Design of Non-recursive Digital Filters

Digital Signal Processing

Notes

Contents

Notes

What is a Digital Filter?

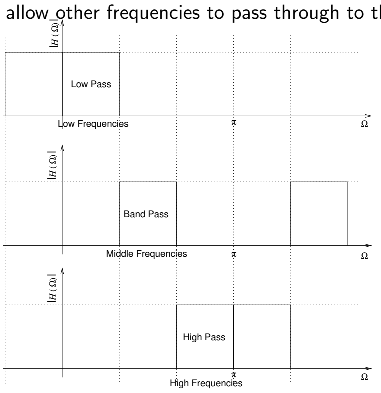
Any digital system can be described as a digital filter. The word "*filter*" means to remove a part of a signal and allow another part to pass through.
The verb "*to filter*" is used in many areas of English language.
Examples

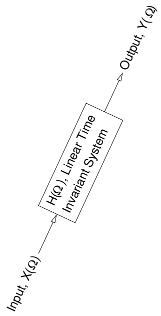
- The water filter cleans the water for drinking.
- The cook filtered the bad fruit from the good for cooking later.
- The air conditioning filters dust from the air.

Notes

What is a Digital Filter?

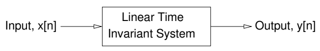
Often used to remove some frequencies from a signal $X(\Omega)$ and to allow other frequencies to pass through to the output $Y(\Omega)$.





Notes

Non-recursive digital filters



What is a *non-recursive digital filter*?

- "*Recursive*" comes from the word "to recur"
  Meaning: to repeat

A recursive filter uses past output values ($y[n-i]$) for the current output $y[n]$:

- *Recursive Filter Example*

$$y[n] = 0.5y[n-1] + 0.5x[n].$$

A non-recursive filter only uses input values $x[n-i]$:

- *Non-recursive Filter Example*

$$y[n] = 0.5x[n-1] + 0.5x[n].$$

Notes

Generalised Non-Recursive Difference Equation

Recall the generalised difference equation for causal LTI systems (see Lecture 02):

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

So a non-recursive digital filter in a causal LTI system is given by:

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

Notes

Non-Recursive Digital Filters

Non-recursive digital filters are often known as

- **Finite Impulse Response (FIR)** Filters

as a non-recursive digital filter has a finite number of coefficients in the impulse response $h[n]$.
Recursive digital filters are often known as

- **Infinite Impulse Response (IIR)** Filters

as the impulse response of an IIR filter has an infinite number of coefficients.
FIR Filters

- Have linear phase characteristics (*i.e.* no phase distortion);
- But they typically require a higher number of computations.

Notes

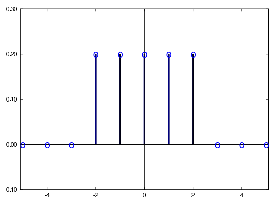
Moving Average Filters

- Moving average filters are usually implemented non-recursively.
- Moving average filters are interesting as they
  - Are useful for some applications
  - But the frequency response is not ideal

$$h[n] = \begin{cases} \frac{1}{k} & \text{if} \quad -k/2 \le n \le k/2 \\ 0 & \text{otherwise} \end{cases} \; ; \text{ where } k \text{ is odd.}$$

Example

$$h[n] = \begin{cases} 0.2 & \text{if} \quad -2 \le n \le 2 \\ 0 & \text{otherwise} \end{cases} .$$

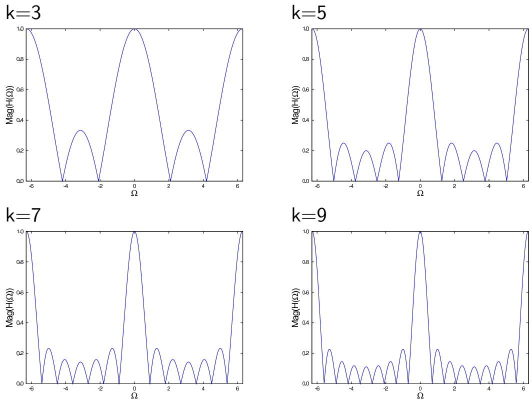


Notes

Moving Average Freq. Response

k=3 k=5

k=7 k=9

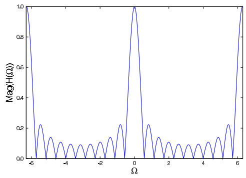


Notes

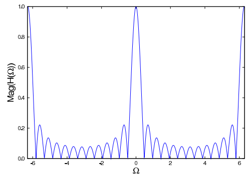
Moving Average Freq. Response

Notes

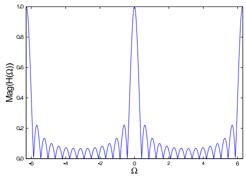
k=11



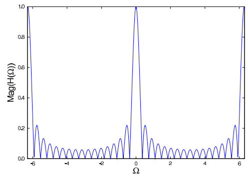
k=13



k=15



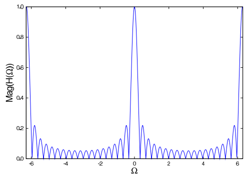
k=17



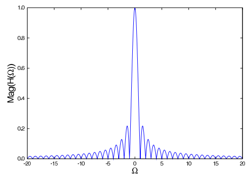
Moving Average Frequency Response and the Sinc Function

Notes

k=19



sinc function



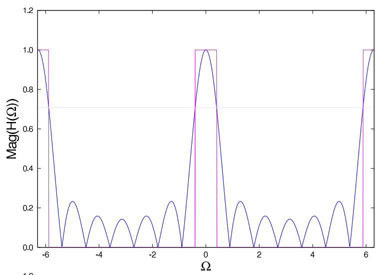
The frequency response tends to a Sinc function defined as:

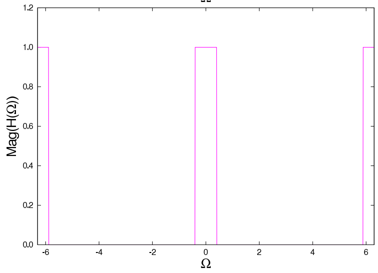
$$H[\Omega] = \mathrm{sinc}(\Omega) = \frac{1}{\Omega}\sin(\Omega).$$

Moving Average Vs Ideal

Notes

Moving average ($k = 7$) (blue) and ideal low pass frequency responses (magenta). Ideal low pass frequency response only.





Ideal Frequency Response Time Domain Representation

What is the time domain representation for the ideal low pass frequency response?

Recall the inverse Fourier Transform is given by:

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

So for an impulse response $h[n]$:

$$h[n] = \frac{1}{2\pi}\int_{2\pi} H(\Omega)\exp(j\Omega n)\mathrm{d}\Omega; \tag{2}$$

or sometimes simpler to use

$$h[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\Omega)\exp(j\Omega n)\mathrm{d}\Omega; \tag{3}$$

which is possible due to periodicity.

Notes

Ideal Frequency Response Time Domain Representation

The ideal low pass frequency response is given by:

$$H(\Omega) = \begin{cases} 1 & \text{for} \quad -\Omega_1 \le \Omega \le \Omega_1, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

This equation states that any frequency Ω between $-\Omega_1$ and $+\Omega_1$ can pass without attenuation ($H(\Omega) = 1$). All other frequencies are stopped altogether.

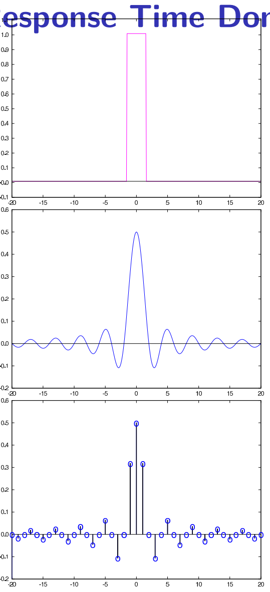
The time domain representation of (4) is given by

$$\begin{aligned}
h[n] =& \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\Omega)\exp(j\Omega n)\mathrm{d}\Omega = \frac{1}{2\pi}\left(\int_{-\pi}^{-\Omega_1} 0\mathrm{d}\Omega + \int_{-\Omega_1}^{\Omega_1} 1\times\exp(j\Omega n)\mathrm{d}\Omega + \int_{\Omega_1}^{\pi} 0\mathrm{d}\Omega\right)\\
=& \frac{1}{2\pi}\left([0]_{-\pi}^{-\Omega_1} + \left[\frac{\exp(j\Omega n)}{jn}\right]_{-\Omega_1}^{\Omega_1} + [0]_{\Omega_1}^{\pi}\right) = \frac{1}{2\pi}\left[\frac{\exp(j\Omega n)}{jn}\right]_{-\Omega_1}^{\Omega_1}\\
=& \frac{1}{2\pi jn}\left(\exp(j\Omega_1 n) - \exp(-j\Omega_1 n)\right) = \frac{1}{2\pi jn}2j\sin(\Omega_1 n) = \frac{1}{\pi n}\sin(\Omega_1 n)\\
=& \frac{\Omega_1}{\pi}\mathrm{sinc}(n\Omega_1)
\end{aligned}$$

where $\mathrm{sinc}(n\Omega_1) = \frac{\sin(n\Omega_1)}{n\Omega_1}$ is known as the sinc function.

Notes

Ideal Low Pass Frequency Response Time Domain Representation



Example

Q. What is the time domain impulse response for an ideal low pass frequency response filter with $\Omega_1 = \pi/2$?

A.

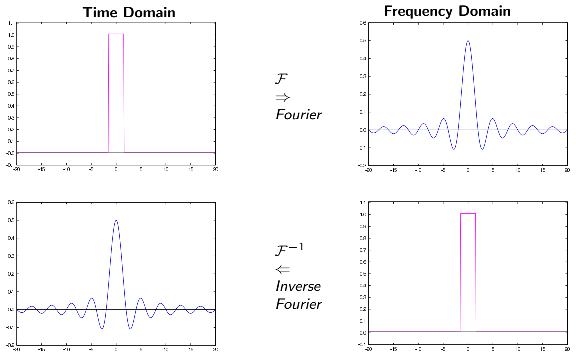
$$\begin{aligned}
h[n] =& \frac{\pi}{2\pi}\mathrm{sinc}\left(n\frac{\pi}{2}\right) = \frac{1}{2}\mathrm{sinc}\left(\frac{n\pi}{2}\right)\\
=& \frac{2}{n\pi}\frac{1}{2}\sin\left(\frac{n\pi}{2}\right)\\
=& \frac{1}{n\pi}\sin\left(\frac{n\pi}{2}\right).
\end{aligned}$$

$h[n]$ is sampled at discrete values of n.

Notes

Sinc Function

- The Sinc function is the Fourier transform of a square wave or impulse response.



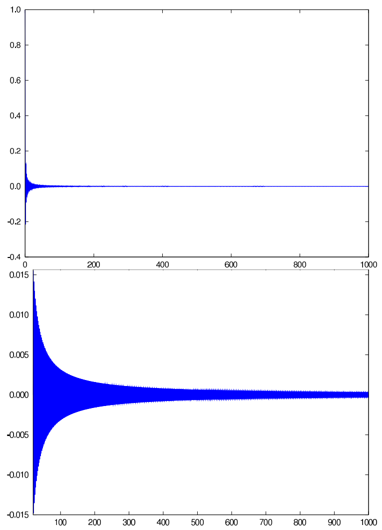
- The Sinc function is also the inverse Fourier transform of a square wave.

Notes

Ideal Frequency Response Time Domain Representation

A Problem!

The Sinc function continues forever.

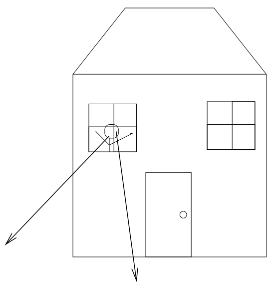


Therefore the time domain representation has to be stopped early.

Notes

How to Stop the Sinc function early?

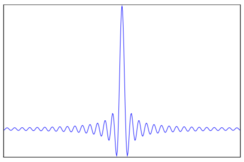
A window can be used...



... to limit the impulse response.

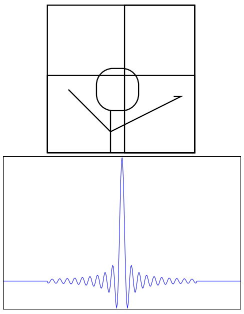
Notes

How to Stop the Sinc function early?



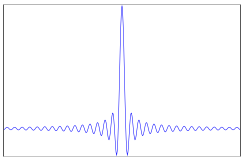
We can use a relatively big window...

$\Rightarrow$



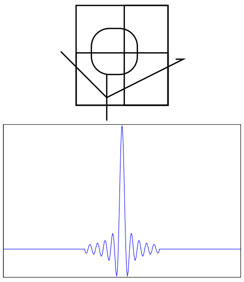
Notes

How to Stop the Sinc function early?



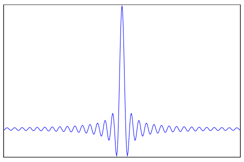
We can use a medium sized window...

$\Rightarrow$



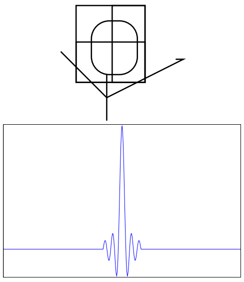
Notes

How to Stop the Sinc function early?



Or we can use a relatively small window...

$\Rightarrow$



Notes

How to Stop the Sinc function early?

The window here $w[n]$ is rectangular and constant, *i.e.*

$$w[n] = \begin{cases} 1 & \text{for } -\frac{\text{window width}}{2} \leq n \leq \frac{\text{window width}}{2} \\ 0 & \text{every where else} \end{cases}$$

usually known as a rectangular window.
The sinc function filter coefficients are multiplied by this window (in the time domain):

$$h_2[n] = h[n] \times w[n].$$

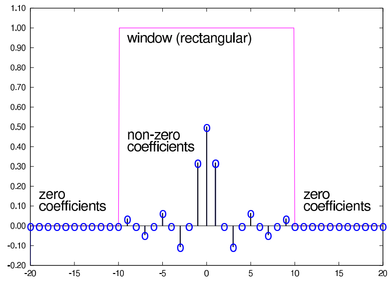
So that

$$h_2[n] = \begin{cases} h[n] & \text{for } -\frac{\text{window width}}{2} \leq n \leq \frac{\text{window width}}{2}, \\ 0 & \text{every where else.} \end{cases}$$

Notes

How to Stop the Sinc function early?

$$h_2[n] = \begin{cases} h[n] & \text{for } -\frac{\text{window width}}{2} \leq n \leq \frac{\text{window width}}{2}, \\ 0 & \text{every where else.} \end{cases}$$

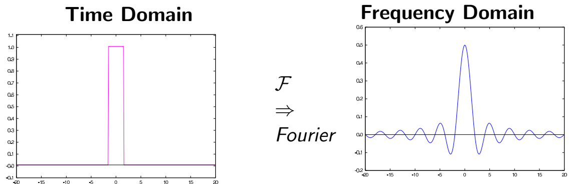


Notes

Rectangular Window Problems!

HOWEVER! The frequency response of this window is not ideal.

Rectangular pulse in time domain = sinc function in frequency domain:



- The original ideal low pass filter in the frequency domain is corrupted by the windowing[1].

[1] Windowing describes multiplication by the window function.

Notes

Rectangular Window Problems!

Time Domain		Frequency Domain
Convolution	$\Longleftrightarrow$	*Multiplication*
$x[n] * y[n]$		$X(\Omega) \times Y(\Omega)$
Multiplication	$\Longleftrightarrow$	*Convolution*
$x[n] \times y[n]$		$X(\Omega) * Y(\Omega)$

Time domain multiplication of $w[n]$ with $h[n]$ is the same as convolution in the frequency domain, *i.e.*

$$\mathcal{F}(w[n] \times h[n]) = W(\Omega) * H(\Omega).$$

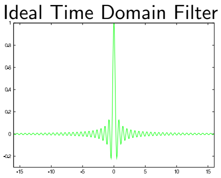
Time domain		Frequency Domain
$w[n]$	$\Longleftrightarrow$	$W(\Omega)$
$h[n]$	$\Longleftrightarrow$	$H(\Omega)$
$h_2[n]$	$\Longleftrightarrow$	$H_2(\Omega)$

$$H_2(\Omega) = \mathcal{F}(h_2(n)) = W(\Omega) * H(\Omega).$$

Notes

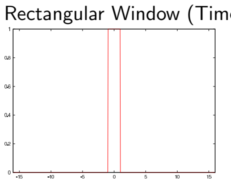
Frequency Domain Effect of (Rect.) Windowing Ideal Low Pass Filter

Ideal Time Domain Filter



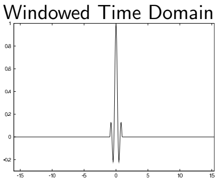
$\times$

Rectangular Window (Time)



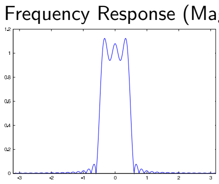
$=$

Windowed Time Domain



$\overset{\mathcal{F}}{\Rightarrow}$

Frequency Response (Mag)

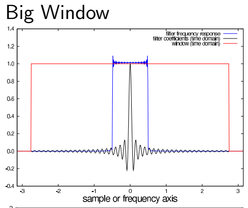


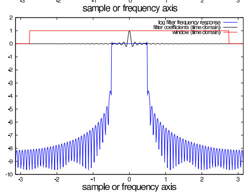
Notes

Frequency Domain Effect of (Rect.) Windowing Ideal Low Pass Filter

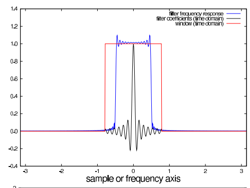
Log - to emphasise the *side lobes*

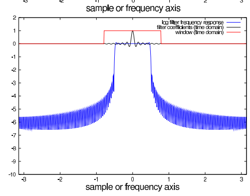
Big Window



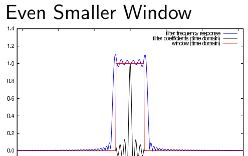


Smaller Window

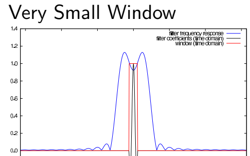




Even Smaller Window



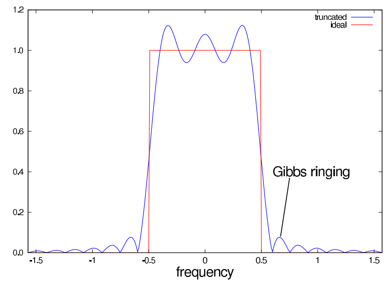
Very Small Window



Notes

Gibbs Ringing or Truncation Artifact

Gibbs artifact = truncation artifact.

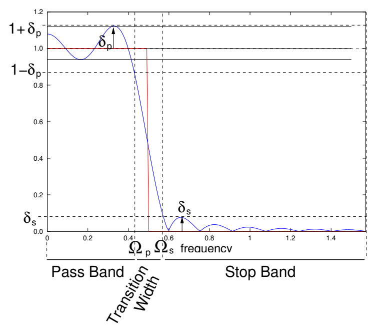


Bumps around sudden changes in signal.

Notes

Filter Parameters

A filter can be described by a number of *parameters*.



- δ_p: pass band ripple
- δ_s: stop band ripple
- Pass Band (little attenuation)
- Transition width
- Stop Band (highest attenuation)

- $\delta_p = 0.12$
- $\delta_s = 0.08$

$\therefore$ Gain at end of pass band is $1 - \delta_p = 0.88$.
Gain and attenuation in decibels (dB):

$$\text{gain}_{\text{dB}} = 20 \log(\text{ gain }).$$

So Gain at end of pass band

$$20 \log(1 - \delta_p) = -1.11\text{dB}.$$

Gain at end of stop band is $1 - \delta_s = 0.08$ or -21.94dB.
Pass band frequencies:

$$\Omega = 0 \text{ to } \Omega_p$$

where $\Omega_p = 0.43$ radians.
Transition width:

Notes

Filter Bandwidth

Filter bandwidth defined by:
"Range of frequencies the filter gain is greater than -3dB".

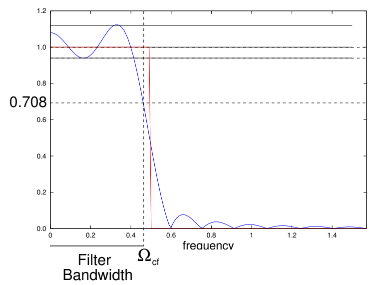
As

$$\text{gain}_{\text{dB}} = 20 \log(\text{ gain }).$$

Then

$$\text{gain} = 10^{\left(\frac{\text{gain}_{\text{dB}}}{20}\right)}.$$

So for $\text{gain}_{\text{dB}} = -3\text{dB}$ then $\text{gain} = 0.708$.



- The cut-off frequency Ω_{cf} corresponds to when the gain falls below -3dB.

Notes

Other Window Types for Truncation (Other than Rectangle)

Other types of window functions can be used to truncate the ideal time domain response:

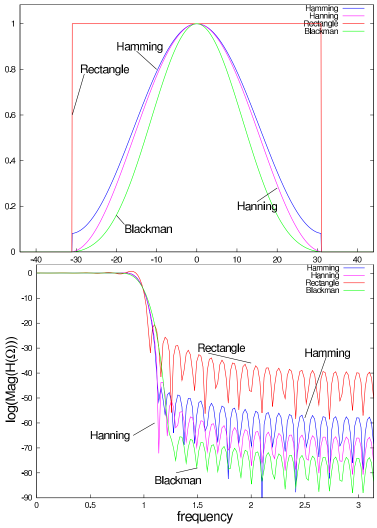
Window Name	Equation
Bartlett/ triangular	$w[n] = \frac{(N+1)-\lvert n\rvert}{(N+1)^2}$ for $\lvert n\rvert \le (N-1)/2$
Hamming	$w[n] = 0.54 + 0.46\cos\left(\frac{2\pi n}{N-1}\right)$ for $\lvert n\rvert \le (N-1)/2$
Hanning	$w[n] = 0.5 + 0.5\cos\left(\frac{2\pi n}{N-1}\right)$ for $\lvert n\rvert \le (N-1)/2$
Kaiser	$w[n] = \frac{1}{I_0(\alpha)} I_0\left(\alpha\sqrt{1-\left(\frac{2n}{N-1}-1\right)^2}\right)$ for $\lvert n\rvert \le (N-1)/2.$

Others include Blackman, Lanczos and Tukey windows.

Notes

Example Frequency Responses with Different Windows

Each window is slightly different in the time domain. Windowed truncation of a filter ideal frequency response.



- Each window type has different properties.
- *e.g.* Stop band attenuation for Blackman is highest but Blackman has long transition width

Notes

FIR Low Pass Filter Design Steps

1. Find the cut-off digital frequency, Ω_{cf}
   1. It may be given directly,
      - *e.g.* $\Omega_{cf} = \pi/4$ radians
   2. Or the sampling frequency and cut off frequencies may be given instead, calculated from $\Omega = 2\pi f/f_s$,
      - *e.g.* $f_s = 100$kHz and $f_{cf} = 12.5$kHz, so that $\Omega_{cf} = 2\pi 12500/100000 = \pi/4$ radians.
2. Calculate the appropriate sinc function for ideal low pass filter:

$$h[n] = \frac{\Omega_1}{\pi}\mathrm{sinc}(n\Omega_1)$$

   where $\Omega_1 = \Omega_{cf}$ is the cut-off frequency.
3. Select a window with appropriate parameters. *e.g.* short transition width or high stop band attenuation.
4. Calculate non-causal time domain impulse response from $h_2[n] = w[n] \times h[n]$.
5. Shift the impulse response to make a causal version $h_3[n] = h_2[n-(N-1)/2]$.

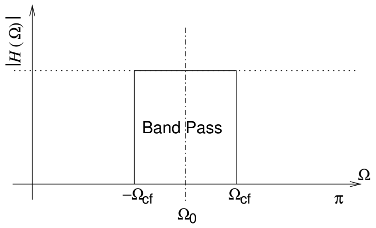
Notes

Band Pass FIR Filter Design

Low pass filter $H(\Omega)$ can be converted to a bandpass filter by:

- Convolution in frequency domain with delta function δ_{Ω_0};
- At centre frequency Ω_0.

Frequency Domain		Time Domain
(*Convolution*)		(*Multiplication*)
$H(\Omega) * \delta_{\Omega_0}$	$\Longleftrightarrow$	$h[n] \times \cos(n\Omega_0)$



The resulting band pass impulse response $h'[n]$

- with bandwidth $2 \times \Omega_{\text{cf}}$,
- using window function $w[n]$,
- and centre frequency Ω_0

is given by

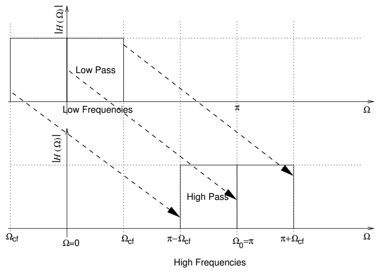
$$h'[n] = h[n] \times w[n] \times \cos(n\Omega_0)$$
$$= \frac{\Omega_{\text{cf}}}{\pi}\text{sinc}(n\Omega_{\text{cf}}) \times w[n] \times \cos(n\Omega_0).$$

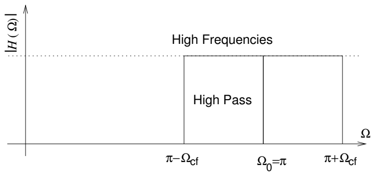
Notes

High Pass FIR Filter

High pass filter can be achieved by

- Shifting the impulse response to $\Omega_0 = \pi$,
- Via multiplication by $\cos(n\Omega_0) = \cos(n\pi)$.
- This is the limit of the unique part of the digital spectrum.





The resulting high pass impulse response $h'[n]$

- with bandwidth Ω_{cf},
- using window function $w[n]$,

is given by

Notes

Other Topics in Filter Design

Band stop is another type of filter,

- Created from a combination of high and low pass filters.

Digital differentiators are common in DSP applications,

- To differentiate a signal, to calculate *e.g.* speed.

Other techniques for FIR filter design include:

- Equiripple filters
  - Optimization of passband and stopband ripples.
- Frequency sampling method
  - Optimization from specified (sampled) frequency response.

Other factors to consider

- Phase response of the filter, not just the magnitude.
  - *e.g. Hilbert transformer* places a 90^o phase shift on a signal.

Notes

Summary

We have covered:

- Moving average filters (and their frequency response)
- Ideal frequency response of FIR filters
- Windowing techniques
- Filter parameters
- Filter design techniques for
  - low pass,
  - band pass
  - and high pass FIR filters

Notes

Notes

Notes