Summary notes for EQ2300 Digital Signal Processing

allowed aid for final exams during 2016

Joakim Jaldén, 2016-01-12

1 Prerequisites

2 The DFT and the FFT

2.1 The discrete Fourier transform

- The discrete Fourier transform (DTFT) is given by

$$X(\nu) = \mathcal{F}\{x[n]\} \triangleq \sum_{n=-\infty}^{\infty} x[n]e^{-j2\pi\nu n}\,, \quad \nu \in \mathbb{R}\,,$$

  assuming that the integral converges, and it's inverse is given by

$$x[n] = \mathcal{F}^{-1}\{X(\nu)\} \triangleq \int_0^1 X(\nu)e^{j2\pi\nu n}d\nu\,, \quad n \in \mathbb{Z}\,.$$

- In the above, $\nu \in \mathbb{R}$ is commonly referred to as the *normalized frequency*.

- Some properties of the DTFT are

  - Linearity: $x[n] \xrightarrow{\mathcal{F}} X(\nu)$, $y[n] \xrightarrow{\mathcal{F}} Y(\nu) \Rightarrow cx[n] + dy[n] \xrightarrow{\mathcal{F}} cX(\nu) + dY(\nu)$

  - Time shift: $x[n-k] \xrightarrow{\mathcal{F}} e^{-j2\pi k\nu}X(\nu)$

  - Conjugate symmetry for *real valued* signals: $x[n] \in \mathbb{R} \;\Rightarrow X(\nu) = X^*(-\nu)$

  - Parseval's relation:
$$\sum_{n=-\infty}^{\infty} \left|x[n]\right|^2 = \int_0^1 \left|X(\nu)\right|^2 d\nu$$

- The DTFT $X(\nu)$ of a discrete time sequence $x[n]$ is always periodic with period 1, i.e., $X(\nu+k) = X(\nu)$ for all integers k.

- Examples of the DTFT and more properties are found in the pink collection of formulas.

- One of the key uses of the DTFT is that it allows us to study linear and time invariant (LTI) systems in the frequency domain according to

$$\xrightarrow[X(\nu)]{x[n]} \boxed{\text{LTI system}} \xrightarrow[Y(\nu) = H(\nu)X(\nu)]{y[n] = h[n] * x[n]}$$

  where $h[n]$ is the system's *impulse response*, where $H(\nu) = \mathcal{F}\{h[n]\}$ is the system's *frequency responce*, and where

$$h[n] * x[n] \triangleq \sum_{m=-\infty}^{\infty} h[m]x[n-m] = \sum_{m=-\infty}^{\infty} x[m]h[n-m]$$

  is the linear convolution operation.

2.2 The discrete Fourier transform

- The N-point discrete Fourier transform (DFT) is given by

$$X[k] = \mathcal{F}_N\{x[n]\} \triangleq \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N} \;, \quad k = 0, \ldots, N-1\,,$$

and it's inverse is given by

$$x[n] = \mathcal{F}_N^{-1}\{X[k]\} \triangleq \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j2\pi kn/N} \;, \quad n = 0, \ldots, N-1\,.$$

- A common alternative notations is

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn} \;, \qquad \text{where} \quad W_N \triangleq e^{-j2\pi/N} \quad (\text{the } N\text{th root of unity})$$

- Some properties of the DFT are
  - Linearity: $x[n] \overset{\mathcal{F}_N}{\to} X[k]$, $y[n] \overset{\mathcal{F}_N}{\to} Y[k] \Rightarrow cx[n] + dy[n] \overset{\mathcal{F}_N}{\to} cX[k] + dY[k]$
  - Parseval's relation:
  $$\sum_{n=-\infty}^{\infty} \big|x[n]\big|^2 = \frac{1}{N}\sum_{k=0}^{N-1} \big|X[k]\big|^2$$
  - Time shift: $x[(n-m) \bmod N] \overset{\mathcal{F}_N}{\to} e^{-j2\pi km/N}X[k] = W_N^{km}X[k]$
  - Convolution in time: $x[n] \, \textcircled{\scriptsize N} \, y[n] \overset{\mathcal{F}_N}{\to} X[k]Y[k]$
  - Multiplication in time: $x[n]y[n] \overset{\mathcal{F}_N}{\to} N^{-1}X[k] \, \textcircled{\scriptsize N} \, Y[k]$

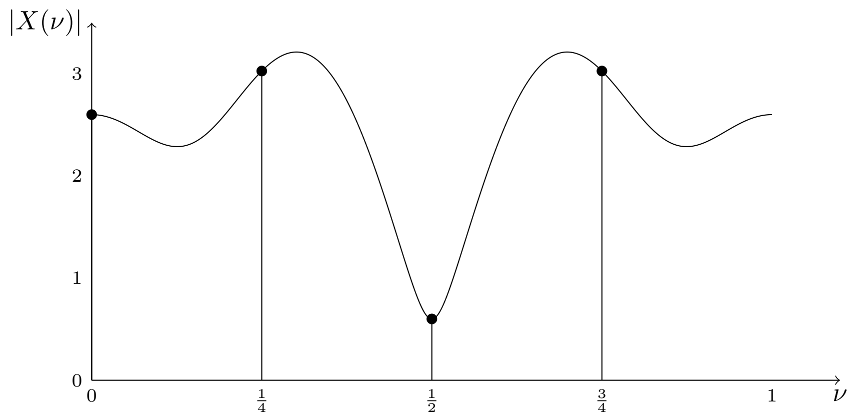
where $n \bmod N$ denotes the modulo N operation (i.e., $n \bmod N$ is for any $n \in \mathbb{Z}$ the unique integer in the range $0, \ldots, N-1$ of the from $n - Nk$ where $k \in \mathbb{Z}$), and $\textcircled{\scriptsize N}$ denotes the *N-point circular convolution* defined by

$$x[n] \, \textcircled{\scriptsize N} \, y[n] \triangleq \sum_{m=0}^{N-1} x[m]y[(n-m) \mod N] = \sum_{m=0}^{N-1} y[m]x[(n-m) \mod N]\,.$$

- The DFT can be used to numerically evaluate the DTFT for normalized frequencies ν_k when $x[n] = 0$ for all $n < 0$ and $n \geq N$ according to

$$X[k] = X(\nu)\Big|_{\nu = k/N}$$

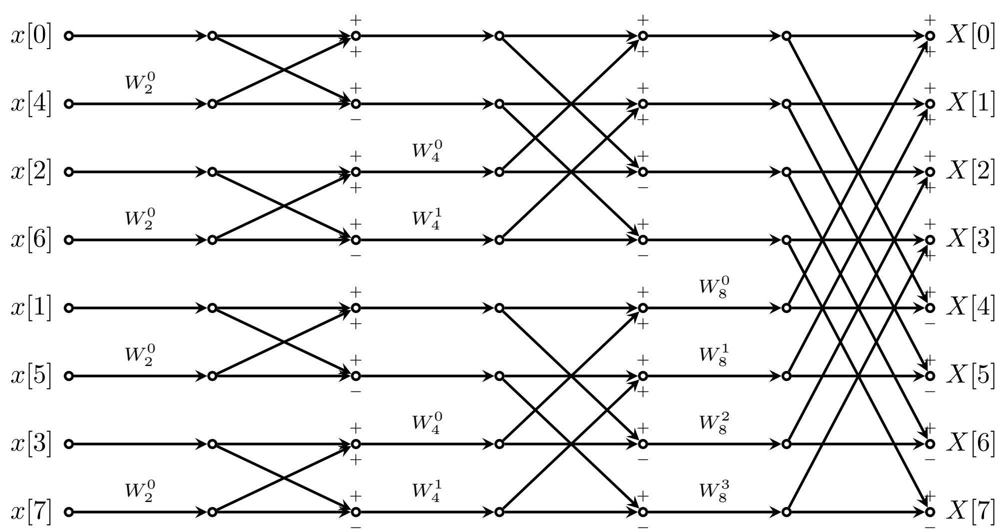
where $X[k] = \mathcal{F}_N\{x[n]\}$ and $X(\nu) = \mathcal{F}\{x[n]\}$. Note that this is not the case when $x[n] \neq 0$ for some $n < 0$ or $n \geq N$. The concept is illustrated by the figure below



for the case where $N = 4$. Zero padding the DFT (increasing N beyond the minimum) can be used to increase the number of normalized frequencies $\nu_k = k/N$ for $k = 0, \ldots, N-1$ where the DTFT is evaluated.

2.3 The fast Fourier transform

- The fast Fourier transform (FFT) is an algorithm (or collection of algorithms) for efficiently computing the DFT for some N.
- The radix-2 FFT (a specific FFT algorithm) assumes $N = 2^p$ for some integer p, and can be illustrated using a graph as shown below for the case where $N = 8 = 2^3$.



- The radix-2 FFT builds on a divide and conquer strategy where an N-point DFT is computed from 2 $N/2$-point DFTs, and so on...
- The numer of complex valued multiplications required for computing the N-point DFT using the radix-2 FFT algorithm is (or more accurately, is well approximated by)

$$C_N = \frac{N}{2} \log_2 N \,.$$

- The numer of complex valued multiplications required for diectly computing the N-point DFT is N^2.

3 Filtering with the FFT

3.1 Linear and circular convolution

- The DFT can be used to filter finite length signals in the frequency domain by recognizing when the *cyclic* convolution computed the values of the *linear* convolution.

- Assume that $x[n]$ has length L ($x[n] = 0$ if $n < 0$ or $n \geq L$) and $h[n]$ has length M ($h[n] = 0$ if $n < 0$ or $n \geq M$). Let $y[n] = h[n] * x[n]$ and $y_N[n] = h[n] \circledN x[n]$. Then

$$y[n] = y_N[n] \quad \text{for} \quad n = 0, \ldots, N-1$$

  if $N \geq L + M - 1$. If $N = L > M$, then $y[n] = y_N[n]$ can only be guaranteed for $n = M-1, \ldots, N-1$.

- The above results can be used to compute $y[n] = h[n] * x[n]$ in the frequency domain from some values of n as $y_N[n] = \mathcal{F}_N^{-1}\{\mathcal{F}_N\{x[n]\} \times \mathcal{F}_N\{h[n]\}\}$. By using the FFT this can lead to fewer complex valued multiplications than the direct computation of $y[n]$ via the convolution sum.

- The overall complexity in terms of complex valued multiplications can be broken down according to

  - $H[k] = \mathcal{F}_N\{h[n]\}$: $N/2 \log_2 N$ multiplications
  - $X[k] = \mathcal{F}_N\{x[n]\}$: $N/2 \log_2 N$ multiplications
  - $Y[k] = H[k]X[k]$: N multiplications
  - $y[n] = \mathcal{F}_N^{-1}\{Y[k]\}$: $N/2 \log_2 N$ multiplications

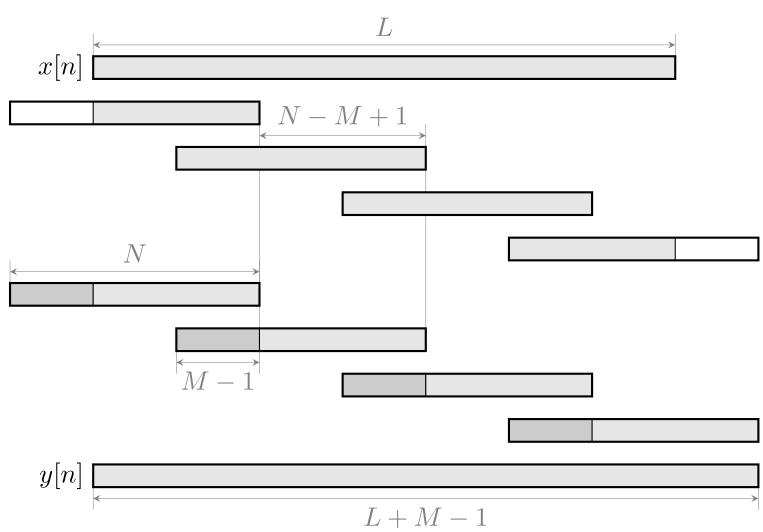
  which leads to a total complexity of

$$C_N = \frac{3N \log_2 N}{2} + N \,.$$

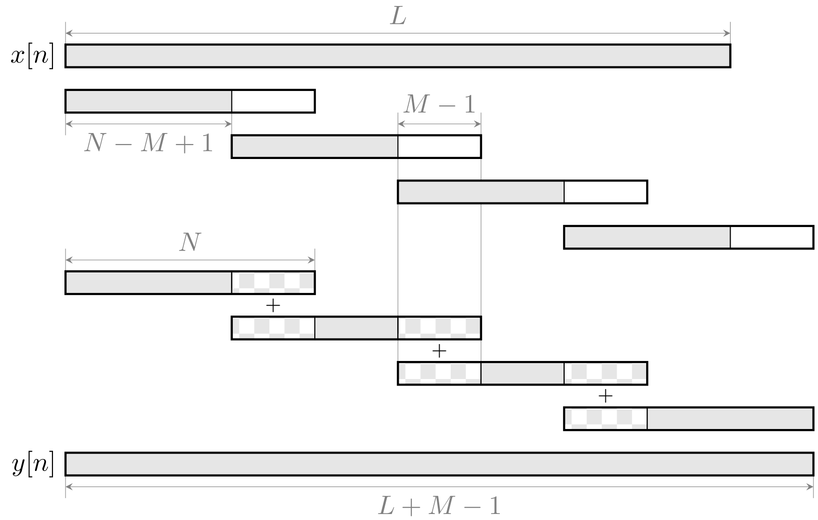
3.2 Overlap save and overlap add

- When $L \gg M$ (where L and M are given as above) it is more computationally efficient to divide $x[n]$ into (overlapping) blocks, filter each block in the frequency domain, and then reconstruct the result in the time domain. This is the basis for the overlap-save and overlap-add methods.

- Overlap-save's block structure can be illustrated as



- Overlap-add's block structure can be illustrated as

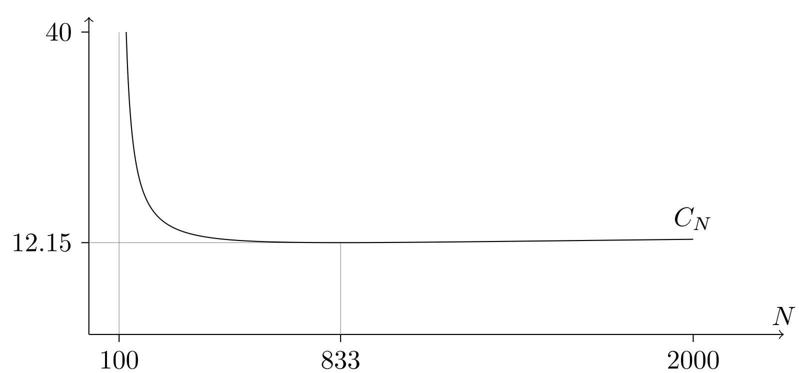


- The complexity *per block* of either overlap-save or overlap-add can be broken down as
  - Get N-point sequence $x_\mathrm{B}[n]$ from $x[n]$
  - Compute $X_\mathrm{B}[k] = \mathcal{F}_N\{x_\mathrm{B}[n]\}$: $N/2\log_2 N$ multiplications
  - Compute $Y_\mathrm{B}[k] = H[k]X_B[k]$: N multiplications
  - Compute $y_\mathrm{B}[n] = \mathcal{F}_N^{-1}\{Y_\mathrm{B}[k]\}$: $N/2\log_2 N$ multiplications

  which yields a total of $N\log_2 N + N$ complex valued multiplications per block. Singe the same $H[k]$can be used for every block, we do not count the complexity of obtaining $H[k]$. Since $N-M+1$ valid samples of $y[n]$ are computed through each block, the complexity in terms of *complex valued multiplications per output sample* becomes

$$C_N \triangleq \frac{N + N\log_2 N}{N-M+1} = \frac{N\log_2 2N}{N-M+1}\,.$$

- The complexity as a function of the block length N is illustrated below



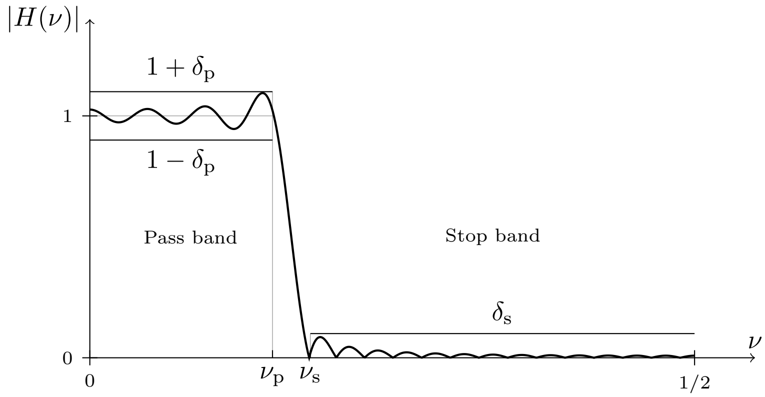
  for the case where $M = 100$. The optimal N, on the form $N = 2^p$ for an integer p and as a function of M, is tabulated below for some different M.

M	N	C_N
5	16	6.67
10	64	8.15
15	64	8.96
20	128	9.39
50	512	11.06
100	1024	12.18
1000	8192	15.94

4 FIR filters and FIR approximations

4.1 Overview

- The filter design problem is to create a filter structure with some filter coefficients that satisfies a set of given design criteria, as illustrated for the case of a low pass filter below.

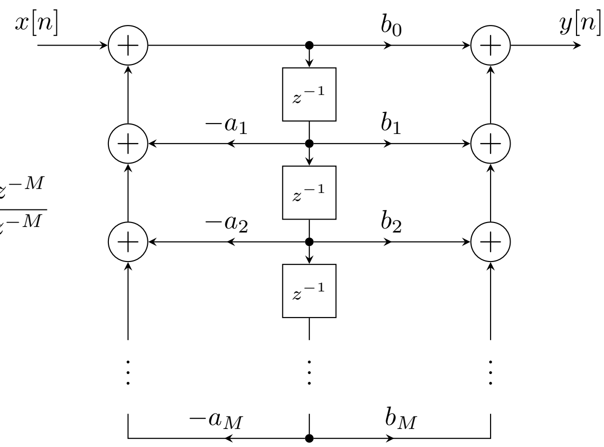


- Implementable filters can generally have an infinite impulse response (IIR) with a transfer function of the form
$$H(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_M z^{-M}} \,,$$
of have an finite impulse response (FIR) with transfer function
$$H(z) = b_0 + b_1 z^{-1} + \cdots + b_M z^{-M} \,,$$
in which case $h[n] = b_n$ for $n = 0, \ldots, M$ and $h[n] = 0$ for $n < 0$ or $n > M$.

- M is referred to as the *order* of the filter, and for FIR filters the *length* of the impulse responce is $N = M + 1$.

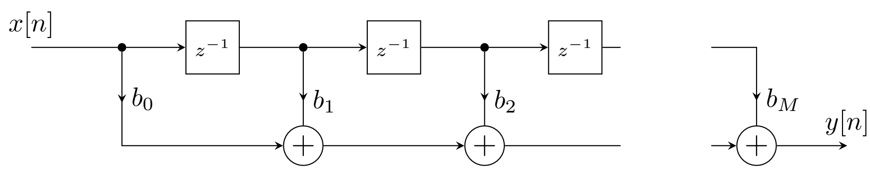
- The *canonic direct form realization* of an IIR filter is

$$H(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_M z^{-M}}$$



- The direct form realization of an FIR filter (also referred to as a tapped delay line) is

$$H(z) = b_0 + b_1 z^{-1} + b_2 z^{-n} + \cdots + b_M z^{-M}$$



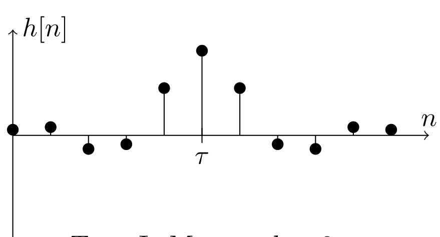
- An Mth order real valued FIR filter $h[n]$ have a constant group delay τ, or *linear phase*, where

$$H(\nu) = B(\nu) e^{-j\frac{\pi}{2}k} e^{-j2\pi\tau\nu} \quad \text{with} \quad B(\nu) \in \mathbb{R}\,,$$

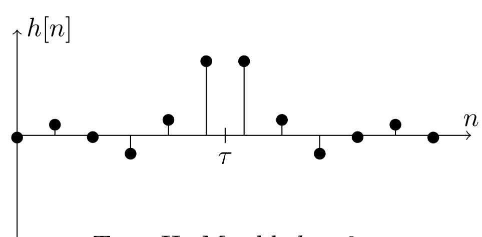
  if and only if it satisfies the symmetry (anti-symmetry) property

$$h[n] = (-1)^k h[M-n] \quad \text{with} \quad k \in \{0,1\} \quad \text{and} \quad \tau = \frac{M}{2}$$

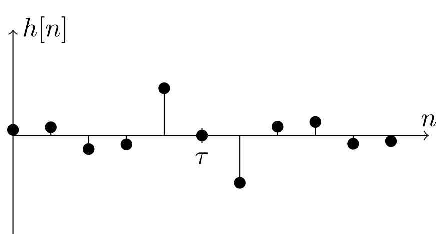
  which leads to four types of possible filters as shown below.



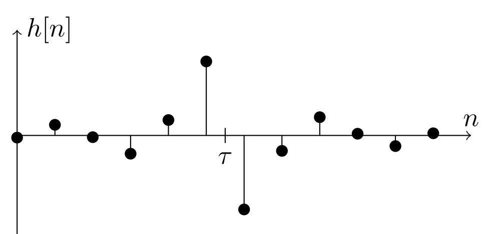
Type I: M even, $k = 0$



Type II: M odd, $k = 0$



Type III: M even, $k = 1$



Type IV: M odd, $k = 1$

4.2 Frequency sampling

- The basic idea of frequency sampling can be described as follows.

  1. Let $D(\nu) \in \mathbb{C}$ be a desired frequency response, and select the filter length N
  2. Let $H[k] = D(k/N)$ for $k = 0, \ldots, N-1$
  3. Obtain $h[n]$ for $k = 0, \ldots, N-1$ by computing the N-point inverse DFT of $H[k]$

  The resulting filter will always be FIR, but special care must be taken in the selection of $D(\nu)$ if one wish to to ensure that $h[n]$ is real valued and has the linear phase property. Special design cases for the different types of linear phase FIR filters are are given next.

- Type I linear phase Mth order FIR design (M even, symmetric filter): Let $D(k/N) = A[k]e^{2\pi\theta[k]}$ with $N = M+1$ and

$$\theta[k] = -\frac{kM}{2(M+1)} \qquad \text{for} \quad k = 0, \ldots, M$$
$$A[k] = A[M-k+1] \qquad \text{for} \quad k = 1, \ldots, M/2$$

  Then

$$h[n] = \frac{1}{M+1}\left[A[0] + 2\sum_{k=1}^{M/2}(-1)^k A[k]\cos\frac{\pi k(1+2n)}{M+1}\right], \quad n = 0, \ldots, M$$

- Type II linear phase Mth order FIR design (M odd, symmetric filter): Let $D(k/N) = A[k]e^{2\pi\theta[k]}$ with $N = M+1$ and

$$\theta[k] = \begin{cases} -\frac{kM}{2(M+1)} & \text{for} \quad k = 0, \ldots, (M-1)/2 \\ \frac{1}{2} - \frac{kM}{2(M+1)} & \text{for} \quad k = (M+3)/2, \ldots, M \end{cases}$$
$$A[k] = A[M-k+1] \qquad \text{for} \quad k = 1, \ldots, (M+1)/2$$
$$A[(M+1)/2] = H(1/2) = 0$$

  Then

$$h[n] = \frac{1}{M+1}\left[A[0] + 2\sum_{k=1}^{(M-1)/2}(-1)^k A[k]\cos\frac{\pi k(1+2n)}{M+1}\right], \quad n = 0, \ldots, M$$

- Type III linear phase Mth order FIR design (M even, anti-symmetric filter): Let $D(k/N) = A[k]e^{2\pi\theta[k]}$ with $N = M+1$ and

$$\theta[k] = \begin{cases} \frac{1}{4} - \frac{kM}{2(M+1)} & \text{for} \quad k = 1, \ldots, M/2 \\ -\frac{1}{4} - \frac{kM}{2(M+1)} & \text{for} \quad k = M/2+1, \ldots, M \end{cases}$$
$$A[k] = A[M-k+1] \qquad \text{for} \quad k = 1, \ldots, M/2$$
$$A[0] = H(0) = 0\,, \quad H(1/2) = 0$$

  Then

$$h[n] = \frac{2}{M+1}\sum_{k=0}^{M/2}(-1)^{k+1}A[k]\sin\frac{\pi k(1+2n)}{M+1}\,, \quad n = 0, \ldots, M$$

- Type IV linear phase Mth order FIR design (M odd, anti-symmetric filter): Let $D(k/N) = A[k]e^{2\pi\theta[k]}$ with $N = M+1$ and

$$\theta[k] = -\frac{1}{4} - \frac{kM}{2(M+1)} \qquad \text{for} \quad k = 1, \ldots, M$$

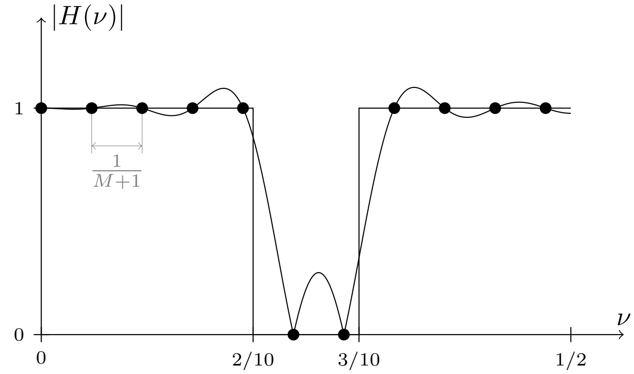
$$A[k] = A[M-k+1] \qquad \text{for} \quad k = 1, \ldots, (M+1)/2$$

$$A[0] = H(0) = 0$$

Then

$$h[n] = \frac{1}{M+1}\left[(-1)^n A[(M+1)/2] + 2\sum_{k=0}^{(M-1)/2} (-1)^k A[k] \sin\frac{\pi k(1+2n)}{M+1}\right]$$

- An example of a Type I band stop filter of order $M = 20$ is given by



4.3 Windowing

- The creation of an order M FIR filter using the window method is given by

$$h[n] = h_{\mathrm{I}}[n]w[n]$$

where

  - $h_{\mathrm{I}}[n]$ is the ideal (desired) filter impulse response (typically infinite length)
  - $w[n]$ is the FIR window function, $w[n] = 0$ for $n < 0$ and $n > M$

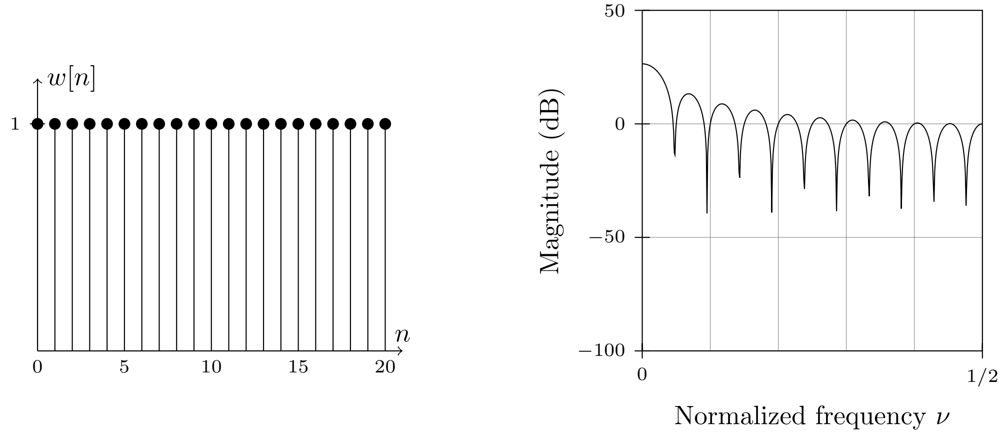
- The resulting frequency response is given by

$$H(\nu) = H_{\mathrm{I}}(\nu) \circledast W(\nu) = \int_{-1/2}^{1/2} H_{\mathrm{I}}(\nu - \tau)W(\tau)d\tau\,,$$

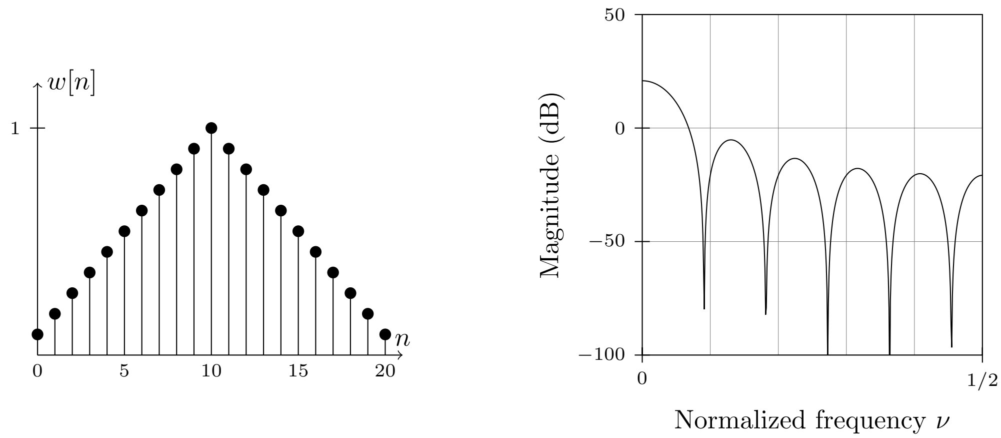
so the characteristics of the design are dictated by the window function (length and type)

- If the ideal response has linear phase, and the window function is real valued and symmetric, the resulting filter design will also have linear phase. Some example windows, and their DTFT (in dB scale) are given next.

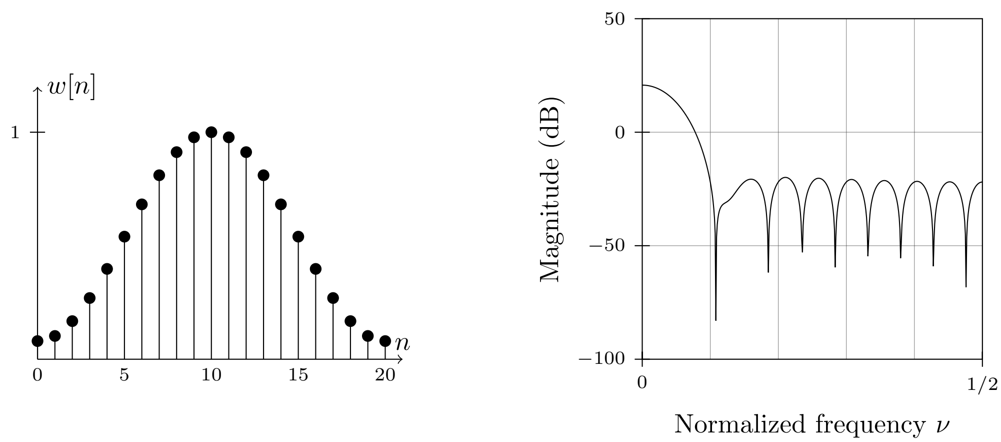
- Rectangular window of length $N = 21$



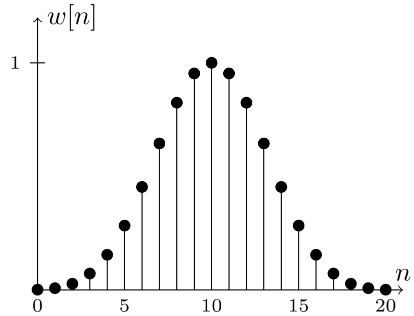
- Triangular window of length $N = 21$

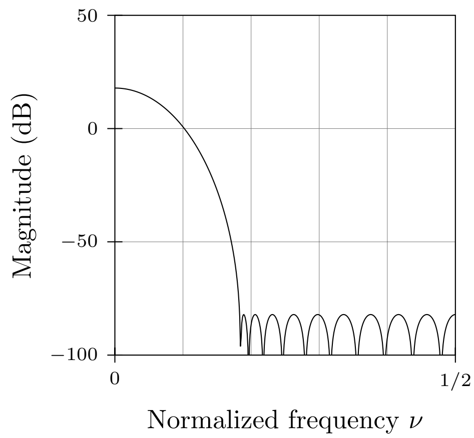


- Hamming window of length $N = 21$



- Chebyshev window of length $N = 21$





- Generally speaking, smoother window functions yield lower sildelobes, and longer windows (longer filters) yield more narrow main lobes (sharper transitions) in the frequency domain.

5 Nonparametric spectral estimation I

5.1 Periodogram

- The peridogram spectrum estimate is given by

$$\hat{P}_x(\nu) = \frac{1}{N}\left|\mathcal{F}\{x[n]w_{\mathrm{R}}[n]\}\right|^2$$

where $x[n]$ for $n = 0, \ldots, N-1$ is the given data (realization) and where

$$w_{\mathrm{R}}[n] = \begin{cases} 1 & 0 \leq n < N \\ 0 & \text{otherwise} \end{cases}$$

is the rectangular window function of length N.

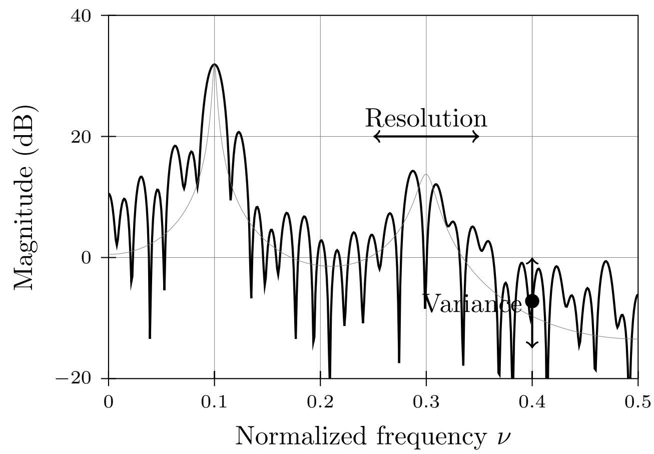
- The periodogram can also be obtained though the following procedure.
  - Estimate of ACF of $x[n]$ from N samples

$$\hat{r}[k] = \begin{cases} \frac{1}{N}\sum_{n=0}^{N-k-1} x[n]x[n+k] & k = 0, \ldots, N-1 \\ \hat{r}^*[-k] & k = -1, \ldots, -(N-1) \\ 0 & |k| > 0 \end{cases}$$

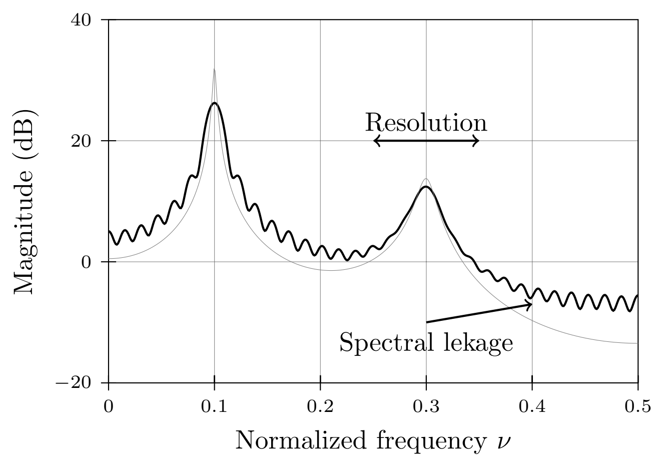
  - Obtain the power spectral density (PSD) estimate by

$$\hat{P}_x(\nu) = \mathcal{F}\{\hat{r}_x[k]\} = \sum_{k=-\infty}^{\infty} \hat{r}_x[k]e^{-j2\pi\nu k}$$

- Both methods are equivalent in terms of the estimate obtained, but the first one is more straightforward to implement and require less computations.
- An example of a periodogram estimate obtained from $N = 64$ samples from an AR4 process is



and the ensemble average (over all possible realizations of the samples) is for the same case given by



- Defining properties of the periodogram are
  - Bias
    $$\mathrm{E}\{\hat{P}_x(\nu)\} = P_x(\nu) \circledast \frac{1}{N}|W_\mathrm{R}(\nu)|^2 = \frac{1}{N}\int_{-1/2}^{1/2} P_x(\tau)|W_\mathrm{R}(\nu-\tau)|^2 d\tau$$
    where $W_R(\nu) = \mathcal{F}\{w_\mathrm{R}[n]\}$,
    $$w_\mathrm{R}[n] = \begin{cases} 1 & n = 0, \dots, N-1 \\ 0 & \text{otherwise} \end{cases}, \qquad |W_\mathrm{R}(\nu)|^2 = \left[\frac{\sin(\pi N \nu)}{\sin(\pi\nu)}\right]^2$$
  - Variance
    $$\mathrm{Var}\{\hat{P}_x(\nu)\} \approx P_x^2(\nu)$$
    where the approximation is good for “noisy” AR type processes and large N
- The properties of the rectangular window determine the resolution and spectral leakage though the bias expression above. These can be altered by choosing another window in the modified periodogram. The variance can be adressed by using methods such as Bartlett’s, Welch’s, or Blackman-Tukey’s methods.

5.2 Modified periodogram

- The modified periodogram is given by

$$\hat{P}_x(\nu) = \frac{1}{NU}\left|\mathcal{F}\{x[n]w[n]\}\right|^2 = \frac{1}{NU}\left|\sum_{n=0}^{N-1} x[n]w[n]e^{-j2\pi\nu n}\right|^2$$

  where

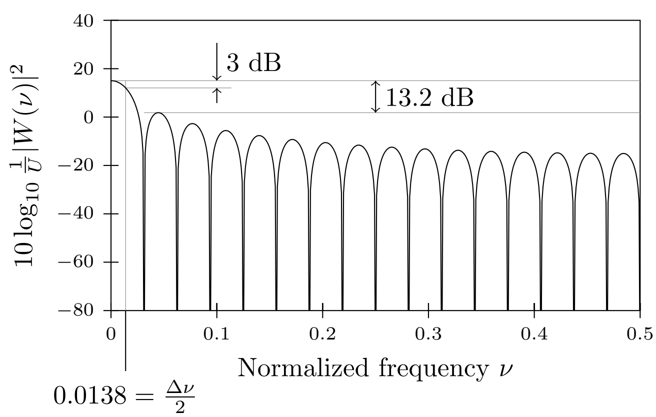
$$U = \frac{1}{N}\sum_{n=0}^{N-1}\left|w[n]\right|^2$$

  and where $w[n]$ is a suitably chosen window function.

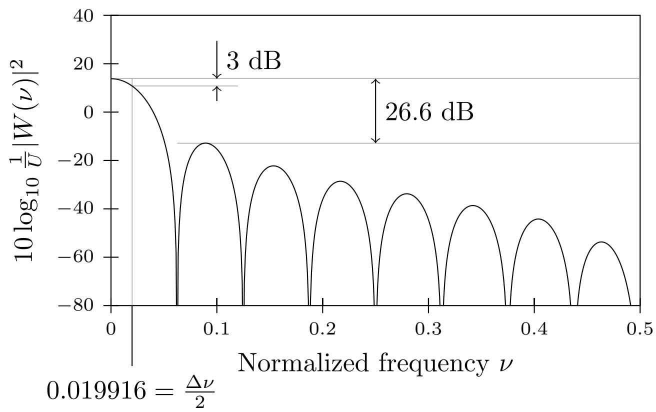
- Defining properties of the modified periodogram are
  - Bias
  $$\mathrm{E}\{\hat{P}_x(\nu)\} = P_x(\nu) \circledast \frac{1}{NU}|W(\nu)|^2$$
    where $W(\nu) = \mathcal{F}\{w[n]\}$.
  - Variance
  $$\mathrm{Var}\{\hat{P}_x(\nu)\} \approx P_x^2(\nu)$$
    where the approximation is good for "noisy" AR type processes and large N

- The chosen window function ultimately determines the modified periodogram spectrum estimate's properties in the frequency domain though the bias formula. Some examples of common window functions (in the frequency domain) are shown next.

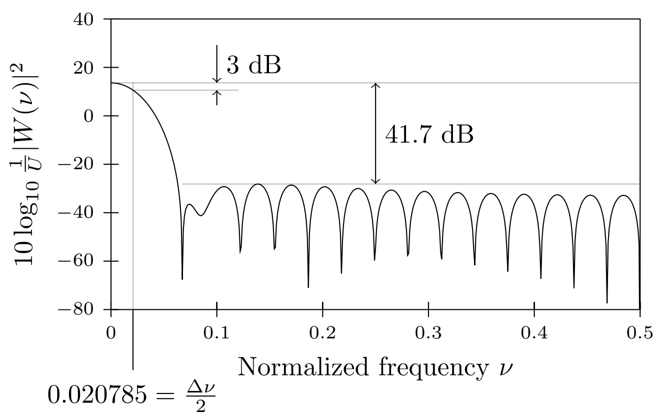
- Rectangular window of length $N = 32$



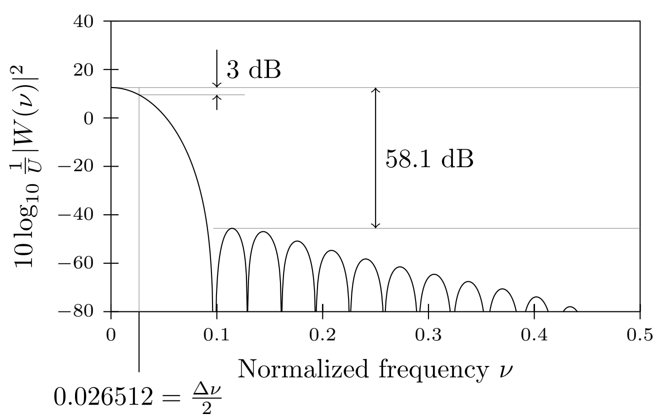
- Triangular window of length $N = 32$



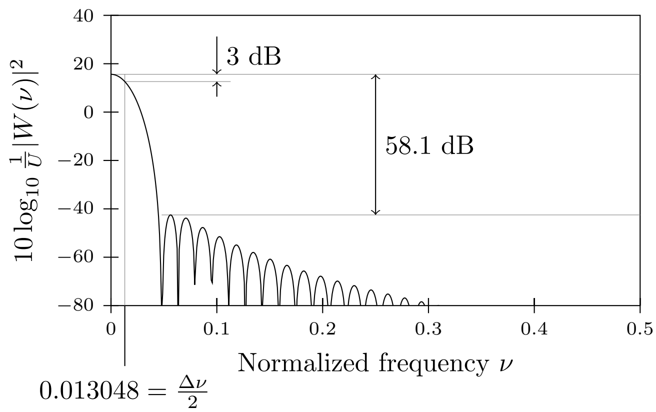
- Hamming window of length $N = 32$



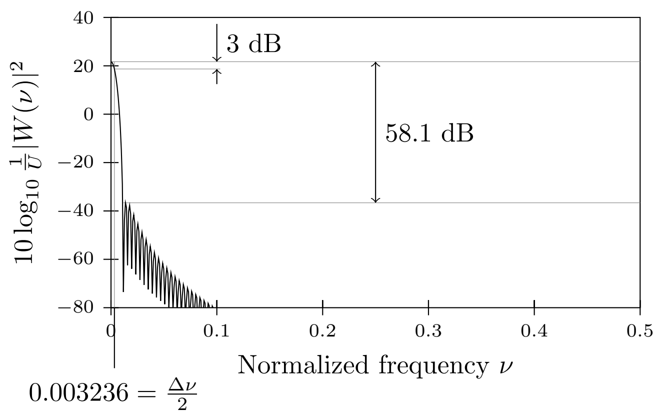
- Blackman window of length $N = 32$



- Blackman window of length $N = 64$



- Blackman window of length $N = 256$



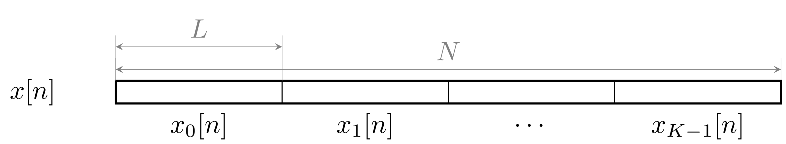
- Key properties of some common windows are

Window	Sidelobe level (dB)	3 dB BW $\Delta\nu$
Rectangular	-13	$0.89/N$
Bartlett	-27	$1.28/N$
Hamming	-43	$1.30/N$
Blackman	-58	$1.68/N$

6 Nonparametric spectral estimation II

6.1 Bartlett's method

- Bartlett's method divides the data into blocks of equal length, and averages the periodograms computed for each individual block. The method can be described as follows

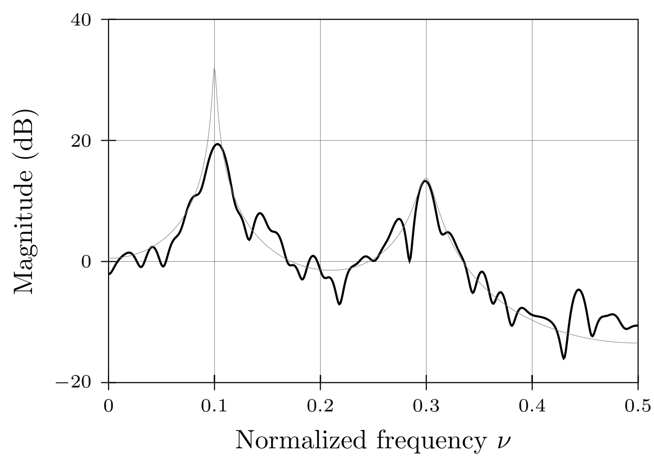


  - Choose K and L such that $KL = N$

- Create K length L blocks $x_k[n] = x[kL+n]$, $n = 0, \ldots, L-1$, $k = 0, \ldots, K-1$
- Average length L periodograms to get

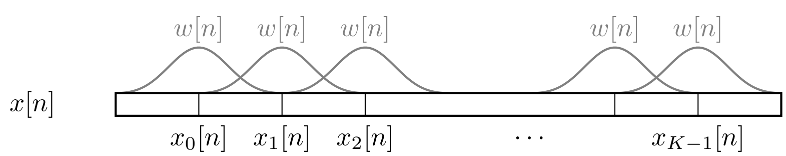
$$\hat{P}_x^{\mathrm{B}}(\nu) = \frac{1}{K} \sum_{k=0}^{K-1} \frac{1}{L} \big| \mathcal{F}\{x_k[n]\} \big|^2$$

- Defining properties of Bartlett's method are
  - Bias
$$\mathrm{E}\{\hat{P}_x^{\mathrm{B}}(\nu)\} = P_x(\nu) \circledast \frac{1}{L} |W_{\mathrm{R}}^{(L)}(\nu)|^2 \quad \text{where} \quad W_{\mathrm{R}}^{(L)}(\nu) = \frac{\sin(\pi L \nu)}{\sin(\pi \nu)}$$
  - Variance
$$\mathrm{Var}\{\hat{P}_x^{\mathrm{B}}(\nu)\} \approx \frac{1}{K} P_x^2(\nu)$$
    where the approximation is good for "noisy" AR type processes and sufficiently large L
  - The resolution is the same as for the length L periodogram
- An example spectrum estimate computed via Bartlett's method for the AR4 process considered before is given below for the parameters $N = 1024$, $L = 64$, $K = 16$.



6.2 Welch's method

- Welch's method divides the data into *overlapping* blocks of equal length, and averages modified periodograms computed for each individual block. The method can be described as follows



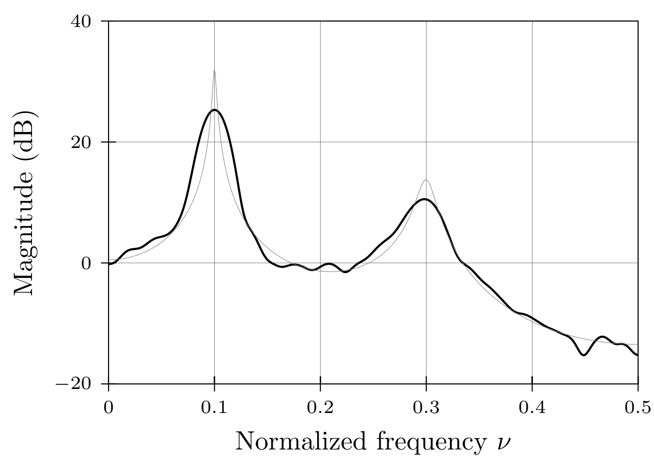
  - Choose K, L, and D such that $KL \geq N$ and $\frac{1}{2}L \leq D \leq L$
  - Create K length L blocks $x_k[n] = x[kD+n]$, $n = 0, \ldots, L-1$, $k = 0, \ldots, K-1$

- Average length L *modified* periodograms to get

$$\hat{P}_x^{\mathrm{W}}(\nu) = \frac{1}{K}\sum_{k=0}^{K-1}\frac{1}{LU}\left|\mathcal{F}\{x_k[n]w[n]\}\right|^2, \qquad U = \frac{1}{L}\sum_{n=0}^{L-1} w^2[n]\,.$$

- Defining properties of Welch's method are
  - Bias
$$\mathrm{E}\{\hat{P}_x^{\mathrm{W}}(\nu)\} = P_x(\nu) \circledast \frac{1}{LU}|W^{(L)}(\nu)|^2 \quad \text{where} \quad W^{(L)}(\nu) = \mathcal{F}\{w[n]\}$$
  - Variance (assuming a triangular window and 50% overlap)
$$\mathrm{Var}\{\hat{P}_x^{\mathrm{W}}(\nu)\} \approx \frac{9}{8K}P_x^2(\nu)$$
    where the approximation is good for "noisy" AR type processes and sufficiently large L
  - The resolution is the same as for the length L modified periodogram

- An example spectrum estimate computed via Bartlett's method for the AR4 process considered before is given below for the parameters $N = 1024$, $L = 64$, $D = 32$, $K = 31$, and a Hamming window in the modified periodogram.



6.3 Blackman-Tukeys's method

- The conceptual view of Blackman-Tukey is

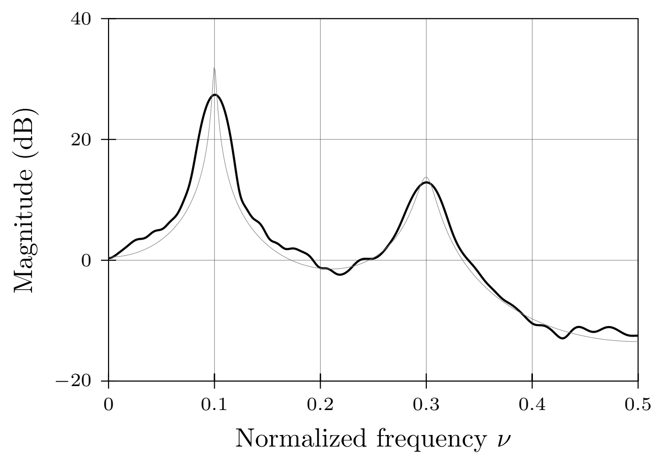
$$\hat{P}_x^{\mathrm{BT}}(\nu) = \hat{P}_x^{\mathrm{Per}}(\nu) \circledast W(\nu)$$

- The straightforward implementation of Blackman-Tukey is given by

$$\hat{P}_x^{\mathrm{BT}}(\nu) = \mathcal{F}\{\hat{r}[k]w[k]\}$$

  where $w[k]$ is symmetric around $k = 0$ and $w[k] = 0$ for $|k| \geq M$ for a given M
  - we generally want to choose $w[k]$ such that $W(\nu) \geq 0$
  - M is often referred to as the maximum lag of Blackman-Tukey's method
  - for proper power normalization we require that $w[0] = 1$

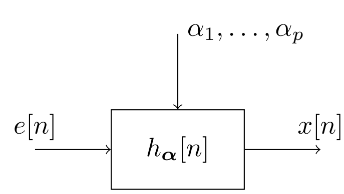
- Defining properties of Blackman-Tukey's method are
  - Bias
  $$\mathrm{E}\{\hat{P}_x^{\mathrm{BT}}(\nu)\} \approx P_x(\nu) \circledast W^{(2M+1)}(\nu) \quad \text{where} \quad W^{(2M+1)}(\nu) = \mathcal{F}\{w[n]\}$$
  - Variance
  $$\mathrm{Var}\{\hat{P}_x^{\mathrm{BT}}(\nu)\} \approx \left[\frac{1}{N}\sum_{k=-M}^{M} w^2[k]\right] P_x^2(\nu)$$
    where the approximation is good for "noisy" AR type processes.
  - The reduction for the variance can be computed explicitly as
  $$\frac{1}{N}\sum_{k=-M}^{M} w^2[k] = \begin{cases} 2M/N & \text{rectangular window} \\ 2M/3N & \text{triangular window} \end{cases}$$
  - The resolution is proportional to $1/M$ and ultimately determined by the window function used
- An example spectrum estimate computed via Blackman-Tukey's method for the AR4 process considered before is given below for the parameters $N = 1024$, $M = 64$, and a Blackman window



7 Parametric spectral estimation

7.1 ARMA models

- Model based parametric spectrum estimation is schematically illustrated below, where $x[n]$is the modelled process and $e[n]$ is unit variance white and zero mean *driving noise.*



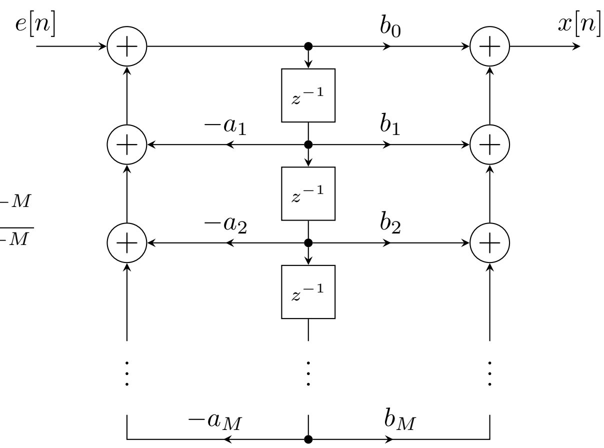
$$
\begin{tikzcd}
 & & \hat{P}_x(\nu) \arrow[dd, leftrightarrow] \\
x[0], \ldots, x[N-1] \arrow[r] & \hat{\alpha}_1, \ldots, \hat{\alpha}_p \arrow[ur, leftrightarrow] \arrow[dr, leftrightarrow] & \\
 & & \hat{r}_x[k]
\end{tikzcd}
$$

- The autoregressive moving average (ARMA) model is given by

$$\boldsymbol{\alpha} = (a_1, \ldots, a_M, b_0, \ldots, b_M)$$

$$H_{\boldsymbol{\alpha}}(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_M z^{-M}}$$

$$\Rightarrow \quad h_{\boldsymbol{\alpha}}[n]$$



- For $\boldsymbol{\alpha} = (a_1, \ldots, a_M, b_0, \ldots, b_M)$, the power spectrum of the ARMA process is

$$P_x(\nu) = \frac{|\sum_{m=0}^{M} b_m e^{-j2\pi\nu m}|^2}{|1 + \sum_{m=1}^{M} a_m e^{-j2\pi\nu m}|^2}$$

- The Yule-Walker equations for a general ARMA process are

$$r_x[k] + \sum_{m=1}^{M} a_m r_x[k-m] = \sum_{m=0}^{M} b_m h_{\boldsymbol{\alpha}}[m-k]\,, \qquad k \in \mathbb{Z}$$

- General idea behind parametric ARMA spectral estimate
  1. Obtain an estimate $\hat{r}_x[k]$ of the ACF $r_x[k]$ from the data $\{x[0], \ldots, x[N-1]\}$
  2. Obtain a parameter estimate $\hat{\boldsymbol{\alpha}} = (\hat{a}_1, \ldots, \hat{a}_M, \hat{b}_0, \ldots, \hat{b}_M)$ by solving

  $$\hat{r}_x[k] + \sum_{m=1}^{M} \hat{a}_m \hat{r}_x[k-m] = \sum_{k=0}^{M} \hat{b}_k h_{\hat{\boldsymbol{\alpha}}}[m-k]$$

     for some finite set of $k \in \mathbb{Z}$
  3. Obtain power spectrum estimate by evaluating

  $$\hat{P}_x(\nu) = \frac{|\sum_{m=0}^{M} \hat{b}_m e^{-j2\pi\nu m}|^2}{|1 + \sum_{m=1}^{M} \hat{a}_m e^{-j2\pi\nu m}|^2}\,.$$

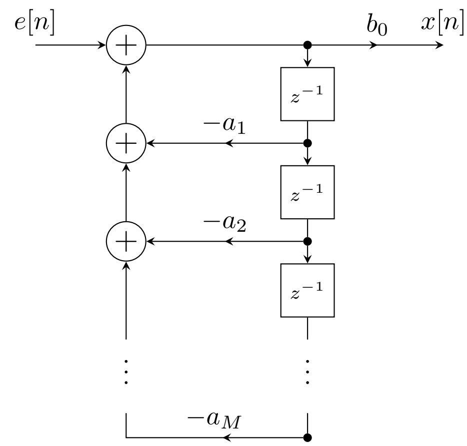
7.2 AR models

- The purely autoregressive (AR) model is given by

$$\boldsymbol{\alpha} = (a_1, \ldots, a_M, b_0)$$

$$H_{\boldsymbol{\alpha}}(z) = \frac{b_0}{1 + a_1 z^{-1} + \cdots + a_M z^{-M}}$$

$$\Rightarrow \quad h_{\boldsymbol{\alpha}}[n]$$



- The Yule-Walker equations for pure AR process are

$$r_x[k] + \sum_{m=1}^{M} a_m r_x[k-m] = b_0^2 \delta[k]\,, \qquad k \in \mathbb{Z}$$

which for $k = 1, \ldots, M$ simplifies to

$$r_x[k] + \sum_{m=1}^{M} a_m r_x[k-m] = 0\,,$$

which can be used to solve for $a_1, \ldots, a_M$, and which for $k = 0$ simplifies to

$$b_0 = \sqrt{r_x[0] + \sum_{m=1}^{M} a_m r_x[m]}\,,$$

which yields b_0.

- The *autocorrelation method* estimates the autocorrelation according to

$$\hat{r}[k] = \begin{cases} \frac{1}{N} \sum_{n=0}^{N-k-1} x[n]x[n+k] & k = 0, \ldots, N-1 \\ \hat{r}[-k] & k = -1, \ldots, -(N-1) \\ 0 & |k| > |N| \end{cases}$$

and solves the Yule-Walker equations with $\hat{r}_x[k]$ in place of $r_x[k]$ for estimates of the model parameters $a_1, \ldots, a_M, b_0)$, for some model order M under the assumption that $N \geq M$.

7.3 MMSE prediction

- Consider the problem of estimating a random variable $y \in \mathbb{R}$ from a random vector $\mathbf{x} \in \mathbb{R}^n$.
  - A linear detector has the form $\hat{y} = \mathbf{w}^{\mathrm{T}} \mathbf{x}$ where $\mathbf{w} \in \mathbb{R}^n$ is a set of weights.

- The best linear detector, in the mean square sense, is given by the solution to

$$\mathbf{R}_x \mathbf{w} = \mathbf{r}_{yx} \qquad \Leftrightarrow \qquad \mathbf{w} = \mathbf{R}_x^{-1} \mathbf{r}_{yx}$$

  where $\mathbf{R}_x = E\{\mathbf{x}\mathbf{x}^{\mathrm{T}}\}$ and $\mathbf{r}_{yx} = E\{y\mathbf{x}\}$.

- The minimum mean square error (MSE) is given by

$$E\{y^2\} - \mathbf{r}_{yx}^{\mathrm{T}} \mathbf{R}_x^{-1} \mathbf{r}_{yx}$$

- The coefficients $\mathbf{w} \in \mathbb{R}^M$ of the Mth order linear minimum mean square error 1-step ahead predictor are related to the ARM parameters $\mathbf{a} \in \mathbb{R}^M$ as $\mathbf{w} = -\mathbf{a}$. Moreover, for the ARM model we have

$$b_0^2 = r_x[0] + \sum_{m=1}^{M} a_m r_x[m] = r_x[0] + \mathbf{a}^{\mathrm{T}} \mathbf{r}_{yx}$$

  and for the 1-step ahead predictor we have

$$\mathrm{MSE} = E\{y^2\} - \mathbf{r}_{yx}^{\mathrm{T}} \mathbf{R}_x^{-1} \mathbf{r}_{yx} = E\{y^2\} - \mathbf{w}^{\mathrm{T}} \mathbf{r}_{yx}$$

  if $\mathbf{w} = \mathbf{R}_x^{-1} \mathbf{r}_{yx}$, which implies that $b_0^2 = \mathrm{MSE}$ because $E\{y^2\} = r_x[0]$ and $\mathbf{w} = -\mathbf{a}$.

- One may choose AR model order by increasing M until no significant further reduction in b_0^2 is observed.

- Assume that N is the number of data-samples, and $b_{0,M}^2$ is driving noise power for the M-th order AR model, then two famous model order selection criteria are

  - Akaike information criterion (AIC): Choose M to minimize

$$\mathrm{AIC}(M) = \ln b_{0,M}^2 + \frac{2M}{N}$$

  - Minimum description length (MDL): Choose M to minimize

$$\mathrm{MDL}(M) = N \ln b_{0,M}^2 + M \ln N$$

8 Multirate signal processing: Decimation and Interpolation

8.1 Downsampling

- *Downsampling* by an integer factor of D is the operation given by

$$y[m] = x[mD] \qquad \forall m \in \mathbb{Z}\,,$$

  which is schematically drawn as

$$x[n] \longrightarrow \boxed{\downarrow D} \longrightarrow y[m]$$

- In terms of the DTFT it holds that

$$Y(\nu) = \frac{1}{D} \sum_{k=0}^{D-1} X\left(\frac{\nu - k}{D}\right) .$$

- In terms of the bilateral z-transform it holds that

$$Y(z) = \frac{1}{D} \sum_{k=0}^{D-1} X\left(z^{1/D} e^{-j2\pi k/D}\right) .$$

- *Proper decimation* refers to the operation

$$\xrightarrow{x[n]} \boxed{h[n]} \xrightarrow{z[n]} \boxed{\downarrow D} \xrightarrow{y[m]}$$

where

$$z[n] = h[n] * x[n] , \qquad y[m] = z[mD] \qquad \forall m \in \mathbb{Z}$$

and

$$\mathcal{F}\{h[n]\} = H(\nu) = \begin{cases} 1 & |\nu| \leq \frac{1}{2D} \\ 0 & \frac{1}{2D} \leq |\nu| \leq \frac{1}{2} \end{cases}$$

- For proper decimation it holds that

$$Y(\nu) = \frac{1}{D} X\left(\frac{\nu}{D}\right) , \quad |\nu| \leq \frac{1}{2} ,$$

and for a general (possibly imperfect) decimation filter with frequency response $H(\nu)$ it holds that

$$Y(\nu) = \frac{1}{D} \sum_{k=0}^{D-1} H\left(\frac{\nu - k}{D}\right) X\left(\frac{\nu - k}{D}\right) .$$

8.2 Upsampling

- *Upsampling* by an integer factor of U is the operation given by

$$y[m] = \begin{cases} x[m/U] & m/U \in \mathbb{Z} \\ 0 & m/U \notin \mathbb{Z} \end{cases}$$

which is schematically drawn as

$$\xrightarrow{x[n]} \boxed{\uparrow U} \xrightarrow{y[m]}$$

- In terms of the DTFT it holds that

$$Y(\nu) = X(\nu U) .$$

- In terms of the bilateral z-transform it holds that

$$Y(z) = X\left(z^U\right) .$$

- *Proper interpolation* refers to the operation

$$\xrightarrow{x[n]} \boxed{\uparrow U} \xrightarrow{z[m]} \boxed{h[m]} \xrightarrow{y[m]}$$

where

$$z[m] = \begin{cases} x[m/U] & m/U \in \mathbb{Z} \\ 0 & m/U \notin \mathbb{Z} \end{cases}, \qquad y[m] = h[m] * z[m] \qquad \forall m \in \mathbb{Z}$$

and

$$\mathcal{F}\{h[m]\} = H(\nu) = \begin{cases} U & |\nu| \le \frac{1}{2U} \\ 0 & \frac{1}{2U} \le |\nu| \le \frac{1}{2} \end{cases}$$

- For proper interpolation it holds that

$$Y(\nu) = \begin{cases} UX(U\nu) & |\nu| \le \frac{1}{2U} \\ 0 & \frac{1}{2U} \le |\nu| \le \frac{1}{2} \end{cases},$$

and for a general (possibly imperfect) interpolation filter with frequency response $H(\nu)$ it holds that

$$Y(\nu) = H(\nu)X(U\nu)\,.$$

8.3 Upsampling and downsampling stochastic signals

- Downsampling a wide sense stationary signal $x[n]$ to $y[m]$ with a factor D gives

$$P_y(\nu) = \frac{1}{D}\sum_{k=0}^{D-1} P_x\left(\frac{\nu - k}{D}\right)$$

- Decimating $x[n]$ to $y[m]$ with a filter $H(\nu)$ gives

$$P_y(\nu) = \frac{1}{D}\sum_{k=0}^{D-1} P_x\left(\frac{\nu - k}{D}\right)\left|H\left(\frac{\nu - k}{D}\right)\right|^2$$

- Proper decimation with a factor D gives

$$P_y(\nu) = \frac{1}{D}P_x\left(\frac{\nu}{D}\right), \quad |\nu| \le \frac{1}{2D}$$

- Upsampling a wide sense stationary signal $x[n]$ to $y[m]$ with a factor U gives (after the inclusion of a uniform random time delay $\Theta \sim \mathcal{U}[0, 1, \ldots, U-1]$ needed to ensure wide sense stationarity in $y[m]$)

$$P_y(\nu) = \frac{1}{U}P_x(U\nu)$$

- Interpolating $x[n]$ to $y[m]$ with a filter $H(\nu)$ gives

$$P_y(\nu) = \frac{1}{U}P_x(U\nu)|H(\nu)|^2$$

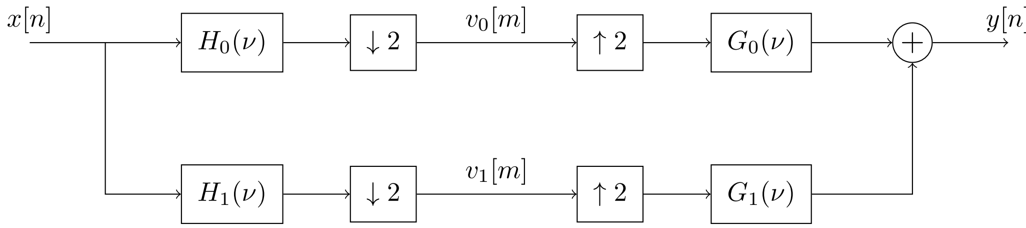
- Proper interpolation with a factor U gives

$$P_y(\nu) = \begin{cases} UP_x(U\nu) & |\nu| \le \frac{1}{2U} \\ 0 & \frac{1}{2U} \le |\nu| \le \frac{1}{2} \end{cases}$$

9 Multirate signal processing: Filter banks

9.1 Filterbanks

- A two branch filterbank, with the task of splitting the signal $x[n]$ into two half rate signals $v_0[m]$ and $v_1[m]$ typically representing the low and high frequency content of the signal repspectively, is illustated below.



- The input output relation of a two branch filter bank is under the bilateral z-transform given by

$$\begin{aligned} Y(z) =& \frac{1}{2}\Big[G_0(z)H_0(z) + G_1(z)H_1(z)\Big]X(z) \\ &+ \underbrace{\frac{1}{2}\Big[G_0(z)H_0(-z) + G_1(z)H_1(-z)\Big]X(-z)}_{\text{aliasing}} \ . \end{aligned}$$

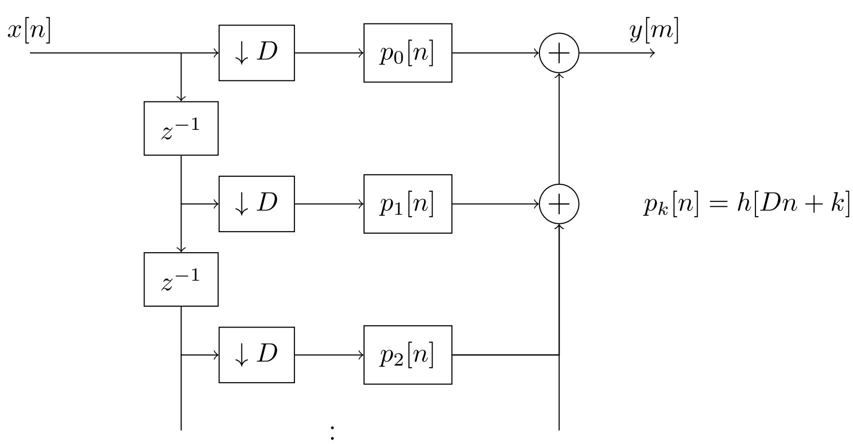
- Perfect reconstruction with delay L, i.e., $y[n] = x[n-L]$, is achieved if

$$\begin{aligned} G_0(z)H_0(z) + G_1(z)H_1(z) &= 2z^{-L} \\ G_0(z)H_0(-z) + G_1(z)H_1(-z) &= 0 \end{aligned}$$

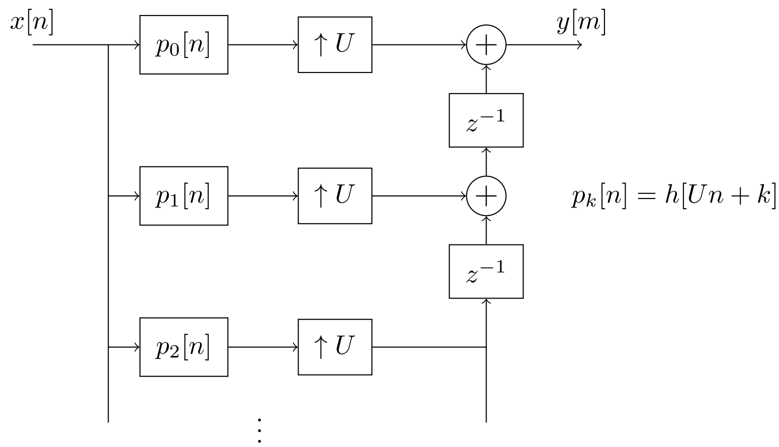
- The symmetry assumptions $G_0(z) = H_1(-z)$ and $G_1(z) = -H_0(-z)$ are sufficient conditions for alias canceling.
- The quadrature mirror filter (QMF) design satisfies the symmetry assumptions above with the filters chosen as $H_0(z) = H(z)$, $H_1(z) = H(-z)$, $G_0(z) = H(z)$ and $G_1(z) = -H(-z)$ for some base (typically low pass) filter $H(z)$.

9.2 Polyphase implementations

- The polyphase implementation of a decimation circuit with downsampling factor D and decimation (anti aliasing) filter $h[n]$ is shown below.



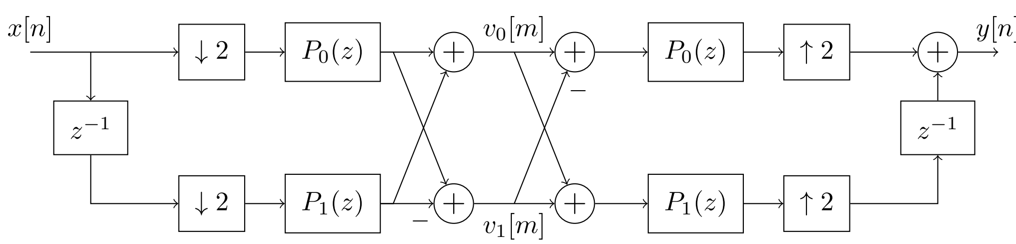
- The polyphase implementation of an interpolation circuit with upsampling factor U and decimation (anti aliasing) filter $h[n]$ is shown below.



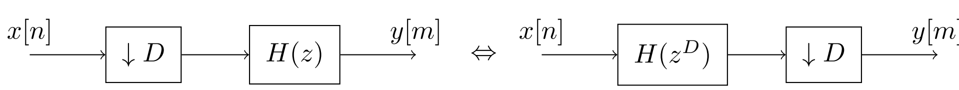
- The transfer function of the base filter $h[n]$ can be obtained as

$$H(z) = \sum_{k=0}^{K-1} z^{-k} P_k(z^K)\,.$$

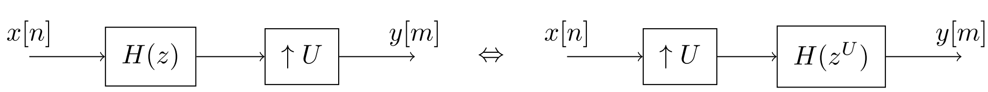
- The polyphase implementation of a two branch polyphase QMF bank is given by



- The noble identities are useful when working with multirate systems and are given schematically by



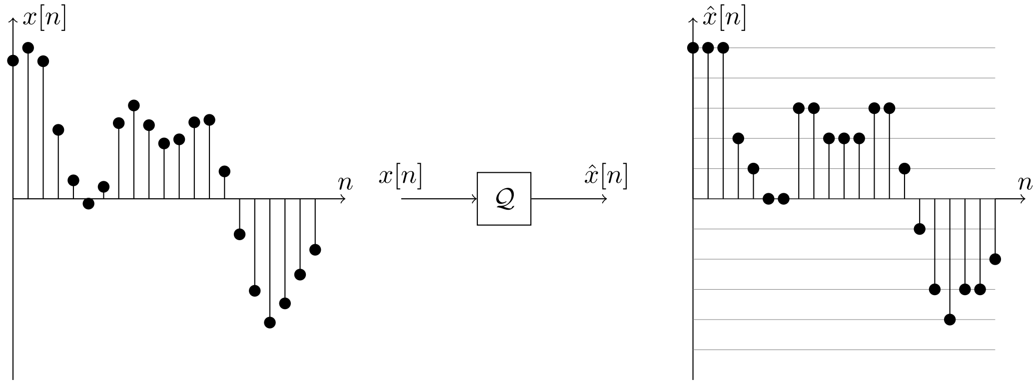
and

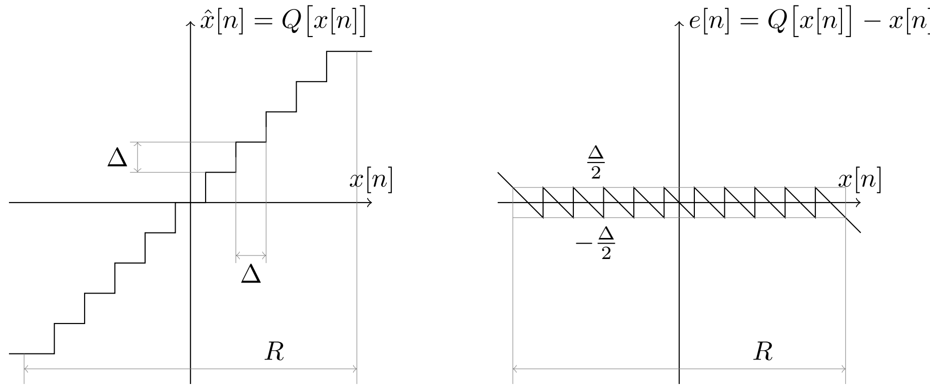


10 Quantization and finite-precision digital signal processing

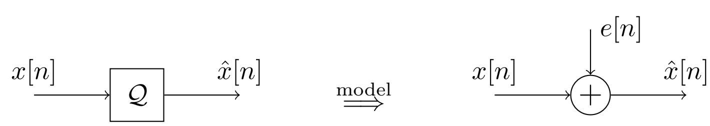
10.1 Quantization

- Signal (amplitude) quantization is illustrated below.





- The additive noise quantization model is given by



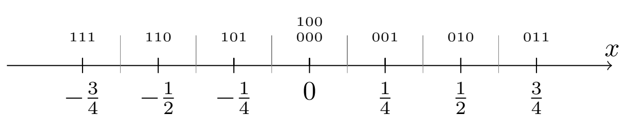
  where

  - $e[n]$ is a white (memoryless) stochastic process, $r_e[k] = E\{e[n]e[n-k]\} = \sigma_e^2 \delta[k]$
  - $e[n]$ is a independent of $x[n]$, $r_{ex}[k] = E\{e[n]x[n-k]\} = 0 \ \forall k$
  - $e[n]$ is uniformly distributed, $e[n] \sim \mathcal{U}[-\Delta/2, \Delta/2]$ which implies $\sigma_e^2 = E\{e^2[n]\} = \Delta^2/12$

- In the fixed point $B+1$ bit signed magnitude representation the collection of bits represents a real number $x \in (-1, 1)$ according to

$$x = (-1)^{b_0} \sum_{i=1}^{B} b_i 2^{-i} \quad \Leftrightarrow \quad (x)_2 = \begin{cases} +0.b_1 b_2 \cdots b_B & b_0 = 0 \\ -0.b_1 b_2 \cdots b_B & b_0 = 1 \end{cases}$$

$B = 2$

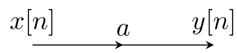


- In the fixed point $B+1$ we have $\Delta = 2^{-B}$ and

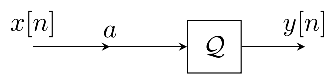
$$\sigma_e^2 = \frac{\Delta^2}{12} .$$

10.2 Fixed point implementations

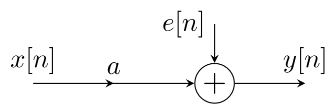
- The varying levels of modeling a fixed point circuit implementation are shown below.
  - Base implementation



  - Quantization based representation of fixed point implementation



  - Additive noise model for quantization noise



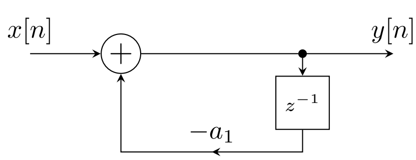
- Statistical characterization of fixed point implementations:
  1. Insert quantization noise source at each point of multiplication, $e_1[n], \ldots, e_M[n]$ for circuit with M total multiplications.
  2. Evaluate noise at output due to every single noise source as

$$\sigma_{e_i}^2 = \frac{2^{-2B}}{12} \sum_{k=0}^{\infty} h_i^2[n]$$

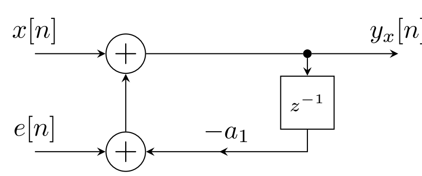
     where $h_i[n]$ is impulse response from noise source to output.
  3. Obtain total noise at output as

$$\sigma_e^2 = \sum_{i=1}^{M} \sigma_{e_i}^2$$

- An example statistical fixed point model of an AR1 filter is shown below.



Fixed point implementation



Statistical noise model

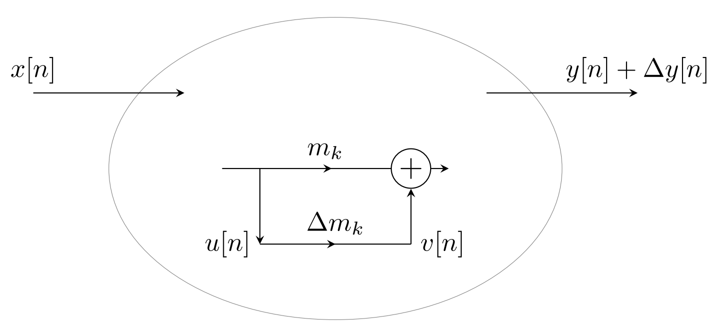
11 Fixed point filter implementations

11.1 Filter coefficient quantization

- A Type I linear phase FIR filter can be constructed as a cascade of smaller components of the form
  - Second order "Type I" component
  $$H_k(z) = c_{k0} + c_{k1}z^{-1} + c_{k0}z^{-2}$$
  - Fourth order "Type I" component
  $$H_k(z) = c_{k0} + c_{k1}z^{-1} + c_{k2}z^{-2} + c_{k1}z^{-3} + c_{k0}z^{-4}$$
  - These components are sufficient for realizing all Type I linear phase FIR filters
  - Quantization of c_{ki}, $i = 0, \ldots, 2$, maintains linear phase property
- For second order Type I factors, the transfer function
$$H_k(z) = c_{k0} + c_{k1}z^{-1} + c_{k0}z^{-2}\,,$$
where $c_{k0}, c_{k1} \in \mathbb{R}$, if z_k is a complex valued zero, $H_k(z_k) = 0$ and $\Im\{z_k\} \neq 0$, then $H_k(z_k^*) = 0$ and $|z_k| = 1$.
- For fourth order Type I factors, the transfer function
$$H_k(z) = c_{k0} + c_{k1}z^{-1} + c_{k2}z^{-2} + c_{k1}z^{-3} + c_{k0}z^{-4}$$
where $c_{k0}, c_{k1}, c_{k2} \in \mathbb{R}$, if z_k is a complex valued zero, $H_k(z_k) = 0$ and $\Im\{z_k\} \neq 0$, then $H_k(z_k^*) = 0$, $H_k(1/z_k) = 0$, and $H_k(1/z_k^*) = 0$.

11.2 Circuit sensitivity

- The sensitivity of $H(z)$ to perturbations in a filter coefficient m_k is given by
$$\frac{\partial}{\partial m_k} H(z) = F_k(z) G_k(z)$$
  - $F_k(z)$ is the transfer function from the input to before the multiplication
  - $G_k(z)$ is the transfer function from an input after the multiplication to the output



Transfer functions: $x[n] \to u[n] : F_k(z)\,,\quad v[n] \to y[n] : G_k(z)$

- Assume a rational transfer function given by

$$H(z) = \frac{B(z)}{A(z)}$$

where the zeros are roots of

$$B(z) = \sum_{k=0}^{M} b_k z^{-k} = b_0 \prod_{k=1}^{M} (1 - z_k z^{-1})$$

and where the poles are roots of

$$A(z) = 1 + \sum_{k=1}^{M} a_k z^{-k} = \prod_{k=1}^{M} (1 - p_k z^{-1})$$

  - If

    $$A(z) = 1 + \sum_{k=1}^{M} a_k z^{-k} = \prod_{k=1}^{M} (1 - p_k z^{-1})$$

    and $A(p_i) = 0$, then

    $$\frac{\partial p_i}{\partial a_k} = \frac{p_i^{M-k}}{\prod_{l \neq i} (p_i - p_l)}$$

    for $i = 1, \ldots, M$ and $k = 1, \ldots, M$.
  - If

    $$B(z) = \sum_{k=0}^{M} b_k z^{-k} = b_0 \prod_{k=1}^{M} (1 - z_k z^{-1})$$

    and $B(z_i) = 0$, then

    $$\frac{\partial z_i}{\partial b_k} = \frac{z_i^{M-k}}{b_0 \prod_{l \neq i} (z_i - z_l)}$$

    for $i = 1, \ldots, M$ and $k = 1, \ldots, M$.

- Many closely packed zeros or poles increases the sensitivity of the zeros or poles to perturbations (quantization) in the filter coefficients.