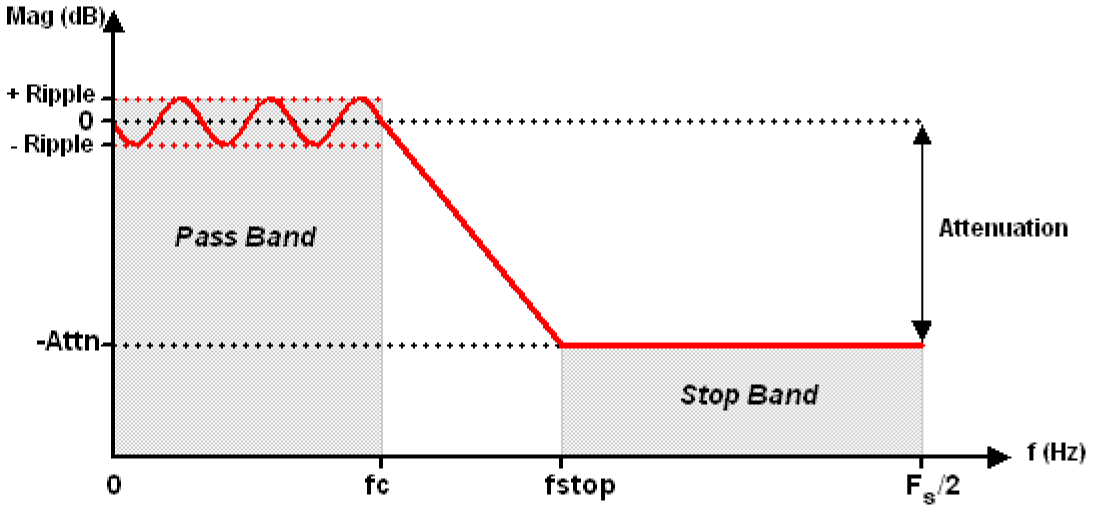
Filters and How to Create Them Using FDATool

In this article I will discuss what types of requirements go into filters, so that you can use MATLAB's **Filter Design Assistant (FDA) Tool** to specify and create filters to get the job done.

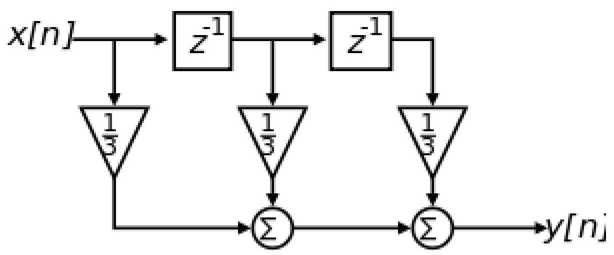
Filter Specifications



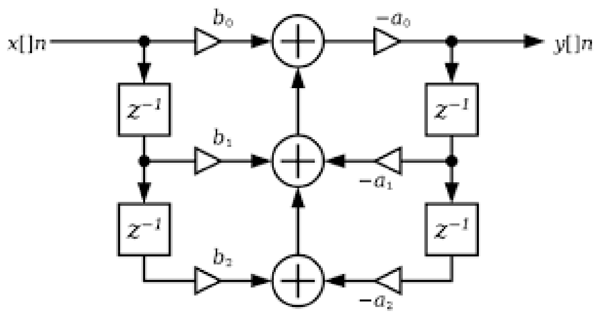
Filter Specifications diagram

<u>Passband:</u> This is the frequency range you do not want attenuated(gone). Your desired signal should be in this range. The highest or lowest frequencies in the passband are the cutoffs, f_c.
<u>Stopband:</u> This is the frequency range you want attenuated(gone) starting with f_{stop}.
The amount of attenuation, Attn allows you to set how much of the unwanted signal is left.
<u>Transition Band:</u> This is the frequency range between the passband and the stopband.
<u>±Ripple:</u> This is how much attenuation or gain is allowed in the passband. A large ripple can add predictable but large distortion to the signal.

Linear filters like we will design in FDATool are just weighted sums of current inputs and old inputs called **Finite Impulse Response (FIR) Filters** or they are weighted sums of current inputs, old inputs, and old outputs called **Infinite Impulse Response (IIR) Filters**.



2nd Order FIR Filter



2nd Order IIR Filter

The order of the filter is how many previous inputs or outputs the current output is a function of. This is the figure of merit for complexity in filter design. Each order comes with two pieces of storage, a coefficient and the previous value, and each order comes with two operations, multiplying the coefficient by the previous value and then adding it to the current output. For IIR filters there is twice this much complexity per order because the filter performs all these operations for both a previous input and a previous output.

Filter Goals

A filter is built to allow some frequencies to pass while stopping others, so several desirable qualities in a filter include:

- **High attenuation in the stopband:** -80dB is close to gone, but this can change with application
- **Small transition band:** Everything here is unwanted, but it is not being attenuated like the unwanted signals in the stopband
- **Low Ripple in the passband:** Distortion should be avoided on your desired signal
- **Minimal Order:** This gives a faster and less taxing filter to implement

FIR Vs IIR

IIR is usually a 10X increase in effectiveness towards specification goals over FIR in exchange for just a 2X complexity on a per order basis. The drawbacks of IIR filters is that they can become unstable (the output may start to oscillate) if designed improperly and they cannot have linear phase.

Linear Phase

Linear phase means all pieces of the signal are affected by the same delay moving through the filter across all frequencies. This is a desirable trait in FIR filters because it leaves the passband signal nearly untouched by the filter which is usually the intuition behind the filter in the first place. The only requirement for linear phase in an FIR filter is that its coefficients are symmetric around some delay. This delay becomes the delay for the filter itself.

$$Linear\ Phase\ Output\ Delay = \frac{Order}{2}$$

<u>Example</u> $y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + b_3 x[n-3] + b_4 x[n-4]$

y[n] is the current output

x[n] is the current input

x[n-m] is the previous input m samples before

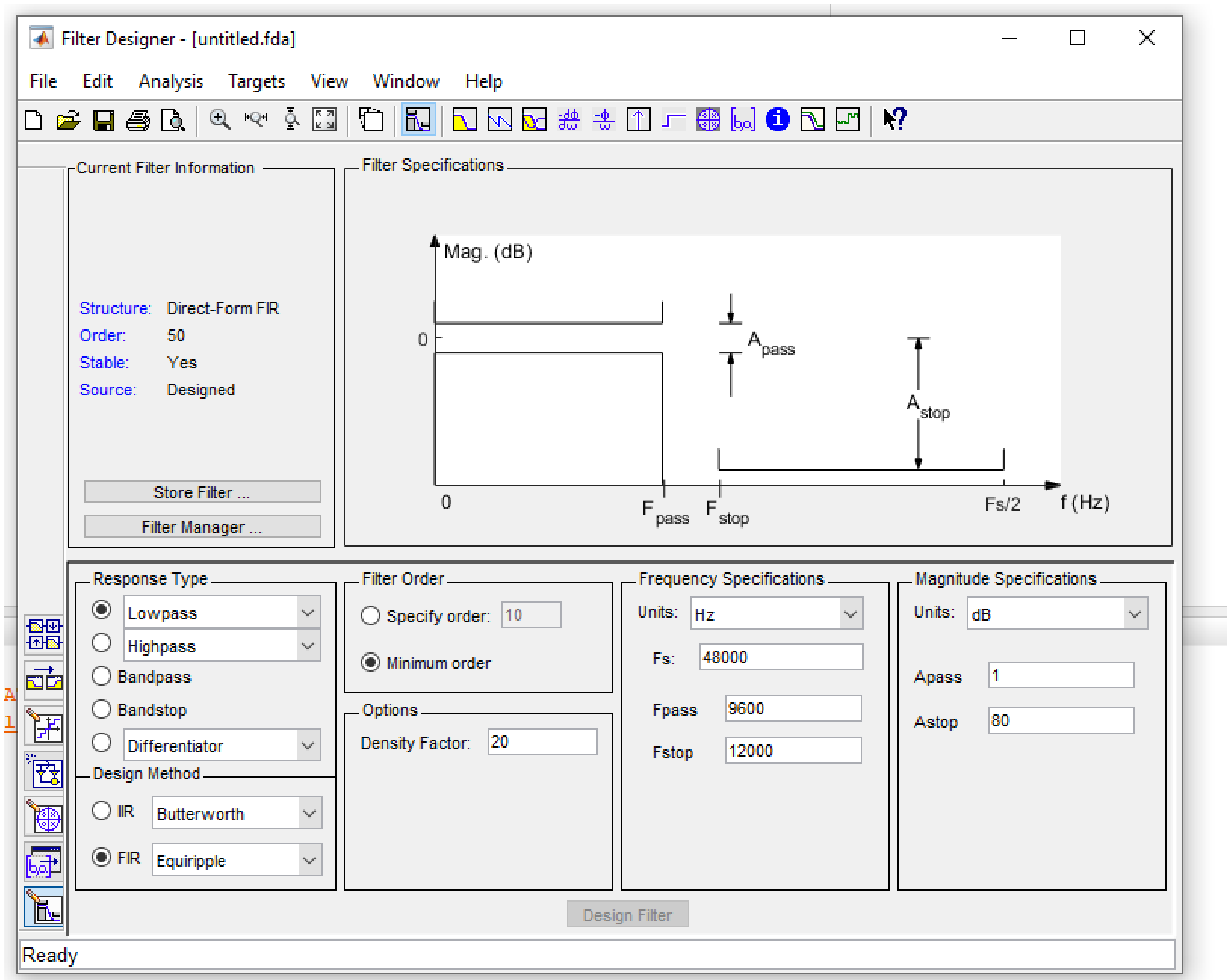
We refer to the filter itself by its transfer function below

$$H(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}$$

This filter is a 4th Order FIR, and it has linear phase if $b_0 = b_4$ and $b_1 = b_3$ leaving b_2 as the midpoint. The proper output of the filter starts after a delay of two samples, so the first two outputs would be thrown out before looking at the data or moving it to the next processing stage.

FDATool

Open Filter Designer
Open MATLAB → Type “fdatool” into the command window



Structure: Direct-Form FIR
Order: 50
Stable: Yes
Source: Designed

Response Type: [✓ Lowpass] Highpass Bandpass Bandstop Differentiator

Design Method: IIR Butterworth [✓ FIR Equiripple]

Filter Order: Specify order: 10 [✓ Minimum order]

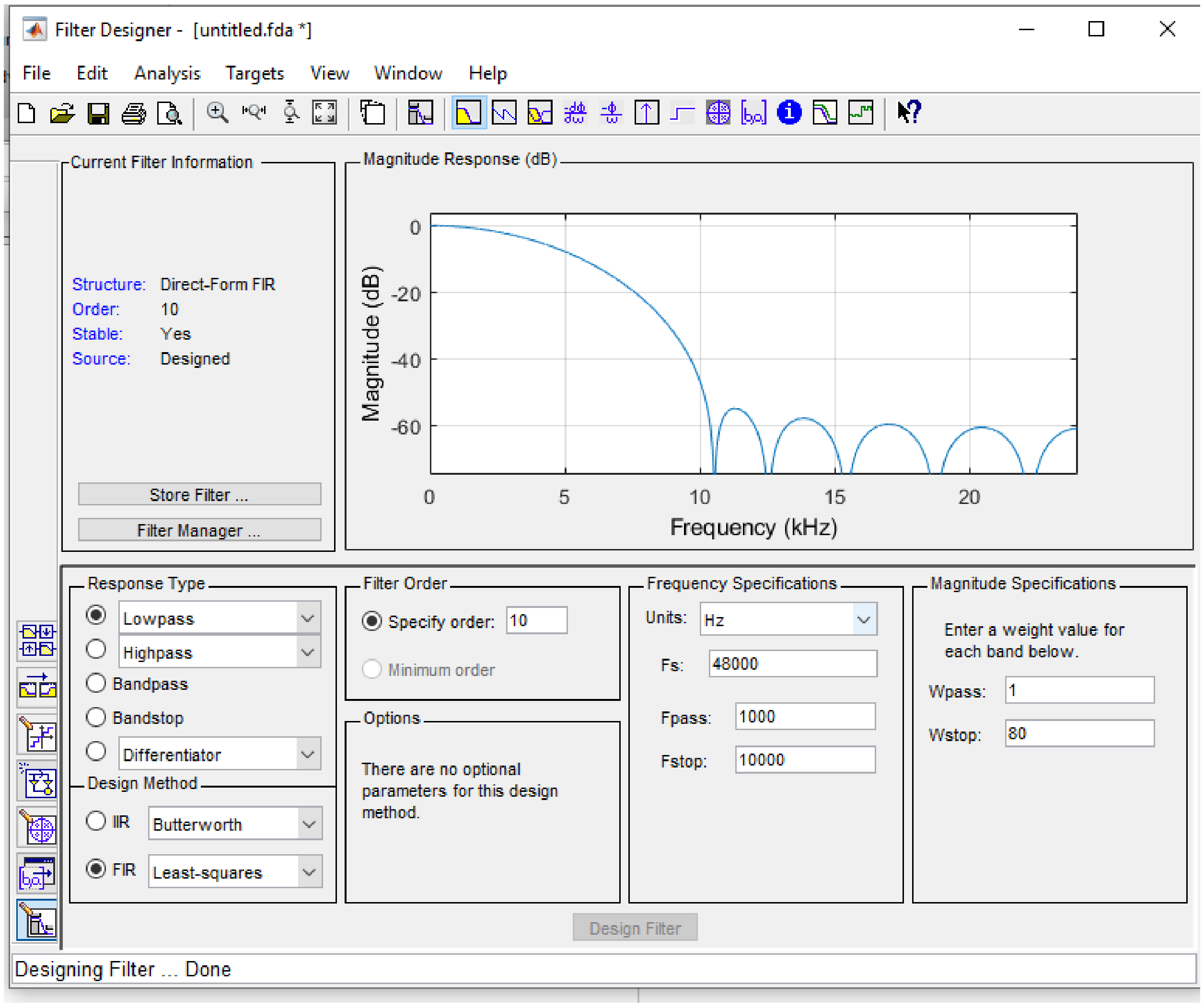
Options: Density Factor: 20

Frequency Specifications: Units: Hz; Fs: 48000; Fpass: 9600; Fstop: 12000

Magnitude Specifications: Units: dB; Apass: 1; Astop: 80

“Current Filter Information” tells you about the filter you have designed so far.
Everything in the lower half is the user specified/filled values of the filter.

<u>Example</u>
We wish to make a lowpass filter passing everything below 1kHz and stopping everything above 10kHz with 80dB of stopband attenuation.
In “ Response Type” select “Lowpass”
In “Design Method” drop down for FIR we will choose the “Least-squares” algorithm
In “Frequency Specifications” set “Fpass” to 1000 and “Fstop” to 10000
In “Magnitude Specifications” set “Wpass” to 1 and “Wstop” to 80


Filter Designer - [untitled.fda *]
File Edit Analysis Targets View Window Help
Store Filter ...
Filter Manager ...
Design Filter

Current Filter Information

Structure:	Direct-Form FIR
Order:	10
Stable:	Yes
Source:	Designed

Magnitude Response (dB)

Response Type

[✓ Lowpass]
Highpass
Bandpass
Bandstop
Differentiator

Design Method

IIR Butterworth
[✓ FIR] Least-squares

Filter Order

[✓ Specify order:] 10
Minimum order

Options

There are no optional parameters for this design method.

Frequency Specifications

Units:	Hz
Fs:	48000
Fpass:	1000
Fstop:	10000

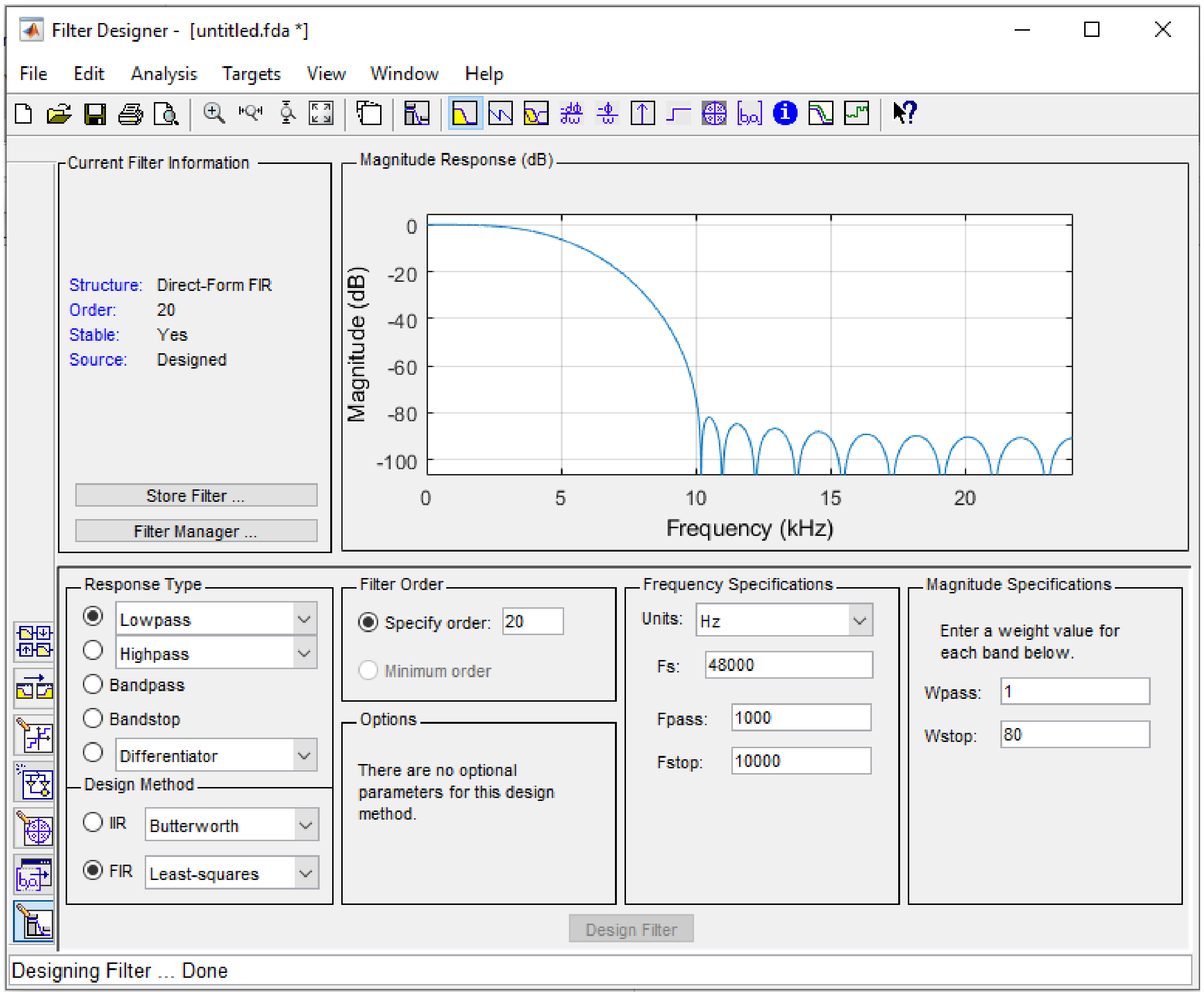
Magnitude Specifications

Enter a weight value for each band below.

Wpass:	1
Wstop:	80

Designing Filter ... Done

The filter just design is flat at 1kHz and attenuated after 10kHz, but it is not attenuated like we asked, so we can either drag the magnitude lower or specify a larger filter order in “Filter Order” such as 20.



Current Filter Information

Structure:	Direct-Form FIR
Order:	20
Stable:	Yes
Source:	Designed

Response Type: [✓ Lowpass] Highpass Bandpass Bandstop Differentiator

Design Method: IIR Butterworth [✓ FIR] Least-squares

Filter Order: [✓ Specify order: 20] Minimum order

Options: There are no optional parameters for this design method.

Frequency Specifications
Units: Hz
Fs: 48000
Fpass: 1000
Fstop: 10000

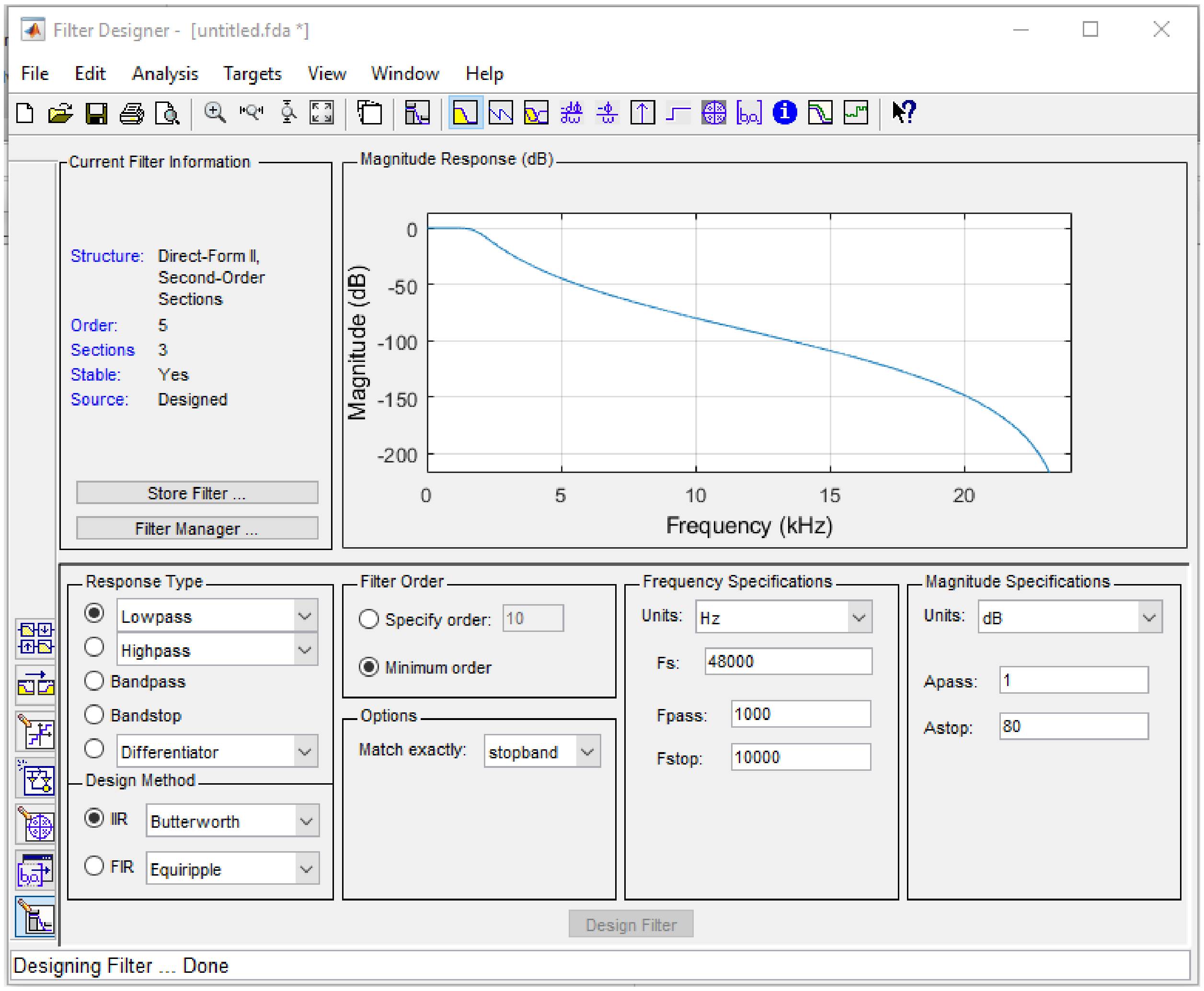
Magnitude Specifications
Enter a weight value for each band below.
Wpass: 1
Wstop: 80

Designing Filter ... Done

Changing the filter order gives a much better filter that meets all our specifications!

If the same specs where given for an IIR filter, we can change the design method to IIR→ Butterworth.

The specification that is left up to the system is **Sampling Frequency (Fs)**. This will determine the range of frequencies your filter is expecting, and it will scale Fpass and Fstop accordingly.



Current Filter Information

Structure: Direct-Form II, Second-Order Sections
Order: 5
Sections 3
Stable: Yes
Source: Designed

Response Type
[✓ Lowpass]
Highpass
Bandpass
Bandstop
Differentiator

Design Method
[✓ IIR] Butterworth
FIR Equiripple

Filter Order
Specify order: 10
[✓ Minimum order]

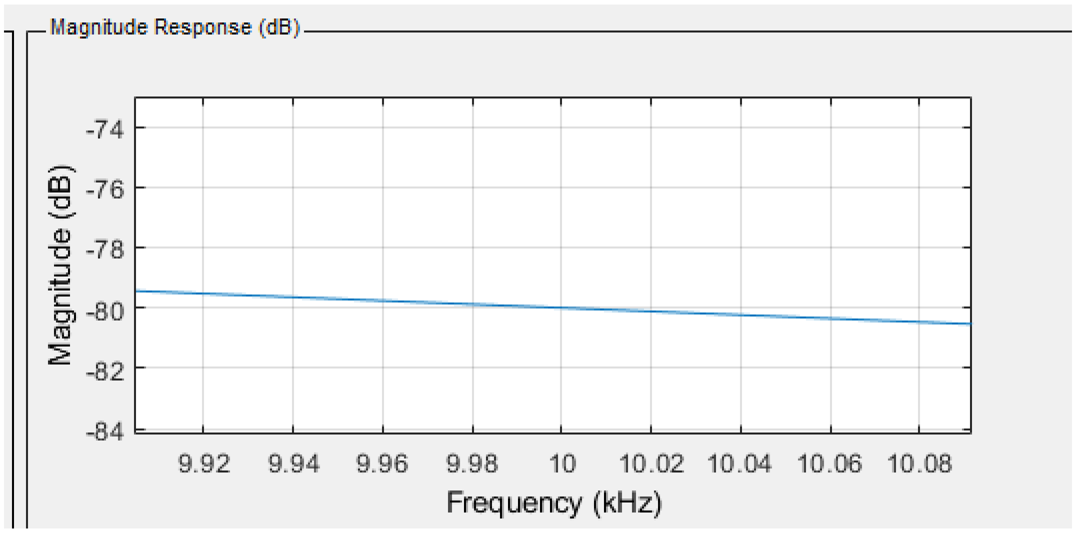
Options
Match exactly: stopband

Frequency Specifications
Units: Hz
Fs: 48000
Fpass: 1000
Fstop: 10000

Magnitude Specifications
Units: dB
Apass: 1
Astop: 80

Designing Filter ... Done

We can confirm the attenuation spec was met by zooming in on the magnitude response in the area of 10kHz.



Exporting Filter Coefficients to C Code

To store this filter’s coefficients for them to be used in a filter implemented in C use:
File → export → choose workspace naming the file Filter_Coefs →
run “fir_dump2c(‘coeff’,’B’,Num,length(Filter_Coefs));” in the command window

This will create a .c and a .h file including an array of coefficients. The filter itself can be implemented in a function that includes these documents.

Calling Filter Design from MATLAB Code

To call this design from MATLAB script to use in MATLAB filter implementation use:
File → Generate MATLAB Code → Filter Design Function

This will create an array that stores the filter coefficients in a logical way for indexing when you go to implement the filter.