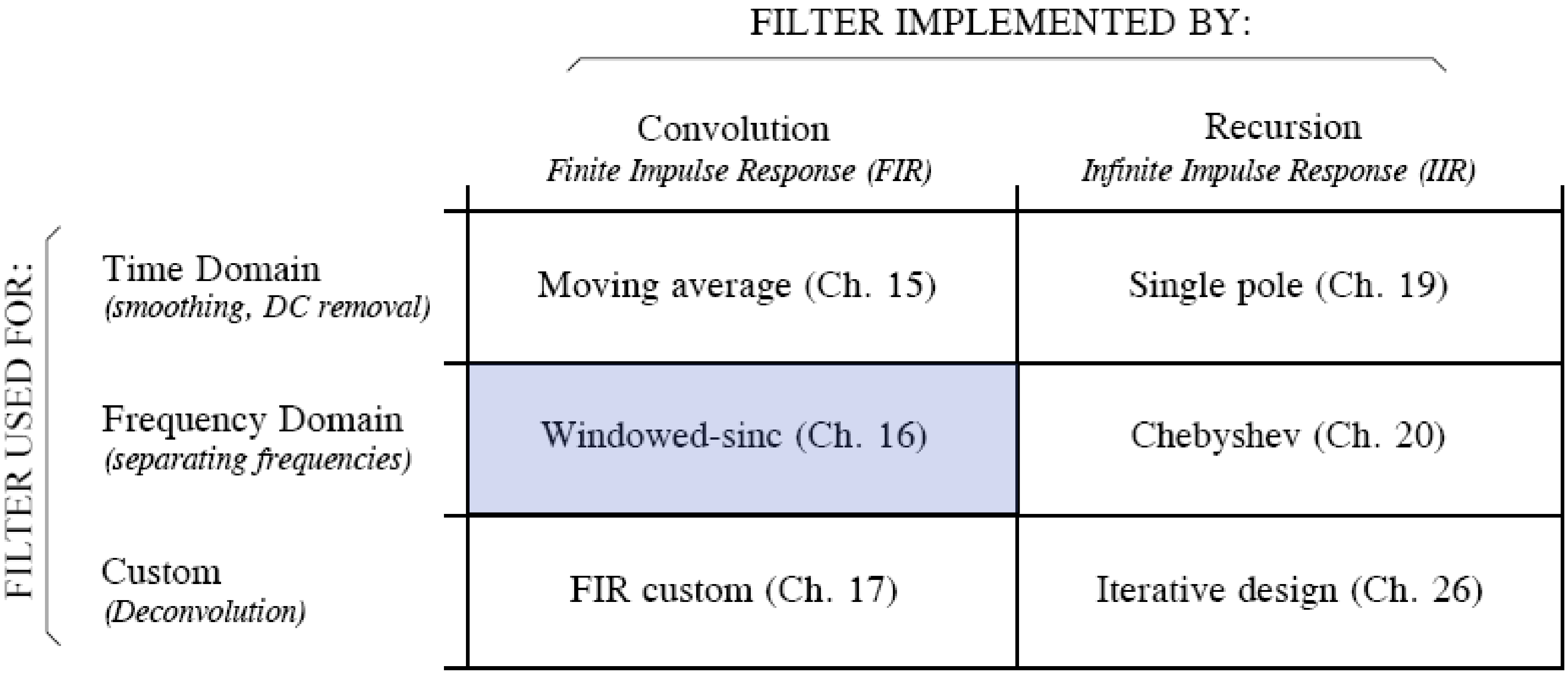
Digital Signal Processing

Windowed SINC Filters

Today’s Topics

- Origins of the Windowed SINC filter
- Impulse Response
- Frequency Response

Filter Classification



		FILTER IMPLEMENTED BY:	
		Convolution *Finite Impulse Response (FIR)*	Recursion *Infinite Impulse Response (IIR)*
FILTER USED FOR:	Time Domain *(smoothing, DC removal)*	Moving average (Ch. 15)	Single pole (Ch. 19)
	Frequency Domain *(separating frequencies)*	Windowed-sinc (Ch. 16)	Chebyshev (Ch. 20)
	Custom *(Deconvolution)*	FIR custom (Ch. 17)	Iterative design (Ch. 26)

Some Properties of the Windowed SINC FIR Filter

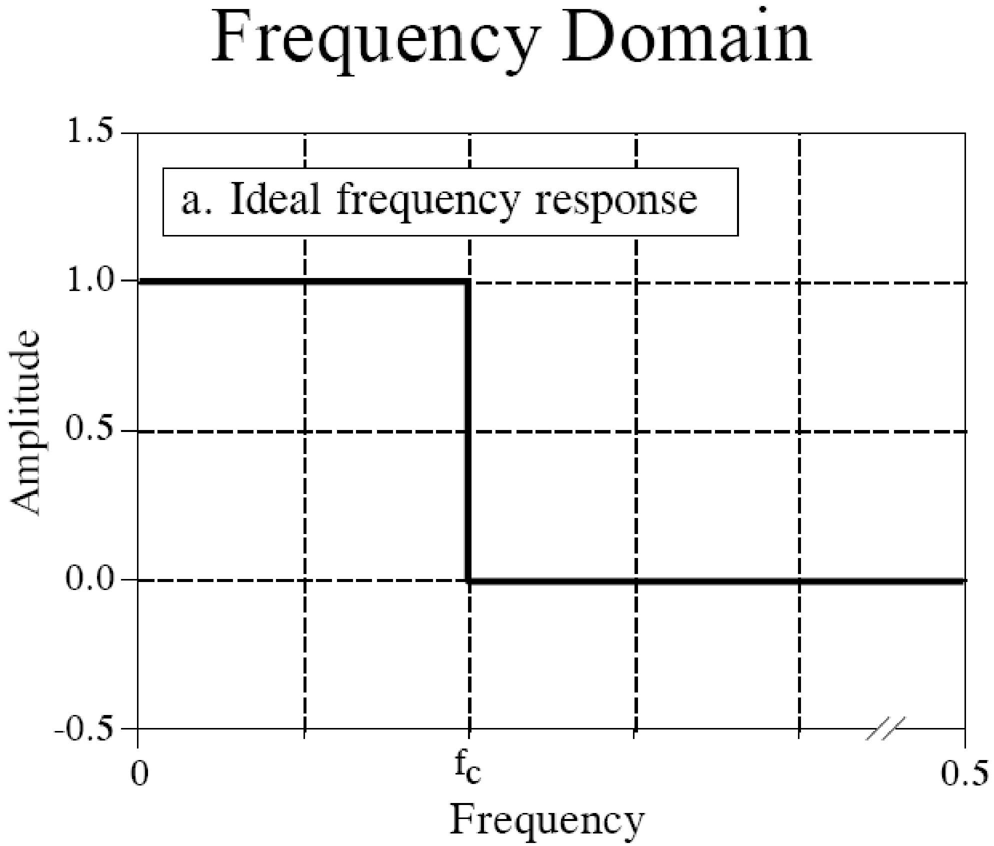
- Better (than the MAV) for separating one band of frequencies from another

- Finite Impulse Response (FIR) structure
  - Stable response
  - Computed using convolution
    - Slower than recursive filters
  - Can use FFT convolution to speed up (multiplication in frequency domain).

- Step response in time domain shows overshoot

The Ideal Low Pass Filter

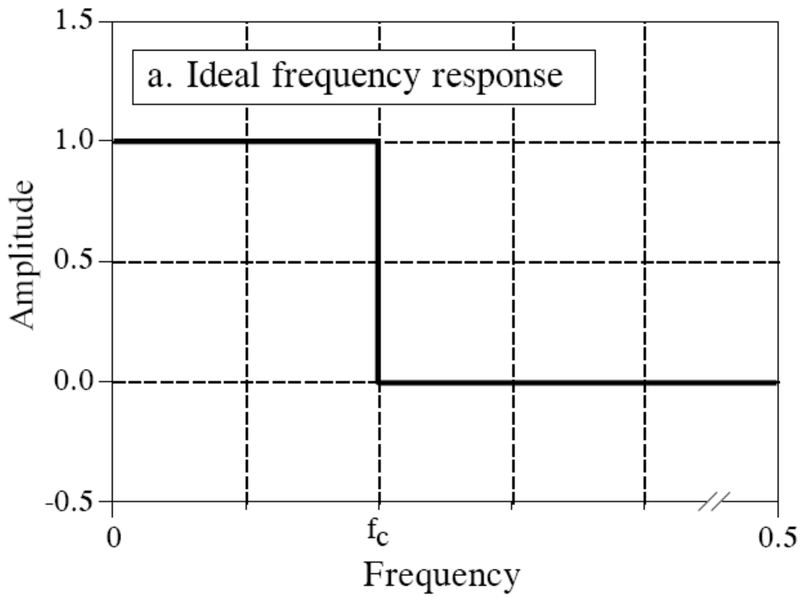
- The ideal low pass is flat in the pass band and has a step response for the attenuation



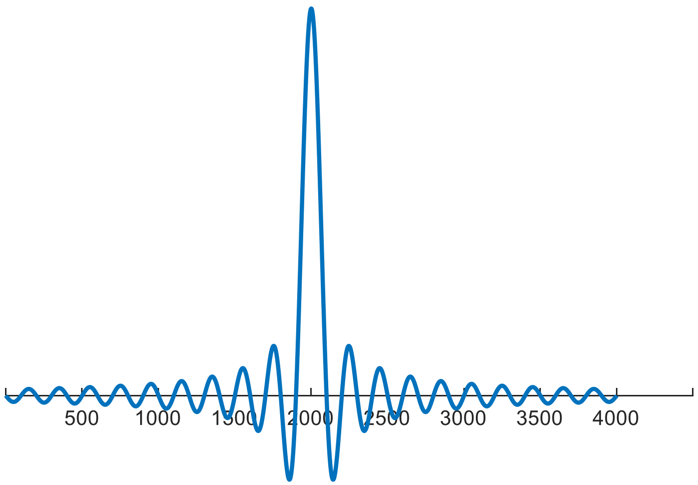
The Ideal Low Pass Filter

- Recognizing the duality between the frequency domain and the time domain, find the impulse response using the inverse Fourier Transform.

Frequency Domain



Time Domain

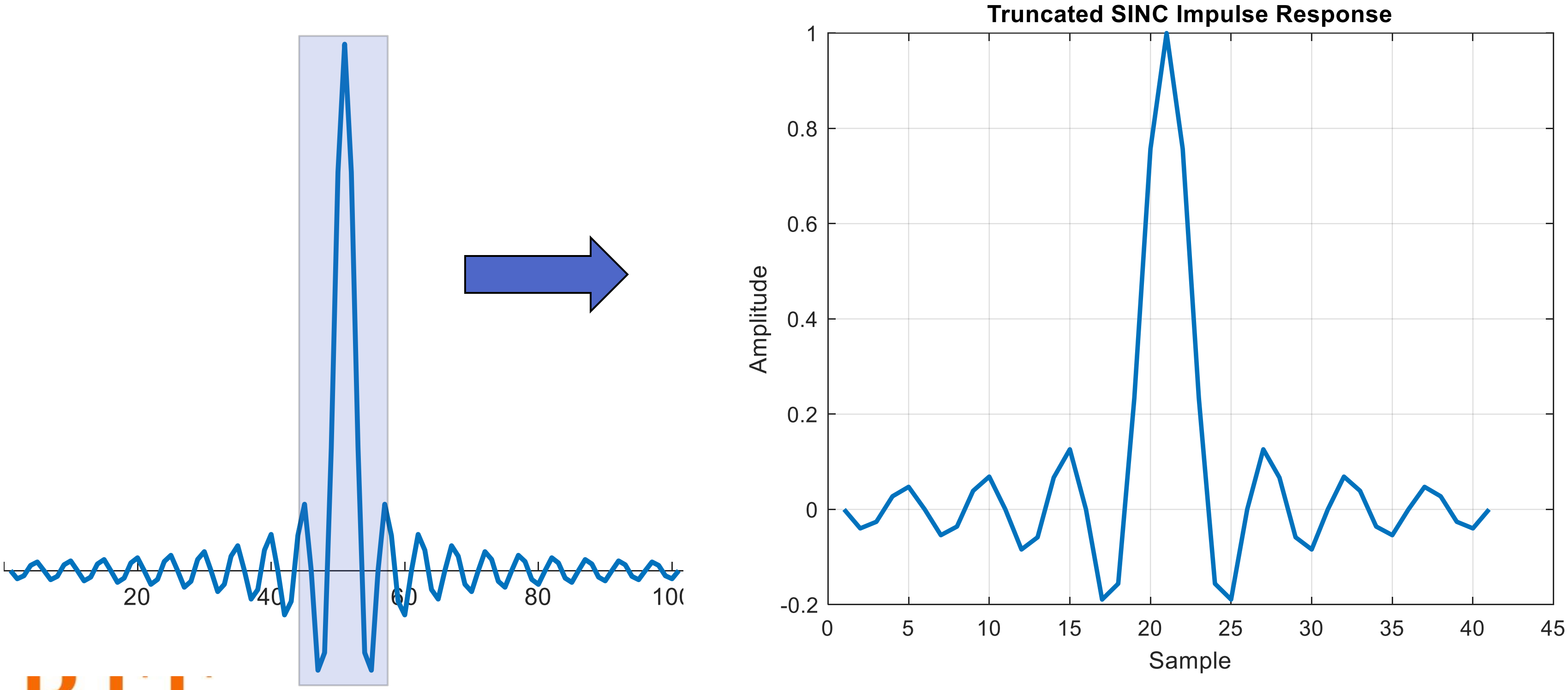


SINC Function

Extends from $-\infty\ to\ \infty$

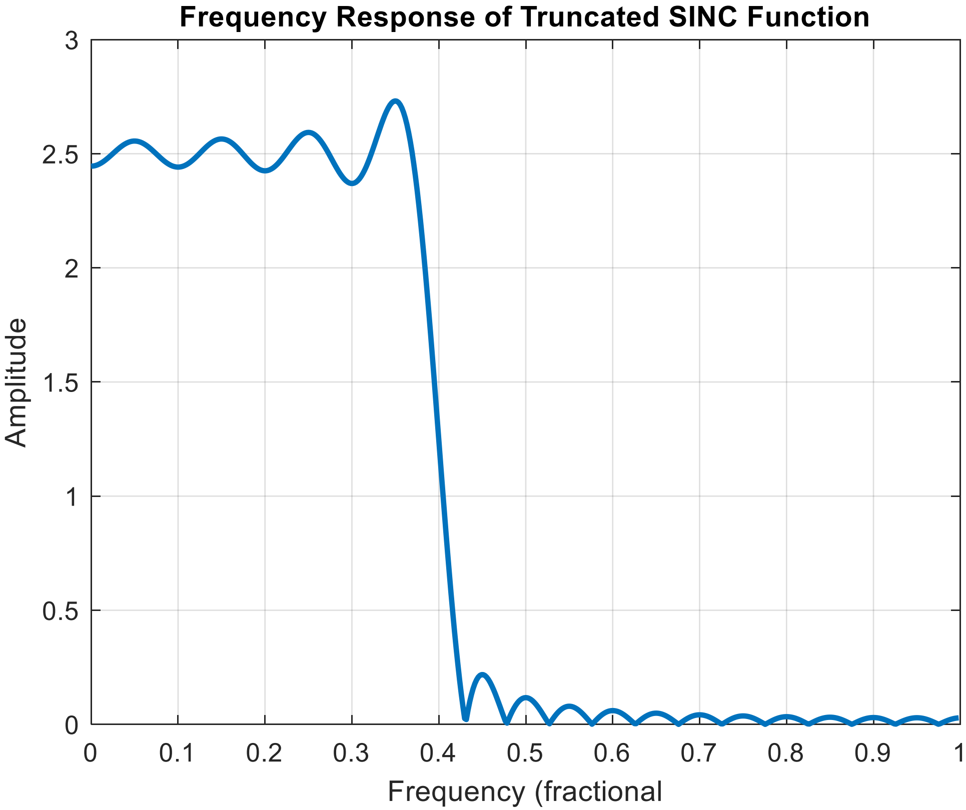
The Ideal Low Pass Filter

- The SINC function extends from $-\infty$ to ∞ in time
- It’s not possible to implement exactly
- Try truncating the sequence – e.g. 41 Samples. Plot the frequency response



The Truncated Impulse Response Frequency Response

- Truncating the response results in large ripple in the pass band and the stop band



Hamming Window

- Modify signal to reduce effect of discontinuities at left and right edges of signal.

- Apply a “window” to ease the end transitions

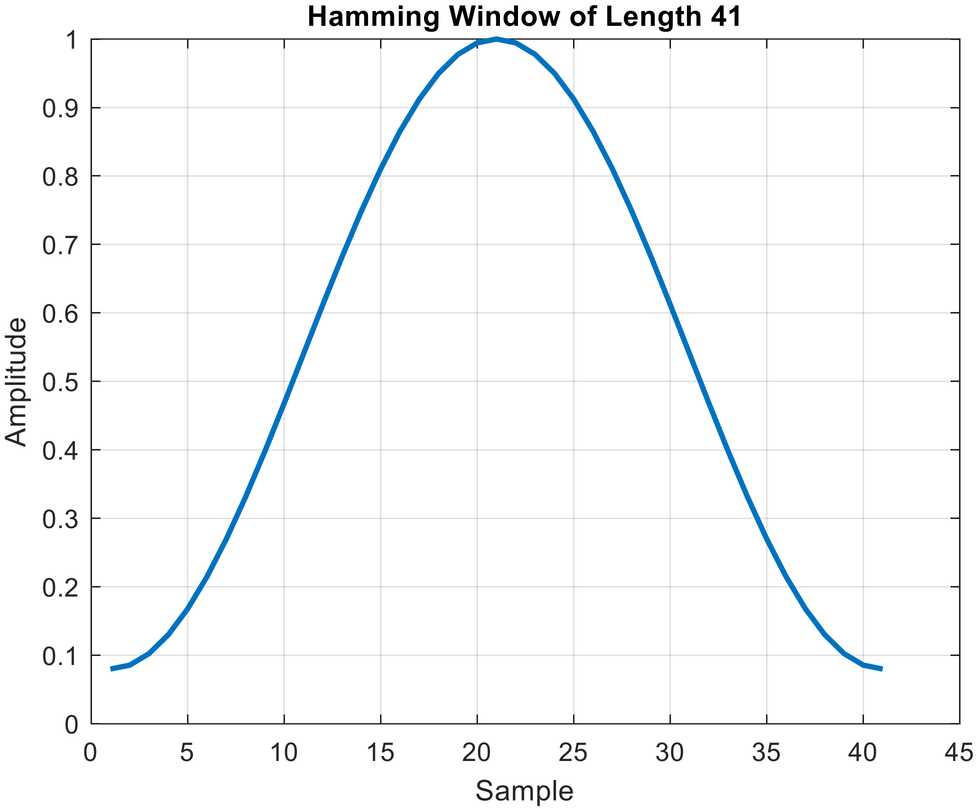
- A Hamming window has a magnitude of

$$w[n] = 0.54\ -0.46\cos\left(\frac{2\pi n}{M-1}\right)$$

- Where M is the window length and n goes from 0 to M-1

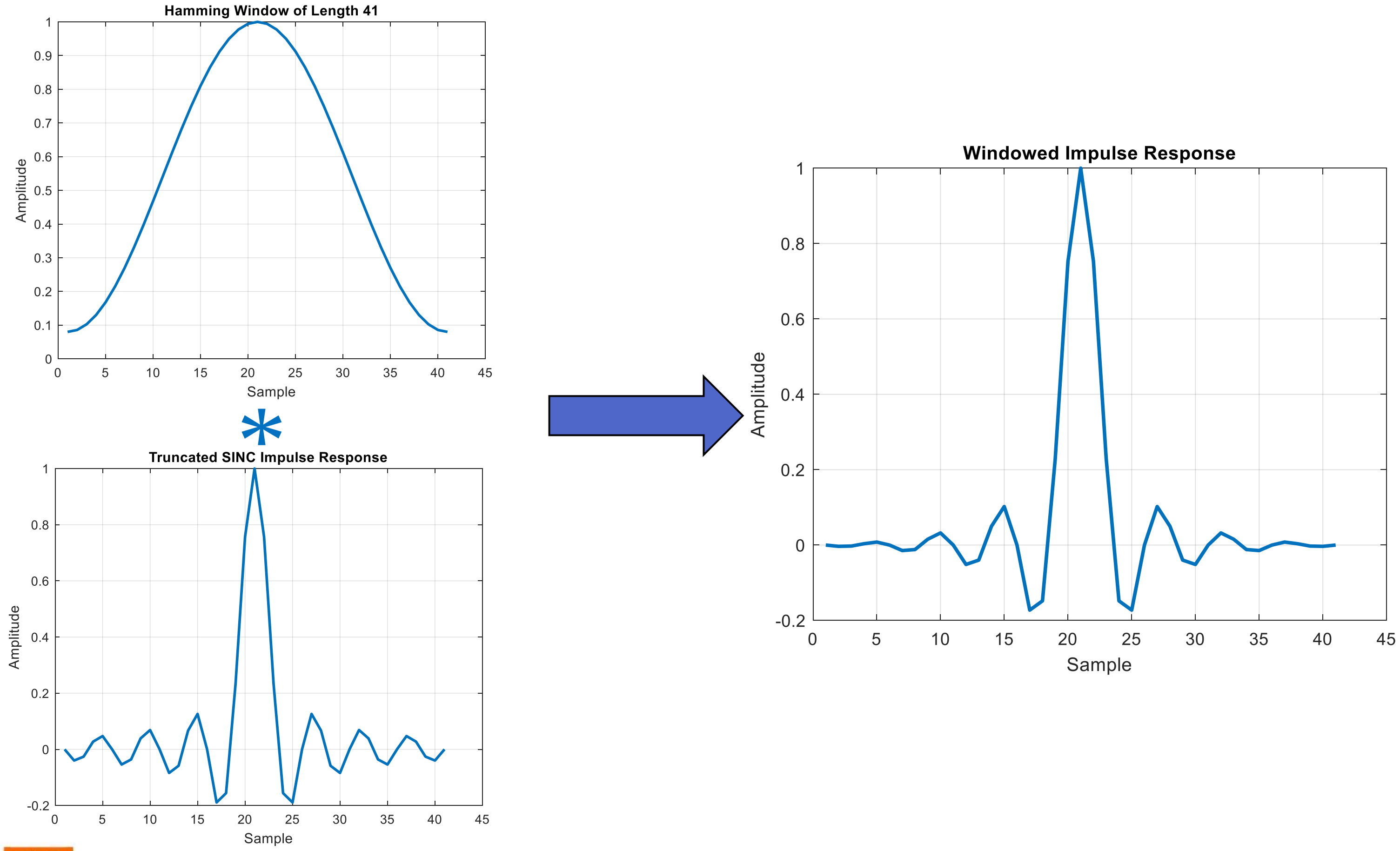
Hamming Window M = 41

$$w[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{M-1}\right)$$



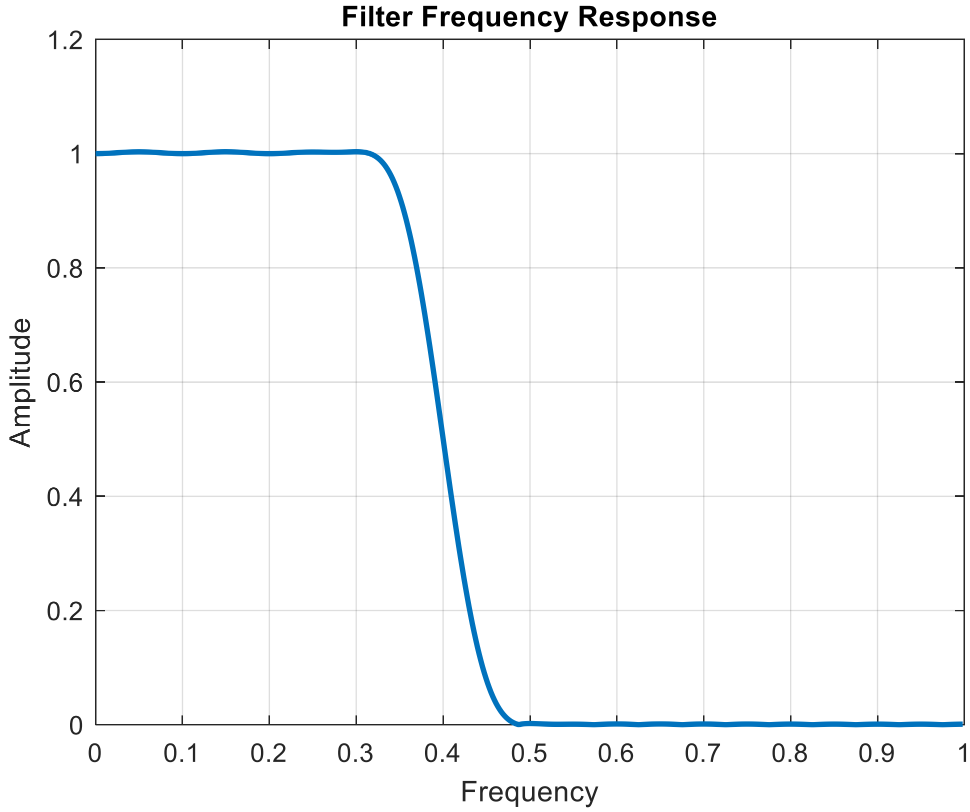
Applying the Window

- Multiply the impulse response by the window point by point



The Windowed SINC Frequency Response

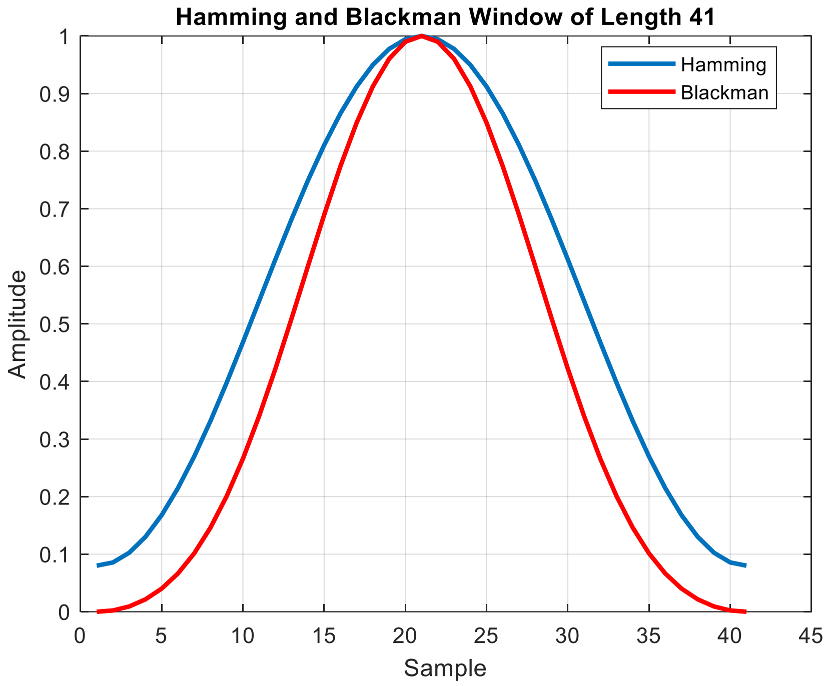
- The windowing has removed the large ripple in the pass band and stop band
- Some ripple remains but is very small



Other Windows

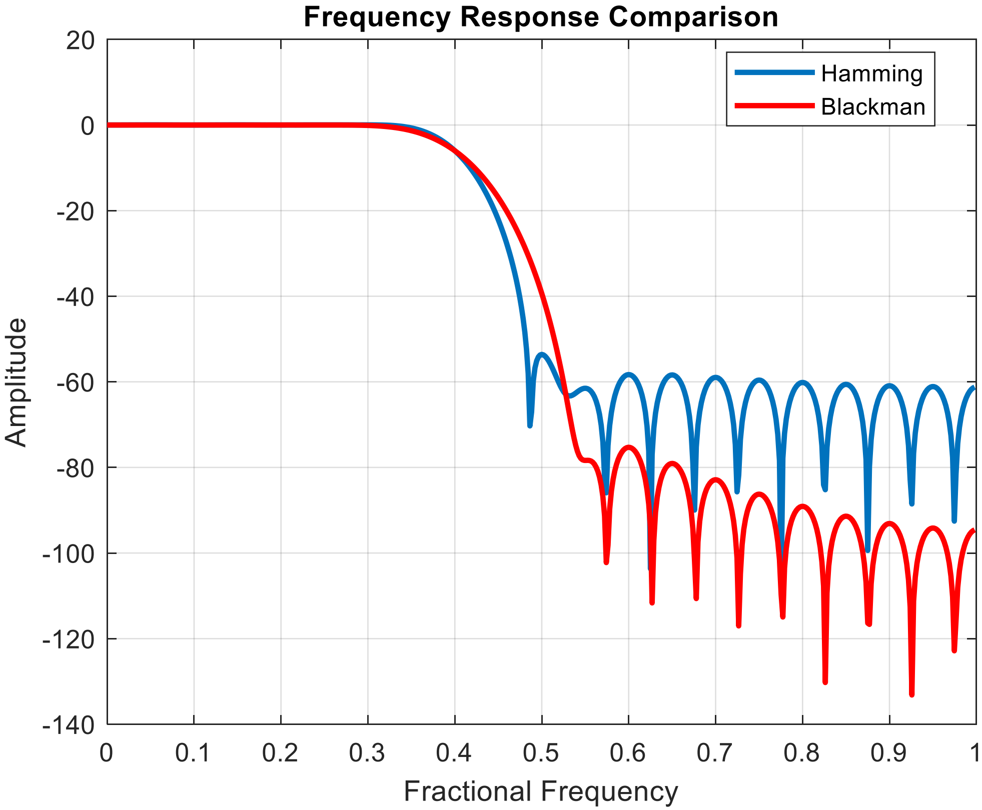
- The Blackman window is also commonly used

$$w[n] = 0.42 \; - 0.5\cos\left(\frac{2\pi n}{M-1}\right) + 0.08\cos\left(\frac{4\pi n}{M-1}\right)$$

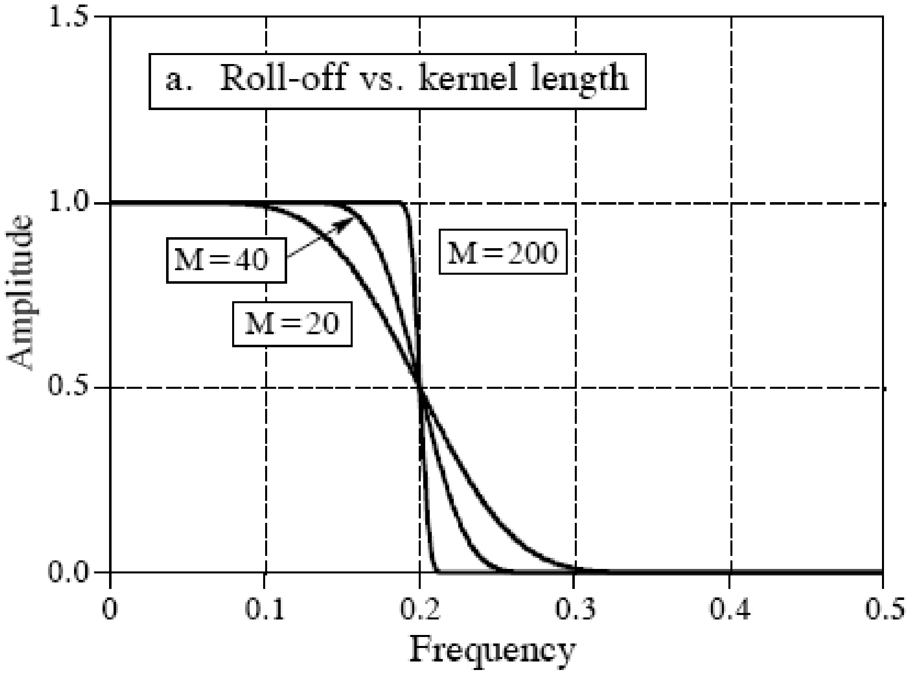


Frequency Response Comparison

- Hamming is Sharper
- Blackman has greater attenuation in the stop band
- Passband ripple similar

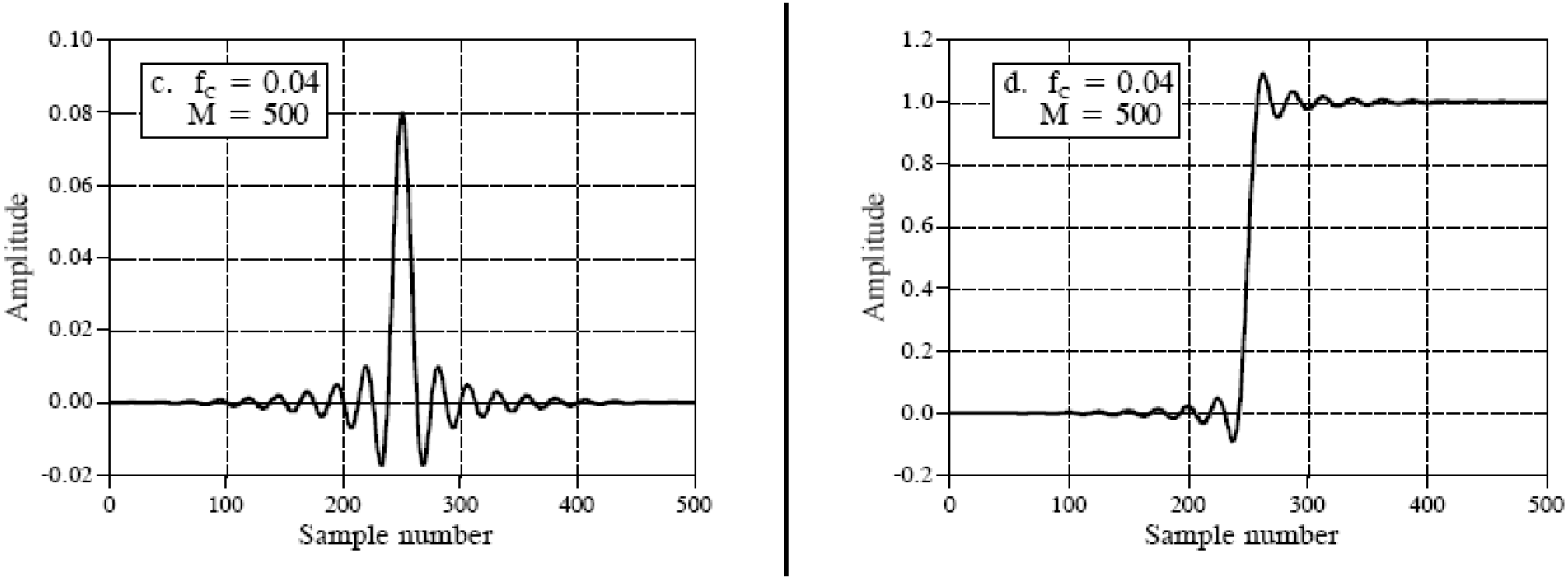


Design Options – Windowed Sinc Filter



- The filter is sharper as the length of the impulse response (kernel) increases

Time Domain Response



- The step response of the filter has overshoot
- The higher the corner frequency the higher the frequency of oscillation

Using FIR_Designer.m

- FIR_Designer.m is a MATLAB function that can help you design windowed SINC filters
  - Available in myCourses
  - Default is to create filters where Fs = 600 BPM
  - Print header for use in FIR_Generic C code
  - Fixed point coefficients
  - Computes HFXPT value
- Can create a HPF using Spectral Inversion

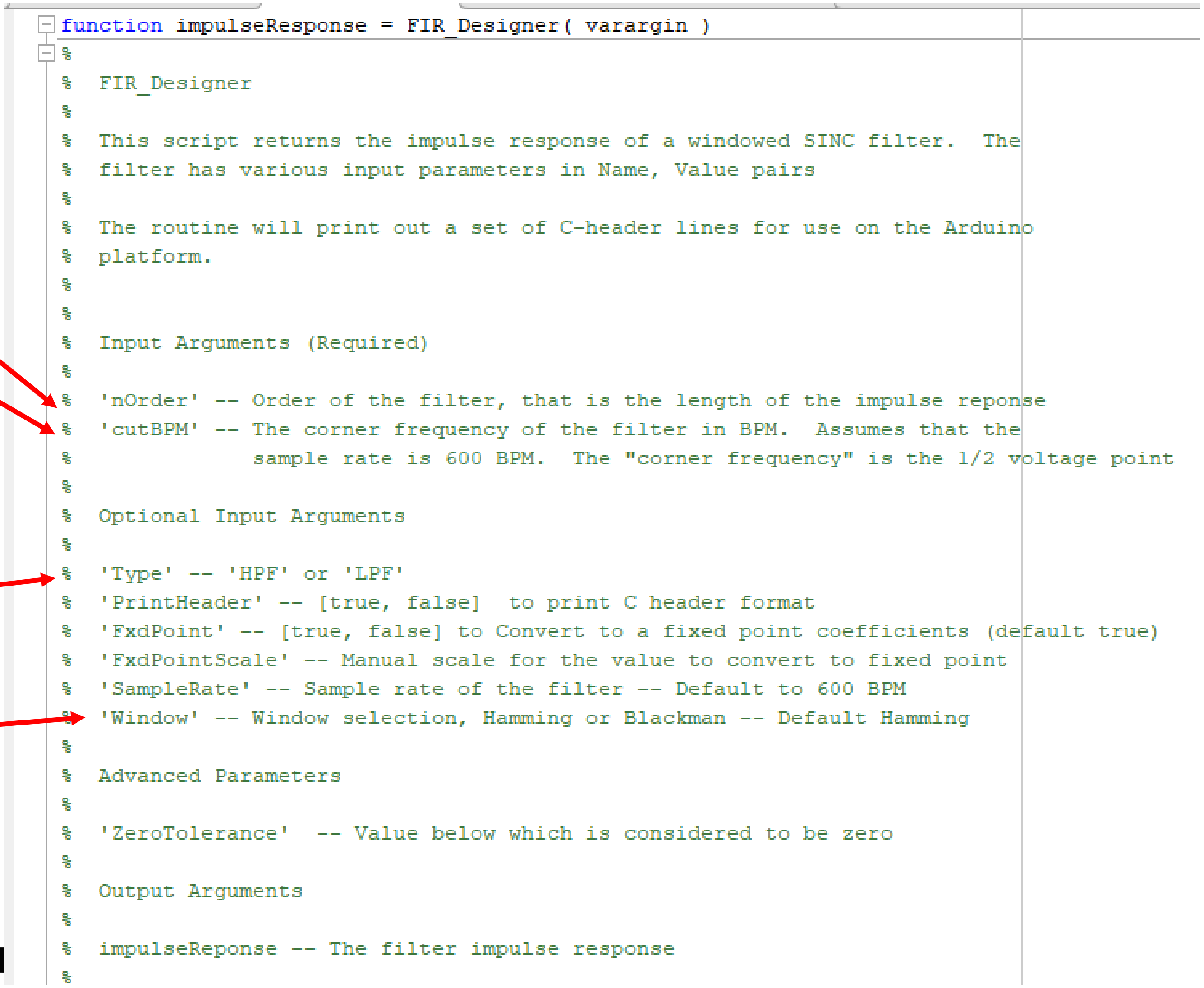
FIR_Designer.m Arguments

Filter order

Corner Freq

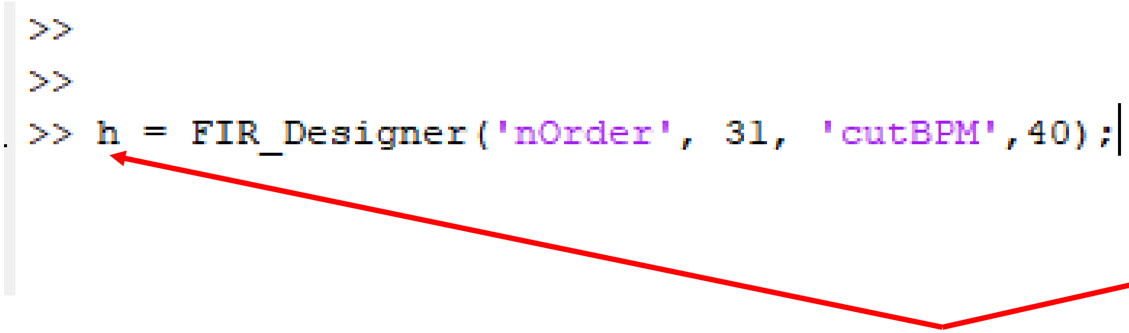
HPF/LPF

Window



```
function impulseResponse = FIR_Designer( varargin )
%
%   FIR_Designer
%
%   This script returns the impulse response of a windowed SINC filter.  The
%   filter has various input parameters in Name, Value pairs
%
%   The routine will print out a set of C-header lines for use on the Arduino
%   platform.
%
%
%   Input Arguments (Required)
%
%   'nOrder' -- Order of the filter, that is the length of the impulse reponse
%   'cutBPM' -- The corner frequency of the filter in BPM.  Assumes that the
%               sample rate is 600 BPM.  The "corner frequency" is the 1/2 voltage point
%
%   Optional Input Arguments
%
%   'Type' -- 'HPF' or 'LPF'
%   'PrintHeader' -- [true, false]  to print C header format
%   'FxdPoint' -- [true, false] to Convert to a fixed point coefficients (default true)
%   'FxdPointScale' -- Manual scale for the value to convert to fixed point
%   'SampleRate' -- Sample rate of the filter -- Default to 600 BPM
%   'Window' -- Window selection, Hamming or Blackman -- Default Hamming
%
%   Advanced Parameters
%
%   'ZeroTolerance'  -- Value below which is considered to be zero
%
%   Output Arguments
%
%   impulseReponse -- The filter impulse response
%
```

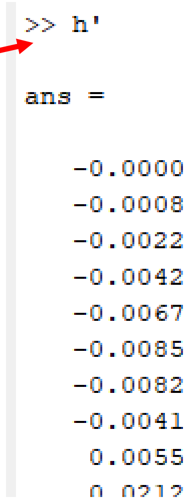
Simple Call



```
>>
>>
>> h = FIR_Designer('nOrder', 31, 'cutBPM',40);
```

- LPF
- Default Fs = 600 BPM

Return Variable h

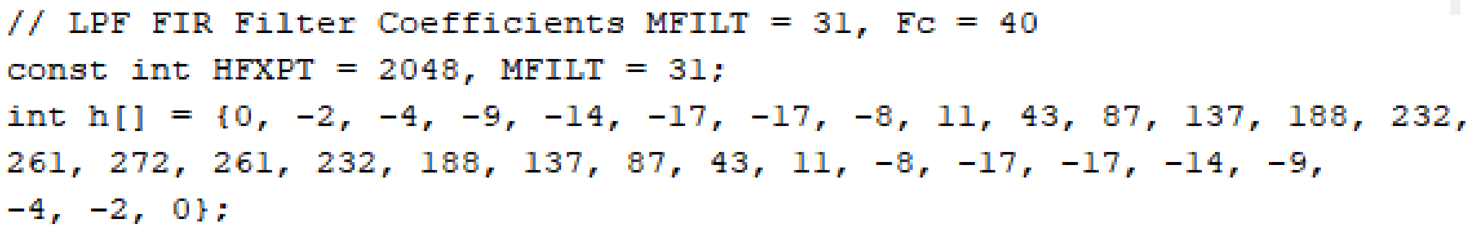


```
>> h'

ans =

   -0.0000
   -0.0008
   -0.0022
   -0.0042
   -0.0067
   -0.0085
   -0.0082
   -0.0041
    0.0055
    0.0212
```

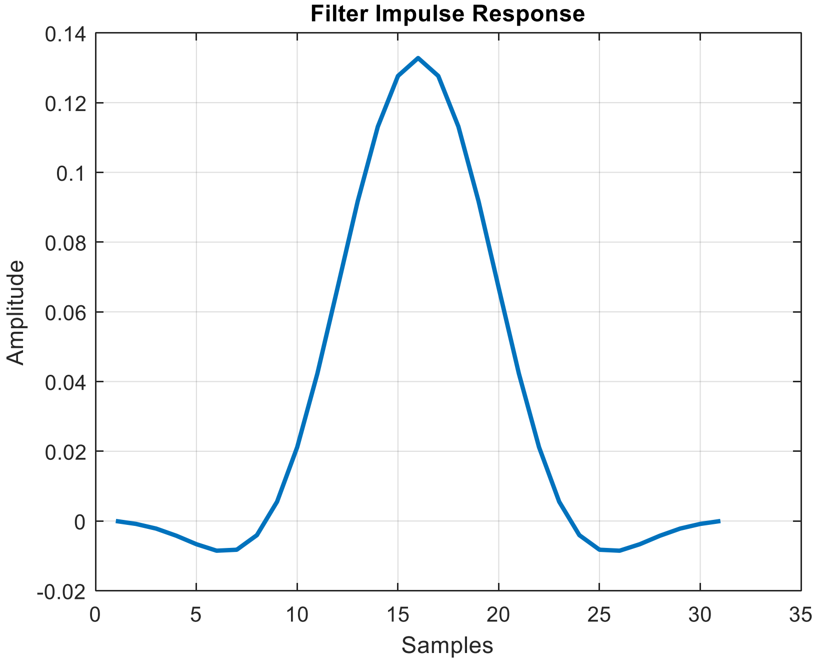
C-Code Output

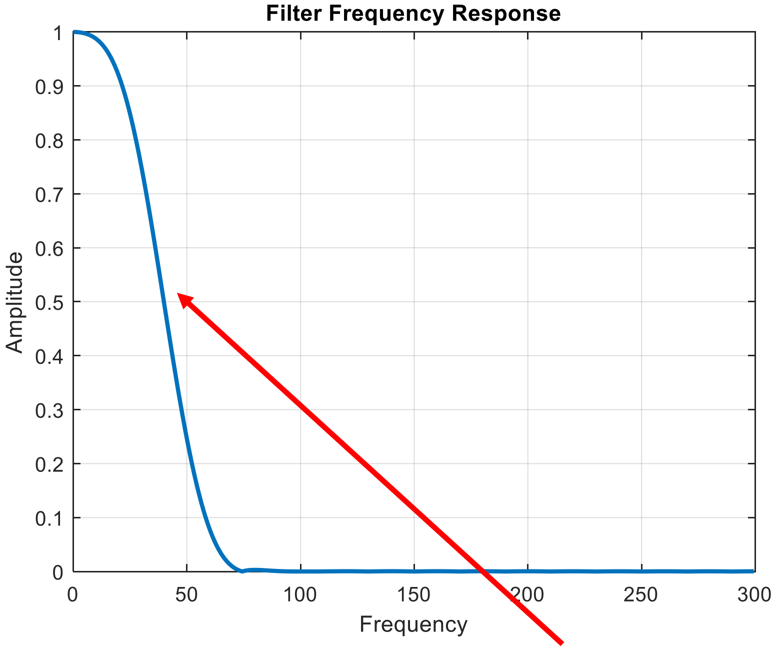


```
// LPF FIR Filter Coefficients MFILT = 31, Fc = 40
const int HFXPT = 2048, MFILT = 31;
int h[] = {0, -2, -4, -9, -14, -17, -17, -8, 11, 43, 87, 137, 188, 232,
261, 272, 261, 232, 188, 137, 87, 43, 11, -8, -17, -17, -14, -9,
-4, -2, 0};
```

Function Output

- Function returns the impulse response and two graphs





cutBPM at 0.5

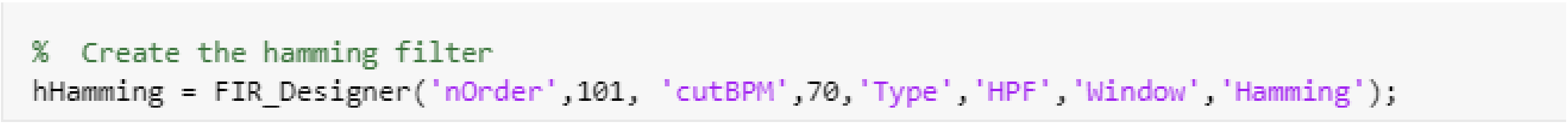
40 BPM

Windowed SINC Filter ICP

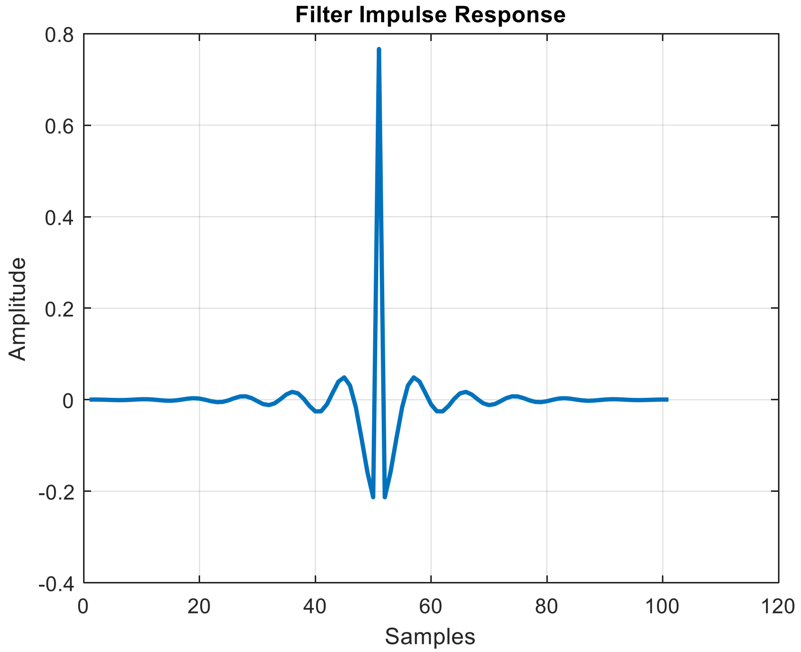
- Design a Windowed SINC filter with the following specifications
  - HPF
  - Filter order – 101
  - Cut frequency (1/2 voltage) – 70
  - Window (Hamming and Blackman)
    - Compare frequency responses in decibels
    - Save the output variable for each filter – Impulse response. Take the FFT of each impulse response

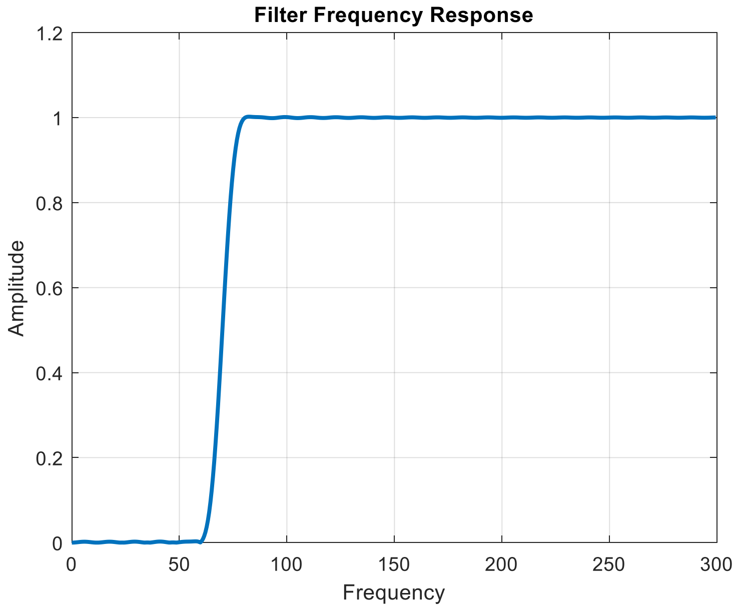
Windowed SINC Filter ICP Hamming Window

NOPRINT



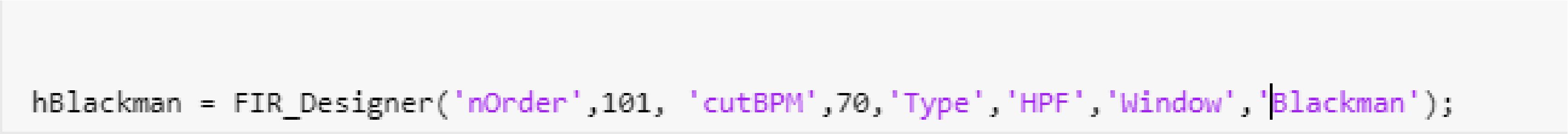
```
%  Create the hamming filter
hHamming = FIR_Designer('nOrder',101, 'cutBPM',70,'Type','HPF','Window','Hamming');
```



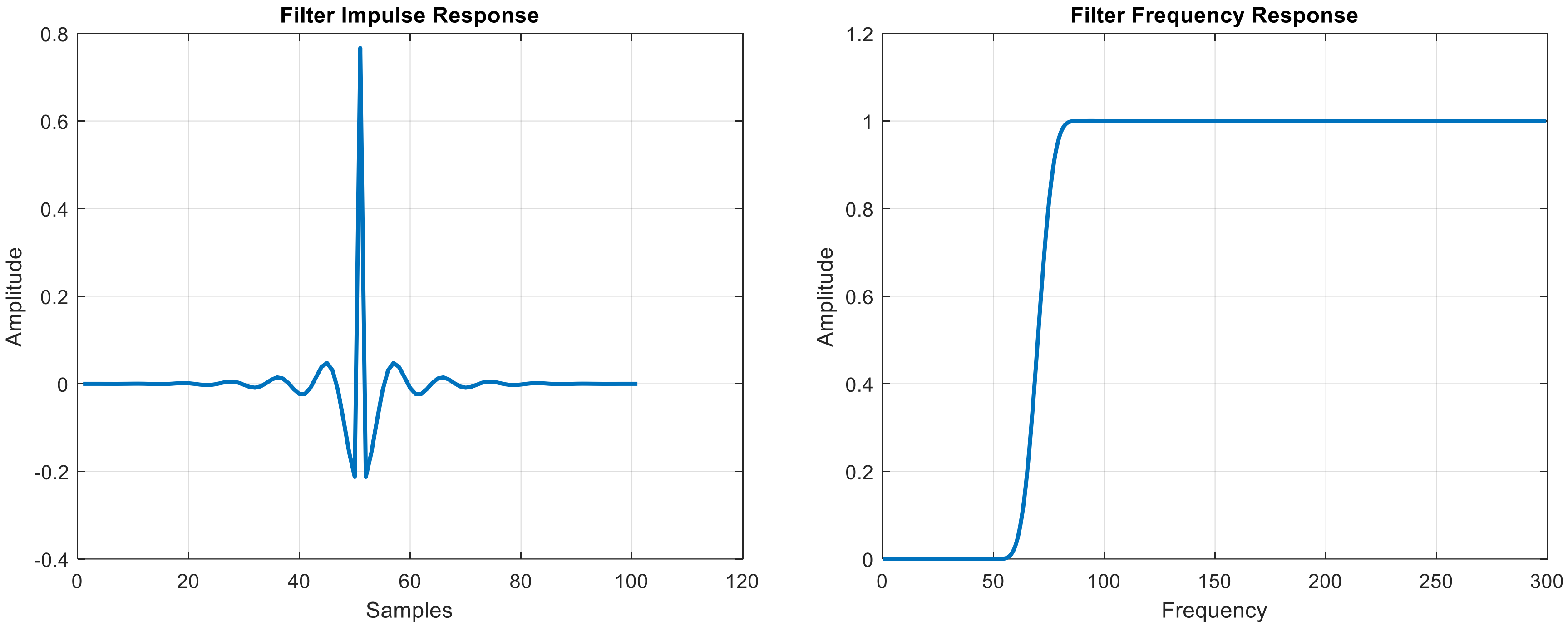


Windowed SINC Filter ICP
Blackman Window

NOPRINT



```
hBlackman = FIR_Designer('nOrder',101, 'cutBPM',70,'Type','HPF','Window','Blackman');
```

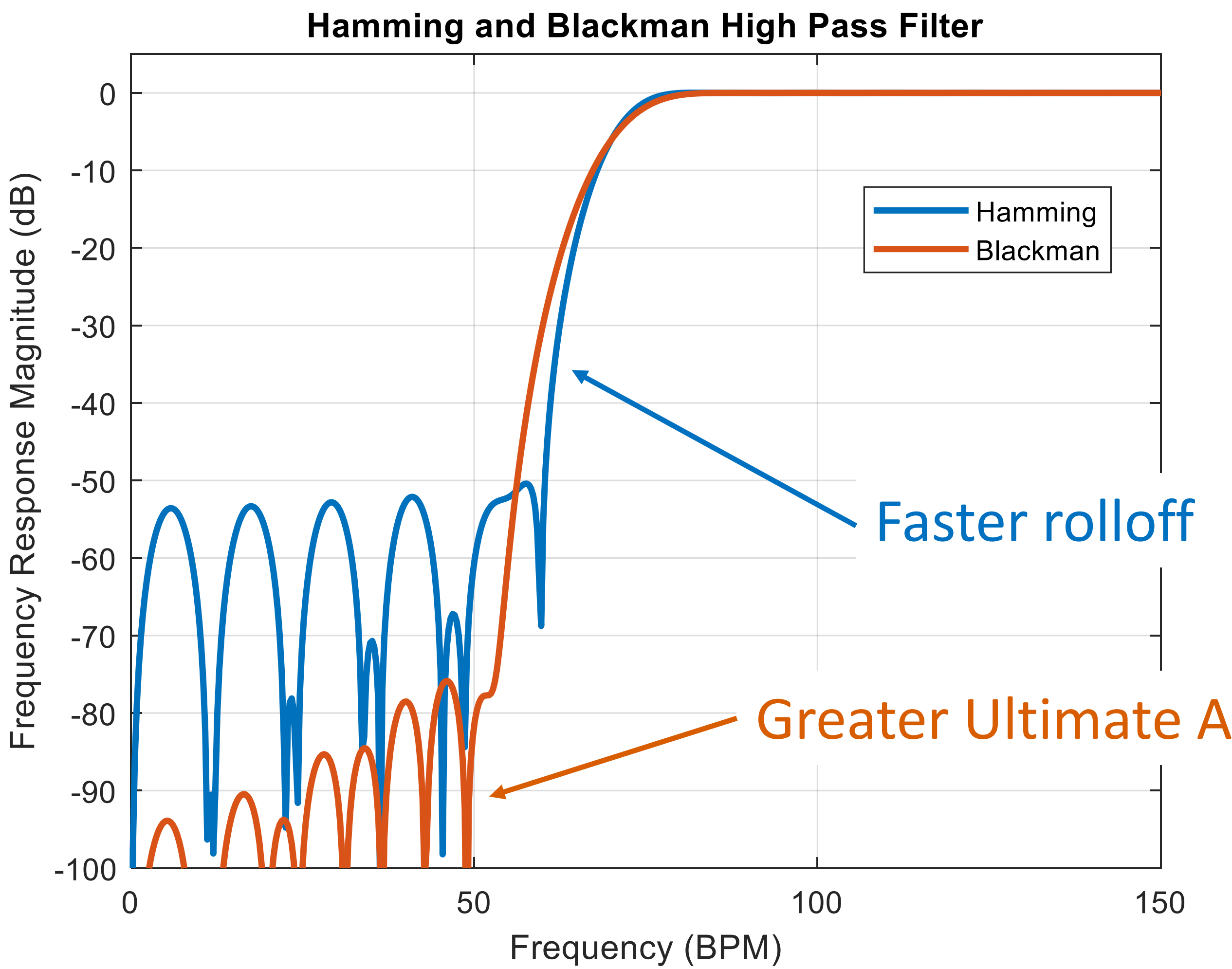


Filter Impulse Response

Filter Frequency Response

Windowed SINC HPF Comparison

NOPRINT



Faster rolloff

Greater Ultimate Attenuation

Summary

- Discussed the ideal low pass filter and its dual impulse response in the frequency domain
- SINC function the Impulse Response but infinite in length
- Truncation of the impulse response causes ripple in the pass and stop band
- Improved the response via windowing
  - Hamming Window
  - Blackman Window
- Compared the Frequency Response between the two filter types