PS403 - Digital Signal processing

5. DSP - Non-Recursive (FIR) Digital Filters

Key Text:

Digital Signal Processing with Computer Applications (2nd Ed.)

Paul A Lynn and Wolfgang Fuerst, (Publisher: John Wiley & Sons, UK)

We will cover in this section

Finite Impulse Response (FIR) filter design

Windowing (Apodization) in DSP

Digital Differentiators

Non-Recursive (FIR) Digital Filters

We have already looked at some non-recursive filters in detail - e.g., Weighted Moving Average (Savitsky-Golay)

We will first consider what the impulse response of a 'perfect or ideal' low pass filter might look like.

Consider the ideal frequency response as shown in figure 5.1. Only the range 'Ω' range from 0 - π (dc - 2 samples/cycle) is unique for an adequately sampled signal. We will make it symmetric about the frequency or 'Ω' axis to simplify the mathmatics).

See Figure 5.1 in Lynn and Fuerst

We know that the general form of the difference equation for any digital filter is given by:

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

Non-Recursive (FIR) Digital Filters

Non-Recursive filters depend only on present and previous inputs $\Rightarrow$

$$y[n] = \sum_{k=0}^{k=M} b_k x[n-k]$$

Direct Convolution Sum

Hence to implement the filter we simply convolve the input signal with the coefficients b_k (which are just the successive terms of h[n] !!!!)

Since the number of terms 'M' must be finite, such filters belong to a class referred to as FINITE IMPULSE RESPONSE (FIR) Filters

Non-Recursive (FIR) Digital Filters

Hence:

I.R. h[n] $\rightarrow$ b_k

ZT(IR) H(Z) $\rightarrow$ $\sum_{k=o}^{M} b_k Z^{-k}$

FT(IR) H(Ω) $\rightarrow$ $\sum_{k=o}^{M} b_k \exp(-jk\Omega)$

The trick to designing FIR filters is to obtain the best approximation to an ideal H(Ω) with as few b_k's (or h[n] terms) as possible, typ. < 100 !

In practice FIR filters are slow (lot of computation - remember e.g., moving average) but they do have some nice redeeming properties

Non-Recursive (FIR) Digital Filters

Those properties are:

A recursive filter is specified in terms of Z-plane zeros only. Hence it is inherently stable since it has no poles at which H(Z) could 'blow up' !!

The non-recursive filter has a linear phase characteristic

As the non-recursive filter has a FIR (finite number of h[n] terms), it can be made symmetric about n = 0 which yields a zero phase characteristic

Consider the impulse response shown in **figure 5.3. (Lynn & Fuerst) -** *a symmetric h[n] with a zero phase characteristic*

The frequency transfer function is:

$$H(\Omega) = \sum_{k=o}^{M} b_k \exp(-jk\Omega)$$

Non-Recursive (FIR) Digital Filters

$$\Rightarrow H(\Omega) = b_0 + 2b_1 Cos\Omega + 2b_2 Cos2\Omega + 2b_3 Cos3\Omega + ...$$

$$= b_0 + 2\sum_{k=1}^{k=M} b_k Cosk\Omega$$

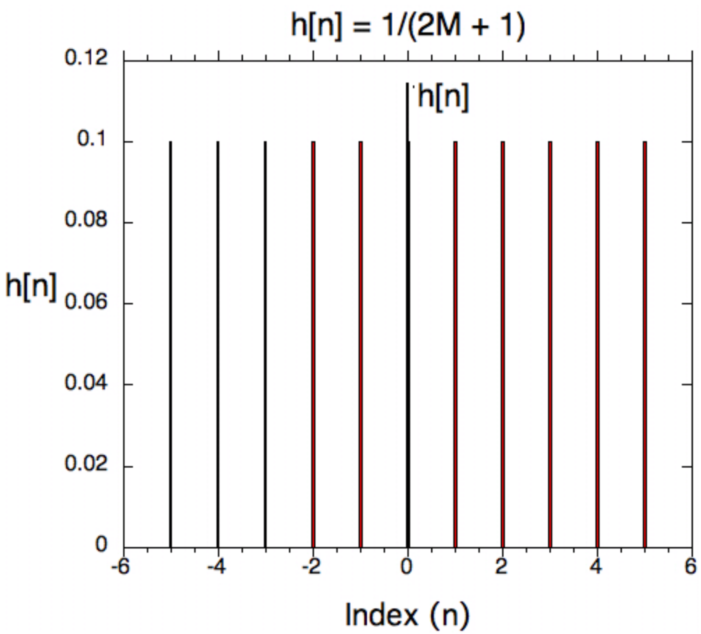
H(Ω) is a real function which implies a zero phase shift at ALL frequencies !

To make the filter CAUSAL we simply shift h[n] by 'M' sampling intervals - (see Figure 5.3b)

As a result |H(Ω)| remains unchanged but
$\Phi_H(\Omega)$ moves to a Linear Phase Characteristic

Non-Recursive (FIR) Digital Filters

Non-Causal, Low Pass Filter (Moving Average Impulse Response)



Consider: 2M + 1 point adjacent Channel averaging filter

$$H(\Omega) = \sum_{k=-M}^{M} b_k \exp(-jk\Omega)$$

$$\Rightarrow H(\Omega) = b_0 + 2b_1 Cos\Omega + 2b_2 Cos2\Omega + 2b_3 Cos3\Omega + \ldots\ldots\ldots$$

$$\Rightarrow H(\Omega) = \frac{1}{2M+1}\{1 + 2Cos\Omega + 2Cos2\Omega + 2Cos3\Omega + \ldots\ldots\ldots + 2CosM\Omega\}$$

Non-Recursive (FIR) Digital Filters

$$H(Z) = \sum_{k=-M}^{k=+M} b_k Z^{-k}$$

$$H(Z) = \frac{1}{5}\left\{Z^{2} + Z^{1} + Z^{0} + Z^{-1} + Z^{-2}\right\}$$

Shift h[n] forward to begin at n = 0, then:

$$\therefore H'(Z) = Z^{-2}H(Z)$$

$$H'(Z) = \frac{1}{5}\left\{1 + Z^{-1} + Z^{-2} + Z^{-3} + Z^{-4}\right\}$$

$$= \frac{1}{5}\left[\frac{1 + Z^{1} + Z^{2} + Z^{3} + Z^{4}}{Z^{4}}\right]$$

4th order pole at the origin

Non-Recursive (FIR) Digital Filters

Zero's occur at $Z = r.\exp(-j\Omega)$, $\Omega = 2\pi n/5$, n = 1, 2, 3 & 4.

Note that there is no zero at $\Omega = 0$ and $\Omega = 2\pi$, $H(\Omega)$ = maximum there !

The pole - zero plot for this filter is shown on figure 5.5 (a). One can tighten up the frequency response by increasing the value of 'M' at the expense of slowing down the settling time of the output.

The pole-zero plot for a 21 term filter is shown in figure 5.5 (b).

There is a 'missing' zero at $Z = 1$, $\theta = 0^{O}$ in each case. In this way, $|H(\Omega)|$ can have a maximum value at $\Omega = 0$. As all '2M' zeros lie on the unit circle, true nulls are obtained in $|H(\Omega)|$ for all $\Omega = 2\pi n/M$!

The 4th and 20th order poles at the origin have no effect on $|H(\Omega)|$

Derivation of Highpass & Bandpass Filters from the Basic Lowpass (Moving Average) Filter

We have looked at a simple moving average filter where the h[n] values have amplitudes of 1/(2M + 1). We see that its frequency response is that of a low pass filter with true nulls are frequencies of $2n\pi/M$.

We require a bandpass filter (BPF) with centre frequency Ω_0

We will use the simple moving average filter as the base to build it !

We infer from experience that such a bandpass filter should have a sinusoidal variation in its impulse response h[n] at frequency Ω_0.

So our first attempt at a BPF design should simple be to take the LPF response and multiply (modulate) it with a function like $\mathrm{Cos}(n\Omega_0)$!

Derivation of Highpass & Bandpass Filters from the Basic Lowpass (Moving Average) Filter

So we can write: $h[n]_{BPF} = h[n]_{LPF}.Cos(n\Omega_0)$

Example: Consider a BPF (21 term, M = 10) given by:

$$h[n] = \frac{1}{2M+1}.Cos\left(\frac{n\pi}{3}\right), -10 \le n \le 10$$

So a BPF with a centre frequency of 6 samples/cycle is required -

The effect is shown in figure 5.6(a) and the corresponding frequency response, obtained with the aid of Program 12 is shown in 5.6(b) -

It is clear that the frequency response is only an approximation (albeit not a bad one) to a BPF with a centre frequency $\Omega_0 = \pi/3$.

Derivation of Highpass & Bandpass Filters from the Basic Lowpass (Moving Average) Filter

NB: For a properly sampled signal we can tune Ω_0 from 0 radians (dc) to π radians (max - 2 samples/cycle) i.e.,
from a LPF through a BPF to a HPF !!

However the main problem with this approach is that we derive 'non-ideal' filters -

Better to turn the process around and **derive (compute) the impulse response h[n] corresponding to the desired H(Ω)**!

Derivation of Non-Recursive Filters using the Fourier Transform Method

Reminder: FT pair - for a discrete (sampled) signal array x[n], it's FT is:

$$X(\Omega) = \sum_{n=-\infty}^{n=\infty} x[n].\exp(-jn\Omega)$$

We can construct the original signal x[n] from its FT using:

$$x[n] = \frac{1}{2\pi}\int_{2\pi} X(\Omega).\exp(jn\Omega)\,d\Omega$$

Or, for a processor, its impulse response h[n] can be obtained from:

$$h[n] = \frac{1}{2\pi}\int_{2\pi} H(\Omega).\exp(jn\Omega)\,d\Omega$$

Derivation of Non-Recursive Filters using the Fourier Transform Method

The key idea here is: Write out the $H(\Omega)$ you desire (ideal) - then derive the corresponding h[n] which in turn yields the b_k coefficients of the corresponding non-recursive filter

Two Problems:

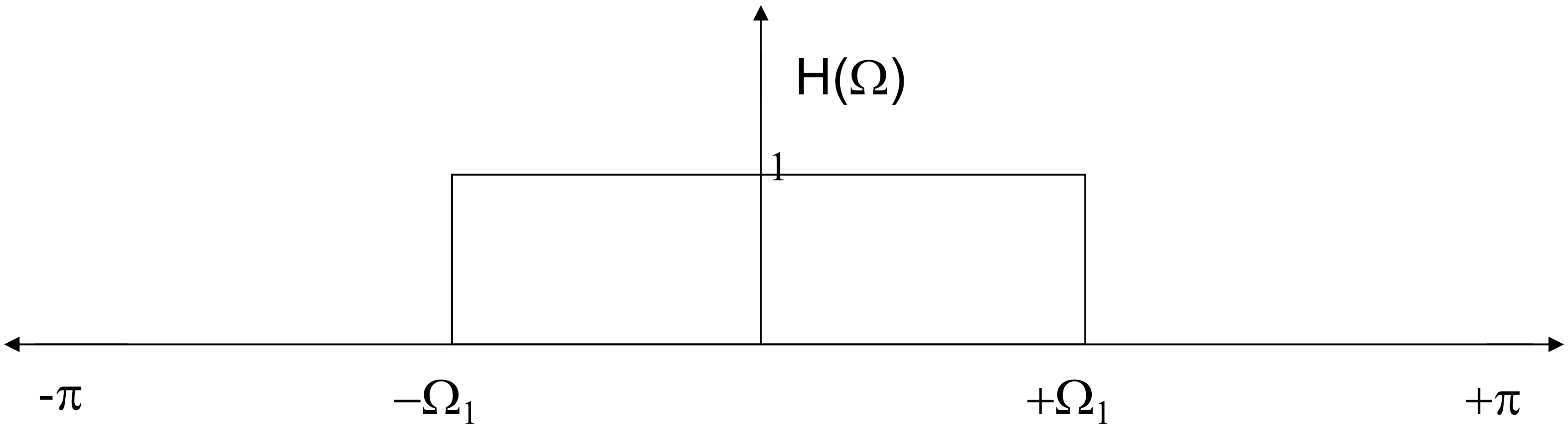
1. It is not always possible to evaluate the integral expression:

$$h[n] = \frac{1}{2\pi} \int_{2\pi} H(\Omega).\exp(jn\Omega)\,d\Omega$$

2. The number of h[n] terms: A desirable $H(\Omega)$ may give rise to a h[n] with many terms and hence a computationally expensive solution - a CPU gas guzzler ! So a compromise between the $H(\Omega)$ desired and a tractable no of terms in the corresponding h[n] must be made.

In relation to point 1 - we always choose simple, ideal filters with zero or Linear Phase Characteristics

Derivation of Non-Recursive Filters using the Fourier Transform Method



Consider an ideal LPF with a cutoff frequency of Ω_1

centred about zero, ergo a zero phase characteristic and H(Ω) is real

$$h[n] = \frac{1}{2\pi}\int_{-\pi}^{+\pi} H(\Omega)\exp(jn\Omega)\,d\Omega$$

$$= \frac{1}{2\pi}\int_{-\Omega_1}^{+\Omega_1} 1.\exp(jn\Omega)\,d\Omega$$

Derivation of Non-Recursive Filters using the Fourier Transform Method

$$= \frac{1}{2\pi}\left[\frac{\exp(jn\Omega)}{jn}\right]_{-\Omega_1}^{+\Omega_1}$$

$$= \frac{1}{2\pi jn}\left\{\exp(jn\Omega_1) - \exp(-jn\Omega_1)\right\}$$

$$\Rightarrow h[n] = \frac{1}{2\pi jn}\left[Cos(n\Omega_1) + jSin(n\Omega_1) - Cos(n\Omega_1) + jSin(n\Omega_1)\right]$$

$$\therefore h[n] = \frac{\Omega_1}{\pi}\frac{Sin(n\Omega_1)}{n\Omega_1} = \frac{\Omega_1}{\pi} Sinc(n\Omega_1)$$

Derivation of Non-Recursive Filters using the Fourier Transform Method

Not an unexpected result - the FT of a rectangular function (waveform) in one domain (space, time etc.) is a 'Sinc' function in the complementary domain - e.g., space domain - FT of a slit (x) is a Sinc (f_x)

Example: Find and sketch the impulse response h[n] of an ideal Low Pass Filter (LPF) with cutoff frequency $\Omega_1 = \pi/5$ (i.e., 10 samples/cycle)

$$h[n] = \frac{1}{n\pi} Sin\left(\frac{n\pi}{5}\right)$$

What is h[n] @ n = 0 ? - Use L'Hopital's Rule

It state that given: $h[x] = \dfrac{y(x)}{z(x)}, \Rightarrow h[0] = \left. \dfrac{\frac{dy}{dx}}{\frac{dz}{dx}} \right|_{n=0}$

Derivation of Non-Recursive Filters using the Fourier Transform Method

$$\Rightarrow h[0] = \left. \frac{\frac{d}{dn}\left\{ Sin\left(n\pi/5 \right) \right\}}{\frac{d}{dn}\left\{ n\pi \right\}} \right|_{n=0}$$

$$= \left. \frac{\pi/5 \; Cos\left(n\pi/5 \right)}{\pi} \right|_{n=0} \quad = \frac{1}{5} = 0.2$$

n	0	1	2	3	4	5	6	7	8
h[n]	0.20000	0.187098	0.151365	0.100910	0.046774	0.00000	-0.031183	-0.043247	-0.037841

Cf: Figure 5.8(a) of Lynn and Fuerst

Repeat for filter with $\Omega_1 = \pi/2$ and Cf: Fig 5.8(b)!

Derivation of Non-Recursive Filters using the Fourier Transform Method

To obtain a non-recursive (FIR) filter with a centre frequency Ω_0 and a bandwidth of $2\Omega_1$, we proceed as before (modulate h[n] for a low pass filter LPF) by a sinusoid at Ω_0 !

$$h'[n] = \underline{h[n]}.Cos(n\Omega_0) = \underline{\frac{1}{n\pi}Sin(n\Omega_1)}Cos(n\Omega_0)$$

We know already that: $H(\Omega) = b_0 + 2\sum_{k=1}^{k=M} b_k Cos(k\Omega)$

Where the b_k's are the values of the **new impulse response h'[n]**

$$\Rightarrow H(\Omega) = |H(\Omega)| = \frac{\Omega_1}{\pi} + 2\sum_{k=1}^{k=M} h'[k]Cos(k\Omega)$$

Derivation of Non-Recursive Filters using the Fourier Transform Method

Program No 12 - Lynn and Fuerst

Cf: Figure 5.9 and read pages 141 - 144 (programs 13 & 14 used)

Remember that FIR filters do not suffer from phase distortion !

Measure of the quality of a FIR filter -

$$e = \int_{2\pi} \left| H_D(\Omega) - H_A(\Omega) \right|^2 d\Omega$$

$H_D(\Omega)$ - desired frequency response
$H_A(\Omega)$ - actual frequency response

Windowing and the Effects of Truncation on h[n]

We already know that to obtain an ideal 'rectangular' filter response $H_D(\Omega)$, we require an infinite number of terms (b_k) and hence an Infinite Impulse Response (IIR) - $h_D[n]$ - {cf: Figure 5.12(a)}

Truncating an IIR to give a FIR is equivalent to multiplying the IIR by a 'Window' function of finite width !

Such a window function (rectangular window) is shown in Figure 5.12(b)

So taking the 'ideal' or 'desired' impulse response $h_D[n]$ and multiplying it by the window w[n], one obtains the 'actual' finite impulse response -

$$h_A[n] = h_D[n].w[n]$$

Windowing and the Effects of Truncation on h[n]

1. We alter the number of terms in $h_A[n]$ by the window length w[n]

2. We have symmetrized each h[n] to obtain a Real $H(\Omega)$ and a Zero Phase response. In real DSP systems we simply introduce the usual time shift and end up with linear phase charachteristics

3. Windowing is obtained by either:

(a) Time domain multiplication - $h_A[n] = h_D[n].w[n]$

(b) Frequency domain convolution - $H_A(\Omega) = H_D(\Omega)*W(\Omega)$

w[n] is sometimes referred to (especially in optical image processing) as 'APODIZATION'.

Windowing and the Effects of Truncation on h[n]

Spectrum of a Rectangular Window

$$W(\Omega) = \sum_{n=-\infty}^{n=+\infty} w[n].\exp(-jn\Omega)$$

Consider a window with 2M+1 terms distributed about n = 0

$$= \sum_{n=-M}^{n=+M} w[n].\exp(-jn\Omega)$$

Rectangular and so all w[n] values are equal (say = 1)

$$= \sum_{n=-M}^{n=+M} 1.\exp(-jn\Omega) \quad = 1 + 2\sum_{n=1}^{n=+M} Cos(n\Omega)$$

Windowing and the Effects of Truncation on h[n]

Program no. 15 evaluates $W(\Omega)$ for 320 Ω values, $0<\Omega<\pi$

The data are plotted on a Log scale of 0 to -50 dB !

One can see from figure 5.13 for 21 [M=10] and 51 term [M=25] windows that there are many sidelobes that exceed the -30 dB level - undesirable !

That said, rectangular windows provide the smallest rms error (e)
Intuitively one can see that it is the sharp cut-in/cut-off of rectangular windows which gives rise to a spread in frequency and a substantial no. of sidelobes of appreciable level (or gain for windows/processors) -

So we might expect that a window with a more gradual switch-on/switch-off might result in a narrower frequency spread - e.g., Triangular/Bartlett

Windowing and the Effects of Truncation on h[n]

Sidelobes -

The study/design of windows or apodization functions w[n] reduces to a study of sidelobes in $H_A(\Omega)$ and $W(\Omega)$. Since sidelobes are small, e.g., < 10% of main lobe level for a rectangular window frequency transfer function, $H_A(\Omega)$ and $W(\Omega)$ are plotted on a log scale.

If G = Filter Gain, Then gain (dB) = $20 \text{ Log}_{10}(G)$

G	$20\text{Log}_{10}(G)$
100	40
10	20
1	0
0.1	-20
0.01	-40
0.001	-60

Since we normalise G to unity, $|H(\Omega)|_{max}$ =1 or 0 dB !

It is clear from figure 5.12 (c) that rectangular windows give rise to unwanted sidelobes in $H_A(\Omega)$

Windowing and the Effects of Truncation on h[n]

Gibbs Phenomenon

With increasing window length the ripples in $H_A(\Omega)$ bunch more closely (around the design frequency Ω_1) - cf: fig 5.9 again. The cut-in/cut-off of the filter also becomes sharper.

Notice that lengthening the window does not reduce ripple magnitudes. In the vicinity of a sudden transition in $H_A(\Omega)$, the maximum ripple is ~ 9%, no matter the length of w[n] -> Gibbs Phenomenon (~1900)

Windowing and the Effects of Truncation on h[n]

Triangular Window Function:

The Bartlett window function and magnitude spectrum are plotted (on a dB scale) on figure 5.14.

We again choose 2M + 1 term values. They run from:

$$\frac{1}{M+1} @ n = 0 \quad \text{to} \quad \frac{1}{(M+1)^2} @ n = \pm M$$

$$W(\Omega) = \sum_{n=-M}^{n=+M} w[n] \exp(-jn\Omega)$$

$$= (M+1) + 2\left\{(M)Cos(\Omega) + (M-1)Cos(2\Omega) \ldots\ldots\ldots\ldots + Cos(M\Omega)\right\}$$

ignoring the $(M+1)^2$ normalisation factor, i.e., assuming amplitudes run from (M+1) at n =0 to 1 at n= $\pm$M

Windowing and the Effects of Truncation on h[n]

$$W(\Omega)\big|_{\Omega=0} = W_{\max} = (M+1) + 2\left\{(M) + \left(M-1\right)\ldots\ldots\ldots\ldots + 1\right\} = \left(M+1\right)^2$$

So, to normalise the Triangular Window gain to unity we divide w[n] by $1/(M+1)^2$

It is then known as a Bartlett Window function

$$w[n] = \frac{\left(M+1\right) - n}{\left(M+1\right)^2}, -M \le n \le +M$$

The corresponding frequency transfer (gain) function is given by:

$$W(\Omega) = \frac{1}{(M+1)^2} + \frac{2}{(M+1)^2}\left\{(M)Cos(\Omega) + \left(M-1\right)Cos(2\Omega)\ldots\ldots\ldots\ldots + Cos(M\Omega)\right\}$$

Windowing and the Effects of Truncation on h[n]

Note:

If you convolve two square pulses you will get a triangular window - so convolving two pulses of width M+1 terms will result in a Bartlett window of width 2M + 1 term values ! Hence the spectrum of a triangular window is given by:

$$H_T(\Omega) = |H_R(\Omega)^2|$$

since time domain convolution ≡ frequency domain multiplication

So sidelobes will drop as the square of the frequency away from the design frequency (or by a factor of 2 on a decibel scale) - Since the first sidelobe in a square or rectangular window has a gain of -13.5dB, the corresonding gain for the same lobe in a Barlett window is -27dB.

Windowing and the Effects of Truncation on h[n]

Note also:

The main lobe (passband) is twice that of a square window for the same number of terms M - remember if you convolve two windows with M+1 terms you will get a window with 2M+1 terms, i.e., a Triangular window with the same main lobe or bandwidth as a Rectangular window requires twice the number of b_k (or h[n]) coefficients to implement with a concomitant cost on CPU time.

Program 15 can be modified - substitute a Bartett w[n] for a rectangular w[n] and see the result in Figure 5.14.

Windowing and the Effects of Truncation on h[n]

Von Hann and Hamming Windows:

An ideal window has a narrow main lobe (passband) and no sidelobes !

Remember - $$H_A(\Omega) = H_D(\Omega) * W(\Omega)$$

Actual = Filter Response	Desired Filter Response	Window Transfer/gain Function

For sharp transitions in $H_A(\Omega)$
=> $W(\Omega)$ must possess a narrow main lobe and no sidelobes
But ripples in $H_A(\Omega)$ depend on the magnitude of the sidelobes in $W(\Omega)$
or
smaller the sidelobes in $W(\Omega)$, the better the ripple performance of $H_A(\Omega)$

$H_R(\Omega)$, rectangular window has the narrowest main lobe for given 'M' but the worst ripple performance - so a tradeoff between these two characteristics (M and bandwidth) is unavoidable !

Windowing and the Effects of Truncation on h[n]

von Hann and Hamming windows, though possessing a main lobe width comparable to the Bartlett Window (for given value of M) they have better sidelobe performance

von Hann - 2M+1 terms

$$w[n] = 0.5 + 0.5Cos\left(\frac{n\pi}{M+1}\right), -M \leq n \leq +M$$

DC level + One Full Cosinusoidal Cycle - See Fig 5.15

Hamming

R W Hamming treated the DC offset and the cosine amplitude as variable quantities to optimise sidelobe performance -

$$w[n] = \alpha + \beta Cos\left(\frac{n\pi}{M}\right), -M \leq n \leq +M, \alpha = 0.54 / \beta = 0.46$$

Windowing and the Effects of Truncation on h[n]

von Hann and Hamming windows

The Frequency Transfer (Gain) Function is as usual given by:

$$W(\Omega) = \sum_{n=-M}^{n=+M} w[n]\exp(-jn\Omega) = w[0] + 2\sum_{k=+1}^{k=+M} w[k]Cos(k\Omega)$$

This operation is effected via Program no 16 - see Figure 5.16.
Codes computes:

$$w[n] = \alpha + \beta Cos\left(\frac{n\pi}{\gamma}\right), -M \le n \le +M$$

Input values: $\alpha, \beta \;\&\; \gamma$

Code computes w[n] (i.e., w[k] above) and then W(Ω) for 320 values and plots them on a dB scale.

Windowing and the Effects of Truncation on h[n]

Cf: Fig 5.16, 51 Term [M = 25] Windows

	1st Sidelobe Gain
• Triangular Window	-27dB
• von Hann Window	-32dB
• Hamming Window	<-40dB

Program no. 17 - Filter Design/ Cf: Fig 5.17

Inputs: Ω_0, Ω_1 (Full band pass = $2\Omega_1$), 2M + 1 and Window Type - it outputs h[n] values and a plot of $|H_A(\Omega)|$ .vs. Ω plot on a dB vertical scale

(a) Rectangular Window
(b) von Hann Window
(c) Hamming Window

$\Omega_0 = 2\pi/3$ or 3 Samples/Cycle
$2\Omega_1$ (bandwidth) = $\pi/18$
M = 25 (51 Term Filter)

Windowing and the Effects of Truncation on h[n]

1. The von Hann/Hamming windows show good sidelobe attenuation
2. Bandpass in both von Hann/Hamming windows is > 10^0, i.e., if one specifies $2\Omega_1 < 10^0$, one will not achieve it for even moderate M

Cf: Fig 5.18, 101 Term [M = 50] Hamming Window

$\Omega_0 = 0$ or LPF

$2\Omega_1$ (bandwidth) $= 2\pi/5$

M = 25 (51 Term Filter)

Note: All sidelobes < -50dB !!!!

Windowing and the Effects of Truncation on h[n]

J. F Kaiser Window

$$w[n] = \frac{I_0\left(\alpha\sqrt{1-\left(\frac{n}{M}\right)^2}\right)}{I_0(\alpha)}, -M \le n \le M$$

I_0: Modified Bessel function of the first kind and zero order.

By varying α one can obtain windows of varying taper !!!

1. for $\alpha = 0$, w[n] =1 for all 'n' - Rectangular Window
2. for $\alpha = 5.44$ one gets a Hamming Window !

Windowing and the Effects of Truncation on h[n]

Designing a Kaiser Window:

Plot the ideal filter response (e.g., as in figure 5.19) $H_D(\Omega)$ showing acceptable ripple level ($\pm\delta$) and transition width (Δ). Then -

Since α controls the window taper (and hence the sidelobe gains or levels), it is determined by the chosen values of δ.

For given δ (and hence α), Δ is determined by the window length 'M'.

If you know δ and M we can can compute w[n] and hence
$W(\Omega) = H_A(\Omega)\ \{\sim H_D(\Omega)\}$

Windowing and the Effects of Truncation on h[n]

Designing a Kaiser Window: Implementation

See "Digital Filters, 2nd Edition, R W Hamming (1983)

1. 'δ' is expressed as an attenuation - $A = -20\text{Log}_{10}(\delta)$

2. 'α' is found from the empirical formulae:

'α' = $0.1102(A - 8.7)$, $A \geq 50$
'α' = $0.5842(A - 21)^{0.4} + 0.07886(A - 21)$, $21 < A < 50$
'α' = 0.0, $A \leq 20$ - (Rectangular window)

Armed with 'A' and choosing 'Δ' (expressed as a fraction of 2π), one can compute the final quantity needed as:

$$M \geq \frac{A - 7.95}{28.72\Delta}$$

rounded up to the nearest integer number !

Windowing and the Effects of Truncation on h[n]

Designing a Kaiser Window: Implementation

If M is too large, reduce 'A' by increasing 'δ' or increase 'Δ' !

One can compute the Bessel function from a power series as:

$$I_0(x) = 1 + \sum_{m=1}^{m=\infty} \left[\left(\frac{x}{2}\right)^m \frac{1}{m!} \right]^2$$

Typically you need include only 10 or so terms (m=10) for a good representation of the function

Windowing and the Effects of Truncation on h[n]

Designing a Kaiser Window: Program No 18

Computes $I_0(x)$ for the first 20 terms: $x = \alpha\sqrt{1-\left(\frac{n}{M}\right)^2}$ - numerator

$x = \alpha$ - denominator

To compute a Kaiser window - input δ, Δ, Ω_p and Ω_s

Example - Figure 5.20

High Pass Filter (HPF) with a 60^{O} bandwidth (BW)

(a) δ = -30dB (0.0316), $\Delta = 15^{O}$, α = 2.1176 and M = 19

(b) δ = -40dB (0.0100), $\Delta = 7.5^{O}$, α = 3.3954 and M = 54

Equi-ripple Filters

FIR filter design boils down to a compromise between the sharpness of the passband edges and magnitude of the sidelobes (for a given number of b_k or h[n] values

Equiripple filters (as the names implies) have sidelobes of approximately similar gain, rather than a maximum near the main lobe and decreasing as one moves away (toward higher frequency) from the main lobe

The main features of equiripple filters are sketched in Fig. 5.21 -

The PASSBAND is: $0 \leq \Omega \leq \Omega_p$
The ACCEPTABLE RIPPLE is: $\pm\delta_1$

The STOPBAND is: $\Omega_s \leq \Omega \leq \pi$
The ACCEPTABLE RIPPLE is: $\pm\delta_2$

Peaks and troughs occur at Ω_1, Ω_2, Ω_3,....

Equi-ripple Filters

$$H(\Omega) = \sum_{k=-M}^{k=+M} b_k \exp(-jk\Omega)$$

$$= b_0 + 2\sum_{k=1}^{k=M} b_k Cos(k\Omega)$$

$$= h[0] + 2\sum_{k=1}^{k=M} h[k] Cos(k\Omega)$$

We can write this as: $H(\Omega) = \sum_{k=0}^{k=M} c_k Cos(\Omega)^k$

So H(Ω) can be expressed as an M^{th} Order trignometric polynomial which can display up to M-1 extrema within $0 \le \Omega \le \pi$!

Equi-ripple Filters

Note also that:

$$H'(\Omega) = \frac{dH(\Omega)}{d\Omega} = -Sin(\Omega)\sum_{k=1}^{k=M} k.c_k Cos(\Omega)^{k-1}$$

@ $\Omega=0$ and $\Omega = \pi$, $Sin(\Omega) = 0$ and $H'(\Omega) = 0$ at these frequencies. Hence $H(\Omega)$ must be a max/min at these frequencies and there are a possible (M-1)+2 = M+1 extrema within the band $0 \leq \Omega \leq \pi$

There are 5 possible parameters that can be varied to design an optical equiripple filter M, δ_1, δ_2, Ω_p and Ω_s

There are two popular approaches to Equiripple FIR filter design - Hermann + Schuessler: Specify M, δ_1 and δ_2, Allow Ω_p and Ω_s to vary. They developed a set of non-linear equations to be minimized.

Parks + McClellan: Specify M, δ_1/δ_2, Ω_p and Ω_s and let δ_1 vary.

The main advantage of the H-S method is that Δ ($=\Omega_p - \Omega_s$) is fixed !

Digital Differentiators

x[n] = displacement
x'[n] = velocity
Useful in anticipatory applications

First Order Difference (FOD): $FOD = x\left[n\right] - x\left[n-1\right]$

$$y[n] = x[n] - x[n-1]$$

$$h[n] = x[n] - x[n-1]$$

$$H\left(\Omega\right) = 1 - \exp\left(j\Omega\right) = 1 - Cos\left(\Omega\right) \underset{1/2}{-} jSin\left(\Omega\right)$$

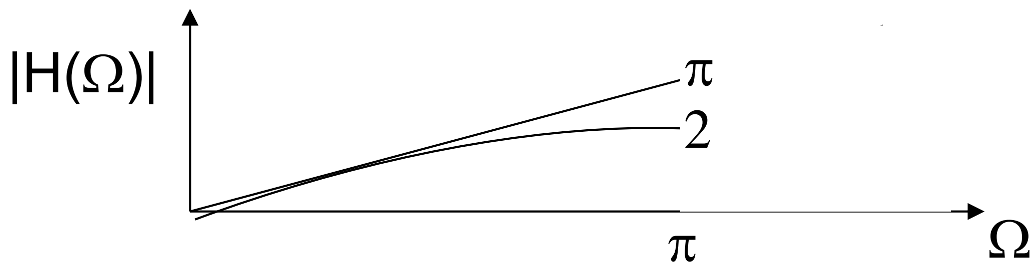
$$\left|H\left(\Omega\right)\right| = \left[\left\{1 - Cos\left(\Omega\right)\right\}^2 + \left\{Sin\left(\Omega\right)\right\}^2\right] = 2Sin\left(\frac{\Omega}{2}\right)$$

Digital Differentiators

In fact an ideal differentiator would have $|H(\Omega)|$ proportional to Ω, e.g., $d/dn\{Sin(n\Omega)\} = \Omega Cos(n\Omega)$ - for Ω close to zero $Sin\Omega() \sim \Omega$

$$\Rightarrow |H(\Omega)| = 2\,Sin\left(\frac{\Omega}{2}\right) \approx \Omega$$

As Ω gets close to π, $|H(\Omega)|$ approaches 2! - it should be π



Note also that d/dn introduces a 90^O phase shift no matter what the value of Ω is - so that phase shift/ transfer function is constant, 90^O for all Ω i.e.,

Digital Differentiators

$$\Phi_H(\Omega) = Tan^{-1}\left[\frac{\operatorname{Im} H(\Omega)}{\operatorname{Re} H(\Omega)}\right] = \frac{\pi}{2}$$

$$\Rightarrow \left[\frac{\operatorname{Im} H(\Omega)}{\operatorname{Re} H(\Omega)}\right] = Tan\left(\frac{\pi}{2}\right) \to \infty$$

$$\Rightarrow \operatorname{Re}\left[H(\Omega)\right] = 0$$

So we have that H(Ω) is purely imaginary !!

$$H(\Omega) = j\Omega$$

Digital Differentiators

Hence the ideal differentiator is a purely imaginary operator !!

Up to this point we have made h[n] symmetric about n=0 and hence H(Ω) becomes real and specificed by cosines only -

A purely imaginary filter operator will be specificed by sine functions only and it will have an impulse response h[n] which is antisymmetric about n=0

Let's evaluate the h[n] corresponding to H(Ω) = jΩ

$$h[n] = \frac{1}{2\pi}\int_{2\pi} H(\Omega)\exp(jn\Omega)\,d\Omega$$

$$= \frac{1}{2\pi}\int_{2\pi} j\Omega\exp(jn\Omega)\,d\Omega$$

Digital Differentiators

Integrate by parts: $\int u\,dv = uv - \int v\,du$

$$u = j\Omega \qquad dv = \exp(jn\Omega)\,d\Omega$$

$$\Rightarrow h[n] = \frac{1}{2\pi}\left[j\Omega \frac{\exp(jn\Omega)}{jn} \Bigg|_{-\pi}^{+\pi} \right] - \frac{1}{2\pi}\left[\int_{-\pi}^{+\pi} \frac{\exp(jn\Omega)}{jn}\, j\,d\Omega \right]$$

$$\Rightarrow h[n] = \frac{1}{2\pi}\left[\exp(jn\Omega)\left\{ \frac{\Omega}{n} - \frac{1}{jn^2} \right\} \Bigg|_{-\pi}^{+\pi} \right]$$

$$\Rightarrow h[n] = \frac{1}{2\pi}\left[\exp(jn\pi)\left\{ \frac{\pi}{n} + \frac{j}{n^2} \right\} - \exp(-jn\pi)\left\{ \frac{-\pi}{n} + \frac{j}{n^2} \right\} \right]$$

Digital Differentiators

For n odd, exp(jnπ) = exp(-jnπ) = -1
For n even, exp(jnπ) = exp(-jnπ) = +1

Hence we can write for an ideal digital differentiator:

$$h[n] = \frac{-1}{n}, n = \pm 1, 3, 5, \ldots\ldots..$$

$$h[n] = \frac{1}{n}, n = \pm 2, 4, 6, \ldots\ldots..$$

$$h[0] = \frac{1}{2\pi} \int_{-\pi}^{+\pi} j\Omega \exp(jn\Omega)\, d\Omega \Big|_{\Omega=0} = 0$$

Digital Differentiators

Cf: Figure 5.24 for h[n] -

Usual problem - we have trade off the number of coefficients (b_k) or h[n] values against an acceptable form of $|H(\Omega)|$ so that it is close to $|H(\Omega)| \sim \Omega$ frequency gain (transfer) function with weak sidelobes !

Solution - Use a Window (Apodize !!)

Program No 19

Use Rectangular or Hamming window to truncate h[n] to 2M + 1 terms.
Cf: Figure 5.25 with M = 10 - increasing M improves transition neat π !

NEXT - IIR FILTERS