IEE1711 Applied signal processing

Julia Berdnikova
julia.berdnikova@taltech.ee

2019 spring

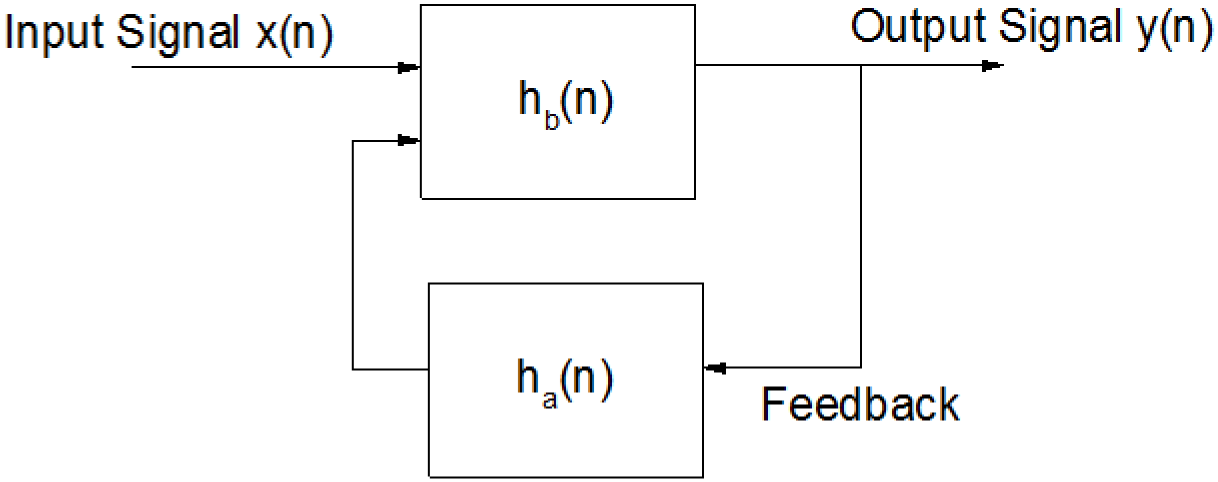
TAL
TECH

Books and Slides

- John G. Proakis, Dimitris K Manolakis, Digital Signal Processing (4th Edition) (Chapter 9.2, 10.1, 10.2)
- Filter Introduction in Moodle from previous course

IEE1710 Signal processing methods and algorithms (Lecture 11)

Infinite impulse response (IIR) filter



The output values of IIR filters are calculated by adding the weighted sum of previous and current input values to the weighted sum of previous output values.

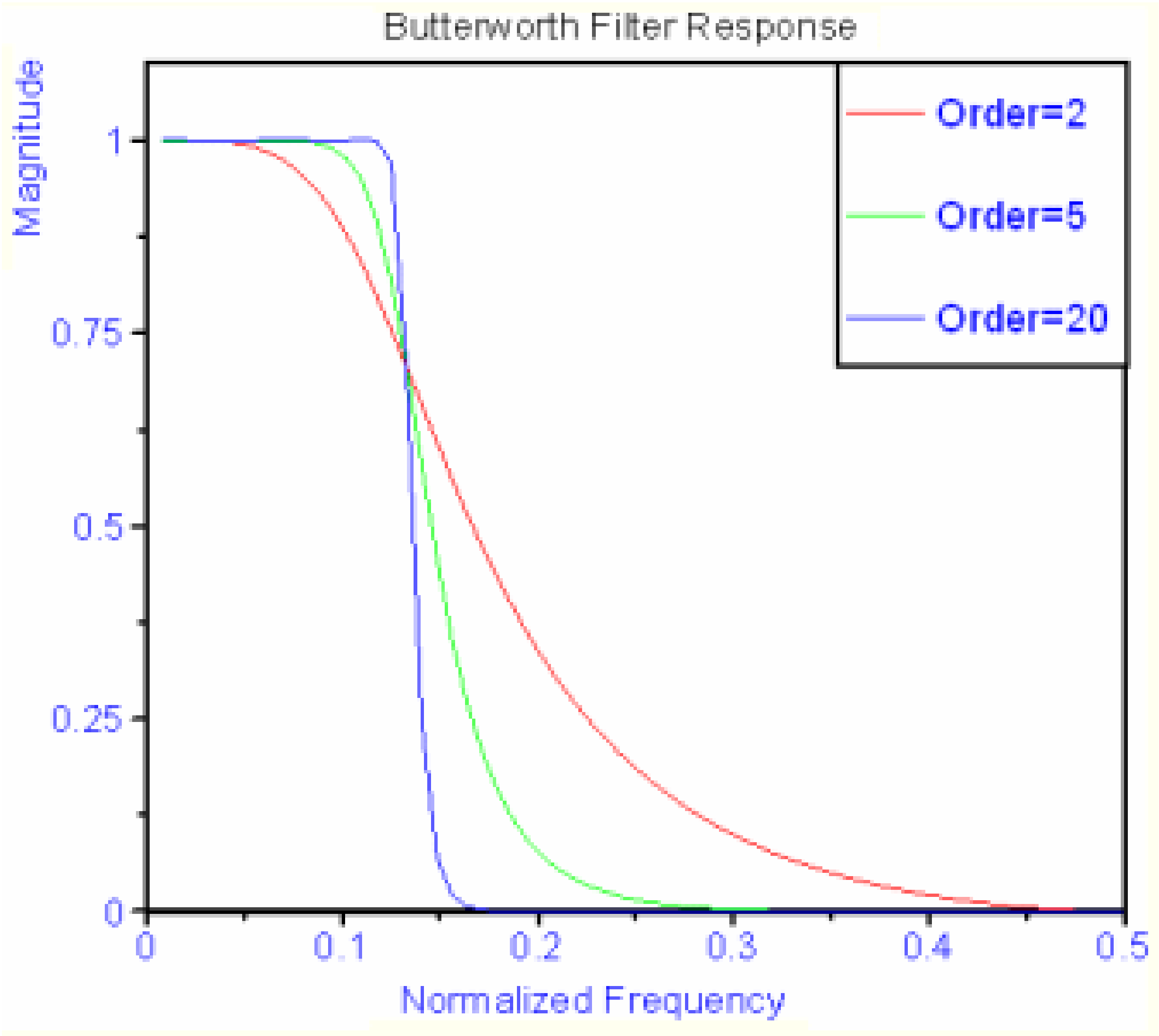
If the input values are *x(n)* and the output values *y(n)* the difference equation defines the IIR filter:

$$y(n) = \frac{1}{a_0}\left[-\sum_{j=1}^{N_y-1} a(j)y(n-j) + \sum_{k=0}^{N_x-1} b(k)x(n-k)\right]$$

The number of forward coefficients N_x and the number of reverse coefficients N_y is usually equal and is the filter order.

https://www.originlab.com/doc/Origin-Help/IIR-Filter-Algorithm
http://zone.ni.com/reference/en-XX/help/370858N-01/genmaths/genmaths/calc_filterfir_iir/

IIR filter order



http://zone.ni.com/reference/en-XX/help/370858N-01/genmaths/genmaths/calc_filterfir_iir/

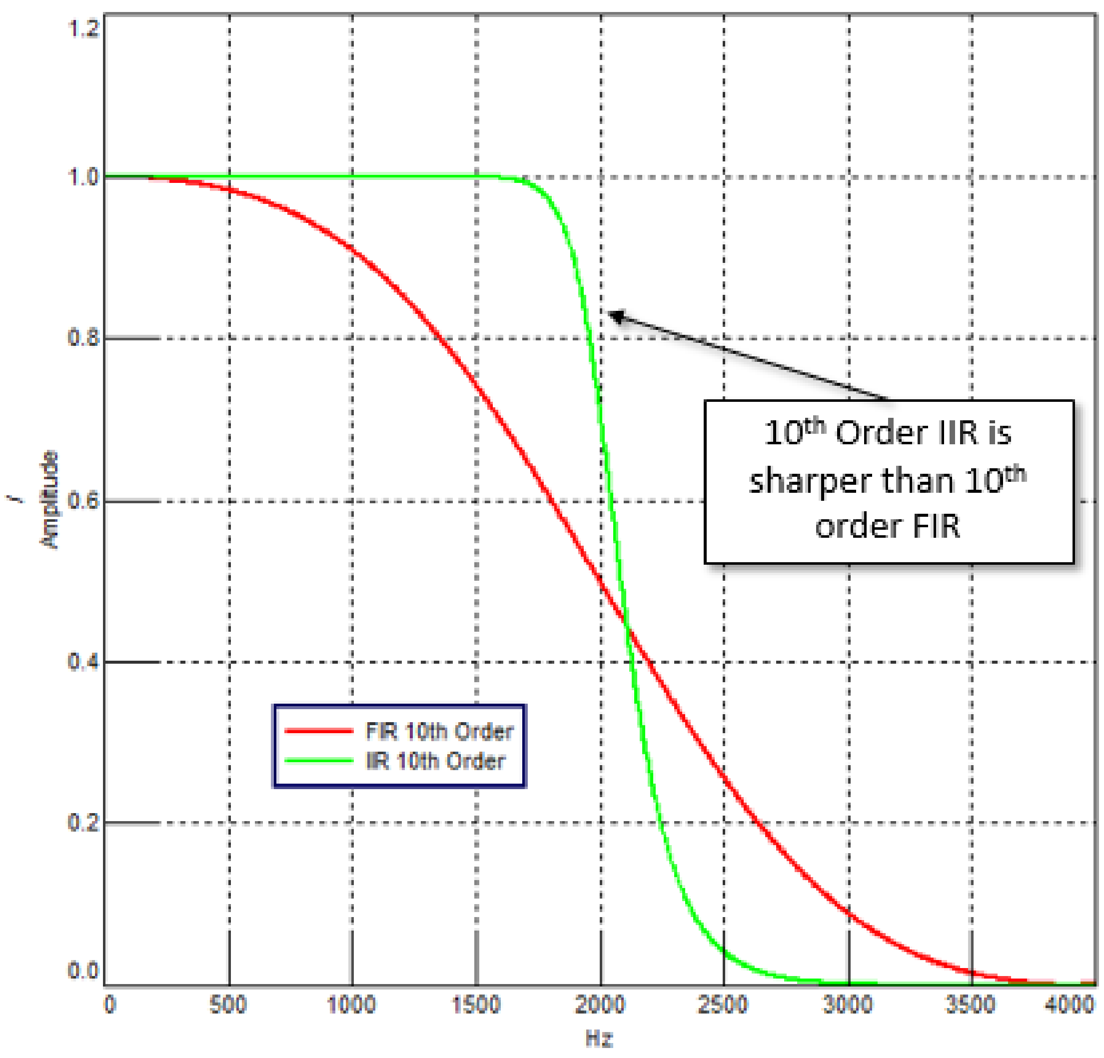
Infinite impulse response (IIR) filter vs Finite impulse response (FIR) filter

FIR filters	IIR filters
non-recursive and stable	recursive (feedback) and not stable
do not have analog equivalent	have analog equivalent
less efficient	more efficient
usually used as anti-aliasing, low pass and baseband filters.	usually used as notch (band reject), band pass functions.
need higher order	lower order to achieve same performance
transfer function have only zeros	transfer function have zeros and poles
require more memory	require less memory
linear phase response	non-linear phase response

http://www.rfwireless-world.com/Terminology/FIR-filter-vs-IIR-filter.html

TAL TECH

IIR Versus FIR Filters
Comparison with same order



https://siemensplm.i.lithium.com/t5/image/serverpage/image-id/65988i70487B55A51D6CA4?v=1.0

IIR filter design

- Specify filter specification.
- Specify low pass analog filter prototype, and the available Butterworth, Chebyshev Type I,

  Chebyshev Type II, Elliptic jtc
- Frequency transform for analog filter
- Convert analog filter into a digital filter

TAL
TECH

Filter prototype functions

- **Butterworth Filters**

The frequency response of the Butterworth filter has no ripples in the passband and the stopband. Therefore it is called a maximally flat filter. The advantage of Butterworth filters is the smooth, monotonically decreasing frequency response in the transition region.

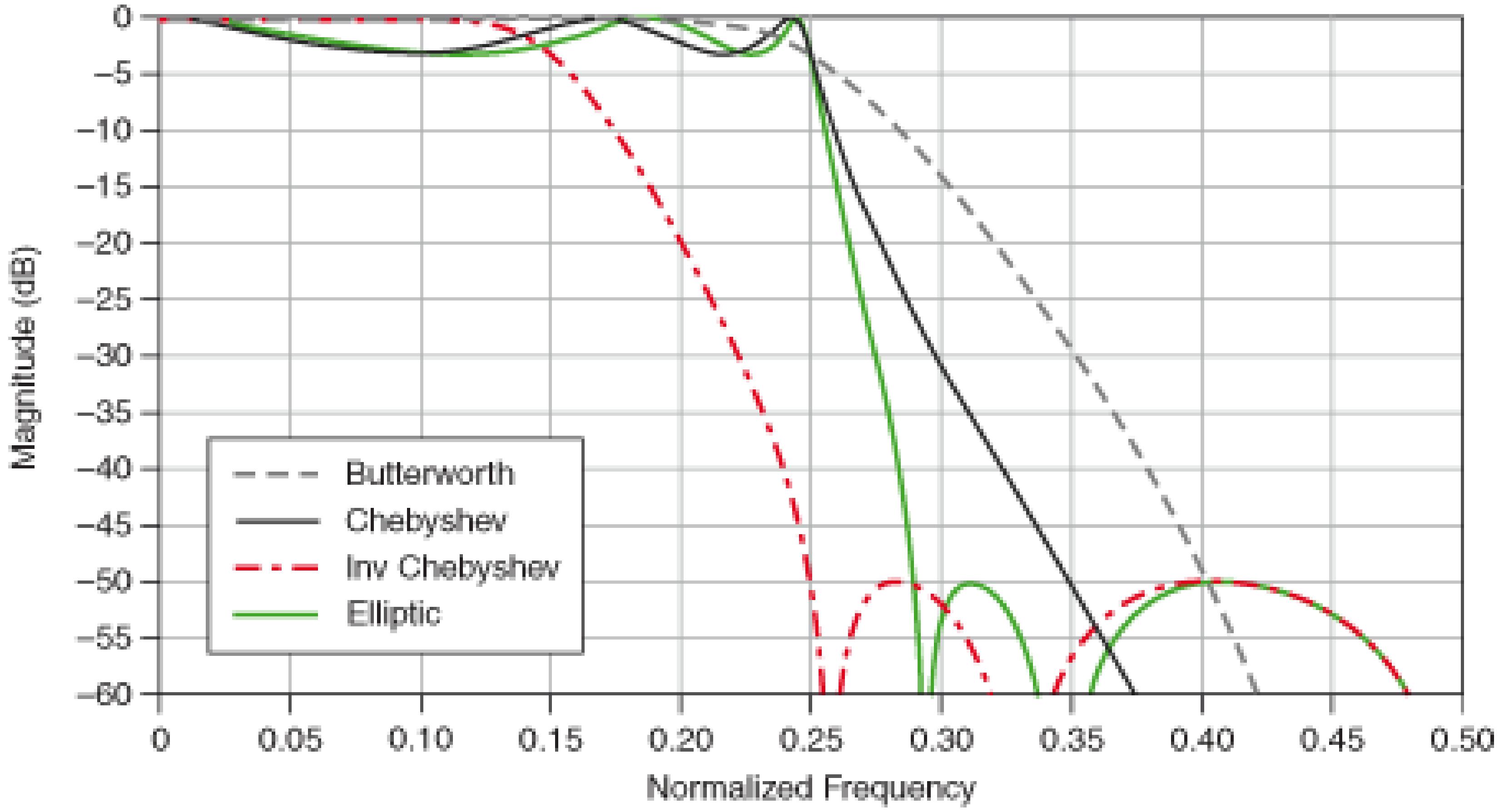
- **Chebyshev Filters**

If the filter is the same, the frequency response of the Chebyshev filter has a norrower transition range than the frequency response of the Butterworth filter which results in a passband with more ripples. The frequency response characteristics of Chebyshev filters have an equiripple magnitude response in the passband, monotonically decreasing magnitude response in the stopband, and a sharper rolloff in the transition region as compared to Butterworth filters of the same order.

- **Bessel Filters**

The frequency response of Bessel filters is similar to the Butterworth filter smooth in the passband and in the stopband. If the filter order is the same, the stopband attenuation of the Bessel filter is much lower than that of the Butterworth filter. Of all filter types the Bessel filter has the widest transition range if the filter order is fixed.

http://zone.ni.com/reference/en-XX/help/370858N-01/genmaths/genmaths/calc_filterfir_iir/



https://www.physik.uzh.ch/local/teaching/SPI301/LV-2015-Help/lvanlsconcepts.chm/lvac_iir_filter_types.html

IIR Filter	**Ripple in Passband?**	**Ripple in Stopband?**	**Transition Bandwidth for a Fixed Order**	**Order for Given Filter Specificatio n**
Butterworth	No	No	Widest	Highest
Chebyshev	Yes	No	Narrower	Lower
Chebyshev II	No	Yes	Narrower	Lower
Elliptic	Yes	Yes	Narrowest	Lowest

https://www.physik.uzh.ch/local/teaching/SPI301/LV-2015-Help/lvanlsconcepts.chm/lvac_iir_filter_types.html

IIR analog filter prototype

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{N} b_i z^{-i}}{\sum_{j=0}^{M} a_j z^{-j}}$$

Method	Squared Magnitude Response Function	Analog Filter Transfer Function
Butterworth	$\lvert H_a(j\Omega)\rvert^2 = \frac{1}{1+\Omega^{2N}}$	$H_a(s) = \frac{q(s)}{p(s)} = \frac{g}{(s-p_1)(s-p_2)\cdots(s-p_N)} = \frac{g}{\prod_{k=1}^{N}(s-p_k)}$
Chebyshev Type I	$\lvert H_a(j\Omega)\rvert^2 = \frac{1}{1+\varepsilon^2 T_N^2(\Omega)}$	$H_a(s) = \frac{q(s)}{p(s)} = \frac{g}{(s-p_1)(s-p_2)\cdots(s-p_N)} = \frac{g}{\prod_{k=1}^{N}(s-p_k)}$
Chebyshev Type II	$\lvert H_a(j\Omega)\rvert^2 = \frac{1}{1+(\varepsilon^2 T_N^2(\frac{1}{\Omega}))^{-1}} = \frac{\varepsilon^2 T_N^2(\frac{1}{\Omega})}{1+\varepsilon^2 T_N^2(\frac{1}{\Omega})}$	$H_a(s) = \frac{q(s)}{p(s)} = g\frac{(s-q_1)(s-q_2)\cdots(s-q_N)}{(s-p_1)(s-p_2)\cdots(s-p_N)} = g\prod_{k=1}^{N}\frac{(s-q_k)}{(s-p_k)}$
Elliptic	$\lvert H_a(j\Omega)\rvert^2 = \frac{1}{1+\varepsilon^2 U_N^2(\Omega)}$	$H_a(s) = \frac{q(s)}{p(s)} = g\frac{(s-q_1)(s-q_2)\cdots(s-q_N)}{(s-p_1)(s-p_2)\cdots(s-p_N)} = g\frac{\prod_{i=1}^{N}(s-q_i)}{\prod_{j=1}^{N}(s-p_j)}$

In the table above, Ω is the frequency, N is the filter order, ε is the maximum oscillation in the passband frequency response, T_N is the Chebyshev polynomial, U_N is the Jacobian elliptic function, g is the scalar gain, s is the plane of Laplace transform, q_k or q_i is the zero, and p_k or p_j is the pole.

TAL TECH

https://www.originlab.com/doc/Origin-Help/IIR-Filter-Algorithm

FILTER USED FOR:	FILTER IMPLEMENTED BY:	
	Convolution *Finite Impulse Response (FIR)*	Recursion *Infinite Impulse Response (IIR)*
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)

TABLE 14-1
Filter classification. Filters can be divided by their *use*, and how they are *implemented*.

https://www.dspguide.com/ch14/6.htm

Matlab design of IIR filters

Filter Method	Description	Filter Functions
Analog Prototyping	Using the poles and zeros of a classical lowpass prototype filter in the continuous (Laplace) domain, obtain a digital filter through frequency transformation and filter discretization.	Complete design functions: besself, butter, cheby1, cheby2, ellip Order estimation functions: buttord, cheb1ord, cheb2ord, ellipord Lowpass analog prototype functions: besselap, buttap, cheb1ap, cheb2ap, ellipap Frequency transformation functions: lp2bp, lp2bs, lp2hp, lp2lp Filter discretization functions: bilinear, impinvar
Direct Design	Design digital filter directly in the discrete time-domain by approximating a piecewise linear magnitude response.	yulewalk
Generalized Butterworth Design	Design lowpass Butterworth filters with more zeros than poles.	maxflat
Parametric Modeling	Find a digital filter that approximates a prescribed time or frequency domain response. (See System Identification Toolbox™ documentation for an extensive collection of parametric modeling tools.)	Time-domain modeling functions: lpc, prony, stmcb Frequency-domain modeling functions: invfreqs, invfreqz

TAL TECH

https://www.mathworks.com/help/signal/ug/iir-filter-design.html

Matlab filter functions

Filter Type	Design Function
Bessel (analog only)	[b,a] = besself(n,Wn,*options*) [z,p,k] = besself(n,Wn,*options*) [A,B,C,D] = besself(n,Wn,*options*)
Butterworth	[b,a] = butter(n,Wn,*options*) [z,p,k] = butter(n,Wn,*options*) [A,B,C,D] = butter(n,Wn,*options*)
Chebyshev Type I	[b,a] = cheby1(n,Rp,Wn,*options*) [z,p,k] = cheby1(n,Rp,Wn,*options*) [A,B,C,D] = cheby1(n,Rp,Wn,*options*)
Chebyshev Type II	[b,a] = cheby2(n,Rs,Wn,*options*) [z,p,k] = cheby2(n,Rs,Wn,*options*) [A,B,C,D] = cheby2(n,Rs,Wn,*options*)
Elliptic	[b,a] = ellip(n,Rp,Rs,Wn,*options*) [z,p,k] = ellip(n,Rp,Rs,Wn,*options*) [A,B,C,D] = ellip(n,Rp,Rs,Wn,*options*)

TAL
TECH

Matlab filter design example

Suppose you want a bandpass filter with a passband from 1000 to 2000 Hz, stopbands starting 500 Hz away on either side, a 10 kHz sampling frequency, at most 1 dB of passband ripple, and at least 60 dB of stopband attenuation.

Butterworth filter ([n,Wn] = buttord(Wp,Ws,Rp,Rs)):

buttord returns the lowest order, ***n***, of the digital Butterworth filter with no more than ***Rp*** dB of passband ripple and at least ***Rs*** dB of attenuation in the stopband. ***Wp*** and ***Ws*** are respectively the passband and stopband edge frequencies of the filter, normalized from 0 to 1, where 1 corresponds to π rad/sample. The scalar (or vector) of corresponding cutoff frequencies, Wn, is also returned.

butter *returns the transfer function coefficients of an* ***n****th-order lowpass digital Butterworth filter*

```
[n,Wn] = buttord([1000 2000]/5000,[500 2500]/5000,1,60)
[b,a] = butter(n, Wn);
freqz(b,a);
dataIn = randn(1000,1);
dataOut = filter(b,a,dataIn);
```

n = 12   Wn = 0.1951 0.4080

Matlab filter design example

Elliptic filter:

```
[n,Wn] = ellipord([1000 2000]/5000,[500 2500]/5000,1,60)
[b,a] = ellip(n,1,60,Wn);
```

n = 5 Wn = 0.2000 0.4000

IIR Filter design example (Matlab)

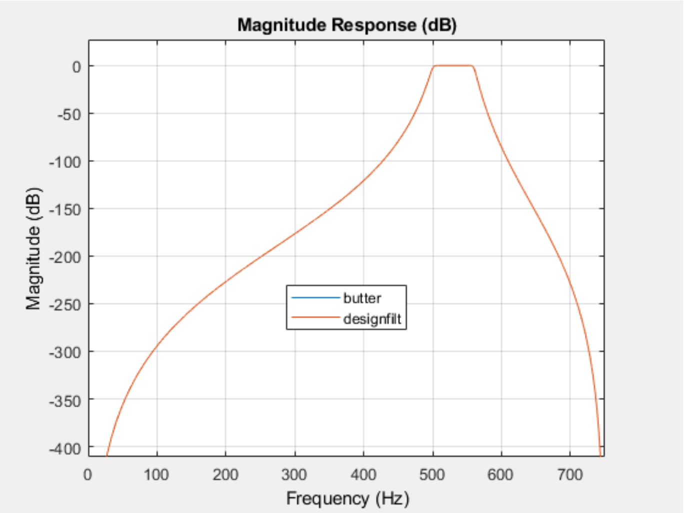
Design a 20th-order Butterworth bandpass filter with a lower cutoff frequency of 500 Hz and a higher cutoff frequency of 560 Hz. Specify a sample rate of 1500 Hz. Use the state-space representation. Design an identical filter using designfilt.

```
[A,B,C,D] = butter(10,[500 560]/750);

 d = designfilt('bandpassiir','FilterOrder',20, 'HalfPowerFrequency1',500,'HalfPowerFrequency2',560, ...
'SampleRate',1500);
```

Convert the state-space representation to second-order sections. Visualize the frequency responses using fvtool.

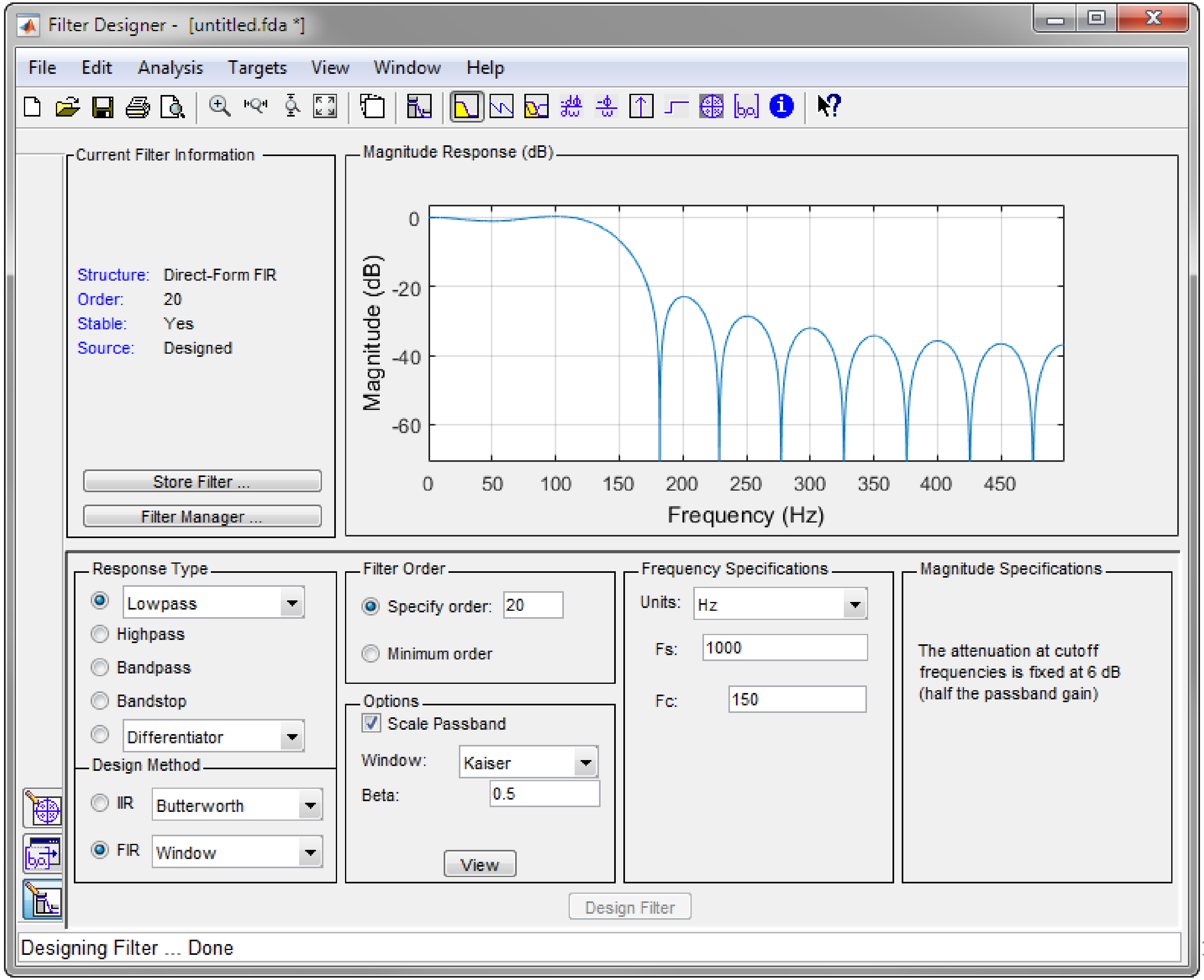
```
sos = ss2sos(A,B,C,D);
fvt = fvtool(sos,d,'Fs',1500);
legend(fvt,'butter','designfilt')
```



https://www.mathworks.com/help/signal/ref/butter.html

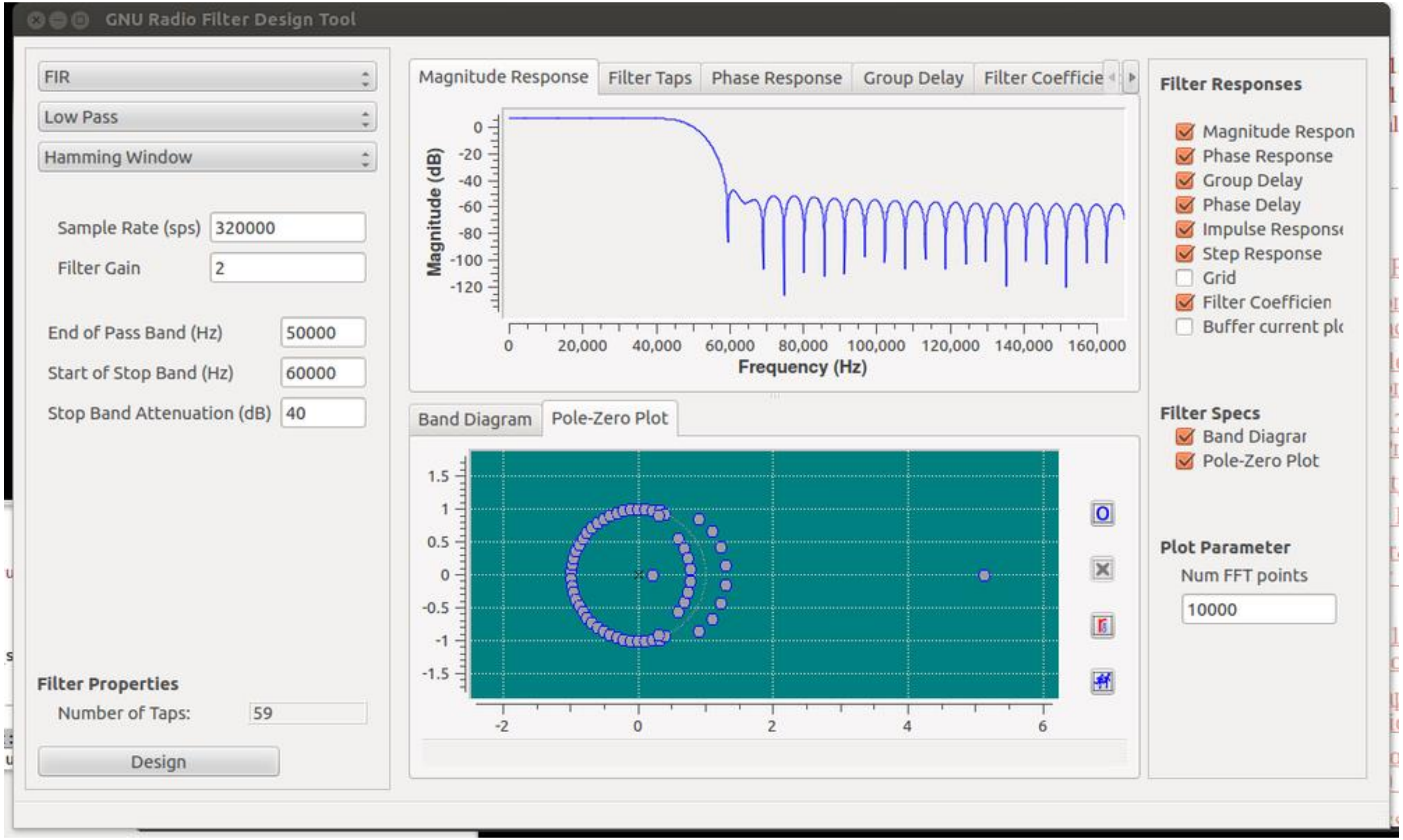
Matlab Filter Designer

https://www.mathworks.com/help/signal/ug/filtering-data-with-signal-processing-toolbox.html

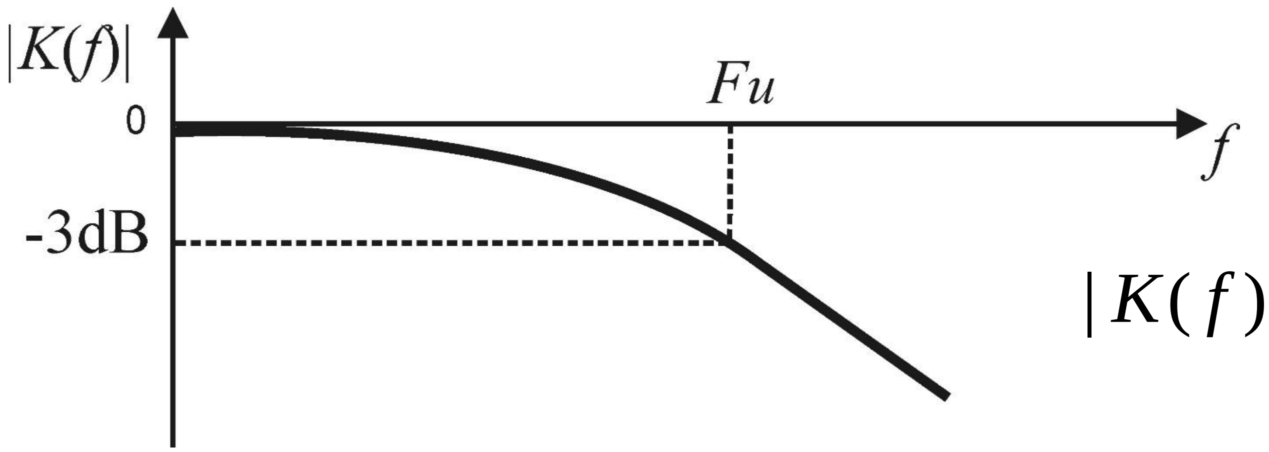


TAL TECH

IIR filters in GNURadio

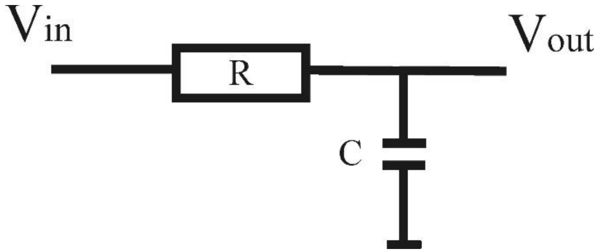


First-order low-pass IIR filter example



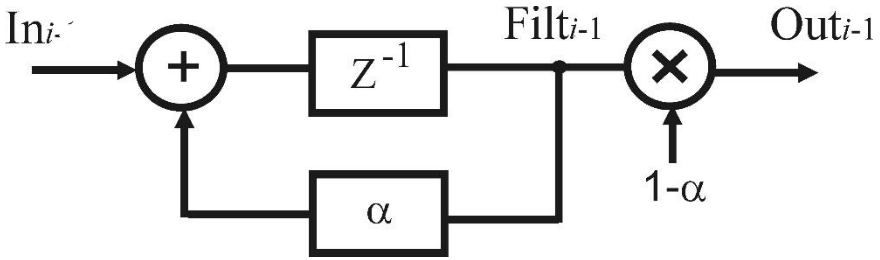
$$|K(f)| = \frac{1}{\sqrt{1 + (f / F_u)^2}}$$

Analog filter



$$F_u = \frac{1}{2\pi RC}$$

Digital filter



$$F_u = -\frac{Fd \ln \alpha}{2\pi}$$

$$0 < \alpha < 1$$