function $\lambda_S(\omega)$ defined on $[0, 2\pi)$ by

$$\lambda_S(\omega) = \sum_k t_k e^{-ik\omega},$$

where t_k are the filter coefficients of S, satisfies

$$\lambda_{S,n} = \lambda_S(2\pi n/N) \text{ for } n = 0, 1, \ldots, N-1$$

for any N. In other words, regardless of N, the vector frequency response lies on the curve λ_S.

Observation 3.15 (Plotting the frequency response): When plotting the frequency response on $[0, 2\pi)$, angular frequencies near 0 and 2π correspond to low frequencies, angular frequencies near π correspond to high frequencies

Observation 3.16 (higher and lower frequencies): When plotting the frequency response on $[-\pi, \pi)$, angular frequencies near 0 correspond to low frequencies, angular frequencies near $\pm\pi$ correspond to high frequencies.

We have that

- The continuous frequency response satisfies $\lambda_S(-\omega) = \overline{\lambda_S(\omega)}$.
- If S is a digital filter, S^T is also a digital filter. Moreover, if the frequency response of S is $\lambda_S(\omega)$, then the frequency response of S^T is $\overline{\lambda_S(\omega)}$.
- If S is symmetric, λ_S is real. Also, if S is antisymmetric (the element on the opposite side of the diagonal is the same, but with opposite sign), λ_S is purely imaginary.
- A digital filter S is an invertible if and only if $\lambda_{S,n} \neq 0$ for all n. In that case S^{-1} is also a digital filter, and $\lambda_{S^{-1},n} = 1/\lambda_{S,n}$.
- If S_1 and S_2 are digital filters, then $S_1 S_2$ also is a digital filter, and $\lambda_{S_1 S_2}(\omega) = \lambda_{S_1}(\omega)\lambda_{S_2}(\omega)$.

Adding echo to sound 1

```
[N,nchannels] = size(x);
z = zeros(N,nchannels);
z(1:d,:) = x(1:d,:);
z((d+1):N,:) = x((d+1):N,:)+c*x(1:(N-d),:);
```

Adding echo to sound 2

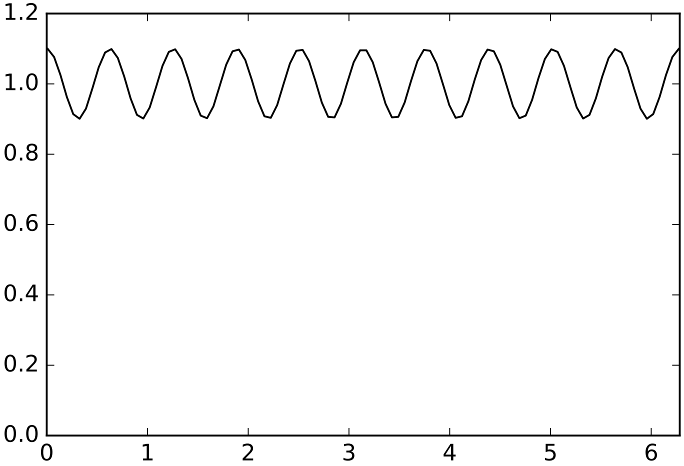

Figure: The frequency response of a filter which adds an echo with damping factor $c = 0.1$ and delay $d = 10$.

Moving average filters

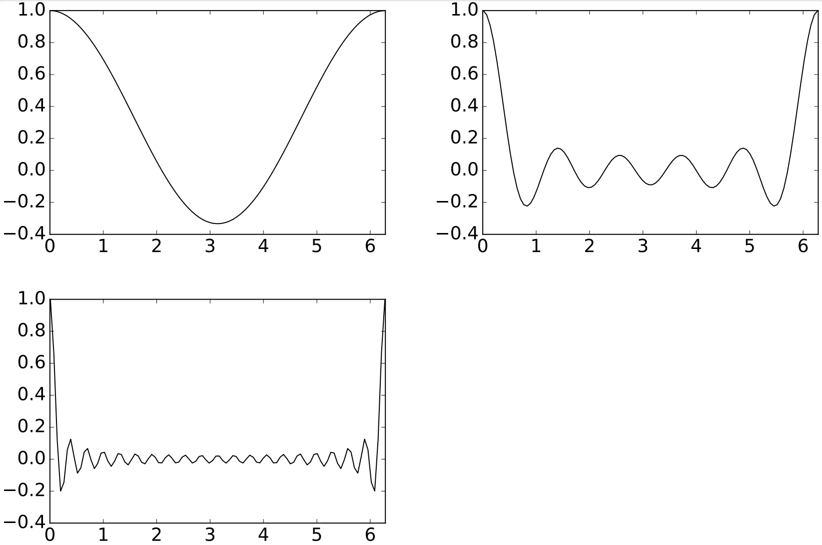

Figure: The frequency response of moving average filters with $L = 1$, $L = 5$, and $L = 20$.

Lowpass and highpass filters, definition 3.21

A filter S is called

- a *lowpass filter* if $\lambda_S(\omega)$ is large when ω is close to 0, and $\lambda_S(\omega) \approx 0$ when ω is close to π (i.e. S keeps low frequencies and annhilates high frequencies),
- a *highpass filter* if $\lambda_S(\omega)$ is large when ω is close to π, and $\lambda_S(\omega) \approx 0$ when ω is close to 0 (i.e. S keeps high frequencies and annhilates low frequencies),
- a *bandpass filter* if $\lambda_S(\omega)$ is large within some interval $[a, b] \subset [0, 2\pi]$, and $\lambda_S(\omega) \approx 0$ outside this interval.

Dropping filter coefficients in ideal lowpass filters

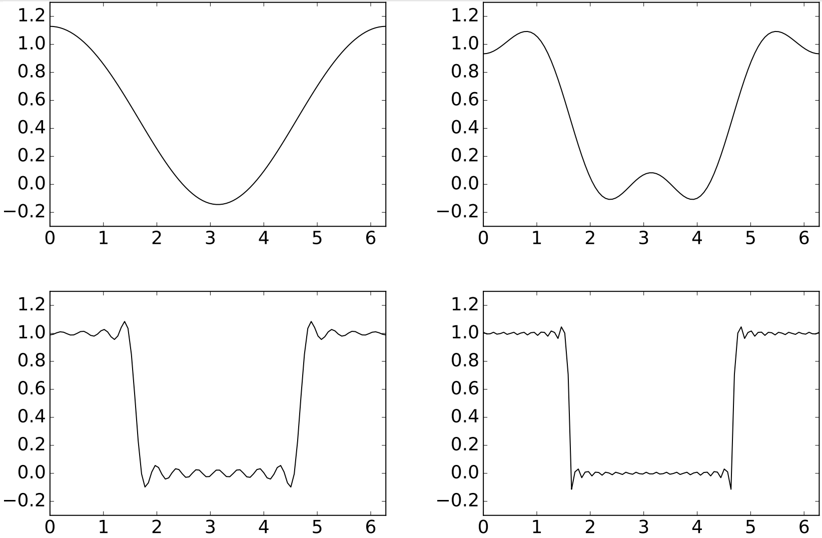

Figure: The frequency response which results by including the first 1/32, the first 1/16, the first 1/4, and and all of the filter coefficients in the ideal lowpass filter.

Filters and the MP3 standard

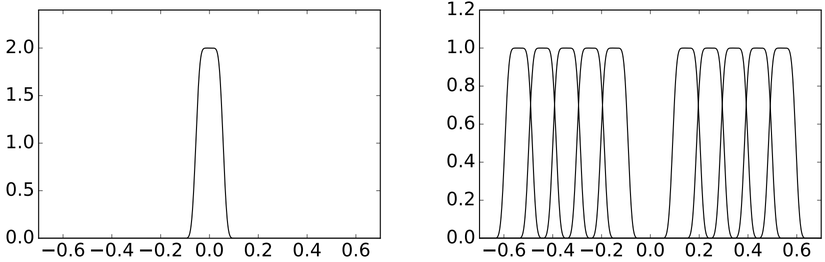

Figure: Frequency responses of some filters used in the MP3 standard. The prototype filter is shown left. The other frequency responses at right are simply shifted copies of this.

Reducing the treble by picking filter coefficients from Pascals triangle

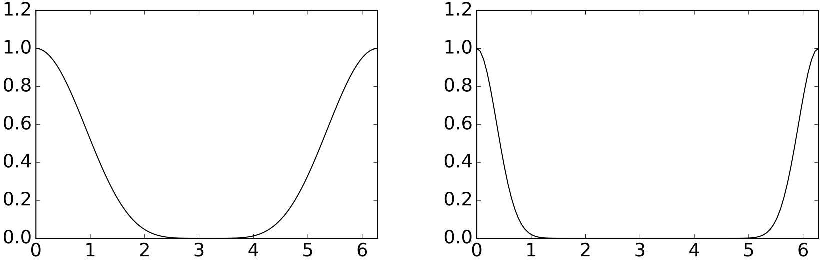

Figure: The frequency response of filters corresponding to iterating the moving average filter $\{\underline{1/2}, 1/2\}$ $k = 5$ and $k = 30$ times (i.e. using row k in Pascal's triangle).

Reducing treble and bass

Observation ?? (Reducing the treble): Let $\boldsymbol{x}$ be the samples of a digital sound, and let S be a filter with coefficients taken from row k of Pascals triangle. Then $S\boldsymbol{x}$ has reduced treble when compared to $\boldsymbol{x}$.

Observation 3.22 (Passing between lowpass- and highpass filters): Assume that S_2 is obtained by adding an alternating sign to the filter coefficicents of S_1. If S_1 is a lowpass filter, then S_2 is a highpass filter. If S_1 is a highpass filter, then S_2 is a lowpass filter.

Observation ?? (Pascals triangle and reducing the bass): Let $\boldsymbol{x}$ be the samples of a digital sound, and let S be a filter with filter coefficients taken from row k of Pascal's triangle, and add an alternating sign to the filter coefficients. Then $S\boldsymbol{x}$ has reduced bass when compared to $\boldsymbol{x}$.